On Low-Overhead and Stable Data Transmission between Channel-Hopping Cognitive Radios

Ching-Chan Wu, Shan-Hung Wu, and Wen-Tsuen Chen, *Fellow, IEEE*

Abstract—*Cognitive radios* (CRs) are proposed to alleviate the huge need for radio spectrum. There are two known steps for a pair of CRs to start communication: the *rendezvous* and *data-channel negotiation*. Despite that the rendezvous can be achieved by some well-studied techniques such as the channel hopping, the strategies for data-channel negotiation receive much less attention and their impact on data transmission performance remains unclear. In this paper, we study existing data-channel negotiation schemes for channel-hopping CRs and observe that 1) for short data transmission, they incur a huge overhead, called *notification delay*, that severely limits the throughput; and 2) for long data transmission, they lead to large overhead, called *interruption delay*, in handling the PU interruption, which makes performance unstable. By carefully re-examining the steps toward low-overhead and stable data transmission, we argue that a key step, called *self-channel selection*, is missing and should precede the rendezvous and data channel selection steps. In this step, each CR selects, in a distributed manner, only a small amount of the most stable channels to be used in the later steps. To realize the self-channel selection, we introduce a *Randomly-Started Stability-Descent* (RSSD) selection algorithm. Expensive simulations are conducted and the results demonstrate the effectiveness of RSSD in reducing 1) the notification delay; 2) the chance of PU interruption; and 3) the interruption delay if PU interruption occurs, which overall improve the performance and quality of data transmission.

Index Terms—Cognitive radios, channel hopping, data-channel negotiation, self-channel selection

◆

1 INTRODUCTION

The need for radio spectrum is huge due to the rapidly increasing number of wireless devices. Literature [1] shows that even in a crowded place such as a city, there are still spectrum holes across time and space. *Cognitive Radios* (CRs) allow the *Secondary Users* (SUs) to opportunistically utilize the *available channels*, i.e., channels that have been allocated to the *Primary Users* (PUs, or called *licensed users*) but are idle currently, to communicate with each other, thus mitigate the spectrum demand.

There are two known steps for a pair of CRs to start communication: the *rendezvous* and *data-channel negotiation*. By rendezvous we mean that two CRs will tune into some common available channels at the same time. One popular technique that guides CRs to rendezvous is *Channel Hopping* (CH) [2], [3], [4], [5]. A CH scheme, programmed in each CR in a network, divides the time of a radio evenly into *time slots*, and requires the radio to hop to a sequence of channels in some predefined order at consecutive slots. This sequence is called the *CH sequence* for that radio. The CH scheme ensures that, by following their CH sequences, two radios can rendezvous with each other within a finite delay called *rendezvous delay*, which amounts to $O(n^2)$, where n is the number of channels each CR can sense. The merit of channel hopping is that it is fully-distributed and allows different pairs of CRs to rendezvous at different channels, preventing congestion.

After rendezvous, two CRs need to agree on one or more common available channels and use these channels as the *data-channels* for data transmission. The data size may be either small (e.g., when transmitting text) or large (e.g., when transmitting multimedia) in the same application scenario. For example, in the battle field, some CRs may exchange their GPS coordinates (small) and some may relay enemy monitoring videos (large). Despite that the rendezvous problem has been widely studied in the literature [6], the strategies for data-channel negotiation receive much less attention and are usually introduced in an ad hoc manner following the rendezvous schemes. The impact of different data-channel negotiation schemes on the data transmission performance, both for small and large data size, remains unclear.

In this paper, we thoroughly review existing data-channel negotiation schemes for channel hopping CRs and give an in-depth study of their performance in data transmission. Generally, existing data-channel negotiation schemes for channel hopping CRs can be categorized into the *in-band* and *out-of-band* schemes. The out-of-band schemes assume there exist dedicated data-channels out of those channels used for rendezvous, while in-band schemes do not assume dedicated data-channels. We find that (1) it is not likely to assign dedicated data-channels for SUs globally as PUs can occupy the dedicated data-channels at any time in any specific region. So the in-band schemes are more realistic for a

---

- *C.-C. Wu and S.-H. Wu are with the Department of Computer Science, National Tsing Hua University, No. 101, Section 2, Kuang-Fu Road, Hsinchu 300, Taiwan.*
  E-mail: ccwu@netdb.cs.nthu.edu.tw, shwu@cs.nthu.edu.tw.
- *W.-T. Chen is with the Institute of Information Science, Academia Sinica, Nankang, Taipei 115, Taiwan, and the Department of Computer Science, National Tsing Hua University, Hsinchu 300, Taiwan.*
  E-mail: chenwt@iis.sinica.edu.tw.

Manuscript received 12 May 2016; revised 10 Sept. 2016; accepted 18 Oct. 2016. Date of publication 1 Nov. 2016; date of current version 2 Aug. 2017.
For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org, and reference the Digital Object Identifier below.
Digital Object Identifier no. 10.1109/TMC.2016.2623715

distributed environment. (2) However, when transmitting small data,[1] the in-band schemes incur delay overhead called the *notification delay*, which amounts to $O(n)$ slots and is usually much longer than the time actually spent in data transmission. This is because that for a pair of channel hopping CRs, the sender need to wait for the receiver to hop to some common channel whenever it wants to begin a new data transmission. The throughput is severely limited. (3) Furthermore, when transmitting big data, the in-band schemes give unstable performance because a longer data transmission time makes the transmission itself vulnerable to the sudden data-channel unavailability due to PUs (called the *PU interruption*). When PU interruption occurs, the sender and receiver need to switch to another common channel to resume the transmission, causing the *interruption delay*, which amounts to the notification delay if that common channel has already been identified in the previous rendezvous; otherwise rendezvous relay (so the sender and receiver can discover new common channel via rendezvous). This unstable performance make CRs hard to support jitter-sensitive applications (e.g., voice/video streams). How to ensure the low overhead, stable data transmission between channel hopping CRs remains an important but open problem.

Instead of proposing a new data-channel negotiation scheme, we argue that a key step is missing: the *self-channel selection*, where each CR i individually selects a small set V_i, $|V_i| \ll n$, of the most stable[2] channels first and only use these channels for rendezvous and data transmission. Since each receiver i hops among $|V_i|$ channels rather than n during channel hopping, the rendezvous delay can be reduced from $O(n^2)$ to $O(|V_i|^2)$ and the notification delay can be reduced from $O(n)$ to $O(|V_i|)$. Given that $|V_i| \ll n$, these both increase the throughput of short data transmission and reduce the interruption delay during long data transmission. Furthermore, because channels used for data transmissions are stable, the PU interruption problem can be alleviated.

Self-channel selection brings new challenges. To preserve connectivity, self-channel selection on a CR needs to ensure that the selected channels will overlap those of another if these two CRs sense any available channel in common (so they can still discover each other in the rendezvous step). Furthermore, the overlaps of the selected channels of different CR pairs should spread out different channels to avoid congestion in data-transmission. Guaranteeing the above is challenging because different CRs sense different available channels and there's no way for one to know the others' available channels before rendezvous. We introduce a *Randomly-Started Stability-Descent* selection algorithm that (1) guarantees the overlap and spread-out, and (2) allows the number of selected channels $|V_i|$ to be as low as $O(\sqrt{n})$ in most situations (3) while making sure that all selected channels have stability above a threshold and are as high as possible.

To the best of our knowledge, this is the first in-depth study on different data transmission techniques between the channel hopping CRs. Following summarizes the contributions of this paper:

- We compare four existing data-channel negotiation schemes for channel hopping CRs through qualitative discussion and quantitative measurement of their performance.
- Based on the results, we observe that current in-band schemes, although fully-distributed, incur a huge overhead (specifically, notification delay) for short data transmissions and the instability for long data transmissions (specifically, occasional interruption delay).
- We introduce the notion of self-channel selection preceding the neighbor discovery and/or data-channel negotiation to walk around the above problems.
- We propose the RSSD algorithm to realize the self-channel selection. We also discuss the practical usage of RSSD and its optimizations. Specifically, we propose a unified method to integrate RSSD with most existing CH rendezvous schemes (for in-band data transmission).
- We conduct extensive simulations and the results show that RSSD achieves 50 percent of delay reduction for small data transmission and can avoid at most 70 percent of interruptions for large data transmission.

The rest of this paper is organized as follows. Section 2 identifies the steps required for CRs to communicate with each other. Section 3 compares performance of existing data-channel negotiation schemes and motivates the self-channel selection. Section 4.1 introduces RSSD. Section 6 evaluates the performance of RSSD and Section 7 reviews related work. Finally, Section 8 concludes the paper.

2 PRELIMINARIES

Next, we briefly describe the practical conditions of the CR environment and review existing CH schemes for rendezvous in such environment, then discuss the notification and re-transmission mechanism in details. Then we review and summarize existing data-channel negotiation schemes.

2.1 The CR Environment

In practice, studies [10], [11] show that almost every geographical area has several unused channels, and the occupied channels D_i and D_j sensed by nearby users are usually similar but not the same. In addition, some patterns of spectrum usage have been observed in the literature.

Condition 1. The usage of many bands, including PLM, fixed/maritime mobile, Paging, SMS, FMS, GPS, satellite, MDS, ITFS, surveillance radar, and TV aux, etc., is usually lower than 10 percent [11], [12].

Condition 2. Channels in the mid- to low-usage bands are often used sporadically (i.e., sparsely and geographically divergently) [1], [10].

Condition 3. In the same region, spectrum holes in some channels last very long while some appear for a very short duration[13].

Based on the above condition, CRs can utilize the low-usage channels at different time and space, and should use the stable channels with long vacancy time to avoid frequent PU interruptions.

---

[1] Specifically, data which require less than a single slot to transmit.

[2] We say that a channel is stable if its availability does not change frequently due to PU activities [7], [8], [9].

2.2 CH Schemes for Rendezvous

Assume that the spectrum sensible to CRs can be divided into a set $U = \{0, 1, 2, \ldots\}$ of channels. Each channel in U is either occupied by nearby Primary Users (PUs) or idle for opportunistic usage for Secondary Users (SUs, or simply CRs), and we let D_i be the set of PU occupied channels that CR i detects. The time of each CR i is divided evenly into *time slots*, denoted as $t_i^{[0]}, t_i^{[1]}, \ldots$. We do *not* assume any timer synchronization between CRs. So given an index x, slots $t_i^{[x]}$ and $t_j^{[x]}$ of two CRs i and j may have arbitrary shift in time. We say that two slots $t_i^{[x]}$ and $t_j^{[y]}$ have *time-overlap* if they overlap for an interval longer than half of a slot, and the slot is set long enough such that the handshaking can be done [2], [3], [6] during the interval.

A CH scheme lets each CR hop to a channel at each time slot and wait for rendezvous with other CRs. Specifically, it generates a Channel Hopping sequence (CH sequence) $S_i = [s_i^{[0]}, s_i^{[1]}, \ldots]$, where $s_i^{[x]} \in U \backslash D_i$, for each CR i such that i hops to channel $s_i^{[0]}$ at slot $t_i^{[0]}$, and $s_i^{[1]}$ at slot $t_i^{[1]}$, and so on.

Definition 1 (Rendezvous). *Given a pair of CRs i and j in a network, the CRs i and j rendezvous if $s_i^{[x]} = s_j^{[y]} = c$ for some x, y, and c, where $t_i^{[x]}$ time-overlaps with $t_j^{[y]}$ and c is in both $U \backslash D_i$ and $U \backslash D_j$.*

Two CRs are said to *rendezvous* if they hop to some common idle channel at a pair of time-overlapping slots. Note that it is impossible for two CRs to rendezvous if they have no idle channels in common, i.e., $(U \backslash D_i) \cap (U \backslash D_j) = \emptyset$.

The Maximum Time To Rendezvous (MTTR) between any CR pairs i and j satisfying $(U \backslash D_i) \cap (U \backslash D_j) = \emptyset$ is an important metric to evaluate a CH scheme. The shorter MTTR the better. Traditional CH schemes [14], [15] give $O(n^2)$ MTTR, where n = |U|. Recent studies [5], [16] propose the short-cycledCH schemes that guarantee $O(|V_i||V_j|)$ MTTR if S_i is defined over some channel set $|V_i| \subseteq |U \backslash D_i|$(i.e., $s_i^{[x]} \in V_i$ for any x), S_j is defined over some $|V_j| \subseteq |U \backslash D_j|$, and $|V_i| \cap |V_j| \neq \emptyset$.

Generally, at rendezvous, two CRs can exchange their (1) channel availabilities so to identify common available channels; and (2) hopping schedules (e.g., CH sequences) for future communication. The channels at rendezvous of different CR pairs are expected to evenly spread among U to avoid congestion. The CH schemes that guarantee this is called *well-formed*.

2.3 Data-Channel Negotiation After Rendezvous

In some practices of radio communications with center control, one may divide the radio spectrum into the dedicated control-channels (for rendezvous and CH schedule exchange) and data-channels. We call a data-channel negotiation scheme as an *Out-of-Band* (OB) scheme if it uses dedicated data-channels out of control channels. For ad hoc CRs (where the channel hopping targets to), OB is not a suitable scheme because without center control, the dedicated data-channel may easily become unavailable due to PU interruptions or SU collisions..

Thus, it is better for ad hoc CRs to employ a scheme that can pick any of the common available channels between a CR pair as a data-channel, including those used for rendezvous. This kind of schemes are called the *In-Band* (IB) schemes. Data-channels in these schemes can be selected based on some criteria (such as bandwidth, channel capacity, and channel state information, etc) indicating the quality of data transmission. A straightforward IB scheme is to choose the channel that gives the best quality of data transmission. However, this scheme may incur serious congestion on that "best channel."

To avoid the concentration of data transmission, a database for querying the current channel usage or a dispatcher distributing transmissions to different channels in a round robin fashion would have worked well [17], [18]. However, these methods requires centralized infrastructures thus cannot apply to ad hoc CRs. Another way is to let CRs reach an agreement either by pooling from or cooperating with neighboring CRs for every transmission [19], [20]. But the agreement may take extremely long time to reach because each pooling/cooperation needs to rendezvous with neighbors first and the rendezvous delay is generally long ($O(n^2)$ slots in the worse case). Moreover, the delayed agreement cannot represent the current channel condition.

From the above, we need a distributed data-channel negotiation scheme that is able to select data-channels in a short time and to spread the transmission evenly to available channels. Recall that a well-formed CH scheme can scatter rendezvous evenly to different channels. Accordingly, CR can leverage this nature of CH scheme right after rendezvous to spread the data transmissions uniformly to different channels. Here, we summarize three data-channel negotiation schemes that work in conjunction with a well-formed CH scheme. Different from rendezvous schemes, the data-channel negotiation schemes can safely assume the role (i.e., sender or receiver) of each CR in a pair, by virtue of the network or application protocols. And we say a data transmission is a *multi-slot transmission* if it spends multiple time slots; otherwise a *single-slot transmission*.

Depending on at which slot the receiver will be notified and what channels to be selected as data-channels, the schemes can be categorized as follows:

In Rendezvous (IR) [21], [22]. Once the data arrives at the sender CR, IR relies on the next rendezvous slot to notify the receiver and start transmission. IR does not change CRs' hopping schedule (i.e., CH sequences) so that a pair of CRs transmit data only in the rendezvous slot. If the transmission time is longer than a time slot, the CR pair will stop the transmission at the end of the slot and resume at the next rendezvous slot. A re-transmission caused by either PU interruption or SU collision also needs to wait for the next rendezvous slot to resume.

Sticky In-band (SIB) [7], [8], [9]**.** Unlike IR, SIB allows the sender CRs to change its hopping schedule for transmission. When the receiver hops to a common available channel, the sender (knowing the receiver's schedule after rendezvous) can actively hop to the same channel at the same time and use that channel as the data-channel. After that, the sender and receiver will stick in this data-channel until the data transmission completes. When encountering a interrupted transmission, the CR pair hops to another common available channel if it exists, otherwise they try to re-rendezvous.

Following In-band (FIB) [3], [23]**.** Like SIB, FIB also allows the sender to change its hopping schedule. The sender waits for the receiver to hop to a common available channel and

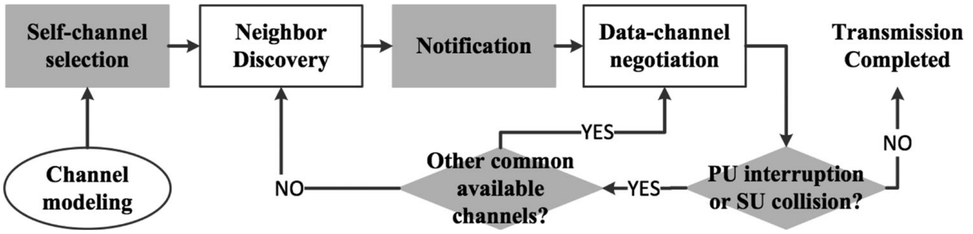


Fig. 1. Steps for CRs to begin data transmission (the shaded steps are currently missing). We propose an optional step called "self-channel selection."

notifies the receiver to start transmission using that channel. The difference is that when the transmission time is longer than a time slot, the sender will follow the schedule of the receiver and hop to the next channel in the receiver's CH sequence in the next slot to continue data transmission. When the current channel in the receiver's schedule is not available for the sender, they will pause transmission during the slot and resume thereafter. FIB employs the same re-transmission procedures as SIB.

The performance of the above data-channel negotiation schemes has not been well-studied.

3 SELF-CHANNEL SELECTION

In this section, we conduct simulations to investigate the performance of existing data-channel negotiation schemes. Our observations motivate a strong need for the self-channel selection step preceding the rendezvous. We then formally define the self-channel selection problem.

3.1 Communication Procedure

The procedure of data transmission for ad hoc CRs are not clearly defined in existing work. Currently, neighbor discovery and data-channel negotiation are two known general steps required at the MAC layer for a pair of CRs to start communication, as shown in Fig. 1. At the very beginning, each CR needs to discover their neighbor CRs by rendezvous. Depending on what information is exchanged at the rendezvous time, two CRs may choose data-channels following different ways characterized by different data-channel negotiation schemes. In addition to the above steps, we identify two parts that are currently not clearly defined, namely the *notification* procedure and *re-negotiation* mechanism of data-channel.

Notification. After rendezvous, there's an often-neglected notification procedure between the sender and receiver. By knowing the schedules (CH sequence) of other CRs, a sender CR can initiate a data transmission as follows. Upon data arrival, the sender first buffers the data and wait for the receiver to hop to a common channel at some coming slot. At that slot, the sender switches to the common channel and sends a notification to the receiver. After this notification procedure, the sender and receiver can negotiate, based on some data-channel negotiation scheme, one or more data-channels/slots to begin data transmission.

Re-Negotiation of Data-Channel. The re-negotiation mechanism is another important but often-neglected part. Since in a CR network, a data-channel may suddenly become unavailable, either due to the occurrence of PUs (called *PU interruption*) or other SUs (called *SU collision*). The data transmission may be interrupted and CR's need to find out another common available channel to resume the transmission. If there was another common channel identified in the previous rendezvous, the CR pair can resume the transmission on that common available channel after a new notification procedure. Otherwise they need to re-rendezvous. The detail re-transmission flow is depicted in the lower right of Fig. 1.

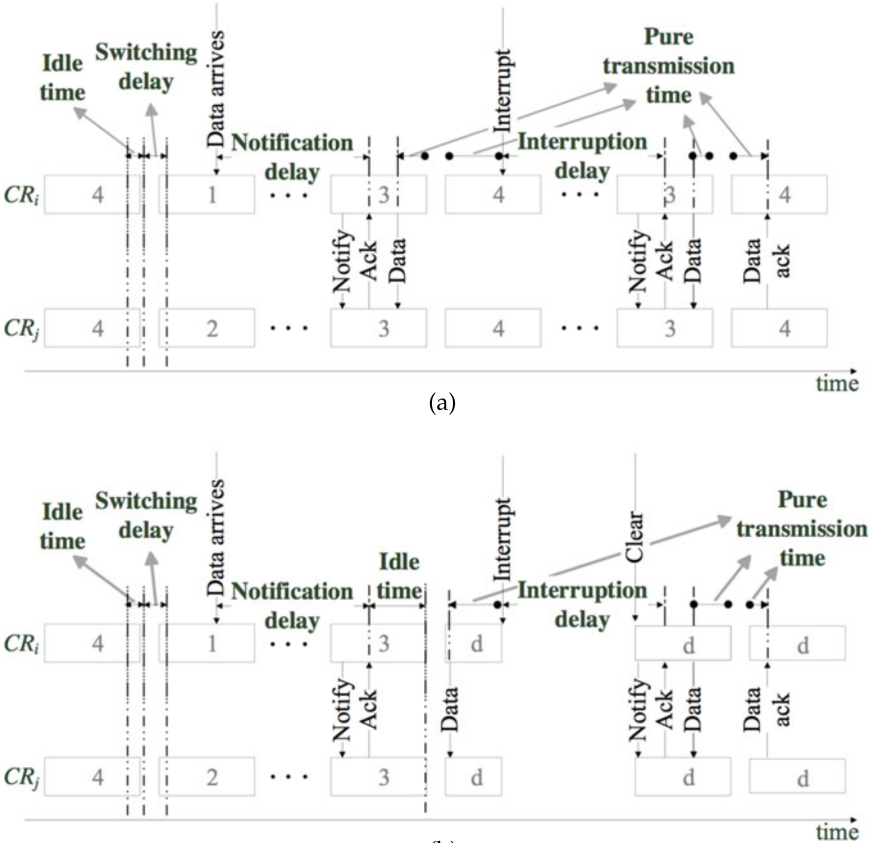
The impact of the above steps on data transmission performance is not well studied so far.

3.2 Experimental Observations

Next, we identify the limitations of the current communication procedure. We follow the conditions shown in Section 2.1 to simulate the CR environment, and the detailed settings are described in Section 6.

Metrics. One of the most direct and important metrics to measure the performance of a data transmission is its delay, as many other metrics such as throughput depend on it. In the following, we categorize and list the delays happening in different communication steps that prolong the data transmission time:

- Switching delay: The time spends on hopping to a new channel and sensing the availability of the new channel, which usually happens when requiring a re-transmission on another common available channel or hopping for avoiding congestion. We separate all switching delays from other delays. IR may waste most time on hopping and FIB also spends non-negligible time for it.
- Notification delay: The interval from the data arrives at the sender to the time the receiver is notified and reaches a consensus on data-channels with the sender. In OB, SIB, and FIB, the sender with data needs to wait for the receiver to hop to a common available channel for notifying it, which last for $O(n)$ slots. For IR, the notification delay may be much higher while the sender needs to wait for the next rendezvous to notify the receiver, where the $O(n^2)$ MTTR is usually several seconds to minutes [3], [4], [21].
- Pure transmission time: The time needed for transmitting data under specified data rate without any interruption, which is independent to data-channel negotiation schemes.
- Interruption delay: The total time needed before re-transmissions can start after the current transmission is interrupted. A data transmission may involve multiple re-transmissions to complete, so the interruption delay may consist of multiple times of the



(a)

(b)

Fig. 2. (a) A case of delay timeline for a pair of CRs using IR. (b) A case of delay timeline for a pair of CRs using OB.

  delay for re-seeking common available channels and/or re-rendezvous.

- Idle & waiting: The time that a CR remains idle during a data transmission or rendezvous, e.g., the random back-off time after collision and the hardware waiting time for channel switching.

To demonstrate the time these delays happen in a communication, we show a case of a data-channel negotiation between two CRs with a small channel set. Given two CRs i and j, $V_i = [1, 3, 4]$ and $V_j = [2, 3, 4]$, the timeline of delays is depicted in Fig. 2. In the figure, a box represents a slot and the number inside the box is the channel that the CR is tuned in. Note that, the time scales in Fig. 2 are adjusted for clarity and may not be true in practice.

The delay timeline of IR is in Fig. 2a. The sender i is going to send the data to the receiver j, and the notification delay starts when the data arrives at i, and it ends when j confirms the notification on a common available channel, channel 3 in this case. The switching delay happens every time when i and j hop to another channel. Interruption delay takes place when some PUs or other CRs interfere with the data-channel, channel 4 in this case i and j are using, and it ends when i meets and re-notifies j on another common available channel, channel 3. The pure transmission time is the sum of the transmission time spent in different slots.

For SIB and FIB, their timelines are very similar to the timeline of IR, so we only describe the differences that only exist after the notification. For CRs using SIB, the sender and receiver stick in the channel they met until the data transmission completes. For CRs using FIB, the sender transmits data in successive channels in the receiver's hopping schedule after the notification.

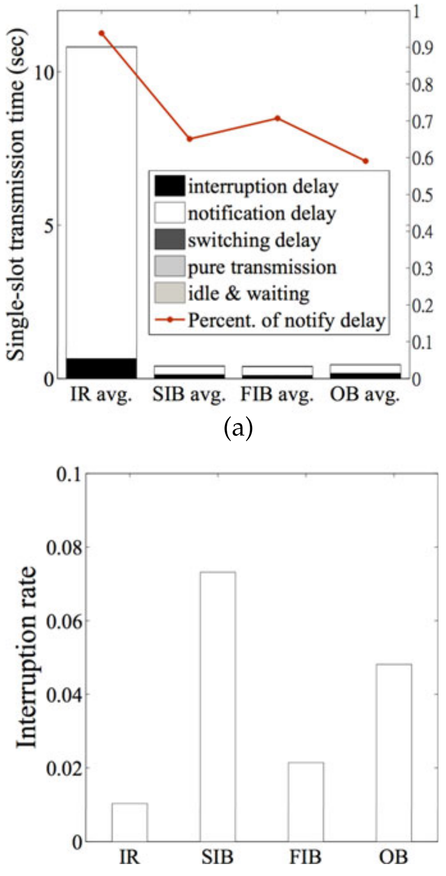
The delay timeline of OB is depicted in Fig. 2b. After the notification, i and j hop to the predefined data-channel d and start the data transmission. When the interruption comes, i and j wait for channel d to be clear of interference again, and then they restart the data transmission after a re-notification.

Note that, the switching delay is not counted in the other delays, and the reason is to better demonstrate that the main overhead of a data-channel negotiation is not the channel hopping, which is the inherent feature of channel-hopping CRs, and can be further reduced.

Next, we run the existing data-channel negotiation schemes in a simulated PU-dynamic environment under the Condition 3 and study the delay performance of short and long transmissions respectively. Detailed settings about the simulation environment are described in Section. 6. Following summarizes our findings:

One-Slot Transmissions Suffer from High Overhead. Fig. 3a shows the breakdown of average data transmission time for transmitting small data. We can see significant overheads than the actual/pure data transmission. Specifically, the notification delay accounts for at least 50 percent of total transmission time for all selection schemes. Another major overhead is the interruption delay, which takes 10~40 percent of the total transmission time.

Multi-Slot Transmissions are Unstable. For large data, Fig. 3b shows the ratio of interrupted transmissions. Although the interruption rate seems not large, it seriously



(a)

(b)

Fig. 3. (a) Single-slot transmission time of the four data-channel negotiation schemes. (b) Interruption rate of all transmissions for multi-slot transmission.

degrades the worst 5 percent of transmissions. Fig. 4 shows the breakdown of data transmission time for the 4 selection schemes. Two kinds of data transmission time are studied: one averaged over all transmissions and another averaged over those with longest 5 percent of delay. By comparing the two, we see that the worst 5 percent transmissions take above 25 percent extra time than the average. This is mainly resulted from the interrupt delay, showing that the long transmissions are vulnerable to PU dynamic. In addition, the idle & waiting time increases relatively large in the worst 5 percent. This reflects the extra time spent in waiting for random back-off on a congested channel. The long transmissions are also sensitive to SU interference/collisions.

Note that the above delay measurements do not suggest which data-channel negotiation scheme is the best. For example, although OB's delay performance seems good, it is very sensitive to the PU interruption and SU collisions—unavailability of the dedicated data-channels lead to an impasse, implying infinite delay and zero throughput. On the other hand, in spite of its poor delay performance, IR may be a good candidate if each sender CR knows what to transmit right upon or before rendezvous and the data are generally small.

3.3 The Need for Self-Channel Selection

As CRs are free to send both small and large data, the above practical findings point out three challenges that need be addressed to improve the data transmission, namely, (1) to shorten the notification and interruption delay, (2) to reduce the chance of interruption, and (3) to avoid congestion to reduce long idle & waiting time for large data.

Instead of trying to improve the data-channel negotiation schemes on-by-one or to propose a new scheme, we identify a step, called the *self-channel selection*, preceding the rendezvous (as shown in Fig. 1) that can benefit all existing data-channel negotiation schemes by addressing the above three challenges simultaneously. Before formally defining this step, we take a quick detour to introduce the concept of channel stability.

The channel stability varies largely among channels under the Condition 3. Thus, the channel stability should reflect the ability and degree that a channel can stay available without PU and SU interferences. Here, we define the stability of a channel as its continuous and available interval started from the current time to the future. Although we cannot know the available time in the future, however, by analyzing past activities of PUs using existing channel modeling techniques [7], [8], [9], we can predict with high confidence whether a channel will be occupied by PUs in a short time. We formally define the channel stability as follows:

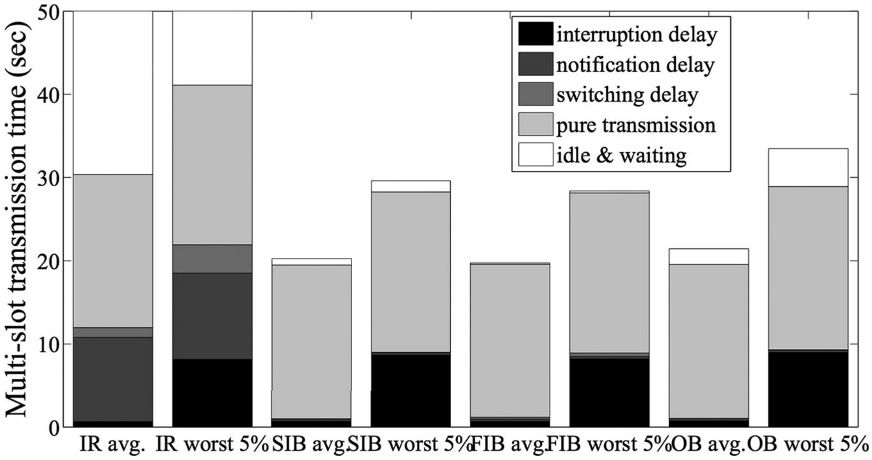


Fig. 4. Multi-slot transmission time of the four data-channel negotiation schemes (the idle and waiting time for IR avg. and IR worst 5 percent are 1,480 and 2,745 seconds, respectively).

Definition 2 (Channel Stability). *Given two channel states as available and unavailable, if a channel is in a state a_0 at a time t_1 and keep in the state until a time t_2, where $t_2 > t_1$, then the stability at t_1 for the channel is defined as $(-1)^i(t_2 - t_1)$, where $i = 0$ if a_0 is the available state, otherwise $i = 1$.*

In a channel stability model, we let t_1 be the current time and predict the changing time of the available state t_2. Thus, a channel will have a positive stability if it is currently available; otherwise negative. There are two common ways to establish the channel stability model based on either centralized or distributed historical channel statistics respectively. In this paper, we assume each CR has a distributed stability model and we define the model L_i of a CR i as a list of channel stability for all channels that i sensed and predicted.

Now we are able to formally define the self-channel selection. Assume that $U = \{0, 1, 2, \ldots\}$, $|U| = n$, is the set of channels sensible to CRs. Let $D_i \subseteq U$ be the set of non-idle channels that CR i perceives. And each CR i has a channel stability model L_i, which provides a sorted list of stable channels that more stable ones in the top of L_i.

Problem 3 (Self-Channel Selection). *On each CR i, select a set of available channels $V_i \subset U \backslash D_i$ such that (a) $|V_i|$ is as small as possible; (b) channels in V_i are in the top of L_i while ensuring that (c) for any pair of CRs i and j, where $(U \backslash D_i) \cap (U \backslash D_j) \neq \emptyset$ so that i and j are able to rendezvous, we have $V_i \cap V_j \neq \emptyset$ so that i and j can still rendezvous by hopping among their respectively self-selected channels following the CH sequences given by a short-cycled CH scheme; and that (d) the overlaps of the selected channels of different CR pairs spread out the entire U to avoid congestion in data transmission.*

The criterion (a) shortens the notification and interruption delay because with a short-cycled CH scheme, the rendezvous delay can be reduced from $O(n^2)$ to $O(|V_i|^2)$ and the notification delay can be reduced from $O(n)$ to $O(|V_i|)$ in SIB and FIB. The criterion (b) reduces the chance of interruption under Condition 3 since channels used for data transmissions are now stable. The criterion (d) reduces long idle & waiting time for large data, and the criterion (c) preserves the network connectivity. Next, we describe our idea to cope with practical conditions mentioned in Section 2.1.

4 THE *RSSD* SELECTION SCHEME

To solve Problem 3, we propose the Randomly-Started Stability-Descent scheme. Although the conditions listed in Section 2.1 implies that communications among CRs are very unstable, we leverage the characteristics of these conditions to design a communication stabilization scheme for these CRs. The intuition of the selection scheme is that we make each CR select relatively stable channels to avoid interruption (by Condition 3) under following rules: we make the CR select a long string of consecutive channels where those channels are under relatively low usage for the CR (by Condition 1), and then for those channels are sporadically PU occupied, we make the CR select some channels that at least one of them overlaps some channel in the selected consecutive channel string of all other CRs. Therefore, CRs can use fewer but more stable channels to communicate and largely reduce the notification and interruption delays. We design our algorithm based on the sporadic channel utilization characteristic under Condition 2, and the detailed algorithm is stated below.

4.1 Algorithm

There are naive methods for self-channel selection. For example, CR pairs can randomly select a common available channel to satisfy criterion (c) but it may be vulnerable to PU interruption, missing criterion (b); or CR pairs can select the most stable channel to satisfy criterion (b), but it may lead to serious congestion since the most stable channel may be the same for nearby CRs, missing criterion (d). One may even model the $|V_i|$'s as quorums returned by a quorum system [3], [24] to satisfy criteria (a) and (d) simultaneously. However, existing construction schemes for quorum systems creates strict rules on placing which channels to which quorums (in order to guarantee the overlaps between quorums). Here, the presence of $\{D_i\}_i$ imposes rules on *not* to place certain channels to certain quorums, which can easily violate the construction rules and prevent the construction schemes from satisfying criterion (c).

We present the Randomly-Started Stability-Descent algorithm that can be run on individual CRs and allow each CR to obtain its V_i in a distributed manner.

The basic ideas of RSSD are to (1) start from a high threshold that selects only few most stable channels, and try to return V_i. If no V_i can be found, then decrease the threshold to consider less stable channels and repeat until V_i is found; and (2) when finding V_i at a particular threshold level, start from the random channel r in U and try to build V_i by inspecting channels in $r+1$, $r+2$, and so on while skipping the unavailable channels in D_i. The steps (1) and (2) are called the *threshold descent* and *random start* steps respectively.

The detailed steps of RSSD are described in Algorithm 1. Here, we say that two channels is the continuous elements as their channel indices differ by 1. Each CR i first determines the stability threshold as the kth stablest channel stability, where k is initialized as the smallest quorum size $2\lfloor\sqrt{n}\rfloor$ [24], for selecting stable channels. It then finds the $c_{\max}$, which is the length of the longest continuous elements in D_i. If $c_{\max} > \lfloor\sqrt{n}\rfloor - 1$, starting from a randomly selected number r, the CR i seeks for the smallest set H_i of continuous elements whose stability is larger than or equal to the threshold and having size larger than $c_{\max}$. Then the CR i finds a set T_i of elements starting from the end of H_i to the beginning of H_i (wrapped) where the distance between every two adjacent elements is as long as possible but shorter than or equal to $|H_i|$. Note that both H_i and T_i may wrap around U, meaning the distance between any two elements a and b are measured by $(b-a) \bmod |U|$. Finally, the CR i obtains $V_i = H_i \cup T_i$ if H_i and T_i can be successfully found during the threshold descent; otherwise, $V_i = U \backslash D_i$. The case for $c_{\max} \leq \lfloor\sqrt{n}\rfloor - 1$ is similar except that the CR i seeks for the longest H_i with elements satisfying the threshold such that $c_{\max} < |H| \leq \lfloor\sqrt{n}\rfloor$. We call that $V_i = H_i \cap T_i$ as a positive return and $V_i = U \backslash D$ as a negative return.

The V_i returned by Algorithm 1 is *not* final. We divide the neighbor discovery step in Fig. 1 into two phases. The first is called the *validation phase*, where each CR hops among channels in $U \backslash D_i$ (as if there was no self-channel selection) for

MTTR slots. For each rendezvous with its neighbors happened during this phase, the CR adds the corresponding common channel to $|V_i|$. Then, the CR enters the second phase called the *stable phase* where $|V_i|$ is final. From now on, the CR hops among only those channels in $|V_i|$ by adopting a short-cycled CH scheme [5], [16].

Algorithm 1. Randomly-Started Stability-Descent Selection Algorithm

```
variables:
  Input: $U$ ($|U| = n$), $D_i$, and $L_i$
  Output: $V_i$
$k \leftarrow 2\lfloor\sqrt{n}\rfloor$
$threshold \leftarrow$ channel stability of the $k$th elements in $L_i$
$c_{\max} \leftarrow$ length of the longest continuous elements in $D_i$
$r \leftarrow$ a random number chosen uniformly from $U$
repeat
  //The threshold descent step
  repeat
    //The random start step
    If $c_{\max} > \lfloor\sqrt{n}\rfloor - 1$ then
      Starting from $r$, find the first set $H$ of continuous elements whose stability $\geq threshold$ such that $|H| = c_{\max} + 1$
      Break if cannot find such H
    end
    else
      Starting from $r$, find the first set $H$ of continuous elements whose stability $\geq threshold$ such that $|H|$ is as close to $\lfloor\sqrt{n}\rfloor$ as possible and $|H| > c_{\max}$
      Break if cannot find such H
    end
    $p \leftarrow$ the last element of $H$
    $T \leftarrow \{p\}$
    repeat
      Starting from $p$, find the farthest element $q$ whose stability $\geq threshold$ such that $(q - p) \bmod (n) \leq |H|$
      Break if cannot find such $q$
      Add $q$ to $T$
      $p \leftarrow q$
    until $q \in H$;
    return $H \cup T$
  until;
  $k$++
  $threshold \leftarrow$ channel stability of the $k$th elements in $L_i$
until $threshold \leq 0$;
return $U\backslash D_i$
```

4.2 Analysis

Next, we analyze the RSSD algorithm in terms of the criteria of Problem 3.

Criterion (c). The self-selected channels preserve network connectivity.

Theorem 4. *Given two CRs i and j with $V_i \subseteq U\backslash D_i$ and $V_j \subseteq U\backslash D_j$ at the stable phase, we have $V_i \cap V_j \neq \emptyset$ if $(U\backslash D_i) \cap (U\backslash D_j) \neq \emptyset$.*

Proof. We consider the cases where (i) both V_i and V_i are positive returns of RSSD, (ii) both V_i and V_i are negative returns, and (iii) otherwise. For case (i), we have $V_i \supseteq H_i \cup T_i$ and $V_j \supseteq H_j \cup T_j$. Define the head, h_i, of V_i as the beginning element of H_i (but not necessary the smallest element of H_i if H_i is wrapped around U). We prove that $V_i \cap V_j \neq \emptyset$ for any i and j. Without loss of generality, assume that $|H_i| > |H_j|$. By definition of RSSD, there exist $|H_i|$ continuous elements from h_i to $h_i + |H_i| - 1$ in V_i. If $h_i \in V_j$, we are done. Otherwise, by definition of RSSD, the distance (modulo n) between every two adjacent elements in V_j must be less than or equal to $|H_j|$. This implies that there exists an element t in $(h_i, h_i + |H_j| - 1]$ in V_j. However, since $h_i + |H_j| - 1 \leq h_i + |H_i| - 1$, the element t is also in H_i. We have $V_i \cap V_j \supseteq \{t\} \neq \emptyset$.

For case (ii), $V_i = U\backslash D_i$ and $V_j = U\backslash D_j$ so $V_i \cap V_j = (U\backslash D_i) \cap (U\backslash D_j) \neq \emptyset$. For case (iii), without loss of generality, assume that V_i is a positive return of RSSD and V_j is negative. During the validation phase of the neighbor discovery step, CR i can rendezvous with CR j ($V_j = U\backslash D_j$ and $(U\backslash D_i) \cap (U\backslash D_j) \neq \emptyset$), so all channels in $(U\backslash D_i) \cap (U\backslash D_j)$ are added to V_i, i.e., $V_i \supseteq H_i \cup T_i \cup ((U\backslash D_i) \cap (U\backslash D_j))$. We have $V_i \cap V_j \neq \emptyset$. □

Criteria (b) and (d). These two criteria can be easily satisfied if V_i's are positive returns of RSSD. For each V_i, the threshold descent ensure that those most stable channels will be included first. And, given a threshold, the RSSD starts seeking channels from a randomly selected channel r, making the channels in different H_i's spread among the entire $U\backslash D_i$. From the previous theorem, two positive V_i and V_i must intersect at a channel in either H_i or H_j. Thus the rendezvous can be spread out in $U\backslash(D_i \cup D_j)$ accordingly. Furthermore, if Condition 2 holds, then CR pairs in different geographical regions sense divergent $D_i \cup D_j$, implying that the rendezvous across regions can spread out in the entire U. The question is: how likely will the RSSD returns positively?

Theorem 5. *Given U and D_i. Let $|U| = n$, $|D_i| = d$, and $c_{\max}$ be the length of the longest continuous elements in D_i. The RSSD is guaranteed to return positively if a) $d \leq n/3$; and b) $c_{\max} < \lfloor(n-d)/d\rfloor$.*

Proof. The $d \leq n/3$ ensures that $c_{\max} < \lfloor(n-d)/d\rfloor$ is feasible. Let g be the number of groups of continuous elements in D_i (including groups with a single element). We have $1 \leq g \leq d - c_{\max} + 1$, where $g = d - c_{\max} + 1$ when there are $d - c_{\max}$ single-element groups and one group of $c_{\max}$ elements. Given these g groups, there exists at least a set L of continuous elements in $U\backslash D_i$ of size larger than or equal to $\lfloor(n-d)/g\rfloor$. Since $g \leq d - c_{\max} + 1$ and $c_{\max} \geq 1$, we have $|L| \geq \lfloor(n-d)/g\rfloor \geq \lfloor(n-d)/(d - c_{\max} + 1)\rfloor \geq \lfloor(n-d)/d\rfloor > c_{\max}$. The set L can be picked as H and the probe scheme returns. □

In practice, the RSSD has a good chance to return positively due to Conditions 1 and 2. For example, let $n = 100$ and $d = n/10 = 10$ by Condition 1. In such a case, $v \leq n/3$ is satisfied, so $c_{\max}$ needs to be larger than or equal to $\lfloor(100-10)/10\rfloor = 9$ to break the guarantee. However, by Condition 2, it is unlikely that 9 out of the 10 occupied channels are continuous. We will study the positive return rate further in the experiments.

Criterion (a). Another advantage of the RSSD is that it is able to produce small-sized quorums in general CR environments.

Theorem 6. *Given U and D_i, let $|U| = n$, $|D_i| = d$, $c_{\max}$ be the length of the longest continuous elements in D_i, and $h = \max(c_{\max} + 1, \sqrt{n})$. The size of a positive return $|V_i|$ of RSSD is at most $h + 2\lceil(n - h)/h\rceil$.*

Proof. From Algorithm 1, the size of a positive return $|V_i|$ is equal to $|H \cup T| = |H| + |T \backslash H| = h + |T \backslash H|$, which is maximal when the distance between any two adjacent elements in $|T \backslash H|$ is minimized. We first prove that for any three successive elements a, b, and c in $|H \backslash T|$, the distance between a and b plus the distance between b and c cannot be shorter than $h + 1$. If the element $a + h$ is not occupied, we have $b = a + h$. In this case, c is at least $a + h + 1$ and we are done. Otherwise, let x be the number of occupied elements counting backward from $a + h$. We have $b = a + h - x$. The element c can be no smaller than $b + x + 1 = (a + h - x) + x + 1 = a + h + 1$. We obtain the proof. This implies that any interval of length h in $N \backslash H$ has at most two elements. Therefore, $|T \backslash H|$ is at most $2\lceil(n-h)/h\rceil$. We have $|H \cup T| = h + |T \backslash H| \leq h + 2\lceil(n - h)/h\rceil$. □

We can see that $|V_i|$ is bounded by $O(\sqrt{n})$ if $c_{\max} = O(\sqrt{n})$. In practice, this is supported by Condition 2. Furthermore, with high chance of positive return given by the previous Theorem, few channels would be added to V_i during the validation phase. The final V_i is likely to be small as well.

Theorem 7. *Given U and D_i. Let $|U| = n$, $|D_i| = d$, and $c_{\max}$ be the length of the longest continuous elements in D_i. The RSSD is guaranteed to have a positive return with smallest quorum size $2\lfloor\sqrt{n}\rfloor$ if $d \leq \lfloor\sqrt{n}\rfloor - 1$.*

Proof. Let d PU-occupied channels separate U evenly into $d + 1$ groups and the length of longest continuous elements in $U \backslash D_i$ will be $n/(d + 1)$, which is the shortest when $|D_i| = d$. The $d \leq \lfloor\sqrt{n}\rfloor - 1$ ensures that the size to the set H to be $\lfloor\sqrt{n}\rfloor$ and implies $c_{\max} \leq \lfloor\sqrt{n}\rfloor - 1$ that the set T can always be found with size $\lfloor\sqrt{n}\rfloor$. Thus size of positive return will be $2\lfloor\sqrt{n}\rfloor$ if $d \leq \lfloor\sqrt{n}\rfloor - 1$. □

Note that given an arbitrary D_i, where $|D_i| > \lfloor\sqrt{n}\rfloor - 1$, RSSD may not always has positive return. Although the set T in Algorithm 1 can always be found by picking the elements right before and after the set of continuously occupied elements of length $c_{\max}$, however, H may not be found by Algorithm 1. A sufficient condition for the positive return is given below.

Theorem 8. *Given U and p, the probability of channel occupancy for every channel. Let $|U| = n$ and $|V_i|$ be the size of positive return of RSSD and $c_{\max}$ be the length of the longest continuous elements in D_i. The probability that $2\lfloor\sqrt{n}\rfloor \leq |V_i| \leq 3\lfloor\sqrt{n}\rfloor$ when $c_{\max} < \lfloor\sqrt{n}\rfloor$ will higher than or equal to $n(1-p)^{\lfloor\sqrt{n}\rfloor}(1 - p^{\frac{\lfloor\sqrt{n}\rfloor}{2}})^{2\lfloor\sqrt{n}\rfloor-2}$.*

Proof. Divide U into $2\lfloor\sqrt{n}\rfloor - 1$ groups of continuous channels that the head group with size of $\lfloor\sqrt{n}\rfloor$ and the other groups with size of $\frac{\lfloor\sqrt{n}\rfloor}{2}$. Let the channels in head group be all available for the CR, and the probability is $(1 - p)^{\lfloor\sqrt{n}\rfloor}$. And we let each group of the other groups has at least one available channel in it, the probability is $(1 - p^{\frac{\lfloor\sqrt{n}\rfloor}{2}})^{2\lfloor\sqrt{n}\rfloor-2}$. We can let the head group be the set H and make sure that the set T can be found due to the distances between any two available channels in other groups will not be larger than $2 \cdot \frac{\lfloor\sqrt{n}\rfloor}{2} - 1$, and the $|V| = |H \cup T|$ will not be larger than $3\lfloor\sqrt{n}\rfloor$. Thus the the probability that $2\lfloor\sqrt{n}\rfloor \leq |V_i| \leq 3\lfloor\sqrt{n}\rfloor$ when $c_{\max} < \lfloor\sqrt{n}\rfloor$ will be larger than $(1 - p)^{\lfloor\sqrt{n}\rfloor}(1 - p^{\frac{\lfloor\sqrt{n}\rfloor}{2}})^{2\lfloor\sqrt{n}\rfloor-2}$. Using the same example above, let $n = 100$ and $p = 0.1$ by Condition 1, the probability for RSSD to return $V_i \leq 3\lfloor\sqrt{n}\rfloor$ is larger than 34.68 percent. However, we do not consider the probability when $c_{\max} \geq \lfloor\sqrt{n}\rfloor$ here, the probability for RSSD to return $|V_i| \leq 3\lfloor\sqrt{n}\rfloor$ may be much larger. □

5 PRACTICAL ADAPTATIONS

Considering some practical application scenarios, we further improve the RSSD by leveraging the additional information the CRs get during the data-channel negotiation. For example, we can leverage the synchronous clocks among a pair of CRs to reduce the re-rendezvous time (the main cause to make a interruption delay to be extremely large) if they synchronize their clocks during data transmission. And after re-rendezvous, the CRs can know their own roles. We can take advantage of known roles to let CRs select fewer and more stable channels to be data-channels.

5.1 Leveraging the Synchronous Clocks

We can leverage the synchronous clocks among CRs to reduce the very long re-rendezvous time from $O(|U|^2)$ to $O(|U|)$ slots. For a pair of CRs consisting of a sender and a receiver, they need to re-rendezvous if they do not have any common available channel after one or more PU interruptions or SU collisions. However, the clocks of the sender and receiver can be synchronized during the data transmission before they re-rendezvous. Legitimately, we can assume that the sender and receiver synchronize their clocks and can start re-rendezvous at the same time. Let both of them use the same cycle length $|U|$. We propose a Sync Channel Hopping (SCH) scheme customized for RSSD with synchronous clocks. We describe SCH in the following:

Let S_i be the CH sequence generated by SCH and let the cycle length $|S_i^{[x]}|$ in every xth cycle be $|U|$. The starting time of rendezvous cycles should be aligned among all CRs in the network, where every CR can know the index of current time slot in the rendezvous cycle. We assign channels to S_i by

$$s_i^{[x,y]} = \begin{cases} y, & x = 0 \text{ and } y \in V_i, \\ \text{an arbitrary element of } V_i, & x = 0 \text{ and } y \notin V_i, \\ s_i^{[x-1,y]}, & \text{otherwise,} \end{cases}$$

where $s_i^{[x,y]} = y$ means that CR i should hop the channel which is indexed y in U (indexed from 0) at the yth slot in the xth cycle and V_i is the selected stable channels by RSSD.

Theorem 9. *Given any pair of CRs i and j with synchronous clocks following the CH sequences S_i and S_j generated by SCH respectively, the CRs i and j will rendezvous within $O(|U|)$ slots as long as both i and j get positive returns by RSSD, i.e., $V_i = H_i \cup T_i$ and $V_j = H_j \cup T_j$.*

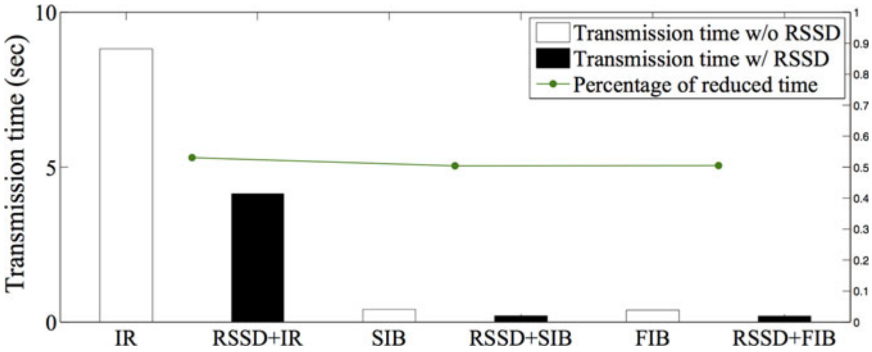


Fig. 5. The average single-slot transmission time and the percentage of reduced time by RSSD under varying methods.

Proof. By Theorem 4, we have $V_i \cap V_j \neq \emptyset$, and without loss of generality, we assume there exists a channel c, $c \in V_i \cap V_j$ and $c \in U$. By definition of SCH, i and j can know when is the cth slot in the aligned rendezvous cycle and both of them will hop to the channel c at that time slot, so that they can rendezvous within the slot. Due to $c \in U$, the MTTR is bounded by $|U|$. □

As a pair of CRs can know if they have no common available channels when the interruption occurs, they can switch to use SCH for re-rendezvous intermediately. Thus, the delay for re-rendezvous is largely reduced and bounded by $O(|U|)$ slots.

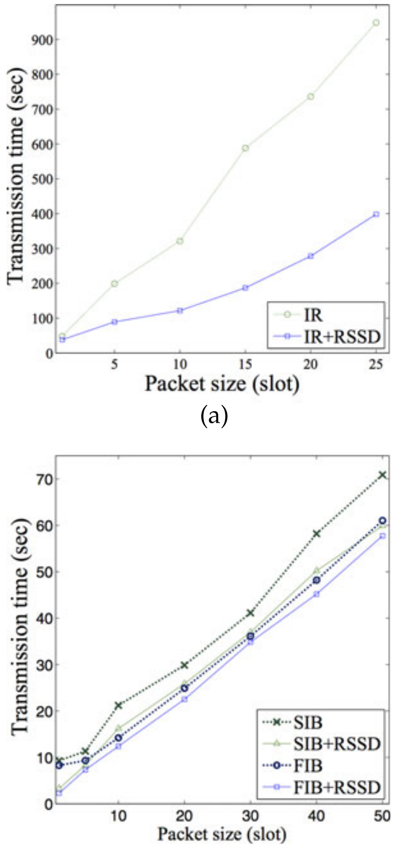
5.2 Leveraging the Known Roles

After re-rendezvous, the CRs have known their own roles (either sender or receiver), and we hope that CRs can select more stable channels than the last selection to avoid re-rendezvous again. We can exploit the known roles to stabilize the selection by lightly modifying the output of RSSD to be the new selection. Our idea is that if a CR can select less channels for data-channel negotiation, it can increase the threshold of channel stability in RSSD algorithm and gets a selection with more stable channels. Let the sender i select a set of stable channels V_i that $V_i = H_i$, where $|H_i| = \sqrt{|U|}$. And let the receiver j select a set of stable channels $V_j = T_j$, where the distance (modulo $|U|$) between every two adjacent elements in V_j must be less than or equal to $\sqrt{|U|}$. By doing that, we make the number of selected channels to be smaller than the original selection for both the sender and receiver. And by the proof of Theorem 4, we have $V_i \cap V_j = H_i \cap T_j \neq \emptyset$ that sender and receiver can find at least one common channel to be their data-channel and it is likely to be more stable than the original selection.

6 PERFORMANCE EVALUATION

We implement RSSD and three data-channel negotiation schemes, IR, FIB and SIB, by modifying WiFi MAC using Network Simulator 3 (NS3). All these implementations follow the IEEE 802.11b protocol to manage the data transmission. We simulate random PU interference by modeling the channel stability based on real measurements [13]. In our default setting, the initial spectrum occupancy is set to 0.1. And we set the time to the next PU interference or each CR, which also means its available time from now on, follows a channel stability model that the time is assigned by the channel vacancy length fitting function derived from the real measurement [13] with TV1 parameters. We first let the channel stability be estimated without any error to see if RSSD can work properly (Figs. 5, 6 and 7), and then we show the performance of RSSD under a practical environment with estimation errors (Fig. 8).

In the simulation, number of universal channels $|U|$ is set to 60. We allocate 30 CRs randomly in a $500\ \mathrm{m}^2$ square and let each CR have a transmitting radius $250\ \mathrm{m}$. A time slot is set to $10\ \mathrm{ms}$ for both the rendezvous scheme and single-slot transmission [5]. A single-slot transmission will last for $9.5\ \mathrm{ms}$ that is close to a time slot. And the data transmission time for a multi-slot transmission is a random variable



(a)

(b)

Fig. 6. (a) The average time over the longest 5 percent of multi-slot transmission of IR and IR with RSSD under varying packet sizes. (b) The average time over the longest 5 percent of multi-slot transmission of SIB and FIB with and without RSSD under varying packet sizes.

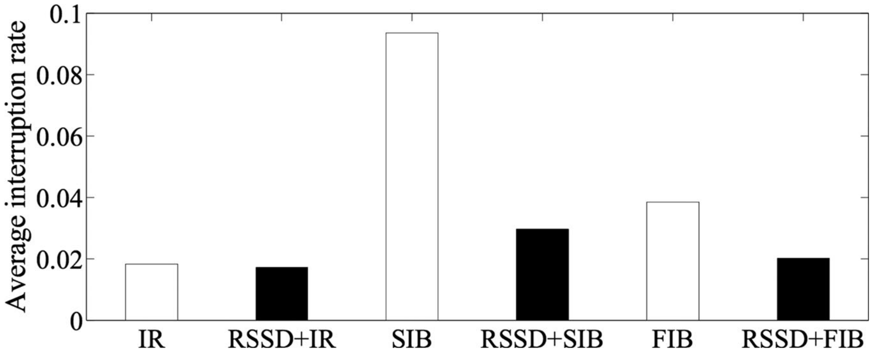


Fig. 7. The average interruption rate over longest 5 percent of multi-slot transmission under varying methods.

under a normal distribution with the mean, 15 s, and the standard deviation, 5 s. Each sender CR has a data queue, and we make the queue be full when starting a simulation. The sender CR can notify and start transmission when hopping to a common available channel with a receiver CR. Each simulation run lasts for $1{,}000$ s, and each data point is obtained by averaging 50 runs.

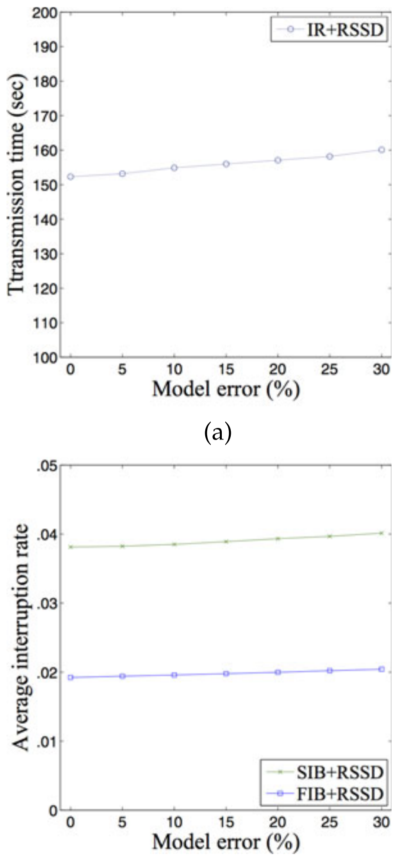
We compare the performance of data-channel negotiation schemes, IR, FIB, SIB and OB, with and without RSSD. For SIB, a CR pair will select the data-channel with largest channel stability between them, and then applies RSSD to rendezvous channels and dedicated data-channels respectively. We do not compare these data-channel negotiation methods with OB due to that OB using the dedicated data channels may fail without a centralized coordination. Besides, after self-channel selection by RSSD, we adopt a short-cycle CH scheme [5] for the neighbor discovery, and perform the practical adaptations mentioned in Section 4.2.

Positive Return Rate: To evaluate the positive return rate of RSSD in the practical environment, we generate 1,000 cases of spectrum occupancy probabilities according to the real spectrum usage measurement [13]. For each simulation, we set PU-occupied probabilities for the channels by one of 1,000 cases. We run the RSSD under 1,000 different cases and get a 99.73 percent of positive return rate. This result coincides with our analysis that RSSD has high probability to get a positive return.

Single-slot Transmission: The performance bottlenecks of the single-slot transmission are the notification delay and the interruption delay. By reducing the size of selected channels from $|U|$ to $|V|$ (approximately $2\lfloor\sqrt{|U|}\rfloor$ mentioned in Section 4.1), a sender CR only needs to traverse less than a half number of original channels to notify the receiver. Fig. 5 gives the experimental evidence, and we can see that RSSD shortens the transmission time by more than 50 percent for all data-channel negotiation schemes. And the re-rendezvous time, which is the largest part of interruption delay, is also largely reduced from $O(|U|^2)$ to $O(|V|^2)$. This also reduces the idle time between two rendezvous slots of IR while the rendezvous time is much smaller.

Multi-slot Transmission: For multi-slot transmission, interruption delay and waiting time for congestion are the two main sources of performance degradation. RSSD mitigate the impact of interruption by selecting stable channels as data-channels. For IR, since the current rendezvous schemes are designed to avoid interruptions from other CRs, IR is inherently robust to PU interruptions that CRs following IR scheme, which exchange data only on their rendezvous channels, will be nearly not affected by interruptions. Because the interruption rate is originally low for IR, RSSD can only reduce few interruptions for IR. (as shown in Fig. 7). However, RSSD greatly decreases IR's transmission time by selecting a much smaller and stable V_i to shorten the waiting time incurred by using V_i for channel hopping as shown in Fig. 6.

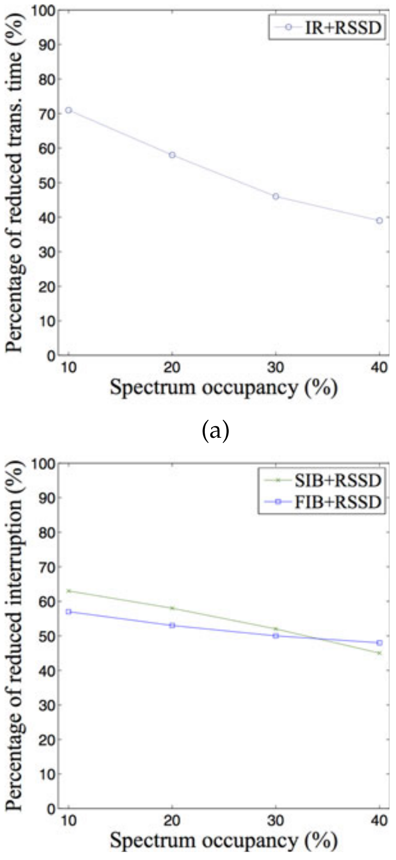
On the other hand, RSSD achieves at least 15 percent reduction for the longest 5 percent transmission for SIB and FIB, because CRs following SIB or FIB are originally prone



(a)

(b)

Fig. 8. (a) The average time over the longest 5 percent of multi-slot transmission of IR with RSSD under varying stability model error. (b) The average interruption rate over longest 5 percent of multi-slot transmission under varying stability model error.



(a)

(b)

Fig. 9. (a) The percentage of reduced average time over the longest 5 percent of multi-slot transmission of IR with RSSD compared to IR without RSSD under varying spectrum occupancies. (b) The percentage of reduced average interruption rate over longest 5 percent of multi-slot transmission of SIB and FIB with RSSD compared to them without RSSD under varying spectrum occupancies.

to be interrupted as we observed in Fig. 4 and RSSD lets CRs not over-crowd into the same most stable channels, RSSD spreads common stable channels of different pair to different channels thus reduce the degree of congestion and SU collisions. Fig. 7 shows the interruption rate which consists of PU interruption rate and SU collision rate. We can see that RSSD reduces 70 percent of interruption for SIB that it tends to stay longer on a data-channel thus having higher chance to be interrupted by an unstable channel. For FIB, which is naturally not easy to be interrupted, RSSD still reduces 56 percent of interruption for it.

Effect of Stability Model Error: We assume the channel stability can be estimated without any error in Algorithm 1, however, this assumption is not practical in the real CR environment. Here, we investigate the effect of model error that we add a random model error under a normal distribution with a mean e to the available time t. We varying e from 0 to $0.3t$ and the results shown in the Fig. 8. We can see that the larger model error only increases slightly higher transmission time for IR and almost does not change the interruption rate for SIB and FIB. Based on the results, we believe that RSSD can work efficiently in real environment even if there are some model error.

Effect of Spectrum Occupancy: We study the RSSD performance under the different spectrum occupancy and show the results in Fig. 9. The higher spectrum occupancy, which means there are fewer channels available for CRs, reduces the performance of RSSD due to the probability of positive return is getting smaller. However, RSSD achieves at least 45 percent reduction of the interruption for SIB and FIB, and can shorten the transmission time as least 39 percent for IR. Even the spectrum occupancy will affect the RSSD performance, RSSD can still improve the data transmission in some higher occupancy environment.

7 RELATED WORK

There are not many studies about the data-channel negotiation and data transmission between channel-hopping CR pairs. Existing CH schemes [3], [21], [22], [23] select the data-channel either use FIB [3], [23] or IR [21], [22] that follow the receiver's schedule to transmit data or transmit in the rendezvous slot. Note that SSCH [22] which adopts IR does not allow the change of hopping schedules, but it will extend the current slot to wait for the completion of the current packet. These works did not discuss the detail communication steps such as notification and re-transmission. And they focus on investigating the performance of multi-hop transmission, which is different from our focus in identifying and eliminating the causes of overhead in pairwise data transmission. Our results can also benefit the multi-hop transmission performance.

Another kind of selection methods [7], [8], [9] adopt SIB and select the most stable channels as the data-channels by using channel modeling or other criteria indicating channel quality. They can also be seen as OB schemes if data-channels are dedicated. They make good use of channel model and can pick out high-quality data-channels. However, these works did not consider different selection strategies for different data sizes. Also, many works [25], [26] address broadcasting issues in CRN, but we focus on unicast between a CR pair may be suitable for more applications. Some work also studies distributed channel selection for Wi-Fi like environments [27], [28] but does not consider characteristics, such as PU dynamics, of cognitive radios.

Being in a separate step, RSSD is complementary to all the above schemes (and even the rendezvous schemes) in that it improves the stability and reduce significant overhead of data transmission.

8 CONCLUSIONS AND FUTURE WORK

In this paper, we review existing data-channel negotiation schemes for channel-hopping CRs, and compare four kinds of existing data-channel negotiation schemes for channel hopping CRs through qualitative discussion and quantitative measurement of their performance. We find and demonstrate that notification delay and interruption delay makes performance unstable in the dynamic environments. Based on the results, we observe that current in-band schemes, although fully-distributed, incur a huge notification delay for short data transmissions and the instability for long data transmissions due to occasional interruption delay. By carefully re-examining the current communication step, we propose a new channel selection step preceding the rendezvous and data channel selection steps, called self-channel selection. We formally define self-channel selection problem, and the RSSD is proposed to solve the problem that minimizes notification and re-transmission cost. We also discuss the practical usage of RSSD and its optimizations. Specifically, we propose a unified method to integrate RSSD with most existing CH rendezvous schemes (for in-band data transmission). Simulation results show that RSSD achieves more than 50 percent delay reduction for single-slot data transmission, and avoids at most 70 percent of interruptions for multi-slot transmission, it also work great even if the channel stability model are not perfect and can still improve the transmission performance when the spectrum occupancy get higher.

However, currently not all data-channel negotiation schemes and environments equally benefit from RSSD on the improvement of stability. A dynamic adjusting method to different schemes and environments, for example, an algorithm to adjust threshold based on the number of neighboring CRs, would be great to be integrated into the self-channel selection algorithm. Another critical issue is that CRs may have varying channel models at different time periods in a day or in a week. An investigation on RSSD and data-channel negotiation performance are essential, and collecting more CR network traces and simulating more realistic channel models are key subjects of our future study.

ACKNOWLEDGMENTS

This work is partially supported by MOST103-2221-E-001-018 and MOST104-2221-E-001-002 of the Ministry of Science and Technology, Taiwan and Thematic Research Grant of Academia Sinica.

REFERENCES

[1] I. Akyildiz, W. Lee, M. Vuran, and S. Mohanty, "Next generation/dynamic spectrum access/cognitive radio wireless networks: A survey," *Comput. Netw.*, vol. 50, no. 13, pp. 2127–2159, 2006.

[2] C. Wu and S. Wu, "On bridging the gap between homogeneous and heterogeneous rendezvous schemes for cognitive radios," in *Proc. ACM Int. Symp. Mobile Ad Hoc Netw. Comput.*, 2013, pp. 207–216.

[3] K. Bian and J. Park, "Maximizing rendezvous diversity in rendezvous protocols for decentralized cognitive radio networks," *IEEE Trans. Mobile Comput.*, vol. 12, no. 7, pp. 1294–1307, Jul. 2013.

[4] Z. Lin, H. Liu, X. Chu, and Y. Leung, "Jump-stay based channel-hopping algorithm with guaranteed rendezvous for cognitive radio networks," in *Proc. IEEE Int. Conf. Comput. Commun.*, 2011, pp. 2444–2452.

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

[6] N. Theis, R. Thomas, and L. DaSilva, "Rendezvous for cognitive radios," *IEEE Trans. Mobile Comput.*, vol. 10, no. 2, pp. 216–227, Feb. 2011.

[7] V. Mishra, C. Lau, and S. Chan, "QoS based spectrum decision framework for cognitive radio networks," in *Proc. 18th Int. Conf. Netw.*, 2012, pp. 18–23.

[8] M. Rehmani, A. Viana, H. Khalife, and S. Fdida, "Surf: A distributed channel selection strategy for data dissemination in multi-hop cognitive radio networks," *Comput. Commun.*, vol. 36, pp. 1172–1185, 2013.

[9] M. Hoyhtya, S. Pollin, and A. Mammela, "Performance improvement with predictive channel selection for cognitive radios," in *Proc. 1st Int. Workshop Cognitive Radio Advanced Spectrum Manage*, 2008, pp. 1–5.

[10] D. Cabric, S. Mishra, and R. Brodersen, "Implementation issues in spectrum sensing for cognitive radios," in *Proc. Asilomar Conf. Signals Syst. Comput.*, 2004, pp. 772–776.

[11] A. Ghasemi and E. Sousa, "Spectrum sensing in cognitive radio networks: requirements, challenges and design trade-offs," *IEEE Commun. Mag.*, vol. 46, no. 4, pp. 32–39, Apr. 2008.

[12] M. McHenry, "Nsf spectrum occupancy measurements project summary," *Shared Spectrum Company Report*, 2005, [Online]. Available: http://www.sharedspectrum.com/papers/spectrum-reports/

[13] S. Yin, D. Chen, Q. Zhang, M. Liu, and S. Li, "Mining spectrum usage data: A large-scale spectrum measurement study," *IEEE Trans. Mobile Comput.*, vol. 11, no. 6, pp. 1033–1046, Jun. 2012.

[14] Z. Lin, H. Liu, X. Chu, and Y.-W. Leung, "Enhanced jump-stay rendezvous algorithm for cognitive radio networks," *IEEE Commun. Lett.*, vol. 17, no. 9, pp. 1742–1745, Sep. 2013.

[15] L. Yu, H. Liu, Y.-W. Leung, X. Chu, and Z. Lin, "Multiple radios for fast rendezvous in cognitive radio networks," *IEEE Trans. Mobile Comput.*, vol. 14, no. 9, pp. 1917–1931, Sep. 2015.

[16] Z. Gu, H. Pu, Q.-S. Hua, and F. C. Lau, "Improved rendezvous algorithms for heterogeneous cognitive radio networks," in *Proc. IEEE INFOCOM*, 2015, pp. 154–162.

[17] C. Cordeiro, K. Challapali, D. Birru, and S. Shankar N, "IEEE 802.22: An introduction to the first wireless standard based on cognitive radios," *J. Commun.*, vol. 1, pp. 38–47, 2006.

[18] C. Stevenson, G. Chouinard, Z. Lei, W. Hu, S. Shellhammer, and W. Caldwell, "IEEE 802.22: The first cognitive radio wireless regional area network standard," *IEEE Commun. Mag.*, vol. 47, no. 1, pp. 130–138, Jan. 2009.

[19] K. Letaief and W. Zhang, "Cooperative communications for cognitive radio networks," in *Proc. IEEE*, vol. 97, no. 5, pp. 878–893, May 2009.

[20] S. M. Mishra, A. Sahai, and R. W. Brodersen, "Cooperative sensing among cognitive radios," in *Proc. IEEE Int. Conf. Commun.*, 2006, pp 1658–1663.

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

[22] P. Bahl, R. Chandra, and J. Dunagan, "SSCH: Slotted seeded channel hopping for capacity improvement in ieee 802.11 ad-hoc wireless networks," in *Proc. ACM Int. Conf. Mobile Comput. Netw.*, 2004, pp. 216–230.

[23] C. Shih, T. Wu, and W. Liao, "DH-MAC: A dynamic channel hopping MAC protocol for cognitive radio networks," in *Proc. IEEE Int. Conf. Commun.*, 2010, pp. 1–5.

[24] W. Luk and T. Wong, "Two new quorum based algorithms for distributed mutual exclusion," in *Proc. IEEE Int. Conf. Distrib. Comput. Syst.*, 1997, pp. 100–106.

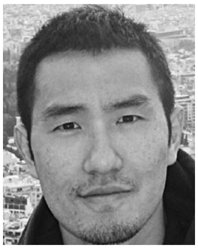
[25] Y. Song and J. Xie, "A distributed broadcast protocol in multi-hop cognitive radio ad hoc networks without a common control channel," in *Proc. IEEE Int. Conf. Comput. Commun.*, 2012, pp. 2273–2281.

[26] T. Wu and W. Liao, "Time-efficient broadcasting in cognitive radio networks," in *Proc. IEEE Int. Conf. Gloal Telecommun.*, 2013, pp. 1191–1196.

[27] P. Bahl, R. Chandra, T. Moscibroda, R. Murty, and M. Welsh, "White space networking with Wi-Fi like connectivity," in *Proc. ACM SIGCOMM Conf. Data Commun.*, 2009, pp. 27–38.

[28] A. Mishra, V. Shrivastava, D. Agrawal, S. Banerjee, and S. Ganguly, "Distributed channel management in uncoordinated wireless environments," in *Proc. ACM Int. Conf. Mobile Comput. Netw.*, 2006, pp. 170–181.

Ching-Chan Wu received the BS degree from the Department of Computer Science, National Tsing Hua University, Taiwan. He is currently working toward the PhD degree in the Department of Computer Science, National Tsing Hua University, Taiwan. His research interests include wireless networks, cognitive radio networks, and mobile data management.



Shan-Hung Wu received the BS degree from the Department of Information Management, National Central University, Jhongli, Taiwan, the MS degree from the Department of Computer Science and Information, National Taiwan University, Taipei, Taiwan, and the PhD degree from the Department of Electrical Engineering, National Taiwan University, Taipei, Taiwan. He is currently an associate professor in the Department of Computer Science, National Tsing Hua University, Hsinchu, Taiwan. Before joining National Tsing Hua University, he was a senior research scientist with Telcordia Technologies and led several Telcordia projects including the clustered in-memory database, video surveillance workbench, and the rating engine of Telcordia ISCP platform. His research interests include wireless networks, database systems, data mining, and mobile data management.

Wen-Tsuen Chen received the BS degree in nuclear engineering from National Tsing Hua University, Taiwan, and the MS and PhD degrees in electrical engineering and computer sciences both from the University of California, Berkeley, in 1970, 1973, and 1976, respectively. He has been with the National Tsing Hua University since 1976 and is a Sun Yunsuan chair professor in the Department of Computer Science. He has served as chairman in the Department, Dean of College of Electrical Engineering and Computer Science, and the president of National Tsing Hua University. Since March 2012, he has joined Academia Sinica, Taiwan, as a distinguished research fellow in the Institute of Information Science. His research interests include computer networks, wireless sensor net- works, mobile computing, and parallel computing. He received numerous awards for his academic accomplishments in computer networking and parallel processing, including the Outstanding Research Award of the National Science Council, Academic Award in Engineering from the Ministry of Education, Technical Achievement Award and Taylor L. Booth Education Award of the IEEE Computer Society, and is currently a lifelong National Chair of the Ministry of Education, Taiwan. He is the founding general chair of the IEEE International Conference on Parallel and Distributed Systems and the general chair of the 2000 IEEE International Conference on Distributed Computing Systems among others. He is a fellow of the IEEE and a fellow of the Chinese Technology Management Association.