A Practical Self-Adaptive Rendezvous Protocol in Cognitive Radio Ad Hoc Networks

Xingya Liu and Jiang Xie
Department of Electrical and Computer Engineering
The University of North Carolina at Charlotte
Email: {xliu33, Linda.Xie}@uncc.edu

Abstract—**In cognitive radio ad-hoc networks, two users rendezvous on a common available channel to realize communications. Most existing rendezvous papers focus on success-guaranteed channel-hopping sequence design. However, the theoretical rendezvous successful rate may suffer from the available channel status changing, collisions on channels, congestion at users, and target users unavailability in practical scenarios. Thus, a practical rendezvous framework that can address these issues is highly demanded. In this paper, we develop analytical models for each possible factor which may influence the performance of rendezvous. Then, based on the analysis of each factor, we propose corresponding schemes and integrate them into a self-adaptive protocol which can adjust its reaction and optimize system parameters to adapt to the dynamic network. Simulation results demonstrate that our proposed protocol gains better performance in terms of true rendezvous successful rate, short rendezvous delay, and low congestion. To the best of our knowledge, this is the first rendezvous protocol that addresses practical issues in realistic communication scenarios in cognitive radio networks.**

I. Introduction

In cognitive radio ad hoc networks (CRAHNs), channel rendezvous is a fundamental operation which requires a pair of nodes, willing to communicate with each other, to meet on a common available channel at the same time, so that information exchange and data transmission can be carried out. In traditional mobile ad hoc networks, a node can easily find the target node's channel by using a common control channel (CCC) to exchange their control information such as node location and channel allocation. However, in CRAHNs, each node is an unlicensed user (or, secondary user (SU)), who can only use the channels which are not occupied by neighboring licensed users (or, primary users (PU)) in an opportunistic manner. Hence, the available channel sets of different SUs in CRAHNs may vary from time to time. Thus, a control channel that is commonly available to all SUs in a network may not exist or cannot last for a long time. Therefore, each SU has no knowledge about other SUs before they rendezvous. Such a blind rendezvous is very challenging and unable to be solved by conventional methods.

Recently, there are quite a few papers aiming to achieve blind rendezvous using the channel-hopping (CH) technique. In this approach, each SU equipped with one cognitive radio first senses the whole spectrum and generates a set of available channels. Then, the SU tunes its radio to the channel in the set one by one following a predefined CH sequence. Thus, two SUs can rendezvous if they hop to a same channel at the same time. However, these CH schemes have various drawbacks when applied to realistic communication scenarios.

Many existing CH schemes are based on impractical assumptions: *(i)* the symmetric assumption [1]–[3], which requires that all SUs have the same available channel set; *(ii)* the time-synchronous assumption [4]–[6], which asks all SUs to start hopping at the same global time; *(iii)* the role preassignment assumption [6], i.e., every node pair is preassigned a role as either a sender or a receiver; and *(iv)* each occupied channel is assumed to be used by one PU. In other words, each channel's availability is associated with one PU's availability. This assumption has been widely adopted by probabilistic-based CH schemes [7], [8] since the activity of PUs usually shows similar patterns along the time. However, in realistic PU networks like 2G/3G, each PU may be randomly assigned a channel on each transmission or several PUs may share one channel when using mechanisms like time division multiple access (TDMA). Therefore, the available channel sets of SUs are unpredictable and time-varying.

Other blind rendezvous schemes based on asymmetric asynchronous models still have drawbacks because they only focus on the rendezvous-guaranteed CH sequence design [9], [10] (i.e., a SU can rendezvous with any other SU within a bounded time if they have at least one common available channel), but ignore the detailed MAC protocol design including time-slot specification, collision avoidance, and congestion control. We explain why a suitable protocol framework is necessarily desired for rendezvous design in the following.

First, since a SU stays on each channel for the same amount of time during the CH, the time it spends on each channel should be as short as possible for the sake of timely rendezvous. Meanwhile, the time should be long enough to support two SUs achieving a basic handshaking process, which, based on the Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA) mechanism in IEEE 802.11, means that a Request-to-Send (RTS) and a Clear-to-Send (CTS) can be successfully exchanged by the sender and the receiver. Therefore, it is natural to define the size of one time slot to be the length of an RTS and a CTS exchange. Thus, the basic rendezvous procedure considered in this paper is illustrated in a time slotted system shown in Fig. 1. Under the considered procedure, each SU hops to a different channel at the beginning of each time slot following the adopted CH sequence. During each time slot, a SU sender sends out an RTS message and a SU receiver listens for potential RTS messages. If the SU sender does not receive a CTS message, it keeps hopping. Otherwise, it indicates that two SUs have met on the same channel and the rendezvous is successful. Then, the two SUs stop hopping and stay on the same channel for data transmissions.

Regarding the collision avoidance issue, most existing papers assume that collisions only happen during the data transmission period, which requires the operation of a spectrum handoff [11]. However, collisions may also happen during a rendezvous process. Besides, collisions may happen not only between PUs and SUs, but also among SUs themselves. More

This work was supported in part by the US National Science Foundation (NSF) under Grant No. 0953644, 1218751, and 1343355.

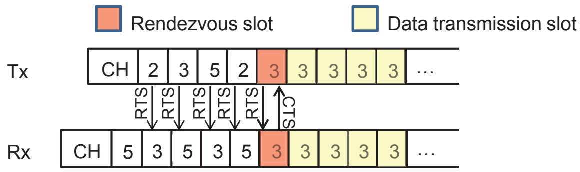


Fig. 1. The rendezvous procedure with the RTS/CTS mechanism.

importantly, collisions, no matter occur in which period or caused by what nodes, may change the available channel status, which further leads to a rendezvous delay or even failure if no corresponding actions are taken. For instance, if channel 3 in Fig. 1 becomes unavailable during the CH, the two SUs have to keep hopping and rendezvous on channel 5 in the next time slot. Since the scanning time of the whole band is much longer than the time slot we defined (i.e., it has been proposed in the IEEE 802.22 standard [12] that in the order of milliseconds per channel should be spent by PHY-layer sensing so as to achieve the desirable level of detection quality, while the transmission time of an RTS and a CTS is usually far less than 1 ms), each SU usually only performs spectrum sensing once to get the available channel set at the beginning of the CH, and then devoutly follows the predefined CH sequence before the rendezvous success. Thus, each SU requires an adjusting ability when any available channel becomes unavailable or vise versa. Although SUs in [13] have the sequence adjusting ability, the proposed approach can only work using the multi-radio technique. We address this issue using the one-radio technique by studying all the factors that may affect channel status changing and design the reasoning-adjusting mechanism to compensate this delay.

At last, the optimal time to stop rendezvous needs to be carefully studied. Since no existing work has mentioned this before, we name it the stopping time to rendezvous (STTR). It cannot be simply assigned as the theoretical upper bound of the time to guarantee rendezvous (maximum time to rendezvous (MTTR)). On one hand, since the expected time-to-rendezvous (ETTR) increases with the number of channels in a network, a long STTR can lead to a high rendezvous success rate. For example, if the total number of channels in a CRAHNs is N, the MTTR is $O(N^2)$ [9], [10], [14]–[17]. If the STTR equals to the MTTR, the theoretical rendezvous successful rate is 100%. However, on the other hand, a long STTR may result in inefficient data packet transmissions and congestion. For example, in IEEE 802.11 based protocols [18]–[21], the data packet size is usually 10 to 25 times of the combined size of the RTS and CTS (10-25 time slots). If there are 10 channels in a CRAHN, in the worst case, the time to rendezvous (TTR) equals 100 time slots. This TTR is approximately 4 to 10 times of the packet transmission time. If we treat the process as a queuing system, it means that the waiting time (rendezvous period) is much longer than the service time (data transmission period). Thus, even a modest data packet arrival rate will lead to significant packet congestion. A shorter STTR can eliminate the congestion, but it will decrease the successful rate of rendezvous. Therefore, selecting a proper STTR plays a crucial role when designing rendezvous protocols under practical scenarios. In our proposed protocol, we calculate the optimal STTR and update it dynamically based on the environmental changes.

To sum up, we propose a practical self-adaptive (PSA) rendezvous protocol in CRAHNs that includes a complete framework from network set-up to successful rendezvous with delay compensation and congestion control. Our proposed distributed protocol is not based on any of the impractical assumptions explained previously. We first develop probabilistic models for the factors which may affect the status of available channels during rendezvous. Then, we design an intelligent delay compensation scheme which can adapt to any existing rendezvous-guaranteed CH scheme. Moreover, we model the packet congestion problem during rendezvous using the queuing theory and propose a control scheme with an optimal STTR. Finally, we propose an integrated protocol which can take corresponding reactions after reasoning the influencing factors, and update the adaptive parameters after learning the dynamic network conditions. Simulation results show that our proposed protocol outperforms other existing rendezvous protocols. We conclude that our proposed protocol can achieve rendezvous with a high successful rate, short rendezvous delay, and low congestion in a practical manner.

The rest of this paper is organized as follows. In Section II, we establish analytical models for each problem mentioned above. An integrated protocol to realize our rendezvous goal is proposed in Section III. Simulation results are shown in Section IV, followed by the conclusions in Section V.

II. Problem Formulation

In this section, we analyze two main problems that a rendezvous process will definitely face in a realistic CRAHN: channel status changing during the CH and data packet congestion caused by rendezvous. We first introduce the system model in our analysis. The parameters used in our analysis are listed in Table I. Note that the values of the parameters used in the examples in the following analysis are from IEEE 802.11-based protocols [18]–[21].

TABLE I. List of Symbol Notations

TTR	The time to rendezvous in the unit of time slots
$ETTR$	The expected time to rendezvous in the unit of time slots
$MTTR$	The maximum time to rendezvous in the unit of time slots
$STTR$	The stopping time to rendezvous in the unit of time slots
N	The total number of channels in the network
M	The number of available channels of a SU
r	The radius of the sensing area of a SU
K_P	The number of neighboring PUs of a SU
K_S	The number of neighboring SUs of a SU
Cp	The channel data rate
T_R	TTR in the unit of seconds
T_E	ETTR in the unit of seconds
T_M	MTTR in the unit of seconds
T_S	STTR in the unit of seconds
T_P	The time for one data packet transmission
P_R	The rendezvous successful rate
λ_P	Average packet arrival rate of PUs in the Poisson distribution
λ_S	Average packet arrival rate of SUs in the Poisson distribution
λ_T	Average packet arrival rate at a sender in the Poisson distribution
λ_R	Average packet arrival rate at a receiver in the Poisson distribution
μ_P	Average packet service rate of PUs
μ_S	Average packet service rate of SUs
L_P	The length of a packet in the unit of time slots
TH	Throughput efficiency of the secondary network

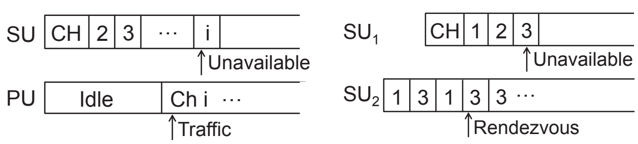
A. System Model

The system consists of finite number of PUs and SUs. Their locations are randomly chosen but are able to maintain the network connectivity. We assume that each PU's transmission range is larger than a SU's sensing range, which is realistic due to SU's power limitation. In the analysis, we assume that both PUs and SUs' traffic rate follows the Poisson distribution. However, this assumption is only used to help us understand how those factors affect rendezvous. Our proposed design does not rely on this assumption. Each time when a PU wants to transmit data, a channel is randomly chosen by the system. In this way, a channel can also be assigned to multiple PUs simultaneously. Therefore, unlike many other papers, we do not rely on the channel state information in our design, which accords with realistic communication systems. In addition, we assume that each SU has two responsibilities, sending and receiving. Since each SU works in a half-duplex mode, when a SU acts as a sender, it cannot be a potential receiver for other neighboring senders. Thus, a long TTR can harm the whole network throughput.

B. Analysis of Channel Status Changing

One common assumption in most existing rendezvous papers is that a channel's availability is stable during the CH. Since these works mainly focus on rendezvous between two nodes within one single hop, channel unstability may happen with a low probability under such simplified scenarios. However, when more neighboring nodes (SUs and PUs) and heavy traffic rates are considered in CRAHNs, the channel status will change more often, which eventually results in a rendezvous delay or failure. We summarize that there are mainly four factors which can affect the status of a channel and mathematically analyze the lower bound of each factor's occurrence probability.

1) PUs' Reoccurrence: A channel will become unavailable to a SU if a neighboring PU re-occupies the channel during rendezvous, as illustrated in Fig. 2(a).



(a) The PU's reoccurrence. (b) The SU's transmission.

Fig. 2. Packet-level factors affecting the channel status during the CH.

First, we consider the one-neighboring-PU scenario, which means that only one PU exists within a SU's sensing range. We model each PU as an $M/G/1$ system. The probability that a PU is idle is $P_0 = 1 - \rho_P$, where $\rho_P = \lambda_P/\mu_P$. Let $N(t)$ be the number of data packet arrivals of the PU in time t and the packet arrivals follow the Poisson distribution. Thus, the probability of n packet arrivals in time t is $Pr[N(t) = n] = \frac{(\lambda_P t)^n}{n!} e^{-\lambda_P t}$. The one PU reoccurrence scenario only happens when the PU is idle at the beginning of the SU's hopping, then it starts traffic during the SU hopping, and the channel the PU used is also in the SU's available channel set. Hence, the probability of one PU reoccurrence is

$$Pr[one\ PU\ reocc.] = (1 - \rho_P)\left(1 - Pr[N(T_R) = 0]\right)\frac{M}{N}. \tag{1}$$

For example, if $\lambda_p = 10pkt/s$, $\rho_P = 0.5$, $N = 20$, $M = 10$, $TTR = 15$ slots, and 1 slot= 600 bits/2Mbps, then the probability is 1.1%.

Then, we consider the multiple-neighboring-PU scenario. We derive the lower bound of the probability of PUs' reoccurrence as

$$Pr[PUs'\ reocc.] = 1 - \left(1 - Pr[one\ PU\ reocc.]\right)^{K_P}. \tag{2}$$

It is a lower bound because the probability that a busy PU becomes idle and starts traffic again during a SU's CH, which may also occupy an available channel of the SU, is not counted. Using the same parameters in the last example and adding the number of SU's neighboring PUs $K_P = 10$, $Pr[PUs'\ reocc.] = 10.5\%$. If the number and traffic rate of neighboring PUs increase, the probability will go even higher.

2) SUs' Transmissions: In CRAHNs, a SU can be close to another SU and at the same time relatively far away from a PU but still in its interference range. In this scenario, when the distant PU or the closed SU starts to transmit, their received signal strength at the SU may be similar. Under the sense-before-access manner, the channel becomes unavailable to the SU no matter who is transmitting on it. As illustrated in Fig. 2(b), SU_2 rendezvous with its targeted SU on channel 3 during SU_1's hopping and keeps transmitting data on this channel. When SU_1 hops to channel 3, the channel is already unavailable.

Using a similar method from Section II-B1, we first consider the one-neighboring-SU scenario. Unlike PUs, a SU is considered busy in both the CH process and the data transmission process. In Fig. 2(b), at the moment when SU_1 starts hopping, SU_2 has three possible states: listening, CH, and transmission. SU_2's listening state will not affect SU_1's channel availability. We also do not consider the last case because SU_2's transmission channel will be excluded in SU_1's available channel set in its scanning process. Under the CH case, since SU_2 either rendezvous before SU_1 does, or after it, we set the probability that SU_2 rendezvous before SU_1 to be 0.5. In fact, this probability is lower than the real case because the hopping process of SU_2 begins earlier than SU_1. Therefore, the lower bound of the probability of one-neighboring-SU transmission is:

$$Pr[one\ SU\ trans.] = \rho_S \frac{T_E}{T_E + T_P}\frac{\gamma}{2}, \tag{3}$$

where $\rho_S = \lambda_S/\mu_S$ and γ is a parameter in [0,1] representing the channel-location-correlation degree between neighboring SUs. γ is very close to 1 in cognitive radio environments as studied in [6], [22]. γ in (3) indicates the probability that the rendezvous channel of SU_2 is in the available channel set of SU_1. Note that T_E can be calculated if N, M, and the CH scheme are known.

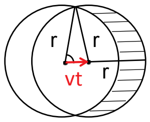
Finally, the lower bound of the probability of SUs' transmissions is:

$$Pr[SUs'\ trans.] = 1 - \left(1 - Pr[one\ SU\ trans.]\right)^{K_S}. \tag{4}$$

This factor usually influences the channel status more often than the previous one, because the traffic rates of SUs are usually at a similar level. For example, if $\gamma = 0.8$, $\rho_S = 0.5$, $L_P = 25$ slots, $ETTR = 10$ slots and 1 slot= 600 bits/2Mbps, then the probability of the one-neighboring-SU transmission scenario is 5.7% and the probability of a 5-neighboring-SU scenario is 25.5%. Note that this probability is not related to the channel data rate, in other words, no matter how fast the link speed can be, the occurrence probability of this factor is still the same.

3) SU's Mobility: A channel will become unavailable to a SU during the CH if the SU keeps moving and encounters new neighboring nodes (PUs and SUs) which are currently using an available channel of the SU. Very few rendezvous papers consider the influence of nodes' mobility. In this paper we also consider PUs and SUs in CRAHNs as static nodes during the CH. However, we give a proof to justify this assumption.

In Fig. 3, the circle represents the sensing range of a SU with a radius r. We assume that the speed of the SU is $\boldsymbol{v}$ which is a relative speed compared to surrounding nodes. The shadow part is the additional sensing area during the SU's moving. Note that the moving time $t = T_R$ and $\alpha = arccos\frac{vt}{2r}$.



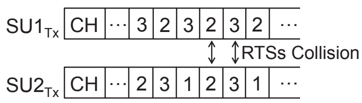


Fig. 3. SU's Mobility. Fig. 4. RTS collision.

If we assume that PUs are evenly distributed in the network, the ratio of K'_P to K_P, i.e., the number of new PUs in the SU's new sensing area to the number of original neighboring PUs of the SU, is the same as the ratio of the shadow area size to the original circular area size, then

$$K'_P = \frac{2\left(\pi r^2 \frac{180-\alpha}{360} - (\pi r^2 \frac{\alpha}{360} - \frac{vt}{2} r sin\alpha)\right)}{\pi r^2} K_P.$$

We can also get the number of new encountered SUs as

$$K'_S = \frac{2\left(\pi r^2 \frac{180-\alpha}{360} - (\pi r^2 \frac{\alpha}{360} - \frac{vt}{2} r sin\alpha)\right)}{\pi r^2} K_S.$$

We consider the probability that a new encountered neighboring PU/SU is busy on one of the SU's available channels as $Pr[busy|PU'] \geq \rho_P \frac{M}{N}$ or $Pr[busy|SU'] \geq \rho_S \frac{T_P}{T_P+T_E}\gamma$. Therefore, the lower bound of the probability of the available channel status changing occurred due to SU's mobility during the CH is

$$Pr(Mobi.) = 1 - \left(1 - \rho_p \frac{M}{N}\right)^{K'_P} \left(1 - \rho_S \frac{T_P}{T_E + T_P}\gamma\right)^{K'_S}. \quad (5)$$

Using the same parameters in the examples of Section II-B-1) and 2) and assuming $\boldsymbol{v} = 10m/s$, $r = 10m$, $Pr(Mobility) = 0.99\%$. Even if we increase the speed $\boldsymbol{v}$ to $30m/s$ which is equivalent to the highway vehicular scenario, $Pr(Mobility) = 2.95\%$ which is still negligible. However, if the ratio of the node speed to its sensing range increases, the above probability will become considerable.

4) RTS Collision: The RTS collision at the receiver side is an unavoidable issue in real networks. It also exists in our proposed protocol. As the factors explained above, the RTS collision will lead to the corresponding channel temporarily unavailable for a receiver, which will also lead to a failed or delayed rendezvous.

The RTS collision scenario is illustrated in Fig. 4. Two SU senders hop to a same channel at the same time. Suppose that they are not in each other's sensing range (otherwise the sense-before-access scheme works), but their RTS messages collide with each other at the receivers located in the overlapping area of their transmission ranges. Thus, no SU receiver can obtain the RTS information and finally the senders may miss their receiver even though they have hopped to the same channel. More importantly, for those synchronous and symmetric rendezvous models under which each SU's CH sequence is generated based on the same algorithm and the available channel set of each SU is the same, the worst case is that they will keep colliding on different channels. Therefore, it is important to design a sequence adjustment mechanism against this situation. We will propose a collision avoidance mechanism in Section III.

In the following we demonstrate the probabilistic model of the RTS collision assuming that both the SU sender and the receiver follow the same CH algorithm during blind rendezvous. Then, we have the lemma:

Lemma 1. *The probability that a sender successfully rendezvous with a receiver is the same as the probability that a sender has an RTS collision with another sender.*

Proof: If a sender and a receiver hop to a same channel at the same time, it is a successful rendezvous. On the other hand, if the receiver is replaced with a sender, it becomes an RTS collision case. ■

Therefore, the RTS collision probability between two senders at one common receiver during the time t is

$$Pr[RTS\ collision|2\ senders] = \frac{t}{T_E}. \quad (6)$$

However, this probability only takes place when the three SUs are preassigned to be senders and receiver, and both senders are in CH status. Since the probability that a SU is in a CH process is $\rho_S \frac{T_E}{T_E+T_P}$, for a receiver, the probability of an RTS collision in time t is

$$Pr[RTS\ collision|2\ SUs] = \left(\rho_S \frac{T_E}{T_E + T_P}\right)^2 \frac{t}{T_E}. \quad (7)$$

It is also a lower bound since the right side of (7) implicates that the two SUs start the CH before $t = 0$, which means that $t \geq 0$. At last, the total RTS collision probability for a receiver with K_S neighbors in time t is shown in (8). Using the same parameters in Section II-B-2), and if we set $t = T_E$, $Pr[RTS\ collision] \approx 18.6\%$. In a moderate case, i.e., $K_S = 3$, $\rho_S = 0.3$, and other parameters are the same, $Pr[RTS\ collision] \approx 2.2\%$.

5) Summary: To conclude, first, the status of SU available channels may depend more on factors *2)* and *4)* which have little relationship with the channel link speed. We study them as the **inner properties** of the blind rendezvous process. Second, in a dense-node high-traffic CRAHN, the above four factors usually affect the channel status simultaneously, which may generate tremendous influence to rendezvous in terms of the successful rate and TTR. If no actions are taken when the channel status changing happens, the performance of the designed CH sequence will degrade significantly. At last, our proposed probabilistic models for analyzing these factors can be used either to design system parameters for mitigating the influence of these factors or to estimate the real rendezvous performance in a network environment under the influence of these factors.

C. Analysis of Data Packet Congestion

Currently, no research has been done on the congestion issue during rendezvous. Similar to the channel status changing issue, the congestion problem is also an inner property inherent with the rendezvous process.

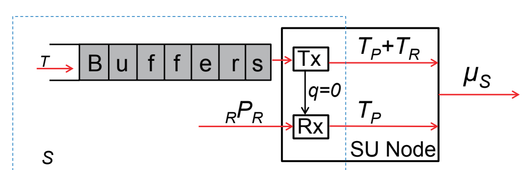


Fig. 5. The congestion model for a SU in CRAHNs.

We propose a congestion model for a SU in CRAHNs as illustrated in Fig. 5. In this model, a SU has traffic arrivals from both inside (generating data packets as a source sender)

$$Pr[RTS\ collision] \geq 1-(1-Pr[RTS\ collision|2\ SUs])^{\binom{K_S}{2}}. \tag{8}$$

and outside (receiving data packets as a relay node or an end receiver) with an average arrival rate λ_T and λ_R, respectively. Without loss of generality, the inside traffic rate is modeled as a Poisson distribution. Besides, the outside packet arrival rate is the combined traffic rate from neighboring SU senders who are aiming to communicate with the considered SU receiver, which by Burke's theorem [23], also follows a Poisson distribution. Therefore, we model each SU node as an $M/G/1$ system with $\rho = \frac{\lambda_S}{\mu_S}$. When there is no traffic and the number of buffered packets q equals 0, the SU becomes a receiver with probability $Pr[q = 0] = 1-\rho$. The service time of a packet from a sender is counted from the beginning of the CH to the end of packet transmission, T_R+T_P. On the other hand, a packet arriving to a receiver infers that the receiver just has a successful rendezvous with a sender. Hence, the service time of a packet from a receiver is Tp. Denote T_I as the average idle time of a SU,

$$T_I = \frac{1-\rho}{\rho}\frac{1}{\mu_S} = \frac{1}{\lambda_S} - \frac{1}{\mu_S}. \tag{9}$$

For a receiver, the packet has a probability to be successfully received, P_R. Since T_R is different in each successful rendezvous, P_R is a function of T_R. If K_T is the number of potential senders for this considered receiver, we can derive the lower bound of P_R as

$$P_R \geq 1-\left(1-\left((1-Pr[N(T_I)=0])\min[\frac{T_I}{T_M},1]\gamma\right)\right)^{K_T}. \tag{10}$$

Since the total service time is between $T_R + T_P$ and T_P, the augment of T_R will lead to the service time $(1/\mu_S)$ increasing, which results in a lower T_I in (9). At the same time, from (10), if T_I decreases, P_R decreases. In other words, the successful rate of the SU receiver decreases. However, if we decrease T_R, the successful rendezvous rate of the SU sender $(\frac{T_R}{T_M})$ decreases. Thus, T_R is a tradeoff parameter.

Moreover, even if there is a mechanism that can fairly distributed the traffic of a SU, i.e., $\lambda_T = \lambda_R P_R$, in such a situation,

$$\begin{aligned}\rho = \frac{\lambda_S}{\mu_S} &= \frac{\lambda_T}{1/\left((1-\rho)T_P+\rho(T_P+T_R)\right)} \\ &= \frac{\lambda_T T_P}{1-\lambda_T T_R},\end{aligned} \tag{11}$$

where $\lambda_T T_P < 1$. If $T_P + T_R > \frac{1}{\lambda_T}$ (a usual case in existing CH algorithms), then $\rho > 1$, which will definitely generate an unstable system with indelible congestion.

In summary, based on the above analysis, existing CH sequence designs may incur congestion at each SU. A proper upper bound of T_R (STTR) should be chosen for the balance of congestion control and rendezvous performance requirements.

III. PSA Design

According to the analysis above, we propose a practical self-adaptive (PSA) rendezvous protocol including collision avoidance, delay compensation, and an optimal STTR.

A. Collision Avoidance

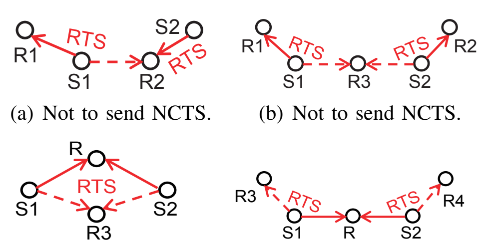
Based on the factors in Section II-B, collisions during a rendezvous process can be classified into two types: one is the collision between RTS and data packets (from both PUs and SUs); the other is the collision between RTSs themselves.

The first type of collisions can be eliminated entirely by setting a current-channel-availability checking period at the beginning of each time slot. In this period, a SU senses the channel to make sure that the current channel is not occupied by another PU or SU. If the channel is not available, the SU does not send an RTS on this channel to avoid the collision between the RTS and the data. Another benefit is that SUs can tell the PU/SU data transmission from a collision, since the RTS collision only happens in the later period. Nevertheless, this mechanism can only avoid data-RTS collisions. The influence of channel status changing on rendezvous successful rate and delay still exists. To address the influence issue, we give a corresponding approach in the next subsection.

The collision between RTSs cannot be eliminated entirely in a proactive way due to the property of blind rendezvous. However, we can design a proper response mechanism so that the impact of RTS collisions can be limited.

1) Challenges: Unlike traditional wireless networks, the SU sender cannot tell whether the target receiver is on the same channel or not until it receives a CTS from the receiver. Thus, if a sender sends an RTS and receives nothing during the listening period, the sender cannot confirm whether it is because of an RTS collision or because that the sender and receiver have not rendezvoused. Therefore, the sender cannot respond as in traditional networks to resend the RTS after a random backup time to avoid further collisions.

2) Proposed Mechanism: We propose a corresponding reaction for the receiver to send Not-Clear-to-Send (NCTS) to notify the senders. However, in our design, we should also be careful of collisions between NCTS and CTS. In Fig. 6, the solid line represents the target RTS transmission and the dashed line represents RTS overhearing. An NCTS should not be sent (by R2 and R3) in the two cases shown in Fig. 6(a) and (b) because at least one RTS is not destined to the victim receiver. Otherwise, a collision between CTS and NCTS will happen: in Fig. 6(a), S1 cannot hear the CTS from its target receiver R1 due to a collision with R2's NCTS; in Fig. 6(b), both rendezvous of S1 and S2 will be ruined by R3's NCTS. On the other hand, cases in Fig. 6(c) and Fig. 6(d) need an NCTS to notify the senders since they are competing for the same receiver. All other scenarios of possible RTS collisions in a complicated network are the combination of these four cases.



(a) Not to send NCTS. (b) Not to send NCTS.

(c) Send NCTS. (d) Send NCTS.

Fig. 6. Different cases for RTS collisions and sending NCTS or not.

Now, how can a SU receiver tell which RTS collision case has happened? We propose a simple yet effective mechanism for the receiver to judge whether to send an NCTS or not after an RTS collision. That is, the first time hearing a collision on a channel during the CH, the receiver takes no action and keeps hopping. The next time the receiver hops to this channel, if it finds the channel becomes unavailability during the checking period, it means that the prior collision accords with the first two cases with a high probability and now the sender is in the transmission period. If it still hears a collision, it means that either case 3 or case 4 happened and may keep happening with a high probability. Thus, this time the receiver chooses to send out an NCTS and stays on this channel to wait for a potential RTS in the near future.

3) Benefits: Under our proposed mechanism, the senders have an opportunity to implement the stay-and-resend scheme just as in traditional wireless networks. At the same time, by adding the reasoning function to receivers, we ensure the sender that its target receiver stays on the same channel and waits for the RTS.

B. Delay Compensation

As mentioned previously, only avoiding collisions cannot mitigate the impact of channel status changing on the rendezvous delay or failure. Thus, we further design a reaction once a SU realizes the change of a channel status.

1) Main Idea: By analyzing the probabilistic models in Section II, we conclude that a lower T_R can benefit the performance under all factors. In addition, as mentioned in Section I, T_R has a correlation with the total number of channels in the network. Hence, if the available channel set of each SU is downsized, a shorter rendezvous time can be achieved. This method has also been proposed in [24]. However, it lacks a rigorous way to downsize the available channel set, in order to avoid excluding common available channels due to the blind information and dynamic environment. On the other hand, the problem will not exist in our proposed protocol because we propose to only cut those channels that become unavailable during the CH. In this way, we can compensate the increased hopping delay caused by the status change of those channels.

2) Motivating Example: We conduct simulations to justify our idea. We use the same parameters and CH sequence as in [24] and assume that a SU knows which channel becomes unavailable during the CH. The TTR and rendezvous successful rate of no adjustment and cut-channel reaction are compared in Table II.

TABLE II. NO ADJUSTMENT VS. CHANNEL REMOVAL

Metrics	A common available channel changes		An uncommon available channel changes			
	No Act	Cut	No Act	Tx. cut	Rx. cut	Both cut
ETTR	4.2 [1.0]	2.2 [0.7]	2.7 [0.8]	2.4 [0.8]	2.4 [0.7]	2.1 [0.7]
MTTR	9.1 [3.0]	3.9 [1.9]	5.5 [2.3]	4.6 [2.2]	4.5 [2.1]	3.8 [1.9]
Suc.	80.15%		100%			

All the numbers in Table II without brackets are the average values and their standard deviation is shown in brackets. From the table we can conclude: *(i)* cutting a common available channel cannot save those lost successful rate, but can mitigate rendezvous delay; *(ii)* if an uncommon available channel has been cut, the successful rate is not affected, but the rendezvous delay is significantly reduced.

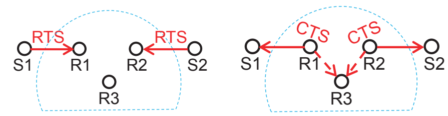
3) Proposed Mechanism: To confirm a channel status change during the CH, besides carrier sensing in the checking period, there are another cases to be considered. If a sender or receiver overhear a CTS from other SU, it is a clear indication that current channel will not be available in a near future. Hence, the sender/receiver will cut this channel directly to downsize the available channel set. Another case is that a SU receiver can also infer that its current channel will become unavailable and cut the channel when overhearing a collision, no matter the collision is generated by CTSs, CTS and NCTS, or NCTSs. We use the CTS collision case to explain the reasoning process. We first prove that only a receiver needs this function.

Lemma 2. *A sender will never overhear an unrelated CTS during the CTS period of the CH.*

Proof: Assume a sender can hear an unrelated CTS, which means that an unrelated receiver has successfully received an RTS from a corresponding sender in the prior period. Since the unrelated receiver is also in the sensing range of the considered sender, the receiver should have also heard the RTS from the considered sender. However, the receiver cannot hear both RTSs due to the collision, which contradicts the assumption. ■

Lemma 3. *A sender cannot hear CTS collisions during the CH.*

Proof: Assume a sender can hear CTS collisions, which means that there are at least two closeby receivers who have sent a CTS at the same time. Since one CTS is only for one target sender, which infers at least one CTS is an unrelated CTS to the sender. This contradicts Lemma 2. ■



(a) The RTS period. (b) The CTS period.

Fig. 7. A case of a CTS collision.

A CTS collision indicates that at least two receivers have successfully received the RTS from their senders and a CTS collision happens at R3 as shown in Fig.7(b). Since both R1 and R2 will transmit on this channel from the next time slot, indicating in a long period this channel is not available, R3 can cut this channel for downsizing.

The CTS-NCTS collision and NCTS collision cases can be analyzed in the same way and reach the same conclusion. The detailed judge-and-reaction algorithm is shown in Section III-D.

C. STTR Optimization

If we remove the role preassignment assumption, the derivation of ρ in (11) in Section II is

$$\rho = \frac{\lambda_R P_R T_R}{1 + \lambda_R P_R T_R - \lambda_S (T_P + T_R)} \qquad (12)$$

Combined with (9), it is very challenging to get an explicit expression of ρ. The parameters such as Ks and λ_R are hard to know during the blind rendezvous process, or change frequently. However, the potential traffic rate from outside of a SU should be similar to the SU's internal traffic rate when the network is controlled in a stable status. Thus, we modify the congestion model in Fig. 5 to better represent both the SU's status and its neighbors'as shown in Fig. 8.

Since it is a loss-tolerant model, we replace ρ with throughput efficiency (TH) in our optimization. A successful packet transmission process begins from the buffer in the sender.

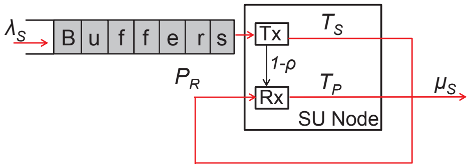


Fig. 8. The optimization model for congestion.

A packet waits for its turn in the buffer to be transmitted. Once the packet is taken from the queue, the sender has to rendezvous with the target receiver in a limited time, which is STTR, to avoid packet congestion. The packet has a probability P_R to be successfully received by the target receiver. Finally, after a successful packet transmission, this packet can be counted as in the throughput between the SU pair. Therefore, $TH = (1-\rho)P_R$.

We can maximize the TH by optimizing the STTR (T_S):

$$\begin{aligned} \underset{T_S}{\text{Maximize}} \quad & (1-\rho)P_R \\ \text{subject to} \quad & P_R = Pr[N(T_S)=0]\frac{T_S}{T_M}, \\ & \rho = \lambda_S(1-\rho)P_R(T_S+T_P) \leq 1. \end{aligned}$$

We denote the ratio of T_S to T_M as the rendezvous successful rate of two SUs within T_S under the ideal situation. For convenience, we replace λ_S with λ.

After countervailing ρ, the expression for TH is

$$TH = \frac{Pr[N(T_S)=0]T_S}{T_M + \lambda Pr[N(T_S)=0]T_S(T_S+T_P)}. \tag{13}$$

To maximize the TH,

$$\frac{d(TH)}{d(T_S)} = 0 \Longrightarrow (1-\lambda T_S)T_M = \lambda e^{-\lambda T_S} T_S^2. \tag{14}$$

Using Taylor expansion,

$$e^{-\lambda T_S} = \frac{1}{\lambda T_S + (\lambda T_S)^2/2 + (\lambda T_S)^3/6 + \ldots}$$

Since λT_S is a value less than 1, we keep the Taylor expansion to the second degree. Thus, the expression for T_S is

$$\lambda^2 T_S^2 + \left(\lambda + \frac{2}{T_M}\right) T_S - 2 = 0. \tag{15}$$

Let $b = \lambda + \frac{2}{T_M}$, finally we have

$$T_S = \frac{-b + \sqrt{b^2 + 8\lambda^2}}{2\lambda^2}. \tag{16}$$

In (16), only two parameters are needed to calculate the optimal STTR: λ_S and $MTTR$. MTTR is a theoretical value pre-known and λ_S can be easily obtained by simply observing a SU's own history. We can further downsize the number of channels to $\sqrt{STTR}$ (keep the first $\sqrt{(STTR)}$ available channels and cut the remaining channels). Thus, the theoretical successful rate is equal to 1 ($MTTR \sim N^2$). This is based on the study of high value γ. In this way, we can dynamically update the optimal STTR to adapt to the changing environment and obtain the maximum throughput efficiency without congestion.

D. Integrated Protocol

We design algorithms for the SU sender and receiver in one SU compositely from Algorithm 1-5. Senders and receivers reason the most possible situation and give corresponding actions. During the execution of the algorithms, they update the optimal STTR to achieve the maximum throughput with congestion control.

The collision types and different periods in one time slot are only analyzed for explaining possible cases and supporting SUs' reactions. For a SU, it is not necessary to know the collision types or to be placed in a strict time synchronization system. For a sender, if it overhears a collision on a channel before sending an RTS, it removes the channel from its available channel set; if it hears a collision after sending an RTS, it takes no action and keeps hopping. For a receiver, if it hears a collision after the CSMA period, i.e., it hears a collision not from the beginning when it hops to a new channel, the receiver ignores it unless it happens again the next time on the same channel. For the latter case, it sends an NCTS. On the other hand, once a receiver hears a collision before leaving a channel, it removes the channel from its available channel set.

Algorithm 1: The integrated protocol for a SU.

Input: length of buffered packets q, current slot t, packet arrival time t', trans. flag $Flag_T$, rescan time period T_{set}, total number of channels N, and set of interarrival times $[int]$.
Output: q, t, $Flag_T$, and $[int]$.
while $mod(t,\ T_{set}) = 0$ **do** ; /* rescan */
  generate available channel set $[Avai_{ch}]$;
  generate CH sequence seq and current channel Ch_i;
while *new packet arrives* **do**
  $q \leftarrow q+1$; $int \leftarrow [int,\ t-t']$; $t' \leftarrow t$;
while *transmission finishes* **do**
  $Flag_T \leftarrow 0$; $q \leftarrow q-1$;
if $Flag_T = 1$ **then**
  transmission;
  else if $q > 0$ **then** ; /* sender */
    Algorithm 2;
    else ; /* receiver */
      Algorithm 4;
$t \leftarrow t+1$;

IV. Performance Evaluation

We conduct simulations to evaluate the performance of our proposed PSA protocol. In our simulation, we do not impose any impractical assumptions explained in Section I which are widely adopted in most of other existing CRN papers. In our simulation, 1) PUs and SUs are randomly distributed in the simulation area, so the number of neighboring PUs/SUs is different for each SU; 2) Each PU is randomly assigned a channel when a new packet needs to be transmitted; and 3) Packet arrivals follow the Poisson distribution and the target receiver of each packet is also randomly chosen, so the rendezvous pair changes dynamically. The parameters used in our simulation are listed in Table III.

Due to the lack of similar complete communication framework in existing papers, we choose the QoS-based CH design

Algorithm 2: A sender ($x1$) sending period.

Input: q, T_R, $[int]$, backup counter T_{bp}, backup time bp, Ch_i, and $[Avai_{ch}]$.
Output: $RTS_{x1\to y1}$, $Avai_{Ch}$, q, T_R, and T_{bp}.
while Ch_i *being used* **do** ; /* CSMA */
  cut Ch_i from $Avai_{ch}$; wait for the next hop;
$\lambda_S \leftarrow 1/mean(int)$; $MTTR \leftarrow N^2$;
Calculate T_S using (17); Transfer to STTR;
Downsize $[Avai_{ch}]$ to $\sqrt{STTR}$;
while $T_R = T_S$ **do** ; /* congestion control */
  $q \leftarrow q-1$; $T_R \leftarrow 0$; $T_{bp} \leftarrow 0$;
  Algorithm 1;
while $T_{bp} = 0$ *or* $T_{bp} = bp$ **do** ; /* not backup */
  send $RTS_{x1\to y1}$;
Algorithm 3;

Algorithm 3: A sender ($x1$) listening period.

Input: CTS, $NCTS$, T_R, T_{bp}, bp, and seq.
Output: $Flag_T$, T_R, T_{bp}, bp, and Ch_i.
if CTS **then** ; /* begin trans. */
  $Flag_T \leftarrow 1$; $T_R \leftarrow 0$; $T_{bp} \leftarrow 0$;
  else if $NCTS$ **then** ; /* backup-resend */
    $T_{bp} \leftarrow T_{bp}+1$; $T_R \leftarrow T_R+1$; $bp \leftarrow rand(4)$;
    else ; /* Nothing received */
      $T_R \leftarrow T_R+1$; $Ch_i \leftarrow Ch(seq+1)$;
Algorithm 1;

Algorithm 4: A receiver ($y1$) listening period.

Input: RTS, T_R, T_{bp}, collision counter $C(Ch_i)$, and T_{bp}.
Output: CTS, $NCTS$, $Flag_T$, T_R, T_{bp}, bp, and $C(Ch_i)$.
while Ch_i *being used* **do** ; /* CSMA */
  cut Ch_i from $Avai_{ch}$; wait for the next hop;
if $RTS_{\to y1}$ **then** ; /* begin trans. */
  $Flag_T \leftarrow 1$; $T_R \leftarrow 0$; $T_{bp} \leftarrow 0$; $C(Ch_i) \leftarrow 0$;
  send CTS_{y1};
  else if $RTS_{\to !y1}$ **then**
    $C(Ch_i) \leftarrow 0$; Algorithm 5;
    else
      if *Collision* **then**
        $C(Ch_i) \leftarrow C(Ch_i)+1$;
        if $C(Ch_i) = 2$ **then** ; /* related RTS collision */
          send $NCTS$; $C(Ch_i) \leftarrow 0$; $T_{bp} \leftarrow 1$;
          Algorithm 5;
          else
            Algorithm 5;

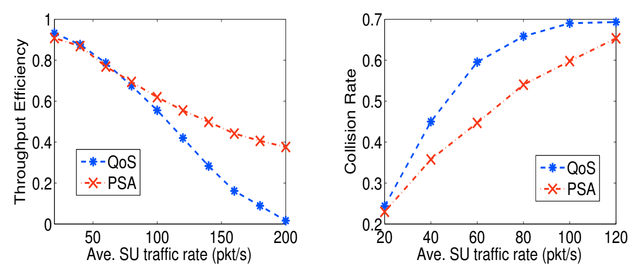
in [24] which is a realistic design based on few assumptions and can achieve good rendezvous performance in terms of high successful rate and low TTR. We will show that even this CH-sequence generating algorithm, after equipped with our proposed framework, still has a considerable space for improvement when being applied to a realistic CRAHN. In this paper, we investigate the following three performance metrics: 1) *Collision Rate*: the probability that a SU has a collision in one time slot; 2) *ETTR*: the average duration from the moment a CH starts to the moment a CTS message is received; and 3) *Throughput Efficiency*: the average rendezvous successful rate of the whole network.

Algorithm 5: A receiver ($y1$) sending period.

Input: CTS, $NCTS$, $Avai_{Ch}$, Ch_i, T_{bp}, and seq.
Output: $Avai_{Ch}$, Ch_i, $C(Ch_i)$, and T_{bp}.
if CTS *or* $NCTS$ *or* $Collision$ **then**
  cut Ch_i from $Avai_{ch}$; $T_{bp} \leftarrow 0$; $C(Ch_i) \leftarrow 0$; wait for the next hop;
  else
    while $T_{bp} = 0$ **do**
      $Ch_i \leftarrow Ch(seq+1)$
Algorithm 1;

TABLE III. SIMULATION PARAMETERS

Average PU packet arrival rate	$50pkt/s$
Number of PUs	100
Number of SUs	50
PU packet size	50 slots
Simulation time	10000 slots
Simulation area	50 m $\times$ 50 m
PU sensing radius	10 m
SU sensing radius	7 m
Channel data rate	2 Mbps
The size of (RTS+CTS)	300 bits



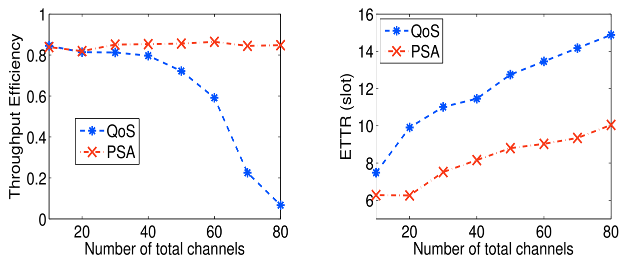
(a) Throughput Efficiency. (b) Collision Rate.

Fig. 9. Performance with λ_S changing. ($N = 15$, $L_P = 20$ slots)

Fig. 9(a) illustrates the throughput efficiency (or, the real successful rate) of the whole network under different traffic conditions of SUs. When the average SU packet arrival rate increases, the throughput efficiency of both protocols decreases, while our PSA protocol can still achieve a higher throughput due to the congestion control and TH maximization scheme. On the other hand, as shown in Fig. 9(b), the collision rate of both protocols increases when activating more SUs in the network. Nevertheless, our protocol enjoys less collision rate since our proposed collision response mechanism can help SUs avoid further collisions.

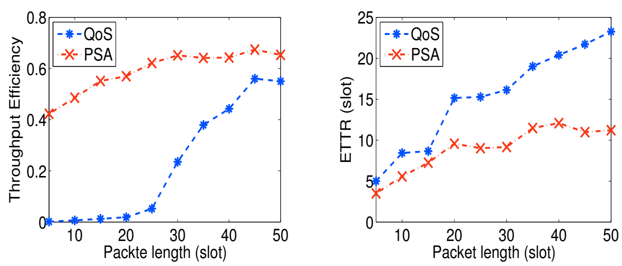
Fig. 10(a) shows the impact of the number of total channels. When the number of channels increases, the successful rate of the QoS scheme decreases. Though the QoS scheme has the available channel set downsize mechanism, it lacks a rigorous method to adapt to the changing channel number. However, our PSA protocol can deal with this issue very well, since our protocol can intelligently downsize the number of available channels. Fig. 10(b) shows the comparison of the ETTR. We can see that even only counting those successful rendezvous cases, our PSA protocol has a shorter rendezvous time because of our proposed delay compensation scheme.

Fig. 11 shows the impact of SU's packet size. In common cases, the packet size of SUs is between 10 to 25 time slots which is shorter than the TTR in most scenarios. We already analyzed in (12) that this size will lead to a congestion system. As shown in Fig. 11(a), when the SU packet size is 5 to 10 time slots, traditional rendezvous protocols have an overwhelming probability to fail the rendezvous. When the packet size of SUs increases, both protocols gain better performance. However, the longer transmission time of one packet will increase the probability of spectrum handoff during a data packet transmission and impact the performance of data transmissions. We will study this tradeoff parameter in the future work.



(a) Throughput Efficiency. (b) ETTR.

Fig. 10. Performance with the number of channels changing. ($\lambda_S = 50pks/s$, $L_P = 20$ slots)



(a) Throughput Efficiency. (b) ETTR.

Fig. 11. Performance with packet length changing. ($N = 10$, $\rho = 0.4$)

V. Conclusion

In this paper, the challenges of practical rendezvous in CRAHNs have been addressed for the first time. A fully self-adaptive rendezvous protocol is proposed without imposing impractical assumptions. By intelligently reasoning the possible scenarios and adjusting parameters dynamically, our proposed protocol can provide high throughput efficiency with collision avoidance and congestion control while achieving a short rendezvous delay. In addition, we also proposed probabilistic models for analyzing the channel-status-changing problem. Simulation results show that our proposed protocol outperforms existing most-close-to-practical scheme in terms of higher throughput efficiency, shorter TTR, and lower collision rate under different network environments.

References

[1] L. A. DaSilva and I. Guerreiro, "Sequence-based rendezvous for dynamic spectrum access," in *Proc. IEEE Symposium on New Frontiers in Dynamic Spectrum Access Networks (DySPAN)*, 2008.

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

[3] J. Shin, D. Yang, and C. Kim, "A channel rendezvous scheme for cognitive radio networks," *IEEE Communications Letters*, vol. 14, no. 10, pp. 954–956, 2010.

[4] C.-F. Shih, T. Y. Wu, and W. Liao, "DH-MAC: A dynamic channel hopping MAC protocol for cognitive radio networks," in *Proc. IEEE International Conference on Communications (ICC)*, 2010.

[5] K. Bian, J. Park, and R. Chen, "A quorun-based framework for establishing control channels in dynamic spectrum access networks," in *Proc. ACM MobiCom*, 2009, pp. 25–36.

[6] C. Xin, M. Song, L. Ma, and C.-C. Shen, "Performance analysis of a control-free dynamic spectrum access scheme," *IEEE Trans. Wireless Communications*, vol. 10, no. 12, pp. 4316–4323, 2011.

[7] L. Jiao and F. Y. Li, "A single radio based channel datarate-aware parallel rendezvous MAC protocol for cognitive radio networks," in *Proc. IEEE Conference on Local Computer Networks (LCN)*, 2009, pp. 392–399.

[8] M. Altamimi, K. Naik, and X. Shen, "Parallel link rendezvous in ad hoc cognitive radio networks," in *Proc. IEEE Global Telecommunications Conference (Globecom)*, 2010.

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

[10] H. Liu, Z. Lin, X. Chu, and Y.-W. Leung, "Ring-walk based channel-hopping algorithms with guaranteed rendezvous for cognitive radio networks," in *Proc. IEEE/ACM Int'l Conference on Cyber, Physical and Social Computing (CPSCom)*, 2010, pp. 755–760.

[11] Y. Song and J. Xie, "ProSpect: A proactive spectrum handoff framework for cognitive radio ad hoc networks without common control channel," *IEEE Trans. Mobile Computing*, vol. 11, no. 7, 2012.

[12] C. Cordeiro, K. Challapali, and M. Ghosh, "Cognitive PHY and MAC layers for dynamic spectrum access and sharing of tv bands," in *Proc. First Intl Workshop Technology and Policy for Accessing Spectrum (TAPAS)*, 2006.

[13] Q. Liu, D. Pang, G. Hu, X. Wang, and X. Zhou, "A neighbor cooperation framework for time-efficient asynchronous channel hopping rendezvous in cognitive radio networks," in *Proc. IEEE International Symposium on Dynamic Spectrum Access Networks (DYSPAN)*, 2012, pp. 529–539.

[14] K. Bian and J.-M. Park, "Asynchronous channel hopping for establishing rendezvous in cognitive radio networks," in *Proc. IEEE INFOCOM*, 2011, pp. 236–240.

[15] R. Gandhi, C.-C. Wang, and Y. C. Hu, "Fast rendezvous for multiple clients for cognitive radios using coordinated channel hopping," in *Proc. IEEE Conference on Sensor, Mesh and Ad Hoc Communications and Networks (SECON)*, 2012, pp. 434–442.

[16] J. Kim, Y. Baek, J. Yun, K. Cho, K. Lee, and J. Han, "A repeated group sequence rendezvous scheme for cognitive radio networks," in *Proc. Spring Congress on Engineering and Technology (S-CET)*, 2012.

[17] G.-Y. Chang, W.-H. Teng, H.-Y. Chen, and J.-P. Sheu, "Novel channel-hopping schemes for cognitive radio networks," *IEEE Trans. Mobile Computing*, 2012.

[18] M. Chowdhury, A. Asaduzzaman, M. F. Kader, and M. O. Rahman, "Design of an efficient MAC protocol for opportunistic cognitive radio networks," in *International Journal of Computer Science and Information Technology (IJCSIT)*, 2012.

[19] M. Juang, K.-C. Wang, and J. Martin, "A measurement study on link capacity of a high stress IEEE 802.11b/g network," in *Proc. International Conference on Computer Communications and Networks (ICCCN)*, 2008, pp. 1–6.

[20] T. Sugimoto, N. Komuro, H. Sekiya, S. Sakata, and K. Yagyu, "Maximum throughput analysis for RTS/CTS-used IEEE 802.11 DCF in wireless multi-hop networks," in *Proc. International Conference on Computer and Communication Engineering (ICCCE)*, 2010, pp. 1–6.

[21] S.-T. Sheu and T.-F. Sheu, "DBASE: a distributed bandwidth allocation/sharing/extension protocol for multimedia over IEEE 802.11 ad hoc wireless LAN," in *Proc. IEEE INFOCOM*, vol. 3, 2001, pp. 1558–1567.

[22] S. Yin, D. Chen, Q. Zhang, and M. Liu, "Mining spectrum usage data: A large-scale spectrum measurement study," *IEEE Trans. Mobile Computing*, 2012.

[23] L. Kleinrock, Ed., *Queueing Systems Volume I: Theory*. New York: A Wiley-Interscience Publication, 1975.

[24] Y. Song and J. Xie, "A QoS-based broadcast protocol for multi-hop cognitive radio ad hoc networks under blind information," in *Proc. IEEE Global Telecommunications Conference (Globecom)*, 2011.