

The Life Cycle of the Rendezvous Problem of Cognitive Radio Ad Hoc Networks: A Survey

Zaw Htike, Choong Seon Hong*, and Sungwon Lee

Department of Computer Engineering, Kyung Hee University, Yongin, Korea
htike@khu.ac.kr, cshong@khu.ac.kr, drsungwon@khu.ac.kr

Abstract

In cognitive radio or dynamic spectrum access networks, a rendezvous represents meeting two or more users on a common channel, and negotiating to establish data communication. The rendezvous problem is one of the most challenging tasks in cognitive radio ad hoc networks. Generally, this problem is simplified by using two well-known mechanisms: the first uses a predefined common control channel, while the second employs a channel hopping procedure. Yet, these two mechanisms form a life cycle, when they simplify the rendezvous problem in cognitive radio networks. The main purpose of this paper is to point out how and why this cycle forms.

Category: Ubiquitous computing

Keywords: Rendezvous problem; Common control channel; Channel hopping; Problem life cycle

I. INTRODUCTION

In recent years, many research works have been involved in the development of cognitive radio (CR) technology, since it has been recognized as a new way to improve the spectral efficiency of wireless networks. In CR networks, cognitive users (often called secondary users [SUs]) are allowed to use free portions of the licensed spectrum or idle channels, without causing any interference to primary users (PUs) [1]. In general, SUs detect the free or idle portions of a channel, and access the channel. When the PU appears on the channel that is currently used by SUs, all SUs must defer their transmissions, and migrate to other available channels [2]. The availability of channels is determined by PU activities, which change dynamically in frequency, space and time; therefore, the set of available channels for each SU might also change dynamically [3]. Thus, at a given time, SUs may operate on dif-

ferent channels independently, as shown in Fig. 1.

However, if a pair of SUs wishes to communicate with each other, they need to rendezvous on a channel that is commonly available to both of them, and exchange control information, such as request-to-send (RTS)/clear-to-

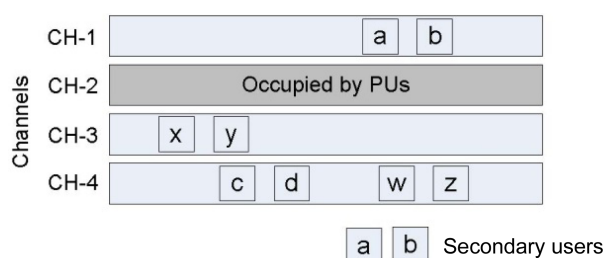


Fig. 1. At any given time, secondary users may operate on different channels independently. PU: primary user.

Open Access <http://dx.doi.org/10.5626/JCSE.2013.7.2.81>

<http://jcse.kiise.org>

This is an Open Access article distributed under the terms of the Creative Commons Attribution Non-Commercial License (<http://creativecommons.org/licenses/by-nc/3.0/>) which permits unrestricted non-commercial use, distribution, and reproduction in any medium, provided the original work is properly cited.

Received 15 February 2013, Accepted 3 May 2013

*Corresponding Author

send (CTS) packets of the distributed coordination function (DCF) of IEEE 802.11 [4]. This task may refer to network coordination in classical wireless networks. The major challenge in CR networks is not coordinating the network, but rather rendezvousing on a common channel. This task is not trivial, since SUs may operate on different channels independently, which gives rise to the *rendezvous problem* of CR networks. Note that this problem is more challenging in CR ad hoc networks, as there is no centralized coordinator.

Currently, there are two well known approaches to enable a rendezvous in CR ad hoc networks: 1) Using a dedicated common control channel (CCC), and 2) using channel hopping.

However, each approach introduces new problems, when they are used to simplify the rendezvous problem. Moreover, these two solutions form a problem life cycle. The main goal of this paper is to point out how and why this cycle is formed. Note that in this paper, we narrow the scope to focus solely on the rendezvous problem of CR networks, with an assumption that every SU is equipped with a single transceiver (radio)—this is an important assumption, because if we consider multi-radio communication, the rendezvous of SUs becomes a simple task. However, using multiple transceivers is neither a cost effective, nor a practical assumption. First, we briefly discuss various pros and cons of each approach, and highlight how and why these two approaches form a life cycle.

II. SOLUTIONS FOR RENDEZVOUS PROBLEM

Many works have addressed the rendezvous problem and have tried to simplify it in different ways. Nonetheless, we categorize these existing solutions into two major groups, based on their operations. We categorize any protocol that enables a rendezvous, by using a dedicated control channel as a *common control channel approach*. Conversely, any protocol that enables a rendezvous without using a CCC is considered part of the *channel hopping approach*.

A. Common Control Channel Approach

The simplest way to solve the rendezvous problem is to use a predefined CCC. Most of the proposed medium access control (MAC) protocols for CR networks were designed by assuming the existence of a CCC, and further assuming that it is available for every SU [5-7].

In fact, this approach originated from the concept of MAC protocols for multi-channel wireless networks [8]. In this approach, one of the available channels is assigned as the CCC, and it serves as a rendezvous channel [9]. All the necessary control information is exchanged among

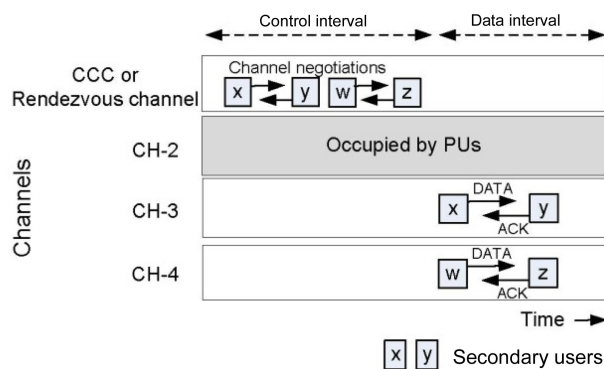


Fig. 2. The basic procedure of rendezvous, channel negotiation and data communication using a common control channel (CCC). PU: primary user.

SUs via the CCC. In this approach, time is divided into two intervals; negotiation, or control interval and data interval. When an SU wants to initiate communication, it first switches to the CCC during the control interval, and attempts to negotiate with the intended receiver or neighbor. After negotiating on the CCC, data communication can be accomplished during the data interval via other available channels, known as data channels. Thus, this approach guarantees that all SUs can rendezvous on CCC, during the control interval.

Fig. 2 illustrates the normal operation of a network with a CCC. As shown in Fig. 2, channel-1 is assigned as CCC, and the rest are dedicated for data communications. All SUs attempt to negotiate on the CCC during the control interval. After the negotiation is completed on the CCC, the SUs move to selected channels, and perform data communications simultaneously during the data interval.

Obviously, using a CCC makes MAC protocols simple and efficient; yet it is often not feasible, or is impractical, due to the following reasons:

1) Lack of CCC Availability

The main drawback of using the CCC approach is that it is susceptible to PU activities. When a PU appears on a CCC, all SUs must defer their transmissions on the CCC, and vacate the channel immediately. Not only does their presence degrade the overall throughput of a CR network, but if a PU transmission period is significantly long on a CCC, the presence of the PU may also block channel access for SUs. Moreover, the available channel sets in CR networks, including the CCC, change dynamically, hindering the establishment of an ever-available control channel for all SUs.

The authors of [10] addressed this problem, by proposing a control channel recovery mechanism. In [10], SUs store a common channel list (CCL), which is updated periodically. If the current CCC is unavailable—e.g., in

the case of PU appearances—SUs select the best channel from the CCL, and use it for a new CCC. Although this process can recover a CCC within a reasonable time interval, it suffers a control channel saturation problem.

Dynamic CCC selection, instead of using dedicated CCC, called CogMesh, was proposed in [11]. In CogMesh, SUs need to collect the information of network topology and neighboring SUs. An SU initiates cluster forming, by selecting a local available channel as CCC, and the SU becomes the cluster head. Neighboring SUs, if the selected CCC is available for them, can join the cluster. Obviously, selecting local available CCC is much easier than selecting the channel that is available to all SUs at the same time. However, since the SUs use local CCC, global network coordination is difficult to achieve. Moreover, it raises other problems, such as neighbor discovery, and topology management.

2) Control Channel Saturation Problem

The principle of the CCC approach forces SUs to transmit all control packets on the CCC. Thus, the collision rate of control packets is high when the number of users in the network is large, since all users use only one channel for negotiation. This procedure obviously degrades the overall throughput of the network, since data transmissions only follow successful negotiations on the CCC. Accordingly, the authors of [12] suggested adjusting the bandwidth of the CCC to reduce the possibility of saturation; however, this solution is not always feasible, since the bandwidth is predetermined, and is usually the same as the data channel bandwidth.

3) Security

Similarly, using a static CCC creates security vulnerability, since it might encourage adversaries to launch powerful attacks, such as control channel jamming attacks [13]. Control channel jamming is the easiest and most effective way to interrupt a network system. The attacker simply needs to inject a strong interference signal into the CCC, in order to disable any control packet exchange on the CCC. This attack will cause a single point of failure within the network.

B. Channel Hopping Approach

The second famous approach to simplify the rendezvous problem in CR ad hoc networks is using channel hopping (also known as sequence-based protocols). The common goals of most of the existing sequence-based protocols are to overcome the drawbacks of a CCC, and to eliminate the need of a CCC. In channel hopping approaches, SUs generate their own channel hopping sequences [14]. When an SU (e.g., a sender) needs to communicate with its neighbor (the receiver), it switches from one channel to another, by following a predefined hopping-sequence, until it finds its neighbor [15, 16].

When two SUs meet on a common channel, they need to exchange the necessary control information to complete rendezvous.

The basic procedure of neighbor discovery can be described as follows: when the SU switches to an available channel, it scans the channel for the presence of PUs and other SU transmissions. If it senses the channel free, it will broadcast a beacon. Note that the beacon is a probe message, and it can be a small packet or signal that can indicate the presence of the SU. Different kinds of control packets are used in different MAC protocols, such as a beacon in [17], RTS in [18], and a connection request in [19]. Nonetheless, the common purpose of transmitting the beacon is to indicate the presence of an SU that is willing to communicate, so that if the intended receiver is on the current channel and receives the beacon, it can reply with an acknowledgment (ACK) message. Once these two users have exchanged the control packets (in this case, beacon/ACK), their rendezvous has been accomplished. Otherwise, the SU will switch to another channel by following the hopping sequence, and broadcast the beacon message again. This process is repeated, until the SU (sender) meets with its intended neighbor (receiver). Once two SUs have rendezvoused on an available channel and exchanged control messages, they can perform negotiation for data communication on the current rendezvous channel.

Fig. 3 illustrates the operation of a channel hopping protocol. In Fig. 3, SUs *x* and *y* find each other, by following their own hopping sequences. When they rendezvous on a channel, which is channel-4, they negotiate for data communication.

At first glance, channel hopping approaches appear to compensate for the problems encountered with the CCC. In this approach, SUs can rendezvous on any available channel. Thus, it is more tolerable to PU activities than CCC approaches, and circumvents long-term blocking by PUs. It can also overcome the control channel saturation problem, as control packets are transmitted via multiple channels. Moreover, channel hopping protocols consider

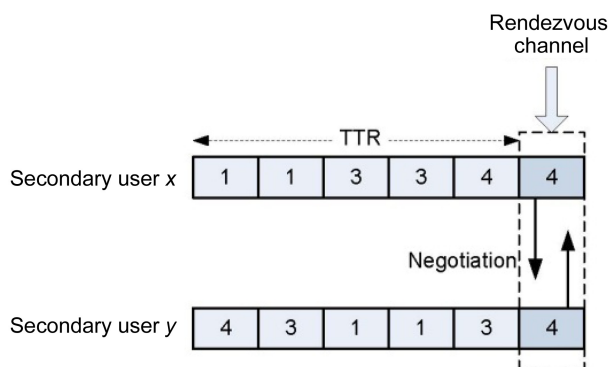


Fig. 3. The basic operation of a channel hopping protocol. TTR: time to rendezvous.

only pair-wise rendezvous connections (i.e., only a sender and a receiver), which do not need a globally available CCC. The problems encountered with control channel jamming attacks have been automatically eliminated, since it does not use any CCC. Nevertheless, channel hopping approaches also introduce the following major problems.

1) Channel Access Delay

In sequence-based protocols, when an SU wants to communicate with its neighbor, it will switch from one channel to another, by following a hopping sequence, until it finds its neighbor. Accordingly, a user needs a significant amount of time to meet with its neighbor, which results in a channel access delay or time to rendezvous (TTR). The value of TTR is typically measured in time slots, and dependent on the channel hopping algorithms.

In [20], a MAC protocol called synchronized MAC (SYN-MAC) was proposed, with the assumption that every node in the network is equipped with two transceivers, one for control signal and one for data communication. In SYN-MAC, channel hopping sequences were created in a round-robin fashion that required tight synchronization among SUs, which is difficult to achieve in an ad hoc environment. In [21], the authors proposed biased pseudo-random sequences. These sequences do not need tight synchronization, but the average TTR may not be bounded. The authors of [22] proposed permutation-based channel hopping sequences. In their proposal, the hopping sequences are the permutation of the number of available channels, N , and the expected time to rendezvous was bounded by a quadratic function of the number of available channels. A quorum-based scheme was proposed in [23], and the authors claimed that rendezvous between any pair of users can occur at least once within N^2 time slots, where N is the number of available channels. The authors of [17] proposed two rendezvous schemes, based on a random algorithm, and an orthogonal sequence-based algorithm. In the random algorithm, an SU wishing to rendezvous its neighbors visits the available channels in random order. The expected TTR values are

$$E_{RA}[TTR] = \frac{N}{p_H} \tag{1}$$

from the random algorithm, and

$$E_{OSA}[TTR] = \frac{N^4 + 2N^2 + 6N - 3}{3N(N+1)} \tag{2}$$

from the orthogonal sequences-based algorithm, where p_H is the probability of a successful handshake, and N is the number of available channels. Clearly, in the sequence-based protocols, the expected TTR values are highly dependent on the number of available channels. The more available channels in the network there are, the larger the expected TTR values. Therefore, in channel hopping pro-

ocols, users may require a significant amount of time to find their neighbors, especially when the number of available channels in the network is large.

2) Complexity

The next difficulty in sequence-based protocols is overcoming the complexity of generating channel hopping sequences. Any channel hopping algorithm should be able to satisfy the following facts.

- When users generate channel hopping sequences, any pair of these sequences should overlap at least once within a sequence period, so that any pair of users who need to communicate can rendezvous.
- The TTR values between any pair of sequences should be reasonable. Note that the TTR value is determined by the channel hopping algorithm described in the previous section.
- It should support unsynchronized channel hopping. This is because, in CR ad hoc networks, there is no practical way to consider two SUs are synchronized, and start the neighbor discovery at the same time.

A simpler way of channel hopping approaches is the jump-stay approach [24]. In this approach, instead of generating complex channel hopping sequences, a user remains in its location (selected channel), while the other exhaustively searches by hopping from one channel to another, thereby guaranteeing rendezvous. This approach also mitigates tight synchronization. However, other undesirable events can possibly occur, such that both users jump and search for each other by hopping, or neither of them searches, and both are in stay mode. For the former case, the maximum TTR value cannot be bounded; and the rendezvous will never occur in the latter case.

In [25], jump-stay is performed by using specific sequences. Thus, users jump (hop) on available channels, and search the neighbors in the jump-pattern, while the neighbors stay and wait on a specific channel in the stay-pattern. However, the main drawback of this protocol is that if different users have different available channels, the expected TTR value will increase exponentially, when the total number of channels increases.

3) Lack of Network Status Information

The most miserable drawback of sequence-based protocols is the lack of information regarding neighbors' communications. Consider the scenario shown in Fig. 4, where there are three available channels, {CH-1, CH-3, CH-4}, and there is no centralized coordinator. Users are currently operating on different channels independently, users a and b are performing data communication on CH-1, c and d are dwelling on CH-3 and CH-4, respectively, while e and f are also performing data communication on CH-4. An SU may have knowledge of communications on the current operating channel by overhearing, but it has no knowledge of the current trans-

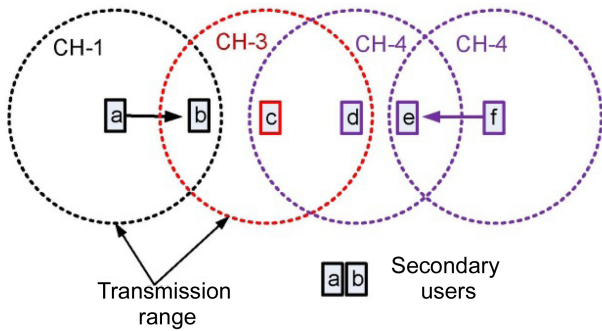


Fig. 4. Secondary users operating on different channels independently.

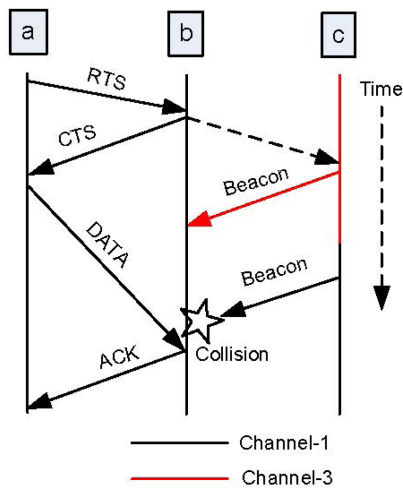


Fig. 5. The multichannel hidden terminal problem. CTS: clear-to-send, RTS: request-to-send, ACK: acknowledgment.

missions on other available channels.

Suppose SU *c* wants to communicate with SU *d*, and let the channel hopping sequence of *c* be {1,3,4,1,3,4,1,3,4}, and that of *d* be {4,4,4,3,3,1,1,1}. SU *c* will visit the channels according to its own hopping sequence, and find the neighboring SU *d*. First, *c* broadcasts the beacon message on CH-1, and will not receive ACK, as there is no user on the channel. Then, *c* switches to CH-3, and senses the presence of PUs and transmission of other SUs on the channel. Although there is current data communication between *a* and *b* on CH-3, SU *c* will detect the channel as free, because it is out of the transmission range of *a*. Therefore, *c* will transmit the beacon message, and it is going to interfere with the data being received by SU *b*, as shown in Fig. 5. SU *b* will not be able to decode the data correctly, and it is called message collision. This scenario represents a *multi-channel hidden terminal problem* [26].

The SU *c* continues neighbor discovery, as it has not received any ACK on CH-3. Therefore, it switches to the CH-4, and senses the presence of PU and other SUs'

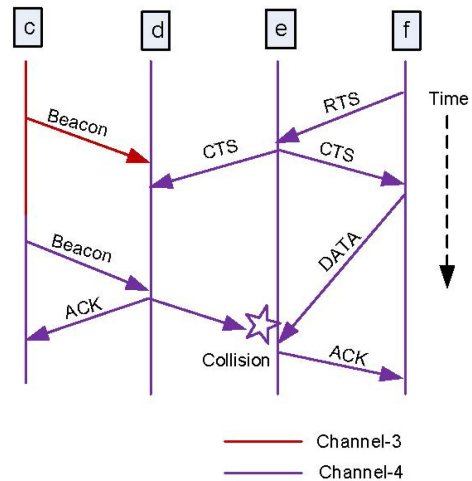


Fig. 6. The multichannel exposed terminal problem. CTS: clear-to-send, RTS: request-to-send, ACK: acknowledgment.

transmissions. Again, *c* senses the channel as free, and broadcasts the beacon message again. In this case, the beacon will be received by *d*, but *d* cannot reply ACK, because if it does, it will interfere with the data being received by SU *e*, as shown in Fig. 6. Consequently, neighbor discovery or rendezvous cannot be accomplished between these two SUs. We will refer to this dilemma as a *multi-channel exposed terminal problem*. These exposed terminal problems are not harmful to current transmissions—although they affect the spectrum efficiency—but hidden terminal problems are quite serious in a broader sense, because they can cause harmful interferences to current transmissions. The main cause of multi-channel hidden terminal problems is the lack of information regarding the current network status.

Moreover, when an SU generates its hopping sequence, all available channels should be included in the sequence. This function is mandatory for almost all channel hopping algorithms; otherwise, they cannot guarantee the rendezvous between a pair of users. Thus, when an SU attempts to rendezvous with its neighbors, it may need to switch to all available channels and broadcast beacon messages. As a consequence, the neighbor discovery of a pair of SUs might interfere with most of the current transmissions in the network. To the best of our knowledge, almost all sequence-based protocols suffer from this multi-channel hidden terminal problem.

The multi-channel hidden terminal problem has been well studied for decades. Currently, the best solution for the multi-channel hidden terminal problem is using a CCC, which can simplify this problem in the following ways: according to the principle of the CCC approach, all users must negotiate on the CCC, before any data communication begins. All neighbor information can easily be broadcast on the CCC, and users can gather the information by overhearing.

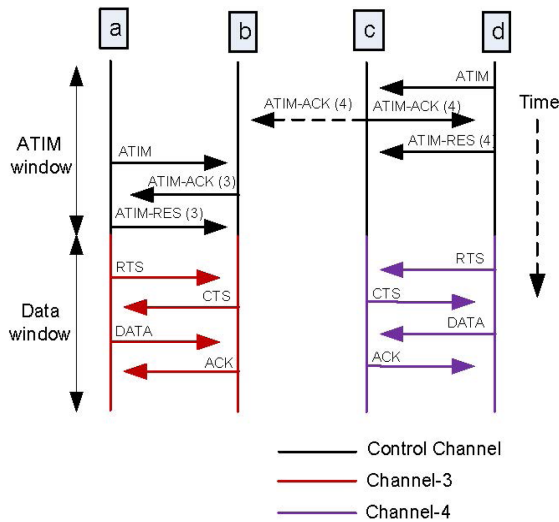


Fig. 7. Process of multichannel medium access control: handling the multichannel hidden terminal problem by exploiting a control channel. ATIM: ad hoc traffic indication message, ACK: acknowledgment, RES: reservation, RTS: request-to-send, CTS: clear-to-send.

A multichannel MAC protocol (MMAC) proposed in [27] might be the best example to show how the multichannel hidden terminal problem is simplified by using a CCC. In MMAC, the network follows the principle of the IEEE 802.11 DCF, and the time is divided into ad hoc traffic indication message (ATIM) windows and data windows (similar to control interval and data interval in CCC approaches). At the ATIM windows, all users switch to the control channel (called the default channel), and nego-

tiate for data communication. The information regarding the upcoming transmission is then embedded in the control packets (in this case, MMAC uses ATIM, ATIM-ACK and ATIM-RES as control packets), which are subsequently exchanged between the sender and the receiver on the control channel. The neighbor nodes that overhear the control packets can determine which channel is going to be used by the neighbors, and how long it is going to remain busy. One-hop neighbors of the sender and/or receiver that overhear the ATIM-ACK and/or ATIM-RES can avoid exploiting the channel, and not interfere in the transmission.

Fig. 7 shows the process of MMAC, and how the multichannel hidden terminal problem is simplified. During ATIM windows, all users hop to control channel, and attempt to negotiate for data communication. Users *c* and *d* choose channel-4 for data communication. User *b* overhears the ATIM-ACK, which indicates that channel-4 is selected by other users; thus *c* realizes that channel-4 will not be available for the next data window. When *a* negotiates with *b* for data communication, *b* chooses channel-3. Therefore, during the data window, users can perform data communication simultaneously on their selected channels. Consequently, this process easily overcomes the multi-channel hidden terminal problems.

Moreover, the neighbor discovery process can be eliminated, by using a CCC. As long as the CCC is available, SUs do not need to perform neighbor discovery. When an SU wants to communicate with its neighbor, it simply switches to the CCC, and attempts to negotiate. As a result, the average TTR value of the CCC approaches is relatively small, compared to that of sequence-based protocols.

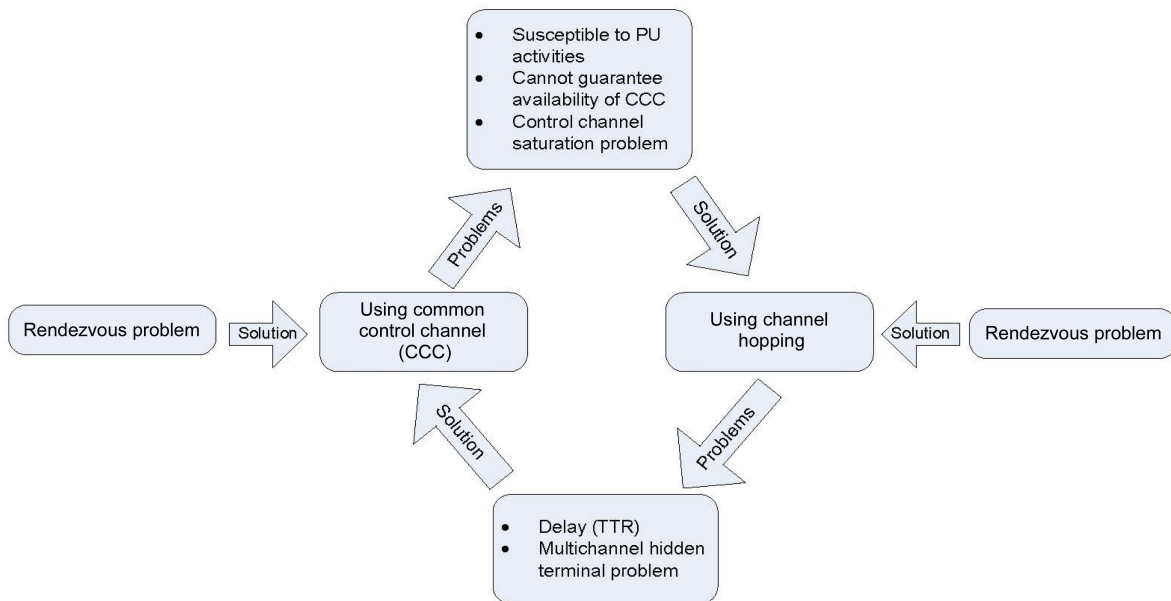


Fig. 8. Problem life cycle of the rendezvous problem of cognitive radio ad hoc networks. PU: primary user, TTR: time to rendezvous.

III. THE LIFE CYCLE

As we mentioned in the previous sections, the two well known solutions for the rendezvous problem of CR networks form a problem life cycle. Fig. 8 illustrates this cycle of CCC approach and sequence-based protocol. The CCC approach can make the protocols simple and efficient, but it cannot guarantee the availability of a CCC. If the CCC is broken—e.g., in the appearance of a primary user—it can induce a single point of failure in the network. Conversely, sequence-based protocols can compensate for problems that the CCC approach experiences due to PU activities; however, sequence-based protocols suffer from the multi-channel hidden terminal problem. The best solution so far for the multi-channel hidden terminal problem is to use a predefined CCC, which has the added benefit of eliminating the neighbor discovery processes, and causing shorter channel access delays. However, the CCC approach is susceptible to PU activities, and so forth.

IV. CONCLUSION

We have presented the problem life cycle of the rendezvous problem of CR ad hoc networks, and we have discussed how and why it is formed. While the rendezvous problem is clearly of fundamental concern, no comprehensive solution yet exists for this problem, at least to the best of our knowledge. Almost all of the proposed solutions suffer from either a lack of availability of a CCC, and/or the multi-channel hidden terminal problem.

ACKNOWLEDGMENTS

This research was supported by the Ministry of Science, ICT & Future Planning (MSIP), Korea, under the Information Technology Research Center support program, supervised by the National IT Industry Promotion Agency (NIPA-2013-H0301-13-2001).

REFERENCES

1. F. Akyildiz, W. Y. Lee, M. C. Vuran, and S. Mohanty, "NeXt generation/dynamic spectrum access/cognitive radio wireless networks: a survey," *Computer Networks*, vol. 50, no. 13, pp. 2127-2159, 2006.
2. A. D. Domenico, E. C. Strinati, and M. G. D. Benedetto, "A survey on MAC strategies for cognitive radio networks," *IEEE Communications Surveys and Tutorials*, vol. 14, no. 1, pp. 21-44, 2012.
3. I. F. Akyildiz, W. Y. Lee, M. C. Vuran, and S. Mohanty, "A survey on spectrum management in cognitive radio networks," *IEEE Communications Magazine*, vol. 46, no. 4, pp. 40-48, 2008.
4. G. Bianchi, "Performance analysis of the IEEE 802.11 distributed coordination function," *IEEE Journal on Selected Areas in Communications*, vol. 18, no. 3, pp. 535-547, 2000.
5. X. Zhang and H. Su, "CREAM-MAC: cognitive radio-enabled multi-channel MAC protocol over dynamic spectrum access networks," *IEEE Journal of Selected Topics in Signal Processing*, vol. 5, no. 1, pp. 110-123, 2011.
6. C. Cordeiro and K. Challapali, "C-MAC: a cognitive MAC protocol for multi-channel wireless networks," in *2nd IEEE International Symposium on New Frontiers in Dynamic Spectrum Access Networks*, Dublin, Ireland, 2007, pp. 147-157.
7. L. Le and E. Hossain, "OSA-MAC: a MAC protocol for opportunistic spectrum access in cognitive radio networks," in *IEEE Wireless Communications and Networking Conference*, Las Vegas, NV, 2008, pp. 1426-1430.
8. Q. Wang, S. Leng, H. Fu, and Y. Zhang, "An IEEE 802.11p-based multichannel MAC scheme with channel coordination for vehicular ad hoc networks," *IEEE Transactions on Intelligent Transportation Systems*, vol. 13, no. 2, pp. 449-458, 2012.
9. I. F. Akyildiz, W. Y. Lee, and K. R. Chowdhury, "CRAHNs: cognitive radio ad hoc networks," *Ad Hoc Networks*, vol. 7, no. 5, pp. 810-836, 2009.
10. B. F. Lo, I. F. Akyildiz, and A. M. Al-Dhelaan, "Efficient recovery control channel design in cognitive radio ad hoc networks," *IEEE Transactions on Vehicular Technology*, vol. 59, no. 9, pp. 4513-4526, 2010.
11. T. Chen, H. Zhang, G. M. Maggio, and I. Chlamtac, "Cog-Mesh: a cluster-based cognitive radio network." in *2nd IEEE International Symposium on New Frontiers in Dynamic Spectrum Access Networks*, Dublin, Ireland, 2007, pp. 168-178.
12. L. Ma, X. Han, and C. C. Shen, "Dynamic open spectrum sharing MAC protocol for wireless ad hoc networks," in *First IEEE International Symposium on New Frontiers in Dynamic Spectrum Access Networks*, Baltimore, MD, 2005, pp. 203-213.
13. S. Liu, L. Lazos, and M. Krunz, "Thwarting control-channel jamming attacks from inside jammers," *IEEE Transactions on Mobile Computing*, vol. 11, no. 9, pp. 1545-1558, 2012.
14. Y. Song and J. Xie, "A distributed broadcast protocol in multi-hop cognitive radio ad hoc networks without a common control channel," in *Proceedings of the IEEE International Conference on Computer Communications*, Orlando, FL, 2012, pp. 2273-2281.
15. C. F. Shih, T. Y. Wu, and W. Liao, "DH-MAC: a dynamic channel hopping MAC protocol for cognitive radio networks," in *IEEE International Conference on Communications*, Cape Town, South Africa, 2010, pp. 1-5.
16. 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.
17. N. C. Theis, R. W. Thomas, and L. A. DaSilva, "Rendezvous for cognitive radios," *IEEE Transactions on Mobile Computing*, vol. 10, no. 2, pp. 216-227, 2011.
18. M. Altamimi, K. Naik, and X. S. Shen, "Parallel link rendezvous in ad hoc cognitive radio networks," in *IEEE Global*

- Telecommunications Conference*, Miami, FL, 2010, pp. 1-6.
19. D. Pu, A. M. Wyglinski, and M. McLernon, "A frequency rendezvous approach for decentralized dynamic spectrum access networks," in *4th International Conference on Cognitive Radio Oriented Wireless Networks and Communications*, Hannover, Germany, 2009, pp. 1-6.
 20. Y. Kondareddy and P. Agrawal, "Synchronized MAC protocol for multi-hop cognitive radio networks," in *IEEE International Conference on Communications*, Beijing, China, 2008, pp. 3198-3202.
 21. C. Cormio and K. R. Chowdhury, "Common control channel design for cognitive radio wireless ad hoc networks using adaptive frequency hopping," *Ad Hoc Networks*, vol. 8, no. 4, pp. 430-438, 2010.
 22. L. DaSilva and I. Guerreiro, "Sequence-based rendezvous for dynamic spectrum access," in *3rd IEEE Symposium on New Frontiers in Dynamic Spectrum Access Networks*, Chicago, IL, 2008, pp. 1-7.
 23. K. Bian, J. M. Park, and R. Chen, "Control channel establishment in cognitive radio networks using channel hopping," *IEEE Journal on Selected Areas in Communications*, vol. 29, no. 4, pp. 689-703, 2011.
 24. E. J. Anderson and R. R. Weber, "The rendezvous problem on discrete locations," *Journal of Applied Probability*, vol. 27, no. 4, pp. 839-851, 1990.
 25. Z. Lin, H. Liu, X. Chu, and Y.W. Leung, "Jump-stay based channel-hopping algorithm with guaranteed rendezvous for cognitive radio networks," in *Proceedings of the IEEE International Conference on Computer Communications*, Shanghai, China, 2011, pp. 2444-2452.
 26. S. C. Jha, M. M. Rashid, V. K. Bhargava, and C. Despins, "Medium access control in distributed cognitive radio networks," *IEEE Wireless Communications*, vol. 18, no. 4, pp. 41-51, 2011.
 27. J. So and N. H. Vaidya, "Multi-channel MAC for ad hoc networks: handling multi-channel hidden terminals using a single transceiver," in *Proceedings of the 5th ACM International Symposium on Mobile Ad Hoc Networking and Computing*, Tokyo, Japan, 2004, pp. 222-233.



Zaw Htike

Zaw Htike received Bachelor of Computer Technology in 2006 from University of Computer Studies, Mandalay, Myanmar. Since 2008, he has been working towards the Ph.D. degree at Department of Computer Engineering, Kyung Hee University, Korea. His research interest includes cognitive radio networks, cooperative communication, and MAC protocols for wireless ad hoc networks.



Choong Seon Hong

Choong Seon Hong received his B.S. and M.S. degrees in Electronic Engineering from Kyung Hee University, Rep. of Korea, in 1983, and 1985, respectively. In 1988, he joined KT, and worked on Broadband Networks as a member of the technical staff. Since September 1993, he joined Keio University, Japan, and received his Ph.D. degree in March, 1997. He had worked for the Telecommunications Network Lab, KT as a senior member of technical staff and as a director of the networking research team until August 1999. Since September 1999, he has been working as a professor of the College of Electronics and Information, Kyung Hee University. He has served as a Program Committee Member and an Organizing Committee Member for International conferences such as NOMS, IM, APNOMS, E2EMON, CCNC, ADSN, ICPP, DIM, WISA, BcN and TINA. His research interest includes ad-hoc networks, network security and management. He is a member of IEEE, IEICE, IPSJ, KIPS, KICS and KIISE.



Sungwon Lee

Sungwon Lee received his B.S. and Ph.D. degrees from Kyung Hee University, Rep. of Korea. He is a professor of the Computer Engineering Department at Kyung Hee University, South Korea. Dr. Lee was a senior engineer of Telecommunications and Networks Division at Samsung Electronics Inc. during 1999 to 2008. He is an editor of the Journal of Korean Institute of Information Scientists and Engineers: Computing Practices and Letters.