PAPER Special Section on Cognitive Radio and Heterogeneous Wireless Networks in Conjunction with Main Topics of CrownCom2011

A Hybrid MAC Protocol for Cognitive Radio Ad Hoc Networks*

Zaw HTIKE†a), **Jun LEE**†b), ***Nonmembers***, **Choong Seon HONG**†c), ***Member***, ***and*** **Sungwon LEE**†d), ***Nonmember***

SUMMARY In cognitive radio networks, secondary users exchange control information to utilize the available channels efficiently, to maintain connectivity, to negotiate for data communication such as sender-receiver handshakes, for neighbor discovery etc. This task is not trivial in cognitive radio networks due to the dynamic nature of network environment. Generally, this problem is tackled by using two famous approaches. The first one is the use of common control channel (CCC) and the second one is using channel hopping (a.k.a sequence-based protocols). The use of CCC simplifies the processes of MAC protocols. However, it may not be feasible in cognitive radio networks as the available channels, including control channel, are dynamically changing according to primary user activities. Channel hopping approaches can tolerate the failure of network due to primary user activities. But it causes significant amount of channel access delay which is known as time to rendezvous (TTR). In this paper, we propose a hybrid protocol of these two mechanisms. This hybrid protocol can maintain connectivity and it can guarantee the secondary users to be able to exchange necessary control information in dynamic environment. In our hybrid protocol, we use multiple control channels. If some control channels are unavailable in case of primary user appearances, secondary users still can communicate on different control channels, so it can be more tolerable primary user activities than normal CCC approaches. Channel hopping is performed only for control channels, so it provides relatively small amount of channel access delay.

key words: *cognitive radio ad hoc networks, control channels, channel hopping*

1. Introduction

Cognitive radio technology has received the attention of many researchers as a new way to improve spectral efficiency of wireless networks. In CR networks, secondary users (SUs) are allowed to use free portions of licensed spectrum or channel in opportunistic fashion without causing any interference to primary users (PUs). All SUs need to scan and detect the free channels to utilize. These sensing results information should be exchanged periodically among SUs to be more reliable and accurate as available channels are dynamically changing [1]. One more important thing is that users need to negotiate by exchanging control information (such as request-to-send (RTS)/clear-to-send (CTS) in 802.11 Distributed Coordination Function (DCF) [2]) to establish data communication on a common channel. These show that spectrum utilization and resource optimization rely on information exchange among secondary users. However, this introduces the most remarkable challenge; how the SUs can exchange these control information in ever-changing dynamic environment.

Normally this problem is simplified by using predefined common control channel (CCC). Most of the proposed MAC protocols for cognitive radio networks were designed by assuming the existence of CCC and it is available for every secondary user (SU) [3], [4]. In this approach, all necessary control information are embedded in control packets and exchanged among SUs via CCC. When a SU wants to initiate any communication, it switches to CCC first and attempts to negotiate with the intended receiver. After negotiation has been done on CCC, data communication can be accomplished in other available channels known as data channels [5], [6]. Figure 1 illustrates the normal operation of a network with a common control channel. In contention phase, all nodes attempt to negotiate on CCC. After negotiation has been done on CCC, nodes** move to selected channels and perform data communications simultaneously.

The main advantage of using CCC is it makes protocols simple and efficient [7]. However, establishing a constantly available CCC in CR networks is a great challenge due to the following reasons.

- *Control channel saturation problem*: Since all SUs use only one CCC for control packets exchange, the control channel may become a bottleneck and it can cause single point of failure [8].
- *Presence of primary users*: When a PU appears on

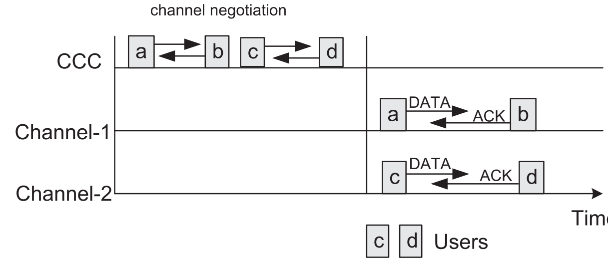

Fig. 1 Process of channel negotiation and data communication with a common control channel.

Manuscript received August 22, 2011.
Manuscript revised November 19, 2011.

†The authors are with the Department of Computer Engineering, Kyung Hee University, 1 Seocheon, Giheung, Yongin, Gyeonggi, 449-701 Korea.

*This work was supported by a grant from the Kyung Hee University in 2011 (KHU-20111209).

a) E-mail: htike@networking.khu.ac.kr
b) E-mail: junlee@networking.khu.ac.kr
c) E-mail: cshong@khu.ac.kr (corresponding author)
d) E-mail: drsungwon@khu.ac.kr

DOI: 10.1587/transcom.E95.B.1135

**In this paper, we use the terms, node and secondary user interchangeably

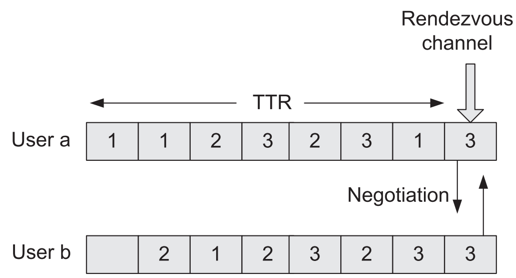

Fig. 2 Operation of a channel hopping protocol.

CCC, all SUs must defer their transmissions and vacate the channel immediately. If primary user's transmission period is significantly long on CCC, the presence of PU may block the SUs to access the channel. It will definitely degrade the overall throughput of CR network.

- *Channel availability*: In CR networks, the available channels are dynamically changing including the CCC. Therefore, an ever-available control channel for all SUs is unlikely to exist.

It seems that channel hopping approaches can compensate the problems of CCC. In channel hopping approaches, SUs generate their own channel hopping sequences. When a SU needs to communicate with its neighbor, it switches one channel after another by following predefined hopping-sequence and finds its neighbor. When two SUs (sender and receiver) rendezvous on a common channel, they exchange control packets and negotiate for data communication [9], [10]. Figure 2 illustrates the operation of a channel hopping protocol. In Fig. 2, user *a* and *b* find each other by following their own hopping sequences. When they meet on channel 3, they perform negotiation for data communication.

The main advantage of these approaches is SUs can rendezvous in every available channel. Therefore, it can overcome control channel saturation problem and tolerate long term blocking of primary users. Moreover, channel hopping protocols consider only pair-wise rendezvous (only a sender and a receiver), so these do not need universally available common control channel.

However, as there is no perfect one-for-all solution, channel hopping approaches also have drawbacks.

- *Channel access delay*: In CH protocols, whenever a SU needs to communicate with its neighbor, it has to find its neighbor by switching one channel after another. SU needs significant amount of time to find its neighbor and it is, normally, called time to rendezvous (TTR). The value of TTR depends on channel hopping algorithm.

- *Complexity*: The main challenge of CH approaches is generating hopping-sequences. Any pair of these sequences should overlap at least once within a sequence period so that any pair of nodes that needs to communicate can rendezvous [11]. Moreover, the TTR values between any pair of sequences should be reasonable.

In [12], channel hopping sequences are created in round robin fashion and it requires tight synchronization among SUs which is difficult to achieve in ad hoc environment. In [13], the authors proposed biased pseudo-random sequences. These do not need tight synchronization but the average TTR may not be bounded. The authors of [9] proposed permutation-based channel hopping sequences. In their proposal, the expected time to rendezvous is bounded by quadratic function of the number of available channels. A quorum-based scheme is proposed in [14] and the authors claimed that rendezvous between any pair of nodes can occur at least once within N^2 time slots, where N is number of available channels. It is obvious that the TTR value is determined by the number of available channels. Therefore, in channel hopping protocols, nodes may take a long time to find their neighbors for control packets exchange, especially when the number of available channels is large [7].

In this paper we propose a hybrid protocol of above two mechanisms. We cannot abandon CCC because of its simplicity and efficiency. In our proposed protocol, we use multiple control channels rather than only one CCC. If some control channels are unavailable in case of PU activities, SUs still can use other free control channels, so it can tolerate long-term blocking of primary users. SUs can negotiate on different control channels independently and this can overcome control channel saturation problem. Channel hopping is performed only for control channels, so that TTR value is relatively small. Moreover, it does not need to generate channel hopping-sequence. SUs just need to memorize the list of control channels or rendezvous channels, so it provides less complexity.

This paper is organized as follows. We describe proposed hybrid protocol in Sect. 2. In Sect. 2, control channels establishing, control channels access mechanism and negotiation for data communication are presented. In Sect. 3, we analyze and simulate proposed hybrid protocol to evaluate the performance. We evaluate our mechanism based on channel access delay, connectivity, collision rates, average throughput of each user and simulation results are presented. We also discuss about trade-off between channel access delay and collision rate. Then, Sect. 4 concludes the paper.

2. Proposed Hybrid Protocol

2.1 Control Channels Establishing

We assume that there are N number of available channels in the network. These are divided into M number of groups which are $G_i \subseteq N, (i = 1, 2, 3, ..., M)$. None of the available channels is overlapping in any two groups, which means $G_i \cap G_j = \phi, i \neq j$. One channel of each group, $(CC_i \in G_i)$, is selected as a control channel. Therefore, in the set of control channels, there are M number of control channels that is $CC_{list} = \{CC_1, CC_2, CC_3, ..., CC_M\}$. This control channel list is stored in every node of the network. In Table 1, a list

Table 1 List of notations.

Term	Definition
N	Number of available channels
M	Number of control channels (Number of channel groups)
G_i	Channel group i
C^i_{list}	Free channel list of group i
C^i_l	Channel l which belongs to group i
CC_{list}	List of control channels
CC_i	Control channel i
SIFS	Short Inter Frame Space
DIFS	Distributed Inter Frame Space

of notations is presented.

2.2 Channel Access Mechanism

We also assume that there is no centralized coordinator in the network and nodes are operating on different available channels independently. Any node in the network could be one of the following states in any time.

- Active state: Node has data to send to other nodes.
- Passive state: Node has no data to send.

If a node is in active state which implies it has some data for other node, so firstly, it must rendezvous with its intended receiver in order to negotiate for data communication. Therefore, active node switches one control channel after another without repeating ($CC_i \in CC_{list}, \forall i \in CC_{list}$) and attempts to rendezvous with its intended receiver. When it switches to a control channel, it senses the channel for the presence of PU and other SUs' transmission. If it senses the PUs' signal, it shall switch to next control channel. If only secondary users' signals are detected, it will perform the random back off. If it senses the channel is free, it will probe whether its intended receiver is on current control channel by broadcasting a preamble (PRE). Preamble is a small packet and it contains only receiver's ID. After transmitting a PRE, it waits for an acknowledgment (ACK). If the intended receiver is in current control channel and receives PRE, it will reply ACK and we can assume that rendezvous has occurred between these two nodes. Then, these two users can perform channel negotiation for data communication. All packet transmissions follow the principle of 802.11 DCF [2] and basic procedure of packet transmissions can be seen in Fig. 3. Total interval of this process is defined as one time slot (T_{slot}) and it can be estimated as

$$T_{slot} = \frac{PRE + ACK}{T_{rate}} + DIFS + SIFS + B, \quad (1)$$

where T_{rate} is transmission rate and B represents the random back-off. While active node is performing random back-off and waiting ACK, it might also receive PRE with its ID. If it receives PRE instead of ACK, it will simply reply ACK and perform channel negotiation.

If active node does not receive any ACK or PRE within a T_{slot}, it shall switch to another control channel and broadcast preamble again. This process is repeated until it rendezvous with the intended receiver or after switching all

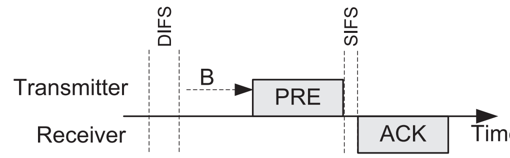

Fig. 3 Procedure of packet transmissions.

Algorithm 1 Channel access algorithm for active node

for Round $i = 1 : M$ **do**
  $CC = rand[CC_{list}]$
  Switches to CC
  if Channel is free **then**
    Broadcasts PRE and waits for ACK
    if Received ACK **then**
      Quit all loops
      Do channel negotiation
    else if Received PRE from neighbor **then**
      Reply ACK
      Quit all loops
      Do channel negotiation
    else
      $CC_{list} = CC_{list} \setminus CC$
    end if
  end if
end for

control channels. The necessary time for active node to switch and find its intended receiver on every control channel can be estimated as

$$T_{round} = T_{slot}.M, \quad (2)$$

where M is number of control channels. If the active node cannot rendezvous with the receiver on any control channel, its receiver might be currently communicating with other nodes on a data channel or it might be in active state too. Therefore, when an active node has spent a T_{round} without meeting with its intended receiver, it will choose active state for next T_{round} with probability P_{active}. The optimal value of P_{active} is described as 0.7525 in [15]. Algorithm 1 shows detailed procedure of channel access mechanism for active nodes.

If a node is in passive state, it chooses one control channel ($CC_i \in CC_{list}$) and tunes its radio to receiving mode. While it is in passive state on control channel CC_i, it performs periodic sensing for all available channels which belong to G_i and makes a free channel list (C^i_{list}). Periodic sensing is performed by passive nodes and as soon as it senses primary user signal on current control channel, it will switch to another control channel immediately. If a passive node does not get any preamble from its potential sender, it is possible that the current control channel is not available for the sender or it is too congested and the PRE cannot be sent. Therefore, if a passive node spends a T_{round} on a control channel without receiving any PRE, it shall switch another control channel $CC_j \in CC_{list}$ and wait one T_{round} again. If it receives preamble from its sender, it will simply reply ACK and these two nodes can perform channel negotiation on current control channel. Detailed procedure for

Algorithm 2 Channel access algorithm for passive node

```
for Round i = 1 : M do
    Switches to CC_i ∈ CC_list
    if The channel is free then
        Wait PRE for one T_round
        if Received PRE then
            Reply ACK
            Quit all loops
            Do channel negotiation
        end if
    end if
end for
```

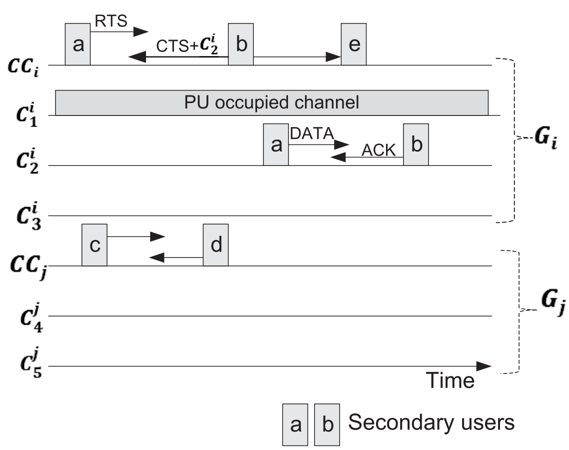

Fig. 4 Channel access negotiation.

the passive node can be found in algorithm 2.

2.3 Channel Negotiation

If two nodes rendezvous on a control channel, they can perform channel negotiation. In negotiation phase, sender sends RTS to receiver as traditional 802.11 DCF. Receiver chooses one channel from its free channel list which is $C_l^i \in C_{list}^i$ and replies it with CTS. After exchanging control packets (RTS and CTS), these two nodes switch to C_l^i and perform data communication. Other neighbor nodes which are currently on CC_i can overhear CTS from the receiver, so that they remove C_l^i from their C_{list}^i. Therefore, any node which overhears the CTS can avoid interference to neighbor nodes' data communication and this can overcome multi-channel hidden terminal problem [8]. Periodic sensing is performed by passive nodes and the free channel lists are updated periodically. Therefore, if data transmission is finished and C_l^i is sensed as free channel in next sensing time, it will be included in new C_{list}^i.

The detailed process of channel access negotiation is illustrated in Fig. 4. Secondary user *a* and user *b* rendezvous on CC_i and *a* transmits RTS to *b*. SU *b* selects C_2^i and replies it with CTS. Then *a* and *b* move to C_2^i and perform data communication. SU *e* is currently on CC_i and it can overhear CTS from *b*. It can know that channel C_2^i is going to be used by neighbors, so that *e* modifies its free channel list as $C_{list}^i = C_{list}^i \backslash C_2^i$. While *a* and *b* negotiating on CC_i, SU *c* and

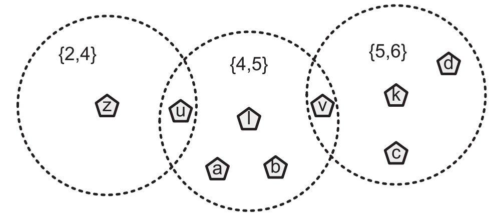

Fig. 5 Multi-hop network.

d can negotiate on CC_j simultaneously. It can significantly reduce control channel saturation problem. Moreover, if a PU appears on a control channel, SUs still can negotiate on different control channels and maintain the connectivity of the network.

2.4 Multi-hop Networks

In multi-hop environment, we need to define center nodes of one-hop neighbors as in clustering. Suppose node *l* is center node and any node which is one-hop away from node *l* belongs to cluster *l*. The edge nodes can be defined as any node which is one-hop away from any two or more center nodes. The control channel lists of edge nodes are generated as $CC_{list}^{passive} = CC_{list}^{l} \bigcap CC_{list}^{k}$ for passive state and $CC_{list}^{active} = CC_{list}^{l} \bigcup CC_{list}^{k}$ for active state, where node *l* and *k* are center nodes and CC_{list}^{l} and CC_{list}^{k} are control channel lists of node *l* and *k* respectively. If the edge node is in passive state, it chooses a control channel from $CC_{list}^{passive}$ and waits its potential sender. If it is in active state, it switches one control channel after another by following CC_{list}^{active} and finds its receiver.

Figure 4 shows an example of multi-hop scenario. In Fig. 5, node *u* and *v* are edge nodes, and node *l*'s and *k*'s control channel lists are $CC_{list}^{l} = \{4, 5\}$ and $CC_{list}^{k} = \{5, 6\}$. So that, node *v* creates its control channel lists as $CC_{list}^{passive} = CC_{list}^{l} \bigcap CC_{list}^{k} = \{5\}$ and $CC_{list}^{active} = CC_{list}^{l} \bigcup CC_{list}^{k} = \{4, 5, 6\}$. Similar to node *v*, node *u*'s control channels lists are $CC_{list}^{active} = \{2, 4, 5\}$ and $CC_{list}^{passive} = \{4\}$. This guarantees that even though there is no global available common control channel, network is universally connected by using multiple control channels.

3. Performance Evaluation

3.1 Channel Access Delay

When a node has data to send, it needs to find its intended receiver first for negotiation. It needs sufficient amount of time to find the receiver and it is called channel access delay or time to rendezvous (TTR). First, we evaluate our protocol based on TTR.

According to the states we discussed in Sect. 2.2, any pair of nodes (sender and receiver) could be in one of the following three events.

- *Event A*: Both nodes are in active state.

- *Event B*: One node (receiver) is in passive state and the other node (sender) is in active state.
- *Event C*: Both nodes are in passive state.

If event A occurs, both nodes try to find each other by switching one control channel after another. In this event, the probability of meeting these two nodes on control channel i is $\frac{1}{M}$. Then, the probability of not meeting at all within a round becomes $(1-\frac{1}{M})^M$. The probability of at least one rendezvous occurs within a round is $1-(1-\frac{1}{M})^M$. Then, the expected time slots for this event can be estimated as:

$$E_A[TTR]=M\left\{1-\left(1-\frac{1}{M}\right)^M\right\}. \tag{3}$$

If event B occurs, sender (active node) shall try to find its receiver by switching one control channel after another while the receiver (passive node) is waiting on a control channel. The probability of meeting on control channel i is $\frac{1}{M}$. If rendezvous has not occurred, sender shall switch to another control channel, let say channel $i+1$, and the probability of meeting on channel $i+1$ becomes $\frac{2}{M}$ and so on. Therefore, the expected time to rendezvous for this event is

$$E_B[TTR]=\sum_{i=1}^{M}\frac{i}{M}=\frac{M+1}{2}. \tag{4}$$

If event C occurs, both nodes are in passive state and we can simply ignore this event.

Suppose, P is the probability of a node being in active state, then we can estimate the overall E [TTR] as follows.

$$E[TTR]=P(1-P)(M+1)+MP^2\left\{1-\left(\frac{M-1}{M}\right)^M\right\}, \tag{5}$$

where M is number of control channels and it is much smaller than number of total available channels, $M<N$. So that the E[TTR] value is relatively small compare to that of random algorithm (RA) which is $\frac{N^2}{pH.N}$ and orthogonal sequence-based algorithm (OSA) which is $\frac{N^4+2N^2+6N-3}{3N(N+1)}$ [16], where pH is probability of a successful handshake and N is number of available channels.

We plot the average TTR values of proposed hybrid protocol (HP), random algorithm (RA) and orthogonal sequence-based algorithm (OSA). We consider worst case scenario for hybrid protocol as number of control channels is the same as number of available channels which is $M=N$. The probability of being in active state is set to be P = 0.5. As shown in Fig. 6, even in worst case scenario, the E[TTR] of hybrid protocol is smaller than that of other sequence-based protocols.

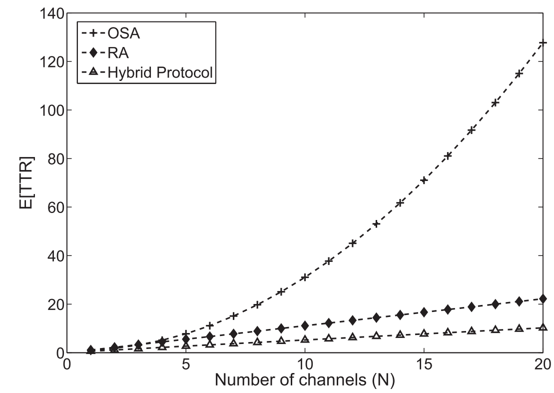

Fig. 6 E[TTR] under number of available channels.

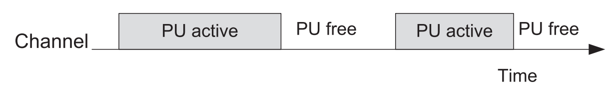

Fig. 7 PU activities on a channel.

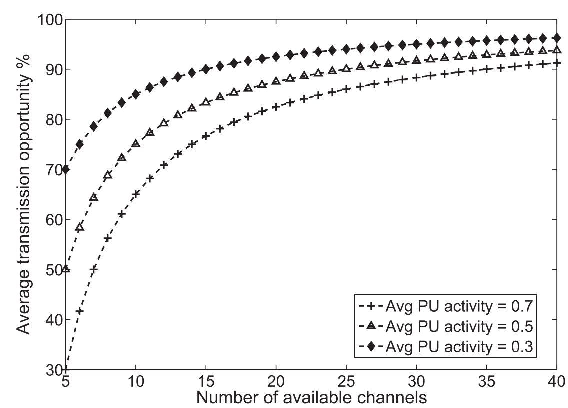

Fig. 8 Average transmission opportunity for SUs under different available channels.

3.2 Connectivity

Secondary users are allowed to use only free or idle potions of channels. In our evaluation and simulations, we assume that SUs avoid PU occupied channels and try to access only free potions of channels. Moreover, when a PU appears on current operating channel, all SU must cease their transmission and vacate to other free channels immediately. Figure 7 shows an example of PU activities on a channel. Transmission opportunities on a channel happen when the PU does not use it. Average transmission opportunity for SUs on channel k is

$$P_k^{idle}=\frac{\sum PU_{free}}{\sum PU_{active}+\sum PU_{free}}, \tag{6}$$

where PU_{active} refers PU occupied periods of a channel and PU_{free} represents idle periods. Average transmission opportunity for SUs is

$$E[T_{opp}]=\frac{1}{N}\sum_{k=1}^{N}P_k^{idle}. \tag{7}$$

We plot the average transmission opportunity for SUs in Fig. 8. We assume that there are five primary users in the network and PU can appear on any available channel at

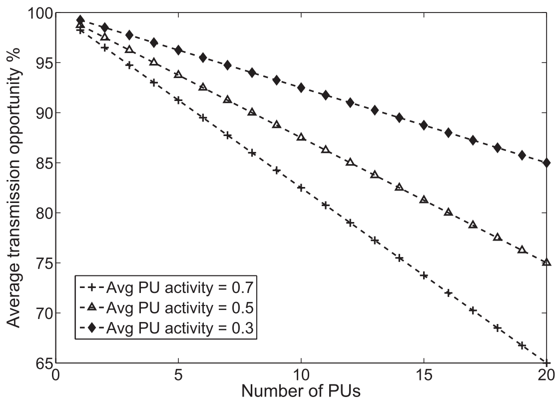

Fig. 9 Average transmission opportunity for SUs against number of PUs.

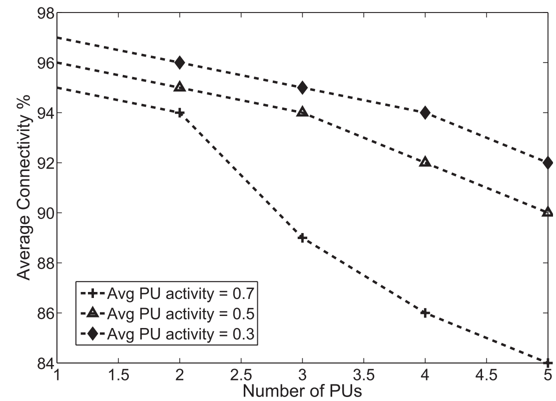

Fig. 10 Connectivity of network against number of primary users.

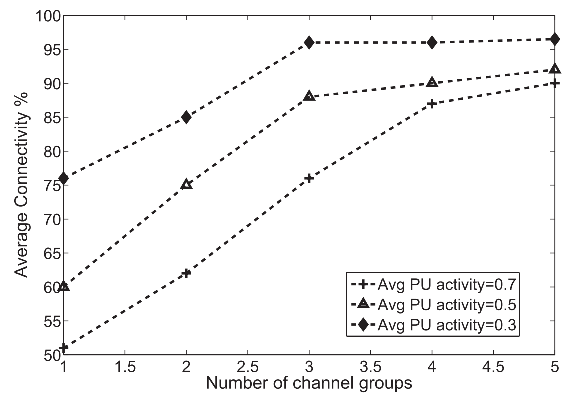

Fig. 11 Connectivity of network with various numbers of channel groups.

any time. We define various primary user activities (average PU_{active}) as 0.7, 0.5 and 0.3. As shown in Fig. 8, the average transmission opportunity for SUs increases as the number of available channels increase.

On the other hand, the number of PUs in the network significantly effects on transmission opportunity for SUs and connectivity of network. Figure 9 illustrates the average transmission opportunity against number of PUs. In this case, we assign number of available channels as N = 40 and average PU activities are also 0.7, 0.5 and 0.3. As shown in Fig. 9, it is significantly decreased as the number of PU increase. These show that the average transmission opportunity for SUs is proportional to number of available channels but inversely proportional to number of PUs.

The connectivity of CR network is determined by the availability of control channels and it can be described as

$$E[T_{opp}^{CC}] = \frac{1}{M} \sum_{\forall k \in CC_{list}} P_k^{idle}. \tag{8}$$

To measure the connectivity, we ran simulations on *Omnet + +* simulator. We assign 20 available channels ($N = 20$) and 4 control channels($M = 4$). It is assumed that PUs can appear on any channel randomly and connectivity is measured based on availability of controls channels. We ran simulations 100 times and take the average value. The results of connectivity against number of PUs are presented in Fig. 10.

We also measure the connectivity of the network under various number of channel groups. In this scenario, we assign five channels in each channel group, $G_i = \{C_1^i, ..., C_5^i\}$. One of five channels is assigned as control channel and the rest are data channels. Number of primary users are set to be four. PU activities are also 0.7, 0.5 and 0.3 and PU appearances are random. As shown in Fig. 11, the large the number of channel groups, the more increase the connectivity.

3.3 Collision Rate and Throughput

In proposed hybrid protocol, SUs need to rendezvous their neighbors on one of control channels and negotiate for data communication. Therefore, collision rates depend on how many users attempt to negotiate on control channels. We simulate our mechanism to measure the control packets collision rate against the number of SUs in the network, as well as the number control channels. In this scenario, the number of primary users is set to be five and average PU activity is 0.5. Number of available channels is defined as 20 ($N = 20$) and SUs are randomly distributed on different channels independently. The control packets (RTS/CTS and PRE/ACK) sizes are defined as 160 bits. After a sender rendezvous with its intended receiver on a control channel, it will attempt to negotiate for data communication as described in Sect. 2.2. When collisions occur, SU retries to negotiate. We assumed that collisions occur when two or more nodes transmit the control packets at the same time. There are two possible ways of collisions occurrence. The first one is two or more nodes sense the channel as free at the same time. The second one is back off time (B) of two or more users reach to zero at the same time.

Simulation results of normal CCC approach and proposed hybrid protocol with various control channel assignments can be seen in Fig. 12.

In normal CCC approach, one of the available channels is assigned as common control channel and the rest are used for data communication. When the number of users increases, collision rate of normal CCC approach also in-

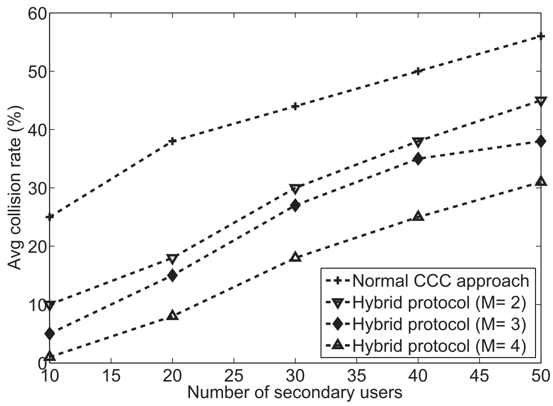

Fig. 12 Control packets collision rates against various numbers of SUs and control channels.

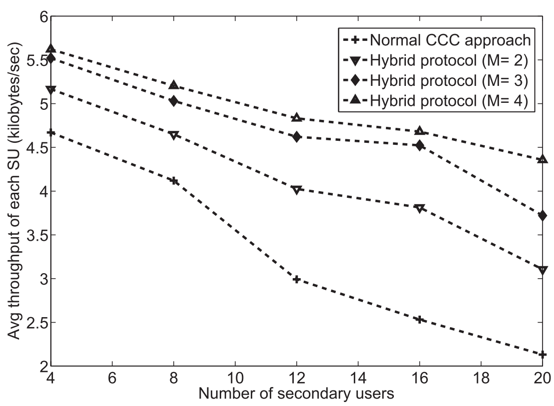

Fig. 13 Throughputs of normal CCC approach and hybrid protocol with various control channels assignments.

creases, because all nodes use only one control channel for negotiation. However, as shown in Fig. 12, we can reduce the collision rate by using multiple control channels. The collision rate of hybrid protocol with two control channels or two channel groups is almost 50% less than normal CCC approach.

The control packets collision rates determine the throughput of the network because data communications can be accomplished only after successful negotiations. We can assume that, there are no data packet collisions among SUs because of the procedure we presented in Sect. 2.3. However, if a PU appears on current data channel, SUs defer the current transmission and switch to the previous control channel for re-negotiation. Throughputs are measured for data packets only and packet sizes are randomly assigned between 512–1024 bytes. The transmission rate is set to be 11 Mbps. Figure 13 presents the throughputs of normal CCC approach and hybrid protocol with various control channel assignments. The collision rate increases as the number of users in the network becomes large and it degrades the average throughput of each user. However, as shown in Fig. 13, we can achieve the desirable throughput by using multiple control channels.

3.4 Discussion

As we have seen, using multiple control channels can significantly reduce the collision rate especially when the number of users in the network is large. It also can enhance the overall throughput of the network. Moreover, using multiple control channels enhances tolerance of the network failures due to primary user activities. On the other hand, using multiple control channels can increase channel access delay and switching overhead. Therefore, the trade-off between collision rate, throughput and channel access delay should be considered carefully.

4. Conclusion

In this paper we have presented a hybrid MAC protocol for cognitive radio ad hoc networks. The proposed hybrid protocol is more tolerable primary user activities than traditional CCC approaches. Moreover, it can overcome control channel saturation problem. It also can guarantee the SUs to be able to exchange control information in dynamic environment. Simulation results validate that the hybrid protocol outperforms the previous works in term of channel access delay, collision rate and average throughput.

References

[1] 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, July 2009.
[2] G. Bianchi, "Performance analysis of the IEEE 802.11 distributed coordination function," IEEE J. Sel. Areas Commun., vol.18, no.3, pp.535–547, March 2000.
[3] X. Zhang and H. Su, "CREAM-MAC: Cognitive radio-enabled multi-channel MAC protocol over dynamic spectrum access networks," IEEE J. Sel. Top. Signal Process., vol.5, no.1, pp.110–123, 2011.
[4] L. Le and E. Hossain, "OSA-MAC: A MAC protocol for opportunistic spectrum access in cognitive radio networks," IEEE WCNC, pp.1426–1430, 2008.
[5] A.C.C. Hsu, D.S.L. Wei, and C.C.J. Kuo, "A cognitive MAC protocol using statistical channel allocation for wireless ad-hoc networks," IEEE WCNC, pp.105–110, 2007.
[6] C. Cordeiro and K. Challapali, "C-MAC: A cognitive MAC protocol for multi-channel wireless networks," IEEE DySpan, pp.147–157, 2008.
[7] B.F. Lo, "A survey of common control channel design in cognitive radio networks," Physical Communication, vol.4, no.1, pp.26–39, March 2011.
[8] J. So and N.H. Vaidya, "Multi-channel MAC for ad hoc networks: Handling multi-channel hidden terminals using a single transceiver," Proc. 5th ACM international symposium on Mobile ad hoc networking and computing (MOBIHOC), May 2004.
[9] L. DaSilva and I. Guerreiro, "Sequence-based rendezvous for dynamic spectrum access," IEEE DySpan, pp.1–7, 2008.
[10] C.F. Shih, T.Y. Wu, and W. Liao, "DH-MAC: A dynamic channel hopping MAC protocol for cognitive radio networks," IEEE ICC, pp.1–5, 2010.
[11] K. Bian, J.M. Park, and R. Chen, "Control channel establishment in cognitive radio networks using channel hopping," IEEE J. Sel. Areas Commun., vol.29, no.4, pp.689–703, 2011.

[12] Y. Kondareddy and P. Agrawal, "Synchronized MAC protocol for multi-hop cognitive radio networks," IEEE ICC, pp.3198–3202, 2008.
[13] C. Cormio and K. 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, June 2010.
[14] K. Bian, J.M. Park, and R. Chen, "A quorum-based framework for establishing control channels in dynamic spectrum access networks," MobiCom 2009, pp.25–36, 2009.
[15] E. Anderson and R. Weber, "The rendezvous problem on discrete locations," J. Applied Probability, vol.27, no.4, pp.839–851, 1990.
[16] N.C. Theis, R.W. Thomas, and L.A. DaSilva, "Rendezvous for cognitive radios," IEEE Trans. Mobile Comput., vol.10, no.2, pp.216–227, Feb. 2011.

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, MAC protocols for wireless ad hoc networks.

Jun Lee received his B.S. degree from Department of Computer Engineering, Kyung Hee University, Korea, in 2010. Since March 2010, he has been working on his M.S. degree in Department of Computer Engineering, Kyung Hee University, Korea. His research interests include advanced wireless network protocols, mobility management, and Ad hoc Network.

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, IPSJ, KIPS, KICS and KISS.

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.