

## acSB: Anti-Collision Selective-Based Broadcast Protocol in CR-AdHocs

Yueyue Li<sup>1,\*</sup>, Zhong Huang<sup>1</sup>, Yugang Ma<sup>2</sup> and Guangjun Wen<sup>1</sup>

**Abstract:** As a fundamental operation in ad hoc networks, broadcast could achieve efficient message propagations. Particularly in the cognitive radio ad hoc network where unlicensed users have different sets of available channels, broadcasts are carried out on multiple channels. Accordingly, channel selection and collision avoidance are challenging issues to balance the efficiency against the reliability of broadcasting. In this paper, an anti-collision selective broadcast protocol, called acSB, is proposed. A channel selection algorithm based on limited neighbor information is considered to maximize success rates of transmissions once the sender and receiver have the same channel. Moreover, an anti-collision scheme is adopted to avoid simultaneous rebroadcasts. Consequently, the proposed broadcast acSB outperforms other approaches in terms of smaller transmission delay, higher message reach rate and fewer broadcast collisions evaluated by simulations under different scenarios.

**Keywords:** Broadcast protocol, selective protocol, collision avoidance, distributed data dissemination, cognitive radio ad hoc network.

### 1 Introduction

Cognitive radio is regarded as a promising technology to obtain better spectrum utilization by enabling unlicensed user to access licensed spectrum bandwidth. Usually, the unlicensed user is called secondary user (SU) while the licensed user is defined as primary user (PU). SUs first sense the licensed channels to make sure they are free to access. A spectrum pool or available channel set containing sensing results is often maintained for further operations. The status of spectrum pool or channel set is changing along with time, space and frequency due to different PU states. With intelligent channel allocation policies, spectrum-sharing techniques and wireless communication technologies, SUs then transfer the information through authorized channels. While the cognitive radio technology is introduced into conventional ad hoc networks, namely cognitive radio ad hoc networks (CR-AdHocs), every node maintains a non-uniform channel set for channel access. Data acquired by SU nodes is from the surroundings of distributed SUs without the

---

<sup>1</sup> University of Electronic Science and Technology of China, Qingshuihe Campus, No. 2006, Xiyuan Ave, West Hi-Tech Zone, Chengdu, Sichuan, 611731, China.

<sup>2</sup> Institute for Infocomm Research, A\*STAR, 1 Fusionopolis Way, #21-01 Connexis (South Tower), 138632, Singapore.

\* Corresponding Author: Yueyue Li. Email: yyli@uestc.edu.cn.

use of common control channels. Particularly, for a single radio design, a node is capable of accessing one channel at a time. In other words, messages are successfully transmitted only if the transceivers are operating on the same channel at the moment. This is a very challenge issue on which the exchange of control messages as well as the actual information through broadcasting is required in CR-AdHocs.

Notice that various elements should be involved in decision makings of channel selection, such as timeliness, global network topology, spectrum status and so on. A tradeoff between delay and reach rate is important for broadcast protocol design in CR-AdHocs. Sureshkumar et al. [Sureshkumar and Rajeswari (2014)] categorized the popular approaches concerning broadcast issues in the field. For instance, random broadcast scheme introduces that only one channel is chosen for transmission by both sender and receiver. The method is flooded with randomness and therefore is at great risk to have a common channel. As a result, reach rate of messages is likely to be small, or even irregular. To achieve a higher reach rate, as spelled out in Sureshkumar et al. [Sureshkumar and Rajeswari (2014)], more than one channel assigned to SUs enhances the possibility of successful connections. Additionally, full broadcasting scheme represents another extreme case. The method discussed in Sureshkumar et al. [Sureshkumar and Rajeswari (2014); Song and Xie (2014)] implements broadcasting and listening behaviors on all allocated channels sequentially, by which the broadcast delay is fairly high, especially when the number of SUs and the number of available channels are large. On balance, an intelligent solution is to construct a subset of channels from the total available ones of each SU, i.e. selective broadcast scheme [Kondareddy and Agrawal (2008)] and minimum broadcast scheme [Arachchige, Venkatesan, Chandrasekaran et al. (2011)], taking into account the global network topology and spectrum status of every node.

Neighbor information is helpful for channel negotiation to balance the successful rate against the delay even though the information is hard to be maintained during the ad hoc broadcasting. There are many literatures state "beacon" for coordinating the coming radios to the first radio to synchronize their tx and rx frequencies into a same channel. Oh et al. [Oh, Ma, Peh et al. (2017)] summarized the possible access techniques in CR including beacon and others. A beacon based neighbor discovery method is put forward in Al-Mathehaji et al. [Al-Mathehaji, Boussakta, Johnston et al. (2015)] to get one-hop neighbor information. Every CR node classifies its channels based on neighbor information and a joint transmitter-receiver channel selection method is proposed to increase the reliability and the reachability of data dissemination. Kondareddy et al. [Kondareddy and Agrawal (2008)] proposed a neighbor graph to track channel information of neighbors. Some other papers investigate method without any neighbor information and most of them are based on channel hopping. In Theis et al. [Theis, Thomas and Dasilva (2010)], two channel hopping algorithms called generated orthogonal sequence (GOS) algorithm and modular clock (MC) algorithm are proposed. However, both algorithms have limitations. GOS is only suitable when two users have the same available channel sets and MC introduces unpredictable delay in different cases. A common control channel design for CR ad hoc networks is proposed, called as adaptive multiple rendezvous control channel (AMRCC) based on frequency hopping in Cormio et al. [Cormio and Chowdhury (2010)]. But the scheme cannot guarantee rendezvous in a finite time. Although a jump-stay based channel-hopping algorithm is proposed for guaranteed delay in Lin et al. [Lin, Liu, Chu et al. (2011)],

the expected delay increases exponentially along with the total number of channels. Also, a broadcast protocol under blind information for multi-hop CR ad hoc networks is proposed in Song et al. [Song and Xie (2014)] to provide QoS (success rate and average delay) support.

Collision is an along-standing problem in terms of broadcasting. It may occur frequently because the unnecessary replicated messages are rebroadcast; or because there are multiple SUs attempt to use a same channel for rebroadcasting. There are very limited papers addressing the collision issue in CR-AdHocs. A typical example is found in Song et al. [Song and Xie (2015)] that authors managed to overcome transmissions at the same time.

In this paper, we conclude problems demanding prompt solutions as follows: 1) to determine appropriate channels for achieving efficient matching between transceivers and 2) to implement a feasible collision avoidance mechanism. For the reasons, a multi-channel selective-based broadcasting protocol with collision avoidance policy is proposed. It challenges the conventional selective broadcast scheme (SB), which presents a wise channel selection method for transmissions but suffers from serious broadcasting collisions resulting in the low success rates of message deliveries. Simulation outcomes provide proof of higher communication performances regarding transmission delay, message reach rate and broadcast collisions. Following the introduction, Section 2 describes the system model and key assumptions. The problem formulation is particularly discussed in Section 3. After reviewing the traditional SB protocol, the procedure of the proposed broadcast protocol acSB is then detailed in Section 4. To verify the performance of acSB, various simulations are carried out in Section 5 to compare with the conventional SB protocol and full broadcast protocol. Finally, we conclude the paper.

## 2 System model and assumptions

We consider a CR-AdHoc system in this paper where  $N_p$  PUs and  $N_s$  SUs exist in a fixed dimension area, sharing  $M$  licensed channels, where  $N_p \in [0, M]$ . PUs may release channels at any moment and this leads to unpredictable fluctuations of available channels for SUs. Referring to the birth-death process [Al-Mathehaji, Boussakta, Johnston et al. (2015)], the probability of the  $i^{th}$  PU being inactive is given by Eq. (1).

$$p_i = \frac{\mu_i}{\lambda_i + \mu_i}, i \geq 1, \mu_0 = 0 \quad (1)$$

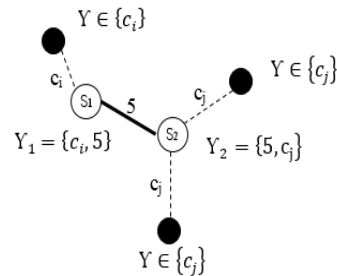
where  $\lambda_i$  and  $\mu_i$ , as the expectation of random variables, indicate the birth rate and death rate of the  $i^{th}$  PU respectively. The idle channel set probability can be calculated based on the inactive PUs as given by Eq. (2).

$$P = \prod_{i=1}^{N_p} p_i \quad (2)$$

The higher  $P$  is, the more idle channels specified as  $P \times M$  are given to  $N_s$  SUs. We denote  $\Gamma$  as the system idle channel where  $\Gamma = \{Y_1, Y_2, \dots, Y_i, \dots, Y_n\}$ .  $Y_i$  represents the available channel set of the  $i^{th}$  SU from  $\Gamma$  depending on its conditions such as position, mobility, energy and so on. A common control channel is defined in the traditional ad hoc networks for nodes, but not in the CR-AdHoc systems. Thus, SUs need to broadcast messages based on their own available channel sets, which should be of different sizes and elements. As an example shown in Fig. 1,  $S_1$  and  $S_2$  with  $Y_1 = \{c_i, 5\}$  and  $Y_2 = \{5, c_j\}$  respectively.  $c_i$

and  $c_j$  represent channel IDs where  $c_i \neq c_j \neq 5$  and  $c_i, c_j \in \mathbb{N}$ . If both SUs choose channel 5 at the same time, the communication succeeds.

In the project, there are several assumptions given. For instance, SUs are capable of sensing idle channels and detecting PUs with synchronized time slots; they are allocated different sizes of available channel spaces; the transmission range and sensing range are set to be the same. Furthermore, the idle channel states are not very dynamic in a certain period

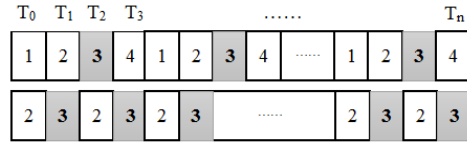


**Figure 1:** An example of multi-channel message propagation

(i.e. using TV channels that are underutilizations) so that the channels are available for implementing the whole broadcasting progress.

### 3 Problem formulation

It is convinced in Kondareddy et al. [Kondareddy and Agrawal (2008)] that an effective coordination requires fast broadcasting of control traffic, especially for applications demanding real-time communications. Broadcast delay is a very important indicator in analyzing the efficiency of broadcasting protocols. Selective broadcasting, certainly, reduces size of broadcasting channels and minimizes the delay of each node to a large degree extent. Apart from this, the performance of broadcast should take the reachability of messages into account. It is particularly observed as a result of successful connections between two nodes through their possible common channels. It is noticed that the factor impacting on reachability can be multifold. One potential issue is how to control channel sequences by which source nodes could pair up with their neighbors successfully as long as there is at least one same channel. For instance, Fig. 2 presents a case that  $S_1$  broadcasts a message on its own available channels  $\{1, 2, 3, 4\}$  and its neighbor  $S_2$  will listen on channels  $\{2, 3\}$ . Since SU may be configured by the single radio, in theory, this requires only one channel at each time for sending or receiving. At  $T_0$ ,  $S_1$  tunes to the channel 1 and  $S_2$  tunes to the channel 2; at  $T_1$ ,  $S_1$  tunes to the channel 2 and  $S_2$  tunes to the channel 3. Unfortunately, they fail to connect each other even though they have the common channel (e.g. 3) theoretically.



**Figure 2:** A negative example of broadcasting sequence

Collisions, which are hard to be evaded, would be the other problem that seriously impacts on broadcasting performance. As a matter of fact, there have been many accounts of this issue discussed in broadcasting protocols of traditional ad hoc networks. Typically, if multiple SUs rebroadcast a message choosing the same channel at a moment, the collision happens and results in failed communications. On the other hand, since nodes are distributed randomly in CR-AdHoc, the neighbor density of each SU varies in the area. A high neighbor density would impose collisions due to concurrent transmissions of SUs via the same channel. A popular strategy is to drop the message which would lose high possibility of successful delivery; and what is more, schemes are designed to add random waiting intervals to the next sending time for each user. However, this could not be directly adopted in our scenarios because the exact communication time for users is unpredictable.

#### 4 Anti-collision selective broadcast protocol (acSB)

##### 4.1 Review of selective broadcast scheme

The proposed broadcasting protocol proceeding from the selective broadcast scheme (SB) expects to improve communication delays in terms of smaller transmission delay, higher reach rate and less collisions. Before making an intensive study of the protocol, it is essential to reformulated critical principles of SB herein regarding subset channel selection. As shown by Kondareddy et al. [Kondareddy and Agrawal (2008)], neighbor graphs and minimal neighbor graphs are two important events. It is assuming that each SU knows its one-hop neighbors' information by broadcasting a beacon to them. Thus, it could construct a neighbor graph with the information of SU neighbors and common channels. As shown in Tab. 1,  $S_1$  has neighbors  $S_2, S_3, S_4, S_5, S_6, S_7$ ; the corresponding communication channels are  $\{1, 4\}, \{1\}, \{1, 2\}, \{2\}, \{2, 3\}$  and  $\{4\}$ .

**Table 1:** Neighbor Graph of  $S_1$

$S_2$	$S_3$	$S_4$	$S_5$	$S_6$	$S_7$
$\{1, 4\}$	$\{1\}$	$\{1, 2\}$	$\{2\}$	$\{2, 3\}$	$\{4\}$

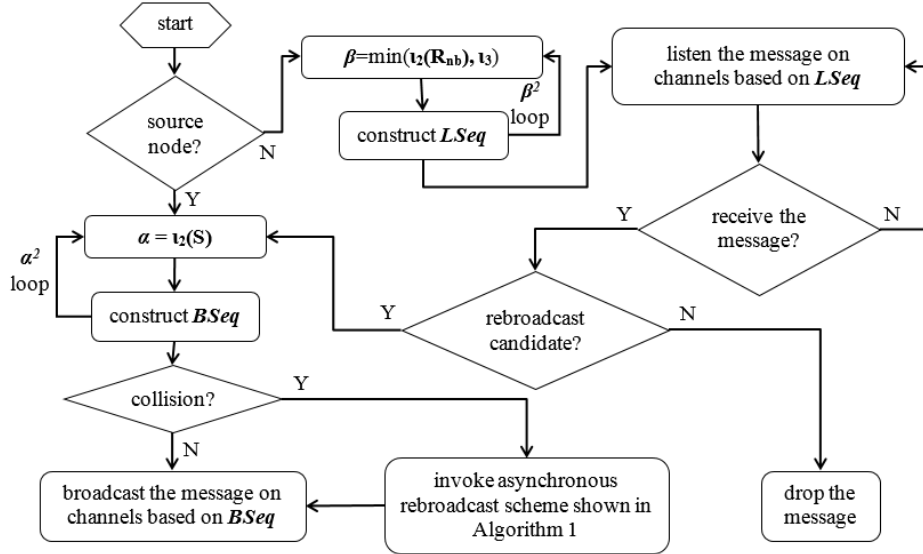
Then a minimal neighbor graph could be organized based on the neighbor graph. There are terms should be stated firstly. One is a set called  $DC$  in Kondareddy et al. [Kondareddy and Agrawal (2008)], being formed by the times of using corresponding channels; the other one is a set called by authors as  $ECS$ , including essential channel set allowing a SU to reach all the neighbors. According to the example of Tab. 1, the  $DC$  is  $\{3, 3, 1, 2\}$ , in accordance with the ascending order of channel indexes. At first time, channel 1 obtains the highest degree of 3 and the id is added to the  $ECS$ , where current  $ECS$  is  $\{1\}$ . Then, the neighbors

$S_2, S_3, S_4$  connected with  $S_1$  by channel 1 are moved from the neighbor table, where only  $S_5, S_6, S_7$  remain. As a result,  $DC$  is updated to be  $\{0, 2, 1, 1\}$ . By parity of reasoning, the progress will end if there is only  $S_1$  left. Channels in the  $ECS$  of  $S_1$  will be then followed in broadcasting one by one.

SB is proper to minimize the number of communication channels and meanwhile to guarantee successful transmissions efficiently. But it does not take into account the concerning problems mentioned in Section 3. This motivates us to go further and put forward an advanced protocol.

#### 4.2 The procedure of acSB

acSB Protocol enhances the conventional SB protocol achieving real-time, wide-coverage and energy-saving communications in the CR-AdHoc system by tackling typical broadcasting problems mentioned in Section 3. In this section, the procedure of the proposed acSB protocol is summarized by the flow chart shown in Fig. 3.



**Figure 3:** The flow chart of the proposed acSB protocol

If this is a source node, it assigns  $t_2(S)$  to  $\alpha$  and constructs the broadcasting sequence ( $BSeq$ ) by  $\alpha^2$  loops. If there are not any collisions, the sender will broadcast the message on corresponding channels; otherwise, it should invoke the asynchronous rebroadcast scheme (Algo. 1) and then broadcast the message. On the other hand, a receiver will compare the length of its available channel set, denoted by  $t_3$ , with the lengths of its neighbors'  $ECS$  channel sets,  $t_2(R_{nb})$ ; and then determines its iteration index  $\beta$  as the minimum value between  $t_2(R_{nb})$  and  $t_3$ . By  $\beta^2$  loops, the SU receiver organizes a listening sequence ( $LSeq$ ). Once the successful connection is established, the SU should decide whether it is a rebroadcast candidate by checking its rebroadcast table. If so, the SU works as a sender who is mainly responsible for generating a message and Sending it out with one broadcast each time; otherwise, communication terminates with discarding the message.

**Algorithm 1:** Asynchronous Rebroadcast**Input:**  $ECS_{S_n}, \iota_1, \iota_2(S_n), ECS'_{S_n} = \emptyset$ .**Output:**  $ECS'_{S_n}$ .

---

```

1: SU  $S_n$  detects a collision on the selected channel;
2:  $i \leftarrow 1$ ;
3: while  $\iota_2(S_n) \leq \iota_1$  and  $ECS_{S_n}(i) \leftarrow \emptyset$  do
4:    $ECS_{S_n}(i) \leftarrow 0$ ;
5:    $i \leftarrow i + 1$ ;
6: end while
7:  $\theta \leftarrow rand(1, \iota_1)$ ;
8:  $ECS'_{S_n} \leftarrow circleshift(ECS_{S_n}, \theta)$ ;
9: return  $ECS'_{S_n}$ 

```

---

Although acSB protocol has been proposed to improve the probability of channel matching between transceivers to a large extent, as the worst case of broadcasting sequence shown in Fig. 2, the connection will be failed eventually. The solution of acSB, enlightened by Song et al. [Song and Xie (2015)], is setting up  $\alpha^2$  times iterations for broadcasting and  $\beta$  times iterations for listening. According to the proof of a theorem in Song et al. [Song and Xie (2015)], only if  $\alpha \leq \beta$ , the successful connection is more likely to be established. Finally, the receiver can receive the message on a channel which is used by a sender. An example for this part can be:  $S_1$  and  $S_2$  obtain channels  $\{1, 2\}$  and  $\{2, 4\}$ , respectively. The broadcasting sequence of  $S_1$  should be  $\{1, 2, 1, 2\}$  for one period (cycle) and meanwhile,  $S_2$  generates its listening sequence as  $\{2, 2, 4, 4\}$ . Evidently they guaranteed to talk on channel 2. Besides, if multiple SUs broadcast the message through the same idle channel at one time, collisions happen. To optimize the problem, acSB implements an asynchronous forwarding policy before broadcasting (Fig. 3). Algorithm 1 elucidates the principle. Initially, a broadcaster  $S_n$  owns an essential channel set  $ECS_{S_n}$  in which broadcasting channels are arrayed with sequential time slots. Assume that  $S_n$  detects collisions on its selected channel. We denote the size of  $\Gamma$  as  $\iota_1$  and the size of  $ECS$  as  $\iota_2$  meeting  $\iota_2 \subseteq \iota_1$ . If  $\iota_2$  of  $S_n$  ( $\iota_2(S_n)$ ) is less than  $\iota_1$ , 0s are added to  $ECS_{S_n}$  until the length is equal to  $\iota_1$ ; then  $S_n$  generates a random number  $\theta$  in the range between 1 and  $\iota_1$ . After that,  $S_n$  carries out  $\theta$  times right-hand circular shift on expanded  $ECS_S$  and then obtains  $ECS'_{S_n}$ , which can reset the latest invoked time of the selected channel for different SUs. For example,  $S_1$  with its  $ECS$  of  $\{1, 2, 3\}$  and  $S_2$  with its  $ECS$  of  $\{1, 4\}$  are assigned random integers as 1 and 3. Since the size of total idle channel set is 5,  $S_1$  expands its  $ECS$  to be  $\{1, 2, 3, 0, 0\}$  and  $S_2$  obtains a channel set  $\{1, 4, 0, 0, 0\}$ . Finally,  $S_1$  and  $S_2$  will perform the broadcasting after receiving the message on their new channel sequences, such as  $\{0, 1, 2, 3, 0\}$  and  $\{0, 0, 0, 1, 4\}$  respectively.

Intuitively, acSB should outperform existing protocols since the selection of subset channel set can shorten transmission delays spent on channels and its collision avoidance policy can

address a problem of broadcasting and finally enhance a possibility of successful transmissions. Let us prove such inferences.

## 5 Performance evaluation

In this section, the performance of the proposed broadcast protocol is evaluated. The network is fixed with 10 unit side-length, where all SUs are randomly distributed.  $N_S$  varies from 5 to 25 and neighbor density  $\rho$  is changed from 0.1 to 1.0 depending on focus of investigations. The total number of system channels is  $M = 100$  and the probability of idle channel set is set to be an undetermined value meeting  $P < 0.5$ . In addition, each SU has random number of available channels between 1 and  $t_2$ . For transmissions, the radius of the sensing range and the transmission range are assumed to be the same, i.e. 1 unit length. The simulation result at each data point in the following graphs shows an average result of 10000 runs. The proposed acSB is compared with the Full Broadcast (FB) and traditional SB protocol. FB is popularly known as each SU can visit its available channels sequentially or randomly [Sureshkumar and Rajeswari (2014)], aiming at the high reach rate of messages. But, it comes at the cost of very long delays.

### 5.1 Critical metrics

#### 5.1.1 Transmission delay

Urgent events, such as traffic accidents, fire hazards, road repairs and so on, require a fast information dissemination to obtain quick responses and solutions. Transmission Delay ( $TD$ ) indicates the average duration that SUs broadcast a message on their selected channels, formulating by Eq. 3,

$$TD = t_2(S) \times T \quad (3)$$

where  $T$  is a minimum time used for broadcasting a message. According to this formula, a smaller  $t_2(S)$  can deduce a smaller  $TD$ .

#### 5.1.2 Message reach rate

Message Reach Rate ( $MRR$ ) indicates the ratio of successful message receiving. In most of cases whatever delay-tolerant or delay-demanding, a large  $MRR$  is preferable. For instance, if the traffic has been tied up for a long time, a car experiencing the event would like to broadcast the information as wide as possible in the area so that cars obtaining the message can avoid the congestion. The message reach rate is formulated as Eq. (4),

$$MRR = (N_S^r \div N_S) \times 100\% \quad (4)$$

where  $N_S^r$  is the number of SU receivers. The formula implies that a higher  $N_S^r$  causes a higher value of  $MRR$ .

#### 5.1.3 Broadcast collision

Broadcast Collision ( $BC$ ), namely the act of colliding, is a common issue discussed in wireless communications. Since it is probably let the quality of communications down, it should be largely controlled and minimized. In fact, redundancy and simultaneous rebroadcast are two typical reasons. The former occurs because there are many unnecessary copies of

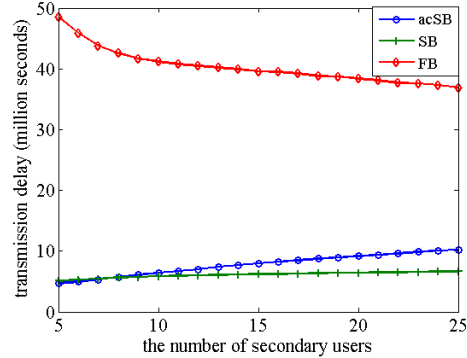


packets in the system waiting for the rebroadcasts; while the latter indicates multiple users attempt to use the same channel for transmissions at the same time. The problem challenges wireless communications, especially for infrastructure less applications. Hence, maintaining the reach rate of messages and the delay of broadcast fixed, to reduce the number of rebroadcasts and to avoid the simultaneous sending will be the main tasks.

### 5.2 The impact of network density

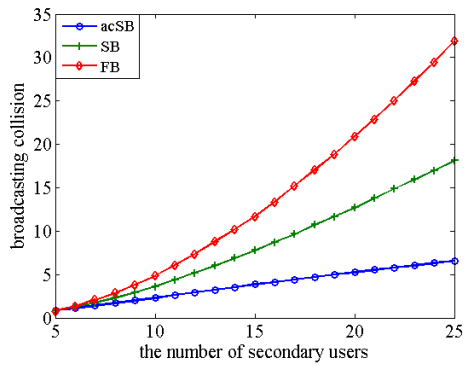
In a network, the increasing number of SU may lead to intensive channel contention. In this section, we will observe simulation results of the transmission time, the reach rate and collisions under different number of SUs with non-uniform channel availability.

As shown in Fig. 4, FB takes approx. six times more delays (40.32 ms) to broadcast a message in the system, compared with SB and acSB protocols (6.1ms and 7.74 ms) in average. The result can be clearly deduced by Eq. 3 that  $TD$  is actually subject to the value of  $t_2(S)$ . FB broadcasts the message on all available channels of SUs, namely  $t_2(S) = t_3$  while acSB requires a subset channels only, as  $t_2(S) < t_3$ . Therefore,  $TD$  of the same SU by acSB is much less than that by FB. Meanwhile, it is noticed in the figure that acSB outputs a longer average delay than SB does. Once there are possible broadcast collisions, acSB with an anti-collision algorithm allows a consistent manner instead of SB and thus its  $TD$  increases.

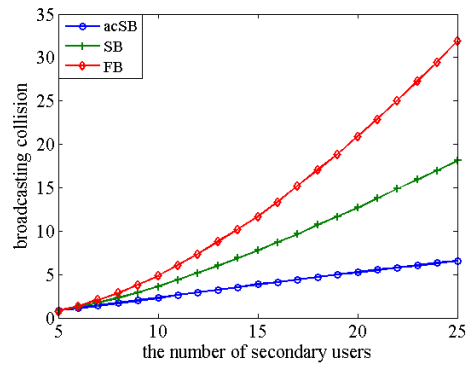


**Figure 4:** Impact of network density on  $TD$

Fig. 5 shows the message reach rate as a function of the number of secondary users maintaining the idle channels (e.g. 20) and neighbor density (e.g. 60%) fixed. When the number of SUs is up to 25,  $MRRs$  of acSB, SB and FB are 99.9%, 98.9% and 100% separately. acSB outperforms the other two schemes with possible reasons that it provides a collision avoidance scheme which increases  $N_5^r$  over SB. Meanwhile, it spent about 7.74 ms in average on channel broadcasting which is much smaller than that of FB.



**Figure 5:** Impact of network density on MRR

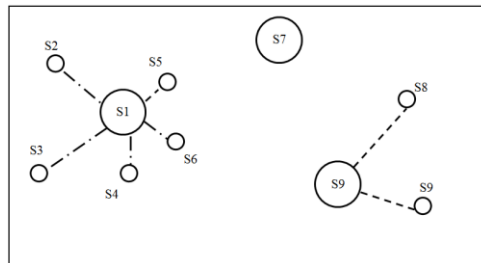


**Figure 6:** Impact of network density on BC

From the discussion in Section 2, the proposed acSB has an advantage that anti-collision policy for simultaneous message forwarding is implemented. In other words, an anticipation of acSB is to reduce the number of collisions while the reachability of messages stays at high point, e.g. 95% or even above. BC indicates the accumulated number of collisions. All BCs shown in Fig. 6 grow with the increasing number of SUs because the channel contentions become serious. At the data point that there are 25 SUs in the system, acSB owns an average collision about 6.5 which is 11.6 and 25.3 fewer than that of SB and FB respectively. As expected, acSB provides the least BC which implies a proof that acSB tackles the simultaneous message forwarding problem successfully without influencing other performance metrics.

**5.3 The impact of neighbor density**

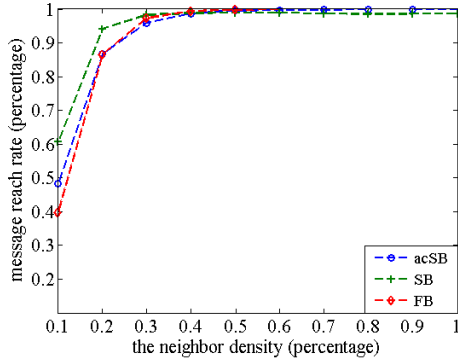
Since SUs may usually be randomly positioned in the areas, the performance of broadcasting should vary according to the type of distributions. Thus, the number of neighbors of a SU varies depending on the locations of SU neighbors. Neighbor density, namely  $\rho$ , is obtained by the number of neighbors over the total number of SUs in the system. Fig. 7 shows a network scenario which contains different neighbor density of 10 SUs. For instance,  $S_1, S_2, S_3, S_4, S_5, S_6$  are located very close to each other therefore the neighbor density of  $S_1$  is higher, i.e. 50% than that of  $S_9$  with neighbors  $S_8$  and  $S_{10}$ , as 20% only.



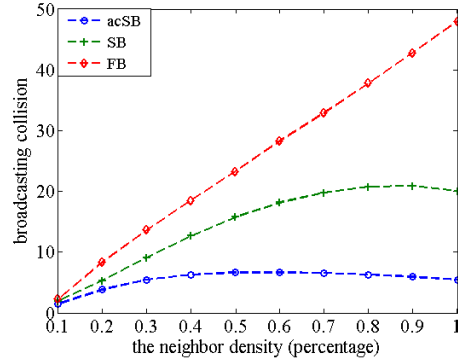
**Figure 7:** Different neighbor density of SU nodes

If the available channels are limited for a certain number of SUs, a large  $\rho$ , probably causes serious concurrent transmissions with same channel selections. Accordingly, simulations are performed in a network of 25 SUs with respect to the changing  $\rho$  from 10% to 100% so as to evaluate the impact of neighbor density on the performance of the proposed acSB.

In Fig. 8, the data points of *MRR* meeting  $\rho \geq 0.5$  stay above 99% in average for all protocols. With such message reach rates, comparatively speaking, acSB achieves the least *BCs* even though  $\rho$  becomes large (Fig. 9). It is noticed by these results that acSB actually does not perform well when the neighbor density is relatively low, e.g. less than 50%. This is because when the density is low, the collision problem is not serious (Fig. 9). Thus, collision avoidance scheme in acSB does not exert the advantage sufficiently. An confident anticipation is given that the network density (amount of SUs) and the neighbor density of each SU (amount of neighbors) is large enough, acSB outperforms traditional SB and FB protocols in addressing communication interferences caused by simultaneous message forwarding.



**Figure 8:** Impact of neighbor density on *MRR*



**Figure 9:** Impact of neighbor density on *BC*

## 6 Conclusion

This paper proposes a novel selective-based broadcasting protocol, acSB, to achieve efficient channel selection and collision avoidance in CR-AdHoc networks, where the non-uniform channel availability of SUs imposes serious challenges to select and maintain improvisational common channels for successfully agile message disseminations. The numerical results show that the acSB protocol obtains average collisions about 3.76 and 5.44 with the impacts of SU amount and neighbor density respectively while the scores of SB (8.4 and 14.4) and FB (13.4 and 25.6) in the corresponding order are relatively high. Besides, the acSB gains an advantage over the other two compared regarding moderate transmission delays (e.g. 7.74 ms) and high message reach rate (e.g. 99.5%).

**References**

- Al-Mathehaji, Y.; Boussakta, S.; Johnston, M.; Fakhrey, H.** (2015): CRBP: A broadcast protocol for cognitive radio ad hoc networks. *IEEE International Conference on Communications*, pp. 7540-7545.
- Arachchige, C. J. L.; Venkatesan, S.; Chandrasekaran, R.; Mittal, N.** (2011): *Minimal Time Broadcasting in Cognitive Radio Networks*. Springer Berlin Heidelberg.
- Cormio, C.; Chowdhury, K. R.** (2010): 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.
- Kondareddy, Y. R.; Agrawal, P.** (2008): Selective broadcasting in multi-hop cognitive radio networks. *Sarnoff Symposium*, pp. 1-5.
- Lin, Z.; Liu, H.; Chu, X.; Leung, Y. W.** (2011): Jump-stay based channel-hopping algorithm with guaranteed rendezvous for cognitive radio networks. *INFOCOM, 2011 Proceedings IEEE*, pp. 2444-2452.
- Oh, S. W.; Ma, Y.; Peh, E.; Tao, M.** (2017): *TV White Space: The First Step Towards Better Utilization of Frequency Spectrum*.
- Song, Y.; Xie, J.** (2014): QB<sup>2</sup>IC: A qos-based broadcast protocol under blind information for multihop cognitive radio ad hoc networks. *IEEE Transactions on Vehicular Technology*, vol. 63, no. 3, pp. 1453-1466.
- Song, Y.; Xie, J.** (2015): Bracer: A distributed broadcast protocol in multi-hop cognitive radio ad hoc networks with collision avoidance. *IEEE Transactions on Mobile Computing*, vol. 14, no. 3, pp. 509-524.
- Sureshkumar, A.; Rajeswari, M.** (2014): A survey of broadcast protocols in multi-hop cognitive radio ad hoc network. *International Journal of Science*, vol. 3, no. 11.
- Theis, N. C.; Thomas, R. W.; Dasilva, L. A.** (2010): Rendezvous for cognitive radios. *IEEE Transactions on Mobile Computing*, vol. 10, no. 2, pp. 216-227.