Priority-Based Spectrum Access in Cognitive D2D Networks for IoT

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

***Abstract*—Device to device (D2D) communication is essential in Internet of things (IoT). Cognitive radio (CR)-enabled device is a promising technique to address D2D communications. Among all the functions supported in IoT, priority-based spectrum access is less investigated yet urgently desired in IoT. An important operation to achieve this capacity is channel hopping (CH) based rendezvous which allows two CR users to meet each other on a common available channel. Unfortunately, existing CH methods cannot help CR users achieve priority-based spectrum access due to various shortcomings. In this paper, we propose PCH, a priority-based spectrum access protocol for cognitive D2D, which can be integrated with any existing CH algorithm. PCH can support priority transmissions with a significantly reduced CH delay, as compared with non-priority transmissions. More importantly, PCH can work under practical scenarios such as the D2D both with priority packets, the D2D handoff, and the overhead/energy constraint IoT. The merits of PCH are proved theoretically and validated against extensive simulations. To the best of our knowledge, this is the first work that investigates priority communications in cognitive IoT.**

I. Introduction

Device-to-device (D2D) communications enable networked devices to exchange information among each other without human control and therefore creates what is known as the Internet-of-Things (IoT) [1]. However, a large number of connected devices, as envisioned for the IoT, will create challenges in terms of spectrum scarcity and significant control overhead, which falls into the solution domain of cognitive radio networks (CRNs) [2]. Therefore, cognitive radio technology is very promising for D2D communications not only because of its dynamic spectrum access capability and efficient spectrum utilization, but also due to its interference management, energy efficiency, and device heterogeneity. It is expected that devices equipped with cognitive radios will be indispensable in order to realize the vision of IoT [3].

In such a network, each cognitive radio-based device (CRD) can dynamically configure their operating parameters to use those channels that will not cause harmful interference to near-by communication-ongoing devices including CRDs and usual devices in other coexisting networks. Due to the changing distribution and activities of all devices in all networks, the available channel set (ACS) of a CRD may be time-varying with its location and time. Hence, unlike traditional wireless networks, a control channel that is commonly available to all CRDs in a CRN may not exist or cannot last for a long time. Instead, any two CRDs with heterogeneous ACSs need to perform channel hopping (CH) [4]–[9] in a decentralized manner to *rendezvous* with each other, i.e., to meet each other on a common available channel before their data transmission.

Although CH enables the spectrum access for cognitive D2D networks, the obligated CH process prevents an important function, *priority-based spectrum access*. Though there is hardly any existing work focusing on priority-based communications in such networks, it is highly desired in IoT. As the fast growth of IoT applications [10], CRD may deal with different types of data or data with different delay constraints. Some applications such as the weather forecast have a higher delay tolerance than the real-time video stream. On the other hand, data of an urgent event need to have a higher priority to be sent through the network. However, all these high priority packets may lose their priority in a cognitive D2D network due to the lack of priority-based spectrum access support.

When two CRDs with the same destination CRD, the one with higher priority data cannot connect to the destination device faster via existing CH methods. In order to differentiate the CH delay for packets with different priorities, one possible way is to use different CH algorithms for different packets, i.e., the faster CH algorithm for the higher priority packet. However, only when both the source CRD and the destination CRD use the same CH algorithm, can the CH process leads to a rendezvous. Unfortunately, the recipient in such networks does not know the priority of a packet before CH. Then, the algorithms used by the D2D pair may be different and thus cannot even guarantee a rendezvous. Another way is to design a downgraded version of a CH algorithm for low priority packets. In other words, the purpose of distinguishing different priorities is achieved by the cost of manually slowing down regular D2D rendezvous.

Besides, the D2D pair may need further actions after rendezvous to achieve priority-based spectrum access. For example, the D2D pair may want to transmit data with a higher priority on the channel with a faster speed. This is realistic especially in emerging 5G networks where users can access several different bands [11]. However, under existing CH efforts, maximizing rendezvous channel diversity [12], [13] is one of the primary design goals. In most cases, the rendezvous channel is not the desired fast channel. Then, the D2D pair needs to handoff [14] to the corresponding desired channel in order to transmit the higher priority data, which adds additional handoff delay [15].

Moreover, some practical scenarios and constraints also need to be considered for the design. For example, if a priority-based method is adopted in the network, for the D2D pair when both are source devices with priority packets to send, can they still connect with each other? If so, can they still rendezvous no later than normal scenarios? Also, will the design need additional actions with extra overhead and energy consumption? Will this additional cost influence the overall performance?

In this paper, we propose a priority-based CH (PCH) MAC protocol which, to the best of our knowledge, is the first distributed cognitive MAC that can achieve priority-based spectrum access in cognitive D2D networks for IoT. The salient features of PCH are summarized as follows:

1) It is both mathematically proved and simulation validated that priority-based spectrum access can be achieved by PCH without causing any longer delay to non-priority D2D rendezvous.
2) PCH can be integrated with any CH algorithm. Even if a CH algorithm with a faster rendezvous delay is proposed in the future, PCH can still adopt it and generate even better performance for priority packets.
3) PCH enjoys fewer handoff operations after rendezvous due to the fact that the rendezvous channel is most of the time the desired channel.
4) PCH can still hold its merits when the destination device is also in the priority-based CH mode.
5) The operation overhead and energy consumption of PCH maintain the same order as the original overhead and consumption in most cases.

The rest of this paper is organized as follows. The system model and design challenges are described in Section II. In Section III, both the theoretical analysis and the PCH construction are explained with details. Simulations results of our proposed protocol are shown in Section IV, followed by the conclusions in Section V.

II. Problem Description

A. System Model

The system considered in this paper divides time evenly into time slots with homogeneous size and consists of a finite number of CRDs and other usual devices in other coexisting networks which can operate on a set of orthogonal channels denoted as $\boldsymbol{C} = \{c_1, c_2, c_3, ..., c_M\}$.

1) Spectrum Access Process: Each time a CRD wants to transmit or listens for a potential transmission, all other transmission-ongoing devices within its sensing range are regarded as primary users (PUs) in CRNs. The CRD senses the primary network to get its available channel set (ACS), S_{CRD}, for the sake of interference-free to PUs, just like a secondary user (SU) in CRNs. This process is done periodically, and the ACS is updated to avoid channel status change due to PU activities. Then, each CRD generates a sequence of channels from its ACS based on the adopted CH algorithm. In each time slot, it hops onto one channel following the sequence. The number of time slots a source CRD hopped before rendezvous on the same channel with its destination CRD is called time to rendezvous (TTR).

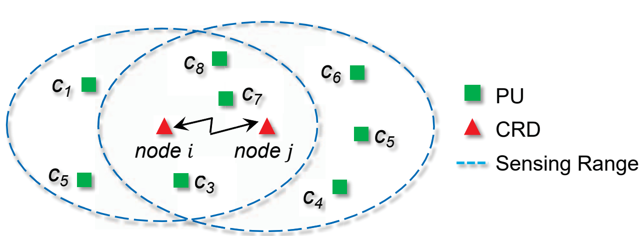
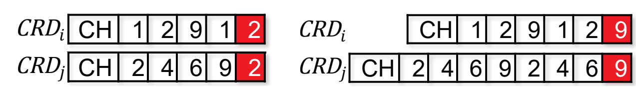


Fig. 1. Heterogeneous ACS scenario.

Note that the ACS of each CRD may not be homogeneous due to the diversity of each CRD's location and sensing capability. As illustrated in Fig. 1, CRD_i and CRD_j have different sensing ranges and the PUs distributed within their sensing ranges are using channels from the set $\{c_1, c_2, c_3, ..., c_9\}$. Therefore, the ACSs of CRD_i and CRD_j in this case are $S_i = \{c_2, c_4, c_6, c_9\}$ and $S_j = \{c_1, c_2, c_9\}$, respectively. Moreover, without loss of generality, we assume that any two CRDs within the transmission range of each other have at least one common available channel.

2) CH Algorithm: The state-of-the-art CH algorithms for heterogeneous ACSs can guarantee the rendezvous for any two SUs who have at least one common available channel within finite time slots. Under these algorithms, the maximum TTR (MTTR) is in the order of M^2, where M is the total number of channels in the primary network, and the expected TTR (ETTR) is $O(M)$.

In our system, we do not require synchronous CH which assumes the rendezvous pair to start CH at the same time. For example, using the example in Fig. 1 and assuming that CRD_i is a source SU and its destination SU is CRD_j, we adopt the CH method proposed in [6] which can guarantee the rendezvous for heterogeneous ACSs in an asynchronous manner. As illustrated in Fig. 2, CRD_i can rendezvous with CRD_j no matter their CH is synchronous or not.



(a) synchronous CH, TTR=5 (b) asynchronous CH, TTR=6

Fig. 2. An example of synchronous and asynchronous CH.

We denote $R^d(P(S_i), P(S_j))$ as the rendezvous time of the source CRD_i with its destination CRD_j. $P(S_x)$ is the permutation function of a given ACS, which depends on the CH algorithm adopted. For example, $P(\{c_1, c_2, c_3\}) = \{3, 3, 3, 2, 2, 1\}$. The hopping sequence follows the permutation periodically. d is the offset slots of CRD_j which can also be a negative integer. Once d, $P(S_i)$, and $P(S_j)$ are determined, a CRD can have one and only one TTR. In the above example, $R^0(\{1, 2, 9\}, \{2, 4, 6, 9\}) = 5$ and $R^2(\{1, 2, 9\}, \{2, 4, 6, 9\}) = 6$.

3) Link Speed Setting: In emerging networks such as the 5G network, users may require different spectrum bands to serve different purposes. Since different bands are centered at different frequencies with different bandwidths, and cognitive radio usually uses different modulation techniques and protocols on its operating channel, the data rate, (or, the link speed) of different channels varies.

Denote $\boldsymbol{V} = \{v_1, v_2, v_3, ..., v_M\}$ as the link speeds of the operational channels in set $\boldsymbol{C}$. For a given packet with length L, the transmission time on each channel is $t_i = \frac{L}{v_i}$, $v_i \in \boldsymbol{V}$. In our model, we reorder the channels in $\boldsymbol{C}$ by their link speed in a **fast-to-slow** order: $S_{SU} = \{c_1, c_2, c_3, ..., c_M | t_1 \leq t_2 \leq ... \leq t_M\}$.

B. Design Challenges and Requirements

In cognitive D2D networks, different packets may have different delay requirements. One intuitive way is to give the packet with a low delay tolerance a higher priority by sending the packet on a higher speed channel. Generally, if there are totally k priorities for different data packets,

$P_{CRD} = \{p_1, p_2, ..., p_k\}$ (ranked based on priority from **high to low**), a CRD should use the corresponding k fast channels $\{c_1, c_2, ..., c_k\}$ for each priority. Such design is illustrated in Fig. 3, where $\{\tau_1, \tau_2, ..., \tau_k\}$ is the spectrum access delay (TTR) for each priority packet. In order to fulfill the design purpose, it requires

$$\tau_i + t_i \le \tau_j + t_j, \qquad i < j. \tag{1}$$

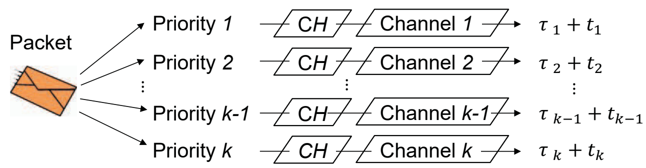


Fig. 3. Ideal design for packets with different priorities.

However, the reality cannot support this ideal design. First, as mentioned in the system model, the ACS of each CRD changes with time and location. Thus, a packet with a fixed priority should not be associated with a fixed transmission channel. Instead, a source CRD (CRD_i) should send its highest-priority packet to its fastest available channel among all its common available channels and so forth: $c(k) = S_{i,j}(k)$, where k is the corresponding priority and $S_{i,j} = S_i \cap S_j$.

Since the rendezvous channel is usually very random (see the example in Fig. 2) and usually not the corresponding channel of a particular priority, the rendezvous pair may need to handoff to the desired channel after the rendezvous if necessary, which causes extra delay.

More importantly, the TTR of each packet may vary a lot. Based on the research in [16], in practical CRNs, the TTR is in the same order and sometimes even longer than the packet transmission time. Such circumstances may cause the inequality in (1) invalid. Therefore, as illustrated in Fig. 4, in order to satisfy (1), the desired design should guarantee that *the packet with a priority enjoys a less or at least the same TTR as compared to the packet without priority*.

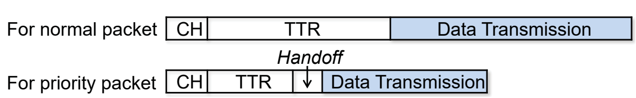


Fig. 4. The desired CH design for priority packet.

III. PCH Design

A. Motivation

We first extract two facts from the analysis of the CH process in the last section. One is the homogeneous time-slot size. In all existing CH schemes, different SUs must use the equilong time slot (T) to guarantee the same unit of their hopping sequences. For example, when SU_i's hopping sequence is $\{1, 2, 1, 2, ...\}$ and SU_j's hopping sequence is $\{2, 2, 3, 3, ...\}$, it implies that the duration time on each of their hopped channel $d(channel)$ is the same. Only when $d(c_1) = d(c_2) = d(c_3) = T$, can the rendezvous-guaranteed sequences take effect. The other fact is heterogeneous transmission time. As we explained in Section II, the transmission time of a packet on different channels with different link speeds has a correspondingly different value.

Consider the source CRD in a time slot stays on a fast channel. The sojourn time of this CRD should be the same as in other time slots when it stays on other relatively slower channels. On the other hand, the packet transmission time on this fast channel is shorter than that on other slower channels. Therefore, there is ample idle time for the CRD in such a time slot. Sometimes, the idle time may be long enough for the CRD to take another transmission on another relatively fast channel.

In addition, the channel switching time and energy cost are relatively small compared with the average packet transmission time on all channels and its energy consumption, according to [17], [18]. Therefore, it is possible to ask the source CRD to hop onto more than one fast channels in one time slot. In this way, a source CRD can keep revisiting some fast channels in some time slots during a CH process. Then, the increased visiting time of those fast channels augments their probability to become the rendezvous channel and reduces the TTR. Meanwhile, the original hopping sequence can be kept. Thus, the ETTR is reduced without worse cases (larger than original MTTR) generated. Moreover, this claim holds under any CH algorithm. We give the proof as follows.

Lemma 1. *Let CRD_i and CRD_j be the source CRD and the destination CRD, respectively. An arbitrary CH algorithm is given. Suppose under a CH process, $\tau = R^d(P(S_i), P(S_j))$. If the source CRD visits multiple channels in one time slot when feasible and the new TTR is represented by τ'. Then, $\forall d$, S_i, and S_j, $\tau' \le R^d(P(S_i), P(S_j))$.*

Proof. Without loss of generality, let $P(S_i) = \{c_{k1}, ..., c_{ki}, ...c_{kN}\}$, where c_{ki} is the channel at the ith position in the permutation and N is the number of available channels. Denote τ^1 as the TTR when CRD_i visits one more channel, say, c_{kj}, during the ith time slot in each periodical hopping round. Then, the new sequence can be expressed as $P^1(S_i) = \{c_{k1}, ..., (c_{ki}, c_{kj}), ..., c_{kN}\}$ where $i, j = 1, 2, ..., N$ and $i \ne j$. Depending on CRD_j's hopping sequence, $P^1(S_i)$ could have the same TTR as using $P(S_i)$, or the same TTR as using $\{c_{k1}, ..., c_{kj}, ..., c_{kN}\}$. Thus, we can derive that

$$\begin{aligned} \tau^1 =& R^d(\{c_{k1}, ..., (c_{ki}, c_{kj}), ..., c_{kN}\}, P(S_j)) \\ =& min\{\ R^d(\{c_{k1}, ..., c_{ki}, ..., c_{kN}\}, P(S_j)), \\ & R^d(\{c_{k1}, ..., c_{kj}, ..., c_{kN}\}, P(S_j))\ \} \\ \le& R^d(\{c_{k1}, ..., c_{ki}, ..., c_{kN}\}, P(S_j)) = \tau. \end{aligned} \tag{2}$$

Iteratively, we can get $\exists y > 0$,

$$\tau^y \le \tau^{y-1} \le ...\tau^2 \le \tau^1. \tag{3}$$

Since $\tau' \in \{\tau^1, \tau^2, ..., \tau^y\}$, then, $\tau' \le \tau$. □

In this proof, $R^d(P(S_i), P(S_j))$ represents all possible CH cases which also include the worst case. Thus, $max(\tau') \le MTTR$. On the other hand, since $ETTR = function(\sum_d \sum_{S_i} \sum_{S_j} R^d(P(S_i), P(S_j)))$, we get $E(\tau') \le ETTR$.

Up to now, when the source CRD proceeds CH in such a manner for packets with high priorities and follows the original CH for normal packets, the design requirement shown in (1) and the effect in Fig. 4 can be achieved in all cases.

B. Refined Design

Next, we refine the proposed method into details to gain better performance based on some latent requirements.

1) Design for the Shortest Delay: From (3), when y is larger, the new CH delay decreases. In other words, the optimal PCH design should visit multiple channels in any feasible time slot as much as possible. For a source CRD, suppose $S = \{c_{k1}, c_{k2}, ..., c_{kN}\}$, and let $\boldsymbol{d} = (d_{k1}, d_{k2}, ..., d_{kN})$ represent the corresponding minimum duration time on each available channel. Since we rank the channels on a fast-to-slow order in our system model, we have

$$d_{k1} \leq d_{k2} \leq ... \leq d_{kN}. \tag{4}$$

For each channel c_{ki}, there is an indicator vector $\boldsymbol{x}(k_i|S)$ showing which other channel can be co-visited with c_{ki} in one time slot. $\boldsymbol{x}(k_i|S) = [l_1^{ki}, l_2^{ki}, ..., l_N^{ki}]^T$, where $l_j^{ki} \in 0, 1$, $j = 1, 2, ..., N$. If $l_j^{ki} = 1$, it means that c_{kj} can be co-visited with c_{ki} in one time slot. By default,

$$l_i^{ki} = 1. \tag{5}$$

As mentioned in the motivating example, the vector should satisfy

$$\boldsymbol{dx}(k_i|S) + T_s \sum_{j=1}^{N} l_j^{ki} \leq T. \tag{6}$$

Therefore, we can minimize the CH delay by maximizing y:

$$\underset{\boldsymbol{X}(S)=[\boldsymbol{x}(k_1|S)\ \dots\ \boldsymbol{x}(k_N|S)]}{\text{Maximize}} \quad y = \sum_{i=1}^{N}\sum_{j=1}^{N} l_j^{ki}, \tag{7}$$
$$\text{subject to} \quad (4), (5), \text{and } (6).$$

Since the indicator vectors of different channels are independent from each other, the problem can be simplified as getting the maximum $\sum_{j=1}^{N} l_j^{ki}$ for a given channel c_{ki}. Consider assign the value '1' to either l_j^{ki} or l_{j+1}^{ki}. Then, the combination of (l_j^{ki}, l_{j+1}^{ki}) could be either $(1, 0)$ or $(0, 1)$. Take them into (6) separately. Because of (4), the left side of (6) calculated with the $(1, 0)$ combination is less than that with the $(0, 1)$ combination. A smaller value of the left side means that we can possibly assign more '1's to $\boldsymbol{x}(k_i|S)$. Iteratively, the optimal solution is to assign '1' to the element in the vector from left to right until (6) cannot be satisfied. Finally, for a given channel c_{ki}, its co-visiting channels $Co(c_{ki}|S)$ can be expressed by the derived $\boldsymbol{x}(k_i|S)$:

$$Co(c_{ki}|S) = \boldsymbol{S} \cdot \boldsymbol{x}(k_i|S)^T. \tag{8}$$

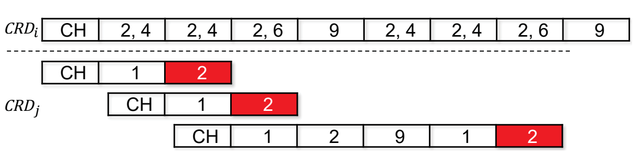


Fig. 5. A PCH example.

Consider the same example in Fig. 2. Suppose $d_i = i$, $Ts = 1$, and $T = 10$. Using the optimal PCH construct method given above, the result is shown in Fig. 5. Now, we have $R^0 = 2$, $R^1 = 3$, and $R^2 = 7$. Each delay is smaller than the previous result (5, 4, and 8, respectively).

2) Design for Same-Role Cases: In previous discussion, we suppose that the role of the destination CRD is either a listening CRD or a source CRD with normal packets to send. However, in practical IoT, the destination CRD could also have priority packets to transmit to another CRD, namely the same-role case.

When the destination CRD is also in a PCH process, both CRDs can visit multiple channels in one time slot. If there is no proper mechanism, even when they are in the potential rendezvous time slot, they may miss each other due to different presence time on their potential rendezvous channel, as illustrated in Fig. 6.

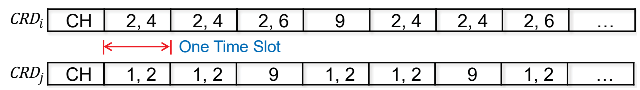


Fig. 6. Rendezvous failure in the same-role case.

In order to guarantee their rendezvous in this case, for each channel, there should be a unified hopping moment for all CRDs in a time slot. Note that the ACS of each CRD is a subset of the total channel set in the primary network. Imagine that there is *a virtual CRD* in the network *whose ACS contains all the channels in the primary network*, i.e., $S = \boldsymbol{C}$. Let this CRD adopt our PCH design. For a given channel c_i, its co-visiting channels $Co(c_i|\boldsymbol{C})$ in a time slot can be derived by (8). Arrange the hopping sequence within a time slot by channel indexes. Then, the moment the CRD hops onto c_i in the time slot is

$$\tau_s(c_i) = \sum_{j=1}^{i-1} (T_s + d_j) l_j^i. \tag{9}$$

This is also the unified visiting moment of c_i in PCH. Since for any other CRD, its ACS is a subeset of $\boldsymbol{C}$, its co-visiting channels of c_i are also a subset of $Co(c_i|\boldsymbol{C})$. In this way, a CRD can confirm the visiting moment of c_i in a time slot.

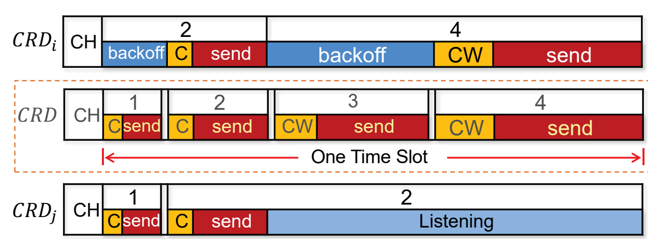


Fig. 7. MAC for both sender soles.

After finding the right moment to hop onto each co-visiting channel in a time slot independently, the rendezvous pair in the same-role scenario can perform even better due to the destination CRD's co-visiting contribution. It is equivalent to enlarge the value of y in Lemma 1's proof, which further decreases the ETTR and MTTR of the original CH algorithm. Again, we use the same example to illustrate our design as shown in Fig. 7. In the figure, CW refers to the *contention window* designed in the CRD MAC layer [19]. Also, note that it is *NOT* necessary to require the time slots or the mini time slots of each SUs to be *strictly synchronized* [20]. Both CRD_i and CRD_j visit channel 1, channel 2, or channel 4 at the corresponding unified visiting moment (illustrated using the virtual-ideal CRD in the middle) in their first time slot. In this case, they will rendezvous on channel 2 with $R^0 = 1$ which is further reduced as compared to the case without the destination SU's participation in PCH when $R^0 = 2$.

Note that a shorter CW can be assigned to the CRD with a higher priority packet. In this way, the packet with a higher priority has a higher chance to be sent first.

C. Protocol Details

Algorithm 1: The PCH protocol for CRD_x

```
Input: ACS of CRD_x S_x, data packet, and adopted CH algorithm;
1: Rank channels in S_x according to their link speeds on a
   fast-to-slow order;
2: Generate the hopping sequence P(S_x);
3: τ = 0;
4: while not rendezvous do
     τ = τ + 1;
     if no packet then
       Hop onto P(S_x)(τ);
       Listening to the channel;
     if packet without priority then
       Hop onto P(S_x)(τ);
       CW; Send message; Listening;
     if packet with priority k then
       c_j = P(S_x)(τ);
       Get Co − ch. = Co(c_j|S_x) using (8);
       i=1;
       while not rendezvous & Co − ch.(i) ∈ Co(c_j|S_x) do
         Get t = τ_s(Co − ch.(i)) using (9);
         Hop onto Co − ch.(i) at moment t;
         CW(k); Send message; Listening;
         i = i + 1;
       Handoff to their kth fastest common channel;
5: Data Transmission;
```

Algorithm 1 gives the entire protocol showing our hopping-channel reconstruct design based on packet priority. As we can see, our proposed PCH is very easy for implementation for any given CH algorithm. Meanwhile, the design is also independent of the CRDs with non-priority packets.

IV. Performance Evaluation

In our simulation, 1) CRDs and regular devices are randomly distributed in the simulation area; 2) each regular device is randomly assigned a channel when a new packet needs to be transmitted; 3) source CRDs are randomly selected, and their packets are randomly assigned with or without priority; and 4) each source CRD randomly chooses a CRD within its transmission range as its destination CRD. The parameters used in our simulation are listed in Table I.

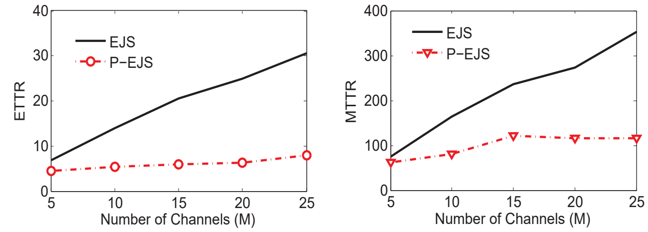
A. Performance Comparison

We adopt a state-of-the-art CH algorithm, Enhanced Jump-Stay (EJS) [5], to evaluate our PCH performance. The ETTR and MTTR of both the EJS and the priority-based EJS (P-EJS) are compared in Fig. 8. Both these protocols can achieve a 100% successful rendezvous rate. The results in Fig. 8 validate our proof that CRDs with a priority packet can always enjoy a lower spectrum access delay including the worst case.

TABLE I
SIMULATION PARAMETERS

Simulation time	10000 slots
Simulation area	50 m $\times$ 50 m
Number of usual/cognitive devices	100/50
CRD sensing radius	10/7 m
Average packet arrival rate	50 pkt/s
Average packet size	50 slots
Channel data rate	$10/i$ Mbps on channel i
The transmit power	$15 - 0.5i$ dbm on channel i [18]
Channel switching time/energy	200 μs / 20 μJ [17]

Moreover, PCH has a better resistance to the primary network change, e.g., when the number of total channels increases.



(a) ETTR performance. (b) MTTR performance.

Fig. 8. TTR vs. M in priority-based EJS

B. Impact of Different Factors

Fig. 9 shows the ETTR performance of both the PCH and the PCH under the same-role scenario with different common available channels. We use γ to represent the common available channel rate of a CRD, $\gamma = \frac{\#\text{of common available channels}}{\#\text{of available channels}}$.

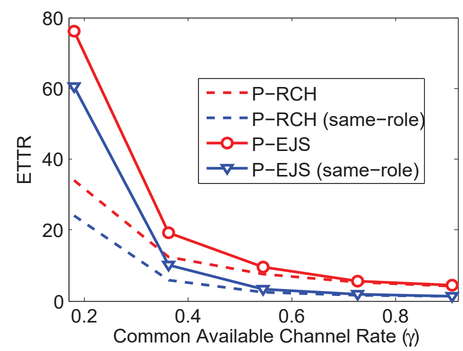
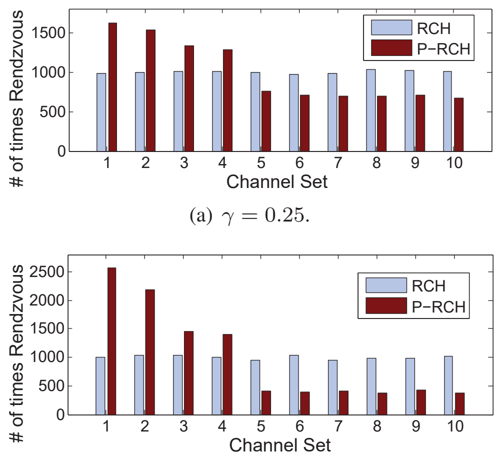


Fig. 9. Impact of the same-role CH.

1) Impact of the Common Available Channel Rate: When γ increases, the ETTR of both PCHs decreases, which satisfies the initial goal of CH. However, the ETTR does not change much when $\gamma > 0.5$. This is because that the ETTR performance mainly depends on the involvement of the fast channels not the number of common available channels. When there are a sufficient number of fast channels, the ETTR performance is dominated by their re-occurrences.

2) Impact of Same-Role Cases: When both CRDs have priority packets to send, PCH helps them rendezvous with each other with even less delay. From Fig. 9, PCH always performs better under the same-role scenario no matter using the Random CH (RCH) or EJS algorithm. Since our design enables same-role CRDs to have the synchronous visiting moment on the common channel within a time slot, the CRD pair enjoys another mini-CH round in each time slot.

C. Rendezvous Channel Distribution



(a) $\gamma = 0.25$.

(b) $\gamma = 0.75$.

Fig. 10. Rendezvous-channel distribution with different γ.

We record the number of times for each channel becoming the rendezvous channel. The rendezvous-channel distribution is shown in Fig. 10. We summarize: *i)* the chance to be the rendezvous channel is evenly distributed over all channels with RCH under various network conditions, which accords with its random feature; *ii)* for a CRD with a priority packet to send, it is more likely to rendezvous on those fast channels (channel 1-4), especially when γ is high, which validates our analysis; and *iii)* since there is no need to handoff if the rendezvous channel is fast, PCH reduces the possible handoff delay.

D. Energy Consumption Performance

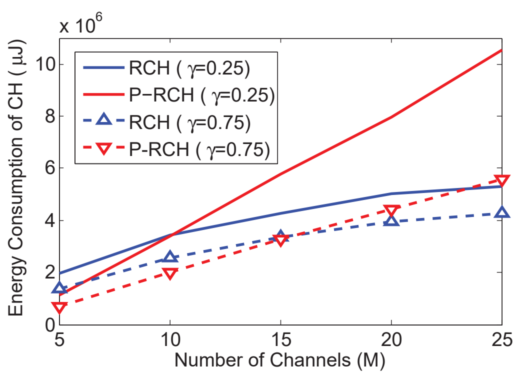


Fig. 11. Energy consumption during CH for different packets.

From Fig. 11, when there are more channels in the system with a small γ, energy cost increases under the PCH since the TTR is large. Nevertheless, when the number of channels is small, the consumption of priority spectrum access is less than that of regular access, despite the low γ. On the other hand, when γ is large, the energy consumption of PCH is very close to normal RCH due to the benefits of the reduced TTR by PCH. In this case, PCH can be used without constraint when the number of channels is less than 15.

V. Conclusion

In this paper, an efficient and adaptive priority-based protocol which can be integrated with any existing CH algorithm, PCH, is proposed. The proposed design is based on co-visiting fast channels within each feasible time slot. Both mathematical proof and simulation results showed that PCH can achieve shorter spectrum access delay for packets with higher priority in cognitive D2D networks. Besides, we further refined and revised our design to deal with different practical issues. Simulation results validated our design goals in various IoT environments. For the first time, the priority-based spectrum access is achieved in cognitive D2D networks for IoT.

References

[1] L. Atzori, A. Iera, and G. Morabito, “The Internet of Things: A survey,” *Computer Networks*, vol. 54, no. 15, pp. 2787–2805, 2010.

[2] I. 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, pp. 2127–2159, 2006.

[3] A. Aijaz and A. H. Aghvami, “Cognitive machine-to-machine communications for Internet-of-Things: A protocol stack perspective,” *IEEE Internet of Things Journal*, vol. 2, no. 2, pp. 103–112, 2015.

[4] Y. Song and J. Xie, “QB^{2}IC: A QoS-based broadcast protocol under blind information for multihop cognitive radio ad hoc networks,” *IEEE Trans. Vehicular Technology*, vol. 63, no. 3, pp. 1453–1466, 2013.

[5] Z. Lin, H. Liu, X. Chu, and Y.-W. Leung, “Enhanced jump-stay rendezvous algorithm for cognitive radio networks,” *IEEE Communications Letters*, vol. 17, no. 9, pp. 1742–1745, 2013.

[6] L. Chen, K. Bian, C. Liu, J.-M. J. Park, and X. Li, “A group-theoretic framework for rendezvous in heterogeneous cognitive radio networks,” in *Proc. MobiHoc*, 2014, pp. 165–174.

[7] I. Chuang, H. Wu, and Y. Kuo, “A fast blind rendezvous method by alternate hop-and-wait channel hopping in cognitive radio networks,” *IEEE Transcations on Mobile Computing*, vol. 13, no. 10, pp. 1536–1233, 2014.

[8] X. Liu and J. Xie, “SUBSET: A joint design of channel selection and channel hopping for fast blind rendezvous in cognitive radio ad hoc networks,” in *Proc. IEEE SECON*, 2015, pp. 426–434.

[9] Y. Song and J. Xie, “BRACER: A distributed broadcast protocol in multi-hop cognitive radio ad hoc networks with collision avoidance,” *IEEE Transcations on Mobile Computing*, vol. 14, no. 3, pp. 509–524, 2015.

[10] M. Shah, S. Zhang, and C. Maple, “Cognitive radio networks for Internet of Things: applications, challenges and future,” in *Proc. International Conference on Automation and Computing (ICAC)*, 2013, pp. 1–6.

[11] E. Hossain and M. Hasan, “5G cellular: key enabling technologies and research challenges,” *IEEE Instrumentation and Measurement Magazine*, vol. 18, no. 3, pp. 11–21, 2015.

[12] K. Bian, J.-M. J. Park, and R. Chen, “A quorum-based framework for establishing control channels in dynamic spectrum access networks,” in *Proc. MobiCom*, 2009, pp. 25–36.

[13] K. Bian and J.-M. Park, “Maximizing rendezvous diversity in rendezvous protocols for decentralized cognitive radio networks,” *IEEE Transcations on Mobile Computing*, vol. 12, no. 7, pp. 1294–1307, 2013.

[14] Y. Song and J. Xie, “Proactive spectrum handoff in cognitive radio ad hoc networks based on common hopping coordination,” in *Proc. IEEE INFOCOM Workshops*, 2010.

[15] X. Liu and J. Xie, “A self-adaptive optimal fragmentation protocol for multi-channel cognitive radio ad hoc networks,” in *Proc. IEEE GLOBECOM*, 2016.

[16] ——, “A practical self-adaptive rendezvous protocol in cognitive radio ad hoc networks,” in *Proc. IEEE INFOCOM*, 2014.

[17] S. Agarwal and S. De, “Impact of channel switching in energy constrained cognitive radio networks,” *IEEE Communications Letters*, vol. 19, no. 6, pp. 977–980, 2015.

[18] Cisco, “MX60/64W and Z1,” https://documentation.meraki.com/MR/Radio_Settings.

[19] X. Liu and J. Xie, “Contention window-based deadlock-free MAC for blind rendezvous in cognitive radio ad hoc networks,” in *Proc. IEEE GLOBECOM*, 2015.

[20] ——, “A slot-asynchronous MAC protocol design for blind rendezvous in cognitive radio networks,” in *Proc. IEEE GLOBECOM*, 2014.