A Simple Distributed Channel Allocation Algorithm for D2D Communication Pairs

Haitao Zhao, *Senior Member, IEEE,* Kaiqi Ding, Nurul I Sarkar, *Senior Member, IEEE,* Jibo Wei, *Member, IEEE*
and Jun Xiong

***Abstract*—Device-to-device (D2D) communication is a promising technology to achieve high spectrum efficiency, low energy consumption, and enhanced system capacity. In D2D communications, the channel allocation plays a crucial role to avoid interference, which usually involves a centralized coordination. With the increasing number of devices or terminals as well as the dynamic network environment, the computation and information exchange during the centralized channel allocation would be a big burden for wireless networks. We tackle this problem by proposing a distributed channel allocation method with blind rendezvous to enable collision-free concurrent transmissions over multiple channels. Specifically, we first present a receiver-oriented channel allocation algorithm to reduce interference and then propose a sender-jump blind channel rendezvous algorithm based on channel hopping techniques. The proposed channel allocation method allows each sender-receiver pair to obtain an appropriate channel for collision-free transmissions requires no centralized coordination. Theoretical analysis, simulation results and testbed experiments validate the proposed solution. Our findings reported in this paper may help network designers to contribute in the development of high-density Internet of Things.**

***Index Terms*—Multi-channel network, Device-to-Device (D2D) communication, channel allocation, distributed algorithm.**

I. INTRODUCTION

Device-to-Device (D2D) communication that provides direct connectivity among devices, sensors and intelligent terminals is a promising technology due to its advantages of high spectrum efficiency, low energy consumption, and enhanced system capacity [1]–[3]. As network density greatly increases, supporting a large number of devices can be a challenging task due to high signal collisions [4], [5]. Besides seeking for better collision avoidance schemes [6], multi-channel communication that exploits non-overlapping channels draws utmost attention [7], [8]. In contrast to the legacy D2D network that aims to maximize the network spectral and energy efficiency while sharing the same spectrum resource, in the multi-channel D2D network the shared spectrum is divided into multiple channels. This model can be used in many applications where dense nodes are deployed, such as cognitive sensor networks, IoTs and remote meters [9]–[11].

To enable multi-channel D2D communications while exploiting the collision-free properties of non-overlapping multiple channels, the main challenge is to design efficient channel allocation methods for sender-receiver pairs. With the increasing number of terminal nodes (e.g., sensors and smart phones) as well as the dynamic network environment, the computation and information exchange in the centralized channel allocation would be a big burden for the central coordination. Therefore, offloading channel allocation to terminal nodes would be a good solution. Existing work on channel allocation strategies mainly focus on allocating channel to each individual node in the network [12] assuming that all individuals will communicate with a base station (BS) or an access point (AP). And the objective of channel allocation is to achieve collision-free allocation [13], fairness [14], or optimization in resource allocation [15], [16]. Recent advances in channel rendezvous approaches [17] can provide efficient solutions to the problem of channel allocations for a pair of sender and receiver, by allowing nodes that operate on multiple channels to meet on a commonly available channel. However, most channel rendezvous approaches allow two node to achieve rendezvous on arbitrary channel as fast as possible without considering interference reduction, network fairness and resource optimization. To fill the gap between channel allocation and channel rendezvous and therefore enable high-efficient D2D communications, in this paper, we propose to make a pair of sender and receiver in multi-channel network meet on a proper channel while maintaining the good properties derived by channel allocation methods. Our main contributions in this paper are highlighted below.

Research problem: We address the channel allocation problem in multi-channel D2D communications to achieve the following desirable requirements. (i)Multiple channels are allocated to sender-receiver pairs instead of individual nodes to achieve collision-free con-current communications; (ii)The channel allocation is operated in a totally distributed manner; (iii) The channel allocation should bring little overhead to the network, requiring no information exchange or handshake between the neighboring nodes; and (iv) The channel allocation scheme should be scalable to large-scale networks and robust to new entrants, i.e., the new entrants to the network should have little effect on existing nodes.

Our solution: The channel allocation is offloaded to terminals from BSs or APs. Specifically, we first propose a distributed receiver-oriented algorithm for collision-free channel

---

This work is supported in part by the National Science Foundation of China (No. 61471376)

H. Zhao, K. Ding, Jibo Wei and Jun Xiong are all with the College of Electric Science, National University of Defense Technology, Changsha, China, 410073. Email: {haitaozhao, dingkaiqi15, wjbhw, xj8765}@nudt.edu.cn

Nurul I Sarkar is with the Department of IT and Software Engineering, Auckland University of Technology, Private Bag 92006, Auckland 1142, New Zealand. E-mail: nurul.sarkar@aut.ac.nz

Manuscript received March 10, 2018.

allocation and then investigate a sender-jump blind rendezvous algorithms to make the sender rendezvous with the receiver on the allocated channel. Both algorithms run on the terminal nodes that can work with their respective local information. And in order to enhance the scalability and the robustness, we propose a easy-to-implement yet effective learning process so that each send-receiver pair will rendezvous on a proper channel based on their experience.

The rest of this paper is organized as follows. Section II reviews relevant literature on channel allocation schemes for communication pairs. In Section III, we proposes distributed channel allocation and rendezvous algorithms for D2D communication pairs. In Section IV, we analyze the convergence property of the proposed algorithms. Section V validates our proposal by simulation and test-bed experiments. And a brief conclusion in Section VI ends the paper.

II. RELATED WORK

Existing work related to our research includes two threads: multi-channel allocation and multi-channel rendezvous.

A. Multi-channel allocation

The main purpose of channel allocation is to design a fast and smooth channel allocation mechanism to minimize the effect of collisions and consequently increase network throughput. This type of problem has been studied extensively in literature where various solutions are proposed at different requirements, which can be broadly classified as centralized solutions and distributed ones.

Centralized solutions: The centralized solutions mostly consider the problem of allocating channel to individual nodes, assuming broadcast [18] or a central station (such as Access Point) that can receive the messages on all channels [19]. Therefore, the central station can allocate the channel to each node based on global information. However, collecting the global information is a nontrivial burden to the network, considering dynamic networks with a large number of nodes. And it usually involves heavy centralized computation such as sub-gradient method [20], packing algorithm [21], coloring algorithm [22], genetic algorithm [23], dynamic programming [24], or needs additional monitoring nodes [25].

Distributed solutions: In distributed wireless networks, there is usually no such central station with supper computational ability, and therefore distributed solutions are more attractive. Under this case, a common control channel (CCC) would be of great help, since all nodes can negotiate channel allocation through the CCC [30]. However, a global CCC is not always available in distributed wireless networks. Besides, there are two main drawbacks of using CCC. First, the CCC is a bottleneck for a large number of nodes to exchange information on it. Second, the CCC is vulnerable to attack. To eliminate the dependance on CCC, researchers in [27]–[30] assume that each node has the ability to sense all the spectrum channels simultaneously. But individual node is usually equipped with a single radio transceiver and therefore cannot operate on different channels simultaneously. Vedantham R. et al. [31] used a distributed approach considering single radio in each node and a game theoretic approach is investigated in [32]. But, both works focus on maximizing link data rate instead of minimizing interference. Saifullah A. et al. [33] formulated a link-based fair channel allocation problem whose objective is to minimize the maximum interference. However, it brings great difficulty in channel hopping.

B. Multi-channel rendezvous

Most research on channel allocation focus on allocating channel to individual nodes, with the assumption that all nodes communicate with the BS or AP. With the increasing requirement for D2D communication, a rising problem is to establish a communication link between sender and receiver that may operate on different channels. As a solution, the multi-channel rendezvous scheme recently attracts a lot of research interests [17], [35]–[41]. And the research on this topic can be classified into three categories, namely, negotiation on control channel, common sequence hopping and blind rendezvous.

Negotiation on control channel : This solution generally includes two processes: control process and transmission process. The first process is to let the nodes meet on a predefined control channel to make agreements. And in the second process, successful pairs tune to their agreed channels for data transmissions [35], [36]. This solution may suffer from the congestion on the control channel in high-density networks because too many nodes could jump to the control channel for negotiation at the same time.

Common sequence hopping: In common sequence hopping, all nodes follow the same channel hopping sequence to switch their channels [37], [38]. These solutions need tight global clock synchronization. However, in practice nodes may quit or join the network at different time, and it is difficult for all nodes in the network to have such a tight global time reference.

Blind rendezvous: These solutions require no centralized controller, control channel or any information exchange. Each node chooses a set of available channels and then hops among these channels to rendezvous with its neighbors [39]–[41]. However, in all of these rendezvous solutions, the sender-receiver pairs achieve rendezvous on arbitrary channel. In other words, they only consider two nodes rendezvous on any of their common channels, but do not consider any fair or optimized resource allocation. And a more important problem is that they may bring possible interference with neighboring nodes in multi-user scenarios [41].

In multi-channel D2D communication networks, we are facing a new problem of allocating channels to sender-receiver pairs rather than individual nodes. In particular, we need to combine the properties of channel allocation and channel rendezvous, i.e., rendezvous a pair of sender and receiver at the most proper channel in a distributed manner. To the best of our knowledge, this problem has not been fully solved yet.

III. DISTRIBUTED CHANNEL ALLOCATION AND RENDEZVOUS ALGORITHMS

We consider a network with N nodes (Fig. 1), where the nodes can be static devices or mobile devices. M channels are shared among these N nodes in the network. Each node is free

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TVT.2018.2865817, IEEE Transactions on Vehicular Technology

to switch among these M channels, but can only use one channel to transmit at a time. In this network, spatially distributed nodes can perform two types of communications: (i) Many-to-one communication, that is, nodes exchange information with an AP; (ii) Device-to-device (D2D) communication, that is, a node communicates directly with its neighboring nodes. The AP can receive information on different channels, however, the nodes can only transmit or receive on one channel at a time. Therefore, for the D2D communication, a communication link must be established first between the sender and the receiver that may operate on different channels. This network model is applicable to sensor networks, IoTs in 5G cellular networks and unmanned aerial vehicle (UAV) swarms. The problem is that from an arbitrary initial network state, how the network obtains the interference-free blind rendezvous on each sender-receiver pair in a distributed manner, while consuming minimized time.

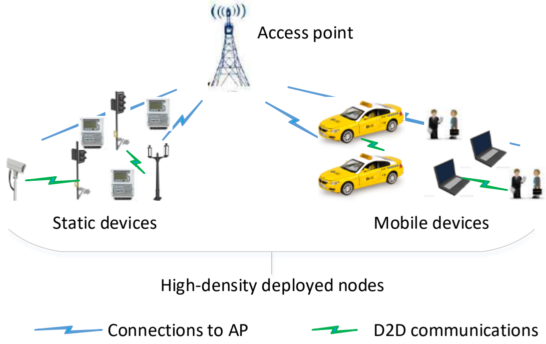


Fig. 1. The system model

To solve the above mentioned problem, we propose a distributed receiver-oriented channel allocation algorithm that allocates different channels to interfering nodes (i.e., potential receivers), followed by a sender-jump blind channel rendezvous algorithm to let the sender meet the receiver at the allocated channel. Note that the interfering nodes also include the hidden nodes besides the neighboring nodes, since the hidden nodes will also bring transmission collisions. And thereafter, each sender-receiver pair can concurrently transmit data on different channels, without causing any interference. In the proposed algorithms, the channel allocation simply relies on the choice of autonomous nodes, and the AP only informs the IDs of neighboring nodes.

A. Receiver-oriented channel allocation

1) Basic idea: Any active node in the network can be either a sender when it is sending data or a receiver otherwise (an idle node will keep listening to the medium for possible data receive, and therefore its role can be seen as a receiver). In our proposal, the channel is allocated to each node when it acts as a receiver. The basic idea is that each node maintains an Available Channel Set (ACS), and the available channels in each node's ACS are randomly sequenced (different nodes have different seeds). A node will switch sequentially onto the channel in the ACS and listen for a period T. If it receives no HELLO packets from other nodes, it will dwell on this channel and start to send HELLO packets every T seconds. When a node receives HELLO packets from another node, it will switch its radio to the next channel in ACS with a probability p.

2) Choice of p: The channel switch probability p can be set as a fixed value. For instance, $p = 1$ indicates that a node will definitely switch its channel once collision occurs and $p = 0.5$ means that the node has the same probability to stay on current channel as to switch to another channel once collision occurs. For a robust solution, we adopt a more efficient choice of p as discussed below.

(i) When there is no interference, $p = 0$; when there is interference, p is at most 0.5. We set p at most 0.5 because when two nodes collide on a channel, the best solution is one stay and the other jump (it is not necessary that both of them jump away).

(ii) When a node stays on a channel for a longer time, it will be less willing to switch channel, which means $p \to 0$ when $t \to \infty$, t is the time that this node has stayed on this channel;

(iii) On the other hand, if a channel is new to this node, it will more like to switch to another channel, which means $p = 0.5$ when $t = 0$.

We select a set of functions that satisfies these properties of p as follows,

$$p = 0.5^{(\alpha \cdot t+1)} \tag{1}$$

where α is the inert factor. Once a node switches to a new channel, it will reset its timer, namely reset $t = 0$.

The selection of p will result to inertia of nodes in channel hopping that elder nodes in the network are more willing to stay its current channel and new entrant is more willing to jump to a new channel. This property contributes to fast convergence when all nodes are cooperative, however, we have to also tackle the problem raised by malicious nodes.

Definition (Malicious nodes): Malicious nodes will dwell on the channel and start to send HELLO packets periodically on the channel, however, it will not switch the channel even if it receives any HELLO packet from other nodes.

Malicious nodes can be detected by any node x, when x receives HELLO packets from other nodes constantly on its dwelling channel for a pre-defined threshold times). After detecting the malicious nodes, x should reset p to the maximum value in order to jump to a new channel and therefore avoid the influence of the malicious nodes. In Fig. 2, we plot the curves of p with different values of α in (1). Our experiments show that setting $\alpha = 0.1$ is proper under the scenarios considered in this research.

The distributed channel allocation algorithm is shown in Algorithm 1, which contains two parallel threads. Thread A is a periodical thread to broadcast HELLO packet and accumulate the channel dwell time. Thread B is an interrupt thread triggered by receiving a HELLO packet.

Since each node only switches channel and listens to the channel without sending any packets before finding a free channel. The power consumption in the channel allocation process is mainly constituted by the electronic power consumption due to channel switch and that due to receiving

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TVT.2018.2865817, IEEE Transactions on Vehicular Technology

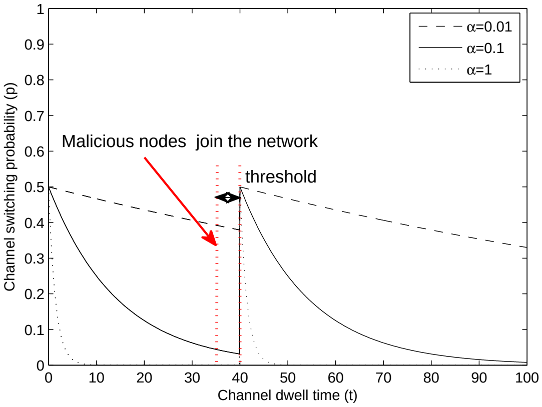


Fig. 2. Channel switch probability p (the threshold for detecting malicious nodes is 5).

Algorithm 1 : Receiver-oriented channel allocation.

```
//Initialization:
Set up the random available channel set (ACS), i.e., Channel($i$), $i = 1, 2, 3, ..., F$, where $F$ is the total number of available channels ($F \leq M$);
Initialize $t = 0$, $i = 1$; $c = 0$, $t_{rec} = 0$;
//Channel allocation:
Thread A: Local information broadcast
while Hello Timeout do
    Broadcast HELLO on the Channel($i$), which encodes its own ID;
    $t = t + 1$;
end while
Tread B: Channel switching
while Received a HELLO packet do
    if $t - t_{rec} < 2$ then
        $c = c + 1$ // Accumulate the times of continuously receiving HELLO;
    else
        $c = 0$
    end if
    if $c > 5$ then
        $t = 0$ // Detect the malicious node and reset $t$
    end if
    Calculate $p$ according to (1)
    if rand(1)<$p$ then
        Set $i = (i + 1) mod(F)$; // Switch the channel with the probability of $p$
        $t = 0$; //Reset $t$ after switching the channel
    end if
    $t_{rec} = t$; // Record the latest time to receive a HELLO
end while
```

possible RTS packets. Both power consumptions are much smaller comparing to the transmitting power.

Another problem we need to consider is the fairness problem to avoid one new entrant can never find a free channel. When a node cannot find a free channel after T_H rounds of channel-switch (T_H is a pre-defined threshold), it will send RTSs continuously along a randomly selected channel K. According to the former description, the older neighboring nodes that dwell on channel K think that a malicious node is attacking this channel and therefore jump away. By this means the new node can take over channel K and avoid the possibility that a new entrant can never acquire channel in the densely deployed network.

B. Sender-jump blind channel rendezvous

After the receiver-oriented channel allocation, the channel of each receiver (i.e., the node that has no data to send) can be seen as fixed and only need the sender (i.e., the node that has data to send) to rendezvous with it. Therefore, we propose a sender-jump blind channel rendezvous, in which the receiver will stay on the allocated channel and the sender will perform channel jump till it meets the receiver on the same channel. The channel rendezvous procedure is illustrated in Fig. 3, where the receiver dwells on Channel 3 and the sender dwells on Channel 4. When the receiver has data to send, it will jump in the sequence of Channels 2, 1, 3 and finally meet the receiver on Channel 3. This blind rendezvous requires no information on the target receiver's allocated channel or control channel.

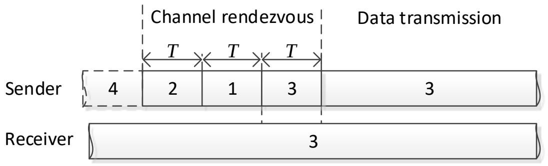


Fig. 3. Illustrating the concept of channel rendezvous.

The channel rendezvous algorithm runs on each node is shown in Algorithm 2. In this algorithm, the sender will send RTS to request for rendezvous only if it received a HELLO on this channel from the intended receiver, which will be confirmed by a CTS from the receiver. We adopt the RTS/CTS confirmation handshake before data transmission based on the following facts. First, there is a possibility that a sender hears a HELLO from its intended receiver while the receiver is still in the process of switching channels (i.e., this channel is not the receiver's dwelling channel). In this case, the immediate transmission from the sender to the receiver will fail. Second, competition will occur when more than two nodes are planning to transmit data to the same receiver. Therefore, a confirmation handshake has to be adopted to confirm the rendezvous. After that, the sender-receiver pair can commence transmissions.

Note that when two or more nodes are trying to rendezvous with one node, they will send RTSs simultaneously to one receiver. And therefore, collision will happen so that they cannot be feedback by the receiver. After experiencing the collision, they will choose to switch to another channel with their respective p or stay at this channel with the probability of $(1 - p)$. Thereafter, the probability that these nodes will collide again along the same channel will decrease due to the following two facts: (i) different nodes are with different p; and (ii) different nodes maintain different ACS.

Algorithm 2 : Sender-jump channel rendezvous.

```
//Initialization:
Given my ACS and the current channel index $i$; The ID of my intended receiver $R$;
Initialize $rendezvous = 0$;
//Channel rendezvous:
while $rendezvous == 0$ do
    $i = (i + 1)mod(F)$;
    Switch myself to Channel($i$) and listen for a period of Timeout;
    if Received a HELLO packet from $R$ then
        Using RTS/CTS to confirm rendezvous;
        if The rendezvous is confirmed then
            $rendezvous = 1$;
        end if
    end if
end while
```

C. Combining channel allocation and rendezvous

As discussed earlier, when a node has no data to send, it is naturally a receiver. And it will turn to sender once it has data to transmit. The node state transition diagram is shown in Fig. 4.

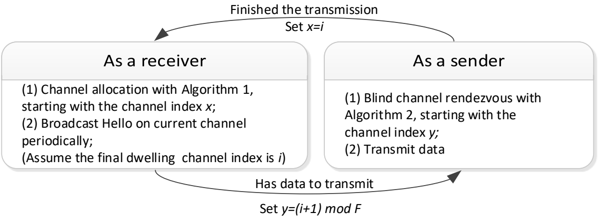


Fig. 4. States transition of a node.

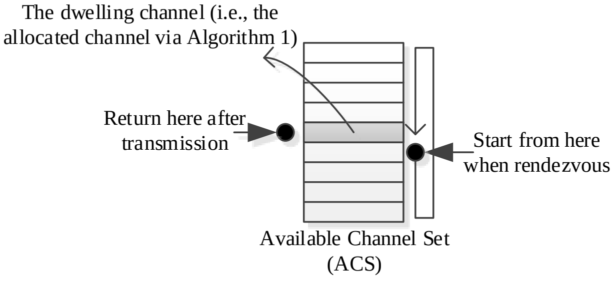


Fig. 5. Combining channel allocation and rendezvous.

As a receiver, a node will dwell on a channel that is different to any adjacent node's channel based on the channel allocation algorithm (Algorithm 1). When a node has data to send it becomes a sender. In order to achieve fast rendezvous with its adjacent receiver, this node will start channel jumping from the channel just next to its dwelling channel in the ACS as illustrated in Fig. 5. Once it finishes transmitting, it will return to the channel before rendezvous (i.e., its original dwelling channel) to avoid dwelling on the same channel as its adjacent nodes.

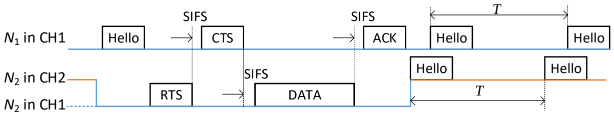


Fig. 6. The procedure of blind channel rendezvous access.

Figure 6 shows the procedure of a successful blind channel rendezvous access. As result of channel allocation, the dwelling channels of Node N_1 and Node N_2 are respectively CH1 and CH2, on which they send HELLO packets. When N_2 has packets to N_1, N_2 will switch its channel and finally meet N_1 at CH1. And then after the RTS/CTS handshake, they start transmissions. When the transmission is completed, N_2 will switch back to its original dwelling channel, sending HELLO packets periodically.

It should be noted that in order to accelerate the rendezvous, each node can create a neighbor nodes' dwelling channel list as follows. When a node obtains channel rendezvous with one of its adjacent node for the first time, it will record this adjacent node's dwelling channel. When the later packets to this adjacent node arrive, it will start to rendezvous from the recorded channel.

IV. THEORETICAL ANALYSIS

In this section we analyze the convergence property of the proposed scheme for sender-receiver pair channel allocation.

Theorem 1: **(Convergence of Algorithm 1)** Given $M \geq r + 1$, where r is the maximal number of interfering nodes to any node in the network, the channel allocation algorithm (Algorithm 1) will converge to interference-free channel allocation in time t with the probability Q_{N_t}. And $Q_{N_t} \to 1$, when $t \to \infty$.

Proof: Denote the total number of nodes in the network as N. At the beginning of a time slot t, n_0 nodes have achieved collision-free channel allocation and n_i interfering nodes are allocated on the same channel i ($i = 1, 2, , m$, where m is the number of channels that are suffering from channel collision). We denote this network state as $[(n_1, n_2 \cdots, n_i, \cdots n_m), n_0]$, and we have

$$\sum_{i=1}^{m} n_i + n_0 = N \tag{2}$$

And then the collided nodes will switch channel with their respective probability of p, and therefore the network state will transit, till it arrives to the state of $[(0), N]$ when the network converges to collision-free channel allocation. In Fig. 7, we plot the state transition flow for 4 interfering nodes in the network given $M \geq 4$, and initially all these 4 nodes are collided on the same channel, where p_{ts} represents the state transition probability. The property of p_{ts} will decide the convergence performance and therefore in the follows we will deduce the calculation of p_{ts}.

Figure 8 illustrates all the possible state transitions at Slot t. During Slot t, we denote $n_{i,t}$ $(0 \leq n_{i,t} \leq n_i - \sum_{k=1}^{t-1} n_{i,k})$ nodes

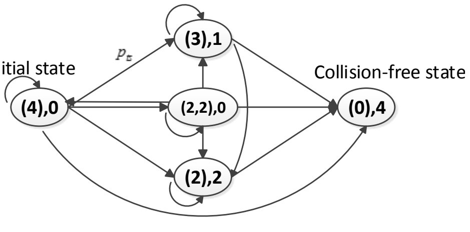


Fig. 7. Channel allocation state transition flow diagram.

out of n_i nodes switch channel and $n_i - \sum\limits_{k=1}^{t} n_{i,k}$ nodes still dwell on Channel i. So the total number of nodes that switch channel are $\sum\limits_{i=1}^{m_{t-1}} n_{i,t} = N_t$, where m_{t-1} is the number of collided channel at the beginning of Slot t (namely after Slot $t-1$), therefore we can calculate the probability that these N_t nodes choose to switch channel as

$$p_{tr} = \prod_{i=1}^{m_{t-1}} \mathrm{C}_{n_i - \sum\limits_{k=1}^{t-1} n_{i,k}}^{n_{i,t}} \cdot \left[\prod_{j=1}^{n_{i,t}} p_j \cdot \prod_{j=1}^{n_i - \sum\limits_{k=1}^{t} n_{i,k}} (1 - p_j) \right] \tag{3}$$

where p_j is the channel switch probability of node j, and $C_n^m = \frac{n!}{m!(n-m)!}$.

Through Slot t, some nodes could obtain collision-free channel while the others may still suffer from channel collision because they happen to switch to the same channel again. We denote $n_{0,t}$ $(0 \le n_{0,t} \le N_t\ )$ nodes finish the channel allocation through Slot t, and $x_{k,t}$ $(0 \le x_{k,t} \le N_t\ ,\ x_{k,t} \ne 1)$ nodes will jump to the same channel which will cause new channel collision. The total number of these new-caused collision channels is a_t. And we have $\sum\limits_{k=1}^{a_t} x_{k,t} + n_{0,t} = N_t$. Therefore, the state probability in Fig. 8 can be calculated as

$$\begin{aligned} & p_s \left[\left(x_{1,t}, x_{2,t}, \cdots, x_{a_t,t} \right), n_{0,t} \right] \\ & = \left[\left(\mathrm{A}_{N_t}^{\sum\limits_{k=1}^{a_t} x_{k,t}} \Big/ \prod_{k=1}^{a_t} \mathrm{A}_{x_{k,t}}^{x_{k,t}} \right) \cdot \mathrm{A}_F^{n_{0,t}+a_t} \right] \Big/ \left(F^{N_t} \cdot l! \right) \end{aligned} \tag{4}$$

where l is the number of equal $x_{k,t}$ $(k = 1, 2 \cdots\cdots m_t)$ (for instance, if $a_t = 3$ and $[x_{1,t}, x_{2,t}, x_{3,t}] = [2, 2, 3]$, $l = 2$). F is the number of current available free channels ($F \le M$). Note that $A_n^m = n \cdot (n-1) \cdots (n-m+1)$. For instance, assuming the total available channel is 8, in Fig. 8, the state probability that two pairs switch to the same channel $p_s \left[(2,2), 0 \right]$ can be calculated as $p_s \left[(2,2), 0 \right] = \frac{\mathrm{A}_4^4}{\mathrm{A}_2^2 \cdot \mathrm{A}_2^2} \cdot \mathrm{A}_8^{0+2} \Big/ 8^4 \cdot 2! = \frac{21}{512}$.

Note that if $n_i - \sum\limits_{k=1}^{t} n_{i,k} = 1$ only one node dwells on this channel i and this node also finishes the channel allocation. Therefore, this state transfers into $n_i - \sum\limits_{k=1}^{t} n_{i,k} = 0$ and $n_{0,t}$ is increased by 1. After Slot t, since $x_{k,t}$ nodes are allocated to the same channel k, these nodes will re-do the channel jumping in Slot $t+1$. In other words, after replacing $\left[\left(n_1 - \sum\limits_{k=1}^{t} n_{1,k}, \cdots n_i - \sum\limits_{k=1}^{t} n_{i,k}, \cdots, n_{m_t} \right), n_0 + \sum\limits_{k=1}^{t} n_{0,k} \right]$, and then the channel jumping process can be repeated iteratively. Therefore, the state transition probability is

$$p_{ts} = p_{tr} \cdot p_s \left[\left(x_{1,t}, x_{2,t}, \cdots, x_{a_t,t} \right), n_{0,t} \right] \tag{5}$$

And we have:

$$\begin{aligned} & p_s \left[n_1 - \sum_{k=1}^{t} n_{1,k}, \cdots n_i - \sum_{k=1}^{t} n_{i,k}, \cdots, n_0 + \sum_{k=1}^{t} n_{0,k} \right] = p_{ts} \\ & \cdot p_s \left[n_1 - \sum_{k=1}^{t-1} n_{1,k}, \cdots n_i - \sum_{k=1}^{t-1} n_{i,k}, \cdots, n_0 + \sum_{k=1}^{t-1} n_{0,k} \right] \end{aligned} \tag{6}$$

When $n_0 + \sum\limits_{k=1}^{t} n_{0,k} = N$, the channel allocation completes.

From the state transition flow in Fig. 7, since $M \ge r+1$, all the state transition will finally convert to the finally state $[(0)\,,\ N]$, which means the overall allocation success probability, denoted by $Q_{N_t}(t)$, approaches 1 with the increase of time slot (i.e., $Q_{N_t} \to 1$, when $t \to \infty$). And $Q_{N_t}(t)$ can be calculated as follows

$$Q_{N_t}(t) = \sum p_s \left[(0)\,,\ n_0 + \sum_{k=1}^{t} n_{0,k} = N \right] \tag{7}$$

■

For any node i, assuming the active interfering nodes around it is r. In the worst case, the first r channels in Node i's ACS are occupied by these r nodes, and Node i will switch its channel every time that it receives HELLO packets from another node. Therefore, Node i need to switch $r+1$ times to find a free channel. Since the ACS of each node is

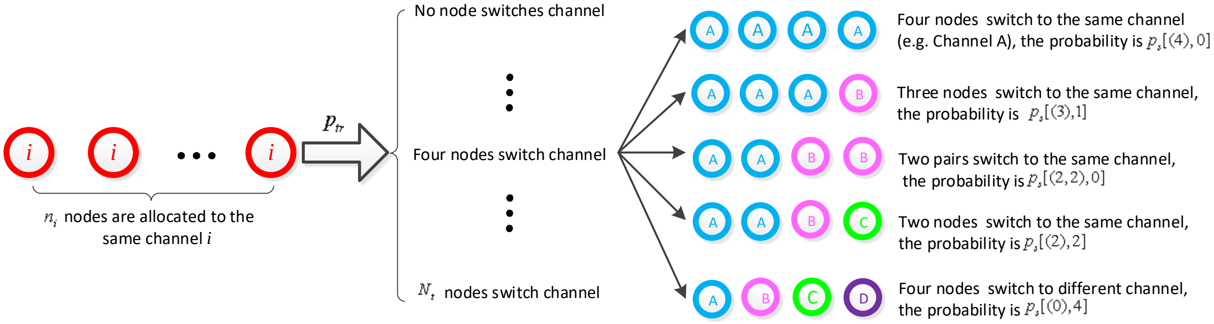
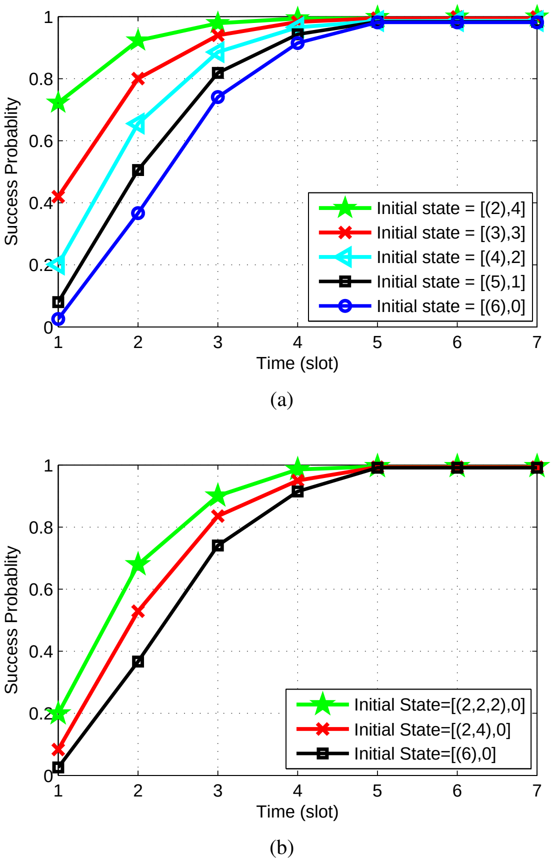


Fig. 8. The possible state transitions at Slot t

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TVT.2018.2865817, IEEE Transactions on Vehicular Technology



(a)

(b)

Fig. 9. Convergence property of the proposed channel allocation: (a) All nodes are initially collided on the same channel; (b) Nodes are initially collided on different channels .

randomly sequenced and a node will listen for a period of T on each switched channel before dwelling on the collision-free channel, the average time that one node can dwell on a free-channel is less than $T(r+1)/2$ considering that Node i may not switch it channel every time when interference occurs (its interfering node will also probably switch channel).

For better understanding, Fig. 9 shows the numerical results of success probability $Q_{N_t}(t)$ varying with time slot (t) (assuming that there are 8 available channels). In Fig. 9a, we set the initial collided node number from 2 to 6, and consider the worst case that all the collided nodes are on the same channel, i.e., the initial state is $\{(x), n_0\}$, where $x = 2, 3, 4, 5, 6$ and $n_0 = 6 - x$. It shows that more nodes initially collided at the same channel will cost more slots to converge. In Fig. 9b, we consider that the nodes are initially collided on different channels: i) 6 nodes are allocated the same channel corresponding to the initial state $[(6), 0]$; ii) 2 nodes are allocated the same channel, meanwhile 4 nodes are allocated another same channel corresponding to the initial state $[(2, 4), 0]$; and iii) three pairs are allocated three different channel corresponding to the initial state $[(2, 2, 2), 0]$.

Although in all three occasions, the collided nodes are all 6, their convergence rate is different. Specifically, more nodes are collided on the same channel the network converge more slowly. On the other hand, if the collided nodes are more evenly on different channel, the network will converge faster. Both Fig. 9a and Fig. 9b demonstrate that the algorithm converges fast. When $t > 6$ the channel allocation success probability is higher than 99% for all the considered scenarios.

Theorem 2: **(Convergence of Algorithm 2)** Given M available channels, the maximal time before one sender rendezvous with its target receiver is $(M-1)T$, and the average time is $(M-1)T/2$.

Proof: Since the receiver is within the transmission range of the sender, it cannot dwell on the same channel as the sender. Therefore, in the worst-case scenario, the sender will switch $(M-1)$ times to rendezvous. As the channel list is randomly set up, the sender's dwelling channel can be any one of these remaining $(M-1)$ channels with the same probability. Therefore, the average switching times are $(M-1)/2$, which consume the time of $(M-1)T/2$. ■

By looking Theorem 1 and Theorem 2, one can observe that the proposed algorithm is convergent.

V. PERFORMANCE EVALUATION

A. *Simulation setup*

We evaluate the proposed algorithm using NS3.25 simulator. Table I lists the MAC layer parameters used in the simulation. Figure 10 shows the simulated network topology. We consider a grid network topology (as opposed to random network topology) for controlling the interference among the nodes, especially when new nodes join the network. Awareness of the interference allows us to evaluate the system performance more effectively. We assume that initially all 16 channels are available to any node (i.e., $M = 16$) and the interference range is 300m (The dotted lines represent that the two nodes are within the interference range).

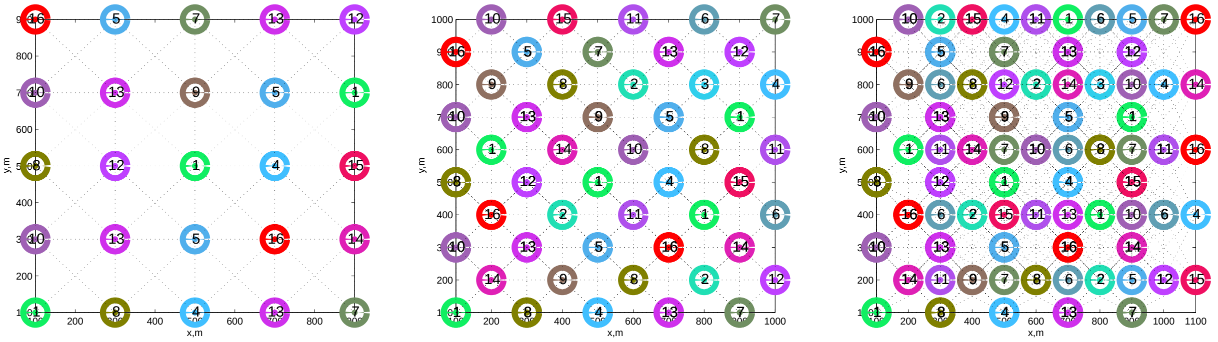
Originally, there are 25 nodes in the network. And we then add 25 more nodes respectively at time slot 5 and 20, to guarantee that the new entrants will interfere with original nodes, the newly joined nodes are evenly distributed in the network as shown in Fig. 10.

TABLE I
SIMULATION PARAMETERS AND VALUES.

Description	Value
Channel Bit Rate	2Mbps
SIFS	28μs
ACK Timeout	200μs
Average Arrival Time	110 slots
Slot Time	50μs
CTS Timeout	200μs

B. *Performance of the channel allocation*

As a consequence of the receiver-oriented channel allocation, Fig. 10 also reports the channel allocation results for Algorithm 1 when all nodes are just deployed and have no data to send (i.e., the number in each node represents the channel that this node finally dwells on). Comparing Fig 10b to 10a and Fig 10c to 10b, one can find that the newly joined nodes will not impact the channel that already allocated to the elder nodes and Algorithm 1 can achieve collision-free channel allocation.



(a) (b) (c)

Fig. 10. Simulation topology: (a) Original 25 nodes; (b) 25 new nodes join the network at time 5; (c) Another 25 nodes join the network at time 20.

In Fig.11, we plot the convergence property of the proposed algorithm in the aforementioned dynamic network scenario. We observe that by using only 4 slots, the network arrives at collision-free state. And at the 5th slot, 25 new nodes entering the network results in 45 node collisions, but the network soon achieve collision-free state again with only 6 slots. At the 20th slot, another 25 new nodes enter the network results in 43 collisions, and this time the network achieve collision-free state again in just 3 slots. It is interesting to note that same number of new entrants will cause similar amount of channel collisions (i.e., 45 and 43 respectively), however, the convergence time consumed at the second time is much less than that consumed at the first time (i.e., 3 is only half of 6). And the later the nodes join in the network, they will have less effect to the original nodes. This is because of the learning ability of the nodes that makes new nodes switch channel with high probability, while the old ones tend to stay their original dwelling channel, this accelerates the convergence speed.

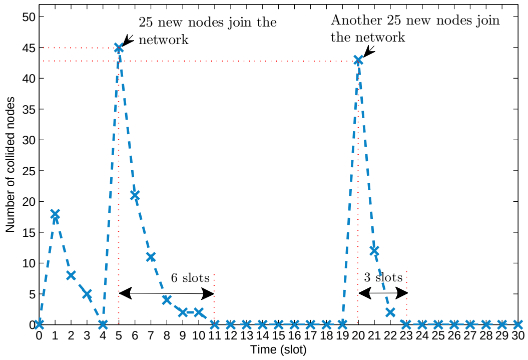


Fig. 11. The number of collided nodes with new nodes joining the network.

In Fig.12, we report the number of node that switch channels varying with time, where we differentiate the new entrants to the original nodes. It clearly shows that the original nodes are less likely to switch the channel with time goes by. This corresponds with the basic idea in Algorithm 1 that when nodes stay on a channel for a longer time it will be less willing to switch channel.

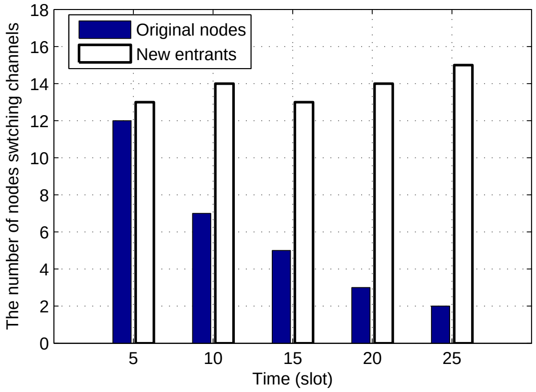


Fig. 12. The number of collided nodes that switch channels.

To validate the robustness of the channel allocation algorithm to malicious nodes, we consider another simulation experiment that let 25 malicious nodes join in the network at the 15th time slot. In Fig.13, we report the number of nodes that switch channels as well as that suffer from collisions varying with time. It shows that during the 15th slot and 20th slot, although nodes are affected by the malicious nodes, only small number of nodes switch the channel as the channel switch probability p is small, resulting to a large number of collided nodes. However, when the malicious nodes are finally detected (the times that constantly receives HELLO packet on the dwelling channel reaches the threshold of 5), p is reset to 0.5, resulting to more nodes choose to switch the channel and thus the sharp decrease on the number of collided nodes. At the 25th time slot, no node is suffering from collisions any more. Using this strategy, the algorithm can effectively avoid the effect of malicious nodes.

Although in the simulation we do not consider the mobility of nodes, the proposed algorithm can also work well with mobile nodes in the network. When a mobile node arrives at a new position, it will keep its allocated channel if it experiences no collision on this channel. On the other hand, if it suffers from collision on its allocated channel, it will switch channel and try to find another free channel, acting as a new entrant.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TVT.2018.2865817, IEEE Transactions on Vehicular Technology

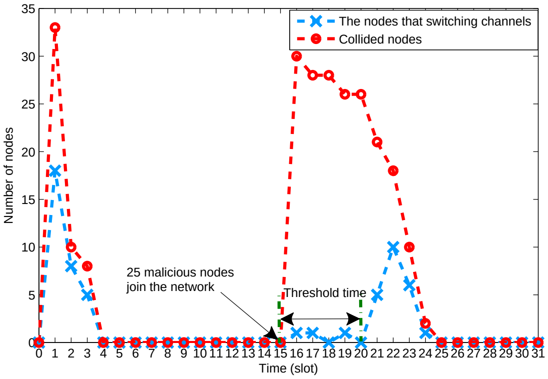


Fig. 13. The situation when malicious nodes join the network.

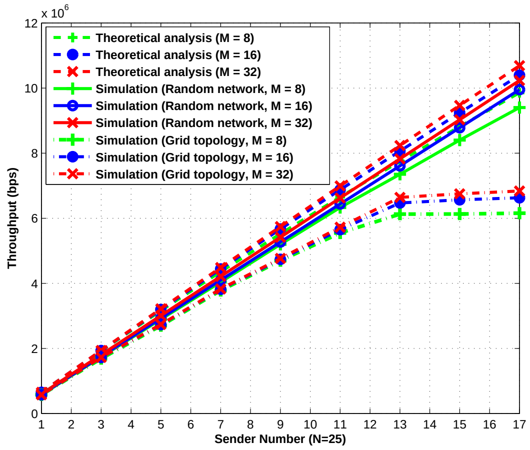


Fig. 14. Saturation throughput: analysis versus simulation.

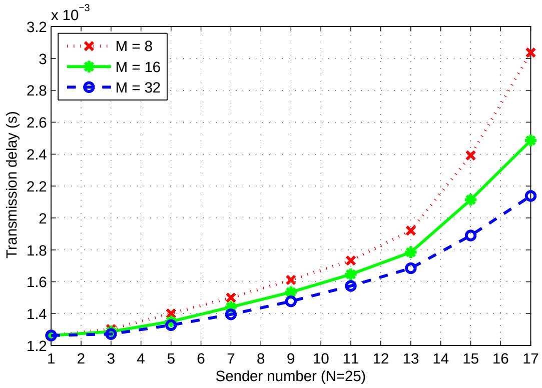


Fig. 15. Transmission delay versus the number of senders.

C. Saturation throughput and transmission delay

In this subsection, we evaluate the throughput performance of the proposed rendezvous access algorithm (Algorithm 2) and compare it to the theoretical analysis results.

In Fig.14, we plot throughput against the number of senders in the network with various channels (M =8, 16 and 32). Both simulation and analysis results are presented for comparison. In the simulation model, we consider two network scenarios: random topology and grid topology. In the random topology 100 nodes are randomly deployed in the network, while in the grid topology 25 nodes are deployed as in Fig. 10a. In both network scenarios, the senders are randomly chosen and each sender's dedicated receiver is also randomly chosen from its adjacent nodes (two senders may compete for transmission to the same receiver).

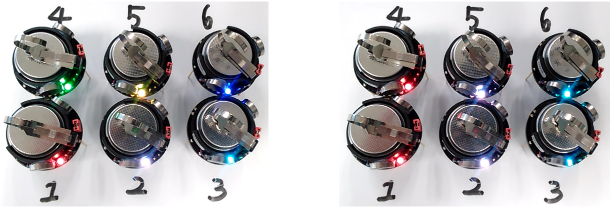
We observe that the simulation results for a 100-node random network are closely match with the theoretical analysis. This is because the higher density of the nodes with random access is closer to the saturation condition. The minor difference between the analysis and simulation results is due to the fact that theoretical results are based on the assumption that each sender always has a receiver. However, in the simulated network, since the sender and the receiver are dynamically chosen, one sender would be another sender's receiver and it could also happen that a sender would not find a free receiver with the increasing number of senders. This phenomena is more evident at $N = 25$ nodes with grid topology. The difference between theoretical and simulation results for grid topology increases with senders. For instance, at $N = 25$ nodes, the throughput is saturated when the senders are more than 12 since there are at most 12 sender-receiver pairs transmitting simultaneously on different channels.

We then consider the transmission delay, which is defined as the time from a sender having a request to start transmission to successful packet transmission. And if there is a collision, the transmission delay also includes retransmission time. Specifically, the transmission delay includes four parts: i) channel hopping time for blind rendezvous; ii) time for channel competition if multiple senders try to transmit on the same channel; iii) packet transmission time; and iv) the time for retransmission if there is collision. In the following simulations, we also adopt the original grid topology of 25 nodes as illustrated in Fig. 10a.

Figure 15 shows the transmission delays with respect to the number of channels. We observe that the lower transmission delay achieved for the larger number of channels. This is because less transmissions are wasted in collisions, especially for large number of sending nodes in the network. The main conclusion is that having more channels not only achieve better throughput but also decrease transmission delays.

D. Testbed illustration

We have used 6 Kilobots as a testbed to demonstrate the effectiveness of our proposed method. The Kilobot is a 3.3 cm tall low-cost robot invented by the Self-organizing Systems Research Group at Harvard University [42]. Each Kilobot is equipped with an infrared transmitter and receiver so that the robots can communicate with each other. In our demonstration, all the 6 Kilobots were placed within communication range. Through the channel allocation, each Kilobot dwells on a different channel, represented by the LED color completion (Fig.16a). When Kilobots 4,5,6 have data to send (their receivers are respectively Kilobots 1, 2, 3), they will successfully rendezvous with the receiver on the allocated channel (Fig.16b). The demonstration shows that our proposed



(a) (b)

Fig. 16. Kilobots demo illustration: (a) Channel allocation complete; (b) Rendezvous complete.

method can work well. The video illustrations can be found at [43].

VI. CONCLUSION

A receiver-oriented channel allocation scheme with a sender-jump blind rendezvous is proposed to solve the sender-receiver pair channel allocation problem in D2D multi-channel communications. Our proposed algorithms can allocate channels to interfering sender-receiver pairs more effectively as a result of learning processes that nodes adopt channel switching strategy based on past history. With the proposed algorithms, the network can perform concurrent transmissions along multiple channels without interference. The work reported can help network planners in deploying multi-channel high-density D2D networks and to contribute in the development of high-density Internet of Things.

REFERENCES

[1] A. Asadi, Q. Wang and V. Mancuso, ”A Survey on Device-to-Device Communication in Cellular Networks”, IEEE Communications Surveys and Tutorials, vol. 16, no. 4, pp. 1801-1819, 2014.

[2] Y. Chen, S. He, F. Hou, Z. Shi, and J. Chen, ”Promoting Device-to-Device Communication in Cellular Networks by Contract-based Incentive Mechanisms”, IEEE Network, vol. 31, no. 3, pp. 14-20, 2017.

[3] Y. Chen, S. He, F. Hou, Z. Shi, and X. Chen, ”Optimal User-Centric Relay Assisted Device-to-Device Communications: An Auction Approach”, IET Communications, vol. 9, no. 3, pp. 386-395, 2015.

[4] H. Zhao, E. Garcia, J. Wei, S. Wang and D. Ma, ”Evaluating the Impact of Network Density, Hidden Nodes and Capture Effect for Throughput Guarantee in Multi-Hop Wireless Networks”, Ad Hoc Networks, vol. 11, no. 1, pp. 54-69, 2013.

[5] H. Zhao, S. Wang, Y. Xi and J. Wei, ”Modeling Intra-flow Contention Problem in Wireless Multi-hop Networks”, IEEE Communications Letters, vol.14, no. 1, pp. 18-20, 2010

[6] H. Zhao, J. Wei, N. I Sarkar and S. Huang, ”E-MAC: An Evolutionary Solution for Collision Avoidance in Wireless Ad Hoc Networks”, Elsevier Journal of Network and Computer Applications(JNCA), vol. 65, pp. 1-11, 2016.

[7] S. Zhang, A. S. Hafid, H. Zhao, S. Wang, ”Cross-Layer Rethink on Sensing-Throughput Tradeoff for Multi-Channel Cognitive Radio Networks”, IEEE Trans. on Wireless Communications, vol. 15, no. 10, pp. 6883-6897, 2016.

[8] H. Zhao, N. I. Sarkar, J. Ben-Othman, O. Holland, A. Maaref and W. Tan, ”Multi-Channel Cognitive Radio Ad Hoc Networks”, IEEE Communications Magazine, vol. 56, no. 4, pp. 134-135, APRIL 2018.

[9] D. Zhang, Z. Chen, M. K. Awad, N. Zhang, H. Zhou and X. S. Shen, ”Utility-Optimal Resource Management and Allocation Algorithm for Energy Harvesting Cognitive Radio Sensor Networks”, IEEE Journal on Selected Areas in Communications, vol. 34, no. 12, pp. 3552-3565, Dec. 2016.

[10] D. Zhang et al., ”Energy-Harvesting-Aided Spectrum Sensing and Data Transmission in Heterogeneous Cognitive Radio Sensor Network”, IEEE Trans. on Vehicular Technology, vol. 66, no. 1, pp. 831-843, Jan. 2017.

[11] S. Zhang, A. S. Hafid, H. Zhao and S. Wang, ”Impact of Heterogeneous Fading Channels in Power Limited Cognitive Radio Networks”, IEEE Trans. On Cognitive Communications and Networking, vol. 4, no. 1, pp. 1-14, March 2018.

[12] S. Jang and S. Bahk, ”A Channel Allocation Algorithm for Reducing the Channel Sensing/Reserving Asymmetry in 802.11ac Networks”, IEEE Trans. on Mobile Computing, vol. 14, no. 3, pp. 458-472, 2015.

[13] L. Liang and G. Feng, ”A Game-Theoretic Framework for Interference Coordination in OFDMA Relay Networks”, IEEE Trans. on Vehicular Technology, vol. 61, no. 1, pp. 321-332, 2012.

[14] L. Qin and D. Zhao, ”Channel Time Allocations and Handoff Management for Fair Throughput in Wireless Mesh Networks”, IEEE Trans. on Vehicular Technology, vol. 64, no. 1, pp. 315-326, 2015.

[15] B. Di, L. Song and Y. Li, ”Sub-Channel Assignment, Power Allocation, and User Scheduling for Non-Orthogonal Multiple Access Networks”, IEEE Trans. on Wireless Communications, vol. 15, no. 11, pp. 7686-7698, 2016.

[16] H. Zhao, E. Garcia, J. Wei and Y. Xi, ”Accurate available bandwidth estimation in IEEE 802.11-based ad hoc networks”, Computer Communications, vol.32, no. 6, pp.1050-57, 2009.

[17] J. Li, H. Zhao, J. Wei, D. Ma and L. Zhou, ”Sender-Jump Receiver-Wait: A Simple Blind Rendezvous Algorithm for Distributed Cognitive Radio Networks”, IEEE Trans. on Mobile Computing, vol. 17, no. 1, pp. 183-196, 2018.

[18] L. Ding, T. Melodia, S. N. Batalama and J. D. Matyjas, ”Distributed Routing, Relay Selection, and Spectrum Allocation in Cognitive and Cooperative Ad Hoc Networks”, in Proc. of IEEE SECON, Boston, MA, 2010, pp. 1-9.

[19] T. H. Lim, W. S. Jeon and D. G. Jeong, ”Centralized channel allocation scheme in densely deployed 802.11 wireless LANs”, in 18th International Conference on Advanced Communication Technology (ICACT), Pyeongchang, 2016, pp. 249-253.

[20] Y. Xue, B. Li and K. Nahrstedt, ”Optimal resource allocation in wireless ad hoc networks: a price-based approach”, IEEE Trans. on Mobile Computing, vol. 5, no. 4, pp. 347-364, 2006.

[21] D. Yang, L. Liu, ”Fragile Bin-packing Problem with Start-Up Weight for Channel Assignment in Cellular Networks”, Journal of Convergence Information Technology, vol. 8, no. 9, pp. 751-759, 2013.

[22] S. Maghsudi and S. Staczak, ”Hybrid Centralized-Distributed Resource Allocation for Device-to-Device Communication Underlaying Cellular Networks”, IEEE Trans. on Vehicular Technology, vol. 65, no. 4, pp. 2481-2495, 2016.

[23] Z. Zhao, Z. Peng, S. Zheng and J. Shang, ”Cognitive radio spectrum allocation using evolutionary algorithms”, IEEE Trans. on Wireless Communications, vol. 8, no. 9, pp. 4421-4425, 2009.

[24] R. Yu, W. Zhong, S. Xie, Y. Zhang and Y. Zhang, ”QoS Differential Scheduling in Cognitive-Radio-Based Smart Grid Networks: An Adaptive Dynamic Programming Approach”, IEEE Trans. on Neural Networks and Learning Systems, vol. 27, no. 2, pp. 435-443, 2016.

[25] F. Kaabi, S. Ghannay, F. Filali, ”Channel Allocation and Routing in Wireless Mesh Networks: A survey and qualitative comparison between schemes”, International Journal of Wireless and Mobile Networks, 2010.

[26] N. Haouari, S. Moussaoui, S. M. Senouci, A. Boualouache and M. A. Messous, ”An efficient management of the control channel bandwidth in VANETs”, in Proc. of IEEE International Conference on Communications (ICC), Paris, 2017, pp. 1-6.

[27] S. Pediaditaki and M. K. Marina, ”A Learning-Based Channel Allocation Protocol for Multi-Radio Wireless Mesh Networks”, in Proc. of IEEE INFOCOM Workshops 2009, Rio de Janeiro, 2009, pp. 1-2.

[28] M. Alicherry, R. Bhatia and L. E. Li, ”Joint Channel Assignment and Routing for Throughput Optimization in Multiradio Wireless Mesh Networks”, IEEE Journal on Selected Areas in Communications, vol. 24, no. 11, pp. 1960-1971, 2006.

[29] D. Wu and P. Mohapatra, ”From Theory to Practice: Evaluating Static Channel Assignments on a Wireless Mesh Network”, in Proc. of IEEE INFOCOM, San Diego, CA, 2010, pp. 1-5.

[30] W. Cheng, X. Cheng, T. Znati, X. Lu and Z. Lu, ”The Complexity of Channel Scheduling in Multi-Radio Multi-Channel Wireless Networks”, in Proc. of IEEE INFOCOM, Rio de Janeiro, 2009, pp. 1512-1520.

[31] Vedantham R, et al., ”Component Based Channel Assignment in Single Radio, Multi-channel Ad hoc Networks”, in Proc. of ACM Mobicom, 2006.

[32] L. Gao and X. Wang, ”A Game Approach for Multi-channel Allocation in Multi-hop Wireless Networks”, in Proc. of ACM Mobihoc, 2008.

[33] A. Saifullah, et al., ”Distributed Channel Allocation Protocols for Wireless Sensor Networks”, IEEE Trans. on Parallel and Distributed Systems, vol. 25, no. 9, pp. 2264-2274, 2014.

[34] S. Zhang, A. S. Hafid, H. Zhao and S. Wang, ”Cross-Layer Rethink on Sensing-Throughput Tradeoff for Multi-Channel Cognitive Radio Networks”, IEEE Trans. on Wireless Communication, vol. 15, no. 10, 2016

[35] K. A. Hafeez, L. Zhao, J. W. Mark, X. Shen and Z. Niu, ”Distributed Multichannel and Mobility-Aware Cluster-Based MAC Protocol for Vehicular Ad Hoc Networks”, IEEE Trans. on Vehicular Technology, vol. 62, no. 8, pp. 3886-3902, 2013.

[36] J. Lee, J. Mo, T. M. Trung, J. Walrand and H. S. W. So, ”Design and Analysis of a Cooperative Multichannel MAC Protocol for Heterogeneous Networks”, IEEE Trans. on Vehicular Technology, vol. 59, no. 7, pp. 3536-3548, 2010.

[37] C. M. Chao, H. Y. Fu and L. R. Zhang, ”A Fast Rendezvous-Guarantee Channel Hopping Protocol for Cognitive Radio Networks”, IEEE Trans. on Vehicular Technology, vol. 64, no. 12, pp. 5804-5816, Dec. 2015.

[38] V. A. Reguera, E. O. Guerra, R. D. Souza, E. M. G. Fernandez and G. Brante, ”Short Channel Hopping Sequence Approach to Rendezvous for Cognitive Networks”, IEEE Communications Letters, vol. 18, no. 2, pp. 289-292, February 2014.

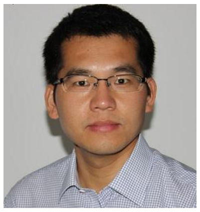
[39] H. Liu, Z. Lin, X. Chu, and Y.-W. Leung, ”Jump-Stay Rendezvous Algorithm for Cognitive Radio Networks”, IEEE Trans. on Parallel and Distributed Systems, vol. 23, no. 10, pp. 1867-1881, 2012.

[40] J. H. Chu, K. T. Feng and J. S. Lin, ”Prioritized Optimal Channel Allocation Schemes for Multi-Channel Vehicular Networks”, IEEE Trans. on Mobile Computing, vol. 14, no. 7, pp. 1463-1474, 2015.

[41] I. H. Chuang, H. Y. Wu and Y. H. Kuo, ”A Fast Blind Rendezvous Method by Alternate Hop-and-Wait Channel Hopping in Cognitive Radio Networks”, IEEE Trans. on Mobile Computing, vol. 13, no. 10, pp. 2171-2184, 2014.

[42] M. Rubenstein, C. Ahler and R. Nagpal, ”Kilobot: A low cost scalable robot system for collective behaviors”, in IEEE International Conference on Robotics and Automation, Saint Paul, MN, 2012, pp. 3293-3298.

[43] Demonstation video. https://www.dropbox.com/s/qw2bhnivag11bfu/DemoIllustration1.mp4?dl=0

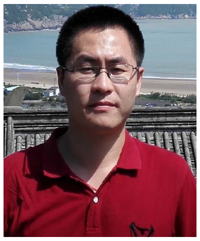


Haitao Zhao received his M.S. and PhD degree in information and communication engineering both from National University of Defense Technology (NUDT), Changsha, China respectively in 2004 and 2009. He is currently an associate professor in College of Electronic Science, NUDT. Prior to this, he visited Queens University of Belfast, UK from April 2008 to July 2009 and Hong Kong Baptist University (HKBU) from August 2014 to February 2015. His main research interests include cognitive radio networks and self-organized networks. He is now a senior member of IEEE, a member of World-wide University Network (WUN) Cognitive Communications Consortium, and also a mentor member for IEEE 1900.1 standard. He has served as TPC member for IEEE ICC' 14-18, Globecom' 15-18 and Guest Editor for several international journals, includes IEEE Communications Magazine and Physical Communications.

Kaiqi Ding received her bachelor degree in communication engineering from Harbin Institute of Technology (HIT) in 2015 and her M.S. degree from National University of Defense Technology (NUDT) in 2017. Her main research interests are cognitive radio networks and channel allocation for multi-channel wireless networks.

Nurul I Sarkar holds a PhD from the University of Auckland and is currently associate professor and leader of the Network and Security Research Group at the Auckland University of Technology, Auckland, New Zealand. He is a member of many professional organizations and societies. Dr Sarkar is a regularly invited keynote speaker, chair, and committee member for various national and international fora. He has published more than 165 refereed articles and served on the editorial review boards of several prestigious journals. Improving the Performance of Wireless LANs: A Practical Guide, his second book has been published by Taylor and Francis in January 2014. Dr. Sarkar is a senior member of IEEE.

Jibo Wei received his B.S. and M.S. degrees from the National University of Defense Technology (NUDT), Changsha, China, in 1989 and 1992, respectively, and the Ph.D. degree from Southeast University, Nanjing, China, in 1998, all in electronic engineering. He is currently a Professor with the Department of Cognitive Communications, NUDT. His research interests include Software Defined Radios (SDR), Cognitive communications and networking, OFDM and MIMO, cooperative communications. Dr. Wei is a member of the IEEE Communication Society and also a member of the IEEE Vehicular Technology Society. He is a council member of the China Institute of Communications. He is also an editorial committee member of the Journal on Communications.



Jun Xiong received his B.S. and Ph.D. degrees from National University of Defense Technology (NUDT), China, in 2009 and 2014, respectively. He is currently a lecturer in College of Electronic Science, NUDT. His research interests include cooperative communications, physical layer security, and resource allocation, where he has published more than 30 refereed articles.