

# Neighbor Discovery in Multi-Receiver Cognitive Radio Networks\*

Srinivasan Krishnamurthy<sup>†</sup>    Neeraj Mittal    R. Chandrasekaran  
S. Venkatesan

School of Engineering and Computer Science

The University of Texas at Dallas, Richardson, TX 75080, USA

kvasan@gmail.com    {neerajm, chandra, venky}@utdallas.edu

## Abstract

A cognitive radio (CR) node is a radio device capable of operating (transmitting and receiving) over multiple channels. As a result, a network consisting of one or more cognitive radio nodes can *adapt* to varying channel availability in its geographical region by dynamically changing the channel (or channels) that nodes use for communication. We investigate the problem of *neighbor discovery* in a network consisting of one or more cognitive radio nodes when nodes have multiple receivers but only a single transmitter. Neighbor discovery, in turn, can be used to solve other important communication problems such as broadcasting and gossiping in an efficient manner. We present a TDMA-based deterministic distributed algorithm for neighbor discovery whose time-complexity is  $M \lceil \frac{N}{r} \rceil + O(\max(M, N) \log r)$ , where  $M$  is the maximum number of channels on which a node can operate,  $N$  denotes the size of the space used to assigned identifiers to nodes,  $r$  is the number of receivers at a node (with  $1 \leq r \leq \min(M, N)$ ).

**Key Words:** cognitive radio, multiple channels, multiple receivers, neighbor discovery, distributed algorithm

---

\*A preliminary version of this paper appeared in the 2006 proceedings of the Symposium on Distributed Computing (DISC) as a brief announcement [13].

<sup>†</sup>The author is currently working at Cisco Systems, Inc.

# 1 Introduction

Cognitive Radio (CR) technology [15] offers a mechanism for flexible and improved utilization of the radio spectrum. The motivation behind CR technology is to enable a group of wireless nodes to utilize unused channels to maximize the communication throughput of their applications, without causing any perceptible drop in the quality of service offered by these channels to their owners. The owner of a (licensed) channel is referred to as *primary user* and all other users of the channel as *secondary users* [1]. A CR network is a collection of CR-enabled secondary users (referred as nodes in rest of this paper) that can act as transmitters or receivers. Each node periodically scans and identifies available channels in the portion of the frequency spectrum that it is built to operate on. A channel is said to be available if a secondary user can transmit and receive messages on the channel for a reasonable amount of time without interference to/from the primary user(s). Since the nodes are likely to be scattered geographically and affected by the presence of (possibly distinct) primary users and/or other sources of interference, the channel availability across nodes is expected to vary. Further, since the set of available channels could change over time [5] and a dedicated central authority that coordinates communication among nodes may not exist at all times, setting up an effective communication mechanism is not straightforward. Nodes in the CR network have to first resolve the following layer-2 configuration problems before network-wide communication can be established:

1. Who are the neighbors of each node?
2. What channel(s) can a pair of neighboring nodes use to communicate with each other?

This gives rise to the *neighbor discovery problem* also referred to as the *neighbor gossip problem* in the literature. We say that two nodes are *neighbors* if they are within wireless transmission range of each other and have at least one common channel in their availability sets. Neighbor discovery is an important problem in radio networks because it can be used as a primitive for solving other important communication problems, such as *broadcasting* a message [2, 3, 11, 10, 4, 7, 12], finding a *globally common channel* [14] or computing a *deterministic transmission schedule* [23], in a more efficient manner. For instance, if each node in the network knows its neighbor, then broadcasting and gossiping problems can be solved in almost linear time using almost linear time leader election and linear time DFS traversal [6]. Clearly, it is desirable to first perform neighbor discovery if several messages have to be broadcast one after another and the neighborhood itself is not expected to

change significantly during this duration.

Most of the current research on wireless networks assumes that the entire network operates on a *single* channel. While some MAC-layer solutions have been proposed for multi-channel networks [8, 9, 17, 18, 19, 20, 21, 22, 24], these solutions usually assume either the existence of a common control channel [21, 22, 24] and/or that every node is equipped with a separate radio interface for each channel [8, 17, 18]. Solutions for multi-channel wireless mesh networks [9, 19, 20] assume that all channels are available throughout the network. To the best of our knowledge, Krishnamurthy *et al.* [14] were the first to propose an algorithm to solve the neighbor discovery problem in a cognitive radio network without a control channel when nodes have a *single* transceiver. They also present an algorithm to compute a *globally common channel*, if one exists, using the algorithm for neighbor discovery [14].

The neighbor discovery problem is non-trivial to solve in a cognitive radio network because nodes can operate over multiple channels and different nodes may have different channel availability sets. This introduces an additional level of complexity. For two neighboring nodes to communicate successfully, both of them should be tuned to the *same channel* at the *same time* in addition to the requirement that sending node's transmission should not collide with another node's transmission which is also within the communication range of the receiving node. Since different nodes may have different sets of channels available at their respective locations, nodes may have to perform neighbor discovery on more than one channel in their availability sets. Therefore it is important for all nodes to be able to detect the completion of neighbor discovery on one channel *simultaneously* and switch to the next channel in a *coordinated* manner.

In this paper, we present a TDMA-based deterministic collision-free neighbor discovery algorithm for a cognitive radio network when nodes have multiple receivers but only a single transmitter. Under this model, during a time-slot, a node can (i) transmit on one channel, (ii) receive on one or more channels (up to the number of receivers), or (iii) be turned off. Simultaneous transmission and reception (even on different channels) at a node is not allowed because of complications in effectively isolating the receivers from the transmitter in hardware. Our algorithm uses  $M \lceil \frac{N}{r} \rceil + O(\max(M, N) \log r)$  time-slots where  $M$  is the maximum number of channels on which a node can operate,  $N$  is the size of the space used to assign identifiers to nodes, and  $r$  is the number of receivers at a node (with  $1 \leq r \leq \min(M, N)$ ). We show elsewhere [16] that any deterministic algorithm for neighbor discovery has to use  $\Omega(MN)$  time-slots, in the worst-case, even when the network contains a single node and collisions are allowed (that is, two or more neighbors of the

same node can transmit on the same channel at the same time). When the network contains a single node, the worst-case time complexity of neighbor discovery is  $\Omega(MN)$  in case it is desirable that nodes be able to detect *termination* of the neighbor discovery algorithm. In this case, if the node does not receive any message during a time-slot, it does not know whether it is because of collision or simply because it does not have any neighbor. The same proof can be easily extended to show a lower bound of  $\Omega(\frac{MN}{r})$  time-slots when each node has  $r$  receivers.

The rest of the paper is organized as follows. Section 2 describes the system model. The neighbor discovery algorithm is presented in Section 3. Finally, we conclude the paper and outline directions for future research in Section 4.

## 2 System Model

A cognitive radio network is modeled as a connected undirected graph on a set of nodes. Each node is assigned a unique identifier from the range  $[1..N]$ . Clearly,  $N$  also acts as an upper bound on the total number of nodes in the network. Hereafter in this paper, we use the phrase “node  $i$ ” to mean “node with identifier  $i$ ”. Let  $\mathcal{A}_{univ} = \{c_1, c_2, \dots, c_M\}$  represent the universal set of available channels that can be potentially used by all nodes for communication. All nodes know  $N$ ,  $M$  and  $\mathcal{A}_{univ}$ . Each node is equipped with  $r$  receivers, where  $1 \leq r \leq \min(M, N)$ , but only one transmitter all of which are capable of operating on any of the  $M$  channels in  $\mathcal{A}_{univ}$ . A node can receive on multiple channels at the same time (using different receivers) but cannot transmit and receive (even on different channels) at the same time. (To enable a node to transmit and receive at the same, the transmitter and the receiver have to be placed sufficiently apart from each other; otherwise the transmission may fry the receiver.) Each node  $i$ , if present, is aware of its channel availability set  $\mathcal{A}_i$ , which may be different for different nodes.

Nodes  $i$  and  $j$  are said to be *neighbors* (represented by an undirected edge or a pair of directed edges in the graph) if  $i$  and  $j$  are within each other’s radio range and  $\mathcal{A}_i \cap \mathcal{A}_j \neq \emptyset$ . Communication between nodes that are not neighbors is achieved by *multi-hop* transmissions.

We assume that time is divided into equal slices called *time-slots*. A message transmitted by a node in time-slot  $t$  is *delivered* to all its neighbors in the same time-slot. A receiving node successfully *receives* a message if and only if (1) one of its receivers is tuned to the channel on which the message is transmitted, and (2) none of its other neighbor transmits on the same channel during that time-slot. Note the distinction between a message being *delivered* at a recipient node

and being *received* by that node. If two or more neighbors of a node  $i$  transmit on the same channel in a given time-slot, a *collision* occurs at  $i$  and  $i$  does not receive any of the transmitted messages. The communication medium is assumed to be loss-free. We also assume that all nodes in the network start executing the neighbor discovery algorithm at the same time.

### 3 The Neighbor Discovery Algorithm

The main idea behind the algorithm is to ensure that every possible pair of nodes that may be present in the network is *able* to communicate on every possible channel. To that end, our algorithm guarantees that, for every triplet  $\langle i, j, c \rangle$ , where  $i$  and  $j$  refer to nodes and  $c$  refers to a channel, there is a time-slot  $t$  such that (i) only  $i$  is scheduled to transmit on  $c$  during  $t$  and (ii) at least one of the receivers at  $j$  is scheduled to be tuned to  $c$  (to receive) during  $t$ . Note that, even if a node is scheduled to transmit on a channel during a time-slot, the actual transmission occurs only if the node is present in the network and the channel is in the availability set of the node. Likewise, even if a receiver at a node is scheduled to be tuned to a channel during a time-slot, the actual tuning occurs only if the node is present in the network and the channel is in the availability set of the node. When describing our algorithm, we use  $\oplus_b$  to denote the modulo  $b$  addition defined as:

$$x \oplus_b y \triangleq ((x + y - 1) \bmod b) + 1$$

Therefore, given a sequence of numbers  $1, 2, \dots, b$ , the subsequence of  $j$  consecutive numbers starting from number  $i$  (with wrap-around) is given by  $i, i \oplus_b 1, \dots, i \oplus_b (j - 1)$ .

Let  $G = \{1, 2, \dots, N\}$  be the set of possible nodes in the network. For ease of exposition, we assume that  $r$  divides both  $N$  and  $M$ . The set  $G$  is partitioned into  $\frac{N}{r}$  groups  $G_1, G_2, \dots, G_{\frac{N}{r}}$  each of size  $r$ . Let  $G_1 = \{1, 2, \dots, r\}$ ,  $G_2 = \{r + 1, r + 2, \dots, 2r\}$ ,  $\dots$ ,  $G_{\frac{N}{r}} = \{N - r + 1, N - r + 2, \dots, N\}$  be the groups. In the description that follows, when a group of nodes  $\{i_1, i_2, \dots, i_x\}$  is said to be scheduled to transmit on channels  $\{c_{j_1}, c_{j_2}, \dots, c_{j_x}\}$  in a time-slot  $t$ , it implies that  $i_1$  is scheduled to transmit on  $c_{j_1}$ ,  $i_2$  is scheduled to transmit on  $c_{j_2}$ , and so on, up to  $i_x$  being scheduled to transmit on  $c_{j_x}$ . When a group of nodes is said to be scheduled to receive on channels  $\{c_{j_1}, c_{j_2}, \dots, c_{j_x}\}$ , it is understood that each node in the group is scheduled to tune at least one receiver to each channel in  $\{c_{j_1}, c_{j_2}, \dots, c_{j_x}\}$ . The nodes may choose to use exactly  $x$  receivers and turn off the remaining  $(r - x)$  receivers.

The algorithm consists of two rounds. Intuitively, the first round handles communication between nodes of different groups (*inter-group* communication), whereas the second round handles

communication between nodes of the same group (*intra-group* communication). The algorithm uses a *divide-and-conquer* approach to solve the problem of efficiently scheduling intra-group communication.

### 3.1 First round

The first round consists of  $\frac{N}{r}$  blocks. Each block consists of  $M$  time-slots. In block  $i$ , nodes in group  $G_i$  are scheduled to transmit and all other nodes are scheduled to listen. Specifically, in time-slot  $j$  of block  $i$ ,  $r$  nodes in  $G_i$ , given by  $(i-1)*r+1, (i-1)*r+2, \dots, i*r$ , are scheduled to transmit on  $r$  channels  $c_j, c_{j \oplus_M 1}, \dots, c_{j \oplus_M (r-1)}$ , respectively. The remaining nodes in the system are scheduled to listen on these  $r$  channels simultaneously using their  $r$  receivers. Table 1 gives an illustration of the first round when  $N = 8, M = 4$  and  $r = 2$ . The following can be easily verified for the first round.

**Lemma 1** *The first round satisfies the following property for each group  $G_k$  with  $1 \leq k \leq \frac{N}{r}$ : for every triplet  $\langle i, j, c \rangle$  such that  $i \in G_k, j \in \{1, \dots, N\} \setminus G_k$  and  $c \in \mathcal{A}_{univ}$ , there is a time-slot during which node  $i$  is scheduled to transmit on channel  $c$  and one of the receivers of node  $j$  is scheduled to listen on channel  $c$ .*

Before we describe the second round, we present an algorithm that we use as a subroutine in the second round.

### 3.2 Algorithm DAC

Consider a group of nodes  $A = \{a_1, a_2, \dots, a_x\}$  and a group of channels  $B = \{b_1, b_2, \dots, b_x\}$  such that each node has at least  $x$  receivers. For ease of exposition, assume that  $x$  is a multiple of 2, that is,  $x = 2y$  for some integer  $y$ . We describe an algorithm, based on divide-and-conquer approach, that ensures that every node in  $A$  is scheduled to listen to every other node in  $A$  on all  $x$  channels in  $O(x \log x)$  time-slots. Before we present the algorithm, we introduce some notation that we use in the description. Let  $\text{DAC}(A_1, B_1)$  and  $\text{DAC}(A_2, B_2)$  be two instances of the algorithm DAC. We use  $\text{DAC}(A_1, B_1) \parallel \text{DAC}(A_2, B_2)$  to denote the algorithm obtained by running the two instances concurrently. Of course, to be able to run the two instances concurrently, they should not interfere with each other. Therefore, in this case, we ensure that  $A_1 \cap A_2 = \emptyset$  and  $B_1 \cap B_2 = \emptyset$ . Further, we also ensure that  $|A_1| = |A_2|$  (which, in turn, implies that  $|B_1| = |B_2|$ ). As a result, the running

Time-slots	Nodes								
	Group $G_1$		Group $G_2$		Group $G_3$		Group $G_4$		
	1	2	3	4	5	6	7	8	
Block 1	1	$T$ $c_1$	$T$ $c_2$	$R$ $\{c_1, c_2\}$	$R$ $\{c_1, c_2\}$	$R$ $\{c_1, c_2\}$	$R$ $\{c_1, c_2\}$	$R$ $\{c_1, c_2\}$	$R$ $\{c_1, c_2\}$
	2	$T$ $c_2$	$T$ $c_3$	$R$ $\{c_2, c_3\}$	$R$ $\{c_2, c_3\}$	$R$ $\{c_2, c_3\}$	$R$ $\{c_2, c_3\}$	$R$ $\{c_2, c_3\}$	$R$ $\{c_2, c_3\}$
	3	$T$ $c_3$	$T$ $c_4$	$R$ $\{c_3, c_4\}$	$R$ $\{c_3, c_4\}$	$R$ $\{c_3, c_4\}$	$R$ $\{c_3, c_4\}$	$R$ $\{c_3, c_4\}$	$R$ $\{c_3, c_4\}$
	4	$T$ $c_4$	$T$ $c_1$	$R$ $\{c_1, c_4\}$	$R$ $\{c_1, c_4\}$	$R$ $\{c_1, c_4\}$	$R$ $\{c_1, c_4\}$	$R$ $\{c_1, c_4\}$	$R$ $\{c_1, c_4\}$
Block 2	5	$R$ $\{c_1, c_2\}$	$R$ $\{c_1, c_2\}$	$T$ $c_1$	$T$ $c_2$	$R$ $\{c_1, c_2\}$	$R$ $\{c_1, c_2\}$	$R$ $\{c_1, c_2\}$	$R$ $\{c_1, c_2\}$
	6	$R$ $\{c_2, c_3\}$	$R$ $\{c_2, c_3\}$	$T$ $c_2$	$T$ $c_3$	$R$ $\{c_2, c_3\}$	$R$ $\{c_2, c_3\}$	$R$ $\{c_2, c_3\}$	$R$ $\{c_2, c_3\}$
	7	$R$ $\{c_3, c_4\}$	$R$ $\{c_3, c_4\}$	$T$ $c_3$	$T$ $c_4$	$R$ $\{c_3, c_4\}$	$R$ $\{c_3, c_4\}$	$R$ $\{c_3, c_4\}$	$R$ $\{c_3, c_4\}$
	8	$R$ $\{c_1, c_4\}$	$R$ $\{c_1, c_4\}$	$T$ $c_4$	$T$ $c_1$	$R$ $\{c_1, c_4\}$	$R$ $\{c_1, c_4\}$	$R$ $\{c_1, c_4\}$	$R$ $\{c_1, c_4\}$
Block 3	9	$R$ $\{c_1, c_2\}$	$R$ $\{c_1, c_2\}$	$R$ $\{c_1, c_2\}$	$R$ $\{c_1, c_2\}$	$T$ $c_1$	$T$ $c_2$	$R$ $\{c_1, c_2\}$	$R$ $\{c_1, c_2\}$
	10	$R$ $\{c_2, c_3\}$	$R$ $\{c_2, c_3\}$	$R$ $\{c_2, c_3\}$	$R$ $\{c_2, c_3\}$	$T$ $c_2$	$T$ $c_3$	$R$ $\{c_2, c_3\}$	$R$ $\{c_2, c_3\}$
	11	$R$ $\{c_3, c_4\}$	$R$ $\{c_3, c_4\}$	$R$ $\{c_3, c_4\}$	$R$ $\{c_3, c_4\}$	$T$ $c_3$	$T$ $c_4$	$R$ $\{c_3, c_4\}$	$R$ $\{c_3, c_4\}$
	12	$R$ $\{c_1, c_4\}$	$R$ $\{c_1, c_4\}$	$R$ $\{c_1, c_4\}$	$R$ $\{c_1, c_4\}$	$T$ $c_4$	$T$ $c_1$	$R$ $\{c_1, c_4\}$	$R$ $\{c_1, c_4\}$
Block 4	13	$R$ $\{c_1, c_2\}$	$R$ $\{c_1, c_2\}$	$R$ $\{c_1, c_2\}$	$R$ $\{c_1, c_2\}$	$R$ $\{c_1, c_2\}$	$R$ $\{c_1, c_2\}$	$T$ $c_1$	$T$ $c_2$
	14	$R$ $\{c_2, c_3\}$	$R$ $\{c_2, c_3\}$	$R$ $\{c_2, c_3\}$	$R$ $\{c_2, c_3\}$	$R$ $\{c_2, c_3\}$	$R$ $\{c_2, c_3\}$	$T$ $c_2$	$T$ $c_3$
	15	$R$ $\{c_3, c_4\}$	$R$ $\{c_3, c_4\}$	$R$ $\{c_3, c_4\}$	$R$ $\{c_3, c_4\}$	$R$ $\{c_3, c_4\}$	$R$ $\{c_3, c_4\}$	$T$ $c_3$	$T$ $c_4$
	16	$R$ $\{c_1, c_4\}$	$R$ $\{c_1, c_4\}$	$R$ $\{c_1, c_4\}$	$R$ $\{c_1, c_4\}$	$R$ $\{c_1, c_4\}$	$R$ $\{c_1, c_4\}$	$T$ $c_4$	$T$ $c_1$

Table 1: An illustration of the first round when  $N = 8$ ,  $M = 4$  and  $r = 2$ . In the table, the symbol  $T$  (respectively,  $R$ ) means that the node is scheduled to transmit (respectively, receive).

time of the resulting algorithm is same as that of the either instance. Likewise, we use  $\text{DAC}(A_1, B_1) \circ \text{DAC}(A_2, B_2)$  to denote the algorithm obtained by running the two instances serially, one-by-one.

We now describe the algorithm. Let  $A = A_1 \cup A_2$  be a partition of  $A$  into two equal halves, where  $A_1 = \{a_1, a_2, \dots, a_y\}$  and  $A_2 = \{a_{y+1}, a_{y+2}, \dots, a_{2y}\}$  (recall that  $x = 2y$ ). The algorithm consists of

three blocks. In the first block consisting of  $x$  time-slots, all nodes in  $A_1$  are scheduled to transmit on all channels in  $B$  whereas nodes in  $A_2$  are scheduled to listen. The second block is similar to the first block except that roles of  $A_1$  and  $A_2$  are reversed, that is, nodes in  $A_2$  are scheduled to transmit while nodes in  $A_1$  are scheduled to listen. Finally, in the third block, we recursively invoke the algorithm if  $y > 1$ . (In other words, the base case for the DAC algorithm occurs when the group contains two nodes.) Let  $B = B_1 \cup B_2$  be a partition of  $B$  into two equal halves. The third block is then given by  $(\text{DAC}(A_1, B_1) \parallel \text{DAC}(A_2, B_2)) \circ (\text{DAC}(A_1, B_2) \parallel \text{DAC}(A_2, B_1))$ . The following can be easily verified for the algorithm DAC.

**Lemma 2** *For every triplet  $\langle i, j, c \rangle$  such that  $\{i, j\} \subseteq A$ ,  $i \neq j$  and  $c \in B$ , there is a time-slot during which node  $i$  is scheduled to transmit on channel  $c$  and one of the receivers of node  $j$  is scheduled to listen on channel  $c$ .*

If  $T(x)$  denotes the running time of  $\text{DAC}(A, B)$ , where  $|A| = |B| = x$ , we get the recurrence relation  $T(x) = 2x + 2T(x/2)$ . This yields  $T(x) = O(x \log x)$ .

### 3.3 Second round

Assume that  $M \leq N$ . (The schedule for the other case, when  $M \geq N$ , is described later.) Further, assume that  $M$  divides  $N$ . Let  $u = \frac{M}{r}$ . The second round is divided into  $\frac{N}{M}$  epochs. In each epoch, only  $u (= \frac{M}{r})$  groups participate, that is, scheduled to transmit and/or receive. Specifically, in epoch  $i$ ,  $u$  groups, given by  $G_{(i-1)*u+1}, G_{(i-1)*u+2}, \dots, G_{i*u}$ , complete their intra-group communication. To that end, we partition the  $M$  channels into  $u$  groups of channels  $C_1, C_2, \dots, C_u$  each of size  $r$ . Each epoch consists of  $u$  sub-epochs. Let  $(i-1)*u = s$ . The schedule for sub-epoch  $j$  in epoch  $i$  is given by:

$$\text{DAC}(G_{s+1}, C_j) \parallel \text{DAC}(G_{s+2}, C_{j \oplus_u 1}) \parallel \dots \parallel \text{DAC}(G_{s+u}, C_{j \oplus_u (u-1)})$$

Let the above schedule to cover inter-group communication when  $M \leq N$  be denoted by WGND (Within Group Neighbor Discovery). Note that WGND takes  $G$  and  $\mathcal{A}_{univ}$  as inputs, where  $|\mathcal{A}_{univ}| \leq |G| = N$ . Now, consider the case of  $M \geq N$  and assume that  $N$  divides  $M$ . The first round is identical to the case of  $M \leq N$ . For the second round, we use the algorithm described for  $M \leq N$  as a subroutine. Let  $v = \frac{M}{N}$ . The universal set of channels  $\mathcal{A}_{univ}$  is partitioned into  $v$  groups of channels  $C_1, C_2, \dots, C_v$ , each of size  $N$ . Note that  $|G| = |C_1| = |C_2| = \dots = |C_v| = N$ .

		Time-slots	Nodes							
			Group $G_1$		Group $G_2$		Group $G_3$		Group $G_4$	
			1	2	3	4	5	6	7	8
Epoch 1	Sub-epoch 1	1	$T$ $c_1$	$R$ $\{c_1\}$	$T$ $c_3$	$R$ $\{c_3\}$				
		2	$T$ $c_2$	$R$ $\{c_2\}$	$T$ $c_4$	$R$ $\{c_4\}$				
		3	$R$ $\{c_1\}$	$T$ $c_1$	$R$ $\{c_3\}$	$T$ $c_3$				
		4	$R$ $\{c_2\}$	$T$ $c_2$	$R$ $\{c_4\}$	$T$ $c_4$				
	Sub-epoch 2	5	$T$ $c_3$	$R$ $\{c_3\}$	$T$ $c_1$	$R$ $\{c_1\}$				
		6	$T$ $c_4$	$R$ $\{c_4\}$	$T$ $c_2$	$R$ $\{c_2\}$				
		7	$R$ $\{c_3\}$	$T$ $c_3$	$R$ $\{c_1\}$	$T$ $c_1$				
		8	$R$ $\{c_4\}$	$T$ $c_4$	$R$ $\{c_2\}$	$T$ $c_2$				
Epoch 2	Sub-epoch 1	9					$T$ $c_1$	$R$ $\{c_1\}$	$T$ $c_3$	$R$ $\{c_3\}$
		10					$T$ $c_2$	$R$ $\{c_2\}$	$T$ $c_4$	$R$ $\{c_4\}$
		11					$R$ $\{c_1\}$	$T$ $c_1$	$R$ $\{c_3\}$	$T$ $c_3$
		12					$R$ $\{c_2\}$	$T$ $c_2$	$R$ $\{c_4\}$	$T$ $c_4$
	Sub-epoch 2	13					$T$ $c_3$	$R$ $\{c_3\}$	$T$ $c_1$	$R$ $\{c_1\}$
		14					$T$ $c_4$	$R$ $\{c_4\}$	$T$ $c_2$	$R$ $\{c_2\}$
		15					$R$ $\{c_3\}$	$T$ $c_3$	$R$ $\{c_1\}$	$T$ $c_1$
		16					$R$ $\{c_4\}$	$T$ $c_4$	$R$ $\{c_2\}$	$T$ $c_2$

Table 2: An illustration of the second round when  $N = 8$ ,  $M = 4$  and  $r = 2$ .

The second round for the case of  $M \geq N$  is then given by:

$$\text{WGND}(G, C_1) \circ \text{WGND}(G, C_2) \circ \text{WGND}(G, C_3) \circ \cdots \circ \text{WGND}(G, C_v)$$

where, as in the case of DAC, the symbol “ $\circ$ ” denotes serial composition of two instances of WGND.

Table 2 gives an illustration of the second round when  $N = 8$ ,  $M = 4$  and  $r = 2$ . The following can be easily verified using Lemma 2 for the second round.

**Lemma 3** *The second round satisfies the following property for each group  $G_k$  with  $1 \leq k \leq \frac{N}{r}$ : for every triplet  $\langle i, j, c \rangle$  such that  $\{i, j\} \subseteq G_k$ ,  $i \neq j$  and  $c \in \mathcal{A}_{univ}$ , there is a time-slot during which node  $i$  is scheduled to transmit on channel  $c$  and one of the receivers of node  $j$  is scheduled to listen on channel  $c$ .*

### 3.4 Proof of Correctness

It follows from Lemma 1 that all triplets corresponding to inter-group communication are covered in the first round. Further, it follows from Lemma 3 that all triplets corresponding to intra-group communication are covered in the second round. Therefore we have:

**Theorem 4** *The neighbor discovery algorithm ensures that if a node  $i$  is a neighbor of node  $j$  on a channel  $c$  then there is a time-slot  $t$  such that only node  $i$  transmits on channel  $c$  during time-slot  $t$  and at least one receiver of node  $j$  is tuned to channel  $c$  during time-slot  $t$ .*

### 3.5 Running time

The first round uses  $M \times \frac{N}{r} = \frac{MN}{r}$  time-slots. When  $M \leq N$ , the second round uses  $\frac{N}{M} \times \frac{M}{r} \times O(r \log r) = O(N \log r)$  time-slots. Note that, so far, we have assumed that  $r$  divides  $M$ ,  $M$  divides  $N$  and  $r$  is a power of 2. If any one of these properties does not hold, the running time increases by a constant factor only. Specifically, for general values of  $M$ ,  $N$  and  $r$ , the total running time is  $M \lceil \frac{N}{r} \rceil + O(N \log r)$ . For  $M \geq N$ , the second round uses  $\lceil \frac{M}{N} \rceil \times O(N \log r) = O(M \log r)$  time-slots yielding a total running time of  $M \lceil \frac{N}{r} \rceil + O(M \log r)$ . Therefore, in general, the running time of the algorithm is given by  $M \lceil \frac{N}{r} \rceil + O(\max(M, N) \log r)$ .

**Theorem 5** *The neighbor discovery algorithm has time complexity of  $M \lceil \frac{N}{r} \rceil + O(\max(M, N) \log r)$ , where  $N$  is the size of the label space,  $M$  is the size of the universal channel set and  $r$  is the number of receivers at a node.*

### 3.6 Discussion

The time-complexity of our algorithm depends on the size of the space from which identifiers are selected and not on actual number of nodes. Of course, the latter is more desirable. One may think that the reason for high time-complexity is because our algorithm is *oblivious* in nature (a node's action does not depend on the knowledge it has gained so far), and ensures the schedule is collision-free. Even with the above restrictions, surprisingly our algorithm is "close" to optimal

because it can be shown that any deterministic algorithm for neighbor discovery has to use  $\Omega(\frac{MN}{r})$  time-slots in the worst case even if (1) the algorithm is adaptive, (2) the network contains a single node, and (3) collisions are allowed [16]. The lower bound holds as long as nodes do not know the actual number of nodes in the network, and nodes cannot distinguish between background noise and collision noise. When the network contains a single node, the worst-case time complexity of neighbor discovery is  $\Omega(MN)$  in case it is desirable that nodes be able to detect *termination* of the neighbor discovery algorithm. In this case, if the node does not receive any message during a time-slot, it does not know whether it is because of collision or simply because it does not have any neighbor. The same proof can be easily extended to show a lower bound of  $\Omega(\frac{MN}{r})$  time-slots when each node has  $r$  receivers.

Although we describe the neighbor discovery algorithm assuming an undirected network (that is, neighborhood relationships are symmetric), our algorithm can also be used (without any modification) by nodes to discover their *incoming* neighbors in a directed network. Additional rounds are required if knowledge of *outgoing* neighbors is also desired.

Note that we assume  $r$  to be at most  $\min(M, N)$ . Even if  $r > M$ , only up to  $M$  receivers can be used during any time-slot to gain useful information about the network. An additional receiver, if used, has to be tuned to one of the  $M$  channels implying that two receivers are tuned to the same channel. Likewise, if  $r > N$ , then only  $N$  receivers (out of  $r$ ) can possibly receive a message during any time-slot. This is because there can be at most  $N$  concurrent transmissions during any time-slot. Therefore, for an oblivious neighbor discovery algorithm (such as the algorithm described in this paper), in which a node knows the entire transmission schedule *a priori*, it does not help to have more  $N$  receivers. This does not mean that a neighbor discovery algorithm cannot be designed to make use of more than  $N$  receivers.

## 4 Conclusion and Future Work

We have presented a TDMA-based deterministic distributed algorithm for neighbor discovery when a node has multiple receivers but only a single transmitter. Each node in the network is assumed to have no prior knowledge of its neighborhood and the channels it could use to communicate with its neighbors. Further, we do not assume the existence of any common control channel. As future work, we plan to identify conditions under which it is possible to devise *faster* algorithms for neighbor discovery whose time-complexity depends on the actual number of nodes in the network

rather than the maximum number of nodes in the network. We also plan to develop *randomized* neighbor discovery algorithms, which may have smaller (expected) running time than deterministic algorithms.

## References

- [1] R. W. Broderson, A. Wolisz, D. Cabric, S. M. Mishra, and D. Willkomm. CORVUS: A Cognitive Radio Approach for Usage of Virtual Unlicensed Spectrum. Available at [http://bwrc.eecs.berkeley.edu/Research/MCMA/CR\\_White\\_paper\\_final1.pdf](http://bwrc.eecs.berkeley.edu/Research/MCMA/CR_White_paper_final1.pdf).
- [2] B. S. Chlebus, L. Gąsieniec, A. Gibbons, A. Pelc, and W. Rytter. Deterministic Broadcasting in Ad Hoc Radio Networks. *Distributed Computing (DC)*, 15(1):27–38, January 2002.
- [3] M. Chrobak, L. Gąsieniec, and W. Rytter. Fast Broadcasting and Gossiping in Radio Networks. *Journal of Algorithms*, 43(2):177–189, 2002.
- [4] A. Czumaj and W. Rytter. Broadcasting Algorithms in Radio Networks with Unknown Topology. In *Proceedings of the 44th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 492–501, Cambridge, Massachusetts, USA, December 2003.
- [5] Federal Communications Commission. *FCC Spectrum Policy Task Force Report, ET Docket No. 02-155*, November 2002.
- [6] L. Gąsieniec, A. Pagourtzis, I. Potapov, and T. Radzik. Deterministic Communication in Radio Networks with Large Labels. Technical Report TR-04-12, Department of Computer Science, King’s College London, May 2004.
- [7] L. Gąsieniec, T. Radzik, and Q. Xin. Faster Deterministic Gossiping in Directed Ad Hoc Radio Networks. In *Proceedings of the 9th Scandinavian Workshop on Algorithm Theory (SWAT)*, pages 397–407, Humlebaek, Denmark, July 2004.
- [8] N. Jain, S. Das, and A. Nasipuri. A Multichannel CSMA MAC Protocol with Receiver-based Channel Selection for Multihop Wireless Networks. In *Proceedings of the IEEE International Conference on Computer Communications and Networks (IC3N)*, Phoenix, Arizona, USA, October 2001.
- [9] M. Kodialam and T. Nandagopal. Characterizing the Capacity Region in Multi-Radio, Multi-Channel Wireless Mesh Networks. In *Proceedings of the IEEE Conference on Computer Communications (INFOCOM)*, August 2005.
- [10] D. R. Kowalski and A. Pelc. Broadcasting in Undirected Ad Hoc Radio Networks. In *Proceedings of the 22nd ACM Symposium on Principles of Distributed Computing (PODC)*, pages 73–82, Boston, Massachusetts, USA, July 2003.
- [11] D. R. Kowalski and A. Pelc. Deterministic Broadcasting Time in Radio Networks of Unknown Topology. In *Proceedings of the 43rd Annual Symposium on Foundations of Computer Science (FOCS)*, pages 63–72, Vancouver, British Columbia, Canada, November 2005.

- [12] D. R. Kowalski and A. Pelc. Time Complexity of Radio Broadcasting: Adaptiveness vs. Obliviousness and Randomization vs. Determinism. *Theoretical Computer Science*, 333(3):355–371, March 2005.
- [13] S. Krishnamurthy, R. Chandrasekaran, N. Mittal, and S. Venkatesan. Brief Announcement: Synchronous Distributed Algorithms for Node Discovery and Configuration in Multi-channel Cognitive Radio Networks. In *Proceedings of the Symposium on Distributed Computing (DISC)*, September 2006.
- [14] S. Krishnamurthy, M. Thoppian, S. Kuppa, R. Chandrasekaran, N. Mital, S. Venkatesan, and R. Prakash. Time-efficient Distributed Layer-2 Auto-configuration for Cognitive Radio Networks. *Computer Networks*, 52(4):831–849, March 2008.
- [15] J. Mitola III. *Cognitive Radio: An Integrated Agent Architecture for Software Defined Radio*. PhD thesis, Royal Institute of Technology (KTH), Stockholm, Sweden, May 2000.
- [16] N. Mittal, S. Krishnamurthy, R. Chandrasekaran, and S. Venkatesan. A Fast Deterministic Algorithm for Neighbor Discovery in Multi-Channel Cognitive Radio Networks. Technical Report UTDCS-14-07, Department of Computer Science, The University of Texas at Dallas, March 2007.
- [17] A. Nasipuri and S. R. Das. Multichannel CSMA with Signal Power-based Channel Selection for Multihop Wireless Networks. In *Proceedings of the Vehicular Technology Conference*, September 2000.
- [18] A. Nasipuri, J. Zhuang, and S. R. Das. A Multichannel CSMA MAC Protocol for Multihop Wireless Networks. In *Proceedings of the IEEE Wireless Communications and Networking Conference*, September 1999.
- [19] A. Raniwala and T. Chiueh. Architecture and Algorithms for an IEEE 802.11-Based Multi-Channel Wireless Mesh Network. In *Proceedings of the IEEE Conference on Computer Communications (INFOCOM)*, March 2005.
- [20] A. Raniwala, K. Gopalan, and T. Chiueh. Centralized Channel Assignment and Routing Algorithms for Multi-channel Wireless Mesh Networks. *ACM SIGMOBILE Mobile Computing and Communications Review*, 8(2):50–65, April 2004.
- [21] J. So and N. H. Vaidya. Multi-channel MAC for Ad Hoc Networks: Handling Multi-Channel Hidden Terminals using a Single Transceiver. In *Proceedings of the 5th ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc)*, pages 222–233, 2004.
- [22] J. So and N. H. Vaidya. Routing and Channel Assignment in Multi-Channel Multi-Hop Wireless Networks with Single Network Interface. In *Proceedings of the 2nd International Conference on Quality of Service in Heterogeneous Wired/Wireless Networks (QShine)*, August 2005.
- [23] M. Thoppian, S. Venkatesan, R. Prakash, and R. Chandrasekaran. MAC-Layer Scheduling in Cognitive Radio based Multi-hop Wireless Networks. In *Proceedings of the International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM)*, pages 191–202, June 2006.
- [24] S. Wu, C. Lin, Y. Tseng, and J. Sheu. A New Multi-Channel MAC Protocol with On-Demand Channel Assignment for Multihop Mobile Ad Hoc Networks. In *Proceedings of the International Symposium on Parallel Architectures, Algorithms and Networks (ISPAN)*, pages 232–232, 2000.