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.2014.2364549, IEEE Transactions on Vehicular Technology

Boundary Helps: Reliable Route Selection With Directional Antennas in Cognitive Radio Networks

Ying Dai,† Jie Wu,† and Yaxiong Zhao‡
Department of Computer and Information Sciences, Temple University†
Google Inc.‡
Email: {ying.dai, jiewu}@temple.edu, yaxiongzhao@google.com

***Abstract*—The unpredictable activities of primary users (PUs) make the channel availabilities in cognitive radio networks (CRNs) very unstable, which causes routing in CRNs to be more difficult than in traditional wireless networks. Specifically, when a source node needs to select a route to reach the destination, the "optimal" route during the route selection phase may not be optimal in the data transmission phase. In this paper, we propose a novel routing protocol based on the traditional source routing protocols. We consider the angle dimension by assuming that directional antennas are equipped on every node, which facilitate the marking of boundaries of PUs. We use the USRP/Gnuradio testbed to show the sensing result differences of different directions at the boundary area of a PU. For every optional route between a source and a destination node, we evaluate its reliability and other performance by evaluating the PU areas it passes through, and by estimating the possible transmission rate. Based on these parameters, we propose an algorithm for route selection, considering both the reliability and delay. Our routing protocol only requires very limited piggyback information, and it is highly adaptable under the dynamic channel availabilities. We evaluate our approach through extensive simulations.**

***Index Terms*—Cognitive radio networks, directional antennas, USRP/Gnuradio, boundary nodes, routing.**

I. Introduction

Nodes, referred to as secondary users (SUs), in *cognitive radio networks* (CRNs) [1] can make opportunistic use of multiple channels not occupied by primary users (PUs). However, when a primary user becomes active and occupies a channel, SUs on that channel need to quit immediately. Therefore, some links can possibly be broken. The dynamics of channel availabilities result in the difficulty of routing in CRNs, and in carrying out end-to-end data transmission.

There have been many works on routing in CRNs [2]. Since the dynamic channel availabilities affect the delay and reliability of each route, the route selection algorithm needs to consider the channel availability situation of each optional route. A simple solution would be that each node collects its own information about PUs, and piggybacks that information to the source node for route selection. However, it will cause a lot of information exchange and burden the control channels.

Most of the existing routing protocols rely on the piggybacked channel information, and build their metrics regarding multiple parameters, e.g., channel availability and route quality. Then the route selection is usually based on these metrics. However, there are two main problems with these protocols. First, the overhead of the piggybacked information is usually too large, which makes it impractical when considering the energy and interference. Second, the piggybacked channel availabilities cannot convey the instant channel situation because, at the time the data is being transmitted, the channel availabilities are possibly different. Therefore, a better protocol should be able to take both the overhead and channel availability dynamics into consideration.

We consider the routing problem in a novel way, and make use of the directional antennas to help route selection. There have been many works done using directional antennas to benefit the data transmission in traditional wireless ad hoc networks. There are two benefits to applying directional antennas. One is the reduction of radio interference. Thus, in CRNs, it increases the spacial reuse opportunities among PUs and SUs, and also for SUs themselves. Another benefit of directional antennas lies in the determination of if a node is located at the boundary of a PU area. The different directional antennas on each node do not need to be globally aligned, which is similar to the directional antenna model in [3].

In our paper, we define the boundary node, and each node is able to decide whether it is, itself, a boundary node or not. The source node makes use of the information returned by boundary nodes along each possible route, measures the channel availability and stability of each route, and chooses the best one to reach the destination node through our algorithm. During the process of route selections by the source node, the boundary nodes' information can be used as a "traffic blocker", which "blocks" traffic from entering too many PU areas. To improve the feasibility of our model, we also consider the situations of imperfect information.

The organization of our paper is as follows. In Section II, we discuss the related works; the problem is defined in Section III; our routing protocol is described in Section IV; we show the improvement of our model's feasibility with imperfect information in Section V; the extensions of error detections are discussed in Section VI; performance evaluations are presented in Section VII; the conclusion is in Section VIII.

II. Related Works

Many works have been done on routing protocols in CRNs. Most of them consider both routing and channel assignments. In [4], the authors propose a protocol called, opportunistic

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.2014.2364549, IEEE Transactions on Vehicular Technology

cognitive routing (OCR), which enables each user to make use of its geographical information and collect channel usage statistics. In [5], the authors study a routing protocol called CRP, which maps the spectrum selection metrics and local PU interference observations to a packet forwarding delay over the control channel. Our work is different from the above protocols, since our model makes use of directional antennas, and only causes very limited overhead while considering the dynamic channel availabilities.

There have also been some works done on the angle dimension of CRNs. In [6], the authors propose a scheme with relays or directional relays for SUs to exploit new spectrum opportunities, and provide higher spectrum efficiency by coexistence of primary and CR users at the same region, time, and spectrum band. In [7], the authors use an electronically steerable parasitic antenna receptor to study the spectrum sensing for cognitive radio. They divide the angular domain into sectors, and detect signals from PUs on a time domain. In [8], multicast communications in CRNs using directional antennas are studied. In [9], the joint optimization of the antenna orientation and spectrum allocation is studied. Our model is different from the above ones, since we make use of the directional antenna to solve the routing problem.

III. Problem Formulation

A. System Model & Constraints

We consider a CRN with the node set, $\{a, b, c, ...\}$. Each node is equipped with directional antennas, which divides the omnidirectional transmission range of each node into a number of sectors. We assume that each node is static. The total available channel set is denoted as M, which is licensed to a set of PUs, whose activities are unknown. During the data transmission, each node selects one sector to send the data. We assume that there is a common control channel (CCC) for nodes to coordinate. Our model can also be extended to environments without a CCC, using the channel rendezvous approaches in [10].

There are several constraints that need to be satisfied for successful data transmission, regardless of which sector is adopted by each user. When node a sends data to node b, they must tune in to the same channel, $m \in M$. Suppose the power used by node a is P_a. The minimum SINR requirement at b is α_b. Therefore, we have

$$\frac{P_a g_{ab}}{N_0 + I} > \alpha_b. \tag{1}$$

Moreover, suppose that the nearby primary user pairs are working on the same channel m. Then, we have

$$\frac{P_p g_{pp}}{N_0 + I + g_{ap} P_a} > \alpha_p, \tag{2}$$

where P_p is the power adopted by $PU-TX$, g_{pp} is the power gain from $PU-TX$ to $PU-RX$, g_{ap} is the power gain from node a to $PU-RX$, and α_p denotes the desired SINR requirement of $PU-RX$. The data transmission from a to b is successful only if the above two constraints are satisfied.

Since the position of any $PU-TX$ is unknown to SUs, to make sure the PU sessions are not interfered with, we strengthen Constraint (2) as for any point within $PU-TX$'s area, instead of only $PU-RX$, where the SINR value is above α_p; P_a must satisfy Constraint (2), no matter which sector a uses. Therefore, when a and $PU-TX$ are working on the same channel, we have the following three situations regarding the constraint of P_a, based on the distance between $PU-TX$ to node a:

$$P_a = \begin{cases} 0 & \frac{P_p g_{pa}}{N_0+I} > \alpha_p, \\ P'_a & \frac{P_p g_{pp'}}{N_0+I} = \alpha_p,\ g_{ap'} P'_a \to 0, \\ P_{max} & \frac{P_p g_{pp'}}{N_0+I} = \alpha_p,\ g_{ap'} P_{max} = 0. \end{cases} \tag{3}$$

The first case is that when the SINR value of $PU-TX$ at a's location is above α_p, a cannot use the channel of $PU-TX$. The second case is that a can use the channel of $PU-TX$, but the interference caused by a cannot make any point that has a SINR greater than or equal to α_p as being less than α_p. p' is a boundary point of $PU-TX$'s transmission area, where the SINR is equal to α_p. The third case is that a is far from $PU-TX$'s transmission area. Node a can transmit at the maximal power P_{max} without causing any significant interference to any point within $PU-TX$'s transmission area.

B. Objective

Suppose there are session requests in the CRN. For a source node S, the objective of our model is to find the route with the minimal delay while ensuring the reliability, as to reach the destination node, D. The channel availabilities on each link play an important factor in determining the overall delay. Since the channel availabilities on each link are dynamic, it is impractical to find the optimal solution. Even if a route provides the minimal delay at a given time, it is unable to ensure the minimal delay during the entire session.

We provide a protocol for routing which considers both delay and reliability, with the help of directional antennas. In our model, to select a route, there are four factors that need to be taken into consideration: the nodes on the route, the sector adopted by each node to transmit, the channel used on that sector, and the power allowed on that channel. Considering that the interference constraints of PUs and SUs are both dynamic, it is impractical to find the optimal solution. We propose an effective routing protocol, which makes use of the directional antennas, and efficiently reduces the overhead during the exchange of control messages on CCC. Our model focuses on piggyback and route selection phases: after the route is selected, the channel management and channel selection for each single link is out of scope.

IV. Routing Protocol with Directional Antennas

A. Overview

Since each node in our model is equipped with a directional antenna, for a node a, we use the 2-tuple (IN_a, OUT_a) where IN_a denotes the sector ID that a packet is received by a, and OUT_a denotes the sector ID that the packet is sent out by a.

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.2014.2364549, IEEE Transactions on Vehicular Technology

Similar to the source routing in traditional ad hoc networks, in our model, the source node first needs to find the route to reach the destination node using the following process:

- The source node, S sends the route request (RREQ) packet through the CCC from all sectors. The RREQ contains the ID of the destination node D, denoted as $< S, D >$. Also, for every sector to which the RREQ is sent out, it also contains the OUT_S. Since it is the source node, the IN_S is empty.
- For any node a that receives the RREQ, it would add its own node ID and broadcast the request through all of its sectors. Moreover, it would also add (IN_a, OUT_a) to the RREQ. Obviously, IN_a is the same for all sectors, from which the RREQ is received. OUT_a differs among the RREQs from different sectors.
- The above two processes continue until the destination node is reached. Then the RREQ will contain all the node IDs from S to D, denoted as $< S, ..., a, ..., D >$. The RREQ also contains the IN and OUT sector IDs of each node on the path, for example, the (IN_a, OUT_a) of node a. In addition, for destination node D, it only contains IN_D, since the OUT_D is empty.

After the destination node D is reached, it will reply with the route reply (RREP) packet over CCC, along with the route information (node IDs and sector numbers) in the RREQ packet, to the source node. The underlying MAC protocol can make use of works in [11], since the RREQ and RREP messages are sent through CCC from all sectors, which convert the MAC problem very similar to that in the traditional wireless networks. The relay node selections and avoiding of $RREQ$ cycles can be performed by applying the approach in [12]. Since, in most cases, there are several routes from S to D, the source node S needs to select one of them. It is intuitive to consider adding the channel availability situation of the corresponding sector on each node to the RREP message along the route, and piggybacking it to S. However, this is impractical. Since the channel availabilities on each sector of each node are dynamic, the channel availability of each node can differ between the piggyback phase and the data transmission phase. Also, it would cause lots of overhead to return the channel availability of every node on the route. In our model, we make use of the directional antennas, and propose an efficient route selection scheme.

B. Boundary Node

We first give the definition of *boundary node* under our model. There are many existing boundary detection algorithms [13]. Our definition here mainly describes what the boundary we refer to in CRNs.

Definition 1: Node a is a boundary node regarding the $PU-TX$ on channel m if the variance, $V_a(m)$, of the sensing results in all sectors of node a is above a threshold, ν. We use $B_a(m) = 1$ to denote that a is the boundary node of $PU-TX$ that occupies channel m. Then

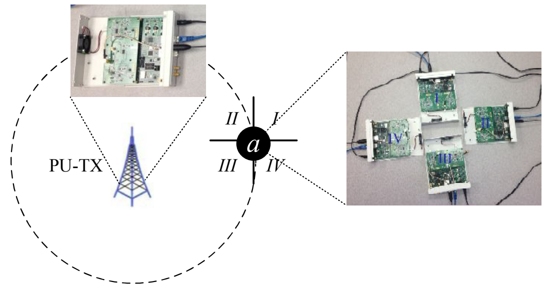


Fig. 1. Testbed for showing the characteristic of a boundary node.

$$B_a(m) = \begin{cases} 0 & V_a(m) \le \nu, \\ 1 & V_a(m) > \nu. \end{cases} \tag{4}$$

From the definition, we can see that the boundary nodes are a region of nodes, rather than a line of nodes. The variance here refers to the variances of sensing results among different sectors, as seen in the following experiment. We use the USRP/Gnuradio testbed to show the difference of the received SINR at different sectors of a boundary node. As shown in Fig. 1, to simulate a SU with a four-directional antenna, we use four USRP N200s, and each of them denotes one sector. Another USRP N200 is used to simulate a $PU - TX$. We use narrowband communication here. The PU sends on the channel with central frequency 1.3005GHz, and the other 4 USRP N200s receive at the same channel. The approximate SINR at sector I is about -50 dB, while the value at sector II is about -87 dB. The differences of SINR values over time at different sectors of a boundary node are very obvious.

Besides, each boundary node records the historical probability, which is the probability of its corresponding PU being active in a constant time range. For example, in Fig. 2, node c is a boundary node with $B_c(m_1) = 1$. It will maintain the active probability $PB_c(m_1)$ of $PU - TX$ in time range T. The value of $PB_c(m_1)$ is updated by node c for channel m every T, since $B_c(m_1) = 1$. We will use the $PB_c(m_1)$ in our piggyback scheme.

Many merits of boundary nodes have been studied in traditional wireless networks. In addition, boundary nodes in CRNs can facilitate the routing path selection. They can help to differentiate the routes that go into or avoid the $PU-TX$'s area. For example, in Fig. 2, suppose that the two $PU-TX$s occupy channel m_1 and m_2, and are randomly active. Route R that goes out from sector III of node c is different from route R' that goes from sector IV. Intuitively, when channel m_1 is unavailable, route R is better than route R' because the following links of c on route R' are more likely to be broken, which is unreliable and would cause more delay.

C. Threshold-Based Piggyback Scheme with Limited Overhead

Having the boundary node definition, each node can identify if it is, itself, a boundary node of a certain primary user during the spectrum sensing phase. Our protocol will make use of boundary nodes during the piggyback phase.

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.2014.2364549, IEEE Transactions on Vehicular Technology

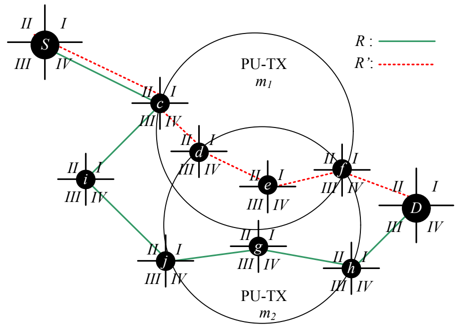


Fig. 2. Two possible routes from S to D.

As stated at the beginning of this section, when node a receives the RREQ packet, it will add both its ID and the 2-tuple (IN_a, OUT_a) to the RREQ. However, if a is a boundary node of the PU occupying channel m, and the active probability of that PU is above a predefined threshold γ, it will add the 4-tuple $(IN_a, OUT_a, m, \mu_a(m))$ to the RREQ, where $\mu_a(m)$ equals 1 or -1, indicating the entrance to, or exiting of, the PU area of channel m. More specifically, we provide the following three cases for node a to decide which information it will add to the current RREQ packet:

- If $\exists m \in M$ that satisfies $B_a(m) = 1$ & $PB_a(m) > \gamma$, and m is unavailable on the sector number OUT_a from which the RREQ is sent out, rather than a's ID and its 2-tuple (IN_a, OUT_a), a would add the 4-tuple $(IN_a, OUT_a, m, \mu_a(m))$ to the RREQ. Here, m is the channel that is unavailable in sector OUT_a, and $\mu_a(m) = 1$, which indicates the entrance to the PU area.
- If $\exists m \in M$ that satisfies $B_a(m) = 1$ & $PB_a(m) > \gamma$, and m is unavailable on the sector number IN_a from which the RREQ is received, a would add its ID and the 4-tuple $(IN_a, OUT_a, m, \mu_a(m))$ to the RREQ, where m is the channel that is unavailable in sector IN_a, and $\mu_a(m) = -1$, which indicates an exit from the PU area.
- Otherwise, a would only add its ID and the 2-tuple (IN_a, OUT_a) to the RREQ.

The first case presents the situation in which the route enters the PU area occupying m, reported by the boundary node a. The second case represents the situation in which the route leaves the PU area occupying m, reported by the boundary node a. In both cases, the PU occupying m does not have to be active at the time when RREQ is transmitted, as long as they are previously measured by the boundary nodes and their active probability measured by a is above a predefined threshold γ. The third case is for nodes that are not boundary nodes, or nodes that are boundary nodes but the active probability of PU is below the threshold γ , regardless of if they are inside or outside the PU areas.

The reason that the active probability of PU during T has to be above the predefined threshold γ is because different PUs have different active levels. For example, some PUs are active much less frequently than other PUs. It is possible that entering these PU areas could achieve a better performance than choosing other routes, which do not go through those PU areas but take longer hop distances to reach the destination. The route selection algorithm is discussed in detail in the following parts.

For example, in Fig. 2, the two $PU-TX$s occupy channels m_1 and m_2. There are two optional routes, R and R', from source S to destination D. Node j in Fig. 2 satisfies the first case, where $B_j(m_2) = 1$, and m_2 is unavailable on sector I ($OUT_j = I$). j would add its ID and the 4-tuple $(II, I, m_2, 1)$ to the RREQ. Node h in Fig. 2 belongs to the second case. Thus, h would add its ID and $(II, I, m_2, -1)$ to the RREQ. Node i in Fig. 2 meets the conditions of the third case. Therefore, i only adds its ID and (I, IV) to the RREQ. The burden of CCC is reduced since only boundary nodes are required to return extra information.

After the destination node D is reached, it copies the route information and the added 2 or 4-tuple information by each node in RREQ, and piggybacks to source node S in RREP. Using route R in Fig. 2 as an example, the RREP would contain the node IDs, $< S, c, i, j, g, h, D >$, on R, and also the 2 or 4-tuple attributes of each node. Among all nodes on route R, j has the 4-tuple $(II, I, m_2, 1)$, and h has the 4-tuple $(II, I, m_2, -1)$. The others have 2-tuple, indicating the IN and OUT sector numbers.

D. Route Selection Algorithm

After the source node receives the RREP along with piggybacked information, it needs to perform the route selection, since there is usually more than one route that can reach the destination node. Due to the dynamics of channel availabilities, it is impractical to estimate the delay of each route and choose the optimal one. To achieve our goal, we provide a heuristic approach to estimate the delay of each route.

The calculation of the route length is conducted by the source node, which makes use of the information contained in the piggybacked RREP packet. We use ab to denote a single link from a to b on the optional route. a and b have a 2-tuple or 4-tuple attribute, depending on whether it is a boundary node or not.

We start by defining whether a link is inside or outside a PU area that occupies channel m. The source node treats the nodes that return a 2-tuple as a non-boundary node. For example, as discussed above, if a node a is a boundary node but the active probability of PU is below a threshold, it only returns a 2-tuple. The source node would treat a as a non-boundary node, as well as other real non-boundary nodes.

Definition 2: A single link ab is inside the PU area that occupies channel m if any of the three cases are satisfied:

- $B_a(m) = 1, \mu_a(m) = 1$;
- $B_a(m) = 0, B_b(m) = 1, \mu_b(m) = -1$;
- $B_a(m) = 0, B_b(m) = 0$, and $\exists c$, which satisfies $B_c(m) = 1$; c is the boundary node nearest to a among all the boundary nodes before a on the given route, and it satisfies $\mu_c(m) = 1$.

Otherwise, the link is outside the PU area of m. For a given route, we use $I_{ab}(m) = 1$ to denote that the link ab is inside

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.2014.2364549, IEEE Transactions on Vehicular Technology

the PU area of m, and $I_{ab}(m) = 0$ to denote that it is outside the PU area of m.

Next, we give examples of the three cases, and one special case (Case 4), in which a link is located within multiple PU areas in Fig. 3:

- Case 1: Link ab is in the PU area since a is a boundary node and $\mu_a(m) = 1$, which means link ab has entered the PU area of channel m and is within it;
- Case 2: Link ab is in the PU area since node b is a boundary node and $\mu_b(m) = -1$, which means link ab is also in the PU area of channel m;
- Case 3: Link ab's closest boundary node for the PU area of channel m is c and $\mu_c(m) = 1$, which means link ab is in the PU area;
- Case 4: For channel m_1, similar to Case 3, link ab's closest boundary node is d and $\mu_d(m_1) = 1$; for channel m_2, link ab's closest boundary node is c and $\mu_c(m_2) = 1$. Therefore, link ab is within two PU areas of channel m_1 and m_2.

From the above discussions, the source node can tell the number of PU areas that a single link passes through. Based on that, we define the weighted length of a single link, which will be used later to define the weighted length of a route. For a node that is a boundary node of multiple PUs, e.g., two PU areas, then it replies with two 4-tuples to indicate the entering or leaving the two PU areas.

Definition 3: For a single link ab on a given route, the weighted length of the single link L_{ab} is calculated as:

$$L_{ab} = \begin{cases} 1 & I_{ab}(m) = 0, \forall m \in M; \\ \frac{|M|}{|M| - \mathcal{C}(m)} & I_{ab}(m) = 1; \end{cases} \tag{5}$$

where $\mathcal{C}(m)$ counts the number of channels on link ab that satisfy $I_{ab}(m) = 1$, which means that ab is inside the PU area of m, and $|M|$ is the number of total channels in the network.

Therefore, we have the definition of the weighted length of a single route.

Definition 4: For a route R, the length of R is:

$$L(R) = \sum_{ab \in R} L(ab), \tag{6}$$

where ab is any link on the route R.

For the source node S, after it receives the RREP from the destination node D, it will retrieve the information in the multiple RREP messages, and select one route to reach D. Suppose that the set of routes S can select from is $\mathcal{R}$. The algorithm for S to select a route from $\mathcal{R}$ is in Algorithm 1. It will choose the route with the minimum weighted length, based on the definition in the previous part.

V. Feasibility Improvement

A. Virtual Boundary Node

It is possible that in a sparse network, not all the boundaries of PUs are detected by nodes in CRNs. When boundaries are not detected, it could cause severe impacts to the weighted length calculations of single links and different routes. The route selected by Algorithm 1 would not be the one that passes the least number of PU areas. Therefore, we propose the "virtual boundary node" scheme, which takes a limited extra communication cost, to overcome the missing boundary detection problem.

Algorithm 1 Route selection from route set $\mathcal{R}$.

Input: $\mathcal{R}$, the route set;
Output: $Route$, the selected route;
$Length = \infty$, $Route = null$;
for $R \in \mathcal{R}$ **do**
  Calculate $L(R)$ using (6;) // Calculate the route length of every R
  if $L(R) < Len$ **then**
    $Route = R$;
    $Length = L(R)$;
return $Route$ // Return the route with minimum length.

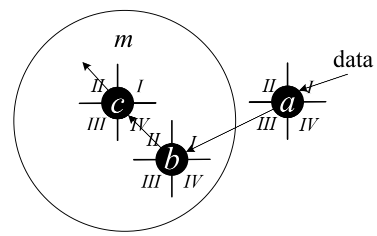


Fig. 4. An example of missing boundary detection for the PU area of m.

Fig. 4 is an example of missing boundary detections. Node a is outside the PU area of channel m, and node b is inside the PU area. Since there is no boundary node of link ab, and ab enters the PU area, the link bc after ab, which is inside the PU area, cannot be detected as being in the PU area of channel m.

To introduce our solution, we first define the virtual boundary node based on the links that pass across the PU boundaries:

Definition 5: For a link ab that crosses the PU boundary of channel m, node b is the virtual boundary node, which is the next hop node of a.

For a link ab, if it crosses a PU boundary, then there are two possibilities according to the direction of the link: either ab enters the PU area, or exits from the PU area. Based on the above virtual boundary node definition, if link ab enters the PU area, it means that a is outside the area and b is inside. If ab exits from the PU area, then a is inside the area and b is outside. In both cases, based on Definition 5, b is the virtual boundary node. For example, in Fig. 4, the virtual boundary node is b.

If node b is a virtual boundary node, although it is not a real boundary node, itself, it would piggyback the 4-tuple information to the source node, which is similar to a boundary node, instead of its 2-tuple information. Therefore, if node b is a virtual boundary node for the PU area of channel m, then we set $B_b(m) = 1$, and the corresponding value of $\mu_b(m)$ is similar to a real boundary node. Our scheme to overcome the missing boundary detection problem only requires the

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.2014.2364549, IEEE Transactions on Vehicular Technology

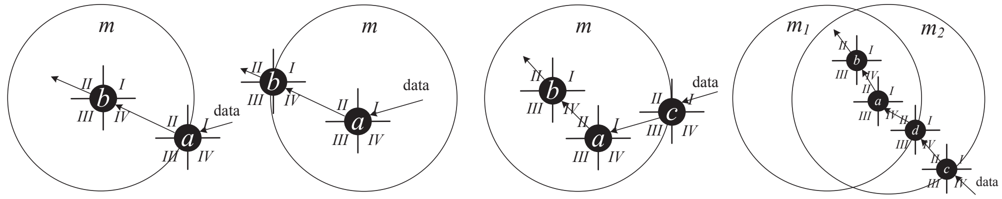


Fig. 3. Four cases of link ab located in one or multiple PU areas, detected by boundary nodes.

Algorithm 2 Virtual boundary settings for node b on link ab.

Input: M_a, M_b, the available channel sets of a and b; M, the total channel set;

Output: $B_b(m), \forall m \in M$; $\mu_b(m)$, the associate value when $B_b(m) = 1$;

1. **for** $m \in M$ **do**
2. **if** $m \in (M_a - M_b)$ **then**
3. $B_b(m) = 1$, $\mu_b(m) = 1$;
4. **else if** $m \in (M_b - M_a)$ **then**
5. $B_b(m) = 1$, $\mu_b(m) = -1$;
6. **else**
7. $B_b(m) = 0$;
8. **return** $B_b(m)(\forall m \in M)$, $\mu_b(m)$ if $B_b(m) = 1$;

information exchange of one-hop nodes. Given a link ab, node a sends its available channel set, M_a, to node b. Node b uses its own available channel set M_b and runs Algorithm 2, to decide if it is a virtual boundary node, itself. If it is, what the value of $\mu_b(m)$ is.

From Algorithm 2, if node b is a virtual boundary node, or $B_b(m) = 1$, then the value of $\mu_b(m)$ has the same meaning of a real boundary node. That is, if $\mu_b(m) = 1$, then link ab enters the PU area of channel m. If $\mu_b(m) = -1$, then link ab exits from the PU area of channel m. For example, in Fig. 4, since $m \in (M_a - M_b)$, then $\mu_b(m) = 1$, which means link ab enters the PU area. During the piggyback phase, the virtual boundary node will piggyback the 4-tuple information, similar to a real boundary node. For example, node b in Fig. 4 would piggyback $(I, II, m, 1)$.

Overall, our scheme indeed causes an extra communication cost to overcome the boundary missing detection problems. However, it only requires the available channel set exchanges among one-hop nodes. A node can decide if it is a virtual boundary node, itself, based on the available channel set of its previous one-hop node on the route. For example, nodes a and c in Fig. 4 do not need to exchange information, or know each other's available channel set. Our scheme, based on virtual boundary nodes, can be easily extended from the previous model, since virtual boundary nodes are treated the same as real boundary nodes and the route selection algorithm is unchanged.

B. Threshold-based Boundary Link

Given a link ab, it is possible that both end nodes are boundary nodes, and link ab itself is located at the boundary of a PU area. Under this situation, it is impractical to simply count link ab as inside or outside the PU area. Therefore, we propose a heuristic solution, which is threshold-based.

The threshold we use here is based on the distance of a and b. If the distance between a and b is above the threshold, then link ab is counted as a link inside the PU area. Otherwise, ab is treated as outside the PU area. This will affect the piggyback information of a and b, since they are both boundary nodes, as shown in the following two cases:

- If ab is treated as inside the PU area of channel m, node a would piggyback the 4-tuple information with $B_a(m) = 1$ and $\mu_a(m) = 1$, indicating the entering of this PU area. If the next-hop node of b on the route is inside the PU area, b would only return 2-tuple information, as a non-boundary node. Otherwise, b would piggyback the 4-tuple information with $B_b(m) = 1$ and $\mu_b(m) = -1$, indicating an exit from the PU area.
- If ab is treated as outside the PU area of channel m, node a would only return 2-tuple information, as a non-boundary node. The piggybacked information by b depends on the next-hop node of it on the route, which is similar to the previous case.

The remaining steps stay the same, e.g., we still calculate the link length based on Definition 3. One thing to notice is that, the threshold we use here can be changed under different bases, e.g., transmission power, or the distance to PUs, according to different requirements.

VI. Extensions

Our model assumes that the sensing results of different nodes, especially boundary nodes, are always correct. However, in real scenarios, it is possible that some sensing results are incorrect. This could cause the boundary nodes, or virtual boundary nodes, to claim the entering or exiting from PU areas incorrectly, which would affect the results of route length calculations and route selections.

Detection of these errors can be performed by the source node. Moreover, it requires the destination node D to piggyback its own sensing results. Since the piggybacked information contains the entering and exiting from PU areas of different channels, the source node can predict the PU areas that it locates at, itself. Then, based on the prediction results, the source node can compare with its own sensing results. If the two results do not match, it means there is error in the sensing results, either by the source node, or by other boundary and virtual boundary nodes on the route. The overview of the source node, S, detecting the errors on one route is:

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.2014.2364549, IEEE Transactions on Vehicular Technology

TABLE I
SIMULATION SETTINGS

Number of nodes	$[100, 300]$
Number of channels	$[10, 25]$
Number of sectors	4
TX power	23 dBm
Noise power	-98 dBm
SINR threshold	10 dB
Number of PUs	$[10, 50]$
Operation range of each PU	$[300, 500]$
Delay for single channel switch	0.1 s

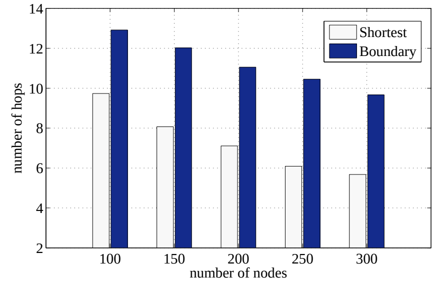


Fig. 5. Number of Hops

1) For each channel $m \in M$, the source node S collects the piggybacked information, which includes the sensing results of destination node D, and sums up all the $u_a(m)$, if $B_a(m) = 1$, $\forall a$ on the route;
2) Source node S maintains two lists, L_{in} and L_{out}, to store the channels whose sums from Step (1) are not 0. If the sum of a channel is greater than 0, the channel is stored in L_{in}. If the sum is negative, the channel is stored in L_{out};
3) S compares the two lists, L_{in} and L_{out}, with its own sensing results. For every channel $m \in L_{in}$, if destination node D's sensing results show that m is available. Then, there is an error on the sensing results of m, either by S, or boundary nodes, including virtual boundary nodes, on the route. For every channel $m \in L_{out}$, if node S's sensing results show that m is available, similarly, there is also an error.

For a boundary node a with $B_a(m) = 1$, if $u_a(m) = 1$, it means the route enters a PU area of channel m. If $u_a(m) = -1$, it means the route exits from the PU area. If channel m is in L_{in}, it means that the sum of $u_a(m)$, $\forall a$ on the route with $B_a(m) = 1$, is greater than 0. It indicates that the route enters more PU areas than it exits from the PU areas of channel m. Therefore, the destination D should be in the PU area of channel m. Then the source node S compares with the sensing results of D. The sensing results of S and D are long term, similar to the boundary nodes and virtual boundary nodes, which indicate whether S and D are inside or outside the PU areas. If the sensing results of D show that m is always available, which means D is outside the PU area, then the contradictory matching results point out the errors of the sensing results, either by the boundary nodes or the destination node. If a channel m is in L_{out}, it means that the route enters less PU areas than it exits from the areas of channel m. Similarly, S should be in the PU area of channel m. If S's sensing results show the different results, an error is detected.

VII. Performance Evaluations

We randomly distribute nodes in a $2,000 \times 2,000$ unit square. Each node has 4 sectors to send and receive data. We generate a number of PUs, which are randomly active on a certain channel. The operation range of each PU is different. The number of nodes is more than the number of PUs, and this ensures that the boundary of each PU is detected. We randomly choose a source and a destination. Then, using our approach, the source establishes a route to reach the destination. The channel availabilities are dynamic during the data transmission, because the PUs are set at a predefined probability to be active on a channel. The settings of our simulation parameters are shown in Table I. For simplicity, we set the channel switch delay to be constant.

The three parameters, the number of nodes, number of channels, and number of PUs, are tunable. We compare our model with the shortest path algorithm, which is to find the shortest path from the source to the destination, without considering the channel availabilities. We evaluate the performance of both models from the following aspects:

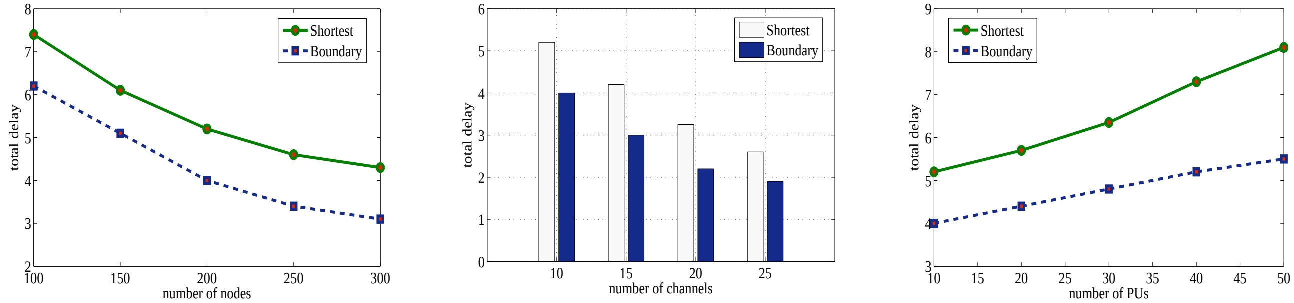
- Number of hops: simply count the number of links for each route, without considering other factors, e.g., the channel availabilities.
- Total delay: the overall delay considering both channel switching delay and data transmission delay of each session.
- Average route length: calculate the average route length in both models using Definition 4.

1) Number of hops: we set the total number of PUs to 10, and the total number of channels to 10. The number of nodes varies from 100 to 300. We calculate the number of hops of each route under both models. The results are shown in Fig. 5. The line labeled as "Shortest" is the result from using the shortest path algorithm, without considering the channel availabilities. The line labeled as "Boundary" is the result from our model. Obviously, the shortest path algorithm has a lower number of hops than does our algorithm. Moreover, in both models, the average number of hops reduces as the number of nodes increases. This is because the connectivity of the network increases when the number of nodes increases.

2) Total delay: we then study the influence of the three network parameters; the results are shown in Fig. 6. In Fig. 6(a), our model achieves about 1.0s less than the model using the shortest algorithm. In Fig. 6(b), the total delay of both models decreases as the total number of channels increases. Under this setting, our model achieves about 20% less total delay than those using the shortest algorithm. In Fig. 6(c), the model's total delay increases when the total number of PUs increases. Our model achieves about 20% less total delay than those using the shortest algorithm.

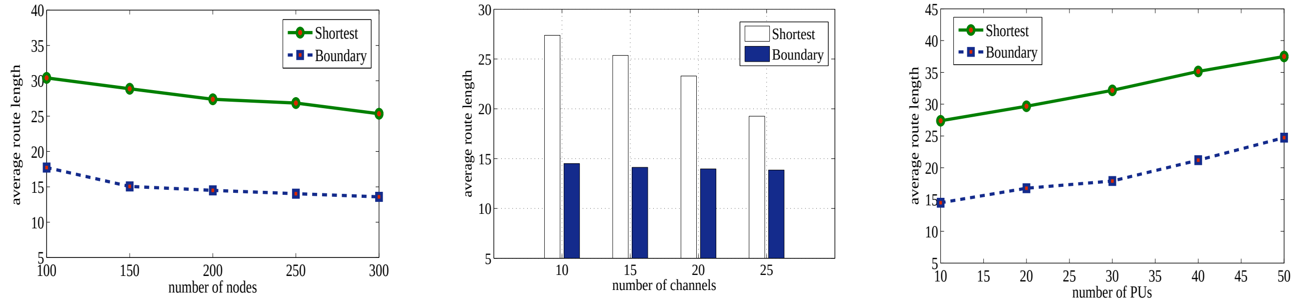
3) Average route length: in Fig. 7(a), the average route length decreases in both models. This is because the number

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.2014.2364549, IEEE Transactions on Vehicular Technology



(a) Varying # of nodes (b) Varying # channels (c) Varying # PUs

Fig. 6. Comparison of the total delay by varying network parameters.



(a) Varying # of nodes (b) Varying # channels (c) Varying # PUs

Fig. 7. Comparison of the average route length by varying network parameters.

of hops in both models decreases when the number of nodes increases. The average route length in our model is about 40% less than the shortest path algorithm. In Fig. 7(b), the average route length using the shortest algorithm is 30% more than in our model. Our model decreases more slightly because it is already close to the minimum value, which equals the number of hops. In Fig. 7(c), the average route length increases in both models when the number of PUs increases. This is because, when there are more PUs, more links are within the PU area. In addition, the average route length in our model is about 40% less than the shortest algorithm.

VIII. Conclusion

In this paper, we propose an efficient model for routing in CRNs, which makes use of boundary nodes. The boundary nodes help to estimate the channel situation of each optional route, and also the number of links located within a PU area on each route. Nodes on each route piggyback the channel availability and path information. We give a novel definition of the route path, and propose an effective algorithm for route selection. An extension of error detection is proposed. We perform numerous simulations to testify the performance of our model.

Acknowledgments

This research was supported in part by NSF grants ECCS 1231461, ECCS 1128209, CNS 1138963, CNS 1065444, and CCF 1028167.

References

[1] I. F. Akyildiz, W.-Y. Lee, and K. R. Chowdhury, "CRAHNs: Cognitive radio ad hoc networks," *Ad Hoc Networks*, vol. 7, pp. 810–836, 2009.

[2] M. Cesana, F. Cuomo, and E. Ekici, "Routing in cognitive radio networks: Challenges and solutions," *Ad Hoc Networks*, 2010.

[3] F. Dai and J. Wu, "Efficient broadcasting in ad hoc wireless networks using directional antennas," *IEEE Transactions on Parallel and Distributed Systems*, 2006.

[4] Y. Liu, L. Cai, and X. Shen, "Spectrum-aware opportunistic routing in multi-hop cognitive radio networks," *IEEE Journal on Selected Areas in Communications*, 2012.

[5] K. R. Chowdhury and I. F. Akyildiz, "CRP: A routing protocol for cognitive radio ad hoc networks," *IEEE Journal on Selected Areas in Communications*, 2011.

[6] G. Zhao, J. Ma, G. Y. Li, T. Wu, Y. Kwon, A. Soong, and C. Yang, "Spatial spectrum holes for cognitive radio with relay-assisted directional transmission," *IEEE Transactions on Wireless Communications*, 2009.

[7] D. Wilcox, E. Tsakalaki, A. Kortun, T. Ratnarajah, C. Papadias, and M. Sellathurai, "On spatial domain cognitive radio using single-radio parasitic antenna arrays," *IEEE Journal on Selected Areas in Communications*, 2013.

[8] W. Guo and X. Huang, "Multicast communications in cognitive radio networks using directional antennas," *Wireless Communications and Mobile Computing*, 2012.

[9] W. Guo, X. Huang, and K. Zhang, "Joint optimization of antenna orientation and spectrum allocation for cognitive radio networks," in *the Forty-Third Asilomar Conference on Signals, Systems and Computers*, 2009.

[10] Y. Dai, J. Wu, and C. Xin, "Virtual backbone construction for cognitive radio networks without common control channel," in *Proceedings of IEEE INFOCOM*, 2013.

[11] R. Vilzmann and C. Bettstetter, "A survey on mac protocols for ad hoc networks with directional antennas," in *EUNICE 2005: Networks and Applications Towards a Ubiquitously Connected World*, 2006.

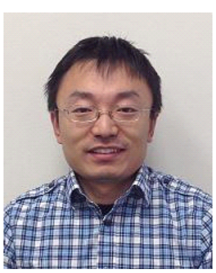
[12] M. Lu, F. Li, and J. Wu, "Efficient opportunistic routing in utility-based ad hoc networks," *IEEE Transactions on Reliability*, 2009.

[13] Y. Zhang, N. Meratnia, and P. Havinga, "Outlier detection techniques for wireless sensor networks: A survey," *IEEE Communications Surveys & Tutorials*, 2010.

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.2014.2364549, IEEE Transactions on Vehicular Technology

Ying Dai received her B.Eng. in Software Engineering from Nanjing University, Nanjing, China, in 2010. She is currently a Ph.D. candidate in the Department of Computer and Information Sciences, Temple University, USA. Her research interests include various topics in cognitive radio networks. She is now working on spectrum sensing, efficient channel assignment, and data transmission in cognitive radio networks. Her publications include papers in TPDS, IEEE Infocom, MASS, and ICCCN.

Jie Wu is the chair and a Laura H. Carnell Professor in the Department of Computer and Information Sciences at Temple University. Prior to joining Temple University, he was a program director at the National Science Foundation and Distinguished Professor at Florida Atlantic University. His current research interests include mo- bile computing and wireless networks, routing protocols, cloud and green computing, network trust and security, and social network applications. Dr. Wu regularly publishes in scholarly journals, conference proceedings, and books. He serves on several editorial boards, including IEEE Transactions on Computers, IEEE Transactions on Service Computing, and Journal of Parallel and Distributed Computing. Dr. Wu was general co-chair/chair for IEEE MASS 2006, IEEE IPDPS 2008 and IEEE ICDCS 2013, as well as program co-chair for IEEE INFOCOM 2011 and CCF CNCC 2013. Currently, he is serving as general chair for ACM MobiHoc 2014. He was an IEEE Computer Society Distinguished Visitor, ACM Distinguished Speaker, and chair for the IEEE Technical Committee on Distributed Processing (TCDP). Dr. Wu is a CCF Distinguished Speaker and a Fellow of the IEEE. He is the recipient of the 2011 China Computer Federation (CCF) Overseas Outstanding Achievement Award.



Yaxiong Zhao is a software engineer at Google. He received the BE degree from Tongji University, Shanghai, China, in 2005, and the ME degree from Tsinghua University, Beijing, China, in 2008. He received his Ph.D. in 2012 from the Computer and Information Sciences Department at Temple University. His research interests include distributed computing systems and computer networks. He has published many first-author papers on wireless sensor networks, delay tolerant networks, and content-based networks. He is a member of the IEEE.