Distributed Rerouting For Multiple Sessions in Cognitive Radio Networks

Ying Dai and Jie Wu
Department of Computer and Information Sciences
Temple University, Philadelphia, PA 19122
Email: {ying.dai, jiewu}@temple.edu

***Abstract*—This paper proposes a distributed rerouting scheme in cognitive radio networks (CRNs). Specifically, we consider how we can reroute for multiple sessions in a situation where switching channels cannot ensure end-to-end connectivity. We propose a ring searching-based framework for rerouting, which is distributed and considers both the channel availabilities and potential rerouting choices of other sessions. Our framework can be applied for the rerouting of multiple sessions in CRNs and does not require any communication among them. The objective in our model is to minimize the queueing delay for each session that needs rerouting. We propose a novel way to estimate the delay for each session. Each node, in our model, that will be receiving RREQ (Route Request) messages maintains two queues: one queue on each node contains a list of nodes whose RREQ is received, and the other one contains the probabilities of it being chosen. We apply an opportunistic overhear approach to predict the probabilities. Also, we take the channel availabilities into account when constructing the rerouting path. The simulation results prove the high performance of our scheme. We also show the influence of different parameter settings, including both network environment parameters and scheme parameters.**

***Index Terms*—Cognitive radio networks, distributed rerouting, multiple sessions, dynamic network environment.**

I. Introduction

Routing in cognitive radio networks (CRNs) [1] is more challenging than in conventional networks for two reasons. Firstly, instead of transmitting on a single channel, the cognitive radio technology enables nodes in CRNs to dynamically switch among different channels for transmission. This results in a different interference model and brings about more complexity in the estimation of throughput or delay. Secondly, routing problems in CRNs are also very different from those in multiple channel networks. The channel assignment and usage in multiple channel networks is relatively static. However, due to the presence of primary users, who have privileges for using channels [2], the channel availabilities in CRNs are dynamic as well as unpredictable. Therefore, the routing problems in CRNs are different from in other wireless networks and are more challenging.

Due to the dynamics of channel availabilities, it is almost impossible to find a relatively static or stable route for a session with delay or throughput constraints [3]. This is because, when one primary user comes up, nodes occupying that channel must quit in order to not interfere with it. To maintain the transmission, the affected nodes may find another available channel to switch to. When these affected nodes are unable to find any other appropriate channel, some links will be broken, which results in the failure of the original routes. To maintain the transmissions between source-destination pairs, the affected nodes need to reroute and find another route to replace the broken link. Therefore, rerouting is also the key point in designing routing protocols for CRNs.

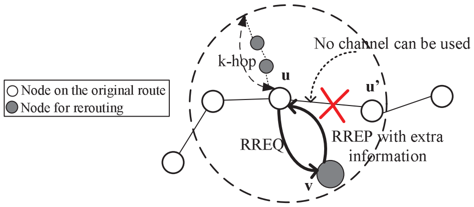


Fig. 1. An example of rerouting in CRNs.

However, designing rerouting protocols under delay constraints in CRNs is difficult. Firstly, the possible choices of other sessions should be considered. It brings complexity to the delay estimation. Secondly, due to the unpredictability of primary users, the total number of currently available channels is undetermined. Thirdly, when multiple sessions reroute together in a close area, for a single node, its nearby nodes are also candidates that may be chosen for rerouting. Therefore, its surrounding network environment cannot be predicted because the channel usage of nearby nodes is unable to be determined in advance.

Considering the difficulties listed above, we propose a general and distributed framework for finding rerouting paths. The general framework is based on ring searching methodology that aims at minimizing the delay of rerouting paths. Unlike previous rerouting protocols in CRNs, our approach makes practical predictions about the delay and does not require any communication among nodes that need rerouting. Both the determined and undetermined factors are taken into account. We make use of the RREQ (Route Request) / RREP (Route Reply) protocol. Like in Fig. 1, when a node u needs rerouting, it would broadcast a RREQ message to nearby nodes within k hops and find different possible routes to replace the broken link from u to u'. Suppose that u's nearby node, v, who receives the RREQ message, would reply with the RREP. Moreover, node v would incorporate the current channel information, estimated delay, along with some other information, to the RREP and send it back to u. The estimated delay here refers to the estimated time duration from when node v receives data from u to when it is sent out to v's next hop. Here, v is chosen by u for rerouting. We will explain how we estimate the delay later. From the point view of u, after receiving the RREP, it would make decisions about whether or not to choose v for rerouting. If no appropriate path can be found within k hops, our approach would increase the searching range, which means increasing k, and repeat the above process until the rerouting path is found.

The problem left is the estimation of the delay. In our framework, we have each node maintain two queues. Each queue contains different information about nodes that are confirmed to use it to construct rerouting paths, and also information about nodes that may possibly use it. Our estimation of the queueing delay also considers the potential choices of other nodes in the queue. Since it is possible that multiple nodes may need rerouting at the same time, the conflicts among them are inevitable. We utilize an approach to resolve the conflicts, which is reserve-based.

We make the three following contributions:

- We propose a general and distributed rerouting framework for CRNs, which is ring searching-based, and it aims at minimizing the delay for multiple sessions. It does not require any communication among multiple sessions and is adaptable to a dynamic network environment.
- We present a novel technique for estimating the delay. Each node constructs the rerouting path based on both the delay estimation and current channel environments.
- We provide a reserve-based approach to resolve conflicts among multiple nodes that need rerouting together in a close area.

The organization of our paper is as follows. In Section II, the related work is presented. The problem formulation is in Section III. Section IV describes our distributed rerouting framework. The conflict resolution is proposed in Section V. Section VI shows our simulation settings and results. Section VII concludes this paper.

II. RELATED WORK

In this section, we introduce the related work from two aspects of routing in CRNs. One is about the approaches that target the route maintenance or route recovery. Another is about research that aims at reducing the delay in CRNs.

A lot of research has been done on the route maintenance in CRNs [4]–[11], which aims at providing a qualified route, despite the appearance of primary users. In [4], Feng et al. propose a handoff scheduling and rerouting scheme. Their protocol assumes the predictability of the primary users and has each flow ready to reroute before the primary users appear. The delays are not considered. In [5], Abbagnale et al. focus on providing a route with more stability. The main idea is to assign weights to routes based the algebraic connectivity. A metric is proposed that can capture path stability and availability over time. Their approach is based on the historical average activity model of primary users. The concept of route maintenance cost is proposed in [11]. The cost represents the effort to maintain the end-to-end connectivity in CRNs. Their approach starts to obtain an optimal path with minimum route maintenance cost under the perfect knowledge of primary users. However, their approach defines the cost in a simple way and is based on the ideal knowledge of primary users.

Moreover, there is a lot of research about routing in CRNs that considers minimizing the delay [12]–[14]. Two kinds of delays are analyzed: switching delay and queueing delay. The switching delay is proposed in [12], [13], which considers the switching delay between different channels and the backoff delay (medium access delay) with a given channel. Each node maintains a metric of the cumulative delay along a candidate route. Their approaches do not consider cases where rerouting would be needed, which would normally be used when there is no channel for a certain link. In [14], Yang et al. also propose a distributed method for local coordination. Their on-demand protocol is a variation of AODV. However, their approach is not well scalable due to lots of information exchanges. The coordination consumes lots of energy, and the potential delay caused by other flows is not considered.

III. PROBLEM FORMULATION

In this section, we first describe the problem. Then we define our system model by giving the assumptions, constraints, and objective.

A. Problem Statement

We consider a network with multiple sessions S. Suppose that there are M channels and N nodes. Each node $u \in N$ has its own set of available channels M_u, where $M_u \subset M$. M_u is dynamic due to the presence of primary users. The possible channel that can be used for a certain link (u, u') must belong to the set, $M_u \cap M_{u'}$. That is, the sender u and the receiver u' must tune in to the same channel in order for a successful transmission. When a primary user becomes active, meaning they are using the same channel that is currently occupied by (u, u'), then (u, u') must switch to another available channel. However, when there is no appropriate channel in set $M_u \cap M_{u'}$, (u, u') would be broken. In this situation, u needs to seek another route to reach u'. This means that u would reroute to replace the broken link (u, u'). Let N' denote a set that consists of nodes that need rerouting at a given time and whose next hop links are broken and cannot be recovered by switching channels.

Since nodes in N' face the rerouting needs together at similar times, the possible choices of them are unpredictable. Our model solves the rerouting problems for them, which minimizes the delay, and does not need any communication among the nodes in N'. The main notations used in our paper are listed in Table I.

B. System Model

Firstly, we assume that the active time period of each session is a constant T, which means that each node is occupied by a certain session for time period T. T can surely be different for different sessions. This assumption is for making

TABLE I
LIST OF NOTATIONS

Notation	Meaning
N/N'	set of nodes (that need rerouting)
$u/u_1/u_2/u_3$	node needs rerouting
$v/v_1/v_2$	node used for constructing rerouting path
M/M_u	set of total available channels (for u)
T	active time period of each session
T_0/T_1	time period for CTR (overhearing RREP)
S_v	set of sessions that are transmitting through v
a_v^m	indicator of whether channel m is used by v
k/k'	the ring searching initial (increasing) hop count
QN/QN_v	queue of node's IDs (maintained on v)
QN_v^u	u's ID maintained on v
QP/QP_v	queue of probabilities (maintained on v)
QP_v^u	estimated probability maintained on v for u
p_v^u	value of QP_v^u before receiving CTR
QL_u	queue length on node u
D^u/D_v^u	estimated delay (on v) for node u
$I(u)$	index of u in QN
$G(u,u')$	a weighted graph for rerouting from u to u'
w_v^u	weight on v assigned by u

the description of our approaches clearer. We also assume that there is a Common Control Channel (CCC) for all nodes, which are secondary users. For each node, we assume that they transmit at a constant power. Suppose that the activities of both primary users and secondary users are very active. Therefore, it is very possible that some links may be broken at a relatively close time period, and multiple nodes may need rerouting within a close area.

For a single node u, at a certain time, let S_u denote the set of sessions transmitting through u. u can only serve one session at a given time. a_u^m $(m \in M_u)$ denotes if the channel is currently being used by u. Then we have:

$$0 \leq |S_u| \leq 1$$

$$0 \leq a_u^m \leq 1.$$

For node u to transmit successfully, the SINR on its receiver should be above a threshold β. Therefore, for a link (u, v) using channel m, we have:

$$SINR_{u,v}^m > \beta, m \in M_u \cap M_v.$$

Suppose that, for node u, there are several rerouting paths to its original next hop node. Subject to the above constraints, the objective here is to find routes that minimize the expected delay for all rerouting nodes, which can be written as:

$$\text{Min} \sum_{u \in N'} D^u.$$

In fact, the delay should consist of queueing delay and switching delay. However, we only consider queueing delay here, as we explained before.

There are two main challenges in our problem. The first is that the influence among nodes in set N' on each other is unpredictable. For a node u, when it makes a choice regarding rerouting paths, it needs to consider the queueing delay. However, since other nodes' decisions in N' cannot be known in advance, which it is very possible to use some of the same nodes in their rerouting paths as u, u is unable to know the exact queueing delay when it chooses the rerouting path. Secondly, the channel situation is dynamic. This can be caused by both primary users and other nodes in N'. Given the dynamic channel situation and the SINR constraints, paths with a higher probability of transmitting successfully should be more likely to be chosen. Therefore, the channel availabilities of nodes on the rerouting path need to be considered as well.

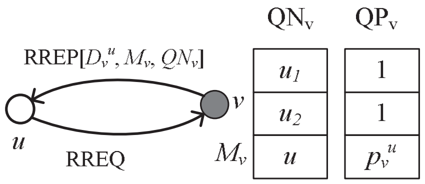


Fig. 2. Two queues maintained by node v.

IV. DISTRIBUTED REROUTING FRAMEWORK

In this section, we first describe the distributed rerouting framework. Then, we describe how to estimate the delay and how to choose nodes for rerouting.

A. *Ring Searching-based Framework*

For a single node that needs rerouting, we apply the ring search method. Initially, it searches the path within k hops. If there is no appropriate path in k hops, then it increases k to $k + k'$ and searches again. Under a specific setting of k, the searching process contains three main phases:

1) For any node $u \in N'$, it sends out the RREQ (Route Requests) message, which contains both the sender's ID and the ID of the rerouting destination;
2) Nodes within u's k-hop distance estimate the delay and reply with the RREP (Route Reply), which is incorporated with the estimated delay and channel availability;
3) u collects all of the RREP messages, makes a decision about which node to use for rerouting, and sends back a CTR(Confirmation To Route) to each chosen node;
4) If a new route is successfully built for u, quit; otherwise, $k = k + k'$ and repeat the above phases.

Our framework makes use of the RREP messages, which contain not only the traditional route reply message, but also the estimated delay for each sender and the current channel environment. The RREQ sender is able to retrieve the useful information in each RREP to construct the rerouting path. If the node cannot find an appropriate path within k hops, it will extend its searching range to $k + k'$. The values of k and k' depend on the network environment. The ring searching increment parameter k' is an important factor for determining the performance of our distributed rerouting scheme. We discuss the influence of its different values in our simulation.

Next, in Steps (2) and (3), the remaining problems lie in estimating the delay and choosing among different nodes to

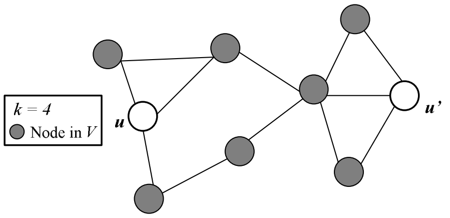


Fig. 3. u: source of rerouting path; u': destination of rerouting path.

Algorithm 1 Delay Estimation and Building RREP

```
while $v$ receives RREQ do
    $u$ = ID of the RREQ sender
    Insert $u$ to $QN_v$
    $v$ overhears RREP messages sent to $u$ within $T_1$
    Calculate $p_v^u$
    Insert $QP_v^u = p_v^u$ to $QP_v$
    Calculate $D_v^u$
    Send back RREP to $u$ with $D_v^u$, $M_v$, and $QL_v$
    if CTR receives within $T_0$ then
        Update $QP_v^u = 1$
    else
        Remove $QN_v^u$ and $QP_v^u$
```

construct the path. We solve the two problems in the following parts.

B. Delay Estimation

When multiple nodes in N' need to reroute at a relatively similar time, it is difficult to estimate the delay for them without knowing each ones' decisions in advance. We propose a novel way for delay estimation, which is distributed and does not require any communication among these nodes in N'.

Each node receiving the RREQ maintains two queues: QN and QP. QN is a queue of confirmed nodes that use it to transmit and nodes that may possibly use it. QP maintains the probabilities of nodes in QN choosing this node for rerouting.

For any node $v \in N$, every time it receives a RREQ from a certain node u, v would put the ID of u in v's QN, as shown in Fig. 2. For values in QP, there are two possibilities. One is for confirmed nodes that use v, and the other is for nodes that may possibly use v. Suppose that $QP_v^u \in QP_v$ is maintained by v and $u \in QN_v$:

$$QP_v^u = \begin{cases} 1 & v \text{ receives CTR from } u \\ p_v^u & v \text{ has not received CTR from } u, 0 < p_u < 1 \end{cases}.$$

Node v has a valid time period T_0 for the CTR, the value of which is pre-defined. If v does not receive the CTR from u within T_0, u and p_v^u would be removed from QN_v and QP_v.

In Fig. 2, nodes u_1 and u_2 are confirmed to use node v since $QP_v^{u_1}$ and $QP_v^{u_2}$ are both equal to 1. Node u may possibly use v because QP_v^u equals p_v^u. The value of p_u is estimated by overhearing other nodes' RREP messages, which are also sent to u. Besides the estimated delay, each node would also put its current queue length and its current channel availability in their RREP. Before v sends back the RREP, it would listen for the RREP messages of other nodes (v') during the time period of T_1. Then we have:

$$p_v^u = (1 - \frac{QL_v}{\sum QL_{v'}}) \times \frac{|M_v|}{\sum |M_{v'}|},$$

where QL_v is the queue length currently maintained by v, $|M_v|$ is the number of currently available channels on v, and v' belongs to the set of nodes that send a RREP to u, which are overheard by v within T_1, plus node v itself.

After QP_v^u is calculated, v would put QP_v^u in QP_v. Therefore, the index of QP_v^u in QP_v is equal to the index of u in QN_v. By having QN_v and QP_v, node v can make the estimation of the delay for u, D_v^u:

$$D_v^u = \sum_{u_1 \in QN_v \& I(u_1) < I(u)} (QP_v^{u_1} \times T),$$

where $I(u_1)$ is the index of u_1 in QN_v. Since $I(u_1) < I(u)$, u_1 denotes every node in QN_v that is also before u in QN_v. After D_v^u is calculated, v will send back the RREP[D_v^u, M_v,QL_v] to u. In this way, the estimated delay and the channel availability are sent back to u.

After sending the RREP, if v receives the CTR from u within T_0, v will update the probability of u in their queue QP_v to be 1. Otherwise, u is removed from both queues.

The algorithm for a certain node v that receives a RREQ is shown in Algorithm 1. Next, we give one theorem to prove the advantages of our algorithm. Another analysis will be given in the following subsection.

Theorem 1. *For two nodes, v_1 and v_2, with the same queue situation, the one that is more stable when facing the unpredictable appearance of primary users has the lower estimated delay.*

Proof. Suppose that the available channel set of v_1 and v_2 is M_{v_1} and M_{v_2}. Suppose they are both possible to be chosen by node u. When primary users become active, some channels would become unavailable. Then nodes with more available channels have a lower probability of having no channel to use. Therefore, the node with more available channels is more stable. Assume that $|M_{v_1}| > |M_{v_2}|$. Then $QP_{v_1}^u > QP_{v_2}^u$. Therefore, under equal conditions, $D_{v_1}^u > D_{v_2}^u$. ■

C. Path Construction

For nodes in set N', after collecting the RREP, they would make decisions about which nodes to use for the rerouting path construction.

Firstly, nodes in set N' would construct a graph $G(u, u') = (V, E, W)$ to find the rerouting path. Fig. 3 is an example. $u \to u'$ is the broken link in the original route, as shown in Fig. 1. $V \subset N$ is the set of nodes that can be used for constructing a new path to replace $u \to u'$. Nodes in V must have a set of available channels for transmission whose SINR is above the threshold. E is the set of links among the nodes in V. W is the weight that u assigns to each node according to the RREP it collects. The weight of $v \in V$ is:

$$w_v^u = \frac{D_v^u}{|M_v|}.$$

The node with more available channels and a smaller estimated delay would have less weight. After the graph is built, u would construct a route to u' with the lowest weight. This can be solved via the greedy algorithm. Then, u constructs the path for rerouting and sends a CTR back to each chosen node.

Now, we prove that the probability of two nodes choosing the same node for rerouting is low.

Theorem 2. *For two nodes, u_1 and $u_2 \in N'$, the probability of them choosing the same node v for rerouting is lower than the random choice scheme.*

Proof. Suppose that the N_{u_1} is the set, that u_1 receives RREP messages from, of nodes with the same hop distance as v, while N_{u_2} for u_2. When using the random choices, the probability that both u_1 and u_2 choose v is:

$$P_1 = \frac{1}{|N_{u_1}|} \times \frac{1}{|N_{u_2}|}.$$

In our scheme, if both u_1 and u_2 choose v, then w_v must be the minimum for both u_1 and u_2, which is:

$$P_2 = p(w_v^{u_1} < w_{v_1}^{u_1}) \times p(w_v^{u_2} < w_{v_2}^{u_2}),$$
$$\forall v_1 \in N_{u_1}, v_2 \in N_{u_1},$$

where $w_v^{u_1}$ is the weight of v, calculated by u_1, and $w_v^{u_2}$ is calculated by u_2. Since the information about other nodes that are the same hop distance away as v to u_1 and u_2 is unknown, the total number of possible situations is greater than $|N_{u_1}|$ and $|N_{u_2}|$. Therefore, $P_1 > P_2$. ■

Therefore, when building rerouting paths for multiple nodes together, the union of their chosen nodes is low, which means that the distance among them is relatively far. In this way, the interference among them is reduced. Also, the channel availability determines the probability of a node to be chosen for rerouting. It increases the reliability of the new route when primary users show up.

V. CONFLICT RESOLUTIONS

When multiple nodes need rerouting together in a close area, there will be conflicts among them. Each node that receives multiple RREP messages needs to determine the positions of them in both of its queues and how it can update their positions based on the received CTR messages. We propose our approach to resolve conflicts. In this section, we first describe the conflicts in our model and then present our conflict resolution approach.

A. Conflict Definition

Node v, that receives multiple RREP messages, first puts them into its QN_v and calculates the probabilities to put them into QP_v, as was stated in the previous section. The sequence of putting them into queues follows the first-come-first-serve method. However, the arriving sequence of the CTR may be different, which does not follow the order in the queues.

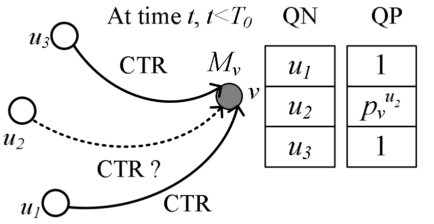


Fig. 4. Conflicts among multiple nodes.

As shown in Fig. 4, in the time period of T_0, when the CTR of node u_3 arrives and no CTR has been received from u_2, v needs to decide the order of u_2 and u_3. The RTS of u_2 arrives before u_3. Therefore, u_2 is supposed to be served before u_3. However, the CTR of u_3 arrives, and u_2 has a probability of not choosing v, which means that the CTR of u_2 may not be received. Then, it would be a waste of time to wait for the CTR of u_2 since u_3 is unable to use v for transmission during the waiting period. Thus, we provide three approaches for dealing with this situation.

B. Reserve-based Conflict Resolution Approach

The reserve-based scheme involves reserving the initial positions of each node by regarding the arrival sequence of the CTR. This means that the positions of the queue are stable. The only situation that will change the positions is when no CTR is received when the valid time period T_0 expires.

VI. SIMULATION

In this section, we introduce our simulation. We show the performance of our scheme compared to two other rerouting schemes for CRNs. Also, the influences of the parameters are studied. Moreover, we compare the three conflict resolutions under different settings.

A. Simulation Settings

We randomly distribute nodes in a 100×100 unit square. At a certain time slot, some active primary users are generated and occupy some previously available channels. Some affected nodes can maintain their transmission through switching channels. Some have to find another path to replace the affected links. Our simulation is based on these nodes that have to reroute. There are two types of parameters in our settings:

- Network environment parameters: the total number of nodes, the total number of channels, and the number of primary users at each time they become active.
- Scheme parameter: k' in the ring searching framework.

The network environment parameters determine the network density, the channel availabilities, and the interference to some extent. The scheme parameter k' is related to our rerouting scheme. The network settings, including both variable parameters and invariable parameters, are shown in Table II.

For a better comparison, we also implement a random rerouting scheme. The random scheme still uses the ring searching-based framework without any preference on selecting nodes for rerouting. Each node that needs rerouting in

TABLE II
SIMULATION SETTINGS

number of nodes	$[100, 300]$
number of available channels	$[10, 30]$
number of primary users	$[30, 90]$
operation range of primary users	$[30, 40]$
time duration for each session$T$	10 slots
hop count k for ring searching base	1
hop count k' for ring searching increment	$[1, 3]$
time duration for overhearing RREP T_1	2 slots
waiting time for CTR T_0	3 slots
TX power	23 dBm
noise power	-98 dBm
SINR threshold	10 dBm

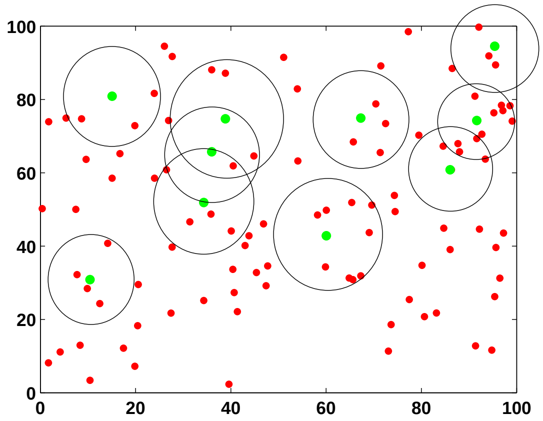


Fig. 5. Rerouting example.

the random scheme would randomly choose a route without considering the queueing delay, as long as each node on the route has an available channel to transmit. In addition, an optimal rerouting scheme, under the ring searching framework, is implemented, which constructs the rerouting path with the minimum queueing delay through searching every possible choice. It keeps increasing the ring range until the achieved delay does not reduce. Then, the achieved minimum delay is considered as the optimal result.

We compare our scheme with the above two schemes, and we also analyze the influences of applying different settings to our scheme parameters on the following performance metrics:

- D: the sum of the delay for all nodes that need rerouting;
- H: the average hop count of all rerouting paths;
- R: the average number of rings, which is equal to the average times of increasing searching range.

We choose one set of parameters and show the demands of our scheme in Fig. 5. The green nodes are those that need rerouting. The circles are their one-hop base searching range, with $k' = 0$. We find that, even if each node searches within their initial range, there are intersections among different searching areas. Nodes in the intersections may possibly to be chosen by several rerouting paths at the same time. Therefore, the queueing delay on these nodes cannot be ignored; this proves the importance of our distributed rerouting scheme.

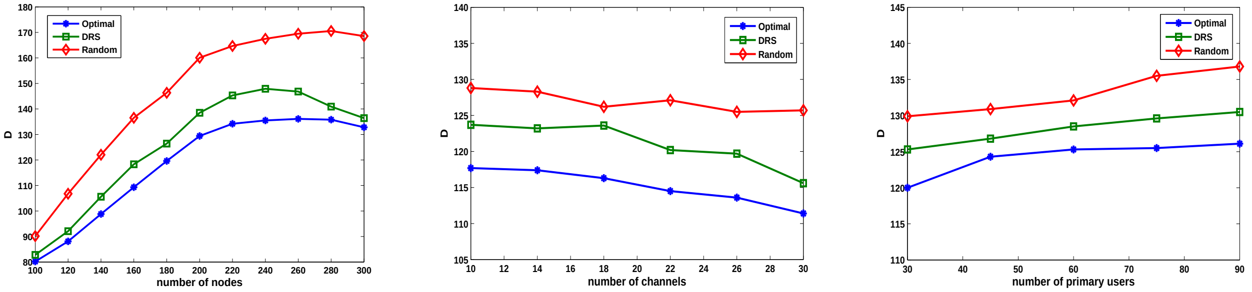
B. Simulation Results

We show our simulation results in this part from two perspectives. The first is the comparison among three rerouting schemes, which are our rerouting scheme, the random scheme, and the optimal scheme. The second is the comparison of the performance under different scheme parameters.

1) Comparison among three schemes: we first compare the D among the three schemes. Fig. 6(a) is the delay comparison after varying the number of nodes. Our distributed rerouting scheme (DRS) is better than the random scheme and achieves more than 80% of the optimal results. When the number of nodes becomes more than 260, the performance becomes better. This is because having more nodes provides more choices. However, before that, the increasing amount of nodes caused more conflicts under the same number of sessions. The delay comparison, through varying the number of channels and the number of primary users, is shown in Fig. 6(b) and Fig. 6(c). The results in both figures show that our scheme is better than the random scheme and reaches more than 90% of the optimal results.

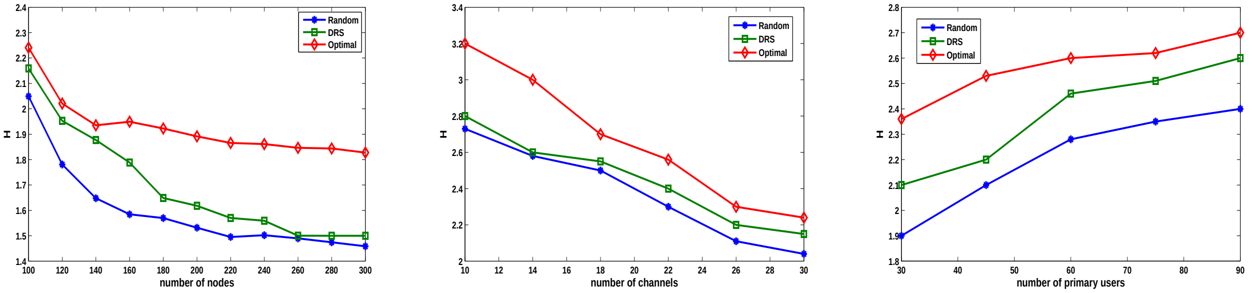
We compare the H among the three schemes under different network parameters. In Fig. 7(a), the hop count is compared by varying the number of nodes. The number of channels is set to 30, and the number of primary users is 30. This shows that the hop count is below 3 on average. When the number of nodes increases, the length of the rerouting path decreases in general. The optimal scheme has the largest hop count on average, and the random scheme results in the lowest hop count. Our scheme is still between the other two. Fig. 7(b) and Fig. 7(c) show the comparison of hop counts through varying the number of channels and the number of primary users, respectively. When the channel availability lowers, the average path length becomes longer. Both figures show that the hop count of our scheme is more than the random scheme but less than the optimal results. Combined with Fig. 6, our scheme needs a much lower number of hop counts while the achieved results are close to the optimal scheme.

In addition, we compare the average number of rings that each scheme uses. The k' is set to 2. The results are shown in Fig. 8. The number of ring searching for the three schemes has a similar trend to what we saw with the number of hop counts, as shown above. In Fig. 8(a), the comparison is conducted through varying the number of nodes. The more nodes there are, the less rounds of ring searching are needed. Our scheme is less than the optimal results and also requires more rounds of ring searching than the random scheme. In Fig. 8(b), we count R through varying the number of channels. When the total number of channels increases, the less rounds of ring searching will be needed. In Fig. 8(c), the channel availability becomes worse when the number of active primary users is more. The rounds of ring searching increase while the channel availability becomes worse. It is obvious that the overhead becomes larger when the rounds of ring searching increase.



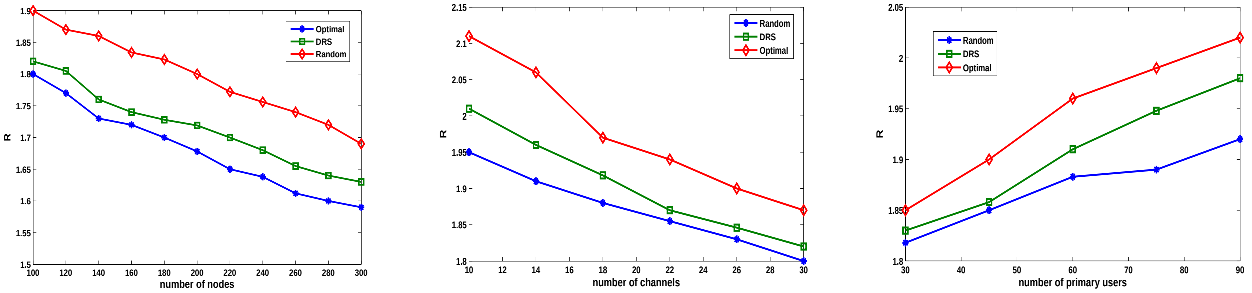
(a) change nodes (b) change channels (c) change primary users

Fig. 6. Comparison of delay under different network environment parameters.



(a) change nodes (b) change channels (c) change primary users

Fig. 7. Comparison of hop count under different network environment parameters.



(a) change nodes (b) change channels (c) change primary users

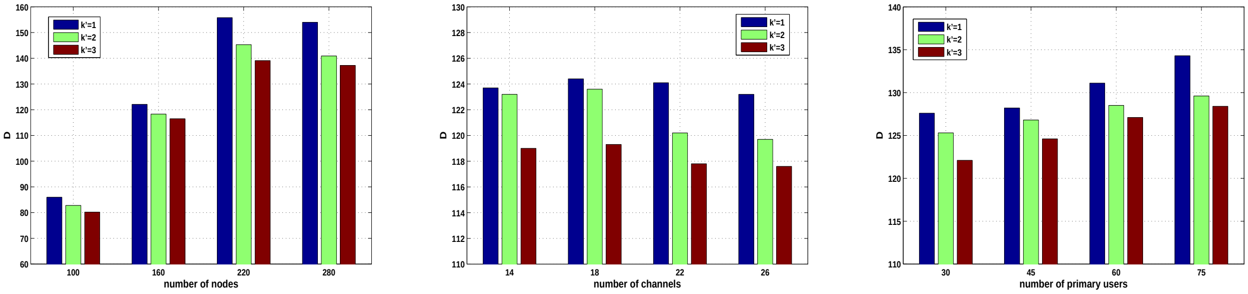
Fig. 8. Comparison of ring searching under different network environment parameters.

Therefore, our scheme causes less overhead compared to the optimal scheme, while the gap in delay of Fig. 6 is small.

From the above results, we can tell that our scheme shows its advantages from all three perspectives compared to the random scheme. Moreover, our scheme is close to the optimal results under the delay metric and causes less overhead.

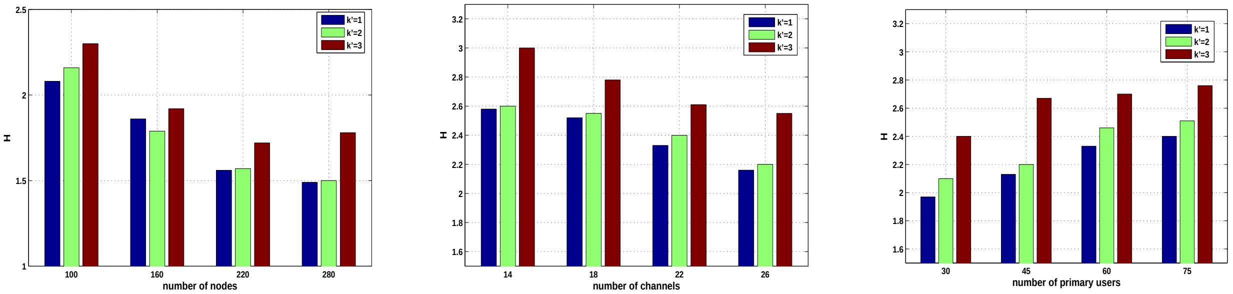
2) Comparison among scheme parameters: we show the influence of different scheme parameters on the performance of our scheme. Firstly, we compare the performance using the metric D under different settings of k'. The value of k' varies from 1 to 3. The results are shown in Fig. 9. In Fig. 9(a), the delay is the highest when $k' = 1$. $k' = 3$ achieves the lowest delay. Compared to the delay achieved by the optimal scheme in Fig. 6(a), $k' = 3$ is very close to the optimal results. In Fig. 9(b) and Fig. 9(c), the trend in both figures is familiar. That is, when the channel availability becomes worse, the delay gets longer. Moreover, considering only the metric D, $k' = 3$ performs the best out of the three. The performance gap of $k' = 3$ in Fig. 9 and the optimal scheme in Fig. 6 is small.

The comparison of metric H is shown Fig. 10. From the three figures, the hop count of the rerouting path is the largest when $k' = 3$. $k' = 1$ has the smallest hop count. Although the delay performance of $k' = 2$ is not as good as $k' = 3$, the average path length is less than $k' = 3$. Combined with the simulation results from previous parts, the H achieved by $k' = 1$ is close to the H achieved by the random scheme in Fig. 7, especially when the number of nodes is small or the channel availability is little. This is because nodes scarcely make the choice to construct rerouting paths in these situations. Moreover, $k' = 3$ takes less hop counts than the optimal scheme. However, sometimes, as shown in Fig. 10(c), when the number of primary users is 45, each node needs a higher hop count to achieve a similar performance to D compared to



(a) change nodes (b) change channels (c) change primary users

Fig. 9. Comparison of delay under different scheme parameters.



(a) change nodes (b) change channels (c) change primary users

Fig. 10. Comparison of hop count under different scheme parameters.

the optimal results.

VII. CONCLUSION

This paper considers the rerouting problem in cognitive radio networks. We propose a distributed rerouting framework, which considers both the channel availability and the network traffic. Our scheme aims at minimizing the queueing delay in total for rerouting multiple sessions. Nodes that need rerouting do not communicate with each other in our scheme. We apply an overhear scheme that enables each node to predict the probability of itself being chosen by nodes in its queue, which in turn helps to estimate the queueing delay for those nodes. We make use of RREP messages to deliver the estimated delay and channel availability. Such information would determine the construction of a rerouting path. In addition, we present three conflict resolution strategies, which are reserve-based, contention-based, and window-based. Extensive simulations are performed. The comparison results among our scheme, the optimal scheme, and the random scheme show that our scheme is better than the random one and comes close to the optimal scheme. We also analyze the influence of different scheme parameters on our scheme.

ACKNOWLEDGMENTS

This research was supported in part by NSF grants ECCS 1128209, CNS 1065444, CCF 1028167, CNS 0948184, and CCF 0830289.

REFERENCES

[1] I. F. Akyildiz, W.-Y. Lee, M. C. Vuran, and S. Mohanty, “Next generation/dynamic spectrum access/cognitive radio wireless networks: A survey,” *Computer Networks*.

[2] F. Commission, “Spectrum policy task force,” Tech. Rep., 2002.

[3] M. Cesana, F. Cuomo, and E. Ekici, “Routing in cognitive radio networks: Challenges and solutions,” *Ad Hoc Networks*, vol. 9, no. 3, pp. 228 – 248, 2011.

[4] W. Feng, J. Cao, C. Zhang, and C. Liu, “Joint optimization of spectrum handoff scheduling and routing in multi-hop multi-radio cognitive networks,” in *Proc. of IEEE ICDCS 09*.

[5] A. Abbagnale and F. Cuomo, “Gymkhana: A connectivity-based routing scheme for cognitive radio ad hoc networks,” in *Proc. of IEEE INFOCOM 10*.

[6] K. Chowdhury and I. Akyildiz, “Crp: A routing protocol for cognitive radio ad hoc networks,” *IEEE Journal on Selected Areas in Communications*, 2011.

[7] Y. Ding and L. Xiao, “Routing and spectrum allocation for video ondemand streaming in cognitive wireless mesh networks,” in *Proc. of IEEE MASS 10*.

[8] K. Chowdhury and M. Felice, “Search: A routing protocol for mobile cognitive radio ad-hoc networks,” *Computer Communications*, 2009.

[9] A. S. Cacciapuoti, M. Caleffi, and L. Paura, “Reactive routing for mobile cognitive radio ad hoc networks,” *Ad Hoc Networks*, 2011.

[10] I.Beltagy, M. Youssef, and M. El-Derini, “A new routing metric and protocol for multipath routing in cognitive networks,” in *Proc. of IEEE WCNC 11*.

[11] I. Filippini, E. Ekici, and M. Cesana, “Minimum maintenance cost routing in cognitive radio networks,” in *Proc. of IEEE MASS 09*.

[12] H. Ma, L. Zheng, X. Ma, and Y. Luo, “Spectrum aware routing for multi-hop cognitive radio networks with a single transceiver,” in *Proc. of IEEE CrownCom 08*.

[13] G. Cheng, W. Liu, Y. Li, and W. Cheng, “Spectrum aware on-demand routing in cognitive radio networks,” in *Proc. of IEEE DySPAN 07*.

[14] Z. Yang, G. Cheng, W. Liu, W. Yuan, and W. Cheng, “Local coordination based routing and spectrum assignment in multi-hop cognitive radio networks,” *Mobile Networks and Applications 2008*.