

Spectrum Decision for Efficient Routing in Cognitive Radio Network

Ramzi Saifan, Ahmed E.Kamal, Yong Guan
Dept. of Electrical and Computer Engineering
Iowa State University, Ames, IA 50011
{ramzi, kamal, guan}@iastate.edu

Abstract—The cognitive radio (CR) nodes in a cognitive radio network (CRN) do not have license to use specific spectrum band. Instead, they use the spectrum bands of the licensed primary users (PU) without interfering with the PU. When the PU becomes active, interfering CRs should leave to another available spectrum band within the PU's tolerable interference delay (TID). Therefore, CRN operates over wide spectrum bands which span many channels. Since each channel is typically licensed to one PU, this requires that channels be sensed separately. This adds monitoring overhead, where the CR should monitor (sense) the channel every TID, which reduces the throughput. For this reason, the node cannot monitor the whole set of channels. Deciding which set of channels to monitor affect other functions in the CRN like routing. Work done on routing in literature assumes that each node maintains a set of available channels which is obtained by sensing. Route setup decision will be made based on the available sets at all nodes. However, there may be some other available channels that the node is not aware of their availability which may enhance the routing quality metric. Also, taking into consideration only the sets of channels available at the CR nodes may preclude finding an end-to-end path. In this paper, we propose a spectrum decision framework that is complementary to the existing routing protocols. This framework is based on two objectives: 1) enhancing the route quality by sensing a few more channels at some nodes. These channels can enhance the quality by: reducing the switching time, requiring shorter sensing time, or expected to be available for longer time; 2) increasing the probability of finding a path by sensing more channels at some nodes in case the routing protocol did not find a path.

Simulation results show that the proposed framework can result in enhancement that can be as high as 100% over the routing protocols that build their decisions based on the available channels at each node only.

I. INTRODUCTION

The increasing demand on wireless spectrum and on broadband wireless services necessitated changes to the legacy wireless policies. The static policy that subjects the wireless spectrum bands to static and exclusive utilization decreases spectrum utilization, because the user who has the license to use a specific spectrum band (called primary user (PU)) does not use it all the time (temporal under-utilization) and in all locations (spatial under-utilization). Moreover, this static policy led to an increased competition for using the ISM wireless spectrum bands which caused these bands to be highly congested. The FCC exploded a bomb when it was announced that wireless spectrum utilization in some bands is less than 15%, and therefore, a more efficient spectrum utilization became required.

One way of increasing spectrum utilization is by adopting

opportunistic spectrum sharing (OSS), in which un-licensed users, also called secondary users (SUs) can use the spectrum band of the PU provided that it does not introduce harmful interference to the PU. The enabling technology for OSS is cognitive radio (CR) which enables the SU to sense the channels and adapt its transmission characteristics accordingly, using software defined radios (SDRs). In SDRs, the components that are traditionally built in hardware (e.g., modulators/demodulators, amplifiers, filters, detectors, etc.) are instead built in software on a personal computer or embedded computing devices.

The use of SDRs and the requirement of protecting the PU from any interference have introduced new challenges: First, the SDR allows the CR node to operate over wide spectrum bands with different characteristics. Second, if the PU can tolerate interference up to t seconds, then the CR should sense the channel at least once every t seconds. If sensing time takes ST ms for one channel and when the CR switches from one channel to another channel to start sensing, it takes SW ms, then a CR must spend $ST + SW$ ms for sensing each channel. This means that the CR if it has only one transceiver and is maintaining a list of N channels, then $N * (ST + SW)$ ms is wasted on sensing, which may be a significant fraction of the t seconds cycle.

Therefore, based on the above, it is evident that choosing the best set of channels can minimize the overhead time (sensing plus switching time), which is known as spectrum decision. The CR node should know which channels it must monitor, where the set of such channels depends on the objective. For example, if the objective is the routing protocol, then the spectrum decision aims to find the channels that will minimize the end-to-end delay or maximize the throughput. If the objective is to increase route stability, then the node can select the channels that are expected to be available for longer times. In this paper, we will consider the spectrum decision problem where the objective is routing.

Routing in CRNs jointly selects the path and the channel to be used on each hop according to a quality objective. The quality objectives are classified into: minimizing end-to-end delay, maximize throughput, minimize interference, and increase path stability. According to [1], routing can be classified into two main types: full spectrum knowledge and local spectrum knowledge. The full spectrum knowledge assumes that there is a central entity that knows all the available channels at each CR node without sensing, thanks to

spectrum availability databases [2]. Indeed, this increases the options and gives better routing results. However, as explained in [1], this is not practical.

The local spectrum knowledge approach is more practical. To the best of our knowledge, all local spectrum knowledge approaches assume that each CR maintains a set of available channels which is obtained by sensing. Then, CRs apply their routing algorithm which finds the path and the channel to be used on each hop such that their quality objective is optimized, and the channel used on each hop, should exist within the set of available channels at both nodes at the two ends of that hop.

To implement these approaches, some questions must be answered: 1) what is the optimal size of the set that should be maintained by each CR node. From the routing point of view, the bigger the set, the better the achievable routing quality objective. However, from sensing point of view, the smaller the set, the less the sensing time overhead, 2) Since, as we mentioned previously, monitoring all the channels consumes considerable time which is also infeasible. Therefore, is the routing decision that was made the best? What if there is another channel that is not used by a PU (available) and enhances the quality objective, but was not selected because the CR is not aware of its availability? and 3) If applying a specific routing algorithm was not able to find a path because, for example, on one or more hops there is no common available channel that is within the set of available channels at both nodes at the two ends of that hop: is there a possibility that there will be another channel that is available at both nodes, but the nodes are not aware of it because they did not sense it?

The objectives of this work can be summarized in: 1) introducing a new framework for spectrum decision which increases the options for a CR node by allowing it to inspect more channels, including the channels that the CR is not aware of their availability. The selection will be according to a specific criterion that takes into consideration the sensing time, the switching time, the access and channel sharing time, and the expected available channel time, 2) use this framework to enhance the performance of the existing routing algorithms, for example, by finding another channel on one hop that increases the throughput or that minimizes the end-to-end delay and 3) if applying the routing algorithm was not able to find a path from a source to a destination, we will use the framework to try to find a path because our framework increases the probability of finding a path since it inspects more channels.

The rest of this paper is organized as follows. The system model is introduced in Section II. The framework for enhancing the throughput of existing routing protocols is in Section III, and in Section IV we explain how to use the suggested framework in increasing the probability of finding an end-to-end path. Simulation results are introduced in Section V. Section VI shows the related work and finally we conclude in section VII.

II. SYSTEM MODEL

The main objective behind this work is to design a spectrum decision framework that will not only consider the set of available channels at each CR node, but also the other channels that are not maintained by the CR nodes and may be available. Our objective is to use this framework to enhance the performance of existing routing protocols, and not to design a new routing protocol that jointly finds the path and the channel to be used on each hop of the path.

We assume that each channel is assigned to one PU who has an exclusive right to use it whenever he wants. If the PU can tolerate interference up to 1 second, then the CR should sense (monitor) the channel periodically every second. If the CR node is maintaining a set of channels, the CR node should sense each of these channels periodically. In addition to the sensing time, the CR node takes some time to switch from one channel to the other. Switching time depends on the frequency step, e.g., to switch from a channel on central frequency, f_1 MHz, to a channel on central frequency, f_2 MHz, the switching time will typically be $\alpha * |f_1 - f_2|$ [3], where α is the switching time per 1 MHz step, and is technology dependent.

We also assume that there exists a routing algorithm that finds the path and the channel to be used on each hop. Therefore, the inputs to our framework are: 1) CRN topology, which consists of the CR nodes and their locations, 2) The outputs of the routing algorithm which are a path from a source to a destination, and the channels selected on each hop, 3) The set of all the channels that the CRN can potentially use, and 4) Some statistics about the PUs activity like the expected active and inactive times, its location, required periodic monitoring time on each CR node, maximum tolerable interference delay which determines how often the CRs should sense the channel.

The output of this framework will be in the form of a set of recommendations to some CRs to sense some channels in order to enhance the routing quality. The recommendations stem from the question: given the output of the routing algorithm which is a path and the channels on that path that are supposed to optimize a specific quality metric, can we enhance the quality of that path by finding other channels on one or more of the hops? The enhancements could be because of finding another channel that requires less monitoring time, less switching time, less access delay, or is being shared between fewer nodes. If the answer to the previous question was yes, then the CR will compare the expected extra cost with the expected benefits that can be gained. If the benefits exceeds the cost, the CR will sense the channel. If the channel found to be available, the CR will start using it. In this paper, we considered the throughput as the quality metric. However, the same approach can be applied on any routing quality metric.

We assume that all channels have the same bandwidth. We also assume that the activities of the PU on channel k can be represented by a birth/death process as in Figure 1, with birth rate (becoming busy), β , and death rate (becoming idle), λ , then the expected time for channel k to be idle within a cycle of activity is $(E_k(\mathcal{H}0) = \frac{1}{\beta})$. Moreover, probability for the

PU's channel to be available, $Pr(\mathcal{H}0) = \frac{\lambda}{\lambda+\beta}$, and probability to be busy, $Pr(\mathcal{H}1) = \frac{\beta}{\lambda+\beta}$.

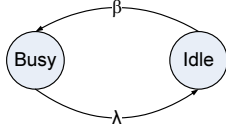


Fig. 1. PU activity model

III. ENHANCING THROUGHPUT

In this section, we assume that there exists a routing algorithm that finds a path which maximizes the throughput given the sets of sensed channels in the CRN, and the switching times. Therefore, the output of this routing algorithm which forms the input to our framework is a multi-hop path with the channel to be used on each of these hops. Suppose that the cycle length is t_c ms, switching time is α ms/1 MHz, and the channel sensing time for channel i by node j is $ST_j(i)$ ms.

We normalize all of our calculations to the cycle length. Since we assume that all channels have the same bandwidth, then the throughput can be measured by the transmission time, which is equal to:

$$\text{Transmission time} = \text{cycle length} - \text{cycle wasted time} \quad (1)$$

Therefore, throughout this section we enhance the throughput by increasing the transmission time per cycle. The wasted time is the time due to channel sensing, switching between channels, switching between multi routes, and due to access and sharing a channel.

If a path is composed of multiple intermediate nodes, node (n_i) has the highest wasted time = t_{wi} ms/cycle, and (n_j) has the next highest wasted time = t_{wj} ms/cycle. Then, during each cycle, the destination will not receive for more than $t_c - t_{wi}$ ms. Therefore, if we decide to maximize the throughput, we should find a way to reduce the wasted time at node i . Also, the upper limit of enhancement is to reduce the wasted time at n_i down to (t_{wj}) . Therefore, if by finding another channel on node i which reduces the wasted time to $(\hat{t}_{wi} < t_{wj})$, the benefit will be upper limited by t_{wj} , which means the benefit will be $t_{wi} - \hat{t}_{wi}$. Whereas if $\hat{t}_{wi} > t_{wj}$, the benefit will be $t_{wi} - \hat{t}_{wi}$.

We will discuss the enhancements from applying the proposed framework to multiple cases (Figure 2). These cases are not exhaustive, but many cases can be only simple extension to these. Throughout all the following cases, we will explain the bottleneck with respect to node c , such that it encounters the maximum delay which reduces the transmission time left. Note that neither node d need be the destination nor node a need be the source. We are showing only part of the path. Throughout all the cases, we are using some dummy numbers just for the purpose of explanation. The input column shows the result of applying the existing routing protocol which maximizes throughput. The next column shows the result of applying our framework. The last column briefly states the

cause of the bottleneck. For example, in the fourth case, using same channel decreases the throughput. Therefore, in the third column, node c chooses to sense another channel and will use it if it is available on both nodes c and d .

A. single path enhancement examples

In this section, we will discuss the cases when there is only one path that is node and channel disjoint with all other paths.

Case 1: Suppose that the output from the routing algorithm is as shown in Figure 2 Case 1 in the input column. Suppose that node d did not sense channel 1 before making the routing decision such that node d built its routing decision based on the list that it was maintaining, and channel 1 was not among that list. According to the figure, nodes a , b , and c are maintaining channel 1. Therefore, they should sense it periodically. Also, nodes c and d are maintaining channel 5 where they sense it periodically. Since node c is maintaining channels 1 and 5 while node d is maintaining channel 5 only, then it should sense these two channels every cycle, and switch between the channels every cycle, as shown in Figure 3.

Case	Input	After applying the framework	Criterion
1			Different channels
2			Sensing time
3			Switching time
4			Same channel
5		A) B)	Interfering node
6		A) B)	Joint node

Fig. 2. Multiple examples on routing enhancements using the proposed framework

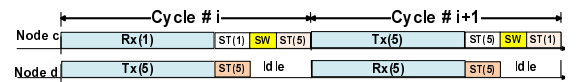


Fig. 3. Time line for nodes c and d ; SW=switching time, ST=sensing time

Our framework recommends enhancement to this routing decision by looking at the bottleneck node, which is node c . This is because node d cannot use channel 1 because it is not within its list of available channels. Therefore, node c needs to switch between channels 1 and 5. Node d can use the idle time during node c switching time, to sense channel 1. According to Figure 3, the idle time at d for this case can be given by the equation:

$$IdleTime_d = ST_c(1) + SW(1,5) + ST_c(5) - ST_d(5) \quad (2)$$

Assuming symmetric switching, i.e., $SW(1,5) = SW(5,1)$, the time overhead for sensing channel 1 by node d is given by the equation:

$$\begin{aligned} SensingOverhead_d &= SW(5,1) + ST_d(1) + SW(1,5) \\ &= 2 * SW(1,5) + ST_d(1) \end{aligned} \quad (3)$$

Then, the cost (C) that node d pays is 0 if the sensing overhead is less than the idle time. Otherwise, it is given by the equation:

$$\begin{aligned} C_d &= SensingOverhead_d - IdleTime_d \\ &= SW(1,5) + ST_d(1) - ST_c(1) - ST_c(5) + ST_d(1) \end{aligned} \quad (4)$$

Then, the expected cost (\bar{C}_1) is:

$$\bar{C}_1 = \begin{cases} zero & \text{if } IdleTime_d \geq SensingOverhead_d \\ C & \text{otherwise} \end{cases} \quad (5)$$

where C is given in equation (4). On the other hand, the gain per cycle (G) that can be achieved by adding channel 1 to the list of channels maintained by node d is given by the following equation:

$$\begin{aligned} G &= \text{Old wasted time} - \text{new wasted time} \\ &= ST_c(1) + SW(1,5) + ST_c(5) - \max\{ST_c(1), ST_d(1)\} \end{aligned} \quad (6)$$

Suppose that the probability of channel 1 being available at node d is ($Pr_d^1(\mathcal{H}0)$) and being busy is ($Pr_d^1(\mathcal{H}1)$). Then, the expected gain (\bar{G}_1) from sensing channel 1 at node d is given by the following:

$$\begin{aligned} \bar{G}_1 &= (\text{gain per cycle}) * (\text{expected \#of idle cycles}) * \text{Pr}(\text{idle}) \\ &= G * \frac{E_1(\mathcal{H}0)}{t_c} * Pr_d^1(\mathcal{H}0) \end{aligned} \quad (7)$$

In case node d has sensed channel 1 and knows for sure that it is busy ($Pr_d^1(\mathcal{H}0) = 0$), then according to Equation (7), the benefit will be zero. Therefore, if the cost is larger than zero, it is useless to sense channel 1 at node d . In other words, by comparing the expected cost (5) to the expected gain (7), we can estimate whether it is cost effective to sense channel 1 at node d or not. Note that in this case we are assuming that node c can send to node d on channel 1 and at the same time node a can send to node b on channel 1. This can happen by using different codes in code division multiple access techniques, or by controlling the transmission power if it is possible. In Case 4, we will show the scenario when it is not possible to simultaneously use the same channel for communication.

Case 2: This case happens when the sensing time of channel 3 at node c ($ST_c(3)$) takes long time such that $SW(1,3) + ST_c(3) > SW(1,5) + ST_c(5)$. The benefit that could be gained from this case is less than Case 1. This may happen if node c is away from the PU that owns channel 3, or because the SNR to the PU is very low, which requires longer sensing time to achieve the required PU detection probability requirement.

Using derivations similar to those in case 1, we derived the final equations for this case. Due to space limitations and since they are similar to those above, we do not show the derivation steps. In this case, the extra cost at node c is the overhead of switching from channel 3 to channel 5, sensing channel 5,

and switching back to channel 3. Therefore, the expected cost (\bar{C}_2) can be given by the following equation:

$$\bar{C}_2 = 2 * SW(3,5) + ST_c(5) \quad (8)$$

And the expected gain (\bar{G}_2) is:

$$\begin{aligned} \bar{G}_2 &= [ST_c(1) + SW(1,3) + ST_c(3) - \\ &\{ST_c(1) + SW(1,5) + ST_c(5)\}] * \frac{E_5(\mathcal{H}0)}{t_c} * Pr_c^5(\mathcal{H}0) \\ &= [SW(1,3) + ST_c(3) - \{SW(1,5) + ST_c(5)\}] * \\ &\frac{E_5(\mathcal{H}0)}{t_c} * Pr_c^5(\mathcal{H}0) \end{aligned} \quad (9)$$

If the expected gain calculated by (9) is less than zero, this means that using channel 5 will be more expensive than using channel 3, because the sensing plus switching time is larger for channel 5. Moreover, if it is positive, but smaller than the expected cost calculated by (8), then it is not beneficial to sense channel 5. However, if it is positive and greater than the expected cost, then node c can sense channel 5. Node d is required to sense channel 5 also. But, we are assuming without loss of generality that the bottleneck is at node c . Therefore, node d can sense channel 5 while node c is sensing channel 5, which means no extra overhead.

Case 3: This case is beneficial in case the switching time is the dominating factor. e.g., in Figure 2 Case 3, since node c is required to switch from channel 1 to channel 10 every cycle. If c can find another channel that minimizes the switching plus sensing time like channel 2 in the figure, this will reduce the wasted time. Since we are assuming linear switching time ($\alpha * |f1 - f2|$), and since node c is required to switch from channel 1 to channel 10, then there is no extra cost for switching because $SW(1,10) = SW(1,2) + SW(2,10)$, otherwise, we should consider the extra switching time. Hence, the expected cost can be given by the equation:

$$\bar{C}_3 = ST_c(2) \quad (10)$$

And the expected gain (\bar{G}_3) is:

$$\begin{aligned} \bar{G}_3 &= [ST_c(1) + SW(1,10) + ST_c(10) - \\ &\{ST_c(1) + SW(1,2) + ST_c(2)\}] * \frac{E_2(\mathcal{H}0)}{t_c} * Pr_c^2(\mathcal{H}0) \\ &= [SW(1,10) + ST_c(10) - \{SW(1,2) + ST_c(2)\}] * \\ &\frac{E_2(\mathcal{H}0)}{t_c} * Pr_c^2(\mathcal{H}0) \end{aligned} \quad (11)$$

Case 4: Under some cases, there are benefits due to switching to other channels and not using the same channel on multiple consecutive hops even if it is available. For example, in Figure 2 Case 4, channel 1 is used for the shown three hops. But, if the used channel-sharing method prevents nodes a and c from simultaneously sending on the same channel because node b will not be able to receive data from a when c is transmitting to d . Figure 4 shows the time lines for nodes c and d . The figure shows that one third of the time the node is idle because node d cannot send on channel 1 when c is receiving on the same channel. The sensing in this case can be done during the idle time. Moreover, during the idle time, node c can

sense some other channels, e.g., channel 5 in Figure 2. And the cost will be zero if $(ST_c(5) + 2 * SW(1, 5)) \leq (\frac{1}{3} * t_c - ST_c(1))$, which is most probably the case. Otherwise, the expected cost will be:

$$\bar{C}_4 = ST_c(5) + 2 * SW(1, 5) - \left\{ \frac{1}{3} * t_c - ST_c(1) \right\} \quad (12)$$

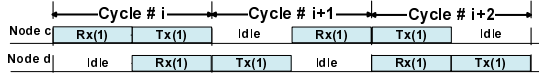


Fig. 4. Time line for nodes c and d in Case 4

If node c found a channel other than channel 1 to be used between c and d , the time lines for them will be very similar to the time lines in Figure 3. Therefore, the expected gain behind using another channel like channel 5 between nodes c and d will be considerable which can be calculated by the following equation:

$$\bar{G}_4 = \left[\frac{1}{3} * t_c - \{ST_c(1) + SW(1, 5) + ST_c(5)\} \right] * \frac{E_5(\mathcal{H}0)}{t_c} * Pr_d^5(\mathcal{H}0) \quad (13)$$

The same method can also be applied between nodes a and b by sensing another channel in order to use it.

B. Multi-path enhancement

In this sub-section, there will be another factor that may affect the throughput of the selected route which is the co-existence of another route that either intersects with the path under study by having a common node, or there is no common node, but there is a channel that is if used on the two routes at the same time, interference happens. Therefore, the two routes alternate on that channel or on that node, which considerably reduces throughput.

Case 5: In this case, as Figure 2 Case 5 shows, suppose that there exists a node, e that is an intermediate node on another path, and this node is very close to another node (c). Both nodes use channel 5 for transmission. Since they are close to each other, they cannot transmit at the same time on channel 5 due to interference. Therefore, nodes d and e should alternate on channel 5. The time lines for node c will be very similar to the one in Figure 4. Due to space limits, we are not showing it. Therefore, one third of the time, node c is idle, and during this idle time, the node can do the sensing for the channels it maintains. Two possible enhancements are shown in Cases 5.A and 5.B in the output column. We will focus here on Case 5.A because we are studying that input route. Therefore, node c will look for another channel (e.g., channel 8 in Figure 2) to be used instead of channel 5. If it is found to be idle, the time line for node c will look like the one in Figure 3.

If the time required to sense channel 8 is less than the idle time, it will sense it and the cost will be zero. Otherwise, the cost will be equal to:

$$\bar{C}_5 = 2 * SW(5, 8) + ST(8) - \left\{ \frac{1}{3} * t_c - (ST_c(1) + SW(1, 5) + ST_c(5)) \right\} \quad (14)$$

While the gain is given in the following equation:

$$\bar{G}_5 = \left[\frac{1}{3} * t_c - \{ST_c(1) + SW(1, 8) + ST_c(8)\} \right] * \frac{E_8(\mathcal{H}0)}{t_c} * Pr_c^8(\mathcal{H}0) \quad (15)$$

Note that we did not subtract the sensing time before the enhancement from the first part because node c was idle for $\frac{1}{3}$ of the cycle where it can sense the channels any time within that time.

Case 6: In this case, node c is the bottleneck because it is an intermediate on two routes, which means it will alternate between the two routes, where it will forward the data of the given route half of the time, and forward the data of the other route for the remaining time as Figure 5 shows. The dashed rectangles are for the other route. The sw rectangles are switching between channels 1 and 5. The sensing times are one for channel 1 and the other for channel 5. Two enhancements could be done as shown in the two cases 6.A and 6.B in the output column of Figure 2. We will explain Case 6.A. Case 6.B will be exactly the same. The main enhancement in Case 6.A is by finding another intermediate node other than node c to forward the data on one of the two routes. This case is different from the previous cases in that it includes finding another node, not just finding another channel.

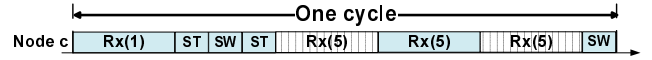


Fig. 5. Time line for node c in Case 6

This could be initiated by node c sending a message to its neighbors telling them if any of them is physically reachable by nodes b and d , even if there is no common channel known to be available. Suppose node e was found with channel 1 available. Then, we can ask nodes e and d to find a common channel to be used for routing data in this route. Suppose that they are interested in channel 8. Then, the cost will be zero if $[2 * SW(5, 8) + ST_d(8)] \leq 0.5 * (t_c + \{ST_c(1) + SW(1, 5) + ST_c(5)\})$. Otherwise, the cost will be given by the following equation:

$$\bar{C}_6 = \{2 * SW(5, 8) + ST_d(8)\} - \{0.5 * (t_c + \{ST_c(1) + SW(1, 5) + ST_c(5)\})\} \quad (16)$$

And the expected gain from this enhancement is:

$$\bar{G}_6 = 0.5 * (t_c + ST_c(1) + 2 * sw(1, 5) + ST_c(5)) - (ST_c(1) + 2 * sw(1, 8) + ST_c(8)) * \frac{E_8(\mathcal{H}0)}{t_c} * Pr_e^8(\mathcal{H}0) \quad (17)$$

C. Protocol

In this sub section, we will introduce a protocol for determining how many channels to sense, which channels to be selected for sensing and when to do the sensing. Case 1 is straight forward, since node d knows that the enhancement can be achieved by looking for the availability of channel 1. Therefore, node d can start sensing channel 1 during the idle time if the expected gain is larger than the expected cost. To

find the expected cost and gain, we use the previous derived equation in the previous section.

Regarding the other cases, there are many options for node c to choose from. Therefore, it needs to know what the best channel is to start with. Algorithm 1 shows the general scenario to follow. The idea will be done by sorting the channels descendingly according to the (payoff = Expected gain - Expected cost).

To find the i^{th} channel to sense, we want to find the channel with the highest payoff among the remaining $(M - i - 1)$ channels. The node loops over all the potential $(M - i - 1)$ channels. In iteration (j) of the loop, the node first, calculates the gain from using that channel (line 5). Second, it will calculate the cost of inspecting that channel which is the cost of inspecting all the previous $(i - 1)$ channels (line 15) plus inspecting the (i^{th}) channel. The cost of inspecting the (i^{th}) channel includes the switching time from the previous channel (f_0) to the iterated (j^{th}) channel (line 6). Initially, the current channel is the channel that node c is using for transmission. The first channel to be sensed is the one with the maximum payoff (lines 20-23). Then, the node assumes that the previous channel (f_0) is the channel that maximizes the payoff for the current outer loop iteration (Line 23).

The cost will be zero when the idle time is larger than the cost. In cases 2 and 3, the idle time will most probably be smaller than that in Cases 4, 5, and 6. If the maximum payoff was not negative, the node will subtract the cost (which is the sensing plus switching times) from the idle time (line 24).

This procedure will be repeated until the node will not be able to find a channel with positive payoff (lines 17-19). Then, number of channels to sense is known (*NumberOfChToSense*) and the order of the channels the node should follow during sensing is also known (f). If the maximum payoff is negative, the node should not sense any channel (*NumberOfChToSense* = 0), which means there is no possible enhancement to the current situation.

Algorithm 1 : Protocol

```

1: NumberOfChToSense  $\leftarrow$  0
2: for  $i=1:M$  do
3:   MaxPayoff  $\leftarrow$  -1, MaxIndex  $\leftarrow$  -1
4:   for  $j=i:M$  do
5:      $G \leftarrow$  Gain( $j$ )
6:      $C \leftarrow$  TotalCost( $i - 1$ ) + Cost( $j, f_0$ )
7:      $P \leftarrow G - C$ 
8:     if ( $P \geq$  MaxPayoff) then
9:       MaxPayoff  $\leftarrow$   $P$ 
10:      MaxIndex  $\leftarrow$   $j$ 
11:     end if
12:   end for
13:   if (MaxPayoff  $\geq$  0) then
14:     NumberOfChToSense ++
15:     TotalCost( $i$ )  $\leftarrow$  TotalCost( $i - 1$ ) + Cost( $f$ (MaxIndex),  $f_0$ )
16:     TotalGain( $i$ )  $\leftarrow$  Gain( $i$ )
17:   else
18:     return NumberOfChToSense,  $f$ 
19:   end if
20:    $Temp \leftarrow f(x)$ 
21:    $f(x) \leftarrow f$ (MaxIndex)
22:    $f$ (MaxIndex)  $\leftarrow$   $Temp$ 
23:    $f_0 \leftarrow f$ (MaxIndex)
24:   IdleTime  $\leftarrow$  IdleTime -  $C$ 
25: end for

```

IV. ENHANCE ROUTING SETUP

Since connectivity in CRN is weaker than other networks and frequently changes because it depends on PU behavior, sometimes if a source (s) wants to setup a path to a destination (d), there will be no path. The reason is that on one or more of the hops, there is no common available channel at both nodes at the two ends of that hop. Nevertheless, there may exist a channel that is available, but the nodes are not aware of its availability because they did not sense it. Existing routing protocols will not be able to find the path. However, it is not reasonable to sense all the channels on all the nodes each time there is a need for a route setup, or there is a discontinuity due to PU reappearance. Therefore, we want to know which nodes are better to sense which channels, and when.

Again, we are not designing a new routing protocol. In case the used routing protocol did not find a path from the source to the destination, we will use our framework to find a path which is supposed to increase the probability of finding a path by increasing the number of channels to be checked at some of the nodes.

A common control channel (CCC) is used to flood the route request packet (RRQP) from the source to the destination. Each intermediate node modifies the value of the quality metric (which is defined below). If the intermediate node does not share an available channel with the upstream node, it only increments the number of discontinuities by one and forwards the RRQP to its neighbors through the CCC again.

The destination will receive multiple RRQPs, each contains a two dimensional metric: 1) the quality value and 2) the number of discontinuities value. After that, and depending on the target, the destination node decides to choose: 1) the path with minimum additional setup time (could be the one with the minimum number of discontinuities), 2) the best quality metric value, or 3) a path that achieves best quality metric value such that the number of discontinuities is less than a given constant. In this paper we will handle the first one. The other two types may be done as a future work.

The quality metric value could be the end-to-end delay, where each intermediate node decides the value of the delay the packet will encounter at the node, add it to the end-to-end delay value in the RRQP, and rebroadcast the RRQP. The node will not re-broadcast the same RRQP again to prevent cycles, except if it has better quality value and/or less number of discontinuities. If the quality metric is throughput, then each node can decide whether it is the bottleneck node or not. In case it is the bottleneck node, then it will modify the RRQP quality value to its throughput. Otherwise, it will not modify it. Then, re-broadcast it.

To estimate the time to find a common channel on one of the hops between two nodes, say x and y , each node initially has a set of available channels. For channel i , that is within the set of available channels at x , but not within the set of available channels at y , the probability to be available at both x and y will be $1 - Pr(\text{not available at } y \text{ given it is available at } x) = 1 - Pr(\mathcal{H}_1 \text{ at } y | \mathcal{H}_0 \text{ at } x)$. Same thing for any channel within the set of available channels at y , but is not within the

set of available channels at x , the probability to be available at x and $y = 1 - Pr(\mathcal{H}1 \text{ at } x | \mathcal{H}0 \text{ at } y)$.

On the other hand, for any other channel that is neither within the available set of channels at x , nor the available set of channels at y , the probability to be available at x and y equals:

$$\begin{aligned} Pr(\mathcal{H}0 \text{ at } x \& \mathcal{H}0 \text{ at } y) &= Pr(\mathcal{H}0 \text{ at } x | \mathcal{H}0 \text{ at } y) * Pr(\mathcal{H}0 \text{ at } y) \\ &= Pr(\mathcal{H}0 \text{ at } y | \mathcal{H}0 \text{ at } x) * Pr(\mathcal{H}0 \text{ at } x) \end{aligned} \quad (18)$$

The conditional probabilities: $Pr(\mathcal{H}1 \text{ at } x | \mathcal{H}0 \text{ at } y)$, $Pr(\mathcal{H}1 \text{ at } y | \mathcal{H}0 \text{ at } x)$, $Pr(\mathcal{H}0 \text{ at } x | \mathcal{H}0 \text{ at } y)$, and $Pr(\mathcal{H}0 \text{ at } y \text{ and } \mathcal{H}0 \text{ at } x)$, can be calculated from the channel model. For example, [4] models the power received by a CR node by a log-normal random variable. Another approach that can be used is the spectrum cartography maps [5].

If we have multiple discontinuities on one path, then the extra time needed to set up the end-to-end path equals the time consumed at one of the intermediate nodes such that it needs the longest time to find a common available channel with the upstream node and/or the downstream node.

To estimate the minimum time required to find a channel to be available on one hop between two nodes (say x and y), we can follow a way similar to the one in Algorithm 1, but taking into consideration only the cost. We find the cost which is the channel's sensing time plus the switching time from the current channel, multiply the cost by (1- probability of channel to be available at both nodes) to find the expected cost. In each round of the outer loop, we find a channel with the minimum expected cost, and assume it as the current channel, and find the next channel and so on.

To estimate the time, we now know: 1) the order of channels to be followed during search from the previous step, 2) the probability for the channel to be available and 3) the sensing time for each channel. We can calculate the time until finding an available channel with high probability (e.g., ≥ 0.95) which is a geometric distribution.

Each node may find a different sequential order. But, the two nodes should follow the same sequential order during sensing. The two nodes will exchange their sensing decisions on the CCC such that if node x finished sensing channel i first, and found it to be busy, it tells y that it is busy and do not continue sensing. In this case the search time will be minimized because we are taking the minimum sensing time at each node plus some extra communication time overhead.

V. SIMULATION RESULTS

We conducted our simulation using Matlab. In the simulation, we studied how the throughput will be affected by the sensing time and the switching time. Switching time is represented by the switching factor (α) which is the time in ms required per 1 MHz frequency step. When a channel in use becomes busy, the CR searches for another available channel. We did not consider the search time because it is out of the scope of this paper and it will not affect the results. For the six cases in Figure 2, we will show the improvement percentage over the traditional protocols, which refers to the protocols

that do not consider sensing other channels. For example, if a node maintains a set of 4 channels, then during route setup, the route decision at that node will be made based on these four channels without sensing more channels.

Throughout the simulation, we assumed that the potential number of channels that the CR can work on is 100 channels, and we simulated the operation for 1000 seconds. The cycle length is taken as 1 second, the CR node should sense each channel it maintains every cycle. Channel bandwidth is 6 MHz. λ and β in Figure 1 for each PU are selected randomly between 0.01 and 0.1. In the first five cases, we only considered the node that has the bottleneck and the downstream node which are similar to nodes c and d in Figure 2, respectively. Since we are studying the throughput, they are enough if we assumed that the bottleneck is at node c . In the sixth case, we considered the three nodes b , c , and d .

To measure the improvement of Case 1, initially, node c will receive on any channel found to be available at c , and will send on any channel found to be available at c and d . In the traditional protocols, they will keep sending on that channel until one of the channels becomes busy. However, in our protocol, node d will check the channel that node c receives on: if it is found to be available, nodes c and d will start using that channel for their communication, otherwise, they will keep on using the same channels. When the channel found to be available, nodes c and d will keep communicating on that channel until it becomes busy. At that point, the two nodes will switch to two new random channels out of those available.

Figures 6.a–b show the effect of sensing time and α on the percentage of improvement over traditional routing protocols in the first case. As α increases, the improvement increases. This is because in case the channel found to be idle at d , nodes c and d will start using it for their communication, and the switching overhead at node c will be zero. For traditional protocols, the overhead increases and the throughput decreases as α increases. Therefore, the improvement increases. The enhancement also increases with increasing the sensing time. This is because in case the channel that c receives on is found to be available also at d , node c will not need to sense two channels every cycle. Instead, it will sense one channel which reduces sensing time and increases the throughput.

The second case improvement happens when node c switches between two channels, and the node can find another channel that requires a shorter sensing time. On the other hand, case 3 happens when node c switches between two channels and the switching time takes long time. Therefore, for Case 2, we selected the sensing time for each channel randomly between 1 and 100 ms for nodes c and d , which is a large range in order for some channels to have longer sensing time than others. For case 3, we considered small sensing times compared to switching time. As shown in Figure 6.b, the improvement is more in case the switching time is increased because our framework will try to find a channel that reduces switching time. For these two cases, searching time that is needed in our framework to find a better channel is considered

as cost, and it is included in the results. When the channel that either node c receives on or it uses to send to d becomes busy, they will switch to a channel randomly. After that in our framework, node c will try to find a channel that reduces the sensing time (Case 2) or a channel that reduces the switching time (Case 3).

Figure 6.a shows a small improvement in the second case (up to 2%). This is because the sensing time range is small (between 1 and 100 ms each cycle) compared to the transmission time (1 second cycle).

In cases 4 and 5, node c is idle for one third of the time. During this idle time, it can perform sensing. Therefore, in traditional routing protocols, increasing the sensing or switching times will not affect the throughput as long as the sensing plus switching times are less than the idle time (one third of the time). But, in our framework, if the node switched to another channel such that it will not be idle one third of the time, increasing sensing or switching time will reduce the throughput because they reduce the transmission time. This explains why the improvement is decreasing with increasing the switching or the sensing time as Figures 6. c and d show. The same thing for the sixth case, but with bigger enhancement. In Case 6, node c is wasting half the time for routing data of the other path. Therefore, if it can find another node to route the data, the throughput will be doubled. For this reason the improvement is close to 100%. In Figure 6.d, the sensing time at each node for each channel is selected randomly between 10 and 50 ms.

Enhancing route setup results: To see the effect of our proposed framework on this metric, we deployed 52 and 102 nodes in an area of size 1000m x 1000m. The source node is located at (0,0) and the destination node is located at (1000,1000). The other nodes are at random locations. The total number of channels that the nodes can select from is 20. Each CR node maintains a set of channels which range from 1 up to 8, as Figure 6.e shows. These channels are selected by each node randomly out of the 20 channels. To find whether there is a path from the given source to the given destination, we modeled the CRN as a graph. The vertices are the CR nodes, and edge between any two nodes exists if they are within the transmission range of each other and they share a common channel. To find a path, we use the breadth first search approach from the source to the destination.

Figure 6.e shows that by increasing the transmission range of the CR node, or by increasing the number of channels the CR node maintains, the probability of finding a path will increase. Also, by increasing the number of nodes in the network, the probability of finding a path increases. In the figure, "400m, n=100" means that the transmission range of the node is 400m and the number of nodes in the network, other than the source and the destination, is 100 node.

Figure 6.f shows the expected time required to setup a path. This time is the extra time required to find common channel between the two nodes that are the two ends of each hop which have no common channel. In all the results, the setup time was less than a second. As the number of

discontinuities increases, the setup time increases because we are choosing the maximum time to find a common channel on each hop. We are also showing the setup time for 4 discontinuities with one and two joint discontinuities. We mean by joint is that one of the nodes has no common channel with both upstream and downstream nodes. This does not make a big difference, because in many cases when a node finds a common channel with its upstream node, the same channel will be also available on the downstream node. For this reason, the curve of 4 discontinuities with zero joint and the curve of 4 discontinuities with one joint cross each other multiple times. In this figure, we conducted the simulation on 100 channels. Sensing time of each channel on each node is selected randomly between 1 and 100 ms.

VI. RELATED WORK

In-band sensing (monitoring) of PU should be done at least once every cycle. The cycle duration is the PU's tolerable interference delay (TID). A large body of research was done on minimizing the in-band sensing. Monitoring time depends on factors like PU SNR, shadowing and fading on the CR node, the distance from the PU, and the PU behavior. The authors in [6], [7] showed that required monitoring time that achieves PU's required detection probability varies from node to node. Therefore, cooperative sensing was proposed [8], [9]. Also, cooperative sensing achieves more accurate results.

Routing decisions depend on the set of available channels. However, routing approaches developed in the literature did not consider sensing overhead. Routing protocols in CRN can be classified into full spectrum knowledge and local spectrum knowledge [1]. In full spectrum knowledge like [10], [11], there is a central entity that has all the information about all the channels and their availability, thanks to the wireless spectrum databases [2]. These approaches, if solved optimally, should give the optimal results since they build their routing decisions based on information about all the channels without the need for sensing the channels. But according to [1], these approaches are not practical.

Local spectrum knowledge approaches assume that each node has some local knowledge about the available channels built through sensing. For example, the routing protocols in [12], [13] tried to maximize the throughput. The authors in [3], [14] tried to minimize end-to-end delay. The authors in [15] established robust paths in diverse spectrum conditions. However, all of these approaches assume that each node, initially has a set of available channels which are determined by sensing. None of these approaches considered the sensing overhead. Therefore, there is a gap between sensing and routing. An approach that can be adopted is to use the wireless spectrum databases [2] in case it is available. But it is not always applicable for all PUs. Another method is to use a sensor network that performs sensing [16]. However, this requires the deployment of a second network which is costly. Our approach is more dynamic and more practical and requires less overhead.

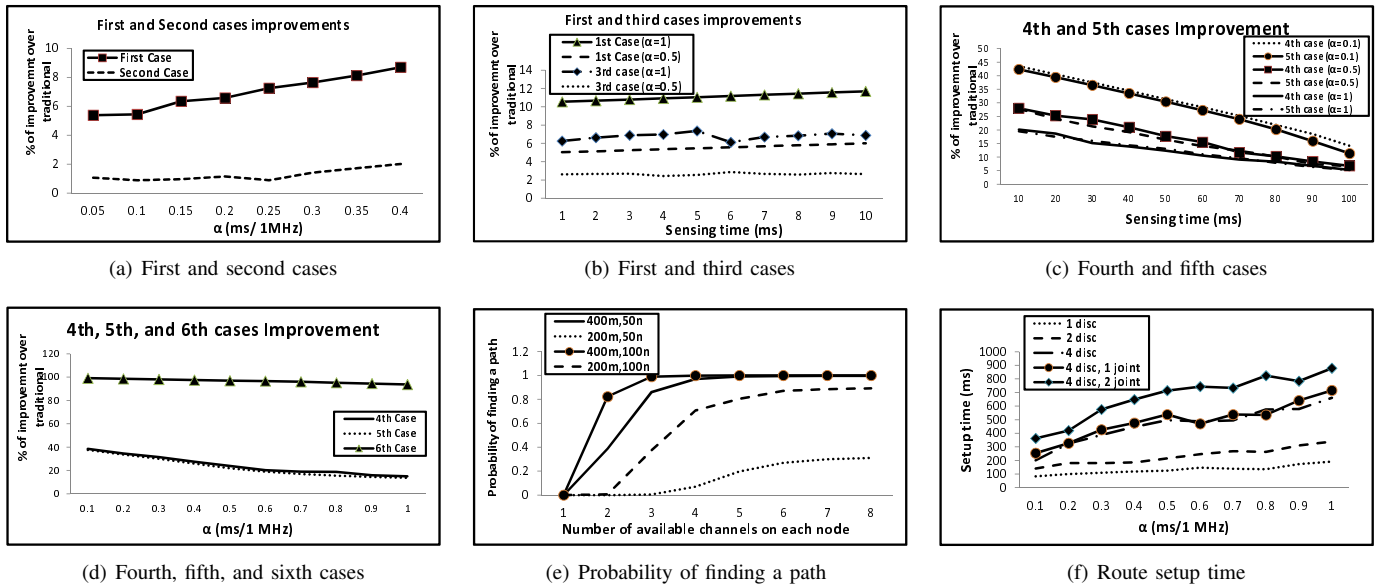


Fig. 6. Simulation results

VII. CONCLUSION

In this work, we introduced a spectrum decision framework that enhances existing routing protocols. In some cases, the achieved enhancement was as high as 100%. Also, existing routing protocols may not be able to find a path from the source to the destination because there may be no common available channel on one or more hops. The proposed framework was able to find a path with a short extra setup time. The framework concept can be summarized in allowing the CR node to inspect more channels by sensing them. The enhancement stems from finding a channel that requires less sensing time, less switching time, a channel that is shared by less number of nodes, or a channel at one node that will not interfere with other paths. Moreover, another enhancement results when the framework finds another node instead of a node that is intermediate on another path. The framework decides which channels to be sensed, on which nodes, and when it is efficient to sense them, taking into consideration the sensing time, channels switching time, PU expected available time, and the probability of the channel being idle.

REFERENCES

- [1] M. Cesana, F. Cuomo, and E. Ekici. Routing in cognitive radio networks: Challenges and solutions. *Ad Hoc Networks*, 9(3):228–248, 2011.
- [2] ET FCC. Docket no. 08-260., *Second Report and Order and Memorandum Opinion and Order*, Nov, 2008.
- [3] G. Cheng, W. Liu, Y. Li, and W. Cheng. Spectrum aware on-demand routing in cognitive radio networks. In *New Frontiers in Dynamic Spectrum Access Networks, 2007. DySPAN 2007. 2nd IEEE International Symposium on*, pages 571–574. IEEE, 2007.
- [4] Theodore Rappaport. *Wireless Communications: Principles and Practice*. Prentice Hall PTR, Upper Saddle River, NJ, USA, 2nd edition, 2001.
- [5] E. Dall’Anese, S.J. Kim, and G.B. Giannakis. Channel gain map tracking via distributed kriging. *Vehicular Technology, IEEE Transactions on*, 60(3):1205–1211, 2011.

- [6] S.M. Mishra, A. Sahai, and R.W. Brodersen. Cooperative sensing among cognitive radios. In *Communications, 2006. ICC’06. IEEE International Conference on*, volume 4, pages 1658–1663. Ieee, 2006.
- [7] I.F. Akyildiz, B.F. Lo, and R. Balakrishnan. Cooperative spectrum sensing in cognitive radio networks: A survey. *Physical Communication*, 2010.
- [8] A. Ghasemi and E.S. Sousa. Collaborative spectrum sensing for opportunistic access in fading environments. In *New Frontiers in Dynamic Spectrum Access Networks, 2005. DySPAN 2005. 2005 First IEEE International Symposium on*, pages 131–136. Ieee, 2005.
- [9] D. Cabric, S.M. Mishra, and R.W. Brodersen. Implementation issues in spectrum sensing for cognitive radios. In *Signals, Systems and Computers, 2004. Conference Record of the Thirty-Eighth Asilomar Conference on*, volume 1, pages 772–776. Ieee, 2004.
- [10] C. Xin, B. Xie, and C.C. Shen. A novel layered graph model for topology formation and routing in dynamic spectrum access networks. In *New Frontiers in Dynamic Spectrum Access Networks, 2005. DySPAN 2005. 2005 First IEEE International Symposium on*, pages 308–317. IEEE, 2005.
- [11] Y.T. Hou, Y. Shi, and H.D. Sherali. Optimal spectrum sharing for multi-hop software defined radio networks. In *INFOCOM 2007. 26th IEEE International Conference on Computer Communications. IEEE*, pages 1–9. IEEE, 2007.
- [12] I. Pefkianakis, S.H.Y. Wong, and S. Lu. Samer: spectrum aware mesh routing in cognitive radio networks. In *New Frontiers in Dynamic Spectrum Access Networks, 2008. DySPAN 2008. 3rd IEEE Symposium on*, pages 1–5. Ieee, 2008.
- [13] L. Ding, T. Melodia, S. Batalama, and M.J. Medley. Rosa: Distributed joint routing and dynamic spectrum allocation in cognitive radio ad hoc networks. In *Proceedings of the 12th ACM international conference on Modeling, analysis and simulation of wireless and mobile systems*, pages 13–20. ACM, 2009.
- [14] G. Cheng, W. Liu, Y. Li, and W. Cheng. Joint on-demand routing and spectrum assignment in cognitive radio networks. In *Communications, 2007. ICC’07. IEEE International Conference on*, pages 6499–6503. Ieee, 2007.
- [15] A. Sampath, L. Yang, L. Cao, H. Zheng, and B.Y. Zhao. High throughput spectrum-aware routing for cognitive radio networks. *Proc. of IEEE Crowncom*, 2008.
- [16] M.B.H. Weiss, S. Delaere, and W.H. Lehr. Sensing as a service: An exploration into practical implementations of dsa. In *New Frontiers in Dynamic Spectrum, 2010 IEEE Symposium on*, pages 1–8. IEEE, 2010.