Abstract—The emergence of cognitive radio network technology provides network nodes with abundant licensed and unlicensed channels to select for routing. Traditional routing methods that minimize hop count often do not leverage the full capabilities of cognitive radio users. They may select high lossy links or choose arbitrarily among the different paths of the same minimum length, regardless of the often large differences in throughput among those paths.

In this paper, we propose a routing algorithm that incorporates the expected transmission count metric (ETX) to select high-quality channels for routing on a hop-by-hop basis. The ETX metric incorporates the effects of link loss ratios, and interference among the successive links of a path, as well as interference caused by licensed users’ activity. Our routing algorithm utilizes all the available channels and selects channels with the best ETX value, allowing us to find paths with higher stability and higher throughput. Simulation in the NS-3 environment shows that our routing protocol has high packet delivery ratio, low end-to-end delay and low hop count in the presence of interference caused by licensed users.

Keywords—routing; ETX; cognitive radio; primary user; secondary user

I. INTRODUCTION

Cognitive radio networks are composed of primary and secondary users that share a common portion of the wireless spectrum. Primary users (PU) have exclusive access to the licensed spectrum. An example of a primary user may be a TV receiver in the TV bands [11]. Secondary users (SU) occupy a channel opportunistically and leave the channel either when they have finished transmission, or when they sense primary user activity [15]. Secondary users must avoid interfering with primary users as much as possible to prevent disruption to primary user traffic. In order to ensure this, cognitive radios must be aware of the characteristics of the spectrum in their environment [5].

The links made between secondary users provide the opportunity for cognitive routing. However, routing in this environment incurs additional challenges when compared to traditional networks. These challenges revolve around the responsibility of secondary users to evacuate licensed bands in response to primary (i.e. licensed) users [3]. This can make routing through the network using a single channel impractical if that channel is occupied by a primary user. In addition, each channel may have different qualities. Determining which channel has the best quality is also a challenge. Lastly, route maintenance in cognitive routing is more challenging than in the non-cognitive case [4]. Primary user activity may result in high latency overhead of intra-route spectrum diversity, which discovers only one route that can evolve through all the available channels. As opposed to inter-route diversity, which repairs a link failure by routing through a new path, intra-route diversity repairs a link failure locally by switching to a different channel on the same path, avoiding rerouting overhead. The only weakness of this approach is that channel switching does not happen until a link fails. Even when there are higher quality channels available, a node still uses the suboptimal channel to transmit the packet, resulting in low throughput.

Due to the low latency overhead of intra-route spectrum diversity, we follow the same approach in designing our cognitive routing algorithm. At the same time, we improve on intra-route diversity by allowing each hop to select the best-quality channel to forward the packet to the next hop. This enhances the overall throughput in our cognitive network. The metric we use to select the channel is based on the expected transmission count metric (ETX) [7]. ETX is valuable in wireless environments as it incorporates the effects of lossy links, interference caused by PU activity, and interference...
among the successive links of a path [13][8]. ETX optimizes for throughput as it minimizes the expected total number of packet transmissions required to successfully deliver a packet to the ultimate destination.

This paper makes two main contributions. First, it presents the design, implementation, and evaluation of our ETX computation protocol. Second, it describes our modification to the intra-route spectrum diversity algorithm to use ETX for channel selection. Our scheme is abbreviated as Cognitive Routing with ETX (CRE).

The remainder of this paper is organized as follows: Section II discusses the details of the ETX computation protocol and the cognitive routing algorithm that leverages ETX to select high-quality channels for routing. Section III presents the results from performance experiments. Section IV reviews the related work. Section V summarizes and concludes this paper.

II. Protocol Design

This section first derives the formula to compute the Expected Transmission Count (ETX) and discusses the important characteristics of ETX in finding paths on multi-hop wireless networks. It then describes the protocol to compute ETX. Finally, it presents the cognitive routing algorithm that incorporates the ETX metric.

A. ETX Metric

The ETX [7] of a link is the predicted number of data transmissions required to send a packet over that link, including retransmissions. The ETX of a link is calculated using the forward and reverse delivery ratios of the link. The forward delivery ratio, \( d_f \), is the measured probability that a data packet successfully arrives at the recipient. The reverse delivery ratio, \( d_r \), is the probability that the ACK packet is successfully received. These two probabilities need not be the same due to different conditions, or different transmission power at each end. The details of measuring these delivery ratios are presented in the next subsection.

The expected probability that a transmission is successfully received and acknowledged is \( d_f \times d_r \). A sender will retransmit a packet that is not successfully acknowledged. The probability of such an event is \( 1 - d_f \times d_r \). Thus, with probability \( d_f \times d_r \), we require 1 transmission. Otherwise, we require \( ETX + 1 \) transmissions to successfully deliver the packet. The expected number of transmissions is then computed as:

\[
ETX = (d_f \times d_r)(1) + (1 - d_f \times d_r)(ETX + 1)
\]

Thus, \( ETX = \frac{1}{d_f \times d_r} \)

ETX has several important characteristics: [7]
- ETX affects throughput as it is computed based on delivery ratios.
- ETX handles asymmetry as it incorporates loss ratios in both forward and reverse directions.
- ETX can select the best route and channel for transmission based on precise link loss ratio measurements on each channel.
- ETX penalizes long multi-hop paths as they suffer interference between successive hops of the same path, resulting in low throughput [10].

B. ETX Computation Protocol

This subsection begins by describing the protocol to compute ETX. Then, an example is given to show how the protocol works in a wireless network.

1) Protocol Description: Secondary users compute ETX by exchanging probe packets (ETX_REQ and ETX REP) with their neighbors. To facilitate ETX computation, each node maintains a window of time \( W \), which is divided into \( M \) equal-length intervals. ETX_REQ is broadcast at the beginning of each time interval. Pseudocode 1 shows the action upon receiving ETX_REQ and ETX REP.

Pseudocode 1  Handle ETX packet arrival

```
1: if packetType is ETX_REQ then
2: Prepare ETX REP packet
3: Piggyback local computed \( d_r \)
4: Unicast ETX REP to sender’s IP address
5: else if packetType is ETX REP then
6: if packet arrives within the time interval then
7: Increment REPcount on channel from receiver to sender
8: else
9: Discard the packet
10: end if
11: end if
```

Pseudocode 2  Compute \( d_f, d_r, \) ETX at the end of time interval

```
1: for each channel \( c \) do
2: \( d_f[c] = \frac{\text{avg(piggybacked } d_r \text{ on channel } c \text{ from all neighbors})}{\text{Theoretical#ofREPwithoutloss}} \)
3: for each neighbor \( n \) do
4: \( d_r[n, \text{on_channel}_c \text{ to } n] = \frac{\text{REPcount[on_channel}_c \text{ to } n]}{\text{Theoretical#ofREPwithoutloss}} \)
5: \( \text{ETX[on_channel}_c \text{ to } n] = d_f[c] \times d_r[n, \text{on_channel}_c \text{ to } n] \)
6: end for
7: end for
```

As mentioned in the previous subsection, the two parameters used to compute ETX is \( d_r \) and \( d_f \). \( d_r \) and \( d_f \) and ETX are computed at the end of each time interval as shown in Pseudocode 2. Note that as time progresses, eventually the time window \( W \) will shift. While the theoretical value of ETX_REQ count (the number of ETX_REQ packets a node should receive without loss) is capped at \( M \), the actual value of ETX_REQ count (REPcount) must be correctly adjusted. For example, consider when time window \( W \) shifts from interval \( t_0 \) to interval \( t_1 \). If there was an ETX_REQ packet arriving within interval \( t_0 \), REPcount is decremented by one. Otherwise, REPcount stays unchanged. The details of this update are not shown in Pseudocode 2.

2) An Example of ETX Computation: Fig. 1 shows our setup. The network consists of three nodes. Each node is
within the broadcast range of each other. For simplicity, we assume there is only one channel in this network, say channel 1. All nodes communicate on that same channel. Each node respectively computes the ETX on channel 1 to all of their neighbors. Without loss of generality, we focus on the ETX computation at node A. Consider the pattern of ETX REP packet lost and received in both directions between two pairs of nodes (B, A) and (C, A) in Table 1.

### Table 1. Pattern of ETX REP packet lost and received

<table>
<thead>
<tr>
<th>Nodes</th>
<th>$t_0$</th>
<th>$t_1$</th>
<th>$t_2$</th>
<th>$t_3$</th>
</tr>
</thead>
<tbody>
<tr>
<td>B→A</td>
<td>Received</td>
<td>Received</td>
<td>Lost</td>
<td>Received</td>
</tr>
<tr>
<td>C→A</td>
<td>Received</td>
<td>Received</td>
<td>Received</td>
<td>Received</td>
</tr>
</tbody>
</table>

Fig. 1. An example of a network setup to show ETX computation

Fig. 2 shows the ETX REP packets and the piggybacked $d_r$ values destined to A from B and C at each time interval. Table 2 shows the values of $d_r$, $d_f$, and ETX at node A. Note that during time interval $t_2$, the ETX REP packet from B to A gets lost. In this case, node A treats the piggybacked $d_r$ value from B to A as 0. As a result, $d_f$ equals to $\frac{0+1}{2} = \frac{1}{2}$. As it can be noticed from Table 2, the values of ETX start out as infinity since at time $t_0$ the initial piggybacked $d_r$ values from B and C are not yet available and are initialized to 0. At time $t_1$, ETX values converge to 1 as ETX REP already arrived at A within the time interval and the piggybacked $d_r$ values became available at the end of time $t_0$. At time $t_2$, since ETX REP packet from B to A on channel 1 gets lost, the ETX values become worse, particularly on channel 1 from A to B. Note that even though ETX REP packet from C to A successfully arrives at A, the ETX value from A to C on channel 1 slightly gets worse because the value of $d_f$ on that channel is affected due to the loss of ETX REP packet on channel 1 from B to A. At time $t_3$, since ETX REP packets successfully arrive at A from each neighbor of A, ETX values are improved.

### C. Cognitive Routing Algorithm

The cognitive routing algorithm in which ETX is implemented is a lightweight version of Ad-hoc On-Demand Distance Vector routing (AODV) [14]. The novelty of our design is that after the path is established, each node along the path has the flexibility to dynamically select the best channel for routing based on the locally-computed optimal ETX value. This is different from any existing works [2] in which they fit the channel to use for routing after the path is already established.

Our routing algorithm follows a pure on-demand route acquisition approach. Nodes that do not lie on active paths neither maintain any routing information nor participate in any periodic routing table exchanges. Further, a node does not have to discover and maintain a route to another node until the two need to communicate, unless the node is offering its service as an intermediate forwarding station to maintain connectivity between two other nodes.

1) Path Discovery: The Path Discovery process is initiated whenever a source node needs to communicate with another node for which it has no routing information in its table. The source node initiates path discovery by broadcasting a route request (RREQ) packet to its neighbors. The RREQ contains the following fields:

\[
< \text{source\_addr}, \text{dest\_addr}, \text{broadcast\_id} >
\]

The pair $< \text{source\_addr}, \text{dest\_addr}, \text{broadcast\_id} >$ uniquely identifies a RREQ. broadcast_id is incremented...
whenever the source issues a new RREQ. Each neighbor either satisfies the RREQ by sending a route reply (RREP) back to the source, or rebroadcasts the RREQ to its own neighbors. Notice that a node may receive multiple copies of the same route broadcast packet from various neighbors. When an intermediate node receives a RREQ, if it has already received a RREQ with the same `broadcast_id` and `source_addr`, it drops the redundant RREQ and does not rebroadcast it. Otherwise, it keeps track of the identity of the RREQ by storing the `source_addr` and the `broadcast_id` of the RREQ.

2) Reverse Path Setup: As the RREQ travels from a source to various destinations, it automatically sets up the reverse path from all nodes back to the source [6]. To set up a reverse path, a node records the address of the neighbor from which it received the first copy of the RREQ. As illustrated in Fig. 3, node A wants to communicate with node D. Node A broadcasts a RREQ. Upon receiving the RREQ from A, node B and E establish the reverse paths to A and rebroadcast the RREQ. Eventually, the reverse path is formed from node D to A. Notice that RREQ is broadcast on all channels regardless of the channel quality (the ETX value). The reason is because channel quality between a node and its neighbors may vary due to different regions of PU activity. For example, in Fig. 3, we might have PU activity in the far left that causes interference on channel 1 only from A to B, but not from A to E. Thus, if node A broadcasts RREQ only on channel 1, while the RREQ reaches node E, it may not reach node B. Hence, to ensure that RREQ reaches as many neighbors as possible, a node will broadcast RREQ on all channels.

3) Forward path Setup: Eventually, a RREQ will arrive at a node (possibly the destination itself) that possesses a current route to the destination. If the receiving node has a route entry for the desired destination, it unicasts a route reply packet (RREP) back to its neighbor from which it received the RREQ. A RREP contains the following information:

```
< source_addr, dest_addr, broadcast_id, 
  one_hop_receiver_addr >
```

By the time a RREQ packet arrives at a node that can supply a route to the destination, a reverse path has been established to the source of the RREQ (see previous subsection). As the RREP travels back to the source, each node along the path sets up a forward pointer to the node from which the RREP came. The field `one_hop_receiver_addr` specifies the address of the next node in the reverse path from the current node. This helps prevent packet flooding in the network. Only the node that has the IP address that matches the `one_hop_receiver_addr` will continue propagating the RREP along the reverse path toward the source node. Notice that each node along the reverse path encodes the next hop IP address in `one_hop_receiver_addr` before forwarding the RREP. The source node can begin data transmission as soon as the first RREP is received (forwarding path setup completes). Both RREP and DATA packets are transmitted on channels with the best ETX value.

Fig. 4 shows the forward path setup as the RREP travels from the destination D to the source node A. As it can be noticed from the figure, as node C receives the RREP from D, it checks its IP address against the `one_hop_receiver_addr`. Since the IP addresses match, it replaces the `one_hop_receiver_addr` with the IP address of its next hop on the reverse path (B) and propagates the packet to B. Eventually, the RREP arrives at node A and the forward path is established.

**III. EVALUATION**

In this section, we evaluate the performance of the ETX computation protocol and the proposed CRE routing scheme in a packet-level simulation. We first describe the simulation setup, followed by the metrics used and the results.

A. Simulation Setup

We performed our simulations using the NS-3.18 network simulator [1]. We conducted three sets of experiments. In the first set of experiments, we evaluate how effective is our ETX computation protocol in capturing channel interference. Fig. 5 shows our network topology setup for the experiment. The topology consists of four cognitive radio nodes (secondary user nodes) and two primary user (PU) nodes. Cognitive radio nodes are laid out on the same row. Node B sits in between two primary user nodes. Each node is only in the
transmission range of its immediate neighbors. For example, in Fig. 5, node C is in the transmission range of node B and D, but out of the transmission range of node A. Finally, we use three channels. Each channel is modeled as a separate network interface in NS-3. This is equivalent to assigning cognitive/secondary users three separate radios, one for each channel. For PU’s transmission, we dedicate channel 1 to the top PU and channel 2 to the bottom PU. Both PUs are active and periodically broadcast HELLO messages on their own dedicated channels. We performed 20 runs. The duration of each run is 100 seconds. For each run, we recorded the ETX values computed for the last 10 intervals. The length of each interval is set to 1 second. We then averaged the results over 20 runs.

![Network topology setup for the first set of experiments](image)

Fig. 5. Network topology setup for the first set of experiments

In the second set of experiments, we evaluate the performance of our proposed CRE scheme. We use two channels. The simulated network consists of 49 nodes, laid out in a 7x7 grid. Nodes move according to the random waypoint model in a square area. We have 10 random nodes sending data packets to 10 other nodes in the network which trigger the route establishment process. We vary the number of primary users. The primary users are selected out of 49 nodes, and they periodically broadcast HELLO messages on the same channel.

In the last set of experiments, we follow similar setup as in the second experiments, but we let primary users periodically switch channels to broadcast HELLO messages, resulting in both channels unavailable at a certain time period between certain nodes.

B. Evaluation Metrics

We compare our proposed CRE scheme against standard AODV. The distinguished difference between standard AODV and our CRE scheme is that AODV is unaware of channel availability. In the case of heavy interference on the established channel, AODV can suffer from significant packet loss along its routing path. We use the following metrics in the experiments:

1) End-to-end delay: the average time it takes a data packet to reach the destination node.

2) Packet delivery ratio: the ratio of data packets successfully received by the destination node to the total number of data packets generated by the source node.

3) Hop count: the average number of hops that the data packets arrive at the destination node.

C. Experimental Results

First, we show the effectiveness of ETX in capturing interference and identifying available channels. Then, we show the performance of our CRE scheme.

1) Effectiveness of ETX: Fig. 6 shows the ETX values computed for each channel in both forward and reverse directions. The ideal value for ETX is 1, which indicates that there are no expected retransmissions for a particular channel and that the channel quality is high. The ETX values are significantly high on channel 1 and 2 from A to B and from C to B. This is because, due to the interference caused by the top and bottom PUs, node B is unable to receive the ETX_REQ sent from A and C. As a result, B does not send ETX_REP to A and C, resulting in high ETX values from A and C to B. For the same reason, B cannot receive the ETX_REP sent from A and C, resulting in high ETX values on channel 1 and 2 from B to A and C. ETX values on channel 3 remain low (close to 1) due to the absence of PU activity on this channel. Finally, ETX values are uniformly low on all three channels between C and D. This is because node C is not in the transmission range of the top and bottom PUs. Therefore, node C is able to receive ETX_REQ and ETX_REP from D, resulting in low ETX values in both directions between C and D.

2) CRE Routing Performance: Fig. 7 and Fig. 8 show the performance of the standard AODV and our proposed CRE across three evaluation metrics. As the number of primary users increase, CRE has significantly lower end-to-end delay, hop count and higher packet delivery ratio. The reason is because the presence of primary users cause interference on the channel which AODV uses to route packets. Since AODV lacks the ability to detect other channel availability, it will suffer packet loss on the busy channel, resulting in lower packet delivery ratio. Furthermore, since the shortest path between a source node and a destination node may be impeded by PUs’ interference, AODV routing has to take longer routing paths, resulting in high end-to-end delay and hop count. Our CRE scheme is robust to PUs’ interference. By using ETX, CRE can detect interference on the busy channel and it subsequently switches to an available channel to route the data packets. As a result, CRE successfully delivers packets using the shortest paths, resulting in low end-to-end-delay, low hop count, and high packet delivery ratio.

In Fig. 8, as PUs can now affect both channels, the performance gap between AODV and CRE is slightly decreased. As two channels are occupied, CRE may have to route through non-shortest paths, resulting in higher end-to-end delay and average hop count. Furthermore, since PUs randomly switch to operate on different channels, certain segments along the established path may become unavailable on both channels, resulting in data packets unable to arrive at the destination.
Thus, CRE's delivery ratio starts deviating from 1. On the other hand, the performance of AODV is improved since the channel AODV is operating on becomes less congested (some PUs switch to a different channel).

IV. RELATED WORK

Cognitive radio networks provide radio devices the ability to opportunistically access licensed bands or efficiently utilize unlicensed bands. CoRoute [9] proposes a geographic routing scheme for vehicular cognitive adhoc networks that allows vehicles to efficiently access WiFi channels to increase the network throughput. Vehicles are considered secondary users in the presence of roadside access points. SUs use the Expected Transmission Time (ETT) and Expected Transmission Count (ETX) metrics to estimate the quality of various channels, and share the results among other SUs to determine the optimal channel for operation. Vehicles are chosen to participate in the forwarding set if they offer the best progress toward the destination and minimize the ETT.

The work in SEARCH [5] uses geographic forwarding in mobile cognitive networks to jointly select a channel and a route that avoid regions of PU activity. The goal is to form paths with low hop counts between a source and a destination by taking advantage of the numerous paths that are available between the two. Routes have the ability to change dynamically according to node mobility and PU activity, while the routes with the lower hop counts are preferred.

Cheng [4] proposed a cognitive on-demand routing protocol that selects a minimum delay path according to the switching and backoff delay experienced at each node. In [12], spectrum sensing and routing are paired in order to allow users to utilize geographic information and learn about spectrum access opportunities at each hop to improve transmission performance. The authors introduce a new metric Cognitive Transport Throughput (CTT) that reflects the relay advantage of each relay candidate by using the expected bit progress per second for one hop relay of a packet according to packet size and channel quality.

V. SUMMARY AND CONCLUSION

We have proposed to use ETX as a metric for multi-hop wireless cognitive network. Channel selection using ETX accounts for link loss ratios, the asymmetry of the loss ratios in the two directions of each link, and the reduction of throughput due to interference among the successive hops of a route and interference caused by PU activity. We have described a protocol to compute ETX. The ETX computation protocol is based on successive exchange of probe packets. The higher the number of probe packets successfully received and acknowledged, the lower the ETX value. We have presented CRE, a cognitive routing algorithm that uses ETX to select high-quality channels for routing. The experimental results show that our routing protocol has high packet delivery ratio, low end-to-end delay and low hop count in the presence of interference caused by primary users.

REFERENCES

(a) ETX value on channel A to B
(b) ETX value on channel B to C
(c) ETX value on channel C to D
(d) ETX value on channel B to A
(e) ETX value on channel C to B
(f) ETX value on channel D to C

Fig. 6. ETX values in the case of interference caused by the top PU on channel 1 and the bottom PU on channel 2

(a) End-to-end delay
(b) Packet delivery ratio
(c) Hop count

Fig. 7. CRE vs AODV when PUs transmit on the same channel

(a) End-to-end delay
(b) Packet delivery ratio
(c) Hop count

Fig. 8. CRE vs AODV when PUs periodically switch and transmit on different channels