Performance Comparison of Routing Protocols for Cognitive Radio Networks

Li Sun, Wei Zheng, Naveen Rawat, Vikramsinh Sawant, Dimitrios Koutsonikolas
University at Buffalo, SUNY, Buffalo, NY, 14260-2500
Email: {*lsun3, wzheng4, naveenra, vikramsi, dimitrio*}*@buffalo.edu*

***Abstract*—Cognitive radio networks (CRNs) have emerged as a promising solution to the ever-growing demand for additional spectrum resources and more efficient spectrum utilization. A large number of routing protocols for CRNs have been proposed recently, each based on different design goals, and evaluated in different scenarios, under different assumptions. However, little is known about the relative performance of all these protocols, let alone the tradeoffs among their different design goals.**

In this paper, we conduct the first detailed, empirical performance comparison of three representative routing protocols for CRNs, under the same realistic set of assumptions. Our extensive simulation study shows that the performance of routing protocols in CRNs is affected by a number of factors, in addition to PU activity, some of which have been largely ignored by the majority of previous works. We find that different protocols perform well under different scenarios, and investigate the causes of the observed performance. Furthermore, we present a generic software architecture for the experimental evaluation of CRN routing protocols on a testbed based on the USRP2 platform, and compare the performance of two protocols on a 6 node testbed. The testbed results confirm the findings of our simulation study.

I. Introduction

The continuously increasing number of WiFi devices has resulted in growing congestion in the crowded ISM bands, putting a potential limit on the evolution of WiFi networking. On the other hand, some licensed bands, e.g., TV broadcast frequencies, remain largely underutilized. In order to satisfy the ever-growing public demand for additional spectrum resources, in November 2008 the FCC issued a ruling permitting unlicensed users (secondary users, SUs) to operate in the so-called white spaces, i.e., unused portions of the TV broadcast frequency band, as long as they do not interfere with licensed users (primary users, PUs). This ruling marks the arrival of cognitive radio networks (CRNs).

In CRNs, SUs have the ability to sense a wide spectrum range, dynamically identify currently unoccupied spectrum blocks, and choose the best available block to transmit, ensuring non-interfering coexistence with PUs [1]. While research on CRNs was initially focused on PHY/MAC layer issues (e.g., [2], [3], [4], [5]), soon the research community realized the great potential of multihop CRNs. By exploiting the unoccupied frequency resources, the cognitive radio technology is expected to largely increase the capacity of multihop wireless networks [6].

However, the unique characteristics of the white spaces, i.e., spatial variation, spectrum fragmentation, and temporal variation [7], make multihop CRNs very different from multihop networks in the ISM band. While in traditional wireless mesh networks (WMNs) the main task of a routing protocol is to discover routes of high quality links, in multihop CRNS the main task changes to ensuring radio resources for SU transmissions while guaranteeing the service for all ongoing PU communications [8]. To fulfill this task, routing in CRNs has to address a number of challenges, including adapting to dynamic changes of spectrum availability, the heterogeneity of resources such as the availability of different channels and radios on the same node, and synchronization between nodes on different channels [9]. Therefore, designing routing protocols for CRNs is a more challenging task than for networks in the ISM bands.

Recently, numerous routing protocols for CRNs have been proposed (e.g., [6], [10], [11], [12], [13], [14], [15], [16], [17], [18], [19], [20], [21], [22]). Besides the main goal of protecting PU transmissions, each protocol is proposed based on different design goals, e.g., maximizing spectrum opportunities, maximizing available bandwidth, minimizing hopcount, minimizing end-to-end delay, etc. The performance of each protocol is evaluated with respect to its specific design goals and sometimes compared against a baseline protocol (e.g., random routing). Moreover, each protocol is evaluated using a different evaluation methodology – different assumptions (e.g., about PU activity), settings, and scenarios, tailored to its specific design goals. Although each methodology offers a deeper understanding of a specific protocol, little is known about the relative performance of all these protocols, let alone the tradeoffs among their different design goals. While extensive performance comparisons have been conducted for multihop routing protocols in the ISM band (e.g., for MANETs [23], [24] or WMNs [25]), almost a decade since the first CRN routing protocol was proposed, there has been (to our best knowledge) no extensive performance comparison of routing protocols for CRNs.

In this paper, we conduct the first extensive empirical performance study of routing protocols for CRNs using both a simulator (ns-2) and a testbed based on the USRP2 platform [26], under the same realistic set of assumptions:

1) SUs have no knowledge about PU activity models and parameters; 2) each SU senses PU activities independently and periodically and learns PU activities online; 3) PUs can interrupt SU communications at any time; 4) the only way for two SUs to learn information about each other (e.g., observation of PU activities in each other's neighborhood) is through communication.

In the simulator, we implement and compare three representative routing protocols for CRNs – Coolest Path [12], SAMER [10], and CRP [13], each with different design objectives. Coolest Path aims to find the path with the highest spectrum availability, which results in path stability. SAMER tries to find the path with the highest throughput by taking into account both PU and SU activities, as well as link quality. CRP is designed either to find a path with minimum end-to-end delay and satisfactory PU protection or to offer the best protection to PU receivers at the cost of some performance degradation for SUs. Furthermore, each protocol takes a different approach with respect to three basic building blocks of a CRN protocol, including 1) how to characterize spectrum opportunities between neighboring SUs, 2) how to define a link metric based on spectrum opportunities and 3) how to select a routing path based on the link metric. Our study reveals the pros and cons of each approach as well as the tradeoffs among the three different design goals in a variety of scenarios.

Our main findings are summarized as follows: 1) Under low PU activity, path stability is not the only factor that affects the performance of a CRN protocol; factors considered by WMN routing protocols, such as link quality and interference among neighboring nodes, should also be taken into account. In such scenarios, SAMER outperforms the other two protocols in terms of both throughput and end-to-end delay. In the presence of multiple flows, SAMER improves the total throughput at the cost of a small reduction in fairness, as it tries to choose disjoint paths for different flows. 2) Under high PU activity, path stability and path length become the dominant factors that affect performance. In such scenarios, Coolest Path with an additive path metric outperforms the other protocols. 3) When the link routing metric ignores link quality, an additive path metric in general performs better than a bottleneck metric, as it limits the path length. In contrast, when link quality is taken into account, longer paths often yield better performance; a similar observation has been made for routing metrics proposed for WMNs, e.g., ETX [27] or ETT [28]. 4) It is important to consider neighbor observations in estimating spectrum opportunities, due to spatial variation in PU activities [7]. CRP often performs poorly, because it estimates spectrum opportunities based only on local observation.

Furthermore, we develop a generic software architecture for the experimental evaluation of CRN routing protocols on a testbed based on the USRP2 platform. Our framework provides an implementation of PHY, MAC, and network layers, which can be used as the basic building blocks for the implementation of any routing protocol. Basic CRN functions such as PU activities, SU periodic sensing, and channel switching capabilities are also supported. Neighboring SUs use a common control channel for exchanging control messages in a distributed way. Based on this architecture, we implement and compare Coolest Path and SAMER on a 6 node testbed. Our testbed results agree with the findings of our simulation study. In spite of the small size of the testbed, we note that this is the first (to our best knowledge) testbed-based performance comparison of two CRN routing protocols.

The rest of the paper is organized as follows. Section II briefly reviews the three CRN protocols we study in this paper. Section III describes our simulation methodology. In Section IV, we present our simulation study. In Section V, we describe the testbed architecture and the experimental evaluation. Section VI discusses the related work. Finally, Section VII concludes the paper.

II. Background

In this section, we provide a brief review of the three routing protocols we consider in our study.

In **Coolest Path** [12], a channel's temperature for an SU link is defined as the fraction of time during which the channel is unavailable due to PU activity in the neighborhood of any of the two SUs. The link's temperature is then defined as the minimum channel temperature among all available channels between the two SUs. Coolest Path provides three different definitions of the path temperature based on the link temperature: (i) *accumulated temperature*, i.e., the sum of the link temperatures along the path, (ii) *highest temperature*, i.e., the maximum link temperature among the links along the path, and (iii) mixed temperature – a combination of the first two. The protocol selects the path with the minimum path temperature. In [12], the performance of mixed temperature was always found to lie between the performance of the other two path metrics. For this reason, we do not consider mixed temperature in our study.

SAMER [10] tries to find a high-throughput path by opportunistically utilizing high-throughput links while still guaranteeing a path's long-term stability. To quantify channel availability, SAMER considers both PU and SU activity. Each SU estimates the fraction of time during which a channel can be used, i.e., it is not used by any PU and any other SU. Since two neighboring nodes may estimate different channel availabilities, the channel availability for a link is given by the smallest of the two values. SAMER's link metric is based on ETT [28], one of the most popular routing metrics for traditional WMNs. For each channel, SAMER estimates the expected throughput as the product of channel availability, link bandwidth, and loss rate. The link metric is then defined as the sum of throughput values of

Table I
QUALITATIVE COMPARISON AMONG THE THREE PROTOCOLS CONSIDERED IN THIS STUDY.

Protocol	Node Channel Availability	Link Channel Availability	Link Metric	Path Metric
Coolest Path	Based on PU activity	Product of channel availabilities observed by two neighbors	Minimum of all available channel temperatures	Accumulated or maximum or mixed link metric values
SAMER	Based on PU and SU activity	Minimum channel availability among two neighbors	Sum of all available channel throughputs	Minimum link metric value
CRP	Based on PU activity	Channel availability observed locally. Neighbor's channel availability is ignored.	Cost function reflecting delay or protection to PU receivers	Accumulated link metric values

all available channels. Hence, different from Coolest Path's link temperature, which reflects only a link's stability, the link metric in SAMER reflects both link stability (channel availability) and link quality (bandwidth, loss rate). The path metric in SAMER is the minimum throughput among all links along a path, i.e., a bottleneck metric.

CRP [13] considers two different routing classes that offer different levels of protection to PUs. Class I aims to minimize the end-to-end delay while still providing satisfactory protection to PUs. On the other hand, Class II allows a level of performance degradation and prioritizes PU protection by selecting as relays SUs that are far from PU receivers. Since in this study we focus on performance, we only consider Class I routes. In CRP, when an SU receives a route request, it selects a rebroadcast delay by calculating a cost function based only on local information. The cost function considers the SU's estimates of channel availabilities, variance of intensities of PU activities, etc. An SU with a lower cost (e.g., with higher channel availability) will rebroadcast the route request earlier. When the destination SU receives a route request, it simply sends a route reply back along the path over which it received the route request, without performing any local computation. Based on this cost-delay mapping, CRP can be easily implemented via minor modifications to AODV [29].

Table I summarizes the differences among the three protocols in the estimation of (i) channel availability for a node or a link, (ii) link metric, and (iii) path metric.

III. SIMULATION METHODOLOGY

For our simulation study, we adopted the ns-2 extended framework proposed in [30], which implements all necessary components for SUs in a CRN [31]: a spectrum sensing block (for detecting PU activity), a spectrum mobility block (for performing spectrum handoff after detecting a PU on the current channel), a spectrum decision block (for channel selection), and a spectrum sharing block (for allowing SUs to share the spectrum and avoid collisions through carrier sensing). A sensing-transmission cycle is also implemented in this framework. Similar to [13], [30], each SU is equipped with one receiving interface for receiving data packets and sensing the spectrum and one transmitting interface for sending data packets. There is also a third interface fixed on the control channel and used only for transmitting/receiving control packets, e.g., route requests/replies and channel switching notifications.

In our simulations, we use a 1-sec sensing-transmission cycle; SUs sense the spectrum in the first 0.1 sec (sensing period) and use the remaining 0.9 sec (data transmission period) to send/receive data. We assume the sensing periods on all SUs are synchronized according to the 802.22 standard [32]. We also assume SUs switch to a new channel rather than waiting on the previous one when they detect PU activity on the current channel. Although an SU is not able to detect PU activity during the data transmission period, our choice of the sensing cycle parameters guarantees that an SU will vacate a channel used by a PU in less that 2.0 sec, which still meets the requirement of the 802.22 standard [32].

In [30], PU activities follow an exponential ON-OFF model proposed in [2]. We use the same model in our study. In the ON-OFF model, the ON state represents the time interval during which a channel is occupied by a PU and the OFF state represents the interval during which a PU is idle and the channel can be used by SUs. Each PU switches between the ON and the OFF state. Although SUs cannot detect PU activity during the data transmission period, we model the impact of PU activity on SU transmissions, by assuming a 20% packet loss probability (due to collision) if a PU is active during an SU transmission.

At the PHY layer, we assume a spectrum band of 11 orthogonal channels with the propagation characteristics of 2.4 GHz. Ten of these channels can be used for data transmissions and one is used as the common control channel. Each interface can be tuned to one channel at a time. The bandwidth of each channel is 6 MHz, which is the same as a TV channel in the UHF band. We assume SU transmission power is the same in every channel. To simulate different channel qualities, we assume the packet loss ratio for every channel follows a uniform distribution between 0 and a maximum loss ratio.

At the MAC layer, we use 802.11b and disable RTS/CTS. To reflect a channel width of 6 MHz (instead of 20 MHz used in 802.11), we scale down the 802.11b data rates by a factor of 6MHz/20MHz. In our simulations, SUs use the highest data rate of 3.3 Mbps (11 Mbps in 802.11b) to transmit data packets and the basic data rate of 0.3 Mbps (1 Mbps in 802.11b) to broadcast control packets.

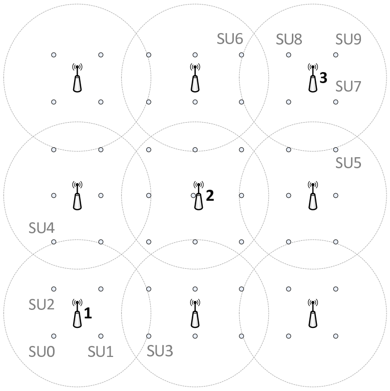


Figure 1. Simulation topology.

At the network layer, we modified AODV to support the three CRN routing protocols. When a route discovery is initiated, a RREQ packet is created at the source and is flooded towards the destination. When a node broadcasts a RREQ, it appends the link metric for the link through which it received the RREQ. The destination receives a number of RREQs over different paths, chooses the least cost path according to the path metric used by the routing protocol, and sends a RREP packet back to the source along the chosen path. Neighboring SUs on the chosen path will select the best channel among all available ones according to the link metric. When the selected channel can no longer be used because of PU activities, i.e, *the link breaks*, the sender-receiver pair will try to repair the link locally by selecting another channel, which is the best among the currently available channels according to the link metric. When no channel is available, the link cannot be used anymore, i.e., *the route breaks*, and a RERR packet is forwarded along the routing path. When the RERR packet arrives at the source node, a new route discovery is initiated by the source node, which buffers packets during this process.

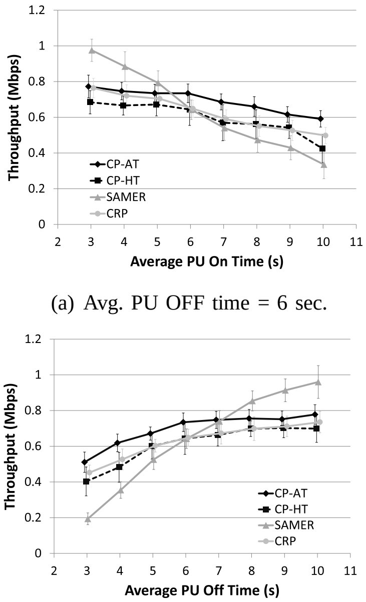
IV. Simulation Study

In this section, we first introduce our simulation setup and then discuss the simulation results.

A. Simulation Setup

We use the topology shown in Figure 1, which is similar to the ones used in [6] and [13]. A square region of side 1200 m is divided into 9 square cells of side 400 m. There are 9 PU locations in the centers of the cells. In each location, there are 10 PUs, operating on the 10 channels which can be used for data transmissions; there is no PU operating on the common control channel. Each PU has an interference range of 250 m. 49 SUs are placed in a grid format; the distance between any two neighboring SUs is 160 m. Each SU has a maximum transmission range of 250 m on each channel. We use SU0 in cell 1 as the source node and SU9 in cell 3 as the destination node, unless otherwise stated.

The default maximum packet loss ratio is set to 0.2. Each simulation runs for 900 sec, during which PUs may become active at any time. In the first 600 sec, SUs only sense the spectrum (during the sensing periods) to learn the statistics of PU activities in their neighborhood. The source node starts transmitting data packets at the 600th second and the data transmission period lasts for 300 sec, during which SUs keep sensing the spectrum and updating their observations of PU activities. Each source sends CBR data traffic over UDP at a rate of 3.3 Mbps with a packet size of 1500 Bytes.



(a) Avg. PU OFF time = 6 sec.

(b) Avg. PU ON time = 6 sec.

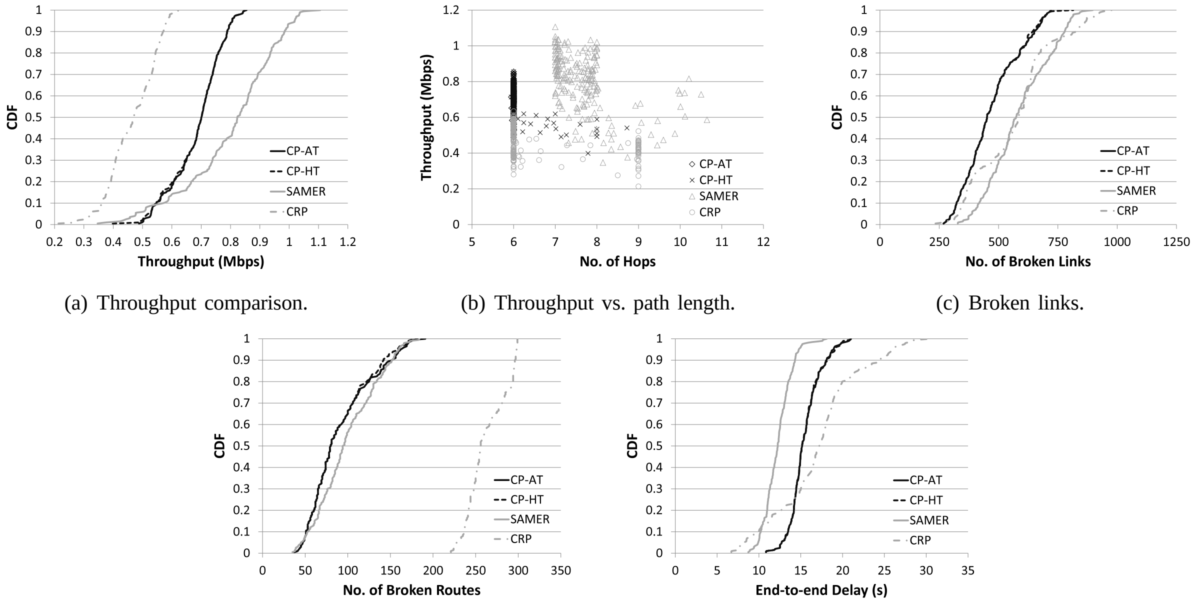
Figure 2. Baseline throughput comparison. For better clarity, the data points for CP-AT, CP-HT, and CRP are shifted horizontally.

We compare the performance of the three routing protocols based on four metrics: throughput, number of broken links, number of broken routes, and average end-to-end delay of successfully delivered packets at the application layer. For scenarios with multiple flows (Section IV-F), we also evaluate fairness using Jain's Fairness Index [33]. For Coolest Path, we consider two versions: Coolest Path with Accumulated Temperature (CP-AT) and with Highest Temperature (CP-HT).

B. Baseline Scenario

We first compare the throughput of the four routing protocols in a baseline scenario, in which we assume all PUs have the same average ON and OFF times. For each combination of average ON/OFF times, we repeat the simulation 20 times, using each time a different seed to generate PU activities, i.e., ON/OFF intervals following an exponential distribution.

In Figure 2(a), we fix the average PU OFF time at 6 sec and vary the average ON time from 3-10 sec. Each point corresponds to the average throughput over 20 simulation runs and the error bars correspond to the standard deviations. We observe that the throughput of all four routing protocols



(a) Throughput comparison.

(b) Throughput vs. path length.

(c) Broken links.

(d) Broken routes.

(e) End-to-end delay comparison.

Figure 3. Localized PU activities.

drops when the intensity of PU activities increases. In this scenario, we find CP-AT always performs better than CP-HT while the performance of CRP lies between CP-AT and CP-HT. Interestingly, SAMER outperforms the other routing protocols when the average ON time is smaller than 6 sec and performs the worst when the average ON time is larger than 6 sec.

In Figure 2(b), we fix the average PU ON time at 6 sec and increase the average OFF time from 3-10 sec. We observe that the relative performance of the four protocols is the same as in Figure 2(a). Again, CP-AT outperforms CRP and CP-HT while SAMER performs the best when the intensity of PU activities is low and performs the worst when the intensity of PU activities is high.

Since the performance trend is the same when we vary either the OFF or the ON time, for simplicity, in the rest of the paper we always fix the average PU ON time at 6 sec and vary the average PU OFF time. In IV-C, IV-D, we study the performance of the four protocols in more complex scenarios with respect to PU activities, and we investigate the causes of the observed performance. In IV-E, we study the impact of the channel loss ratio. Finally, in IV-F, we study the performance in the presence of multiple flows.

C. Localized PU Activities

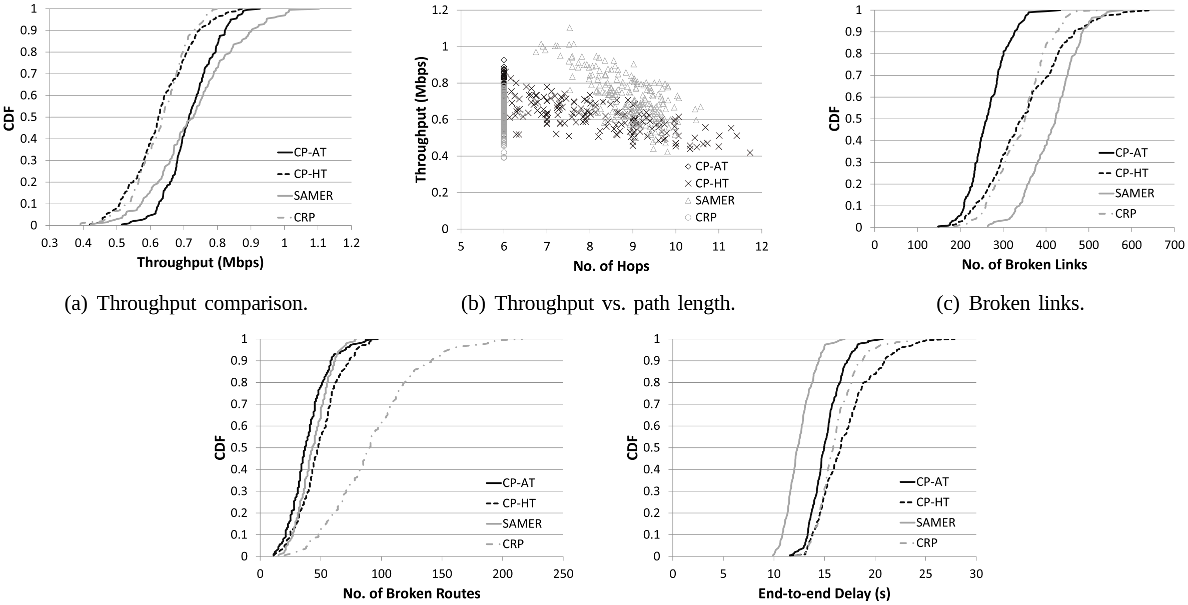
We now compare the four protocols in a more realistic scenario where PU activities vary in different locations. In [7], the authors point out that rural and suburban regions exhibit a much lower degree of spectrum fragmentation and more contiguous spectrum than urban areas. To simulate this scenario, we use different average OFF time values for PUs in different cells. In each cell, all PUs are assigned the same average OFF time, chosen uniformly from the interval 2-11 sec. We use 200 different seeds to select average OFF times and generate PU activities.

Throughput comparison Figure 3(a) plots the Cumulative Distribution Function (CDF) of the 200 throughput values for each protocol. In general, SAMER performs the best and CRP performs the worst among the four protocols while the performance of CP-HT is very close to the performance of CP-AT. In the median case, SAMER outperforms CP-AT by 17.56% and CRP by 75.46%.

Path lengths Figure 3(b) plots a scatterplot of the throughput against the average routing path length for each of the 200 simulation runs. We observe that, in general, SAMER chooses longer routes than the other protocols. There are two reasons for this behavior. First, SAMER's link metric considers both spectrum opportunities and link qualities. Hence, the protocol often prefers longer paths consisting of higher quality links, similar to link quality-based routing metrics for traditional WMNs, e.g., ETX or ETT. Second, SAMER's path routing metric is a bottleneck metric; hence, the route selection does not take path length into account.

CP-HT's path metric is also a bottleneck metric, but its link metric considers only PU activities. As a result, in most cases it still selects the shortest path (6 hops, through cells 1, 2, 3 in Figure 1). On the other hand, CP-AT and CRP tend to choose shorter paths because of their additive path metrics. However, CRP sets a lower limit on spectrum availabilities and an upper limit on spectrum availability variance, and, in some cases, it prefers a longer path, when nodes on the shortest path do not satisfy these constraints.

In spite of choosing longer paths, SAMER still achieves



(a) Throughput comparison.

(b) Throughput vs. path length.

(c) Broken links.

(d) Broken routes.

(e) End-to-end delay.

Figure 4. Random PU activities – Avg. PU OFF time: 2-11 sec.

the highest throughput among the four protocols. This is because SAMER's link metric also takes contention among SUs into account, i.e., it tries to avoid assigning the same channel to two nodes in the same interference range. More importantly, SAMER is also able to find more stable paths. Unlike Coolest Path, which uses the best channel to calculate link metric values, SAMER considers the potential throughput values over all possible channels and avoids SUs with only one good channel. As a result, the protocol selects relays in cells with lower PU activity gaining advantage in a scenario where the intensity of PU activities varies per cell. This advantage is lost when all cells exhibit high PU activity (Figures 2(a) and 2(b) in the baseline scenario).

Broken links/routes Figures 3(c) and 3(d) plot the CDFs of the number of broken links and broken routes, respectively, for each protocol. We observe that CP-AT and CP-HT experience the smallest number of broken links and broken routes; this demonstrates the protocol's effectiveness in finding stable paths, and is consistent with the protocol's design goal. However, as Figure 3(a) shows, path stability is not the only factor that affects throughout in a CRN; traditional factors such as link quality and interference from neighboring nodes should also be taken into account.

SAMER experiences more broken links than Coolest Path (27% more in the median case), however, many of the link breaks do not result in route breaks – the median number of broken routes is 94 for SAMER and 80 for Coolest Path. By considering all possible channels in the relay selection, SAMER is able to select relays with many available channels and is often able to choose a new channel when a link breaks (due to PU appearance on the current channel) without resorting to an end-to-end route discovery.

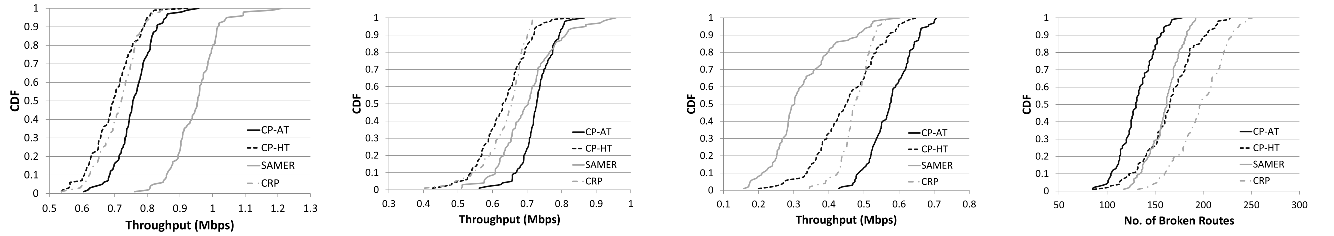
CRP experiences the largest number of broken links and broken routes. This is due to the fact that CRP estimates statistics of PU activities using only local observations and ignores PU activities on the other end of the link. Although CRP sets a threshold on channel availability locally, it is not guaranteed that a link can be indeed utilized with a probability that satisfies that threshold when neighboring SUs are impacted by PUs in different locations. Consequently, the protocol suffers from a large number of broken routes, which result in low throughput, as shown in Figure 3(a).[1]

End-to-end delay Figure 3(e) plots the CDFs of the end-to-end delay for each protocol. We note that we used a large buffer size at the link layer in the simulator in order to maximize throughput, which resulted in long end-to-end delays. We observe that SAMER achieves again the best performance, simultaneously optimizing throughput and delay. We also observe that CRP performs worse than Coolest Path although it is designed to minimize end-to-end delay. The reason is again the large number of broken routes. The source node has to buffer packets and wait for a new route discovery every time the route breaks.

D. Random PU Activities

In this section, we simulate a scenario in which the average OFF time for each PU is chosen uniformly from the interval 2-11 sec, independent of its location. Compared to the scenario in IV-C, this scenario is characterized by

[1] A unique feature in the design of CRP, which we did not consider in this study, is that the protocol prefers channels with longer transmission range. Hence, its relative performance may improve if the available channels are distributed over a large frequency band with varying spectrum propagation characteristics, as in [13].



(a) Throughput comparison (Avg. PU OFF time: 8-11 sec).

(b) Throughput comparison (Avg. PU OFF time: 5-8 sec).

(c) Throughput comparison (Avg. PU OFF time: 2-5 sec).

(d) Broken routes (Avg. PU OFF time: 2-5 sec).

Figure 5. Random PU activities – Varying PU OFF time.

more diversity in terms of PU activities. Similar to IV-C, we use 200 different seeds to select average OFF times and generate PU activities.

Throughput comparison Figure 4(a) plots the throughput CDFs for each protocol and Figure 4(b) plots the throughput against the average path length for each run. We observe that in this scenario CP-AT clearly outperforms CP-HT. The reason is the difference in the path lengths. The median path length for CP-HT is 7.74 hops while CP-AT chooses the shortest path (6 hops) in almost all cases. In contrast, in the scenario of Section IV-C, CP-HT chose the same routing paths as CP-AT in most cases (Figure 3(b)). We also observe that SAMER and CP-AT outperform CRP and CP-HT. However, in contrast to the localized PU activity scenario in IV-C, there is no clear winner in this scenario; the median throughput is almost the same for both protocols. CP-AT outperforms SAMER in half of the simulation runs (those yielding throughputs lower than 0.7 Mbps) and SAMER outperforms CP-AT in the other half.

Broken links/routes We plot the CDFs of the number of broken links in Figure 4(c), and the CDFs of the number of broken routes in Figure 4(d). We observe that SAMER again has the largest number of broken links and CRP has the largest number of broken routes, similar to in Section IV-C. CP-AT is the most stable protocol, experiencing the smallest number of both broken links and routes. CP-HT has a much smaller number of broken routes compared to CRP but it appears to be more sensitive to broken routes – even a small number of broken routes results in low throughput, as we saw in Figure 4(a). The reason is the long routing paths which result in high packet loss in case of route breaks, since there are more packets buffered along a long path.

End-to-end delay Figure 4(e) plots the CDFs of the end-to-end delay for each protocol. SAMER again performs the best and CP-AT outperforms CRP. However, different from Section IV-C, CRP performs better than CP-HT. This is because the path lengths of CP-HT are much longer in this scenario than in the previous one.

Varying PU activity To study the performance of SAMER and Coolest Path in more detail, we divide the average PU OFF time into three smaller ranges – 2-5 sec, 5-8 sec, and 8-11 sec. For each range, we repeated the simulation with 100 different seeds. In Figures 5(a), 5(b), and 5(c), we plot the throughput CDFs for the three ranges. We observe that the performance of SAMER drops as the average PU OFF time decreases - SAMER performs the best in the range of 8-11 sec and the worst in the range of 2-5 sec, similar to the baseline scenario in Section IV-B. This also explains the overall result in Figure 4(a).

Figure 5(d) plots the CDF of the number of broken routes for the range of 2-5 sec. The number of broken routes for SAMER and CP-HT is very high compared to Figures 3(d), 4(d) – the median numbers are 162 and 165, respectively. Moreover, SAMER selects longer paths than CP-HT (the median path length for the two protocols is 8.86 and 7.86 hops, respectively). The combined effect of many route breaks and large path lengths makes SAMER perform poorly under high PU activity. This also explains SAMER's poor performance under high PU activity in the baseline scenario.

E. Impact of Channel Loss Ratio

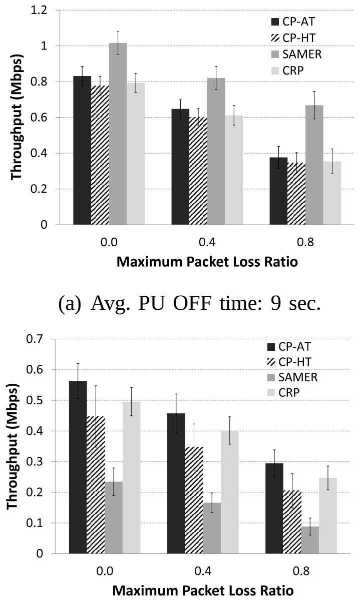
To study the impact of loss ratio on protocol performance, we consider again the baseline scenario and repeat the simulations for maximum packet loss ratio of 0.0, 0.4, and 0.8. The results are plotted in Figures 6(a), 6(b), for average PU OFF time equal to 9 sec and 3 sec, respectively.

In Figure 6(a), we observe that under low PU activity, SAMER achieves the highest throughput among the four protocols. Furthermore, SAMER is the most robust protocol to packet loss. When the maximum packet loss ratio increases from 0.0 to 0.8, SAMER's throughput drops by only 34%, while the throughput of the remaining three protocols drops by about 55%. When the intensity of PU activities is not high, incorporating loss ratio in the link metric of a CRN protocol improves the protocol's performance, similar to the case of routing metrics for traditional WMNs.

On the other hand, Figure 6(b) shows that SAMER performs the worst among the four protocols under high PU activity. In that case, incorporating loss ratio in the routing metric does not help and protocols which ignore loss ratio and choose routes using PU activity as the only criterion achieve better performance.

F. Multiple Data Flows

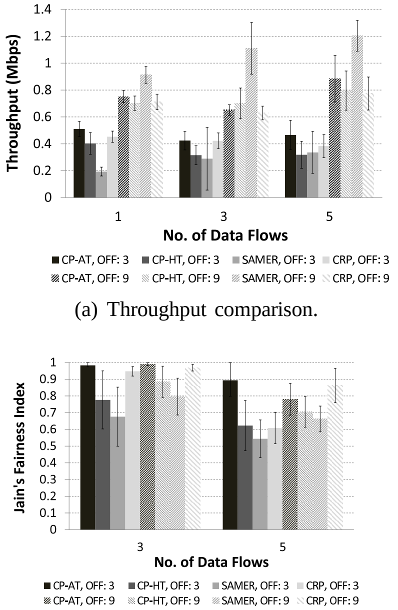
To study the performance of each protocol with multiple data flows, we conducted simulations with 3 and 5 flows



(a) Avg. PU OFF time: 9 sec.

(b) Avg. PU OFF time: 3 sec.

Figure 6. Average throughput as a function of the loss ratio.



(a) Throughput comparison.

(b) Fairness comparison.

Figure 7. Multiple data flows.

using the baseline scenario. In the topology shown in Figure 1, we selected the source-destination pairs SU0-SU9, SU1-SU8, and SU2-SU7, for the simulations with 3 data flows. For 5 data flows, we added 2 more source-destination pairs - SU3-SU6 and SU4-SU5. We repeated the simulations with the same 20 seeds used in Section IV-B for average PU OFF time equal to 3 and 9 sec. Figure 7(a) plots the total throughput, and Figure 7(b) plots Jain's Fairness Index, with 1, 3, and 5 flows.

In Figure 7(a), we observe that SAMER achieves the highest throughput, regardless of the number of flows when the intensity of PU activities is not high (OFF time 9 sec). Moreover, when the number of flows increases from 1 to 3, the total throughput increases with SAMER but drops with the other three protocols. This is because protocols which only consider PU activities in estimating spectrum availabilities are more likely to share a large part of the routing path for all three flows (note that all 3 source/destination nodes are impacted by PUs in the same cell), resulting in high contention among SUs. In contrast, SAMER considers interference from neighboring SUs in estimating spectrum availability and selects more disjoint paths. In Figure 7(b), we observe that the increase in SAMER's total throughput under low PU activity comes at the cost of reduced fairness compared to the other three protocols; SAMER penalizes some flows by routing them over longer paths in attempt to reduce the amount of SU interference. In contrast, CP-AT and CRP have the highest fairness, at the cost of reduced throughput.

On the other hand, when the intensity of PU activities is high (PU OFF time 3 sec), SAMER achieves, in general, the lowest performance in terms of both throughput and fairness, similar to our observations in IV-B and IV-D. In this case, CP-AT achieves the highest throughput followed closely by CRP. However, CP-AT achieves much better fairness than CRP, especially as the number of flows increases. CP-HT outperforms SAMER in terms of throughput in the presence of a single flow, but the gap diminishes with multiple flows. Its fairness index though remains higher than SAMER's, regardless of the number of flows.

V. Testbed Prototyping and Evaluation

In this section, we present a generic software architecture for the experimental evaluation of CRN routing protocols on a testbed based on the USRP2 platform [26]. The architecture provides a complete design of PHY, MAC, and network layers. Basic CRN functions such as PU activities, SU periodic sensing, and channel switching capabilities are also supported. Based on this architecture, we prototype and compare CP-AT, CP-HT, and SAMER on a testbed consisting of 6 USRP2 nodes. We describe the software architecture in Section V-A and the hardware configuration and experimental setup of our testbed in Section V-B. Finally, we discuss the results in Section V-C.

A. Architecture

As shown in Figure 8, the software architecture consists of a *Data/Decision plane*, a *Routing plane*, and a *Communication plane*.

The *Data/Decision plane* is responsible for channel switching and scheduling data transmissions among nodes, using either CSMA with the help of RTS/CTS control

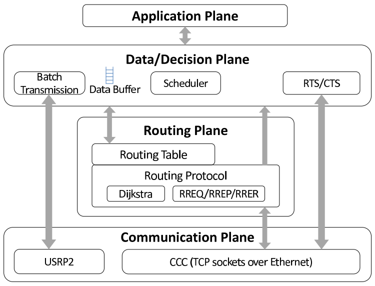


Figure 8. Testbed software architecture

packets or TDMA. Since the nodes in our testbed are half-duplex, they cannot transmit and receive simultaneously (unlike in the simulations). In the case of CSMA, RTS is used by a node A to reserve a channel with its downstream node B. When node B is free to receive data packets, it switches frequency and replies with a CTS to indicate its availability to node A. In the case of TDMA, each node maintains a schedule with the Tx/Rx slots and the corresponding channels. Due to the high overhead of per packet channel reservation and switching, a node transmits a batch of packets in each slot or between two RTS/CTS exchanges. A data buffer is implemented on each node to store packets for future transmissions.

The Data/Decision plane is also responsible for maintaining the sensing-transmission cycle. Each node follows a schedule according to which it checks PU activity during the sensing period and sends/receives data during the data transmission period. Due to the limited number of USRPs, we do not use USRPs as PUs, but instead we emulate PU activity by providing each node with an input file describing PU activity in its neighborhood over time (ON/OFF intervals). SUs "sense" PU activities by remaining idle during each sensing period and looking up PU activity in their input file.

The *Routing plane* manages the Routing Table and implements the route discovery and route maintenance mechanisms which we described in Section III. The Data/Decision Plane refers to the Routing Table before forwarding data to another node and then uses the underlying Communication plane to transmit data or control packets. When it receives a RERR packet from the Routing Plane, which indicates the routing path is broken because of PU activities, the Data/Decision Plane cleans the data buffer.

The *Communication plane* is responsible for data/control packet exchange among neighboring nodes. While data packets are sent over a wireless channel using USRP2, control packets (RTS, CTS, RREQ, RREP, RERR, etc.) are sent via TCP sockets over a Gigabit Ethernet interface, which emulates an out-of-band common control channel. Furthermore, similar to [34] and [35], communication on the control channel and the data channel is handled by two different threads on the host. In emulating the common control channel, we only establish TCP connections between nodes which are neighbors in a given topology, so that broadcast packets are received by nodes reachable according to that topology. An all-wireless common control channel is left as a future extension.

B. Hardware and Experimental Setup

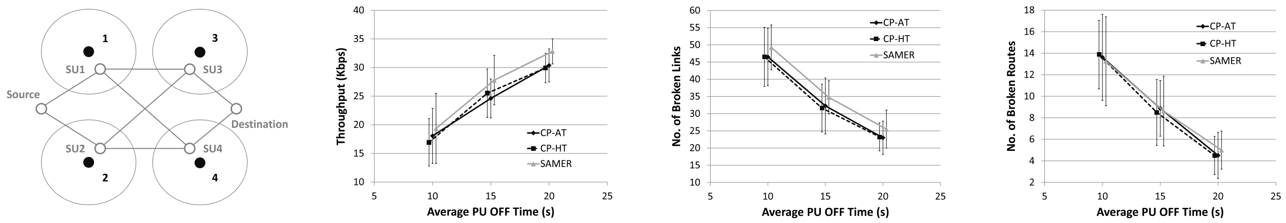
We implemented CP-AT, CP-HT, and SAMER on a testbed consisting of six nodes. Each testbed node consists of a PC running Ubuntu 12.04 and a USRP2. Each PC has two Gigabit Ethernet interface cards. One of them is used to connect to USRP2 and the other one is used to enable the common control channel over a Gigabit Ethernet backbone. Each USRP2 is equipped with a half-duplex daughterboard (XCVR2450). We use a TDMA MAC protocol in our experiments and allow non-interfering links to transmit in the same time slot.

In our experiments, each node can use 5 channels for data transmissions. The center frequencies are 2.512 GHz, 2.513 GHz, 2.514 GHz, 2.515 GHz, and 2.516 GHz. Each channel has a bandwidth of 0.2MHz. On each node, we used a batch size of 100 packets, a packet size of 500 bytes, OFDM with BPSK, and the default transmit power of USRP2. The sensing-transmission cycle consists of a 1 sec sensing period and a 3 sec transmission period. Each experiment runs for 1200 seconds, with the first 1000 seconds used for observing PU activities and the remaining 200 seconds for data transfer. In our experiments, all PUs have an average ON time of 15 sec and an average PU OFF time of 10, 15, and 20 sec.

C. Testbed Results

Figure 9(a) shows the topology we used for our experiments, in which 4 SUs used as relays are impacted by PUs in 4 different locations. Similar to our simulations, we assume there is a PU on each data channel in each location. For each of the three average PU OFF times, we use 8 seeds to generate PU activities. Due to the temporal variability of the wireless environment in a real testbed, we repeat the experiment 5 times for each seed. Figures 9(b), 9(c), and 9(d) plot the average throughput, number of broken links, and number of broken routes, respectively, for the three protocols, as a function of the average PU OFF time. The error bars show the standard deviations.

We observe that SAMER provides the highest throughput and the performance gap is larger under low PU activity (average PU OFF time 20 sec). This result is consistent to our simulation results. Contrary to our simulation results, SAMER's performance is not severely impacted by high intensity of PU activities; SAMER still outperforms CP-AT and CP-HT when the average PU OFF time is 10 sec. This is because, in this topology, SAMER has the same path length as the other two protocols; the length of all possible paths is 3 hops. Moreover, all routing paths cross locations with similar intensity of PU activities; as a result,



(a) Topology. (b) Throughput comparison. (c) Broken links. (d) Broken routes.

Figure 9. Testbed evaluation. For better clarity, in Figures 9(b), 9(c), 9(d), the data points for SAMER and CP-HT are shifted horizontally.

all three protocols have similar numbers of broken routes, as shown in Figure 9(d). Therefore, SAMER achieves higher throughput, by choosing the path with the lowest amount of SU interference (highest channel diversity) and/or lowest loss ratio, among the available paths of equal length and similar PU activity. Although the number of link breaks is higher for SAMER (Figure 9(c)), similar to our simulations results, many of these breaks do not result in route breaks, as we explained in Section IV-C.

We also observe that CP-AT and CP-HT perform similarly in this topology, as they both choose one of the available 3-hop paths of similar PU activity, without taking into account SU activity or link quality. This is also consistent to our simulation results, when CP-AT and CP-HT both choose the shortest path (Figures 3(a), 3(b)).

VI. Related Work

In recent years, numerous routing protocols for CRNs have been proposed with different design goals, e.g., maximizing throughput [14], [15], [10], [16], [17], [18], minimizing delay [11], [19], [6], [13], maximizing route stability [20], [12], minimizing route recovery/maintenance cost [21], [22], etc. Many of those protocols [14], [15], [20], [11], [16], [19], [17] assume static channel availability and do not include PU dynamics in their routing metrics. Consequently, such protocols are similar to those proposed for multi-channel WMNs and cannot deal with temporal variations of spectrum availability in CRNs. Among works which take PU dynamics into account, some focus on analytical studies, e.g., [22] and [18], and some others propose protocols relying on transmission power adaptation, e.g., [18] and [6]. A performance comparison of these types of protocols is left as future work.

The majority of routing protocols for CRNs are only compared against protocols which do not take PU dynamics into account, e.g., [10], [21], [12], [13]. One exception is [6], in which the authors compare the proposed protocol against SAMER. In their evaluation, the authors do not implement a sensing functionality on the SUs to learn PU activities online, in a distributed way, but instead they assume each SU has complete a priori knowledge of the model and parameters of PU activities. In [30], the authors conduct a comparison study of single-path and multi-path AODV with three routing metrics – ETX, ETT, and hop count – in a multihop CRN. However, neither AODV nor any of the three routing metrics are designed for CRNs. To our best knowledge, our work is the first extensive empirical performance comparison study of routing protocols in CRNs.

Most of the existing routing protocols for CRNs have only been evaluated in simulators, primarily due to the difficulty to build a CRN testbed. Although a number of cognitive radio platforms has become available in the past few years (see [36] for a survey), the majority of CRN protocols that have been evaluated on testbeds are MAC/PHY layer protocols (e.g., [7], [34]).

A notable exception is Coolest Path [12] which was evaluated in a 6-node USRP-based testbed. The authors in [12] only compare Coolest Path against random routing and they do not provide details about the testbed architecture, e.g., about the MAC/Network layer or the implementation of the common control channel. Furthermore, in their testbed evaluation, the authors use the route switch ratio (which is proportional to the number of route breaks) as the performance metric. In our study, we found that a lower switch ratio (number of route breaks) does not always result in higher throughput.

The authors in [34] and [35] build a small testbed of 3 USRPs for the evaluation of their proposed routing protocols. The testbed has some similarities to ours, e.g., the common control channel is implemented as an Ethernet interface. However, their evaluation included only single-hop experiments. In contrast, in our testbed, we evaluate the performance of two routing protocols in multihop topologies with more USRPs.

VII. Conclusion

In this paper, we conducted the first detailed empirical performance study of routing protocols for CRNs using both the ns-2 simulator and a testbed based on the USRP2 platform. Our main findings are: i) Taking link quality and interference among SUs into account can greatly improve throughput and end-to-end delay under low PU activity; in contrast, path stability and path length become the dominant factors that affect performance under high PU activity. ii) Considering interference among SUs in the case of multiple flows can result in more disjoint paths and increase total throughput at the cost of reduced fairness. iii) Link and path stability are not always good performance indicators.

iv) For link routing metrics that ignore link quality, limiting the path length through the use of an additive instead of a bottleneck path metric typically improves performance. This conclusion does not always hold true for link quality-based routing metrics. v) Estimating spectrum availability based only on local observations cannot guarantee path stability.

Overall, we found that the performance of routing protocols in CRNs is affected by a number of factors, in addition to PU activity, and different protocols perform well under different scenarios. Our study motivates the design of self-adaptive protocols that choose different link/path routing metrics in different scenarios, in an online manner. We plan to investigate this direction as part of our future work.

REFERENCES

[1] M. Cesana, F. Cuomob, and E. Ekicic, "Routing in cognitive radio networks: Challenges and solutions", *Elsevier Ad Hoc Networks*, vol. 9, no. 3, pp. 228–248, May 2011.

[2] W.-Y. Lee and I. F. Akyildiz, "Optimal spectrum sensing framework for cognitive radio networks", *IEEE Transactions on Wireless Communications*, vol. 7, no. 10, pp. 3845–3857, 2008.

[3] P. Setoodeh and S. S. Haykin, "Robust transmit power control for cognitive radio", *Proceedings of the IEEE*, vol. 97, no. 5, pp. 915–939, 2009.

[4] C. Cordeiro and K. S. Challapali, "C-MAC: A cognitive MAC protocol for multi-channel wireless networks", in *Proc. of IEEE DySpan*, 2007.

[5] J. Jia, Q. Zhang, and X. Shen, "HC-MAC: A hardware-constrained cognitive MAC for efficient spectrum management", *IEEE Journal on Selected Areas in Communication*, vol. 26, no. 1, January 2008.

[6] K. C. How, M. Ma, and Y. Qin, "Routing and QoS provisioning in cognitive radio networks", *Elsevier Computer Networks*, vol. 55, no. 1, p. 330342, January 2011.

[7] P. Bahl, R. Chandra, T. Moscibroda, R. Murty, and M. Welsh, "White space networking with Wi-Fi like connectivity", in *Proc. of ACM SIGCOMM 2009*, 2009.

[8] H. Khalife, N. M. Malouch, and S. Fdida, "Multihop cognitive radio networks: to route or not to route", *IEEE Network Magazine*, July-August 2009.

[9] M. Youssef, M. Ibrahim, M. Abdelatif, L. Chen, and A. V. Vasilakos, "Routing metrics of cognitive radio networks: a survey", *IEEE Communications Surveys and Tutorials*, 2013.

[10] I. Pefkianakis, S. H. Wong, and S. Lu, "SAMER: Spectrum aware mesh routing in cognitive radio networks", in *Proc. of IEEE DySpan*, 2008.

[11] 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 (Kluwer)*, vol. 13, no. 1-2, pp. 67–81, April 2008.

[12] X. Huang, D. Lu, P. Li, and Y. Fang, "Coolest Path: Spectrum mobility aware routing metrics in cognitive ad hoc networks", in *Proc. of IEEE ICDCS*, 2011.

[13] K. R. Chowdhury and I. F. Akyildiz, "CRP: A routing protocol for cognitive radio ad hoc networks", *IEEE Journal on Selected Areas in Communication*, vol. 29, no. 4, April 2011.

[14] C. Xin, B. Xie, and C.-C. Shen, "A novel layered graph model for topology formation and routing in dynamic spectrum access networks", in *Proc. of IEEE DySPAN*, 2005.

[15] Q. Wang and H. Zheng, "Route and spectrum selection in dynamic spectrum networks", in *Proc. of IEEE CCNC*, 2006.

[16] A. Sampath, L. Yang, L. Cao, H. Zheng, and B. Y. Zhao, "High throughput spectrum-aware routing for cognitive radio networks", in *Proc. of IEEE Crowncom*, 2008.

[17] A. S. Cacciapuoti, M. Caleffi, and L. Paura, "Reactive routing for mobile cognitive radio ad hoc networks", *Elsevier Ad Hoc Networks*, vol. 10, no. 5, pp. 803–815, July 2012.

[18] 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 *Proc. of ACM MSWiM*, 2009.

[19] K. Chowdhury and M. Felice, "Search: A routing protocol for mobile cognitive radio ad-hoc networks", *Elsevier Computer Communications*, vol. 32, no. 18, pp. 1983–1997, December 2009.

[20] S. Deng, J. Chen, H. He, and W. Tang, "Collaborative strategy for route and spectrum selection in cognitive radio networks", in *Proc. of IEEE FGCN*, 2007.

[21] G.-M. Zhu, I. F. Akyildiz, and G.-S. Kuo, "STOD-RP: A spectrum-tree based on-demand routing protocol for multi-hop cognitive radio networks", in *Proc. of Globecom*, 2008.

[22] I. Filippini, E. Ekici, and M. Cesana, "Minimum maintenance cost routing in cognitive radio networks", in *Proc. of IEEE MASS*, 2009.

[23] J. Broch, D. A. Maltz, D. B. Johnson, Y.-C. Hu, and J. Jetcheva, "A Performance Comparison of Multi-Hop Wireless Ad Hoc Network Routing Protocols", in *Proc. of ACM MobiCom*, October 1998.

[24] S. R. Das, C. E. Perkins, and E. M. Royer, "Performance comparison of two on-demand routing protocols for ad hoc networks", in *Proc. of IEEE INFOCOM*, March 2000.

[25] R. Draves, J. Padhye, and B. Zill, "Comparison of routing metrics for static multi-hop wireless networks", in *Proc. of ACM SIGCOMM*, 2004.

[26] E. Inc., "Universal Software Radio Peripheral", http://ettus.com.

[27] D. S. J. D. Couto, D. Aguayo, J. C. Bicket, and R. Morris, "A high-throughput path metric for multi-hop wireless routing", in *Proc. of ACM MobiCom*, 2003.

[28] R. Draves, J. Padhye, and B. Zill, "Routing in multi-radio, multi-hop wireless mesh networks", in *Proc. of ACM MobiCom*, 2004.

[29] C. E. Perkins and E. M. Royer, "Ad hoc on-demand distance vector routing", in *Proc. of IEEE WMCSA*, February 1999.

[30] M. D. Felice, K. R. Chowdhury, W. Kim, A. Kassler, and L. Bononi, "End-to-end protocols for cognitive radio ad hoc networks: An evaluation study", *Elsevier Performance Evaluation*, vol. 68, no. 9, pp. 859–875, September 2011.

[31] I. F. Akyildiz, W.-Y. Lee, and K. R. Chowdhury, "CRAHNs: Cognitive radio ad hoc networks", *Elsevier Ad Hoc Networks*, vol. 7, no. 5, pp. 810–836, July 2009.

[32] "IEEE 802.22 working group on wireless regional area networks", http://www.ieee802.org/22/.

[33] R. K. Jain, D.-M. W. Chiu, and W. R. Hawe, "A quantitative measure of fairness and discrimination for resource allocation in shared computer systems", DEC-TR-301, Digital Equipment Corporation, Tech. Rep., September 1984.

[34] L. Ding, P. B. Nagaraju, T. Melodia, S. N. Batalama, D. A. Pados, and J. D. Matyjas, "Software-defined joint routing and waveform selection for cognitive ad hoc networks", in *Proc. of IEEE MILCOM*, 2010.

[35] P. Nagaraju, L. Ding, T. Melodia, S. Batalama, D. Pados, and J. Matyjas, "Implementation of a distributed joint routing and dynamic spectrum allocation algorithm on USRP2 radios", in *Proc. of IEEE SECON*, 2010.

[36] K. R. Chowdhury and T. Melodia, "Platforms and testbeds for experimental evaluation of cognitive ad hoc networks", *IEEE Communications Magazine*, vol. 48, no. 9, September 2010.