Primary User Aware k-hop Routing for Cognitive Radio Networks

Arsany Guirguis
Comp. and Sys. Eng. Dept.
Alexandria University, Egypt
arsany.guirguis@alexu.edu.eg

Mohamed Ibrahim
Dept. of Computer Science
Rutgers University, USA
mibrahim.ahmed@rutgers.edu

Karim Seddik
American University in Cairo
Egypt
kseddik@aucegypt.edu

Khaled Harras
Computer Science Dept.
Carnegie Mellon University
kharras@cs.cmu.edu

Fadel Digham
National Telec. Reg. Authority
Egypt
fdigham@tra.gov.eg

Moustafa Youssef
Wireless Research Center
Egypt-Japan Uni. of Sci. and Tech.
moustafa.youssef@ejust.edu.eg

***Abstract*—We propose a primary user-aware k-hop routing scheme that can be plugged into any cognitive radio network routing protocol to adapt, in real time, to the environmental changes. The main use of this scheme is to make the compromise required between the route overhead and its optimality based on a user-defined utility function. We analytically derive the optimal discovery radius (k) that achieves this target. Evaluations on NS2 show that our scheme can enhance the current routing protocols in terms of throughput with minimal overhead.**

I. Introduction

Cognitive Radio Networks (CRNs) present a promising solution for spectrum scarcity in wireless networks to cope with the ever-increasing demand for mobile communications. In CRNs, unlicensed secondary users (SUs) opportunistically utilize vacant portions of the spectrum without interfering with licensed primary users (PUs). This promises a large set of potential applications, given the scarcity of the unlicensed wireless spectrum, including high-demand and highly crowded distributed mobile applications such as the Internet of things, high-quality mobile video, and disaster or emergency response settings. Despite this promise, one of the main problems that impacts the performance of multi-hop CRNs is routing. Compared to traditional ad hoc networks, routing in CRNs has to deal with unique challenges such as dynamic spectrum availability due to the stochastic behavior of primary and secondary users, resource heterogeneity resulting from the availability of different channels and radios on the same node, and synchronization between nodes on different channels.

To tackle these routing challenges in CRNs, routing protocols for CRNs have attracted attention from a large number of researchers [1]. These protocols can be categorized into two main classes: global and local routing protocols. Topology-based (global) routing protocols, e.g. [2]–[4], discover all possible routes to the destination by flooding the network with control packets and selecting the optimal route based on a defined routing metric. Despite this optimality, these protocols do not scale or adapt to support topological changes as a result of high mobility or variations in network size. On the other hand, geographic (local) routing protocols, e.g. [5]–[8], make localized greedy decisions at each hop by selecting the

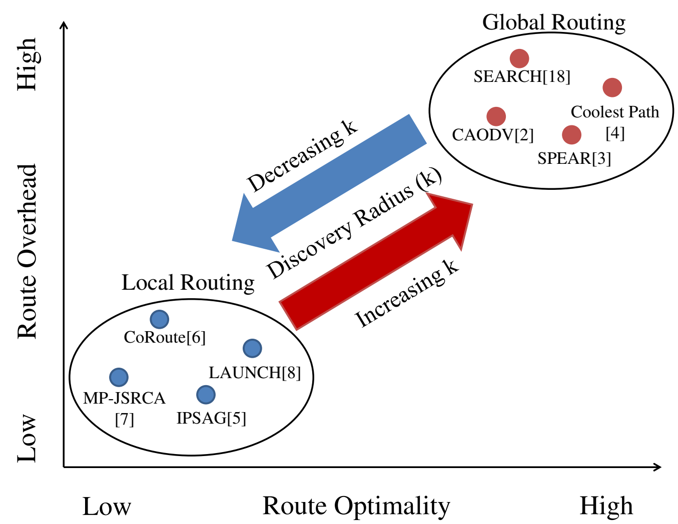


Fig. 1: *The tradeoff between local and global routing*

best one-hop neighbor from those geographically closer to the destination. Such greedy approaches take local optimal decisions to rapidly adapt to network dynamics without flooding the network with control packets. However, they suffer from the overhead of dealing with local voids. Consequently, an inevitable tradeoff exists between the optimality of a chosen route and the routing overhead as shown in Figure 1 which shows the spectrum of different routing protocols in CRNs.

We therefore propose *PAK*: a **P**rimary User **A**ware **k**-hop route discovery scheme that can explore this spectrum of protocols between geographic and topology-based routing. *PAK* is not designed to act as yet another routing protocol. Instead, it can be plugged into any routing protocol with minimal changes. Based on a user-defined utility function that balances overhead and route optimality, it can dynamically find the best discovery radius, k, in real time for each node in the network.

We derive, mathematically, a utility function that quantifies the tradeoff discussed. We then evaluate *PAK* by integrating it with a sample routing protocol for CRNs [2] (with minor modification that will be discussed later) using NS2 [9]. Our results demonstrate that in typical CRN operation scenarios, performance is significantly enhanced, in terms of throughput with minimal incurred overhead, when gradually increasing the discovery radius.

Overall, our contributions in this paper are threefold:

1) We propose a Primary User Aware k-hop route discovery scheme to be plugged to any routing protocol in cognitive radio networks.
2) We present a mathematical analysis of the optimal discovery radius, in terms of the most optimal route with least overhead, based on a user-defined utility function.
3) We integrate *PAK* with a traditional routing protocol for CRNs and evaluate its performance.

The remainder of this paper is organized as follows. Section II discusses the related work. Section III presents our system model, assumptions and the discovery scheme used by *PAK*. We then provide a mathematical analysis of the optimum k in Section IV. We evaluate the proposed system in Section V. Finally, Section VI concludes the paper and provides directions for future work.

II. Related Work

In traditional ***wired*** networks, the tradeoff problem between global and local routing is discussed extensively. For example, in the scope of the Internet, the concept of Autonomous Systems (Intra and Inter AS routing) is used to achieve this tradeoff. Along the same line, hybrid routing techniques have been investigated before for large scale networks in the context of mobile ***ad hoc networks*** [10], whose challenges are different from CRNs. For example, Terminodes [11] proposes two modes of operation where a greedy geographic approach is used for long distances and switches to a topological global mode when it approaches the destination.

Current routing protocols in CRNs, however, are either static geographic (local) approaches (based on 1 hop information [5], [7], [8] or 2 hop information [6]) or static topological (global) approaches [3], [4], [12]–[19]. Local geographic approaches discover one- or two-hop neighbors and pick the best route according to the used routing metric. These metrics include location-aided metrics [5], where the nearest neighbor to the destination will be used as a next hop. Moreover, another class of local approaches uses PU aware routing metrics where offline statistics are leveraged to estimate PU behavior [6], [8]. Local approaches suffer from picking local optimal routes and may be trapped in local voids which can be resolved using perimeter routing [20]. Global routing approaches, on the other hand, flood the network with control packets, and hence, pick the optimal route. However, these routing approaches do not scale to support large, highly-loaded, and dynamic networks in which their performance significantly degrades.

Despite this extensive routing research in CRNs, studying the impact of varying the number of discovered hops (k), as shown in Figure 1, on routing optimality and its overhead has not been investigated before in the context of CRNs. Moreover, current routing protocols in CRNs are static protocols that cannot adapt with the environment changes, e.g., dynamic changes in the number of SUs and number of active connections. All of these are the subject of the proposed scheme.

III. System Model

In this section, we present our system assumptions and then provide a brief overview on the proposed discovery scheme.

A. System Assumptions

We consider an ad hoc cognitive radio network. PUs are located uniformly in the deployment area. PUs' activities are modeled as an ON-OFF birth-death process, where the periods of the ON and OFF periods follow two independent exponential distributions with birth parameter λ and death parameter β depending on the traffic of the PUs [21]. All PUs are homogeneous in terms of their parameters and transmission ranges. We assume that PUs are stationary. This is common in many CRN scenarios such as TV white space-based CRNs. We also do not make any specific assumption on the MAC or higher layer protocols for the PUs' system.

We further assume that SUs are located uniformly in the two dimensional Cartesian space and each SU knows its own location and the location of its direct neighbors. Assuming knowing only the location of the destination is a typical and valid assumption in many applications including military and sensor networks where reporting nodes know the locations of the sink nodes. Without loss of generality, exchanging routing control packets can be done through a Common Control Channel (CCC). Finally, we assume that our scheme will be plugged on routing protocol that are already running in the background [22], [23].

B. Discovery Scheme Overview

Figure 2 shows how *PAK* operates in a 2-hop neighborhood discovery scenario, where Node E (Src) tries to reach Node N (Dst). Node E starts the discovery process by broadcasting a route request (RREQ) packet for Node N on the CCC and waits for the replies. As shown, some nodes will not reply based on their location relative to the source and the destination. When the source node receives replies, it chooses one of the k-hop neighbor, i.e. nodes at the k-hop perimeter (**we call it a mega-hop neighbor**) according to the returned routing metric which describes the PUs activity along the route.

IV. Analytical Model

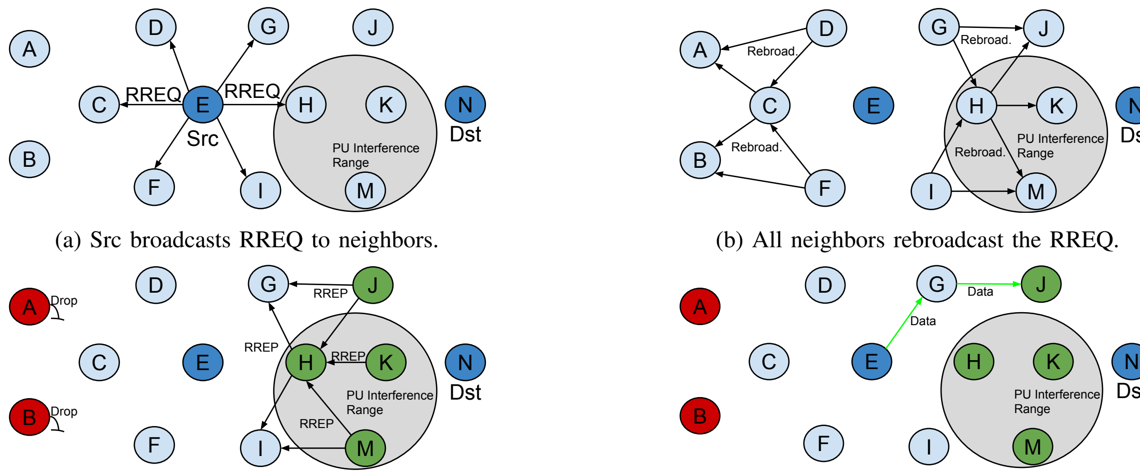
A. Notations

Table I summarizes the notations used in the paper. We model a CRN as a graph $G = (V_{SUs}; V_{PUs}; E)$, where V_{SUs} represents the set of SU nodes, V_{PUs} is the set of PU nodes and E is the set of graph edges.

B. Problem Formulation

We propose a utility function U that describes the relation between two weighted metrics:

1) **Optimality Metric ($Opt(k)$)**: This metric quantifies the advantage of discovering larger k as more information about PUs will be gained, leading to better routes.
2) **Overhead Metric ($Over(k)$)**: This metric quantifies the overhead of discovering larger k in terms of



(a) Src broadcasts RREQ to neighbors.

(b) All neighbors rebroadcast the RREQ.

(c) Only the 2-hop neighbors, that are nearer to Dst than Src, reply.

(d) Src picks the optimal route and forwards the packet to the best 2-hop neighbor.

Fig. 2: *A 2-hop neighborhood scenario.* Note that all communications occur on the CCC, which is independent of the PU activity. All nodes inform the source about the PU activity near them. The source node selects the optimal route based on a user-defined utility function.

TABLE I: Mathematical Notations

Symbol	Description
k	Discovery Radius
T_r	SUs' transmission range
$T_{r_{pu}}$	PUs' transmission range
l	The side length of the square deployment area
n	The number of SUs
n_{pu}	The number of PUs
n_{pu_k}	The number of PUs within k-hops
d	The average node degree
μ	SUs' density
α	The weight of optimality metric in the utility function. This is user-defined
τ	The period within which PU activity is observed
λ	Activity rate of each PU
p_{pu}	The probability to get affected by any given PU
p_{not}	The probability of not being affected by any PU

control packets that flood the network to discover the route to the destination.

Different representations can be used for both metrics. For example, optimality metric may capture loss ratio, end-to-end delay, among others. In *PAK*, the optimality metric is designed to describe the PUs behavior so that the nature of CRNs is captured[1]. Based on this, we note that both the optimality and overhead metrics increase with k. Therefore, we use the following utility function to combine them:

$$U(k) = \alpha Opt(k) - (1-\alpha) Over(k) \tag{1}$$

where α is a parameter that determines the user preference.

Let Src be a source node establishing a connection with Node Dst. The Src node will discover only k^* hops, where k^* is the optimal hops in terms of the routing utility function:

$$k^* = \arg\max_k U(k). \tag{2}$$

Therefore, our goal now is to develop a mathematical formula for the two functions: $Over(k)$ and $Opt(k)$, that can be used to find the optimal k.

C. Control Overhead Analysis

1) Average Number of Neighbors within k-hops: In this section, we study the relation between k and the routing overhead, i.e., the total number of control packets to discover k hops. In order to find this relation, we first need to derive a formula for the average number of nodes within k hops from the sender.

Let n_k be a discrete random variable representing the number of nodes within k hops. Then, the average number of nodes that are within k hops from the sender:

$$E(n_k) = np_k,$$

where p_k is the probability for a node to be within k hops from the sender. Given the assumption of uniform distribution of SU nodes in the deployment area, p_k is given by:

$$p_k = \frac{\pi r^2}{l^2} = \frac{\pi k^2 T_r^2}{l^2};$$

where $r = kT_r$ is the radius of the area of the k-hop neighborhood and l^2 is the total deployment area. This probability can further be simplified using the average node degree $d = \frac{n\pi T_r^2}{l^2}$ [24]) as

$$p_k = \frac{dk^2}{n}.$$

Then, we have

$$E(n_k) = dk^2. \tag{3}$$

2) Overall Control Overhead: The routing overhead includes the number of times that the route request (RREQ) packet is rebroadcast and number of route replies (RREP)

[1]We define the optimality metric mathematically later in this section.

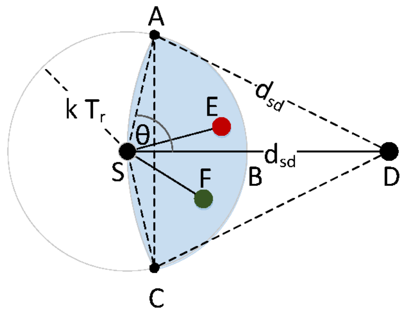


Fig. 3: *The forwarding area in a k-hop Geographic Forwarding.*

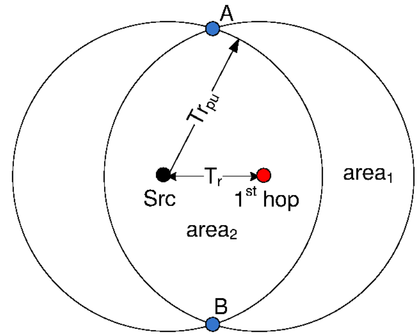


Fig. 4: *Area of intersection between circles formed by nodes of the chosen route. The radius of the circles is $T_{r_{pu}}$ since any PU within this range will affect the nodes.*

unicasted. RREQ messages are forwarded through the edges of a spanning tree, rooted at the sender of the RREQ as shown in [24]. Let M_{RREQ} be the average number of RREQ messages broadcast within k hops from the sender of the RREQ. Each node will rebroadcast the RREQ message just once for a certain source-destination pair during a specified time. Then, M_{RREQ} is the average number of non-leaf nodes in a breadth-first tree of the graph rooted at the route requester node.

$$M_{RREQ} = 1 + \sum_{i=1}^{k-1} a_i, \tag{4}$$

where a_i is the average number of nodes that are exactly i hops from the requester node which, from Equation (3), is given by:

$$a_i = E(n_i) - E(n_{i-1}) = di^2 - d(i-1)^2 = d(2i-1).$$

Then, Equation (4) can be simplified into the following expression:

$$M_{RREQ} = 1 + \frac{2d(k-1)^2}{2} = 1 + d(k-1)^2.$$

On the other hand, an RREP message will be unicasted k times (from k-hop neighbors) until it reaches the route requester. Only the mega-hop that has smaller distance to the destination (D) than the distance between the source (S) and the destination will reply to the route request. Given that the number of k-hop neighbors is $d(2k-1)$, the average number of RREP messages, M_{RREP} is given by the following expression:

$$M_{RREP} = \frac{area_{ABCS}}{\pi k^2 T_r^2} dk(2k-1),$$

where $area_{ABCS}$ (refer to Figure 3) is the area that contains the nodes which can respond to the source node. It can be shown that $area_{ABSC}$ is given by:

$$area_{ABCS} = T_r^2 k^2 \left(\theta - \frac{\sin 2\theta}{2} + \frac{\sec^2\theta}{4}\left(\pi - 2\theta - \frac{\sin 4\theta}{2}\right)\right),$$

Then, the average of the total number of control packets sent to discover k-hop neighbors is given by:

$$\begin{aligned} &Over(k,\theta) \\ &= M_{RREQ} + M_{RREP} \\ &= 1 + d(k-1)^2 \\ &+ \frac{1}{\pi}\left(\theta - \frac{\sin 2\theta}{2} + \frac{\sec^2\theta}{4}\left(\pi - 2\theta - \frac{\sin 4\theta}{2}\right)\right) dk(2k-1). \end{aligned}$$

D. Optimality Metric Analysis

Since *PAK* is designed to work in CRNs, we propose an optimality metric that quantifies the robustness of the chosen route in terms of the effect of PUs on this route. So, the best (optimal) route is the one that has the least PUs effect. According to our chosen PUs model, the probability of PU activity during a given period τ is given by:

$$p_{pu} = 1 - e^{-\tau\lambda}.$$

Then the probability of not being affected by any PU (from those that are existing within the discovery range from the source to the destination) is given by:

$$p_{not} = (1 - p_{pu})^{n_{pu_k}},$$

where n_{pu_k} is the number of PUs within the discovery area. This quantity is given by:

$$n_{pu_k} = \frac{n_{pu} * \text{disc. area from source to megahop destination}}{l^2}.$$

To get the discovered area from source to megahop destination, we assume that $T_r \le T_{r_{pu}}$ since SUs should send using less power than that of PUs. According to Figure 4 the total discovered area can be given by:

$$\begin{aligned} \text{Discovered Area} &= \text{Area of circle} + k * area_1 \\ &= \pi T_{r_{pu}}^2 + k * (\pi T_{r_{pu}}^2 - area_2). \end{aligned}$$

where area of the intersection between the two circles ($area_2$) is given by:

$$area_2 = 2\left(T_{r_{pu}}^2 \cos^{-1}\left(\frac{T_r}{2T_{r_{pu}}}\right) - \left(\frac{T_r}{2}\right)\sqrt{T_{r_{pu}}^2 - \left(\frac{T_r}{2}\right)^2}\right).$$

Finally, we take the optimality metric as the information gained by knowing the probability of the PU affecting a certain route given as:

$$Opt(k) = -\log_2 p_{not} = -n_{pu_k}\log_2(1 - p_{pu}).$$

E. Optimal Discovery Radius

In this subsection, we solve the utility expression to find the optimal k. Noting that the overhead metric depends on both k and θ (Figure 3), we first average over the distribution of θ to factor it out from the equation and let the utility function depend only on k.

For simplicity, we assume that θ follows a **uniform distribution** and its values range from 0 to $\frac{\pi}{2}$ Therefore, we can simplify $U(k)$ as:

$$U(k) = \frac{1}{\frac{\pi}{2}} \int_0^{\frac{\pi}{2}} U(k,\theta) d\theta$$

$$= -\alpha n_{pu_k} \log_2(1-p_{pu}) - (1-\alpha)(1+d(k-1)^2+11.29dk(k-1)). \quad (5)$$

Therefore, the optimal value of k is:

$$k^* = \frac{13.29d(1-\alpha) - \alpha \frac{n_{pu}}{l^2}(\pi T_{r_{pu}}^2 - area_2)\log_2(1-p_{pu})}{47.16d(1-\alpha)}. \quad (6)$$

V. Performance Evaluation

In this section, we evaluate *PAK* via NS2 simulations. We first describe our simulation setup, parameters and metrics used. Then we present and discuss the simulation results.

A. Simulation Setup

We used a multi-channel version of NS2 [25]. *PAK* can be plugged on any routing protocol. However, in our simulations, we implemented and used a modified version of CAODV [2] as the underlying routing protocol on which we plug *PAK*. Our modified version works similar to the default CAODV but uses the optimality metric defined in the previous section as its route selection metric. We use the IEEE 802.11 as the MAC protocol and a CBR traffic model for the generated packets from the SUs.

TABLE II: Experiments parameters.

Parameter	Value range	Nominal Value
Number of SUs (n)	100	100
Number of PUs (n_{pu})	2-10	2
Number of active connections	10	10
SU transmission range T_r (m)	125	125
PU transmission range $T_{r_{pu}}$ (m)	140	140
Number of channels	2	2
Packet size (Byte)	512	512
Data Rate per source (Kpbs)	16	16
Network Capacity (Mbps)	1.5	1.5
Square Deployment area side length (m)	1000	1000
User Utility Parameter (α)	0.1-0.9	0.5
Activity period τ (sec.)	1	1
PUs Activity Parameter (λ)	0.5	0.5

B. Experimental Parameters

Table II summarizes the experimental parameters. PUs are uniformly located over the available channels in the area of interest.

C. Metrics

We evaluate *PAK* using these metrics:

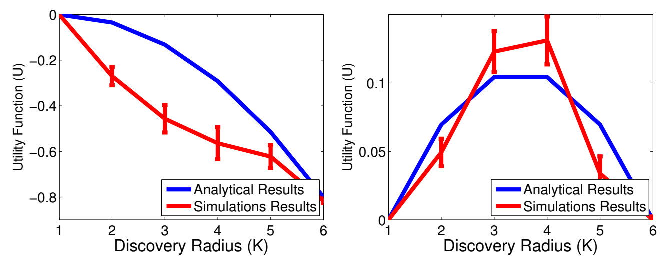
1) Throughput: number of bits transmitted correctly from source to destination per second.
2) Routing overhead ratio: ratio of number of transmitted control packets to total number of transmitted data packets.

We limit our results to these two metrics only due to size limits. We choose the achieved throughput to quantify the optimality of the route and the overhead ratio to trace its overhead.

D. Experimental Results

We first validate our analytical results and study the effect of the user utility parameter (α) on performance. Finally, we show the effect of mismatching k on performance. However, some experimental results are omitted due to size constraints.

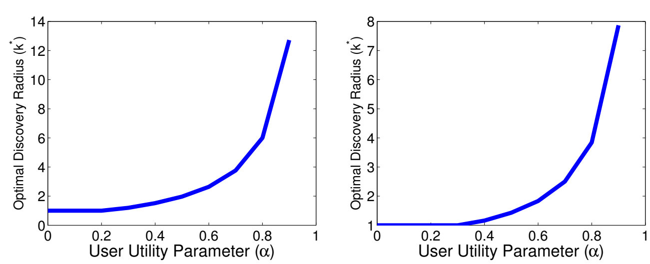
1) **Validating Analytical Expressions**: Figure 5 studies the gap between local and global routing based on the user defined utility weight (α). The figure shows that, for different user goals in terms of routing optimality and overhead, the proposed scheme enables the required compromise. We can also see that the simulation results match the analytical results in the general behaviour. The difference between simulation and analytical results could be due to the uniform angle distribution assumption.



(a) α = 0.1 (favors low overhead) - Optimal $k = 1$

(b) α = 0.5. (equal weight to both overhead and optimality) - Optimal $k = 4$

Fig. 5: Effect of changing k on the utility function at different values of α obtained through both analysis and simulations. For both subfigures, the goal is to obtain the best utility value. This is different based on k. Traditional algorithms that have a fixed k cannot adapt to dynamic network conditions or user desires. As shown, optimal value of k depends on the user-defined weight α



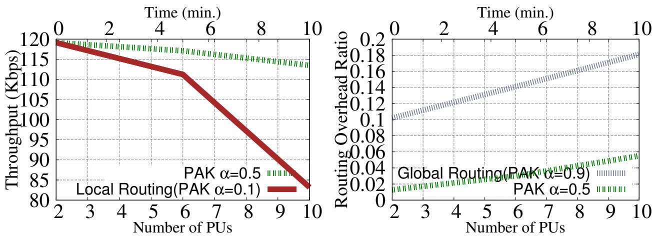
(a) n:n_{pu} = 25:3

(b) n:n_{pu} = 100:3

Fig. 6: Effect of changing α on k in different cases of network topology (different ratios of n to n_{pu}). Increasing α leads to getting larger discovery radius (k) returning more optimal route with higher overhead.

2) **The Effect of User Utility Parameter** α: Figure 6 shows the effect of α on k. Setting α to low value favors the traditional local routing, and hence optimal value for k will be smaller. However, one can choose

high value for α (and hence high value for k) in case of preferring the global routing approach. So, user can achieve the desired behavior by setting α to an appropriate value.



(a) Throughput (b) Routing Overhead Ratio

Fig. 7: Effect of changing Number of PUs with time on throughput and routing overhead ratio. We can see that *PAK* adapts well with the topological change to give better performance than the routing protocol that uses fixed value for k.

3) **Effect of Changes in Network Dynamics**: In Figure 7 we see how *PAK* adapts to the change of the number of PUs along the time. It is clear that *PAK* sets an appropriate value for k as network topology changes whereas fixing k to a certain value (as in global or local approach) gives bad performance. Figure 7a shows that local approach performance degrades significantly while increasing number of PUs. However, setting α to 0.5 allows *PAK* to choose an appropriate value for k to keep the good performance. The same applies for Figure 7b where overhead increases a lot when number of PUs increases with time in the case of global routing (fixing k to ∞). But, *PAK* performance remains acceptable with this change.

VI. Conclusion and Future Work

We proposed a new scheme for adaptive routing discovery in CRNs where the number of hops to be discovered can be set to adapt with the network topology. We studied the tradeoff between the routing optimality and overhead as a function of the number of discovered hops both analytically and through simulations. The results showed the advantages of the proposed scheme and how it can be applied on top of any routing protocol. We are currently extending our work by applying our discovery scheme over different classes of routing protocols in CRNs as well as experimenting with other distributions for network nodes. Furthermore, we are implementing our scheme over Cognitive routing protocols on some of the emerging testbeds like CogFrame [26] and CRESCENT [27] to see the performance on real scenarios.

Acknowledgement

This work is supported in part by a grant from the Egyptian National Telecommunication Regulatory Authority (NTRA).

References

[1] Youssef et al. Routing metrics of cognitive radio networks: A survey. *Communications Surveys & Tutorials, IEEE*, 16(1):92–109, 2014.

[2] Cacciapuoti et al. CAODV: Routing in mobile ad-hoc cognitive radio networks. *Wireless Days (WD)*, 2010.

[3] Sampath et al. High throughput spectrum-aware routing for cognitive radio networks. *Proc. of IEEE Crowncom*, 2008.

[4] Huang et al. Coolest Path: Spectrum Mobility Aware Routing Metrics in Cognitive Ad Hoc Networks. In *Distributed Computing Systems (ICDCS), 2011 31st International Conference on*, pages 182 –191, 2011.

[5] Badoi et al. IPSAG: An IP Spectrum Aware Geographic Routing Algorithm Proposal for Multi-hop Cognitive Radio Networks. In *Communications (COMM), 2010 8th International Conference on*, pages 491–496. IEEE, 2010.

[6] Kim et al. CoRoute: A New Cognitive Anypath Vehicular Routing Protocol. *Wireless Communications and Mobile Computing*, 11(12):1588–1602, 2011.

[7] Tang et al. A Joint Design for Distributed Stable Routing and Channel Assignment Over Multi-Hop and Multi-Flow Mobile Ad Hoc Cognitive Networks. *Industrial Informatics, IEEE Transactions on*, PP(99):1–1, 2012.

[8] Habak et al. A location-aided routing protocol for cognitive radio networks. In *IEEE ICNC*, 2013.

[9] S. Floyd S. McCanne. The Network Simulator NS-2. [Online] http://www.isi.edu/nsnam/ns/.

[10] Mauve et al. A survey on position-based routing in mobile ad hoc networks. *Network, IEEE*, 15(6):30–39, 2001.

[11] Blazevic et al. Self organized terminode routing. *Cluster Computing*, 5(2):205–218, 2002.

[12] Cheng et al. Joint On-Demand Routing and Spectrum Assignment in Cognitive Radio Networks. *Proceedings of IEEE ICC 2007*, June 2007.

[13] H. Ma, L. Zheng, X. Ma, and Y. luo. Spectrum Aware Routing for Multi-Hop Cognitive Radio Networks with a Single Transceiver. *Proceedings of IEEE CrownCom*, 2008.

[14] G. Cheng, W. Liu, Y. Li, and W. Cheng. Spectrum Aware On-Demand Routing in Cognitive Radio Networks. *Proceedings of IEEE DySPAN*, 2007.

[15] H. Khalife, S. Ahuja, N. Malouch, and M. Krunz. Probabilistic Path Selection in Opportunistic Cognitive Radio Networks. *Proceedings of IEEE GLOBECOM*, 2008.

[16] M. Pan, R. Huang, and Y. Fang. Cost design for opportunistic multi-hop routing in cognitive radio networks. *Proceedings of IEEE MILCOM*, 2008.

[17] Xin et al. A novel layered graph model for topology formation and routing in dynamic spectrum access networks. *Proceedings of IEEE DySPAN*, 2005.

[18] Chowdhury et al. Search: A routing protocol for mobile cognitive radio ad-hoc networks. *Computer Communications*, 32(18):1983–1997, 2009.

[19] Beltagy et al. A new routing metric and protocol for multipath routing in cognitive networks. In *Wireless Communications and Networking Conference (WCNC), 2011 IEEE*, pages 974–979. IEEE, 2011.

[20] Karp et al. GPSR: greedy perimeter stateless routing for wireless networks. In *Proceedings of the 6th annual international conference on Mobile computing and networking*, pages 243–254. ACM, 2000.

[21] Lee et al. Optimal spectrum sensing framework for cognitive radio networks. *Wireless Communications, IEEE Transactions on*, 7(10):3845–3857, 2008.

[22] Karmoose et al. Dead zone penetration protocol for cognitive radio networks. In *Wireless and Mobile Computing, Networking and Communications (WiMob), 2013 IEEE 9th International Conference on*, pages 529–536. IEEE, 2013.

[23] Guirguis et al. Primary user-aware network coding for multi-hop cognitive radio networks. *Proceedings of IEEE GLOBECOM*, 2014.

[24] Youssef et al. Overlapping multihop clustering for wireless sensor networks. *Parallel and Distributed Systems, IEEE Transactions on*, 20(12):1844–1856, 2009.

[25] Cognitive radio extension. *[Online]. Available: http://stuweb.ee.mtu.edu/ljialian/.*

[26] Saeed et al. A low-cost large-scale framework for cognitive radio routing protocols testing. In *Communications (ICC), 2013 IEEE International Conference on*, pages 2900–2904. IEEE, 2013.

[27] Asal et al. Crescent: a modular cost-efficient open-access testbed for cognitive radio networks routing protocols. In *Proceedings of the 19th annual international conference on Mobile computing & networking*, pages 179–182. ACM, 2013.