This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMC.2018.2828104, IEEE Transactions on Mobile Computing

Primary User-aware Optimal Discovery Routing for Cognitive Radio Networks

Arsany Guirguis[‡], *Student Member, IEEE*, Fadel Digham[¶], *Member, IEEE*, Karim Seddik[⋆], *Senior Member, IEEE*, Mohamed Ibrahim[†], *Member, IEEE*, Khaled A. Harras[§], *Senior Member, IEEE* and Moustafa Youssef[⋆⋆], *Senior Member, IEEE*

[‡] Department of Computer and Systems Engineering, Alexandria University, Egypt.
[¶] National Telecommunication Regulatory Authority, Egypt.
[⋆] Electronics and Communications Engineering Department, American University in Cairo, Egypt.
[†] Department of Computer Science, Rutgers University, USA.
[§] Computer Science Department, Carnegie Mellon University, USA.
[⋆⋆] Egypt-Japan University of Science and Technology (E-JUST), Egypt.
Email: arsany.guirguis@alexu.edu.eg, fdigham@tra.gov.eg, kseddik@aucegypt.edu,
mibrahim.ahmed@rutgers.edu, kharras@cs.cmu.edu, moustafa.youssef@ejust.edu.eg.

Abstract*—Routing protocols in multi-hop cognitive radio networks (CRNs) can be classified into two main categories: local and global routing. Local routing protocols aim at decreasing the overhead of the routing process while exploring the route by choosing, in a greedy manner, one of the direct neighbors. On the contrary, global routing protocols choose the optimal route by exploring the whole network to the destination paying the flooding overhead cost. In this paper, we propose a primary user-aware k-hop routing scheme where k is the discovery radius. This scheme can be plugged into any CRN routing protocol to adapt, in real time, to network dynamics like the number and activity of primary users. The aim of this scheme is to cover the gap between local and global routing protocols for CRNs. It is based on balancing the routing overhead and the route optimality, in terms of primary users avoidance, according to a user-defined utility function. We analytically derive the optimal discovery radius (k) that achieves this target. Evaluations on NS2 with a side-by-side comparison with traditional CRNs protocols show that our scheme can achieve the user-defined balance between the route optimality, which in turn reflected on throughput and packet delivery ratio, and the routing overhead in real time.

Keywords*—*Cognitive radio networks, Routing optimality-scalability tradeoff, Routing protocols

I. Introduction

[1]Cognitive Radio Networks (CRNs) present a promising solution for spectrum scarcity in wireless networks to cope with the ever-increasing demand for higher bandwidth in mobile communications [1], [2]. 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 distributed mobile applications for high-demand and highly-crowded scenarios such as the Internet of Things, high-quality mobile video, and disaster or emergency response settings. One example of these applications is the recent Spectrum Collaboration Challenge (SC2) proposed by DARPA in 2016 [3]. In this challenge, "*competitors will reimagine a new, more efficient wireless paradigm in which radio networks autonomously collaborate to dynamically determine how the spectrum should be used moment to moment*". Despite this promise, one of the main problems that impact the performance of multi-hop CRNs is routing. Compared to traditional ad hoc networks, routing in CRNs has to deal with the unique challenges of dynamic spectrum availability (due to the stochastic behavior of primary and secondary users) [4], resource heterogeneity (resulting from the availability of different channels and radios on the same node), and synchronization between nodes on different channels, among others.

To tackle these routing challenges in CRNs, routing protocols have attracted the attention of a large number of researchers [5], [6]. These protocols can be categorized into two main classes: global and local routing protocols. Topology-based (global) routing protocols, e.g. [7]–[11], discover all possible routes to the destination by flooding the network with control packets and select the optimal route based on a defined routing metric. In general, there is a large number of ways for defining optimality. One approach is to design a multi-objective optimization function; this would be cumbersome and could rather focus on one metric to address its impact from different perspectives. Despite this optimality, these protocols do not scale and are not able to quickly 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. [12]–[18], make localized greedy decisions at each hop by selecting the 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 falling into local optima.

[1] An earlier version of this paper appeared in the proceedings of IEEE Global Communications Conference (GLOBECOM) 2015.

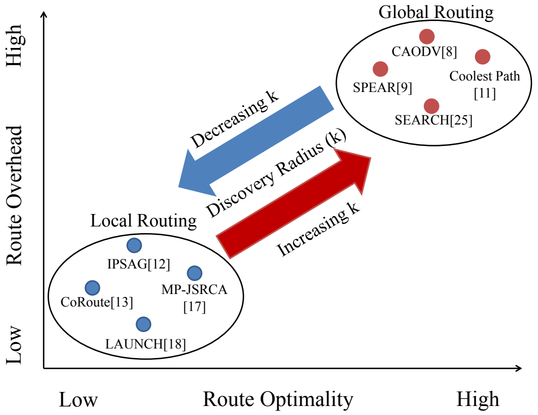
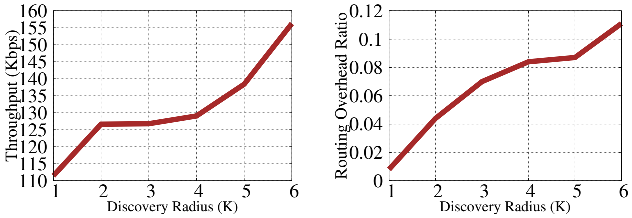


Fig. 1: The tradeoff between local and global routing.

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. The ultimate factor distinguishing such solutions is the discovery radius k. We define the discovery radius k as the number of hops to which the route requests are broadcast before sending a route reply. Lower values of k, usually one, result in low overhead non-optimal solutions, which is the typical case in geographic-based routing protocols. At the other end of the spectrum, increasing k widens the route discovery space, ultimately leading to optimal solutions (for $k = \infty$) at the expense of high overhead. Figure 2 quantifies the effect of changing k on throughput and routing overhead. Figure 2a shows the change of throughput with the increase of k. Throughput is defined by the number of bits per second communicated successfully from the source to the destination. This quantity increases with increasing k, as more information about the network can be known to be taken into consideration while choosing the best route. On the other hand, Figure 2b draws the change of the routing overhead with increasing k. The routing overhead ratio is defined by the ratio of bits communicated as control packets to those communicated as data packets. Generally, at low values of k, the routing overhead is minimized but the discovered route is not optimal, leading to a low throughput. On the other hand, getting an optimal route (and hence maximizing the throughput) ends up with being penalized by a higher overhead. Essentially, in highly dynamic large-scale networks, none of these protocols at both ends of the spectrum that use a fixed discovery radius can gracefully adapt to changing network conditions.

We therefore propose *PAK*: a **P**rimary User **A**ware **k**-hop route discovery scheme that can explore the undiscovered 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. In particular, based on a user-defined utility function that balances overhead and route optimality, *PAK* can dynamically find the best discovery radius, k, in real time for each node in the network.

For that, we specifically study analytically the effect of varying the discovery radius, k, on the optimality and overhead metrics. This allows us to use k as an explicit controllable parameter rather than a constant value as in traditional routing protocols. We then evaluate *PAK* by integrating it with a sample routing protocol for CRNs [8] using NS2 simulations [19]. Our results demonstrate that, in typical CRN operation scenarios, *PAK* adapts in real time to the changing network dynamics and always achieves the user preferred balance between route optimality and overhead regardless of the network topology.



(a) Throughput

(b) Routing Overhead Ratio

Fig. 2: General effect of changing k on the network metrics using NS2 simulations. The used topology consists of 100 SUs, 2 PUs and 10 active connections with a rate of 16 Kbps for each one. Other parameters are listed in Table II.

Overall, our contributions in this paper are threefold:

1) We propose *PAK*: a Primary user-Aware k-hop route discovery scheme that can be plugged on top of any routing protocol for cognitive radio networks. The proposed scheme covers the design space in CRN routing protocols between local and global routing protocols.
2) We present a mathematical analysis for the best discovery radius 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.

II. Related Work

In traditional ***wired*** networks, the tradeoff problem between global and local routing is studied 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. Similarly, the tradeoff between route optimality and scalability has been also studied in the context of ***wireless*** networks. For example, [20] balances this tradeoff for opportunistic routing protocols by presenting a graph partitioning method to decompose a large-scale wireless network into small autonomous sub-topologies using local information, where each sub-topology could realize local optimal opportunistic routing by itself. Likewise, in the context of mobile ***ad hoc networks***, different classes of routing protocols have been introduced [21], [22] including greedy geographical, direction-restricted flooding and hybrid approaches. For example, Terminodes [23] proposes two modes of operation in which a greedy geographic approach is used for long distances and switches to a topological global mode when it approaches the destination. [24] uses virtual backbone routing to achieve the scalability preserving the route optimality. It combines local proactive and global reactive routing components over a variable-sized zone hierarchy. However, it uses a fixed zone size for a particular environment. Therefore, it cannot adapt to the network dynamic changes.

Current routing protocols in CRNs, however, are either static geographic (local) approaches [12]–[18] or static topological (global) approaches [7]–[11], [25], [26]. Local geographic approaches discover one- or two-hop neighbors and pick the best route according to the used routing metric. These metrics include pure location-aided metrics (k= 1 or 2) [12], [15], [16], where the nearest neighbor to the destination is used as a next hop regardless of any other network metric. Another class of local approaches uses PU-aware routing metrics in addition to the location information where offline statistics are leveraged to estimate the PU behavior [13], [14], [18]. Thus, in this class of protocols, greedy decisions are taken but PUs activities are also taken into consideration when calculating the routing metric. In summary, local approaches suffer from picking local optimal routes and may be trapped in local voids, which can be resolved using perimeter routing [27]. Global routing approaches, on the other hand, flood the network with control packets (k= ∞), 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 to the network changes, e.g., dynamic changes in the number of SUs and number of active connections. All of these common issues are the subject of the proposed scheme.

III. System Model

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

A. System Assumptions

We consider an ad hoc cognitive radio network where PUs hold the right to use the licensed spectrum while SUs can access only the unoccupied portions of the spectrum, i.e. SUs must evade the channels when PUs become active. PUs are uniformly located 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 [28]. These parameters can be estimated using offline statistics, local sensing information, or through implicit feedback in full-duplex communications. All PUs are homogeneous in terms of their parameters and transmission ranges. We assume that SUs and PUs channels follow the unit disc model [29]. Also, 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[2] in the two-dimensional Cartesian space and each SU knows its own location and the location of its direct neighbors. A node can estimate its location using any of the current localization systems, including GPS [30] or mobile-based systems [31]. Moreover, a sender can obtain the location information of the ultimate destination via out of band services that map node addresses to locations, or have it disseminated through the network. 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. Although having the locations of all nodes of the network may lead to better and more robust routes, we do not assume having nodes' locations except for the sink and the neighboring nodes; we believe that disseminating the nodes' locations information to the whole network and keeping it up-to-date requires a huge overhead, which degrades our system performance. Without loss of generality, exchanging routing control packets can be done through a Common Control Channel (CCC). Moreover, our model does not assume any specific frequency band.

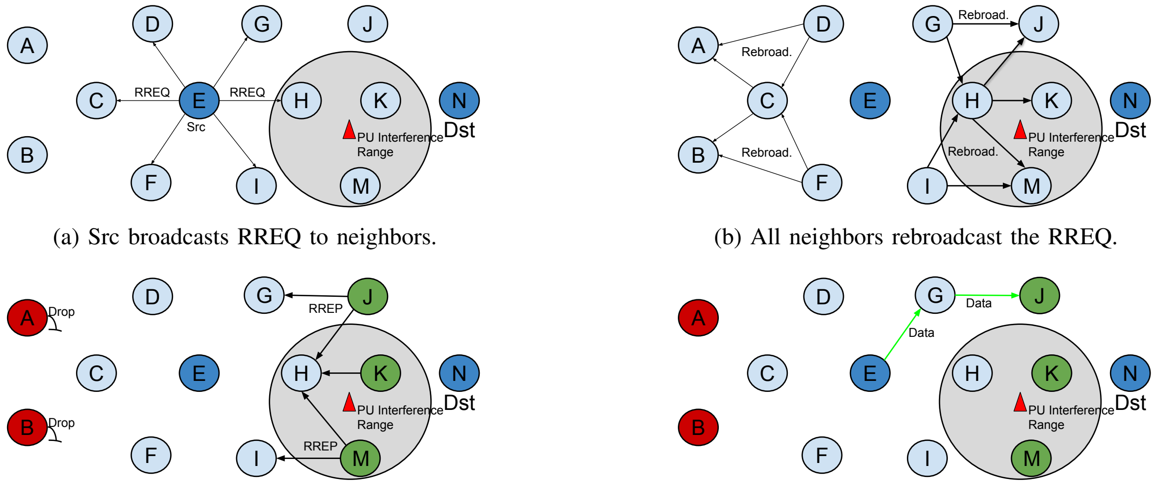
B. Discovery Scheme Overview

Figure 3 shows how *PAK* operates in a 2-hop neighborhood discovery scenario (i.e. $k = 2$), 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 (Figure 3a).

The RREQ packet will be rebroadcast by receiving nodes until it reaches the destination or a k-hop neighbor (2-hop neighbors in this case) from Node E as shown in Figure 3b. The destination location information is used to take the decision of whether to send a RREP packet or not (Figure 3c). Specifically, RREQ packets that reach k-hop neighbors that are further away from the destination than the source are discarded and no RREP is sent.

If the RREQ packet reaches a k-hop neighbor (**we call it a mega-hop neighbor**) or the destination, then this neighbor replies with the routing metric based on the used routing protocol in a RREP packet. Each node in the reverse route will combine its local computed metric with the accumulated route metric in the RREP packet based on the routing protocol metric. Finally, the source will choose the route to the selected mega-hop neighbor corresponding to the best metric based on the replies that reach the source within a pre-specified period. The packet is then forwarded to this neighbor, which starts the same k-hop discovery process until the destination is reached. It is quite important to note that if $k \neq \infty$, the routing protocol acts similar to any location-based routing which can fall in local optimal. This problem is addressed through routing around the local minimum area, as in [27].

[2] We do some experiments with SUs locations that follow Gaussian distribution in Section V.



(a) Src broadcasts RREQ to neighbors.

(b) All neighbors rebroadcast the RREQ.

(c) Only the 2-hop neighbors that are closer to Dst than Src reply. Red nodes drop the RREQ packet where green nodes respond with a RREP packet.

(d) Src picks the optimal route based on the replies it receives and forwards the packet to the best 2-hop neighbor.

Fig. 3: *A 2-hop neighborhood scenario.* Note that all communications occur on the CCC, which is independent of the PUs activity. All nodes inform the source about the PUs activity near them. The source node selects the optimal route based on a user-defined utility function. Circles represent SUs and the traingle represents the PU.

IV. Analytical Model

In this section, we define a utility function that reflects the user preference by balancing the route optimality and the routing overhead as a function of k. We show analytically how network metrics change depending on the value k and find the optimal value of k that achieves the user-defined target.

A. Notations

Table I summarizes the notations used in the paper. We model a CRN as a graph $G = (V_{\text{SUs}}; V_{\text{PUs}}; E)$, where V_{SUs} represents the set of SU nodes and two vertices are connected by an edge $e \in E$ if they are within the transmission range, T_r, of each other. V_{PUs} is the set of PU nodes and E is the set of edges in the SU network (i.e., connections among SUs). The network size, $n_{\text{su}} = |V_{\text{SUs}}|$, represents the number of SUs in the network and $n_{\text{pu}} = |V_{\text{PUs}}|$ describes the number of PUs in the network. Therefore, the node density, $\mu = \frac{n_{\text{su}}}{l^2}$, assuming a square deployment area with a side length l. Let k be the discovery radius. Therefore, the area of k-hop neighborhood can be approximated as a circle with a radius $r = kT_r$.

TABLE I: Mathematical Notations

Symbol	Description
k	Discovery radius
T_r	SUs' transmission range
$T_{r_{\text{pu}}}$	PUs' transmission range
l	The side length of the square deployment area
n_{su}	The total number of SUs
n_{pu}	The total number of PUs
n_{pu_k}	The number of PUs within k-hops
d	The average SU degree
d_{sd}	The distance between the source and the destination
μ	SUs' density
α	The user-defined weight of optimality metric in the utility function.
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

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 using higher k for discovery, resulting in gaining more information about PUs and leading to better routes. In this paper, we consider the optimal route as the one which has the least interference on the surrounding PUs. Formally, least interference means the least probability of PUs activity in the region of transmission. Although primary receivers (not the transmitters) are those which need to be protected from SUs interference, in our model, we consider protecting primary users in general, as a PU can work as a transmitter or a receiver at any time. The interference of the PUs on the SUs is also implicitly captured in our model. If an SU exists in a PU transmission range, the former's capacity decreases. Thus, improving this metric decreases the probability of affecting the PUs

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMC.2018.2828104, IEEE Transactions on Mobile Computing

and increasing the SUs capacity as well.

2) **Overhead Metric (Over(k))**: This metric quantifies the overhead of discovering larger k in terms of control packets that flood the network to discover the route to the destination. Moreover, this metric implicitly captures the energy consumed in discovering the route as the consumed energy increases with increasing the SUs' transmission. Thus, getting the number of transmissions done by SUs to discover the route can quantify the energy consumed by them for route discovery.

Our model evaluates the interference as follows: In general, the level of the interference at a PU depends on the received power from the SUs. There is no interference at the PU if the received SU signal power is lower than the noise threshold. This happens when the PU is outside of the SU transmission range. If the PU is inside the SU's transmission range, the former may suffer from interference from the SU's signal. In this case, the SINR is lower (than the first case) and the rate, that the SU can achieve, is lower too. If the SU interference power becomes more significant, the two signals (PU signal and SU interfering signal) are said to have collided and the node's radio drops both of them.

In this model, the optimality metric is related to the transmitted data on the data channel, while the overhead metric is related to the transmitted control packets on the CCC. The latter is typically a scarce shared resource with a low data rate compared to the data transfer channel. Thus, we target minimizing the communicated data on it according to the overhead metric. Yet, the data channel is less scarce so that more data can be communicated through it. However, it suffers from the presence of PUs (that should be protected). Therefore, we use the optimality metric to capture this requirement (protecting the PUs from SUs' interference). We do not guarantee the full protection of PUs but we try to maximize the probability of protecting them. Finally, it is important to note that if the model is changed to consider only one channel to communicate both control and data messages (in-band CCC), the PUs' activities should be taken into consideration while calculating the overhead. We believe that our model can accommodate this with minimal changes.

We note that both the optimality and overhead metrics increase with k. Therefore, we use the following utility function to combine them:

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

where α is a weighting parameter that determines the user preference and N is a normalization constant accounting for the unit differences between overhead and optimality metrics which we discuss in Section IV-G. Higher values of α, e.g. $\alpha = 1$, favor the optimality metric (leading to higher values for k), while lower values for α favor reducing the routing overhead, leading to lower values for k. Although both metrics in the equation could have different ranges, α and N capture this difference.

Let Src be some node that has data to be forwarded towards some destination Dst. The Src node will discover only k^* hops, where k^* is the optimal number of hops in terms of the routing utility function:

$$k^* = \underset{k}{\operatorname{argmax}}\, U(k) \tag{2}$$

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

C. Control Overhead Analysis

In this section, we study the relation between k and the routing overhead, i.e., the total number of transmitted control packets to discover k hops. In order to find this relation, we first derive the average number of nodes within k hops from the sender. Then, we quantify the control overhead in terms of the transmitted RREQ and RREP packets.

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

$$E(n_k) = n_{\text{su}} p_k$$

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 [32]:

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

where $r = kT_r$ is the radius of the area of the k-hop neighborhood and l^2 is the deployment area. This probability can further be simplified using the average node degree $d \triangleq \frac{n_{\text{su}} \pi T_r^2}{l^2}$ [33] as

$$p_k = \frac{dk^2}{n_{\text{su}}}$$

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 the number of route replies (RREP) unicast.

Let M_{RREQ} be the average number of the RREQ messages broadcast within k hops from the sender of the RREQ. Each node (within k-hop) will rebroadcast the RREQ message just once for a certain source-destination pair during a specified time. Then, M_{RREQ} is the number of non-leaf nodes in a breadth-first tree of the graph rooted at the route requester node.

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

where a_i is the number of nodes that are, on average, 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:

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

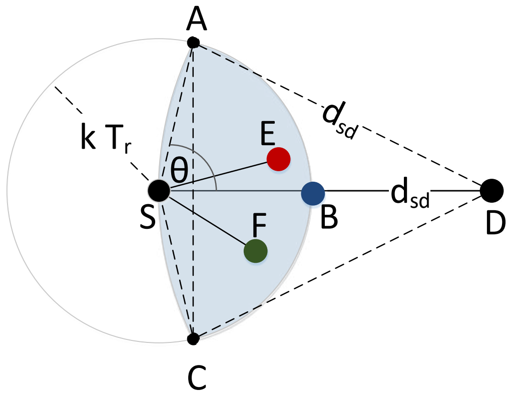


Fig. 4: The allowed forwarding area for the source node S where the destination node is D. Points A and C are on the boundaries of this area since they are kT_r away from S and d_{sd} away from D. Nodes E and F are the potential next mega-hops since they are in the allowed forwarding area of S.

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

$$M_{\text{RREP}} = \frac{\text{area}_{\text{ABCS}}}{\pi k^2 T_r^2} dk(2k-1) \tag{5}$$

where area$_{\text{ABCS}}$ (Figure 4) is the area that contains the nodes whose distances to destination are shorter than that from source to destination, i.e., the responding nodes:

$$\text{area}_{\text{ABCS}} = T_r^2 k^2 \left(\theta - \frac{\sin 2\theta}{2}\right) + d_{sd}^2 \left(\pi - 2\theta - \frac{\sin 4\theta}{2}\right) \tag{6}$$

What follows is to calculate the value of θ. Given that $SD = AD = d_{sd}$, then $\angle SAD = \theta$, leading to:

$$\cos\theta = \frac{kT_r}{2d_{sd}} \tag{7}$$

Substituting in Equation (6), area$_{\text{ABCS}}$ can be written as:

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

Therefore, from Equation (5), M_{RREP} is given by:

$$\begin{aligned} M_{\text{RREP}} = \frac{dk(2k-1)}{\pi} \Bigg(& \theta - \frac{\sin 2\theta}{2} \\ & + \frac{\sec^2\theta}{4}\left(\pi - 2\theta - \frac{\sin 4\theta}{2}\right)\Bigg) \end{aligned} \tag{8}$$

and the total number of control packets sent to discover k-hop neighbors is:

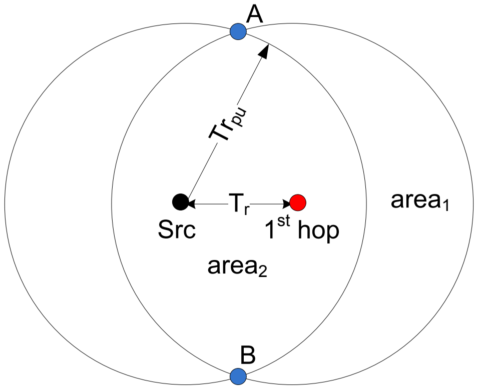


Fig. 5: Area of intersection between circles formed by nodes of the chosen route. The radius of the circles is $T_{r_{\text{pu}}}$ since any PU within this range will affect the nodes.

$$\begin{aligned} & Over(k,\theta) \\ & \quad = M_{\text{RREQ}} + M_{\text{RREP}} \\ & \quad = 1 + d(k-1)^2 \\ & \quad + \frac{1}{\pi}\left(\theta - \frac{\sin 2\theta}{2} + \frac{\sec^2\theta}{4}\left(\pi - 2\theta - \frac{\sin 4\theta}{2}\right)\right) dk(2k-1) \end{aligned} \tag{9}$$

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. Specifically, the best (optimal) route is the one that experiences the least PUs effect/activity. The probability of a PU activity during a given period τ is given by:

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

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

$$p_{\text{not}} = (1 - p_{\text{pu}})^{n_{\text{pu}_k}}$$

where n_{pu_k} is the number of PUs within the discovery area:

$$n_{\text{pu}_k} = \frac{n_{\text{pu}} \times \text{discovered area from source to a mega-hop dest.}}{l^2}$$

Without loss of generality, let the SU transmit range (T_r) be $\leq$ the PUs transmit range ($T_{r_{\text{pu}}}$), then from Figure 5, the total discovered area within which the whole link is protected from the PUs effect is given by:

$$\begin{aligned} \text{Discovered Area} &= \text{Area of circle} + k \times \text{area}_1 \\ &= \pi T_{r_{\text{pu}}}^2 + k \times (\pi T_{r_{\text{pu}}}^2 - \text{area}_2). \end{aligned}$$

where the area of the intersection between the two circles (area$_2$) is given by:

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

We finally set the optimality metric as the information gained by knowing the activity probability of the PUs affecting a certain route as:

$$\text{Opt(k)} = -\log_2 p_{\text{not}} = -n_{\text{pu}_k} \log_2(1 - p_{\text{pu}}) \qquad (10)$$

E. Optimal Discovery Radius

In this section, we solve the utility expression to find the optimal k. Noting that, from Equation (2), the overhead metric depends on both k and θ, we start by deriving the distribution of θ. Then, we get an expression for the average utility function. Finally, we find the expression for the optimal k^*.

1) Finding θ distribution: From Equation (7), we can note that θ has one-to-one mapping with d_{sd} given that k and T_r will be constant for a given network deployment. So, by knowing the distribution of d_{sd}, we can get θ distribution directly. However, d_{sd} is the distance between the source and the destination which are located uniformly in the deployment area. So, d_{sd} can be considered as the distance between two points that are randomly located in a square of side length l. According to [34], the probability density function of d_{sd} is given by:

$$f(d_{sd}) = \frac{4d_{sd}}{l^4}\phi(d_{sd}) \qquad (11)$$

where $\phi(d_{sd})$ is given by:

$$\phi(d_{sd}) = \begin{cases} \frac{\pi}{2}l^2 - 2ld_{sd} + \frac{d_{sd}^2}{2} & 0 \le d_{sd} < l \\ \\ l^2\left[\sin^{-1}(\frac{l}{d_{sd}}) - \cos^{-1}(\frac{l}{d_{sd}})\right] - l^2 - \frac{d_{sd}^2}{2} & \\ +2l\sqrt{d_{sd}^2 - l^2} & l \le d_{sd} \le \sqrt{2}l. \end{cases}$$

Then, we can write the distribution of θ as follows:

$$g(\theta) = \frac{f(d_{sd})}{\mid \frac{\partial\theta}{\partial d_{sd}} \mid} \qquad (12)$$

which can be reduced to (Appendix A):

$$g(\theta) = \frac{k^2T_r^2}{l^4(1-T)}\sec^2(\theta)\tan(\theta)\psi(\theta) \qquad (13)$$

where

$$\psi(\theta) = \begin{cases} \frac{\pi}{2}l^2 - lkT_r\sec(\theta) + \frac{1}{8}k^2T_r^2\sec^2(\theta) \\ \qquad 0 \le \theta < \cos^{-1}\left(\frac{kT_r}{2l}\right) \\ \\ l^2\left[\sin^{-1}\left(\frac{2l\cos(\theta)}{kT_r}\right) - \cos^{-1}\left(\frac{2l\cos(\theta)}{kT_r}\right)\right] - l^2 \\ -\frac{1}{8}k^2T_r^2\sec^2(\theta) + 2l\sqrt{\frac{1}{4}k^2T_r^2\sec^2(\theta) - l^2} \\ \qquad \cos^{-1}(\frac{kT_r}{2l}) \le \theta \le \cos^{-1}\left(\frac{kT_r}{2\sqrt{2}l}\right). \end{cases}$$

and

$$T = \frac{k^2T_r^2(3k^2T_r^2 - 32kT_rl + 24\pi l^2)}{96l^4}$$

for the typical case when $\frac{kT_r}{2} \le l$ (Appendix A). Figure 6 shows the obtained θ distribution.

2) Average utility function: To obtain a closed-form expression for k, we experimented with different approximations for the θ distribution (Appendices B and C). We found that the different approximations lead to the same optimal value as the exact distribution (as we show in Section IV-F). Therefore, we use the approximation that leads to the simplest computations. Specifically, assuming that θ follows a uniform distribution, and its values range from 0 to $\frac{\pi}{2}$; the average utility function $U(k)$ can be simplified as:

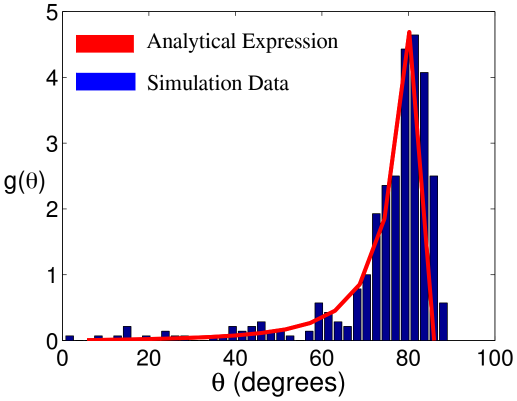
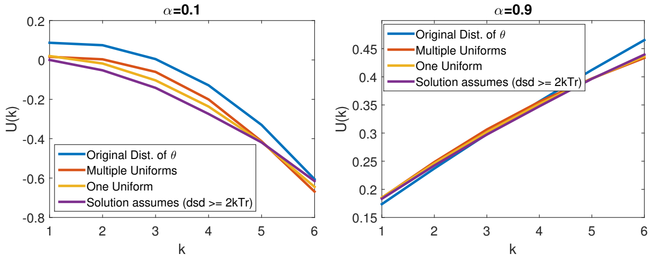


Fig. 6: θ distribution as drawn from the analytical formula and histogram of simulation data at k=2.



(a) α = 0.1 (b) α = 0.9

Fig. 7: Effect of changing k on the utility function for different values of α drawn for different representations for $U(k)$.

$$U(k) = -\alpha N n_{\text{pu}_k} \log_2(1 - p_{\text{pu}}) - (1-\alpha)(1 + d(k-1)^2 + 11.29dk(k-1)) \qquad (14)$$

3) Finding k^*: Solving for k^* leads to:

$$k^* = \left\lceil \frac{13.29d(1-\alpha) - \alpha N\frac{n_{\text{pu}}}{l^2}(\pi T_{r_{\text{pu}}}^2 - \text{area}_2)\log_2(1 - p_{\text{pu}})}{47.16d(1-\alpha)} \right\rceil \qquad (15)$$

F. Comparison Between Different Functions of U(k)

In this section, we compare the different approximation functions of $U(k)$: Assuming θ follows a uniform distribution (used in the previous section), different uniform distributions based on the value of k (Appendix B), as well as assuming that the destination is far away from the source, i.e. at least two mega-hops separate the source and destination (Appendix C). Figure 7 shows the value of $U(k)$ obtained using the exact θ distribution and the three approximations for different values of α. The figure shows that the four functions have the same behavior in terms of choosing the optimal k (which corresponds to the maximum value of $U(k)$). This finding gives us some key points about the effect of θ distribution. First, θ has only a slight effect on the value of $U(k)$ since it affects only the overhead metric without affecting the optimality one. Also, discretizing the value of k^* hides the effect of θ on it as k^* value changes slightly with the value of θ. Therefore, for simplicity and mathematical tractability, we rely on the uniform approximation to get the optimal value for k as described in the previous subsection. Moreover, Appendix D validates this finding via simulations.

G. Practical Notes

In this section, we discuss some practical considerations and apply them to the obtained equation of k^*. In particular, we note that the range of values for the overhead metric is different from that of the optimality metric. Therefore, we add a normalization constant N to make the ranges comparable as follows:

$$k^* = \left\lceil \frac{13.29d(1-\alpha) - \alpha N C \log_2(1-p_{pu})}{47.16d(1-\alpha)} \right\rceil \tag{16}$$

where C is constant and is given by:

$$C = \frac{n_{pu}}{l^2}(\pi T_{r_{pu}}^2 - \text{area}_2)$$

To obtain the value of N, we can use the condition on the value of k^* as follows:

$$k_{min} \leq k^* \leq k_{max}$$

where, k_{min} and k_{max} are the lower and the upper bound of the allowed values of k. Typically, the value of k_{min} is 1. On the other hand, the maximum possible value of k is when the source and the destination are on the two diagonals of the deployment area (on the network diagonal) so that the distance between them is $\sqrt{2}l$. In this case, the value of k^* should be limited by $k_{\text{max}} = \lceil \frac{\sqrt{2}l}{T_r} \rceil$. So, this leads to the following condition on the value of k^*:

$$1 \leq k^* \leq \left\lceil \frac{\sqrt{2}l}{T_r} \right\rceil$$

Consequently, the values of N can be given by:

$$N \geq \frac{33.87d(1-\alpha)}{-\alpha C \log_2(1-p_{\text{pu}})}$$

and

$$N \leq \frac{d(1-\alpha)\left(47.16\sqrt{2}\frac{l}{T_r} - 13.29\right)}{-\alpha C \log_2(1-p_{\text{pu}})}$$

Along the same line, to get the best value of N (which allows α to reflect the actual weight of both overhead and optimality metrics), we normalize both metrics and hide this normalization factor in N (Appendix E). In such case, N is given by:

$$N = \frac{dT_r^2 + 14dl^2 - 8\sqrt{2}T_r dl}{-3T_r^2 n_{\text{pu}} \log_2(1-p_{\text{pu}})} \tag{17}$$

Note that Equation (16) already captures the special cases when $\alpha = 0 (k = 1)$ and $\alpha = 1 (k = \infty)$.

H. Discussion

In this section, we discuss some assumptions and choices we have made in our model. First, we do not allow for the concurrent transmission of SUs and PUs as the main goal is to protect the PUs from the SUs interference. For example, choosing a high-latency route that guarantees not interfering with PUs is better than a low-latency one that may interfere with the PUs. Thus, we believe that other SUs-centric metrics, like delay and throughput, have a lower priority than protecting PUs from the SUs' interference. Based on that, we consider the number of PUs in the k-hop discovery and their activities.

Moreover, it is quite important to note that having geographical information is essential if $k \neq \infty$ as, in this case, data should be delivered to an intermediate node before reaching the destination (as in any location-aided routing protocol). However, if $k = \infty$, the protocol can work properly without the location information. But even in this case, we believe that having the geographical information reduces the complexity of exploring the route.

Finally, some assumptions were made for mathematical tractability, which can be relaxed in a future work. These assumptions include the uniformity of SUs locations in the deployment area, stationary SUs assumption, the unit disc model for all nodes channels, PUs homogeneity assumption, and the interference model of the SUs on the PUs. However, it is important to mention that we relax some of these assumptions in simulations as discussed in Section V.

V. Performance Evaluation

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

A. Simulation Setup

We used a multi-channel version of NS2 [35] for our simulations as well as a modified version of CAODV [8] as the underlying routing protocol on which we plug *PAK*. CAODV is an extension of the AODV protocol of ad hoc networks [36] that enables multiple channels and PUs awareness in CRNs. Our modified version works similar to the default CAODV but uses the optimality metric defined in the previous section as its route selection metric. SUs are deployed uniformly in the deployment area and the source and the destination in each experiment are selected randomly. We assume that PUs are independent from each other. This means that each of them can send or receive data independently from the other PUs. Following our system model, we assume that all PUs are

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMC.2018.2828104, IEEE Transactions on Mobile Computing

TABLE II: Experiments parameters.

Parameter	Value range	Nominal Value
Number of SUs n_{su}	10-200	100
Number of PUs n_{pu}	2-10	2
Number of active connections	1-20	10
SU transmission range T_r (m)	125	125
PU transmission range $T_{r_{\text{pu}}}$ (m)	140	140
Channel erasure probability	0.1	0.1
Maximum location error (m)	0 - 20	0
SUs speed (m/s)	0 - 20	0
Packet size (Byte)	512	512
Data rate per source (Kbps)	4 - 80	16
Network capacity (Mbps)	1.5	1.5
Square deployment area side len. l (m)	1000	1000
User utility parameter α	0.1-0.9	0.5
Activity period τ (sec.)	1	1
PUs activity parameter λ	0.5-8	0.5

homogenous; this includes the ON and OFF times parameters and their transmission ranges, as described in Section III. Finally, we use the IEEE 802.11 as the MAC protocol and the constant bit rate (CBR) traffic model for the generated traffic from the SUs. Each simulation experiment spans 200 seconds. All of the reported results are derived via simulations.

B. Experimental Parameters and Metrics

Table II summarizes the experimental parameters. PUs are uniformly located over the available channels in the area of interest. We also evaluate *PAK* using four metrics:

1) Throughput: number of bits transmitted correctly from source to destination per second. This is calculated by averaging the throughput of all sources over the whole simulation period.
2) Average end-to-end delay: average time taken by all packets to reach the destination from the source. This is calculated by averaging the delay of all packets transmitted from all sources over the whole simulation period.
3) Packet delivery ratio[3]: percentage of packets that reach the destination from all sent packets.
4) Routing overhead ratio: the ratio of the number of transmitted control packets to the total number of transmitted data packets.

Although the optimality metric we chose is the interference on the PUs, we measure the overall optimality of our protocol using general network performance metrics like throughput and the packet delivery ratio. In general, the network throughput is related to the interference level as [37]:

$$R = B\log(1 + \text{SINR}), \tag{18}$$

where R is the maximum user data rate (the available capacity), B is the available bandwidth, and SINR is the signal-to–interference-noise-ratio. Although we calculate throughput by getting the actual number of bits reached the destination, Equation 18 draws the relation between network throughput (used for evaluation) and the interference (used in the mathematical derivation). By avoiding PUs, SUs can reduce the effect of PUs interference at SUs and, at the same time, SUs can protect the PUs from the interference caused by SUs' transmission, i.e., protecting PUs results in interference reduction at both the PUs and the SUs.

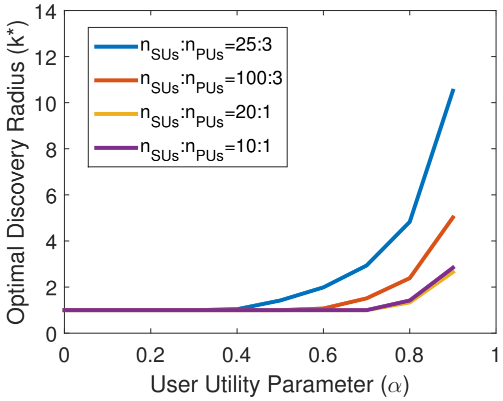


Fig. 8: Effect of changing α on k^* for different cases of network topology (different ratios of n_{su} to n_{pu}). Increasing α leads to favoring larger discovery radius (k) representing better route with higher overhead.

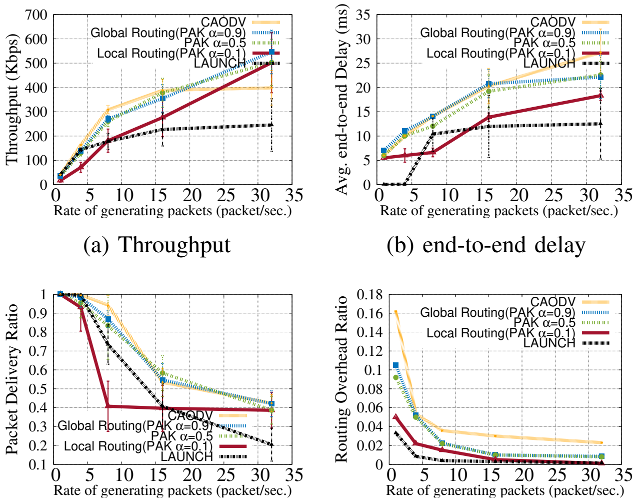
C. Experimental Results

We first study the effect of the user utility parameter (α) on performance. Then, we compare *PAK* with one protocol from each main category of routing approaches, the global and the local approaches, which are CAODV and LAUNCH, respectively. Finally, we show the effect of mismatching k on performance.

1) The effect of user utility parameter (α): Figure 8 shows the effect of α on k^*. The figure is drawn for different network setups in terms of the number of secondary and primary users. Setting α to a low value favors smaller k, which maps to the traditional local routing. On the other hand, choosing a high value for α leads to a high value for k, leading to the global routing approach in the extreme case. More importantly, the figure shows that the optimal value of k ***changes significantly*** based on the user utility, which can be achieved by *PAK*, as compared to traditional routing protocols for CRNs.

2) Capturing the spectrum gap between local and global routing: In Figure 9, we show a comparison between *PAK*, with different values of α, and two other protocols which are CAODV [8] and LAUNCH [18]. CAODV is a global routing protocol (corresponding to α = 1), and LAUNCH is an example of local routing protocols (corresponding to *PAK* with α = 0). We can see that using *PAK* with choosing different values for α can efficiently span the spectrum between the two extreme approaches of routing. Thus, a user can use *PAK* as a local routing protocol (by setting α = 0) or as a global one (by setting α = 1).

[3] In our system model, we assume the disc channel model in which no transmission errors exist; this is for mathematical tractability. However, in simulations, we release this assumption by assuming a fading channel in which wireless transmission errors may occur. Therefore, the packet delivery ratio has a meaning in this case.



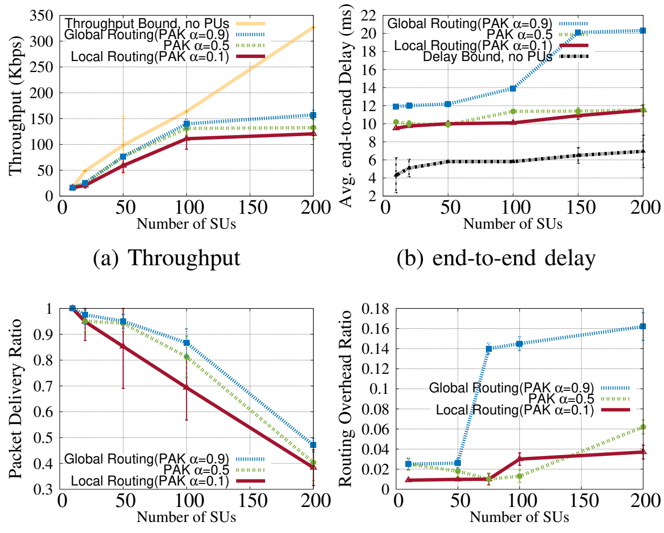
(a) Throughput (b) end-to-end delay

(c) Packet Delivery Ratio (d) Routing Overhead Ratio

Fig. 9: Comparing *PAK* with CAODV and LAUNCH and studying the effect of changing packets generation rate on the required metrics. Note that setting α to 0.1 corresponds to setting k to 1 which is the typical value of discovery radius in local routing protocols. On the other hand, setting α to 0.9 corresponds to having $k = \infty$ which is the case of global routing protocols. This note can be also applied to all figures in this section.

Moreover, Figure 9 shows the impact of increasing the rate of generated packets on different performance metrics. Increasing the offered load leads to increased throughput and reduced routing overhead at the expense of increasing the end-to-end delay and decreasing the packet delivery ratio. Nonetheless, Figure 9 also shows that using a fixed k makes the user stuck at one of these extremes (for example, using CAODV will get the optimal route but with a high overhead). However, using *PAK*, each user can independently select his desired utility function and hence can achieve different performance metrics fitting his own criteria.

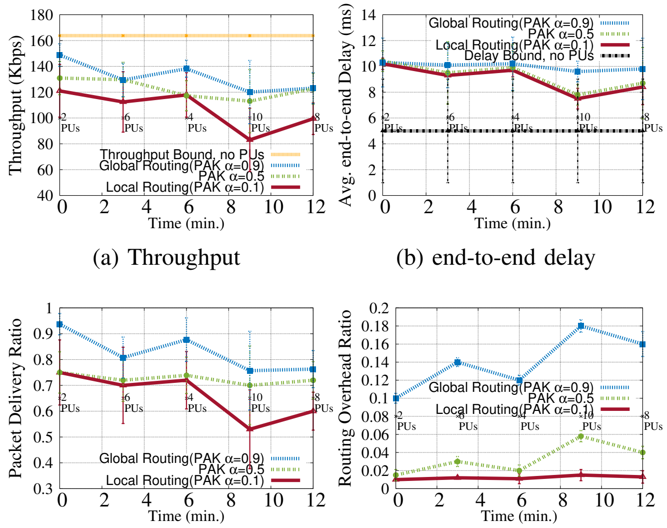
3) The effect of changing SUs density: Figure 10 shows the effect of changing the SUs density on performance. The effect is similar to the effect of the increasing the packet generation rate with the exception that increasing SUs leads to increasing the routing overhead, due to the extra route discovery overhead with new users, as compared to just increasing the offered load without increasing the number of users. This figure also includes bounds on both throughput and end-to-end delay (Figures 10a and 10b)[4]. These bounds define the maximum throughput and the minimum delay we can achieve in such network configurations. The maximum throughput is obtained through applying a global routing approach (i.e., $k = \infty$) with the absence of PUs. On the other hand, we get the minimum end-to-end delay through applying a shortest path algorithm in a similar network without PUs too. We believe that these values constitute the bounds on our protocol performance in such networks.



(a) Throughput (b) end-to-end delay

(c) Packet Delivery Ratio (d) Routing Overhead Ratio

Fig. 10: Effect of changing the number of SUs (while fixing the percentage of sources to the total number of nodes to 10%).



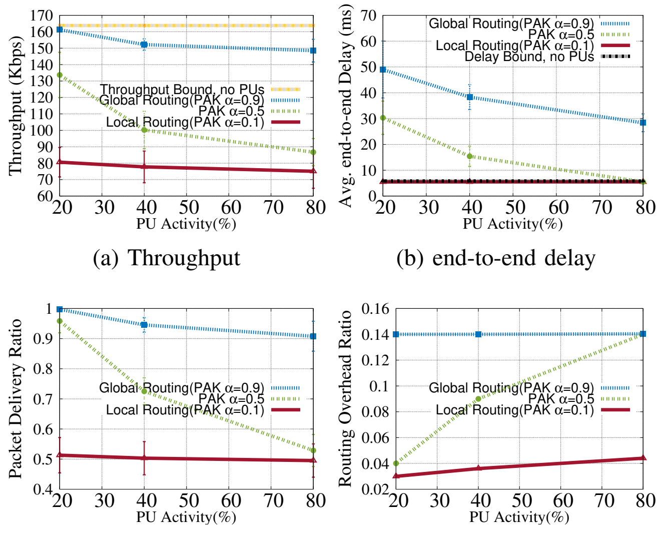
(a) Throughput (b) end-to-end delay

(c) Packet Delivery Ratio (d) Routing Overhead Ratio

Fig. 11: Effect of changing the number of PUs (as an example of changing network dynamics) with time on different metrics. We can see that *PAK* (at $\alpha = 0.5$) adapts well with the topological change to give better performance than the routing protocols that use a fixed value for k.

4) The effect of changing network dynamics with time: In Figure 11, we show how *PAK* adapts to dynamic changes with time. As an instance of changing the network dynamics, we choose to change the number of PUs. Based on that, in this figure, we show how PAK adapts to changing the number of PUs over time to preserve the user-defined balance between both the optimality and the overhead metrics. We can see that *PAK* can select an appropriate value for k as the network topology changes whereas fixing k to a certain value (as in

[4] The same applies to the next two figures (Figures 11 and 12).



(a) Throughput (b) end-to-end delay

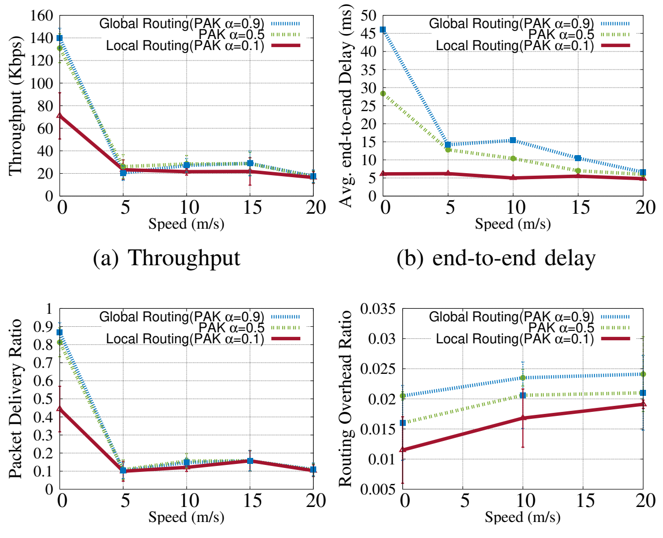
(c) Packet Delivery Ratio (d) Routing Overhead Ratio

Fig. 12: Effect of changing the PUs activity on the performance metrics.

the global or local approach) gives an overall bad performance. Figure 11a and 11c show that the local approach performance degrades significantly when the number of PUs increases. This happens since the local routing does not account for the expected activities of PUs along an extended path. 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 11b and 11d where the overhead increases significantly when the number of PUs increases at some time in the case of global routing (fixing $k = \infty$).

5) The effect of changing the PUs activities: Figure 12 shows the effect of changing the PUs activities on the performance metrics. Generally, increasing the PUs activities leads to worse performance in terms of throughput and packet delivery ratio. However, using a higher value for α leads to better and more robust routes as more PUs are discovered. Choosing a route that interferes with a highly-active PUs may lead to being idle (SU that cannot transmit) for a long time; this affects the performance greatly.

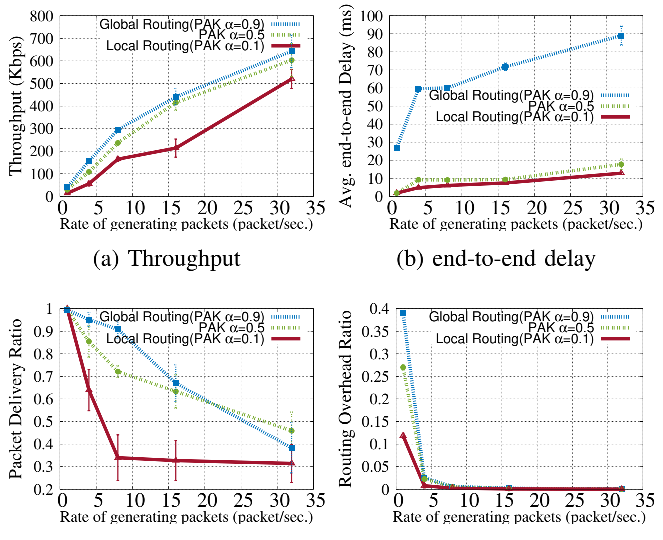
6) The effect of changing the SUs speed: Figure 13 shows the effect of changing the SUs speed on the performance of *PAK*. Although we assume that SUs are stationary, for the mathematical tractability, we experiment, in this figure, with SUs which are moving[5] according to the Random Way Point model [38]. Generally, the performance, in terms of throughput and packet delivery ratio, decreases with increasing the nodes' speed where the overhead increases at the same time. This is due to the instability of the found routes when the nodes are moving, especially at high speeds. At this point, the performance of *PAK*, while using different values of α, converges to be nearly the same. However, we can still see that the routing overhead is higher when using a higher value of α. Thus, the user can still control the overhead using the offered parameter α.



(a) Throughput (b) end-to-end delay

(c) Packet Delivery Ratio (d) Routing Overhead Ratio

Fig. 13: Effect of changing the SUs speed on the performance metrics.



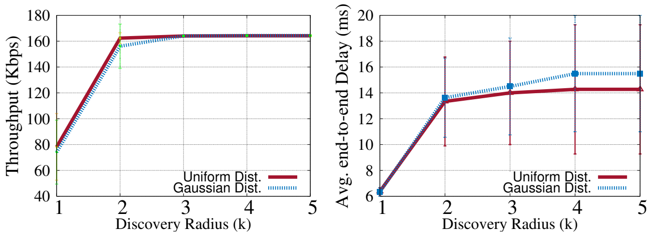
(a) Throughput (b) end-to-end delay

(c) Packet Delivery Ratio (d) Routing Overhead Ratio

Fig. 14: Effect of changing the packets generation rate while distributing SUs' locations according to the Gaussian distribution.

7) The effect of changing the SUs' locations distribution: Figure 14 shows the effect of changing rate when SUs are distributed according to the Gaussian distribution. In our analysis, we assume that SUs are distributed uniformly in the deployment area. This assumption is motivated by the fact that we address adhoc networks, in which nodes do not have a known specific shape or distribution. However, in this figure, we evaluate the performance of *PAK* when SUs are distributed according to the Gaussian distribution. Most of the nodes are located near the center of the deployment area (mean is the

[5] Source and destination nodes are moving too in the same way like other SUs.



(a) Throughput

(b) end-to-end delay

Fig. 15: The effect of changing k on the performance metrics when SUs are deployed randomly according to two different distributions, namely the Uniform and the Gaussian distributions.

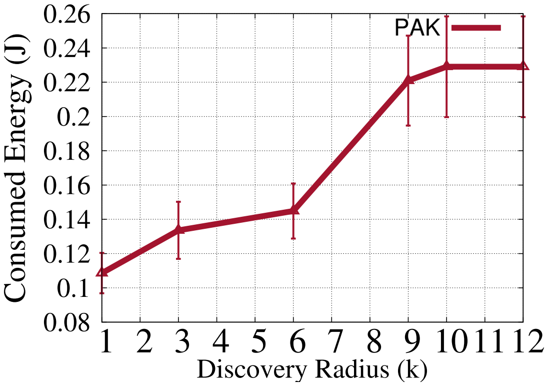
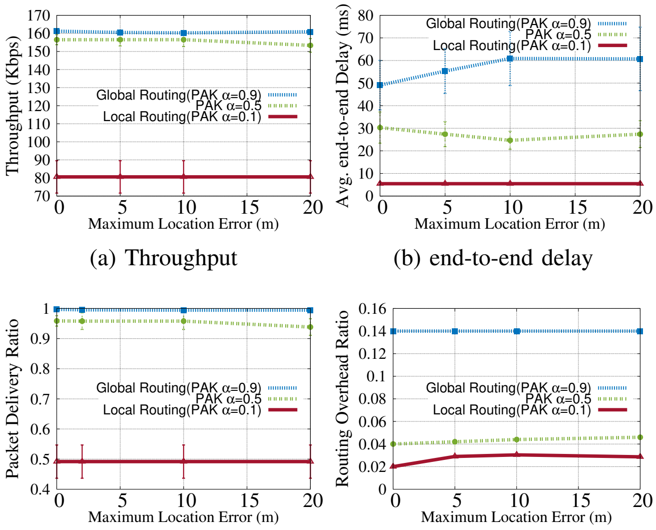


Fig. 16: Energy consumed by the whole network, due to the control packets transmissions, while changing the discovery radius (k).

center of the deployment area) with a variance equal to $100m$. *PAK* has similar performance, with different values for α, to the already-discussed figures. Setting a higher value for α leads to a higher throughput and overhead as well while setting a smaller one to α decreases the route optimality and the routing overhead.

Along the same line, Figure 15 shows the effect of changing k on the performance metrics in case of deploying SUs according to two different distributions: Uniform and Gaussian distributions. It shows that changing k has the same effect on performance metrics for both distributions. Based on that, k^* would not change with changing the distribution from Uniform to Gaussian.

8) Evaluating the energy consumption: Figure 16 shows the energy consumed in the whole network due to the control packets transmissions[6]. For the energy consumption model, we use the same transmission and reception model and parameters proposed by [39]. We can see that the energy consumed increases with increasing the discovery radius (k) as more transmissions are required. Thus, global protocols (which use larger radii) consume more energy than local ones. Moreover, increasing k above some threshold ($k = 10$ here) will not consume more energy as this radius is larger than the network diameter; in this case, *PAK* performs like the global protocols



(a) Throughput

(b) end-to-end delay

(c) Packet Delivery Ratio

(d) Routing Overhead Ratio

Fig. 17: Effect of different errors in estimating nodes' locations on the performance metrics.

(in which $\alpha = 1$ and $k = \infty$).

9) The effect of location uncertainty: In our system model, we assume that we have no errors in estimating the nodes' locations. Figure 17 shows the effect of having some error in estimating the nodes' locations on *PAK*'s performance. In this figure, we show that having a bounded (small) error in estimating the location does not affect the system performance. As the accuracy of the modern localization systems is a few meters, we test the effect of (up to) 20 meters error in estimating the nodes' locations. We can see that this error in nodes' localization does not, generally, affect the performance. This happens as the location information is used for guiding the routing process towards the destination location. Thus, few meters error will not have significant effect on the system performance.

VI. Conclusion

We propose a new scheme for adaptive routing discovery in CRNs where the number of hops to be discovered can be adapted with the network topology based on a user-defined utility function. We study the tradeoff between the route optimality and the routing overhead as a function of the number of discovered hops both analytically and through simulations. Results show the advantages of the proposed scheme and how it can adapt to different user requirements. This work can be expanded in several directions. First, we can apply our discovery scheme over different classes of routing protocols in CRNs. Another future direction is to investigate new and more complex mathematical approaches for deriving the optimal k. Furthermore, experimenting our approach while having multiple channels and a channel selection scheme would be a good direction for future research. Finally, we are planning to extend our protocol by explicitly considering other functions for the optimality metric.

[6] Energy consumed due to the data transmissions can be extracted too from Figure 16 and the routing overhead ratio figures.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMC.2018.2828104, IEEE Transactions on Mobile Computing

References

[1] Peter Steenkiste, Douglas Sicker, Gary Minden, and Dipankar Raychaudhuri. NFS Workshop on Future Directions in Cognitive Radio Network Research. Technical report, 2009.

[2] Suman Banerjee, Dapeng Oliver Wu, Xiaojun Lin, and Xinyu Zhang. NFS Workshop on Future Directions in Wireless Networking. Technical report, 2013.

[3] DARPA. Spectrum Collaboration Challenge. https://spectrumcollaborationchallenge.com/, 2016. [Online; accessed 28-October-2017].

[4] E Fadel, Muhammad Faheem, Vehbi C Gungor, Laila Nassef, N Akkari, Muhammad Ghulam Abbas Malik, Suleiman Almasri, and Ian F Akyildiz. Spectrum-aware bio-inspired routing in cognitive radio sensor networks for smart grid applications. *Computer Communications*, 101:106–120, 2017.

[5] Moustafa Youssef, Mohammad Ibrahim, Mohamed Abdelatif, Lin Chen, and Athanasios V Vasilakos. Routing metrics of cognitive radio networks: A survey. *Communications Surveys & Tutorials, IEEE*, 16(1):92–109, 2014.

[6] Shriram Kulkarni and Shriram Markande. Comparative study of routing protocols in cognitive radio networks. In *Pervasive Computing (ICPC), 2015 International Conference on*, pages 1–5. IEEE, 2015.

[7] Angela Sara Cacciapuoti, Marcello Caleffi, and Luigi Paura. Reactive Routing for Mobile Cognitive Radio Ad Hoc Networks. *Ad Hoc Networks*, 10(5):803–815, 2012.

[8] Angela Sara Cacciapuoti, Cosimo Calcagno, Marcello Caleffi, and Luigi Paura. CAODV: Routing in mobile ad-hoc cognitive radio networks. *Wireless Days (WD)*, 2010.

[9] Ashwin Sampath, Lei Yang, Lili Cao, Haitao Zheng, and Ben Y Zhao. High throughput spectrum-aware routing for cognitive radio networks. *Proc. of IEEE Crowncom*, 2008.

[10] Bing Xia, M.H. Wahab, Yang Yang, Zhong Fan, and M. Sooriyabandara. Reinforcement learning based spectrum-aware routing in multi-hop cognitive radio networks. In *Cognitive Radio Oriented Wireless Networks and Communications. CROWNCOM '09. 4th International Conference on*, june 2009.

[11] Xiaoxia Huang, Dianjie Lu, Pan Li, and Yuguang Fang. 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.

[12] Cornelia-Ionela Badoi, Victor Croitoru, and Ramjee Prasad. IPSAG: An IP Spectrum Aware Geographic Routing Algorithm Proposal for Multi-hop Cognitive Radio Networks. In *Communications (COMM), 8th International Conference on*, pages 491–496. IEEE, 2010.

[13] Wooseong Kim, Mario Gerla, Soon Y Oh, Kevin Lee, and Andreas Kassler. CoRoute: A New Cognitive Anypath Vehicular Routing Protocol. *Wireless Communications and Mobile Computing*, 11(12):1588–1602, 2011.

[14] Xing Tang, Yanan Chang, and Kunxiao Zhou. Geographical Oportunistic Routing in Dynamic Multi-hop Cognitive Radio Networks. In *Computing, Communications and Applications Conference (ComComAp)*, pages 256–261. IEEE, 2012.

[15] Min Xie, Wei Zhang, and Kai-Kit Wong. A Geometric Approach to Improve Spectrum Efficiency for Cognitive Relay Networks. *Wireless Communications, IEEE Transactions on*, 9(1):268–281, 2010.

[16] Junseok Kim and Marwan Krunz. Spectrum-Aware Beaconless Geographical Routing Protocol for Mobile Cognitive Radio Networks. In *Global Telecommunications Conference (GLOBECOM), IEEE*, pages 1–5. IEEE, 2011.

[17] Feilong Tang, Leonard Barolli, and Jie Li. 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.

[18] Karim Habak, Mohammed Abdelatif, Hazem Hagrass, Karim Rizc, and Moustafa Youssef. A location-aided routing protocol for cognitive radio networks. In *IEEE ICNC*, 2013.

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

[20] Yanhua Li, Abedelaziz Mohaisen, and Zhi-Li Zhang. Trading optimality for scalability in large-scale opportunistic routing. *IEEE T. Vehicular Technology*, 62(5):2253–2263, 2013.

[21] Martin Mauve, Jörg Widmer, and Hannes Hartenstein. A survey on position-based routing in mobile ad hoc networks. *Network, IEEE*, 15(6):30–39, 2001.

[22] Silvia Giordano, Ivan Stojmenovic, and Ljubica Blazevic. Position based routing algorithms for ad hoc networks: A taxonomy. *Ad hoc wireless networking*, 8:64, 2003.

[23] Ljubica Blažević, Silvia Giordano, and Jean-Yves Le Boudec. Self organized terminode routing. *Cluster Computing*, 5(2):205–218, 2002.

[24] Ben Liang and Zygmunt J Haas. Hybrid routing in ad hoc networks with a dynamic virtual backbone. *Wireless Communications, IEEE Transactions on*, 5(6):1392–1405, 2006.

[25] Kaushik R Chowdhury and Marco Di Felice. Search: A routing protocol for mobile cognitive radio ad-hoc networks. *Computer Communications*, 32(18):1983–1997, 2009.

[26] Shishir Borkar and SM Ali. Enhancing opportunistic routing for cognitive radio network. 2017.

[27] Brad Karp and Hsiang-Tsung Kung. 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.

[28] Won-Yeol Lee and Ian F Akyildiz. Optimal spectrum sensing framework for cognitive radio networks. *Wireless Communications, IEEE Transactions on*, 7(10):3845–3857, 2008.

[29] Alberto Cerpa, Jennifer L Wong, Louane Kuang, Miodrag Potkonjak, and Deborah Estrin. Statistical model of lossy links in wireless sensor networks. In *Proceedings of the 4th international symposium on Information processing in sensor networks*, page 11. IEEE Press, 2005.

[30] P. Enge and P. Misra. Special issue on GPS: The Global Positioning System. *Proceedings of the IEEE*, 1999.

[31] M. Ibrahim and M. Youssef. CellSense: An Accurate Energy-Efficient GSM Positioning System. *Vehicular Technology, IEEE Transactions on*, 61(1):286–296, 2012.

[32] Andreas Savvides, Chih-Chieh Han, and Mani B Strivastava. Dynamic fine-grained localization in ad-hoc networks of sensors. In *Proceedings of the 7th annual international conference on Mobile computing and networking*, pages 166–179. ACM, 2001.

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

[34] AM Mathai, P Moschopoulos, and G Pederzoli. Random points associated with rectangles. *Rendiconti del Circolo Matematico di Palermo*, 48(1):163–190, 1999.

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

[36] C. Perkins, E. Belding-Royer, and S. Das. Ad hoc on-demand distance vector (AODV) routing, 2003.

[37] Thomas M. Cover and Joy A. Thomas. *Elements of Information Theory*. Wiley-Interscience, New York, NY, USA, 1991.

[38] Guolong Lin, Guevara Noubir, and Rajmohan Rajaraman. Mobility models for ad hoc network simulation. In *INFOCOM 2004. Twenty-third AnnualJoint Conference of the IEEE Computer and Communications Societies*, volume 1. IEEE, 2004.

[39] Wendi Rabiner Heinzelman, Amit Sinha, Alice Wang, and Anantha P Chandrakasan. Energy-scalable algorithms and protocols for wireless microsensor networks. In *Acoustics, Speech, and Signal Processing,*

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMC.2018.2828104, IEEE Transactions on Mobile Computing

2000. ICASSP'00. Proceedings. 2000 IEEE International Conference on, volume 6, pages 3722–3725. IEEE, 2000.

Arsany Guirguis is a Research Assistant at Computer and Systems Department, Faculty of Engineering, Alexandria University, Egypt. He received his B.Sc. and M.Sc. in Computer and Systems Engineering from Alexandria University, Egypt in 2014 and 2017 respectively. His research interests include cognitive radio networks, routing protocols and cooperative routing. Arsany is a recipient of the Certificate of Honor for being ranked second in College of Engineering and first on Computer and Systems Department, Alexandria University in 2014. He has also received the Best Graduation Project Award from the same department in 2014.

Dr. Fadel Digham is the Executive Director of the Research and Development (R&D) department at the National Telecom Regulatory Authority (NTRA) of Egypt. Dr Digham is also an Adjunct Professor at Nile University and Instructor at Cairo University. He worked in the past for both Academia and Industry including the University of Minnesota in USA, Mitsubishi research labs in USA, and Alcatel in Egypt. He is Co-chairing a joint study group on spectrum management at the International Telecommunication Union (ITU). Dr. Digham is a member in the steering committee of the IoT forum in Egypt and was a board member of the Industry Advisory Board of the IEEE Wireless Communications Engineering Technologies Certification.

Dr. Fadel received his B.Sc. (with honor) in Electronics and Communications from Cairo University in 1995, and his PhD from the University of Minnesota (Twin Cities) in USA in 2005.

Karim G. Seddik (S' 04, M' 08, SM' 14) is an associate professor in the Electronics and Communications Engineering Department at the American University in Cairo (AUC), Egypt. Before joining AUC, he was an assistant professor in the Electrical Engineering Department at Alexandria University, Egypt. Dr. Seddik received the B.S. (with highest honors) and M.S. degrees in electrical engineering from Alexandria University, Alexandria, Egypt, in 2001 and 2004, respectively. He received his Ph.D. degree at the Electrical and Computer Engineering Department, University of Maryland, College Park 2008. His research interests include cooperative communications and networking, MIMO–OFDM systems, cognitive radio, layered channel coding, and distributed detection in wireless sensor networks.

Dr. Seddik has served on the technical program committees of numerous IEEE conferences in the areas of wireless networks and mobile computing. Dr. Seddik is a recipient of the Certificate of Honor from the Egyptian President for being ranked first among all departments in College of Engineering, Alexandria University in 2002, the Graduate School Fellowship from the University of Maryland in 2004 and 2005 and the Future Faculty Program Fellowship from the University of Maryland in 2007.

Mohamed Ibrahim is working toward a Ph.D. degree from the Department of Computer Science, Rutgers University. He is a research assistant in Wireless Information Network Lab (WINLAB), where he has been working on mobile systems including mobile context sensing, visible light communication and privacy protection for context services. He received his B.Sc degree in Computer Science from Alexandria University in Egypt, in 2009, and M.Sc. in Communications and Wireless Technology from Nile University in Egypt, in 2011. His research interests include mobile wireless networks, sensor networks, location determination technologies, and pervasive computing. He is the recipient of the best paper award at the ACM VLCS Workshop, held with MobiCom 2016 conference.

Dr. Khaled Harras is an Associate Professor at Carnegie Melon University Qatar (CMUQ), and Director of the Computer Science program there. Dr. Harras is the founder and director of the Networking Systems Lab (NSL) at CMUQ. For the past 14 years, he has been working on the areas of delay/disruption tolerant networks, intermittent networking, video sensor networks, unmanned air vehicles (UAVs), bandwidth aggregation in multi-interfaced devices, mobile edge network architectures, and mobile computational offloading. He also has expertise in the domains of wireless and mobile networks measurement, building real systems and testbeds, designing and implementing architectures and frameworks based on real data. Dr. Harras has collaborations with various institutions and companies including QU, TAMUQ, QCRI, Nile University, EJUST, UCSB, Georgia Tech, and Boeing. He has more than 80 refereed publications in international prestigious journals, conferences, and workshops, and 4 US patents. Along with his research group in the past few years, he has won the best computing research award in Qatar twice, and received two best paper awards. To date, he has been involved in or managing research grants that amount to more than 3 Million USD, and has supervised over 30 different personnel including undergraduate and graduate students, postdoctoral researchers, and research engineers.

Moustafa Youssef is a Professor at Egypt-Japan University of Science and Technology (E-JUST) and Founder & Director of the Wireless Research Center of Excellence, Egypt. His research interests include mobile wireless networks, mobile computing, location determination technologies, pervasive computing, and network security. He is an associate editor for the ACM TSAS, a previous area editor of the ACM MC2R and served on the organizing and technical committees of numerous prestigious conferences. Prof. Youssef is the recipient of the 2003 University of Maryland Invention of the Year award, the 2010 TWAS-AAS-Microsoft Award for Young Scientists, the 2012 Egyptian State Award, the 2013 and 2014 COMESA Innovation Awards, the 2013 ACM SIGSpatial GIS Conference Best Paper Award, among many others. He is also an ACM Distinguished Scientist and an ACM Distinguished Speaker.