Computer and Information Science; Vol. 8, No. 3; 2015
ISSN 1913-8989 E-ISSN 1913-8997
Published by Canadian Center of Science and Education

Maximum Lifetime Communication Topologies of Secondary User Nodes in Cognitive Radio Ad Hoc Networks

Natarajan Meghanathan[1]

[1] Department of Computer Science, Jackson State University, Jackson, MS, USA

Correspondence: Natarajan Meghanathan, Department of Computer Science, Mailbox 18839, Jackson State University, Jackson, MS 39217, USA. Tel: 1-601-979-3661. E-mail: natarajan.meghanathan@jsums.edu

Received: March 14, 2015 Accepted: April 1, 2015 Online Published: April 6, 2015

doi:10.5539/cis.v8n3p1 URL: http://dx.doi.org/10.5539/cis.v8n3p1

The research is financed by the NASA EPSCoR sub award (#: NNX14AN38A) from University of Mississippi.

Abstract

A Cognitive radio ad hoc network (CRAHN) is an ad hoc network of primary user (PU) nodes and secondary user (SU) nodes, wherein the SU nodes use the licensed channels of the PU nodes when not in use. As the availability of the PU channels fluctuates with time, communication topologies spanning the SU nodes have to be dynamically and frequently reconfigured. Given the complete knowledge of the availability of the PU channels, we propose a generic benchmarking algorithm that determines a sequence of stable communication topologies spanning all of the SU nodes such that the number of transitions from one instance of the topology to another is the global minimum. At any time instant t when we need a stable communication topology spanning the entire network, we look for the largest value of k such that the intersection of the static SU graphs from time instants t to $t+k$, defined as the mobile graph $G_{t...t+k}(\text{SU}) = G_t(\text{SU}) \cap G_{t+1}(\text{SU}) \cap .... \cap G_{t+k}(\text{SU})$, is connected and that the mobile graph $G_{t...t+k+1}(\text{SU})$ is not connected. We repeat the above procedure for the entire network session to determine a sequence of longest-living instances of the mobile graphs and the corresponding instances of the communication topology of interest (say a shortest path tree rooted at a source SU node) such that the number of topology transitions is the minimum. We prove the theoretical correctness of the algorithm and evaluate its effectiveness by implementing it to determine a sequence of shortest path trees of the maximum lifetime.

Keywords: benchmarking algorithm, cognitive radio ad hoc networks, communication topology, minimum hop routing, shortest path trees, stability

1. Introduction

Cognitive radios are software-defined radios that can dynamically adapt their transmission parameters to suit to the available channels (frequencies) in the operating environment (Akyildiz et al., 2006). A cognitive radio network (CRN) is a wireless network whose devices are embedded with cognitive radios that can sense the available channels in the neighborhood and switch the communication channel, if needed. Cognitive radios are considered a promising solution to alleviate the problem of spectrum scarcity (Zhao & Sadler, 2007). A CRN comprises of two types of users: primary users (PUs) and secondary users (SUs); the PUs own licensed channels and the SUs do not own any licensed channel (Akyildiz et al., 2006). The SUs employ the available licensed channels of the PUs that are not in use. When the PUs of the currently used channels become active, an SU either relinquishes the channel and switches to any other available PU channel, called spectrum overlay, or reduces its transmission range and continues to use the channel without interfering the licensed PU, called spectrum underlay (Zhao & Sadler, 2007). For the rest of this paper, we restrict ourselves to the spectrum overlay approach.

Our focus in this paper is on a category of CRNs called the cognitive radio ad hoc networks (CRAHNs): a self-organized network comprising of the PU nodes and SU nodes (Akyildiz et al., 2009). The PU nodes and SU nodes operate with a limited transmission range and are peers of each other. For simplicity, we assume each PU node to own a distinct licensed channel, identified using the PU node's ID. A PU channel is considered to be available for use by an SU node if the two nodes are within the transmission range of each other and that the PU node is turned OFF (i.e., not active). Two SU nodes are connected with a link if they share at least one available

PU channel in their common neighborhood. Depending on the activity status of the PU nodes, the set of PU channels available for the SU nodes to use changes dynamically with time. Thus, even if the SU nodes and PU nodes are static, communication topologies (like paths and trees) that connect the SU nodes may have to be frequently reconfigured based on the PU channels' availability in the SU-SU neighborhoods. The routing solutions for CRAHNs proposed so far have focused mainly on discovering optimal routes with respect to traditional metrics like hop count (Xin et al., 2005), end-to-end delay, with the channel switch delay considered an integral component of end-to-end delays (Cheng et al., 2007a; Cheng et al., 2007b; Ma et al., 2008), throughput (Shi & Hou, 2008), energy consumption (Pyo & Hasegawa, 2007) and etc. Not much work has been done on determining stable paths or trees whose lifetime is longer. Specifically, to the best of our knowledge, we could not find any work on determining stable sequence of a communication topology that spans (i.e., connect all the SU nodes) the entire CRAHN network of SU nodes.

Our objective in this paper is to develop a generic benchmarking algorithm that can be used to arrive at upper bounds for the lifetime of any communication topology that spans the SU nodes. We call the algorithm as the Maximum Lifetime Communication Topology (MLCT) algorithm. Given that there is a polynomial-time algorithm or heuristic to determine the communication topology of interest (say, shortest path tree, minimum spanning tree, connected dominating set, etc) that spans the entire network of SU nodes, the objective of the MLCT algorithm is to find a stable sequence of instances of the communication topology for the lifetime of the network. In this paper, we choose shortest path trees as the representative communication topology of interest, as these trees are widely used to determine minimum hop paths from a source node to every other node in a network (a standard routing problem in any network), including CRAHNs. As our focus here is to develop a benchmarking algorithm, we take a centralized approach and assume the availability of any relevant information that could contribute towards arriving at the best solution. Accordingly, we assume that the availability status (ON or OFF) of every PU channel at any time instant within the duration of the network session is known before-hand.

We model the CRAHN network of SU nodes at any time t as a static graph G_t(SU) comprising of edges between any two SU nodes if and only if the Euclidean distance between the two nodes is less than or equal to the transmission range of the nodes and if there is at least one PU channel available in their mutually intersecting neighborhood at t. Assuming the information about PU channel availability for future is known a priori, the proposed MLCT algorithm determines a sequence of maximum lifetime instances of the communication topology of interest (say, shortest path trees rooted at a source SU node s) as follows: At any particular time instant t when we want to find a stable instance of the communication topology of interest, we determine a long-living mobile graph $G_{t...t+k}$(SU) = G_t(SU) ∩ G_{t+1}(SU) ∩ .... ∩ G_{t+k}(SU) such that $G_{t...t+k}$(SU) is connected (i.e., spans all the SU nodes), but $G_{t...t+k+1}$(SU) is not connected. We simply run an appropriate algorithm to determine the communication topology of interest, like the Breadth First Search algorithm (Cormen et al., 2009) for shortest path trees, on $G_{t...t+k}$(SU) and such a topology is guaranteed to exist in each of the constituent static graphs G_t(SU), G_{t+1}(SU), ..., G_{t+k}(SU). We repeat the above procedure for the entire network session to determine a stable sequence of instances of the mobile graphs and the corresponding communication topology of interest such that the number of transitions from one instance of the topology to another instance in the sequence is the global minimum. The average of the lifetimes of the mobile graphs form an upper bound (benchmark) for the average lifetime of any communication topology (be it a shortest path tree, spanning tree, connected dominating set, etc., determined by a centralized or distributed algorithm) that spans the entire network of SU nodes in the CRAHN under identical operating conditions.

We use shortest path tree as the representative communication topology of interest and show how the MLCT algorithm could be used to determine a sequence of maximum lifetime shortest path trees (MLSPTs) such that the average of the lifetimes of the MLSPTs form an upper bound to the average lifetime incurred for shortest path trees in CRAHNs. We compare the performance of MLSPTs to a sequence of minimum-hop based shortest path trees (MHSPTs) that are determined as explained below: At any particular time instant t when we are in need of a minimum-hop shortest path tree (rooted at a particular source node s), we run the BFS (Breadth First Search) algorithm on G_t(SU) and determine a minimum-hop shortest tree rooted at s and continue to use the tree in the ensuing static SU graphs, G_{t+1}(SU), G_{t+2}(SU), ... as long as the tree exists. Upon failure to exist in a static graph G_{t+j}(SU), we find a new minimum-hop shortest path tree in G_{t+j}(SU) and continue to use it until it exists. This procedure is continued for the entire network session. The above approach taken for MHSPTs of not looking at the future and choosing the best topology based on the current static graph G_t(SU) is the standard approach used by the distributed routing protocols (e.g., Perkins & Royer, 1999; Meghanathan, 2009) for ad hoc networks.

Simulation results indicate the lifetime of MLSPTs to be significantly higher than that of MHSPTs, especially with increase in the number of PU channels and their availability time in the SU-SU neighborhood. Since the root nodes of the shortest path trees are randomly chosen for each time instant, the average height of a shortest path tree in the simulations could be construed as a measure of the average network diameter (i.e., the maximum of the minimum number of hops between two nodes in the network). We observe that there is very minimal increase in the average height of the MLSPTs compared to that of the MHSPTs. Thus, there is no tradeoff between stability and hop count while determining benchmarks for maximum lifetime shortest path trees in CRAHNs.

The rest of the paper is organized as follows: Section 2 presents the network model and assumptions. Section 3 presents the MLCT algorithm to determine stable sequence of any network-wide communication topology and its application to determine maximum lifetime shortest path trees. We also prove the correctness of the algorithm, analyze its run-time complexity as well as illustrate its working to determine MLSPTs through an example. Section 4 presents simulation results comparing the performance of the MLSPTs and MHSPTs with respect to tree lifetime and height. Section 5 discusses related work in the literature for routing in CRAHNs and our contributions in this paper. Section 6 concludes the paper. Throughout the paper, we use the terms 'route' and 'path', 'edge' and 'link', 'node' and 'vertex' interchangeably; they mean the same.

2. Network Model and Assumptions

We assume a static and centralized setup of the CRAHN comprising of the primary user nodes (PU nodes) and secondary user nodes (SU nodes), each of which are uniquely identified by an ID. Each PU node is assumed to own a licensed channel that has a unique frequency and is identified with the ID of the PU node itself. An SU node identifies a PU channel on the basis of the latter's ID. The SU nodes and PU nodes are assumed to be randomly distributed in a 2-dimensional network. The transmission range (R) for both the categories of nodes is assumed to be identical. Using the node locations and transmission range R, we could identify the PU node neighbors for each SU node. A PU node is said to be a neighbor of an SU node if the Euclidean distance between them is within the transmission range. Accordingly, we say that a PU channel is available for use by an SU node at a particular time instant only if the corresponding PU node is in the neighborhood of the SU node (which is always the case, as the nodes are static) and that the PU node is turned OFF and not using its licensed channel at that time instant.

Each PU channel is independently turned OFF and ON (alternately) for the entire network session. To begin with, a PU channel is randomly turned ON or OFF and continues to stay so for a time period randomly chosen from the range [0…MAX_{Random_ON}] or [0…MAX_{Random_OFF}] respectively. After this, the PU channel changes its status from ON to OFF or OFF to ON (depending on the case) and chooses to be in the new status for a random time period selected from the above range. This procedure is repeated throughout the network session.

We model two types of SU graphs: (i) A time-invariant SU graph G_i(SU) comprises of only the SU nodes and there exists an SU-SU edge only if the Euclidean distance between the end-to-end SU nodes of the edge is within the transmission range, R. (ii) A time-variant SU graph G_t(SU) that comprises of both the SU nodes and PU nodes; there exists an SU-SU edge only if the edge exists in G_i(SU) and there exists at least one idle PU node (that is turned OFF and its licensed channel is available for use) in the neighborhood of both the end-to-end SU nodes of the edge. We sample the network periodically for every t_{sample} seconds. We model a time-variant mobile SU graph $G_{t...t+k}$(SU) = G_t(SU) ∩ G_{t+1}(SU) ∩ .... ∩ G_{t+k}(SU) such that $G_{t...t+k}$(SU) is connected (i.e., spans all the SU nodes), wherein G_t(SU), G_{t+1}(SU), G_{t+2}(SU),...., G_{t+k}(SU) are respectively the static SU graphs at time instants sampled at time instants t, t+1, t+2, ..., and t+k.

3. Generic Algorithm to Determine Maximum Lifetime Communication Topology

Given the knowledge of the future availability of the PU channels, the MLCT algorithms works as follows: At a time instant t for which we want to find a stable communication topology that spans the entire network of SU nodes (say a shortest path tree rooted at source SU node s), we check if the static SU graph G_t(SU) is connected (i.e., the SU nodes are reachable from each other through one or more paths). We can check the connectivity of an SU graph (static graph or mobile graph) using the BFS algorithm. If all the vertices of the graph can be visited by running BFS from any arbitrarily chosen node, then the graph is connected; otherwise, not. If G_t(SU) is connected, the network-wide communication topology of interest exists at time instant t. We then proceed to the next sampling time instant t+1 and take the intersection of the graphs G_t(SU) and G_{t+1}(SU) and refer to the intersection graph as the mobile graph $\mathrm{G}_{t...t+1}$(SU) = G_t(SU) ∩ G_{t+1}(SU). If $\mathrm{G}_{t...t+1}$(SU) is connected, we continue to proceed to the next sampling time instant t+2 and check whether $\mathrm{G}_{t...t+2}$(SU) = G_t(SU) ∩ G_{t+1}(SU) ∩ G_{t+2}(SU) is connected. We continue like this until we come across sampling time instants t+k and t+k+1 such that

$G_{t...t+k}$(SU) is connected and $G_{t...t+k+1}$(SU) is not connected. If $G_{t...t+k}$(SU) is connected, it implies the communication topology of interest (say a shortest path tree rooted at source node *s*) exists in each of the static graph snapshots G_t(SU), G_{t+1}(SU), ..., G_{t+k}(SU). We could use that communication topology across all of these time instants *t*, *t*+1, ..., *t*+*k*. We repeat the above procedure starting from time instant *t*+*k*+1 to find the next stable mobile graph and communication topology of the sequence and continue likewise until the end of the network session.

```
Input: Static graphs $G_t$(SU) for all the sampling time instants $t \in 1....T$ of the network session
            Source node $s$
Output: Stable sequence of mobile graphs and maximum lifetime shortest path trees (MLSPTs)
Auxiliary variables: $i, j$
Initialization: $i = 1, j = 1$, $MLSPT_{1...T}$(SU) = $\phi$; $Stable\_G_{1...T}$(SU) = $\phi$
Begin MLCT Algorithm
    while ( $i \le T$ ) do
        Find a mobile graph $G_{i...j}$(SU) = $G_i$(SU) $\cap$ $G_{i+1}$(SU) $\cap$ ... $\cap$ $G_j$(SU) such that
                $G_{i...j}$(SU) is connected and $G_{i...j+1}$(SU) is not connected.
        $Stable\_G_{1...T}$(SU) = $Stable\_G_{1...T}$(SU)  U {$G_{i...j}$(SU)}
        Find a shortest path tree $SPT_{i...j}$(SU) = BFS($G_{i...j}$(SU), $s$)
        $MLSPT_{1...T}$(SU) = $MLSPT_{1...T}$(SU) U {$SPT_{i...j}$(SU)}
        $i = j+1$
    end while
    return $MLSPT_{1...T}$(SU) and $Stable\_G_{1...T}$(SU)
End MLCT Algorithm
```

Figure 1. Pseudo code of the maximum lifetime communication topology algorithm and its application to determine stable sequence of maximum lifetime shortest path trees (MLSPTs)

The MLCT algorithm finds for us a stable sequence of instances of the mobile graph and the communication topology of interest (say the shortest path tree) such that the number of transitions from one instance of the topology to another in the sequence is the global minimum. The average lifetime of the mobile graphs in the stable sequence found by the MLCT algorithm would serve as an upper bound (benchmark) for communication topologies (be it a shortest path tree, spanning tree, connected dominating set, etc) that span all the SU nodes found by any centralized or distributed algorithm for CRAHNs.

Figure 1 presents the pseudo code of the MLCT algorithm with respect to determining a stable sequence of mobile graphs and the corresponding instances of shortest path trees. As one can see, the MLCT algorithm is generic and could be used to find a stable sequence of any communication topology that spans the entire CRAHN network of SU nodes, provided there exists an underlying algorithm or heuristic to determine that communication topology (e.g., shortest path tree, minimum spanning tree, connected dominating set, etc) on a network graph. The tradeoff we anticipate is that the stable sequence determined for the communication topology of interest may not be optimal with respect to the metric for which it is known for. For example, the shortest path tree determined on a sequence of mobile graphs $G_{1...j}$(SU), $G_{j+1...k+1}$(SU), ... $G_{r+1...T}$(SU) may not have the same average tree height (a measure of the minimum number of hops for a path between any two SU nodes) compared to the average height of a sequence of shortest path trees determined on the individual static graphs G_1(SU), G_2(SU), ..., G_T(SU) where time instants 1 and *T* are respectively the beginning and ending time instants of the network session. This is because, when we aim for stability and determine a mobile graph $G_{t...t+k}$(SU) that exists for the longest time, the probability of finding an SU-SU edge in the mobile graph decreases and we mostly end up with the minimal number of SU-SU edges that would be needed to keep the SU nodes connected. A common shortest path tree determined on such a mobile graph $G_{t...t+k}$(SU) might have a height larger than the average height of the individual shortest path trees determined on each of the static graphs G_t(SU), G_{t+1}(SU), ..., G_{t+k}(SU).

3.1 Time Complexity of the MLCT Algorithm

The time complexity of the MLCT algorithm depends on the time complexity to determine the mobile graphs and the number of times we run the BFS algorithm to determine the connectivity of the static graphs and mobile graphs as well as on the time complexity of the algorithm or heuristic employed to determine the underlying communication topology of interest. The number of edges in any static SU graph cannot be more than the number of edges in the time-invariant SU graph G_i(SU). Let E be the number of edges in G_i(SU) and this is also the maximum number of edges in a static graph G_t(SU) at any time instant t as well as in a mobile graph $G_{t...t+k}$(SU) at any time instant $t+k$. The complexity of running BFS on an SU graph of V nodes is $\Theta(V+E)$, where V and E are respectively the number of SU nodes and SU-SU edges in the graph.

Given a mobile graph $G_{t...t+k}$(SU), we arrive at mobile graph $G_{t...t+k+1}$(SU) by simply checking whether each of the edges in $G_{t...t+k}$(SU) exists in the static graph G_{t+k+1}(SU). $G_{t...t+k+1}$(SU) contains only those edges of $G_{t...t+k}$(SU) that also exist in G_{t+k+1}(SU). Edges in G_{t+k+1}(SU) that do not exist in $G_{t...t+k}$(SU) are not considered for inclusion in $G_{t...t+k+1}$(SU). Thus, the time complexity to determine a mobile graph $G_{t...t+k+1}$(SU) is the time complexity to determine the mobile graph $G_{t...t+k}$(SU) plus the number of edges in $G_{t...t+k}$(SU), which could be at most E. In each iteration of the algorithm, we either move from mobile graph $G_{t...t+k}$(SU) to $G_{t...t+k+1}$(SU) if the latter is connected or start the algorithm fresh from G_{t+k+1}(SU) if $G_{t...t+k+1}$(SU) is not connected. Either way, the algorithm advances by one time instant ($t+k$ to $t+k+1$) in each iteration. Hence, the number of iterations of the algorithm is T, the duration of the network session (also the number of static graph samples generated). In each such iteration, we spend $\Theta(E)$ time to form a mobile graph. At the worst case, we may have to run the BFS algorithm twice in each iteration: once to check whether a mobile graph $G_{t...t+k}$(SU) is connected and (if it is not connected) to check whether the static graph G_{t+k}(SU) is connected. Hence, the time complexity to determine the mobile graphs and check for the connectivity of the mobile graphs/static graphs across all the iterations is $\Theta(\ (E + V + E)*T)$.

In the case of maximum lifetime shortest path trees (MLSPTs), the underlying algorithm used to determine the communication topology of interest is also BFS. We run the BFS algorithm for the communication topology once in each iteration: either on the mobile graph $G_{t...t+k}$(SU) if it is connected or on the static graph G_{t+k}(SU) if the former is not connected. Thus, the time complexity incurred to run the BFS algorithm (as the algorithm to determine the communication topology of interest) across all the T iterations is $\Theta(T*(V+E))$. Putting together all the time complexities, we arrive at the overall-time complexity of the MLCT algorithm to determine the MLSPTs to be: $\Theta((E+V+E)*T) + \Theta(T*(V+E)) = \Theta(T*(V+E))$. Note that this will also be the same time-complexity to determine a sequence of minimum-hop shortest path trees by running the $\Theta(V+E)$-BFS algorithm on each of the static graph snapshots. Thus, the MLCT algorithm could determine the stable MLSPTs without any substantial increase in the run-time complexity.

In general, if $\Theta(X)$ is the run-time complexity to determine the communication topology of interest on a particular SU graph, the overall-time complexity to run the MLCT algorithm to determine a stable sequence of the communication topology of interest would be: $\Theta(\mathrm{T}*(V+E+X)\ )$. For example, if one wants to determine a sequence of stable minimum spanning trees using the MLCT algorithm, the overall-time complexity would be: $\Theta(T*(V+E+ElogE))$, where $\Theta(ElogE)$ is the time-complexity to determine minimum spanning tree on a graph of E edges (in turn dependent on the number of vertices) according to the well-known Kruskal's algorithm (Cormen et al., 2009). Note that the weight of an edge in a mobile graph $G_{t...t+k}$(SU) is the sum of the weights of the edge in the constituent static graphs G_t(SU), G_{t+1}(SU), ..., G_{t+k}(SU).

To determine network-wide communication topologies with unit values for the edge weights (for example: minimum hop shortest path trees and minimum connected dominating sets), the sequence of stable mobile graphs, Stable_$G_{1...T}$(SU), obtained by running the MLCT algorithm once for a particular input sequence of the static SU graphs covering the entire duration of the network session would be sufficient to determine each of such network-wide communication topologies; one could simply run the appropriate algorithms to determine those communication topologies on the *same* Stable_$G_{1...T}$(SU).

3.2 Example to Illustrate the Execution of the MLCT Algorithm

Figure 2 illustrates an informative example for the execution of the MLCT algorithm to determine a stable sequence of shortest path trees across five time instants 1, 2, ..., 5. The SU nodes are represented as larger circles (white background) with the node ID inside the circles; the PU nodes are represented as smaller circles (color background: either blue or red) and node IDs are not assigned. In the static graphs at time instants 1, 2, ..., 5, an active PU node (i.e., is turned ON) is shown as a red-colored node and an idle PU node (i.e., is turned OFF) is shown as a blue-colored node. The dotted links represent the presence of a PU node within the transmission range of the SU node(s). The time-invariant initial graph of SU nodes represents a network of SU nodes with all possible links: there exists a link between any two SU nodes in this graph if the two SU nodes are neighbors.

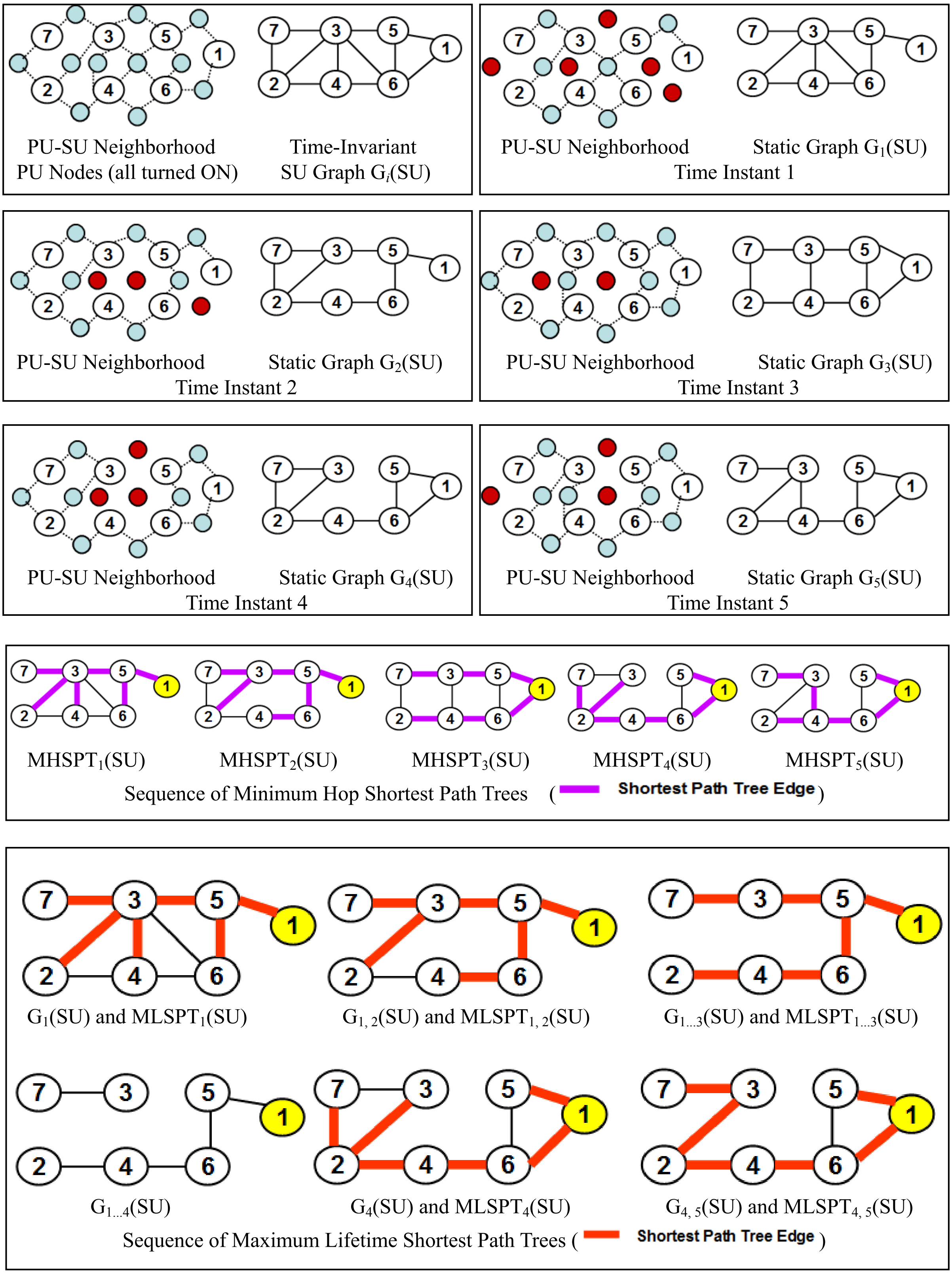


Figure 2. Example to Illustrate the Execution of the Maximum Lifetime Communication Topology

In the static SU graphs for time instants 1, 2, ..., 5, there exists an SU-SU edge only if the edge exists in the time-invariant SU graph as well as there exists an available/idle PU node within the transmission range of the two SU nodes at that time instant. Depending on the availability status of the PU nodes, one or more SU-SU edges in the time-invariant SU graph may not be present in the static SU graphs captured in the time instants 1, 2, ..., 5. We observe the static SU graphs captured at time instants 1, 2, ..., 5 to differ depending on the availability of the PU nodes.

Let node 1 be the source node rooted at which we are interested in finding the shortest path tree. The shortest path tree on an individual graph or mobile graph is determined by running the BFS algorithm, starting from node 1. Just for simplicity and uniformity in the example illustrated, we break the tie to explore the neighborhood of the vertices in the increasing order of their IDs as well as we visit the neighbors of a particular vertex in the increasing order of their IDs. However, in the simulations (Section 4), we randomly break the ties to visit the neighbors. In the example graphs of Figure 2, we observe that a shortest path tree for a particular SU graph (captured at a specific time instant) does not exist in the subsequent SU graphs and needs to be reconfigured for every time instant, leading to a total of four transitions (tree changes) when we aim for minimum-hop/minimum-height shortest path trees (MHSPTs) when we want to determine one at a particular time instant. The time-averaged height of the shortest path trees across these five time instants is (3+3+3+4+4)/5 = 3.4. We observe the MLCT algorithm to be able to find a mobile graph that spans the time instants 1, 2 and 3 (but does not exist further) and a mobile graph that spans the time instants 4 and 5; thus there is a need for only one tree transition when we use the MLCT algorithm to determine a stable sequence of shortest path trees across time instants 1, 2, ..., 5. The time-averaged height of the maximum lifetime shortest path trees (MLSPTs) is (4+4+4+5+5)/5 = 4.4. Though we observe the average tree height of the MLSPTs (4.4) to be appreciably larger than the average tree height of the MHSPTs (3.4) - indicating the possibility of a stability-hop count tradeoff, we do not observe any such appreciable tradeoffs in the simulation results (see Sections 4.1 and 4.2).

3.3 Proof of Correctness of the MLCT Algorithm

We now prove that the MLCT algorithm does discover a stable sequence of the mobile graphs and the corresponding instances of the communication topology of interest such that the number of transitions from one instance of the topology to another in the sequence is the global minimum (i.e., optimum). We use the approach of proof by contradiction. Let m be the number of transitions (change from one instance of the topology to another) in the sequence of instances of the communication topology determined by the MLCT algorithm. Let there be a hypothetical algorithm that determines a sequence of instances of the communication topology such that the number of transitions is m' wherein $m' < m$. If such a hypothetical algorithm exists, for m to be greater than m', (without loss of generality) there should be a mobile graph $G_{t+p...t+s}$(SU) for which the MLCT algorithm should have incurred at least one transition in the time span from $t+p$ to $t+s$, where t is the beginning time instant of the network session and $p < s$ are the increments to t indicating sampling time instants $t+p$ and $t+s$ respectively, but that the hypothetical algorithm did not incur any transition in the communication topology from $t+p$ to $t+s$. If the hypothetical algorithm were to find a network-wide communication topology that exists from time instants $t+p$ to $t+s$, the mobile graph $G_{t+p...t+s}$(SU) must be connected. But, the MLCT algorithm underwent at least one transition during the time period $t+p$ to $t+s$ because the mobile graph $G_{t+p...t+s}$(SU) was not connected. This implies the hypothetical algorithm cannot find an instance of the mobile graph and the communication topology of interest whose duration of existence is larger than that determined by the MLCT algorithm. Hence, the sequence of instances of the communication topology determined by the hypothetical algorithm should have underwent at least the same number of transitions as the sequence of instances of the communication topology determined by the MLCT algorithm, which means $m' \geq m$. This is a contradiction to the initial hypothesis that $m' < m$ and hence the hypothesis cannot be true. Thus, the MLCT algorithm determines a stable sequence of the mobile graphs and the corresponding instances of the communication topology of interest such that the number of topology transitions is the global minimum (optimum).

4. Simulations

We developed a discrete-event simulator in Java for Cognitive Radio Ad hoc Networks (CRAHNs) and implemented the Maximum Lifetime Communication Topology (MLCT) algorithm and adapted it to determine a sequence of maximum lifetime shortest path trees (MLSPTs) among the SU nodes. We also implemented the algorithm to determine a sequence of minimum hop shortest path trees (MHSPTs) that determines a MHSPT on a particular static SU graph and uses it in the static SU graphs for the subsequent time instants as long as the MHSPT exists. Such an approach is referred to as the Least Overhead Routing Approach, LORA (Abolhasan et al., 2004), commonly used by the distributed routing protocols for ad hoc networks.

The network topology is sampled for every 0.25 seconds to generate the static SU graphs. Simulations are run for a total of 1000 seconds; thus, the number of static SU graphs constructed is 4001, including the static graph of SU nodes at time 0. The source node for a particular MHSPT or MLSPT is chosen randomly from the set of

SU nodes. As MHSPTs follow the LORA strategy, a chosen source node for a particular MHSPT is continued to be used as the source node for the entire lifetime of the MHSPT. For the MLCT-MLSPT algorithm, the source node for a particular mobile graph $G_{t...t+k}$(SU) is randomly chosen among the SU nodes. According to the MLCT algorithm, when we find out that a mobile graph that existed from time instant t to $t+k$ no longer exists at time instant $t+k+1$, we run the BFS algorithm on the mobile graph $G_{t...t+k}$(SU), starting from a randomly chosen source SU node and the shortest path tree determined in $G_{t...t+k}$(SU) is considered to be the maximum lifetime shortest path tree (rooted at the particular source node) used for each of the time instants t to $t+k$.

The network is of dimensions 1000m x 1000m. The number of SU nodes is fixed at 50; the number of PU nodes is varied from 25, 40, 50, 75 to 100 nodes. Accordingly, we define the PU-SU ratio to be the ratio of the number of PU nodes to the number of SU nodes. For the above said values of PU nodes and SU nodes, the PU-SU ratio values range from 0.5 to 2.0. The PU nodes and SU nodes are identified with a unique ID; the PU channels are identified based on the ID of the PU node that holds their license. Each PU node is assumed to own a channel of distinct frequency. The transmission range of the SU nodes as well as the PU nodes is fixed to be 250m.

A PU node is considered to be in the neighborhood of an SU node if the Euclidean distance between them is within the transmission range (250m). We construct a time-invariant graph of the SU nodes - G_i(SU) that indicates the set of all possible edges between any two SU nodes in the network: an SU-SU edge exists in G_i(SU) if the Euclidean distance between the two SU nodes is within the transmission range (250m). At any time instant, an SU-SU edge exists only if the edge exists in G_i(SU) and at least one PU channel is available in their mutually intersecting neighborhood (i.e., a PU node in their mutually intersecting neighborhood is idle - turned OFF) at that time instant. The values of the parameters MAX_{Random_ON} and MAX_{Random_OFF} are each: 5, 10 and 20 seconds. A particular simulation is thus run for a fixed value of PU-SU ratio, MAX_{Random_ON} and MAX_{Random_OFF}, leading to a total of 5*3*3 = 45 scenarios for the operating conditions. The results (shown in Figures 3 and 4) are average values obtained from 20 runs of the simulations for each combination of values for PU-SU ratio, MAX_{Random_ON} and MAX_{Random_OFF}.

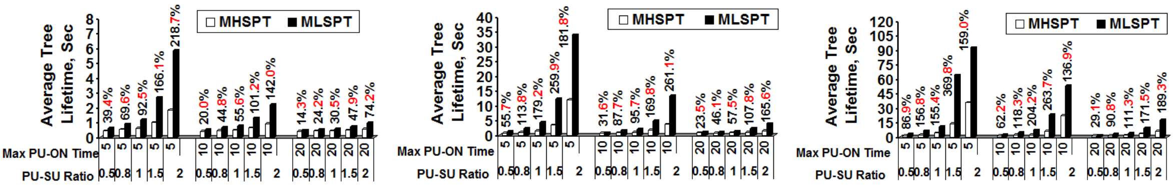
The performance metrics analyzed in the simulations are: (i) *Average Tree Lifetime*: This is the average of the lifetime of the shortest path trees determined according to the MLCT-MLSPT and MHSPT algorithms for particular values of PU-SU ratio, MAX_{Random_ON} and MAX_{Random_OFF}. The larger the value for the average tree lifetime, more stable is the tree determined by the corresponding algorithm. (ii) *Average Tree Height*: This is the time-averaged height of the shortest path trees determined for the sequence of MLSPTs and MHSPTs. The tree height is a measure of the diameter of the network (the maximum of the minimum number of hops for a path between any two nodes).

4.1 Average Tree Lifetime

We observe that the MLSPTs incur significantly larger values of tree lifetime compared to the MHSPTs for all the scenarios. The average tree lifetime incurred for the MLSPTs form benchmarks for the average lifetime that could be incurred for any centralized or distributed algorithm developed to determine shortest path trees spanning the SU nodes (or for that matter any network-wide communication topology spanning all the SU nodes) of a CRAHN. The percentage values shown in Figure 3 are based on the ratio of the lifetime of the MLSPTs to that of the MHSPTs, indicating the percentage difference in the lifetimes of the two trees. For a fixed value of the MAX_{Random_ON} and MAX_{Random_OFF} parameters, the average lifetime for both the MLSPTs and MHSPTs increases as the PU-SU ratio increases; nevertheless, the rate at which the average lifetime of the MLSPTs increases is significantly faster than the rate at which the average lifetime of the MHSPTs increases. In other words, the difference in the average lifetime incurred by the MLSPTs and MHSPTs increases as the PU-SU ratio increases. This is attributed to the larger number of PU channels (with increase in the number of PU nodes) accessible to the SU nodes. As a result, the SU-SU links are likely to exist for a longer lifetime. The MLCT algorithm leverages the a priori future knowledge about the PU channels and their availability and is able to find a long-living shortest path tree that exists in several of the static SU graphs spanning a mobile graph; on the other hand, the MHSPT algorithm could make use of only information about the availability of the PU channels in the particular SU graph at which it is run.

As the PU-SU ratio increases, the average lifetime of the MLSPTs increases and the increase is further enhanced with increase in the available time of the PU channels (i.e., with increase in the value of MAX_{Random_OFF}). We observe that for a MAX_{Random_OFF} value of 5 seconds, the average lifetime of the MLSPTs increased by a factor of 2 to 9 (the rate of increase gets smaller for larger values of MAX_{Random_ON}) as the PU-SU ratio is increased from 0.5 to 2.0; on the other hand, for a MAX_{Random_OFF} value of 20 seconds and a fixed MAX_{Random_ON}, the average lifetime of the MLSPTs increased by a factor of 9 to 30 (the rate of increase is smaller for larger values of

MAX_{Random_ON}) as the PU-SU ratio is increased from 0.5 to 2.0. As stated above, when the PU channels are used by their PU nodes for a longer time and remain unavailable for the SU nodes, the rate of increase for the average lifetime of the MLSPTs reduces. A similar trend is observed for the MHSPTs too, albeit with a relatively lower magnitude of increase of average tree lifetime when the increase in the PU-SU ratio is complemented with an increase in the available time of the PU channels. As the number of PU channels increases (i.e., increase in the PU-SU ratio) and the availability time of the PU channels increases, we observe an increase in the percentage difference between the lifetimes of the MLSPTs and MHSPTs.



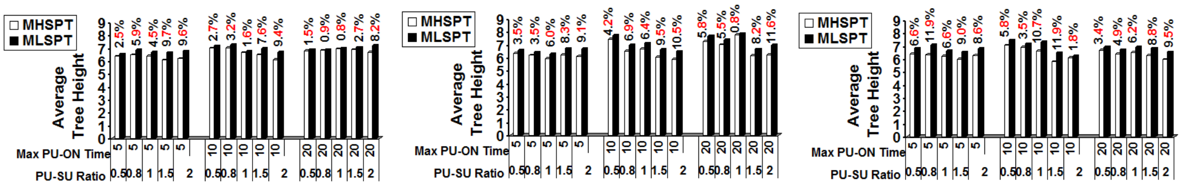
MAX_{Random_OFF} = 5 seconds MAX_{Random_OFF} = 10 seconds MAX_{Random_OFF} = 20 seconds

Figure 3. Lifetime of Maximum Lifetime Shortest Path Trees and Minimum Hop Shortest Path Trees

For a fixed MAX_{Random_OFF} value, as the MAX_{Random_ON} value increases, the average lifetime of the MLSPTs and MHSPTs decreases and the difference between the average lifetimes of the two trees narrows down for smaller values of PU-SU ratio. This is because when the number of available PU channels are lower and these channels are occupied for a longer time, the MLCT algorithm could not find a long-living shortest path tree that exists for several time instants. Nevertheless, the difference between the average lifetimes of the two trees is appreciable even under such scenarios (i.e., at smaller values of PU-SU ratio along with larger values of MAX_{Random_ON} and smaller values of MAX_{Random_OFF}): MLSPTs incur at least 14-30% larger lifetime than that of the MHSPTs.

4.2 Average Tree Height

We observe the MLSPTs to incur at most 12% larger height than MHSPTs. There is no definite trend in the difference in the height between the two trees across the simulation conditions. The overall average percentage difference in height between the two trees (when averaged over all the scenarios) is 6.2%. With the significant gains in the lifetime of the MLSPTs and an inconsequential increase in the height of the trees, we could infer that there is no stability-hop count tradeoff in CRAHNs, unlike the MANETs and mobile sensor networks wherein we have observed a stability-hop count tradeoff (Meghanathan, 2008) and stability-data gathering delay tradeoff (Meghanathan & Mumford, 2013) respectively. Overall, we observe a slight increase in the percentage difference in the height between the two trees with increase in the PU-SU ratio and/or increase in the availability time of the PU channels. This could be attributed to the existence of connected mobile graphs that span over several time instants, but with the bare minimal edges to keep the graph connected. The MLSPTs determined on such sparse mobile graphs are likely to sustain a relatively larger height (as also evidenced in the example of Figure 2). In other words, in scenarios conducive for finding long-living stable trees, the percentage difference in the height of the two trees increases. Nevertheless, the increase in the height of the MLSPTs is not significantly larger than that observed in scenarios that limit the existence of the connected mobile graphs to fewer instants (true for scenarios with larger values of MAX_{Random_ON} and low-moderate values of PU-SU ratio and smaller values of MAX_{Random_OFF}).



MAX_{Random_OFF} = 5 seconds MAX_{Random_OFF} = 10 seconds MAX_{Random_OFF} = 20 seconds

Figure 4. Height of Maximum Lifetime Shortest Path Trees and Minimum Hop Shortest Path Trees

5. Related Work and Our Contributions

Routing solutions for CRAHNs are either full spectrum knowledge based or local spectrum knowledge based.

The solutions based on the full spectrum knowledge assume each SU node to be completely aware of all the available PU channels in the network and choose optimal routes with respect to either minimum number of hops per SU-SU path (Xin et al., 2005), maximum conflict-free assignment (Zhou et al., 2009) of PU channels or minimum number of channel switches per SU-SU path (Xin et al., 2008); there is bound to be switching of channels when none of the PU channels available for the end nodes of an SU-SU link are the same as the preferred PU channels for one or both the end nodes to which they stay tuned by default for transmission and reception. Such full spectrum knowledge-based solutions take a centralized approach like we took in this paper; however, the full spectrum knowledge for the current time instant alone cannot be used to arrive at benchmarks for the routing metric, if one intends to use the LORA strategy of staying on with a route as long as it exists. In order to arrive at benchmark values for the routing metric and still use the LORA strategy, it is imperative that the routing algorithm knows the future topology changes or at least predict them and make use of. The approach taken in our paper to make use of the knowledge about future topology changes (i.e., availability of the PU channels) to arrive at optimal benchmark values for the maximum tree lifetime could be construed as a strategy for developing distributed algorithms to determine stable communication topologies.

The routing solutions based on the local spectrum knowledge are distributed solutions that make use of the spectrum information collected in the neighborhood through the common control channel (Lo, 2011). The routing solutions based on local spectrum knowledge could be classified according to the class of metric they are targeted to optimize in a distributed fashion. The minimum power routing protocol (Pyo & Hasegawa, 2007) discovers SU-SU paths that minimize the energy consumption incurred due to broadcast discovery, transmission, channel switching and reception. The bandwidth footprint (BFP) minimization-based routing protocol (Shi & Hou, 2008) discovers *s-d* (source-destination) routes that minimally impact the interference area of the SU nodes (called the bandwidth footprint) that are part of the ongoing *s-d* sessions. While analyzing the tradeoffs between farthest neighbor routing (FNR) and nearest neighbor routing (NNR) for CRAHNs, it was observed that FNR provides better end-to-end channel utilization and reliability, whereas NNR is relatively more energy-efficient (Xie et al., 2010). With regards to minimizing delay-sensitive routing protocols for CRAHNs, Cheng et. al (2007a) and Ma et. al (2008) developed protocols to minimize the channel switching and access delays and Cheng et. al (2007b) developed protocol to minimize the queuing delays at the intermediate nodes.

The proposed MLCT algorithm could be adapted to determine stable communication topologies that could also incur lower values for power consumption, interference or channel switch delay. For a mobile graph, the weight of an SU-SU edge is simply the sum of the weights of the edge in the constituent static graphs. The weights of the SU-SU edges in the static graphs have to be just suitably modeled to suit to the communication topology of interest. For example, to determine stable minimum power consumption routes, one could model the weight of an SU-SU edge at a particular time instant as the sum of the energy lost due to reception, transmission (function of the distance between the two end SU nodes at that time instant) and channel switching (depending on the preferred PU channels of the two end SU nodes at that time instant), and simply run the Dijkstra algorithm (Cormen et al., 2009) on the long-living connected mobile graphs to determine a sequence of stable minimum power consumption *s-d* paths such that the energy lost due to broadcast route discovery (route transitions) is the global minimum. Likewise, to determine stable minimum interference routes, one could model the weight of an SU-SU edge for a static graph to be the number of peer SU nodes that are also in the neighborhood of the PU channel used by the edge for transmission and reception, and simply run the Dijkstra algorithm on the long-living connected mobile graphs to determine a sequence of stable minimum interference *s-d* paths such that the number of route transitions is the global minimum.

To the best of our knowledge, we have not come across any work that has proposed a benchmarking algorithm to determine stable communication topologies for CRAHNs. In earlier works, separate benchmarking algorithms have been proposed based on the idea of taking graph intersections to determine stable sequence of unicast paths, multicast Steiner trees and broadcast connected dominating sets (Meghanathan, 2008) for mobile ad hoc networks (MANETs) and to determine stable sequence of data gathering trees (Meghanathan & Mumford, 2013) for wireless mobile sensor networks (WMSNs). The characteristic of both MANETs and WMSNs is that the nodes are mobile and it is the mobility of the nodes that triggers the topology changes. On the other hand, nodes in the CRAHNs considered in this research are static and it is the availability of the PU channels that changes dynamically with time, triggering changes in the communication topology of interest. Moreover, the benchmarking algorithms proposed for MANETs and WMSNs focus on a specific communication topology of interest (like an *s-d* path, multicast tree, data gathering tree, etc) and are not proposed as generic algorithms. In this paper, we propose the MLCT algorithm as a generic algorithm that can be used to determine a stable sequence of instances of long-living mobile graphs such that the average lifetime of these mobile graphs will be the upper bound (benchmark) for the average lifetime incurred for any communication topology of interest that spans the entire network of SU nodes (be it a shortest path tree, minimum spanning tree, minimum connected dominating set, etc); the shortest path tree has been just chosen as a representative communication topology of interest to demonstrate the working of the MLCT algorithm and its analysis.

6. Conclusions

The high-level contribution of this paper is the proposal of a centralized benchmarking algorithm to determine stable sequence of mobile graphs and the corresponding instances of a communication topology that spans all the secondary user (SU) nodes in a cognitive radio ad hoc network (CRAHN) environment. We make use of the notion of mobile graph for the SU nodes that exists over a sequence of time instants and captures only the SU-SU edges that exist in all these time instants. Referred to as the Maximum Lifetime Communication Topology (MLCT) algorithm, the algorithm is generic in nature and can be applied to determine a stable sequence of any communication topology of interest (like shortest path trees, minimum spanning trees, minimum connected dominating set, etc) as long as there is an algorithm or heuristic to determine the topology. The MLCT algorithm is assumed to know beforehand about the sequence of topology changes in the future. Whenever the communication topology of interest is needed at a particular time instant, we determine a mobile graph that exists for the longest time since that time instant and run an algorithm (appropriate to determine the communication topology of interest) on the mobile graph to determine the communication topology that is guaranteed to exist in all the constituent static graphs. The run-time complexity of the proposed algorithm is a function of the time complexity of the algorithm used to determine the underlying communication topology of interest as well as the time complexity of the Breadth First Search (BFS) algorithm to check for connectivity of the mobile graphs.

In addition to describing a generic version of the MLCT algorithm, we have specifically discussed its application to determine a sequence of maximum lifetime shortest path trees (MLSPTs) and compared the performance of the MLSPTs with that of minimum hop shortest path trees (MHSPTs) through extensive simulations. Simulation results indicate the average lifetime of the MLSPTs to truly serve as benchmarks for average tree lifetime of shortest path trees among the SU nodes in a CRAHN as the MLSPTs incur substantially longer lifetime compared to the MHSPTs without incurring any significant increase in the tree height. The lifetime of the MLSPTs could be anywhere from 14% to 370% larger than that of the lifetime of the MHSPTs; whereas, the increase in the height for the MLSPTs could be at most 12%. Thus, there is no stability-hop count tradeoff when we aim for stable shortest path trees spanning the SU nodes in a CRAHN.

Note that the MLCT algorithm uses a greedy strategy of determining a mobile graph that spans for the largest number of time instants starting from the current time instant and running an appropriate algorithm for the communication topology of interest on the mobile graph. We have shown that such a greedy strategy does indeed gives an optimal solution for the minimum number of transitions (and hence a maximum lifetime) among the instances of the communication topology determined for the duration of the network session. Greedy strategies rarely give optimal solutions with exceptions like the Dijkstra algorithm for minimum-weight path trees, Prim's and Kruskal's algorithms for minimum spanning trees, Huffman algorithm for prefix-free variable-bit encoding, etc (Cormen et al., 2009). Thus, the MLCT algorithm joins the league of these greedy strategy based algorithms that give an optimal solution for the problem they are meant for.

Acknowledgments

This research is funded by the NASA EPSCoR sub award (#: NNX14AN38A) from University of Mississippi.

References

Abolhasan, M., Wysocki, T., & Dutkiewicz, E. (2004). A Review of Routing Protocols for Mobile Ad hoc Networks. *Ad hoc Networks*, *2*(1), 1-22. http://dx.doi.org/10.1016/S1570-8705(03)00043-X

Akyildiz, I. F., Lee, W. Y., Vuran, M. C., & Mohanty, S. (2006). Next Generation/ Dynamic Spectrum Access/Cognitive Radio Wireless Networks: A Survey. *Computer Networks*, *50*(13), 2127-2519. http://dx.doi.org/10.1016/j.comnet.2006.05.001

Akyildiz, I. F., Lee, W. Y., & Chowdhury, K. R. (2009). CRAHNs: Cognitive Radio Ad hoc Networks. *Ad hoc Networks*, *7*(5), 810-836. http://dx.doi.org/10.1016/j.adhoc.2009.01.001

Cheng, G., Liu, W., Li, Y., & Cheng, W. (2007a). Spectrum Aware On-demand Routing in Cognitive Radio Networks. Paper presented at the 2nd IEEE International Symposium on New Frontiers in Dynamic Spectrum Access Networks, Dublin, Ireland. http://dx.doi.org/10.1109/DYSPAN.2007.80

Cheng, G., Liu, W., Li, Y., & Cheng, W. (2007b). Joint On-demand Routing and Spectrum Assignment in Cognitive Radio Networks. Paper presented at the IEEE International Conference on Communications, Glasgow, Scotland. http://dx.doi.org/10.1109/ICC.2007.1075

Cormen, T. H., Leiserson, C. E., Rivest, R. L., & Stein, C. (2009). *Introduction to Algorithms* (3rd ed.) MIT Press.

Lo, B. F. (2011). A Survey of Common Control Channel Design in Cognitive Radio Networks. *Physical Communication*, *4*(1), 26-39. http://dx.doi.org/10.1016/j.phycom.2010.12.004

Ma, H., Zheng, L., Ma, X., & Iuo, Y. (2008). Spectrum Aware Routing for Multi-hop Cognitive Radio Networks with a Single Transceiver. Paper presented at the 3rd International Conference on Cognitive Radio Oriented Wireless Networks and Communications, Singapore. http://dx.doi.org/10.1109/CROWNCOM.2008.4562509

Meghanathan, N. (2008). On the Stability of Paths, Steiner Trees and Connected Dominating Sets in Mobile Ad hoc Networks. *Ad hoc Networks*, *6*(5), 744-769. http://dx.doi.org/10.1016/j.adhoc.2007.06.005

Meghanathan, N. (2009). A Location Prediction Based Reactive Routing Protocol to Minimize the Number of Route Discoveries and Hop Count per Path in Mobile Ad hoc Networks. *The Computer Journal*, *52*(4), 461-482. http://dx.doi.org/10.1093/comjnl/bxn051

Meghanathan, N., & Mumford, P., (2013). A Benchmarking Algorithm to Determine Maximum Stability Data Gathering Trees for Wireless Mobile Sensor Networks. Paper presented at the 10th International Conference on Information Technology: New Generations, Las Vegas, NV, USA. http://dx.doi.org/10.1109/ITNG.2013.83

Perkins, C. E., & Royer, E. M. (1999). Ad hoc On-demand Distance Vector Routing. Paper presented at the 2nd Annual IEEE Workshop on Mobile Computing Systems and Applications, New Orleans, LA, USA. http://dx.doi.org/10.1109/MCSA.1999.749281

Pyo, C. W., & Hasegawa, M. (2007). Minimum Weight Routing based on a Common Link Control Radio for Cognitive Wireless Ad hoc Networks. Paper presented at the International Conference on Wireless Communications and Mobile Computing, Honolulu, HI, USA. http://dx.doi.org/10.1145/1280940.1281026

Shi, Y., & Hou, Y. (2008). A Distributed Optimization Algorithm for Multi-hop Cognitive Radio Networks. Paper presented at the 27th IEEE Conference on Computer Communications, Phoenix, AZ, USA. http://dx.doi.org/10.1109/INFOCOM.2008.186

Xie, M., Zhang, W., & Wong, K-K. (2010). A Geometric Approach to Improve Spectrum Efficiency for Cognitive Relay Networks. *IEEE Transactions on Wireless Communications*, *9*(1), 268-281. http://dx.doi.org/10.1109/TWC.2010.01.090180

Xin, C., Xie, B., & Shen, C. C. (2005). A Novel Layered Graph Model for Topology Formation and Routing in Dynamic Spectrum Access Networks. Paper presented at the 1st IEEE International Symposium on New Frontiers in Dynamic Spectrum Access Networks, Baltimore, USA. http://dx.doi.org/10.1109/DYSPAN.2005.1542647

Xin, C., Ma, L., & Shen, C. C. (2008). A Path-centric Channel Assignment Framework for Cognitive Radio Wireless Networks. *Mobile Networks and Applications*, *13*(5), 463-476. http://dx.doi.org/10.1007/s11036-008-0084-y

Zhao, Q., & Sadler, B. (2007). A Survey of Dynamic Spectrum Access. *IEEE Signal Processing Magazine*, *24*(3), 79-89. http://dx.doi.org/10.1109/MSP.2007.361604

Zhou, X., Lin, L., Wang, J., & Zhang, X. (2009). Cross-layer Routing Design in Cognitive Radio Networks by Colored Multi graph Model. *Wireless Personal Communications*, *49*(1), 123-131. http://dx.doi.org/10.1007/s11277-008-9561-7

Copyrights

Copyright for this article is retained by the author(s), with first publication rights granted to the journal.

This is an open-access article distributed under the terms and conditions of the Creative Commons Attribution license (http://creativecommons.org/licenses/by/3.0/).