Mobile Netw Appl (2008) 13:463–476
DOI 10.1007/s11036-008-0084-y

A Path-Centric Channel Assignment Framework for Cognitive Radio Wireless Networks

Chunsheng Xin · Liangping Ma · Chien-Chung Shen

Published online: 8 July 2008

Abstract Today's static spectrum allocation policy results in a situation where the available spectrum is being exhausted while many licensed spectrum bands are under-utilized. To resolve the spectrum exhaustion problem, the cognitive radio wireless network, termed *CogNet* in this paper, has recently been proposed to enable unlicensed users to dynamically access the licensed spectrum bands that are unused in either temporal or spatial domain, through spectrum-agile cognitive radios. The CogNet plays the role of secondary user in this shared spectrum access framework, and the spectrum bands accessible by CogNets are inherently heterogeneous and dynamic. To establish the communication infrastructure for a CogNet, the cognitive radio of each CogNet node detects the accessible spectrum bands and chooses one as its operating frequency, a process termed *channel assignment*. In this paper we propose a graph-based *path-centric* channel assignment framework to model multi-hop ad hoc CogNets and perform channel assignment from a network perspective. Simulation results show that the path-centric channel assignment framework outperforms traditional *link-centric* approach.

Keywords cognitive radio wireless network · dynamic spectrum access · path-centric channel assignment

Chunsheng Xin was supported in part by National Science Foundation under grant CNS-0721313.

Liangping Ma was supported in part by National Science Foundation under grant CNS-0721230.

Chien-Chung Shen was supported in part by National Science Foundation under grants CNS-0721361, CNS-0347460 and CNS-0335302.

C. Xin (✉)
Dept. of Computer Science, Norfolk State University, Norfolk, VA 23504, USA
e-mail: cxin@nsu.edu

L. Ma
Argon ST, Inc., San Diego, CA 92121, USA
e-mail: lma@argonst.com

C.-C. Shen
Dept. of Computer and Information Sciences, University of Delaware, Newark, DE 19716, USA
e-mail: cshen@cis.udel.edu

1 Introduction

Access to wireless spectrum has been chronically regulated with *static* spectrum allocation policy since the early 20th century, where each wireless service is assigned a fixed block of spectrum for exclusive usage. With recent rapid proliferation of wireless services, the remaining spectrum available for these new wireless services is being exhausted, which is known to be the *spectrum scarcity* problem in the literature [1]. Nevertheless, according to studies sponsored by the Federal Communications Commission (FCC) [2, p.9–16], many allocated spectrum blocks are used only in certain geographical areas and/or in short periods of time. In light of this fact, one solution to the spectrum scarcity problem is to allow other users to share the under-utilized spectrum, provided that the licensed users of such spectrum are not interfered. Since the accessibility of a block of spectrum by unlicensed users varies over time, such spectrum sharing is called *dynamic spectrum access*.

On the policy side, FCC is currently sponsoring studies on spectrum allocation and regulatory policies reform to pave the way for dynamic spectrum access. On the technology side, the spectrum-agile cognitive radio can tune to a wide spectrum range, identify the spectrum bands *currently* not being used by licensed users, and operate on one of these available spectrum bands. The combination of advance in cognitive radio technology and reform of spectrum allocation policy promises to enable cognitive radio wireless networks (termed *CogNets* in this paper) with dynamic spectrum access capability. In particular, FCC released the Notice of Proposed Rulemaking [3] in 2004 to allow unlicensed users to access idle TV channels. Since then, there have been many studies on utilizing idle TV channels by unlicensed users (e.g., see [4–7] and references therein). These studies showed that although the availability of idle TV channels varies over time, there still exist many idle TV channels at any moment and any location that are available to unlicensed users for effective data communications.

In CogNets, to establish the communication infrastructure, each node uses its cognitive radio to detect the *accessible* spectrum bands that are available for dynamic spectrum access, and chooses one as its operating frequency. The process of assigning a frequency band, which is often referred to as a *channel* in the literature, to each radio interface as its operating frequency is called *channel assignment*. Channel assignment is a major technique to achieve high spectrum utilization and traffic throughput. As such there have been extensive research on channel assignment in multi-channel ad hoc networks based on IEEE 802.11 [8–20]. However, solutions for traditional multi-channel wireless networks are not suitable for CogNets, due to the challenges raised by dynamic spectrum access. For example, the algorithms in [8–12] assumed N radios at each node, one for each channel. This would result in prohibitively high cost due to the expected high cost of cognitive radios. The algorithms in [14–18] used two radios, and assigned a common control channel among all nodes to one radio. Each node then uses this control channel to dynamically negotiate a data channel for another radio when transporting packets to a neighbor. Nevertheless, there may not exist a common channel for a CogNet due to its role as a secondary user in dynamic spectrum access. Furthermore, these existing algorithms implicitly require that all channels are static and accessible at every node, which, although offers a great convenience for coordination in channel negotiation and switching, is not realistic for CogNets.

More recently, the work of [21] studied channel assignment for multi-hop ad hoc CogNets, and assumed one single radio per node. With this algorithm, CogNet nodes form groups in a distributed operating mode, with each group having one common channel. Groups are connected via boundary nodes between groups by using a *time-division multiple access* MAC protocol. A boundary node switches its radio among the channels of its connected groups at different time slots. The data transport between nodes in different groups goes through the boundary nodes between groups. In other words, such a CogNet is time-slotted and synchronized to achieve connectivity between groups.

All the above approaches to channel assignment are essentially link (MAC) layer solutions and consider channel assignment/negotiation only between neighboring nodes. We term these approaches as the *link-centric* channel assignment. In this paper, we study channel assignment from a network (layer) perspective, and propose a *path-centric* channel assignment framework for CogNets. In this path-centric framework, channel assignment is driven by the routing protocol and directly affects network performance such as throughput, rather than indirectly affects network performance from the MAC layer as by the link-centric channel assignment. This framework is particularly designed for multi-hop ad hoc CogNets, although it also works well in traditional multi-channel wireless networks using IEEE 802.11. The framework includes both a modeling tool named *layered graph* to model CogNets and the path-centric channel assignment algorithm utilizing the layered graph.

In the remainder of the paper, we first discuss in Section 2 the general assumptions for our work. Section 3 introduces the layered graph modeling tool for CogNet. In Section 4, we discuss the path-centric channel assignment using the layered graph. Section 5 evaluates the performance of the path-centric channel assignment algorithm, and Section 6 concludes the paper.

2 Assumptions and motivations

In this paper, we use a generic term *channel* for a block of spectrum that was allocated to some licensed user but is accessible for unlicensed users for dynamic spectrum access. It can be the spectrum band of one licensed user, e.g. a TV channel, or a sub-band divided from a larger spectrum band of a licensed user, e.g., an IEEE 802.11 channel. The channels can be either orthogonal as in IEEE 802.11 networks, or non-orthogonal. The channels dynamically arrive/depart, i.e., dynamically change their status between accessible and inaccessible for unlicensed users (i.e., CogNet nodes). The channel lifetime—the duration accessible to unlicensed users—is assumed reasonably long, e.g., an idle TV channel usually lasts for hours. Without loss of generality, we assume that the cognitive radio is a single-interface radio and the interface is half duplex. The cognitive radio can dynamically switch among different channels, but operates on only one channel at a specific time. Our channel assignment framework allows different nodes to have different number of cognitive radios, although in practice, most nodes would likely have only one radio due to the cost concern. One major motivation of equipping more than one radio to a CogNet node is to let the node act as a gateway to connect networks operating in spectrum bands significantly apart, e.g., one network operating in channels around 900 MHz, and the other operating in channels around 2.4 GHz. Since it takes a non-negligible amount of time for a typical radio to switch between such diversely separated bands, it is not practical to use one radio to connect such networks through dynamic spectrum switching.

We assume that each CogNet node can detect both the accessible channels and the neighbors reachable on these channels. Furthermore, this channel information together with the number of radios of each node can be disseminated in the CogNet to every other node or to a *designated node*, which then utilizes the information to perform the path-centric channel assignment and routing path computation for the CogNet. If there is a control channel in the CogNet, the process of detecting and disseminating accessible channels can be performed over the control channel. In the case that there exists no control channel in the CogNet, we assume that the CogNet nodes have already been self-organized to form a preliminary communication infrastructure. Then the channel and radio information, possibly with other information such as traffic load statistics, can be piggybacked as part of the routing protocol control messages, such as the link-state advertisement or distance vector exchange. For instance, a link-centric channel assignment algorithm such as the one described in [21] could be utilized for CogNet self-organization since such link-centric channel assignment algorithm operates in a *local* mode, i.e., a node only talks to its neighbors to determine its channel assignment (see Section 5 for details). However, such self-organized CogNet usually forms sub-optimal channel assignment and routing paths. The objective of our proposed path-centric channel assignment framework is to configure the network into a near-optimal communication topology to achieve better performance.

3 Layered graph model for CogNet[1]

The routing path computation and channel assignment in the CogNet is challenged by both the channel heterogeneity and dynamic channel arrival/departure. Without a comprehensive modeling tool, a computed routing path may conflict with the channel assignment and becomes not committable. We propose a layered graph to model the CogNet. The graph accurately models the channel information at each node and streamlines the interplay between routing path computation and channel assignment for the CogNet, which results in an easy procedure of searching shortest paths in the graph. Since the routing path computation and channel assignment are carried out in an integrated manner, both activities are guaranteed to be consistent with each other. As described earlier, the discovered channels and number of radios of every node are disseminated to either every node or a designated node, and the channel and radio information may be piggybacked in the routing protocol control messages with a small overhead. Based on the channel and radio information, a CogNet node constructs the layered graph as follows. Let N denote the number of channels and $\mathcal{G}$ denote the layered graph. The graph $\mathcal{G}$ has N logical layers, with one layer corresponding to one channel. For each node in the CogNet, graph $\mathcal{G}$ adds this node and N *subnodes* associated with this node as its vertices, with one subnode in each layer. For example, for node A in the CogNet, we add node A and subnodes $A_1, \ldots, A_N$ to $\mathcal{G}$, with subnode A_i in layer i of $\mathcal{G}$. The edges in the corresponding layered graph are constructed as three different types, *access*, *horizontal*, and *vertical edges*, as follows:

- *Access edges* connect a node to its subnodes, e.g., connect node A to subnode A_1,
- *Horizontal edges* connect subnodes in the same logical layer, representing the reachability between nodes at the channel corresponding to this logical layer, and
- *Vertical edges* connect subnodes that are associated with the same node and are in different layers, indicating the data forwarding capability between different channels at a node.

Algorithm 1 constructs the layered graph, given a CogNet with the channel reachability and radio information of each node. Figure 1 presents a constructed layered graph for a 3-node CogNet shown in Fig. 2,

[1] A preliminary version of the graph model has been presented in IEEE DySPAN 2005 [22] and CrownCom 2007 [23].

Algorithm 1 Layered graph construction

Input: A CogNet with the channel reachability and radio information between nodes.
/* Add vertices and access edges */
for each node A in the CogNet **do**
  Add A and $A_1, \ldots, A_N$ to be vertices of graph $\mathcal{G}$.
  for $1 \le i \le N$ **do**
    Add the access edge (A, A_i) to graph $\mathcal{G}$. (We use (A_i, B_i) to indicate a bidirectional edge between A_i and B_i, which is equivalent to two unidirectional edges, $\langle A_i, B_i \rangle$ and $\langle B_i, A_i \rangle$.)
  end for
end for
/* Add horizontal edges */
for $1 \le i \le N$ **do**
  for every node pair A and B in the CogNet **do**
    If node A and B can reach each other in channel i, then add the horizontal edge (A_i, B_i) to graph $\mathcal{G}$.
  end for
end for
/* Add vertical edges */
for each node A in the CogNet, and $1 \le i \ne j \le N$ **do**
  If both A_i and A_j have horizontal edges to other nodes, add vertical edge (A_i, A_j) to $\mathcal{G}$.
end for

where channel 1 is available between nodes A and B, and between nodes B and C, channel 2 is available between nodes A and B, and channel 3 is available between nodes B and C. All edges shown in Fig. 1 are bidirectional edges.

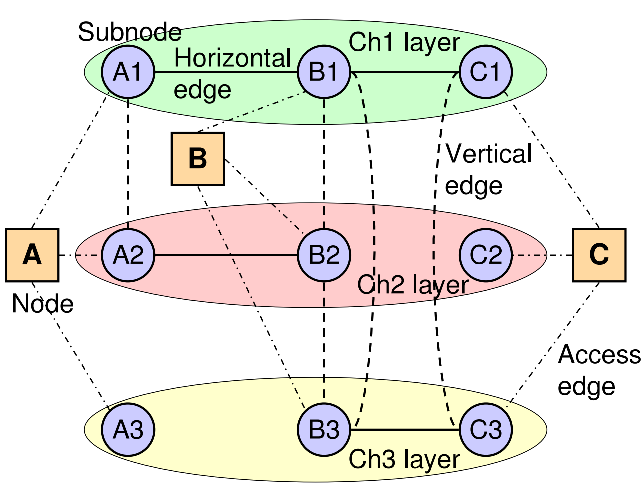


Figure 1 A sample layered graph

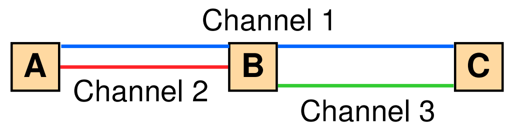


Figure 2 A 3-node CogNet

With the layered graph, the process of finding a routing path between two nodes simply becomes searching a shortest path in the layered graph. In a layered graph, the vertical edges offer the flexibility to choose different channels on the incoming and outgoing links of a node. The radio of this node can receive packets from one channel and switch to another channel to forward packets to the next hop, which reduces the interference between neighboring nodes on a path and improves spectrum utilization. The ‘cost’ of a vertical edge can be set related to the channel switching cost. Alternatively, we can set the cost of a vertical edge related to the signal quality of the channel indicated by the destination subnode[2] of the vertical edge, the interference level between two channels indicated by the source and destination subnodes of the vertical edge, the preference to channels of certain licensed users that are more tolerable to secondary users over the channels of other licensed users, or a combination of all these metrics. With the vertical edge, the interference between adjacent non-orthogonal channels can also be well addressed by setting a high cost to the vertical edges going to subnodes corresponding to adjacent channels.

We can further extend the layered graph model to offer more flexibility to control the outgoing channel selection for the packets received on a specific channel. For example, a node may want to avoid using the same channel in both the upstream and downstream links when forwarding packets. Another example is that one may want to consider node cost in the routing path computation, in addition to link cost, so that some CogNet nodes are preferred over other nodes in traffic forwarding, e.g., because the former has more power to forward traffic, or there is a security concern with the latter. In addition, the cost setting of vertical edges in the layered graph needs to be carefully done to avoid the situation that the total cost of multiple consecutive vertical edges is less than the cost of another vertical edge with the same end nodes, e.g., the total cost of edge (B_1, B_2) and (B_2, B_3) is less than the cost of edge (B_1, B_3) in Fig. 1. In order to provide the additional flexibility and controllability, we introduce an auxiliary

[2] The bidirectional vertical edge includes two unidirectional vertical edges, one in each direction.

subnode A'_i for each subnode A_i in graph $\mathcal{G}$. We connect the *primary* subnode A_i to its auxiliary subnode A'_i with a directional edge, which is also classified as a vertical edge. Furthermore, we explicitly use two unidirectional edges, one in each direction, to replace each bidirectional access, horizontal and vertical edge. The incoming vertical and access edges that are previously terminated at the primary subnode A_i, and outgoing horizontal edges that are previously originated from the primary subnode A_i are now terminated/originated at/from the auxiliary subnode A'_i. The outgoing vertical and access edges and the incoming horizontal edges are still originated/terminated from/at the primary subnode A_i. Figure 3 illustrates the added auxiliary subnodes for all of the subnodes associated with node A, and the edges connected to the primary and auxiliary subnodes, respectively. With the added auxiliary subnodes, the channel selection priority can be realized through assigning a cost to each vertical edge. For example, in Fig. 3, the shortest path search or the routing path computation would favor channel 3 as the outgoing channel, if the incoming channel is channel 1 on this path, since the cost of vertical edge $\langle A_1, A'_3 \rangle$ is 1, which is smaller than the cost of vertical edges $\langle A_1, A'_2 \rangle$ and $\langle A_1, A'_1 \rangle$. Similarly channel 2 has higher priority over channel 1 since the cost of $\langle A_1, A'_2 \rangle$ is smaller than $\langle A_1, A'_1 \rangle$. Furthermore, imposing the node cost in the routing path computation can be easily achieved by setting the cost of vertical edges between the primary and the auxiliary subnodes associated with a node related to the cost of this node. Figure 4 illustrates the extended layered graph for the CogNet in Fig. 2, with the added auxiliary subnodes. (The access edges to node B and C are not shown for a clearer view.)

The full connection between subnodes creates $O(N^2)$ vertical edges for each node in the CogNet, which may slow down the process of shortest path search. In many scenarios, a CogNet node only needs to choose an outgoing channel that is different from the incoming channel during the outgoing channel selection process for the computation of a routing path. For instance, if the incoming channel is 1, the node simply wants to avoid selecting channel 1 as the outgoing channel for this routing path, but does not mind to select either channel 2 or 3. In this case, we can add one *cross-connecting subnode* (XSN) between the primary subnodes and auxiliary subnodes associated with a node, as illustrated in Fig. 5. Each primary subnode is connected to its auxiliary subnode with one vertical edge, and to the XSN with another vertical edge. We can set different costs on these two vertical edges to give preference to one over the other. With the XSN, the number of vertical edges are $3N$ for the subnodes associated with one node. Let the number of nodes in the CogNet be M. The total number of edges in the layered graph is then $O(NM)$ and the total number of vertices is $O(NM)$. Using a heap-based implementation, the running time of a shortest path algorithm is $O(NM \cdot \log(NM))$, which scales well.

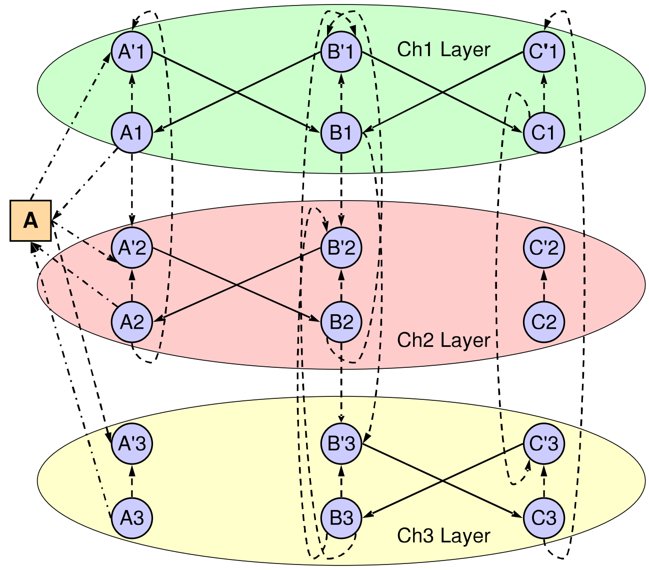


Figure 4 The extended layered graph with auxiliary subnodes

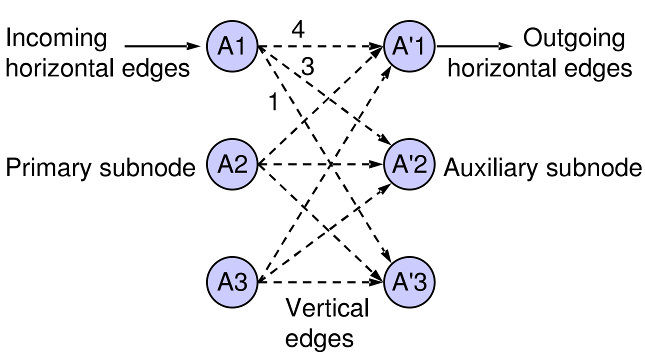
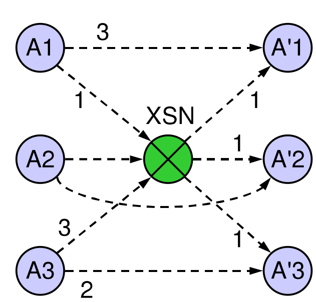


Figure 3 Auxiliary subnodes and vertical edges in the layered graph

Figure 5 Reduce the number of vertical edges in the layered graph



4 Path-centric channel assignment[3]

The objective of switching channels in a multi-channel wireless network is to reduce packet collision among neighboring nodes using the same channel and thus increase throughput. Such objective is the major motivation of the work in [8–20]. Nevertheless using multiple channels is not always beneficial. In some scenarios, using a common channel for all nodes may be better. For example, when the traffic volume traversing a node is light, the collision should be low and thus switching the radio of this node among different channels may not be beneficial, because the channel switching overhead may result in lower throughput than using a common channel. On the other hand, when the traffic volume is high, switching the radio between channels is clearly beneficial. Therefore, whether using a common or different channels for neighboring nodes should be dependent on the collision level, which in turn depends on the traffic loads traversing these nodes. The traffic load going through a specific node or link is determined by the routing protocol. However, link-centric channel assignment algorithms cannot capture this feature well. This motivates us to couple routing decision with channel assignment, and let channel assignment be driven by the routing protocol, so that some nodes use a common channel and others switch their radios among different channels, instead of simply letting all nodes switch radios among different channels based on a local decision as in the previous work. Next we introduce our path-centric channel assignment algorithm. For the ease of description, we first present the algorithm in a centralized mode, i.e., running this algorithm on a designated node that has collected information from every node. At the end of the section, we discuss how this algorithm operates in a distributed mode.

We utilize both traffic information and routing path computation to determine the channel assignment for each node. As discussed in Section 2, the traffic load information can be collected by every node and disseminated to the network together with the channel and neighbor information, e.g., through piggybacking in the routing protocol control messages. After the layered graph is constructed, we first compute the routing path (the minimum cost path) on the layered graph between a node pair with the largest traffic volume,[4] by setting some initial cost to horizontal and vertical edges. The cost of horizontal edges are set relevant to the traffic load on this link. The cost of vertical edges can be set to indicate the channel switching overhead or to achieve other objectives, as discussed earlier. Note that when computing a routing path for a node pair, the access edges, except the ones connected to the source and destination node, do not participate in the computation. After a routing path is computed, for each node corresponding to an intermediate subnode in the path, if it has an available radio that has not been assigned to a channel,[5] we assign it to the channel corresponding to the subnode[6] on the path. We call this channel as the *primary channel* of this radio, because, as to be discussed later, this radio may dynamically switch to other channels when transporting traffic between another node pair. We label the subnode(s)[7] of a node that corresponds to the primary channel(s) of this node's radio(s) as *active subnode(s)* and all other subnodes of this node as *inactive subnodes*. (Initially all nodes are *unlabeled*.) In other words, the subnodes in the path determine the channel assignment of their corresponding nodes if there are available radios in these nodes. Next we change the edges incident to inactive subnodes as follows: the incoming horizontal edges and outgoing vertical edges are removed.[8] In the routing path computation, we should not allow the traffic to go to an inactive subnode through a horizontal edge, because the corresponding node usually does not have a radio tuned to the channel indicated by this inactive subnode to receive traffic. Therefore we remove the incoming horizontal edges incident to an inactive subnode so that there is no routing path going to an inactive subnode through a horizontal edge. This will become more clear later in the discussion of an example in Fig. 11. We keep the incoming vertical edges and outgoing horizontal edges incident to an inactive subnode, because this would allow a node to switch its radio from the primary channel to the channel indicated by an inactive subnode to transport traffic. Algorithm 2 describes the integrated routing path computation and channel assignment. Note that

[3]A preliminary version of this algorithm has been presented in CrownCom 2007 [23].

[4]Note that instead of using traffic load information, other metrics can be used to select a node pair for processing. For example, we may process node pairs in the order of the existing routing path length that is formed in the initially self-organized CogNet.

[5]Our algorithm is flexible to allow different nodes to have different number of radios, and can capitalize on the scenario that some nodes have more than one radio.

[6]A subnode in layer *i* indicates channel *i*. For example, subnode A_1 indicates channel 1.

[7]In the case of the extended layered graph, the subnode here refers to both the primary and auxiliary subnodes.

[8]In the case of extended layered graph, both the incoming horizontal edges and outgoing vertical edges are incident to the primary subnode.

Algorithm 2 Path-centric channel assignment

1. Sort the nodes pairs by some metric, e.g., sort them at the descending order of traffic load.
2. Pick next node pair in the sorted list and compute a path between the selected node pair on the layered graph.
3. For each subnode A_i along the path, if there is an available radio interface in node A that has not been assigned with a channel, assign it to channel i (associated with subnode A_i), label A_i as an active subnode. If there is no more available radio in node A, mark all the unlabeled subnodes as inactive subnodes.
4. Update the vertical and horizontal edges incident to the subnodes on the path as follows.
   1) Remove all outgoing vertical edges incident to all inactive subnodes, but keep the incoming vertical edges.
   2) Remove the incoming horizontal edges incident to all inactive subnodes, but keep the outgoing horizontal edges.
   3) Increase the cost of horizontal edges incident to all subnodes on the path and the neighboring subnodes of each subnode on the path, as illustrated in Fig. 9, to indicate increased collision probability at these links/nodes.
5. If there is still a node pair in the sorted list to be processed, go to step 2.

in the case of the extended layered graph, the subnode in Algorithm 2 refers to both the primary and the corresponding auxiliary subnodes, e.g., both A_i and A'_i. In the centralized operating mode, after the channel assignment and routing paths for all nodes have been determined, the designated node sends such information to every node, e.g., through piggybacking in the routing protocol messages or a control channel.

We give an example for the path-centric channel assignment. Suppose we want to compute a routing path from node A to B in Fig. 6, where every node has one cognitive radio. First, we set the edge costs. For example, initially, we can set the cost of vertical edges as 1, the cost of horizontal edges as 2, and the cost of access edges of nodes A and B as 0, as illustrated in Fig. 7. With these edge costs, a shortest path can be computed by a standard shortest path algorithm. Suppose that the path {A, A_1, B_1,B} is computed, with path cost 2, illustrated in Fig. 8. Since the radios of nodes A and B have not been assigned to channels, channel 1 is assigned to the radios of both nodes A and B to

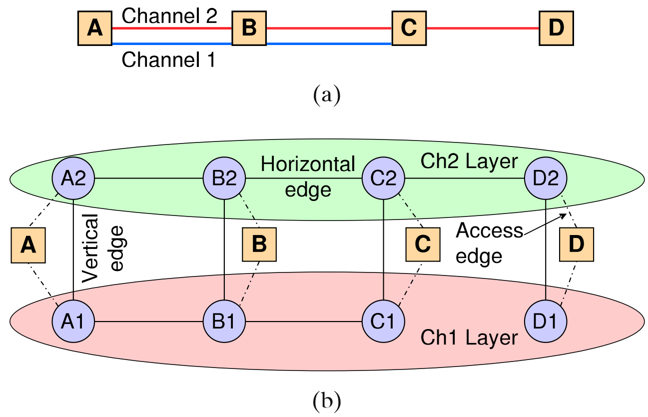


Figure 6 A 4-node CogNet and the constructed layered graph. **a** CogNet. **b** Layered graph

become their primary channel. Subnodes A_1 and B_1 are then labeled as active subnodes and subnodes A_2 and B_2 are labeled as inactive subnodes. The outgoing vertical edges and incoming horizontal edges incident to inactive subnodes A_2 and B_2 are removed as shown in the figure. Note that the bidirectional horizontal edge (A_2, B_2) is removed because this is actually equivalent to two unidirectional incoming horizontal edges, one to subnode A_2 and the other to B_2. At last, we update the cost of the horizontal edges so that in the next routing path computation, the horizontal edge cost is proportional to link usage. We also need to update the cost of horizontal edges that are not on the path but can be interfered by data communication on this path. These horizontal edges are the ones incident to the subnodes that are not on the path but are the neighbors of the subnodes on the path. Figure 9 illustrates the neighboring subnodes of the ones on the path, and the horizontal edges that are incident to these subnodes and need to increase edge cost. Here we assume that nodes F, G, H, and K are close to the routing path from node A to node B. Subnodes F_1 and G_1 are the neighbors of subnodes A_1 and B_1 that are on the routing path. The horizontal edges incident to these subnodes are drawn as bold lines. When the horizontal edge $A_1 \leftrightarrow B_1$ is

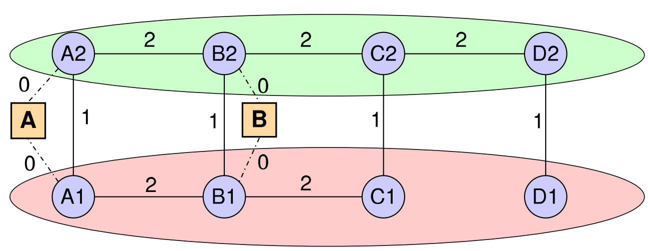


Figure 7 Example edge costs

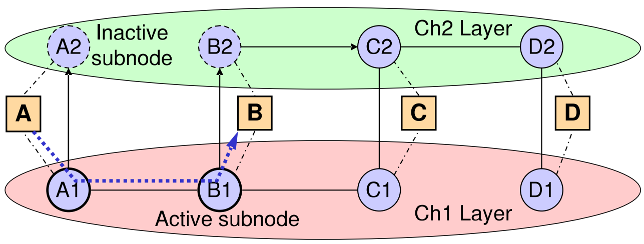


Figure 8 Updated layered graph after path **A**→**B** is computed

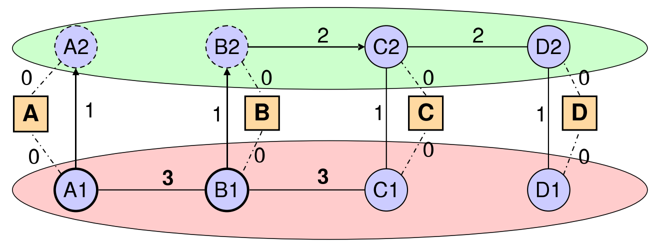


Figure 10 Updated edge costs

transmitting traffic, the bold horizontal edges in Fig. 9 are directly or indirectly interfered. For example, if node A is sending traffic to node B on channel 1, i.e., on the horizontal edge $A_1 \rightarrow B_1$, node G cannot send traffic to node K on the horizontal edge $G_1 \rightarrow K_1$ (using channel 1). Figure 10 illustrates the updated edge costs based on the costs in Fig. 7. The costs of edges $A_1 \leftrightarrow B_1$ and $B_1 \leftrightarrow C_1$ have been increased by 1 because they are the edges that will be interfered by data transmission on the routing path {A, A_1, B_1,B}.

Next we pick the node pair with the second largest traffic volume and calculate a shortest routing path in the layered graph. For all unlabeled subnodes on the path, we assign channels to the radios of their corresponding nodes as discussed above. For all active subnodes on the path, the channel assignment for their corresponding nodes have been determined previously during processing other node pairs. If there is an inactive subnode on the path that was labeled during processing another node pair previously, the upstream edge incident to this subnode must be a vertical edge, which means the corresponding node needs to switch its radio from its primary channel to the channel corresponding to this inactive subnode when transporting traffic on this routing path. On the other hand, the downstream edge incident to this inactive subnode must be a horizontal edge and the downstream subnode must be either an active subnode or an unlabeled subnode. In the latter case, the node corresponding to the downstream subnode assigns the channel indicated by this subnode to its radio. We let the node use a buffer to store packets to the downstream node since their radios are not in the same primary channel. When the buffer is full, this node switches the radio from its primary channel to the channel indicated by the inactive subnode, i.e., the primary channel of the downstream node's radio, to send out the buffered packets. Thus the channel assignment of intermediate nodes is again implicitly determined by this routing path. After updating the subnode status and horizontal edge cost, we proceed to calculate the routing path for the third node pair, and so on and so forth until the routing paths for all node pairs are calculated.

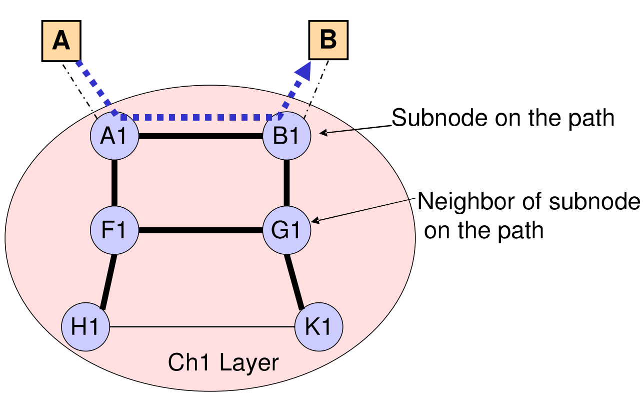


Figure 9 Updating the cost of horizontal edges

Figure 11 gives examples for the above discussions. Suppose that the node pair C→D has the second largest traffic volume. Then a routing path {C, C_2, D_2,D} can be computed, as in the case for node pair A→B. The radios of nodes C and D, which have not been assigned with channels, are both assigned with channel 2. Subnodes C_2 and D_2 are labeled as active subnodes and subnodes C_1 and D_1 are correspondingly labeled as inactive subnodes, as illustrated in Fig. 11. Now suppose that we are going to compute the routing path for node pair A→D. Based on the current layered graph, we will get a path as the dotted line illustrated in Fig. 11. The radios of all nodes on this path have already been assigned with channels earlier, with the radios of nodes B and C being in different primary channels. When a burst of packets on this path arrives at node B, it switches its radio from its primary channel to channel

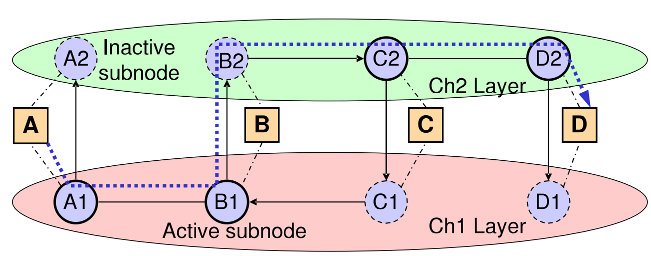


Figure 11 Computing the routing path from node A to D

2 at an on-demand basis to forward packets to node C. Note that in most link-centric channel assignment algorithms, nodes periodically and frequently switch between channels to maintain link-level connectivity to neighbors, which is not needed in our approach if there is no traffic to the neighboring node.

In the preceding discussions, we have examined how the path-centric channel assignment algorithm operates in the centralized mode. This algorithm can operate in the distributed mode as well, through a similar mechanism as the IP routing protocols. In this scenario, every node collects the channel and neighbor information together with traffic load statistics from all other nodes, piggybacked in the routing protocol messages in a self-organized CogNet, or through a control channel in the CogNet. Each node independently runs the path-centric channel assignment algorithm, same as above. After running this algorithm, the routing paths between node pairs are computed and the primary channel for the radio of each node is determined. The difference from the centralized mode is that in the distributed mode, this node does not send out the channel assignment of each node determined by this node. Instead, it only assigns its own radio with the channel determined by the algorithm, and records the computed routing paths from itself to other nodes. If every node collects the same or approximately the same channel and neighbor information, the channel assignment for every node that is independently determined at different nodes should be the same or approximately the same. Thus the channel assignment at each node in the distributed mode is the same or approximately the same as in the centralized mode.

5 Performance evaluation

We evaluate our algorithm in two sample networks, a 16-node torus network and a 25-node random mesh network. Each CogNet node has one radio. In the former topology, nodes are equally separated in a 1×1 unit plane and the radio transmission radius of each node is set as 0.35 to form a regular torus topology, as illustrated in Fig. 12. In the latter topology, nodes are randomly placed in the 1×1 unit plane through a uniform random variable for the location. The radio transmission radius of each node is also set as 0.35. This is due to the fact that although the node density of the mesh topology is higher than the torus topology, the node location in the mesh topology is random, and hence it is necessary to make the radio transmission radius slightly larger to avoid network partition. We assume 10 channels in our experiments, and the channel bandwidth is 10 Mbps. Each of these channels has probability 0.4 of being accessible and probability 0.6 of being inaccessible at a node, determined by a uniform random variable. The channel switching time is assumed to be 200 microseconds. The traffic loads between node pairs are random variables uniformly distributed over $[0, d]$ where d is the maximum packets arrival rate per second. Each node generates Poisson packet traffic. The packet size is randomly generated as follows with the packet header size excluded: generate 1-byte packets with a probability of 0.15, to simulate the TCP connection setup/termination packets; generate 1400-byte packets with a probability of 0.60, to simulate the full size Ethernet frame; and uniformly generate 2 to 1399-byte packets with a total probability of 0.25, to simulate the last packet in a burst of packets carrying a single TCP data segment from users.[9] Such packet size distribution is consistent with the statistics of network packet traces from the National Laboratory for Applied Network Research (http://www.nlanr.net). We use Carrier Sense Multiple Access in the MAC layer. The radio retransmits each packet up to 3 times in case of busy medium. It uses exponential backoff time with the initial waiting time being a random duration between 0 and 3 full packet transmission time.

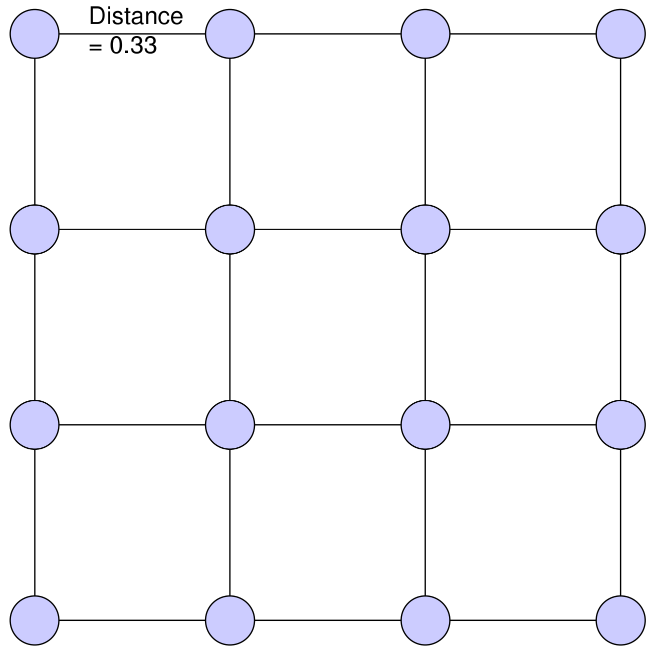


Figure 12 The 16-node torus network

[9]A large TCP data segment from users is partitioned into a burst of packets for transportation in the network, with all packets except the last one in full size (1400 bytes), and the last packet not in full size, since the data segment size is rarely in the exact multiples of the full packet size.

We use the extended layered graph for the routing path computation and channel assignment. The cost of all access edges is set as 1. The horizontal edge cost is initially set as 10, and adjusted during the routing path computation. Whenever a routing path is computed, the cost of all horizontal edges incident to a subnode on the path and its neighboring subnodes is increased, to discourage other node pairs from using these horizontal edges in their routing paths, as illustrated in Fig. 13. Note that some of these horizontal edges are directly interfered by the traffic transmission on the path, while others such as the horizontal edge $G_1 \leftrightarrow K_1$ are indirectly interfered. Therefore, we increase the cost of the former by 2 and the cost of the latter by 1. The cost of the vertical edge from a primary subnode to its own auxiliary subnode, i.e., $\langle A_i, A'_i \rangle$, is set as 10. We have classified the experiments into three scenarios based on the cost setting for the vertical edges from a primary subnode to the auxiliary subnodes of other primary subnodes, i.e., $\langle A_i, A'_k \rangle$ for $k \neq i$. The cost of such vertical edges is all set to 5, 10, or 20, which are accordingly denoted as *small cost scenario* (SCS), *medium cost scenario* (MCS), and *large cost scenario* (LCS), respectively.

In the simulation, we compare our algorithm to the link-centric channel assignment algorithm for CogNet proposed in [21], which has been briefly introduced in Section 1. We describe it in more details here. One of the major objectives of the algorithm in [21] is to make all nodes in a CogNet form a connected network by appropriate channel assignment, so that any two nodes can communicate. Due to the dynamic and heterogeneous channels at each node, it is usually not possible to use one common channel to connect all nodes into one connected network. In this algorithm, each node selects a channel for its radio through a local decision and does not need to coordinate with nodes beyond its direct neighbors. This is particularly suitable for the initial self-organization of a CogNet. Every node scans the accessible channels and discovers the neighbors reachable in each of these channels. This node also exchanges control information with its neighbors at each channel, e.g., the number of nodes that can be reached by the neighbor in single or multiple hops at the corresponding channel. The node then selects a channel based on this information, e.g., the channel that can reach the maximum number of nodes. The nodes selecting the same channel form a group. In other words, the nodes in each group have a common channel and can communicate with each other. To transport traffic between two nodes in different groups, the boundary nodes between one or more groups periodically switch among channels of these groups, so that the traffic can be carried over from one group to another group. Figure 14 shows an example of this algorithm. The five nodes form two groups, using channels 1 and 2, respectively. The boundary node C periodically switches between channels 1 and 2 to reach the nodes in both group 1 and group 2, so that the nodes in different groups can communicate with each other. Other nodes are fixed at the assigned channel. In our implementation, after scanning channels and discovering neighbors, each node simply joins the largest group, i.e., selects the channel of the neighbor in the largest group. The boundary nodes switch channels between different groups every 5 ms.

Figure 15 illustrates the performance of the 16-node torus network. Figure 15a plots the throughput, calculated as the number of received packets divided by the number of generated packets, using the path-centric and link-centric channel assignment algorithms, respectively. The presented throughput is an average value from 6 experiments using different seeds. The throughput in terms of bytes is similar to the one in terms of packets and is omitted here. From

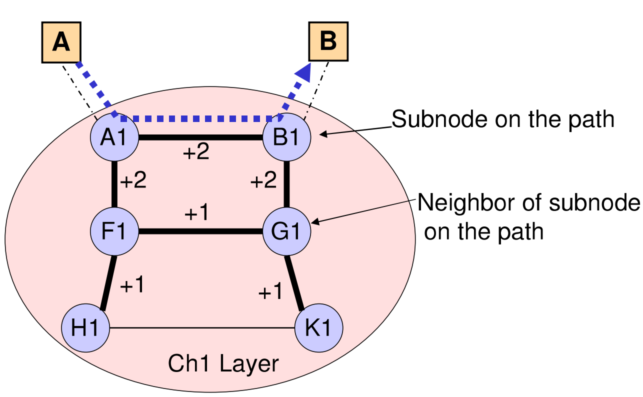


Figure 13 The cost updating for horizontal edges in simulations

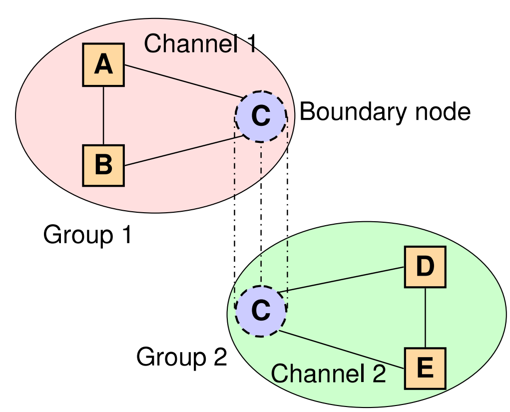


Figure 14 Illustration of the channel assignment algorithm in [21]

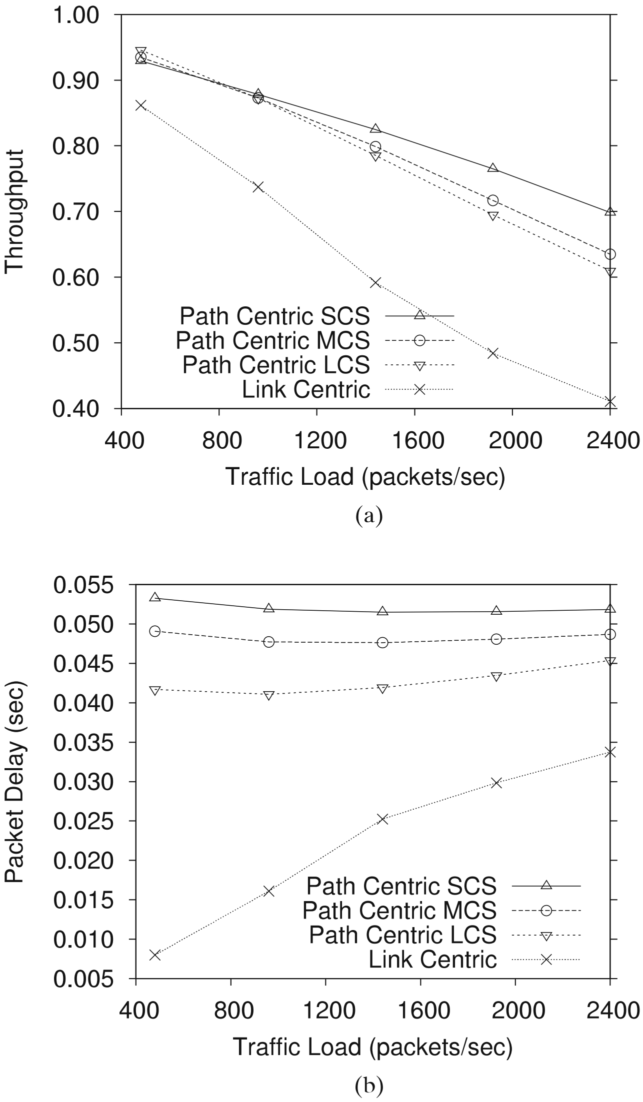


Figure 15 Performance of the 16-node torus network. **a** Throughput. **b** Average packet delay

Fig. 15a, we can observe that our channel assignment algorithm significantly performs better. This is due to the coordination between the routing protocol and the channel assignment, which results in load balancing on different channels and on-demand channel switching to reduce overhead. When the traffic load is light, a node stays on its primary channel most of time and does not switch frequently between channels, to avoid channel switching overhead. Among the three scenarios of path-centric channel assignment, SCS outperforms both MCS and LCS. In other words, when the cost of vertical edges from a (primary) subnode to the auxiliary subnodes in other layers is smaller than the cost of the vertical edge from this subnode to its own auxiliary subnode, the performance is the best. On the other hand, LCS, which assigns a larger cost to the vertical edges going to the auxiliary subnodes in other layers and thus favors the selection of the same channel on the downstream link as the one from the upstream link, has the worst performance as shown in Fig. 15a. This confirms that selecting different channels on adjacent links of the routing paths reduces interference and thus increases performance.

Next we examine the average packet delay as illustrated in Fig. 15b. A long packet delay in the network may result in problems for higher layer protocols. Hence, in the simulation, we put a limit for the packet delay. If a packet has traveled more than 100 ms (including transmission and queuing delay), the packet is dropped. As discussed in the last section, in the path-centric channel assignment algorithm, when a packet arrives, if the downstream node is not in the same channel of this node, this packet is placed in a buffer. In the experiments, we assume that the buffer size is 50 packets. When the buffer is full, the node switches its radio to the channel of the downstream node and sends these buffered packets in a burst mode. Thus the average packet delay increases due to this queueing time in the buffer and channel switching. On the other hand, many nodes in the link-centric channel assignment algorithm operate at the same channel and the packet delay is smaller. The packet delay for the link-centric channel assignment algorithm increases with heavier traffic load because the probability of collision is higher, which results in a longer retransmission delay. For the path-centric channel assignment algorithm, one of the major factors of the packet delay is the queueing time in the buffer and the channel switching due to the packets going to neighbors on a different channel. Thus the packet delay only slightly increases when the traffic load increases. From Fig. 15b, the packet delay in the scenario of SCS is larger than the other two scenarios of the path-centric channel assignment algorithm. This is because in SCS, more packets are successfully transported, which results in higher throughput, but also leads to a longer queueing delay.

Now we study the performance of the 25-node mesh network where nodes are randomly distributed. Figure 16 plots the throughput and average packet delay. With the random mesh topology, the performance gain of our path-centric channel assignment algorithm further increases. The average packet delay for the path-centric channel assignment algorithm is almost flat with even slight decrement. As discussed before, the major factors of the packet delay are the queueing time and the channel switching due to the packets going to neighbors on a different channel. With heavier traffic load, this queueing time would decrease because the

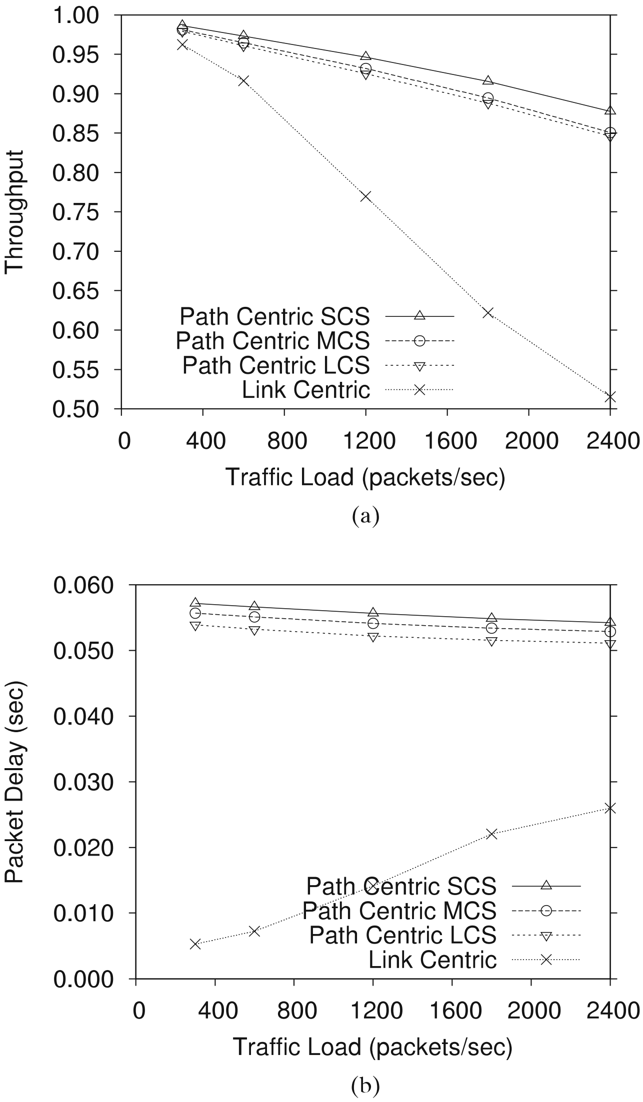


Figure 16 Performance of the 25-node random mesh network. **a** Throughput. **b** Average packet delay

time spent in the buffer decreases, since it takes less time for the buffer to become full than in the case of heavier traffic load.

In the simulation results discussed above, we assumed that only the neighboring nodes are interfered by data transmission at a given node. In some environments highly sensitive to interference, the radio signal generated by data transmission at a given node may extend beyond the neighboring nodes and interfere non-neighboring nodes. Note that if a non-neighboring node is not too far from the given node, it can still hear the radio signal, although the data cannot be correctly received (otherwise it would be a neighboring node), i.e., for this node to correctly receive the data, there is a certain threshold of the signal to interference ratio. We examine the performance of channel assignment in such environments. The radio transmission radius is still 0.35 as before, but the interference radius is set as 0.7, i.e., data transmission at a node interferes all nodes within 0.7 unit distance. Figure 17 plots the throughput and packet delay in the 25-node random mesh network with the large interference radius. We can observe that the performance gain of the path-centric channel assignment algorithm over the link-centric channel assignment algorithm is still large in this scenario, compared with the results in Fig. 16a. Note that the throughput in Fig. 17a is lower than the one in Fig. 16a because of higher interference.

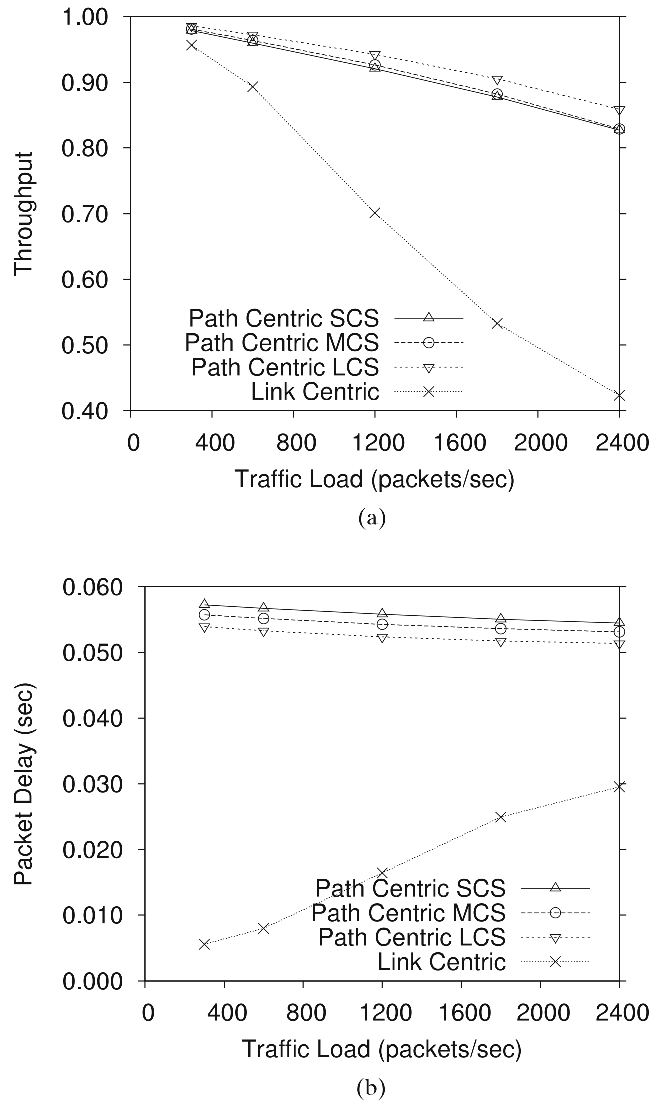


Figure 17 Performance of the 25-node random mesh network with a large interference radius. **a** Throughput. **b** Average packet delay

6 Conclusion and future work

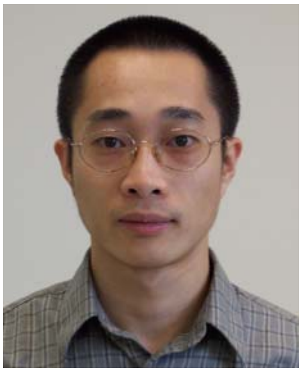
In this paper, we have proposed a path-centric channel assignment framework for cognitive radio wireless networks (CogNets), which takes a different perspective from traditional channel assignment approaches for multi-channel wireless networks. The proposed framework includes both a modeling tool named *layered graph* for CogNets and a channel assignment algorithm utilizing the layered graph. The framework couples both routing and channel assignment, and determines the channel assignment for each node to achieve globally optimized performance, rather than focusing on the local node with non-coordinated channel assignment. The numerical results show that the path-centric channel assignment outperforms the traditional link-centric channel assignment algorithms. As the future work, we plan to add the dynamics of channel arrival/departure and consider the overhead of the channel reassignment when the working channels become inaccessible due to the preemption by licensed users.

References

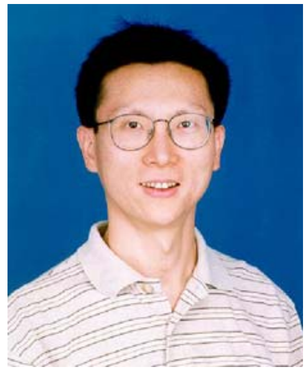
1. NTIA (2003) US frequency allocation chart. http://www.ntia.doc.gov/osmhome/allochrt.pdf
2. FCC Spectrum Policy Task Force (2002) Report of spectrum efficiency working group. http://www.fcc.gov/sptf/files/SEWGFinalReport_1.pdf. Accessed November 2002
3. FCC (2004) Unlicensed operation in the TV broadcast bands. ET Docket No. 04-186, Notice of Proposed Rulemaking (NPRM)
4. Cordeiro C, Challapali K, Birru D, Shankar S (2005) IEEE 802.22: the first wirldwide wireless standard based on cognitive radios. In: Proc. IEEE DySPAN, Baltimore, pp 328–337, 8–11 November 2005
5. Brown T (2005) An analysis of unlicensed device operation in licensed broadcast service bands. In: Proc. IEEE DySPAN, Baltimore, pp 11–29, 8–11 November 2005
6. Leu AE, Steadman K, McHenry M, Bates J (2005) Ultra sensitive TV detector measurements. In: Proc. IEEE DySPAN, Baltimore, pp 30–36, 8–11 November 2005
7. Poston JD, Horne WD (2005) Discontiguous OFDM consideration for dynamic spectrum accessin idle TV channels. In: Proc. IEEE DySPAN, Baltimore, pp 607–610, 8–11 November 2005
8. Draves R, Padhye J, Zill B (2004) Routing in multi-radio, multi-hop wireless mesh networks. In: Proc. ACM MobiCom, Philadelphia, pp 114–128, 1 October 2004
9. Nasipuri A, Das SR (1999) A multichannel CSMA MAC protocol for multihop wireless networks. In: Proc. IEEE wireless communications and networking conference (WCNC), New Orleans, 21–24 September 1999
10. Nasipuri A, Das SR (2000) Multichannel CSMA with signal power-based channel selection for multihop wireless networks. In: Proc. IEEE fall vehicular technology conference (VTC), Boston, 24–28 September 2000
11. Jain N, Das SR, Nasipuri A (2001) A multichannel MAC protocol with receiver-based channel selection for multihop wireless networks. In: Proc. IEEE international conference on computer communication and networks (ICCCN), Phoenix, October 2001
12. Adya A, Bahl P, Padhye J, Wolman A, Zhou L (2004) A multi-radio unification protocol for IEEE 802.11 wireless networks. In: Proc. IEEE international conference on broadband networks (BroadNets), San Jose, 25–29 October 2004
13. Raniwala A, Gopalan K, Chiueh T (2004) Centralized channel assignment and routing algorithms for multi-channel wireless networks. Mobile Comput Commun Rev 8(2):50–65
14. Kyasanur P, Vaidya NH (2004) Routing and interface assignment in multi-channel multi-interface wireless networks, Coordinated Science Laboratory, University of Illinois at Urbana-Champaign, Technical Report
15. Wu S-L, Lin C-Y, Tseng Y-C, Sheu J-P (2000) A new multichannel MAC protocol with on-demand channel assignment for multi-hop mobile ad hoc networks. In: Proc. Int'l symposium on parallel architectures, algorithms and networks (I-SPAN), Dallas, 7–9 December 2000
16. Wu S-L, Tseng Y-C, Lin C-Y, Sheu J-P (2002) A multichannel MAC with power control for multi-hop mobile ad hoc networks. Comput J 45(1):101–110
17. Hung W-C, Law E, Leon-Garcia A (2002) A dynamic multichannel MAC for ad hoc LAN. In: Proc. 21st Biennial symposium on communications, pp. 31–35, Kingston, June 2002
18. So J, Vaidya NH (2004) Multi-channel MAC for ad hoc networks: handling multi-channel hidden terminals using a single transceiver. In: Proc. ACM MobiHoc, Tokyo, pp 222–233, 24–26 May 2004
19. Shacham N, King P (1987) Architectures and performance multichannel multihop packet radio networks. IEEE J Sel Areas Commun 5(6):1013–1025
20. Bahl P, Chandra R, Dunagan J (2004) SSCH: slotted seeded channel hopping for capacity improvement in IEEE 802.11 ad-hoc wireless networks. In: Proc. ACM MobiCom, Philadelphia, 1 October 2004
21. Zhao J, Zheng H, Yang G-H (2005) Distributed coordination in dynamic spectrum allocation networks. In: Proc. IEEE DySPAN, Baltimore, pp. 259–278, 8–11 November 2005
22. Xin C, Xie B, Shen C-C (2005) A novel layered graph model for topology formation and routing in dynamic spectrum access networks. In: Proc. IEEE DySPAN, Baltimore, pp 308–317, 8–11 November 2005
23. Xin C, Ma L, Shen C-C (2007) Path-centric channel assignment in cognitive radio wireless networks. In: Proc. CrownCom, Orlando, 31 July–3 August 2007

Chunsheng Xin received the Ph.D. degree in computer science from State University of New York at Buffalo in 2002. From 2000 to 2002, he was a Research Co-Op in Nokia Research Center,

Boston. From 2002, he is an assistant professor in the Computer Science Department, Norfolk State University, Norfolk, Virginia. His research interests include optical networks, cognitive radio wireless networks, and performance evaluation and modeling.



Liangping Ma received his B.S. degree in Physics from Wuhan University, Hubei, China, in 1998, and his Ph.D. degree in Electrical Engineering from the University of Delaware, Newark, DE, in 2004. He was with the University of Delaware as a Postdoctoral Research Fellow. Since 2005, he has been with San Diego Research Center, Inc. (now part of Argon ST, Inc.), San Diego, CA, as a Research Staff Member. His research interests include medium access control (MAC), spectrum agile radios, and signal processing.



Chien-Chung Shen received his B.S. and M.S. degrees from National Chiao Tung University, Taiwan, and his Ph.D. degree from UCLA, all in computer science. He was a senior research scientist at Bellcore (now Telcordia) Applied Research working on control and management of broadband networks. He is now an associate professor in the Department of Computer and Information Sciences of the University of Delaware, and a recipient of NSF CAREER Award. His research interests include ad hoc and sensor networks, dynamic spectrum management, control and management of broadband networks, distributed object and peer-to-peer computing, and simulation. He is a member of both ACM and IEEE.