TIME-EFFICIENT LAYER-2 AUTO-CONFIGURATION FOR COGNITIVE RADIOS

S.Krishnamurthy
Computer Engineering Program
The University of Texas at Dallas
Richardson, TX 75083, U.S.A
email: ksrini@student.utdallas.edu

M.Thoppian, S.Kuppa, S.Venkatesan, R.Chandrasekaran, N.Mittal and R.Prakash
Department of Computer Science
The University of Texas at Dallas
Richardson, TX 75083, U.S.A
email: {mansi, ksrikant, venky, chandra, neerajm, ravip}@utdallas.edu

ABSTRACT

Cognitive radios (CR) have the ability to dynamically adapt to local spectrum availability. In a network comprised of CR-enabled devices, layer-2 auto-configuration involves determining a common set of channels (also referred to as the global channel set) to facilitate communication among participating nodes. This is a unique challenge as nodes in the CR network are unaware of (a) their neighbors and (b) the channels on which they can communicate with a neighbor. In this paper, we propose a time-efficient distributed algorithm for layer-2 auto-configuration for a CR network. Our algorithm finds the global channel set in $O(N(M+D))$ timeslots, where N is the maximum number of nodes deployed, M is the maximum number of available channels for communication and D is the diameter of the network. All nodes know N and M. We present a diameter-aware and a diameter-unaware version of the algorithm. For highly sparse networks (such as a linear chain topology where $D = N - 1$), with $N = 40$ and $M = 80$, the diameter-aware configuration protocol terminates within 8 $seconds$ and the diameter-unaware version terminates within 12 $seconds$.

KEY WORDS

Cognitive Radio, Multi-channel, Auto-configuration, Distributed Algorithm

1 Introduction

Cognitive radio (CR) technology [1] allows wireless devices to dynamically adapt to spectrum availability in their geographical region. The owner of a (licensed) channel is referred to as *primary user* and all other users of the channel as *secondary users* [2]. CR technology enables secondary users to scan and identify available channels in a frequency spectrum. A channel is said to be *available* if the secondary user can send and receive messages on the channel without interfering with the primary user(s).

Communication infrastructure based on CR technology is of particular relevance in defense and relief operations. Since the usage of spectrum varies widely from one region to another [3], communication among users (soldiers in a platoon or relief personnel in a disaster-prone area) must rely on a dynamic channel assignment scheme. Also, in military applications and disaster recovery scenarios such as the shuttle recovery effort in east Texas or hurricane affected areas, significant parts of the spectrum are likely to be available for use by secondary users.

When a secondary user (hereafter, referred to as a node in the CR network) independently scans the spectrum usage and maintains the set of available channels, the following layer-2 auto-configuration issues arise:

1. How do nodes detect their neighbors and collectively form a communication infrastructure in the absence of a central authority?
2. How do nodes decide on the set of channel(s) that can be used for communication?

Wireless communication among neighbors is possible if the source node and destination(s) tune to the common channel at the same time. In the layer-2 auto-configuration problem in a CR network, a *common set of channels* (referred to as the *global channel set*, $\mathcal{G}$, in this paper) needs to be determined. The motivations behind finding $\mathcal{G}$ are:
(i) There may be multiple groups of nodes deployed in a geographical area, say in a military operation (there may be many platoons, with each platoon being a group) or at the site of a natural disaster (firemen, paramedics, police being three groups). It is important that each group chooses a unique channel for communication among themselves with few nodes acting as gateways between groups.
(ii) Tuning overheads are incurred when nodes have to switch from one frequency to another. By making nodes communicate on a globally common channel, say c_{global}, we can avoid such tuning overheads.
(iii) Node mobility leads to frequent changes in network topology. For such systems, communication over c_{global} provides a simple and effective solution ([4], [5]).
(iv) Since c_{global} is available at all the nodes, which could be distributed over a wide geographical region, using a globally common channel (such as c_{global}) leads to a fairly stable communication infrastructure.

Finding a common set of available channels (i.e. global channel set) for communication is non-trivial because of the divergence in the sets of available channels at individual nodes and the absence of a central authority[1]. The complexity of the problem is further increased due to the following reasons:
(i) Nodes do not have prior knowledge about the number and identities of nodes in their neighborhood.
(ii) Nodes are unaware of the existence of a common channel and its identity (if there is one).
(iii) Changes in neighborhood due to node mobility can play a significant role in computing $\mathcal{G}$. So, it is very important that the distributed computation terminates quickly.

---

[1] This is because communication infrastructures in military and relief operations are usually ad hoc in nature.

Let N be the total number of possible nodes and $\mathcal{A}_{univ}$ be a set of M possible channels the nodes can operate on. In this paper, we propose a layer-2 auto-configuration protocol that enables the nodes to dynamically compute the global channel set, $\mathcal{G}$ in a distributed manner. All nodes know $\mathcal{A}_{univ}$ and the value of N[2]. We present both diameter-aware and diameter-unaware versions of the protocol. The worst-case time complexity of both versions of the algorithm is $O\left(N\left(M+D\right)\right)$ timeslots. If D is known, then the number of bits exchanged per message is $O(M)$. Otherwise, the number of bits per message is $O(M+logN)$. The protocol runs independent of the cardinality of the global channel set $\mathcal{G}$. To the best of our knowledge, there is no existing work in the literature that addresses this layer-2 auto-configuration problem for cognitive radios.

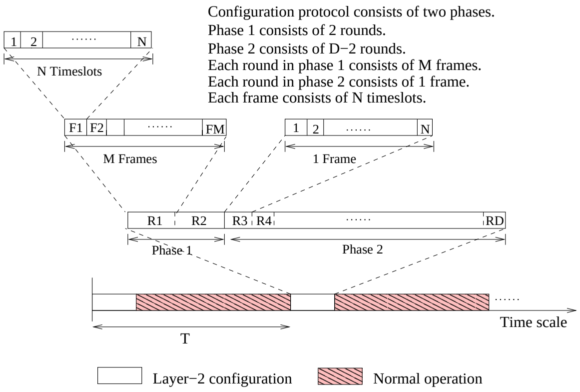


Figure 1. Operation cycle of a diameter-aware node

2 System model

Throughout this paper, we consider a mobile multi-hop wireless network formed by a group of CR-enabled nodes.

2.1 Node characteristics

Every node, say i, is assigned a unique identifier, say UID_i in the range $[1 \ldots N]$, where N is an upper bound on the total number of nodes. Since the envisioned applications are military and relief operations where the maximum number of soldiers in a platoon or firemen assigned for relief efforts is known a priori, we assume that the maximum number of nodes (N) is known to every node. For simplicity, we assume $UID_i = i$ throughout this paper. Every node has a single wireless transceiver that can transmit or receive in any of the M channels of $\mathcal{A}_{univ}$. Each node is equipped with a GPS receiver [6] to enable time synchronization among nodes. In military and disaster relief operations, it has recently become a norm for personnel to carry GPS-equipped devices ([7]).

2.2 Medium characteristics

We assume that the communication medium is loss-free. Let $\mathcal{A}_{univ} = \{c_1, c_2 \ldots c_M\}$ represent the universal set of available channels that can be potentially used by all nodes for communication. All the nodes know N and $\mathcal{A}_{univ}$. (For example, $\mathcal{A}_{univ}$ can include a set of 80 channels in the 3-5 GHz range.) We assume that every channel has a unique identity.

2.3 Network operation

Nodes in the CR network perform one of the following two operations at any time: (i) layer-2 auto-configuration, or (ii) normal operation as shown in Figure 1. These two operations repeat periodically every T time units. The time at which each operation starts is known in advance to all the nodes. This is made possible by exploiting the GPS capability [6] and by letting every node know the value of T[3]. (T is configurable.) During the layer-2 auto-configuration, nodes learn about other participating nodes in the network and also determine the global channel set, $\mathcal{G}$. In the normal mode of operation, the nodes may behave similar to the nodes in any other multi-hop wireless network, such as MANET [8] or mesh network [9]. It is necessary to periodically invoke the layer-2 auto-configuration protocol to account for varying global channel set, $\mathcal{G}$, due to the changes in network topology and/or channel availability set maintained by individual nodes. (See Section 4.3 for more details.)

3 Layer-2 auto-configuration protocol

During layer-2 auto-configuration process, time division multiple access (TDMA) scheme is used for communication among nodes. Time is split into equal intervals referred to as *frames* as shown in Figure 1. Each frame is further divided into N *timeslots*, each of equal length. The slot assignment for a node, say i, is done in advance according to its UID_i. Since $UID_i = i$, node i transmits during the i^{th} slot in each frame (see Figure 1) and all other nodes are in receive mode. This ensures that every node in the network gets one chance to transmit without collisions during each frame.

3.1 Data structures

The following data structures are maintained at every node i:

UID_i	Identity of the node i
$\mathcal{A}_i$	Set of available channels from $\mathcal{A}_{univ}$ at i (sorted)
NBR_i	Set containing one-hop neighbors of node i
pc_i	Preferred channel for node i
$\mathcal{G}$	Global channel set
r	Current round number

Availability set, $\mathcal{A}_i$ is obtained by node i by scanning the medium and estimating the available channels. This step is

---

[2] When the radios are built, they have a range of frequencies they can operate on.

[3] Note that all nodes can have 00:00:00 GMT on January 1, 2000 as the starting reference time.

external to our algorithm. Nodes i and j are *neighbors* of each other if both are within communication range of each other and $\mathcal{A}_i \cap \mathcal{A}_j \neq \emptyset$. A *preferred channel* for node i is a channel on which transmissions by i can be received by all of its neighbors. If there are multiple common channels, any one of them can be selected as a preferred channel. The concept of a *round* is defined to monitor the progress of the configuration protocol, like any other synchronous distributed algorithm [10]. Initially, $UID_i = i$, $NBR_i = \emptyset, pc_i = NULL, \mathcal{G} = \mathcal{A}_i$ and $r = 0$.

3.2 Diameter-aware auto-configuration

Assume that all the nodes are aware of the diameter of the network, D. The following algorithm determines the global channel set. The algorithm consists of two phases (see Figure 1).

3.2.1 First phase

The first phase consists of two rounds. During the first round, each node i determines its *preferred channel* (pc_i). During the second round (of the first phase) all nodes exchange their *preferred channel* information with their respective neighbors. Each round consists of M frames $(F_1, F_2 \ldots F_M)$ and each frame consists of N slots as shown in Figure 1. During frame F_j $(1 \leq j \leq M)$, every node i with $c_j \in \mathcal{A}_i$ tunes its transceiver to channel c_j.

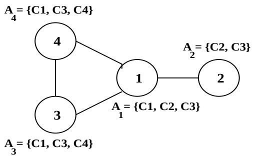


Figure 2. A Sample CR Network

PHASE 1, ROUND 1

	Timeslot 1	Timeslot 2	Timeslot 3	Timeslot 4
Frame 1	{C1, C2, C3}		{C1, C3, C4}	{C1, C3, C4}
Frame 2	{C1, C2, C3}	{C2, C3}		
Frame 3	{C1, C2, C3}	{C2, C3}	{C1, C3, C4}	{C1, C3, C4}
Frame 4			{C1, C3, C4}	{C1, C3, C4}

PHASE 1, ROUND 2

	Timeslot 1	Timeslot 2	Timeslot 3	Timeslot 4
Frame 1	{C3}		{C1, C3}	{C1, C3}
Frame 2	{C3}	{C2, C3}		
Frame 3	{C3}	{C2, C3}	{C1, C3}	{C1, C3}
Frame 4			{C1, C3}	{C1, C3}

Figure 3. Transmissions during Phase I

First Round: Consider the j^{th} frame F_j. All nodes tune to channel c_j and every node i transmits the contents of its set $\mathcal{G}$ on channel $c_j \in \mathcal{A}_i$ during i^{th} timeslot of frame F_j. This corresponds to $((j-1) \times N + i)^{th}$ timeslot from the beginning of the first round. During the remaining time slots in this frame, node i is in the receive mode. If $c_j \notin \mathcal{A}_i$, then node i remains silent during the i^{th} slot of frame F_j. If node i receives node j's transmission (during the j^{th} slot) of a frame F_k, node i learns that $j \in NBR_i$ and $c_k \in A_j$. In this case, node i updates its data structures.

Consider the sample network shown in Figure 2. Here, node 2 transmits the set $\mathcal{G} = \mathcal{A}_2 = \{c_2, c_3\}$ during frames F_2 and F_3 at $((2-1) \times 4 + 2) = 6^{th}$ and $((3-1) \times 4 + 2) = 10^{th}$ timeslots (from the beginning of the first round). During slots $2 = (1-1) \times 4 + 2$, and $14 = (4-1) \times 4 + 2$, node 2 remains silent since c_1 and c_4 do not belong to $\mathcal{A}_2$. Figure 3 shows the transmissions of the four nodes during the first round. After the first round, each node i knows the identities of all its one-hop neighbors (maintained locally in set NBR_i) and their respective availability sets. Node i updates the set $\mathcal{G}$ as follows: $\mathcal{G} = \mathcal{G} \cap (\cap_{j \in NBR_i} A_j)$. It also increments r by one.

Node i can select one of the channels in the updated set $\mathcal{G}$ as its *preferred channel* (pc_i) for transmission. When node i transmits on its preferred channel during i^{th} timeslot, its transmission can be received by all of its one-hop neighbors if all nodes $j \in NBR_i$ tune to pc_i during i^{th} timeslot. For the sample network in Figure 2, $NBR_1 = \{2, 3, 4\}$ and the updated set $\mathcal{G}$ for node 1 is $\mathcal{G} \cap (\cap_{j \in NBR_1} A_j) = \{c_3\}$. So, when node 1 transmits on the channel c_3 during the first timeslot, its transmission can be heard by all of its one-hop neighbors if they tune to c_3. After the first round, only node i knows about its preferred channel. Before node i begins to transmit *only* on pc_i, it has to inform its neighbors about its selection. The second round of the algorithm is used for this.

Second round: Nodes exchange the updated set $\mathcal{G}$ in the second round. Every node i transmits $\mathcal{G}$ on channel $c_j \in \mathcal{A}_i$ during i^{th} timeslot of frame F_j. Once again, consider the sample network shown in Figure 2. After the first round, the updated set $\mathcal{G}$ at nodes $1 \ldots 4$ are $\{c_3\}$, $\{c_2, c_3\}$, $\{c_1, c_3\}$ and $\{c_1, c_3\}$, respectively. Transmission by nodes $1 \ldots 4$ during the second round are shown in Figure 3. Nodes use a deterministic mechanism to select a preferred channel from among the channels in $\mathcal{G}$ at the end of the first round (for example, the smallest channel). A node implicitly conveys this information to all its one-hop neighbors during the second round by transmitting its updated $\mathcal{G}$. At the end of the second round, each node i updates the set $\mathcal{G}$ as described before and increments r by one. In our example, the updated set $\mathcal{G}$ after the second round at nodes $1 \ldots 4$ are $\{c_3\}$, $\{c_3\}$, $\{c_3\}$ and $\{c_3\}$, respectively. Note that this updated set now gives the set of channels that are common to a node and all other nodes that are within its 2-hop distance.

3.2.2 Second phase

The second phase of the algorithm consists of $D-2$ frames with each frame divided into N timeslots (see Figure 1). Each node i now transmits only on its preferred channel, pc_i, (that was agreed upon during the first phase) during its pre-assigned timeslot. This effectively reduces the number of timeslots, and in turn, reduces the time complexity of the algorithm. As in the first phase, each node i continues to transmit its updated set $\mathcal{G}$. At the end of the k^{th} frame of the second phase, each node i knows the set of channels that are common to i and all the nodes that are within $(k+2)$ hops from node i. Thus, for $k = D-2$ $(r = D)$, every node i is aware of the global channel set and the algorithm terminates.

Upon termination, the proposed auto-configuration protocol provides the following:

- All the nodes are able to identify their one-hop neighbors.
- Every node i learns the set of channels that is common to itself and all nodes within k-hop distance from node i, for each $1 \leq k \leq D$. This information is very useful when $\mathcal{G} = \emptyset$ to support some communication infrastructure (see Section 4.1).
- It identifies the global channel set $\mathcal{G}$, and hence, enables the normal operation (that follows the auto-configuration process) to take place on one of the channels in the set $\mathcal{G}$ (if non-empty).

While the auto-configuration process uses slotted time, any other communication mechanism (such as scheduling, contention or reservation-based scheme) can be used during the normal operation of the nodes.

3.3 Diameter-unaware auto-configuration

Consider the case where all the nodes are not aware of the diameter of the network, D. Note that even though all the nodes know the set $\mathcal{G}$ in D rounds (see proof of correctness in Section 3.4), the nodes lack sufficient local knowledge to determine that D rounds have been completed and the auto-configuration protocol may be terminated.

Peleg proposed a distributed time-optimal leader election algorithm that runs in $O(D)$ time even when the nodes in the general network lack the knowledge of D [10]. We propose to run Peleg's algorithm in parallel with our layer-2 auto-configuration protocol to determine the terminating condition in a diameter-unaware scenario. Before proceeding further, recall that the term *round* signifies the duration it takes for every node in the network to communicate with all of its neighbors. Peleg's algorithm runs for at least $\left(\frac{3D}{2}+2\right)$ rounds. Since the set $\mathcal{G}$ is available in D rounds and each node only performs a set intersection operation at the end of every round, further rounds of same procedure will not change $\mathcal{G}$. Thus, running Peleg's algorithm in parallel with the auto-configuration protocol will not affect its correctness as far as determining the set $\mathcal{G}$ is concerned.

To run Peleg's algorithm in parallel with the configuration protocol, every node i includes the following two additional pieces of information along with the set $\mathcal{G}$ during each transmission: (i) estimate of the leader (highest UID value seen so far), say UID_L, and (ii) estimate of longest distance (in number of hops), say d, from node L to any node in the network. This information is updated every round. A potential leader node, say j, receives increasing values of d at the end of every second round. Upon receiving three consecutive identical values of d, node j concludes that it is the leader in the network and it has implicitly communicated with all nodes in the network. It then broadcasts a signal for other nodes to terminate. For more details on this algorithm, readers are referred to the research note in [10].

3.4 Proof of correctness

Let $\mathcal{G}_{ik}$ be the set of channels that are common to node i and all the other nodes that are within k-hops from node i. Global channel set $\mathcal{G} = \mathcal{G}_{iD}$.

Theorem: Consider an arbitrary node i in the network. At the end of D rounds, where D is the diameter of the network, node i is aware of $\mathcal{G}_{iD}$.

Proof (*by induction*):

Basis step $(m = 1)$: Consider a node x that is a neighbor of node i. Let $c_k \in \mathcal{A}_i \cap \mathcal{A}_x$. During k^{th} frame, node i is in receive mode on c_k except in the i^{th} timeslot (when it transmits). As a result, node i will hear x's transmission on c_k in the x^{th} timeslot of k^{th} frame. Similarly, node i will hear from all its neighbors with c_k in their availability set during the k^{th} frame (of N timeslots). As M is the maximum number of channels in $\mathcal{A}_{univ}$, by the end of NM timeslots, node i would have heard from all its neighbors. Thus, at the end of the first round, node i is aware of $\mathcal{G}_{i1}$.

Induction Hypothesis: Assume the result is true for some $m \in \mathbf{Z}^+ \wedge 1 < m < D$, i.e. at the end of m^{th} round, node i is aware of $\mathcal{G}_{im}$.

Inductive Step: Consider the $(m+1)^{st}$ round. Node i hears from all its neighbors during this round. By inductive hypothesis, for any arbitrary node x that is a neighbor of i, i receives $\mathcal{G}_{xm}$ during the $(m+1)^{st}$ round. Consider a node j that is m-hops away from node x and greater than m-hops away from node i. The set $\mathcal{G}_{xm}$ includes the information about the set $\mathcal{A}_j$. The node j can now be at most $m+1$ hops away from i. At the end of $(m+1)^{st}$ round, i will be aware of $\mathcal{A}_j$ through $\mathcal{G}_{xm}$. This is true for any arbitrary node j that is in the $(m+1)$-hop neighborhood of node i. Thus, $\mathcal{G}_{i(m+1)}$ is available at the end of $(m+1)$ rounds.

Hence, at the end of D rounds, node i has $\mathcal{G}_{iD}$, which is same as the global channel set $\mathcal{G}$. ■

3.5 Complexity analysis

The diameter-aware auto-configuration protocol requires D rounds for completion. The first two rounds require NM timeslots each and the remaining $(D-2)$ rounds require N timeslots for each round. Thus, the time complexity of the protocol is $N\left(2M+(D-2)\right)$ timeslots. As each node transmits information related only to its channel set, the number of bits carried per payload is $O(M)$.

For a network of 40 nodes[4] and 80 channels[5], a linear chain topology (diameter, $D = N-1$) and timeslot duration of $1\ ms$ [12], the diameter-aware protocol terminates within $8\ seconds$. Note that the timeslot duration of $1\ ms$ includes the time required for changing channel frequency, preamble required to establish message bit synchronization, and guard bands for synchronization error and propagation time[6].

Peleg's time-optimal leader election algorithm terminates in $3d+2$ rounds, where $d \leq D \leq 2d$. In the

---

[4] In the army, a unit of thirty to forty soldiers forms a platoon.

[5] In IEEE 802.11b devices operating in the 2.4 GHz band [11], there are three non-overlapping channels (Channels 1, 6 and 11) that are $25\ MHz$ apart. Assuming the same distribution, a $2\ GHz$ band may be divided into 80 non-overlapping channels.

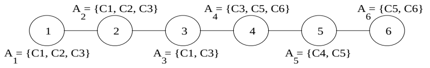
[6] Details on computation of timeslot duration were obtained from personal communication with Jeff Barton, Rockwell Collins Inc.

worst case, $d = D$. Thus, our diameter-unaware auto-configuration protocol requires $2NM$ timeslots (for the first two rounds) and $3ND$ timeslots for termination detection. Thus, the time complexity of the diameter-unaware protocol is $N\,(2M + 3D)$ timeslots. This protocol requires every node to transmit the channel set, estimate of highest UID node and the estimate of longest distance from the highest UID node to any other node in the network. Thus, it requires $O(M + logN)$ bits per message payload. Once again, for $N = 40$, $M = 80$, $D = N - 1$ and timeslot duration of $1\ ms$, the diameter-unaware protocol terminates within $12\ seconds$.

4 Discussion

In this section, we introduce some special cases and discuss how the proposed configuration protocol behaves under such circumstances.

4.1 Comments on empty global channel set



PHASE 1, ROUND 1

	Timeslot 1	Timeslot 2	Timeslot 3	Timeslot 4	Timeslot 5	Timeslot 6
Frame 1	{C1, C2, C3}	{C1, C2, C3}	{C1, C3}			
Frame 2	{C1, C2, C3}	{C1, C2, C3}				
Frame 3	{C1, C2, C3}	{C1, C2, C3}	{C1, C3}	{C3, C5, C6}		
Frame 4					{C4, C5}	
Frame 5				{C3, C5, C6}	{C4, C5}	{C5, C6}
Frame 6				{C3, C5, C6}		{C5, C6}

PHASE 1, ROUND 2

	Timeslot 1	Timeslot 2	Timeslot 3	Timeslot 4	Timeslot 5	Timeslot 6
Frame 1	{C1, C2, C3}	{C1, C3}	{C3}			
Frame 2	{C1, C2, C3}	{C1, C3}				
Frame 3	{C1, C2, C3}	{C1, C3}	{C3}	{}		
Frame 4					{C5}	
Frame 5				{}	{C5}	{C5}
Frame 6				{}		{C5}

PHASE 2, ROUND 3

Timeslot 1	Timeslot 2	Timeslot 3	Timeslot 4	Timeslot 5	Timeslot 6
{C1, C3}	{C3}	{}		{}	{C5}

PHASE 2, ROUND 4

Timeslot 1	Timeslot 2	Timeslot 3	Timeslot 4	Timeslot 5	Timeslot 6
{C3}	{}	{}		{}	{}

PHASE 2, ROUND 5

Timeslot 1	Timeslot 2	Timeslot 3	Timeslot 4	Timeslot 5	Timeslot 6
{}	{}	{}		{}	{}

Figure 4. Algorithm execution when set $\mathcal{G}$ is empty

If the global channel set, $\mathcal{G}$, is non-empty, the following two conditions have to be true at each node i:

$$(\mathcal{A}_i \cap \mathcal{A}_j) \neq \emptyset,\ \forall\ j \in NBR_i \qquad (1)$$

$$\mathcal{A}_i \cap (\cap_{j \in NBR_i} \mathcal{A}_j) \neq \emptyset \qquad (2)$$

Condition (1) formally defines NBR_i as stated in Section 3.1. It implies that node i has at least one common channel with *each neighbor*. Condition (2) implies that node i has at least one common channel with *all its neighbors*. Note that $(2) \Rightarrow (1)$, but the converse need not be true and this would lead to an empty global channel set as illustrated in Figure 4. Here, $\mathcal{A}_4 \cap \mathcal{A}_3 = \{c_3\}$ and $\mathcal{A}_4 \cap \mathcal{A}_5 = \{c_5\}$, but $\mathcal{A}_4 \cap (\mathcal{A}_3 \cap \mathcal{A}_5) = \emptyset$. As shown in Figure 4, each node correctly determines the set $\mathcal{G}$ to be empty after $D = 5$ rounds. If the set $\mathcal{G}$ is empty, nodes can always revert back to the last non-empty set $\mathcal{G}$ that was recorded at the end of a round. This can be done by having nodes maintain an additional data structure such as $\mathcal{G}_{ik}$ defined in section 3.4. In this example, $\mathcal{G}_{11} = \{c_1, c_2, c_3\}$, $\mathcal{G}_{12} = \{c_1, c_3\}$, $\mathcal{G}_{13} = \{c_3\}$ and $\mathcal{G}_{14} = \emptyset$. Thus, node 1 can deduce that channel c_3 is common to itself and all other nodes that are three hops away. Similarly, $\mathcal{G}_{61} = \{c_5, c_6\}$, $\mathcal{G}_{62} = \{c_5\}$ and $\mathcal{G}_{63} = \emptyset$. Thus, node 6 can deduce that channel c_5 is common to itself and all other nodes that are two hops away. The nodes 1, 2, 3 and 4 can form a cluster and communicate among themselves using channel c_3. Similarly, nodes 4, 5 and 6 can form another cluster and communicate using channel c_5. For inter-cluster communication, node 4 can act as a gateway node, as node 4 can communicate with nodes in either cluster.

4.2 Comments on auto-configuration overheads

Layer-2 auto-configuration operation and normal operation are repeated periodically every T time units. Alternating between layer-2 auto-configuration and normal operation requires that the normal operation be suspended for some time every T time units. This may disrupt ongoing higher layer communication (for example, TCP connections), which may be a high penalty to pay, especially for networks with low mobility. So, instead of alternating between these two modes of operation, it would be better to interleave the auto-configuration rounds between the normal operation so that configuration process is continuously ongoing. For this,

- The normal operation may also need to have a slotted and framed structure. This is to ensure that the context switching (which is more frequent here) may be done in a manner that is independent of the communication protocol used for normal operation.
- Traffic-bearing slots during normal operation may be longer than the timeslot duration of the proposed auto-configuration protocol. One could possibly pack several configuration slots into “borrowed” traffic slots.

4.3 Comments on changes to the global channel set, $\mathcal{G}$

Some of the factors that affect the integrity of the global channel set $\mathcal{G}$ computed by the proposed layer-2 auto-configuration protocol are:

- Nodes may not turn on their radios at the same time, and hence, may invoke the auto-configuration protocol at different times.
- Network topology changes. New nodes in the network could arrive or the existing nodes from the network

could depart at any time. Thus, it is possible that a single network could get partitioned and one or more such partitions could merge later to form a single network.

- Changes to channel availability set maintained by individual nodes (possibly due to arrival of the primary user of the globally common channel, c_{global}) will also trigger re-computation of the global channel set.

To address changes to the set $\mathcal{G}$ due to all the above factors, we claim that the auto-configuration protocol has to be re-invoked. Consider a newly arrived node. If the newly arrived node decreases the cardinality of the set $\mathcal{G}$ by at least one, then we term it as a *contributing node*. Suppose a run of the auto-configuration protocol resulted in the selection of c_{global} for communication among existing nodes. When a contributing node (say j) arrives, it would not be able to communicate with the existing nodes in the network if $c_{global} \notin \mathcal{A}_j$. Thus, nodes in the neighborhood of j would remain unaware of j's arrival. Due to this lack of knowledge, they would have to scan through all the M channels in $\mathcal{A}_{univ}$. Also, the total number of new nodes joining the network is not known a priori. Thus, a time-slotted mechanism appears to be needed, whereby each node transmits in its pre-assigned timeslot. A total of NM slots would be required to detect a newly arrived node and learn about its availability set. After this, $O(D)$ rounds would be required to propagate this information throughout the network. This is equivalent to re-invoking the auto-configuration protocol. Thus, in order to handle changes to globally common channel c_{global} and/or the set $\mathcal{G}$, every node in the network re-invokes the layer-2 auto-configuration protocol every T time units (see Figure 1), where T is significantly larger than the time required for the auto-configuration protocol to terminate. For example, we may choose T to be equal to $20\times$time taken by the auto-configuration protocol.

5 Conclusion

In this paper, we addressed the layer-2 auto-configuration problem in a CR network and presented a distributed algorithm for finding a global channel set wherein nodes have no prior knowledge of their neighborhood. Our algorithm consists of two phases. In the first phase, every node learns its neighborhood information and selects a preferred channel for transmission. In the second phase, nodes exchange messages on the preferred channels to compute the global channel set. We showed that all nodes in the network determine the global channel set in $O(N(M+D))$ timeslots for both diameter-aware and diameter-unaware versions of the algorithm. For reasonable network deployment scenarios, the time taken is of the order of tens of seconds. The proposed solution also provides every node the set of channels that are common to itself and all other nodes that are k-hops away. This information is particularly useful when the global channel set is empty to facilitate a communication infrastructure among clusters of nodes connected through gateways. We further believe that $O(N(M+D))$ timeslots is a fairly tight lower bound for this problem. Future work will focus on proving lower bounds and developing heuristics that might execute faster for various special scenarios.

Acknowledgments

We would like to thank Jeff Barton of Rockwell Collins Inc. for his valuable inputs and suggestions.

References

[1] J. Mitola, *Cognitive radio: An integrated agent architecture for software-defined radio*. PhD thesis, Royal Institute of Technology (KTH), 2000.

[2] R. W. Broderson et al., “Corvus: A cognitive radio approach for usage of virtual unlicensed spectrum.” White Paper, June 2004.

[3] *FCC, Spectrum Policy Task Force Report, ET Docket No. 02-155*, November 2002.

[4] Jinyang Li, Charles Blake, Douglas S.J. De Couto, Hu Imm Lee and Robert Morris, “Capacity of ad Hoc wireless networks,” in *Proc.* 7^{th} *Annual International Conference on Mobile Computing and Networking*, 2001.

[5] Matthias Grossglauser and David N. C. Tse, “Mobility increases the capacity of ad hoc wireless networks,” in *IEEE/ACM Trans. on Networking*, August 2002.

[6] A. S. Brown, “Embedding GPS receivers in software defined radio,” in *Proc. Cognitive Radios*, 2004.

[7] *Frost and Sullivan's North American GPS equipment markets, Report A601-22*, 2003.

[8] S. Corson and J. Macker, “Mobile Ad Hoc Networking (MANET): Routing Protocol Performance Issues and Evaluation Considerations,” in *RFC 2501*, January 1999.

[9] Ian F. Akyildiz, Xudong Wang and Weilin Wang, “Wireless mesh networks: a survey,” in *Elsevier Computer Networks*, January 2005.

[10] David Peleg, “Time-optimal leader election in general networks,” in *J. of Parallel and Distributed Computing*, January 1990.

[11] *IEEE Standard for Wireless LAN Medium Access Control (MAC) and Physical (PHY) Layer Specifications, ANSI/IEEE Std 802.11, 1999 Edition*.

[12] *TCI 8067 Spectrum processor data specification, Webpage - http://www.tcibr.com/PDFs/8067webs.pdf*, 2000.