Time-efficient Layer-2 Auto-configuration for Cognitive Radios‡

S. Kuppa, M. Thoppian, S. Krishnamurthy, S. Venkatesan, R. Chandrasekaran, N. Mittal and R. Prakash
School of Engineering and Computer Science
{ksrikant@, mansi@, ksrini@student., venky@, chandra@, neerajm@, ravip@}utdallas.edu
The University of Texas at Dallas, Richardson, TX-75083

Abstract—***Cognitive radios*** $(\boldsymbol{CR})$ ***have the capability 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 who their neighbors are, and also, the channels on which they can communicate among themselves. 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 $O(N \cdot (M + D))$ 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. Assuming all nodes are aware of N and M, we present both diameter-aware and diameter-unaware versions of the algorithm. For highly sparse networks*** (***like 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 $seconds$.

I. Introduction

With recent proliferation of wireless communication devices, it is being claimed that there shall be an acute shortage of bandwidth in near future. The veracity of this claim is debatable. If we consider the frequency bands already assigned for various applications (e.g. television transmission, cellular communication, etc.) we could say that the number of available bands is diminishing [5]. However, if we examine the usage of the frequency spectrum at a particular place at a given time, we find that a significant portion of the spectrum is under-utilized ([6], [14]). For example, in several cities the television channels in the VHF and UHF bands are unassigned.

Cognitive radio (CR) technology [11] allows wireless devices to dynamically adapt based on spectrum availability in their geographical region. The owner of a channel is referred to as *primary user* and all other users of the channel as *secondary users* [14]. CR technology enables secondary users to periodically scan and identify available channels[1] in the frequency spectrum. The secondary users are then able to communicate in the identified available channels without interfering with the primary user(s). This kind of communication infrastructure is of particular relevance in defense and relief operations. Since the usage of frequency spectrum varies widely from one region to another [6], communication among users (soldiers in a platoon or relief personnel in a disaster-prone area) must rely on a dynamic channel assignment scheme.

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:

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

A. Motivation

Wireless communication among neighboring nodes takes place when: (i) a node has a message to transmit, and (ii) the source node as well as all of its neighbors participating in the group communication are tuned to the common channel at the same time. In general, such a communication is needed for the exchange of: (i) vital control information such as HELLO/Keep-Alive messages, ARP broadcasts, routing table updates, etc. or (ii) application data, such as file transfers, service requests and grants, etc.

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. Our motivation behind finding the set $\mathcal{G}$ is the following:

- 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 these groups) 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.
- Tuning overheads are incurred when nodes have to switch from one frequency to another. For example, tuning speeds could be of the order of 1 ms for 10 MHz step in the frequency range 20 MHz-3 GHz [4]. By making nodes communicate on the globally common channel, say C_{global}, we can avoid such tuning overheads.
- Node mobility leads to frequent changes in network topology. For such systems, communication over C_{global} provides a simple and effective solution ([12], [13]).
- Let $\mathcal{N}$ be the set of all nodes in the network and $\mathcal{A}_i$ be the set of available channels maintained by node i. Since the nodes can be widely distributed over a geographical region, the fact that $C_{global} \in \mathcal{A}_i$ $\forall i \in \mathcal{N}$ implies that C_{global} is available over a wide region. Hence, using a globally common channel (such as C_{global}) leads to a fairly stable communication infrastructure.

‡Supported in part by an NSF CAREER grant and STTR Grants from NIST and ONR.

[1]By identifying a channel as *available*, it means that the secondary node can send and receive messages on the channel.

B. Layer-2 auto-configuration as an intersection set determination problem

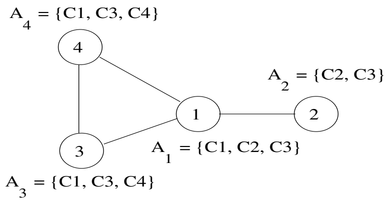


Fig. 1. A sample network showing connectivity among CR nodes and their set of available channels

Consider a simple network consisting of four CR nodes as shown in Figure 1. In the figure, the set of available channels for each node is also shown. Let $\mathcal{A}_1$, $\mathcal{A}_2$, $\mathcal{A}_3$ and $\mathcal{A}_4$ be the set of available channels at nodes 1, 2, 3 and 4, respectively. We assume that the set $\mathcal{A}_i$ maintained by node i is a *sorted* list of available channels at node i. The sorting could be done according to the channel identities. Thus, from Figure 1, we see that $\mathcal{A}_1 = \{C_1, C_2, C_3\}$, $\mathcal{A}_2 = \{C_2, C_3\}$, $\mathcal{A}_3 = \{C_1, C_3, C_4\}$ and $\mathcal{A}_4 = \{C_1, C_3, C_4\}$. The problem of determining which common channel(s) can be used by a node, say 1, to communicate with all other nodes (here nodes 2, 3 and 4), is equivalent to finding the global channel set $\mathcal{G} = \cap_{i \in \mathcal{N}} \mathcal{A}_i = \mathcal{A}_1 \cap \mathcal{A}_2 \cap \mathcal{A}_3 \cap \mathcal{A}_4$. In this example, $\mathcal{G} = \{C_3\}$. Note that it is preferable to find the intersection in a fully distributed manner without the use of a central authority[2]. This necessitates exchange of messages carrying information about the set $\mathcal{A}$ maintained at each node. Further complexity to the problem of computing the intersection set arises due to the following:

(i) Nodes do not have prior knowledge about who their neighbors are and how many nodes there are in their neighborhood.
(ii) Nodes are unaware of the existence of a common channel and its identity (if there is one) that can be used to exchange the necessary messages to compute the intersection set.
(iii) Changes in neighborhood due to node mobility can play a significant role in computing the intersection set. So, it is very important that the distributed computation terminates quickly.

C. Our contribution

As the set of available channels identified by individual nodes differs from one node to another, computing a common set of available channels (i.e. global channel set) for communication is a non-trivial task. 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.

Let N be the total number of possible nodes and M be the total number of 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, provided all nodes are aware of N and M. We present both diameter-aware and diameter-unaware versions of the protocol. The diameter-aware protocol terminates in $O\left(N \cdot (M + D)\right)$ timeslots, where D is the diameter of the network. For as many as 40 nodes[3] and 80 channels[4], a linear chain topology with $D = N - 1$ and the duration of a timeslot being $1\ ms$, the diameter-aware configuration protocol terminates within $8\ seconds$. On the other hand, the diameter-unaware version terminates within $12\ seconds$ - a highly desirable outcome. Even when the nodes are unaware of the value of D, the worst-case time complexity of the algorithm still remains $O\left(N \cdot (M + D)\right)$ timeslots. This is done by employing the terminating condition of the time-optimal leader election algorithm for general networks proposed in [9]. If the diameter D is known, then the number of bits exchanged per message is $O(M)$. Otherwise, the number of bits per message is $O(M + logN)$.

Key contributions of this paper are:

- The proposed configuration protocol helps CR nodes identify all their neighbors.
- The protocol identifies the set of channels that are common to all the nodes of the group.
- The configuration protocol runs independent of the whether the global channel set $\mathcal{G}$ is empty or not.

In summary, computing the global channel set, $\mathcal{G}$, without any prior neighborhood knowledge is a unique challenge that we address in this paper.

The rest of this paper is organized as follows. Section II presents the system model and assumptions. Section III describes the proposed distributed solution in detail followed by the proof of correctness and complexity analysis. Section IV discusses some special cases and analyzes the performance of the proposed solution. Finally, section V concludes this paper.

II. System model

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

A. Node characteristics

Every node, say i, is assigned a unique identifier, say UID_i in the range $[1 \dots N]$, 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. To simplify hardware and power requirements of the equipment carried by the soldier or relief personnel, we assume every node carries only a single transceiver. All nodes are equipped with GPS [8] 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]).

B. Medium characteristics

We assume that the communication medium is loss-free. In a lossy environment, the proposed auto-configuration protocol needs to run on top of a reliable communication mechanism which can be implemented by exchanging sequence numbers and acknowledgments. Let $\mathcal{A}_{univ} = \{C_1, C_2 \dots C_M\}$

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

[3]In an army, a platoon is a unit of thirty to forty soldiers ([3], [2]).

[4]In IEEE 802.11b devices operating in the 2.4 GHz band [1], 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.

represent the universal set of available channels that can be potentially used by all nodes for communication. Let M be the cardinality of the set $\mathcal{A}_{univ}$. We assume that M is fixed and known to all the nodes in the network a priori. The CR-enabled devices can be pre-configured with the value of M. For example, $\mathcal{A}_{univ}$ can include a set of 80 channels around the 3-5 GHz frequency range. We assume that every channel has a unique identity and the channel identities are known in advance to all nodes.

C. Network operation

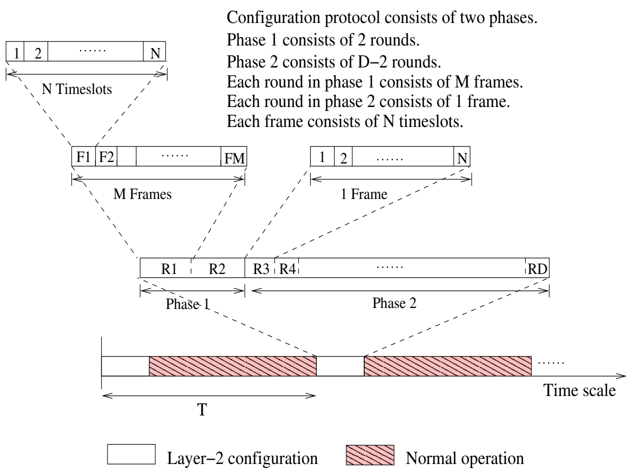


Fig. 2. Operation cycle of the network

We envision that nodes in the CR network perform one of the following two operations at any given instant of time: (i) layer-2 auto-configuration, or (ii) normal operation as shown in Figure 2. These two operations repeat periodically every T time units and the time instant at which each operation is invoked is known in advance to all the nodes in the network. This is made possible by exploiting the GPS capability [8] and by letting every node know the value of T. 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, like MANET [15] or mesh network [10]. So, they may be exchanging data related to file transfers, service requests and even control information like route discovery and maintenance, IP configuration, etc. 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 individually nodes (see Section IV-C for more details).

III. 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 2. 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 gets to transmit during the i^{th} slot in each frame (see Figure 2) 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.

A. Data structures

Following data structures are maintained at every node i:

UID_i	Identity of the node i
$\mathcal{A}_i$	Sorted list of identities of available channels at node i
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, $\mathcal{A}_i$ could be populated by every node i as a bootstrap process prior to the initiation of the configuration protocol. A node j can be a one-hop *neighbor* of node i 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 the node can be heard by all of its one-hop neighbors. If there are multiple such 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 [9]. Initially, $UID_i = i$, $NBR_i = \emptyset$, $PC_i = NULL$, $\mathcal{G} = \mathcal{A}_i$ and $r = 0$.

B. Diameter-aware auto-configuration

Let us first assume that all the nodes are aware of the diameter of the network, D. Then the following algorithm determines the global channel set. The algorithm consists of two phases (see Figure 2).

	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}

Fig. 3. Transmissions during the first M frames of the first phase

1) First phase: At the end of the first phase, each node i determines its *preferred channel* (PC_i) for transmission and sends PC_i to all of its neighboring nodes. The first phase consists of two rounds and each round consists of M frames $(F_1, F_2 \ldots F_M)$ as shown in Figure 2. 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.

In the first round, 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. 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. Consider the sample network shown in Figure 1. Here, node 2 transmits the set $\mathcal{G} = \mathcal{A}_2 = \{C_2, C_3\}$ during $((2-1) \times 4 + 2) = 6^{th}$ and $((3-1) \times 4 + 2) = 10^{th}$ timeslot. 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 nodes $1 \ldots 4$ 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 example, updated set $\mathcal{G}$ for node 1 is $\mathcal{G} \cap (\cap_{j \in NBR_1} A_j) = \{C_3\}$, where $NBR_1 = \{2, 3, 4\}$. 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 to let the neighbors of node i know about its preferred channel.

	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}

Fig. 4. Transmissions during the second M frames of the first phase

During the second round, nodes exchange the updated set $\mathcal{G}$. As before, 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 1. 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 4. Nodes can learn about the preferred channel of all its one-hop neighbors if we assume that every node i selects the smallest channel in the set it transmits during the second round. Once again, at the end of the second round, each node i updates the set $\mathcal{G}$ as described before and increments r by one. Now, 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.

2) Second phase: The second phase of the algorithm consists of $D-2$ frames with each frame divided into N timeslots (see Figure 2). 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 k^{th} frame (i.e. k^{th} round) of the second phase of the algorithm, each node i is aware of the set of channels that are common to node i and all other nodes that are within $(k+2)$ hops from node i. Thus, for $k = D-2$, every node i is aware of the global channel set and the algorithm terminates. In the sample network shown in Figure 1, $D = 2$. Hence, nodes $1 \ldots 4$ learn about the global channel set $\mathcal{G}$ at the end of the first phase and do not run the second phase of the algorithm (see Figure 4).

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 kind of communication infrastructure (see Section IV-A).
- 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).

Once the auto-configuration process is completed, any other communication mechanism (like scheduling, contention or reservation-based scheme) can be used during the normal operation of the nodes.

C. 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 the notion that the nodes have the set $\mathcal{G}$ in D rounds is still valid (see proof of correctness in Section III-D), the nodes lack sufficient local knowledge to determine that D rounds have 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 [9]. 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 $\mathcal{G}$: (i) estimate of the leader (highest UID value seen so far), say UID_x, and (ii) estimate of longest distance (in number of hops), say d, from node x 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 [9].

D. 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-hop distance from node i. By definition, the 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 on 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 $\mathcal{G}$. ■

E. Complexity analysis

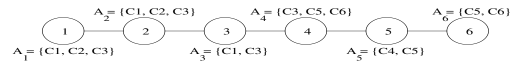
The diameter-aware auto-configuration protocol requires D rounds for completion. The first two rounds require a total of $2MN$ timeslots and remaining $(D-2)$ rounds require $(D-2) \cdot N$ timeslots. Thus, the time complexity of the protocol is $(2M + (D-2)) \cdot N$ timeslots. As each node transmits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

For $N = 40$, $M = 80$, a linear chain topology with $D = (N-1)$ and timeslot duration of $1\ ms$ [4], 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[5].

Peleg's time-optimal leader election algorithm terminates in $3d + 2$ rounds, where $d \leq D \leq 2d$. In worst case, $d = D$. Thus, our diameter-unaware auto-configuration protocol requires $2MN$ timeslots (for the first two rounds) and $3DN$ timeslots for termination detection. Thus, the time complexity of the diameter-unaware protocol is $(2M + 3D) \cdot N$ timeslots. This protocol requires every node to transmit 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 say $1\ ms$, the diameter-unaware protocol terminates within $12\ seconds$.

IV. Discussion

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



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
{ }	{ }	{ }		{ }	{ }

Fig. 5. Illustrating algorithm execution when set $\mathcal{G}$ is empty

A. Comments on empty global channel set

In the sample network shown in Figure 1, the global channel set, $\mathcal{G}$, was non-empty. In that example, the following two conditions were true at each node i:

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

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

Condition (1) formally defines NBR_i as stated in Section III-A. 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 5. 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 5, 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 III-D. 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.

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

B. Comments on auto-configuration overheads

In Section II-C, we mention that the layer-2 auto-configuration operation and normal operation are repeated periodically every T time units. Alternating between layer-2 auto-configuration and normal operation require the normal operation to be stalled 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.

C. 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. If the availability set of a 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 would be required, 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 2), where T is much larger than the time it requires for the auto-configuration protocol to terminate, say $T = 20\times$time taken by the auto-configuration protocol.

V. Conclusion

In this paper, we addressed the layer-2 auto-configuration problem in a CR network and presented 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 chosen preferred channel to compute the global channel set. We showed that all nodes in the network determine the global channel set in $O(N \cdot (M+D))$ timeslots. 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.

Acknowledgments

We would like to thank Jeff Barton of Rockwell Collins Inc. for providing valuable information about frequency spectrum usage and also answering several questions about cognitive radio technology.

References

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

[2] *MSN Encarta Dictionary, Webpage - http://encarta.msn.com/dictionary_1861738652/platoon.html*.

[3] *Wikipedia - The Free Encyclopedia. Article on Platoon, Webpage - http://en.wikipedia.org/wiki/Platoon*.

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

[5] *FCC, Spectrum Policy Task Force Report, ET Docket No. 02-135*, November 2002.

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

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

[8] A. S. Brown. Embedding GPS receivers in software defined radio. In *Proceedings of Cognitive Radios*, March 2004.

[9] David Peleg. Time-optimal leader election in general networks. In *Journal of parallel and distributed computing*, pages 96–99, January 1990.

[10] Ian F. Akyildiz, Xudong Wang and Weilin Wang. Wireless mesh networks: a survey. In *Elsevier Computer Networks*, pages 1–43, January 2005.

[11] J. Mitola. *Cognitive radio: An integrated agent architecture for software-defined radio*, 2000.

[12] Jinyang Li, Charles Blake, Douglas S.J. De Couto, Hu Imm Lee and Robert Morris. Capacity of ad Hoc wireless networks. In *Proceedings of the 7^{th} annual international conference on Mobile computing and networking*, pages 61–69, July 2001.

[13] Matthias Grossglauser and David N. C. Tse. Mobility increases the capacity of ad hoc wireless networks. In *IEEE/ACM Transactions on Networking*, pages 477–486, August 2002.

[14] R. W. Broderson et al. *Corvus: A cognitive radio approach for usage of virtual unlicensed spectrum, Webpage - http://bwrc.eecs.berkeley.edu/Research/MCMA/CR_White_paper_final1.pdf*.

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