Cognitive Radio Network Setup without a Common Control Channel

Yogesh R Kondareddy*, Prathima Agrawal* and Krishna Sivalingam†
*Electrical and Computer Engineering, Auburn University, E-mail: {kondayr, agrawpr}@auburn.edu
†Computer Science and Electrical Engineering, University of Maryland Baltimore County, Email: Krishna@cs.umbs.edu

Abstract— **The concept of Cognitive Radio Networks has introduced a new way of sharing the open spectrum flexibly and efficiently. However, there are several issues that hinder the deployment of such dynamic networks. The common control channel problem is one of such issue. Cognitive radio networks are designed by assuming the availability of a dedicated control channel. In this paper, we identify and discuss the *Network Setup Problem* as a part of the *Common Control Channel Problem*. Probabilistic and deterministic ways to start the initial communication and setup a Cognitive Radio network without the need of having a common control channel in both centralized and multi-hop scenarios are suggested. Extensive MATLAB simulations validate the effectiveness of the algorithms.**

Keywords— Cognitive Radio Network, Common Control Channel.

I. INTRODUCTION

Cognitive Radio Network (CRN) is a group of opportunistic users communicating with each other using the spectrum holes. A spectrum hole is a part of the licensed spectrum not used by the owner for a period of time [1]. The concept of cognitive radio networks was introduced to increase the spectrum usage of such underutilized bands. The opportunistic users are called secondary users (Cognitive Users) and the licensed owners of bands of spectrum are called primary users.

In a *CRN*, the Cognitive Users (CUs) communicate only in those frequencies in which the primary users (PUs) are inactive. So, the CUs should scan for the unused bands (channels) from time to time. This process is called spectrum sensing. After this stage, every CU has a list of free channels. The list of free channels may differ from one CU to another. Two CUs can communicate if there is at-least one common channel in their free channel lists. Since the unused spectrum is shared among a group of independent users, there should be a way to control and coordinate access to the spectrum. This can be achieved using a centralized control or by a cooperative distributed approach. In a centralized architecture, a single entity, called the Cognitive Base Station (CBS), controls the usage of the spectrum by CUs [2]. The Cognitive Base Station (CBS) gathers the information like the list of free channels of each node either by sensing its entire domain or by integrating the individual CUs sensed data. It maintains a database of all the collected information. When two CUs want to start a session, they request the CBS for channel allocation. The CBS looks into the list of free channels of each CU in its database and assigns a channel which is common to both of them. The database has to be updated regularly since the list of free channels will change with PUs traffic. The negotiations between the CBS and CUs are usually assumed to be carried on a dedicated control channel [3]. Intuitively, a separate dedicated channel for control signals would seem a simple solution. But a dedicated CCC has several drawbacks as discussed in [4]. Firstly, a dedicated channel for control signals is wasteful of channel resources. Secondly, a control channel would get saturated as the number of users increases similar to a multi-hop network as identified in [5]. Thirdly, an adversary can cripple the dedicated control channel by intentionally flooding the control channel. This is the Denial of Service (DoS) attack as discussed in [6]. So it was suggested in [7] to choose one of the free channels as the control channel. When PU of the chosen channel returns, a new control channel is picked. But nothing was mentioned about how the first node contacts the CBS and how would it be informed about the chosen control channel for the first time. This is called the *Network Setup Problem* in this paper.

In the second type of network architecture which is a distributed (multi-hop) scenario, the CUs have to cooperatively coordinate to coexist and access the free channels. The information sensed by a CU should be shared with other users in the network to enable certain essential tasks like route discovery in a *CRN*. Since, each CU has multiple channels to choose from, a distributed CRN is a multi-hop multi-channel network with dynamic channel set for each user. In a multi-channel network, the control information like the choice of the communicating channel is negotiated on a pre-defined common control channel. Again, dedicating a control channel for the entire network is not a good idea for the above mentioned reasons and choosing a free channel as the control channel might not work because the chosen channel might not be free with all the users. Most of the recent papers proposed MAC protocols which avoid a common control channel but none of them focused on how to setup the initial network (*Network Setup Problem*) i.e. how would a CU contact another CU before it can start anything?

Addressing and solving the *Network setup problem* is the motive of this paper. A deterministic and probabilistic way of scanning the channels by a CU to connect to the CBS is proposed. The proposed mechanisms are also extended to a multi-hop scenario in which a CU searches for another CU.

The rest of the paper is organized as follows. Section II discusses the common control channel problem in detail. The existing work is shown in section III. Section IV discusses the proposed solutions. Section V presents simulation results and section VI concludes the paper.

978-1-4244-2677-5/08/$25.00 ©2008 IEEE

II. NETWORK SETUP PROBLEM

In this section the *Network Setup Problem* (NSP) is described.

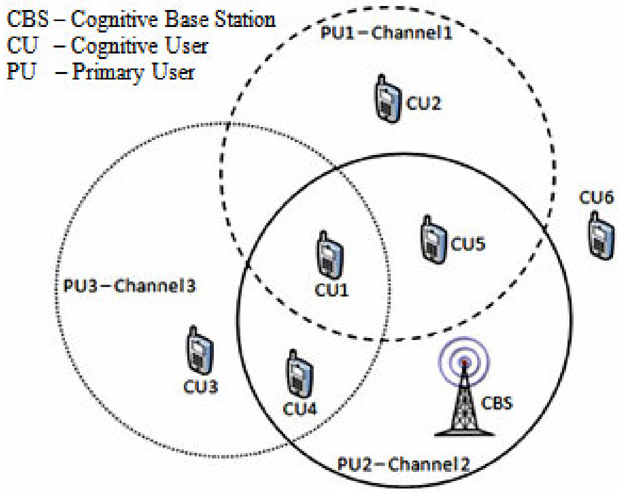

Figure 1. A group of CUs among three primary users.

Fig. 1 illustrates a centralized architecture in which there are three PUs each one occupying a channel. The circles represent the interference range of each PU. There are six CUs and a CBS. Suppose that there are totally three channels available. A channel is said to be free for a CU to communicate in, if the PU of that channel is inactive in its premises or if it is not in the interference range of that PU. The set of such free channels of a CU is referred to as *Free Channel Set* (FCS). If all the PUs are active, the FCS of each user will look like in Fig 2. It can be observed that since CU6 is not in the interference range of any PU it has all the three channels free. It is possible that each user has a choice of more than one channel as it is in the case of CU2, CU3 and CU6.

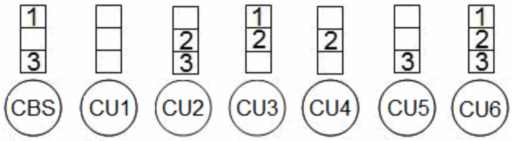

Figure 2. Free Channel Sets of the CUs and CBS

In the *Initial State of the Network*:

- A CU is a totally independent node.
- No CU has any information about its neighbors or the CBS.
- CBS also does not have any information regarding the CUs around it.

To setup a Cognitive Radio Network, the 6 users have to contact the CBS and notify their presence. A CU can communicate with CBS only if they both transmit and listen in the same channel. Since they both have a set of channels, they can possibly communicate only if they have at-least one channel common in their FCS. For example, CU2 in Fig 2 can communicate with CBS since they have channel 3 in common. But, since neither of the CUs has any information about the free channels of the CBS and there is no dedicated control channel, there should be a protocol for the nodes to strategically search for the CBS to setup the network. In a practical scenario, there can be many more channels in the FCS of each user making the situation more complicated. This is called the *Network Setup Problem (NSP)*. NSP in a centralized scenario represents the following questions:

- Who should beacon in the search process: the CBS or the CU?
- In which channel should the CU or the CBS beacon?
- How much time should the CUs search for?

A similar problem arises in the case of a multi-hop scenario in which there is no base station. Fig 1 depicts such a scenario if the CBS is removed. In this case the CUs have to identify their neighbors to form a Multi-hop Cognitive Radio Network (MHCRN) and the same questions apply for every pair of CUs.

NSP basically occurs due to the absence of a Common Control Channel (CCC). So NSP is a part of a bigger CCC problem which is explained below.

Common Control Channel Problem

As discussed earlier, two users in a CRN are connected if they have a common channel for communication. It is possible that each user has a choice of more than one channel. In that case, the sender and the receiver need to agree upon a common communicating channel which is available to both. The initial handshake signals to negotiate the choice of a common channel are called *control signals.* But such negotiations require communication over a common signaling channel. This is called the *common control channel problem (CCCP).* This problem is illustrated in more detail using Fig. 3.

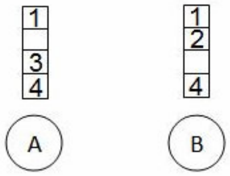

Figure 3. Two cognitive nodes with a set of free channels.

Fig 3 shows a more generalized scenario of two nodes which represent a pair of CUs in a multi-hop CRN or a CU and a CBS in the case of a centralized CRN. Node A has channels 1, 3 and 4 available and node B has 1, 2 and 4 available. These available channels form the FCS of the respective pair of nodes. Suppose that the network is in its *initial state* i.e. A is unaware of B's channel set and vice versa. It can be seen from the figure that channels 1 and 4 are common among the two nodes. When node A wants to transmit to node B, A and B should:

a) Identify its neighbors and negotiate their *channel sets – Network setup problem.*

b) Exchange 'Request to Send' (RTS) and 'Clear to Send' (CTS) messages to reserve a channel for communication in a manner similar to IEEE 802.11 Distributed Coordination function (DCF) – *Design a MAC protocol* without a CCC.

These control messages in turn have to be negotiated via a channel. So a channel is required to choose a channel! The later part has been addressed in several papers [8-10]. [8] and [9] assume a CCC which is one among the available channels. [10] proposes a method in which a group of users which are close together form a sub-ad hoc network and select a channel for communicating control information. The former part of CCCP is what we differentiated as *Network Setup Problem (NSP)* and are focusing on in this paper. In the next section three solutions to the NSP are proposed.

III. NETWORK SETUP MECHANISMS

In this section, three different protocols to address the NSP will be explained. The protocols define a scan and search procedure for the CBS and CUs so that they can initiate a CRN. Before that it is important to discuss the capabilities of a CU and a CBS and some of the terms used in the coming discussion.

A *Cognitive User* is capable of shifting his frequency of operation. A simple CU is equipped with one Cognitive Radio (CR) and it can scan a channel at a maximum rate of Rcu channels per second.

A *Cognitive Base Station* is at-least equipped with two CRs. It is an added advantage if it is assumed that a CBS is capable of scanning the channels faster than a CU at a rate of Rcbs. But, the lack of this assumption does not affect the working of the protocol in anyway.

Primary User's Traffic Rate (PUTR): It is defined as the average rate at which the primary user changes his state (active/inactive). This is an important factor because the channel availability is directly related to PUTR. Higher PUTR implies that channel availability each CU fluctuates at a higher rate.

Number of Channels (N): The total spectrum in which the Cognitive Users can operate is divided into a fixed number of channels; *N*. It should be noted that *N* can be possibly very large varying from tens to thousands of channels. Though the proposed protocols do not depend on the value of *N,* for the convenience of pictorial representation *N* will be chosen very small.

All the proposed protocols are initially discussed for those architectures (centralized/distributed) which the protocol best suits to.

A. Exhaustive Protocol

This protocol implements exhaustive search and will be referred to as EX Mechanism. The channels are searched from lower to higher frequencies by both the CBS and CUs. CBS is assigned the task of sending beacons because of its superior infrastructure in terms of hardware and energy. It is also assumed that PUs traffic does not vary in one search cycle.

In a *Centralized Architecture*, CBS maintains a timer which counts to T_S seconds. It initially starts its search from the channel with lowest frequency and starts its timer T_S. It shifts to the next channel when the timer expires. In each time slot, the channel is scanned for the presence of a PU. If the channel is not free, then CBS will immediately shift to the next channel and resets the timer. If the channel is free, a beacon is sent indicating its presence in that channel. It will wait for a response for the rest of the time slot till the T_S timer expires and then tune to next channel starting its timer again. If in the mean time a response is received from a CU, a different Cognitive Radio is assigned the task of carrying on the negotiations with the CU and CBS continuous its search for other potential users. After all the channels are searched, it will restart from the lowest frequency again. If all the *N* channels were free, CBS would take $N\times T_S$ seconds to complete a cycle of searching all the channels.

Every CU maintains a *Wait* timer, T_W which is set to $N\times T_S$. It initially starts from the channel with lowest frequency and scans for the availability. If the channel is not free, it shifts to the next channel and resets its timer. If the channel is free, it waits for a beacon from the CBS till the timer T_W expires. Since, CBS will search all the channels at-least once in T_W seconds, the CU can be sure of receiving a beacon if the channel it was listening to is free with CBS. The total process is illustrated using Fig 4. Each block in the figure represents a channel. So, there are totally 10 channels with each of CBS and CU. A shaded block means that PU is active in that channel. CBS starts its search from channel by setting its timer T_S. Since the first channel is not available it will reset its timer and shift to the second channel. As the CBS scans and sees that channel 2 available, it beacons in this channel and waits till the timer expires for a response. Similarly a CU starts from the first channel and waits for T_W seconds and will not receive any beacon because CBS does not beacon in that channel. After the T_W timer expires, CU shifts to the next channel where it will receive a beacon from the CBS and respond to the beacon and request a connection. It should be observed that a CU will receive a beacon in a maximum time of $N^2\times T_S$ seconds if at-least one channel is free with both CBS and CU.

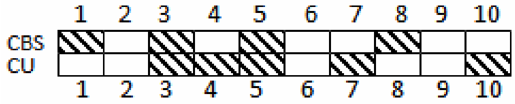

Figure 4. Channel availability at the CBS and CU.

In a *Multi-Hop Architecture* there is no CBS and CUs have to search for each-other. So, the protocol is modified such that a CU will wait for the beacons if the network is already initiated, otherwise, it will initiate the network by sending beacons for the following CUs. To know whether the network is initiated or not, a CU has to make sure it is not the first user in the network. So it will wait for a beacon in every channel for T_W seconds. The cycle completes in $N^2\times T_S$ seconds. If it receives a beacon, it acknowledges it and shares the channel information. If it does not receive a beacon, then it is considered to be the first CU in the network and it starts sending beacons as the CBS did in the centralized architecture.

This protocol guarantees that a CU will be able to contact the CBS within a worst case of $N^2\times T_S$ seconds if a common free channel exists between the CBS and CU. So, it is a deterministic solution and in will be seen that most of the times, the search time is much lesser than the worst case scenario. In the next subsection, a probabilistic solution is proposed.

B. Random Protocol

Unlike the deterministic approach (EX-protocol), this is probabilistic approach. This protocol is useful in situations where the number of channels N is not known precisely.

In a *Centralized scenario*, CBS does the same basic tasks which were explained in EX-protocol except that the channels are chosen randomly. The first channel is chosen randomly and is checked for availability. If the channel is not free, another channel is chosen randomly and the timer T_S is reset. If it is available, a beacon is sent and the CBS waits for a response till the timer expires. CBS will keep choosing one of the C channels randomly. For example in Fig 4, if the CU happened to choose channel 8 as the first channel and wait for a beacon in that channel continuously, it would never get a beacon since that channel is not available for CBS unless the PU in that channel stops using it. So, shifting the channel periodically is necessary.

There is little difference in the CU's tasks from the tasks of a CU in EX-protocol. The first difference is that the channels are chosen randomly. There is one more variable which the CU maintains which is the number of *Wait Slots*, W_S. The *wait* timer, T_W is now set to $W_S \times T_S$ instead of $N \times T_S$ as in the case of EX-protocol. If the CU does not receive a beacon, it will choose a different channel. The shifting of channel is necessary because, if the CU waits in the same channel continuously waiting for a beacon and suppose the chosen channel is not available for CBS, then the CU would not receive a beacon at all. If The CU receives a beacon it responds to it and negotiates the channel information.

The value of W_S is chosen strategically depending on range of channels which the CU is capable of scanning, C. In $W_S \times T_S$ seconds, CBS would have searched at-least W_S channels. If all the channels were free, the probability that the CU will receive a beacon in one of its wait time T_W is:

$$Pr = 1-(1-\frac{1}{C})^{Tw}$$

The actual probability depends on the probability of a channel being free with both CBS and CU.

In a *Multi-hop scenario*, CUs will exactly follow the same rules as they did in a centralized scenario and additionally send beacons for every T_S seconds. Moreover, the *Wait Slots*, W_S is randomly chosen from a predefined range of numbers. This makes each CU search the channels at different rate which emulates the centralized scenario. Unlike in EX-protocol, a CU cannot wait for a specified period of time for a beacon and be sure that it is the first user if it did not receive a beacon. This is because of the randomness due to which it is not possible to define a definite maximum time period during which CBS would have scanned all the channels at-least once. So, when a CU wants to initiate or join a network, it chooses a random W_S and a random channel in which it beacons for every T_S seconds. Upon the successful reception of a beacon by any CU, it acknowledges and exchanges the channel information with the sender.

C. Sequential Protocol

This protocol is a modified version of EX-protocol to make it more suitable for a multi-hop network. So, the multi-hop scenario is explained first and then extended to the centralized scenario. In this protocol, the total number of channels N, is assumed to be known.

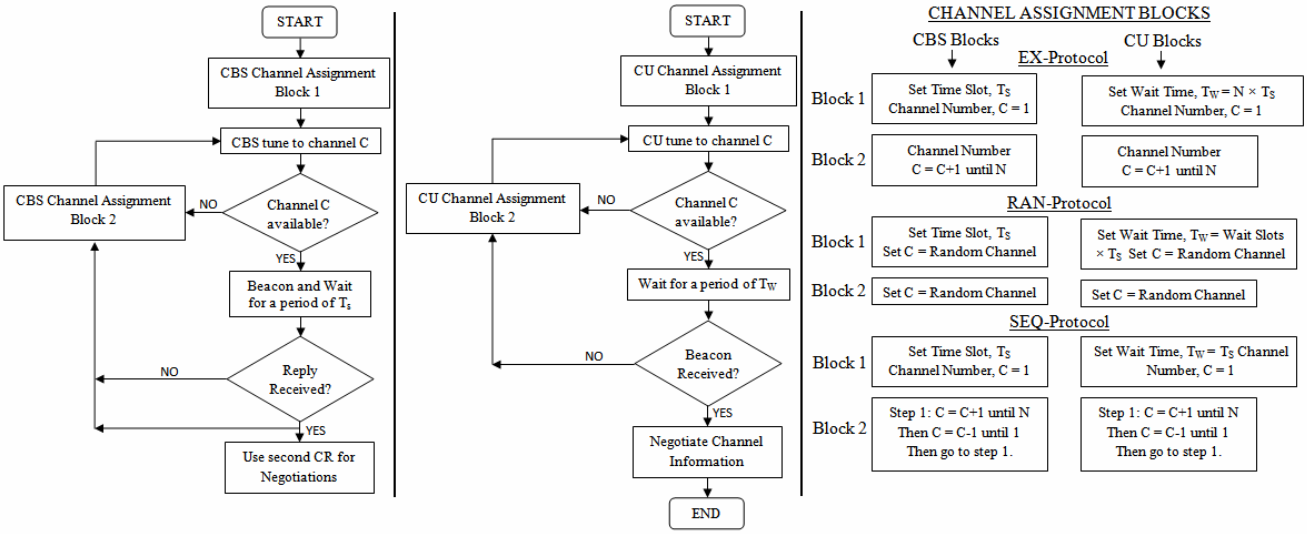

Figure 5. Algorithms for the proposed protocols.

Multi-hop Scenario: The time slot, T_S is chosen similar to other protocols. The CUs start from a random channel. The next channel is chosen in the increasing order of frequencies. After the last channel is reached, the next channel is chosen in decreasing order of frequencies and not from the lowest again. If the chosen channel is not available the Cu shifts to the next channel. If the channel is available, it stays for a period of T_S that channel and sends a beacon during that period. If it receives an acknowledgement, its neighboring CU has received its beacon and they exchange the control information. The same thing happens if the CU receives a beacon. Due to the symmetry in the CU tasks, this protocol is more suitable for a multi-hop network.

In a *Centralized Scenario*, the only difference is that the CU does not beacon instead just listens in its chosen channel for a beacon.

Fig. 5 shows the generalized flow charts of the tasks performed by the CBS and CU. The channel assignment blocks in these flow charts should be replaced by the blocks shown on the right side which differentiate the protocols from each other. In the following section, the working of the proposed protocols is studied using simulations.

IV. SIMULATION STUDY

In this section the protocols are simulated and their performance is studied and the three protocols are compared with each other. The simulation setup used in all these experiments is shown below.

Simulation setup

MATLAB has been used for all simulations. The number of channels, N is varied from 10 to 1000 in each simulation. Each point on the graphs is an average of 500 simulations. The *Primary User's* traffic is compensated by setting a probability of channel availability. The probabilities are set such that they represent realistic scenarios. In [10] it has been observed a CU's neighbors will have the same channel states with high probability i.e., if a CU has a channel available, it's highly probable that its neighbor has the same channel available. So, the probability of channel availabilities is chosen as shown below:

- The probability of a channel having the same status (available/ not available) with both CBS and CU is 80%.
- The probability of a channel with the same status at CBS and CU being available is 50%.
- The probability of a channel to be available at one of CBS or CU is 20%.
- The probability of a channel with the different status at CBS and CU being available is 50%.

Other specifications used in the simulations are:

- The CU arrival time is randomly chosen.
- The value of *Time Slot*, $T_S = 1$ sec.
- The beacon time duration is chosen as, $T_b = 100$ msec.
- The time taken to shift to a channel and check its availability = 100 msec.

A. *Search Time*

During the network setup, *Setup Time* is the crucial factor. The total network setup time is directly proportional to the time each CU takes to find the CBS and connect to it. The time taken for a CU to receive a beacon from the CBS is measured i.e., the time taken before a *Cognitive User* connects to the *Cognitive Base Station* is referred to as *Search Time* in the rest of the discussion. So, in this section the *Search Time* of the three protocols is compared.

Fig 6 shows the average *Search Time* of the protocols as the number of channels is varied. Each point is an average of 500 simulations. It is observed that the EX-protocol takes the least *Search Time* and the SEQ-protocol takes the highest *Search Time.* So, EX-protocol is efficient when the total number of channels, C is known. If C, is unknown then RAN protocol is a good choice compared to SEQ-protocol.

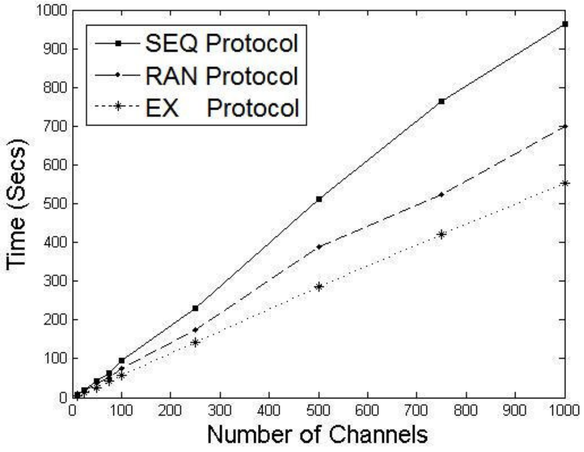

Figure 6. Plot showing the Search Time of the three protocols as the number of channels are varied.

We have seen earlier that RAN-protocol uses a variable quantity called Wait Slots, $W_{S.}$ Fig 7 shows the effect of number of Wait slots W_S on the average *Search Time* when the RAN-protocol is used. It can be observed that the *Search Time* is more for higher values as well as lower values of W_S. So, there is an optimum value of W_S, for achieving minimum search time. In this case it is 5.

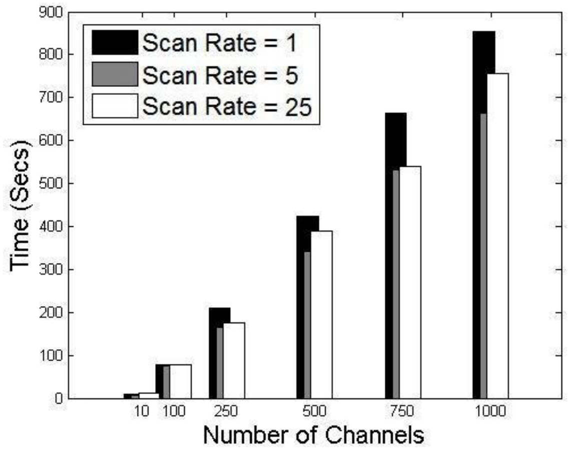

Figure 7. A plot showing the affect of the number of the Wait Slots in RAN-protocol as the number of channels are varied .

B. *Number of Scans*

Number of scans is defined as the number of channels the user has search or shifted to in the before search time. The more the *Number of Scans* is, higher is the energy usage and the time taken. A CBS is supposed to be more robust than a CU which has a limited energy to spend. So, it is always better to have less *Number of Scans* for a CU compared to CBS.

Table 1 shows the average *Number of Scans* of a CBS and a CU for each protocol. The EX-protocol offers the least *Number of Scans* for a CU which is suitable in cases of energy crisis. It is observed that RAN-protocol offers lesser *Number of Scans* for a CBS. In SEQ-protocol, the *Number of Scans* is almost equal for CBS and CU due to the symmetry in the protocol. It can be concluded from the table that since EX-protocol offers least number of scans for a CU, it is more suitable for a centralized scenario and RAN-protocol is suitable for a multi-hop scenario.

Table 1. Number of scans in the search time.

Number of Channels	SEQ		RAN		EX	
	CBS	CU	CBS	CU	CBS	CU
10	13	14	15	6	14	3
50	74	82	65	20	54	3
100	162	171	132	38	100	3
250	441	425	316	88	223	3
500	866	903	655	190	517	3
750	1336	1365	944	271	714	3
1000	1708	1786	1331	375	999	3

C. *Failures*

If a CU is not able to connect to a CBS in a specified period of time then it is considered as a *Failure*. It was observed that in case of SEQ-protocol, there were some failures. The time period after which the search is considered a failure is 5000 seconds in the following simulations.

Fig 8 shows the number of *Failures* for 500 simulations. It can be observed that the number of failures decrease as the number of channels is increased. This protocol though does not perform well is suitable for a multi-hop scenario because of its symmetry.

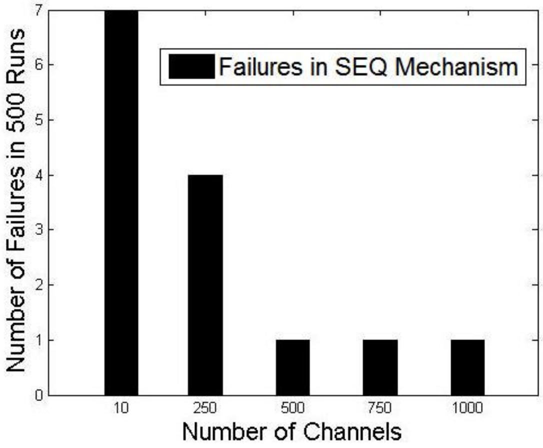

Figure 8. Number of failures in SEQ-protocol for 500 random simulations.

V. CONCLUSION

In this paper we have identified the *Network Setup Problem* (*NSP*) which is a part of the *Common Control Channel Problem*. The *NSP* is discussed in detail and three protocols are proposed to set up a centralized *Cognitive Radio Network* (*CRN*) or a Multi-hop *CRN* (*MHCRN*). The protocols were verified and the results are analyzed using MATLAB simulations. It was observed that the EX-protocol is very efficient in a centralized scenario; SEQ-protocol is more suitable for a multi-hop network. But, EX-protocol cannot be used when the exact number of possible channels is not known. RAN-protocol, is useful in such scenarios and moreover it has optimum performance compared to SEQ and EX-protocols.

VI. ACKNOWLEDGMENT

Part of the research was supported by a grant from Air Force Office of Scientific Research (AFOSR) grant No. FA9550-06-1-0103.

REFERENCES:

[1] Q. Zhao, B.M. Sadler, “A Survey of Dynamic Spectrum Access: Signal Processing, Networking, and Regulatory Policy,” IEEE Signal Processing Magazine, May, 2007.

[2] Chunsheng Xin, Bo Xie, Chien-Chung Shen, “A novel layered graph model for topology formation and routing in dynamic spectrum access networks”, Proc. IEEE DySPAN 2005, November 2005, pp. 308-317.

[3] Akyildiz, Ian F. ,Lee, Won-Yeol, Vuran, Mehmet C, Mohanty, Shantidev, “NeXt generation/dynamic spectrum access/cognitive radio wireless networks: A survey”, Computer Networks, v 50, n 13, Sep 15, 2006, p 2127-2159.

[4] J. Zhao, H. Zheng, G.-H. Yang, “Distributed coordination in dynamic spectrum allocation networks”, in: Proc. IEEE DySPAN 2005, pp. 259-268, November 2005.

[5] J. So, N. Vaidya; “Multi-Channel MAC for Ad Hoc Networks: Handling Multi-Channel Hidden Terminals Using A Single Transceiver'”, Proc. ACM MobiHoc 2004.

[6] K. Bian and J.-M. Park, "MAC-layer misbehaviors in multi-hop cognitive radio networks," 2006 US - Korea Conference on Science, Technology, and Entrepreneurship (UKC2006), Aug. 2006.

[7] J. Zhao, H. Zheng, G.-H. Yang, “Distributed coordination in dynamic spectrum allocation networks”, in: Proc. IEEE DySPAN 2005, pp. 259-268, November 2005.

[8] S. Krishnamurthy, M. Thoppian, S. Venkatesan, and R. Prakash, “Control Channel-based MAC-layer Configuration, Routing and Situation Awareness for Cognitive Radio Networks”, in: MILCOM 2005, Atlantic City, NJ, October 2005.

[9] C. M. Cordeiro and K. Challapali, "C-MAC: A Cognitive MAC Protocol for Multi-Channel Wireless Networks", in IEEE Symposium on New Frontiers in Dynamic Spectrum Access Networks, April 2007.

[10] Q. Zhao, L. Tong, A. Swami, and Y. Chen, “Decentralized Cognitive MAC for Opportunistic Spectrum Access in Ad Hoc Networks: A POMDP Framework”, submitted to IEEE Journal on Selected Areas in Communications, February, 2006.