This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the ICC 2008 proceedings.

Synchronized MAC Protocol For Multi-hop Cognitive Radio Networks

Yogesh R Kondareddy and Prathima Agrawal
Department of Electrical and Computer Engineering
Auburn University, Auburn, AL 36849
E-mail: {kondayr, agrawpr}@auburn.edu

***Abstract*— Cognitive networks enable efficient sharing of the radio spectrum. Multi-hop cognitive network is a cooperative network in which cognitive users take help of their neighbors to forward data to the destination. Control signals used to enable cooperation communicate through a common control channel (CCC). Such usage introduces conditions like channel saturation which degrades the overall performance of the network. Thus, exchanging control information is a major challenge in cognitive radio networks. This paper proposes an alternative MAC protocol for multi-hop cognitive radio networks in which the use of a CCC is avoided. The scheme is applicable in heterogeneous environments where channels have different bandwidths and frequencies of operation. It inherently provides a solution to issues like CCC saturation problem, Denial of Service attacks (DoS) and multi-channel hidden problem. The proposed protocol is shown to provide better connectivity and higher throughput than a CCC based protocol, especially when the network is congested.**

Keywords- Cognitive Network, medium access control, control channel

I. INTRODUCTION:

A cognitive network is an opportunistic network. *Spectrum opportunity* deals with the usage of an available (free) channel that is a part of the spectrum which is not currently used by primary users [1]. The licensed owner of a frequency band is called a *primary user* and the one who utilizes spectrum opportunities for communication is called a *secondary user*. When the receiver is not in the transmitting range of the sender, data is forwarded through several hops forming a Multi-Hop Cognitive Radio Network (MHCRN). But unlike in a normal multi-hop network in which all users operate in the same channel, users in a MHCRN use different frequencies depending on spectrum availability. As a result, two users are connected depending on whether they have a common frequency band for operation.

A MHCRN is, in many ways, similar to a multi-channel network. In both networks, each user has a set of channels available for communication. When two users want to communicate, they negotiate via a common control channel (CCC) to select a communicating channel. Two major differences in these two network environments are: a) the number of channels available at each node is fixed in a multi-channel network whereas it is a variable in a MHCRN. It is possible that a user has no available channel at all due to the complete occupancy of the spectrum by primary users. b) In general, the channels in a multi-channel environment have equal transmission ranges and bandwidths unlike in a MHCRN in which the environment is heterogeneous. Thus, a MHCRN is a combination of a multi-hop and a multi-channel network. The protocols used in multi-channel networks cannot be applied to a MHCRN due to the above mentioned differences in the two networks. However, the issues and challenges related to these networks apply to a MHCRN as well. For example, CCC and multi-channel hidden terminal problems [2] which are related to a multi-channel network are common to a MHCRN.

In this paper, a new MAC protocol for MHCRNs is proposed which avoids the need of a dedicated CCC and solves the multi-channel hidden terminal problem [2]. The main idea is to divide total time into fixed-time intervals, each representing one of the available channels. At the beginning of each time slot, all nodes in the network listen to a channel which the time slot represents for exchanging control signals. Thus, all nodes in the network are synchronized.

The rest of the paper is organized as follows: Section II identifies some of the major issues such as CCC saturation and multi channel hidden terminal problem in a MHCRN. Section III reviews existing protocols and discusses the drawbacks. Section IV describes the proposed MAC layer protocol. Section V presents the simulation results and Section VI concludes the paper.

II. ISSUES IN MULTI-HOP COGNITIVE NETWORKS

In this section, we describe the common control channel (CCC) problem and briefly explain the multi-channel hidden terminal problem [2] in the context of a MHCRN.

A. *The Common Control Channel problem*

As discussed earlier, two users in a MHCRN 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*. This problem is illustrated in more detail using Fig. 1. A cognitive user is referred to as a 'node' in the rest of the paper.

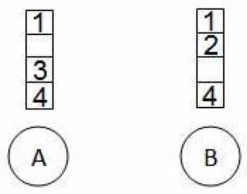

Figure 1. Two cognitive nodes with a set of free channels.

Suppose Node A has channels 1, 3 and 4 available and node B has 1, 2 and 4 available as shown in Fig. 1. These available channels form the *channel set* of the respective pair of nodes. Also, suppose that 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) negotiate their *channel sets*, 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). These control messages in turn have to be negotiated via a channel. Intuitively, a separate dedicated channel for control signals would seem a simple solution. But a dedicated CCC has several drawbacks as discussed in [3]. 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 [2]. 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 [4].

Another solution to exchange control messages is to choose a channel among the available channels as the control channel. For example, in Fig. 1, channel 1 or 4 can be chosen as the control channel. When the primary user of that channel returns, a new channel which is available to all users is chosen. This approach is not feasible because the probability that a particular channel is available to all users is small. Moreover, the available channels may vary in the frequency of operation, bandwidth and transmitting range. Due to such heterogeneity in the transmission range, the connectivity and scalability of the network varies with the control channel because a channel with shorter transmission range may not cover all the areas covered by a channel with a longer transmission range. Thus, there is a need for a better protocol which avoids the use of a CCC and which takes heterogeneity into account while choosing a channel for communication.

B. Multi-channel hidden terminal problem in MHCRNs

The multi-channel hidden terminal problem was identified in [2] for multi-channel networks. The same problem is extended to a cognitive network environment in [5]. It is briefly explained below.

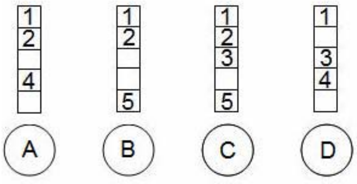

Figure 2. Four nodes with respective available channels

Fig. 2 represents 4 nodes with their respective channel sets. Suppose that only adjacent nodes are in transmitting range. Since channel 1 is available to all nodes, suppose that channel 1 is chosen as the control channel and that node C and D are already communicating using channel 3. When node A wants to transmit a packet to node B in channel 2, it sends an RTS to B on the control channel (channel 1 in this case). B sends a CTS proposing channel 2 for data communication. Node A sends a confirmation message to B and to its neighbors that channel 2 is reserved for data communication. But since C was communicating in channel 3, it did not receive the CTS from B. So C assumes that channel 2 is free and might initiate a communication with node B in channel 2 resulting in a collision. This is called the *multi-channel hidden terminal problem*. In [5], this problem is addressed by allocating special time slots. In these time slots the communicating pair of nodes gets updated from its neighboring nodes about any potential hidden terminals in their vicinity. Though, the problem is solved successfully using this method, a CCC is still used for control signal exchange.

To avoid the above mentioned disadvantages, a MAC protocol which does not need a pre-allotted control channel and which solves the multi-channel hidden terminal problem, is proposed in this paper.

III. RELATED WORK

Considerable number of MAC-protocols have been proposed for cognitive networks previously [7-15]. But most of these assume a CCC which limits the robustness of the network in many ways as pointed out earlier. [7][8] assume a CCC which is one among the available channels. Similarly [9][10][11] also assume a CCC for the purpose of exchanging control signals. There are a few proposals which solve the problem of CCC partially. Since the probability that a CCC is available at every node is small, [12] 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. If the primary user of the channel returns, a different channel which is available to everyone in the sub-group is chosen. It is assumed that one of the members of the group has the capability to connect to the neighboring groups. Though it is an indirect solution to the problem of availability of a CCC, it does not completely eliminate the dependency on a CCC. There is still a possibility that a user can pose a DoS [13] attack. Moreover, the group head that is responsible for sharing of the information between two groups gets a chance to act selfishly on its data. [5] solves multi-channel hidden node problem but still uses a CCC for control signal exchanges.

Though the MAC layer misbehaviors have been pointed out in [4], there is no existing protocol which considers the control channel saturation problem, multi-channel hidden terminal problem and heterogeneity of channels simultaneously. Compared to the above proposals, our protocol does not require a separate control channel for the purpose of control signal exchange. Instead it requires synchronization among all nodes. Though it is an extra requirement on the network, it will be shown in later sections that it achieves better throughput and network connectivity than maintaining a separate control channel.

IV. PROPOSED SYNCHRONIZED MAC (SYN-MAC) PROTOCOL

In this section, the proposed scheme is presented. Before that, the assumptions are summarized below.

- Every node is assumed to be equipped with two radios. One of the two radios is used for just listening (*listening radio*) to the control signals and the other for both receiving and transmitting data (*data radio*).
- The maximum number of channels at each node is *N,* but the channels available at each node may vary with the primary user's traffic.

The proposed protocol will be referred to as Synchronized MAC (SYN-MAC) from now on.

A. Network initialization state

Initially, when there are no cognitive users (*nodes*) to form a network or when the new user wants to form a sub-group independent of the existing users, the network is said to be in the *initialization state*.

In the *network initialization state,* the first node divides time into N number of equal time slots of fixed duration 'T_c', since there are N possible channels. Each time slot is dedicated to one channel for control signal exchange. The node then beacons in all its available channels at the beginning of the corresponding time slots. The following nodes choose one of the channels and listen for beacon messages to synchronize their *listening radios.* Since the first node broadcasts in all its available channels, the following nodes can choose any channel and be sure to receive a beacon message within '$N \times T_c$' seconds. After it receives a message, the nodes exchange information about their channel sets. If it did not receive a beacon, then it is considered to be the first node.

At the end of the *network initialization state,* all nodes are synchronized and every node has the information about its neighbors and their respective channel sets. Nodes being synchronized mean that at the beginning of every slot, the *listening radio* of every node tunes to the respective channel which the slot represents and listens in that channel. It is analogous to passing a token among the channels and every node is listening to that particular channel at a given time. The continuous scanning (listening) of channels is necessary for three reasons which are: a) To keep track of primary user's presence, b) For exchanging control signals, c) To avoid multi-channel hidden node problem.

B. Exchange of control signals and data

When a node wants to start a communication, it should exchange the required control signals. To exchange the control signals it chooses one of the channels common between itself and its neighbor. It then waits for the time slot which represents the chosen channel. Since all nodes will be listening to that channel in that slot duration, it will start exchanging its control signals with its neighbor.

Unlike the exchange of control signals which need to be exchanged only at the beginning of specified time slots, the data is exchanged after exchanging the control signals. So exchange of data occurs in an un-synchronized fashion using the second radio (*data radio*). Control signals or information is exchanged among the nodes whenever an event occurs. These events are called *information events.* There are 4 *information events (IEs)* which are:

IE-1: When a new node enters the network, it should notify its arrival to its neighboring nodes.

IE-2: When the available channel list at a node changes due to the primary user traffic, the node's neighbors have to be updated about its new channel list.

IE-3: When a node starts, stops or changes its channel of communication, the information is forwarded to its neighbors to enable them to know whether the data packets can be forwarded through the communicating node.

IE4: When a node wants to communicate with its neighbor, it sends a set of control signals to inform its intent to start a communication in a particular channel. This event is followed by an acknowledgement by the neighbor to convey its acceptance/denial. On acceptance, data transfer takes place on the negotiated channel without any delay.

The complete process of starting a communication is illustrated with an example shown in Fig. 3.

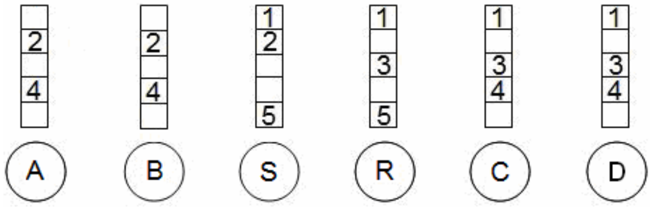

Figure 3. Six cognitive nodes with a set of free channels at each node.

Consider a network of 6 nodes as shown in Fig. 3. There are a total of 5 possible channels among them. The array of blocks above each node represents the available channels at that node. Since the total number of channels is 5, time is divided into 5 slots and the *listening radio* of the nodes keep listening to successive channels at the beginning of the respective slots. Now, suppose Sender (S) wants to start a communication with Receiver (R). Since the nodes know the available channel sets of their neighbors, node S sees that it has channels 1 and 5 in common with node R. It chooses one of these channels for communicating with node R. If channel 1 is chosen, then node S waits for a random back off time (shown using solid shading in Fig. 4) and starts its negotiations, similar to IEEE 802.11 DCF. Once the negotiation is successful the data transfer takes place in channel 1 immediately. This is shown in Fig. 4. Now, suppose that node B observes that primary user of channel 4 has returned. So, it generates an *IE2* which contains its new channel set. Since node B knows that it can reach its neighbors through channel 2, it waits for the time slot which represents channel 2, backs off for a random time and then transmits its information (*IE2*). Nodes S and A, on receiving this information learn that node B will not be available on channel 4. Similarly, when node C sees that primary user of channel 4 has returned, it waits till the slot representing channel 3 and then broadcasts the information to nodes R and D. All activities described above are summarized in Fig. 4.

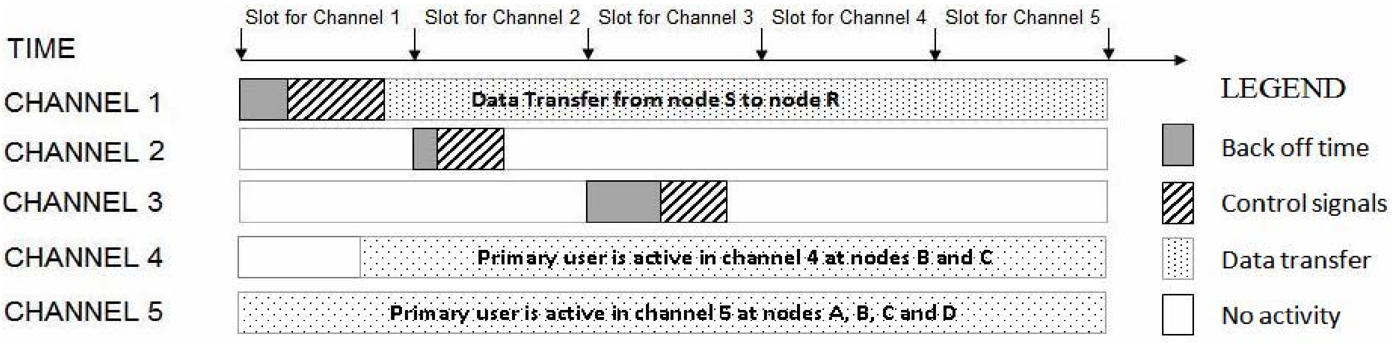

Figure 4. Five channels with the control and data transfer events in their respective time slots.

With the above explanation, we will demonstrate the advantages of the protocol over CCC based protocols now. We will also show how different issues discussed in section II are solved using our protocol.

Firstly, it should be clear by now that there is no dedicated CCC for control signals, so there would be no concept of control channel saturation. It will be shown in the next section that there is a significant benefit in terms of throughput from not having a CCC in a MHCRN. Also, in Fig. 3, it is seen that there is not even one channel common at all of the six nodes and hence control signal exchange could not have been possible using the CCC based protocols. But in SYN-MAC communication could be established as discussed.

Secondly, observe that when node S wanted to transmit to R, it chose channel 1 and sent an RTS to R and node R sent CTS back to S. Suppose that nodes C and D are already communicating over channel 3. Though nodes C and D are busy communicating data in channel 3, since the *listening radio* of C is listening to channel 1, it receives the CTS sent by node R and hence notes that channel 1 will be busy for the 'Network Allocation Vector' (NAV) amount of time. But for the synchronization and the extra radio (*listening radio*), multi-channel hidden terminal problem could not be avoided.

Thirdly, suppose the transmitting range of channel 1 is so short that node S can't reach R through that channel and that of channel 5 is long enough to reach its adjacent node. This is an example of heterogeneous environment. When node S wants to send a packet to R as discussed, S chooses channel 5 now, instead of channel 1 and starts its negotiations in the fifth time slot. Hence maximum connectivity is possible in a heterogeneous environment also.

In the following section, the effectiveness of the proposed protocol is demonstrated using simulations.

V. SIMULATION RESULTS

In this section the SYN-MAC protocol is compared with the CCC based protocol proposed in [7] for throughput performance and network connectivity.

A. *Throughput performance*

NS2 with CMU wireless extensions is used for this part of simulations. A multi-hop network with 80 nodes, randomly placed in a 1000*m* × 1000*m* area is considered. 40 nodes are chosen randomly as the sources and the other 40 nodes as destinations. The transmission range of each node is set to 250*m*. A set of three channels is chosen, each of which is available at each node with a probability of p = 80 %. The flow rate is varied for each connection to increase the network traffic and the throughput performance of CCC-MAC and SYN-MAC is compared. Each point in the graph in Fig. 5 is an average of 100 simulations.

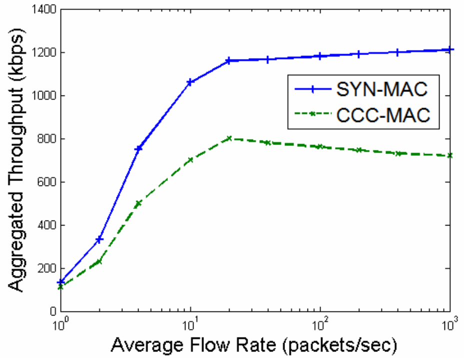

Figure 5. Average Throughput vs. Flow Rate. Packet size is 512 bytes.

Fig. 5 shows the aggregated throughput of both the protocols as the network traffic is increased. The throughput of SYN-MAC is significantly higher than that of CCC-MAC. The major reason for this behavior is that a CCC among all the nodes is not always available and so many times a connection is not established. Due to these failures the throughput is significantly lower in CCC-MAC.

It can also be observed that when the traffic is very high, the throughput of CCC-MAC starts dropping. This is because a single channel is used for control signal exchange. With increased traffic, contention of control packets increases and throughput degrades. Whereas, in SYN-MAC the control signal traffic is shared among all available channels and there is lower contention and hence, better throughput.

B. *Network Connectivity*

Now, we compare the network connectivity of CCC-MAC and SYN-MAC protocols using MATLAB simulations. Network connectivity is defined as the maximum percentage of nodes which are connected together either directly or through several hops. A network of 10 nodes randomly deployed in a 500*m* × 500*m* area is considered. Each node is assigned a set of channels. Each channel is available at a node with a probability of *p*= 80%.

Fig. 6 shows the percentage of network connectivity as a function of the number of channels. It can be observed that the SYN-MAC protocol assures higher connectivity than the CCC-MAC approach. As the number of channels is increased the probability that a common channel is found among all nodes increases. The rate of increase in SYN-MAC is higher than that of CCC-MAC. The SYN-MAC assures nearly 100% connectivity while the CCC-MAC provides only 65% connectivity for a group of 10 nodes and a set of 10 channels.

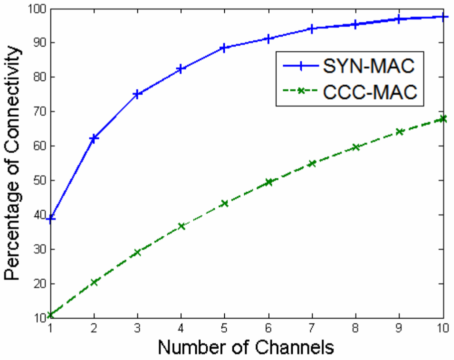

Figure 6. Percentage of network connectivity as the number of channels are varied for a group of 10 nodes.

Fig. 7 shows the percentage of network connectivity versus the number of nodes. It is observed that for a fixed number of channels, as the number of nodes is increased, the connectivity of both the approaches drop. But the fall in the CCC approach is very steep. This is due to the fact that as the number of nodes increases, the probability that a common channel is available at all the nodes decreases. For a network of just 10 nodes, the percentage connectivity of CCC-MAC fell to nearly 65% and that of SYN-MAC is 93%. So, it can be concluded that the proposed protocol provides higher network connectivity than the CCC approach.

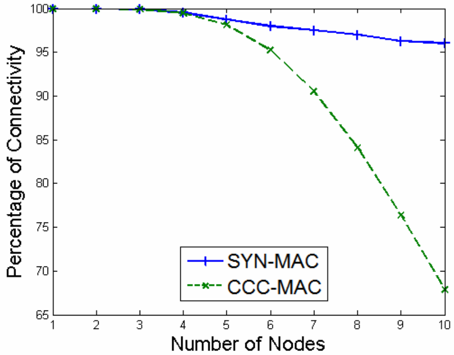

Figure 7. Percentage of network connectivity as the number of nodes are varied for a set of 10 channels.

VI. CONCLUSION

In this paper we have presented a MAC protocol for MHCRNs which avoids the need for a common control channel for the entire network. This automatically eliminates the control channel saturation problem and DoS attacks. The multi-channel hidden terminal problem is solved by introducing synchronization into the protocol. The protocol is also applicable to heterogeneous channels. NS2 simulation results show that SYN-MAC achieves higher throughput than CCC based protocols. It was also demonstrated through MATLAB simulations that SYN-MAC offers higher network connectivity than CCC-MAC.

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

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

[4] 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.

[5] Hao Nan, Sang-jo Yoo and Tae-In Hyon, “Distributed Coordinated Spectrum Sharing MAC Protocol for Cognitive Radio”, IEEE International Symposium on New Frontiers in Dynamic Spectrum Access Networks, pp. 240–249, April 2007.

[6] Liangping Ma, Xiaofeng Han and Chien-Chung Shen, “Dynamic Open Spectrum Sharing MAC Protocol for Wireless Ad Hoc Networks”, IEEE International Symposium on New Frontiers in Dynamic Spectrum Access Networks, pp. 203–213, November 2005.

[7] 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.

[8] 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.

[9] S.-L. Wu , C.-Y Lin , Y.-C. Tseng , J.-P. Sheu, “A New Multi-Channel MAC Protocol with On-Demand Channel Assignment for Multi-Hop Mobile Ad Hoc Networks”, Proceedings of the 2000 International Symposium on Parallel Architectures, Algorithms and Networks (ISPAN '00), p.232, December 07-07, 2000.

[10] Thoppian, M.; Venkatesan, S.; Prakash, R.; Chandrasekaran, R., “MAClayer scheduling in cognitive radio based multihop wireless networks”, World of Wireless, Mobile and Multimedia Networks, 2006.

[11] Q. Zhao, L. Tong, and A. Swami, “Decentralized cognitive MAC for dynamic spectrum access”, in Proc. of IEEE Symposium on New Frontiers in Dynamic Spectrum Access Networks, Nov. 2005.

[12] 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.

[13] Pradeep Kyasanur , Nitin H. Vaidya, Selfish MAC Layer Misbehavior in Wireless Networks, IEEE Transactions on Mobile Computing, v.4 n.5, p.502-516, September 2005.

[14] Q. Zhao, L. Tong, and A. Swami, “A Cross-Layer Approach to Cognitive MAC for Spectrum Agility”, in Proc. of Asilomar Conference on Signals, Systems, and Computers, Nov. 2005.

[15] Athanassios V. Adamis, Konstantinos N. Maliatsos, Philip Constantinou, “A New MAC Protocol with Control Channel Auto-Discovery for Self-Deployed Cognitive Radio Networks”, in Program for European Wireless 2007.