COMAS: a C0operative MultiAgent architecture for Spectrum sharing

Usama Mir, Leila Merghem-Boulahia and Dominique Gaïti

ICD/ERA, UMR 6279, Université de Technologie de Troyes, 12 rue Marie Curie, 10010 Troyes Cedex, France
{usama.mir, leila.merghem_boulahia, dominique.gaiti}@utt.fr

Abstract. Static spectrum allocation is a major problem in recent wireless network domains. Generally, these allocations lead to inefficient usage creating empty spectrum holes or white spaces. Thus, some alternatives must be ensured in order to mitigate the current spectrum scarcity. An effective technology to ensure dynamic spectrum usage is cognitive radio, which seeks the unutilized spectrum portions opportunistically and shares them with the neighboring devices. Using cognitive radio capabilities, the nodes are not restrained to static spectrum utilization, rather they can choose it opportunistically. However, since nodes generally have a limited knowledge about their environment, we claim that cooperative behavior can provide them with the necessary information to solve the global issues. Therefore, in this paper, we develop a novel approach for spectrum allocation using a multiagent system that enables cognitive radio devices to work cooperatively with their neighboring licensed (or primary user) devices in order to utilize the available spectrum dynamically. The fundamental aspect of our approach is the deployment of an agent on each device which cooperates with its neighboring agents in order to have a better spectrum sharing. Considering the concurrent, distributed and autonomous nature of the proposed approach, Petri nets are adopted to model the cooperative behaviors of primary and cognitive radio users. Then, we analyze the performance of proposed approach under ad-hoc conditions where the users are poissonly distributed with the continuous changes in their neighborhoods. The experimental results show that our approach achieves good performance in terms of spectrum access, sustaining lower communication overhead.

Keywords: Cognitive radios, dynamic spectrum sharing, multiagent systems, cooperation, Petri Nets.

I. INTRODUCTION

The deployment of modern day wireless devices follows the static spectrum usage where spectrum is assigned to a licensed user for longer durations. This static spectrum assignment is considered to be extremely favorable in order to avoid the device-level collisions; however, it leads to radio spectrum shortage problem creating empty spectrum *holes*. According to [24], in both rural and urban areas, spectrum usage can go as low as 10-15%, resulting in huge amount of spectrum to be wasted. As a result, the newly arriving unlicensed devices are forced to use unlicensed bands leading to inefficient and crowded spectrum utilizations.

Cognitive radio (CR), firstly coined by *Mitola* [18], is considered to be an efficient technology to enable dynamic and opportunistic spectrum sharing. Generally, a CR (or secondary) user senses the nearby empty spectrum portions and is capable of sharing them with the neighboring devices, without interrupting the working of licensed (or primary) users. It continuously monitors the environmental radio frequency (RF) signals and alters its transmission and reception parameters in order to better perform its functions. However, one of the key issues in CR networks is to avoid device level collisions and interferences while maintaining efficient spectrum usage. We argue that a non-cooperative node can cause harmful interference to its neighbors and hence can reduce the overall spectrum usage.

One effective solution to enable inter-device cooperation and information exchange is multiagent system [20]. Basically, a multiagent system (MAS) is composed of groups of autonomous agents working together by having frequent interactions with each other, in order to solve the tasks which are beyond the capabilities of a single agent [7]. Each agent works dynamically to fulfill its user needs and has partial knowledge and imperfect information about the nearby environment. In addition, an efficiently designed CR device, with an agent embarked on it, is capable of interacting with neighboring radios to form a dynamic and collaborative network and provides a rationale to conceptualize new spectrum sharing techniques for CR networks. This rationale is particularly attractive and equally important to create spectrum sharing solutions that can work in dynamic, distributed and open wireless networks domains.

Therefore, in this paper, an MAS based approach is proposed, where the primary and secondary users are equipped with agents. The secondary user (SU) agents coexist and cooperate with the legacy PU agents using the

message passing and decision making mechanisms of contract net protocol (CNP) [25]. The whole environment is ad-hoc with the frequent changes in the neighborhoods of primary and secondary users. Moreover, in order to capture the agents' interactions under mobile conditions, Petri net (PN) modeling is used [8]. The graphical and analytical nature of PN allows us to visualize the detail feasibility analysis of agents' internal behaviors when they have to make spectrum sharing deals/agreements. While passing through several cooperative stages, we study the inter-agent message exchange in order to make cooperative decisions. In this context, our previous works [29][30] have focused on proposing a cooperative spectrum sharing framework and analyzing its behavior. Unlike our previous contributions, in this work we deploy our primary and secondary users according to Poisson distributions [17] and monitor their arrivals and departures under ad-hoc conditions. These distributions help us identifying the exponential time periods (or holding times) for which the users utilize the available spectrum.

The rest of the paper is organized as follows. Prior works related to dynamic spectrum sharing are summarized in Section II. Section III formulates the problem statement. In Section IV, we propose our cooperative framework. Petri net model along with some important properties of our design is presented in Section V. Section VI depicts simulation results. Finally, Section VII concludes our work with the future perspectives.

II. RELATED WORK

In the recent past, several dynamic spectrum sharing approaches have been proposed using different techniques such as game-theory and auctions, medium access control protocols, etc. In fact, some researchers have also drawn their focus towards multiagent based approaches for spectrum sharing. In [32], an MAS is used for managing spectrum resources across several wireless LANs (WLANs), collocated in a geographical area. Each access point (AP) located in a WLAN contains an agent which interacts with the neighboring AP agents (located in other WLANs) to form an MAS. The internal architecture of an agent consists of two parts: *predictive parameter estimation,* which generates parameter estimates using the signal characteristics received from WLAN environments and *resource management optimization*, which decides the suitable spectrum bands to be selected. The proposed approach is explained conceptually, but none of the analysis and experiments are shown. On contrary, the works proposed in [1][10] consider market-based auctions for dynamic spectrum sharing. The SUs working as *consumer agents* submit their bids to the PUs (or *auctioneer agents*), which shows their willingness for spectrum sharing. The *auctioneer agents* then share the spectrum based on the received bids. The ultimate aim of using auctions is to provide an incentive for SUs to maximize their spectrum usage (and hence the utility), while allowing network to achieve *Nash Equilibrium*. However, considering the competitive nature of market based approaches, it is hard to develop agents with unselfish behaviors.

According to *Weib et al* [9], agents should have the ability to learn from their past states in order to better perform their following actions/moves. This MAS learning can provide significant contribution to spectrum allocation in CR networks, if the devices have the knowledge of their past sharing patterns and neighborhood movements. In vicinity, the solutions based on MAS learning are presented in [11][15]. Basically, the SUs periodically share the relative traffic information on the sensed channels (they are likely to be using in near future), with the neighboring devices. Based on this information exchange, multiagent learning (i.e. delay sensitive and Q-learning) algorithms are proposed which allow the CR users to dynamically and autonomously optimize their transmission power on a selected channel and to avoid the inter-device interferences. Conversely, sometimes these learning algorithms can create a situation where the agents have weak assumptions about other agents' spectrum usage making the task of getting accurate information more difficult.

A different cooperative approach named as DSAP (dynamic spectrum access protocol) is presented in [31]. This approach is based on the concept of centralized server which is responsible for leasing spectrum to the requesting users in a small geographical region. The server also maintains a global view of the network's channel conditions through a series of frequent information exchanges with its clients. However, centralized server can become a huge bottleneck in diverse network conditions.

Game-theoretical solutions are considered to be a perfect match of nature for dynamic spectrum allocations. Mostly, in these approaches [6][23], to efficiently utilize the scarce spectrum resource, PUs adopt the roles of the *leaders*, by selecting a subset of neighboring SUs and granting them spectrum access. In return, SUs work as the *followers*, by paying PUs the relative price for spectrum utilization and maximizing their utilities in terms of spectrum access for a specific time period. Yet, each user focuses on maximizing its individual usage without taking into account the others, showing selfish behaviors. As a result, in order to allow players to work interdependently, cooperative games are proposed [2][19]. In cooperative games, the SUs' transmission powers and spectrum usage are common knowledge and their utility functions are chosen in order to maximize the global utility. At the same

time, cooperative approaches require a feedback from each player to be sent to the centralized server about its utility function, increasing the overall algorithmic complexity.

A local bargaining approach is presented in [21] where the CR users self-organize into small bargaining groups. The group formation process starts by the initiator CR node, sending a group formation request to its neighbors for a subset of spectrum portions. The interested neighbors acknowledge the request and the bargaining group is formed ensuring minimum spectrum allocation to each group member. The experimental results prove that local bargaining performs similar to greedy approach [14] incurring less communication overhead.

Beside from local bargaining and game-theoretical approaches, some authors suggest that the spectrum sharing problems are similar to MAC issues [26], where several users try to access the same channel and their access should be coordinated with the neighboring users to avoid interferences. In MAC-based spectrum sharing [13][27], when an SU is using a specific channel, both the transmitter and the receiver synchronize themselves by sending a busy tone signal through the associated control channel, such that the signal interferences should be avoided. Nevertheless, sending frequent busy tones can interrupt the neighboring devices, because each time they have to stop their normal working flow in order to listen to the busy tone on the control channel.

Our proposed approach is different from the above as we consider a framework where PU agents are working together with the SUs in order to enable dynamic spectrum allocations. The agents' internal behaviors are cooperative and unselfish, which allow them to help maximizing each other's utility functions. To the best of our knowledge, the idea of deploying a cooperative MAS over CR networks under ad-hoc conditions along with the modeling of users' spectrum sharing process using Petri nets and detailing their internal message structures, has not been previously addressed. Therefore, we think our work will provide a novel contribution to the current dynamic spectrum access literature.

III. PROBLEM DESCRIPTION

Before formulizing our problem statement, let us consider an ad-hoc network scenario[1] in Figure 1. The figure depicts an emergency situation during an accident in a very remote area, where the user is in a non-covered zone (i.e. the radio resources at this moment are not available) or the radio access technology requires an energy that the terminal (a mobile, a laptop, or a PDA) does not own. In this case, an SU should observe the nearby PUs (PU_1 and PU_2) and sense their transmission signals to identify the available spectrum bands. Then, its agent can cooperate with the conforming primary user agents to make dynamic spectrum sharing agreements.
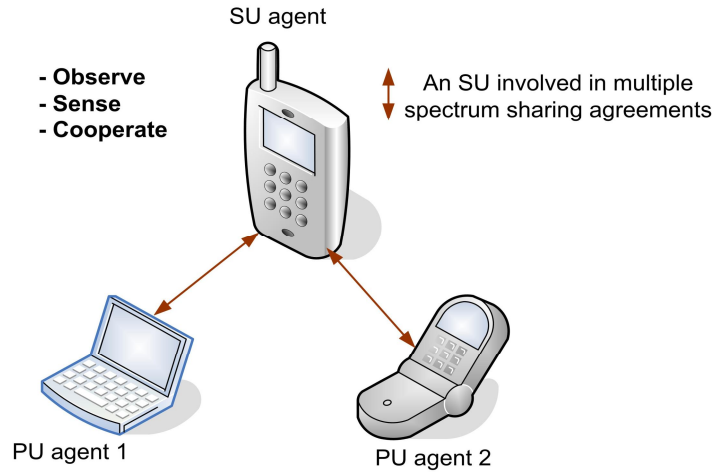

Figure 1. Spectrum sharing under an ad-hoc network

We now formulize our problem. Let $G = (N, A)$ be a directed network consisting of a set of mobile nodes N such that $(SU \cup PU) \in N$ and a set of directed arcs A. Each directed arc $(i, j) \in A$ connects a secondary user SU_i to a

---

[1] A scenario of two PUs and one SU is shown just as an example for depicting the deployment of cooperative agents under ad-hoc network setting.

primary user PU_j. Similarly, we can denote the directed arc $(j, i) \in A$ to connect PU_j to SU_i. The secondary users are cooperating with the neighboring primary users to have a spectrum sharing agreement. We assume that s_{ij} is the amount of spectrum a secondary user *'i'* is desiring to get from a primary user *'j'*. Similarly, t_{ij} is the amount of time for which *'i'* wants to utilize the spectrum and p_{ij} is the price it is willing to pay to *'j'*. On the other hand, for a primary user *'j'*, s_{ji} is the amount of spectrum it is willing to share with *'i'*, t_{ji} is the respected time limit and p_{ji} is the price it is expecting to get after sharing its spectrum. We can formulate the above model for each secondary user *'i'* as:

$$\text{Maximize} \sum_{(i,j) \in A} s_{ij} t_{ij} \qquad (1)$$

Subject to

$$\text{Minimize} \sum_{(i,j) \in A} p_{ij} \quad \forall\, SU \in N \qquad (2)$$

Similarly for primary users:

$$\text{Maximize} \sum_{(j,i) \in A} p_{ji} \qquad (3)$$

Subject to

$$\text{Minimize} \sum_{(j,i) \in A} s_{ji} t_{ji} \quad \forall\, PU \in N \quad (4)$$

And
$$l_{ji} \leq s_{ji} \leq u_{ji}$$

where l_{ji} and u_{ji} are the lower and upper bounds of available spectrum of *'j'*. This means that *'i'* cannot ask for an amount of spectrum above this limit.

IV. COOPERATIVE SPECTRUM ALLOCATION FRAMEWORK

In this section, we explain the internal architectures of primary and secondary users along with their working behaviors. Basically, our design (as shown in Figure 2) is based on five different interlinked parts that embody the working of our cooperative approach:

- Dynamic spectrum sensor;
- Spectrum characterizer;
- Secondary user interface;
- Agent's knowledge module; and
- Agent's cooperation module.

The working of these modules is described, in the following. Obviously, the primary user does not contain the cognitive radio module, while the agent module is common in both the primary and secondary users. Note that, in essence, spectrum sensing and characterizing is beyond the scope of our work, thus our focus will be on agent module which is the key functionary to enable cooperation between primary and secondary users.

A. Dynamic Spectrum Sensor

The function of dynamic spectrum sensor (DSS) is the sensing of radio spectrum *holes* by continuously monitoring the neighboring PU signals. Several techniques such as PU's weak signal and its energy detection [34], cooperative detection [3], etc can be used to perform spectrum sensing. For DSS, it is also necessary that the sensing is performed by considering a real-time dynamic environment, because it is not obvious at what time a spectrum band is occupied or when it is free. Thus, all the factors such as PU's signal power with the respected noise, spectrum traffic (by calculating the number of current users and taking into account the application type), sampling time and intervals must be kept in consideration.

B. Spectrum Characterizer

Spectrum characterizing can be considered as a sub-function of spectrum sensing. Basically, our spectrum characterizer (SC) module functions as to arrange/divide the spectrum *holes* information (received through DSS)

according to capacity. In a simple way, to create a capacity based descending ordered list of neighboring PUs, SC uses the Shannon Theorem :

$$C = B \log_2(1 + SNR)$$

Where C is the capacity in *bits per second*, B is the bandwidth measured in *hetrz* and *SNR* is the respected signal-to-noise ratio in *watts*. For more details, the complete derivation and formulation of the above equation is found in [4].
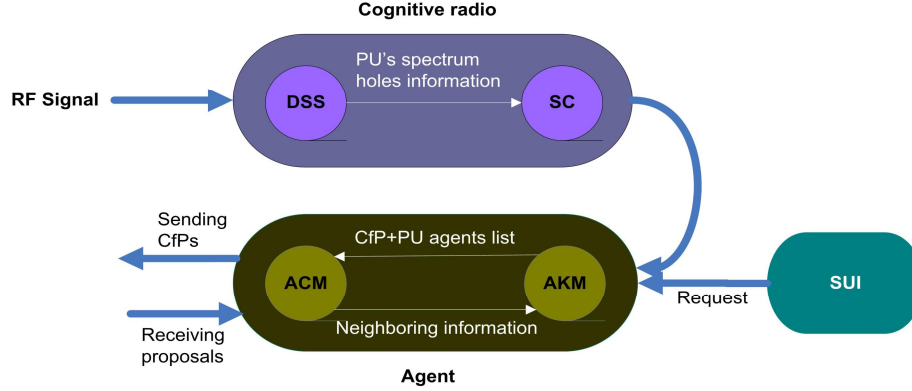

Figure 2. Various functioning modules for primary and secondary users

C. Secondary User Interface

The third part, secondary user interface (SUI) sends a *request* message to the agent module, whenever a user wants to have a portion of spectrum (for internet surfing, watching high quality videos, etc.). The message is of the form *req (s,t)*, where *s* is the amount of spectrum needed by the SU depending upon its application in use, for a time duration *t*. In reality, the user's request depends upon the application to be used. For example, if a user runs a Skype based multimedia application on its PDA or cell phone on daily bases, then each time this application is executed, its request for spectrum utilization will remain the same.

D. Agent's Knowledge Module

Agent's knowledge module (AKM) gets PUs' characterization information from SC module which serves as a motivation for agents that subsets of neighbors having unutilized spectrum portions are available. This list is not permanent rather it is updated and maintained on regular time intervals. Secondary user's AKM (or SU-AKM) also gets the *req* message from SUI module and based on the inputs from both the modules, it prepares a *call for proposal (CfP)* message:

$$CfP (SUID, s, t, d)$$

Where *SUID* is the secondary user's ID (or its agent's identification) and it is used to help PU to reply back to the corresponding SU, *s* is the amount of spectrum needed by the SU, *t* is the desired time limit (or holding time) for the spectrum utilization and *d* is the deadline to receive the PUs' responses (*proposals*). Parallel to *CfP* creation, SU-AKM maintains the neighboring PUs' information that is received via frequent interactions between the agents, along with a list of previously received *proposals* (if there exist any). This information includes the leaving and joining of neighboring nodes in a network and their current spectrum status and it helps an SU to create a more precise *CfP*. Uniformly, the PU-AKM module functions almost in the same manner by maintaining the neighboring SUs' arrivals and departures information and a list of their previous spectrum demands.

E. Agent's Cooperation Module

Agent cooperation module (ACM) manages the cooperation between primary and secondary users. After the reception of a *CfP* message from SU-AKM, the SU-ACM sends the received *CfP* to the neighboring and currently available PU agents. The PUs are considered to be available, if they still exist in the corresponding SU's neighborhood with their spectrum portions. Besides, SU-ACM also performs the main decision for an SU by selecting the appropriate *proposal*. In much the same way, PU-ACM chooses the most suitable *CfP* for a PU and sends the *proposal* in response. Finally, the appropriate agreement for both the primary and secondary users is the one which is profitable and maximizes their utility values.

On average, the utility for a PU is the price paid by SU agents for their spectrum utilization divided by the amount of spectrum it has shared for the respected time period. An SU agent's utility is represented as its spectrum usage for the required time divided by the corresponding price paid to the PUs.

Accordingly, Figures 3 and 4 delineate the behavioral working of secondary and primary users respectively. Both the behaviors show the same characterizing, analyzing, sending, receiving and deciding steps mentioned before. The spectrum sharing process for an SU starts by getting the characterization results and the user requirements and continues until the sending of *CfPs* and receiving of the *proposals*. The process ends either by having an agreement or disagreement. For a PU, the process follows the same pattern by first analyzing the received *CfPs*, sending the *proposals* as responses and finally ending the process either by receiving an *accept* or a *reject* message from the conforming SU.
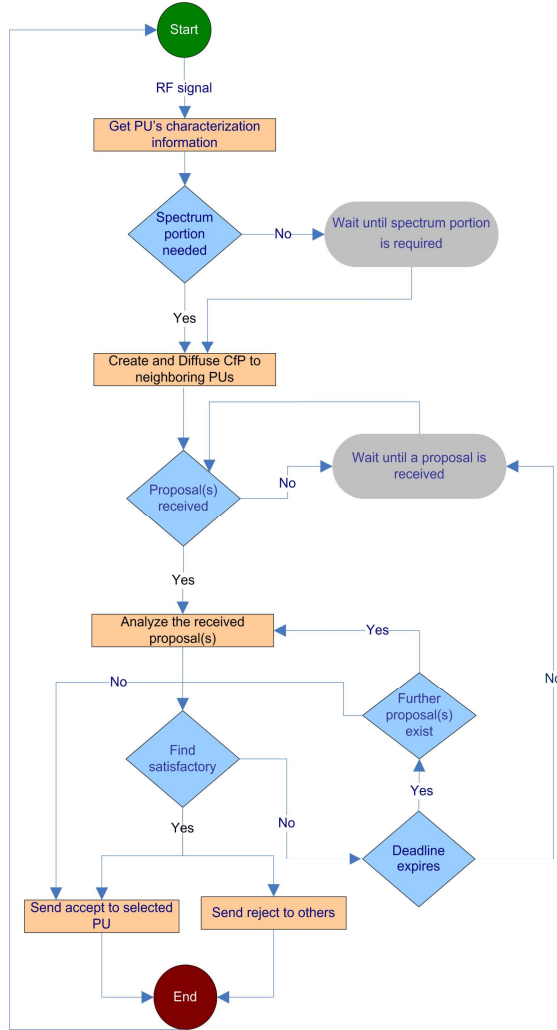
Figure 3. Secondary user's behavior   Figure 4. Primary user's behavior
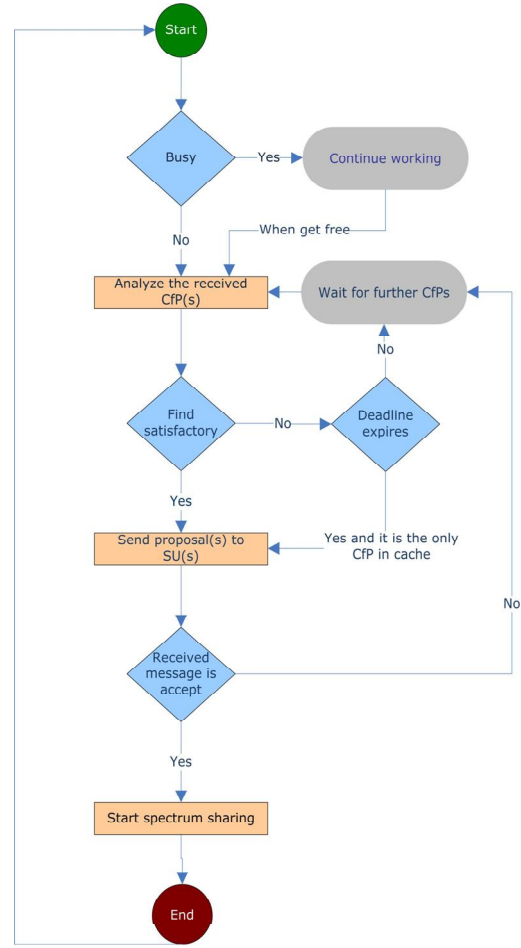
V.   PETRI NET MODEL FOR THE COOPERATIVE APPROACH

Petri Net (PN) [8] is a graphical tool for the formal description of the flow of activities in complex systems. Generally, PNs are used to represent the logical interactions among nodes, devices and parts of a system. Their discrete and distributed nature makes them highly suitable to model inter-agent interactions and capture the dynamics of decentralized environments. A simple PN model is shown in Figure 5. Basically, it contains two types

of nodes, namely a set of places $P = \{p_1, p_2, ... p_n\}$ and a set of transitions $T = \{t_1, t_2, ... t_m\}$. A place is represented by a circle and a transition can be shown by a bar (or a box). Further, a PN consists of a set of inputs I, outputs O and the markings M (assignment of *tokens* to the places). The marking of a PN is a vector, the components of which are positive integer values. The dimension of this vector is equal to the number of places. A *token* (represented by a small filled circle) is moved from one place to another when a transition is fired. For example, in Figure 5, firing of transitions $t_1, t_2, t_3,$ and t_4 can allocate several resources from one agent to another and these firings can be represented by removal of *tokens* from input places and their addition to the output places.
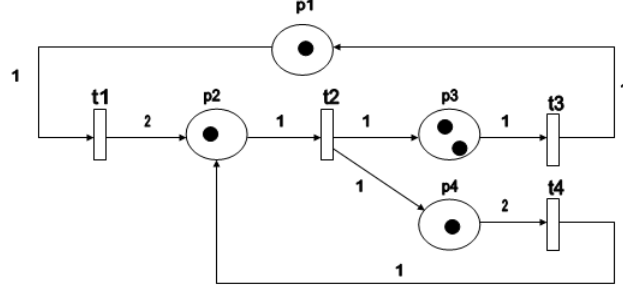

Figure 5. A simple PN with initial marking Mo = {1,1,2,1} [16]

A. Modeling Spectrum Sharing Agreement/Disagreement using PN

A cooperative spectrum sharing model between a primary and a secondary user is a five tuple $N = \{P, T, I, O, M\}$, where P is the finite set of places, T is a finite set of transitions, I and O, are the input and output functions which specify the input and output places of transitions and $M = \{M_0, M_f\}$ is the set of markings such as M_o and M_f denote the sets of initial and final markings, after firing all the transitions. We also denote by:

- $*t$ is the set of input places p of a transition t: $(p,t) \in I$.
- $t*$ is the set of output places p of a transition t: $(t,p) \in O$.
- $*p$ is the set of input transitions t of a place p: $(t,p) \in I$.
- $p*$ is the set of output transitions t of a place p: $(p,t) \in O$.

Places represent several states of primary and secondary users during a spectrum sharing agreement/disagreement. A transition $t \in T$ is enabled when an event is about to occur (e.g. a *CfP* is ready to be sent) and it is fired when the event occurs (i.e. a *CfP* has been successfully sent). Firing a transition will remove *token(s)* from each $*t$ and will add them to $t*$. Formally, firing transitions consists of transforming the $m_o \in M_o$ into $m_f \in M_f$ as follows [16]:

$$m_f(p) = \begin{cases} m_o(p) - firedtoken(s) & \text{if } p \in *t \\ m_o(p) + firedtoken(s) & \text{if } p \in t* \\ m_o(p) & \text{otherwise} \end{cases} \quad (5)$$

The cooperation process of an SU with two neighboring PUs to make a spectrum sharing agreement is shown in Figure 6, while the description of various states and transition is summarized in Tables 1 and 2. The spectrum sharing process starts with places p_1, p_4 and p_5 where SU_i is ready to send $CfPs$ and PU_1 and PU_2 are ready to receive them. Firing transition t_1 removes a token from place p_1 and adds it to p_2, p_3 and p_6. Thus when transition t_1 is fired, the $CfPs$ are sent from SU_i to PU_1 and PU_2. Both the tokens from p_2 and p_3 enable transitions t_2 and t_3 and firing of these two transitions adds one *token* each to p_7 and p_8, showing the reception of $CfPs$ by PU_1 and PU_2. Similarly, remainder of the message passing process follows the same pattern (of *tokens* removal and addition) where SU_i accepts PU_1 for spectrum sharing and it rejects PU_2 due to its unsatisfactory proposal. The states and transitions involved in these message exchanges are:

$proposals = \{p_9, p_{10}, p_{11}, p_{12}, t_4, t_5, t_6\}$, $accept = \{p_4, p_{13}, p_{15}, p_{16}, p_{20}, t_7, t_9, (t_{12} \text{ or } t_{13})\}$ and $reject = \{p_1, p_{14}, p_5, p_{17}, p_{18}, t_8, t_{10}, (t_{14} \text{ or } t_{15})\}$.

After the reception of an *accept* message, the spectrum sharing is started between SU_i and PU_1 and it continues until the spectrum is completely utilized and the respected price is paid. A successful spectrum sharing contains rest of the states and transitions of Figure 6. The primary users will behave in the same manner when they have to deal with multiple $CfPs$ at a time. Moreover, if a secondary user receives more than one *proposal* which is equally satisfactory, then the decision will be made on the FIFO bases. Finally, Table 3 depicts the initial and final markings

of *tokens* after firing all the transitions. It is clear that the value of M_f becomes '3', only when a spectrum portion is been shared or the price is been paid.
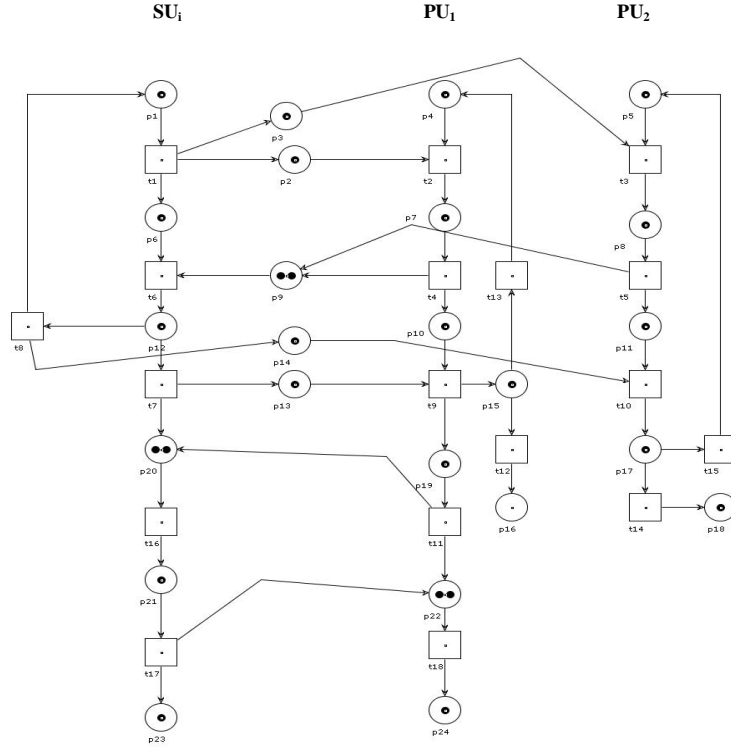

Figure 6. Spectrum sharing process between an SU and two PUs

Table 1. Spectrum sharing states

p_1	SU_i: ready to send *CfP*
p_2	PU_1: PU agent's cache (*CfP* arrives)
p_3	PU_2: PU agent's cache (*CfP* arrives)
p_4	PU_1: ready to receive *CfP*
p_5	PU_2: ready to receive *CfP*
p_6	SU_i: *CfP* sent and wait for *proposals*
p_7	PU_1: *CfP* received
p_8	PU_2: *CfP* received
p_9	SU_i: SU agent's cache (*proposals* arrive)
p_{10}	PU_1: *proposal* sent and wait for the final response
p_{11}	PU_2: *proposal* sent and wait for the final response
p_{12}	SU_i: *proposal* received
p_{13}	PU_1: PU agent's cache (*accept* arrives)
p_{14}	PU_2: PU agent's cache (*reject* arrives)
p_{15}	PU_1: temporary *waiting* phase
p_{16}	PU_1: further *CfP* receiving stopped
p_{17}	PU_2: *reject* received and temporary waiting phase
p_{18}	PU_2: further *CfP* receiving stopped
p_{19}	PU_1: ready to *share* the acquired spectrum
p_{20}	SU_i: ready to *utilize* spectrum
p_{21}	SU_i: spectrum utilized and ready to *pay* price
p_{22}	PU_1: spectrum shared and ready to *receive* price
p_{23}	SU_i: price *paid*
p_{24}	PU_1: price *received*

Table 2. Various transitional phases

t_1	SU_i: send *CfP*
t_2	PU_1: receive *CfP*
t_3	PU_2: receive *CfP*
t_4	PU_1: send *proposal*
t_5	PU_2: send *proposal*
t_6	SU_i: receive *proposal*
t_7	SU_i: send *accept*
t_8	SU_i: send *reject*
t_9	PU_1: receive response (*accept*)
t_{10}	PU_2: receive response (*reject*)
t_{11}	PU_1: start *sharing* the spectrum
t_{12}	PU_1: continue receiving further *CfPs*
t_{13}	PU_1: stop receiving further *CfPs*
t_{14}	PU_2: stop receiving further *CfPs*
t_{15}	PU_2: continue receiving further *CfPs*
t_{16}	SU_i: start *utilizing* the acquired spectrum
t_{17}	SU_i: *pay* price
t_{18}	PU_1: *receive* price

B. Some Definitions

By analyzing our proposed model along with the tables presented above, we can conclude the following few definitions:

Definition 1: For a secondary user $SU_i \in SU$, the Petri net $N_i = \{P_i, T_i, I_i, O_i, M_i\}$ for an empty part of the spectrum s_i for time t_i, is a working model if and only if the movement from initial marking M_{osi} to final marking M_{fsi}, after firing all transitions, results in the maximization of its utility function U_{sui} such that $U_{sui} = (s_i \times t_i)/c_i$ where c_i is the corresponding price SU_i is asked to pay for its spectrum utilization. Similarly, for a primary user $PU_j \in PU$, a Petri net $N_j = \{P_j, T_j, I_j, O_j, M_j\}$ for an empty spectrum portion s_j with the associated price c_j, the movement from initial marking M_{opj} to final marking M_{fpj}, results in the maximization of its utility function U_{puj} such that $U_{puj} = c_j/(s_j \times t_j)$. Both the primary and secondary users work cooperatively to maximize each other's utility functions. Especially, the primary users send the proposals which are in their own profit as well as of the requesting secondary users.

Table 3. Initial and final markings

	M_O	t_1	t_2	t_3	t_4	t_5	t_6	t_7	t_8	t_9	t_{10}	t_{11}	t_{12}	t_{13}	t_{14}	t_{15}	t_{16}	t_{17}	t_{18}	M_f
p_1	1	-1							+1											1
p_2	1	+1	-1																	1
p_3	1	+1		-1																1
p_4	1		-1											+1						1
p_5	1			-1												+1				1
p_6	1	+1					-1													1
p_7	1		+1		-1															1
p_8	1			+1		-1														1
p_9	2				+1	-1	-2													2
p_{10}	1				+1					-1										1
p_{11}	1					+1					-1									1
p_{12}	1						+1	-1	-1											0
p_{13}	1							+1		-1										1
p_{14}	1								+1		-1									1
p_{15}	1									+1			-1	-1						0
p_{16}	1													+1						2
p_{17}	1											+1			-1	-1				0
p_{18}	1														+1					2
p_{19}	1									+1		-1								1
p_{20}	2							+1				+1					-1			3
p_{21}	1																+1	-1		1
p_{22}	2											+1						+1	-1	3
p_{23}	1																	+1		2
p_{24}	1																		+1	2

Definition 2: A Petri net $N_s = \{P_s, T_s, I_s, O_s, M_s\}$ for an empty part of the spectrum s, is said to be unsuccessful if moving from M_{os} to M_{fs}, results in no change in the utility functions of both the participating secondary and primary users such that:

$$U_{sui} = 0 \text{ and } U_{puj} = 0, \; \forall \{SU_i, PU_j\} \in \{SU, PU\} \qquad (6)$$
$$\text{where } \{U_{sui}, U_{puj}\} \in \mathbb{R}$$

In our proposed model, the PN between SU_i and PU_2 proved to be unsuccessful, because after performing all the cooperation steps, the utility functions of both the users remain unchanged, resulting in spectrum sharing disagreement.

C. Behavioral Properties

To verify the efficiency of our approach, we provide the behavioral properties of the proposed PN model. These properties, when interpreted in the context of the modeled system, allow the system designer to understand the working of the considered network. By behavioral properties we mean the properties which are dependent on all the

markings of a PN i.e. the initial and final markings are interlinked. Thus, we provide here the most important behavioral properties such as reachability, boundness and liveness [28].

Reachability

In order to find out the reachability of the proposed model, it is necessary to locate a sequence of events or transitions which would transform an initial marking to a final marking by representing the required functioning behavior of the network. This transformation is similar to real networks where a sequence of steps (i.e. sending and receiving of messages, accepting requests) would allow the network to achieve its required goals. The reachability also indicates the presence of the real network related facts, which reflect the behavior of participating nodes. For our proposed model in Figure 6, let m_o be the initial marking for a state p, then the final marking m_f reachable from m_o can be written as:

$$m_f = m_o + v \tag{7}$$

Where v is the vector containing the firings related to p. v can be denoted as:

$$v = \sum_{t \in {}^* p} \varphi_{(t,p)} - \sum_{t \in p^*} \sigma_{(t,p)} \tag{8}$$

Here, $\varphi_{t,p}$ and $\sigma_{p,t}$ are the number of tokens added and consumed from p. For all $p \in P$, equation (7) can be written as:

$$M_f = M_o + V \tag{9}$$

Where M_o and M_f are the initial and final marking sets for all $p \in P$ and V is a vector containing all v related to all p.

Reachability is very important for our proposed model as it helps us calculating the final markings reachable from initial markings and shows the flow of tokens from place to place. To prove this property from our proposed PN let us give an example. As mentioned before, Table 3 shows all the initial and final markings with the addition and subtraction of tokens for Figure 6. From this table, we can calculate V and M_o as follows:

$$V^T = (0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ -1\ 0\ 0\ -1\ 1\ -1\ 1\ 0\ 1\ 0\ 1\ 1\ 1) \tag{10}$$

$$M_o^T = (1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 2\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 2\ 1\ 2\ 1\ 1) \tag{11}$$

Using (10) and (11), the equation (9) becomes:

$$M_f^T = (1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 2\ 1\ 1\ 0\ 1\ 1\ 0\ 2\ 0\ 2\ 1\ 3\ 1\ 3\ 2\ 2) \tag{12}$$

Thus in our PN, the addition of vector V with initial marking set M_o enables us to reach the final marking set M_f.

Boundness

A place p_i of a PN model is said to be k-bounded if the number of tokens in p_i always remain less than or equal to k i.e. $x(p_i) \leq k \ \forall \ p \in P$, where k an integer value > 0 and $x(p_i)$ are the number of tokens in place p_i. p_i is always bounded when it is k-bounded. Boundness is an important property in order to check the design errors in a PN model. For instance, some tokens may permanently stay in places and create serious bottlenecks for the whole PN. In Figure 6, boundness holds for all the places as the number of tokens in each place are within $0 < x(p) \leq k \ \forall \ p \in P$, where $k = \{1,2\}$.

In contrast to reachability which is verified using marking sets, boundness can be easily verified using coverability graph [16]. Coverability graph represents all the possible markings of a PN model in the form of a simple tree where M_o represents the root node, all the markings reachable from M_o are the nodes and the arcs represent the firing of transitions. Therefore, for simplicity we construct a coverability graph in Figure 7 for the initial part of Figure 6 i.e. for states $(p_1, p_2, p_3, p_4, p_5, p_6, p_7, p_8, p_9, p_{10}, p_{11}, p_{12})$ and transitions $(t_1, t_2, t_3, t_4, t_5, t_6, t_8)$. From the graph, we notice that states $(p_1, p_2, p_3, p_4, p_5, p_6, p_7, p_8, p_{10}, p_{11}, p_{12})$ are 1-bounded and state p_9 is 2-bounded. Thus, the graph remains k-bounded for all the states and value of k never becomes 0. Boundness can similarly be verified for all the remaining states and transitions of Figure 6.

Liveness

Definition A: In wireless networks, different tasks are performed at several time instances and similar is the case with Petri net models where many transitions are ready to fire at distinct times. Generally, a PN is live when there exists at least one transition $t \in T$ which is ready to be fired at a particular time instance. Formally,

$$\sum_{t \in T} \varrho_t \left(\Gamma \right) \geq 1 \qquad (7)$$

Where Γ is the time instance and ϱ_t is the function indicating either a transition t is ready to fire or not [35]. Our PN remains live during a spectrum sharing process because at least one $t \in T$ always remains ready to be fired. For example, when PU_1 and PU_2 receive the *CfPs*, the transitions t_4 and t_5 are in their ready states and they are fired when PU_1 and PU_2 send their proposals to SU_i. Then, at the next time instance, transition t_6 is ready to fire and is fired when SU_i sends its responses to PU_1 and PU_2 and similar is the case with the other transitions. Thus, the PN remains live for the whole process of spectrum sharing.
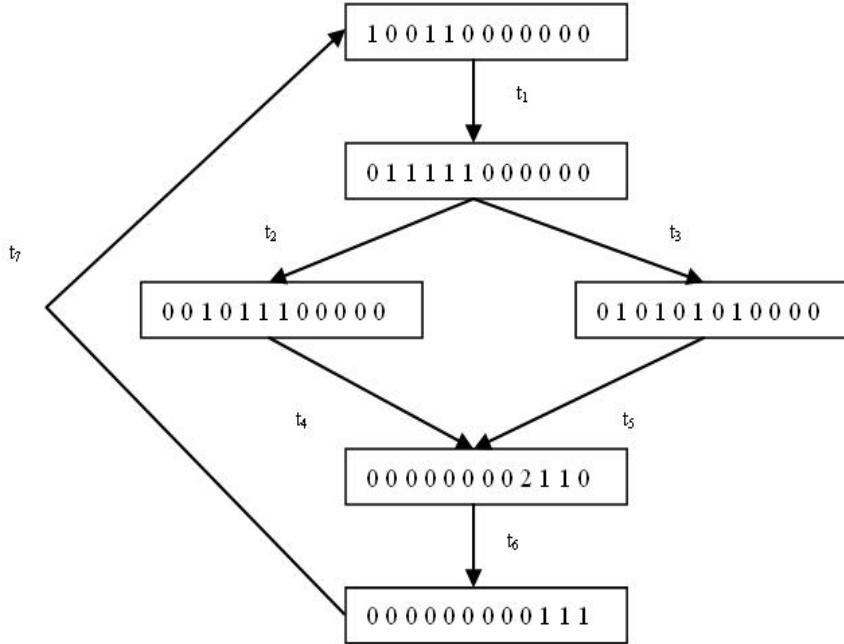
Figure 7. Coverability graph for some states and transitions of Figure 6

Definition B: A PN is said to be live if and only if, from any node in its reachability graph, it is possible to find a directed path having this node as origin. In other words, the PN is live when each node of its directed graph is the origin of at least one arc. If we consider the coverability graph of Figure 7, we notice that each node contains at least one arc as origin, thus this coverability graph is live. Likewise, for the proposed PN model of Figure 6, we can clearly see that each transition contains at least one arc to reach its next state until the spectrum sharing process arrives to its end. Therefore, we can conclude that the proposed PN is live.

VI. EXPERIMENTAL RESULTS

A. Setup

In this section, we present various numerical results to evaluate the working of the proposed cooperative approach, based on the following simulation setup. Multiple sets of primary and secondary users are randomly placed

according to Poisson distribution [17] in a noiseless and mobile ad-hoc network with the continuous change in their neighborhoods. The SUs cooperate with the PUs in order to make spectrum sharing deals/agreements. According to the studies presented in [6][33], we set the elapsed simulation time T_E to 90 minutes. All the simulations are conducted in Java Application Development Environment (JADE) [36], over a PC with 2.4 GHZ dual processor and 4GB memory.

The parameter selection is shown in Table 4. As mentioned in [5], we set the bandwidth of each spectrum portion to *3.75MHz*. The simulation is conducted for a total of 10 runs (i.e. r_N=10)[2] and the average values are taken for plotting the graphs. According to [22], the fractional percentage of the time for which the spectrum is being used by the PUs (or holding time) in an urban environment follows the exponential distribution with mean μ_p and is measured as approximately 40 to 45%, thus we set the mean spectrum usage of PUs to 40%. Considering the capacity of a single machine, the maximum number of primary and secondary users is 100 in total. Moreover, the rates λ_p and λ_s denote the Poisson distributions of primary and secondary users. For simplicity, we fix the value of λ_s to 5 and compare our parameters at various values of λ_p (i.e. λ_p = 3, 4, and 5). This variation factor helps us understanding the behaviors of SUs when they have to deal with different numbers of PUs. Additionally, during a simulation run of 90 minutes, we also observe distinct values of our parameters over five different time intervals of 18 minutes each, ensuring that the parameters can be compared across several time instants. We calculate a primary user's utility as the price paid by SUs for spectrum utilization divided by the amount of spectrum it has shared for the respected time period. Likewise, a secondary user's utility is represented as its spectrum usage for the required time divided by the price paid by the PUs. Finally, the number of non-allocated spectrum portions measures the overall percentage of spectrum deficit *(L_{def})*.

Table 4. Fixed values of parameters

PARAMETERS	VALUE
Size of a Spectrum portion (*B*)	3.75 MHz
Distribution of SU (λ_s)	5
Distribution of PU (λ_p)	{3, 4, 5}
Total simulation runs (r_N)	10
Elapsed simulation time (T_E)	90 minutes
PUs' mean spectrum usage	40%
Maximum number of PUs (N_{pmax})	50
Maximum number of SUs (N_{smax})	50

B. Obtained Results

Firstly, we show the histograms of PUs' spectrum usage at several values of λ_p in Figure 8. All three histograms depict the spectrum usage probability of PUs at different instances of time. At early stages, this probability is high, but later, the spectrum is mostly unutilized and thus, the SUs can have a higher percentage of successful spectrum sharing agreements during these periods.

Successful Agreements

We now plot the average number of successful spectrum sharing agreements between primary and secondary users. By a successful agreement, we mean that an SU has utilized the assigned spectrum according to its requirement. Formally for an agreement *k*,

$$k = \begin{cases} \text{Successful} & \text{when } \frac{t^{su}_{utilized}}{t^{su}_{requested}} = 1 \\ \text{Unsuccessful} & \text{otherwise} \end{cases} \qquad (10)$$

And if we denote a successful agreement by $k_{success}$, we can write the summation as:

$$A_{(s)} = \sum k_{success} \qquad (11)$$

---

[2] We have also verified our simulation with different values of elapsed time for several runs (i.e. r_N = 20, 30, 40, 50, …) and nevertheless the agents behaviors remain the same.

Where $A_{(s)}$ represents the number of successful agreements and $t^{su}_{utilized}$ and $t^{su}_{requested}$ are the time values for which spectrum is being utilized and requested by the secondary user. Of course, due to the cooperative nature of agents, the secondary user pays the agreed price (for its spectrum utilization) to the respected primary user.

Figures. 9(a) and 9(b) show three curves of A_s plotted using several numbers of SUs (N_s) at various time values. From the figures, the differences in A_s at $\lambda_p = 3$, 4 and 5 are clearly observable. For small N_s, the values of A_s are marginally the same, because most of the PUs are busy during the early time periods and it is difficult for SUs to find the required PUs with the available spectrum portions. When N_s reaches higher values with simulation time greater than 50 minutes, the distinction between A_s is much clearer. This distinction also shows that fewer successful agreements are made when $\lambda_p = 3$, because the number of available PUs are less. But, when we increase λ_p to 4 and 5, the values of A_s almost reach to their maximum positions. Furthermore, we can observe that for bigger values of λ_p, the cooperation between the primary and secondary users is still very effective, as more SUs have made successful spectrum sharing agreements.
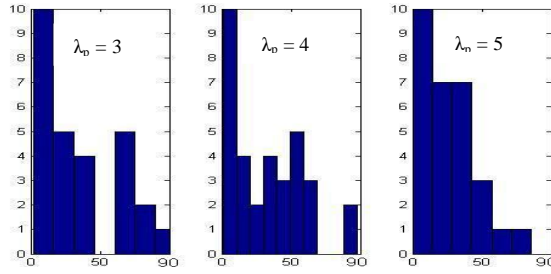
Figure 8. Primary users' spectrum utilization
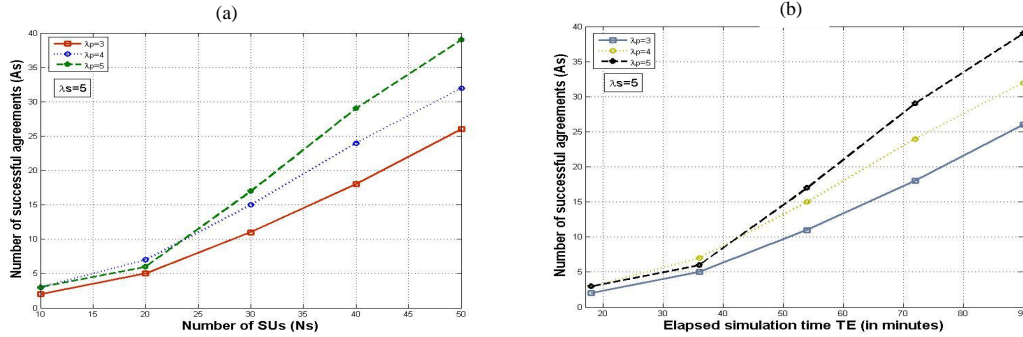
Figure 9. Successful agreements with (a) number of SUs and (b) elapsed simulation time

Percentage Utility

Figure 10 compares the average percentage utility of SUs at different time values. Particularly, the utilities are within 20 to 55% when $\lambda_p=3$, while they can reach 70% (80% respectively) when $\lambda_p=4$ ($\lambda_p=5$ resp.). The graph is in conjunction with Figures 8 and 9, where at time values greater than 50 minutes, the SUs can find more available PUs for spectrum sharing. Moreover, it is noticeable that the utility values can augment up to 80% (with $\lambda_p=5$) showing higher efficiency in terms of spectrum utilization. Nevertheless, complete satisfaction is not achieved, considering that the environment is ad-hoc and the PUs are hesitant to share their spectrum portions for longer time durations. In contrast, Figure 11 shows that the average percentage utility of PUs is almost 90% for all three values of λ_p, since the PUs are always less (or equal) in number compared to SUs and there is relatively a higher chance that they can easily find SUs to share their unutilized spectrum.

We now compare the average percentage utility values of primary and secondary users' achieved through simulations to the optimal value. Optimal value can be achieved when the average percentage utilities of primary and secondary users are fully satisfied (i.e. 100%). Figure 12 summarizes the results. We observe that the values achieved through experiments are very close to the optimal value and they can reach almost 90% showing good utility based performance of our approach.
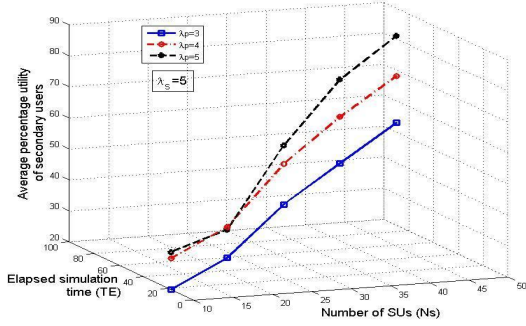
Figure 10. Percentage utility values of SUs at different time instances
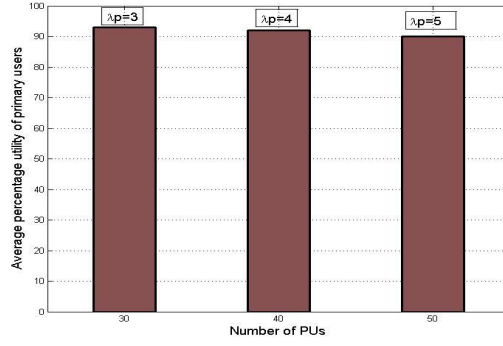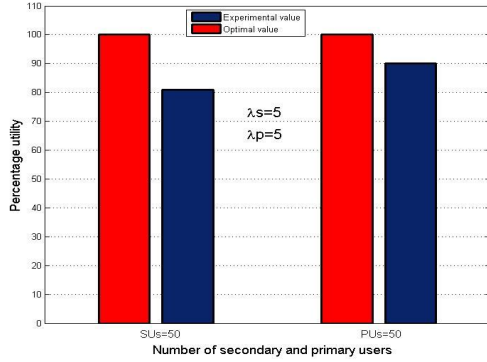

Figure 11. Percentage utility of PUs

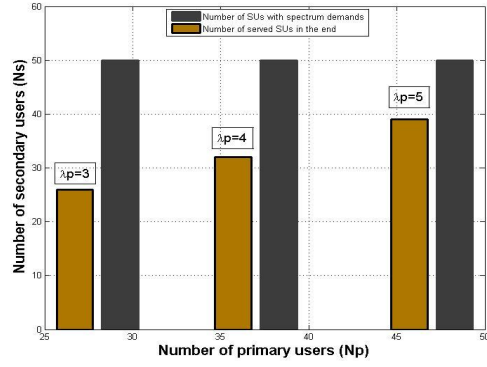Figure 12. Comparison to the optimal value

Figure 13. SUs with spectrum demands vs. number of served SUs

Another way of showing above results is to compare the number of SUs with spectrum demands and those which are served in the end. For this depiction, we fix the maximum number of SUs to 50 and observe the results for different values of PUs, in Figure 13. Served SUs are those which have completely obtained the required spectrum. In all the three comparisons, even though a large amount of SUs have been completely served, still a percentage of them remains unsuccessful. Thus, under ad-hoc situations, despite the primary and secondary users are equal in numbers, the results are not fully achieved.

Communication Cost

The number of cooperation messages for successful spectrum sharing agreements determines the average communication cost (M_{cost}). Formally,

$$M_{cost} = \frac{\sum_{h=1}^{A_{(s)}} m_h}{A_{(s)}} \qquad (12)$$

Where m_h is the number of messages sent and received for a successful agreement h such that $1 \leq h \leq A_{(s)}$.

Different values of M_{cost} along with several numbers of successful agreements are plotted in Figure 14. In the figure, the value of M_{cost} is initially 4, but it climbs to an average of 8 messages per agreement. This increasing pattern in M_{cost} is directly relational to number of PUs (N_p) i.e. when the available PUs in an SU's neighborhood are less in number, the message exchange between the users is not high. Similarly, when N_p increases, the SUs can find more PUs in their neighborhood causing M_{cost} to increase. However, the values of M_{cost} are not very high, showing the communication efficiency of our approach.
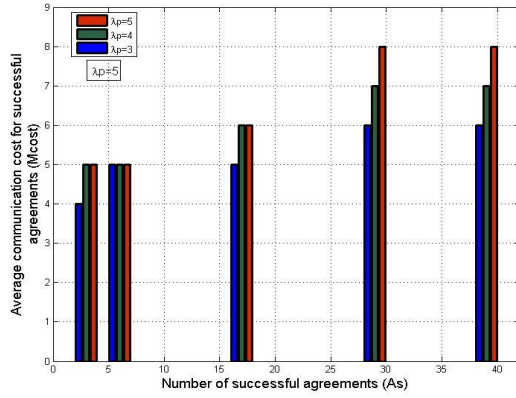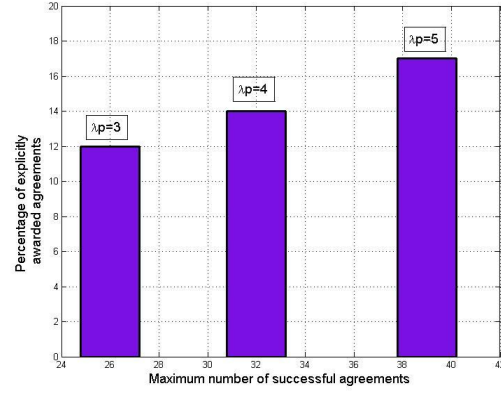
Figure 14. Communication cost

Figure 15. Explicit spectrum sharing agreements

Explicitly awarded spectrum sharing agreements

Another important novelty of our approach is the ability of PUs to make explicit agreements with the neighboring SUs. The necessity of explicit agreements arrives in situations where the corresponding PU's spectrum portion is in utilization at the time of *CfP* reception and therefore its current spectrum status is set to "busy". In our approach, the PUs can still send their explicit *proposals* to SUs when they get unoccupied. Each PU maintains a list of recent *CfPs* (stored in its cache) and accordingly, it sends the *proposal* to the most suitable one whose deadline is not yet expired. In relation, Figure 15 delineates the percentage of explicitly awarded spectrum sharing agreements. It is clearly envisaged that almost 10 to 20% of the agreements have been explicitly awarded by the PUs.

Spectrum Deficit

Figure 16 depicts the percentage spectrum deficit (S_{def}). One important reason to draw this graph is to see whether the values of S_{def} increase with large numbers of agents. This observation also shows the performance degradation of the whole system with an increased amount of traffic. In the corresponding figure, initially the values of S_{def} are within 60-80%, because at these stages most of the PUs are occupied. Later, the values of S_{def} continue to decrease on a steady pace, as more PUs become available. Still, there is not a rapid degradation in overall system performance, showing the efficiency of our proposed solution.
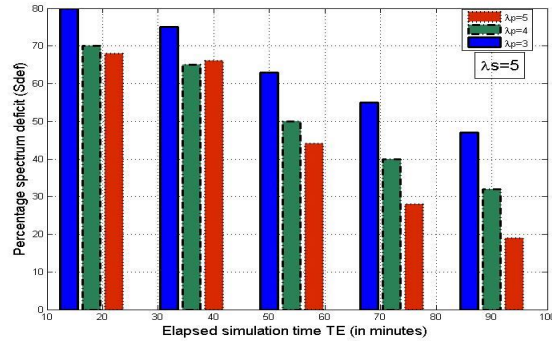

Figure 16. Spectrum deficit

Comparison with other approaches

Finally, we compare our solution to greedy algorithm [14], cooperative local bargaining [21] and dynamic spectrum access protocol (DSAP) [31]. In greedy algorithm most of the PUs are self-interested and they are hesitant

to share the available spectrum, until they get the highest offer maximizing their individual utility. Local bargaining is cooperative where the users exchange messages and they self-organize into bargaining groups for spectrum sharing. DSAP is based on the concept of centralized licensed server which is responsible for leasing spectrum to the requesting users. Thus, we compare our approach to three different solutions showing greedy, cooperative and centralized behaviors, respectively. All the solutions are implemented under our ad-hoc scenario and the users are deployed according to Poisson processes with $\lambda_p = 5$ and $\lambda_s = 5$.

Figures 17(a) and 17(b) compare the average achieved utility values by the SUs and the communication cost associated with successful agreements, respectively. We notice that the utility values and communication cost for greedy approach is very high considering the selfish nature of the neighboring PUs. Consequently, most of the time SUs receive unsatisfactory *proposals* and several messages are wasted. The local bargaining approach is limited to one-to-one bargaining, where an SU can bargain with only one PU at a time. Thus, local bargaining achieves similar utility to greedy approach at a reduced communication cost. Considering DSAP, the users exchange several kinds of messages including discover, offer, request, acknowledge, reclaim, etc making its communication cost higher. However, due to its centralized nature where several users cooperate with only one primary user for their spectrum assignments, DSAP behaves similar to our approach in terms of communication cost with lesser utility. On the other hand, our approach shows improvements in utility values compare to all three approaches incurring slightly higher communication cost than local bargaining.
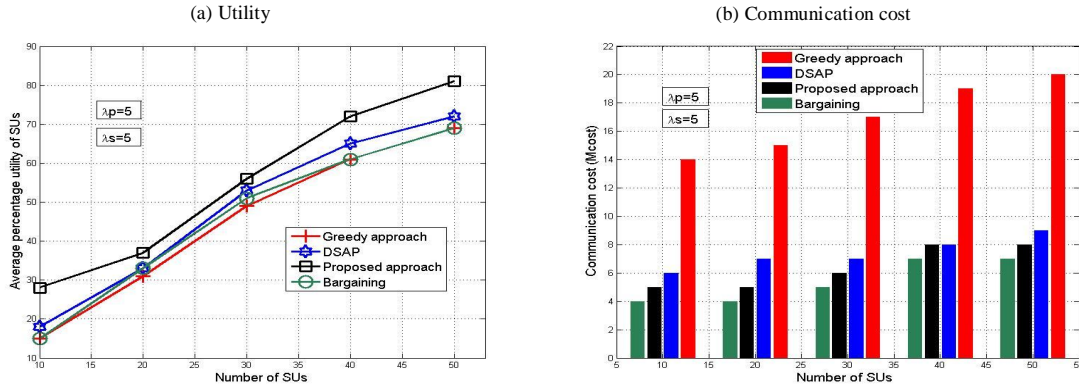

Figure 17. Comparison with greedy, local bargaining and DSAP approaches

VII. CONCLUSION AND FUTURE PERSPECTIVES

In this paper, a cooperative approach to enable dynamic spectrum sharing is presented. The solution is based on multiagent cooperation where the primary and secondary users exchange bilateral messages to make spectrum sharing agreements in an ad-hoc manner. The behavioral modeling of our approach based on Petri nets proves its efficiency for dynamic and distributed environments. Experimental results show that, compare to greedy, bargaining and centralized solutions, our cooperative solution works very effectively without having higher communication cost.

One important area of our future research corresponds to unlicensed spectrum sharing where the spectrum can be viewed as an open "pool" and all the devices are of equal rights and priorities i.e. none of the devices have exclusive license for spectrum usage. The unlicensed devices (using multiagent system) can form necessary coalitions in order to maximize spectrum usage and minimize interferences. We are also planning to develop mathematical models for our approach to analyze its working with large numbers of agents.

ACKNOWLEDGEMENT

REFERENCES

[1] A. Tonmukayakul, & M. B. Weiss, M. B. H, "An agent-based model for secondary use of radio spectrum," in *Proc. IEEE New frontiers Dynamic Spectrum Access Networks*, Baltimore, 2005, pp. 467-475.

[2] C. Ghosh, D. P. Agrawal and M. B. Rao, "Channel capacity maximization in cooperative cognitive radio networks using game theory," *ACM Mobile Computing and Communications*, vol. 13, pp. 2-13, 2009.

[3] C. Qi, J. Wang, and S. Li, "Weighted-clustering cooperative spectrum sensing in cognitive radio context," in *Proc. International Conference on Communications and Mobile Computing*, 2009, pp. 102-106.

[4] Clancy, T. C. (2006). *Dynamic spectrum access in cognitive radio networks*. Ph. D. dissertation, University of Maryland, MD, USA.

[5] D. I. Kim, L. B. Le, and E. Hossain, "Joint rate and power allocation for cognitive radios in dynamic spectrum access environment," *IEEE Trans. Wireless Communications,* vol. 7, pp. 5517–5527, Dec. 2008.

[6] D. Niyato, and E. Hossain, "Competitive pricing for spectrum sharing in cognitive radio networks: dynamic game, inefficiency of Nash equilibrium, and collusion," *IEEE Trans. Selected Areas in Communications*, vol. 308, pp. 192-202, 2008.

[7] Durfee, E. H., and Lesser, V. "Negotiating task decomposition and allocation using partial global planning," *Distributed Artificial Intelligence Journal*, Volume 2, pp. 229–244, 1989.

[8] F.–S. Hsieh, "Developing cooperation mechanism for multi-agent systems with Petri nets**."** *Engineering Applications of Artificial Intelligence*, vol. 22, pp. 616-627, June 2009.

[9] G. Weib, and S. Sen, "Adaptation and Learning in Multiagent Systems," Springer Verlag, Berlin, 1996.

[10] H. J. Kloeck, and F. Jondra, "Multi-agent radio resource allocation," *ACM Mobile Networks and Applications*, vol. 11, pp. 813-824, December 2006.

[11] H. Li, "Multi-agent Q-Learning of channel selection in multi-user cognitive radio systems: A two by two case," in *Proc. IEEE International Conf. Systems, Man and Cybernetics*, San Antonio, TX, 2009, pp. 1395-1422.

[12] H. Peng, Zheng, and B. Y. Zhao, "Utilization and fairness in spectrum assignment for opportunistic spectrum access," *ACM/Springer Mobile Networks and Applications*, vol. 11, pp. 555 – 576, Aug. 2006.

[13] H. Wang, H. Qin, and L. Zhu, "A survey on MAC protocols for opportunistic spectrum access in cognitive radio networks," in *Proc. Int. Conf. Computer Science and Software Engineering*, 2008, pp. 214–218.

[14] H. Zheng, and C. Peng, "Collaboration and fairness in opportunistic spectrum access," in *Proc. 40th Annu. IEEE International Conf. Communications*, Seoul, 2005, pp. 3132–3136.

[15] H.-P. Shiang, and M. Van der Schaar, "Delay-sensitive resource management in multi-hop cognitive radio networks," in *Proc. IEEE New frontiers Dynamic Spectrum Access Networks*, Chicago, 2008, pp. 14-17.

[16] J- M. Proth, and X. Xie, "Petri nets: A tool for design and management of manufacturing systems," John Wiley & Sons, West Sussex, 1996.

[17] J. F. C. Kingman, "Poisson Processes," Clarendon Press, England, 1993.

[18] J. Mitola, "Cognitive radio: an integrated agent architecture for software defined radio," Ph.D Thesis, KTH Royal Institute of Technology, Sweden, 2000.

[19] K. Hamdi, and K. B. Letaief, "Cooperative communications for cognitive radio networks," in *Proc. 8th Annu. Postgraduate Symposium. Convergence of Telecommunications, Networking and Broadcasting,* Liverpool, 2007, pp. 878-893.

[20] K. P. Sycara, "Multiagent systems," *Artificial Intelligence Magazine*, vol. 19, pp. 79-92, 1998.

[21] L. Cao, and H. Zheng, "Distributed spectrum allocation via local bargaining," in *Proc. 2nd Annu. IEEE Conference on Sensor and Ad Hoc Communications and Networks*, Santa Clara, 2005, pp. 475-486.

[22] M. A. McHenry, "NSF spectrum occupancy measurements project summary," August 2005.

[23] M. Liu, S. H. A. Ahmad and Y. Wu, "Congestion games with resource reuse and applications in spectrum sharing,**"** *International Conf. Game Theory for Networks,* Istanbul, 2009, pp. 171-179.

[24] M. Mchenry, "Spectrum white space measurements," New America Foundation Broadband Forum (June 2003).

[25] R. G. Smith, "The contract net protocol: High-level communication and control in a distributed problem solver," *IEEE Trans. Comput*, vol. 29, pp. 1104–1113, December 1980.

[26] S. Kumar, V. S. Raghavan, and J. Deng, "Medium access control protocols for ad-hoc wireless networks: a survey," *International Journal of Ad-hoc Networks*, vol. 4, pp. 326-358, May 2006.

[27] S. Sengupta, M. Chatterjee, and K. Kwiat, "Dynamic spectrum access in cognitive radio based tactical networks," in *Proc. Annu. IEEE Conference Wireless Communications & Networking*, Budapest, 2009, pp. 1343-1348.

[28] T. Murata, "Petri nets: Properties, analysis and applications," *Proceedings of the IEEE*, vol. 77, pp. 541–580, April 1989.

[29] U. Mir, L. Merghem-Boulahia, and D. Gaïti, "A cooperative multiagent based spectrum sharing". in *Proc. 6th Annu. Advanced International Conference on Telecommunications,* Barcelona, 2010.

[30] U. Mir, L. Merghem-Boulahia, and D. Gaïti, "Multiagent based spectrum sharing using Petri nets," in *Proc. International Workshop on Artificial Intelligence and Distributed Systems at PAAMS'10,* Salamanca, 2010.

[31] V. Brik, E. Rozner, S. Banarjee, P. Bahl, "DSAP: a Protocol for Coordinated Spectrum Access," in: *Proc. IEEE New frontiers Dynamic Spectrum Access Networks*, Baltimore, 2005, pp. 611–614.

[32] X. Jiang, H. Ivan, and R. Anita, "Cognitive radio resource management using multi-agent systems," in *Proc. IEEE Consumer Communications and Networking Conference*, Las Vegas, 2007, pp. 1123-1127.

[33] Y. Xing, "Cognitive radio networks: learning, games and optimization," Ph.D. Thesis, Stevens Institute of Technology, USA, 2006.

[34] Z. Wie, R. K. Mallik, and K. B. Letaief, "Optimization of Cooperative Spectrum Sensing with Energy Detection in Cognitive Radio Networks," *IEEE Trans. Wireless Communications*, vol. 8, pp. 5761-5766, 2009.

[35] Z. Liu, "Performance Analysis Of Stochastic Timed Petri Nets using Linear Programming Approach," *IEEE Trans: Software Engineering*, vol. 24, pp. 1014-1030, 1998.

[36] http://jade.tilab.com/