ISSN: 2229-6948(ONLINE)
DOI: 10.21917/ijct.2012.0088

PREDICTION BASED CHANNEL-HOPPING ALGORITHM FOR RENDEZVOUS IN COGNITIVE RADIO NETWORKS

Dhananjay Kumar[1], A. Magesh[2] and U. Mohammed Hussain[3]

Department of Information Technology, Madras Institute of Technology, Anna University Chennai, India

E-mail: [1]dhananjay@annauniv.edu, [2]magster_magesh@yahoo.co.in and [3]hussainub@gmail.com

Abstract

Most common works for rendezvous in cognitive radio networks deal only with two user scenarios involving two secondary users and variable primary users and aim at reducing the time-to-rendezvous. A common control channel for the establishment of communication is not considered and hence the work comes under the category of 'Blind Rendezvous'. Our work deal with multi-user scenario and provides a methodology for the users to find each other in the very first time slot spent for rendezvous or otherwise called the first-attempt-rendezvous. The secondary users make use of the history of past communications to enable them to predict the frequency channel that the user expects the rendezvous user to be. Our approach prevents greedy decision making between the users involved by the use of a cut-off time period for attempting rendezvous. Simulation results show that the time-to-rendezvous (TTR) is greatly reduced upon comparison with other popular rendezvous algorithms.

Keywords:

Cognitive Radio, Rendezvous, Time to Rendezvous, First Attempt Rendezvous, Channel Hopping

1. INTRODUCTION

In a cognitive radio networks (CRN), the term rendezvous means establishment of a communication link between two secondary users (SU) and maintain the same when the channel availability changes [1]. In the context of dynamic spectrum access (DSA) in CRN, rendezvous refers to the ability of two secondary users to find an available channel for communication without providing any hindrance to the primary users to whom the spectrum is licensed to. Upon the presence of a primary user (PU) the radios in rendezvous must find an alternate available channel to continue the communication.

There can be a large number of vacant channels at any point in time and the secondary users may rendezvous in any of those vacant channels, provided the PU is not active in the chosen channel at the given time [2]. To achieve rendezvous the two radios or nodes need to select the same channel for communication and one of the radios need to be sensing the medium while the other is transmitting a beacon to establish the handshake required for the commencement of rendezvous [3]. The control channel must also be free from primary user activity and as well as from rendezvous between other pairs of radios in the multi-user (multiple secondary users) network.

When a common control channel for the establishment of communication is not considered such types of rendezvous are categorized as blind rendezvous [4]. The protocol that is used to achieve the handshake between the users is adopted from link rendezvous protocol [5]-[6]. The users follow the protocol to complete the rendezvous handshake and to implement it one of the radios must be in sensing state while the other is sending a beacon signal.

In blind rendezvous, inherently all vacant channels are potentially available for the exchange of control and data [7]. It now becomes the responsibility of the radios to determine the availability of the channels to establish a link in any one of the available channels. When this is extended to a multi-user case then there is even competition between the SUs during rendezvous for channel access. This increases the complexity in finding a vacant channel to rendezvous.

The proposed frequency prediction algorithm uses the history of past communications to predict the frequency that ensure rendezvous between the SUs in the very first time slot. If history information is unavailable, we propose and analyse a modified version of modular clock algorithm [2]. To establish the efficiency of the proposed algorithm a comparative analysis and simulation with the original modular clock algorithm is presented here.

The remainder of the paper is organized as follows. Section 2 describes the survey of earlier works in the field of rendezvous of cognitive radios. Section 3 deals with the system design and Section 4 describes in detail the proposed algorithm. Results and evaluation of the proposed algorithm is mentioned in Section 5, and Section 6 concludes the work.

2. RELATED WORK

Neighbor discovery [8] is referred to as the process wherein the radios attempt to arrive on the same frequency channel to begin communication. The methodology of neighbor discovery through random [2] channel selection is one of the simplest of algorithms wherein the radio chooses one of the N available channels with a selection probability of 1/N. Modular based rendezvous [9], which aims to achieve rendezvous through modular operations making the radio move linearly from channel to channel with a defined hop distance. A modulus operation is performed each time to ensure that the radio stays and operates within the frequency channels defined or otherwise called the operating frequency. So there is a need for the radios to know the total number of frequency channels available, for the modular operation to be performed.

The other common type of rendezvous algorithm is jump-stay algorithms [10] wherein the radios are either in jump mode or in stay mode which enables the SU in jump mode to achieve rendezvous with the SU in stay mode. Jump-stay algorithms do not require clock synchronization between the SUs involved. Sequence based rendezvous [11] are mainly based on a sequence generation function to generate sequences with finite number of guaranteed intersections. But it is essential for the rendezvous channels to be unrestricted. Restricting the rendezvous channels to the sequence generated increases the risk of failures due to the presence of primary user activity. Other works in rendezvous had their basis from concepts of game theory [12].

3. SYSTEM MODEL

For our current study, we consider a CRN consisting of K secondary users (K ≥ 2), who coexist with one or more primary users. The primary users hold the license to use the spectrum, whereas the secondary users may make use of the spectrum when it is not utilized by the primary user. The secondary user must never compete with the primary user and must relinquish the channel it is using when the primary user needs it. We consider the spectrum to be divided into M non-overlapping orthogonal channels (M > 1) [13]. We also assume a uniform labelling methodology where these channels are labelled as 1, 2, 3, …, M and the labelling is well-known to every user in the network [13].

The whole set of potential available channels for the users is denoted by C={c1,….,cM}, in which ci denotes the ith channel (i =1, 2, …. , M). The set of available channels Ci is a subset of the total number of channels observed, Ci ⊆ C. The rest of the channels may be used by the primary user. The set Ci changes every timeslot based upon the primary user activity.

Our proposed model makes an assumption that all radios sense the same number of open or available channels. It also includes the assumption that all the radios observe the same channel labelling.

3.1 TIME SLOTS IN RENDEZVOUS

We assume time to be slotted. Each radio may change channels between timeslots to search for other radios, but cannot change channels in between timeslots. During each timeslot, a radio begins by sensing the medium for the presence of PU activity. If it does not sense the presence of any user, it will transmit a beacon. The beacon transmission is followed by a listening period, during when it waits for a response from the opposite party. Fig.1 shows the constituents of a timeslot.

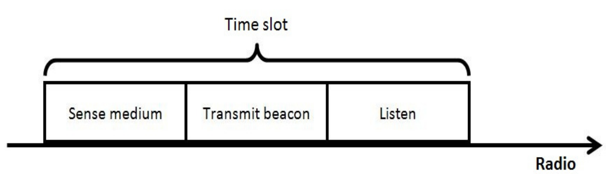

Fig.1. Single Time slot

3.2 BASIC ARCHITECTURE

The secondary user radios passage linearly from channel to channel. Each radio stays on each channel for defined number of timeslots before it moves to the subsequent frequency channel.

3.2.1 Stay Value or Stay ID:

Each radio is assigned with a stay value or called the stay ID. It defines the number of timeslots that each radio has to spend on each frequency channel before moving on to the next subsequent channel. This stay value is derived from the unique number that is assigned to each secondary user. All SUs involved will be able to compute the stay value of any other SU. For example if the stay value of a SU is 3 then the SU spends 3 timeslots on every frequency channel before hopping to the subsequent one. This scenario is described in Fig.2.

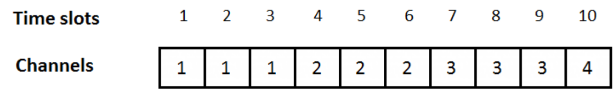

Fig.2. Frequency hopping in rendezvous in the absence of PU activity

There might be a case where a primary user wants to use the channel occupied by the SU. In this case the SU moves to the subsequent channel and the remaining timeslots that has to be spent on the previous frequency channel is carried forward. Fig.3 showcases this scenario wherein the PU intercepts the SU at the 4th timeslot. So the SU moves to the subsequent frequency channel which is 3 and spends 5 (3+2) timeslots on that channel.

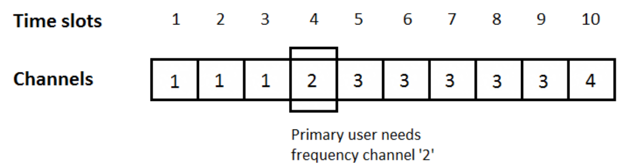

Fig.3. Frequency hopping in rendezvous in presence of primary user activity

3.3 SECONDARY USER BEHAVIOUR

The secondary user in the network is expected to be in the stay mode sensing the medium and trying to receive transmission beacon from other SUs.

The SUs stays on each frequency channel for the stay ID number of timeslots and moves on to the next subsequent channel linearly. When the SU reaches the end of the working frequency a modulus operation performed takes the SU to the beginning frequency channel of the working frequency [13]. So there is a necessity to know the total number of channels present for this operation to take place. During this phase the SUs will be sensing the medium for beacons from other SUs. The secondary user follows the above mentioned method till it is in stay mode. When the user acquires motivation to achieve rendezvous with another SU, then the SU saves its position and the current time and moves to another frequency position to establish communication. Now, the radio is in the jump mode. When the rendezvous is complete it uses the saved information to compute the frequency channel and tunes to it.

3.4 HISTORY STORED IN SECONDARY USERS

The SU maintains information required for the prediction of frequencies to assist rendezvous. Each SU has a store of history of previous rendezvous that has occurred between every other SU in the network. The information that it has in store for every user in the network are,

a) Time when the previous rendezvous has occurred.
b) Frequency channel where the previous rendezvous has taken place.

3.5 RENDEZVOUS CUT-OFF

Since multiple secondary users are involved, greedy strategies need to be prevented. The SUs necessarily work such that the performance of the group is maximized [14]. So there is a rendezvous cut-off time for every SU. This is the total number of timeslots that the user should spend in trying to achieve rendezvous. If the user is unable to achieve rendezvous within this time period then the SU should return to stay state to allow other users trying to achieve rendezvous with it. The SU shall try rendezvous with the same user again after a particular time period.

3.6 FIRST ATTEMPT RENDEZVOUS

Usually rendezvous between radios take a few timeslots to be established. But if the SU is able to achieve rendezvous in the very first time slot that it spends then it is termed as first attempt rendezvous. Our algorithm enables the SUs to achieve first attempt rendezvous.

4. FREQUENCY PREDICTION ALGORITHM

The system allows every user to predict the frequency channel where the rendezvous user would be in. The other user needs to be in stay state for prediction to work precisely. If that user has hopped to some other channel then first time rendezvous shall not occur. History of previous communication and the stay value of each user enable the SUs to correctly predict the frequency channel the rendezvous user would be in. The algorithm for the frequency prediction is given in Algorithm 4.1.

This assumes that the history information is available for every pair of users trying to achieve rendezvous. Since the channel hopping of the SUs is linear and based on the stay value of the user, other SUs in the network can predict the frequency channel it would be at present assuming the previous rendezvous and the frequency channel where the rendezvous has occurred is known. Before attempting rendezvous the SU saves its current time, *save time* and its current frequency channel *save post* to re-calculate its frequency once the rendezvous is complete. The number of timeslots since the previous rendezvous with user i can be estimated by analogy. Let t_{slot} be the duration of each timeslot and t_{curr} be the present time and t_i be the time when the previous rendezvous has occurred with radio i. If t_{diff} represents the number of timeslots since the previous rendezvous between the pair of SUs, this can be given by,

$$t_{diff} = (t_{curr} - t_i) / t_{slot} \tag{1}$$

The frequency displacement can be calculated from the time difference t_{diff} and stay value of the user *stay_val* is the stay value of the rendezvous user. Since each SU stays on each frequency channel by a *stay_val* (number of timeslots), the SU predicts the rendezvous SU to be f_{disp} frequency channels forward linearly and it is expressed as,

$$f_{disp} = (t_{diff} \; / \; stay\_val) \tag{2}$$

The SU predicts the frequency channel where the user would be at present by using f_{disp} calculated in Eq.(2) and f_i which is the frequency where the previous rendezvous has occurred with radio i. So the predicted frequency channel for the SU is represented by f_{curr} and is given by,

$$f_{curr} = (f_i + f_{disp}) \bmod m_i \tag{3}$$

where m_i is the total number of channels in the operating frequency. The SU now tunes to f_{curr} and tries to achieve rendezvous. A modular operation is performed with the total number of channels in the operating frequency (m_i) which prevents the user from hopping to undesired channels.

Once the rendezvous is complete, the SU re-calculates the frequency it needs to be using the saved time and saved frequency. The number of timeslots spent during rendezvous is found using,

$$t_k = (t_{curr} - save\_time) / t_{slot} \tag{4}$$

where *save_time* is the time when SU started to attempt rendezvous.

Since the frequency hopping of the radios is linear the radios re-calculates its frequency using the saved frequency and the *stay_ID* of the user. The user now move back to f_{curr} which is the re-calculated frequency of the user using,

$$f_{curr} = save\_pos + t_k \; / stay\_ID[a] \tag{5}$$

where *stay_ID*[*a*] is the stay value of the user attempting rendezvous with "*a*" number of time slot the SU stay at a particular frequency.

If rendezvous is not achieved in the first time slot the radio tries to rendezvous till the cut-off time is reached. The cut-of time is taken as twice the ID of the rendezvous user number of time slots since the start of rendezvous. It searches the next *stay_ID* number of frequency channels twice during the cut-off period. The searching process during rendezvous during time *t=0 to cutoff* need to be computed. If $j(t)$ holds the frequency channel during the time period t of attempt to rendezvous it can be given by,

$$j(t) = (f_{curr} + (t \bmod stay\_ID)) \bmod (m_i) \tag{6}$$

ALGORITHM 4.1

BEGIN

LET i be the rendezvous user

OBSERVE m_i, the number of channels within the operating frequency

RETREIVE $t_i = time\_history_i$

$f_i = freq\_history_i$

$stay\_val = stay\_ID_i$

$cutoff = save\_val * 2$

$t_{diff} = (t_{curr} - t_i) / t_{slot}$

$f_{disp} = floor\ (t_{diff} / save\_val)$

$save\_pos = f_{curr}$

$save\_time = t_{curr}$

$f_{curr} = (f_i + f_{disp})$ mod (m_i)

WHILE attempting rendezvous

FOR t = 0 to *cutoff* do

$j(t+1) = (f_{curr} + t$ mod $stay\_val$ () mod (m_i)

IF $j(t+1)$ has no primary user activity

THEN
c = $c_{j(t+1)}$
attempt rendezvous on channel c
END IF
END FOR
END WHILE
$t_{diff} = (t_{curr} - save\_time) / t_{slot}$
$f_{curr} = save\_pos + t_{diff} / stay\_ID[a]$
END

The next scenario deals with the situation wherein the SU does not possess any history information regarding the secondary user with which it is trying to achieve rendezvous. In this scenario, till the cut-off time is reached the SU uses the modular clock algorithm [2] to attempt rendezvous with any of the users present within the operating frequency. If rendezvous is successful then the SU checks if the user is the rendezvous user. If then, the communication takes place and the user updates the history values for subsequent rendezvous with the same SU. If the user is not the rendezvous user, then it attempts to retrieve the history information of the rendezvous user, if available. Else the user follows the algorithm trying to find the rendezvous SU or any alternate SU with the required history information. This continues till the cut-off period is reached (Algorithm 4.2).

ALGORITHM 4.2
BEGIN
LET *i* be the user whose history information is not available
OBSERVE m_i the number of channels within the operating frequency
CALCULATE *p*, the next largest prime to m_i
$j(0) = rand[0, m_i]j(0)$
WHILE not rendezvous do
CHOOSE *r* from [0, *p*] randomly
FOR $t = 0$ to $2p$ do
IF cut-off reached **THEN**
break jump mode
Re-calculate frequency to start stay mode
END IF
$j(t+1) = (j(t)+r)\mathrm{mod}(p)$
IF $j(t+1)<m_i$ **THEN**
$c = c_{j(t+1)}$
ELSE
$c = c_{(j(t+1)\mathrm{mod}(mi))}$
END IF
attempt rendezvous on channel *c*
IF rendezvous successful **THEN**
IF user is rendezvous user *i* **THEN**
Rendezvous
Store history values
ELSE
IF history information of *i* available
THEN
get history information
Predict frequency[algorithm 4.1]
END IF
END IF
END IF
END FOR
END WHILE
END

Fig.4 provides a complete flow of the proposed frequency prediction algorithm. If the past information is available then the user saves the current time and frequency and uses Algorithm 4.1 to predict the frequency. It senses the channel for primary user activity and upon the absence of the same, it checks for the presence of the rendezvous user. If the rendezvous user is present then the communication starts else the user searches the subsequent frequency channels till the cut-off time period is reached. Once the rendezvous is complete it re-calculates its frequency and tunes to it and resumes the stay mode. Under the absence of past data, modified modular clock algorithm (Algorithm 4.2) is used.

4.1 THE CIRCUMSTANCES THAT PREVENT THE FIRST ATTEMPT RENDEZVOUS

The activities of primary user as well as other SU is not always predictable. Furthermore, the other user may not be available for some reason when the rendezvous process is initiated. This outlines two scenarios when first attempt rendezvous fails.

4.1.1 User Activity in Predicted Channels:

Due to primary user activity the SUs are forced to move to the subsequent frequencies which cause the prediction to fail in the first attempt. The user activity also includes the secondary user activity i.e. other SUs might rendezvous in the selected channel. Even if the channel is unoccupied at that instant in time, primary user activity during the previous timeslots might play a role in the rendezvous user moving to subsequent channels.

4.1.2 User Attempting Rendezvous with Another User:

The other reason is that the user with whom the rendezvous is trying to be established with is in rendezvous with another radio in the multi-user network.

Let P_{pri} be the probability of primary user occupying channel *i* and P_{secB} be the probability of user B being in rendezvous with another SU which initially is expected to be in stay mode in channel i. So now the probability of first attempt rendezvous P_{far} occurring in channel i between the two SUs is given by,

$$P_{far} = (1-P_{pri})(1-P_{\sec B}) \quad (7)$$

4.2 MODIFIED MODULAR CLOCK ALGORITHM

The original performance analysis of modular clock algorithm was proposed for a 2-user case [2]. $P_{modclock}\{TTR<p\}$ gives the probability of modular clock algorithm succeeding

within p timeslot. Here the value of p is assigned to the cut-off value. The modified algorithm tries to achieve rendezvous with not 1-user but any user within the operating frequency. If K is the total number of secondary users in the network then the probability of achieving rendezvous for the modified modular clock algorithm is given in Eq.(8). $P_{modclock[i]}$ represents the probability of achieving rendezvous with user i within p time slots.

$$P_{modclock-modified} = \sum_{i=1}^{K} P_{modclock}[i] \quad (8)$$

The probability of finding one of K radios increases the performance of modular clock algorithm by K times which is computed as $P_{modclock\text{-}modified}$ in Eq.(8).

5. IMPLEMENTATION AND EVALUATION

The proposed system was simulated using MATLAB v.2009 and the results for various cases were recorded. The performance of frequency prediction algorithm was compared with the modular clock algorithm [2].

5.1 SIMULATION OF ALGORITHMS

The simulation process involved running the modular clock algorithm and the proposed algorithm against the same set of inputs for various degrees of primary user activity. It was observed that the proposed frequency prediction algorithm produces much lower average time to rendezvous compared with that of the modular clock algorithm (Fig.5). The TTR in modular clock algorithm keeps increasing as the number of channels increase whereas in the proposed algorithm the TTR shows a flat response for increasing number of channels.

Table.1 compares the average time to rendezvous for both the algorithms with respect to the number of channels available. A total number of 10 secondary users and a percentage of primary user channel occupancy of 35% were considered for the simulation. The table shows reduced average TTR for the proposed frequency prediction algorithm. For greater number of channels the average TTR remains the same for the proposed algorithm.

Upon running the same simulation for a several hundreds of iterations, a flat response is observed signifying the average time to rendezvous for the two algorithms as shown in Fig.6. During every iteration of execution, the average timeslots to rendezvous varies between 5 and 7 timeslots for frequency prediction algorithm and between 6 and 9 timeslots for the modular clock algorithm. Repeating the same for hundreds of iterations, the flat response signifies the average TTR of frequency prediction algorithm and modular clock algorithm to be at 6 and 9 timeslots respectively.

Table.1. Comparison of Time-to-rendezvous

No. of Channels	Time To Rendezvous	
	Modular Clock	Frequency Prediction
10	7.7332	5.9615
20	8.0957	6.1797
30	8.4775	6.8628
40	8.9154	6.8548

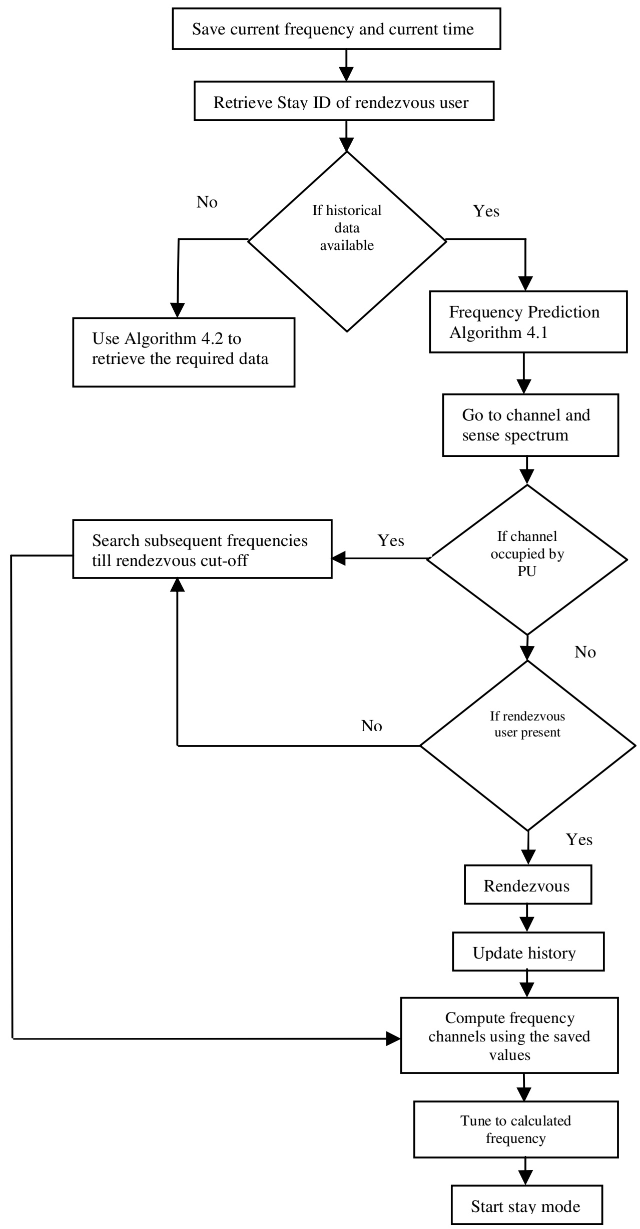

Fig.4. Control Flow of Prediction Algorithm

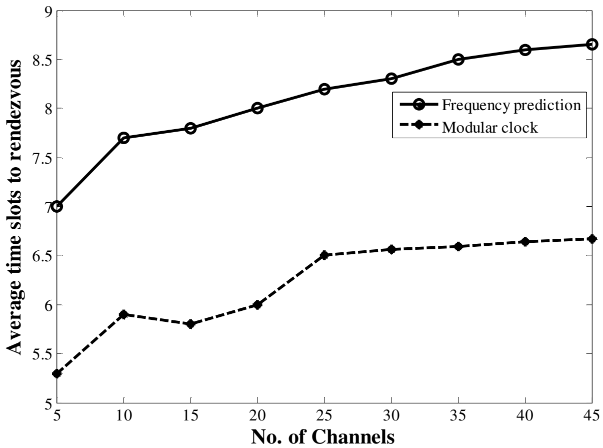

Fig.5. Modular Clock vs. Frequency prediction

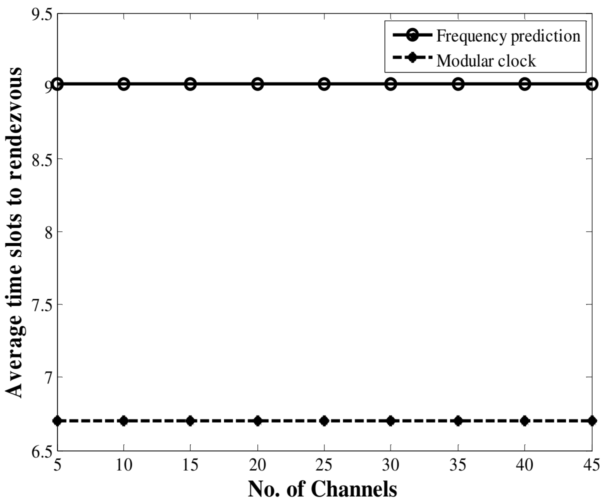

Fig.6. Modular Clock algorithm vs. Frequency prediction [2]

The maximum and average TTR for both the algorithms are compared in the bar chart in Fig.7. The maximum and the average TTR of the proposed frequency prediction algorithm are found to be at a reduced value upon comparison with the modular clock algorithm. The difference of values between the average and maximum TTR are very close to each other which substantiates the fact that maximum TTR is upper bounded for both the algorithms.

5.2 FIRST ATTEMPT RENDEZVOUS

In the simulation carried out primary user occupancy is set to 40% and the motivation probability of each SU to rendezvous with other SUs is set to 0.60. The tests were run to analyse the occurrence of first time rendezvous. It is observed that the first attempt rendezvous has occurred successfully in 36% of total rendezvous as listed in Fig.8. Reducing the primary user activity and motivation probability of the SUs, the occurrence of first attempt rendezvous will increase. The other successful rendezvous have occurred not in the first time slot but in further time slots spent for rendezvous.

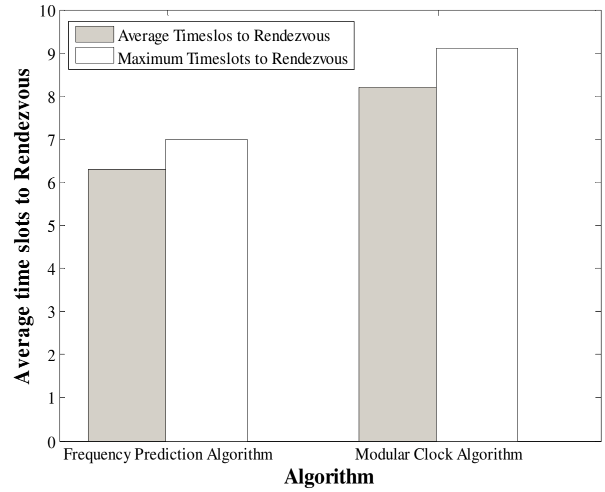

Fig.7. Modular Clock algorithm vs. Frequency prediction – Average and Maximum TTR

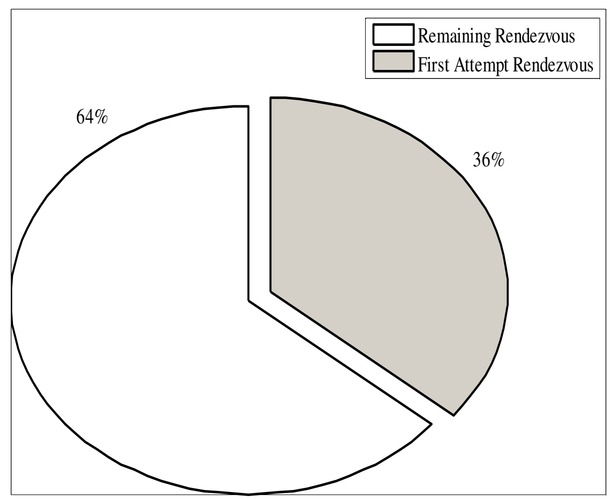

Fig.8. First attempt rendezvous in proposed algorithm

6. CONCLUSION

In this paper a prediction based model was presented which provides a methodology to enable the radios to achieve rendezvous in the very first time-slot that it spends trying to establish communication. It also prevents the multi-users involved in rendezvous from taking greedy decisions by the use of a cut-off time period. The frequency prediction algorithm assures first attempt rendezvous with a high degree of occurrence. It was observed that the first time rendezvous is increased when the motivation to perform rendezvous is not frequent between the SUs. In future we plan to extend the frequency prediction algorithm to a different model of rendezvous wherein the numbers of common channels observed between the two users are not the same. This requirement may demand additional intelligence needed to tackle the issue.

REFERENCES

[1] Y. Xiao and F. Hu, "*Cognitive Radio Networks*", CRC Press, 2008.

[2] N.C. Theis, R.W. Thomas and L.A. DaSilva, "Rendezvous for cognitive radios", *IEEE Transactions on Mobile Computing*, Vol. 10, No. 2, pp. 216-227, 2011.

[3] Y. Kondareddy, P. Agrawal and K. Sivalingam, "Cognitive radio network setup without a common control channel", *Proceedings of IEEE International Conference on Military Communications*, pp. 1-6, 2008.

[4] Bruce Fette, "*Cognitive radio technology*", Academic Press, 2006

[5] Horine and D. Turgut, "Link Rendezvous Protocol for Cognitive Radio Networks", *Proceedings of IEEE International Symposium on New Frontiers in Dynamic Spectrum Access Networks*, pp. 444-447, 2007.

[6] Horine and D. Turgut, "Performance Analysis of Link Rendezvous Protocol for Cognitive Radio Networks", *Proceedings of Second IEEE International Conference on Cognitive Radio Oriented Wireless Networks and Communications*, pp. 503-507, 2007.

[7] Y. Kondareddy, P. Agrawal and K. Sivalingam, "Cognitive radio network setup without a common control channel", *Proceedings of IEEE International Conference on Military Communications*, pp. 1-6, 2008.

[8] K. Balachandran and J.H. Kang, "Neighbor Discovery with Dynamic Spectrum Access in Adhoc Networks", *Proceedings of IEEE Conference on Vehicular Technology*, Vol. 2, pp. 512-517, 2006.

[9] Nicholas C. Theis, "*The Modular Clock Algorithm for Blind Rendezvous*", AFIT/GCS, 2009.

[10] Zhiyong Lin, Hai Liu, Xiaowen Chu and Yiu-Wing Leung, "Jump-Stay Based Channel-hopping Algorithm with Guaranteed Rendezvous for Cognitive Radio Networks", *IEEE International Conference on Computer Communications*, pp. 2444 - 2452, 2011.

[11] L. DaSilva and I. Guerreiro, "Sequence-based rendezvous for dynamic spectrum access", *Proceedings of IEEE International Symposium on New Frontiers in Dynamic Spectrum Access Networks*, pp. 1-7, 2008.

[12] S. Alpern and S. Gal, "*The Theory of Search Game and Rendezvous*", Springer, 2003.

[13] J.L. Arachchige, S. Venkatesan and N. Mittal, "An asynchronous neighbour discovery algorithm for cognitive radio networks", *Proceedings of IEEE International Symposium on New Frontiers in Dynamic Spectrum Access Networks*, pp. 1-5, 2008.

[14] Y. Xiao and F. Hu, "*Cognitive Radio Networks*", CRC Press, 2008.