



# Quality of Service Aware Channel Hopping Rendezvous in Cognitive Radio Networks for Internet of Things

Mohd Asifuddola<sup>1</sup> and Mohammad Ahsan Chishti<sup>1</sup>

<sup>1</sup>Department of Computer Science & Engineering, National Institute of Technology, Srinagar, Kashmir, India

<sup>#</sup>Polytechnic, Maulana Azad National Urdu University, India

<sup>\*</sup>Department of I.T, School of Engineering & Technology, Central University of Kashmir, Kashmir, India

Received 20 May 2020, Revised 25 Jul. 2020, Accepted 31 Jul. 2020, Published 8 Feb. 2021

**Abstract:** In the last few years, there has been significant growth in the number of smart wireless devices termed as Internet of Things (IoT). With the expansion in the number of IoT devices, the demand for wireless spectrum has also been increased tremendously while the spectrum is limited. Cognitive Radio Network (CRN) has emerged as a promising solution to the spectrum scarcity problem. In CRN, the unlicensed heterogeneous wireless devices have to dynamically make use of the spectrum which is not in use by the licensed users. However, establishing a common communication link between two heterogeneous wireless devices requires them to rendezvous on a commonly available channel which is a great challenge in a dynamic access environment. The time taken by two wireless devices to rendezvous on a commonly available channel is referred to as Time to Rendezvous (TTR). Depending upon Primary user (PU) temporal activity and required Quality of Service (QoS) by wireless devices, in this paper, we propose a novel QoS aware algorithm that can substantially reduce TTR of all the existing rendezvous algorithms. The algorithm takes into account the channel availability duration depending upon PU temporal activity and maximizes the utilization of these channels according to their availability duration. The algorithm works by reducing the number of channels to be hopped in the process of rendezvous. We develop a mathematical model to analyze the performance of the developed algorithm when applied to existing rendezvous algorithms. Both analytical and simulation results prove that the developed algorithm reduces the MTTR, increases the network throughput by providing required QoS to the wireless IoT devices and supports efficient utilization of wireless spectrum.

**Keywords:** Cognitive Radio Networks (CRN), Internet of Things (IoT), Common Control Channel (CCC), Rendezvous.

## 1. INTRODUCTION

Internet of Things (IoT) technology has emerged as a paradigm shift technology by making things or objects sense and react accordingly. IoT is a network of heterogeneous interconnected things embedded with different sensors and actuators that interact with each other using addressing methods based on standard communication protocols [1].

In the recent years, there has been tremendous increase in the number of IoT devices each requiring their share of wireless spectrum for connectivity. It has been found that within a smart IoT environment, while some portion of the spectrum is congested or over-utilized, some portion of the spectrum is not utilized efficiently or remains underutilized [2-3]. For instance, a smart home is equipped with various wireless IoT devices such as smart doors, smart refrigerator, smart fans, health monitoring sensors, smart electricity meter and have been pre-

assigned a certain portion of licensed wireless spectrum. While, smart electricity meter sensors and health monitoring sensors transmits data on a frequent basis, devices such as smart fans and smart doors needs to transmit occasionally which makes the spectrum assigned to these devices under-utilized. This temporally differentiated but spatially co-located spectrum utilization leads to inefficiency in the spectrum utilization. To increase the spectrum utilization, what could be done is to allocate the spectrum assigned to smart fans and smart doors to smart chairs or smart curtains whenever smart fans and doors are not transmitting. This efficient use of the spectrum is the logic behind the Cognitive Radio Network (CRN) [4].

Cognitive Radio Network (CRN) has emerged as a promising solution to the spectrum scarcity problem and to the problem of inefficient use of the wireless spectrum in IoT based application areas. CR technology enables sharing the wireless spectrum between licensed users or



primary users (PUs) and unlicensed users or secondary users (SUs) in an opportunistic manner called Dynamic Spectrum Access (DSA) [5]. Sharing of spectrum is done in such a manner that SUs access the licensed channels when the channels are not in use by PUs. Therefore, the licensed channels are used in an opportunistic manner providing performance guarantees to the PUs because PUs have higher priority as they are the licensed owners of the channel. The portion of the spectrum which is not in use temporally or spatially is referred to as spectrum holes [4]. SUs are allowed to use these spectrum holes for their transmission as long as they are available and must leave them for PUs on PU return.

#### A. Why CRN for IoT ?

The existing wireless communication protocols employed in IoT applications such as WLAN, Bluetooth and cellular network works on traditional static spectrum assignment policy and do not have any mechanism to sense the spectrum holes available. Due to the unavailability of this intelligent feature, most of the portion of the precious wireless spectrum remains unutilized. Also, there is no mechanism of sharing the spectrum dynamically in the current spectrum assignment strategies which is considered as a major drawback of the existing spectrum assignment schemes in the context of efficient utilization [4]. With the use of CRN in the communication protocols employed in IoT applications, the spectrum band allocated to a few number of devices can serve well typically a large number of devices by not asking for more spectrum to be allocated especially for them. Therefore, a small portion of the wireless spectrum either in the licensed cellular network or in the 2.4GHz industrial, scientific and medical (ISM) band can serve more number of devices which earlier was serving only few devices.

It is estimated that without assimilating the cognitive capability, IoT will not sustain potentials of growing challenges [6]. The ongoing research in the CR-IoT domain has been stressed upon providing solutions for the challenges in different domains [6-12]. It is estimated that the future IoT applications and devices will not only be smart or intelligent but will also have the feature of CRN implemented which enables them to adapt according to the environment in which they operate. This adaptation will pave the way for the system to be more efficient in terms of spectrum utilization primarily. Therefore, it is necessary to integrate CR with IoT for a more efficient overall system.

#### B. Rendezvous in Cognitive Radio Networks

In cognitive radio networks (CRNs), the set of available channels may differ for different SUs and owing to the dynamic behavior of channel availability, it is a tedious job to establish communication link between the SUs. The SUs in a CRN must be able to meet on a

channel which is commonly available to both of them for the purpose of communication and this process is referred to as rendezvous [13]. Rendezvous process includes exchanging various control information such as handshake and data communication. So far, mainly two methods have been employed in the existing works for the purpose of rendezvous; either with the help of a centralized controller which suggests a common control channel (CCC) [14] or by channel hopping (CH) [15]. Employing a CCC for the purpose of rendezvous eases out the process but have many drawbacks which makes it not much of a better option as compared to later. The drawbacks of using a single CCC includes congestion, jamming attack, single point of failure and so on. Due to these inefficiencies of CCC approach, CH technique is preferable in practice for the purpose of rendezvous which does not need any CCC [15]. As the name suggests, the idea behind channel hopping technique is to let SUs hop on its available channels in some predefined sequential order over time slots which is referred to as channel hopping sequence. Each of the SU follows its assigned CH sequence which may be different for each SU or may be same. After a complete period of CH, the sequence might remain same and the SU just repeats the sequence or the sequence might get shifted according to some predefined algorithm. The objective of such an algorithm is to make two SUs hop on to the same channel at the same time slot achieving rendezvous provided that channel is available to both of the SUs. Therefore, the prime objective in this approach is to generate CH sequences in such a way that the rendezvous is achieved without employing any centralized controller or CCC and this is why these systems are also referred to as 'blind rendezvous systems' [5]. Time to Rendezvous (TTR) is one of the main performance metric in CH approach towards rendezvous. TTR is defined as number of time slots required for the two SUs to achieve rendezvous. The worst case TTR is referred to as Maximum Time to Rendezvous (MTTR) for a CH sequence [13]. Therefore, most of the existing works have focused on generating CH sequences guaranteeing the rendezvous while lowering the MTTR. While these algorithms have been able to provide rendezvous solutions, they haven't considered the Quality of Service (QoS) required by the SUs. For example, in a smart home scenario, few applications might have low data rate requirement and can work with discontinuous connectivity like temperature sensors. On the other hand, few applications might have high data rate requirement and needs continuous connectivity like smart health monitoring and streaming which means the QoS level required is different for each application. Therefore, without taking into account the QoS level required by SUs in an IoT environment, the performance of the system cannot be improved just by focusing on lowering the MTTR and achieving rendezvous. The CH algorithms must take into account the QoS level required by each specific application and accordingly provide services to the applications in

minimum time. Also, it is an obvious fact that high QoS demanding applications need to rendezvous on channels which are available for longer durations because they are not delay tolerable and any discontinuity in connectivity would degrade their service. Similarly, Low QoS demanding applications need to rendezvous on channels which are available for small durations because discontinuity in connectivity would not hamper their operation as explained above. The existing works in this domain have focused on lowering the MTTR without considering this important parameter of QoS. Mere lowering the MTTR without considering the type of service required may result in an overall inefficient system. In CRN, for better utilization of available channels and to enhance the spectrum efficiency, all available channels regardless of their availability durations (less or more) must be taken for use accordingly [16].

Therefore, in this paper we have formulated a new method of achieving rendezvous between SUs taking into account the QoS required by SUs and channel availability durations. The data of high QoS demanding applications will be termed as High Priority (HP) data and the data of low QoS demanding applications will be termed as Low Priority (LP) data. In the proposed method, the CH sequence generation depends upon the type of application (HP or LP) and on the basis of availability duration of channels. The behaviour of channel usage by PUs is taken into consideration in this work to characterize the available channels according to their availability durations based on the information retrieved from the circular window  $|S|$  which keeps the record of availability duration of all channels and updates it based on periodic sensing results. This information is used to assign channels which are available for small and longer durations to LP and HP data respectively and accordingly CH sequence consists of only those channels which can serve the purpose. The motive behind this strategy has benefits which are as follows:

- All available channels regardless of their availability durations will be utilized according to the type of application they are being assigned to. This improves spectrum usage efficiency by utilizing all channels according to their availability durations.
- Required QoS will be provided to the IoT applications.
- MTTR will be lowered (improved) by employing the proposed technique for all the existing CH algorithms as will be proved in later sections. As opposed to the native rendezvous schemes which takes into account all available channels for generating the CH sequences, the proposed method considers only channels according to the required QoS.

- While native rendezvous schemes have throughput degradation, the proposed method performs better when duty cycle of PUs is high.

The remainder of this paper is organized as follows: Some state of the art channel hopping algorithms are reviewed in Section 2. System model and problem formulation is presented in Section 3. We propose QoS aware rendezvous algorithm and its theoretical analysis in Section 4. Mathematical simulation results are presented in Section 5. Finally, Section 6 concludes the paper.

## 2. RELATED WORK

Due to the limitations of a CCC for rendezvous, many researchers focus on CH rendezvous algorithms for establishing communication link in CRNs [7,13,17-30]. The above studies have shown to be the solutions for rendezvous in CRN. Without any pre-defined procedure or without using any of the CH algorithm, SUs hops on available channels in random order which may result in no rendezvous at all. Using any of the CH algorithm, SUs pre-computes the sequence on which they are going to hop. For example, if there are 5 channels available to a SU such as  $\{1, 2, 3, 4, 5\}$  and for these available channels, one of the CH sequence possible is  $[3, 5, 1, 2, 4]$  according to some CH algorithm. The SU hops among these channels according to the employed CH algorithm. Once the CH among the available channel completes, the sequence will be repeated or the hopping pattern might get shifted in the next hopping period as prescribed by the employed CH algorithm. When one cycle of hopping among all available channels completes, it is called hopping period. For the above example, the hopping period is 5. There might be different method of generation of CH sequence for sender and receiver or it can be same depending upon the CH algorithm employed. Also, the available channel set of sender might be different from the available channel set of receiver. This scenario is termed as '*Asymmetric*' model [13]. However, for rendezvous to be successful, there must be at least one commonly channel between sender and receiver. For example, as shown in Figure. 1, the available channel set of sender is  $\{2, 3, 6, 7, 8\}$  and available channel set of receiver is  $\{1, 4, 5, 7, 9\}$ . The rendezvous will be successful only when channel 7 aligns at the same time slot for the hopping sequences of sender and receiver as channel 7 in the only commonly available channel between sender and receiver. As shown in Figure. 1, hopping according to the CH sequence results in rendezvous in fourth time slot. In this case, sender and receiver both start their hopping pattern at the same time and hence termed as Synchronous. So, Figure. 1 shows an example of *Asymmetric Synchronous*

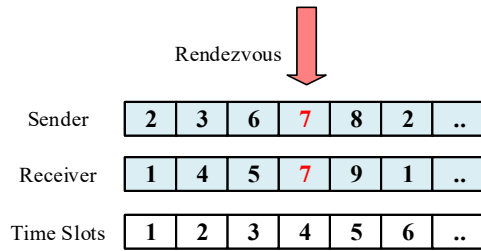


Figure. 1 Asymmetric Synchronous Rendezvous

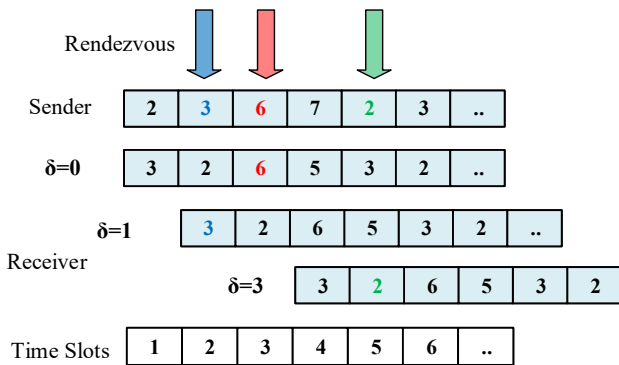


Figure. 2 Asymmetric Asynchronous Rendezvous

[31] CH algorithm for rendezvous. When the available channel set of sender and receiver are same, it is called 'Symmetric' model. For obvious reasons, the MTTR will be less in the symmetric model as all channels are common between sender and receiver. When sender and receiver start their CH sequence at different instants of time, it is referred to as 'Asynchronous' CH [31]. Figure 2 shows an example of Asynchronous CH algorithm. In this example, the available channel set of Sender is {2,3,6,7} and available channels set of receiver is {2,3,5,6}. We have taken a parameter ' $\delta$ ' which is the time drift between the hopping pattern of sender and receiver in the asynchronous case. When  $\delta = 0$ , it is synchronous. When  $\delta \neq 0$ , the CH is asynchronous. In Figure. 2, it is shown that for synchronous CH, the rendezvous occurs in the third time slot. When  $\delta = 1$ , rendezvous occurs in the first time slot as time slot is counted when both sender and receiver starts hopping. Also, when  $\delta = 3$ , rendezvous occurs in the second time slot. Similarly, when the available channel sets of sender and receiver are same, we will have either 'Symmetric Synchronous' or 'Symmetric Asynchronous' channel hopping [31]. In literature, some researchers have distinguished between Symmetric and Asymmetric on the basis of role assigned to the two nodes attempting rendezvous. When role is not pre assigned before starting CH, it is termed as Symmetric CH and when one node is assigned the role of sender and other is assigned the role of receiver, it is termed as Asymmetric CH [32]. In most of the existing CH algorithms, the

former definition of Asymmetric and Symmetric is more prevalent. Various CH algorithms have been proposed so far for each model.

### 3. SYSTEM MODEL

We have taken a CRN environment where several IoT devices co-exists each requiring different QoS. A basic network model is shown in Figure. 3. A smart home network has been taken for illustration with 6 devices operating in a CRN environment. The SU devices have cognitive capability and can configure according to the changing network conditions. The term cognitive capability refers to the ability of the devices to detect spectrum holes and also has a cognitive radio for data transfer. After detection of unutilized channels, the available channel set of each user might be different. There is no restriction on the type of model to be taken in the proposed approach and the approach is valid for Symmetric, Asymmetric, Synchronous and Asynchronous models. We have assumed that time is divided into slots and the slot boundary may nor may not be aligned of the CH sequences of Sender and Receiver. We have assumed no global system clock and thus there is no need of time synchronization as given in [20]. The duration of each slot is taken double of the time required for a complete handshake process between sender and receiver [20]. The benefit of considering no time synchronization is that the CH sequence can rendezvous even when the slot boundaries are not aligned and therefore any of them can start their CH process any time. Typically, 0.27 ms is the time required for rendezvous process [20]. Therefore, 0.54 ms is the duration of time slot in this model. In a CRN environment, each SU has the capability of sensing the spectrum holes [2-4, 16]. After the detection of available channels, SUs records this information in a circular window of size  $|S|$  [16].

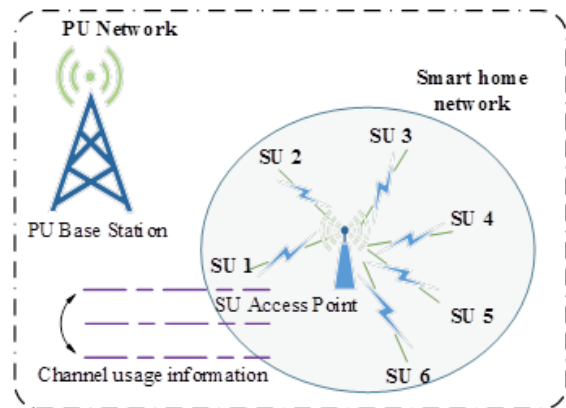


Figure. 3. Network Model

The recent sensing results are stored in this window and this record updates periodically. The instantaneous

average of the sensing results stored in  $|S|$  tells about the type of PU traffic and the average channel availability duration  $T_{ch}$ . Low  $T_{ch}$  indicates high PU traffic (heavy use) and high  $T_{ch}$  indicates low PU traffic (sparse use). The low and high channel availability duration serves well LP QoS and HP QoS applications respectively as explained in Section 1. Therefore, depending on the type of service a SU is serving, each SU sets some minimum channel duration requirement criteria which is termed as threshold duration  $T_{th}$ . It has been re-iterated that LP QoS applications can served well with channels having low  $T_{ch}$  but HP QoS applications need high  $T_{ch}$  channels being delay sensitive and having a time out period. The parameter  $T_{th}$  defines the minimum duration of channel availability below which the transmission gets interrupted. High QoS demanding applications (HP data) need channels which are available for duration of more than  $T_{th}$  and the communication of low QoS (LP data) demanding applications can be completed with channels which are available for short durations and several short durations channels can be assigned to LP data providing intermittent transmission. The motive behind this strategy is because HP data has certain time out period after which the packets loss probability increases whereas for LP data, there is no time out period of SU packets. Therefore, CH sequence of HP QoS applications consists of channels which are sparsely used and CH sequence of LP QoS applications consists of channels which are heavily used. The benefit of this approach is each type of channels; heavily used or sparsely used will be utilized accordingly providing efficient spectrum utilization and instead of total number of channels, CH sequences now consists of less number of channels which results in low MTTR as will be proved in Section 4.

Let  $U$  be the total number of SUs in the network and  $M$  be the set of total number of channels in the network indexed from  $[0$  to  $M-1]$ . Out of these  $M$  channels, each SU may have different set of available channels (Asymmetric) or may have same set of available channels (Symmetric). The proposed method of achieving rendezvous is valid for both. Let  $C_A$  be the set of available channels of SU  $A$  contains  $p$  channels such that  $p \leq M$ ,  $C_A = \{0$  to  $p-1\}$  and  $C_B$  be the set of available channels of SU  $B$  contains  $q$  channels such that  $q \leq M$ ,  $C_B = \{0$  to  $q-1\}$ .  $C_A$  and  $C_B$  are the subset of  $M$ . In the existing approaches, all channels of subset  $C_A$  and  $C_B$  takes part in the process of rendezvous. In the proposed approach, we are further reducing the size of sets  $C_A$  and  $C_B$  by taking into account the threshold value  $T_{th}$  and the reason for this will be clarified in Section 4. We have taken the probability of missed detection and probability of false alarm to be minimum such that it does not affect the result analysis to be presented in the later sections. For the process of rendezvous to be successful, two SUs must hop on the same channel (which is commonly

available to both of them) at the same time slot. There is no restriction of starting the CH sequence at the same time and any SU can start hopping any time (asynchronous). The aim is to generate the CH sequences for the SUs such that they should rendezvous in the least possible time to achieve the full benefits of available channels in a CRN environment where the availability of channels is highly dynamic. When the rendezvous is successful in some finite time, it is said to be guaranteed rendezvous. As mentioned in the previous section, researchers have provided some state of the art guaranteed rendezvous algorithms in the recent past and in this paper, we are determined to further lower the TTR and thus MTTR of these existing algorithms by the method adopted which is presented in the next section.

#### 4. PROPOSED ALGORITHM

Based on the motivation in the preceding section, we are presenting our algorithm in this section.

##### A. Basic Idea

In all the existing rendezvous algorithms, the total number of available channels is of importance as MTTR depends on the total number of channels. If a CRN system has  $M$  number of total available channels, then MTTR of each of the existing rendezvous algorithms depends on this value  $M$ . It is also to be observed from Section 2 that the less the value of  $M$ , less will be MTTR and vice versa. So the basic idea behind the proposed approach is to reduce this value of  $M$  on which MTTR depends. When  $M$  is reduced, MTTR will be reduced automatically. As explained in the previous section, the available channel set of sender and receiver may have different channels (Asymmetric) and the existing CH algorithms prescribe hopping on these channels by following a pre-defined CH sequence. All the existing algorithms consider all available channels for generating their hopping sequences. In the proposed approach, we are not taking all channels available to a SU for generating the CH sequence rather taking only application specific channels and omitting those channels which are not suitable for the required QoS. QoS specific channels have been discussed in detail in [16]. Therefore, in generating CH sequence, we are further reducing the size of the subset  $C_A$  and  $C_B$  based on the required QoS by SUs. For example, let the number of available channels of SU  $A$  be 10 and out of these 10 channels, 6 are having availability duration more than the set threshold value  $T_{th}$  and 4 channels have availability duration less than  $T_{th}$ . Now, if the SUs need to access available channels for their communication requirement and also if their required QoS is HP. In this situation, the SUs generate their CH sequences taking only those channels which have availability duration more than the threshold value  $T_{th}$ . The channels which have availability duration less than  $T_{th}$  will not be there in the CH sequence. This way, the channel set  $C_A$  is reduced to 6 now. Similarly, for the channel set  $C_B$ . Generalizing this



approach, let before applying this method, there were initially  $M$  channels and the required QoS is HP. Let channel availability duration is represented as  $T_{ch}$  and  $n$  be the number of channels which have availability duration less than  $T_{th}$ . Applying the proposed method, new set of channels for generating the CH sequence is  $Z$  where  $Z$  is given by

$$Z = M - n.(T_{ch} < T_{th}) \quad (1)$$

$Z$  represents new set of channels which is obtained after eliminating the channels which have  $T_{ch}$  less than  $T_{th}$  as the required QoS is HP. Similarly, when required QoS is LP, then the channels which have  $T_{ch}$  less than  $T_{th}$  will be considered for generating the CH sequence and the channels which have  $T_{ch}$  greater than  $T_{th}$  will be eliminated. So, when required QoS is LP, new set of channels for generating the CH sequence will consist of  $Z$  channels instead of  $M$  given by

$$Z = M - n.(T_{ch} > T_{th}) \quad (2)$$

Let  $\theta$  be a parameter corresponding to  $n$  such that ( $0 < \theta \leq 1$ ). In other words,  $\theta$  can be defined as a percentage of the total number of channels  $M$  such that the number of channels available to a user is equal to  $\theta M$  after implementing the proposed algorithm. For example,  $\theta = 0.9$  means that only 90 percent of the available channels have availability duration in respect of the requirement and therefore the rest 10 percent of channels must be removed from the hopping sequence. As  $\theta$  decreases,  $n$  increases and thus  $Z$  decreases. It is to be noted that in this approach, the value of  $T_{th}$  plays an important role. The optimal value of  $T_{th}$  must be set according to the channel availability duration requirement of LP or HP data. The pseudo code of the proposed algorithm is given. spectrum sensing results must be available beforehand before implementing the proposed algorithm. In any CRN environment, channels must be available for secondary usage which is the meaning of line 1. As explained in the preceding section, circular window  $|S|$  has the updated sensing results. Based on the information available from  $|S|$ , the channels will be classified as either sparsely used or heavily used, given in line 2.

This channel classification also tells about the PU channel usage pattern. Now, instead of all available channels  $M$ , only those channels that meets the need of the required QoS by the SUs will be taken for generating the CH sequence. The required QoS has been classified in HP and LP as explained earlier in the preceding section. There is a parameter  $T_{th}$  taken which is referred to as threshold value of availability duration and  $T_{ch}$  represents any channel availability duration. For HP data, channels which have  $T_{ch} > T_{th}$  are of use because the channels which have  $T_{ch} < T_{th}$  would not satisfy the QoS requirement of SUs owing to be available for the time less than minimum required time. Considering these channels in generating CH sequence would rather degrade the QoS

### The Proposed Algorithm

#### Require:

Spectrum sensing result,  $S \neq 0$ .

PU usage pattern: Q (0: Sparse use, 1: Heavy use)

Type of QoS required: P (H: High, L: Low)

$M$  = Total no. of available channels.

$T_{ch}$  = Channel availability duration

$T_{th}$  = Threshold level of channel availability duration.

$n.T_{ch}$  = no. of channels having availability duration  $T_{ch}$

$N$  = No. of channels having availability duration corresponding to required QoS;  $N \leq M$ .

$Z$  = New set of available channels

```

1:  If  $S \neq 0$  then
2:      if  $P == H$  then
3:           $N = n.T_{ch} < T_{th}$ 
4:          While ( $N \neq 0$ ) do
5:               $Z = M - N$  // new set of channels
6:          if  $Q == 0$  then
7:               $N \approx \text{low} \ \& \ Z \approx \text{large}$ 
8:          else if  $Q == 1$  then
9:               $N \approx \text{large} \ \& \ Z \approx \text{less}$ 
10:         end if
11:         end while
12:      else if  $P == L$  then
13:           $N = n.T_{ch} > T_{th}$ 
14:          While ( $N \neq 0$ ) do
15:               $Z = M - N$  //new set of channels
16:          if  $Q == 0$  then
17:               $N \approx \text{large} \ \& \ Z \approx \text{less}$ 
18:          else if  $Q == 1$  then
19:               $N \approx \text{low} \ \& \ Z \approx \text{large}$ 
20:          end if
21:          end while
22:      end if
23:  end if

```

required by SUs and also would lead to more MTTR. Considering channels which have  $T_{ch} > T_{th}$  for HP data would be advantageous in the sense that the required QoS will be provided to the SUs and also as some channels which have  $T_{ch} < T_{th}$  gets eliminated from the CH sequence, the MTTR will be reduced. In line 3,  $N$  is the number of channels by which the total number of channels  $M$  gets lowered by. When required QoS is HP,  $N$  is the number of channels having  $T_{ch} < T_{th}$  (given by Eq. 1) and when required QoS is LP,  $N$  is the number of channels having  $T_{ch} > T_{th}$  (given by Eq. 2) given in line 13. Eliminating  $N$  number of channels from total number of channels, we will get new set of total available channels  $Z$  for generating the CH sequences. This new set of available channels  $Z$  is less than  $M$  or in worst case will be equal to  $M$ . It is to be noted that the channel usage by PU has an important role in all CH algorithms and so does in the proposed approach.

HP data will be served well when channels are sparsely used and LP data will be served well in both sparsely or heavily used PU activity. However, for LP data, heavily used channels can be allocated because LP data can tolerate delay or intermittent transmission and the channels which have low  $T_{ch}$  will also be utilized resulting in improved spectrum efficiency. So, there are two type of services we need to serve in a CRN-IoT environment, high QoS demanding or HP data and low QoS or LP data. For the first section of the proposed approach, lines 10 to 20 are for HP data. For the sparsely used case, channels which have  $T_{ch} > T_{th}$  will be in abundance and serves well the HP data. On the other hand, channels which have  $T_{ch} < T_{th}$  will be less in numbers. So, there would not be much difference between the original number of available channels and new set of available channels. In this condition, the impact on lowering the MTTR will be less. Similarly, when channels are heavily used more number of channels which have  $T_{ch} < T_{th}$  will be in abundance and channels which have  $T_{ch} > T_{th}$  will be less in number. The result is new set of available channels will be very less in number than original available channels. In this condition, MTTR will be lowered by significant amount as the CH sequence now have less number of channels. Similarly, when the required QoS is LP, lines 12 to 23 can be understood the same way.

### B. Algorithm Analysis

The proposed method is applicable to all existing CH rendezvous algorithms and provides required QoS to the SUs with reduced MTTR than the existing ones. We will prove this by taking few state of the art CH rendezvous algorithms. Before proceeding further, it has been observed from the discussion in the preceding section that when required QoS is HP, the CH sequence consists of channels which have  $T_{ch} > T_{th}$ . Based on this observation, we have Lemma 1 and Lemma 2 as follows.

**Lemma 1.** For HP data, given total number of available channels  $M$ , the CH sequence will have  $(M - N)$  number of channels where  $N$  is the number of channels having ( $T_{ch} < T_{th}$ .)

**Lemma 2.** For LP data, given total number of available channels  $M$ , the CH sequence will have  $(M - N)$  number of channels where  $N$  is the number of channels having ( $T_{ch} > T_{th}$ .)

Lemma 1 and Lemma 2 has been proved in the preceding section using Eqs. 1 and 2.

**Theorem:** Any two SUs having channel hopping sequence generated by any CH rendezvous algorithm will rendezvous with  $MTTR = f(M - N)$ .

**Proof:** Above theorem has been derived as a result of discussion in the preceding section. It can further be proved by taking example of an existing rendezvous

algorithm. We have taken Jump stay (JS) algorithm [13] to prove our algorithm. In JS algorithm, if  $M$  is the total number of channels and  $P$  is the smallest prime number not less than  $M$ , than MTTR is  $3P$  time slots for Symmetric model. For example, if  $M = 4$  than  $P$  is 5 (prime number not less than 4). According to JS algorithm, MTTR will be  $3 \times 5 = 15$  time slots. Now, when the proposed algorithm is implemented into the JS algorithm we have following observations:

Let required QoS is HP data and when the channels in the circular window  $|S|$  are analyzed for their availability duration  $T_{ch}$ , Let 3 channels have  $T_{ch} > T_{th}$  where  $T_{th}$  is the minimum time required to service the HP data as explained in the preceding section. Now, according to the proposed algorithm, only these 3 channels are to be taken for generating the CH sequence. So, the new value of  $M$  is now 3 instead of 4 (eliminating one channel that has  $T_{ch} < T_{th}$ ). For  $M = 3$ , MTTR is now  $3 \times 3 = 9$ . In this example, MTTR is reduced from 15 time slots to just 9 time slots when the proposed algorithm is implemented into JS algorithm. Similarly, the proposed algorithm can be implemented into any other CH rendezvous algorithm. We have seen in this example that without implementing the proposed algorithm, MTTR depends on the value of  $M$  which means MTTR is function of  $M$ . Now, when the proposed algorithm is implemented,  $M$  is reduced to  $M - N$  ( $N$  is 1 in the example). With the proposed algorithm implemented, MTTR depends on the value of  $M - N$  which means MTTR is function of  $M - N$ . For the sparse use of channels by PUs, channels which have  $T_{ch} > T_{th}$  will be more in number and so when required QoS is LP, the value of  $N$  will be large which results in less number of new set of channels using which the CH sequence will be generated. The effect of this will be on minimizing the time to rendezvous and so MTTR will also be lowered as compared to the case when QoS is not considered. Similarly, for heavily used of channels by PUs, channels which have  $T_{ch} < T_{th}$  will be more in number and so when required QoS is HP, the value of  $N$  will be large which results in less number of new set of channels using which the CH sequence will be generated. The effect of this will be on minimizing the time to rendezvous and so MTTR will also be lowered in this case also compared to the case when QoS is not considered. It is also to be noted that in these cases when the value of  $N$  is small, then it will not affect much the value of MTTR. Also, the value of  $N$  is highly dependent on the network conditions which is highly dynamic and it may increase or decrease according to changing network conditions. Whatever be the value of  $N$  in both the cases (HP or LP), the generated CH sequence will not take  $N$  number of channels in the hopping sequence. As discussed previously, we have represented effect of  $N$  in terms of  $\theta$ . When  $\theta$  tends to 1,  $N$  tends to 0. This means that all available channels are of importance and have to be used in generating the CH



sequence. As  $\theta$  moves away from 1 towards 0, the value of  $N$  increases which means only channels of required QoS will be used and  $N$  number of channels will be removed before generating the CH sequence.

## 5. PERFORMANCE ANALYSIS

We evaluate the performance of the proposed algorithm both analytically and through simulations. MTTR and throughput are the two performance metrics to analyse the performance of the proposed algorithm. Two state of the art rendezvous algorithms have been taken for performance evaluation; 1) Jump Stay Algorithm [13] and 2) Sender Jump Receiver Wait Algorithm [17]. Both of these CH rendezvous algorithms are applicable to symmetric as well as asymmetric channel availability models. While symmetric model has low MTTR as compared to asymmetric model because of the fact that all channels are common between two CH SUs, evaluation of MTTR is done for only one commonly available channel between sender and receiver SUs in asymmetric model. In the simulations we perform, we assume that each SU has the information of the total number of available channels and their availability duration. The information is obtained from the circular window  $|S|$ . In addition to MTTR, throughput has also been evaluated for the proposed rendezvous method (separately for HP data and LP data) and native rendezvous algorithms which is defined as the percentage of number of SUs served in a CRN with respect to the duty cycle of PUs. Here, duty cycle ' $T$ ' refers to average number of busy channels due to active PUs given by the expression

$$T = \frac{T_{ON}}{T_{ON} + T_{OFF}} \quad (3)$$

In the above expression,  $T_{ON}$  and  $T_{OFF}$  refers to average active and inactive duration of PUs. Without loss of generality, as  $T$  increases, throughput degrades in the native schemes because of the fact as  $T$  increases, less number of channels will be available to SUs with low  $T_{ch}$ . The results have been obtained for MTTR and throughput of these CH algorithms and then the effect of the proposed algorithm has been observed using MATLAB simulations. We shall discuss the effect of proposed algorithm on each of the said algorithms one by one.

*A. Jump Stay (JS) Algorithm* [13]: Without going much in the details of how the CH sequence is generated using these algorithms, MTTR achieved by these algorithms has been considered for evaluating the performance of the proposed algorithm. JS algorithm achieves guaranteed rendezvous in both Symmetric and Asymmetric cases and we shall observe the effect in both the cases. First we will observe the effect for Symmetric channel availability for which MTTR given by JS is  $3P$  time slots where  $P$  is the smallest prime number not less

than the number of available channels  $M$ . The effectiveness of the proposed algorithm can be seen by taking two cases as given. As given in section 4,  $\theta$  is the proportion of total channels  $M$  which satisfies the requirement criteria (for LP or HP) and hence  $M\theta$  number of channels will be used in the hopping sequence. As  $\theta$ , the number of channels to be used in the hopping sequence will decrease and therefore the MTTR will decrease.

**Case 1.** When the proposed algorithm is not implemented, we have assumed  $M=100$  for which  $MTTR = 303$  time slots as shown in Figure. 4. This is the value of MTTR when channel availability duration and QoS required by SU has not been considered and CH sequence is generated using all the available channels.

**Case 2.** When the proposed algorithm is implemented which takes into account the QoS required and channel availability duration, we have following observations. We have assumed  $M=100$  and used the factor  $\theta$  to show the effect of removing the channels from the hopping sequence which does not satisfy the required QoS criteria.  $\theta = 1$  means that all available channels  $M$  have to be used in the hopping sequence as all the channels satisfies the required QoS criteria (all channels have  $T_{ch} < T_{th}$  for LP or  $T_{ch} > T_{th}$  for HP). As  $\theta$  varies from 1 to 0, channels with required QoS decreases and thus corresponding number of channels will be removed from the hopping sequence. Therefore, the more the  $\theta$  moves away from 1 towards 0, more and more number of channels will be removed from the hopping sequence.

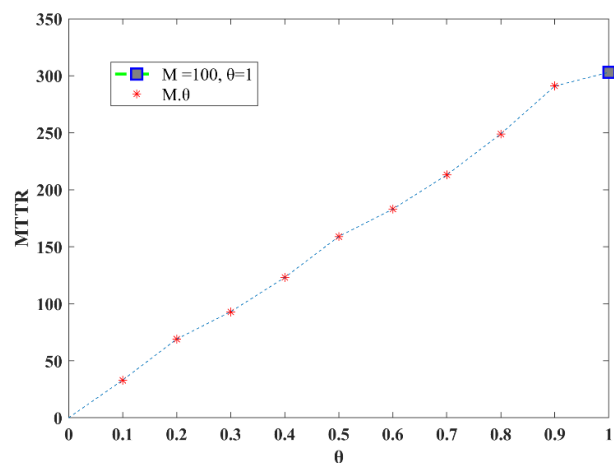


Figure. 4. Effect of  $\theta$  on MTTR in Symmetric JS [13]



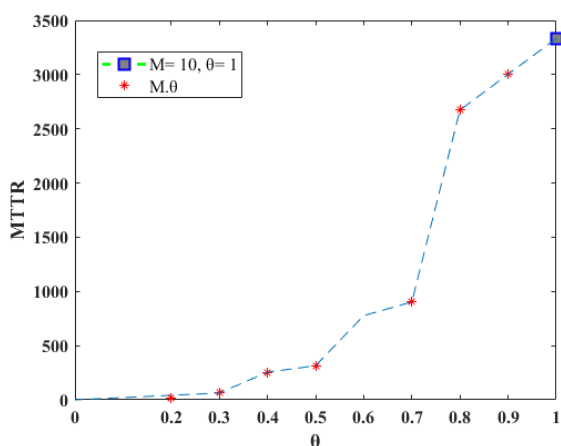


Figure 5. Effect of  $\theta$  on MTTR in Asymmetric JS [13]

As shown in Figure. 4, without considering the required QoS, MTTR is 303 time slots for  $M=100$ . According to the proposed algorithm, when required QoS is taken into account, MTTR starts decreasing as  $\theta$  decreases. As shown, when  $\theta=0.9$ ,  $M$  has been reduced from 100 to 90 which means 10 percent of the channels does not satisfy the required QoS and thus MTTR is also reduced from 303 time slots to 291 time slots. For HP data, these 10 percent of channels have  $T_{ch} < T_{th}$  and  $T_{ch} > T_{th}$  for LP data. Similarly, it is shown that as  $\theta$  reduces further, number of channels in the hopping sequence also lowered and thus MTTR decreases as only those channels will be used for generating the CH sequence which fulfils the required QoS. Similarly, the effect can be seen in the Asymmetric CH JS algorithm. We have taken  $M=10$ ,  $C=1$  (number of commonly available channels between sender and receiver). For these values of  $M$  and  $C$ , the effect of  $\theta$  on the value of MTTR can be seen in Figure. 5. Next we shall see the effect on MTTR for Sender Jump Receiver Wait algorithm.

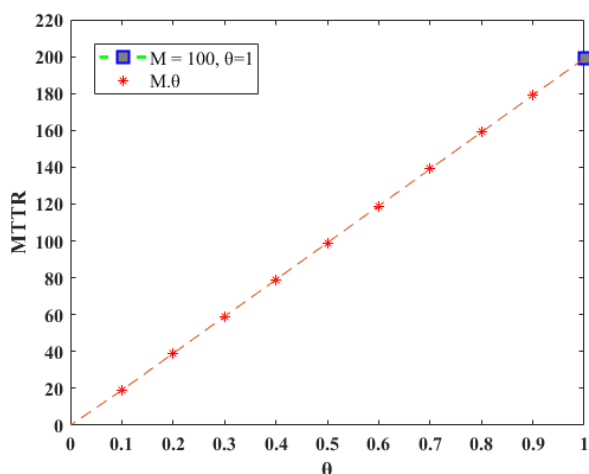


Figure 6. Effect of  $\theta$  on MTTR in Symmetric SJ-RW [17]

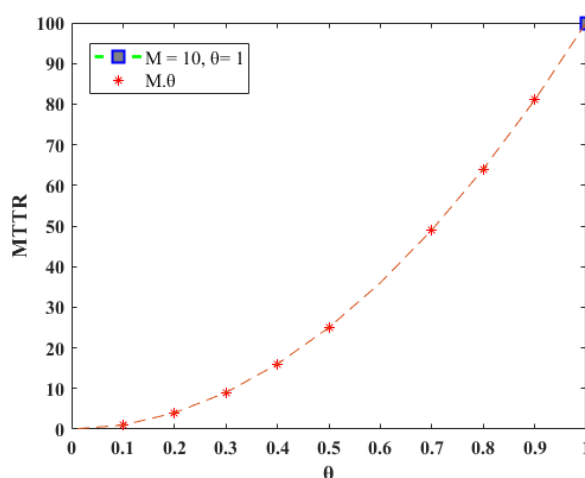


Figure 7. Effect of  $\theta$  on MTTR in Asymmetric SJ-RW [17]

**B. Sender Jump-Receiver Wait (SJ-RW) algorithm [17]** achieves better MTTR than JS. SJ-RW algorithm achieves guaranteed rendezvous for Symmetric as well as for Asymmetric channel availability. For Symmetric case, the behaviour of  $\theta$  on MTTR is shown in Figure. 6 and in Figure. 7 for asymmetric case.  $M$  is the number of channels available to sender and receiver. We shall see the performance of the proposed algorithm by taking two cases as given.

**Case 1.** When channel availability duration and QoS required by SU has is not considered and CH sequence is generated using all the available channels, for  $M=100$ , MTTR is 199 time slots as shown in Figure. 6.

**Case 2.** When the proposed algorithm is implemented which takes into account the QoS required and channel availability duration, we have following observations. For symmetric case, we have assumed  $M=100$  and used the factor  $\theta$  to show the effect of removing the channels from the hopping sequence which does not satisfy the required QoS criteria. As discussed for JS [13], MTTR reduces as  $\theta$  reduces. As  $\theta$  varies from 1 to 0, channels with required QoS decreases and thus corresponding number of channels will be removed from the hopping sequence. Therefore, the more the  $\theta$  moves away from 1 towards 0, more and more number of channels will be removed from the hopping sequence. Similarly, the effect of the proposed algorithm can be seen in the Asymmetric CH SJ-RW algorithm [17] for which worst case MTTR is there when  $C=1$  (commonly available channel between sender and receiver). For  $M=10$  and without implementing the proposed algorithm, MTTR is 100 time slots as shown in Figure. 7. But when proposed algorithm is considered, as  $\theta$  reduces (in accordance with channels which does not satisfy the required QoS criteria), MTTR starts decreasing as shown in Figure. 7. MTTR reduces in respect of  $\theta$  as now instead of  $M$ , only those channels have been considered for generating the CH sequence that fulfils the required QoS criteria.



### C. Throughput

CH rendezvous schemes proposed so far in literature do not consider QoS and channel availability criteria in selecting channels for generating CH sequences and main focus has been on minimizing the TTR. In such CH environments, throughput degrades as duty cycle (given by Eq. 3) of PUs increases as shown in Figure 8. This is due to the fact that in native CH schemes, all SUs (LP data or HP data) have equal opportunity to rendezvous on all available channels. As a result of this, there arises several performance degradation factors which includes multi user rendezvous on the same channel resulting in collision and increase in packet drop rate due to non-availability of channels when duty cycle is high. On the other hand, the proposed CH rendezvous approach suggests separate search domain of channels for LP data and HP data. As a result, probability of collision gets minimized and the users will be served with required type of QoS. As shown in Figure 8, throughput of LP QoS applications is better than HP QoS and native scheme because of the fact that even when the duty cycle is high, communication requirement of LP QoS applications is fulfilled with channels having low availability duration. For HP QoS applications, having separate search domain, overall throughput is better than the native CH rendezvous schemes as shown in Figure 8. Therefore, with the proposed method of CH sequence generation which takes into account the QoS criteria and channel availability duration, overall performance in terms of throughput has been improved as compared to native CH schemes.

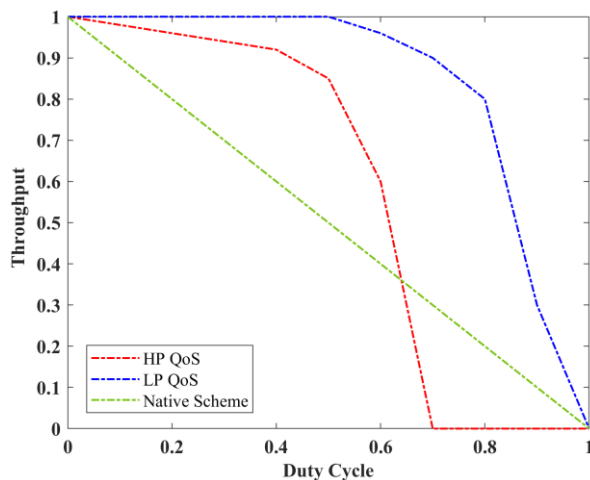


Figure 8. Throughput w.r.t varying duty cycle

In the analysis of both the algorithms as explained, it has been found that MTTR is a function of number of channels in the hopping sequence. Also, existing rendezvous algorithms have not considered the required

QoS by SUs and focused only on achieving lowest possible MTTR. However, without considering the required QoS, even if low MTTR is achieved it doesn't guarantee the increase in overall throughput because it may happen that a HP QoS requirement by SUs assigned channels of low availability duration and vice versa. It has also been shown that when required QoS is taken into account before generating the CH sequences for link establishment, MTTR is reduced substantially and better throughput has been achieved as compared to the native CH rendezvous schemes. Apart from achieving low MTTR, required QoS is being provided in using this approach. Also, all channels regardless of their availability duration are utilized accordingly and hence spectrum utilization efficiency is increased. Here, we have presented the effect on two CH rendezvous algorithms but the approach adopted can be applied to any of the existing CH rendezvous algorithms.

## 6. CONCLUSION

In this paper, a QoS aware CH algorithm has been proposed for achieving rendezvous. Owing to the fact that the QoS required by different users in an IoT environment is different, not all channels (on the basis of availability duration) in a CH sequence are of use to some applications. This motivation has been used in building the proposed algorithm. When Implemented into the existing CH algorithms, the proposed algorithm not only reduces MTTR and increases network throughput but also takes care of the QoS required by SUs. Also, all types of channels (regardless of their availability duration) not in use by PUs are shown to be utilized accordingly which enhances the spectrum utilization efficiency. Extensive theoretical analysis and simulation results have verified the performance improvement of CH algorithms in terms of MTTR and throughput and required type of QoS is provided to the SUs.

## REFERENCES

- [1] Atzori, L., Iera, A., & Morabito, G. (2010). The Internet of Things: A survey. *Computer Networks*, 54(15), 2787–2805. <https://doi.org/10.1016/j.comnet.2010.05.010>
- [2] Bany Salameh, H. A., Almajali, S., Ayyash, M., & Elgala, H. (2018). Spectrum Assignment in Cognitive Radio Networks for Internet-of-Things Delay-Sensitive Applications Under Jamming Attacks. *IEEE Internet of Things Journal*, 5(3), 1904–1913. <https://doi.org/10.1109/JIOT.2018.2817339>
- [3] Aslam, S., Ejaz, W., & Ibnkahla, M. (2018). Energy and Spectral Efficient Cognitive Radio Sensor Networks for Internet of Things. *IEEE Internet of Things Journal*, 5(4), 3220–3233. <https://doi.org/10.1109/JIOT.2018.2837354>
- [4] Akyildiz, I. F., Lee, W.-Y., Vuran, M. C., & Mohanty, S. (2006). NeXt generation/dynamic spectrum access/cognitive radio wireless networks: A survey. *Computer Networks*, 50(13), 2127–2159. <https://doi.org/10.1016/j.comnet.2006.05.001>



- [5] DaSilva, L. A., & Guerreiro, I. (2008). Sequence-Based Rendezvous for Dynamic Spectrum Access. *2008 3rd IEEE Symposium on New Frontiers in Dynamic Spectrum Access Networks*. <https://doi.org/10.1109/DYSPAN.2008.52>
- [6] Wu, Q., Ding, G., Xu, Y., Feng, S., Du, Z., Wang, J., & Long, K. (2014a). Cognitive Internet of Things: A New Paradigm Beyond Connection. *IEEE Internet of Things Journal*, 1(2), 129–143. <https://doi.org/10.1109/IJOT.2014.2311513>
- [7] Li, T., Yuan, J., & Torlak, M. (2018). Network Throughput Optimization for Random Access Narrowband Cognitive Radio Internet of Things (NB-CR-IoT). *IEEE Internet of Things Journal*, 5(3), 1436–1448. <https://doi.org/10.1109/IJOT.2017.2789217>
- [8] Miz, V., & Hahanov, V. (2014). Smart traffic light in terms of the cognitive road traffic management system (CTMS) based on the Internet of Things. *Proceedings of IEEE East-West Design & Test Symposium (EWDTS 2014)*. <https://doi.org/10.1109/EWDTS.2014.7027102>
- [9] Aijaz, A., & Aghvami, A. H. (2015). Cognitive Machine-to-Machine Communications for Internet-of-Things: A Protocol Stack Perspective. *IEEE Internet of Things Journal*, 2(2), 103–112. <https://doi.org/10.1109/IJOT.2015.2390775>
- [10] Tervonen, J., Mikhaylov, K., Pieska, S., Jamsa, J., & Heikkilä, M. (2014). Cognitive Internet-of-Things solutions enabled by wireless sensor and actuator networks. *2014 5th IEEE Conference on Cognitive Infocommunications (CogInfoCom)*. <https://doi.org/10.1109/CogInfoCom.2014.7020426>
- [11] Sallai, G. (2013). From telecommunications to cognitive infocommunications and internet of things - Phases of digital convergence. *2013 IEEE 17th International Conference on Intelligent Engineering Systems (INES)*. <https://doi.org/10.1109/INES.2013.6632803>
- [12] Alberti, A. M., Mazzer, D., Bontempo, M. M., de Oliveira, L. H., da Rosa Righi, R., & Cerqueira Sodr e, A. (2017). Cognitive radio in the context of internet of things using a novel future internet architecture called NovaGenesis. *Computers & Electrical Engineering*, 57, 147–161. <https://doi.org/10.1016/j.compeleceng.2016.07.008>
- [13] Liu, H., Lin, Z., Chu, X., & Leung, Y.-W. (2012). Jump-Stay Rendezvous Algorithm for Cognitive Radio Networks. *IEEE Transactions on Parallel and Distributed Systems*, 23(10), 1867–1881. <https://doi.org/10.1109/TPDS.2012.22>
- [14] Asifuddola, M., Ram, S., Begh, G. R., & Rather, G. M. (2013). Vacate on demand algorithm in cognitive radio networks. *2013 IEEE International Conference on Signal Processing, Computing and Control (ISPCCC)*. <https://doi.org/10.1109/ISPCCC.2013.6663430>
- [15] Theis, N. C., Thomas, R. W., & DaSilva, L. A. (2011). Rendezvous for Cognitive Radios. *IEEE Transactions on Mobile Computing*, 10(2), 216–227. <https://doi.org/10.1109/TMC.2010.60>
- [16] Uyanik, G. S., & Oktug, S. (2017). Cognitive channel selection and scheduling for multi-channel dynamic spectrum access networks considering QoS levels. *Ad Hoc Networks*, 62, 22–34. <https://doi.org/10.1016/j.adhoc.2017.04.003>
- [17] Li, J., Zhao, H., Wei, J., Ma, D., & Zhou, L. (2018). Sender-Jump Receiver-Wait: A Simple Blind Rendezvous Algorithm for Distributed Cognitive Radio Networks. *IEEE Transactions on Mobile Computing*, 17(1), 183–196. <https://doi.org/10.1109/TMC.2017.2703166>
- [18] Chang, G.-Y., Wang, S.-Y., & Liu, Y.-X. (2017). A Jamming-Resistant Channel Hopping Scheme for Cognitive Radio Networks. *IEEE Transactions on Wireless Communications*, 16(10), 6712–6725. <https://doi.org/10.1109/TWC.2017.2728659>
- [19] Yang, B., Liang, W., Zheng, M., & Liang, Y.-C. (2016). Fully Distributed Channel-Hopping Algorithms for Rendezvous Setup in Cognitive Multiradio Networks. *IEEE Transactions on Vehicular Technology*, 65(10), 8629–8643. <https://doi.org/10.1109/TVT.2015.2510359>
- [20] Yu, L., Liu, H., Leung, Y.-W., Chu, X., & Lin, Z. (2015). Multiple Radios for Fast Rendezvous in Cognitive Radio Networks. *IEEE Transactions on Mobile Computing*, 14(9), 1917–1931. <https://doi.org/10.1109/TMC.2014.2366746>
- [21] Paul, R., & Choi, Y.-J. (2016). Adaptive Rendezvous for Heterogeneous Channel Environments in Cognitive Radio Networks. *IEEE Transactions on Wireless Communications*, 15(11), 7753–7765. <https://doi.org/10.1109/TWC.2016.2607170>
- [22] Chao, C.-M., Hsu, C.-Y., & Ling, Y.-T. (2016). Efficient Asynchronous Channel Hopping Design for Cognitive Radio Networks. *IEEE Transactions on Vehicular Technology*, 65(9), 6888–6900. <https://doi.org/10.1109/TVT.2015.2490722>
- [23] Wu, C.-C., Wu, S.-H., & Chen, W.-T. (2017). On Low-Overhead and Stable Data Transmission between Channel-Hopping Cognitive Radios. *IEEE Transactions on Mobile Computing*, 16(9), 2574–2587. <https://doi.org/10.1109/TMC.2016.2623715>
- [24] Bian, K., & “Jerry” Park, J.-M. (2013). Maximizing Rendezvous Diversity in Rendezvous Protocols for Decentralized Cognitive Radio Networks. *IEEE Transactions on Mobile Computing*, 12(7), 1294–1307. <https://doi.org/10.1109/TMC.2012.103>
- [25] Chen, W.-C., Yang, G.-C., Chang, M.-K., & Kwong, W. C. (2017). Construction and Analysis of Shift-Invariant, Asynchronous-Symmetric Channel-Hopping Sequences for Cognitive Radio Networks. *IEEE Transactions on Communications*, 65(4), 1494–1506. <https://doi.org/10.1109/TCOMM.2017.2653158>
- [26] Li, A., Han, G., & Ohtsuki, T. (2019a). A Fast Blind Scheme With Full Rendezvous Diversity for Heterogeneous Cognitive Radio Networks. *IEEE Transactions on Cognitive Communications and Networking*, 5(3), 805–818. <https://doi.org/10.1109/TCCN.2019.2917666>
- [27] Sheu, J.-P., & Lin, J.-J. (2018). A Multi-Radio Rendezvous Algorithm Based on Chinese Remainder Theorem in Heterogeneous Cognitive Radio Networks. *IEEE Transactions on Mobile Computing*, 17(9), 1980–1990. <https://doi.org/10.1109/TMC.2018.2790408>
- [28] Tan, X. J., Zhou, C., & Chen, J. (2017). Symmetric Channel Hopping for Blind Rendezvous in Cognitive Radio Networks Based on Union of Disjoint Difference Sets. *IEEE Transactions on Vehicular Technology*, 66(11), 10233–10248. <https://doi.org/10.1109/TVT.2017.2726352>
- [29] Chang, C.-S., Chen, C.-Y., Lee, D.-S., & Liao, W. (2017). Efficient Encoding of User IDs for Nearly Optimal Expected Time-To-Rendezvous in Heterogeneous Cognitive Radio Networks. *IEEE/ACM Transactions on Networking*, 25(6), 3323–3337. <https://doi.org/10.1109/TNET.2017.2734695>
- [30] Chao, C.-M., & Fu, H.-Y. (2018). Supporting fast and fair rendezvous for cognitive radio networks. *Journal of Network and Computer Applications*, 113, 98–108. <https://doi.org/10.1016/j.jnca.2018.03.032>
- [31] Chang, G.-Y., Teng, W.-H., Chen, H.-Y., & Sheu, J.-P. (2014). Novel Channel-Hopping Schemes for Cognitive Radio Networks. *IEEE Transactions on Mobile Computing*, 13(2), 407–421. <https://doi.org/10.1109/TMC.2012.260>
- [32] Sheu, J.-P., Su, C.-W., & Chang, G.-Y. (2016). Asynchronous Quorum-Based Blind Rendezvous Schemes for Cognitive Radio Networks. *IEEE Transactions on Communications*, 64(3), 918–930. <https://doi.org/10.1109/TCOMM.2015.2508803>



**Mohd Asifuddola** is an Assistant professor in the Department of Electronics & Communication Engineering, Polytechnic, Maulana Azad National Urdu University, Hyderabad, India. He is pursuing the Ph.D. degree in the Department of Computer Science & Engineering, (N.I.T), Srinagar, J&K, India. His current research interests include Cognitive Radio Networks, Internet of Things, Next-Generation Networks and Computer Networks.



**Mohammad Ahsan Chishti** is an Associate Professor in the Department of Information Technology at Central university of Kashmir, Kashmir, India. He received his Ph.D. in Computer Science from N.I.T Srinagar, J&K, India in 2016. He has more than 50 research publications to his credit and 12 patents with two granted International Patents. He is a Young Scientist awardee from Department of Science & Technology, Government of Jammu and Kashmir for the year 2009-2010 and IEI young engineer awardee 2015. He is Senior Member of IEEE, Member IEI, Life Member CSI, and Member IETE. He is a certified White belt in Six Sigma by Six Sigma Advantage Inc. of USA (SSAI).