ASCH: A Novel Asymmetric Synchronous Channel Hopping Algorithm for Cognitive Radio Networks

Sulagna Mohapatra, Prasan Kumar Sahoo
Dept. of Computer Science and Information Engineering
Chang Gung University, Kwei-Shan, 33302, Taiwan (ROC)
Email: m0229029@stmail.cgu.edu.tw, pksahoo@mail.cgu.edu.tw

***Abstract*—Cognitive Radio Network (CRN) is a growing cutting edge technology in the field of wireless communication. The fundamental idea behind CRN is to allow some unlicensed Secondary Users (SUs) who can utilize the spectrum holes in absence of the licensed Primary Users (PUs). Channel Hopping (CH) procedure is followed by the SUs to establish the rendezvous process. In this paper, a novel Asymmetric Synchronous Channel Hopping (ASCH) protocol is proposed in which different SUs can have different sets of channels to establish the rendezvous. Simulation results show that our protocol can achieve better performance over the exiting channel hopping protocols in terms of degree of rendezvous, Maximum Time To Rendezvous (MTTR), and throughput.**

I. Introduction

The recent study of wireless communication indicates the fact about under-utilization of licensed spectrum and over-crowded in the unlicensed spectrum due to extensive growth of wireless devices. To handle those issues, Federal Communications Commission (FCC) has introduced Cognitive Radio Network (CRN) technology, which enhances the Dynamic Spectrum Access (DSA). CRN comprises with both licensed Primary Users (PUs) and unlicensed Secondary Users (SUs). The main objective of CRN is to utilize the unused spectrum fairly among all the SUs in the absence of PUs [3]-[8].

In CRN, each SU is equipped with one or more radios to sense the spectrum holes. SUs can communicate with each other through the process of "rendezvous" [4]-[8]. They need to meet with each other in the same available channel at the same time instance to establish rendezvous, which includes both the process of negotiation and data transfer. Rendezvous process of SUs is carried out either by the Common Control Channel (CCC) or by Channel Hopping (CH) approach [1], [3]. In CCC approach, SUs make their negotiation in the control channel and then switch to the negotiated data channel for further data transmission.

Control channel saturation problem, jamming attack due to single control channel, selection of control channel overhead [1], [2] are some of the drawbacks in CCC approach. In channel hopping approach, each SU follows a particular channel hopping sequence. By using CH sequences, each SU hops from one channel to another to make its rendezvous successful [3]-[8]. In this approach, the network is considered as either synchronous or asynchronous and the channel type can be symmetric or asymmetric. In a synchronous environment, normally all SUs are entered into the network at the same time, whereas in asynchronous scenario SUs are arrived at different time instance without any clock synchronization. SUs are used symmetric channel type, i.e. all SUs have same availability of channels whereas in asymmetric approach the availability of channel is different for each SU [4]-[8].

The performance of SUs following the channel hopping approach is mainly evaluated in terms of degree of rendezvous and Time To Rendezvous (TTR). Generally, degree of rendezvous is defined as how many number of times two SUs can rendezvous with each other in a CH sequence. The TTR of a SU is explained as the number of time slots required for a SU to rendezvous with another SU. The worst case of TTR is called as MTTR (Maximum Time To Rendezvous).

The rest of the paper is organized as follows. Related works with our contribution are given in Section II. The proposed Asymmetric Synchronous Channel Hopping protocol is described in Section III. Performance evaluation of the protocol is made in Section IV and concluding remarks are made in Section V.

II. Related work

Recently, most of the research work in Cognitive Radio Networks (CRN) focus on symmetric synchronous, symmetric asynchronous and asymmetric asynchronous channel environment. Though authors propose Common Control Channel (CCC) or Channel Hopping (CH) approaches for the propose of rendezvous between two SUs, each of them have their own advantage and disadvantages. The CH approach can overcome many problems of CCC such as long time blocking by PUs, control channel saturation problem, but it can have many other challenges [2], [4]-[8] such as assignment of licensed channels to unlicensed users, proper channel utilization by SUs, collision avoidance among SUs, increase in degree of rendezvous and to minimize the MTTR.

In [4], authors propose two protocols named as PRICH (Round-Robin Indemnity-channel Hopping) and CACH (Cycle-Adjustable Channel Hopping) in symmetric synchronous environment. However, in both protocols, degree of rendezvous is very small as SUs can rendezvous in N out of $N * (N + 1)$ time slots. A Rendezvous Couple Channel Hopping (RCCH) mechanism is proposed in [5], where a SU generates an alternative sequence in anticlockwise if it has any data to send. In this protocol, though the channel utilization ratio is increased by balancing the channel loading,

978-1-4799-6664-6/16/$31.00 ©2016 IEEE

the degree of rendezvous is also very small with large MTTR = $\frac{N}{2}$ for N number of licensed channels. The Asynchronous Rendezvous Channel Hopping (ARCH) scheme proposed in [5] brings rendezvous between the sender and receiver though they can have distinct time-parity. However, for N number of available channels, the degree of rendezvous is only $\frac{1}{N}$ as compared to the total number of time slots available in the entire channel hopping sequence. Beside, the MTTR value is large as it is $2N - 1$.

Authors in [6] propose E-AHW (Enhanced Alternate Hop and Wait) protocol in asynchronous symmetric and asymmetric environment. In E-AHW, the degree of rendezvous is very small, which occurs only in a single channel. Hence, if that particular channel is busy with PUs, no rendezvous is possible at all and also the MTTR value is very large. Authors in [7] design the jump and stay pattern algorithm in an asynchronous symmetric and asymmetric environment, where MTTR between two SU is $3P$ for $P > N$ and P is a prime number. In this mechanism, rendezvous probability in jump pattern is very small and there will be no rendezvous in stay pattern, if both SUs stay in different channels. Beside, two SUs need to wait for an entire inner round for rendezvous and therefore the MTTR value is very large.

In [8], authors design T-CH and D-CH algorithms, which are non-ID and ID based, respectively. In T-CH, both the sender and the receiver use same channel hopping sequence. Although it increases the degree of rendezvous, there is higher probability of collision in multiuser case. Since, each SU is assigned with a unique ID in D-CH, the degree of rendezvous and MTTR are also limited as two SUs obviously will have distinct IDs. From the survey of current literature, it can be concluded that performance of the most channel hopping protocols in terms of degree of rendezvous, available channel utilization and MTTR is not so encouraging. In this paper, an Asymmetric Synchronous Channel Hopping (ASCH) protocol is proposed to increase the degree of rendezvous, and to minimize the MTTR. The main contributions of this paper can be summarized as follows.

- The proposed protocol can work for any number of channels.
- The *Asymmetric Synchronous Channel Hopping Scheme (ASCH)* is introduced, which can achieve guaranteed rendezvous between any two SUs.
- The MTTR for any two SUs in this proposed ASCH scheme is one.

III. Asymmetric Synchronous Channel Hopping (ASCH) Mechanism

In this paper, a CRN that consists of a set of licensed PUs and unlicensed SUs is considered. Each SU is equipped with one half-duplex radio transceiver, which can have the capacity to sense the spectrum holes. An asymmetric channel hopping paradigm is considered in which different users can have different sets of available channels. Let N be the set of available channels in the CRN indexed from [0, $|N|$-1] and the channels those are present in the set N be $\{c_0, c_1, ..., c_i, ..., c_{N-1}\}$. Based on the conditions of asymmetric channel hopping, let C_A = $\{c_0, c_2, c_i, ..., c_{N-2}\}$ and C_B = $\{c_2, c_3, c_i, ..., c_{N-1}\}$ be the different sets of available channels for secondary users SU_A and SU_B, respectively. It is assumed that there is at least one common channel between both SUs, which can be the operational or rendezvous channel for both of them. Hence, $\exists\ c_i \in (C_A \cap C_B)$ such that $C_A \cap C_B \neq \emptyset$.

The entire network is divided into multiple time slots or slot indices starting from 0, 1, ... , T_l-1, where T_l = length of the hopping sequence. The hopping process is executed between different SUs to find a common vacant channel in which negotiation and data transfer are accomplished during the communication. The channel hopping pattern of a particular SU is determined by its channel hopping sequence (*CH_Seq*) and is expressed as CH_Seq = $\{(0, c_0), (1, c_1), ..., (i, c_i), ..., (T_l\text{-}1, c_{T_l-1})\}$, where $c_i \in [0, |N|\text{-}1]$ represents the channel number visited by a SU at i^{th} time slot in a CH sequence. The entire CH sequence of a SU comprises two parameters, i.e. slot index and channel number. If two SUs hop on the same channel at the same slot index, they can listen to each other and use this channel as the operational channel for their communication.

Due to heterogeneity nature of the channels, it is very challenging to achieve guaranteed rendezvous in an asymmetric environment. It is not possible for the SUs to meet in the common channel frequently, which reduces the degree of rendezvous and increases the value of TTR between SUs.

A. Finite Projective Plane (FPP)

A Finite Projective Plane (FPP) of order n with $n > 0$ is a collection of (n^2+n+1) lines and (n^2+n+1) points. Some of the properties of FPP are: 1. Every line contains $(n+1)$ number points. 2. Every point contains $(n+1)$ number lines. 3. Any two distinct lines intersect exactly at one point. 4. Any two distinct point lie on exactly one line. As shown in Fig. 1, an example of FPP of order n = 1 is presented. From the property (1) and (4), it can be inferred that $\nexists\ P_i \in L_i$ such that P_i = P_j, where P_i, P_j are any point $\in L_i$ and L_i is a line out of n^2+n+1 lines. According to property (2) and (3), $\exists$ $P_i \in \{L_1, L_2, ..., L_{n+1}\}$ such that $(L_1 \cap L_2 \cap .... \cap L_{n+1})$ = P_i. Therefore, each point must be associated with another n^2+n points, which are present within any $n+1$ number of lines out of (n^2+n+1) number of generated lines.

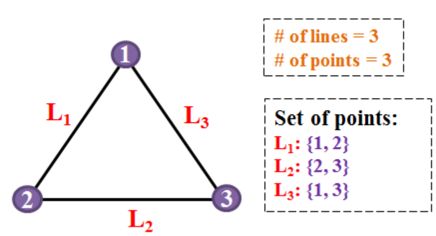

Fig. 1: An example of Finite Projective Plane of order n = 1.

B. Generation of general sequences

In our proposed channel hopping protocol, each CH sequence comprises a set of *General Sequences* denoted as *GS* = $\{s_0,\ s_1,\ s_2,\ ...,s_{m-1}\}$, where $m < |N|$. Each SU creates its own channel hopping sequences by taking different permutations of general sequences. It is assumed that all licensed channels of the set *N* are arranged in a ring and following steps are used to generate the general sequences.

Step-1:
: All licensed channels of set *N* are arranged in a ring along clockwise direction.

Step-2:
: A pivot element is selected whose value $p = \lfloor |N|/2 \rfloor$.

Step-3:
: Divide the channels into three sets as N_{front} *(N_f), N_{middle} (N_m), and N_{back} (N_b)* with help of *pivot element p*. The channels inside the sets can be expressed as $N_m = \{p^{th} element of set N\} = N[p]$, $N_f = \{0, 1,\ \ldots,\ (|N_m| - 1)\}$, $N_b = \{(|N_m| + 1,\ \ldots,\ (|N| - 1)\}$.

Step-4:
: Initial channel of the set N_f will go for an anticlockwise shift and then have an anticlockwise movement by choosing a set of channel-shifting seed set *r*. The element of set r can be calculated as follows: if $(|N_f|\%2 == 0)$, r = $\{0,\ 2,\ 4,\ \ldots,\ (|N_f|\ -\ 2)\}$, otherwise r = $\{0,\ 1,\ 3,\ \ldots,\ (|N_f| - 2)\}$. If channel-shifting seed = 0, then only clockwise movement is done without any shifting.

Step-5:
: New sets are created from the channels of set N_f by using each element of set *r*.

Step-6:
: Union operations are carried out between the set (N_m), (N_b) and with each newly generated set (N_f). Each concatenation generates a general sequence s_0, s_1, ..., s_{m-1}.

Step-7:
: The same procedure is repeated from Step-3 to Step-6 on set N_b for generating the rest of the general sequences.

Step-8:
: A sequence set *GS* is generated containing *m* number of sequences, which is called as *General Sequences* namely s_0, s_1, ..., s_{m-1}.

Let us consider a CRN with 9 channels, i.e. $|N|$ = 9. Then, the pivot element is calculated as $p = \lfloor \frac{9}{2} \rfloor$ = 4. According to Step-3, set N_m contains $p-th$ channel of set N, i.e. N_m = $\{3\}$. Similarly, set N_f contains channels $\{0, 1, 2\}$ and N_b contains channels $\{4, 5, 6, 7, 8\}$. All the channels in set N_f and N_b are also arranged in a ring. Initially channel-shifting seed set r is calculated based on set N_f. Since, $|N_f|$ is odd, i.e. 3, r = $\{0, 1\}$. When $r = 0$ is applied on set N_f, only the clockwise movement is done without any shifting. Hence, after applying $r = 0$, there is no change in the channel order of set N_f. When $r = 1$ is applied on set N_f, initial channel becomes 2, i.e. one anticlockwise shifting is done from initial channel 0. After shifting is made, an anticlockwise movement is carried out starting from the channel 2. After anticlockwise movement, a new set of N_f is generated, which contains the channel order like N_f = $\{2, 1, 0\}$. The general sequences s_0 and s_1 are generated under the union operations between the channels present in the set N_m and N_b with each generated sets N_f = $\{0, 1, 2\}$ and $\{2, 1, 0\}$.

It is to be noted that rest of the general sequences are generated by applying shifting seed on set N_b, which contains channels $\{4, 5, 6, 7, 8\}$. Again the shifting seed set r is calculated based on the set N_b. Since, $|N_b|$ = 5, r = $\{0, 1, 3\}$, as N_b - 2 = 3. After applying each shifting seed on channels of set N_b, three new sets as N_b = $\{4, 5, 6, 7, 8\}$ for $r = 0$, N_b = $\{8, 7, 6, 5, 4\}$ for $r = 1$ and N_b = $\{6, 5, 4, 8, 7\}$ for $r = 3$ are generated. General sequences are generated by performing union operations between the channels of set N_f = $\{0, 1, 2\}$ and set N_m with each newly generated set of N_b.

C. The ASCH Mechanism

In this section, the Asymmetric Synchronous Channel Hopping (ASCH) mechanism is designed by using FPP as a tool. The common points of FPP between the lines can map to the common channel between the SUs and other points can be considered as the corresponding slot numbers, where the common channel is going to be replaced. All $(n^2 + n)$ numbers of points are arranged in $(n+1)\times n$ matrix, which is known as a *slot_matrix*. Each column is associated with one general sequence starting from s_0, s_1, ..., s_{m-1}. If $m > n$, each column value of the matrix is again repeated after each n number of sequences. Overview of the proposed ASCH protocol is described in the following steps.

Step-1:
: Taking *N* number of available channels, *m* number of general sequences are generated as $\{s_0, s_1, ..., s_{m-1}\}$, which are shared by all SUs.

Step-2:
: Each SU prepares a permutation of sequences from those *m* number of general sequences to create its own channel hopping sequence.

Step-3:
: A common channel is identified by both SUs randomly from the set of common available channels. If $SU1_{AV}$ and $SU2_{AV}$ are the available channel sets of SU1 and SU2, respectively, the common available channel set $(CAV_{SU1,SU2})$ between them is $SU1_{AV}$ $\cap$ $SU2_{AV}$.

Step-4:
: FPP of order n is considered, which satisfies the condition $(n^2\ +\ n\ +\ 1) > |N|$.

Step-5:
: A *slot_matrix* is created by using FPP of order *n*, which contains the elements from 0 to $(n^2 + n)$ except the randomly selected common channel.

Step-6:

All elements are arranged in the $(n+1) \times n$ matrix along the column such that col_0 contains the value from 0 to n, col_1 contains the value from $(n+1)$ through $((n+(n+1))$ and so on.

Step-7:

If any element of the *slot_matrix* is equal to the randomly selected common channel number, then that element is not stored inside the matrix.

Step-8:

The randomly selected common available channel is replaced inside the general sequences whose slot indexes are identified by the column of the *slot_matrix*. The permuted order of general sequence is taken for the replacement.

Let s_i be a general sequence that comprises *N* number of slot indexes starting from 0 to *N - 1*. In this step, randomly selected common available channel is replaced in those slots of the general sequence s_i, where the slot indexes are identified by col_j of the *slot_matrix*. Out of *m* number of sequences, s_0, s_1, ...,s_{n-1} are related with col_0, col_1, ..., col_{n-1}, respectively. If *m* > *n*, again the column values are repeated from col_0 for the sequence s_n. If any of the values v_i inside the column is greater than the length of the sequence, i.e. $|N|$, one modulus operation is carried out, i.e. $(v_i \% |N|)$.

Step-9:

CH sequences is generated by each SU by concatenating the permuted general sequences, which are modified by using Step-8, i.e. the permuted order modified sequences are only concatenated one after another to create a CH sequence of length $m * |N|$.

After creation of general sequences, a SU needs to choose a permutation of general sequences. Let PS be the set that contains the permutated series of general sequences. The relationship between the general sequences and the *slot_matrix* *mat* can be explained as follows. $PS[0]$ corresponds to the first column, i.e. $mat[0][0], mat[1][0], ..., mat[n][0]$. Similarly, $PS[n-1]$ corresponds to the $(n-1)^{th}$ column, i.e. $mat[0][n-1], mat[1][n-1], ..., mat[n][n-1]$. For the sequence $PS[n+1]$, the values are again repeated such as the values present in $mat[0][0], mat[1][0], ..., mat[n][0]$, when $m > n$. The main idea of using FPP is to make some slots as the guaranteed rendezvous slots between two SUs within each general sequence. The slot number is identified by the column values of the *slot_matrix*.

As shown in Fig. 2, an example is considered, where available channels $|N| = 9$. The available channels for $SU1$ and $SU2$ are $SU1_{AV}$={0, 3, 4, 8} and $SU2_{AV}$={1, 5, 6, 8}, respectively. Hence, the common available channel is {8}. $SU1$ creates its own CH sequence by choosing the permutation s_0, s_1, s_4, s_3, and s_2, where $SU2$ creates its hopping sequence by using the permutation s_1, s_2, s_3, s_4, and s_0. In this example, our goal is to show the number of rendezvous between two SUs without using the concept of FPP. From Fig. 2, it can be observed that the degree of rendezvous is only 3 out of 45 time slots as both SUs have limited opportunity to meet at a common channel.

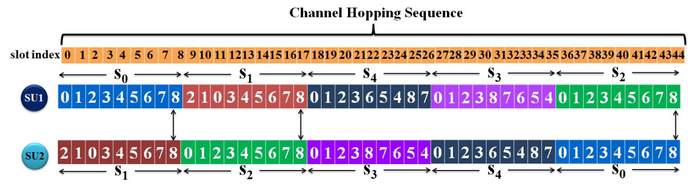

Fig. 2: Rendezvous between two SUs in asymmetric synchronous scenario without using the concept of FPP.

As shown in Fig. 3(a), the *slot_matrix* is constructed from the FPP of order $n = 3$, which satisfies the property (n^2+n+1)>$|N|$. By considering $n = 3$, the *slot_matrix* contains the values starting from 0 to 12 except the common channel 8. Each column of the *slot_matrix* contains $(n+1)$ number of values. Hence, col_0 contains from 0 to 3. Similarly, col_1 contains the values from 4 to 7 and col_2 from 9 to 12. In Fig. 3(b), it is demonstrated how the common channel replacement is done within the permutated set of general sequences of SU A.

In Fig. 3(b), $s_0 \rightarrow col_0$ represents that it is related to col_0. Hence, the common channel is replaced in the slots 0, 1, 2, 3 inside the general sequence s_0. Likewise, s_1 maps to the col_1, and therefore, the common channel is replaced in the slots 4, 5, 6, 7 in s_1. col_3 contains the values 9, 10, 11, 12, which are $< N$. Hence, a modular operation is carried out and after s_2, again the values are repeated starting from the initial column, i.e. col_0. The same method is also applied for SU B after choosing the permutation of general sequence s_1, s_2, s_3, s_4, and s_0.

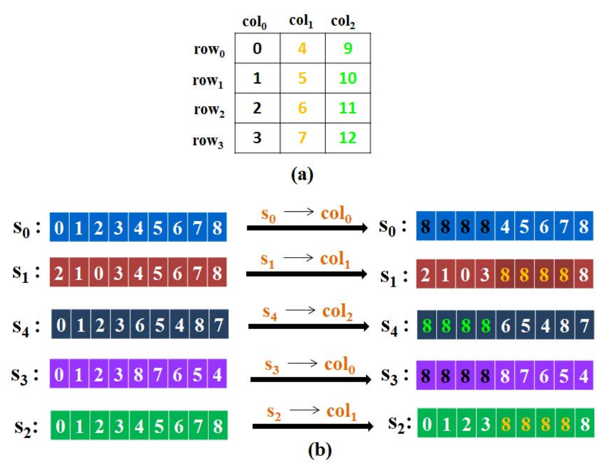

Fig. 3: Demonstration of *slot_matrix* and common channel replacement within each general sequence of SU A.

Fig. 4 shows the rendezvous between both SUs in the common available channel, i.e. on channel 8 after applying the concept of FPP. From the Fig. 4, we can observe that the degree of rendezvous is increased from 3 to 23.

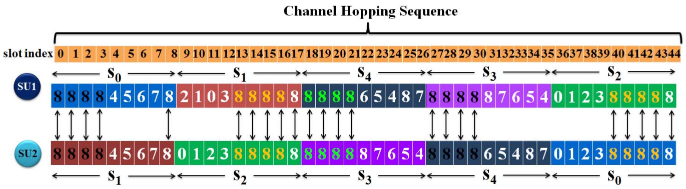

Fig. 4: Rendezvous between both SUs after applying the concept of FPP.

In ASCH, *slot_matrix* contains all the numbers starting from 0 to (n^2+n). Hence, the common channel replacement within the general sequence is done starting from the initial slot 0. If the common channel between two SUs is 0, then replacement will start from slot number 1. SUs will wait only one time slot for the rendezvous. Hence, *MTTR* of ASCH protocol is 1. According to definition, *IRI* = $t_j - t_i$, where t_i, t_j are two consecutive time slots when rendezvous occurs. The degree of replaced slot in each general sequence is always $(n+1)$. Consider s_i and s_j are two consecutive sequences. Let $0, ..., n$ be the slots of s_i, where replacement of common channel is done. Hence, the number of unutilized slots of s_i = $N-(n+1)$. In the next sequence s_j, replacement is done from the slot number $n+1$ according to the *slot_matrix*. Since, *IRI* between s_i and s_j can be calculated as $\{N-(n+1)\}+\{n+1\} = N$, *IRI* in ASCH protocol is N.

IV. Simulations

Performance of the proposed *ASCH* protocol is evaluated using OMNeT++ simulator. First, the ASCH protocol is simulated with and without considering the FPP and then its evaluation is made to compare it with some recent protocols such as *SARCH* [5], *JS* [7], *EJS* [9], in asymmetric environment. Considering t_i and t_j are the entering slot indexes of $SU1$ and $SU2$, respectively, it is assumed that t_i = t_j = 0. The impact of throughput, average TTR (ATTR), and percentage of rendezvous with respect to the number of channels are evaluated for two users considering the asymmetric and synchronous environment. In the simulation, 50 PUs and 100 SUs are deployed over an area of 1000m × 1000m. The entire network is divided into multiple time slots, where duration of each time slot is taken to be 0.02 sec. In order to achieve the guaranteed rendezvous within very small TTR value, we have fixed some channels as the common channels between any two SUs.

As shown in Fig. 5, the impact of throughput over number of channels is evaluated. It is observed that throughput is always directly proportional to the rendezvous. Protocols like *JS* and *EJS* contain one jump and one stay pattern. Due to blind rendezvous nature of both protocols, it is very difficult to find one common available channel in both stay and jump patterns, which reduces the throughput. In *SARCH*, the throughput is gradually decreases with increase in channel number. This happens due to decrease in the number of rendezvous and increase in *IRI*. The throughput of ASCH is large as compared to other protocols due to guaranteed rendezvous.

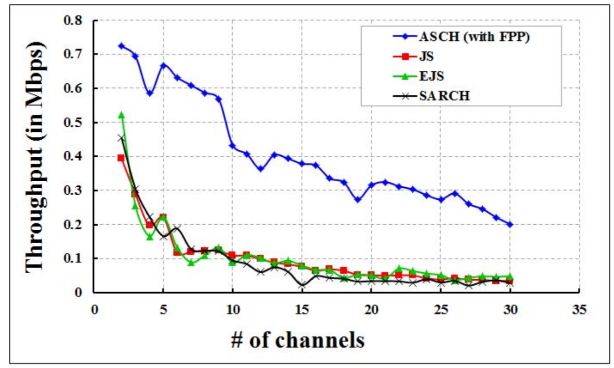

Fig. 5: Throughput (Two-users) vs. number of channels.

In Fig. 6, the comparison of *ASCH* with other protocols are made in terms of ATTR. In case of *ASCH* using FPP, the ATTR is very small as the replacement of channels depends on the *slot_matrix*, which contains the value starting from 0 to (n^2+n). Hence, the waiting time for SUs decreases. In both *JS* and *EJS*, the ATTR value is very large due to asymmetric nature of channels, which is $\approx 3P$ and $4P$, respectively. This makes SUs to wait for another round of rendezvous in common channel. In *SARCH*, the length of the sequence is $2N*(2N+1)$, where the degree of rendezvous is $\approx 2*(2N+1)$. *IRI* of *SARCH* is also longer. Therefore, a SU needs to wait more number of time slots to rendezvous, which increases the ATTR.

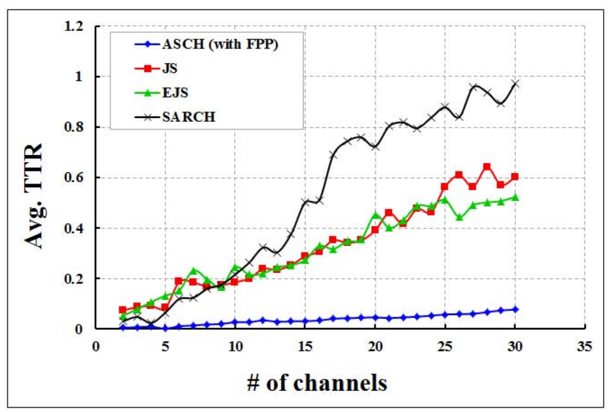

Fig. 6: ATTR vs. number of channels.

From Fig. 7, it is observed that the percentage of rendezvous is very high in ASCH as compared to other protocols. In ASCH, we have guaranteed at least $m*(n+1)$ number of rendezvous in each CH sequences, where n is the order of the FPP. In JS and EJS the percentage of the rendezvous is very small due to random selection of the channels in the jump pattern and for choosing different channels as stay pattern. In SARCH, the degree of rendezvous decreases with increase in the channel number. In SARCH, if SUs select different rotation seeds, they get a very rare chance to meet at the common channel in a CH sequence, which decreases the percentage of rendezvous.

It is to be noted that in the first scenario of ASCH, CH

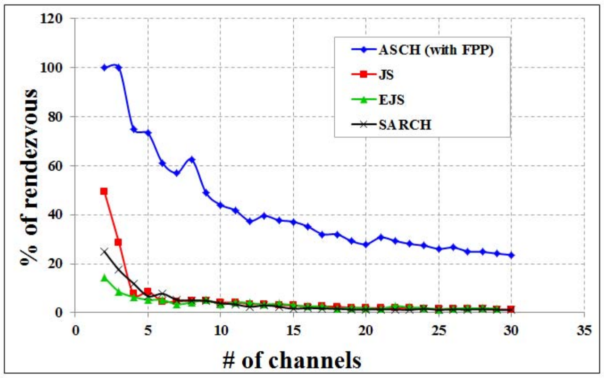

Fig. 7: Percentage of rendezvous vs. number of channels.

sequence consists of general sequences, which are modified by using the concept of FPP. In the second scenario, channel hopping sequence is created by the SUs without using the FPP, i.e. the CH sequence is only comprises with permuted general sequences. In this scenario, a single common available channel is considered between any two SUs. As shown in Fig. 8, the impact of throughput over number of channels is simulated. It is already proved that the degree of rendezvous in case of ASCH using FPP is higher as compared to the channel hopping sequence without using FPP. More rendezvous allows SUs to transmit data more frequently, which increases the throughput of the SUs. In case of ASCH using FPP, initially the throughput is very high though gradually it decreases due to decrease in the percentage of rendezvous with increase in the channel number.

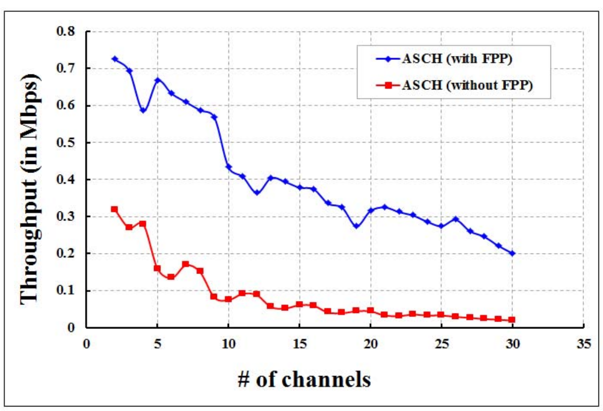

Fig. 8: Throughput of ASCH with and without using FPP.

From Fig. 9, we find that percentage of rendezvous is very high in ASCH using FPP as compared to the case of ASCH without using FPP. In the latter case, SUs get very few chances to meet, i.e. maximum *m*, where *m* is the number of general sequences at the common channel, if there is a single common channel is shared among both users. As discussed earlier, there is at least $m * (n + 1)$ number of guaranteed rendezvous in each CH sequences in case of ASCH using FPP. Although in ASCH using FPP, percentage of rendezvous decreases with channel numbers, it is always higher than the ASCH without using FPP.

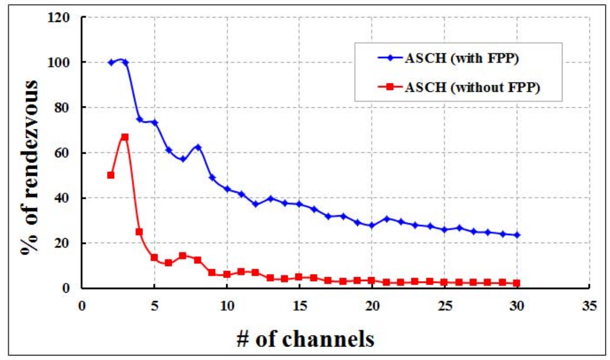

Fig. 9: Percentage of rendezvous with and without using FPP.

V. Conclusion

In this paper, a novel Asymmetric Synchronous Channel Hopping scheme is designed, which guarantees the rendezvous between the SUs with minimum value of MTTR. A new concept of Finite Projective Plane is introduced in ASCH to maximize the available common channel utilization within the entire CH sequence. By evaluating our protocol, it is found that ASCH outperforms over related asymmetric channel hopping protocols in terms of throughput, MTTR and percentage of rendezvous.

References

[1] Brandon F. Lo, "A survey of common control channel design in cognitive radio networks", *Physical Communication*, vol. 4, Issue 1, pp. 26-39, March, 2011.

[2] Sengupta, S.; Subbalakshmi, K.P., "Open research issues in multi-hop cognitive radio networks", *IEEE Magazine on Communications*, vol. 51, no. 4, pp. 168-176, April, 2013.

[3] Zaw Htike, Choong Seon Hong, Sungwon Lee, "The Life Cycle of the Rendezvous Problem of Cognitive Radio Ad Hoc Networks: A Survey", *Journal of Computing Science and Engineering, JCSE*, vol. 7, no. 2, pp. 81-88, June, 2013.

[4] Tsung-Ying Wu; Wanjiun Liao; Cheng-Shang Chang, "CACH: Cycle-Adjustable Channel hopping for control channel establishment in cognitive radio networks", *in Proc. IEEE INFOCOM*, pp. 2706-2714, May 2, 2014.

[5] Guey-Yun Chang; Wen-Hung Teng; Hao-Yu Chen; Jang-Ping Sheu, "Novel Channel-Hopping Schemes for Cognitive Radio Networks", *IEEE Transactions on Mobile Computing*, vol. 13, no. 2, pp. 407-421, Feb, 2014.

[6] I-Hsun Chuang; Hsiao-Yun Wu; Yau-Hwang Kuo, "A Fast Blind Rendezvous Method by Alternate Hop-and-Wait Channel Hopping in Cognitive Radio Networks", *IEEE Transactions on Mobile Computing*, vol. 13, no. 10, pp. 2171-2184, Oct, 2014.

[7] Hai Liu; Zhiyong Lin; Xiaowen Chu; Leung, Y.-W., "Jump-Stay Rendezvous Algorithm for Cognitive Radio Networks", *IEEE Transactions on Parallel and Distributed Systems*, vol. 23, no. 10, pp. 1867-1881, Oct, 2012.

[8] Chang, Guey-Yun; Huang, Jen-Feng; Wang, Yao-Shian, "Matrix-based channel hopping algorithms for cognitive radio networks", *IEEE Transactions on Wireless Communications*, vol. 14, no. 5, pp. 2755-2768, May, 2015.

[9] Zhiyong Lin; Hai Liu; Xiaowen Chu; Yiu-Wing Leung, "Enhanced Jump-Stay Rendezvous Algorithm for Cognitive Radio Networks", *IEEE Communications Letters*, vol. 17, no. 9, pp. 1742-1745, September, 2013.