논문 18-43-02-01 The Journal of Korean Institute of Communications and Information Sciences '18-02 Vol.43 No.02
https://doi.org/10.7840/kics.2018.43.2.209

인지 라디오 네트워크에서의 랑데뷰 알고리즘에 관한 연구

가브리엘 앙크탕◆, 김 용 철°

A Survey on Rendezvous Algorithms in Cognitive Radio Networks

Gabriel Anquetin◆, Yongchul Kim°

요 약

무선인지 네트워크 기술은 허가받은 주파수 대역에서의 혼잡한 상황을 해결하기 위한 솔루션으로 각광을 받고 있는 기술이다. 허가받지 않은 사용자 노드들(Secondary Users) 에게 허가된 사용자 노드들(Primary Users)에 방해가 되지 않는 범위에서 허가된 채널에 접속하여 데이터를 전송할 수 있도록 허용해 주는 기술이다. 그러므로 랑데뷰 알고리즘은 허가받지 않은 사용자 노드들이 서로 같은 채널에서 데이터를 교환 할 수 있도록 해주는 중요한 알고리즘이다. 본 논문에서 지금까지 연구된 랑데뷰 알고리즘들을 각각의 장단점을 비교하여 분류 하였으며, 특히 최근에 이슈가 되고 있는 재밍 공격 상황에서의 랑데뷰 알고리즘 성능을 분석하여 이를 포함한 기준으로 기존 랑데뷰 알고리즘들을 재분류하였다.

Key Words : Channel Hopping Scheme, Jamming Attacks, Rendezvous Algorithms, CRN

ABSTRACT

Cognitive Radio Network (CRN) is a solution to the congestion of the licensed channels spectrum. It allows Secondary Users (SUs) to access licensed channels and exchange data without causing interference to Primary Users (PUs). Thus, rendezvous is a fundamental step to meet and communicate between SUs. In this paper, we have classified many rendezvous algorithms created so far, using their pros and cons. Especially we analyzed the performance of rendezvous algorithm under Jamming Attacks since it has recently become an issue. We then re-classified rendezvous algorithms by including jamming resistance.

Ⅰ. Introduction

With the increasing use of wireless devices, we are facing congestion in the unlicensed spectrum whereas licensed channels include many idle channels. Spectrum sharing and regulation, introduced in [18], remains an issue and spectrum is far from being used in an efficient manner. A great response appears to be a technique called Opportunistic Spectrum Access (OSA) which assumes that the network has a hierarchical structure. Licensed users (i.e., PUs) use the network

※ 본 논문은 육군사관학교 화랑대연구소의 2018년도 연구활동비 지원을 받아 연구되었음.
◆ First Author : Korea Military Academy Department of Electrical Engineering, gabriel.anquetin@gmail.com, 정회원
° Corresponding Author : (ORCID:0000-0003-1393-8711)Korea Military Academy Department of Electrical Engineering, kyc6454@kma.ac.kr, 종신회원
논문번호 : KICS2017-11-346, Received November 13, 2017; Revised January 7, 2018; Accepted January 10, 2018

as they want while SUs equipped with cognitive radios opportunistically accede to idle channels after having sensed them without causing any interference to PUs. In order to exchange data, users have to meet on the same channel, which is the first step, and then hop among available channels in a synchronous way. The first step is called rendezvous: that is the handshaking between users. They must perform a rendezvous algorithm. Over the last decade, many researchers have studied these algorithms and main methods have emerged [1]. To have a better overview of rendezvous algorithms created so far, this paper introduces a classification of these algorithms, including the jamming attack resistance.

The main method is the Channel Hopping (CH), in which the network is time-slotted. During this process, at each time-slot users jump (hop) among available channels detected beforehand in order to rendezvous. A rendezvous is established if users jump on the same channel in the same time-slot. What is important to note is that users might not be aware of the presence of each other.

Some researchers tried to compute reliable CH algorithms using servers, called centralized controller, to help users to find each other on the network. However, major issues have appeared. The server always turns into the bottleneck: the network is easy to be jammed, and a problem on the server hits every network user. Therefore, only decentralized systems seem trustworthy and are the target of main researches. In the same way, reliable rendezvous algorithms using Common Control Channel (CCC) look unfeasible. CCC is a channel which has to be available for all users in the network (global CCC), or available in an area (local CCC), where users can exchange some data to allow an easier rendezvous. But this method which aims to facilitate rendezvous and to reduce congestion on the spectrum has the opposite result: a channel available for every user is unfeasible in experiments and increases congestion because every user has to join this channel before rendezvous. Besides, a simple jamming attack could interrupt the entire network.

In order to classify rendezvous algorithms, we need several criteria. We are in the case of decentralized systems without using CCC, which are called blind rendezvous algorithms.

- One of the most important criteria is the symmetry. Users are under the symmetric model if they have the same available channel set. In practice, we can assume that users in a very small area are under symmetric model. Yet, the most general case is the asymmetric model, in which users have different available channel sets, but at least one common channel.
- Some algorithms require time-synchronization between SUs, while the most general case is the asynchronous case. As a fact, the most common situation is the one with a timing offset between time-slots of users. This is the synchronicity criterion.
- We also have to know if the algorithm is role-based. Some algorithms require pre-assigned roles for SUs. A user must be a transmitter while another is the receiver. Thus, role-based algorithms are much more efficient than the non-role-based, called common-strategy algorithms. However, role-based algorithms are difficult to carry out in many cases because two receivers cannot rendezvous each other (or two transmitters).
- Many algorithms do not distinguish SUs so they do not require IDs. (IP addresses in that case). Yet, some algorithms use IDs of users to create different channel hopping sequences for users. So, there are anonymous algorithms and non-anonymous algorithms.
- Jamming resistance also allows us to classify algorithms. In fact, algorithms often use sequences without any randomness. It makes them vulnerable to jamming attacks whose purpose is to reduce rendezvous probability and characteristics of rendezvous algorithms.
- A recent algorithm introduces the static/dynamic criterion [16]. In a static setting, user available channels are the same during all the rendezvous process. This is an ideal case because new channels might be available during the rendezvous

process. Dynamic rendezvous algorithms are designed to take into account these channels.

We will also distinguish rendezvous algorithms with a main metric, the Time-To-Rendezvous (TTR). TTR can be defined as the number of time-slots it takes to establish the rendezvous between SUs once every SUs has begun its CH sequence. Maximum Time-To-Rendezvous (MTTR) is the TTR in the theoretical worst scenario. Thus, a rendezvous algorithm can guarantee rendezvous only if he has a finite MTTR. For example, the algorithm which randomly selects channels has an infinite MTTR because it is possible that users will not rendezvous each other, in the worst-case. Expected Time-To-Rendezvous (ETTR) is the TTR that a SU expect to require to achieve its rendezvous with another SU. It is also important, because an algorithm that has a bad MTTR can be very effective with a good ETTR. Again, random algorithm has an infinite MTTR, however this algorithm can achieve a decent ETTR. This paper aims to use criterion introduced previously to classify existing rendezvous algorithms. The second part of this paper deals with conventional algorithms, while the third part tackles algorithms with jamming attacks. Then we will summarize main ideas in the fourth part.

Ⅱ. Conventional Algorithms

To introduce various conventional algorithms, we will use the following model. We consider a network with PUs and $N>1$ SUs using one cognitive radio antenna. Let M be the number of non-overlapping licensed available channels. Let P be the smallest prime number greater than M. Let C be the whole available channel set, $C=\{c_1, c_2, ..., c_M\}$ and C_k be the channel set available for the kth user. A channel is said to be available if SUs can exchange data without causing any interference to PUs.

Every algorithm use a different channel hopping sequence. Fig. 1 illustrates the principle of this fundamental method in rendezvous algorithm.

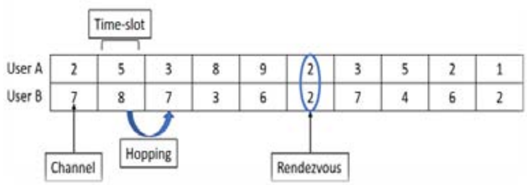

Fig. 1. Channel hopping principle

Some of the first rendezvous algorithms are those which require time-synchronisation. An algorithm named SSCH proposed by Bahl et al. [3] guarantees rendezvous under symmetric synchronous model. It is not very convenient but it is one of the first rendezvous algorithm. This algorithm was based on pairs (multiple and seed) picked by the user so that CH sequences were created with these pairs. Other pioneer algorithms are Quorum-based Channel Hopping (QCH) algorithms [2]. M-QCH and L-QCH are able to guarantee rendezvous under asymmetric and synchronous model. A-QCH can guarantee rendezvous under asynchronous model but only in networks with two channels.

Many algorithms aim to provide rendezvous without time-synchronization. This is more complex but rendezvous is more feasible in practice. Generated Orthogonal Sequences (GOS) algorithm provides rendezvous under symmetric asynchronous model, using identical pre-designed channel hopping to rendezvous. The MTTR of this algorithm is *O(M2)*, which is high under symmetric model but it is a pioneer in asynchronous algorithms [4]. Theis et al. [4] also proposed the Modular Clock (MC) algorithm and Modified Modular Clock (MMC) algorithm. The first one is for symmetric model while the second one is designed for asymmetric model. Neither of these algorithms provide a guaranteed rendezvous but they are effective in practice (ETTR is *O(P)* for MC which is good under symmetric model) and provide rendezvous under specific conditions. They are based on modulo operations with two parameters, r and P, a rate and a prime number respectively. Users hop on the channel indexed $(i+r)\%P$ where i is the index of the current channel, trying to hop on the same channel in the same time-slot.

In [5] and [6], Yang et al. introduced two algorithms named Deterministic Rendezvous Sequence (DRSEQ) and Channel Rendezvous Sequence (CRSEQ) are proposed. DRSEQ provides a guaranteed rendezvous in at most $2M+1$ time-slots under symmetric model, which is good. Yet, it does not work under asymmetric model. The main idea is to generate CH sequence of $2M+1$ indices numbered as follows : $\{1,2,\cdots,M,e,M,\cdots,2,1\}$ where e denotes an empty slot. Fig. 2 shows how rendezvous is established between two nodes. SU A repeats the same CH sequence whereas SU B creates a time offset at each pattern to enable SUs to meet each other. Blue squares indicate when rendezvous happens.

The CRSEQ guarantees rendezvous for asymmetric model in at most $P(3P-1)$ time-slots which is very decent. This algorithm is based on triangular number $T_n = n(n+1)/2$ and modulo operations. However CRSEQ is weak under symmetric model, where DRSEQ is better for example.

Ring-Walk (RW) based algorithm [7] is a well known category of rendezvous algorithm. Channels are seen as vertices in a ring, and users hop on vertices to meet each other. Different velocities are assigned to users. They hop among channels in clockwise direction so that users with higher velocities meet users with lower velocities and this is guaranteed rendezvous under both symmetric and asymmetric model. MTTR are $O(M \times N)$ and $O(M^2 \times N)$ respectively for symmetric and asymmetric cases, which is not satisfactory, but the idea can be improved. We can also note that this algorithm uses the ID of user to generate CH sequence, which is an additionnal information to provide.

SU A	1	2	3	e	3	2	1	1	2	3	e	...
SU B	1	2	3	e	3	2	1	1	2	3	e	...
		1	2	3	e	3	2	1	1	2	3	...
			1	2	3	e	3	2	1	1	2	...
				1	2	3	e	3	2	1	1	...
					1	2	3	e	3	2	1	...
						1	2	3	e	3	2	...
							1	2	3	e	3	...

Fig. 2. Channel Hopping sequence in DRSEQ algorithm

More recently, Full Diversity Channel Hopping (FDCH) algorithm was proposed in [8] by Guerra et al. It is a RW based algorithm which aims to provide a good TTR and to maximize rendezvous diversity (the number of channels in which a rendezvous may occur). FDCH can be divided into two cases : FDCH-Role-Based (FDCH-RB) and FDCH Common Strategy (FDCH-CS). The first one assumes that rendezvous is established between a transmitter and a receiver. Roles are pre-assigned to users to establish rendezvous. The idea is that a transmitter walks in clockwise direction while a receiver walks in anticlockwise direction as shown in Fig. 3.

Rendezvous is achieved in $O(M)$ time-slots under asymmetric and asynchronous model which is good, but it is due to pre-assigned roles. Blue squares illustrate sensed channels of SUs.

FDCH-CS does not use a role-based strategy and is very efficient. Prerequisite is to have two cognitive radio antennas by user. The algorithm uses the same idea, but users do not have to choose between clockwise or anticlockwise direction because the two cognitive radio antennas allow them to walk in both directions, as illustrated in Fig. 4. MTTR is $(T-1)/2$ where T is the number of vertices in the ring, so $T>M$. This is a very good MTTR, provided that SUs have two cognitive radio antennas. Blue squares illustrate sensed channels of SUs.

Jump-Stay (JS) based algorithms are also famous ones. Lin et al. [9] introduced this method, and improved it in [10] with the Enhanced Jump-Stay

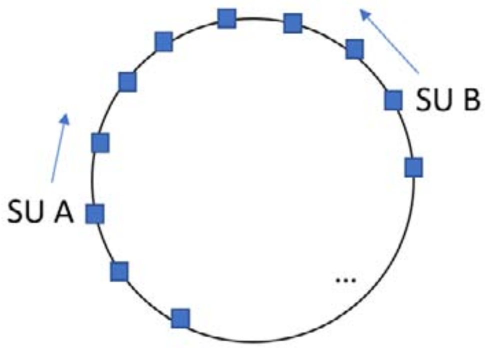

Fig. 3. Channel hopping idea in FDCH-RB algorithm

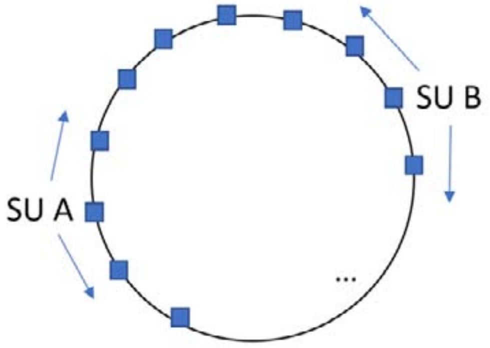

Fig. 4. Channel hopping idea in FDCH-CS algorithm

(EJS). JS based algorithms are designed for the most general case, asymmetric, asynchronous without roles. To create a JS channel hopping sequence, it requires an initial starting index i_0 in [1,P] and a step-length *r* in [1,M]. Each user generates jump patterns and stay patterns. One round is one jump pattern during *2P* time-slots and one stay pattern during *P* time-slots. During jump pattern, the user continuously hops on available channels and stays on a channel during the stay pattern. The hopping sequence begins at the starting index i_0, the user hops between channels during *2P* time-slots with the step-length *r* using modulo operations. After that jump pattern, the user stays on the channel *r* during *P* time-slots, this is the stay pattern. Fig. 5 shows an example of CH sequence, where *M=4* (i.e., *P=5*), starting index are $i_1 = 1$ and $i_2 = 3$, and step-length are $r_1 = 4$ and $r_2 = 2$. JS algorithm guarantees rendezvous in at most *3P* time-slots for symmetric model, and *3MP(P-G)+3P* time-slots where *G* is the common available channel set between users. This MTTR is decent, and JS works for asymmetric and asynchronous situations. The EJS algorithm uses the same idea but a round lasts *4P* with *3P* jump pattern and *P* stay pattern, they also add a replacement code so that in the asymmetric case MTTR is *O(P2)* instead of *O(P3)*, which is much more effective.

The Alternate Hop-and-Wait (AHW) algorithm proposed by Chuang et al. [11] has the same JS characteristics. Yet, this algorithm uses the binary ID of SU to create CH sequences. If the Less

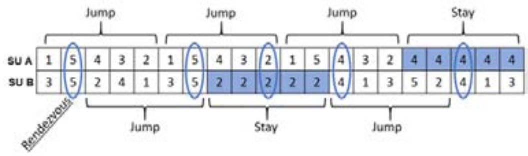

Fig. 5. EJS channel hopping example

Significant Bit (LSB) is 0, the SU performs a Wait/Hop/Hop elementary sequence (same as Stay/Jump/Jump). If the LSB is 1, the SU performs a Hop/Hop/Hop elementary sequence. Atfer being used, the LSB is right-permuted and so becomes the most significant bit. When every bit of the ID has been used, we obtain the CH sequence. We assume that $\log(D)$ is the length of the ID of the user in binary. Therefore, rendezvous is achieved in $O(P \times \log(D))$ for symmetric model and $O(P^2 \times \log(D))$ for asymmetric model, where $\log(D)$ is the length of the binary ID of a node. This algorithm is effective in practice, being the best of JS based algorithms.

Based on Disjoint Relaxed Difference Set (DRDS), the DRDS algorithm is introduced in [17]. This algorithm can achieve a constant approximation to the lower bound of rendezvous algorithms by using the equivalence between channel hopping and disjoint relaxed difference set. Moreover, this algorithm is effective for the most general model (aynchronous, asymmetric without additionnal information) and rendezvous is achieved in $O(M)$ for symmetric model and $O(M^2)$ for asymmetric model. This algorithm is also used in [16] to establish a dynamic rendezvous. Authors have designed a rendezvous algorithm taking into account channels which become available during the rendezvous process.

Recently, many algorithms have been created, tackling different issues of rendezvous algorithm to improve TTR. For example, Interleaved Sequences based on Available Channel set (ISAC) algorithm in [12] has been designed to generate CH sequences only based on the available channel set instead of the whole channel set. Conversion Based Hopping (CBH) algorithm in [13] is an oblivious blind rendezvous algorithm, it means that it works like a classic blind rendezvous algorithm but SUs do not see the same channel labels when sensing, which is a more general case.

Ⅲ. Algorithms under Jamming Attacks

3.1 Jamming Attacks

All the algorithms we have studied so far are conventional algorithms because they do not take into account jamming attacks on the network. Jamming attacks are able to mitigate the effects of any algorithm which use sequences without any randomness, that is to say every algorithms mentioned above. The basic strategy of a jamming attack is to find what is repetitive in the code, to imitate the rendezvous algorithm to jam so that rendezvous probability reduces a lot.

Oh et al. [14] proposed a Channel Detecting Jamming Attack (CDJA) which is designed to jam JS algorithm. As explained previously, JS generates CH sequences with an index channel *i* and a step-length *r*, these are parameters the jammer needs in order to act. CDJA requires that the jammer listens one or two channels. Fig. 6 and Fig. 7 explain it with one and two listening channels. In Fig. 6, the jammer has only one listening channel, so he is able to listen to channel 2 then channel 1,

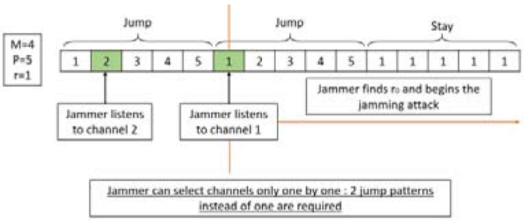

Fig. 6. Example of step-length research with one listening channel

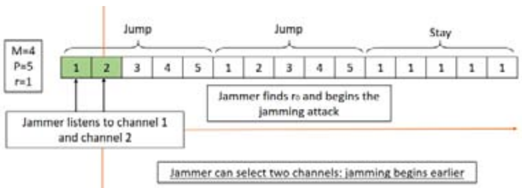

Fig. 7. Example of step-length research with two listening channels

whereas in Fig. 7 the jammer can listen to both channel 1 and 2: the jamming begins earlier. In both figures, SUs are performing JS algorithms, and parameters used are shown. As described in [9], the channel in the jump pattern is determined by: $c_i = (i_0 + t_i \times r_0 - 1)\%P + 1$. So at time t_1, the jammer listens to a user on c_1 and at time t_2 on c_2. Therefore we can find r_0 :

$$c_1 = (i_0 + t_1 \times r_0 - 1)\%P + 1 \quad (1)$$

$$c_2 = (i_0 + t_2 \times r_0 - 1)\%P + 1 \quad (2)$$

(2)-(1) gives : $((t_2 - t_1) \times r_0)\%P = (c_2 - c_1)$, thus, $r_0 = (P \times k + (c_2 - c_1))/(t_2 - t_1)$.

The jammer uses r_0 to find i_0 with (1) or (2) and is able to prevent the rendezvous from being established. As a result, the jammer only has to use (3).

$$C_{next} = (C_{last} + r_0)\%P + 1 \quad (3)$$

Concerning asymmetric and asynchronous cases, the idea is the same with minor changes. Fig. 8 illustrates the case of asynchronous CDJA. The main difference between synchronous and asynchronous cases is the fact that jammer cannot estimate the beginning of the communication. The jammer can still estimate r_0 and jam channels on the hopping sequence, but without i_0 it is impossible to find the beginning of the stay pattern which is important to be able to jam every channel available for users. As we can see in Fig. 8, the jammer must listen to the channel r_0 to detect when the user stays on r_0 two

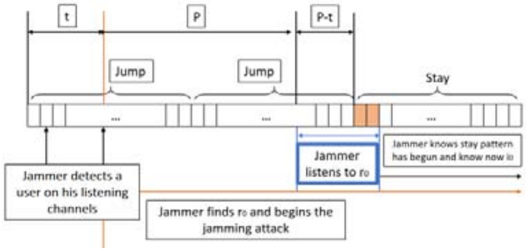

Fig. 8. Example of asynchronous research of step-length and starting index with two listening channels

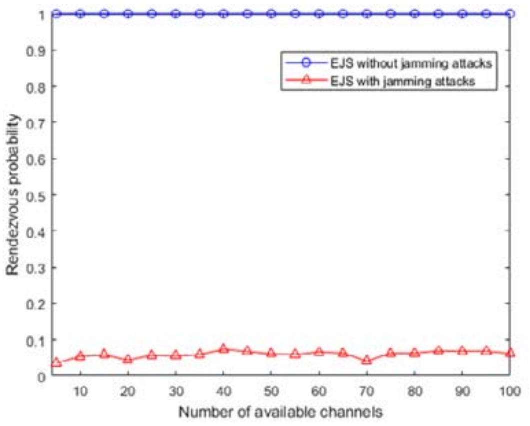

Fig. 9. Rendezvous probability for symmetric EJS with and without SCDJ attacks

time-slots in a row and be aware of the beginning of the stay pattern.

Moreover, Oh et al. [15] introduced jamming attacks against EJS, named Symmetric Channel Detecting Jamming (SCDJ) and Asymmetric Channel Detecting Jamming (ACDJ). As a fact, EJS use *4P* rounds instead of *3P* rounds like JS, so the jamming method uses the same parameters i_0 and r_0 to jam. SCDJ cannot be used for asymmetric channel because of the replacement algorithm for unavailable channels of a user which happens in asymmetric case. Fig. 9 illustrates the collapse of the rendezvous probability under jamming attacks. In this example, EJS rendezvous probability drops significantly from 100% to less than 10% because of jamming attacks. On the other hand, an algorithm which randomly picks channels is particularly strong against this attack, as expected.

3.2 Jamming Resistant Algorithms

A jamming-resistant algorithm must perform well in both jamming and non-jamming situations. The rendezvous must be guaranteed when there is no jamming attack while probability must remain high even during jamming attacks.

In [14], an algorithm named Role-based Channel Rendezvous (RCR) is proposed as a good alternative against jamming attacks. As shown in Fig. 10, the sender, SU A, generates randomly-permutated sequence of *M* channels, 5 in that case, while the

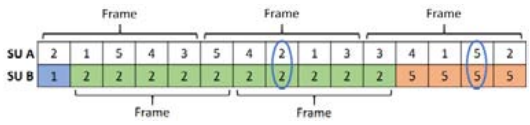

Fig. 10. An example of rendezvous in the RCR Symmetric scheme

receiver, SU B, stays on a randomly selected channel during two frames (*2M* time-slots, 10 in that case). Thus, jammers cannot find any step-length or repetitive parts in the channel hopping sequence so that rendezvous is guaranteed in at most *2M* time-slots, which is decent for a role-based algorithm.

Introduced in [15], Random Enhanced Jump-Stay (REJS) algorithm puts randomness in EJS so that it greatly improves probability of rendezvous under jamming attacks. The idea is to use random operations instead of modulo operations for the remapping and replacement steps in EJS algorithm. For example, when the CH sequence returns a channel index out of the available set of a user, a remapping with random operations is done, which makes the algorithm reliable against jamming attacks. Its MTTR is the same as EJS, and its ETTR is better than EJS, even without jamming attacks.

Ⅳ. Summary

Table I summarizes our study. We can note that time-synchronization is not required in most algorithms, only pioneer algorithms used it, because it is too much restrictive. Besides, additional information allows algorithms to improve their TTR but is sometimes not convenient for SUs. Since 2011, every algorithm works and guarantees rendezvous with both symmetric and asymmetric models, asymmetric model is necessary for large areas, where SUs have sometimes only one or two common available channels, and symmetric is very useful in small area to establish a rendezvous very quickly.

SSCH, GOS, and quorum-based algorithms (L-QCH/A-QCH) are pioneer works with unique methods but are not useful in practice. The fact of

Table 1. Main characteristics of rendezvous algorithms
X means that the scenario is impossible, log(D) is the length of the ID of the user in binary

Algorithm	Year	time synchronization	need additionnal information	work under asymetric model	guarantee rendezvous (sym/asym)	MTTR: symmetric	MTTR: asymmetric	Jamming resistant
Random	~	NO	NO	YES	NO	X	X	YES
SSCH	2004	YES	NO	NO	SYM	Unknown	X	NO
RW	2010	NO	ID and network size	YES	BOTH	O(MN)	$O(M^2N)$	NO
DRSEQ	2010	NO	NO	NO	SYM	O(M)	X	NO
CRSEQ	2010	NO	NO	YES	BOTH	$O(P^2)$	$O(P^2)$	NO
GOS	2011	NO	NO	NO	SYM	$O(M^2)$	X	NO
MC	2011	NO	NO	NO	NO	X	X	NO
MMC	2011	NO	NO	YES	NO	X	X	NO
L-QCH	2011	YES	NO	NO	SYM	Unknown	X	NO
A-QCH	2011	YES	NO	YES	BOTH	Unknown	Unknown	NO
JS	2011	NO	NO	YES	BOTH	O(P)	$O(MP^2)$	NO
EJS	2013	NO	NO	YES	BOTH	O(P)	$O(P^2)$	NO
AHW	2013	NO	ID required	YES	BOTH	O(P logD)	$O(P^2 \log D)$	NO
DRDS	2013	NO	NO	YES	BOTH	O(M)	$O(M^2)$	NO
FDCH-RB	2015	NO	Role-based	YES	BOTH	O(M)	O(M)	NO
FDCH-CS	2015	NO	2 radio antennas	YES	BOTH	O(M)	O(M)	NO
RCR	2013	NO	Role-based	YES	BOTH	O(M)	O(M)	YES
REJS	2016	NO	NO	YES	BOTH	O(P)	$O(P^2)$	YES

needing time-synchronization or being unable to rendezvous SUs in asymmetric scheme is problematic. Moreover, their average TTR are often very high. DRSEQ and CRSEQ are pioneer works but their methods ensure a fast rendezvous without time-synchronization, and DRSEQ is one of the best symmetric rendezvous algorithms for its ETTR which is very low. MC and MMC are that kind of algorithm which is unable to guarantee rendezvous but is effective in practice.

Jump-Stay has long been the reference of rendezvous algorithms. It is a simple and effective algorithm which guarantees rendezvous in any case except under jamming attacks. EJS is willingly worse than JS in symmetric model to be better in asymmetric case which is a weakness for JS.

AHW, FDCH-RB and FDCH-CS are examples of very efficient algorithms requiring additional information. Their TTR are very good but it has aprice. However, these algorithms are not jamming resistant. Jamming attacks case is important to be taken into account because it is essential to have an algorithm available in any network environment. That is why RCR and REJS are reliable algorithms. Yet, it is impossible to get TTR as good as conventional algorithms due to the randomness inside jamming resistant algorithms.

Ⅴ. Conclusion

In this paper, we introduced an overview of the up to date rendezvous algorithms. We can notice progress made over the last decade. Many methods have emerged concerning blind rendezvous algorithms, in the aim of improving TTR. These methods demand sometimes a lot of information, like users' ID, network size or demand a role-based model. Jamming attacks are a real problem when it comes to rendezvous but some algorithms include randomness in their code to overcome this issue. Thus, latest algorithms proposed allow SUs to rendezvous with asynchronous and asymmetric model in a good TTR, even under jamming attacks.

References

[1] H. Liu, Z. Lin, X. Chu, and Y.-W. Leung, "Taxonomy and challenges of rendezvous

algorithms in cognitive radio networks," in *Proc. ICNC*, pp. 645-649, 2012.

[2] K. Bian, J.-M. Park, and R. Chen, "A quorum-based framework for establishing control channels in dynamic spectrum access networks," in *Proc. MobiCom'09*, pp. 25-36, Beijing, China, Sept. 2009.

[3] P. Bahl, R. Chandra, and J. Dunagan, "SSCH: slotted seeded channel hopping for capacity improvement in IEEE 802.11 ad-hoc wireless networks," in *Proc. ACM MobiCom 2004*, pp. 216-230, Sept. 2004.

[4] N. C. Theis, R. W. Thomas, and L. A. DaSilva, "Rendezvous for cognitive radios," *IEEE Trans. Mob. Comput.*, vol. 10, no. 2, pp. 216-227, 2010.

[5] D. Yang, J. Shin, and C. Kim, "Deterministic rendezvous scheme in multichannel access networks," *Electron. Lett.*, vol. 46, no. 20, pp. 1402-1404, 2010.

[6] J. Shin, D. Yang, and C. Kim, "A channel rendezvous scheme for cognitive radio networks," *IEEE Commun. Lett.*, vol. 14, no. 10, pp. 954-956, 2010.

[7] H. Liu, Z.Y. Lin, X. W. Chu, and Y.-W. Leung, "Ring-walk based channel hopping algorithms with guaranteed rendezvous for cognitive radio networks," in *Proc. 2nd IEEE WiSARN2010-FALL, in Conjunction with IEEE/ACM CPSCom*, pp. 755-760, Dec. 2010.

[8] E. O. Guerra, V. A. Reguera, R. D. Souza, E. G. Fernandez, and M. E. Pellenz, "Systematic construction of common channel hopping rendezvous strategies in cognitive radio networks," in *J. Wireless Commun. and Netw.*, Dec. 2015.

[9] Z. Y. Lin, H. Liu, X. W. Chu, and Y.-W. Leung, "Jump-stay based channel hopping algorithm with guaranteed rendezvous for cognitive radio networks," in *Proc. IEEE INFOCOM*, pp. 2444-2452, Apr. 2011.

[10] Z. Lin, H. Liu, X. Chu, and Y. Leung, "Enhanced jump-stay rendezvous algorithm for cognitive radio networks," *IEEE Commun. Lett.*, vol. 17, no. 9, pp. 1742-1745, 2013.

[11] I. Chuang, H.-Y. Wu, K.-R. Lee, and Y.-H. Kuo, "Alternate hop-and-wait channel rendezvous method for cognitive radio networks," in *INFOCOM*, pp. 746-754, 2013.

[12] L. Yu, H. Liu, Y.-W. Leung, X. Chu, and Z. Lin, "Efficient channel-hopping rendezvous algorithm based on available channel set," *IEEE ICC*, 2014.

[13] Z. Gu, Q.-S. Hua, and W. Dai, "Fully distributed algorithms for blind rendezvous in cognitive radio networks," *Mobihoc*, pp. 155-164, Philadelphia, Pennsylvania, USA, Aug. 2014.

[14] Y.-H. Oh and D. J. Thuente, "Channel detecting jamming attacks against jump-stay based channel hopping rendezvous algorithms for cognitive radio networks," *2013 IEEE ICCCN*, pp. 1-9, Jul.-Aug. 2013.

[15] Y.-H. Oh and D. J. Thuente, "Jamming and advanced modular-based blind rendezvous algorithms for cognitive radio networks," *2016 IEEE WoWMoM*, pp. 1-10, Jun. 2016.

[16] H. Pu, Z. Gu, X. Lin, Q.-S. Hua, and H. Jin. "Dynamic rendezvous algorithms for cognitive radio network," *ICC*, pp. 1-6, May 2016.

[17] Z. Gu, Q.-S. Hua, Y. Wang, and F. C. M. Lau, "Nearly optimal asynchronous blind rendezvous algorithm for cognitive radio networks," *2013 IEEE SECON*, pp. 371-379, New Orleans, LA, 2013.

[18] S. Bhattarai, J. M. J. Park, B. Gao, K. Bian, and W. Lehr, "An overview of dynamic spectrum sharing: Ongoing initiatives, challenges, and a roadmap for future research," in *IEEE Trans. Cognitive Commun. and Netw.*, vol. 2, no. 2, pp. 110-128, Jun. 2016.

가브리엘 앙크탕 (Gabriel Anquetin)

Gabriel Anquetin was born in Rouen, Normandy, France, in 1994. Second Lieutenant at the French Military Academy, he is specialized in electrical engineering. He is currently pursuing the master degree.

김 용 철 (Yongchul Kim)

1998년 3월 : 육군사관학교 전자공학과 학사

2001년 11월 : University of Surrey, UK 전자공학과 석사

2011년 12월 : North Carolina State University, USA 전기컴퓨터 공학과 박사

2012년 2월~현재 : 육군사관학교 전자공학과 부교수

<관심분야> WiMAX, Relay Networks, Ad-hoc Networks, Wireless Jamming.