향상된 점프-스테이 알고리즘의 무작위화 개선에 대한 연구

테오 샤바트[◆], 김 용 철[°]

Exploring Randomized Improvements to the Enhanced Jump-Stay Algorithm

Chavatte Théo[◆], Yongchul Kim[°]

요 약

인지 라디오 네트워크(Cognitive Radio Networks, CRNs)는 보조 사용자(Secondary Users, SUs)가 주 사용자(Primary Users, PUs)와의 간섭을 피하면서 사용되지 않는 주파수 대역을 동적으로 접근할 수 있도록 하여 효율적인 스펙트럼 활용을 가능하게 한다. CRNs에서 중요한 도전 과제 중 하나는 비동기성, 동적 채널 가용성, 사전 조율 부족으로 인해 복잡해지는 동시 접속(rendezvous)을 달성하는 것이다. 이러한 문제를 해결하기 위해 향상된 점프-스테이(Enhanced Jump-Stay, EJS) 알고리즘은 결정론적 채널 호핑 방식을 통해 유한 시간 내에 동시 접속을 보장하며 효과적인 방법으로 평가받아 왔다. 그러나 EJS는 동적이거나 예측 불가능한 환경에서는 한계를 보이기 때문에, 이를 극복하기 위해 본 논문에서는 무작위화된 향상 점프-스테이(Randomized Enhanced Jump-Stay, REJS) 알고리즘을 제안하여 확률론적 개선을 도입해 어려운 조건에서도 유연성과 강건성을 향상시키고자 하였다. 또한, 대칭 및 비대칭 시나리오에서 EJS와 REJS를 비교하고, 동시 접속 시간(Time to Rendezvous, TTR)과 같은 지표를 통해 성능을 분석하였으며, 시뮬레이션 결과를 통해 EJS는 다수의 공통 채널이 존재하는 안정적인 환경에서 우수한 성능을 보이는 반면, REJS는 제약이 많은 환경에서 뛰어난 성능을 보이는 것을 확인할 수 있었다. 본 연구를 통해 다양한 CRN 환경에서 성능을 최적화하기 위해 EJS와 REJS를 적응적으로 결합하는 하이브리드 접근 방식의 가능성을 확인 할 수 있었다.

Key Words : Cognitive Radio, Rendezvous Algorithm, Jump-Stay Algorithm

ABSTRACT

Cognitive Radio Networks (CRNs) enable efficient spectrum utilization by allowing Secondary Users (SUs) to dynamically access unused frequency bands while avoiding interference with Primary Users (PUs). A critical challenge in CRNs is achieving rendezvous, which is complicated by asynchrony, dynamic channel availability, and lack of prior coordination. The Enhanced Jump-Stay (EJS) algorithm has been effective in addressing these challenges through deterministic channel hopping, ensuring rendezvous in finite time. However, EJS shows limitations in dynamic or unpredictable environments. To address these challenges, this paper proposes the Randomized Enhanced Jump-Stay (REJS) algorithm, which introduces probabilistic improvements to enhance

※ 본 논문은 육군사관학교 화랑대연구소의 2025년도 연구활동비 지원을 받아 연구되었음.

◆ First Author : French Military Academy Department of Electrical Engineering, theo.chavatte0501@gmail.com, 정회원
° Corresponding Author : Korea Military Academy Department of Electrical Engineering, kyc6454@gmail.com, 종신회원

논문번호 : 202412-319-E-RN, Received December 14, 2024; Revised February 28, 2025; Accepted March 28, 2025

flexibility and robustness under challenging conditions. Furthermore, it compares the performance of EJS and REJS in symmetric and asymmetric scenarios, analyzing metrics such as Time to Rendezvous (TTR). Simulation results demonstrate that while EJS performs exceptionally well in stable environments with multiple common channels, REJS excels in highly constrained conditions. This study highlights the potential for hybrid approaches that adaptively combine EJS and REJS to optimize performance across diverse CRN environments.

Ⅰ. Introduction

Cognitive Radio Networks (CRNs) offer a promising solution to address the inefficient utilization of the radio spectrum, a critical resource in the modern era of connected devices. This approach has been extensively explored, emphasizing its potential to enhance spectrum utilization in dynamic environments [1-3]. This paradigm is particularly relevant for applications in 5G and beyond, where dynamic spectrum sharing becomes crucial[4,5]. These networks allow Secondary Users (SUs) to dynamically access unused frequency bands without interfering with Primary Users (PUs), enhancing spectrum efficiency. However, a key challenge in CRNs is achieving rendezvous, the process by which two SUs synchronize on a common channel to establish communication. This challenge has been analyzed in depth, highlighting issues such as asynchrony and dynamic channel availability[5,6,9]. This task is complicated by asynchrony, dynamic channel availability, and the lack of prior coordination. Algorithms such as the Enhanced Jump-Stay (EJS) have been developed to address these issues using a deterministic approach that ensures rendezvous in a finite time[1,7].

While EJS is effective in stable environments, its performance can be limited in highly dynamic or unpredictable conditions. To address these limitations, we propose probabilistic enhancements to the EJS algorithm, incorporating randomized elements to improve its flexibility and robustness in challenging scenarios. Such adaptations are particularly relevant for large-scale underlay CRNs with dynamic spectrum access[6]. These randomized modifications aim to adapt better to variable channel conditions, offering potential improvements in rendezvous success rates and efficiency.

The rest of this paper is structured as follows. Section Ⅱ provides a detailed explanation of the Enhanced Jump-Stay (EJS) algorithm, including its channel hopping sequences and theoretical properties for achieving rendezvous. Section Ⅲ introduces the Randomized Enhanced Jump-Stay (REJS) algorithm, explaining its probabilistic modifications designed to enhance flexibility and robustness in dynamic environments. Section Ⅳ evaluates the performance of EJS and REJS through simulations, comparing their efficiency using Time to Rendezvous (TTR) as a key metric under various conditions. Finally, Section Ⅴ concludes the paper by summarizing key findings and discussing potential future improvements.

Ⅱ. EJS algorithm

The EJS method creates channel hopping (CH) sequences with jump patterns and stay patterns. This method has been extensively analyzed for its efficiency in stable environments, providing theoretical guarantees for rendezvous[5]. Such guarantees have been explored further, demonstrating deterministic properties under symmetric and asymmetric scenarios [7]. Before presenting the theoretical foundation of this algorithm, we define the total number of available channels as M and the smallest prime number greater than M as P. During a jump pattern, SUs hop among channels during P time- slots, starting from a channel whose index is i_0 (included in [1,P]). These SUs jump with a step-length r (included in [1,M]). A stay pattern is designed to stay P time-slots on the channel indexed r. The purpose of the stay period is to ensure that if another SU is hopping through channels, there is a higher probability of rendezvous occurring. By staying on a fixed channel for a designated period, the SU increases the chances of aligning with another SU that may be following a different hopping pattern. Meanwhile, the jump period serves to explore different channels systematically, allowing the SU to discover potential rendezvous opportunities across a broader range of available channels. The EJS rendezvous algorithm makes SUs jump 3 time-slots in a row and stay one time-slot on the channel *r*. In the EJS scheme, the channel hopping sequence in the jump-pattern is determined by the following formula:

$$c_i = \left(i_0 + t_i \times r - 1\right) \bmod P + 1$$

where c_i represents the channel assigned at the i-th time slot, i_0 is the initial channel index, selected within the range [1,*P*], t_i denotes the current time slot index, *r* is the step length used for hopping between channels, *P* is the smallest prime number greater than the total number of available channels, *M*. And, if $c_i > M$, we remap c_i as:

$$c_i = c_i \bmod M + 1$$

For instance, if an SU starts from channel c_0=2, step length *r*=1, and prime number *P*=5, the channel at the first time slot (*i*=1) is computed as:

$$c_1 = (2 + 1 \times 1 - 1) \bmod 5 + 1 = 3$$

similarly, for *i*=2:

$$c_2 = (2 + 2 \times 1 - 1) \bmod 5 + 1 = 4$$

This structured approach ensures a systematic channel hopping sequence while maintaining deterministic rendezvous properties.

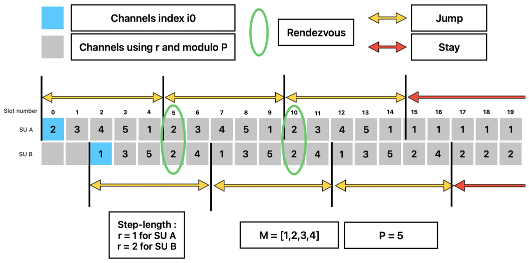


Fig. 1. EJS channel hopping example

Ⅲ. REJS Algorithm

The EJS scheme was proposed to enhance the previous JS system, reducing the Expected Time to Rendezvous (ETTR) for the asymmetric model from $O(P^3)$ to $O(P^2)$. EJS appears to be one of the best blind rendezvous algorithms for CRNs due to its non-deterministic CH sequences and its guaranteed rendezvous times for both symmetric and asymmetric models. For this reason, we chose it as a foundation to study the not well developed REJS algorithm.

The REJS algorithm, in turn, strengthens the asymmetric EJS scheme by replacing biased channel selections with random selections, both for remapping channels and for replacing unavailable channels. In the REJS algorithm, randomness is introduced in two key aspects: the selection of hopping channels and the determination of the stay pattern. Unlike EJS, which follows a strict modular-based approach for channel selection, REJS allows for probabilistic channel allocation to increase adaptability in dynamic conditions. By replacing deterministic remapping with randomized selection, REJS minimizes the risk of repeated unsuccessful rendezvous attempts and enhances performance in asymmetric scenarios.

The channel hopping sequence in REJS is formulated as follows. At each time slot *k*, the selected channel c_k is determined based on whether the previously selected channel belongs to the set of available channels for the SU. The hopping rule is given by:

$$c_k = \begin{cases} rand\_sample\,(SU_A, 1), & \text{if } k = 0 \\ (c_{k-1} + r - 1) \bmod M + 1, & \text{if } c_{k-1} \in SU_A \\ rand\_sample\,(SU_A, 1), & \text{if } c_{k-1} \notin SU_A \end{cases}$$

where c_k represents the selected channel at the k-th time slot, *r* denotes the step length used for hopping, and M refers to the total number of available channels. The function *rand_sample*(SU_A ,1) selects a random channel from the set of available channels for SU_A. If the computed channel does not belong to SU_A's available set, it is replaced with a randomly selected channel, ensuring that the sequence remains dynamic and unpredictable. Consequently, the channel c_k of the jump pattern is determined by the Algorithm 1 and 2.

```
Function REJS sequence
$c_1 = \text{randsample}(\text{SU A}, 1)$
for $k = 1 : 3P+1$
    if $c_k + r \in \text{SU A}$
        $c_{k+1} = c_k + r$
    else
        $c_{k+1} = \text{randsample}(\text{SU A}, 1)$
for $k = 3P+1 : 4P$
    $c_{k+1} = c_k$
```

Algorithm. 1. Pseudo code of REJS CH sequence

```
Function REJS rendezvous
$seqA = \text{REJSsequence}(\text{SU A}, r, M)$
$seqB = \text{REJSsequence}(\text{SU B}, r, M)$
$lenA = \text{length(seqA)}$
for $i = (1 + offset) : lenA$
    if $seqA(i) == seqB(i - offset)$
        $time = i$
return $i$
```

Algorithm. 2. Pseudo code of REJS rendezvous

In addition to modifying the hopping sequence, REJS introduces a probabilistic stay pattern. Unlike EJS, where the stay channel follows a fixed rule, REJS selects the stay channel dynamically based on the available channels of the secondary user. This process is expressed as:

$$c_s = rand\_sample(SU_B, 1)$$

where c_s represents the stay channel, and $rand\_sample(SU_B,1)$ selects a random channel from the set of available channels for SU_B. By employing this probabilistic approach, REJS reduces the likelihood of persistent mismatches between hopping sequences, thereby improving the overall rendezvous success rate. The reason for selecting a new stay channel each time is to increase the probability of rendezvous by diversifying the channel selection. If the same stay channel were used repeatedly, the rendezvous opportunity could be limited to specific conditions where another SU happens to be following a matching sequence.

To illustrate the effectiveness of REJS, consider a scenario where an SU starts at channel c_0=2, with a step length of r=1 and a total of M=4 available channels. The hopping sequence proceeds as follows:

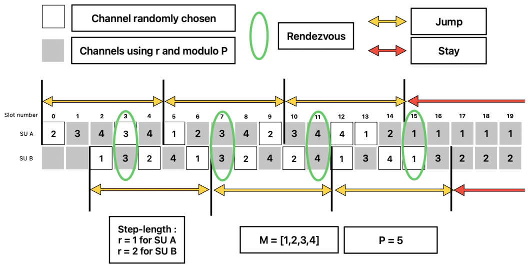
$$c_1 = (c_0 + r - 1) \bmod M + 1 = 3$$
$$c_2 = (c_1 + r - 1) \bmod M + 1 = 4$$

If the computed channel c_3 is unavailable in the set of SU_A's available channels, it is replaced by a randomly selected channel:

$$c_3 = rand\_sample(SU_A, 1)$$

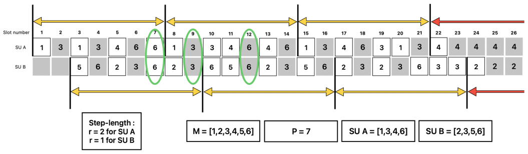
This approach ensures that the hopping sequence remains adaptable to changing conditions, allowing for a higher probability of successful rendezvous in environments with fluctuating channel availability. Examples of channel hopping sequences of REJS are depicted in Figure 2 and 3.

The introduction of randomness in REJS provides several advantages over the deterministic EJS algorithm. By incorporating probabilistic channel selection, REJS is more resilient to jamming attacks and unpredictable interference. The flexibility of randomized hopping reduces the likelihood of repeated collisions, making the algorithm particularly effective in asymmetric scenarios where common channels are scarce[9]. Furthermore, the improved adaptability al-



Slot number	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
SU A	2	3	4	3	4	1	2	3	4	2	3	4	4	1	2	1	1	1	1	1
SU B			1	3	2	4	1	3	2	4	2	4	1	3	4	1	3	2	2	2

Fig. 2. REJS channel hopping example



Slot number	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26
SU A	1	3	1	3	4	6	6	1	3	3	4	6	4	6	1	3	4	6	3	1	3	4	4	4	4	4
SU B			5	6	2	3	6	2	3	6	5	6	2	3	5	6	2	3	2	3	6	3	3	2	2	2

Fig. 3. Another REJS channel hopping example with a different step-length and number of channels

lows REJS to perform well in highly dynamic environments, where fixed hopping patterns may not be optimal.

Overall, the REJS algorithm represents a significant improvement over the traditional EJS approach by introducing a dynamic and probabilistic hopping mechanism. By modifying both the channel hopping sequence and the stay pattern, REJS enhances the robustness and flexibility of the rendezvous process in CRNs. The next section presents a comprehensive performance evaluation, comparing REJS and EJS under various conditions to assess their effectiveness in achieving successful rendezvous.

Ⅳ. Performance evaluation

We then studied our algorithm written in Matlab by modifying numerous parameters to conclude on the guarantee of rendezvous and to assess whether the performance of our algorithm was better than the existing algorithms.

It can be observed that in this Figure 4, in general, the TTR increases with M. This means that as the number of available channels increases, it becomes more difficult and takes more time for SUs to synchronize on a common channel to establish communication. In particular, the curve shows a more moderate increase in TTR when M is low (up to around 70 channels), which indicates that the algorithm handles situations with a limited number of channels fairly well. However, when M exceeds 70, the TTR increases more rapidly, suggesting that synchronization becomes increasingly complex as the number of channels grows. The shaded area between the upper and lower bounds represents the 95% confidence interval for the TTR values, allowing for the visualization of the possible deviation around the estimated average value. This area also shows that, while the algorithm remains relatively stable for low values of M, it becomes more uncertain as the number of channels increases.

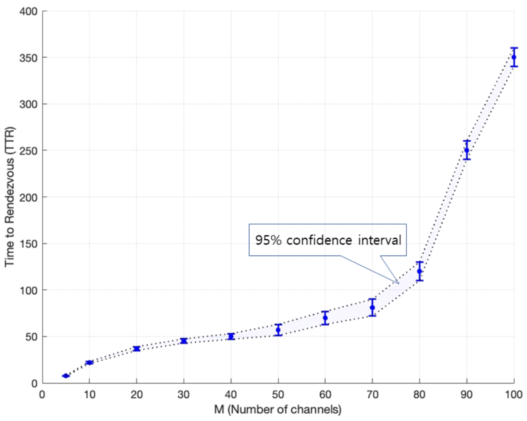


Fig. 4. Average TTR for REJS Rendezvous

The rapid increase in TTR for large M indicates a potential limitation of REJS in highly dynamic environments with a large channel pool. To mitigate this issue, one possible enhancement is to introduce a hybrid mechanism where REJS initially employs a semi-random selection method, prioritizing channels with historically higher successful rendezvous rates. Another approach is to integrate an adaptive stay duration, where the stay period dynamically adjusts based on recent unsuccessful attempts, reducing unnecessary randomization and improving synchronization efficiency. Future work can focus on optimizing these enhancements to balance adaptability and efficiency.

Figure 5 compares the performance of the EJS and REJS algorithms in terms of TTR, considering two scenarios: $G1$, where there is only 1 common channel between the two users, and $G10$, where there are 10 common channels.

In the $G1$ case, a clear difference in performance is observed between EJS and REJS. The EJS $G1$ curve

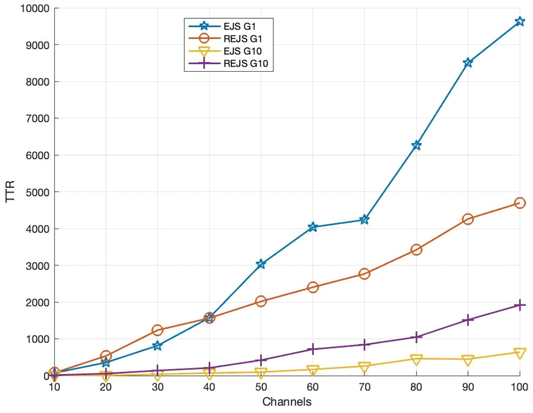


Fig. 5. Average TTR for the asymmetric EJS and REJS schemes

exhibits a steep increase in TTR as the number of channels grows, particularly when there are more than 70 channels. By the time there are 100 channels, the TTR reaches nearly 10,000, showing that EJS struggles significantly in this scenario. In contrast, REJS $G1$ shows a much more moderate increase in TTR. The randomized enhancements in REJS allow it to locate the single common channel far more efficiently than EJS, resulting in consistently lower TTR values. This demonstrates that REJS is particularly effective in challenging situations where there is only one common channel.

In the $G10$ case, the performance trend changes. Both algorithms benefit from the larger number of shared channels, leading to significantly lower TTR values compared to $G1$. However, EJS $G10$ performs better than REJS $G10$. EJS $G10$ maintains lower TTR values across all channel counts, showing its efficiency in scenarios with multiple common channels. The deterministic nature of EJS appears to be more advantageous in these cases, as it systematically explores the channels and leverages the increased availability of common resources. On the other hand, the randomness in REJS, while helpful in $G1$, seems to slightly hinder its efficiency in $G10$ scenarios, where a deterministic approach like EJS performs better. To make the above explanation easier to understand, the performance differences between the two algorithms are summarized in Table 1.

Table 1. Comparison of EJS and REJS Algorithms

Criteria	EJS	REJS
Channel Selection	Deterministic, using modular arithmetic for channel hopping.	Randomized selection, dynamically remapping unavailable channels.
TTR in Stable Environments	Lower TTR due to structured channel hopping.	Slightly higher TTR compared to EJS when multiple common channels exist.
Performance in Highly Dynamic Environments	Decreases significantly when channels are unstable or unpredictable.	More robust, as randomization reduces predictability issues.
Handling Asymmetric Scenarios	Efficient but still constrained by deterministic channel hopping.	Improves rendezvous probability by dynamically adapting to channel availability.
Jamming Resistance	Vulnerable due to its deterministic nature.	More resistant to jamming as it does not follow a predictable hopping pattern.

The results highlight the differing strengths of EJS and REJS depending on the scenario. In the $G1$ case, where there is only one common channel, REJS clearly outperforms EJS, showcasing its ability to adapt to more challenging and dynamic environments. However, in the $G10$ case, where there are multiple common channels, EJS proves to be more effective, achieving lower TTR values than REJS. This suggests that while REJS significantly outperforms EJS in $G1$ due to its adaptive nature, EJS is more efficient in $G10$, benefiting from its deterministic structure when multiple common channels are available. However, it is important to note that $G1$ represents a more challenging and realistic scenario in many CRN applications, particularly in highly dynamic or congested environments where common channels are scarce. In such cases, REJS provides a robust solution by reducing the risk of repeated failures due to fixed hopping sequences. Furthermore, EJS relies on strict deterministic patterns that can be exploited in adversarial environments. Since REJS introduces randomness in both channel hopping and stay selection, it is more resilient to jamming attacks and unpredictable interference. This adaptability makes REJS a preferable option in security-sensitive applications and dynamic spectrum access scenarios, even in environments with a larger number of common channels.

Therefore, it would be wise to create an adaptive algorithm. On the one hand, in the overlay approach, it is feasible to use only the EJS algorithm to take advantage of its performance when no jamming occurs and then switch to our REJS algorithm when there is a suspicion of jamming. On the other hand, if the underlay approach is possible in an environment, it would be more prudent to use our algorithm, as it outperforms when a large number of channels is available. Future research can further enhance its adaptability by incorporating hybrid approaches that adjust the balance between randomness and determinism based on network conditions. Additionally, REJS can be optimized to perform better in *G*10 scenarios by refining its channel selection strategy or incorporating learning-based adjustments to its hopping sequence.

Ⅴ. Conclusion

This paper explores the efficiency and limitations of the EJS and REJS algorithms for CRNs. The EJS algorithm, while effective in stable environments with multiple common channels (*G*10 scenario), shows its limitations in situations where only one common channel is available (*G*1 scenario) or when the total number of channels significantly increases. In contrast, the REJS algorithm, with its randomized enhancements, outperforms EJS in more challenging scenarios, particularly in dynamic conditions or when the number of common channels is low. The findings highlight the importance of adapting the strategy based on environmental conditions: using EJS in favorable situations with multiple common channels and no interference, and favoring REJS in the presence of jamming or under less predictable conditions. Future development should focus on creating an adaptive approach that combines the strengths of both algorithms. Incorporating adaptive cognitive techniques aligned with evolving 5G architectures could further optimize performance in real-world scenarios[10]. A hybrid strategy could leverage EJS in the absence of jamming while switching to REJS when disruptions are suspected. This work lays a solid foundation for improving performance in complex and dynamic CRN environments.

References

[1] Z. Li and D. J. Thuente, "Extensions and improvements of jump-stay rendezvous algorithms for cognitive radios," *2017 Int. Conf. Selected Topics in MoWNet*, 2017. (https://doi.org/10.1109/MoWNet.2017.8045952)

[2] M. Robin and Y. Kim, "A survey on rendezvous algorithms in cognitive radio networks under jamming attacks," *J. Inf. and Security*, vol. 21, no. 1, pp. 209-219, Mar. 2021. (https://doi.org/10.33778/kcsa.2021.21.1.209)

[3] A. Amanna and J. H. Reed, "Survey of cognitive radio architectures," in *Proc. IEEE SOUTHEASTCON 2010*, 2010, pp. 292-297. Charlotte, NC, USA, Mar. 2010. (https://doi.org/10.1109/SECON.2010.5453869)

[4] A. Ahmad, S. Ahmad, M. H. Rehmani, and N. U. Hassan, "A survey on radio resource allocation in cognitive radio sensor networks," in *IEEE Commun. Surv. & Tuts.*, vol. 17, no. 2, pp. 888-917, Secondquarter 2015. (https://doi.org/10.1109/COMST.2015.2401597)

[5] B. T. J. Maharaj and B. S. Awoyemi, *Developments in Cognitive Radio Networks: Future Directions for Beyond 5G*, Springer, 2022. (https://doi.org/10.1007/978-3-030-64653-0)

[6] Y. Liu, Y. Deng, and M. Elkashlan, "Non-orthogonal multiple access in large-scale underlay cognitive radio networks," *IEEE Trans. Veh. Technol.*, vol. 65, no. 12, pp. 11405-11409, 2018. (https://doi.org/10.1109/TVT.2016.2524694)

[7] C.-S. Chang, Y.-C. Chang, and J.-P. Sheu, "A quasi-random algorithm for anonymous rendezvous in heterogeneous cognitive radio networks," *arXiv preprint arXiv:1902.06933*, 2019. (https://doi.org/10.48550/arXiv.1902.06933)

[8] E. O. Guerra, V. A. Reguera, C. Duran-Faundez, and T. M. T. Nguyen, "Channel hopping for blind rendezvous in cognitive radio networks: A review," *Computer Commun.*, vol. 195, pp. 82-98, Dec. 2022. (https://doi.org/10.1016/j.comcom.2022.08.011)

[9] Q. Liu, Z. Lin, Z. Wei, J. Wen, C. Yi, and H. Liu, "A quinary coding and matrix structure-based channel hopping algorithm for blind

rendezvous in cognitive radio networks," *arXiv preprint arXiv:2208.10867*, 2022. (https://doi.org/10.48550/arXiv.2208.10867)

[10] H. Arslan and E. Başar, *Flexible and Cognitive Radio Access Technologies for 5G and Beyond*, Institution of Engineering and Technology, 2020. (https://doi.org/10.1049/PBTE092E)

샤바트 테오 (Chavatte Théo)

Chavatte Théo was born in Cosne Cours sur Loire, France, in 2001. 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 교수

<관심분야> Wireless networks, Satellite communications, Military tactical communication system.