Revisiting the Performance of the Modular Clock Algorithm for Distributed Blind Rendezvous in Cognitive Radio Networks

Michel Barbeau[1], Gimer Cervera[2], Joaquin Garcia-Alfaro[3]
and Evangelos Kranakis[1]

[1] School of Computer Science, Carleton University, K1S 5B6, Ottawa, Ontario, Canada {barbeau,kranakis}@scs.carleton.ca

[2] Universidad Tecnológica Metropolitana, 97279, Merida, Yuc., Mexico gimer.cervera@utmetropolitana.edu.mx

[3] Telecom SudParis, 91000, Evry, France joaquin.garcia-alfaro@acm.org

Abstract. We reexamine the *modular clock algorithm* for distributed blind rendezvous in cognitive radio networks. It proceeds in rounds. Each round consists of scanning twice a block of generated channels. The modular clock algorithm inspired the creation of the *jump-stay rendezvous* algorithm. It augments the modular clock with a *stay-on-one-channel* pattern. This enhancement guarantees rendezvous in one round. We make the observation that as the number of channels increases, the significance of the stay-on-one-channel pattern decreases. We revisit the performance analysis of the two-user symmetric case of the modular clock algorithm. We compare its performance with a random and the jump-stay rendezvous algorithms. Let m be the number of channels. Let p be the smallest prime number greater than m. The expected time-to-rendezvous of the random and jump-stay algorithms are m and p, respectively. Theis et al.'s analysis of the modular clock algorithm concludes a maximum expected time-to-rendezvous slightly larger than $2p$ time slots. Our analysis shows that the expected time-to-rendezvous of the modular clock algorithm is no more than $3p/4$ time slots.

1 Introduction

The cognitive radio network approach aims at a more intense use of the radio spectrum. Indeed, segments of radio spectrum allocated to communications services are often underused. Dynamic spectrum access has been proposed to address this issue. For instance two classes of users, primary and secondary, may be defined and have simultaneous access to a shared segment of radio spectrum. Priority is granted to the primary users. They may access and use their allocated radio spectrum segment anytime. Secondary users may be active and use the residual air time left when primary users are not active.

We assume that the radio spectrum segment is channelized. Secondary users can communicate over idle channels of the radio spectrum segment as long as they do not create interference to the primary users. Primary users may jump in anytime. Secondary users know what the channels are, but they do not know which ones among them are available. For a group of secondary users, dynamically finding idle channels and making rendezvous on a common channel, available to all, are challenging issues.

Assuming a number of possible channels, how can a group of secondary users make rendezvous on a communication channel? The problem can be addressed using a central controller, a distributed approach with a dedicated common control channel or a distributed blind rendezvous approach. We focus on the latter. Secondary users hop over a set of channels attempting to make rendezvous. Secondary users may have a common channel set (the symmetric model) or different, but non disjoint, channel sets (the asymmetric model).

Time is divided into equal length intervals called time slots. There are two conditions for a successful rendezvous: being on the same channel during a time slot and a successful protocol handshake. These two conditions can be modeled individually and independently. The probability of a successful rendezvous is the product of the probability of being on the same channel during a time slot and a successful protocol handshake. In this paper, the focus is on achieving the condition *being on the same channel during a time slot*.

We are interested in minimizing the time required by two secondary users to make rendezvous. To achieve the condition *being on the same channel during a time slot*, we consider the *modular clock rendezvous algorithm* [17]. It inspired the authors of the jump-stay rendezvous algorithm [10, 14, 12], augmenting modular clock with a *stay-on-one-channel* pattern. This addition guarantees rendezvous in one round, in the symmetric case. We make the following observation. In these algorithms, channel hopping is done according to a randomized step increment. As the number of channels increases, the probability that two different users generate different step increments grows, a requirement to make rendezvous happen during hopping. The significance of the *stay-on-one-channel* pattern in the jump-stay rendezvous algorithm drops.

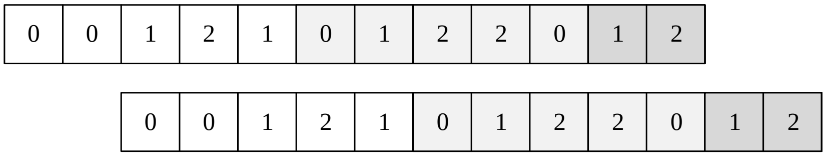
Let m denote the number of channels (a positive integer). Let p be the smallest prime number greater than m. The modular clock rendezvous algorithm proceeds in rounds consisting of two hopping phases of p time slots each. It generates blocks of p channels in accordance with the jump-stay rendezvous algorithm (*stay-on-one-channel* pattern omitted). After each round, a new block of p channels is generated. We revisit the performance analysis of the modular clock algorithm. The expected time-to-rendezvous (TTR) of the random and jump-stay algorithms are m and p time slots, respectively. Theis et al.'s analysis of the modular clock algorithm concludes a maximum expected TTR slightly larger than $2p$ time slots [17]. Our analysis shows that the expected TTR of the modular clock algorithm is no more than $3p/4$ time slots.

In Section 2, we review related work. The *modular clock rendezvous* algorithm used for our analysis is described in Section 3. The estimation of the expected TTR is done in Section 4. Simulation results are presented in Section 5. We conclude with Section 6.

2 Related Work

The performance of the channel hopping algorithms is evaluated using the TTR metric. In the two users case, from the moment both users are running, it is the number of time slots required to achieve rendezvous. An algorithm with a finite maximum TTR is said to be *guaranteed rendezvous*.

Related works include the random channel and orthogonal-sequence-based algorithms of Theis et al. [17, 7]. The random channel algorithm visits all channels in a random order. For each time slot, a channel is selected among the m channels with uniform probability. The user is tuned to that channel for the whole time slot. Under the symmetric model, the expected TTR is m time slots. Under the asymmetric model, the expected TTR is m^2/g time slots. In both models, rendezvous is not guaranteed.



0 0 1 2 1 0 1 2 2 0 1 2

0 0 1 2 1 0 1 2 2 0 1 2

Fig. 1. Orthogonal-sequence-based channel hopping.

With the orthogonal-sequence-based algorithm, channels are visited according to the same pattern by all nodes. By construction, two hopping users are eventually on the same channel. Rendezvous is guaranteed. Let $s_0, s_1, \ldots, s_{m-1}$ be a permutation of the m channels, the hopping pattern is

$$s_0, s_0, s_1, \ldots, s_{m-1}, s_1, s_0, s_1, \ldots, s_{m-1} \ldots s_{m-1}, s_0, s_1, \ldots, s_{m-1}.$$

Two hopping users are illustrated in Figure 1. In that example, m is three. The nodes make rendezvous in the third time slot, from the start of the second user. Rendezvous is guaranteed within $m(m+1)$ time slots.

Shin et al. have proposed the channel rendezvous sequence algorithm [16]. Rendezvous is guaranteed to take place. The asynchronous user ring-walk algorithm has been proposed by Lin et al. [11, 13]. Preference is given to channels with low interference to primary users. Rendezvous is not guaranteed to take place.

Bahl et al. proposed an approach for WiFi/802.11 networks [1]. Rendezvous is guaranteed to take place under the symmetric model. Krishnamurthy et al. proposed a two-phase algorithm [9]. The first phase is for neighbor discovery. It is conducted on common local channels. In the second phase, a global common channel is determined among the participating users. Bian et al. use a quorum principle [6, 4, 5]. Rendezvous is guaranteed. They have a solution for a two-channel case. Yang et al. have proposed an algorithm based on the k-shift-invariant concept that guarantees rendezvous [19].

Lin et al. authored the *enhanced jump-stay rendezvous* algorithm [10, 14, 12], hereafter called the *jump-stay rendezvous* algorithm. It is designed for multiple synchronous users with guaranteed rendezvous. We illustrate the principle with two users. Each secondary user implements a cyclic behavior consisting of four equal length phases. The first three are identical. The secondary user hops from channel-to-channel. All channels are visited. Each hop lasts for the duration of one time slot. During the last phase, the secondary user stays on the same channel for the whole duration.

Channel hopping is performed according to a pattern determined by the following procedure. Let p be the smallest prime number greater than m (the number of channels). For instance, if there are four channels, then p is five. Hopping is performed in steps of r units, with $r \in \{1, \ldots, m\}$, and starting index $i \in \{0, \ldots, p-1\}$. Each phase consists of p time slots. In the first three phases, hopping is performed for p time slots. During the fourth phase, the secondary user stays on channel r for p time slots. The total length of a cycle, called a *round*, is therefore $4p$ time slots. Let us index the time slots with variable $t = 0, 1, 2, \ldots, 4p-1$. As a function of p, r, i and t; a *channel number pattern* is generated according to the formulae

$$j = (i + tr) \mod p \text{ for } t = 0, 1, 2, \ldots, 3p-1, \tag{1}$$

$$j = r \text{ for } t = 3p, 3p+1, \ldots, 4p-1. \tag{2}$$

In the hopping phases, defined by Equation 1, the sequence of generated channel numbers is such that any window of length p time slots is a permutation of the numbers $0, \ldots, p-1$. Channel indices range from zero to $m-1$. The indices of the corresponding channels are obtained as $c = j \mod m$. Every channel is visited at least once during any interval of p time slots.

0	1	2	3	0	0	1	2	3	0	0	1	2	3	0	1	1	1	1	1

0	2	0	1	3	0	2	0	1	3	0	2	0	1	3	2	2	2	2	2

Fig. 2. Two jump-stay rendezvous sequences, round synchronized users.

Two example jump-stay rendezvous patterns are shown in Figure 2. Let us say, the upper sequence is performed by User 1 and the lower one by User 2. In both examples, m is equal to four and p is equal to five. Each line represents a cyclic behavior. Each number corresponds to a channel visited during a time slot. Each phase consists of five time slots. The channels of the three hopping phases are listed first. The constant channel of the stay phase follows. In the first example, r is equal to one. It is equal to two in the second example. The start index (i) is zero in both examples. The users make rendezvous when they are on a common channel number during the same time slot, which occurs in the first time slot in the example of Figure 2. The TTR is one. Note that in this example, users are round synchronized. Figure 3 shows another example where the sequences are the same as in Figure 2, but users are not round synchronized. With respect to User 1, User 2 starts in the fourth time slot. They make rendezvous in the third time slot time from the start of User 2. The TTR is three.

0	1	2	3	0	0	1	2	3	0	0	1	2	3	0	1	1	1	1	1			
			0	2	0	1	3	0	2	0	1	3	0	2	0	1	3	2	2	2	2	2

Fig. 3. Two jump-stay rendezvous sequences, non round synchronized users.

The initial values of the step increment r and start index i are selected at random. The index is incremented to the successor value, modulo p, after each round. Given a sequence generated with $r = r_1$ and another sequence generated with $r = r_2$, with $r_1 \neq r_2$, then any *jump pattern* window of p time slots of the first sequence has a common channel time slot with an overlapping *jump pattern* window of p time slots of the second sequence [12].

Lin et al. address the followings cases: two symmetric users, two asymmetric users and multiple-users. In companion papers, we have improved the analysis of the jump-stay rendezvous algorithm under the symmetric model [2] and developed a new analysis for the asymmetric model in [3]. Under the two-user symmetric model, the expected TTR of the jump-stay rendezvous algorithm is equal to p time slots [2]. Under the two-user asymmetric model, assuming that g is the number of common channels (less than or equal to m), the expected TTR of the jump-stay rendezvous algorithm is [3]

$$\frac{p+1}{1+g} \text{ time slots.} \tag{3}$$

The modular clock algorithm has been originally proposed by Theis et al. [17]. It is based on ideas initially introduced by DaSilva and Guerreiro [7]. It is analogous to the jump-stay rendezvous algorithm, but the stay pattern is not performed. Two-node rendezvous is guaranteed when they hop using different step increments, i.e., different values for r. Because of the absence of the stay pattern, rendezvous does not occur when they start hopping on different channels with identical step increments. When a node fails to rendezvous for $2p$ time slots, it switches to a different step increment. In the modular clock algorithm, described by Theis et al. [17], the step increment r is in $\{0, \ldots, p-1\}$. In the jump-stay rendezvous algorithm it is in $\{1, \ldots, m\}$. In both cases, the generated sequences of p channels share the same aforementioned mathematical properties. Practical evaluations of the modular clock and random algorithms have been conducted by Robertson et al. using the GNU radio framework [15].

3 The Modular Clock Algorithm

The modular clock rendezvous algorithm proceeds in rounds. Each round consists of two phases, of p time slots each. In the sequel, we use the channel number pattern formula of the jump-stay rendezvous algorithm, i.e., Equation 1. It is mathematically equivalent to the formula used for the original presentation of the modular clock algorithm. In other words, every user generates blocks of p channels, following the jump-stay algorithm, but the stay pattern is omitted. Each round consists of two times p jumps (a block of p channels). After each round, each user randomly generates a new starting index i and step length r.

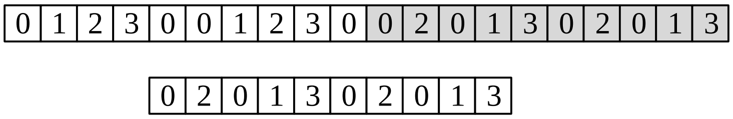


Fig. 4. Two modular clock rendezvous sequences, non round synchronized users.

An example is shown in Figure 4. The upper band represents the sequence of channels visited by User 1, the lower band the ones scanned by User 2. The sequences are as in Figures 2 and 3. The users are not *round synchronized*. In this example, User 2 starts after User 1 has started. Two rounds for User 1 are shown and one round for User 2. The first round of User 2 overlaps the first and second rounds of User 1. User 1 uses different step increments (r) for each round. The TTR is one.

The following can be observed. In the jump-stay rendezvous algorithm, for each round the probability that two users generate different step increments (their r) is proportional to the number of channels, i.e., m. The larger m is, the more likely that two users pick different step increments. As a consequence, the usage of the stay pattern becomes less significant in the performance of the algorithm. This is confirmed by the upcoming analysis and simulation results.

4 Estimation of the Expected TTR

We assume that there are two users: User 1 and User 2. They respectively use step increments r_1 and r_2. Without loss of generality, we assume that User 2 starts when or after User 1 has started. A round is made of two modular clock sequences performed by User 2. We start counting the TTR from the time slot when User 2 starts. We first state the main result of this work.

Theorem 1. *The expected TTR of the modular clock rendezvous algorithm is at most* $\frac{3p}{4}$ *time slots.*

Proof. In the upcoming Lemma 2, it is shown that the expected number of rounds is equal to one. This is because the probability of success P of a round is the parameter of a geometric random variable with mean $1/P$. $1/P$ is equal to one asymptotically in m. Furthermore, asymptotically in m there are only two cases with non-null probability (Cases 1.1 and 2.2). Their expected number of time slots required to make rendezvous are $\frac{p+1}{2}$ and $\frac{2p+1}{2}$, respectively. Using their respective probability these translate to

$$\frac{p+1}{2p} \cdot \frac{m-1}{m} \cdot \frac{p+1}{2} + \frac{p-1}{2p} \cdot \frac{m-1}{m} \cdot \frac{2p+1}{2} \text{ time slots.}$$

Asymptotically in m, this is equal to $3p/4$ time slots. We make this statement mathematically precise in the following two Lemmas.

We define the following function that is used in several mathematical expressions in the sequel:

$$S_m(k) := \sum_{l=1}^{k} \left[1 - \left(\frac{m-1}{m} \right)^{l} \right] \tag{4}$$

Lemma 1. *For any m, let p be the smallest prime number bigger than m. Then*

$$\frac{S_m(2p)}{2p} \approx \frac{1}{2} - \frac{1}{2e^2}, \tag{5}$$

asymptotically in m.

Proof. Elementary calculations on the function $S_m(k)$ yield the following identity

$$S_m(k) = k - (m-1) + (m-1) \left(\frac{m-1}{m} \right)^{k}. \tag{6}$$

We are interested in deriving the asymptotic of $S_m(k)$ when $k = 2p$. Recall that p was chosen to be the smallest prime number greater than m. Using well-known results in number theory concerning the difference between consecutive primes, it is easily seen that p is lower than $m + m^{6/11}$ (see [8], Section A9 for additional bounds and discussion). Therefore, since $\left(\frac{m-1}{m}\right)^m \to \frac{1}{e}$, as $m \to \infty$, we have

$$S_m(2p) \approx p - \frac{m-1}{e^2}, \tag{7}$$

asymptotically in m, where e denotes Euler's constant. Since $\frac{m}{p} \to 1$ as $m \to \infty$, Lemma 1 follows.

Lemma 2. *The probability of success of a round is:*

$$P \geq \frac{p+1}{2p} \cdot \frac{m-1}{m} + \frac{p-1}{2p} \cdot \frac{m-1}{m}$$

Proof. The analysis is structured into two main cases, with respect to the overlap, in time slots, between the current rounds of two users. In the first case, it is assumed that the overlap is greater than or equal to p. In the second case, it is assumed that the overlap is less than p.

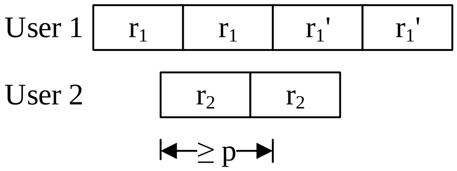


Fig. 5. Overlap is greater than or equal to p.

Case 1 (overlap is greater than or equal to p): The overlap is greater than or equal to p time slots, but lower than or equal to $2p$ time slots. This case occurs with probability

$$\frac{p+1}{2p} \tag{8}$$

because User 2 starts from the first to the $p+1$-th time slot from the beginning of User 1. This case is illustrated in Figure 7. The round of User 2 partially overlaps over the first and second rounds of User 1. In the first round of User 1, the step increment is r_1. In the second round, it is r_1'. The step increment of User 2 is r_2. There are four subcases.

Case 1.1 $(r_1 \neq r_2)$*:* In their current round, both users select different step increments, i.e., $(r_1 \neq r_2)$. The users make rendezvous in a maximum of p time slots. On average, they make rendezvous in

$$\frac{1}{p}\sum_{i}^{p} i = \frac{p+1}{2} \text{ time slots.}$$

The probability of this subcase is $\frac{p+1}{2p} \cdot \frac{m-1}{m}$, because two users pick different step increments with that probability.

Case 1.2 $(r_1 = r_2)$ *and* $(r_1' = r_2)$*:* Both users select the same step increment. Rendezvous is not guaranteed to happen. However, in User 1's round each hop with index i in $1, \ldots, 2p$ can be seen as a Bernoulli trial with probability of success, i.e., rendezvous, $1/m$ (the two users pick the same channel) and probability of failure $\frac{m-1}{m}$ (the two users pick different channels). The two users meet with probability $\frac{S_m(2p)}{2p}$. This subcase occurs with probability $\frac{p+1}{2p} \cdot \frac{1}{m^2}$.

Case 1.3 $(r_1 = r_2)$ *and* $(r_1' \neq r_2)$ *and overlap is equal to p:* This subcase is illustrated in Figure 6. Rendezvous is guaranteed to occur during the second half of User 1's round, i.e., in a maximum of $2p$ time slots. We may assume that they are equally probable. Rendezvous is made with an average of $\frac{2p+1}{2}$ time slots. This subcase occurs with probability $\frac{1}{2p} \cdot \frac{1}{m} \cdot \frac{m-1}{m} = \frac{1}{2p} \cdot \frac{m-1}{m^2}$.

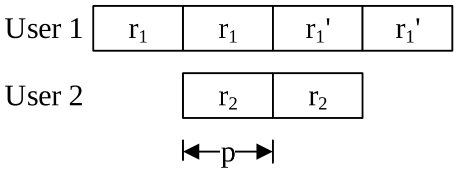


Fig. 6. Overlap is equal to p, r_1 and r_2 are equal, and r_1' and r_2 are different.

Case 1.4 $(r_1 = r_2)$ and $(r_1' \neq r_2)$ and overlap is greater than p: If $(r_1 = r_2)$ and $(r_1' \neq r_2)$, then rendezvous is not guaranteed to happen. In User 1's round each hop with index i in $1, \ldots, 2p$ can be seen as a Bernoulli trial with probability of success $1/m$ and probability of failure $\frac{m-1}{m}$. The two users meet with probability $\frac{S_m(2p)}{2p}$. This subcase occurs with probability $\frac{p}{2p} \cdot \frac{p}{m} \cdot \frac{m-1}{m} = \frac{p}{2p} \cdot \frac{m-1}{m^2}$.

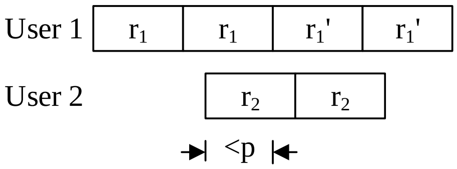


Fig. 7. Overlap is lower than p.

Case 2 (overlap is lower than p): The overlap is lower than p time slots. This case occurs with probability

$$\frac{p-1}{2p} \tag{9}$$

because User 2 starts from the first to the $p+2$-th time slot from the start of User 1. There are two subcases.

Case 2.1 $(r_1' \neq r_2)$: Rendezvous is guaranteed to occur during the second half of User 1's round, i.e., in a maximum of $2p$ time slots. We may assume that they are equally probable. Rendezvous is made with an average of $\frac{2p+1}{2p}$ time slots. This subcase occurs with probability $\frac{p-1}{2p} \cdot \frac{m-1}{m}$.

Case 2.2 $(r_1' = r_2')$: In User 1's round, each hop with index i in $1, \ldots, 2p$ can be seen as a Bernoulli trial with probability of success $1/m$ and probability of failure $\frac{m-1}{m}$. The two users meet with probability $\frac{S_m(2p)}{2p}$.

The probability of success of a round is:

$$\begin{aligned} P = {} & \frac{p+1}{2p}\left(\frac{m-1}{m} + \frac{1}{m^2}\cdot\frac{S_m(2p)}{2p}\right) \\ & + \frac{m-1}{m^2}\left(\frac{1}{2p} + \frac{p}{2p}\cdot\frac{S_m(2p)}{2p}\right) \\ & + \frac{p-1}{2p}\left(\frac{m-1}{m} + \frac{1}{m}\cdot\frac{S_m(2p)}{2p}\right) \end{aligned}$$

Asymptotically in m, only Cases 1.1 and 2.2 are significant. We can therefore derive the following lower bound on P:

$$P \geq \frac{p+1}{2p}\cdot\frac{m-1}{m} + \frac{p-1}{2p}\cdot\frac{m-1}{m}$$

5 Evaluation

Figure 8 plots the TTRs obtained with an OMNeT++ [18] simulation of the jump-stay, random and modular clock algorithms, for two-user scenarios. 95% confidence intervals are shown as small horizontal bars. On the x-axis, the number of channels m ranges from 10 to 100 channels. On the y-axis, the mean TTR is plotted as a function of the number of channels for two users. Numbers obtained through simulations are labelled *Jump-stay (simulations)*, *Random (simulations)* and *Modular clock (simulations)*. The expected TTR (ETTR), calculated using the analytical models, is also plotted for the jump-stay, random and modular clock algorithms. The analytical expected TTR for the jump-stay algorithm is labelled *Jump-stay (ETTR)*. It is calculated using expression $\frac{p+1}{1+g}$ time slots, i.e., Equation 3. Simulations results are slightly better. The analytic expected TTR for the random algorithm, i.e., m time slots (Section 2), is labelled *Random (ETTR)*. The analytical expected TTR for the modular clock algorithm is labelled *Modular clock (ETTR)*. It is calculated using equation $\frac{3p}{4}$ time slots, i.e., Theorem 1. The simulations results are slightly better than the analytic model. For the jump-stay and modular clock algorithms, simulations yield better results than the analytic models. It means that there are slightly more rendezvous opportunities than what the analytical models can capture. Analytical models provide upper bounds. Our simulation confirms that the expected TTR of the modular clock algorithm is no more than $3p/4$ time slots. Simulations performance from worst to best are with random, jump-stay and modular clock algorithms.

6 Conclusion

We have revisited the performance of the *modular clock rendezvous* algorithm. We compared with the performance of the jump-stay rendezvous algorithm. In contrast, the modular clock algorithm does only two hopping phases, of p time slots each. Each round consists of two phases. Rendezvous is not guaranteed. However, our analysis and simulation confirm that as the number of channels increases, the relevance of the stay pattern in the jump-stay rendezvous algorithm drops. Better performance can be expected with the modular clock algorithm. Theis et al.'s analysis of the modular clock algorithm concludes a maximum expected TTR slightly larger than $2p$. Our analysis concludes that the expected TTR of the modular clock algorithm is no more than $3p/4$. This has been confirmed through simulation.

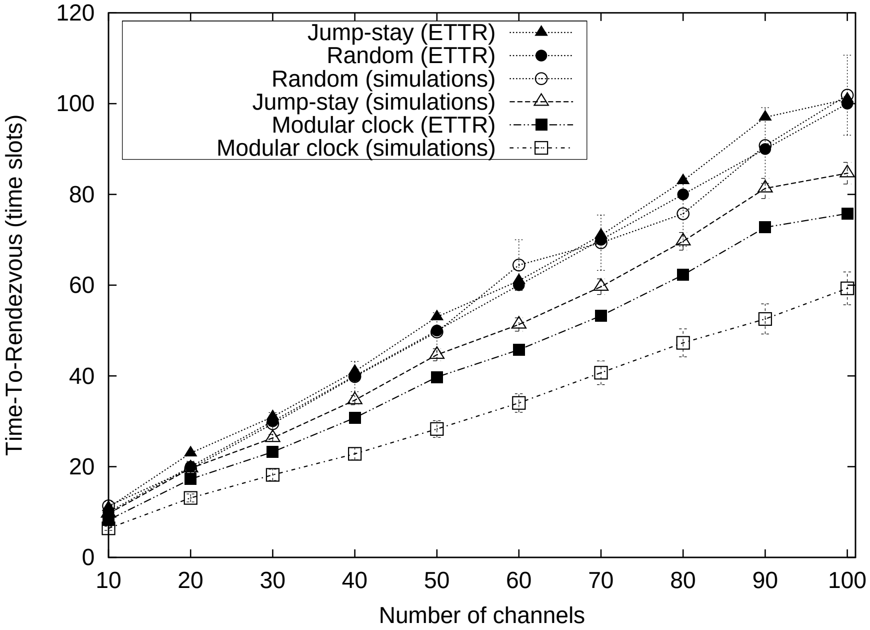


Fig. 8. Mean and expected TTR for the jump-stay, random and modular clock algorithms (with two users, 10 to 100 channels).

Acknowledgments

We acknowledge financial support from Natural Sciences and Engineering Research Council of Canada, Spanish Ministry of Science (projects CONSOLIDER INGENIO 2010 CSD2007-0004 ARES and TIN2011-27076-C03-02 CO-PRIVACY) and Innovation and Ministry of Education of Mexico (PROMEP).

References

1. P. Bahl, R. Chandra, and J. Dunagan. SSCH: slotted seeded channel hopping for capacity improvement in IEEE 802.11 ad-hoc wireless networks. In *Proceedings of the 10th annual international conference on Mobile computing and networking*, MobiCom, pages 216–230, New York, NY, USA, 2004. ACM.
2. M. Barbeau, G. Cervera, J. Garcia-Alfaro, and E. Kranakis. A new analytic model for the cognitive radio jump-stay algorithm. In *IFIP Wireless Days*, pages 1–3, November 2013.
3. M. Barbeau, G. Cervera, J. Garcia-Alfaro, and E. Kranakis. A new analysis of the cognitive radio jump-stay algorithm under the asymmetric model. In *IEEE ICC 2014 - Cognitive Radio and Networks Symposium*, 2014.
4. K. Bian and J.-M. Park. Asynchronous channel hopping for establishing rendezvous in cognitive radio networks. In *Proceedings of IEEE INFOCOM*, pages 236–240, 2011.
5. K. Bian and J.-M. Park. Maximizing rendezvous diversity in rendezvous protocols for decentralized cognitive radio networks. *IEEE Transactions on Mobile Computing*, 12(7):1294–1307, 2013.
6. K. Bian, J.-M. Park, and R. Chen. A quorum-based framework for establishing control channels in dynamic spectrum access networks. In *Proceedings of the 15th annual international conference on Mobile computing and networking*, MobiCom, pages 25–36, New York, NY, USA, 2009. ACM.
7. L.A. DaSilva and I. Guerreiro. Sequence-based rendezvous for dynamic spectrum access. In *3rd IEEE Symposium on New Frontiers in Dynamic Spectrum Access Networks (DySPAN)*, pages 1–7, 2008.
8. R. Guy. *Unsolved problems in number theory*, volume 1. Springer, 3rd edition, 2004.
9. S. Krishnamurthy, M. Thoppian, S. Kuppa, R. Chandrasekaran, N. Mittal, S. Venkatesan, and R. Prakash. Time-efficient distributed layer-2 auto-configuration for cognitive radio networks. *Computer Networks*, 52(4):831–849, 2008.
10. Z. Lin, H. Liu, X. Chu, and Y.-W. Leung. Jump-stay based channel-hopping algorithm with guaranteed rendezvous for cognitive radio networks. In *Proceedings of IEEE INFOCOM*, pages 2444–2452, April 2011.
11. Z. Lin, H. Liu, X. Chu, and Y. .W Leung. Ring-walk rendezvous algorithms for cognitive radio networks. *Ad-Hoc and Sensor Wireless Networks*, 16(4):243–271, 2012.
12. Z. Lin, H. Liu, X. Chu, and Leung Y.-W. Enhanced jump-stay rendezvous algorithm for cognitive radio networks. *IEEE Communications Letters*, pages 1–4, 2013.

13. H. Liu, Z. Lin, X. Chu, and Y.-W. Leung. Ring-walk based channel-hopping algorithms with guaranteed rendezvous for cognitive radio networks. In *IEEE/ACM Int'l Conference on Cyber, Physical and Social Computing (CPSCom), Green Computing and Communications (GreenCom)*, pages 755–760, 2010.
14. H. Liu, Z. Lin, X. Chu, and Y.-W. Leung. Jump-stay rendezvous algorithm for cognitive radio networks. *IEEE Transactions on Parallel and Distributed Systems*, 23(10):1867–1881, 2012.
15. A. Robertson, L. Tran, J. Molnar, and E.-H.F. Fu. Experimental comparison of blind rendezvous algorithms for tactical networks. In *IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM)*, pages 1–6, 2012.
16. J. Shin, D. Yang, and C. Kim. A channel rendezvous scheme for cognitive radio networks. *IEEE Communications Letters*, 14(10):954–956, 2010.
17. N.C. Theis, R.W. Thomas, and L.A. DaSilva. Rendezvous for cognitive radios. *Mobile Computing, IEEE Transactions on*, 10(2):216–227, 2011.
18. A. Varga and R. Hornig. An overview of the OMNeT++ simulation environment. In *1st International conference on Simulation tools and techniques for communications, networks and systems & workshops (Simutools)*, 2008.
19. D. Yang, J. Shin, and C. Kim. Deterministic rendezvous scheme in multichannel access networks. *Electronics Letters*, 46(20):1402–1404, 2010.

1. F. Cuppens, N. Cuppens-Boulahia, and J. Garcia-Alfaro. Misconfiguration management of network security components. In *7th International Symposium on System and Information Security (SSI 2005)*, pages 1–10, 2005.
2. F. Cuppens, N. Cuppens-Boulahia, J. Garcia-Alfaro, T. Moataz, and X. Rimasson. Handling stateful firewall anomalies. In *IFIP International Information Security Conference (SEC 2012); Information Security and Privacy Research*, pages 174–186. Springer, June 2012.
3. J. Garcia-Alfaro, F. Autrel, J. Borrell, S. Castillo, F. Cuppens, and G. Navarro-Arribas. Decentralized publish-subscribe system to prevent coordinated attacks via alert correlation. In *6th International Conference on Information and Communications Security (ICICS 2004)*, pages 223–235. Springer, October 2004.
4. J. Garcia-Alfaro, M. Barbeau, and E. Kranakis. Analysis of Threats to the Security of EPC Networks. In *Communication Networks and Services Research Conference, 2008. CNSR 2008. 6th Annual*, pages 67–74. IEEE, May 2008.
5. J. Garcia-Alfaro, M. Barbeau, and E. Kranakis. Security threats on epc based rfid systems. In *Fifth International Conference on Information Technology: New Generations, 2008 (ITNG 2008)*, pages 1242–1244. IEEE, 2008.
6. J. Garcia-Alfaro, M. Barbeau, and E. Kranakis. Secure localization of nodes in wireless sensor networks with limited number of truth tellers. In *Seventh Annual Conference on Communication Networks and Services Research (CNSR'09)*, pages 86–93. IEEE, 2009.
7. J. Garcia-Alfaro, N. Boulahia-Cuppens, and F. Cuppens. Complete analysis of configuration rules to guarantee reliable network security policies. *International Journal of Information Security*, 7(2):103–122, April 2008.
8. J. Garcia-Alfaro, F. Cuppens, and N. Cuppens-Boulahia. Towards filtering and alerting rule rewriting on single-component policies. In *International Conference on Computer Safety, Reliability, and Security (SAFECOMP 2006)*, pages 182–194. Springer, September 2006.
9. J. Garcia-Alfaro, F. Cuppens, and N. Cuppens-Boulahia. Aggregating and deploying network access control policies. In *Second International Conference on Availability, Reliability and Security (ARES 2007)*, pages 532–542. IEEE, April 2007.
10. J. Garcia-Alfaro and G. Navarro-Arribas. A Survey on Detection Techniques to Prevent Cross-Site Scripting Attacks on Current Web Applications. In *International Workshop on Critical Information Infrastructures Security (CRITIS 2007)*, volume 5141, pages 287–298. Springer, October 2007.
11. J. Melia-Segui, J. Garcia-Alfaro, and J. Herrera-Joancomarti. J3gen: A prng for low-cost passive rfid. *Sensors*, 13(3):3816–3830, 2013.
12. S. Preda, F. Cuppens, N. Cuppens-Boulahia, J. Garcia-Alfaro, and L. Toutain. Dynamic deployment of context-aware access control policies for constrained security devices. *Journal of Systems and Software*, 84(7):1144–1159, July 2011.
13. S. Preda, F. Cuppens, N. Cuppens-Boulahia, J. Garcia-Alfaro, L. Toutain, and Y. Elrakaiby. Semantic context aware security policy deployment. In *4th International Symposium on Information, Computer, and Communications Security (ASIACCS 2009)*, pages 251–261. ACM, March 2009.
14. S. Preda, N. Cuppens-Boulahia, F. Cuppens, J. Garcia-Alfaro, and L. Toutain. Model-driven security policy deployment: property oriented approach. In *International Symposium on Engineering Secure Software and Systems*, pages 123–139. Springer, February 2010.