HAL
open science

Channel selection using a multiple radio model

Michel Barbeau, Gimer Cervera, Joaquin Garcia-Alfaro, Evangelos Kranakis

To cite this version:

Michel Barbeau, Gimer Cervera, Joaquin Garcia-Alfaro, Evangelos Kranakis. Channel selection using a multiple radio model. Journal of Network and Computer Applications (JNCA), 2016, 64 (C), pp.113 - 123. 10.1016/j.jnca.2016.01.021 . hal-01332350

HAL Id: hal-01332350

https://hal.science/hal-01332350v1

Submitted on 15 Jun 2016

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Channel Selection Using a Multiple Radio Model[1]

Michel Barbeau[a], Gimer Cervera[b], Joaquin Garcia-Alfaro[c], Evangelos Kranakis[a]

[a] *School of Computer Science, Carleton University, K1S 5B6, Ottawa, Ontario, Canada Email: {barbeau,kranakis}@scs.carleton.ca*
[b] *Universidad Tecnológica Metropolitana, 97279, Merida, Yuc., Mexico Email: gimer.cervera@utmetropolitana.edu.mx*
[c] *Telecom SudParis, CNRS Samovar UMR 5157, Evry, France Email: joaquin.garcia-alfaro@acm.org*

Abstract

How can a group of distributed secondary users make rendezvous on one among a set of available channels, whose exact content is a priori unknown to the participants? Let us assume that secondary users scan the set of channels, attempting to make rendezvous with each other. Each user has several radios that are concurrently used to achieve rendezvous. We propose two rendezvous algorithms for users equipped with several radios each. We study in detail the multiple user case and the asymmetric case, where the users have different but overlapping channel sets. The performance of the algorithms are analyzed and evaluated through simulation. Equations modeling the worst case performance and expected performance are developed.

Keywords: Channel selection, cognitive radio network, cognitive wireless network, dynamic spectrum access, rendezvous.

1. Introduction

The demand for wireless continues to grow. Wireless traffic is increasing. Devices, such as smart phones and tablets, are numerous and bandwidth hungry. The numbers of wireless users, devices and applications are all booming as occupants of some of the segments of the radio spectrum. Radio spectrum is a limited natural resource. Lack of available radio spectrum is an

[1]This work is an extended and revised version of our former paper that developed a solution for a two-user, two-radio per user model [1].

issue with respect to the introduction of new applications. Indeed, the radio spectrum from nine KHz to 275 GHz has been entirely allocated to various services. In theory, there is no room for new services and accommodating growth. This is dubbed the *spectrum crunch* problem. Nevertheless, not all allocated bandwidth is currently being intensively used. For instance, a limited number of frequencies allocated to television, space exploration and defense are occupied every-time, everywhere. Measurement experiments observed a remarkably low usage of the radio spectrum. For example, McHenry et al. [2] have concluded that over 80% of the allocated spectrum is unused. The cognitive radio paradigm aims at improving the radio spectrum usage efficiency and support of the expected growth of wireless traffic.

Opportunistic spectrum access is a cognitive radio approach. It works on the assumption that certain radio bands are allocated to a primary service (e.g., television) and a secondary service (e.g., computer networks). There are primary users and secondary users. A secondary user cannot cause harmful interference to the transmissions of a primary user. Opportunistic spectrum usage is a medium access model for secondary users. Primary users may access the wireless medium anytime. Secondary users must always monitor activities of primary users. They can only use residual air time. Secondary users must relinquish channels to primary users when the latter become active. Spectrum utilization can be improved by opportunistically transmitting in spectrum holes. An important question is: where are the spectrum holes? There are two approaches for finding them: database and sensing. In the database approach [3, 4], secondary users query a database to find channels that are available for their operation. This approach requires a database-server infrastructure and a communication protocol between the secondary users and servers. In the sensing approach, secondary users observe the spectrum. They uncover unoccupied channels.

To be able to network together, secondary users meet and agree on one common channel. In the sequel, it is assumed that the secondary users are synchronous. Time is divided in slots of equal length. A rendezvous occurs within one time slot. There are two conditions for a successful rendezvous: a successful protocol handshake and being on the same channel during a time slot. These two conditions can be considered separately. They can be modeled individually and independently. The probability of a successful rendezvous is the product of the probability of a successful protocol handshake and probability of being on the same channel during a time slot. The focus of this paper is on the latter aspect. We address the problem of finding a com-

mon channel by secondary users, on which they can network. The problem of finding and selecting a common channel can be approached using either a central controller, a dedicated common control channel or a distributed blind rendezvous technique. A blind rendezvous technique may use channel scanning. Each secondary user scans a set of channels looking for a rendezvous with a peer. Participating users may all have a common channel set, in the symmetric case, or a different, but non disjoint channel set, in the asymmetric case. The goal is to make the secondary users rendezvous on a common channel in a minimum number of time slots.

The problem addressed specifically in this paper is enabling communications for a group of secondary users by making rendezvous on an available channel. We assume that each secondary user scans the set of channels, attempting to make rendezvous with other secondary users. Time is divided into equal length intervals called time slots. During one time slot, each user is tuned into one or several channels, simultaneously. Two or several users make rendezvous when they are all tuned into a common and same channel during a time slot. Most of the research works conducted so far on this problem assume a single radio per user. We assume that each user has several radios that are concurrently used to achieve rendezvous with other users. With the current software-defined radio technology, multi-radio operation is perfectly doable.

Three cognitive radio paradigms have been identified [5], namely, underlay, overlay and interweave. They refer to the model of spectrum usage by secondary users with respect to primary users. In the underlay model, secondary users are allowed to transmit until interference created to primary users remains below a threshold [6, 7, 8]. In the overlay model, because of their transmission technique, secondary user transmissions have no impact on the performance of primary users. In the interweave model, secondary users detect non-occupied spectrum segments and use them to communicate. The work presented in this paper falls into the interweave category.

We present two rendezvous algorithms, with a *bidirectional* behavior, for users that have $2k$ radios each, where k is a non-null positive integer. The algorithms differ in the way they are initialized. In the first algorithm, $2k$-*point algorithm*, each user picks $2k$ random starting channels. In the second algorithm, k-*point algorithm*, each user chooses k random starting channels. Let m be the number of available channels (assumed to be an odd number without loss of generality). The performance of our algorithms is determined by the Time-To Rendezvous (TTR) measure. We study first the two-user

symmetric case, which means that the two users share exactly the same channel set. For the two-user symmetric case, the first algorithm achieves worst case performance in $m-1$ time slots. The second algorithm achieves worst case performance in $(m-1)/2$ time slots, but at the expense of an additional constraint, i.e., users must start running the algorithm at the same time slot. Their expected performance is almost the same, $\lceil \frac{m}{2k+1} \rceil$ or $\lceil \frac{m}{2k+2} \rceil$ time slots, asymptotically in k. Next we study, the more general multiple-user asymmetric case, which means that users may hold different channels sets, but with at least one channel in common. Equations modeling the worst case performance and expected performance are developed for all cases.

In Section 2, we review related work. We study a randomized algorithm in Section 3. The bidirectional algorithm exploiting several radios per user is described in Sections 4 and 5. We compare the theoretical performance of our algorithms with related algorithms in Section 6. Simulation results are presented in Section 7. We conclude with Section 8.

2. Related Work

The problem of finding and selecting a common channel, by secondary users, can be approached using either a central controller, a dedicated common control channel or a distributed blind rendezvous technique. The blind rendezvous technique may use channel scanning. Each secondary user scans a set of channels looking to make rendezvous with a peer. Participating users may have a common channel set, under the symmetric model, or different, but non disjoint, channel sets, under the asymmetric model. The performance of the channel scanning 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 work includes the random channel and orthogonal-sequence-based algorithms of Theis et al. [9, 10]. The random channel algorithm visits all channels in a random order. For each time slot, a channel is selected among the available channels with uniform probability. The user is tuned on to that channel for the whole time slot. Rendezvous is not guaranteed. The asynchronous user ring-walk algorithm has been proposed by Lin et al. [11, 12]. 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 [13]. Rendezvous is guaranteed to take place under the symmetric model. Krishnamurthy et al. proposed a two-phase algorithm [14]. 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 [15, 16, 17]. 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 [18]. Lin et al. authored the (enhanced) jump-stay rendezvous algorithm [19, 20, 21]. It is designed for multiple users with guaranteed rendezvous. The modular clock algorithm was originally proposed by Theis et al. [9]. It is based on ideas initially introduced by DaSilva and Guerreiro [10]. It is analogous to the jump-stay rendezvous algorithm, but the stay pattern is not performed [22]. Two-node rendezvous is guaranteed when they scan using different step increments. Practical evaluations of the modular clock algorithm, and random algorithm, have been conducted by Robertson et al. using the GNU radio framework [23]. More recent related contributions include the ones described in the papers of Chang and Huang [24], Reguera et al. [25], Gu et al. [26] and Chang et al. [27].

All the aforementioned works assume a single radio per user. In this paper, we assume that each user has two or more radios that can be used simultaneously to achieve rendezvous. Yu et al. [28] have conducted research in that direction. They proposed the *role-based parallel sequence* (RPS) algorithm where users are equipped with multiple radios. In fact, each user has one dedicated radio and several general radios. The dedicated radio stays on a specific channel for a number of time slots. Then, switches to another one in a round-robin manner. The remaining general radios scan all the available channels, in parallel. For example, when a user has k general radios, k channels are simultaneously scanned per time slot. All available channels are scanned in a *parallel round-robin manner*. The number of radios may vary from user to user. The greater the number of general radios, the faster the scan of all available channels. Given a pair of users, it is expected that rendezvous is made between the dedicated radio of one user and one of the general radios of the other. Both the asymmetric and symmetric models are considered. Our work is compared with the work of Yu et al. in Sections 6 and 7.

3. Randomized Algorithm

Let m, a non-null positive integer, be the number of channels. Let C be the set of m channels $\{0, 1, \ldots, m-1\}$. Let k, a non-null positive integer, be the number of radios. Let $\{R_1, \ldots, R_k\}$ be the set of k radios. Let c be a non-empty subset of set C. The cardinal of c, i.e., $|c|$, is lower than or equal to k. An assignment of channels to the k radios is a surjective function f from the domain $\{R_1, \ldots, R_k\}$ to the co-domain c. Function f determines the channel to which each radio is tuned.

Lemma 1. *Let f be a channel assignment surjective function from the domain $\{R_1, \ldots, R_k\}$ to the co-domain $c \subseteq C$, with $1 < |c| \leq k$. The number of different possibilities for channel assignment function f is determined by the product $S(k, |c|)|c|!$, where*[2]

$$S(k, |c|) = \frac{1}{|c|!} \sum_{i=0}^{|c|} (-1)^i \binom{|c|}{i} (|c| - i)^k. \tag{1}$$

Proof. It follows from the fact that the number of surjective functions from a set of cardinal k to a set of cardinal $|c|$ is equal to $S(k, |c|)|c|!$, where $S(k, |c|)$ is the Stirling number of the second kind. It counts the number of partitions of a set of size k into $|c|$ nonempty, disjoint subsets. Further, it is known that $S(k, |c|)$ is defined as in Equation (1). □

Let us consider the following randomized algorithm. There are two users, $i = 1, 2$. Each user i is equipped with k radios. Each radio is represented by the symbol $R_{i,j}$, for $j = 1, 2, \ldots, k$. The two users are synchronous. Time is divided in slots of equal length. A rendezvous between the two users occurs within one time slot. For every time slot, each user randomly and independently selects a channel assignment function f_i, defined from domain $\{R_{i,1}, \ldots, R_{i,k}\}$ to a channel non-empty subset c_i of C. We have that $|c_i|$ is lower than or equal to k, that is to say, the number of channels is not greater than the number of radios. Success is obtained when at least one pair of radios (one from each user) are tuned to the same channel. That is to say, $f_1(q)$ is equal to $f_2(r)$, for q and r in $\{1, 2, \ldots, k\}$. Note that if the number of channels m is lower than the total number of radios, i.e.,

[2] The symbol "!" denotes the factorial operator.

$2k$, then the problem can be solved in one time slot. It suffices to select f_1 and f_2 such that they are injective. In that case, they map two radios, one from each user, to a common channel because of the limited number of possibilities. In the sequel, we are interested in the case where m is greater than or equal to $2k$. Intuitively, each radio may very well repeatedly select channel assignment functions such that rendezvous is impossible, they do not map two radios, from different users, to a common channel. By nature, a randomized algorithm does not guarantee that all channels are visited, including the ones where rendezvous might happen. The following theorem formally shows the expected TTR of the randomized algorithm

Theorem 1. *Le us consider two users $i = 1, 2$. Each user i is equipped with k radios $R_{i,j}$, with $j = 1, 2, \ldots, k$ and $2k \leq m$. The two users are synchronous. Time is divided in slots of equal length. For every time slot, each user randomly and independently selects a channel assignment function f_i defined from domain $\{R_{i,1}, \ldots, R_{i,k}\}$ to a non-empty channel subset c_i of C, with $|c_i| \leq k$. The two users make rendezvous in a time slot when $f_1(q)$ is equal to $f_2(r)$, for q and r in $\{1, 2, \ldots, k\}$. The expected TTR (ETTR) of the randomized algorithm is*

$$\frac{1}{1 - \frac{1}{m^{2k}} \sum_{i,j=1,\ldots,k} \binom{m}{i} S(k,i) i! \binom{m-i}{j} S(k,j) j!} \text{ time slots.} \tag{2}$$

Proof. Let us consider the following experiment. User 1 selects at random a non-empty set of channels $c_1 \subseteq C$, with $|c_1| \leq k$. User 2 selects at random a non-empty set of channels $c_2 \subseteq C$, with $|c_2| \leq k$. What is the probability that $c_1 \cap c_2 \neq \emptyset$? That is to say, they succeed to make rendezvous. Observe that the probability of failing to make rendezvous is

$$q = \Pr[c_1 \cap c_2 = \emptyset] \tag{3}$$

$$= \Pr[c_1 \cap c_2 = \emptyset \ \& \ |c_1| = i \ \& \ |c_1| = j] \tag{4}$$

$$= \frac{1}{m^{2k}} \sum_{i,j=1,\ldots,k} \binom{m}{i} S(k,i) i! \binom{m-i}{j} S(k,j) j!. \tag{5}$$

The factor $1/m^{2k}$ counts the number of possible functions pairs, i.e., f_1 and f_2, where each function is from a domain of cardinal k to a co-domain of

cardinal m. In the summation, the first two multiplicands count the number of subsets of cardinal i, chosen form a set of cardinal m, times the number of injective functions from a domain of cardinal k to a co-domain of cardinal i. The last two multiplicands count the number of subsets of cardinal j, chosen form a set of cardinal $m-i$, times the number of injective functions from a domain of cardinal k to a co-domain of cardinal j. The summation counts the total number of surjective functions pair f_1 and f_2 with empty co-domain intersection. Therefore, we conclude that the probability of a successful rendezvous is

$$p = \Pr[c_1 \cap c_2 \neq \emptyset] \tag{6}$$

$$= 1 - q \tag{7}$$

$$= 1 - \frac{1}{m^{2k}} \sum_{i,j=1,\ldots,k} \binom{m}{i} S(k,i) i! \binom{m-i}{j} S(k,j) j!. \tag{8}$$

Each time slot can be seen as a Bernoulli trial with probability of success, i.e., rendezvous, p and probability of failure q. The sequence of Bernoulli trials is continued until the first success, the TTR, which counts the number of trials, is a geometric random variable with $\Pr[TTR = t] = q^{t-1}p$. The average TTR is then,

$$E[TTR] = \frac{1}{p} = \frac{1}{1 - \frac{1}{m^{2k}} \sum_{i,j=1,\ldots,k} \binom{m}{i} S(k,i) i! \binom{m-i}{j} S(k,j) j!} \text{ time slots.} \tag{9}$$

This proves the theorem. □

Our analysis of the randomized case is consistent with the conclusions of Yu et al. [29]. In their paper, although, they analyze a randomized algorithm where the channel assignment function is always injective. In contrast, our analysis does not make this assumption. The maximum TTR of the randomized algorithm is infinity, since we have shown in Theorem 1 that during every trial, the probability of failing to make rendezvous (q) is always nonzero. In terms of ETTR, the randomized algorithm outperforms all other algorithms. However, it does not guarantee rendezvous. In the sequel, we concentrate on algorithms that guarantee rendezvous. Their ETTR is not better than the ETTR of the randomized algorithm because they implement some *exhaustive search* strategies to achieve the *guaranteed rendezvous* property. In that

context, the performance of the randomized algorithm is a baseline case for the purposes of comparisons. Using Equation 2, Figure 1 plots the ETTR

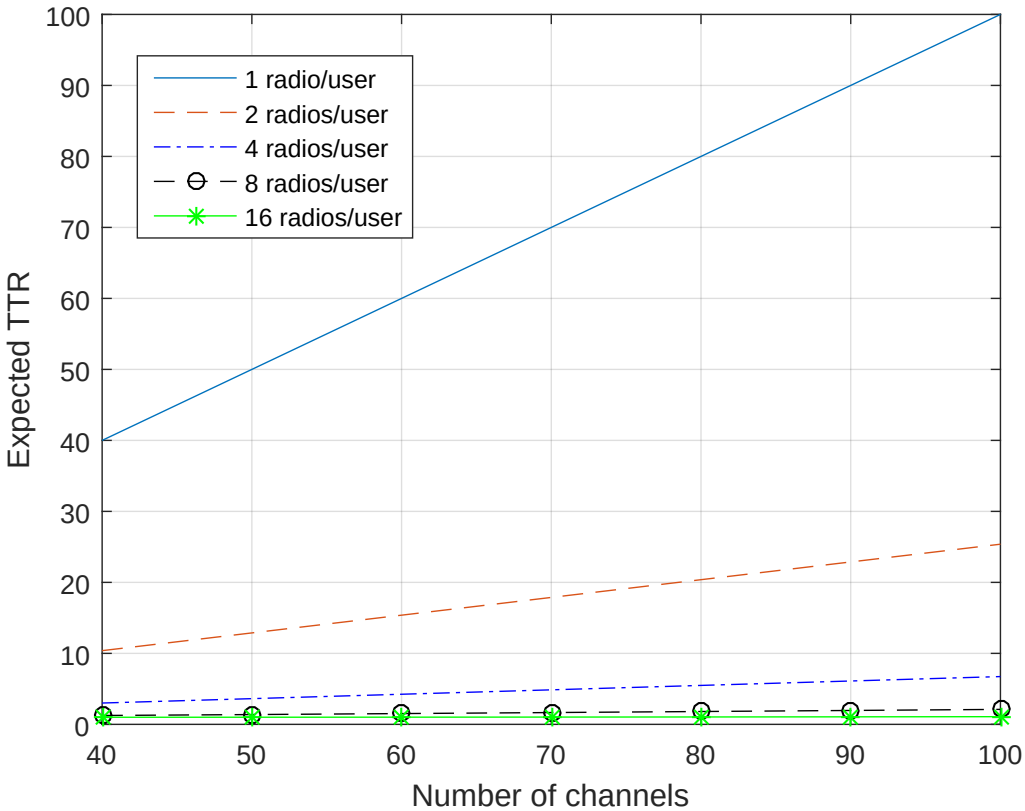


Figure 1: ETTR (in time slots) for the randomized algorithm, for one, two, four, eight and 16 radios per user.

(in time slots) for the randomized algorithm, for one, two, four, eight and 16 radios per user. This plot highlights the benefits of using simultaneously multiple radios. Each time, doubling the number of radios increases the performance by more than a factor of two.

It is assumed that users are synchronized on time slots. Practically, this can be achieved using the clock signal from the global positioning system. The start of every new time slot can be marked by the start of every second, decisecond, centisecond or millisecond.

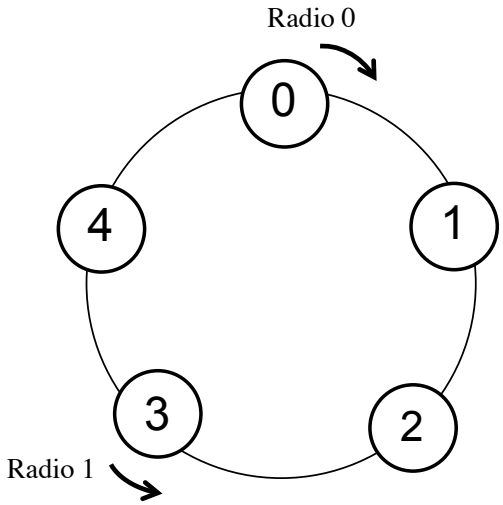


Figure 2: A ring with m equal to five channels.

4. Bidirectional Algorithms with Multiple Radios

We present our guaranteed rendezvous algorithms. Let m be the number of available channels, a positive integer. With the *bidirectional algorithm*, the channel numbers are arranged consecutively on a ring of size m, as in Figure 2. In this example, there are five channels: 0, 1, 2, 3 and 4. Each radio implements the following behavior. Scanning of the channels is done either in the clockwise (CW) or counterclockwise (CCW) direction. Let us assume that there are two users and that each user has two radios. Radio 0 of user 1 is tuned to channel zero, scanning the channel ring CW. Radio 1 of user 2 is tuned to channel three, scanning the channel ring CCW. The distance d is the number of edges separating the two radios on the channel ring, scanning toward each other. In this example, d is three edges, an odd number. With m channels, d is in $0, 1, \ldots, m-1$. Each radio scans one channel per time slot. In Figure 3, after two time slots, the two radios mutually crossover. They are tuned to channels one and two. However, this time the distance is four edges, an even number. After scanning for two additional time slots, they make rendezvous on channel four.

In the sequel, we present two rendezvous algorithms for two users equipped with $2k$ radios each, where k is a positive integer. The algorithms differ in the way they are initialized. In the first algorithm, $2k$ random starting channels are chosen. In the second algorithm, each user chooses k random starting

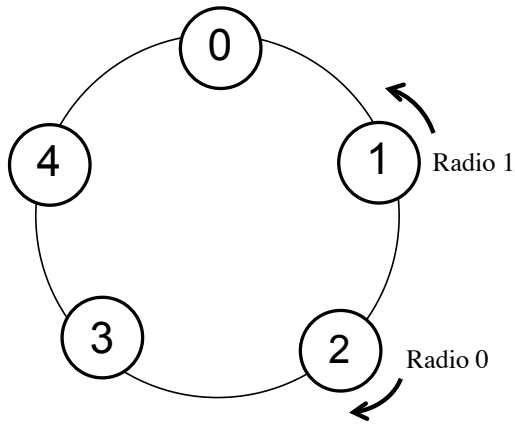


Figure 3: Crossover after two time slots, rendezvous is made after scanning for two more time slots.

channels.

The two users are indexed with variable $i = 1, 2$. Each user i is equipped with radios, say $R_{i,2j-1}$ and $R_{i,2j}$, with $j = 1, 2, \ldots, k$. Success is obtained when at least one pair of radios (one from each user) accomplishes rendezvous, i.e., they are tuned to a common channel.

Note that the TTR can be measured either in number of steps (denoted with the random variable H) or in number of time slots (denoted with the random variable TS). Regardless of the way it is being measured, the magnitude of the TTR is not affected. H and TS differ only by a constant term, namely $TS = H + 1$. This simple observation simplifies the proofs. We tacitly make use of it in the sequel.

4.1. 2k−Point Algorithm

The first procedure is called the $2k$-*point algorithm* (Algorithm 1). Each user i in $1, 2$, for each radio pair $R_{i,2j-1}, R_{i,2j}$, with j in $1, 2, \ldots, k$, randomly and independently selects two start channels in $0, 1, 2, \ldots, m-1$, denoted as $c_{i,2j-1}$ and $c_{i,2j}$. The radio $R_{i,2j-1}$ scans CW from channel $c_{i,2j-1}$, while the radio $R_{i,2j}$ scans CCW from channel $c_{i,2j}$. Scanning is performed until rendezvous is achieved by the two users.

Theorem 2. *Let us assume that m is an odd integer. Algorithm 1 (*$2k$-point algorithm*) always accomplishes rendezvous in at most $m-1$ time slots.*

Algorithm 1 $2k$-Point Algorithm: User $i = 1, 2$

for $j = 1, 2, \ldots, k$
  select start channels $c_{i,2j-1}$ and $c_{i,2j}$ randomly and independently
  scan CW with radio $R_{i,2j-1}$ starting from $c_{i,2j-1}$
  scan CCW with radio $R_{i,2j}$ starting from $c_{i,2j}$
stop when meet on same channel with other user

Furthermore, rendezvous occurs in at most $\left\lceil \frac{m}{2k+1} \right\rceil$ *expected number of steps, asymptotically in* m.

Proof. Each user has $2k$ radios. For $j = 1, 2, \ldots, k$, the radio pair $R_{1,2j-1}, R_{1,2j}$, of the first user, is coupled with the radio pair $R_{2,2j-1}, R_{2,2j}$, of the second user, in the following manner. Radio $R_{1,2j-1}$ is paired with radio $R_{2,2j}$ and radio $R_{1,2j}$ is paired with radio $R_{2,2j-1}$. It follows that the radios in the pair $R_{1,2j-1}, R_{2,2j}$ scan in opposite directions starting from channels $c_{1,2j-1}, c_{2,2i}$, respectively. Also, the radios in the pair $R_{1,2j}, R_{2,2j-1}$ scan in opposite directions starting from channels $c_{1,2j}, c_{2,2j-1}$, respectively. Let d_j be the distance in time edges between $c_{1,2j-1}$ and $c_{2,2i}$, and d'_j be the distance in edges between $c_{1,2j}$ and $c_{2,2j-1}$. We have that $d_j, d'_j \in \{0, 1, 2, \ldots, m-1\}$. Because m is odd, note that when d_j, d'_j are even, their maximum value is $m-1$. When they are odd, their maximum value is $m-2$.

There are four cases to consider.

1. If both d_j, d'_j are even, then both pairs accomplish rendezvous in $\frac{d_j}{2}$ and $\frac{d'_j}{2}$ steps, respectively.
2. If d_j is even and d'_j is odd then one pair accomplishes rendezvous in $\frac{d_j}{2}$ steps. The other pair crosses over in $\frac{d'_j+1}{2}$ steps and makes rendezvous in additional $\frac{m-1}{2}$ steps. Giving a total of $\frac{d'_j+1}{2} + \frac{m-1}{2} = \frac{m+d'_j}{2}$ steps.
3. Similarly, if d_j is odd and d'_j is even, then one pair accomplishes rendezvous in $\frac{m+d_j}{2}$ steps and the other in $\frac{d'_j}{2}$ steps.
4. Finally, if both d_j, d'_j are odd then the pairs accomplish rendezvous in $\frac{m+d_j}{2}$ and $\frac{m+d'_j}{2}$ steps.

In all cases, the number of steps is not greater than $m-1$. We conclude that rendezvous is accomplished within at most either $m-1$ steps or m time slots.

We now look at the expected number of time slots. Let H_j be a random variable measuring the number of steps until rendezvous, as a function of the initial distance d_j in edges. The previous discussion shows that the following identity is valid:

$$H_j = \begin{cases} d_j/2 & \text{if } d_j \text{ is even} \\ (m+d_j)/2 & \text{if } d_j \text{ is odd} \end{cases} \tag{10}$$

Equation (10) determines the value of H_j as a function of the distance d_j in edges. The random variables H'_j is defined analogously.

The random variables $H_1, H_2, \ldots, H_k$ and $H'_1, H'_2, \ldots, H'_k$ are independent and identically distributed. For any value h realized by the random variable H_j (h in $0, 1, \ldots, m-1$), we have that

$$\begin{aligned} \Pr[H_j = h] &= \Pr[H_j = h \ \&\ h \leq (m-1)/2] + \Pr[H_j = h \ \&\ h \geq (m+1)/2] \\ &= \frac{1}{2}\Pr[H_j = h \mid h \leq (m-1)/2] + \frac{1}{2}\Pr[H_j = h \mid h \geq (m+1)/2] \\ &= \frac{1}{2m} + \frac{1}{2m} \\ &= \frac{1}{m}. \end{aligned}$$

This implies that $\Pr[H_j \geq h] = \frac{m-h-1}{m}$. Since the random variables H_j and H'_j are independent and identically distributed, we have that

$$\begin{aligned} E\left[\min_{1\leq i\leq k} \min\{H_j, H'_j\}\right] &= \sum_{h=0}^{m-1} \Pr\left[\min_{1\leq i\leq k} \min\{H_j, H'_j\} \geq h\right] \\ &= \sum_{h=0}^{m-1} \prod_{j=1}^{k} \Pr\left[H_j \geq h\right] \Pr\left[H'_j \geq h\right] \\ &= \sum_{h=0}^{m-1} (\Pr\left[H \geq h\right])^{2k}. \end{aligned}$$

We conclude that

$$
\begin{aligned}
E\left[\min_{1\le i\le k}\min\{H_j, H'_j\}\right] &= \sum_{h=0}^{m-1}\left(\frac{m-h-1}{m}\right)^{2k} \\
&= \frac{1}{m^{2k}}\sum_{h=0}^{m-1}(m-h-1)^{2k} \\
&= \frac{1}{m^{2k}}\sum_{i=0}^{m-1} i^{2k} \\
&\le \frac{m}{2k+1} \\
&\le \left\lceil \frac{m}{2k+1} \right\rceil \text{ steps,}
\end{aligned}
$$

asymptotically in m.[3] This completes the proof of Theorem 2. □

The analysis in Theorem 2 implicitly assumes that both users start running the algorithm at the same time slot. This assumption simplifies the analysis. It is not required. Although, in the following algorithm the same assumption is present and required. The reward is a significant improvement in the worst case performance, $(m-1)/2$ steps, for the Algorithm 2, versus $m-1$ steps for Algorithm 1.

4.2. k-Point Algorithm

The second procedure is called the *k-point algorithm* (Algorithm 2). Each user i in $1, 2$, for each radio pair $R_{i,2j-1}, R_{i,2j}$, with j in $1, 2, \ldots, k$, randomly and independently selects a start channel in $0, 1, 2, \ldots, m-1$, denoted as $c_{i,j}$. Both radios in the pair start on the same channel. The radio $R_{i,2j-1}$ scans CW, while the radio $R_{i,2j}$ scans CCW. Scanning is performed until rendezvous is achieved by the two users.

Theorem 3. *Let us assume that m is an odd integer. Algorithm 2 (*k-point algorithm*) accomplishes rendezvous in at most $(m-1)/2$ steps. Further, it accomplishes rendezvous in $\left\lceil \frac{m}{2k+2} \right\rceil$ expected number of steps, asymptotically in m.*

[3] The $\lim_{n\to\infty} \frac{\sum_{i=0}^{n} i^p}{n^{p+1}/(p+1)}$ is equal to one.

Algorithm 2 k-Point Algorithm: User $i = 1, 2$

for $j = 1, 2, \ldots, k$
  select start channel $c_{i,j}$ randomly and independently
  scan CW with radio $R_{i,2j-1}$ starting from $c_{i,j}$
  scan CCW with radio $R_{i,2j}$ starting from $c_{i,j}$
stop when meet on same channel with other user

Proof. For $j = 1, 2, \ldots, k$, the radio pair $R_{1,2j-1}, R_{1,2j}$, of the first user, is coupled with the radio pair $R_{2,2j-1}, R_{2,2j}$, of the second user, in the following manner. Radio $R_{1,2j-1}$ is paired with radio $R_{2,2j}$. Radio $R_{2,2j-1}$ is paired with radio $R_{1,2j}$. It follows that the radio pair $R_{1,2j-1}, R_{2,2j}$ scan in opposite directions starting from channels $c_{1,j}$ and $c_{2,j}$. Also, the radio pair $R_{1,2j}, R_{2,2j-1}$ scan in opposite directions starting from channels $c_{1,j}$ and $c_{2,j}$.

First, we look at the worst-case TTR. According to the direction of movement, the start distance between the first radio pair is d_j edges, with d_j in $0, 1, \ldots, m-1$. It is $m - d_j$ steps for the second radio pair. Let us first assume that d_j is even. Note that in this case the maximum value for d_j is $m - 1$. The first pair accomplishes rendezvous in $\frac{d_j}{2}$ steps. The second pair crosses over in $\frac{m-d_j+1}{2}$ steps and make rendezvous in additional $\frac{m-1}{2}$ steps, thus giving a total of $\frac{m-d_j+1}{2} + \frac{m-1}{2} = m - \frac{d_j}{2}$ steps. Because the maximum value for d_j is $m - 1$, we have that the term $\frac{d_j}{2}$ is smaller than or equal to the term $m - \frac{d_j}{2}$. The first pair meets first in a maximum of $\frac{m-1}{2}$ steps.

Similarly, if d_j is odd then $m - d_j$ is even. Note that in this case the maximum value for d_j is $m - 2$. Using the previous observations, the first radio pair accomplishes rendezvous in $\frac{d_j+1}{2} + \frac{m-1}{2} = \frac{m+d_j}{2}$ steps while the second pair makes it in $\frac{m-d_j}{2}$ steps. The term $\frac{m-d_j}{2}$ is smaller than the term $\frac{m+d_j}{2}$. In all cases, the number of steps is not greater than $\frac{m-1}{2}$. We conclude that rendezvous is accomplished within at most $\frac{m-1}{2}$ steps.

We look at the expected number of time slots. As a function of the initial distance d_j in edges, let H_j be a random variable measuring the number of steps until rendezvous. The previous discussion shows that the following identity holds true:

$$H_j = \begin{cases} d_j/2 & \text{if } d_j \text{ is even} \\ (m - d_j)/2 & \text{if } d_j \text{ is odd} \end{cases} \tag{11}$$

Equation (11) determines the value of H_j as a function of the distance d_j in

edges. Let the random variable D_j be the distance between $c_{1,j}$ and $c_{2,j}$.

The random variables $H_1, H_2, \ldots, H_k$ are independent and identically distributed. Similarly for the random variables $D_1, D_2, \ldots, D_k$. For the sake of simplicity, in the sequel we denote a pair of variables H_j, D_j as H, D. Observe that $\Pr[D = d] = \frac{2}{m}$, for any $d \leq (m-1)/2$. However, since by assumption m is odd, for any integer $d \leq m$, d is even if and only if $m - d$ is odd. Therefore for any value h realized by the random variable H (h in $0, 1, \ldots, (m-1)/2$), either the event $D = 2h$ or event $D = m - 2h$ is valid, in other words:

$$H = h \Leftrightarrow \begin{cases} D = 2h & \text{if } D \text{ is even} \\ D = m - 2h & \text{if } D \text{ is odd} \end{cases}$$

That implies that $\Pr[H = h] = \frac{2}{m}$, for all h. From this observation, we calculate the distribution of H,

$$\begin{aligned} \Pr[H > h] &= \sum_{i=h+1}^{(m-1)/2} \Pr[H = i] \\ &= \sum_{i=h+1}^{(m-1)/2} \frac{2}{m} \\ &= \left(\frac{m-1}{2} - h\right) \frac{2}{m} \\ &= \frac{m - 2h - 1}{m} \end{aligned} \tag{12}$$

as well as the expected value of H,

$$\begin{aligned} E[H] &= \sum_{h=0}^{(m-1)/2} h \cdot \Pr[H = h] \\ &= \frac{2}{m} \sum_{h=0}^{(m-1)/2} h \\ &= \frac{(m-1)(m+1)}{4m} \\ &= \frac{m}{4} - \frac{1}{4m} \text{ steps.} \end{aligned}$$

We are interested in calculating the expected number of steps until rendezvous by at least one radio pair. Rendezvous is accomplished when the first

radio pair (one from each user) meets at a channel of the ring. Therefore the expected number of steps until rendezvous does not exceed the following quantity

$$E\left[\min_{1\le i\le k} H_j\right].$$

Since the random variables H_j are independent and identically distributed, we conclude that

$$\begin{aligned}
E\left[\min_{1\le i\le k} H_j\right] &= \sum_{h=0}^{(m-1)/2} \Pr\left[\min_{1\le i\le k} H_j > h\right] \\
&= \sum_{h=0}^{(m-1)/2} \prod_{i=1}^{k} \Pr\left[H_j > h\right] \\
&= \sum_{h=0}^{(m-1)/2} \left(\Pr\left[H > h\right]\right)^k.
\end{aligned}$$

It remains to evaluate this sum. Using Equation (12), we conclude that

$$\begin{aligned}
E\left[\min_{1\le i\le k} H_j\right] &= \sum_{h=0}^{(m-1)/2} \left(\frac{m-2h-1}{m}\right)^k \\
&= \frac{1}{m^k} \sum_{h=0}^{(m-1)/2} (m-2h-1)^k \\
&= \frac{1}{m^k} \sum_{\substack{i=0 \\ i \text{ even}}}^{m-1} i^k \\
&= \frac{2^k}{m^k} \sum_{i=0}^{(m-1)/2} i^k \\
&\le \frac{m}{2k+2} \text{ steps,}
\end{aligned}$$

asymptotically in m. This completes the proof of the theorem. □

For the k-point algorithm, Figure 4 plots the ETTR for 10 to 100 channels and two, four, eight and 16 radios. Here is why the k-point algorithm worst case performance is better than the $2k$-point algorithm worst case performance. In the $2k$-point case, given two pairs of coupled radios, one pair

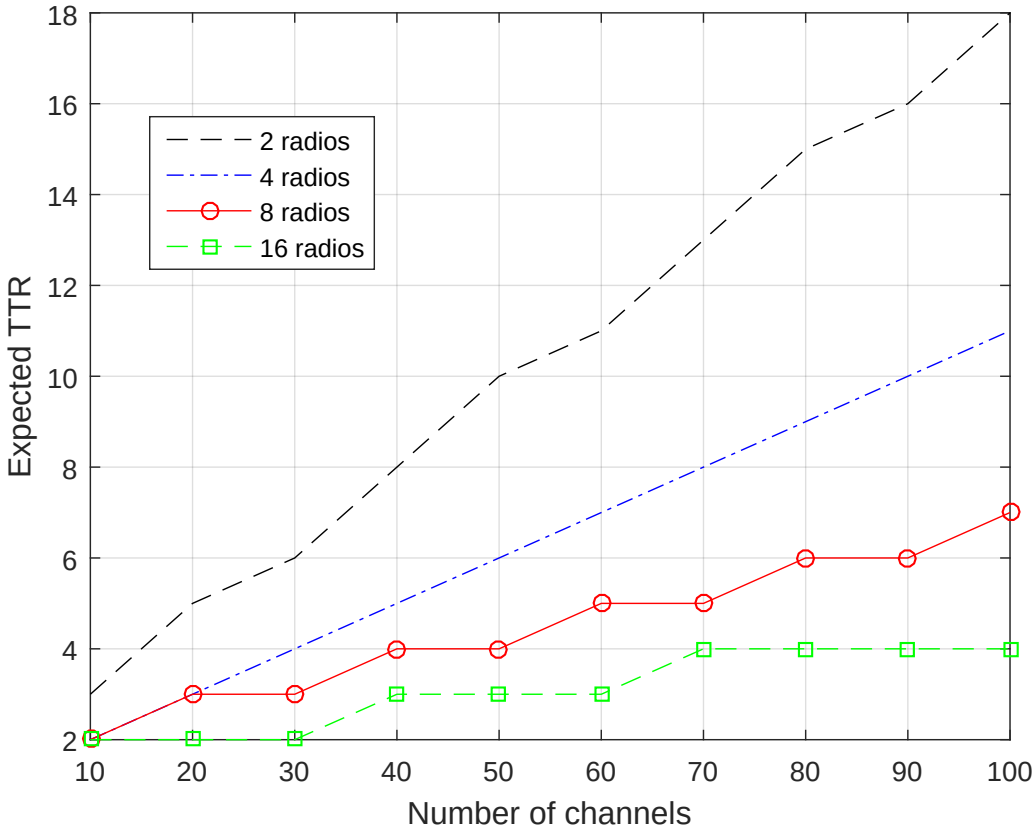


Figure 4: ETTR (in time slots) for the k-point algorithm.

from each user, the two distances can be both odd and require crossovers in both cases before rendezvous is achieved. In the k-point case, because m is assumed to be odd, given two pairs of coupled radios, if the distance is odd for one pair (crossover required), it is even for the other pair (no crossover required).

5. Asymmetric and Multiple-User Cases

In this section, we handle the asymmetric and multiple user cases. Under the asymmetric model, the secondary users have different sets of available channels. If the sets are overlapping, which we assume, they can eventually make rendezvous. The goal is to make the secondary users rendezvous on a common channel in a minimum amount of time.

Referring to the scenario depicted in Figures 2 and 3, let us assume that channel four is not available to user 1, all others are. There are no changes for user 2. In these conditions, rendezvous on channel four fails. The users repeatedly crossover and never make rendezvous. This cyclic behavior needs to be broken. In this example, the initial distance is even. An analogous behavior is obtained when the initial distance is odd.

When all channels are available, with the $2k$-point algorithm (Algorithm 1) we know that rendezvous does not require more than $m-1$ steps. Let $c_{i,2j-1}$ and $c_{i,2j}$ be the start channels of Radio 0 of user 1 and Radio 1 of user 2. After the completion of $m-1$ steps, let the new start channel of Radio 0 be $(c_{i,2j-1}+m-1) \mod m$. Let the new start channel of Radio 1 be $c_{i,2j}+1 \mod m$. For the Figures 2 and 3 scenario, it means that after *visiting* channel four, Radio 1 jumps to channel two. After *visiting* channel four, Radio 0 does channel four again. After one step, they crossover again. After three more steps, they finally meet on channel three. If the channel happens to be also not available to both, then this procedure is repeatedly applied until rendezvous is made.

Each sequence of *start channel selection* followed by $m-1$ steps is called a round. This procedure is called the *$2k$-point-asymmetric algorithm* (Algorithm 3). Each user i in $1,2$, for each radio pair $R_{i,2j-1}, R_{i,2j}$, with j in $1,2,\ldots,k$, randomly and independently selects two start channels in the set $\{0,1,2,\ldots,m-1\}$, denoted as $c_{i,2j-1}$ and $c_{i,2j}$. The radio $R_{i,2j-1}$ scans CW channel $c_{i,2j-1}$, while the radio $R_{i,2j}$ scans CCW from channel $c_{i,2j}$. Scanning is performed until rendezvous is achieved by the two users or $m-1$ steps have been completed. After the completion of a cycle of $m-1$ steps, for each radio pair $R_{i,2j-1}, R_{i,2j}$ the new starting channels are $c_{i,2j-1} = (c_{i,2j-1}+m-1) \mod m$ and $c_{i,2j} = c_{i,2j}+1 \mod m$. This is repeated m times.

Theorem 4. *Let us assume that m is an odd integer and that w is the number of channels that two users have in common, with $1 \leq w \leq m$. The $2k$-point-asymmetric algorithm (Algorithm 3) accomplishes rendezvous in at most $(m-w+1)\cdot(m-1)$ steps and with a $\left\lceil \frac{m^2}{w(2k+1)} \right\rceil$ expected number of steps, asymptotically in m.*

Proof. Rendezvous is made in at most $(m-w+1)\cdot(m-1)$ steps, because there is a maximum of $m-w+1$ rounds, due to the assumption on the range of w, and each round takes at most $(m-1)$ steps.

We calculate the expected number of steps. With index r in $1,\ldots,m-w$, each round is a trial with probability of success, i.e., rendezvous, $\frac{w}{m-r+1}$ (the

Algorithm 3 $2k$-Point-Asymmetric Algorithm: User $i = 1, 2$

```
for j = 1, 2, ..., k
    select start channels c_{i,2j-1} and c_{i,2j} randomly and independently
for l = 1, 2, ..., m
    for m − 1 steps
        scan CW with radio R_{i,2j-1} starting from c_{i,2j-1} and
        scan CCW with radio R_{i,2j} starting from c_{i,2j}
        stop when meet on same channel with other user
    c_{i,2j-1} = (c_{i,2j-1} + m − 1)  mod m
    c_{i,2j} = c_{i,2j} + 1  mod m
```

two users tune in a channel available to both) and probability of failure $1 - \frac{w}{m-r+1}$ (the two users tune in a channel not available to both). Due to the assumption that w common channels exist (with $1 \leq w \leq m$), the probability of success in the $(m - w + 1)$th round is one. The algorithm stops there, at the latest.

Let R be the random variable that counts the number of rounds until success. At least one round is required and a maximum of $m - w + 1$ rounds are needed. Then, we see that

$$\Pr[R = r] = \begin{cases} \prod_{r>1, s=2}^{r} \left(1 - \frac{w}{m-s+2}\right) \cdot \frac{w}{m-r+1} & r = 1, \ldots, m-w \\ 1 - \sum_{s=1}^{m-w} \Pr[R = s] & r = m-w+1. \end{cases}$$

The average of R is then

$$\begin{aligned} E[R] &= \sum_{r=1}^{m-w+1} r \Pr[R = r] \\ &= \sum_{r=1}^{m-w} r \Pr[R = r] + (m - w + 1) \cdot \left(1 - \sum_{r=1}^{m-w} \Pr[R = r]\right). \end{aligned}$$

For $r = 1, \ldots, m - w$, note that (see proof in Appendix A)

$$\left(1 - \frac{w}{m}\right)^{r-1} \frac{w}{m} \leq \Pr[R = r]. \tag{13}$$

Hence,

$$\sum_{r=1}^{m-w} r \Pr[R=r] + (m-w+1) \cdot \left(1 - \sum_{r=1}^{m-w} \Pr[R=r]\right)$$

$$\begin{aligned}
&\leq \sum_{r=1}^{m-w} r \left(\Pr[R=r] - \left[\Pr[R=r] - \left(1-\frac{w}{m}\right)^{r-1} \frac{w}{m}\right]\right) \\
&+ (m-w+1) \left(1 - \sum_{r=1}^{m-w} \left(\Pr[R=r] - \left[\Pr[R=r] - \left(1-\frac{w}{m}\right)^{r-1} \frac{w}{m}\right]\right)\right) \\
&= \sum_{r=1}^{m-w} r \left(1-\frac{w}{m}\right)^{r-1} \frac{w}{m} + (m-w+1) \left(1 - \sum_{r=1}^{m-w} r \left(1-\frac{w}{m}\right)^{r-1} \frac{w}{m}\right) \\
&= \sum_{r=1}^{m-w} r \left(1-\frac{w}{m}\right)^{r-1} \frac{w}{m} + (m-w+1) \sum_{r=m-w+1}^{\infty} \left(1-\frac{w}{m}\right)^{r-1} \frac{w}{m} \\
&\leq \sum_{r=1}^{m-w} r \left(1-\frac{w}{m}\right)^{r-1} \frac{w}{m} + \sum_{r=m-w+1}^{\infty} r \left(1-\frac{w}{m}\right)^{r-1} \frac{w}{m} \\
&= \sum_{r=1}^{\infty} r \left(1-\frac{w}{m}\right)^{r-1} \frac{w}{m} = \frac{m}{w}
\end{aligned}$$

The proof uses the fact that $\left(1-\frac{w}{m}\right)^{r-1} \frac{w}{m}$ defines the probability mass function and

$$\sum_{r=1}^{\infty} r \left(1-\frac{w}{m}\right)^{r-1} \frac{m}{w}$$

defines the expected value of a geometric random variable with parameter w/m, support $\{1, 2, 3, \ldots\}$ and mean m/w.

□

Similarly to Algorithm 3, it is possible to derive an algorithm for the asymmetric case from the k-point algorithm (Algorithm 2). The following can be concluded.

Theorem 5. *Let us assume that m is an odd integer and w is the number of channels that two users have in common, $1 \leq w \leq m$. The k-point-asymmetric algorithm accomplishes rendezvous in at most $(m-w+1) \cdot (m-$*

1)/2 *steps and with a* $\left\lceil \frac{m^2}{w(2k+2)} \right\rceil$ *expected number of steps, asymptotically in* m.

The proof is similar to the one of Theorem 4. It is omitted.

For the multiple-user case (asymmetric or symmetric), pairwise rendezvous is repeatedly applied to make global rendezvous. We adapt an idea originally presented by Lin et al. [11, 12]. When two users $i = 1, 2$ succeed to rendezvous, they exchange their parameters $c_{i,2j-1}$ and $c_{i,2j}$, j in $1, 2, \ldots, k$. For j in $1, 2, \ldots, k$, the smallest channel number is adopted by both users. In the sequel, the two users scan the same way. In the multiple-user case, this is done repeatedly, pairwise, until all secondary users scan the same way.

Theorem 6. *Let n be the number of users, an integer greater than one. Using the $2k$-point-asymmetric algorithm (Algorithm 3), all users rendezvous in at most $\lceil \log_2 n \rceil \cdot (m - w + 1) \cdot (m - 1)$ steps and with a $\left\lceil \log_2 n \cdot \frac{m^2}{w(2k+1)} \right\rceil$ expected number of steps, asymptotically in m. Using the k-point-asymmetric algorithm, all users make rendezvous in at most $\lceil \log_2 n \rceil \cdot (m-w+1) \cdot (m-1)/2$ steps and with an $\left\lceil \log_2 n \cdot \frac{m^2}{w(2k+2)} \right\rceil$ expected number of steps, asymptotically in m.*

Proof. The multiplier $\log_2 n$ comes from the fact that in each round the number of entities to rendezvous is divided by two. □

6. Comparison of Algorithms

We assume that each user has two or more radios that can be used simultaneously to achieve rendezvous. From a quantitative point of view, fair performance comparisons are possible with algorithms also making this assumption. To the best of our knowledge not much work has been published in that direction. Yu et al. [28] have conducted research in that direction, the *role-based parallel sequence* (RPS) algorithm. At the end, we have four comparable algorithms: RPS, randomized, $2k$-point and k-point. The ETTR is an evaluation metric applicable to all. Let p be the smallest primer number greater than or equal to m. Let k be the number of radios. Assuming, a homogeneous cognitive radio network, i.e., all nodes have the same number of radios, for the RPS algorithm the ETTR (also the maximum TTR) is

$$\left\lceil \frac{p}{k-1} \right\rceil \text{ time slots.} \tag{14}$$

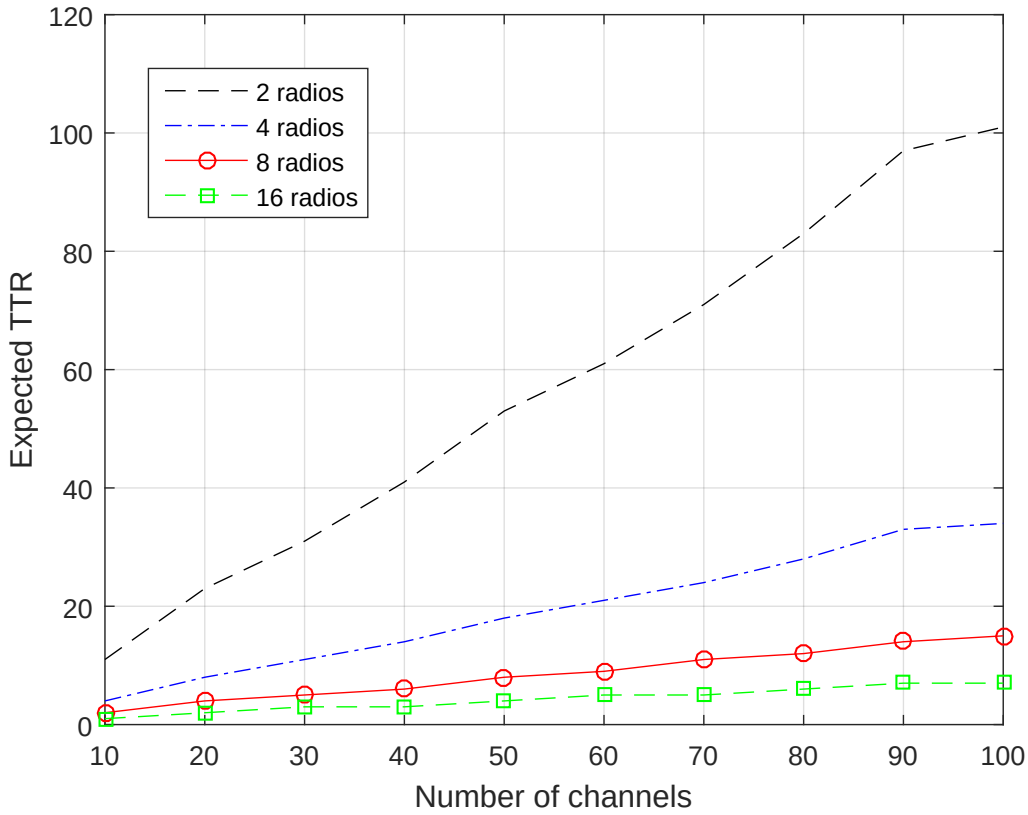


Figure 5: ETTR (in time slots) for the RPS algorithm.

For the RPS algorithm, Figure 5 plots the ETTR for 10 to 100 channels and two, four, eight and 16 radios. The ETTR for the randomized and k-point algorithms are shown in Figures 1 and 4. Note that k-point is slightly better than $2k$-point (see Theorems 2 and 3). They are both better than RPS. The randomized algorithm does better than all, when the number of radios is greater than four.

7. Simulations

Simulations have been conducted in the OMNeT++ environment [30]. The boxplots of Figures 6 and 7 show the performance of the simulated k-point algorithm. Along the x-axis, the number of channels (m) varies from 41 to 201. The y-axis corresponds to the TTR. The analytic ETTRs for the RPS [28], k-point and randomized algorithms are also plotted. The goal is

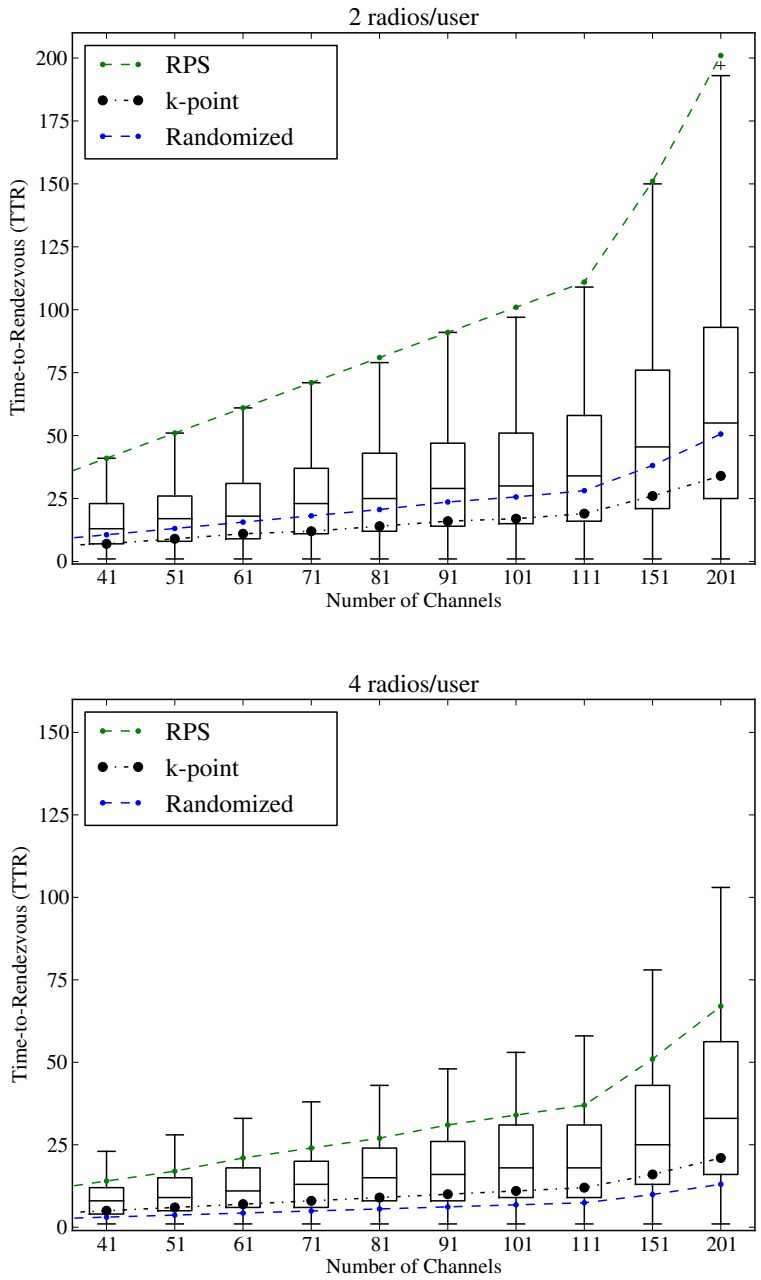
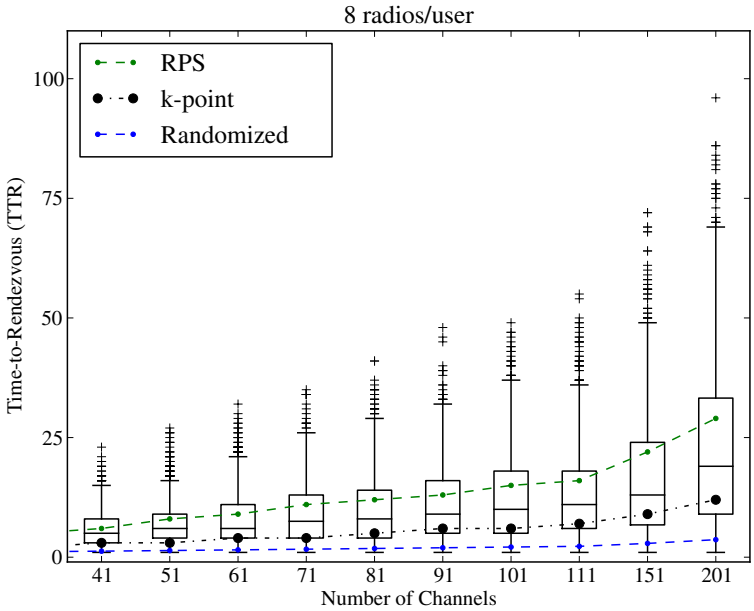


Figure 6: Boxplots of performance results obtained with the OMNeT++ simulations of the k-point algorithm for two and four radios. Curves depict the analytic ETTR for the RPS [28], k-point and randomized algorithms.



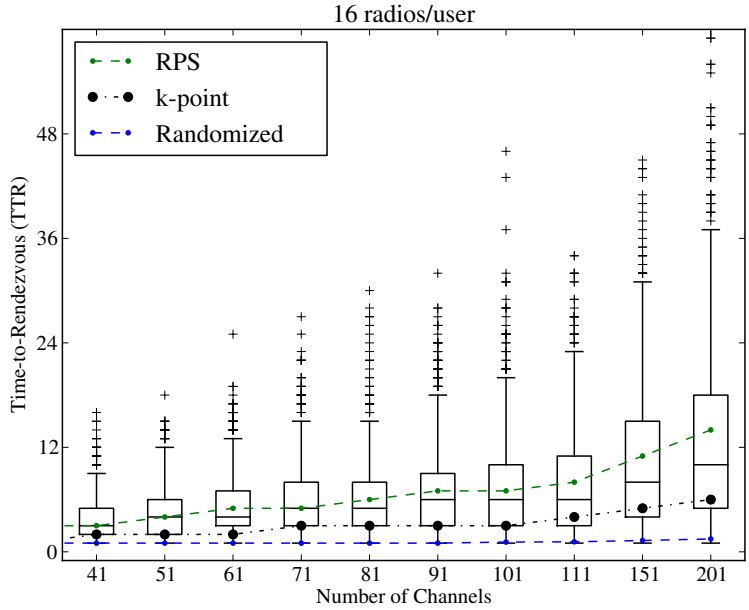


Figure 7: Boxplots of performance results obtained with the OMNeT++ simulations of the k-point algorithm for eight and 16 radios. Curves depict the analytic ETTR for the RPS [28], k-point and randomized algorithms.

to compare the simulation results with the ETTR of the models discussed in Sections 2, 3 and 4.

Each boxplot describes the statistical dispersion of the data. For each value of number of channels, the ranked data is divided into four equal groups. Each group, comprises a quarter of the data. They are delimited by three values called *quartiles*. The box bottom indicates the first quartile. The boxed horizontal bar corresponds to the second quartile, i.e., the median. The box top indicates the third quartile. The lowest bar corresponds to the lowest datum still within 1.5 of the interquartile range (i.e., difference between the second and first quartiles) down of the first quartile. The highest bar corresponds to the highest datum still within 1.5 the interquartile range (i.e., difference between the third and second quartiles) up of the third quartile. Crosses correspond to extremities, i.e., outliars. The ETTR of the k-point model discussed in Section 4 predicts results slightly better than the simulation results.

The boxplots of Figures 8, 9 and 10 show the results of the simulated $2k$-point asymmetric algorithm. The upper-side boxplots are for a two user-eight radios each (k is 8) case. The number of channel varies from 11 to 101. The number of common channels (w) is one in Figure 8, three in Figure 9 and 10 in Figure 10. The common channels are selected randomly. The non-common channels are also selected randomly, without repetition. The bottom-side boxplots are for a four user-eight radios each (k is 8) case. The number of channel also varies from 11 to 101. The number of common channels (w) is also one, three or 10. The band inside each boxplot corresponds to the median. The ETTR, according to Theorem 6, is shown. The simulation results match well the theoretical performance, in particular for the three and ten common channel cases.

Figure 11 plots the simulation average TTR for the k-point symmetric algorithm with multiple users, from two to 64. Each user has 16 radios. The number of channels varies from 11 to 101. The number of common channels (w) is 10. For multiple user and multiple radio cases, the simulation results are better than what is predicted by the theory. Indeed, there are opportunities for rendezvous across pairs that are not captured by Theorem 6. For the multiple user case, we hypothesize function

$$\left\lceil \log_2 n \cdot \frac{m^2}{w(2k \cdot \log_2 \log_2(n) + 1)} \right\rceil . \tag{15}$$

With respect to the equation of Theorem 6, note the presence of term

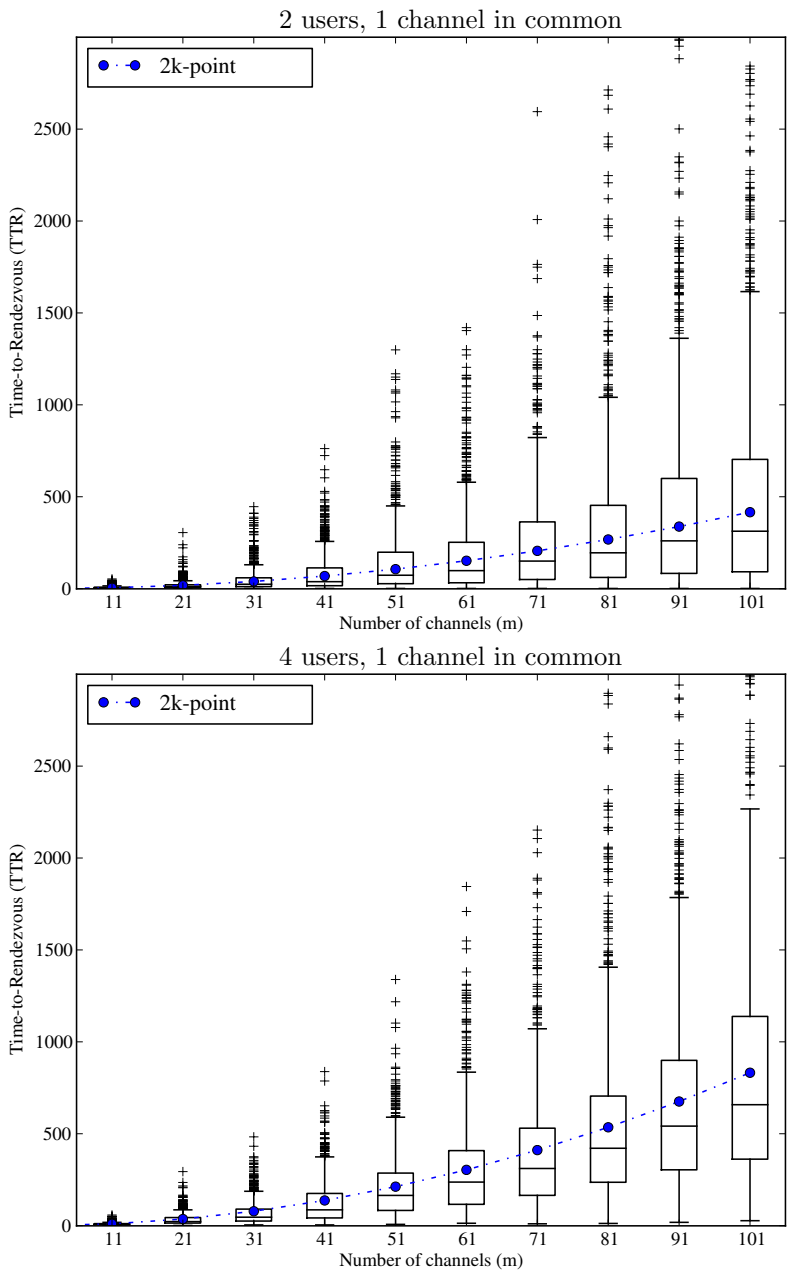


Figure 8: Boxplots depicting the performance results obtained with the OMNeT++ simulations of the $2k$-point-asymmetric algorithm. Upper-side boxplots are for the two user-one channel in common case (k is 8). Bottom-side boxplots are for the four user-one channel in common case (k is 8). Curves depict the analytic ETTR for the $2k$-point-asymmetric algorithm.

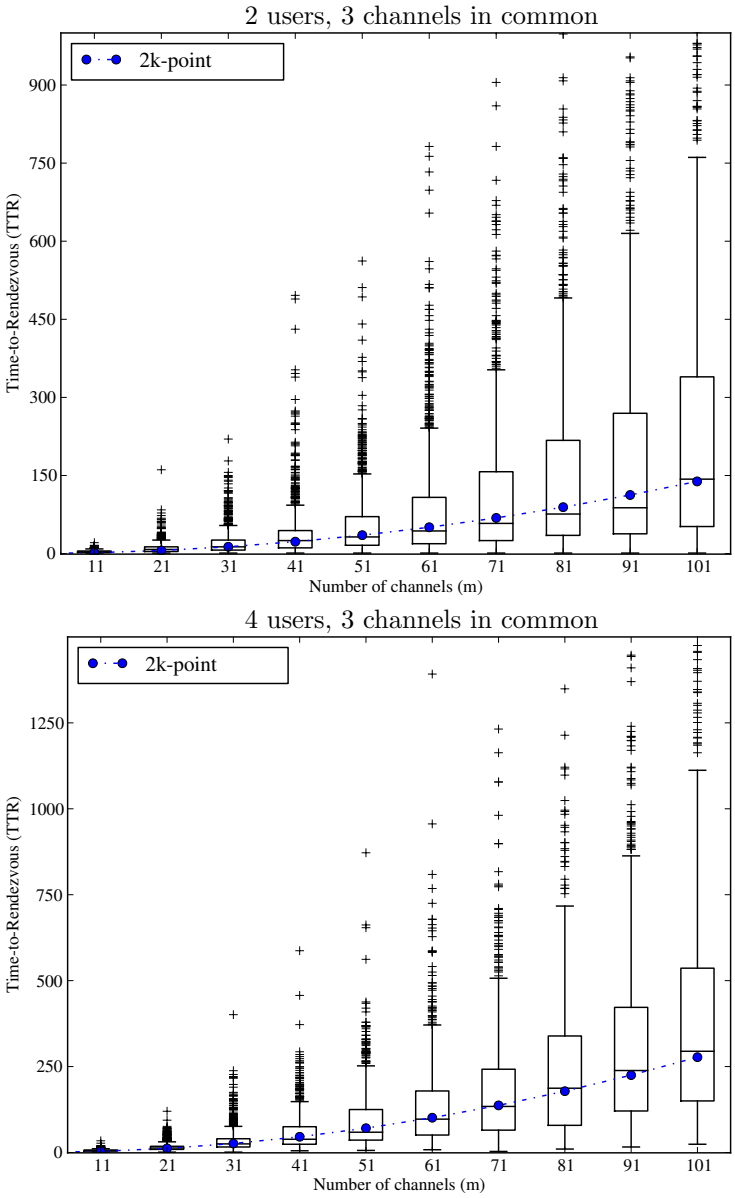


Figure 9: Boxplots depicting the performance results obtained with the OMNeT++ simulations of the $2k$-point-asymmetric algorithm. Upper-side boxplots are for the two user-three channels in common case (k is 8). Bottom-side boxplots are for the four user-three channels in common case (k is 8). Curves depict the analytic ETTR for the $2k$-point-asymmetric algorithm.

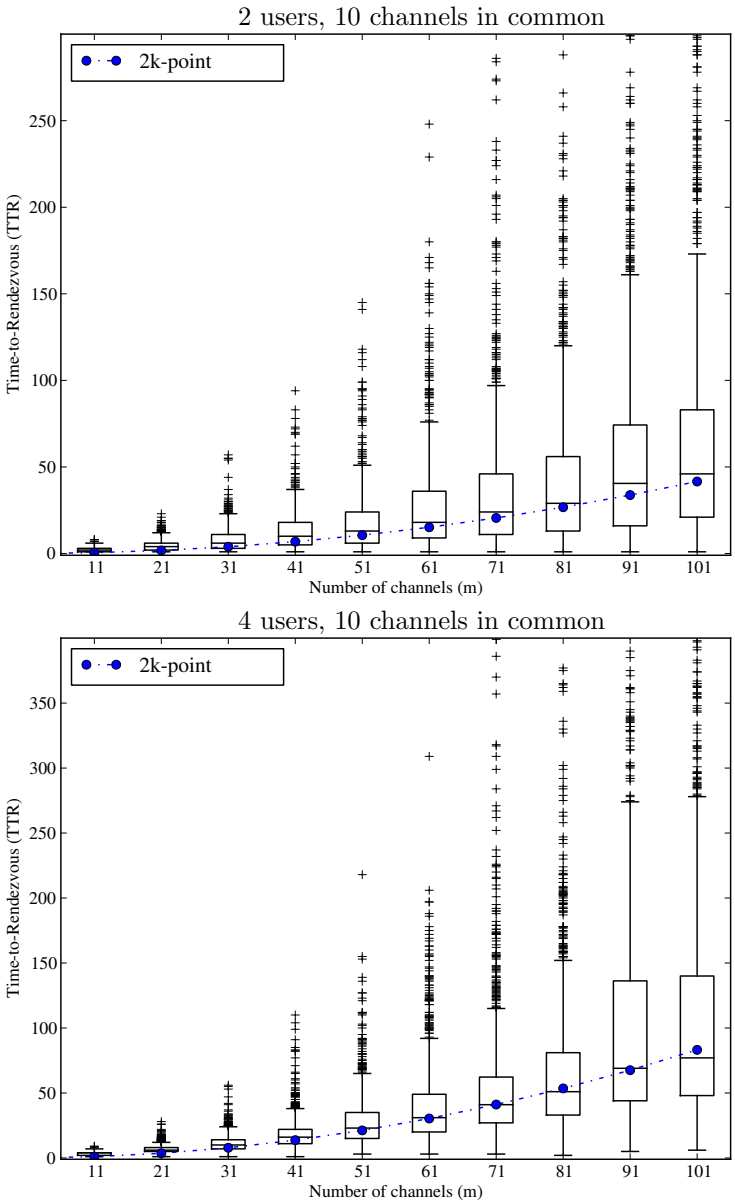


Figure 10: Boxplots depicting the performance results obtained with the OMNeT++ simulations of the $2k$-point-asymmetric algorithm. Upper-side boxplots are for the two user-ten channels in common case (k is 8). Bottom-side boxplots are for the four user-ten channels in common case (k is 8). Curves depict the analytic ETTR for the $2k$-point-asymmetric algorithm.

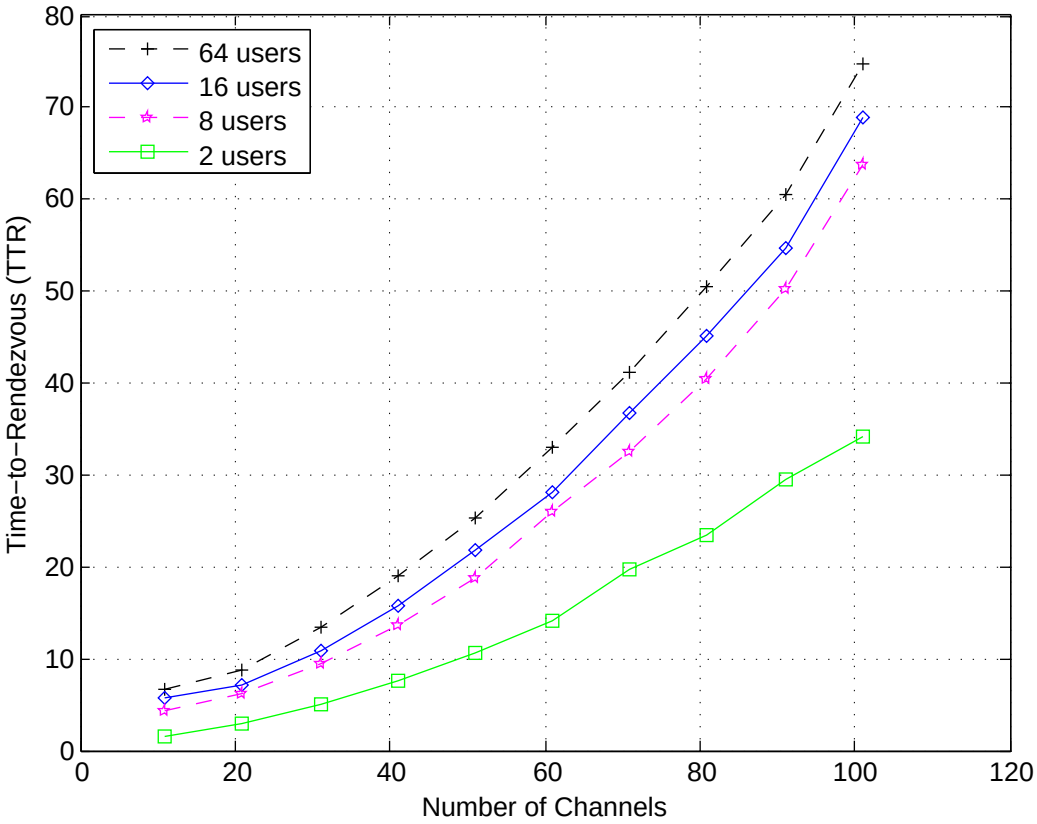


Figure 11: Simulation average TTR for the k-point-asymmetric algorithm. There are multiple users, from two to 64. Each user has 16 radios. The number of channels varies from 11 to 101. The number of common channels (w) is 10.

$\log_2 \log_2(n)$ in the denominator. Figure 12 shows, for the 64 user case, the simulation average TTR, ETTR according to Theorem 6 and ETTR according to Equation (15). The number of common channels (w) is 10. We ran the Kolmogorov-Smirnov goodness-of-fit statistical test [31] on the sample data and Equation (15). The statistical test yields to acceptance of the null hypothesis that the distribution follows Equation (15), at the 5% level of significance.

8. Conclusion

We have addressed the problem of secondary users with multiple radios making rendezvous on any of the available channels. We have introduced

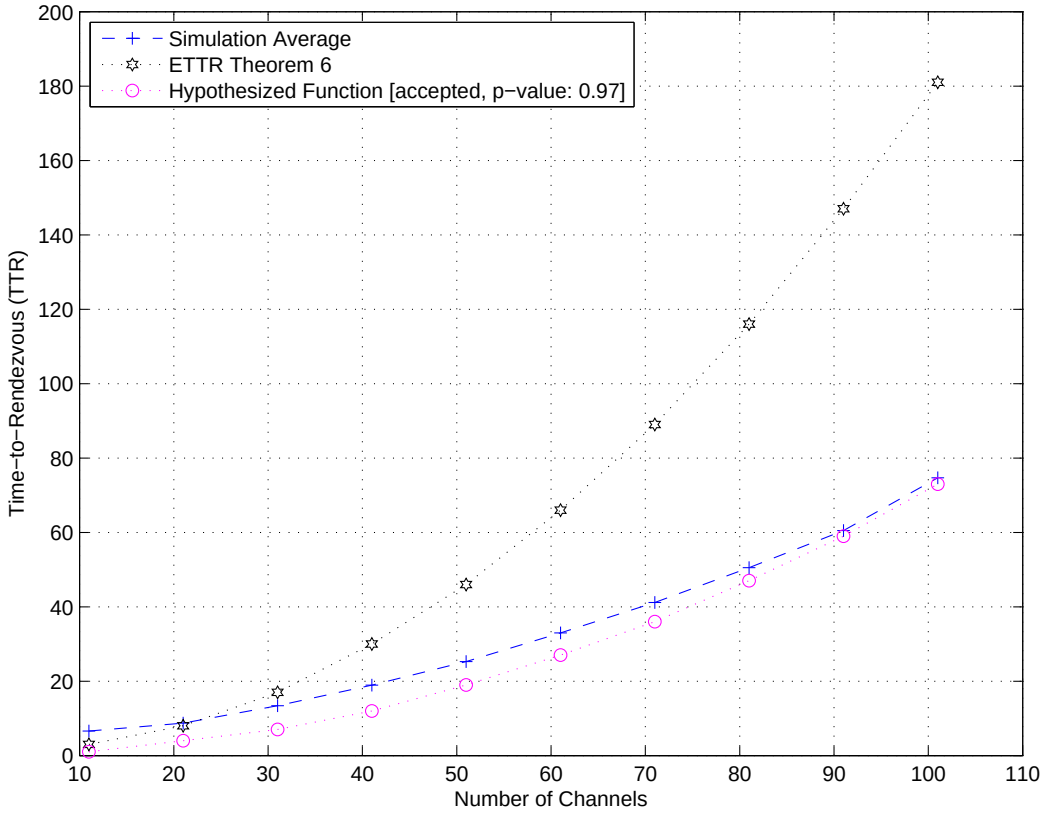


Figure 12: TTR for the k-point-asymmetric algorithm. There are 64 users. Each user has 16 radios. The number of channels varies from 11 to 101. The number of common channels (w) is 10.

two bidirectional algorithms, the $2k$-point and k-point procedures. We have addressed the two-user case, with a common channel set (the symmetric condition) or with different but overlapping channel sets (the asymmetric condition). The results have been generalized to an arbitrary number of secondary users. Equations have been developed to model the worst case and expected performance. A performance model has also been developed empirically using simulation data. Homogeneous secondary users have been assumed, i.e., they all have the same number of radios. This assumption is present for the purposes of the analysis. The algorithms work also in the heterogeneous case, where secondary users might have different numbers of radios.

Acknowledgment

We acknowledge financial support from Natural Sciences and Engineering Research Council of Canada and Spanish Ministry of Science (project TIN2011-27076-C03-02 CO-PRIVACY).

References

[1] M. Barbeau, G. Cervera, J. Garcia-Alfaro, E. Kranakis, The bidirectional algorithm for channel selection using a two-radio model, in: IEEE 80th Vehicular Technology Conference (VTC2014-Fall), 2014.

[2] M. McHenry, Nsf spectrum occupancy measurements project summary, in: First International Workshop on Technology and Policy for Accessing Spectrum (TAPAS), (co-located with WICON), Boston, MA., 2006.

[3] K. Ishizu, H. Murakami, H. Harada, TV white space database for coexistence of primary-secondary and secondary-secondary systems in mesh networking, in: 15th International Symposium on Wireless Personal Multimedia Communications (WPMC), 2012, pp. 118–122.

[4] C. Sun, G. Villardi, L. Zhou, Y. Alemseged, H. Tran, H. Harada, Optimizing the coexistence performance of secondary-user networks under primary-user constraints for dynamic spectrum access, IEEE Transactions on Vehicular Technology 61 (8) (2012) 3665–3676.

[5] P. Kolodzy, Cognitive radio fundamentals, in: SDR Forum, Singapore, 2005.

[6] C. Cormio, K. Chowdhury, An adaptive multiple rendezvous control channel for cognitive radio wireless ad hoc networks, in: 8th IEEE International Conference on Pervasive Computing and Communications Workshops (PERCOM Workshops), 2010, pp. 346–351.

[7] A. Mesodiakaki, F. Adelantado, A. Antonopoulos, L. Alonso, C. Verikoukis, Fairness evaluation of a secondary network coexistence scheme, in: IEEE 18th International Workshop on Computer Aided Modeling and Design of Communication Links and Networks (CAMAD), 2013, pp. 180–184.

[8] A. Mesodiakaki, F. Adelantado, L. Alonso, C. Verikoukis, Performance analysis of a cognitive radio contention-aware channel selection algorithm, IEEE Transactions on Vehicular Technology 64 (5) (2015) 1958–1972.

[9] N. Theis, R. Thomas, L. DaSilva, Rendezvous for cognitive radios, Mobile Computing, IEEE Transactions on 10 (2) (2011) 216–227.

[10] L. DaSilva, I. Guerreiro, Sequence-based rendezvous for dynamic spectrum access, in: 3rd IEEE Symposium on New Frontiers in Dynamic Spectrum Access Networks (DySPAN), 2008, pp. 1–7.

[11] Z. Lin, H. Liu, X. Chu, Y.-W. Leung, Ring-walk rendezvous algorithms for cognitive radio networks, Ad-Hoc and Sensor Wireless Networks 16 (4) (2012) 243–271.

[12] H. Liu, Z. Lin, X. Chu, 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), 2010, pp. 755–760.

[13] P. Bahl, R. Chandra, 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, ACM, New York, NY, USA, 2004, pp. 216–230. doi:10.1145/1023720.1023742.
URL `http://doi.acm.org/10.1145/1023720.1023742`

[14] S. Krishnamurthy, M. Thoppian, S. Kuppa, R. Chandrasekaran, N. Mittal, S. Venkatesan, R. Prakash, Time-efficient distributed layer-2 auto-configuration for cognitive radio networks, Computer Networks 52 (4) (2008) 831–849.

[15] K. Bian, J.-M. Park, 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, ACM, New York, NY, USA, 2009, pp. 25–36. doi:10.1145/1614320.1614324.
URL `http://doi.acm.org/10.1145/1614320.1614324`

[16] K. Bian, J.-M. Park, Asynchronous channel hopping for establishing rendezvous in cognitive radio networks, in: Proceedings of IEEE INFOCOM, 2011, pp. 236–240.

[17] K. Bian, J.-M. Park, Maximizing rendezvous diversity in rendezvous protocols for decentralized cognitive radio networks, IEEE Transactions on Mobile Computing 12 (7) (2013) 1294–1307.

[18] D. Yang, J. Shin, C. Kim, Deterministic rendezvous scheme in multi-channel access networks, Electronics Letters 46 (20) (2010) 1402–1404.

[19] Z. Lin, H. Liu, X. Chu, Y.-W. Leung, Jump-stay based channel-hopping algorithm with guaranteed rendezvous for cognitive radio networks, in: Proceedings of IEEE INFOCOM, 2011, pp. 2444–2452.

[20] H. Liu, Z. Lin, X. Chu, Y.-W. Leung, Jump-stay rendezvous algorithm for cognitive radio networks, IEEE Transactions on Parallel and Distributed Systems 23 (10) (2012) 1867–1881.

[21] Z. Lin, H. Liu, X. Chu, Y. Leung, Enhanced jump-stay rendezvous algorithm for cognitive radio networks, IEEE Communications Letters (2013) 1–4.

[22] M. Barbeau, G. Cervera, J. Garcia-Alfaro, 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.

[23] A. Robertson, L. Tran, J. Molnar, E.-H. Fu, Experimental comparison of blind rendezvous algorithms for tactical networks, in: IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM), 2012, pp. 1–6.

[24] G.-Y. Chang, J.-F. Huang, A fast rendezvous channel-hopping algorithm for cognitive radio networks, IEEE Communications Letters 17 (7) (2013) 1475–1478.

[25] V. Reguera, E. O. Guerra, R. D. Souza, E. Fernandez, G. Brante, Short channel hopping sequence approach to rendezvous for cognitive networks, IEEE Communications Letters 18 (2) (2014) 289–292.

[26] Z. Gu, Q.-S. Hua, Y. Wang, F. Lau, Nearly optimal asynchronous blind rendezvous algorithm for cognitive radio networks, in: 10th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks (SECON), 2013, pp. 371–379.

[27] G.-Y. Chang, W.-H. Teng, H.-Y. Chen, J.-P. Sheu, Novel channel-hopping schemes for cognitive radio networks, IEEE Transactions on Mobile Computing 13 (2) (2014) 407–421.

[28] L. Yu, H. Liu, Y.-W. Leung, X. Chu, Z. Lin, Multiple radios for effective rendezvous in cognitive radio networks, in: IEEE International Conference on Communications (ICC), 2013, pp. 2857–2862.

[29] L. Yu, H. Liu, Y.-W. Leung, X. Chu, Z. Lin, Multiple radios for effective rendezvous in cognitive radio networks: Proof of theoretical results, Tech. rep., Department of Computer Science, Hong Kong Baptist University (September 2012).

[30] A. Varga, 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.

[31] A. Kolmogorov, Sulla determinazione empirica di una legge di distribuzione, G. Ist. Ital. Attuari 4 (1933) 83–91.

Appendix A.

Lemma 2. *For $m, r, w \in \{1, 2, 3, \dots\}$, with $w \leq m$ and $r \leq m - w + 1$,*

$$\left(1 - \frac{w}{m}\right)^{r-1} \frac{w}{m} \leq \prod_{r>1, s=2}^{r} \left(1 - \frac{w}{m - s + 2}\right) \cdot \frac{w}{m - r + 1}.$$

Proof. The proof is by induction on r.

Basis ($r = 1$).

$$\left(1-\frac{w}{m}\right)^{r-1}\frac{w}{m} \leq \prod_{r>1,s=2}^{r}\left(1-\frac{w}{m-s+2}\right)\cdot\frac{w}{m-r+1} \iff$$
$$\left(1-\frac{w}{m}\right)^{1-1}\frac{w}{m} \leq \frac{w}{m-1+1} \iff$$
$$\frac{w}{m} \leq \frac{w}{m}$$

Induction ($r+1$, with $1 \leq r \leq m-w$). The proof uses the fact that

$$1-\frac{w}{m} = \frac{m-w}{m} \leq \frac{m-w-(r-1)}{m-r} = \left(1-\frac{w}{m-r+1}\right)\cdot\frac{m-r+1}{m-r}.$$

We have,

$$\left(1-\frac{w}{m}\right)^{r-1}\frac{w}{m} \leq \prod_{r>1,s=2}^{r}\left(1-\frac{w}{m-s+2}\right)\cdot\frac{w}{m-r+1} \Rightarrow$$
$$\left(1-\frac{w}{m}\right)^{r-1}\left(1-\frac{w}{m}\right)\frac{w}{m} \leq$$
$$\prod_{r>1,s=2}^{r}\left(1-\frac{w}{m-s+2}\right)\cdot\left(1-\frac{w}{m-r+1}\right)\cdot\frac{m-r+1}{m-r}\cdot\frac{w}{m-r+1} \Rightarrow$$
$$\left(1-\frac{w}{m}\right)^{r}\frac{w}{m} \leq \prod_{r>1,s=2}^{r+1}\left(1-\frac{w}{m-s+2}\right)\cdot\frac{w}{m-r}.$$

□