The Bidirectional Algorithm for Channel Selection Using a Two-radio Model

Michel Barbeau*, Gimer Cervera‡, Joaquin Garcia-Alfaro† and Evangelos Kranakis*

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

Abstract—We study the problem of establishing rendezvous between two secondary users. We assume that each user has two radios that can be used concurrently. We present the bidirectional algorithm that exploits the two radios. Assuming the availability of \( m \) channels, rendezvous between two start-asynchronous users is guaranteed within a delay of \( m \) time slots. The expected time-to-rendezvous is \( m/3 \) time slots. Assuming users are start-synchronous, rendezvous is made in at most \( (m + 1)/2 \) time slots. The expected time-to-rendezvous is \( \frac{m}{2} + 1 - \frac{1}{4m} \) time slots.

Index Terms—Channel selection, cognitive radio network, cognitive wireless network, dynamic spectrum access, rendezvous.

I. INTRODUCTION

Over idle channels of a radio spectrum segment, we assume that secondary users can communicate as long as they do not create interference to primary users. The problem is for two secondary users to make rendezvous on one of the available channels. Each secondary user hops over the set of channels attempting to make rendezvous with the other secondary user. Time is divided into equal length intervals called time slots. The two users make rendezvous when they are on the same channel during a time slot. We assume that each user has two radios that are concurrently used to achieve rendezvous with the other user.

The bidirectional algorithm is introduced. Let \( m \) be the number of available channels. Firstly, we consider the case where users do not start at the same time, i.e., they are start-asynchronous. We show that rendezvous is guaranteed within a delay of \( m \) time slots. The expected time-to-rendezvous (TTR) is \( m/3 \) time slots. Secondly, we consider the case where users start at the same time, i.e., they are start-synchronous. Rendezvous is made in at most \( (m + 1)/2 \) time slots. The expected TTR is \( \frac{m}{2} + 1 - \frac{1}{4m} \) time slots.

In Section II, we review related work. We make a comparison with a purely random algorithm, studied in Section III. The bidirectional algorithm is described in Section IV. Simulation results are presented in Section V. We conclude with Section VI.

II. RELATED WORK

The performance of 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. [1], [2]. The random channel algorithm, visiting all channels in a random order, does guarantee rendezvous. In the asynchronous user ring-walk algorithm, of Lin et al. [3], [4], preference is given to channels with low interference to primary users. Rendezvous is not guaranteed. Bahl et al. proposed an approach for WiFi networks [5]. Rendezvous is guaranteed to take place under the symmetric model. Krishnamurthy et al. developed a two-phase algorithm [6]. Following a first phase for neighbor discovery, conducted on common local channels, a global common channel is determined among the participants in the second phase. Bian et al. use a quorum principle on a two-channel case [7]–[9]. Rendezvous is guaranteed. Yang et al. introduced an algorithm based on the k-shift-invariant concept that guarantees rendezvous [10]. Lin et al. authored the (enhanced) jump-stay rendezvous algorithm [11]–[13]. It is designed for multiple users with guaranteed rendezvous. The modular clock algorithm has been originally presented by Theis et al. [1]. It is based on ideas of DaSilva and Guerreiro [2]. 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. Practical evaluations of the modular clock and random algorithms have been conducted by Robertson et al. using the GNU radio framework [14]. More recent related contributions are described in the papers of Chang and Huang [15], Reguera et al. [16], Gu et al. [17] and Chang et al. [18]. All the aforementioned works assume a single radio per user. Recently, Yu et al. [19] have conducted research in that direction. They proposed the role-based parallel sequence (RPS) algorithm where users are equipped with multiple radios.
III. Random Algorithm

Let \( m \) denote the number of channels (a positive integer). Channel numbers range from zero to \( m-1 \). We assume that there are two users: 1 and 2. Each user is equipped with two radios, say radios 0 and 1. For the sake of comparison, we introduce an algorithm that makes each user visit channels randomly, that is, Algorithm 1.

Algorithm 1 Random Algorithm

```
while True do
    Randomly select a start channel \( c_0 \) in 0, 1, 2, \ldots, \( m-1 \)
    Randomly select a start channel \( c_1 \) in 0, 1, 2, \ldots, \( m-1 \)
    if rendezvous on channel \( c_0 \) or \( c_1 \) then exit
end if
end while
```

**Lemma 1:** The expected TTR of the random algorithm is

\[
m^3 \cdot \frac{m^3 - (m - 1) \cdot [m - 1 + (m - 2)^2]}{m^3 - (m - 1) \cdot [m - 1 + (m - 2)^2]} \text{ time slots.} \tag{1}
\]

**Proof.** Each time slot can be seen as a Bernoulli trial with probability of success, i.e., rendezvous, \( p \) and probability of failure \( q = 1 - p \). The probability of a trial failure is

\[
q = \frac{m \cdot (m - 1)^2 + m \cdot (m - 1) \cdot (m - 2)^2}{m^4}. \tag{2}
\]

Indeed, there are two pairs of radios: user 1 radios 0 and 1 and user 2 radios 0 and 1. Each radio can be tuned to any of the \( m \) different channels. Hence, there is a total of \( m^4 \) different channel combinations (the denominator in Equation (2)). Among them, there are two cases where rendezvous fails. In case 1 (the left-term numerator in Equation (2)), both radios of user 1 are on the same channel, there are \( m \) such combinations, and radios 0 and 1 of user 2 are tuned to two other channels, there are \( (m - 1)^2 \) such combinations. In the second case (the right-term numerator in Equation (2)), radios 0 and 1 of user 1 are tuned to two different channels, there are \( m \cdot (m - 1) \) such combinations, and radios 0 and 1 of user 2 are tuned to other channels, there are \( (m - 2)^2 \) such combinations. Equation (2) can be rewritten as \( \frac{(m-1) \cdot [m - 1 + (m - 2)^2]}{m^4} \). Hence, \( p = 1 - q = 1 - \frac{(m-1) \cdot [m - 1 + (m - 2)^2]}{m^4} \). 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 mean

\[
1 - \frac{1}{p} = \frac{m^3}{m^3 - (m - 1) \cdot [m - 1 + (m - 2)^2]} \text{ time slots.}
\]

Note that the random algorithm does not guarantee rendezvous.

IV. The Bidirectional Algorithm

We arrange the channel numbers consecutively on a ring of size \( m \), as in Figure 1. In this example, there are five channels. Radio 0 of user 1 is tuned to channel zero, scanning channels clockwise (CW). Radio 1 of user 2 is tuned to channel three, scanning counterclockwise (CCW). The distance \( d \) is the number of hops separating the two radios on the channel ring, scanning toward each other. In this example, \( d \) is three hops, an odd number. With \( m \) channels, \( d \) is in \( 0, 1, \ldots, m-1 \). While scanning, each user traverses one hop per time slot. In Figure 2, after two time slots, the two radios mutually crossover. They are tuned to channels one and two. However, this time the distance is four hops, an even number. After scanning for two additional time slots, they make rendezvous on channel four.

Firstly, let us consider users that are start-asynchronous, i.e., they may start at different time slots. We assume, that they are time slot-synchronous. Every user executes Algorithm 2.

Algorithm 2 Bidirectional Algorithm

```
Randomly select a start channel \( c_0 \) in 0, 1, 2, \ldots, \( m-1 \)
Randomly select a start channel \( c_1 \) in 0, 1, 2, \ldots, \( m-1 \)
while True do
    if rendezvous on channel \( c_0 \) or \( c_1 \) then exit
    \( c_0 = (c_0 + 1) \mod m \)
    \( c_1 = (m + c_1 - 1) \mod m \)
end while
```

**Theorem 1:** In the start-asynchronous case, the bidirectional algorithm accomplishes rendezvous in at most \( m/3 \) time slots. The expected TTR is \( m/3 \) time slots, asymptotically in \( m \).

**Proof.** First, we look at the worst-case TTR. Because each user has two radios scanning in opposite directions, there are...
two pairs of radios, each consisting of one radio from user 1 and one radio from user 2, scanning in opposite directions. In the worst case, the distance between two radios from two different users scanning in opposite directions on the ring is \( m - 2 \) hops (an odd number). Crossover happens exactly after \( \frac{m-1}{2} \) time slots. After the crossover, the distance between the two users becomes equal to \( m - 1 \) hops (an even number). The two users make rendezvous in \( \frac{m-1}{2} \) additional time slots. Counting the initial time slot, the total number of time slots until rendezvous never exceeds \( m \).

We now look at the expected TTR. We have three cases to consider. In the first case, for exactly one pair of radios the distance is even. Let \( d \) hops be their distance. The users make rendezvous in \( 1 + d/2 \) time slots. In the second case, the distance is even for the two pairs. If \( d \) hops is the minimum of the two distances, then the users make rendezvous in \( 1 + d/2 \) time slots. In the third case, for both pairs of radios the distance is odd. Let \( d \) hops be the minimum distance. The users make rendezvous in \( 1 + \frac{d+1}{2} + \frac{m-1}{2} \) time slots. Observe that the first case occurs with probability \( \frac{1}{2} \), the second case with probability \( \frac{1}{4} \), and the last case with probability \( \frac{1}{4} \). We now analyze the expected number of time slots in each case.

Case 1. For exactly one pair of radios the distance is even. If \( d \) is even, then they rendezvous in \( 1 + \frac{d}{2} \) time slots. The expected TTR can be expressed as

\[
\frac{2}{m+1} \sum_{d=0}^{m-1} \left( 1 + \frac{d}{2} \right) = 1 + \frac{m(m-1)}{4(m+1)} \approx \frac{m}{4}, \tag{3}
\]

asymptotically in \( m \). The factor \( \frac{2}{m+1} \) is due to the fact that there are \((m+1)/2\) possible even distance values between two radios hopping in opposite directions.

Case 2. For both pairs of radios the distance is even. There are two random variables involved, namely \( N_0 \) and \( N_1 \), and \( N_i \) (for \( i = 1, 2 \)) is the number of hops between the two radios, of two different users, hopping in opposite directions. Let \( A_i \) denote the event that \( N_i \) is an even integer. Conditioning over the event \( A_i \) we observe that \( \Pr[N_i = 2j | A_i] = \frac{1}{m/2} \). Therefore, \( \Pr[N_i > d | A_i] \) is equal to

\[
\sum_{m-1 \geq 2j > d} \Pr[N_i = 2j | A_i] = \sum_{(m-1)/2 \geq j > d/2} \frac{1}{m/2}. \tag{4}
\]

It follows from the last sum (4) above that \( \Pr[N_i > d | A_i] = \frac{m-1-d}{m} \) if \( d \) is even, and \( \Pr[N_i > d | A_i] = \frac{m-d}{m} \) if \( d \) is odd. Clearly, the random variables \( N_0, N_1 \) are independent and identically distributed. Let \( A = A_0 \cap A_1 \). Conditioning over the event \( A \), and using the formulas for \( \Pr[N_i > d | A_i] \) derived above we conclude that

\[
E[\min\{N_0, N_1\} | A] = \sum_d \Pr[\min\{N_0, N_1\} > d | A] = \sum_d \Pr[N_0 > d | A_0] \Pr[N_1 > d | A_1] \]
\[
= \sum_d (\Pr[N_0 > d | A_0])^2 \approx \frac{1}{m^2} \sum_{d=0}^{m-1} d^2 \]
\[
= \frac{m(m+1)(2m+1)}{6m^2} \approx \frac{m}{3},
\]

asymptotically in \( m \). It follows that, asymptotically in \( m \), the expected minimum distance is \( \frac{m}{3} \) hops, while rendezvous occurs in expected

\[
\frac{1}{2} \cdot \frac{m}{3} = \frac{m}{6}
\]

time slots, asymptotically in \( m \).

Case 3. For both pairs of radios the distance is odd. If \( d \) is odd, then they crossover after \( 1 + \frac{d+1}{2} \) time slots. They rendezvous in \( \frac{m-1}{2} \) additional time slots. Reusing the argumentation of Case 2, asymptotically in \( m \), the expected minimum distance is \( \frac{m}{4} \) hops. The expected TTR can be expressed as \( 1 + \frac{1}{2} \cdot \frac{m}{4} + \frac{m-1}{2} = 1 + \frac{m}{6} + \frac{m-1}{2} \). Asymptotically in \( m \), the expected TTR is

\[
\frac{m}{3}.
\tag{6}
\]

Combining Equations (3), (5), and (6), it follows that expected TTR is \( \frac{1}{2} \cdot \frac{m}{4} + \frac{m}{6} + \frac{1}{2} \cdot \frac{m}{3} = \frac{m}{6} \), asymptotically in \( m \). ■

Algorithm 3 Bidirectional Algorithm - Start Synchronous

Randomly assign to \( c_0, c_1 \) a start channel in \( 0, 1, \ldots, m-1 \)
while True do
if rendezvous on channel \( c_0 \) or \( c_1 \) then exit
\( c_0 = (c_0 + 1) \mod m \)
\( c_1 = (m + c_1 - 1) \mod m \)
end while

Algorithm 3 is a slight modification of Algorithm 2. For each user, both radios are starting on the same randomly selected channel, still scanning in opposite directions. We assume that both users start at the same time, i.e., they are start-synchronous.

For \( i = 1, 2 \), let us say user ‘\( i \)’ radios are \( R^{(1)}_i \) and \( R^{(2)}_i \). Across users, radio \( R^{(1)}_1 \) is paired with radio \( R^{(2)}_2 \). Radio \( R^{(1)}_2 \) is paired with radio \( R^{(2)}_1 \). Radios in the pair \( R^{(1)}_1, R^{(2)}_2 \) (or \( R^{(2)}_1, R^{(1)}_2 \)) scan in opposite directions. At start, the distance between one pair is \( d \) hops while it is \( m - d \) hops for the other. If \( d \) is even, then one pair rendezvouses in \( \frac{d}{2} \) hops, while the other pair crossovers in \( \frac{m-d}{2} \) hops and rendezvouses in additional \( \frac{m-1}{2} \) hops. Giving a total of \( \frac{m-d+1}{2} + \frac{m-1}{2} = m - \frac{d}{2} \) hops. Similarly, if \( d \) is odd then \( m - d \) is even. Using the previous observation, one pair accomplishes rendezvouses in \( \frac{m-d}{2} \) hops while the other in \( \frac{d+1}{2} + \frac{m-1}{2} = \frac{m+d}{2} \) hops. In either case, the number of hops for rendezvous is at most
the minimum of the two values above, which implies that the worst-case number of time slots is at most \( (m+1)/2 \).

**Theorem 2.** In the start-synchronous case, the bidirectional algorithm accomplishes rendezvous in at most \( (m+1)/2 \) time slots. Moreover, the expected number of time slots until rendezvous is \( m \cdot \frac{1}{4} + 1 - \frac{1}{4m} \).

**Proof.** Before we proceed to the main proof, we note that the TTR can be measured either as a number of hops or as a number of time slots. Regardless of the way it is measured, the magnitude of the TTR is not affected since \( TS = H + 1 \). To simplify the proof, we tacitly make use of this simple observation in the sequel.

We now look at the expected TTR. The discussion outlined above shows that the following identity is valid

\[
H(d) = \begin{cases} 
\frac{d}{2} & \text{if } d \text{ is even} \\
\frac{(m-d)}{2} & \text{if } d \text{ is odd}
\end{cases}
\]

(7)

where \( H(d) \) is a random variable measuring the number of hops until rendezvous, when the initial distance is \( d \).

Equation (7) displays the value of \( H(d) \) as a function of the distance \( d \). Let the random variable \( D \) be the distance between \( c_0 \) and \( c_1 \). We can now calculate the expected number of hops:

\[
E[H] = \frac{(m-1)/2}{\sum_{d=0}^{m} H(d) \cdot \Pr[D = d]} = \sum_{d=0}^{m} H(d) \cdot \frac{2}{m}
\]

\[
= \frac{2}{m} \left( \frac{(m-1)/2}{\sum_{d=0}^{\text{even}} d} + \sum_{d=1}^{\text{odd}} \frac{m-d}{2} \right)
\]

\[
= \frac{1}{m} \left( \frac{(m-1)/2}{\sum_{d=0}^{\text{even}} d} + \sum_{d=1}^{\text{odd}} (m-d) \right)
\]

\[
= \frac{1}{m} \sum_{d=0}^{m-1} d = \frac{(m-1)(m+1)}{4m} = m \cdot \frac{1}{4} - \frac{1}{4m}.
\]

By the observation at the beginning of the proof, we have that the number of time slots \( TS = H + 1 \). Therefore, the expected number of time slots is at most

\[
\frac{m}{4} + 1 - \frac{1}{4m}.
\]

(8)

**V. Simulations**

Simulations have been conducted in the OMNeT++ environment [20]. The boxplot of Figure 3 shows the performance of the simulated random algorithm. On the \( x \)-axis, the number of channels \( m \) varies from 11 to 101. The \( y \)-axis corresponds to the TTR. The mean TTR obtained through simulation is plotted. The expected TTR, according to the model of Equation 1, is also plotted. The simulation results are consistent with the analytic model. The boxplot describes the statistical dispersion of the data. For each value of \( m \), 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 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., outliers.

The boxplot of Figure 4 shows the performance of the simulated bidirectional algorithm, \textit{start-asynchronous} case.

The expected TTR, according to the model of Theorem 2, are plotted. Again, the simulation results are consistent with the analytic model. The random algorithm slightly outperforms the bidirectional algorithm, on average. The bidirectional algorithm, however, guarantees rendezvous in finite time. Moreover, the boxplots
show that the bidirectional algorithm is subject to substantially less statistical dispersion than the random algorithm. In terms of cost, the bidirectional algorithm does not need to repeatedly generate random channel numbers.

The boxplot of Figure 5 shows the performance of the simulated bidirectional algorithm, in the start-synchronous case. The mean TTR, obtained through simulation, and expected TTR, according to Equation 8, are plotted. Simulation results and analytic model are consistent.

VI. CONCLUSION

We have addressed the problem of two users with two radios making rendezvous on any of the $m$ available channels. We have introduced the bidirectional algorithm. Rendezvous is guaranteed within $m$ time slots, in the start-asynchronous case, and $(m + 1)/2$ time slots, in the start-synchronous case. The expected TTR is $m/3$ time slots, in the start-asynchronous case, and $m/2 + 1 - 1/4m$ time slots, in the start-synchronous case. The performance has been confirmed through simulation. The work has been extended to an arbitrary number of radios in a companion paper [21].

ACKNOWLEDGMENT

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 COPRIVACY) and Innovation and Ministry of Education of Mexico (PROMEP).

REFERENCES


