Chapter 3
Rendezvous Theory

Abstract In distributed computing, the entities of the system are assumed to be able to communicate with each other and predominantly they adopt the message passing model (point-to-point model, broadcast model or others) for communication. However, many works abstracted away the detailed communication scheme and assume that two entities can transmit messages to each other if they are connected in the network. Actually, each entity has many external ports for communication as described in Chap. 2, and two neighbors can only communicate with each other if indeed a communication link can be constructed between the ports they choose. Rendezvous is referred to as the process of two entities choosing the connected ports for link establishment and data transmission. In this chapter, we introduce the rendezvous problem, how it is applied in distributed computing, and in which areas. In Sect. 3.1, we give the definition of the rendezvous problem, present rendezvous examples in distributed systems including multichannel wireless networks and cognitive radio networks in Sects. 3.2 and 3.3. Afterwards, we define rendezvous in distributed system in Sect. 3.4 and present some simple distributed algorithms in Sect. 3.5.

3.1 What is the Rendezvous Problem?

To begin with, consider the following scenario: two young people want to meet each other at some place (such as a supermarket or a large park), but they cannot find each other since it is a huge area and they do not have the specific location to meet. Rendezvous, which means getting together at exactly the same place, asks how they can meet quickly. Obviously, each person can choose between waiting in a fixed place or looking for the other person by searching the different places; obviously, if they both wait, they will never rendezvous. The objective is to design efficient strategies for them to achieve rendezvous as quickly as possible under the assumption that neither one knows the other person's strategy and they do not make any plans beforehand.

This rendezvous problem was first officially introduced in [1] and the continuous version of the problem was formalized in [2]. Following the pioneering work, the rendezvous problem has been studied and applied in many research areas such as

Z. Gu et al., *Rendezvous in Distributed Systems*,
DOI 10.1007/978-981-10-3680-4_3

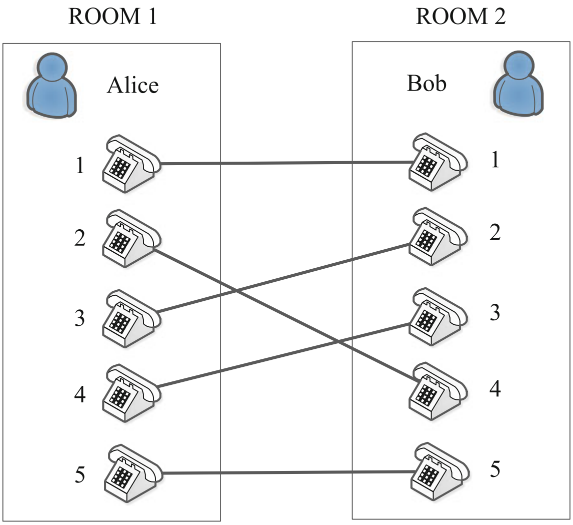


Fig. 3.1 An example of telephone coordination problem

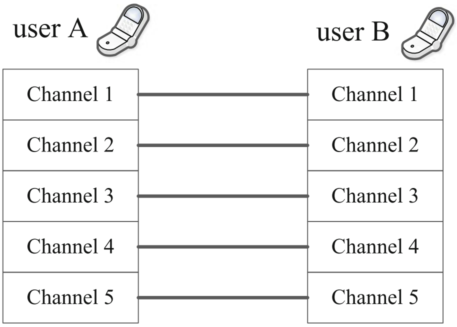
graph theory [3, 4, 6, 18, 33], control theory [16, 39], and mobile robots [19, 28, 38]. In recent years, the rendezvous problem has been applied to wireless networks in designing efficient multichannel MAC protocols, where the wireless devices can access multiple orthogonal channels for communication in order to increase capacity and throughput [34, 35]. In addition, the rendezvous problem for Cognitive Radio Networks (CRNs) has been drawing a lot of attention in the last decade, since CRNs are the most typical multichannel wireless networks and they are the most promising prototype for the next generation wireless networks.

We first introduce a simple example of the rendezvous problem, which is known as the *telephone coordination problem* [1]. As depicted in Fig. 3.1, two people (Alice and Bob) are isolated in two rooms and they can only communicate through the telephones placed in each room. However, they do not know how these telephones are connected and neither person knows when the other will pick which telephone. If both users pick telephone 1 at the same time by luck, they can hear from each other and we call that a *rendezvous*. However, if Alice happens to pick telephone 2 while Bob picks telephone $k \neq 4$, they will not hear anything from the other room and they have to choose between holding the telephone or picking another one, which is similar to the starting scenario in this section.

3.2 Rendezvous in Multichannel Wireless Networks

In multichannel wireless networks (MWN), as in Fig. 3.2, the wireless device is like the person in the example of telephone coordination problem (Fig. 3.1), and the telephones placed in the rooms correspond to the wireless channels that the devices

Fig. 3.2 Rendezvous for multichannel wireless networks



can use for communication, and the connection patterns between the telephones can be thought of as the real physical connections among the channels. Similar to telephone coordination, if both devices choose the same channel at the same time, rendezvous is achieved and they can communicate with each other. The difference between telephone coordination and multichannel wireless networks is that the wireless devices have the same labels as the wireless channels and rendezvous protocols can be designed based on the shared labels.

3.3 Rendezvous in Cognitive Radio Networks

Cognitive Radio Network (CRN) has been hailed as a promising network scheme for solving the spectrum scarcity problem, and rendezvous for CRNs is defined as “the process of one smart or cognitive radio finding another in the spectrum band of interest” [41]. some works also refer to the problem as *neighbor discovery* [7, 9, 52]. In [7, 31], rendezvous is regarded as “the process of two or more radios of users to meet and establish a link on a common channel”, which is more formal and intuitive. Actually, establishing a communication link on a channel involves many detailed implementation steps, such as beaconing and handshaking [41]; we refer to rendezvous process as the process of *choosing the same common channel at the same time*. Although CRN is one special type of multichannel wireless networks, the rendezvous problem in CRN is quite different from both the telephone coordination problem and rendezvous in multichannel wireless networks.

Take Fig. 3.3 as an example, and assume there are six licensed channels $\{1, 2, 3, 4, 5, 6\}$ for both primary users (PUs), i.e. the users who own the licensed spectrum, and secondary users (SUs), i.e. the users who can only use the unlicensed spectrum; PU 1 occupies channels $\{3, 4\}$, PU 2 occupies channels $\{6\}$, PU 3 occupies channels $\{1, 2\}$, and PU 4 occupies channels $\{2, 3\}$, and then the network can be thought of as in Fig. 3.3. Due to the PUs’ occupancy of the licensed channels, not all channels are available for the SUs and this is a huge difference from both the telephone coordination problem and multichannel wireless networks. As shown in the figure,

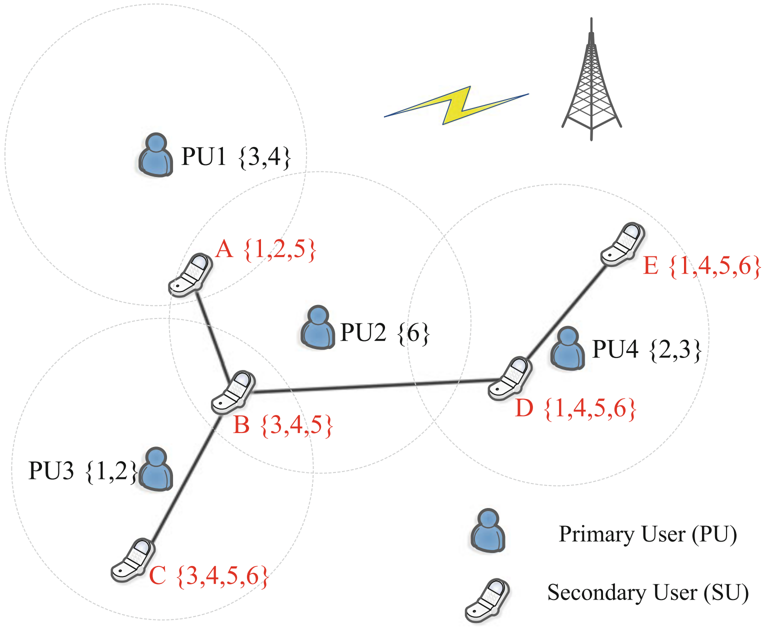


Fig. 3.3 Rendezvous example for cognitive radio network

users A and B can rendezvous on channel 5, users B and C can rendezvous on channels $\{3, 4, 5\}$, users B and D can rendezvous on channels $\{4, 5\}$, and users D and E can rendezvous on channels $\{1, 4, 5, 6\}$. The objective of rendezvous is to design channel accessing algorithms such that every pair of neighboring users (SUs) can access the same available licensed channel at the same time as quickly as possible. Compared with the traditional rendezvous problem, rendezvous in CRN is more challenging for the following reasons:

(1) The wireless devices (SUs) can only access the unused licensed frequency bands (channels) after the spectrum sensing stage, which means not all channels are available to choose from;
(2) traditional multichannel wireless networks assume each frequency band has a fixed label that all SUs are aware of. However, different entities or service providers may have different labels even for the same frequency band in CRN and is no standard for everyone to subscribe to;
(3) different wireless devices may have different capabilities to sense the licensed spectrum, which means the SUs can only sense a portion of the spectrum and they may have only a small common fraction of the total spectrum available for communication;
(4) the status of the licensed spectrum may vary according to the licensed users' (PUs') activities, which implies the sensed available channels can change dynamically.

The first challenge comes from the fact that different SUs may have different available channels to use, which makes the rendezvous problem harder. For example, if both SUs choose channel 1 in Fig. 3.2, they cannot rendezvous if the channel is occupied by nearby PUs, which implies they do not have the chance to use the channel.

The second challenge is due to frequency bands being labeled differently or locally by different SUs. For example, the SUs in Fig. 3.2 cannot rendezvous on channel 1 if they represent different frequency bands. Many extant works assume the labels are fixed and all devices share the same information. The study of a practical problem called *oblivious blind rendezvous* was proposed in [5, 23] where the SUs have no common labels of the channels.

The third challenge arises from the different sensing capabilities of different SUs, and they constitute the so called *Heterogeneous Cognitive Radio Network (HCRN)* [37, 49, 50].

The forth challenge reflects the impact of PUs on the SUs; whenever the PUs occupy the spectrum, the SUs have to release it and exploit new opportunities for rendezvous. Therefore, the status of the licensed channels varies temporally and rendezvous needs to be achieved in a dynamic way.

3.4 Rendezvous in Distributed Systems

From the above examples of the rendezvous problem, we now have the essential meaning of rendezvous in distributed systems: to construct a communication link from the entities' connected ports (or chosen channels) for message transmission.

Considering two neighboring entities in a distributed system, suppose each one has N external ports (which corresponds to N telephones in the telephone coordination problem). Suppose these external ports are labeled with distinguishable identifiers, and some external ports between two neighbors are connected. For example, Fig. 2.4 shows the connections between the neighbors. For node a and b, they can only transmit data to each other when they choose the connected ports, even if they are connected in the network.

Rendezvous in distributed systems is similar to the problem in cognitive radio networks, where each entity (node) does not know the other's information, such as how many ports are connected, how the ports are labeled, etc. All entities have to find out their connections with neighbors on the basis of its local information, i.e. the external ports and perhaps some distinguishable identifiers (such as ID, MAC address, etc.).

Clearly, rendezvous plays an important and fundamental role in distributed systems, which reveals the underlying communication principles. This process can be adopted to construct wireless networks, discover neighbors for autonomous robots, find the existence of other sensor nodes in sensor networks, etc. In this book, we focus on introducing some basic principles and elementary methods of designing efficient rendezvous algorithms, and presenting some elegant results for the field.

3.5 Distributed Rendezvous Algorithms

As very few works have studied the rendezvous process of choosing the connected ports in distributed systems, we mainly highlight some results of rendezvous in some specific systems. Combining these results, we then introduce how to design efficient rendezvous algorithms in distributed systems.

3.5.1 Distributed Telephone Coordination Algorithms

The *telephone coordination problem* seeks to minimize the elapsed time until two isolated individuals can pick the pair of connected telephones. We formally introduce the problem as follows:

Two persons (Alice and Bob) are isolated in two rooms, and there are N telephones in each room. These N telephones are pairwise connected but no one knows how they are connected. Alice and Bob can only communicate with each other by picking up the connected telephones at the same time. Suppose they try round after round, where each person can choose one telephone in each round. If their chosen telephones are connected, rendezvous is achieved and they can communicate. Each person does not know the other person's information (such as identifiers, or telephone chosen strategy); the problem seeks to minimize the expected time (i.e. number of rounds) to achieve rendezvous.

For simplicity, assume the telephones of each person is labeled as $\{1, 2, \ldots, N\}$ randomly, and telephone i of Alice is connected to certain telephone j of Bob (which is not known to the two users). A naïve but reasonable strategy is to pick the telephone randomly in each round. We show that such an algorithm uses expected N rounds for two people to rendezvous.

Lemma 3.1 *Random picking strategy has expected time to rendezvous $ETTR = N$ for two people.*

Proof Let r_t be the event that Alice and Bob pick a pair of connected telephones in the t-th round. Since each person chooses the telephone randomly,

$$Pr(r_t) = \frac{1}{N} \tag{3.1}$$

which computes the probability of event r_t happening. Let r'_t be the event that the users can rendezvous in the t-th round for the first time, then

$$Pr(r'_t) = Pr(\overline{r_1} \bigcap \overline{r_2} \bigcap \ldots \bigcap \overline{r_{t-1}} \bigcap r_t) = \left(1 - \frac{1}{N}\right)^{(t-1)} \cdot \frac{1}{n} \tag{3.2}$$

Thus, we derive the expected time to rendezvous as:

$$\begin{aligned} ETTR &= \sum_{t=1}^{\infty} t \cdot Pr(r_t') \\ &= \sum_{t=1}^{\infty} t \cdot \left(1 - \frac{1}{N}\right)^{(t-1)} \cdot \frac{1}{N} \\ &= N \end{aligned}$$

Following this expected rendezvous time, we can conclude that the users can achieve rendezvous within $O(N \log N)$ rounds with high probability.

Lemma 3.2 *Random picking strategy guarantees rendezvous in* $O(N \log N)$ *rounds for two people with high probability.*

Proof As shown in Lemma 15.2, the probability to rendezvous in each round t is $Pr(r_t) = \frac{1}{N}$. Since both persons select their telephones randomly in each round, r_t, r_t' are independent for any $t \neq t'$. Therefore, the probability that they do not rendezvous in $cN \log N$ (c is a constant) rounds is bounded by:

$$Pr(\overline{r_1} \bigcap \overline{r_2} \bigcap \ldots \bigcap \overline{r_{cN \log N}}) = \left(1 - \frac{1}{N}\right)^{cN \log N} \tag{3.3}$$

When $N \to \infty$,

$$Pr(\overline{r_1} \bigcap \overline{r_2} \bigcap \ldots \bigcap \overline{r_{cN \log N}}) = e^{-c \log N} = \frac{1}{N^c} \tag{3.4}$$

and thus rendezvous happens in $O(N \log N)$ time slots with high probability $1 - \frac{1}{N^c}$.

The random picking strategy is simple. A better algorithm is proposed in [6], which is called the Anderson-Weber (AW) strategy. This algorithm works as follows:

(1) Choose a random value $i \in [1, N]$ and pick the telephone with label i in the first round;

(2) choose a constant value $p \in [0, 1]$ and pick the telephone with label i for the next $N - 1$ rounds with probability p, or pick the telephones in the next $N - 1$ rounds according to a random permutation of label set $\{1, 2, \ldots, i - 1, i + 1, \ldots, N\}$ (with probability $1 - p$);

(3) If rendezvous does not happen, repeat the second step.

In [6], the AW strategy is proved to be better than random picking strategy and the expected time to rendezvous ($ETTR$) is about $0.829N$. We will introduce the details in Chap. 15.

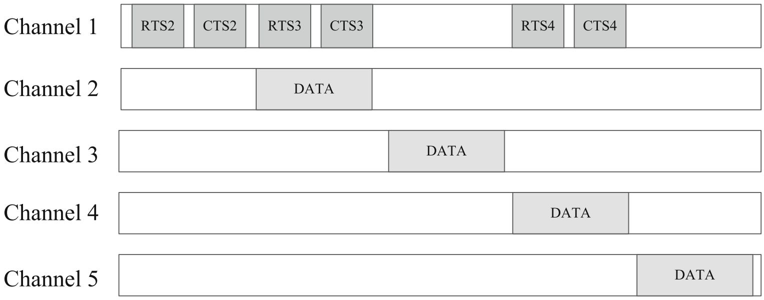


Fig. 3.4 Dedicated Common Control Channel based rendezvous protocol

3.5.2 *Distributed Rendezvous Algorithms for Multichannel Networks*

In traditional wireless networks, there are mainly two types of multichannel MAC protocols for rendezvous: dedicated common control channel (CCC) based rendezvous [25, 47, 48] and channel hopping based rendezvous [8, 13, 42, 43, 45].

The intuitive idea of *dedicated CCC based rendezvous protocols* is to schedule the requests of establishing a link for communication in a dedicated fixed channel. As depicted in Fig. 3.4, channel 1 is the control channel via which the users make an agreement for rendezvous.

For example, one device (we denote it as user A for simplicity) wants to transmit data to user B on channel 2; it sends a request-to-send (RTS) packet on channel 1 and user B responses with a clear-to-send (CTS) packet if it is ready for communication. Then the users utilize channel 2 for communication and no other users can occupy the channel until they finish the transmission. Dynamic Channel Allocation (DCA) [47], Dynamic Channel Allocation with Power Control (DCA-PC) [48], and Dynamic Private Channel (DPC) [25] are some typical examples of this method.

The main advantage of the approach is that it can ensure all users be aware of the status of each channel through communication on the dedicated control channel. Moreover, time synchronization is not needed, which means the users can always achieve rendezvous via the control channel no matter when they want to communicate with others. However, there are several disadvantages:

(1) If the number of channels that can be used is small, the efficiency of utilizing the channels for communications decreases since the dedicated control channel cannot be used for transmission between two specific users;
(2) the dedicated control channel can be easily congested if the number of users is large;
(3) once an attacker finds out the frequency of the dedicated control channel, it can attack and block the channel, which results in no more rendezvous subsequently.

In order to overcome the first disadvantage, a new method called *split phase* was proposed, where time is divided into a sequence of control phase and data (transmis-

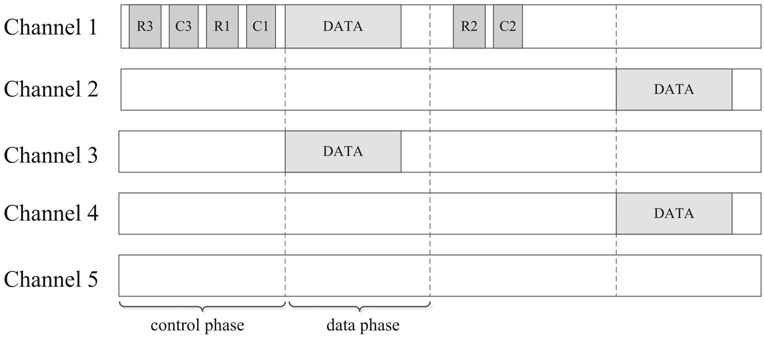


Fig. 3.5 Split phase method

sion) phase. All users utilize the dedicated control channel to make an agreement on the schedule of the channels during the control phase, and they communicate through the scheduled channels in the data transmission phase.

As depicted in Fig. 3.5, time is divided into a sequence of two phases (control phase and data phase) and the users can send RTS or CTS (denoted as R and C in the figure) messages during the control phase on the dedicated common control channel (channel 1 in the figure); then two pairs of users utilize different channels (channels 1 and 3 for the first data phase in the figure) for communication in the data transmission phase. Multichannel Access Protocol (MAP) [13] is one example of this method. Obviously, this method could increase the efficiency of using all channels, but the users need to be synchronized, i.e. they should start the process at the same time.

In view of the disadvantages surrounding the common control channel, many rendezvous protocols not based on a dedicated CCC were proposed and there are mainly two approaches: common hopping and parallel rendezvous [8, 35, 43, 45, 46].

The intuitive idea of *common hopping* is: all users hop through all channels synchronously. If a pair of devices (users) want to communicate with each other, they stop hopping and use the current channel. Once they finish, they rejoin the common hopping pattern just like the other users.

For example, there are five wireless channels as depicted in Fig. 3.6 and all users hop through the channels with the pattern 1, 2, 3, 4, 5, 1, 2, 3, 4, 5, ... If one device (called user A) wants to communicate with user B, it sends an RTS on the current hopping channel (for example, channel 5), and then user B responses with a CTS message on channel 5 if it is ready. Then both users stop hopping and use channel 5 for communication. Once the communication finishes, they rejoin the other users and continue hopping through the channels.

Channel Hopping Multiple Access (CHMA) [45] and Channel Hopping multiple Access with packet Trains (CHAT) [46] are two examples of this approach. The main advantage of this approach is that it uses all channels for communication,

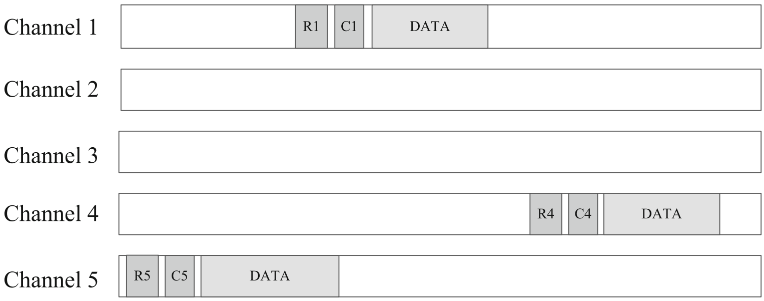


Fig. 3.6 Common hopping rendezvous protocol

which improves the efficiency of channels' usage. However, there are also some disadvantages:

(1) All users have to be synchronized, which means they have to start the process at the same time. This is a serious limitation in reality;
(2) once two users have occupied a channel for communication, the others may also still hop onto this channel. Clearly, this should be forbidden. Some works use carrier sensing to determine if the channel is busy, but this is costly in implementation [51].

The intuitive idea of *parallel rendezvous* is: the users can make an agreement on different channels by hopping through the channels according to different patterns. Slotted Seeded Channel Hopping (SSCH) [8] and Multi-channel MAC (McMAC) [43] are two examples of this method, where the users hop through the channels according to many different pseudorandom sequences that are decided by some seeds. The users can achieve rendezvous on different channels once the user is aware of the other user's seed and rendezvous happens on different channels.

3.5.3 Distributed Rendezvous Algorithms for Cognitive Radio Networks

However, these proposed rendezvous protocols are not applicable to cognitive radio networks (CRNs). There are two general types of rendezvous algorithms for CRN: centralized algorithms and decentralized algorithms.

Similar to the dedicated CCC method in traditional multichannel MAC protocols, *centralized rendezvous algorithms* for CRNs assume a central controller or the existence of a dedicated Common Control Channel (CCC) [1, 3, 32]. It is clear that the users can make an agreement through the central controller or the CCC, which simplifies the rendezvous process.

Although the centralized system is simple to implement and it can provide strong controllability of the channels, it is not flexible or scalable. There are several disadvantages.

(1) First, the central controller or the dedicated CCC can easily get congested as the number of users increases.

(2) Second, the centralized system is vulnerable to adversary attacks since the central controller or the CCC is vital for all rendezvous and for all the users in the network.

(3) Third, the cost to maintain the controller or the CCC is high because it needs to detect the channels' statuses, and to schedule the requests for communication from large quantities of wireless devices.

Therefore, failure or overloading of the central controller or the CCC could lead to dire consequences, and thus centralized algorithms are generally not practical.

Decentralized rendezvous algorithms without such central units have been proposed to avoid the drawbacks of centralized algorithms. There are mainly two categories depending on whether CCC is required.

Some decentralized algorithms establish local CCCs through which a user can contact their neighbors [2, 5]. These algorithms however incur substantial overhead in establishing and maintaining local CCCs.

Since central unit or the dedicated CCC has its inherent limitations, some researchers turned to decentralized algorithms without CCC, which are called *blind rendezvous algorithms* [1–4, 7, 8, 10–12, 17, 20, 26, 27, 36, 53]. Cognitive radios are autonomous and they are able to sense the licensed spectrum and access the available frequency bands on their own. Therefore, it is reasonable that the cognitive radios can communicate with others without relying on some other infrastructure such as some central unit or the CCC. Therefore, blind rendezvous algorithms are more practical and applicable for such intelligent cognitive radios, and large-scale CRNs can be constructed.

The existing blind rendezvous algorithms are divided into two classes: Global Sequence (GS) based rendezvous algorithms [7, 8, 21, 30, 40, 44] and Local Sequence (LS) based rendezvous algorithms [14, 15, 22, 24, 29].

The intuitive idea of *Global Sequence (GS)* based rendezvous algorithm is similar to that of the common hopping approach in traditional multichannel rendezvous protocols. According to the number of licensed channels and their labels, a sequence containing all these channels (labels of the channels) is constructed and all users (SUs) hop through the channels by repeating the sequence, which is called the *Global Sequence*.

For example, we can construct a sequence of length 15 on the basis of 3 channels $\{1, 2, 3\}$, as follows:

$$GS = \{1, 1, 2, 1, 3, 3, 3, 1, 2, 2, 3, 2, 1, 2, 3\} \tag{3.5}$$

Then two users can hop through the channels by repeating the sequence and they can achieve rendezvous at the same time no matter when they start. Suppose time is divided into slots of equal length and the user accesses a channel in each time slot

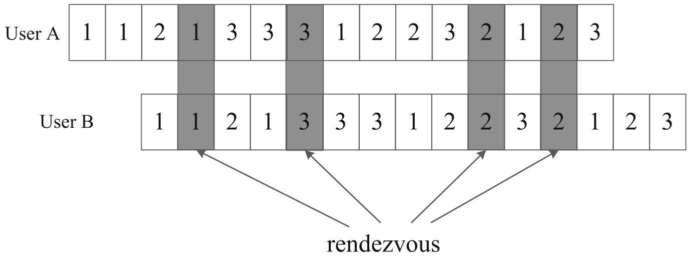


Fig. 3.7 An example of Global Sequence based rendezvous

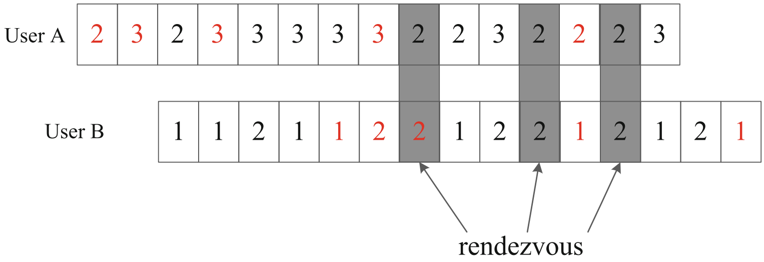


Fig. 3.8 An example of Global Sequence based rendezvous: replacement happens when some channel is not available

according to the sequence. As depicted in Fig. 3.7, two users can achieve rendezvous on all the three channels even if they are asynchronous, i.e. they do not start at the same time.

However, a user may not need to access all licensed channels according to the sequence since some of them may be occupied by the licensed users (PUs); then the user chooses another available channel randomly to continue. For example, in Fig. 3.8, channel 1 is not available for user A and it has to replace it by any available one from {2, 3} (the channels in red represent the replaced ones). Similarly, channel 3 is not available for user B and so a replacement is also needed. Subsequently, the two users rendezvous on channel 2, as shown in the figure.

The first rendezvous happens since user A chooses a random available channel (2) to replace channel 1 in the sequence, and the other two rendezvous situations are generated according to the original channel hopping sequence.

Actually, if the constructed GS has some good properties, rendezvous can always be guaranteed once two users share at least one common available channel, no matter how many channels are available for each user and when do they start the process [3]. We call these GS based rendezvous algorithms and the technique to hop through the licensed channels according to some pre-defined sequences is called *Channel Hopping (CH)* [1, 2, 7, 8, 10, 11]. Jump-Stay (JS) algorithm [7], Channel Rendezvous Sequence (CRSEQ) [8] and Disjoint Relaxed Difference Set (DRDS) based rendezvous algorithm [21] are some state-of-the-art GS based rendezvous algorithms. We will introduce these algorithms in Chap. 8.

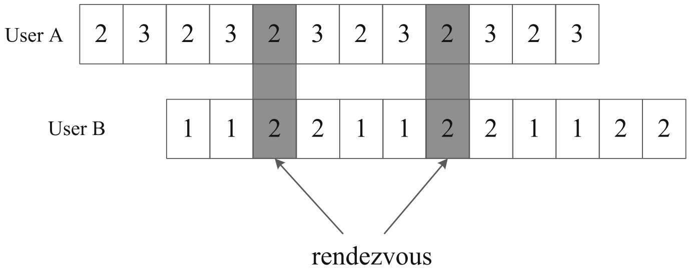


Fig. 3.9 An example of Local Sequence based rendezvous

The other type of blind rendezvous algorithm is *Local Sequence (LS)* based rendezvous algorithms, which is firstly defined in [4].

GS based rendezvous algorithms construct a fixed length sequence for all users and it is inefficient when the number of available channels accounts for a small fraction of all licensed channels. LS based algorithms can accelerate the rendezvous process and the intuitive idea is to construct different sequences for different users based on the local information.

In order to design different sequences, most works assume each user in the network has a distinct identifier (ID) and different sequences can be constructed on the basis of the user's ID and its sensed available channels. As depicted in Fig. 3.9, user *A* hops through its available channels {2, 3} by repeating the sequence {2, 3}, while user *B* accesses the available channels {1, 2} according to the sequence {1, 1, 2, 2}, rendezvous happens on channel 2 in a shorter time compared with Fig. 3.8.

The first algorithm belonging to this type is the Ring-Walk (RW) rendezvous algorithm [5]. Following that, Alternate Hop-and-Wait (AHW) [2], Local Sequence (LS) based rendezvous algorithm, and Modified Local Sequence (MLS) based rendezvous algorithm were proposed, where the the user's ID plays an important role in sequence construction. There are also some works that establish different sequences without the users' ID [1], which we will introduce later in this book.

References

1. Alpern, S. (1976). Hide and seek game. In *Seminar at the Institute* für *Höhere Studien*, Vienna.
2. Alpern, S. (1995). The rendezvous search problem. *SIAM Journal of Control and Optimization*, *33*(3).
3. Alpern, S., Baston, V. J., & Essegaier, S. (1999). Rendezvous search on a graph. *Journal of Applied Probability*, *36*(1), 223–231.
4. Alpern, S., & Beck, A. (1999). Asymmetric rendezvous on the line is a double linear search problem. *Mathematics of Operations Research*, *24*(3), 604–618.
5. Alpern, S., & Pikounis, M. (2000). The telephone coordination game. *Game Theory and Applications*, *5*, 1–10.
6. Anderson, E. J., & Weber, R. R. (1990). The rendezvous problem on discrete locations. *Journal of Applied Probability*, *28*, 839–851.

7. Arachchige, C. J. L., Venkatesan, S. & Mittal, N. (2008). An asynchronous neighbor discovery algorithm for cognitive radio networks. In *DySPAN*.
8. Bahl, P., Chandra, R., & Dunagan, J. (2004). SSCH: Slotted seeded channel hopping for capacity improvement in IEEE 802.11 Ad Hoc wireless networks. In *MobiCom*.
9. Balachandran, K., & Kang, J. H. (2006). Neighbor discovery with dynamic spectrum access in adhoc networks. In *Vehicular Technology Conference*.
10. Bian, K., Park, J.-M., & Chen, R. (2009). A quorum-based framework for establishing control channels in dynamic spectrum access networks. In *Mobicom*.
11. Bian, K., & Park, J.-M. (2011). asynchronous channel hopping for establishing rendezvous in cognitive radio networks. In *IEEE INFOCOM*.
12. Bian, K., Park, J.-M., & Chen, R. (2011). Control channel establishment in cognitive radio networks using channel hopping. *IEEE Journal on Selected Areas in Communications*, *29*(4), 689–703.
13. Chen, J., Sheu, S., & Yang, C. (2003). A new multichannel access protocol for IEEE 802.11 ad hoc wireless LANs. In *Proceeding of 14th IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC)*.
14. Chen, S., Russell, A., Samanta, A. & Sundaram, R. (2014). Deterministic blind rendezvous in cognitive radio networks. In *ICDCS*.
15. Chuang, I., Wu, H.-Y., Lee, K.-R., & Kuo, Y.-H. (2013). Alternate hop-and-wait channel rendezvous method for cognitive radio networks. In *INFOCOM*.
16. Conte, G., & Pennesi, P. (2010). The rendezvous problem with discontinuous control policies. *IEEE Transaction on Automatic Control*, *55*(1), 279–283.
17. Dai, Y., Wu, J., & Xin, C. (2013). Virtual Backbone Construction for Cognitive Radio Networks without Common Control Channel. In *INFOCOM*.
18. Dessmark, A., Fraigniaud, P., & Pelc, A. (2003). Deterministic rendezvous in graphs. In *ESA*.
19. Dudek, G., & Roy, N. (1997). Multi-robot rendezvous in unknown environments. In *AAAI*.
20. Gandhi, R., Wang, C.-C.. & Hu, Y.C. (2012). Fast rendezvous for multiple clients for cognitive radiso using coordinated channel hopping. In *SECON*.
21. Gu, Z., Hua, Q.-S., Wang, Y., & Lau, F. C. M. (2013). Nearly optimal asynchronous blind rendezvous algorithm for cognitive radio networks. In *SECON*.
22. Gu, Z., Hua, Q.-S., & Dai, W. (2014). Local sequence based rendezvous algorithms for cognitive radio networks. In *SECON*.
23. Gu, Z., Hua, Q.-S., Wang, Y. & Lau, F. C. M. (2014). Oblivious rendezvous in cognitive radio networks. In *SIROCCO*.
24. Gu, Z., Hua, Q.-S., & Dai, W. (2014). Fully distributed algorithms for blind rendezvous in cognitive radio networks. In *MOBIHOC*.
25. Hung, W.-C., Law, K. L. E., & Leon-Garcia, A. (2002). A dynamic multi-channel MAC for ad hoc LAN. In *Proceeding of 21st Biennial Symposium on Communications*.
26. Kondareddy, Y., Agrawal, P., & Sivalingam, K. (2008) Cognitive radio network setup without a common control channel. In *MILCOM*.
27. Lazos, L., Liu, S., & Krunz, M. (2009). Spectrum opportunity-based control channel assignment in cognitive radio networks. In *SECON*.
28. Lin, Z., Francis, B., & Maggiore, M. (2009). Getting mobile autonomous robots to rendezvous. *Control of uncertain systems*, pp. 119–137.
29. Lin, Z., Liu, H., Chu, X., & Leung, Y.-W. (2012). Ring-walk rendezvous algorithms for cognitive radio networks. *Ad Hoc & Sensor Wireless Networks*.
30. Liu, H., Lin, Z., Chu, X., & Leung, Y.-W. (2012). Jump-stay rendezvous algorithm for cognitive radio networks. *IEEE Transactions on Parallel and Distributed Systems*, *23*(10), 1867–1881.
31. Liu, H., Lin, Z., Chu, X. & Leung, Y.-W. (2012). Taxonomy and challenges of rendezvous algorithms in cognitive radio networks. In *ICNC*.
32. Lo, B. F. (2011). A survey of common control channel design in cognitive radio networks. *Physical Communication*, *4*, 26–39.
33. Marco, G. D., Gargano, L., Kranakis, E., & Krizanc, D. (2006). Asynchronous deterministic rendezvous in graphs. *Theoretical Computer Science*, *355*, 315–326.

34. Mo, J., So, H.-S. W., & Walrand, J. (2005). Comparison of multi-channel MAC protocols. In *MSWiM*.
35. Mo, J., So, H.-S. W., & Walrand, J. (2008). comparison of multichannel MAC protocols. *IEEE Transaction on Mobile Computing*, *7*(1), 50–65.
36. Perez-Romero, J., Salient, O., Agusti, R., & Giupponi, L. (2007). A novel on-demand cognitive pilot channel enabling dynamic spectrum allocation. In *DySPAN*.
37. Romaszko, S. (2013). A rendezvous protocol with the heterogeneous spectrum availability analysis for cognitive radio ad hoc networks. *Journal of Electrical and Computer Engineering*.
38. Roy, N., & Dudek, G. (2001). Collaborative robot exploration and rendezvous: Algorithms, performance bounds and observations. *Autonomous Robots*, *11*, 117–136.
39. Schenato, L., & Zampieri, S. (2006). Optimal rendezvous control for randomized communication topologies. In *CDC*.
40. Shin, J., Yang, D., & Kim, C. (2010). A channel rendezvous scheme for cognitive radio networks. *IEEE Communications Letters*, *14*(10), 954–956.
41. Silvius, M. D., MacKenzie, A. B., & Bostian, C. W. (2007). A survey of neighbor discovery protocols in modern wireless systems and their application to the design and implementation of "rendezvous" algorithms in smart and cognitive radios. In *Virginia Tech Symposium on Wireless Personal Communications*.
42. So, J. & Vaidya, N. (2004). Multi-channel MAC for ad hoc networks: Handling multi-channel hidden terminals using a single transceiver. In *MobiHoc*.
43. So, H.W., Walrand, J., & Mo, J. (2007). McMAC: A multi-channel MAC proposal for ad hoc wireless networks. In *Proceeding of IEEE Wireless Communications and Networking Conference (WCNC)*.
44. Theis, N. C., Thomas, R. W., & DaSilva, L. A. (2011). Rendezvous for cognitive radios. *IEEE Transactions on Mobile Computing*, *10*(2), 216–227.
45. Tzamaloukas, A., & Garcia-Luna-Aceves, J. J. (2000). Channel-hopping multiple access. In *ICC*.
46. Tzamaloukas, A., & Garcia-Luna-Aceves, J. (2000). Channel-hopping multiple access with packet trains for ad hoc networks. In *Proceeding of IEEE Device Multimedia Communications (MoMuC)*.
47. Wu, S.-L., Lin, Y., Tseng, Y.-C., & Sheu, J.-P. (2000). A new multi-channel MAC protocol with on-demand channel assignment for mobile ad hoc networks. Algorithms and Networks (ISPAN). In *Proceeding of International Symposium on Parallel Architectures*.
48. Wu, S.-L., Lin, C.-Y., Tseng, Y.-C., Lin, C.-Y., & Sheu, J.-P. (2002). A multi-channel MAC protocol with power control for multi-hop mobile ad hoc networks. *The Computer Journal*, *45*(1), 101–110.
49. Wu, C.-C., & Wu, S.-H. (2013). On bridging the gap between homogeneous and heterogeneous rendezvous schemes for cognitive radios. In *MobiHoc*.
50. Wu, S.-H., & Wu, C.-C., Hon, W.-K., & Shin, K. G. (2014). Rendezvous for heterogeneous spectrum-agile devices. In *INFOCOM*.
51. Zannoth, M., Ruhlicke, T., & Klepser, B.-U. (2004). A highly integrated dual-band multimode wireless LAN transceiver. *IEEE Journal of Solid-State Circuits*, *39*(7), 1191–1195.
52. Zhang, D., He, T., Ye, F., Ganti, R. & Lei, H. (2012). EQS: Neighbor discovery and rendezvous maintenance with extended quorum system for mobile sensing applications. In *ICDCS*.
53. Zhao, J., Zheng, H., & Yang, G.-H. (2005). Distributed coordination in dynamic spectrum allocation networks. In *DySPAN*.