# Chapter 11
# Oblivious Blind Rendezvous

**Abstract** Time is divided into slots of equal length and each user can access an available channel in each time slot. Rendezvous is achieved only when the users access the same channel in the same time slot. All the extant blind rendezvous algorithms assume they know the global parameter $N$ and the labels of these $N$ channels, and some works [1] also assume each user knows the number of users in the network. In this part, we introduce the *oblivious blind rendezvous problem*, where *oblivious* means the entities' ports are labeled locally. As introduced in Part II, most blind rendezvous algorithms assume that all entities can see the same labels of the connected ports. However, this assumption is impractical in many distributed systems. For example, in cognitive radio networks, many works assume the licensed spectrum is divided into $N$ non-overlapping channels with fixed labels $\{1, 2, \ldots, N\}$, and each user can access the channel not occupied by any nearby PUs as an *available channel*. However, this assumption may not align with the reality when designing blind rendezvous algorithms. Actually, all users may not see the same labels for the licensed channels. For example, the 'TV white space' that could be sensed by the users has operating frequencies ranging from 470–790 MHz in Europe [2, 4], but it is located in the VHF (i.e. very high frequency) (54–216 MHz) and UHF (i.e. ultra high frequency) (470–698 MHz) bands in the United States [3]. Obviously, the labeling of this space could be different and the same frequency band (channel) may be assigned different labels under different administrations. In a general distributed system, each user has $N$ external ports and it can label these ports locally from $\{1, 2, \ldots, N\}$ in order to distinguish them. Any port $k$ of user $u_i$ may not be connected with port $k$ of user $u_j$ since both users may only use $k$ to identify the different ports. In some special applications, the ports may be labeled according to a global rule. For example, the FTP service uses port 21 of the computers, and the default port for WWW service is 80. We study a more general situation where the users do not have a common labeling rule, and this can be used in many general applications. In this chapter, we first present the system model for the oblivious blind rendezvous problem, in Sect. 11.1; then we introduce the commonly used metrics for evaluation in Sect. 11.2. The problem definition is provided in Sect. 11.3 and we give examples of oblivious blind rendezvous for better understanding in Sect. 11.4. Finally, we summarize the chapter in Sect. 11.5.



Z. Gu et al., *Rendezvous in Distributed Systems*,
DOI 10.1007/978-981-10-3680-4_11

## 11.1 System Model

In this part, we present the rendezvous algorithms for different types of rendezvous settings, on the basis that the ports are oblivious, i.e. all entities do not apply the same labeling rules. Therefore, the rendezvous settings can be represented as:

$$RS_{oblivous} =< Alg, Time, Port, ID, Obli > \tag{11.1}$$

where $Alg \in \{Alg - AS, Alg - S\}$, $Time \in \{Syn, Asyn\}$, $Port \in \{Port - S, Port - AS\}$, and $ID \in \{Non - Anon, Anon\}$.

Technically speaking, suppose there are $M(M \geq 2)$ users in a distributed system, and each user has $N(N \geq 1)$ external ports. In the $RS_{oblivous}$ setting, the ports of each user can be labeled freely by the user itself. Denote all users as:

$$\{u_1, u_2, \ldots, u_M\} \tag{11.2}$$

For simplicity, we assume that each external port have a universal label which is not seen by the users. Denote the set of ports with universal labels as:

$$U = \{u_1, u_2, \ldots, u_N\} \tag{11.3}$$

For any user $u_i$, suppose the user labels the $N$ ports locally as:

$$\{p_i(1), p_i(2), \ldots, p_i(N)\} \tag{11.4}$$

For any two users $u_i, u_j$, the ports $p_i(k)$ and $p_j(k)$ may not be connected after the local labeling.

Suppose the adopted rendezvous algorithms of the users are:

$$\{F_1, F_2, \ldots, F_M\} \tag{11.5}$$

respectively (we suppose user $u_i$ runs algorithm $F_i$).

(1) In the $Alg - AS$ setting, for any two users $u_i, u_j, i \neq j$, $F_i$ and $F_j$ could be different (we use $F_i \neq F_j$ to indicate that they are different).
(2) In the $Alg - S$ setting, all users share the same algorithm, i.e. $\forall i, j \in [1, M]$, $F_i = F_j$.

Similar to blind rendezvous in the distributed systems, time is also assumed to be divided into slots of equal length $2t$, where $t$ is the sufficient time for establishing a communication link between two connected ports. Suppose the system is slot-aligned and each user can choose a port for rendezvous attempt in each time slot. If two users' time slots are not aligned, we can also transfer it to slot-aligned scenario as in Fig. 5.3 in Chap. 5.

Denote the start time of the users as:

$$\{t_1, t_2, \ldots, t_M\} \tag{11.6}$$

respectively, where the start time of user $u_i$ is $t_i$.

(1) In the $Syn$ setting, all users have the same start time, i.e. $\forall i, j \in [1, M], t_i = t_j$.
(2) In the $Asyn$ setting, for any two users $u_i, u_j, i \neq j, t_i$ and $t_j$ could be different, i.e. $t_i \neq t_j$.

Similar to the blind rendezvous problem, some ports of each user may be occupied by other services, and the user could use only a fraction of the $N$ external ports. We say a port is *available* if it is not occupied by others and the user can choose it for communication. For any user $u_i$, it can sense an available port set as $C_i \subseteq U$. Although user $u_i$ may have already labelled the ports locally, it can also label the available ports as

$$C_i = \{c_i(1), c_i(2), \ldots, c_i(k_i)\} \tag{11.7}$$

where $k_i = |C_i|$ represents the number of available ports. Actually, we can regard each available port $c_i(j)$ as a port with global label $u_l$.

(1) In the $Port - S$ setting, all users have the same *global* available ports, i.e. for each user $u_i$ and user $u_j$, $k_i = k_j$, and $\forall l_i \in [1, k_i]$, there exists $l_j \in [1, k_j]$ such that port $c_i(l_i)$ and port $c_j(l_j)$ correspond to the same global label (they are connected).
(2) In the $Port - AS$ setting, all users may not have the same available ports, i.e. for each user $u_i$ and user $u_j$, $\exists l_i \in [1, k_i]$ such that port $c_i(l_i)$ is not connected to any available port in $C_j$.

In the $Port - AS$ setting, in order to guarantee rendezvous, two neighboring users must have at least one common available port, i.e. for any two neighboring users $u_i, u_j$, there exists $l_i \in [1, k_i]$ and $l_j \in [1, k_j]$ such that $c_i(l_i)$ and $c_j(l_j)$ correspond to the same global port, which indicates they are connected. For simplicity, we denote $C_i \bigcap C_j \neq \emptyset$.

In designing oblivious blind rendezvous algorithms, the users' identifiers (IDs) play an important role. Therefore, we define the settings as follows:

(1) In the $Anon$ setting, all users are anonymous and they have no distinct identifiers.
(2) In the $Non - Anon$ setting, each user has a distinct identifier (ID). Denote user $u_i$'s ID as $I_i$. For any two users $u_i, u_j, i \neq j, I_i$ and $I_j$ are different, i.e. $I_i \neq I_j$.

Actually, as there are $M$ users in all and we suppose the a user's ID is a distinct number in the range $[1, \hat{M}]$, where $\hat{M}$ means the maximum ID value for the users. Some works assume $\hat{M} = M$, which means the user could only have continuous IDs in the range [1]. In this book, we assume $\hat{M}$ could be larger than $M$, but we assume it is bounded as $\hat{M} \leq N^c$ where $c$ could be any arbitrary large constant. For simplicity, we re-use notation $M$ to denote $\hat{M}$ in the following chapters.

## 11.2 Metrics

We use *Time to Rendezvous* ($TTR$) to measure the efficiency of rendezvous algorithms. As introduced in the model, the start time of user $u_i$ is denoted as $t_i$. Suppose the finish time of user $u_i$ is $d_i$, where $d_i > t_i$. For any two neighboring users $u_i$ and $u_j$, both users will finish the process at the same time if they achieve rendezvous, thus $d_i = d_j$.

We define the time to rendezvous between two users in oblivious blind rendezvous as:

**Definition 11.1** For two neighboring users $u_i$ and $u_j$, suppose their start time are $t_i, t_j$ respectively, and their finish time are $d_i, d_j$ where $d_i = d_j = d$. The time to rendezvous is defined as:

$$TTR = d - \max\{t_i, t_j\} \tag{11.8}$$

We also denote the rendezvous time as the elapsed time of the user who starts the rendezvous process later. We define the time to rendezvous among all $M$ users as:

**Definition 11.2** Considering all user $\{u_1, u_2, \ldots, u_M\}$ in the system, denote their start time and finish time as $\{t_1, t_2, \ldots, t_M\}$ and $\{d_1, d_2, \ldots, d_M\}$ respectively. The time to rendezvous is defined as:

$$TTR = \max_{1 \le i \le M} d_i - \max_{1 \le i \le M} t_i \tag{11.9}$$

We also use two important metrics to evaluate the proposed rendezvous algorithms:

(1) *Maximum Time to Rendezvous* ($MTTR$) represents the maximum time used to rendezvous in all different situations, such as different available ports, different start times, etc.
(2) *Expected Time to Rendezvous* ($ETTR$) represents the expected time used to rendezvous in all different situations.

$MTTR$ reveals the performance of the rendezvous algorithm in the worst situation, while $ETTR$ reveals the average performance.

## 11.3 Problem Definition

As described in the System Model, there are $M$ users and their available ports can be denoted as:

$$\{C_1, C_2, \ldots, C_M\} \tag{11.10}$$

If rendezvous happens for all users, denote the common available port set as:

$$G = \bigcap_{i=1}^{M} C_i \neq \emptyset \tag{11.11}$$

which are the common available ports (with global labels). Before we define the rendezvous problem for multiple users in the system, we first formulate the *oblivious blind rendezvous (OBR)* problem between two users, as follows:

**Problem 11.1** OBR-2: Given an available channel set $C \subseteq U$ and the ID $I \in [1, M]$, design an algorithm to access global ports over different time slots $t : f_{C,I}(t) \in C$ such that for any two users $u_i$ and $u_j$ with $C_i, C_j \subseteq U$, $C_i \bigcap C_j \neq \emptyset$ and ID $I_i, I_j \in [1, M]$, $I_i \neq I_j$ respectively,

$$\forall \delta, \exists T_\delta, \ s.t. \ f^i_{C_i,I_i}(T_\delta + \delta) = f^j_{C_j,I_j}(T_\delta). \tag{11.12}$$

The $TTR$ value is $T_\delta$ when user $u_j$ starts the rendezvous process $\delta$ time slots later than user $u_i$. The $MTTR$ value of the algorithms is defined as:

$$MTTR_{f^i,f^j} = \max_{\forall\delta} T_\delta \tag{11.13}$$

The objective is to design rendezvous algorithms with bounded $MTTR$ value to guarantee rendezvous between two users. Notice that, $f^i$ represents the algorithm user $u_i$ adopts. If we are to design *symmetric algorithms* for the users, both users $u_i$ and $u_j$ should adopt the same algorithm, i.e. $f^i = f^j$.

*Remark 11.1* When user $u_j$ starts the rendezvous process earlier than user $u_i$, $\delta < 0$ in Eq. (11.12).

Based on the rendezvous problem definition of two users, we formulate the *Oblivious Blind Rendezvous Problem for Multiple Users in the Multihop system* as follows:

**Problem 11.2** Consider a multihop system with $M(M \geq 2)$ users where each user has a distinct ID $I \in [1, M]$. Denote the available port set for user $u_i$ as:

$$C_i = \{c_i(1), c_i(2), \ldots, c_i(k_i)\} \tag{11.14}$$

where $k_i = |C_i|$. Let $G = \bigcap_i C_i$ and $G \neq \emptyset$. Design distributed algorithms for the users such that all users are guaranteed to rendezvous on the same port in $G$, regardless of the different times when the users begin the process.

## 11.4 Examples of Oblivious Blind Rendezvous

Figure 11.1 is an example for OBR-2. Assume there are 5 external ports:

$$U = \{u_1, u_2, u_3, u_4, u_5\} \tag{11.15}$$

of which $u_1, u_3$ are available for user $u_a$ with ID $I_a = 1$:

$$C_a = \{c_a(1), c_a(2)\} \tag{11.16}$$

and the ports are labeled locally as:

$$\begin{cases} c_a(1) = u_1 \\ c_a(2) = u_3 \end{cases}$$

Meanwhile, $u_1, u_2, u_4, u_5$ are available for user $u_b$ with ID $I_b = 2$:

$$C_b = \{c_b(1), c_b(2), c_b(3), c_b(4)\} \tag{11.17}$$

and the ports are labeled locally as:

$$\begin{cases} c_b(1) = u_5 \\ c_b(2) = u_4 \\ c_b(3) = u_2 \\ c_b(4) = u_1 \end{cases}$$

Consider a simple algorithm: user $u_a$ repeats accessing the ports according to the sequence:

$$\{c_a(1), c_a(1), c_a(2), c_a(2)\} \tag{11.18}$$

while user $u_b$ accesses ports according to the sequence:

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| user $u_a$ | $\text{Time}_a$ | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| | Sequence | $c_a(1)$ | $c_a(1)$ | $c_a(2)$ | $c_a(2)$ | $c_a(1)$ | $c_a(1)$ | $c_a(2)$ | $c_a(2)$ |
| | Port | $u_1$ | $u_1$ | $u_3$ | $u_3$ | $u_1$ | $u_1$ | $u_3$ | $u_3$ |

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| user $u_b$ | $\text{Time}_b$ | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| | Sequence | $c_b(1)$ | $c_b(2)$ | $c_b(3)$ | $c_b(4)$ | $c_b(1)$ | $c_b(2)$ | $c_b(3)$ | $c_b(4)$ |
| | Port | $u_5$ | $u_2$ | $u_4$ | $u_1$ | $u_5$ | $u_2$ | $u_4$ | $u_1$ |

**Fig. 11.1** An example of OBR-2

| | | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 9 | 10 |
|---|---|---|---|---|---|---|---|---|---|---|
| | Time$_a$ | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 9 | 10 |
| user $u_a$ | Sequence | 1 | 2 | 1 | 2 | 1 | 2 | 1 | 2 | 1 |
| | Port | $c_a(1)$ | $c_a(2)$ | $c_a(1)$ | $c_a(2)$ | $c_a(1)$ | $c_a(2)$ | $c_a(1)$ | $c_a(2)$ | $c_a(1)$ |
| user $u_b$ | Sequence | - | 1 | 2 | 3 | 4 | 1 | 2 | 3 | 4 |
| | Port | - | $c_b(1)$ | $c_b(2)$ | $c_b(3)$ | $c_b(4)$ | $c_b(1)$ | $c_b(2)$ | $c_b(3)$ | $c_b(4)$ |

**Fig. 11.2** An example of OBR-2 when the users adopt a symmetric algorithm and $\delta = 1$

| | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | Time$_a$ | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 9 | 10 |
| user $u_a$ | Sequence | 1 | 2 | 1 | 2 | 1 | 2 | 1 | 2 | 1 |
| | Port | $c_a(1)$ | $c_a(2)$ | $c_a(1)$ | $c_a(2)$ | $c_a(1)$ | $c_a(2)$ | $c_a(1)$ | $c_a(2)$ | $c_a(1)$ |
| user $u_b$ | Sequence | - | - | 1 | 2 | 3 | 4 | 1 | 2 | 3 |
| | Port | - | - | $c_b(1)$ | $c_b(2)$ | $c_b(3)$ | $c_b(4)$ | $c_b(1)$ | $c_b(2)$ | $c_b(3)$ |

**Fig. 11.3** An example of OBR-2 when the users adopt a symmetric algorithm and $\delta = 2$

$$\{c_b(1), c_b(2), c_b(3), c_b(4)\} \tag{11.19}$$

When user $u_b$ starts the process $\delta = 2$ time slots later, rendezvous can be achieved on port $u_1$ with $TTR = 4$ when $c_a(1) = c_b(4) = u_1$, as illustrated in Fig. 11.1.

However, it is easy to check that the above simple algorithm cannot guarantee rendezvous for all scenarios such as $\delta = 0$.

In Fig. 11.1, two users run different strategies to achieve rendezvous, which is impossible in practice since all users should run the same algorithm, i.e. symmetric algorithm. Figures 11.2 and 11.3 show another example of OBR-2 where the users share the same strategy. Similar to the above, user $u_a$ and user $u_b$ have the same available port sets, as in Fig. 11.1, i.e. user $u_a$ has two available ports, and the available port set is:

$$C_a = \{c_a(1), c_a(2)\} \tag{11.20}$$

and user $u_a$ has four and the set is:

$$C_b = \{c_b(1), c_b(2), c_b(3), c_b(4)\} \tag{11.21}$$

but only one common available port exists:

$$c_a(1) = c_b(4) = u_1 \tag{11.22}$$

Different from Fig. 11.1, both users run a same simple algorithm: each user accesses the ports by repeating the sequence:

$$\{1, 2, \ldots, k\} \tag{11.23}$$

which are of local labels, where $k$ is the number of available ports. Thus user $u_a$ repeats accessing the ports:

$$\{c_a(1), c_a(2), c_a(1), c_a(2), \ldots\} \tag{11.24}$$

until rendezvous, and similarly for user $u_b$.

For the asynchronous scenario, supposing user $u_b$ starts the attempt $\delta = 1$ time slot later, rendezvous is achieved as depicted in Fig. 11.2 at time slot 5 since $c_a(1) = c_b(4)$. However, it is easy to see that the above simple algorithm cannot guarantee rendezvous for all scenarios such as when $\delta = 2$, as illustrated in Fig. 11.3.

Combining the two examples, we aim to design efficient distributed algorithms with bounded $TTR$ values for different types of rendezvous settings.

## 11.5 Chapter Summary

In this part, we propose the oblivious blind rendezvous problem and present different types of rendezvous algorithms. Oblivious blind rendezvous assumes that the external ports are not labeled by a universal rule, and the users have to label the ports themselves locally. In Chap. 12, we design asymmetric algorithms for the users, which is similar to the blind rendezvous problem. In Chap. 13, we study symmetric algorithms for the users in a distributed system. We first assume the users are non-anonymous and they can design algorithms on the basis of the distinguishable identifiers. The method of designing fully distributed rendezvous algorithms is then presented in Chap. 14 where no global information is utilized in rendezvous, such as the number of external ports, the number of users in the system, and the maximum identifiers for the users. We study oblivious blind rendezvous for anonymous users in Chap. 15 and we introduce several randomized algorithms. Finally, we extend the oblivious blind rendezvous between two users to the rendezvous problem among multiple users in a multi-hop system in Chap. 16.

To begin with, we introduce the oblivious blind rendezvous problem, with examples. Different from blind rendezvous, the external ports are not labeled globally and the users may see different "local" labels of a pair of connected ports. In this chapter, we introduce the system model including several aspects in a rendezvous setting: $Algorithm$, $Time$, $Port$ and $ID$. We will present algorithms for different rendezvous settings. We also use Maximum Time to Rendezvous ($MTTR$) and Expected Time to Rendezvous ($ETTR$) to evaluate the rendezvous algorithms. These two metrics are used to evaluate the performance of the worst situation and the average performance respectively. We also provide some examples of oblivious blind rendezvous to demonstrate the differences with blind rendezvous.

## References

1. 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*.
2. ETSI. (2012). EN 301 598 white space devices (WSD); wireless access systems operating in the 470 MHz to 790 MHz frequency band.
3. Flores, R.E. Guerra, A.B., & Kightly, E.W. (2013). IEEE 802.11af: A standard for TV white space spectrum sharing. *IEEE Communications Magazine*, *62*, 92–100.
4. Ofcom. (2013). Regulatory Requirements for White Space Devices in the UHF TV band. http://www.cept.org/Documents/se-43/6161/.