Chapter 15
Oblivious Blind Rendezvous for Anonymous Users

Abstract In this chapter, we present *symmetric algorithms* for the blind rendezvous problem between two *anonymous* users. In the setting, we fix Alg and ID as:

$$RS =< Alg - S, Time, Port, Anon, Obli > \tag{15.1}$$

where $Port \in \{Port - S, Port - AS\}$ and $Time \in \{Syn, Asyn\}$. It is easy to see that there are 4 different rendezvous settings when Alg is fixed as symmetric, ID is fixed as anonymous, and $Label$ is fixed as oblivious. Different from Chaps. 13 and 14, we assume the users have no distinct identifiers to break symmetry in distributed computing. This anonymous setting makes the oblivious blind rendezvous problem difficult. In Sect. 15.1, we show the hardness due to such anonymity which gives rise to the result that no deterministic algorithm could exist for the oblivious blind rendezvous problem. Then, we present in Sect. 15.2 an efficient randomized algorithm for two *port-symmetric* users no matter whether they are synchronous or asynchronous, which achieves short expected time to rendezvous. For the most difficult setting, where the users are *port-asymmetric*, we present randomized algorithms that work well for both synchronous and asynchronous users. Finally, we summarize the chapter in Sect. 15.4.

15.1 Hardness of Anonymity

In this section, we show that there is no deterministic distributed algorithm for the OBR problem between two anonymous users, i.e. the users do not have unique identifiers (IDs) or distinguishable information.

Theorem 15.1 *There is no deterministic distributed algorithm for the OBR problem between two anonymous users.*

Proof Suppose there exists such a deterministic algorithm:

$$F : f \mapsto [1, N] \tag{15.2}$$

Z. Gu et al., *Rendezvous in Distributed Systems*,
DOI 10.1007/978-981-10-3680-4_15

for the OBR problem between two anonymous users. Consider two users A and B with two available port sets C_A, C_B satisfying

$$k_a = k_b = \lceil N/2 \rceil + 1 \tag{15.3}$$

where $k_a = |C_A|, k_b = |C_B|$ represent the number of available ports for each user; we set:

$$\forall i, c_a(i) \neq c_b(i) \tag{15.4}$$

where $c_a(i)$ and $c_b(i)$ represent the ports that are labeled as i locally by users A and B respectively.

Since $k_a + k_b > N$, at least one common available port exists between sets C_A and C_B, which is the elementary condition that the users could rendezvous. However, we show that the algorithm F cannot guarantee rendezvous.

Let $\delta = 0$, i.e. two users start the rendezvous algorithm F at the same time, and denote a_t, b_t as the labels of the ports to access in time slot t respectively; thus:

$$\begin{aligned} a_t &= f(a_1, a_2, \ldots, a_{t-1}, k_a, N) \\ b_t &= f(b_1, b_2, \ldots, b_{t-1}, k_b, N) \end{aligned}$$

since the chosen port in time slot t is only related to the user's local information: the local labels of the ports, the number of available ports, the number of all ports N (notice that, N may not be known in advance, but we show the theorem even when this value is available for the users).

We prove that users A and B will choose the ports with the same local label $a_t = b_t$ in time slot t through the inductive method.

(1) When both users start the rendezvous algorithm, they only find out that there are $\lceil N/2 \rceil + 1$ available ports and they are indistinguishable from each other. Thus they will make the same choice and access the port with the same local label in the first time slot, i.e. $a_0 = b_0$;

(2) Suppose $a_i = b_i$ when $0 \leq i \leq t-1$. Then user A should access port a_t as:

$$a_t = f(a_1, a_2, \ldots, a_{t-1}, k_a, N) \tag{15.5}$$

while user B should access port b_t with label:

$$b_t = f(b_1, b_2, \ldots, b_{t-1}, k_b, N) \tag{15.6}$$

Since $a_i = b_i$ when $0 \leq i \leq t-1$, $k_a = k_b$, both users have the same input to the deterministic algorithm F and the outputs of the algorithm should be the same, i.e. $a_t = b_t$.

Combining the two aspects, both users A and B choose the ports with the same local label $a_t = b_t$ for any time slot t. Since:

$$c_a(i) \neq c_b(i), \forall i \in [1, \lceil N/2 \rceil + 1] \tag{15.7}$$

(obviously, this setting can be easy fulfilled). Rendezvous never happens even for two synchronous users. Therefore, no deterministic algorithm exists for the OBR problem between two anonymous users.

15.2 Port-Symmetric Rendezvous

In this section, we handle the port-symmetric rendezvous where two users have the same set of available ports. For simplicity, we assume all ports are available and it is easy to extend the algorithm to the general port-symmetric setting.

Recall the telephone coordination problem [1] (introduced in Sect. 1): two users A and B are isolated in two rooms and there are N telephones in each of them. The telephones are pairwise connected in some unknown fashion. For simplicity, assuming the telephones are labeled $\{1, 2, \ldots N\}$ randomly (locally) for each user, and telephone $i \in [1, N]$ of user A is connected to a certain telephone j of user B, but they do not know the connection pattern. Time is also assumed to be divided into slots of equal length and the user can select one telephone in each time slot by sending a "hello" message. If they pick a pair of connected telephones in the same time slot, they can hear from each other and it is called *rendezvous* (all time slots are regarded as aligned). Time to rendezvous (*TTR*) denotes the time cost when all users have begun the selection process and the objective is to minimize the expected time to rendezvous (*ETTR*).

When all the ports are available for the two anonymous users, it is similar to the telephone coordination problem. One simple and intuitive idea is random selection, where each user selects a random port to attempt rendezvous. This method has expected time to rendezvous (*ETTR*) as N time slots and it seems to be the best solution.

However, a better algorithm called the Anderson-Weber strategy (AW) is proposed in [2]; for two *synchronous* users and it works as follows:

(1) Choose a random value $i \in [1, N]$ and pick the i-th telephone in the first time slot;
(2) choose a constant $p \in [0, 1]$ and the user picks the i-th telephone for the next $N - 1$ time slot with probability p, or picks the telephones in the next $N - 1$ time slots according to a random permutation of set $\{1, 2, \ldots, i-1, i+1, \ldots, N\}$ (with probability $1 - p$);
(3) if rendezvous does not happen, repeat the second step.

It has been proved that the AW strategy is optimal when $N = 2, p = \frac{1}{2}$ (this is also shown in [3]) and $N = 3, p = \frac{1}{3}$ [1, 4, 5, 7]. It has also been conjectured that AW is asymptotically optimal when $N \geq 4$ (specifically, $ETTR = 0.8289N$ and $p = 0.2475$ when $N \to \infty$). In [6], it is proved that the AW strategy is not optimal when $N = 4$ and to find an optimal algorithm even for two synchronous users is still an open problem. In addition, the AW strategy does not work for asynchronous users. In this section, we present a randomized algorithm which works well for both synchronous and asynchronous users.

Before we describe the algorithm, we present some useful results from probability theory.

Let A be an event, $Pr(A)$ denote the probability event A happens and $Pr(\overline{A}) = 1 - Pr(A)$ the probability that event A does not happen. Let $\{B_1, B_2, \dots, B_n\}$ be a set of disjoint events whose union is the entire sample space; then according to the law of total probability:

$$Pr(A) = \sum_{i=1}^{n} Pr(A \bigcap B_i) = \sum_{i=1}^{n} Pr(A|B_i) \cdot Pr(B_i) \tag{15.8}$$

Suppose X is a random variable and denote $E(X)$ as the expectation of X. If events $\{B_1, B_2, \dots, B_n\}$ are mutually exclusive and exhaustive, according to the law of total expectation:

$$E(X) = \sum_{i=1}^{n} E(X|B_i) \cdot Pr(B_i) \tag{15.9}$$

Let $[N]$ denote the set $\{1, 2, \dots, N\}$, and A_N^k be the number of methods selecting k elements out of $[N]$:

$$A_N^k = N(N-1)\cdots(N-k+1) \tag{15.10}$$

15.2.1 Intuitive Ideas

To begin, we show a lower bound of the expected time to rendezvous (*ETTR*) when two users are allowed to use asymmetric strategies (i.e. different algorithms). Then we derive $ETTR = N$ for the random selection algorithm. Combining the two results, we then describe the intuitive ideas in designing the proposed randomized distributed algorithm.

Lemma 15.1 *For any distributed algorithm solving the OBR problem between two anonymous users, the expected time to rendezvous satisfies:*

$$ETTR \geq \frac{N+1}{2} \tag{15.11}$$

even when the users are allowed to use asymmetric algorithms.

Proof This lemma can be derived as in [2]. Let r_t be the event that two users select the same universal port (the local labels of the port may be different) in the t-th time slot. Without loss of generality, suppose user A starts later than user B; t is the time stamp of user A since time to rendezvous (TTR) records the time cost when two users have both begun the process.

Since the users do not know the other's labels of the ports, we have:

$$Pr(r_t) = \frac{1}{N} \tag{15.12}$$

Note that, r_t means they can rendezvous in the t-th time slot, but not necessarily for the first time. Thus the probability two users rendezvous in the first t time slots can be bounded as:

$$Pr(r_1 \bigcup r_2 \bigcup \ldots \bigcup r_t) \leq \min \left\{ 1, \sum_{i=1}^{t} Pr(r_i) \right\} = \min\{1, \frac{t}{N}\} \tag{15.13}$$

The bound on the right side of the inequality is achieved by the strategy $\mathscr{S}$:

* One user accesses a fixed port all the time, while the other user hops through the ports according to a random permutation of $[N]$.

Obviously, we can derive the expected time to rendezvous for this strategy as:

$$ETTR = \frac{\sum_{i=1}^{n} i}{N} = \frac{N+1}{2} \tag{15.14}$$

and thus the lemma holds.

Although the strategy $\mathscr{S}$ can guarantee fast rendezvous for two anonymous users, it is inapplicable to the OBR-2 problem since two anonymous users cannot decide which role to take.

When it comes to the situation in which two asynchronous users should run a symmetric algorithm, random selection seems to be reasonable, which can be denoted as $\mathscr{R}$:

* Each user accesses a port randomly in each time slot.

We derive the expected time to rendezvous and show the efficiency of the strategy.

Lemma 15.2 *$\mathscr{R}$ has expected time to rendezvous $ETTR = N$ for two asynchronous users.*

Proof Let r_t be the event that the users access the same universal port (the local labels may be different) in the t-th time slot. Since both users access the port randomly, we can derive:

$$Pr(r_t) = \frac{1}{N} \tag{15.15}$$

Let r'_t be the event that the users can rendezvous in the t-th time slot for the first time; then we have:

$$Pr(r_t') = Pr(\overline{r_1} \bigcap \overline{r_2} \bigcap \ldots \bigcap \overline{r_{t-1}} \bigcap r_t) = \left(1 - \frac{1}{N}\right)^{(t-1)} \cdot \frac{1}{N} \tag{15.16}$$

Therefore, we can compute the *ETTR* value as:

$$ETTR = \sum_{t=1}^{\infty} t \cdot Pr(r_t') = \sum_{t=1}^{\infty} t \cdot \left(1 - \frac{1}{N}\right)^{(t-1)} \cdot \frac{1}{N} = N \tag{15.17}$$

So the lemma holds.

Lemma 15.3 *$\mathscr{R}$ guarantees rendezvous in $O(N \log N)$ time slots for two asynchronous users with high probability.*

Proof As shown in Lemma 15.2, the probability to rendezvous in each time slot t is $Pr(r_t) = \frac{1}{N}$. Since strategy $\mathscr{R}$ accesses the ports randomly for every time slot, events r_t, r_t' are independent for any $t \neq t'$. So the probability that they do not rendezvous in $cN \log N$ (c is a constant) time slots is bounded by:

$$Pr(\overline{r_1} \bigcap \overline{r_2} \bigcap \ldots \bigcap \overline{r_{cN \log N}}) = \left(1 - \frac{1}{N}\right)^{cN \log N} \tag{15.18}$$

When $N \rightarrow \infty$, we derive that:

$$Pr(\overline{r_1} \bigcap \overline{r_2} \bigcap \ldots \bigcap \overline{r_{cN \log N}}) = e^{-c \log N} = \frac{1}{N^c} \tag{15.19}$$

Therefore, rendezvous happens in $O(N \log N)$ time slots with high probability $1 - \frac{1}{N^c}$, which concludes the lemma.

Though strategy $\mathscr{S}$ designs asymmetric algorithms for two users, the idea that one user waits while the other user hops through all ports provides an important foundation for designing efficient randomized algorithms. The strategy $\mathscr{R}$ seems to be the best randomized algorithm where we use pure randomization in making decisions. However, if we could combine both intuitions to design randomized algorithms, we may achieve better results.

15.2.2 Stay or Random Selection Algorithm

In the section, we introduce a simple randomized distributed algorithm called *Stay or Random Selection (SRS)* that achieves rendezvous faster than random selection ($\mathscr{R}$).

As shown in Algorithm 15.1, the user makes a choice at the beginning of each *block*, which is defined as N consecutive time slots. If the chosen random value $p' \leq p$ (p is a constant we need to compute and define), the user accesses a random port and waits at it for a block of time slots; otherwise a random permutation of $[N]$ is generated and the user accesses the corresponding port in the permutation for each time slot of the block. We denote the first choice as the *stay pattern* and the second one as the *jump pattern*. The user keeps this process until rendezvous.

Algorithm 15.1 Stay or Random Selection Algorithm

```
p is a pre-defined constant in [0, 1];
while Not rendezvous do
    Select a random value p' ∈ [0, 1];
    if p' ≤ p then
        Select a random number in [N] and access the corresponding port for the following N time slots;
    else
        Generate a random permutation of [N] and access the corresponding ports in the following N time slots according to the permutation;
    end if
end while
```

The intuitive ideas of $\mathscr{S}$ and $\mathscr{R}$ are combined in our algorithm. Although the description of the algorithm is simple, finding the optimal value of p that minimizes the *ETTR* is very difficult. Compared with the AW strategy for the telephone coordination problem, our algorithm also works for two asynchronous users, which is not treated in existing works.

15.2.3 Synchronous Users Scenario

The SRS algorithm is applicable for both synchronous and asynchronous users. In this section, we analyze the rendezvous efficiency for two synchronous users and compute the appropriate p value in Algorithm 15.1.

In the synchronous situation, two users start the algorithm at the same time. As shown in Algorithm 15.1, time is divided into blocks of length N. At the beginning of each block, the user decides to be in the stay or jump pattern. Denote $r(S, J)$ as the event that user A is in the stay pattern and user B is in the jump pattern. The other three events are denoted as $r(S, S)$, $r(J, S)$, $r(J, J)$ similarly. Denote the expected time to rendezvous (*ETTR*) for synchronous users as T_s which can be formulated as:

$$T_S = E(S, J)Pr(S, J) + E(J, S)Pr(J, S) + E(S, S)Pr(S, S) + E(J, J)Pr(J, J) \quad (15.20)$$

where $Pr(S, J)$ is the probability that event $r(S, J)$ happens, $Pr(J, S)$ the probability that event $r(J, S)$ happens, $Pr(S, S)$ the probability that event $r(S, S)$ happens and $Pr(J, J)$ the probability that event $r(J, J)$ happens. Similarly, $E(S, J)$ is the expected time to rendezvous if user A is in the stay pattern and user B is in the jump pattern; $E(J, S)$ is the expected time to rendezvous if user A is in the jump pattern and user

B in the stay pattern; $E(S, S)$ is the expected time to rendezvous if user A is in the stay pattern and user B is in the stay pattern; and $E(J, J)$ is the expected time to rendezvous if user A is in the jump pattern and user B is in the jump pattern.

We first analyze the *ETTR* values for the four events respectively.

(1) Event $r(S, S)$:
When both users choose the stay pattern, the only chance to rendezvous is that the ports they select represent the same global port. Thus the probability to rendezvous is:

$$\frac{Pr(S, S) = 1}{N} \tag{15.21}$$

and 1 time slot is needed when rendezvous happens. Therefore

$$E(S, S) = \frac{1}{N} \cdot 1 + \left(1 - \frac{1}{N}\right)(N + T_s) \tag{15.22}$$

(2) Event $r(S, J)$ and $r(J, S)$:
When one user chooses the stay pattern while the other one is in the jump pattern, rendezvous happens for certain:

$$Pr(S, J) = Pr(J, S) = 1 \tag{15.23}$$

and the expected time to rendezvous is:

$$E(S, J) = E(J, S) = \frac{N + 1}{2} \tag{15.24}$$

(3) Event $r(J, J)$:
When two users are both in the jump pattern, the expected rendezvous time is formulated as in Lemma 15.4.

Lemma 15.4 *If two users are both in the jump pattern, the expected rendezvous time is:*

$$E(J, J) = (N + 1)(1 - p(N + 1, 0) - p(N, 0)) + p(N, 0)(N + T_s) \tag{15.25}$$

where $p(N, 0)$ *is the probability that the users cannot rendezvous according to two random permutations of* $[N]$ *they generate respectively.*

Proof Let J_1, J_2 be the permutations of $[N]$ that the users generate respectively when they are in the jump pattern. Let variable m be the first time they meet on a specific position. Supposing J_1, J_2 rendezvous exactly $x \geq 1$ times, then:

$$Pr(m \leq i) = \frac{\binom{N+1-i}{x}}{\binom{N}{x}}, \forall 1 \leq i \leq N - x + 1. \tag{15.26}$$

For any given N and fixed value $1 \le x \le N$, denote the expected time to rendezvous as $E(m, x)$ which can be formulated as:

$$\begin{aligned} E(m,x) &= \sum_{i=1}^{N-x+1} i \cdot Pr(m=i) \\ &= \sum_{i=1}^{N-x+1} Pr(m \le i) \\ &= \sum_{i=1}^{N-x+1} \frac{\binom{N+1-i}{x}}{\binom{N}{x}} \\ &= \frac{N+1}{x+1} \end{aligned} \tag{15.27}$$

We accumulate the expectations for all possible N and x to derive:

$$\begin{aligned} E(J,J) &= \sum_{x=0}^{N} p(N,x) \cdot E(m,x) \\ &= \sum_{x=1}^{N} p(N,x) \cdot E(m,x) + p(N,0)(N+T_s) \end{aligned} \tag{15.28}$$

here $p(N, x)$ is the probability that J_1, J_2 rendezvous exactly $x \ge 1$ times. On the basis that x rendezvous points exist between J_1, J_2, the remaining part cannot rendezvous and the probability is denoted as $p(N-x, 0)$. As the x rendezvous points have $x!$ different permutations, we derive:

$$p(N,x) = \frac{p(N-x,0)}{x!} \tag{15.29}$$

Combining Eq. (15.27), we get

$$\begin{aligned} \sum_{x=1}^{N} p(N,x) \cdot E(m,x) &= \sum_{x=1}^{n} E(m,x) \cdot \frac{p(N-x,0)}{x!} \\ &= (N+1) \cdot \sum_{x=1}^{n} \frac{p(N-x,0)}{(x+1)!} \\ &= (N+1) \cdot \sum_{x=1}^{n} p(N+1,x+1) \\ &= (N+1)(1-p(N,0)-p(N+1,0)) \end{aligned} \tag{15.30}$$

Plugging this into the formulation of $E(J, J)$, the lemma holds.

Then we need to calculate $p(N, 0)$ which denotes the probability that J_1, J_2 do not rendezvous. Assuming J_1 is the permutation generated by user A, we count the number of permutations (J_2) that do not rendezvous with J_1 (denote the number as D_N), which can be computed as in Lemma 15.5.

Lemma 15.5 $D_N = N! \cdot \sum_{k=0}^{N}(-1)^k \cdot \frac{1}{k!}$. *When N is large enough,* $D_N = \lfloor \frac{N!}{e} \rfloor$

Proof Consider two permutations J_1, J_2 of $[N]$ generated by user A and user B respectively. Let $J_1(i), J_2(i)$ be the labels of the i-th position. Since no rendezvous happens, $J_1(N)$ does not represent the same universal port as $J_2(N)$. Suppose $J_2(N)$ and $J_1(i), 1 \leq i < N$ represent the same universal port while $J_1(N)$ and $J_2(j), 1 \leq j < N$ represent the same universal port.

(1) If $i = j$, rendezvous cannot happen for all other $N - 2$ positions and the number of such permutations is D_{N-2};
(2) if $i \neq j$, the number of such permutations is D_{N-1}.

Therefore, we can compute:

$$D_N = (N-1)(D_{N-1} + D_{N-2}) \tag{15.31}$$

It is easy to see $D_1 = 0, D_2 = 1$ and $p(N, 0) = \frac{D_N}{N!}$. Plugging these into the equation we get:

$$N!p(N, 0) = (N-1)((N-1)!p(N-1, 0) + (N-2)!p(N-2, 0))$$

After the transformation we get:

$$N!p(N, 0) - N!p(N-1, 0) = -(N-1)!p(N-1, 0) + (N-1)!p(N-2, 0) \tag{15.32}$$

Let $w(N) = N!p(N, 0) - N!p(N-1, 0)$, we can solve the above equation as:

$$w(N) = -w(N-1) = (-1)^{N-1}w(1) = (-1)^{(N-1)}$$

Then, we have:

$$p(N, 0) - p(N-1, 0) = \frac{1}{N!}(-1)^N \tag{15.33}$$

and we can solve the equation as:

$$p(N, 0) = \frac{1}{N!}\sum_{i=0}^{N}(-1)^i \frac{1}{i!} \tag{15.34}$$

Therefore, D_N is computed as:

$$D_N = N! \cdot p(N, 0) = N! \cdot \sum_{k=0}^{N}(-1)^k \cdot \frac{1}{k!} \tag{15.35}$$

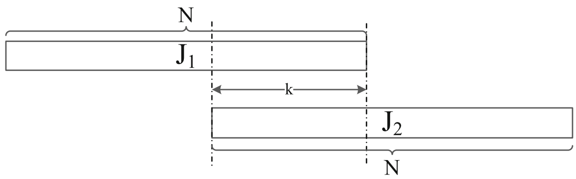

Fig. 15.1 An example of the overlapping between two random permutations

When $N \to \infty$, $p(N, 0)$ is the Taylor expansion of e^{-1}, and thus $D_N = \lfloor \frac{N!}{e} \rfloor$. So the lemma holds.

Since $p(N, 0) = \frac{D_N}{N!}$, we can combine Eqs. (15.20)–(15.25) to derive the expected time to rendezvous as in Theorem 15.2.

Theorem 15.2 *The expected time to rendezvous (ETTR) of the SRS algorithm (Algorithm 15.1) for two* synchronous and port-symmetric *users can be formulated as:*

$$T_s = \frac{T_1 + T_2 + T_3}{1 - p^2(1 - \frac{1}{N}) - (1 - p)^2 p(N, 0)} \tag{15.36}$$

where:

$$\begin{aligned} T_1 &= p(1 - p)(N + 1) \\ T_2 &= (1 - p)^2[(N + 1)(1 - p(N, 0) - p(N + 1, 0)) + p(N, 0) \cdot N] \\ T_3 &= [p^2(\frac{1}{N} + N - 1)] \end{aligned} \tag{15.37}$$

In order to find out the optimal p that minimizes T_s, let $\frac{dT_s}{dp} = 0$, and we can compute the value of p. When $N \to \infty$, $p \approx 0.2475$ and $T_s \approx 0.8289N$, which matches the state-of-the-art results [2].

15.2.4 Asynchronous Users Scenario

In order to analyze the algorithm for two asynchronous users, we present a method to derive the *ETTR* value for a general situation, i.e. an arbitrary N value. Similar to the analysis for two synchronous users, we first consider the scenario where two users are both in the jump pattern and are in the asynchronous situation.

Suppose sequences J_1, J_2 are two random permutations of $[N]$ generated by users A and B respectively. Let $r(N, k)$ denote the event that two users rendezvous in the overlapping fragment of length k (as in Fig. 15.1) and $R(N, k)$ denote the corresponding variable. Let $p(N, k, j)$ be the probability that they rendezvous exactly j times in the overlapping part; it is obvious that:

$$Pr(\overline{r(N, k)}) = p(N, k, 0) \tag{15.38}$$

We introduce Lemmas 15.6–15.8 to compute $p(N, k, j)$ and $E(R(N, k))$. To begin with, we introduce the inclusion-exclusion principle.

For two sets A, B, the cardinality of set $A \bigcup B$ can be computed as:

$$|A \bigcup B| = |A| + |B| - |A \bigcap B| \tag{15.39}$$

When there are multiple sets $A_1, A_2, \dots, A_n$, we can compute the cardinality of set $\bigcup_{i=1}^{n} A_i$ as:

$$\begin{aligned} |\bigcup_{i=1}^{n}| &= \sum_{i=1}^{n} A_i - \sum_{1 \le i < j \le n} |A_i \bigcap A_j| + \sum_{1 \le i < j < k \le n} |A_i \bigcap A_j \bigcap A_k| \\ &\quad + \cdots + (-1)^{n-1} |A_1 \bigcap A_2 \bigcap \dots \bigcap A_n| \\ &= \sum_{k=1}^{n} (-1)^{k+1} \left(\sum_{1 \le i_1 < \dots < i_k \le n} |A_{i_1} \bigcap \cdots \bigcap A_{i_k}| \right) \end{aligned} \tag{15.40}$$

Lemma 15.6 $p(N, k, 0) = \sum_{i=0}^{k} (-1)^i \cdot \frac{\binom{k}{i}}{A_N^i}$.

Proof This lemma can be derived easily. Denote $q(N, k, i)$ as the probability that two users rendezvous at least i times in the overlapping fragment which has length k; when $1 \le i \le k$, we have:

$$q(N, k, i) = \frac{\binom{k}{i} \cdot (N - i)!}{N!} = \frac{\binom{k}{i}}{A_N^i} \tag{15.41}$$

Applying the inclusion-exclusion principle,

$$p(N, k, 0) = 1 - q(N, k, 1) + q(N, k, 2) + \cdots + (-1)^i q(N, k, i) = \sum_{i=0}^{k} (-1)^i \cdot \frac{\binom{k}{i}}{A_N^i} \tag{15.42}$$

so the lemma holds.

Lemma 15.7 $p(N, k, j) = p(N - j, k - j, 0) \cdot \frac{\binom{k}{j}}{A_N^j}$.

Proof Let $D(N, k, j)$ denote the number of permutations when J_1, J_2 have overlapping length k and exactly j rendezvous points. It is obvious that:

$$D(N, k, j) = N! \cdot N! \cdot p(N, k, j) \tag{15.43}$$

Similarly, we compute:

$$D(N - j, k - j, 0) = (N - j)! \cdot (N - j)! \cdot p(N - j, k - j, 0) \tag{15.44}$$

For any instance of the $D(N-j, k-j, 0)$ situations, it can be transformed into some instance in the $D(N, k, j)$ situations. Clearly, there are $\binom{N}{j}$ numbers (rendezvous points) that can be chosen, and there are $k-j+1$ positions to place the first number, $k-j+2$ positions for the second one, until $k-j+j$ positions for the j-th number. Thus, we derive:

$$D(N, k, j) = D(N-j, k-j, 0) \cdot \binom{N}{j} \cdot \frac{k!}{(k-j)!} \tag{15.45}$$

Combining the relationships of $D(N, k, j), p(N, k, j)$ and $D(N-j, k-j, 0)$, $p(N-j, k-j, 0)$, we get:

$$\begin{aligned} p(N, k, j) = \frac{D(N, k, j)}{N! \cdot N!} &= \frac{D(N-j, k-j, 0) \cdot \binom{N}{j} \cdot \frac{k!}{(k-j)!}}{N! \cdot N!} \\ &= p(N-j, k-j, 0) \cdot \frac{(N-j)! \cdot (N-j)!}{N! \cdot N!} \cdot \binom{N}{j} \cdot \frac{k!}{(k-j)!} \\ &= p(N-j, k-j, 0) \cdot \frac{\binom{k}{j}}{A_N^j} \end{aligned} \tag{15.46}$$

Thus the lemma holds.

Similar to Lemma 15.4, we bound the *ETTR* of $R(N, k)$ in Lemma 15.8.

Lemma 15.8 $E(R(N, k)) = (N+1)(1 - p(N+1, k+1, 0)) - (k+1)$ $p(N, k, 0)$.

Proof When $j > k$, $p(N, k, j) = 0$ and we accumulate the probabilities when $j = 0, 1, \ldots, k$ as:

$$\sum_{j=0}^{k} p(N, k, j) = p(N, k, 0) + p(N, k, 1) + \cdots + p(N, k, k) = 1. \tag{15.47}$$

Supposing two users rendezvous exactly j times in the overlapping part of length k (denote the event as $r(N, k, j)$). Let $r_{k,j,1}$ be the time when they first rendezvous and let q_i be the probability that $r_{k,j,1}$ is no more than i, thus:

$$q_i = Pr(r_{k,j,1} \le i \mid r(N, k, j)) = \frac{\binom{k+1-i}{j}}{\binom{k}{i}} \tag{15.48}$$

where $i \le k+1-j$. When $i > k+1-j$, $q_i = 0$. We can formulate the expected time of the first rendezvous as:

$$
\begin{aligned}
E(r_{k,j,1} \mid r(N,k,j)) &= \sum_{i=1}^{k+1-j} i \cdot Pr(r_{k,j,1} = i \mid r(N,k,j)) \\
&= \sum_{i=1}^{k+1-j} Pr(r_{k,j,1} \leq i \mid r(N,k,j)) \\
&= \sum_{i=1}^{k+1-j} q_i = \frac{\sum_{i=1}^{k+1-j} \binom{k+1-i}{j}}{\binom{k}{j}} \\
&= \frac{k+1}{j+1}
\end{aligned} \tag{15.49}
$$

Thus we accumulate all the expectations when $j = 1, 2, \ldots, k$ as:

$$
\begin{aligned}
E(R(N,k)) &= \sum_{j=1}^{k} p(N,k,j) \cdot E(r_{k,j,1} \mid r(N,k,j)) \\
&= \sum_{j=1}^{k} p(N-j, k-j, 0) \cdot \frac{\binom{k}{j}}{A_N^j} \cdot \frac{k+1}{j+1} \\
&= (N+1) \cdot \sum_{j=1}^{k} p(N-j, k-j, 0) \frac{\binom{k+1}{j+1}}{A_{N+1}^{j+1}} \\
&= (N+1) \cdot \sum_{j=1}^{k} p(N+1, k+1, j+1) \\
&= (N+1)(1 - p(N+1, k+1, 0)) - (k+1)p(N,k,0)
\end{aligned} \tag{15.50}
$$

We use Lemma 15.7 and plug in Eq. (15.47) to derive Eq. (15.50), and thus the lemma holds.

Without loss of generality, suppose user B starts the algorithm δ time slots later than user A. Since each user makes a choice every N time slots independently, we consider the situation as user B starts:

$$
d = \delta \mod N \tag{15.51}
$$

time slots later than user A. Let T_1 be the *ETTR* value when user A is in the stay pattern, and T_2 be the *ETTR* value when user A is in the jump pattern. Then we derive the *ETTR* value for the asynchronous scenario as follows.

Theorem 15.3 *For an arbitrary N, the optimal p of Algorithm 15.1 can be determined numerically and the minimized ETTR is computed as:*

$$
ETTR = p \cdot T_1 + (1-p) \cdot T_2 \tag{15.52}
$$

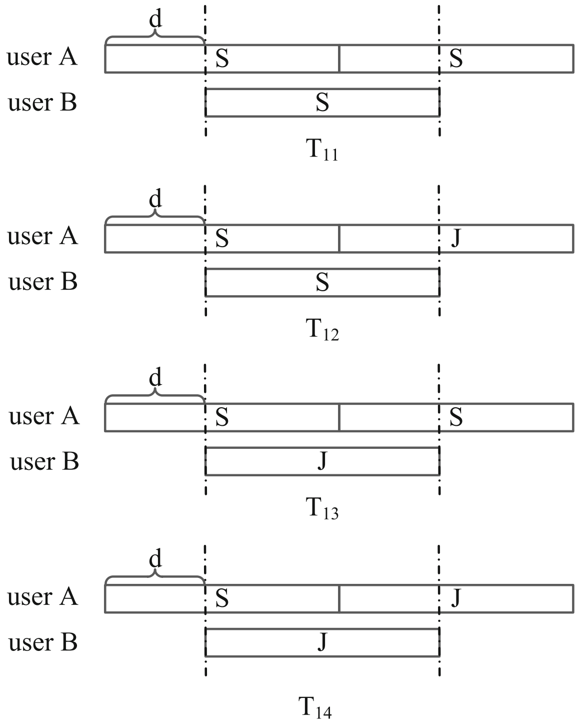

Fig. 15.2 Different situations of asynchronous rendezvous scenario when computing T_1

Proof In order to compute T_1, T_2, there are 4 situations respectively as shown in Figs. 15.2 and 15.3.

Since we treat every N time slots as a block, user B's first block intersects with user A's two consecutive blocks. Let B_1 denote user B's first block's pattern, and A_1, A_2 denote user A's two intersecting blocks' patterns. For simplicity, we write $B_1 = S$ for the stay pattern and $B_1 = J$ for the jump pattern. Thus:

$$\begin{aligned} T_1 &= ETTR(A_1 = S) \\ T_2 &= ETTR(A_1 = J) \end{aligned} \tag{15.53}$$

Denote $A_1 \bigcap B_1$ and $A_2 \bigcap B_1$ as the overlapping fragment, and thus:

$$\begin{aligned} |A_1 \bigcap B_1| &= N - d \\ |A_2 \bigcap B_1| &= d \end{aligned} \tag{15.54}$$

here $|.|$ represents the length of the overlapping part. The situations of the overlapping fragments are also illustrated in Figs. 15.2 and 15.3.

As depicted in Fig. 15.2, we denote

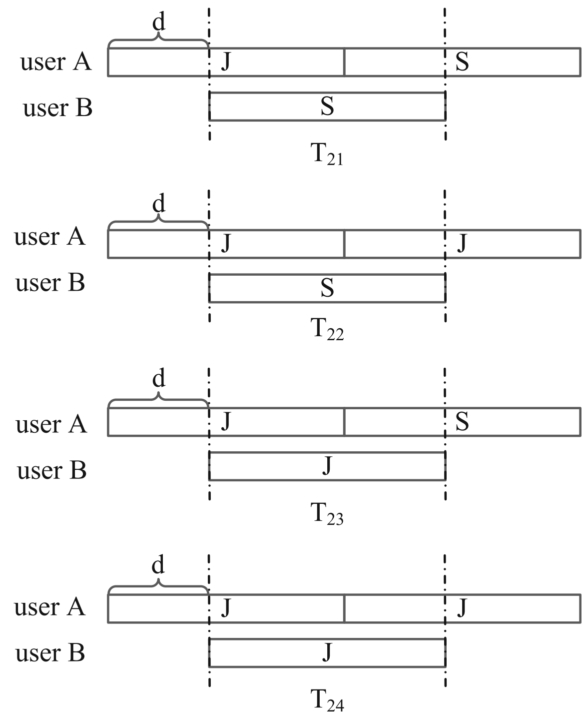

Fig. 15.3 Different situations of asynchronous rendezvous scenario when computing T_2

$$\begin{aligned} T_{11} &= ETTR(B_1 = S, A_2 = S \mid A_1 = S) \\ T_{12} &= ETTR(B_1 = S, A_2 = J \mid A_1 = S) \\ T_{13} &= ETTR(B_1 = J, A_2 = S \mid A_1 = S) \\ T_{14} &= ETTR(B_1 = J, A_2 = J \mid A_1 = S) \end{aligned} \quad (15.55)$$

and we can get the formulation of T_1:

$$T_1 = p^2 \cdot T_{11} + p(1-p) \cdot T_{12} + (1-p)p \cdot T_{13} + (1-p)^2 \cdot T_{14} \quad (15.56)$$

Similarly, from Fig. 15.3, we derive Eq. (15.57) for T_2:

$$T_2 = p^2 \cdot T_{21} + p(1-p) \cdot T_{22} + (1-p)p \cdot T_{23} + (1-p)^2 \cdot T_{24} \quad (15.57)$$

where:

$$\begin{aligned} T_{21} &= ETTR(B_1 = S, A_2 = S \mid A_1 = J) \\ T_{22} &= ETTR(B_1 = S, A_2 = J \mid A_1 = J) \\ T_{23} &= ETTR(B_1 = J, A_2 = S \mid A_1 = J) \\ T_{24} &= ETTR(B_1 = J, A_2 = J \mid A_1 = J) \end{aligned} \quad (15.58)$$

Now we present the method to produce the expressions of $T_{11}, T_{12}, T_{13}, T_{14}$ and $T_{21}, T_{22}, T_{23}, T_{24}$.

Let $r_1(T_{ij})$ and $r_2(T_{ij})$ be the events that the users rendezvous in $A_1 \bigcap B_1$, $A_2 \bigcap B_1$, respectively, and $R_1(T_{ij})$, $R_2(T_{ij})$ be the corresponding variables, where $1 \leq i \leq 2$, $1 \leq j \leq 4$. We derive the formulation of T_{ij} as:

$$\begin{aligned} T_{ij} =& Pr(\overline{r_1(T_{ij})})[(N-d) + Pr(r_2(T_{ij})) \cdot E(R_2(T_{ij})) + Pr(\overline{r_2(T_{ij})}) \cdot (d + T_\delta) \\ &+ Pr(r_1(T_{ij})) \cdot E(R_1(T_{ij})) \end{aligned} \tag{15.59}$$

where $\delta = (j-1) \mod 2 + 1$ (i.e. $\delta = 1$ when $j = 1, 3$; otherwise $\delta = 2$). Thus we can plug in the following probabilities and expectations to generate Eqs. (15.60) and (15.61).

For event $r_1(T_{11})$, we compute:

$$\begin{cases} Pr(r_1(T_{11})) = \frac{1}{N} \\ E(R_1(T_{11})) = 1 \end{cases}$$

For event $r_2(T_{11})$, we compute:

$$\begin{cases} Pr(r_2(T_{11})) = \frac{1}{N} \\ E(R_2(T_{11})) = 1 \end{cases}$$

For event $r_1(T_{12})$, we compute:

$$\begin{cases} Pr(r_1(T_{12})) = \frac{1}{N} \\ E(R_1(T_{12})) = 1 \end{cases}$$

For event $r_2(T_{12})$, we compute:

$$\begin{cases} Pr(r_2(T_{12})) = \frac{d}{N} \\ E(R_2(T_{12})) = \frac{d+1}{2} \end{cases}$$

For event $r_1(T_{13})$, we compute:

$$\begin{cases} Pr(r_1(T_{13})) = \frac{N-d}{N} \\ E(R_1(T_{13})) = \frac{N-d+1}{2} \end{cases}$$

For event $r_2(T_{13})$, we compute:

$$\begin{cases} Pr(r_2(T_{13})) = \frac{d}{N} \\ E(R_2(T_{13})) = \frac{d+1}{2} \end{cases}$$

For event $r_1(T_{14})$, we compute:

$$\begin{cases} Pr(r_1(T_{14})) = \frac{N-d}{N} \\ E(R_1(T_{14})) = \frac{N-d+1}{2} \end{cases}$$

For event $r_2(T_{14})$, we compute:

$$\begin{cases} Pr(\overline{r_2(T_{14})}) = p(N, d, 0) \\ Pr(r_2(T_{14})) \cdot E(R_2(T_{11})) = E(R(N, d)) \end{cases}$$

For event $r_1(T_{21})$, we compute:

$$\begin{cases} Pr(r_1(T_{21})) = \frac{N-d}{N} \\ E(R_1(T_{21})) = \frac{N-d+1}{2} \end{cases}$$

For event $r_2(T_{21})$, we compute:

$$\begin{cases} Pr(r_2(T_{21})) = \frac{1}{N} \\ E(R_2(T_{21})) = 1 \end{cases}$$

For event $r_1(T_{22})$, we compute:

$$\begin{cases} Pr(r_1(T_{22})) = \frac{N-d}{N} \\ E(R_1(T_{21})) = \frac{N-d+1}{2} \end{cases}$$

For event $r_2(T_{22})$, we compute:

$$\begin{cases} Pr(r_2(T_{22})) = \frac{d}{N} \\ E(R_2(T_{22})) = \frac{d+1}{2} \end{cases}$$

For event $r_1(T_{23})$, we compute:

$$\begin{cases} Pr(\overline{r_1(T_{23})}) = p(N, N-d, 0) \\ Pr(r_1(T_{23})) \cdot E(R_1(T_{23})) = E(R(N, N-d)) \end{cases}$$

For event $r_2(T_{23})$, we compute:

$$\begin{cases} Pr(r_2(T_{23})) = \frac{d}{N} \\ E(R_2(T_{23})) = \frac{d+1}{2} \end{cases}$$

For event $r_1(T_{24})$, we compute:

Table 15.1 Optimal p and minimized $ETTR$ values in Algorithm 15.1

N	Optimal p	$ETTR$	$ETTR/N$
3	0.302	2.887	0.9624
5	0.280	4.749	0.9499
10	0.233	9.332	0.9332
50	0.206	45.765	0.9159
100	0.203	91.354	0.9135
200	0.202	182.467	0.9123
500	0.201	455.806	0.9116
1000	0.200	911.369	0.9113
2000	0.200	1822.432	0.9112
10000	0.200	911.149	0.9111
$N \to \infty$	0.200	$0.9111N$	0.9111

$$\begin{cases} Pr(\overline{r_1(T_{24})}) = p(N, N-d, 0) \\ Pr(r_1(T_{24})) \cdot E(R_1(T_{24})) = E(R(N, N-d)) \end{cases}$$

For event $r_2(T_{24})$, we compute:

$$\begin{cases} Pr(\overline{r_2(T_{24})}) = p(N, d, 0) \\ Pr(r_2(T_{24})) \cdot E(R_2(T_{24})) = E(R(N, d)) \end{cases}$$

Combining these equations, we can derive the expressions of $T_{11}, T_{12}, T_{13}, T_{14}$, as follows:

$$\begin{cases} T_{11} = \frac{1}{N} \cdot 1 + \frac{N-1}{N} \cdot \frac{1}{N} \cdot (N-d+1) + (\frac{N-1}{N})^2 \cdot (N+T_1) \\ T_{12} = \frac{1}{N} \cdot 1 + \frac{N-1}{N} \cdot \frac{d}{N} \cdot (N-d+\frac{d+1}{2}) + \frac{N-1}{N} \cdot \frac{N-d}{N} \cdot (N+T_2) \\ T_{13} = \frac{N-d}{N} \cdot \frac{N-d+1}{2} + \frac{d}{N} \cdot \frac{d}{N} \cdot (N-d+\frac{d+1}{2}) + \frac{d}{N} \cdot \frac{N-d}{N} \cdot (N+T_1) \\ T_{14} = \frac{N-d}{N} \cdot \frac{N-d+1}{2} + \frac{d}{N} \cdot (N-d+E(R(N,d))) + \frac{d}{N} \cdot p(N,d,0) \cdot (d+T_2) \end{cases} \tag{15.60}$$

Similarly, we derive the expression of $T_{21}, T_{22}, T_{23}, T_{24}$:

$$\begin{cases} T_{21} = \frac{N-d}{N} \cdot \frac{N-d+1}{2} + \frac{d}{N} \cdot \frac{1}{N} \cdot (N-d+1) + \frac{d}{N} \cdot \frac{N-1}{N} \cdot (N+T_1) \\ T_{22} = \frac{N-d}{N} \cdot \frac{N-d+1}{2} + \frac{d}{N} \cdot \frac{d}{N} \cdot (N-d+\frac{d+1}{2}) + \frac{d}{N} \cdot \frac{N-d}{N} \cdot (N+T_2) \\ T_{23} = E(R(N, N-d)) + p(N, N-d, 0) \cdot \frac{d}{N} \cdot (N-d+\frac{d+1}{2}) \\ \qquad +p(N, N-d, 0) \cdot \frac{N-d}{N} \cdot (N+T_1) \\ T_{24} = E(R(N, N-d)) + p(N, N-d, 0)(N-d+E(R(N,d))) \\ \qquad +p(N, N-d, 0)p(N,d,0)(d+T_2) \end{cases} \tag{15.61}$$

Combining Eqs. (15.52)–(15.61), p is optimized numerically for arbitrary N and the minimized *ETTR* of our algorithm can be computed as Eq. (15.52). Table 15.1 lists some results derived through this numerical method.

15.3 Port-Asymmetric Rendezvous

In Sect. 15.2, we introduce a good method that works better than picking a random port for rendezvous when the users have symmetric available ports. In this section, we handle the port-asymmetric situations and present several randomized algorithms.

15.3.1 Random Picking Algorithm

One trivial way to handle the oblivious blind rendezvous between two anonymous, port-asymmetric users is to pick the available port for rendezvous randomly. We describe such an algorithm in Algorithm 15.2.

Algorithm 15.2 Random Picking Algorithm

Denote the set of the user's available port set as C;
Denote $C = \{c(1), c(2), \ldots, c(k)\}$ where $k = |C|$;
$t := 0$;
while Not terminated **do**
  Pick a random number $i \in [1, k]$ and access port $c(i)$ in time t;
  $t := t + 1$;
end while

As depicted in the algorithm, the user has k available ports and it labels these ports locally as:

$$\{c(1), c(2), \ldots, c(l)\} \tag{15.62}$$

where each port $c(i)$ corresponds to a global port, but the user does not know the relationship between them. We derive the time complexity of achieving rendezvous with high probability.

Consider any two neighboring users u_a and u_b, and suppose the corresponding available ports sets are

$$\begin{aligned} C_a &= \{c_a(1), c_a(2), \ldots, c_a(k_a)\} \\ C_b &= \{c_b(1), c_b(2), \ldots, c_b(k_b)\} \end{aligned} \tag{15.63}$$

respectively, where $k_a = |C_a|$ and $k_b = |C_b|$ record the number of available ports. Since we study rendezvous between two port-asymmetric users, sets C_a, C_b can be different.

Denote $C_g = C_a \bigcap C_b$, which represents the set of common available ports between user u_a and user u_b. Notice that, two users have at least one common available port and $|C_g| \geq 1$. We derive below the expected time to rendezvous of the random picking algorithm.

Lemma 15.9 *The expected time to rendezvous of the random picking algorithm is* $ETTR = \frac{|C_a||C_b|}{|C_g|}$ *for two port-asymmetric users.*

Proof Let r_t be the event when both users access the same universal port (the local labels may be different) in the t-th time slot. Since both users access the port randomly, we analyze the probability as follows.

User u_a accesses each available port randomly, and the probability of accessing each port $c_a(i)$ is:

$$Pr_a(i) = \frac{1}{|C_a|} \tag{15.64}$$

Similarly, the probability of user u_b accessing each port $c_b(j)$ is:

$$Pr_b(j) = \frac{1}{|C_b|} \tag{15.65}$$

Therefore, the probability of user u_a accessing port $c_a(i)$ and user u_b accessing port $c_b(j)$ at the same time is:

$$Pr(u_a accesses c_a(i), u_b accesses c_b(j)) = \frac{1}{|C_a||C_b|} \tag{15.66}$$

As there are $|C_g|$ common available ports for both users, for each port $c_g(l) \in C_g$, there exist i, j such that:

$$\begin{cases} c_a(i) = c_g(l) \\ c_b(j) = c_g(l) \end{cases}$$

here "=" means they correspond to the same universal port. Therefore, there are $|C_g|$ situations where they may access the same port and the probability is:

$$Pr(r_t) = \frac{|C_g|}{|C_a||C_b|} \tag{15.67}$$

Since both users make decisions randomly and independently in each time slot, we can compute the *ETTR* value as:

$$\begin{aligned} ETTR &= \sum_{t=1}^{\infty} t \cdot Pr(r_t) \\ &= \sum_{t=1}^{\infty} t \cdot \left(1 - \frac{|C_g|}{|C_a||C_b|}\right)^{(t-1)} \cdot \frac{|C_g|}{|C_a||C_b|} \\ &= \frac{|C_a||C_b|}{|C_g|} \end{aligned} \tag{15.68}$$

Therefore, the lemma holds.

15.3.2 Random Prime Selection and Sequential Accessing Algorithm

Though the random picking algorithm has short expected time to rendezvous, it cannot guarantee rendezvous within a bounded number of time slots with high probability. Actually, we can design another algorithm that guarantees rendezvous if a certain condition is satisfied.

Algorithm 15.3 Random Prime Selection and Sequential Accessing Algorithm

1: Denote the set of the user's available port set as C;
2: Denote $C = \{c(1), c(2), \ldots, c(k)\}$ where $k = |C|$;
3: Choose a random prime number $p \in [k, 3k]$;
4: $t := 0$;
5: **while** Not terminated **do**
6: $\quad x := t \mod p$;
7: $\quad index := (x - 1) \mod k + 1$;
8: $\quad$ Access port $c(index)$ for rendezvous;
9: $\quad t := t + 1$;
10: **end while**

As described in Algorithm 15.3, suppose the user has k available ports and it chooses a random prime number p in the range of $[k, 3k]$. After picking prime number p, the user accesses port sequentially by its local labels from 1 to p. However, p may be larger than k and we map the number in $[k + 1, p]$ to $[1, k]$ as in Line 7. For example, $k = 2$ and we choose $p = 3$, and the user accesses the ports as in Fig. 15.4.

For two users u_a and u_b, denote their available port sets as:

$$\begin{aligned} C_a &= \{c_a(1), c_a(2), \ldots, c_a(k_a)\} \\ C_b &= \{c_b(1), c_b(2), \ldots, c_b(k_b)\} \end{aligned} \tag{15.69}$$

Fig. 15.4 An example of Algorithm 15.3

Time	1	2	3	4	5	6	7	8	9
Sequence	c(1)	c(2)	c(3)	c(1)	c(2)	c(3)	c(1)	c(2)	c(3)
Port	c(1)	c(2)	c(1)	c(1)	c(2)	c(1)	c(1)	c(2)	c(1)

respectively, where $k_a = |C_a|$ and $k_b = |C_b|$ record the number of available ports. Denote the chosen prime numbers for two users as p_a and p_b. We show that they can rendezvous within $p_a p_b$ time slots for sure, if $p_a \neq p_b$.

Theorem 15.4 *Two port-asymmetric users (synchronous or asynchronous) can achieve rendezvous within $p_a p_b$ time slots under the situation that $p_a \neq p_b$.*

Proof Denote the port accessing sequences of user u_a and u_b as:

$$S_a = \{c_a(1), c_a(2), \ldots, c_a(p_a), c_a(1), c_a(2), \ldots, c_a(p_a), \ldots\} \tag{15.70}$$

and

$$S_b = \{c_b(1), c_b(2), \ldots, c_b(p_b), c_b(1), c_b(2), \ldots, c_b(p_b), \ldots\} \tag{15.71}$$

We do not consider the situation where some port in $(c_a(k_a), c_a(p_a)]$ may not exist. Suppose user u_a is δ time slots earlier than user u_b. Consider one common available port c_g between two users. Suppose it corresponds to port $c_a(i)$ of user u_a and port $c_b(j)$ of user u_b. Suppose both users can rendezvous on port c_g after user u_b starts t time slots; then we deduce that:

$$\begin{cases} t + \delta \mod p_a \equiv i \\ t \mod p_b \equiv j \end{cases}$$

According to the Chinese Remainder Theorem (see Chap. 9, Theorem 9.1), such value t must exist which satisfies both equations and $t \leq p_a p_b$. Therefore, two users can always achieve rendezvous no matter when they start.

However, if both users choose the same prime number, they may never rendezvous if they happen to miss the common available port. However, the probability of such a failure is small.

15.4 Chapter Summary

In this chapter, we study the oblivious blind rendezvous (OBR) problem for two anonymous users that are indistinguishable from each other.

In the beginning, we show an impossibility result that no deterministic algorithm can tackle the OBR-2 problem even when the users start the rendezvous process at the same time (i.e. synchronous users). Then, we propose a randomized distributed algorithm called Stay or Random Selection (SRS) for a special situation in which all ports are available for the users, which performs better than randomly accessing all ports. Finally, we present several randomized algorithms for port-asymmetric users on the basis of a random picking strategy.

When all N ports are available, two anonymous users adopting the random selection method have expected time to rendezvous (*ETTR*) in N time slots. We prove that the optimal strategy when two users can run asymmetric algorithms, i.e. different strategies, has $ETTR = \frac{N+1}{2}$ time slots, where one user accesses a fixed port and the other accesses the ports according to a random permutation of the N ports. The SRS algorithm combines both ideas: the user accesses a fixed port for N time slots with probability p or accesses the ports according to a random permutation of the N ports (with probability $1 - p$).

Although the description of SRS is simple, it is difficult to compute the appropriate p value that minimizes the expected time to rendezvous. In the chapter, we show the complicated analyses for both synchronous and asynchronous situations:

(1) For two synchronous users, the $ETTR$ is derived in Theorem 15.2 and the optimal value of p can be derived numerically. When $N \to \infty$, $p \approx 0.2475$ and $ETTR \approx 0.8289N$, which matches the state-of-the-art result [2];

(2) For two asynchronous users, the $ETTR$ is derived in Theorem 15.3. Some detailed parameters are listed in Table 15.1 and when $N \to \infty$, $p \approx 0.200$, $ETTR = 0.9111N$.

Therefore, the SRS algorithm works better than random selection, which is an elegant and surprising result. However, we cannot claim that SRS is the optimal algorithm and one future direction is to explore the optimal algorithm when two users should run a symmetric strategy. Moreover, when not all ports are available for the users, which should be more practical, we need to design efficient randomized distributed algorithms that have a good performance in the future.

For the port-asymmetric rendezvous setting, the random picking algorithm can achieve rendezvous in $ETTR = \frac{|C_a||C_b|}{|C_g|}$ time slots, where C_a, C_b represent the number of available ports of the two users, while C_g denotes the number of common available ports. We also present another algorithm called the Random Prime Selection and Sequential Accessing Algorithm, which has good performance and the failure probability of no rendezvous within $p_a p_b$ time slots is very low, where p_a, p_b are two chosen prime numbers in the algorithm.

References

1. Alpern, S., & Pikounis, M. (2000). The telephone coordination game. *Game Theory Application*, *5*, 1–10.
2. Anderson, E. J., & Weber, R. R. (1990). The rendezvous problem on discrete locations. *Journal of Applied Probability*, *28*, 839–851.
3. Crawford, V. P., & Haller, H. (1990). Learning how to cooperate: Optimal play in repeated coordination game. *Econometrica*, *58*(3), 571–596.
4. Fan, J. (2009). Symmetric rendezvous problem with overlooking. Ph.D. thesis, University of Cambridge.
5. Weber, R. R. (2006). The optimal strategy for symmetric rendezvous search on three locations. arXiv:0906.5447v1.
6. Weber, R. (2009). The Anderson-Weber strategy is not optimal for symmetric rendezvous search on K_4. arXiv:0912.0670.
7. Weber, R. R. (2012). Optimal symmetric rendezvous search on three locations. *Mathematics of Operations Research*, *37*, 111–122.