Chapter 14
Fully Distributed Rendezvous Algorithm for Non-anonymous Users

Abstract In Chap. 13, we present efficient distributed algorithms for both synchronous and asynchronous users that are non-anonymous. These algorithms utilize global information such as the number of the external ports N and the number of users M (or the maximum value for the users' identifier (ID)). In practical large scale networks, it is difficult for the users to know these information beforehand. For example, in cognitive radio networks, no general standard exists dividing the total licensed spectrum into N channels, such as the IEEE 802.11 standard which only concerns frequencies ranging 470–710 MHz [1], and so it is impractical for the users to know the value of N. Moreover, all users are physically dispersed in the system and they may join or leave freely, and hence they cannot know the number of users in advance as there is no central controller. Therefore, it is desirable to design a *fully distributed algorithm* where only the users' local information would be utilized. Actually, in a general distributed system, this kind of local information is limited to the user's ID and the number of the user's available ports since there exists no global labels for the ports. In Sect. 14.1, we present the first fully distributed algorithm called the Conversion Based Hopping (CBH) algorithm, which guarantees oblivious blind rendezvous in a short time. The correctness and complexity are analyzed in Sect. 14.2. We summarize the chapter in Sect. 14.3.

14.1 Conversion Based Hopping Algorithm

The SCH algorithm in the preceding chapter cannot work for two asynchronous users because the synchronous check stage could not work when the users start at different time slots. However, we can use the intuitive idea of the hop stage of the SCH algorithm to design distributed algorithms for two asynchronous users. Moreover, the SCH algorithm assumes each user has an estimation of N but the proposed algorithm in this chapter (we called *Conversion Based Hopping Algorithm*, or CBH for short) only uses the user's local information: the ID and the number of available ports.

Suppose the user's ID is I and the available port set is C. The CBH algorithm is described in Algorithm 14.1. With local input (I, C), Algorithm 14.1 finds the smallest prime number $p \geq \max\{k, 3\}$ where $k = |C|$ and invokes ID Conversion $(I, p - 1)$ to get the results d. The ID Conversion is described in Algorithm 13.1.

Z. Gu et al., *Rendezvous in Distributed Systems*,
DOI 10.1007/978-981-10-3680-4_14

Algorithm 14.1 Conversion Based Hopping Algorithm

1: *Input:* I, C;
2: $k := |C|$;
3: Find the smallest prime numbers $p \geq \max\{k, 3\}$;
4: $l := \lfloor log_{p-1} I \rfloor$;
5: Invoke ID Conversion $(I, p-1)$ and the output is d;
6: **if** $(l+2) \mod 2 = 0$ **then**
7: $\quad l_p := l+2$; $D := \{0, d_0+1, d_1+1, \ldots, d_l+1\}$
8: **else**
9: $\quad l_p := l+3$, $D := \{0, 1, d_0+1, d_1+1, \ldots, d_l+1\}$
10: **end if**
11: $T := 2l_p \cdot p^2$, $FL := 2l_p \cdot p$, $SL = 2p$;
12: **while** Not rendezvous **do**
13: $\quad t' := t \mod T$;
14: $\quad x := \lfloor t'/FL \rfloor, x' = t' \mod FL$;
15: $\quad y_1 := \lfloor x'/SL \rfloor, y_2 = x' \mod SL$;
16: $\quad z := x + D(y_1) \cdot y_2 \mod p+1$;
17: $\quad z' := (z-1) \mod k+1$, access port $c(z') \in C$;
18: $\quad t = t+1$;
19: **end while**

Then we construct the array D containing l_p numbers as in Lines 6–10, where l_p is defined to be an even number, which is different from Algorithm 13.2. Following the preprocessing, Algorithm 14.1 generates a sequence of length $T = 2l_p \cdot p^2$ as in Lines 13–16. This sequence consists of p frames of equal length $FL = 2l_p \cdot p$, where each frame contains l_p segments of length $SL = 2p$. In Line 17 of the algorithm, the sequence is mapped from $[1, p]$ to $[1, k]$ and the corresponding port is accessed by the user.

We illustrate the construction of the sequence in Fig. 14.1. It consists of p frames:

$$\{F_0, F_1, \ldots, F_{p-1}\} \tag{14.1}$$

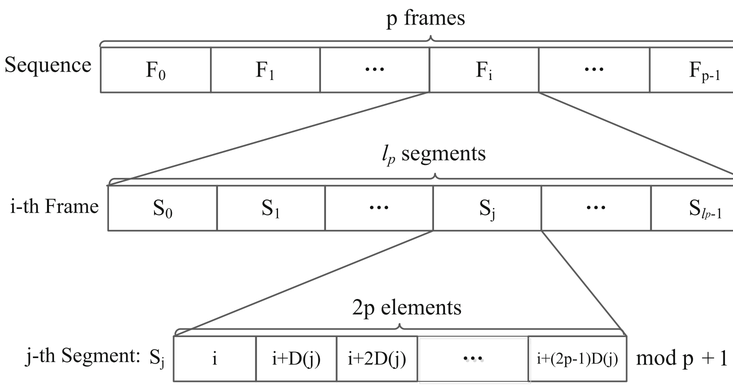

Fig. 14.1 The construction of the $T = 2l_p \cdot p^2$ sequence in CBH (Algorithm 14.1)

and each frame has l_p segments:

$$\{S_0, S_1, \ldots, S_{l_p-1}\} \tag{14.2}$$

The way to generate segment S_j of frame F_i is to construct $2p$ numbers, starting with i and the hopping step is $D(j)$; then the k-th number is constructed as such:

$$(i + kD(j)) \mod p + 1 \tag{14.3}$$

Each segment contains $2p$ numbers and this is to eliminate the asynchronous situation through doubling the length, which is similar to the method of transforming time slots into slot-aligned scenario.

There are two intuitive ideas in designing the CBH algorithm. The first one comes from the SCH algorithm when the corresponding prime numbers of the two users in Line 3 satisfy $p_a \neq p_b$, and each user repeating its own ports can guarantee rendezvous. When $p_a = p_b$, distinct IDs have different representations through the ID conversion, thus accessing the ports with these hopping steps may assure rendezvous. The proposed CBH algorithm combines these two principles and it has good performance as analyzed in the next section.

14.2 Correctness and Complexity

Assume two asynchronous users (u_a and u_b) run Algorithm 14.1 with inputs (I_a, C_a) and (I_b, C_b) where $C_a \bigcap C_b \neq \emptyset$, $I_a \neq I_b$ ($I_a, I_b \in [1, M]$). Without loss of generality, suppose user u_b is $\delta \geq 0$ time slots later. Denote the variables used for two users in Algorithm 14.1 as:

$$\begin{cases} (k_a, p_a, l_a, l_{p_a}, D_a, T_a, FL_a, SL_a, t_a) \\ (k_b, p_b, l_b, l_{p_b}, D_b, T_b, FL_b, SL_b, t_b) \end{cases}$$

Since $C_a \bigcap C_b \neq \emptyset$, there exists a port with global label $u' \in C_a \bigcap C_b$ and there exist $1 \leq i \leq k_a$, $1 \leq j \leq k_b$ such that

$$\begin{cases} c_a(i) = u' \\ c_b(j) = u' \end{cases}$$

We derive the time complexity to achieve rendezvous based on the following three situations:

(1) $p_a = p_b = p$ and $l_{p_a} = l_{p_b} = l_p$;
(2) $p_a = p_b = p$ but $l_{p_a} \neq l_{p_b}$;
(3) $p_a \neq p_b$;

Lemma 14.1 *If $p_a = p_b = p$ and $l_{p_a} = l_{p_b} = l_p$, rendezvous between users u_a and u_b can be guaranteed in $T = 2l_p \cdot p^2$ time slots.*

Proof If $0 \le \delta \mod 2p < p$, there exists $x^* \ge 0, 0 \le y_1^* < l_p, 0 \le y_2^* < p$ such that:

$$\delta = x^* \cdot (2pl_p) + y_1^* \cdot (2p) + y_2^* \tag{14.4}$$

Suppose users u_a and u_b can achieve rendezvous on port u' at time t_a, t_b respectively, and there exists $x(a), x(b) > 0, 0 \le y_1(a), y_1(b) < l_p, 0 \le y_2(a) < 2p, 0 \le y_2(b) < p$ such that:

$$t_a = x(a) \cdot (2pl_p) + y_1(a) \cdot (2p) + y_2(a) \tag{14.5}$$
$$t_b = x(b) \cdot (2pl_p) + y_1(b) \cdot (2p) + y_2(b) \tag{14.6}$$

From Lines 13–16 of Algorithm 14.1, the corresponding z values for two users could be generated to be i, j, thus:

$$x(a) + D_a(y_1(a)) \cdot y_2(a) \mod p + 1 = i \tag{14.7}$$
$$x(b) + D_b(y_1(b)) \cdot y_2(b) \mod p + 1 = j \tag{14.8}$$

Since user u_b is δ time slots later, we rewrite it as:

$$t_a = t_b + \delta \tag{14.9}$$

Plug Eqs. (14.4)–(14.6), we can get:

$$\begin{aligned} [x(a) - x(b) - x^*] \cdot (2pl) + [y_1(a) - y_1(b) - y_1^*] \cdot (2p) \\ +[y_2(a) - y_2(b) - y_2^*] = 0 \end{aligned} \tag{14.10}$$

Since $y_2(b) \in [0, p)$, $y_2(a) - y_2(b) - y_2^* = 0$. Combining this with Eqs. (14.7)–(14.8), we can derive:

$$\begin{aligned} [D_a(y_1(a)) - D_b(y_1(b))] \cdot y_2(b) + D_a(y_1(a)) \cdot y_2^* = \\ i - x(a) - (j - x(b)) \mod p \end{aligned} \tag{14.11}$$

If we can find values $y_1(a), y_1(b)$ satisfying:

$$\begin{cases} D(y_1(a)) - D(y_1(b)) \neq 0 \\ y_1(a) - y_1(b) - y_1^* \mod l_p = 0 \end{cases}$$

Equation (14.11) can be solved under the constraint Eq. (14.10). We compute $y_1(a), y_1(b)$ as follows:

$$\begin{cases} y_1(a) = y_1(b) = k & \text{If } y_1^* = 0 \\ y_1(a) = y_1^*, y_1(b) = 0 & \text{If } 0 < y_1^* \le l_p - 1 \end{cases} \tag{14.12}$$

If $y_1^* = 0$, there exist $1 \le k \le l_p - 1$ such that $D_a(k) \ne D_b(k)$ from ID conversion. If $0 < y_1^* \le l_p - 1$, $D_a(y_1(a)) - D_b(y_1(b)) = D_a(y_1^*) > 0$. Thus such $y_1(a)$, $y_1(b)$ exist and $y_1(a) - y_1(b) - y_1^* = 0$.

Since $D_a(y_1(a)) - D_b(y_1(b)) \ne 0$, $y_2(b)$ can be computed from Eq. (14.11) as follows. We plug in equation:

$$x(a) - x(b) = x^* \tag{14.13}$$

from the constraint Eq. (14.10). Then, we compute:

$$x(b) = j - 1 - D_b(y_1(b)) \cdot y_2(b) \mod p \tag{14.14}$$

and thus $x(b) \in [0, p)$. So the time to rendezvous is:

$$TTR = t_b = x(b) \cdot (2pl_p) + y_1(b) \cdot (2p) + y_2(b) \tag{14.15}$$

and it is bounded by $2l_p \cdot p^2$.

For example, users u_a and u_b have inputs $I_a = 5$, $|C_a| = 4$, $I_b = 20$, $|C_b| = 5$ and $c_a(2) = c_b(4)$ is their only common available port. Thus $p_a = p_b = 5$, $l_a = l_b = 4$ and $D_a = \{0, 1, 2, 2\}$, $D_b = \{0, 2, 2, 0\}$.

Let $\delta = 2014$ and it can be rewritten as:

$$\delta = 50 \cdot 40 + 1 \cdot 10 + 4 \tag{14.16}$$

Thus we compute the values according to Eq. (14.4) as:

$$x^* = 50,\ y_1^* = 1,\ y_2^* = 4 \tag{14.17}$$

Since $y_1^* = 1$, from Eq. (14.12), we know:

$$\begin{aligned} y_1(a) = y_1^* &= 1 \\ y_1(b) &= 0 \\ x(a) - x(b) = x^* &= 50 \end{aligned}$$

From Eq. (14.11), $y_2(b) = 4$ and $x(b) = 3$. Thus

$$\begin{aligned} t_b &= 3 * 40 + 4 = 124 \\ t_a &= t_b + \delta = 2138 \end{aligned}$$

We can check that user u_a accesses port $c_a(2)$ and u_b accesses port $c_b(4)$ at the same time.

If $p \leq \delta \mod 2p < 2p$, the *TTR* value is also bounded by $2l_p \cdot p^2$ time slots using the same technique above. We omit the details and the readers may deduce this situation. Therefore, the lemma holds.

Lemma 14.2 *If $p_a = p_b = p$ but $l_{p_a} \neq l_{p_b}$, rendezvous between users u_a and u_b can be guaranteed in $T = 2\min\{l_{p_a}, l_{p_b}\} \cdot p^2$ time slots.*

Proof If $0 \leq \delta \mod 2p < p$, there exists $x^* \geq 0, 0 \leq y_1^* < l_{p_a}, 0 \leq y_2^* < p$ such that:

$$\delta = x^* \cdot (2pl_{p_a}) + y_1^* \cdot (2p) + y_2^* \tag{14.18}$$

Suppose two users can rendezvous on port u' at time t_a, t_b respectively, we have:

$$t_a = x(a) \cdot (2pl_{p_a}) + y_1(a) \cdot (2p) + y_2(a)$$
$$t_b = x(b) \cdot (2pl_{p_b}) + y_2(b) \cdot (2p) + y_2(b)$$

where $x(a), x(b) > 0, 0 \leq y_1(a) < l_{p_a}, 0 \leq y_1(b) < l_{p_b}, 0 \leq y_2(a) < 2p, 0 \leq y_2(b) < p$. Combining these with $t_a = t_b + \delta$ to derive:

$$\begin{aligned}[l_{p_a}x(a) - l_{p_b}x(b) - l_{p_a}x^* + y_1(a) - y_1(b) - y_1^*] \cdot 2p \\ +y_2(a) - y_2(b) - y_2^* = 0\end{aligned}$$

Similarly, we have:

$$\begin{cases} l_{p_a} \cdot x(a) - l_{p_b} \cdot x(b) - l_{p_a} \cdot x^* + y_1(a) - y_1(b) - y_1^* = 0 \\ y_2(a) - y_2(b) - y_2^* = 0 \end{cases} \tag{14.19}$$

We can also formulate Eqs. (14.7)–(14.8). If $l_{p_a} > l_{p_b}$, we let:

$$\begin{cases} y_1(a) = 0 \\ y_1(b) = k \neq 0 \end{cases}$$

From Eq. (14.7), we can derive:

$$x(a) = (i-1) \mod p \tag{14.20}$$

Plugging this into Eq. (14.19), we have:

$$l_{p_b}x(b) = [l_{p_a}x(a) - l_{p_a}x^* - y_1(b) - y_1^*] \mod p \tag{14.21}$$

Since l_{p_b} is an even number, $x(b) \in [0, p)$ can be computed obviously. Then from Eq. (14.8), $y_2(b) \in [0, p)$ can be derived. Therefore, the TTR value is:

$$t_b = x(b) \cdot (2pl_{p_b}) + y_1(b) \cdot (2p) + y_2(b) \leq 2p^2 l_{p_b} \tag{14.22}$$

If $l_{p_a} < l_{p_b}$, we can bound the time to rendezvous as:

$$TTR = t_a - \delta = (x(a) - x^*) \cdot (2pl_{p_a}) + (y_1(a) - y_1^*) \cdot (2p) + (y_2(a) - y_2^*) \leq 2p^2 l_{p_a} \tag{14.23}$$

Thus, $MTTR \leq 2\min\{l_{p_a}, l_{p_b}\}p^2$.

If $p \leq \delta \mod 2p < 2p$, the *TTR* value is also bounded by:

$$T = 2\min\{l_{p_a}, l_{p_b}\} \cdot p^2 \tag{14.24}$$

time slots using the same technique above. Thus the lemma holds.

Lemma 14.3 *If $p_a \neq p_b$, rendezvous between users u_a and u_b can be guaranteed in $T = 2l_p \cdot p^2$ time slots, where $p = \max\{p_a, p_b\}$ and l_p is the corresponding value from $\{l_{p_a}, l_{p_b}\}$.*

Proof This lemma can be concluded similarly. Suppose $p_a < p_b$, we can derive the following equations:

$$x(a) + D_a(y_1(a)) \cdot y_2(a) \mod p_a + 1 = i$$
$$x(b) + D_b(y_1(b)) \cdot y_2(b) \mod p_b + 1 = j$$

Let $y_1(b) = 0$, then:

$$x(b) = (j - 1) \mod p_b \tag{14.25}$$

and $y_2(b) \in [0, 2p_b)$. Suppose:

$$x(a) = i' \mod p_a \tag{14.26}$$

and $y_1(a) \neq 0$, then $y_2(a)$ exists. Since $t_a = t_b + \delta$ and we know:

$$t_a = x(a) \cdot (2p_a l_{p_a}) + y_1(a) \cdot (2p_a) + y_2(a)$$
$$t_b = x(b) \cdot (2p_b l_{p_b}) + y_2(b) \cdot (2p_b) + y_2(b)$$

We can find value:

$$x(a) = i' + v(a)p_a \tag{14.27}$$

satisfying $\delta_b(v_b) + \delta - \delta_a(v_a) \in [2p_a - 2p_b, T_a)$ where $T_a = 2p_a^2 l_{p_a}$ is define as above, where:

$$\delta_b(v_b) = (2p_b l_{p_b}) \cdot (j - 1 + v(b)p_b)$$
$$\delta_a(v_a) = (2p_a l_{p_a}) \cdot x(a)$$

Obviously, we can compute:

$$\delta_b(0) \mod T_a \geq 2p_a - 2p_b \tag{14.28}$$

We let:

$$\begin{aligned} v(b) &= 0 \\ v(a) &= \lfloor (\delta_b(0) + \delta)/T_a \rfloor \\ i' &= \lfloor (\delta_b(0) + \delta - v(a)T_a)/FL_a \rfloor \end{aligned}$$

where $FL = 2p_a l_{p_a}$; then variables $y_1(a)$, $y_2(a)$, $y_2(b)$ can be determined. Thus, the time to rendezvous can be computed as:

$$TTR = t_b = x(b) \cdot (2p_b l_{p_b}) + y_1(b) \cdot (2p_b) + y_2(b) \leq 2p_b^2 l_{p_b} \tag{14.29}$$

If $p_a > p_b$, we can derive the time complexity using a similar technique:

$$TTR \leq 2p_a^2 l_{p_a} \tag{14.30}$$

Therefore, the lemma holds.

Combining Lemmas 14.1–14.3, we can conclude:

Theorem 14.1 *Two users running the CBH algorithm (Algorithm 14.1) can achieve rendezvous in $MTTR = 2l_p \cdot p^2$ time slots where $p = \max\{p_a, p_b\}$ and l_p is the corresponding value from $\{l_{p_a}, l_{p_b}\}$.*

This theorem reveals that: the CBH algorithm can guarantee oblivious blind rendezvous between two users in a short time and it is comparable to the lower bound in Theorem 13.5 for most cases. More precisely, $MTTR = 2l_p \cdot p^2 = O(k^2)$ time slots if l_p is a constant, which implies the corresponding ID is a polynomial function of p, where $k = \max\{k_a, k_b\}$.

For example, if $k_a > k_b$ (which implies $p_a \geq p_b$), and I_a is bounded by $I_a \leq p_a^c$ where c can be an arbitrary large constant, $MTTR = 2l_{p_a} \cdot p_a^2 = O(k_a^2)$. If $k_b = \Theta(k_a)$ and $k_g = o(k_a)$, the $MTTR$ value is comparable with the lower bound in Theorem 13.5 (see Chap. 13.3).

Remark 14.1 In Lemma 14.3, the $MTTR$ value can be bounded by $2l_p p^2$ time slots where $p = \min\{p_a, p_b\}$ for most cases: when $p_a < p_b$, if

$$(2p_a l_a j) \mod T_b \geq 2(p_b - p_a) \tag{14.31}$$

or

$$T_a \mod T_b = \Omega(p_b) \tag{14.32}$$

the $MTTR$ value could be very small.

14.3 Chapter Summary

In this chapter, we present the first fully distributed rendezvous algorithm for two non-anonymous users, called Conversion Based Hopping (CBH). The CBH algorithm only utilizes the user's local information: the ID and the number of available ports and it is independent of the global parameters: the number of all ports N and the maximum value for the users' ID M.

The CBH algorithm combines the intuitive idea of the MLS algorithm (in Chap. 9) and the SCH/MSH algorithm (in Chap. 13): the user's ID is first scaled (converted) to a new number given the base value that is related to the number of available ports; then the algorithm constructs hopping sequences by different hopping steps. The CBH algorithm guarantees rendezvous in $O((\max\{|C_a|, |C_b|\})^2)$ time slots under most circumstances where C_a, C_b represent the sets of two users' available ports. When the number of available ports is small, the CBH algorithm outperforms some state-of-the-art global sequence based rendezvous algorithms.

The CBH algorithm has many advantages when compared with traditional non-oblivious blind rendezvous algorithms:

(1) The CBH algorithm uses very little information. Only the user's ID and the number of available ports are used in designing the CBH algorithm. It does not require global information, such as the number of ports, the maximum ID value, the labels of the ports. Some traditional non-oblivious blind rendezvous algorithm may not utilize the user's ID either, but they may need the value of all ports, or the labels of these ports.

(2) The CBH algorithm is also suitable for non-oblivious setting, where the external ports have global labels. Compared with the state-of-the-art rendezvous algorithms, the CBH algorithm has good performance when the users' number of available ports is small.

Reference

1. A. B. Flores, R. E. Guerra, and E. W. Kightly. IEEE 802.11af: A Standard for TV White Space Spectrum Sharing. *IEEE Communications Magazine*, 2013.