# RANDOMIZED RENDEZ-VOUS WITH LIMITED MEMORY

Evangelos Kranakis Danny Krizanc Pat Morin

September 27, 2007

Abstract. We present a tradeoff between the expected time for two identical agents to rendez-vous on a synchronous, anonymous, oriented ring and the memory requirements of the agents. In particular, we show there exists a $2t$ state agent, which can achieve rendez-vous on an $n$ node ring in expected time $O(n^2/2^t + 2^t)$ and that any $t/2$ state agent requires expected time $\Omega(n^2/2^t)$. As a corollary we observe that $\Theta(\log \log n)$ bits of memory are necessary and sufficient to achieve rendez-vous in linear time.

## 1 Introduction

The problem of *rendez-vous* (the gathering of agents widely dispersed in some domain at a common place and time) has been studied under many guises and in many settings [2, 15, 4, 5, 7, 6, 8, 10, 9, 14, 12, 18, 20, 21, 22]. (See Reference [13] for a survey of recent results.) In this paper we consider the problem of autonomous mobile software agents gathering in a distributed network. This is a fundamental operation useful in such applications as web-crawling, distributed search, meeting scheduling, etc. In particular, we study the problem of two identical agents attempting to rendez-vous on a synchronous anonymous ring.

We consider the standard model of a synchronous anonymous oriented $n$-node ring [19]. The nodes are assumed to have no identities, the computation proceeds in synchronous steps and the edges of the ring are labelled `clockwise` and `counterclockwise` in a consistent fashion. We model the agents as identical probabilistic finite automata $A = \langle S, \delta, s_0 \rangle$ where $S$ is the set of states of the automata including $s_0$ the initial state and the special state `halt`, and $\delta : S \times C \times P \to S \times M$ where $C = \{H, T\}$ represents a random coin flip, $P = \{\texttt{present}, \texttt{notpresent}\}$ represents a predicate indicating the presence of the other agent at a node, and $M = \{-1, 0, +1\}$ represents the potential moves the agent may make, $+1$ representing clockwise, $-1$ counterclockwise and $0$ stay at the current node. During each synchronous step, depending upon its current state, the answer to a query for the presence of the other agent, and the value of an independent random coin flip with probability of `heads` equal to $1/2$, the agent uses $\delta$ in order to change its state and either move across the edge labelled `clockwise`, move across the edge labelled `counterclockwise` or stay at the current node. We assume that the agent halts once it detects the presence of the other agent at a node. Rendezvous occurs when both agents halt on the same node. The complexity measures we are interested in are the expected time (the number of synchronous steps) to rendez-vous (where the expectation is taken over all sequences of coin flips of the two agents) and the size ($|S|$) or memory requirement ($\log_2 |S|$) of the agents.

A number of researchers have observed that using random walks one can design $O(1)$ state agents that will rendez-vous in polynomial number steps on any network [3]. For the ring the expected time for two random walks to meet is easily shown to be $O(n^2)$. (See Reference [11] for an example proof of this fact.) This expected time bound can be improved by considering the following strategy. Repeat the following until rendez-vous is achieved: flip a fair coin and walk $n/2$ steps clockwise if the result is heads, $n/2$ steps counterclockwise if the result is tails. If the two agents choose different directions (which they do with probability 1/2) then they will rendez-vous (at least in the case where they start at an even distance apart). It is easy to see that the expected time until rendez-vous in this case is $O(n)$. Alpern refers to this strategy as Coin Half Tour and studies it in detail [1]. Note that the agents are required to count up to $n/2$ and thus require $\Omega(n)$ states or $\Omega(\log n)$ bits of memory to perform this algorithm. The main result of this paper is that this memory requirement can be reduced to $O(\log\log n)$ bits while still achieving rendezvous in $O(n)$ expected time, and this is optimal.

Below we show a tradeoff between the size of the agents and the time required for them to rendez-vous. We prove there exists a $2t$ state algorithm, which can achieve rendez-vous on an $n$ node ring in expected time $O(n^2/2^t + 2^t)$ and that any $t/2$ state algorithm requires expected time $\Omega(n^2/2^t)$. As a corollary we observe that $\Theta(\log\log n)$ bits of memory are necessary and sufficient to achieve rendez-vous in linear time. Section 2 contains some preliminary results, section 3 our upper bound and section 4 the lower bound.

## 2 Preliminaries

### 2.1 Martingales, Stopping Times, and Wald's Equations

In this section, we review some results on stochastic processes that are used several times in our proofs. The material in this section is based on the presentation in Ross' textbook [17, Chapter 6]. Let $X = X_1, X_2, X_3, \dots$ be a sequence of random variables and let $Q = Q_1, Q_2, Q_3 \dots$ be a sequence of random variables in which $Q_i$ is a function of $X_1, \dots, X_i$. Then we say that $Q$ *is a martingale with respect to* $X$ if, for all $i$, $\mathrm{E}[|Q_i|] < \infty$ and $\mathrm{E}[Q_{i+1} \mid X_1, \dots, X_i] = Q_i$.

A positive integer-valued random variable $T$ is a *stopping time* for the sequence $X_1, X_2, X_3, \dots$ if the event $T = i$ is determined by the values $X_1, \dots, X_i$. In particular, the event $T = i$ is independent of the values $X_{i+1}, X_{i+2}, \dots$. Some of our results rely on the Martingale Stopping Theorem:

**Theorem 1** (Martingale Stopping Theorem)**.** *If $Q_1, Q_2, Q_3, \dots$ is a martingale with respect to $X_1, X_2, X_3, \dots$ and $T$ is a stopping time for $X_1, X_2, X_3, \dots$ then*

$$\mathrm{E}[Q_T] = \mathrm{E}[Q_1]$$

*provided that at least one of the following holds:*

1. *$Q_i$ is uniformly bounded for all $i \le T$,*
2. *$T$ is bounded, or*
3. *$\mathrm{E}[T] \le \infty$ and there exists an $M < \infty$ such that*

$$\mathrm{E}\left[|Q_{i+1} - Q_i| \mid X_1, \dots, X_i\right] < M \ .$$

If $X_1, X_2, X_3, \dots$ is a sequence of i.i.d. random variables with expected value $\mathrm{E}[X] < \infty$ and variance $\mathrm{var}(X) < \infty$ then by applying Theorem 1 on the sequence $Q_i = \sum_{j=1}^{i}(X_j - \mathrm{E}[X])$ we obtain

Wald's Equation:

$$\mathrm{E}\left[\sum_{i=1}^{T} X_i\right] = \mathrm{E}[T] \cdot \mathrm{E}[X] \quad (1)$$

whenever $T$ is a stopping time for $X_1, X_2, X_3, \ldots$ Similarly, we can derive a version of Wald's Equation for the variance by considering the martingale $Q_i = \left(\sum_{j=1}^{i}(X_j - \mathrm{E}[X]\right)^2 - i \cdot \mathrm{var}(X)$ to obtain

$$\mathrm{var}\left(\sum_{i=1}^{T} X_i\right) = \mathrm{E}\left[\left(\sum_{i=1}^{T}(X_i - \mathrm{E}[X_i])\right)^2\right] = \mathrm{E}[T] \cdot \mathrm{var}(X) \ . \quad (2)$$

### 2.2 A Lemma on Random Walks

Let $X_1, X_2, X_3, \ldots \in \{-1, +1\}$ be independent random variables with

$$\Pr\{X_i = -1\} = \Pr\{X_i = +1\} = 1/2$$

and let $S_i = \sum_{j=1}^{i} X_j$. The sequence $S_1, S_2, S_3, \ldots$ is a *simple random walk* on the line, where each $X_i$ represents a step to the left ($X_i = -1$) or a step to the right ($X_i = +1$). Define the *hitting time* $h_m$ as

$$h_m = \min\{i : |S_i| = m\} \ \ ,$$

which is the number of steps in a simple random walk before it travels a distance of $m$ from its starting location. The following result is well-known (see, e.g., Reference [16]):

**Lemma 1.** $\mathrm{E}[h_m] = m^2$.

Applying Markov's Inequality with Lemma 1 yields the following useful corollary

**Corollary 1.** $\Pr\{\max\{|S_i| : i \in \{1, \ldots, 2m^2\}\} \geq m\} \geq 1/2$ .

In other words, Corollary 1 says that, at least half the time, at some point during the first $2m^2$ steps of a simple random walk, the walk is at distance $m$ from its starting location.

Let $Y_1, \ldots, Y_m$ be i.i.d. non-negative random variables with finite expectation $r = \mathrm{E}[Y_i]$, independent of $X_1, \ldots, X_m$, and with the property that

$$\Pr\{Y_i \geq \alpha r\} \geq 1/2$$

for some constant $\alpha > 0$. The following lemma considers a modified random walk in which the $i$th step is of length $Y_i$:

**Lemma 2.** *Let $X_1, \ldots, X_m$ and $Y_1, \ldots, Y_m$ be defined as above. Then there exists constants $\beta, \kappa > 0$ such that*

$$\Pr\left\{\max\left\{\left|\sum_{i=1}^{m'} X_i Y_i\right| : m' \in \{1, \ldots, m\}\right\} \geq \beta r\sqrt{m}\right\} \geq \kappa \ .$$

*Proof.* We will define 3 events $E_1, E_2, E_3$ such that $\Pr\{E_1 \cap E_2 \cap E_3\} \geq 1/8$ and, if $E_1$, $E_2$, and $E_3$ all occur, then there exists a value $m' \in \{1, \ldots, m\}$ such that $\left|\sum_{i=1}^{m'} X_i Y_i\right| \geq \alpha r \sqrt{m}/2^{3/2}$. This will prove the lemma for $\kappa = 1/8$ and $\beta = \alpha/2^{3/2}$.

Let $E_1$ be the event that there exists a value $m' \in \{1, \ldots, m\}$ such that

$$\left|\sum_{i=1}^{m'} X_i\right| \geq \sqrt{m/2} \ .$$

By Corollary 1, $\Pr\{E_1\} \geq 1/2$. Assume $E_1$ occurs and, without loss of generality, assume $\sum_{i=1}^{m'} X_i > 0$.

Let $I^+ = \{i \in \{1, \ldots, m'\} : X_i = +1\}$ and $I^- = \{1, \ldots, m'\} \setminus I^+$. We further partition $I^+$ into two sets $I_1^+$ and $I_2^+$ where $I_1^+$ contains the smallest $|I^-|$ elements of $I^+$ and $I_2^+$ contains the remaining elements. Note that, with these definitions, $|I_1^+| = |I^-|$ and that $|I_2^+| = \sum_{i=1}^{m'} X_i$. Let $E_2$ be the event that

$$\sum_{i \in I_1^+} X_i Y_i + \sum_{i \in I^-} X_i Y_i \geq 0$$

which is equivalent to $\sum_{i \in I_1^+} Y_i \geq \sum_{i \in I^-} Y_i$ and observe that, by symmetry, $\Pr\{E_2 | E_1\} \geq 1/2$.

Finally, let $E_3$ be the event

$$\sum_{i \in I_2^+} X_i Y_i \geq \alpha r |I_2^+|/2$$

To bound $\Pr\{E_3 | E_1 \cap E_2\}$, let $T = \left|\{i \in I_2^+ : Y_i \geq \alpha r\}\right|$ and observe that $T \geq |I_2^+|/2$ implies $E_3$. Now, $T$ is a binomial$(|I_2^+|, p)$ random variable for $p \geq 1/2$ so its median value is at least $p|I_2^+| \geq |I_2^+|/2$ and therefore $\Pr\{E_3 | E_1 \cap E_2\} \geq \Pr\{T \geq |I_2^+|/2\} \geq 1/2$.

We have just shown that $\Pr\{E_1 \cap E_2 \cap E_3\} \geq 1/8$. To complete the proof we observe that, if $E_1$, $E_2$ and $E_3$ occur then

$$\begin{aligned} \sum_{i=1}^{m'} X_i Y_i &= \sum_{i \in I_1^+} X_i Y_i + \sum_{i \in I^-} X_i Y_i + \sum_{i \in I_2^+} X_i Y_i \\ &\geq \sum_{i \in I_2^+} X_i Y_i \\ &\geq \alpha r |I_2^+|/2 \\ &\geq \alpha r \sqrt{m}/2^{3/2} \ . \end{aligned}$$

□

### 2.3 An Approximate Counter

In the previous section we have shown that, if we can generate random variables $Y_i$ that are frequently large, then we can speed up the rate at which a random walk moves away from its starting location. In this section we consider how to generate these frequently-large random variables. Consider a random variable $Y$ generated by the following algorithm:

BigRand($t$)

```
1: Y ← C ← 0
2: while C < t do
3:    Y ← Y + 1
4:    if a coin toss comes up heads then
5:       C ← C + 1
6:    else
7:       C ← 0
8: return Y
```

**Lemma 3.** *Let $Y$ be the output of Algorithm* BigRand($t$). *Then*

1. $\mathrm{E}[Y] = 2^t(2 - 1/2^{t-1})$ *and*
2. $\Pr\{Y \geq \mathrm{E}[Y]/2\} \geq 1/2$.

*Proof.* To compute the expected value of $Y$ we observe that the algorithm begins by tossing a sequence of $i-1$ heads and then either (a) returning to the initial state if the $i$th coin toss is a tail or (b) terminating if $i = 2^t$. The first case occurs with probability $1/2^i$ and the second case occurs with probability $1/2^t$. In this way, we obtain the equation

$$\mathrm{E}[Y] = \sum_{i=1}^{t} \frac{1}{2^i}\left(i + \mathrm{E}[Y]\right) + \frac{t}{2^t} \ .$$

Rearranging terms and multiplying by $2^t$, we obtain

$$\mathrm{E}[Y] = 2^t(2 - 1/2^{t-1}) \ .$$

To prove the second part of the lemma, consider the number of times the counter $C$ is reset to 0 in Line 7 of the algorithm. This number is a geometric($1/2^t$) random variable and its expected value is therefore $2^t \geq \mathrm{E}[Y]/2$. Since the number of times Line 7 executes is a lower bound on the number of times the value of $Y$ is incremented (Line 3), this completes the proof. □

## 3 The Rendez-Vous Algorithm

Consider the following algorithm used by an agent to make a random walk on a ring. The agent repeatedly performs the following steps: (1) toss a coin to determine a direction $d \in \{\texttt{clockwise}, \texttt{counterclockwise}\}$ then (2) run algorithm BigRand($t$) replacing each increment of the variable $Y$ with a step in direction $d$. By using $t$ states for a clockwise counter and $t$ states for a counterclockwise counter this algorithm can be implemented by a $2t$ state finite automata. (Or using one bit to remember the direction $d$ and $\log t$ bits to keep track of the counter $C$ in the BigRand algorithm, it can be implemented by an agent having only $1 + \log_2 t$ bits of memory.)

We call $m$ iterations of the above algorithm a *round*. Together, Lemma 2 and Lemma 3 imply that, during a round, with probability at least $\kappa$, an agent will travel a distance of at least $\beta 2^t \sqrt{m}$ from its starting location. Set

$$m = \left\lceil \frac{n^2}{\beta^2 2^{2t}} \right\rceil$$

and consider what happens when two agents $A$ and $B$ both execute this rendez-vous algorithm. During the first round of $A$'s execution, with probability at least $\kappa$, agent $A$ will have visited agent $B$'s starting location. Furthermore, with probability at least $1/2$ agent $B$ will not have moved away from $A$ when this occurs, so the paths of agents $A$ and $B$ will cross, and a rendez-vous will occur, with probability at least $\kappa/2$.

By Lemma 3, the expected number of steps taken for $A$ to execute the $i$th round is at most

$$\mathrm{E}[M_i] \leq m2^t \ .$$

The variables $M_1, M_2, \cdots$ are independent and the algorithm terminates when $A$ and $B$ rendez-vous. If we define $T$ as the round in which agents $A$ and $B$ rendez-vous then the time to rendez-vous is bounded by

$$\sum_{i=1}^{T} M_i \ .$$

Note that the event $T = j$ is independent of $M_{j+1}, M_{j+2}, \ldots$ so $T$ is a stopping time for the sequence $M_1, M_2, \ldots$ so, by Wald's Equation

$$E\left[\sum_{i=1}^{T} M_i\right] = E[T] \cdot E[M_1] \leq \frac{2}{\kappa} \cdot m2^t \ .$$

This completes the proof of our first theorem.

**Theorem 2.** *There exists a rendez-vous algorithm in which each agent has at most* $2t$ *states and whose expected rendez-vous time is* $O(n^2/2^t + 2^t)$.

## 4 The Lower Bound

Next we show that the algorithm in Section 3 is optimal.

The model of computation for the lower bound represents a rendez-vous algorithm $\mathcal{A}$ as a probablistic finite automata having $t$ states. Each vertex of the automata has two outgoing edges representing the two possible results of a coin toss and each edge $e$ is labelled with a real number $\ell(e) \in [-1, +1]$. The edge label of $e$ is represented as a step of length $|\ell(e)|$ with this step being counterclockwise if $\ell(e) < 0$ and clockwise if $\ell(e) > 0$. As before, both agents use identical automata and start in the same state. The rendez-vous process is complete once the distance between the two agents is at most 1. This model is stronger than the model used for upper bound, since the edge labels are no longer restricted to be in the discrete set $\{-1, 0, +1\}$ and the definition of a rendezvous has been slightly relaxed.

### 4.1 Well-Behaved Algorithms and Reset Times

We say that an algorithm is *well-behaved* if the directed graph of the state machine has only one strongly connected component that contains all nodes. We are particularly interested in intervals between consecutive visits to the start state, which we will call *rounds*.

**Lemma 4.** *Let* $R$ *be the number of steps during a round. Then* $\mathrm{E}[R] \leq 2^t$ *and* $\mathrm{E}[R^2] \leq c2^{2t}$.

*Proof.* For each state $v$ of $\mathcal{A}$'s automata fix a shortest path (a sequence of edges) leading from $v$ to the start state. For an automata that is currently at $v$ we say that the next step is a *success* if it traverses the first edge of this path, otherwise we say that the next step is a *failure*.

Each round can be further refined into *phases*, where every phase consists of 0 or more successes followed by either a failure or by reaching the start vertex. Let $X_i$ denote the length of the $i$th phase and note that $X_i$ is dominated[1] by a geometric(1/2) random variable $X'_i$, so $\mathrm{E}[X_i] \leq \mathrm{E}[X'_i] \leq 2$. On the other hand, if a phase lasts $t-1$ steps then the start vertex is reached. Therefore, the probability of reaching the start vertex during any particular phase is at least $1/2^{t-1}$ and the number $T$ of phases is dominated by a geometric($1/2^{t-1}$) random variable $T'$, so $\mathrm{E}[T] \leq \mathrm{E}[T'] \leq 2^{t-1}$. Therefore, by Wald's Equation

$$\mathrm{E}[R] = \mathrm{E}\left[\sum_{i=1}^{T} X_i\right] \leq \mathrm{E}\left[\sum_{i=1}^{T'} X'_i\right] = \mathrm{E}[T'] \cdot \mathrm{E}[X'_1] = 2^t \ .$$

For the second part of the lemma, we can apply Wald's Equation for the variance (2) to obtain

$$\begin{aligned}
\mathrm{E}[R^2] &= \mathrm{E}\left[\left(\sum_{i=1}^{T} X_i\right)^2\right] \\
&\leq \mathrm{E}\left[\left(\sum_{i=1}^{T'} X'_i\right)^2\right] \\
&= \mathrm{var}\left(\sum_{i=1}^{T'} X'_i\right) + (\mathrm{E}[T'] \cdot \mathrm{E}[X'_1])^2 \\
&= \mathrm{E}[T'] \cdot \mathrm{var}(X_1) + (\mathrm{E}[T'] \cdot \mathrm{E}[X'_1])^2 \\
&\leq 2^{t-1} \cdot 4 + (2^{2t-1} \cdot 8) \\
&\leq 5 \cdot 2^{2t}
\end{aligned}$$

as required. □

### 4.2 Unbiasing Algorithms

Note that $\mathrm{E}[R]$ can be expressed another way: For an edge $e$ of the state machine, let $f(e)$ be the expected number of times the edge $e$ is traversed during a round. The *reset time* of algorithm $\mathcal{A}$ is then defined as

$$\mathrm{reset}(\mathcal{A}) = \sum_e f(e) = \mathrm{E}[R] \ .$$

The *bias* of a well-behaved algorithm $\mathcal{A}$ is defined as

$$\mathrm{bias}(\mathcal{A}) = \sum_e f(e) \cdot \ell(e) \ ,$$

which is the expected sum of the edge labels encoutered during a round. We say that $\mathcal{A}$ is *unbiased* if $\mathrm{bias}(\mathcal{A}) = 0$, otherwise we say that $\mathcal{A}$ is *biased*.

[1] A random variable $X$ dominates a random variable $Y$ if $\Pr\{X > x\} \geq \Pr\{Y > x\}$ for all $x \in \mathbb{R}$.

Biased algorithms are somewhat more difficult to study. However, observe that, for any algorithm $\mathcal{A}$ we can replace every edge label $\ell(e)$ with the value $\ell(e) - x$ for any real number $x$ and obtain an equivalent algorithm in the sense that, if two agents $A$ and $B$ execute the modified algorithm following the same sequence of state transitions then $A$ and $B$ will rendez-vous after exactly the same number of steps. In particular, if we replace each edge label $\ell(e)$ with the value

$$\ell'(e) = \ell(e) - \frac{\text{bias}(\mathcal{A})}{\text{reset}(\mathcal{A})}$$

then we obtain an algorithm $\mathcal{A}'$ with $\text{bias}(\mathcal{A}') = 0$. Furthermore, since $|\text{bias}(\mathcal{A})| \leq \text{reset}(\mathcal{A})$, every edge label $\ell'(e)$ has $-2 \leq \ell'(e) \leq 2$. This gives the following relation between biased and unbiased algorithms:

**Lemma 5.** *Let $\mathcal{A}$ be a well-behaved $t$-state algorithm with expected rendez-vous time $R$. Then there exists a well-behaved unbiased $t$-state algorithm $\mathcal{A}'$ with expected rendez-vous time at most $2R$.*

### 4.3 The Lower Bound for Well-Behaved Algorithms

We now have all the tools in place to prove the lower bound for the case of well-behaved algorithms.

**Lemma 6.** *Let $\mathcal{A}$ be a well-behaved $t$-state algorithm. Then the expected rendez-vous time of $\mathcal{A}$ is $\Omega(n^2/2^{2t})$.*

*Proof.* Suppose the agents are placed at antipodal locations on an $n$ node ring, so that the distance between them is $n/2$. We will show that there exists constants $c > 0$ and $p > 0$ such that, after $cn^2/2^t$ steps, with probability at least $p$ neither agent will have travelled a distance greater than $n/4$ from their starting location. Thus, the expected rendez-vous time is at least $pcn^2/2^t = \Omega(n^2/2^t)$.

By Lemma 5 we can assume that $\mathcal{A}$ is unbiased. Consider the actions of a single agent starting at location 0. The actions of the agent proceed in rounds where, during the $i$th round, the agent takes $R_i$ steps and the sum of edge labels encountered during these steps is $X_i$. Note that the random variables $X_1, X_2, \ldots$ are i.i.d. with expectation $\text{E}[X] = 0$ and variance $\text{E}[X^2]$. Since the absolute value of $X_i$ is bounded from above by $R_i$, we have the inequalities $\text{E}[|X_i|] \leq \text{E}[R_i]$ and $\text{E}[X_i^2] \leq \text{E}[R_i^2]$.

Let $S_i = \left|\sum_{j=1}^{i} X_j\right|$, for $i = 0, 1, \ldots$ be the agent's distance from their starting location at the end of the $i$th round. Let $Q_i = S_i^2 - i\text{E}[X^2]$ and observe that the sequence $Q_1, Q_2, \ldots$ is a martingale with respect to the sequence $X_1, X_2, \ldots$ [17, Example 6.1d]. Define

$$T = \min\{i : S_i \geq m\} \ ,$$

and observe that this is equivalent to

$$T = \min\{i : Q_i \geq m^2 - i\text{E}[X^2]\} \ .$$

The random variable $T$ is a *stopping time* for the martingale $Q_1, Q_2, \ldots$ so, by the Theorem 1

$$\text{E}[Q_T] = \text{E}[Q_1] = \text{E}[(X_1)^2 - \text{E}[X^2]] = 0 \ . \tag{3}$$

However, by definition $Q_T \geq m^2 - T \cdot \text{E}[X^2]$, so

$$\text{E}[Q_T] \geq \text{E}[m^2 - T \cdot \text{E}[X^2]] = m^2 - \text{E}[T] \cdot \text{E}[X^2] \ . \tag{4}$$

Equating the right hand sides of (3) and (4) gives

$$\mathrm{E}[T] \geq \frac{m^2}{\mathrm{E}[X^2]} \ .$$

Furthermore, the expected number of steps taken by the agent during these $T$ rounds is, by Wald's Equation,

$$\mathrm{E}\left[\sum_{i=1}^{T} R_i\right] = \mathrm{E}[T] \cdot \mathrm{E}[R_1] \geq \frac{m^2 \mathrm{E}[R]}{\mathrm{E}[R^2]} \geq \frac{m^2 \mathrm{E}[R]}{c 2^{2t}} \geq \frac{m^2}{c 2^{2t}} \ ,$$

where the last two inequalities follow from Lemma 4 and the fact that $R \geq 1$. □

### 4.4 Badly-Behaved Algorithms

Finally, we consider the case where the algorithm $\mathcal{A}$ is not well-behaved. In this case, $\mathcal{A}$'s automata contains a set of *terminal components*. These are disjoint sets of vertices of the automata that are strongly connected and that have no outgoing edges (edges with source in the component and target outside the component). From each terminal component, select an arbitrary vertex and call it the *terminal start state* for that terminal component. An argument similar to that given in Lemma 4 proves:

**Lemma 7.** *The expected time to reach some terminal start state is at most* $2^t$*.*

Observe that each terminal component defines a well-behaved algorithm. Let $c$ be the number of terminal components and let $t_1, \ldots, t_c$ be the sizes of these terminal components. When two agents execute the same algorithm $\mathcal{A}$, Lemma 7 and Markov's Inequality imply that the probability that both agents reach the same terminal component after at most $2^{t+2}$ steps is at least $1/2c$. By applying Lemma 6 to each component, we can therefore lower bound the expected rendez-vous time by

$$\frac{1}{2c}\Omega(n^2/2^{t-c}) \quad \geq \quad \Omega(n^2/2^{2t}) \ ,$$

Substituting $t' = t/2$ into the above completes the proof of our second theorem:

**Theorem 3.** *Any* $t/2$*-state rendez-vous algorithm has expected rendez-vous time* $\Omega(n^2/2^t)$*.*

### 4.5 Linear Time Rendez-vous

We observe that Theorems 2 and 3 immediately imply:

**Theorem 4.** $\Theta(\log \log n)$ *bits of memory are necessary and sufficient to achieve rendez-vous in linear time on an* $n$ *node ring.*

## References


[1] S. Alpern. The rendezvous search problem. *SIAM Journal of Control and Optimization*, 33:673–683, 1995.

[2] S. Alpern and S. Gal. *The Theory of Search Games and Rendezvous*. Kluwer Academic Publishers, Norwell, Massachusetts, 2003.

[3] D. Coppersmith, P. Tetali, and P. Winkler. Collisions among random walks on a graph. *SIAM Journal of Discrete Mathematics*, 6:363–374, 1993.

[4] A. Dessmark, P. Fraigniaud, and A. Pelc. Deterministic rendezvous in graphs. In *Proc. 11th ESA*, pages 184–195, 2003.

[5] S. Dobrev, P. Flocchini, G. Prencipe, and N. Santoro. Multiple agents rendezvous in a ring in spite of a black hole. In *Proc. Symposium on Principles of Distributed Systems (OPODIS '03)*, volume 3144 of *LNCS*, pages 34–46, 2004.

[6] P. Flocchini, E. Kranakis, D. Krizanc, F. Luccio, N. Santoro, and C. Sawchuk. Mobile agent rendezvous when tokens fail. In *Proc. of 11th Sirocco*, pages 161–172, 2004.

[7] P. Flocchini, E. Kranakis, D. Krizanc, N. Santoro, and C. Sawchuk. Multiple mobile agent rendezvous in the ring. In *Proc. LATIN 2004*, volume 2976 of *LNCS*, pages 599–608, 2004.

[8] L. Gasieniec, E. Kranakis, D. Krizanc, and X. Zhang. Optimal memory rendezvous of anonymous mobile agents in a uni-directional ring. In *Proc. of 32nd SOFSEM*, pages 282–292, 2006.

[9] D. Kowalski and A. Malinowski. How to meet in an anonymous network. In *Proc. 13th Sirocco*, 2006.

[10] D. Kowalski and A. Pelc. Polynomial deterministic rendezvous in arbitrary graphs. In *Proc. 15th ISAAC*, 2004.

[11] E. Kranakis and D. Krizanc. An algorithmic theory of mobile agents. In *Symposium on Trustworthy Global Computing*, 2006.

[12] E. Kranakis, D. Krizanc, and E. Markou. Mobile agent rendezvous in a synchronous torus. In *Proc. of LATIN 2006*, pages 653–664, 2006.

[13] E. Kranakis, D. Krizanc, and S. Rajasbaum. Mobile agent rendezvous: A survey. In *Proc. 13th Sirocco*, pages 1–9, 2006.

[14] E. Kranakis, D. Krizanc, N. Santoro, and C. Sawchuk. Mobile agent rendezvous search problem in the ring. In *Proc. International Conference on Distributed Computing Systems (ICDCS)*, pages 592–599, 2003.

[15] G. De Marco, L Gargano, E. Kranakis, D. Krizanc, A. Pelc, and U. Vacaro. Asynchronous deterministic rendezvous in graphs. In *Proc. 30th MFCS*, pages 271–282, 2005.

[16] M. Mitzenmacher and E. Upfal. *Probability and Computing: Randomized Algorithms and Probabilistic Analysis*. Cambridge University Press, 2005.

[17] S. M. Ross. *Probability Models for Computer Science*. Harcourt Academic Press, 2002.

[18] N. Roy and G. Dudek. Collaborative robot exploration and rendezvous: Algorithms, performance bounds and observations. *Autonomous Robots*, 11:117–136, 2001.

[19] N. Santoro. *Design and Analysis of Distributed Algorithms*. Wiley, Hoboken, NJ, 2006.

[20] C. Sawchuk. *Mobile Agent Rendezvous in the Ring*. PhD thesis, Carleton University, School of Computer Science, Ottawa, Canada, 2004.

[21] I. Suzuki and M. Yamashita. Distributed anonymous mobile robots: Formation of geometric patterns. *SIAM Journal of Computing*, 28:1347–1363, 1999.

[22] X. Yu and M. Yung. Agent rendezvous: A dynamic symmetry-breaking problem. In *Proceedings of ICALP96*, volume 1099 of *LNCS*, pages 610–621, 1996.