Yugoslav Journal of Operations Research
4 (1994), Number 1, 67-77

# EQUITABLE PROBABILISTIC ELECTIONS IN RING NETWORKS

Jernej POLAJNAR*

*School of Technical Sciences, University of Novi Sad*
*Dositeja Obradovića 7, 21000 Novi Sad, Yugoslavia*

Tomislav UNKAŠEVIĆ

*Institute of Applied Mathematics and Electronics*
*Kneza Miloša 37, 11000 Belgrade, Yugoslavia*

Abstract: A probabilistic algorithm for leader election in ring networks is described and analyzed. The election is based on comparison of random priority numbers drawn independently by each station from a globally defined finite integer range, with ties resolved by additional election rounds, and with all stations having equal chances of being elected. The algorithm is shown to be partially correct and to terminate with probability 1. The main result of the paper is an explicit formula for the probability distribution of election time (i.e., the number of rounds). It is also shown that all moments of the distribution exist. A distinctive feature of the underlying ring model is the assumption that each receiving station can distinguish its own messages from those of others, but that the stations are otherwise indistinguishable and use no global information. This assumption is motivated by a reliability argument based on reconfigurable local area networks of ring topology.

Key words: Distributed algorithms, probabilistic algorithms, ring networks, election of ring leader, election time, permutation runs.

## 1. INTRODUCTION

In a distributed computer system consisting of *n stations* connected by a communication network, it is sometimes necessary to single out a unique station, called the *leader*, that will perform an operation of global significance for the system. If there exists no central coordinator that could designate the leader, we are facing the problem of *distributed election* of the leader. The problem can be analyzed under a

---

*Mailing address is address of the second author variety of specific assumptions concerning network topology, synchronization, determinism, distinguishability of stations, and global information available to them. A substantial part of research on this topic has so far concentrated on ring networks of two different types. In the first type, ring stations are assumed to have unique identifiers (see e.g. [3]), and deterministic algorithms can be used to elect a particular station that has the largest identifier or satisfies some other criteria. In the second type, ring stations are indistinguishable, and leader election is impossible unless probabilistic techniques are used to break the inherent symmetry of the ring [1]. Moreover, in the absence of station identifiers, a receiving station cannot distinguish its own messages from those of others unless it is given some global information, such as ring size. In either case, the primary goal is to develop correct election algorithms and optimize their performance with respect to suitable complexity measures. Both types of network permit unrestricted use of information available to stations.

In this paper we describe and analyze a leader election algorithm which is essentially different from previous work in that it *restricts* the way in which stations may use the available information. Specifically, we assume that each station can directly decide whether a received message is its own or comes from another source, but cannot derive any other information from station identifiers. In other words, each station can distinguish itself from others, but can see no distinction between other stations. As a consequence, the election problem remains *symmetric*, and probabilistic algorithms must be used, but the weakened *distinguishability* concept still suffices to make probabilistic election without global information possible. The algorithm is based on the familiar method of comparing random *priorities* that are independently drawn by each station from a globally defined finite range of integers. Distinct stations may happen to draw equal priorities, in which case additional election rounds are needed. The election method is *equitable* in the sense that all stations have identical chances of being elected. Each station can exchange with others only random data, which prevents it from deriving any global information about the system through communication, except for testing *solitude*, i.e., the status of being the only active station in the ring.

The approach that restricts the use of station identifiers has been motivated by a reliability argument, developed in [4, 5] in the context of reconfigurable local area networks of ring topology. The gist of the argument is that, in the presence of faults, leader election should be based on reliability criteria, and if failure is followed by a new election, reelection of faulty stations should be avoided by randomization. In particular, if all stations have identical reliability properties, and assuming a certain fault model, equitable probabilistic elections were shown to have the best properties.

Our analysis of the algorithm starts by showing its correctness. In order to analyze the election time, defined as the number of rounds used to elect the leader, we first derive a combinatorial result. The result determines the number of all permutations of a multiset that have a given number of 'circular' runs, i.e., runs that can extend beyond $n$-th position and continue at the first, as permitted by the ring topology of the network. The main result is an explicit formula for the distribution of election time. Finally, we prove that all moments of the distribution exist.

## 2. THE ELECTION ALGORITHM

In this section we first describe the distributed system model, and then formulate the election algorithm and prove its correctness.

The distributed system consists of $n$ stations $C_i$, $i \in I_n$, where $n > 1$ and $I_n = \{0, \dots, n-1\}$. A network of ring topology connects each station $C_i$ with station $C_{i \oplus 1}$, where $i \oplus 1 = i + 1$ for $i < n-1$, and $(n-1) \oplus 1 = 0$. In general, operations $\oplus$ and $\ominus$ on $I_n$ are defined as follows:

$$x \oplus y \overset{def}{=} ((x + y) \bmod n)$$

$$x \ominus y \overset{def}{=} ((x - y) \bmod n)$$

For some $m > 1$, the set $J_m = \{1, \dots, m\}$ is called the *priority range* of the system.

In the course of an election, each station $C_i$ can invoke function ***random*(1, *m*)** to generate a random *priority* $x_i \in J_m$. $C_i$ can also use procedures ***send*(*i*, $x_i$)**, to transmit a message containing its identifier $i$ and priority $x_i$, and ***receive*(*j*, $x_j$)**, to receive and remove from the ring an incoming message of the same format. However, the only permitted use of the received identifier $j$ is in testing whether $i = j$, i.e., whether $C_i$ itself or another station is the source of the message.

The leader election algorithm works as follows. Initially, all stations are *active*. In each iteration of the loop, called an election *round*, some stations may become inactive. The algorithm terminates when no active stations exist. One may find it simpler to think of election rounds in different stations as being performed synchronously. If such external synchronization existed, each station would need an input buffer that can hold a single message. However, the algorithm can be performed asynchronously, in which case an $n$–element input buffer is needed.

```
type    station = 0..(n-1);
        priority = 1..m;
function leader(i: station);
var active: boolean;
    j: station;
    x_i, x_j: priority;
begin
  active := true;
  repeat
     x_i := random(1, m);
     send(i, x_i);
     receive(j, x_j);
     active := x_i ≤ x_j and i ≠ j;
     leader := i = j;
  until not active;
end {leader};
```

Note that only active neighbors communicate in each round. The message sent by $C_i$ will be received and removed by its 'downstream' active neighbor. The message received by $C_i$ was sent by its 'upstream' active neighbor $C_j$. When $i = j$, the closest neighbor of $C_i$ in either direction is itself, which means that $C_i$ is the only remaining active station, and hence the leader. Otherwise, $C_i$ gets eliminated when it has a higher priority number, which means a lower precedence, than its upstream active neighbor $C_j$. Thus the progress in each round depends not only on the priorities chosen by stations, but also on the arrangement of those priorities along the logical ring formed by the currently active stations.

Our main purpose in observing the election process is to follow how the number of active stations is being gradually reduced from $n$ to 1. The system *state* is therefore defined as the number of currently active stations. An election round in which that number is actually reduced is called *productive*. However, note that all active stations can be assigned equal priorities, in which case none is eliminated and the round is called *unproductive*. In general, the probability of *transition* of the system from state $i$ to state $j$ in $k$ rounds will be denoted by $p(i, j, k)$. In particular, $p(n, 1, k)$ is then the probability that the election will be completed in exactly $k$ rounds, since an election cannot end with an unproductive round.

Since the election algorithm is probabilistic, we can speak of its correctness only in probabilistic terms.

**THEOREM 1.** The election algorithm terminates with probability 1 and elects a unique leader.

PROOF. To show partial correctness, suppose the algorithm terminates, and let $i_1 < \dots < i_k$, where $k \geq 1$, be the indices of active stations at the beginning of the last round. Suppose $k > 1$. Then $x_{i_1} < \dots < x_{i_k}$ since $i_2, \dots, i_k$ get eliminated in the last round, and hence by transitivity $x_{i_1} < x_{i_k}$. However, since $i_1$ gets eliminated as well, it follows that $x_{i_k} < x_{i_1}$ – a contradiction. Thus $k = 1$, and $i_1$ gets eliminated since $i = j = i_1$, which implies that $C_{i_1}$ is the unique leader. To show termination suppose the contrary, namely that an execution of the algorithm never terminates. Then the election process must reach its minimum number $i$ of active stations, where $i > 1$, after some number of rounds, and remain in that state forever. Let $\pi(n, i)$ be the probability that the system reaches state $i$ in a finite number of rounds. An unproductive round in state $i$ arises when all $i$ stations choose the same priority (out of $m$ possibilities); hence its probability is $p(i, i, 1) = m(1/m)^i = (1/m)^{i-1}$. The probability of $j$ unproductive rounds in state $i$ is then $p(i, i, j) = p(i, i, 1)^j = (1/m)^{j(i-1)}$. The combined probability of transition from state $n$ to a fixed state $i$ in an arbitrary number $k$ of rounds (with last round productive), and then remaining in state $i$ for another $j$ rounds is

$$Q_{ij} = \pi(n, i)\, p(i, i, j) \leq p(i, i, j) = (1/m)^{j(i-1)}$$

The probability that the election never terminates is

$$Q = \lim_{j \to \infty} \sum_{i=2}^{n} Q_{ij} \leq (n-1) \lim_{j \to \infty} (1/m)^j = 0$$

## 3. PRIORITY ASSIGNMENTS

It is evident from the election algorithm that in each productive round there exist two active stations $C_i$ and $C_j$ with priorities $x_i$ and $x_j$, respectively, such that $C_i$ is the first upstream active neighbor of $i$ and $x_i < x_j$. In each instance of such situation, station $C_i$ will become inactive. More generally, if priorities of a sequence of consecutive (in the direction of ring traffic) active stations form a maximal monotonically increasing chain, then all stations in that sequence, except the first one, will be inactive in the next election round. Below we formally define and investigate such chains in order to determine the number of election rounds needed to complete an election. We use the term 'R–chain' to emphasize the underlying ring topology, in which the successor of $C_n$ is station $C_1$.

**DEFINITION 1.** A sequence of priorities $x_r, x_{r\oplus 1}, \dots, x_s$ chosen by stations $C_r, C_{r\oplus 1}, \dots, C_s$, where $s \ominus r \geq 0$, is called a *non-decreasing R–chain* if and only if $x_r \leq \dots \leq x_s$, $x_{r\ominus 1} > x_r$, and $x_s > x_{s\oplus 1}$. A sequence of priorities $x_r, x_{r\oplus 1}, \dots, x_s$ is called an *increasing R–chain* if and only if $x_r < \dots < x_s$, $x_{r\ominus 1} \geq x_r$, and $x_s \geq x_{s\oplus 1}$.

Thus, R–chains are maximal chains of priorities found in a ring. For this reason, an increasing R–chain is not necessarily a non-decreasing R–chain (but is always a part of one). In dealing with R–chains it is useful to rely on a simpler concept of 'L–chains' which are fully contained within the 'linear' sequence $x_1, \dots, x_n$ and do not depend on ring topology. In other words, $C_n$ has no successor in L–chains, and $C_1$ has no predecessor, as if we had cut the ring after $C_n$ to obtain a linear ('bus') topology.

**DEFINITION 2.** A sequence of priorities $x_r, x_{r+1}, \dots, x_s$ chosen by the stations $C_r, C_{r+1}, \dots, C_s$, where $s \geq r$, is called a *non-decreasing L–chain* if and only if (1) $x_r \leq \dots \leq x_s$, (2) $r = 1$ or $x_{r-1} > x_r$, and (3) $s = n$ or $x_s > x_{s+1}$. A sequence of priorities $x_r, x_{r+1}, \dots, x_s$ is called an *increasing L–chain* if and only if (1) $x_r < \dots < x_s$, (2) $r = 1$ or $x_{r-1} \geq x_r$, and (3) $s = n$ or $x_s \geq x_{s+1}$.

For notational simplicity, we have defined R–chains and L–chains with respect to the complete system configuration with $n$ stations, and we shall derive their properties using the same context. However, those properties will later also be applied to smaller configurations, consisting only of active nodes. The same remark refers to the notation for priority assignments that we now introduce.

**NOTATION 1.** For all integers $n$, $k$, $1 \leq k \leq n$, let $\mathcal{L}_{nk}$ be the set of all priority assignments $l : I_n \to J_m$ with exactly $k$ *increasing L–chains* and $L_{nk}$ the cardinality of $\mathcal{L}_{nk}$. Let $\tilde{\mathcal{L}}_{nk}$ and $\tilde{L}_{nk}$ be analogously defined for *non-decreasing L–chains*, $\mathcal{R}_{nk}$ and $R_{nk}$ for *increasing R–chains*, and $\tilde{\mathcal{R}}_{nk}$ and $\tilde{R}_{nk}$ for *non-decreasing R–chains*.

In proving properties of chains we shall need the following result.

**Lemma 1.** For all integers $n \geq 1$, $m \geq 1$, $a \geq 0$, and $n_j \geq 0$, $1 \leq j \leq m$,

$$\sum_{n_1+\cdots+n_m=n} \prod_{j=1}^{m} \binom{a+n_j-1}{n_j} = \binom{ma+n-1}{n} \tag{1}$$

**Proof.** We identify the coefficients of $x^n$ in the expansions

$$f(x) = (1-x)^{-ma} = \sum_{n=0}^{\infty} \binom{-ma}{n} x^n = \sum_{n=0}^{\infty} \binom{ma+n-1}{n} x^n \tag{2}$$

and

$$f(x) = \prod_{j=1}^{m} (1-x)^{-a} = \sum_{n=0}^{\infty} \left( \sum_{n_1+\cdots+n_m=n} \prod_{j=1}^{m} \binom{a+n_j-1}{n_j} \right) x^n \tag{3}$$

**Proposition 1.** The numbers $L_{nk}$ and $\tilde{L}_{nk}$ of priority assignments with exactly $k$ L–chains have the following properties:

(a) $L_{n\,n-k+1} = \tilde{L}_{n,k}$

(b) $\tilde{L}_{n,k} = \sum_{i=0}^{k} (-1)^k \binom{n+1}{i} \binom{m(k-i)+n-1}{n}$

**Proof.** To show part (a), let $h : \tilde{L}_{nk} \to L_{n\,n-k+1}$ be the mapping defined by $h(\tilde{l})(i) = \tilde{l}(n-i-1)$ for all $\tilde{l} \in \tilde{L}_{nk}$ and $i \in I_n$. To show that $\tilde{l} \in \tilde{L}_{nk}$ indeed implies that $l = h(\tilde{l})$ belongs to $L_{n\,n-k+1}$, note that for all $i$, $0 \leq i < n-1$, $\tilde{l}(i) > \tilde{l}(i+1)$ holds if and only if $l(n-i-2) < l(n-i-1)$. In other words, a non–decreasing chain boundary between $\tilde{l}(i)$ and $\tilde{l}(i+1)$ exists if and only if there exists no increasing chain boundary between $l(n-i-2)$ and $l(n-i-1)$. Now $\tilde{l} \in \tilde{L}_{nk}$ implies that there are $k$ non–decreasing chains in $\tilde{l}$, and hence $k-1$ values of $i$ corresponding to non–decreasing chain boundaries between $\tilde{l}(i)$ and $\tilde{l}(i+1)$. Since $i$ has $n-1$ possible values, there are $(n-1) - (k-1) = n-k$ values of $i$ such that there is an increasing chain boundary between $l(n-i-2)$ and $l(n-i-1)$, and hence $n-k+1$ increasing chains in $l$. Thus $h(\tilde{L}_{nk}) \subseteq L_{n\,n-k+1}$. The fact that $h$ is a bijection now follows trivially from the definition. Hence $L_{n\,n-k+1} = \tilde{L}_{n,k}$.

To show (b), let $\tilde{L}_{nk}^{n_1 \ldots n_m}$ denote the number of priority assignments with exactly $k$ non–decreasing L–chains and with exactly $n_1$ ones, $n_2$ twos, ... ,$n_m$ $m$–s, where $\sum_{i=1}^{m} n_i = n$ and $n_i \geq 0$ for all $i \in J_m$. Note that $\tilde{L}_{nk}^{n_1 \ldots n_m}$ is the number of permutations with exactly $k$ runs of the multiset $\{n_1 \cdot 1, n_2 \cdot 2, \dots, n_m \cdot m\}$. It was shown in Exercise 5.1.3.12 of [2] that

$$\tilde{L}_{nk}^{n_1 \dots n_m} = \sum_{i=0}^{k} (-1)^i \binom{n+1}{i} \prod_{j=1}^{m} \binom{n_j - 1 + k - i}{n_j} \tag{4}$$

From the definition it is clear that

$$\tilde{L}_{nk} = \sum_{n_1 + \dots + n_m = n} L_{nk}^{n_1 \dots n_m} \tag{5}$$

Substituting (4) into (5) and changing the order of summation,

$$\tilde{L}_{nk} = \sum_{i=0}^{k} (-1)^i \binom{n+1}{i} \sum_{n_1 + \dots + n_m = n} \prod_{j=1}^{m} \binom{n_j - 1 + k - i}{n_j}$$

and the statement follows directly by Lemma 1.

**THEOREM 2.** The numbers $R_{nk}$ and $L_{nk}$ satisfy the relationship

$$R_{nk} = \binom{n}{k} \sum_{j=1}^{k} (-1)^{j-1} \frac{L_{n\,k-j+1}}{\binom{n-1}{k-j}}$$

PROOF. Let $\mathcal{R}_n = \bigcup_{k=1}^{n} \mathcal{R}_{nk}$, and $\mathcal{L}_n = \bigcup_{k=1}^{n} \mathcal{L}_{nk}$, and $\overline{\mathcal{R}}_n = \mathcal{R}_n \times I_n$. Intuitively, each $(r, i) \in \overline{\mathcal{R}}_n$ is interpreted as a probability assignment $r \in \mathcal{R}_n$ in ring system with a 'cut' between $C_i$ and $C_{i\oplus 1}$. The (increasing) chains of $(r, i)$ coincide with the (increasing) R–chains of $r$, except that a chain cannot cross the cut. Thus for $r \in \mathcal{R}_{nk}$, $(r, i)$ has $k$ chains if $r(i) \geq r(i \oplus 1)$ and $k + 1$ chains otherwise. Let $\overline{\mathcal{R}}_{nk}$ be the set of all elements of $\overline{\mathcal{R}}_n$ with exactly $k$ chains and let $\overline{R}_{nk}$ be the cardinality of $\overline{\mathcal{R}}_{nk}$. Clearly, for each $r \in \mathcal{R}_{nk}$, there are $k$ values of $i$ such that $(r, i) \in \overline{\mathcal{R}}_{nk}$, and $n-k$ values of $i$ such that $(r, i) \in \overline{\mathcal{R}}_{n\,k+1}$. Conversely, $\overline{R}_{nk}$ contains $k$ elements of the form $(r, i)$ for each $r \in \mathcal{R}_{nk}$, and $n - k + 1$ elements $(r, i)$ for each $r \in \mathcal{R}_{nk-1}$. Thus

$$\overline{R}_{nk} = kR_{nk} + (n - k + 1)R_{n\,k-1} \tag{6}$$

Next, let $h : \overline{\mathcal{R}}_n \to \mathcal{L}_n$ be the mapping $h(r, i)(j) \stackrel{def}{=} r(j \oplus i)$ for all $(r, i) \in \overline{\mathcal{R}}_n$ and all $i \in I_n$. Intuitively, $h(r, i)$ can be viewed as the sequence of priorities of $r$, cyclically rearranged to start at the cut, i.e., with $r(i \oplus 1)$ rather than with $r(1)$. Since the chains of $(r, i)$ are precisely the (increasing) L–chains of $h(r, i)$, it follows that $h(\overline{\mathcal{R}}_{nk}) = \mathcal{L}_{nk}$. Note that $h(r, i) = h(r', i')$ is equivalent to $(\forall j)\, r(j \oplus i) = r'(j \oplus i')$, i.e. to the condition that $r'$ can be obtained by 'circular shifting' of $r$. It follows that for each $(r, i)$ the set $\{ (r', i') \mid h(r', i') = h(r, i) \}$ has exactly $n$ elements, which implies that $\overline{R}_{nk} = nL_{nk}$.

Substituting into (6), we get

$$\begin{aligned} R_{n0} &\overset{def}{=} 0 \\ nL_{nk} &= kR_{nk} + (n-k+1)R_{n\,k-1} \quad \text{for } k \in I_n \end{aligned} \tag{7}$$

and hence the recurrence relation for $R_{nk}$:

$$R_{nk} = \frac{1}{k}\left[nL_{nk} - (n-k+1)R_{n\,k-1}\right] \quad \text{for } k \in I_n \tag{8}$$

The statement of the theorem follows from (8) by straightforward mathematical induction on $k$.

## 4. THE PROBABILITY DISTRIBUTION OF ELECTION TIME

Our intention is to find the probability that the system changes its state from $n$ to 1 using $k$ rounds. Let $p(i,j,k)$ denote the probability of transition from state $i$ to state $j$ in $k$ rounds, and let $p(i,j)$ be an abbreviation for $p(i,j,1)$. For $k = 0$ we have $p(i,j,0) = \delta_{ij}$ where $\delta_{ij}$ is the Kronecker delta symbol. From the definition it is clear that

$$p(i,j,k+1) = \begin{cases} \sum_{l=j}^{i} p(i,l,k)\,p(l,j) & \text{if } i \ge j \text{ and } i > 1 \\ 0 & \text{otherwise} \end{cases} \tag{9}$$

By using the last recurrence relation, state transition probabilities of the form $p(i,j,k)$ can be reduced to those of the form $p(i,j)$. The latter are computed as follows.

**PROPOSITION 2.** The probability of transition from state $i$ to $j$ in one round is:

$$p(i,j) = \frac{R_{i,j}}{m^i} \tag{10}$$

PROOF. By Definition 1, a single round transition from state $i$ to state $j$ will take place if the current priority assignment $r$ has exactly $j$ increasing R–chains. Thus $p(i,j)$ is the probability that $r \in \mathcal{R}_{ij}$. Since the number of assignments in $\mathcal{R}_{ij}$ is $R_{ij}$, and the number of all possible assignments in state $i$ is $m^i$, the statement follows.

We can now prove the main result of the paper. Let us define a random variable $T_n$, called the *election time*, as the number of rounds used in the election process. The probability distribution of election time is given by the following result.

**THEOREM 3.** The probability of election in exactly $k$ rounds is

$$P\{T_n = k\} = \sum_{r=1}^{\min(k,n-1)} \sum_{\substack{n=i_1>\cdots>i_{r+1}=1 \\ d_1+\cdots+d_r=k-r}} \prod_{j=1}^{r} m^{d_j(1-i_j)}\, p(i_j, i_{j+1})$$

Proof. Suppose the election has $k$ rounds, out of which $r$ are productive. Clearly, $r$ is at least 1 and at most $\min(k, n-1)$. Descending from state $n$ to 1, the system assumes $r+1$ distinct states $i_j$ and has $d_j \ge 0$ unproductive rounds in $i_j$, $1 \le j \le r+1$. The election starts in state $i_j = n$ and terminates in $i_{r+1} = 1$, so $d_{r+1} = 0$. The total number of unproductive rounds is $\sum_{j=1}^{r} d_j = k - r$. The probability of $d_j$ unproductive rounds in state $i_j$ followed by a productive round leading to state $i_{j+1}$ is $p(i_j, i_j, d_j)\, p(i_j, i_{j+1}, 1)$; the probability that this occurs for all $j$ is obtained as a product over $j$. Thus

$$P\{T_n = k\} = \sum_{r=1}^{\min(k,n-1)} \sum_{\substack{n=i_1>\cdots>i_{r+1}=1 \\ d_1+\cdots+d_r=k-r}} \prod_{j=1}^{r} p(i_j, i_j, d_j)\, p(i_j, i_{j+1}, 1) \tag{11}$$

To complete the proof, note that $p(i_j, i_{j+1}, 1) = p(i_j, i_{j+1})$, $p(i_j, i_j, d_j) = [\,p(i_j, i_j, 1)\,]^{d_j}$, and $p(i_j, i_j, 1) = m(1/m)^{i_j} = m^{1-i_j}$, and the statement immediately follows.

## 5. CONVERGENCE RESULTS

In this section we first establish an upper bound on $P\{T_n = k\}$, and then use it to prove the existence of all moments of the probability distribution.

**Lemma 2.** For $k \geq n$, the probability distribution of election time $T_n$ satisfies the inequality

$$P\{T_n = k\} \leq m^{n-k-1} k^{n-2}$$

Proof. In the expression for $P\{T_n = k\}$ as given by Theorem 3, we first note that for each $j \leq r$, $i_j \geq 2$ holds and hence $m^{d_j(1-i_j)} \leq m^{-d_j}$. Since $p(i_j, i_{j+1}) \leq 1$, it follows that

$$\prod_{j=1}^{r} m^{d_j(1-i_j)} p(i_j, i_{j+1}) \leq \prod_{j=1}^{r} m^{-d_j} = m^{r-k} \tag{12}$$

Furthermore, observe that $r-1$ distinct integers $i_2, \ldots, i_r$ can be selected from $\{2, \ldots, n-1\}$ in $\binom{n-2}{r-1}$ ways, and that the number $k-r$ of unproductive rounds can be partitioned into $d_1, \ldots, d_r$ in $\binom{k-r+r-1}{r-1} = \binom{k-1}{r-1}$. We get

$$P\{T_n = k\} \leq \sum_{r=1}^{n-1} \binom{n-2}{r-1} \binom{k-1}{r-1} m^{r-k} \tag{13}$$

Using $\binom{k-1}{r-1} \leq (k-1)^{r-1}$ we obtain

$$\begin{aligned} P\{T_n = k\} &\leq \sum_{r=1}^{n-1} \binom{n-2}{r-1} (m(k-1))^{r-1} m^{1-k} = \\ &= m^{1-k}(mk - m + 1)^{n-2} \\ &\leq m^{n-k-1} k^{n-2} \end{aligned} \tag{14}$$

which completes the proof.

An immediate consequence of Lemma 2 is the following result.

**THEOREM 4.** The expected value, variance, and all higher-level moments of the probability distribution of election time $T_n$ exist.

PROOF. It suffices to show that the infinite series

$$S_q(T_n) = \sum_{k=1}^{\infty} k^q P\{T_n = k\} \tag{15}$$

converges for all $q \geq 0$. By Lemma 2, for $k > n$, the general term of (15) satisfies

$$k^q P\{T_n = k\} \leq m^{n-1-k} k^{n+q-2} \tag{16}$$

The series $\sum_{k=0}^{\infty} k^s x^k$ is convergent in the interval $(-1, 1)$. Putting $s = n + q - 2$ and $x = 1/m$ we see that the series $m^{n-1} \sum_{k=0}^{\infty} k^{n+q-2} (1/m)^k$ is convergent for all $m$, $n$, $q$ and $S_q(T_n)$ exist for all $q$. Then $E = S_1(T_n)$ is the expected value of election time and, in general,

$$M_q = \sum_{k=1}^{\infty} (k - E)^q P\{T_n = k\} = \sum_{i=0}^{q} (-1)^{q-i} \binom{q}{i} E^{q-i} S_i(T_n) \tag{17}$$

is the moment of level $q$ of the probability distribution.

## 6. CONCLUSION

A simple probabilistic algorithm for leader election in a ring has been formulated and proved correct. The probability distribution of its election time has been established, and all moments of the distribution have been proved to exist. These results form a suitable basis for further study of this and related algorithms. Such study may be of interest since the comparison of random priorities, employed by the present algorithm, is a basic probabilistic election technique. The algorithm is applicable to ring models in which stations have unique identifiers. In particular, this allows for deterministic resolution of ties that could speed up the election. However, our algorithm avoids deterministic symmetry-breaking mechanisms and restricts the use of station identifiers to testing whether or not a received message is one' s own. This feature places it in a separate class of leader election algorithms. It also gives rise to a possible research topic, namely a systematic investigation of the proposed class of leader election algorithms for ring (and perhaps other) networks, including the search for those with minimum election time, as well as a study of potential application areas.

## REFERENCES

[1] Itai,A., and Rodeh,M., "Symmetry breaking in distributed networks", *Information and Computation* 88 (1990) 60-87.

[2] Knuth,D.E., *The Art of Computer Programming, vol. 3, Sorting and Searching*, 2nd ed., Addison-Wesley, 1973.

[3] Lynch N.A., and .Goldman,K.J., "Distributed Algorithms" MIT/LCS/RSS 5, Research Seminar Series, Massachusetts Institute of Technology, 1989.

[4] Polajnar,J., and Mastilović,Dj. "A probabilistic election algorithm for FDDI fault recovery", presented at the *12-th World Computer Congress, IFIP'92*, Madrid, Spain, Sept. 7-11, 1992.

[5] Polajnar,J., Mastilović,Dj., and Pardulski,B., "Decentralized loopback reconfiguration of a bidirectional ring LAN", in: *Proc. Eighth Symp. on Reliable Distributed Systems*, Seattle, Oct. (1989) 72-78.