A Survey on Leader Election Protocols for Radio Networks *

Koji Nakano† Stephan Olariu‡

Abstract

A radio network is a distributed system with no central arbiter, consisting of n radio transceivers, henceforth referred to as stations. We assume that the stations are identical and cannot be distinguished by serial or manufacturing number. The leader election problem asks to designate one of the station as leader.

In this work, we focus on single-channel, single-hop radio networks. We assume that time is slotted and all transmissions occur at slot boundaries. In each time slot the stations transmit on the channel with some probability until, eventually, one of the stations is declared leader. The history *of a station up to time slot t is captured by the status of the channel and the transmission activity of the station in each of the t time slots.*

From the perspective of how much of the history information is used, we identify three types of leader election protocols for single-channel, single-hop radio networks: **oblivious** *if no history information is used,* **uniform** *if only the history of the status of the channel is used, and* **non-uniform** *if the stations use both the status of channel and the transmission activity.*

The main goal of this paper is to provide a survey of recent leader election protocols for radio networks.

1 Introduction

A radio network (RN, for short) is a distributed system with no central arbiter, consisting of n radio transceivers, henceforth referred to as *stations*. In a *single-channel* RN the stations communicate over a unique radio frequency channel known to all the stations. A RN is said to be *single-hop* when all the stations are within transmission range of each other. In this paper we focus on single-channel, single-hop radio networks.

Radio transmission is *isotropic*, that is, when a station is transmitting all the stations in its communication range receive the packet. We note here that this is in sharp contrast with the basic point-to-point assumption in wireline networks where a station can specify a unique destination station. We employ the commonly-accepted assumption that when two or more stations are transmitting on a channel in the same time slot, the corresponding packets *collide* and are garbled beyond recognition. It is customary to distinguish among radio networks in terms of their *collision detection* capabilities. In the RN with collision detection the status of a radio channel in a time slot is:

NULL: if no station transmitted in the current time slot,

SINGLE: if exactly one station transmitted in the current time slot,

COLLISION: if two or more stations transmitted the channel in the current time slot.

The problem that we survey in this paper is the classical *leader election* problem which asks to designate one of the station in the network as *leader*. In other words, after executing the leader election protocol, exactly one station learns that it was elected leader, while the remaining stations learn the identity of the leader. Historically, the leader election problem has been addressed in wireline networks, where each station can specify a destination station.

The leader election problem can be studied in the following three scenarios:

Scenario 1: The number n of stations is known in advance;

Scenario 2: The number n of stations is unknown, but an upper bound u on n is known in advance;

---

*Work supported, in part, by the Telecommunications Advancement Foudation, by the NSF grant CCR-9522093, and by ONR grant N00014-97-1-0526

†Graduate School of Information Science, Japan Advanced Institute of Science and Technology, Tatsunokuchi, Ishikawa 923-1292, Japan, knakano@jaist.ac.jp

‡Department of Computer Science, Old Dominion University, Norfolk, Virginia 23529, USA, olariu@cs.odu.edu

Scenario 3: Neither the number of stations nor an upper bound on this number is known in advance.

It is intuitively clear that the task of leader election is the easiest in Scenario 1 and the hardest in Scenario 3, with Scenario 2 being in-between the two.

Randomized leader election protocols designed for single-channel, single-hop radio networks work as follows: in each time slot, the stations transmit on the channel with some probability. As we will discuss shortly, this probability may or may not be the same for individual stations. If the status of the channel is SINGLE, the unique station that has transmitted is declared the leader. If the status is not SINGLE, the above is repeated until, eventually, a leader is elected. Suppose that a leader election protocol runs for t time slots and a leader has still not been elected at that time. The *history* of a station up to time slot t is captured by **the status of the channel**: The status of the channel in each of the t time slots, that is, a sequence of {NULL,COLLISION} of length t. **transmit/not-transmit**: The transmission activity of the station in each of the t time slots, that is, a sequence of {transmit,not-transmit} of length t.

It should be clear that its history contains all the information that a station can obtain in t time slots. From the perspective of how much of the history information is used, we identify three types of leader election protocols for single-channel, single-hop radio networks:

oblivious: In time slot i, $(1 \leq i)$, every station transmits with probability p_i. The probability p_i is fixed beforehand and does not depend on the history

uniform: In time slot i, $(1 \leq i)$, all the stations transmit with the *same* probability p_i. Here p_i is a function of the history of the status of channel in time slots $1, 2, \ldots, i-1$

non-uniform: In each time slot, every station determines its transmission probability depending on its own history.

An *oblivious* leader election protocol is uniquely determined by a sequence $P = \langle p_1, p_2, \ldots \rangle$ of probabilities. In time slot i, $(1 \leq i)$, every station transmits with probability p_i. A leader is elected if the status of the channel is SINGLE. Clearly, oblivious leader election protocols also work for radio networks with no collision detection, in which the stations cannot distinguish between NULL and COLLISION.

A *uniform* leader election protocol is uniquely determined by a binary tree T of probabilities. T has nodes $p_{i,j}$, $(1 \leq i; 1 \leq j \leq 2^{i-1})$, each corresponding to a probability. Node $p_{i,j}$ has left child $p_{i+1,2j-1}$ and right child $p_{i+1,2j}$. The leader election protocol traverses T from the root as follows: initially, the protocol is positioned at the root $p_{1,1}$. If in time slot i the protocol is positioned at node $p_{i,j}$ then every station transmits on the channel with probability $p_{i,j}$. If the status of the channel is SINGLE, the unique station that has transmitted becomes the leader and the protocol terminates. If the status of channel is NULL, the protocol moves to the left child $p_{i+1,2j-1}$; if the status is COLLISION the protocol moves to the right child $p_{i+1,2j}$.

Similarly, a *non-uniform leader election protocol* is captured by a ternary tree T with nodes $p_{i,j}$, $(1 \leq i; 1 \leq j \leq 3^{i-1})$, each corresponding to a probability. The children of node $p_{i,j}$ are, in left to right order, $p_{i+1,3j-2}$, $p_{i+1,3j-1}$ and $p_{i+1,3j}$. Each station traverses T from the root as follows. Initially, all the stations are positioned at the root $p_{1,1}$. If in time slot i a station is positioned at node $p_{i,j}$ then it transmits with probability $p_{i,j}$. If the status of the channel is SINGLE, the unique station that has transmitted becomes the leader and the protocol terminates. If the status of the channel is NULL, the station moves to $p_{i+1,3j-2}$. If the status of channel is COLLISION, then the station moves to $p_{i+1,3j-1}$ or $p_{i+1,3j}$ depending on whether or not is has transmitted in time slot i.

Several randomized protocols for single-channel, single-hop networks have been presented in the literature. Metcalfe and Boggs [2] presented an oblivious leader election protocol for Scenario 1 that is guaranteed to terminate in $O(1)$ expected time slots. Their protocol is very simple: every station keeps transmitting on the channel with probability $\frac{1}{n}$. When the status of channel becomes SINGLE, the unique station that has transmitted is declared the leader. Recently, Nakano and Olariu [3] presented two non-uniform leader election protocols for Scenario 3. The first one terminates, with probability $1-\frac{1}{n}$, in $O(\log n)$ time slots. The second one terminates with probability $1 - \frac{1}{\log n}$ in $O(\log\log n)$ time slots. The main drawback of these protocols is that the "high probability" expressed by either $1 - \frac{1}{n}$ or $1 - \frac{1}{\log n}$ becomes meaningless for small values of n. For example, the $O(\log\log n)$-time protocol may take a very large number of time slots to terminate. True, this only happens with probability at most $\frac{1}{\log n}$. However, when n is small, this probability is non-negligible. To address this shortcoming, Nakano and Olariu [4] improved this protocol to terminate, with probability exceeding $1 - \frac{1}{f}$, in $\log\log n + 2.78\log f + o(\log\log n + \log f)$ time slots. Nakano and Olariu [5] also presented an oblivious leader election protocol for Scenario 3 terminating with probability at least $1 - \frac{1}{f}$, in $O(\min((\log n)^2 + (\log f)^2, f^{\frac{3}{5}} \log n))$ time slots.

In a landmark paper, Willard [7] presented a uniform leader election protocol for the conditions of Scenario 2 terminating in $\log\log u + O(1)$ expected time slots. Willard's protocol involves two stages: the first stage, using binary search, guesses in $\log\log u$ time slots a number i, $(0 \le i \le \log u)$, satisfying $2^i \le n < 2^{i+1}$. Once this approximation for n is available, the second stage elects a leader in $O(1)$ expected time slots using the protocol of [2]. Thus, the protocol elects a leader in $\log\log u + O(1)$ expected time slots. Willard [7] went on to improve this protocol to run under the conditions of Scenario 3 in $\log\log n + o(\log\log n)$ expected time slots. The first stage of the improved protocol uses the technique presented in Bentley and Yao [1], which finds an integer i satisfying $2^i \le n < 2^{i+1}$, bypassing the need for a known upper-bound u on n. More recently, Nakano and Olariu with probability exceeding $1 - \frac{1}{f}$, in $\log\log n + o(\log\log n) + O(\log f)$ time slots. Our uniform leader election features the same performance as the non-uniform leader election protocol of [4] even though all the stations transmit with the same probability in each time slot.

In this paper, we survey known leader election protocols. We refer the reader to Table 1 for the characteristics of these protocols.

2 Oblivious leader election protocols

The main goal of this section is to discuss oblivious leader election protocols for radio networks for Scenarios 1, 2, and 3.

2.1 Oblivious leader election for Scenario 1

Let $P = \langle p_1, p_2, p_3, \ldots\rangle$ be an arbitrary sequence of probabilities and suppose that in time slot i each of the n stations of the RN is transmitting on the channel with probability p_i. If the status of the channel is SINGLE, the unique station that has transmitted becomes the leader. Otherwise, in time slot $i+1$ every station transmits with probability p_{i+1}. This is repeated until either the sequence P is exhausted or the status of the channel is, eventually, SINGLE. The details are spelled out in the following protocol.

Protocol `Election(`P`)`
for $i \leftarrow 1$ to $|P|$ do
  each station transmits with probability p_i
    and all stations monitor the channel;
  if the status of the channel is SINGLE **then** the
    station that has transmitted becomes the leader.
endfor

Clearly, since every station transmits with the same probability p_i in time slot i, `Election(`P`)` is oblivious for any sequence P of probabilities. Since correctness is easy to see, we now turn to the task of evaluating the number of time slots it takes protocol `Election(`P`)` to terminate. The status of the channel is SINGLE with probability $p_i(1-p_i)^{n-1}$. Simple calculations show that if we choose $p_i = \frac{1}{n}$, the probability is maximized. In this case,

$$p_i(1-p_i)^{n-1} \quad = \quad \left(1 - \frac{1}{n}\right)^{n-1} > \frac{1}{e}.$$

Therefore, we choose $P = \langle \frac{1}{n}, \frac{1}{n}, \frac{1}{n}, \ldots\rangle$. Now, each iteration of the **for** loop in protocol `Election(`$P = \langle \frac{1}{n}, \frac{1}{n}, \frac{1}{n}, \ldots\rangle$`)` succeeds in electing a leader with probability exceeding $\frac{1}{e}$. Hence, t trials fails to elect a leader with probability $(1 - \frac{1}{e})^t < e^{-\frac{t}{e}}$. Let f be a parameter satisfying $\frac{1}{f} = e^{-\frac{t}{e}}$. Then, we have $t = e \ln f$. Therefore, we have the following lemma:

Lemma 2.1 `Election(`$\langle \frac{1}{n}, \frac{1}{n}, \frac{1}{n}, \ldots\rangle$`)` *elects a leader in $e \ln f$ time slots with probability at least $1 - \frac{1}{f}$ for any $f \ge 1$.*

Note that the value of n must be known to every station in order to perform `Election(`$\langle \frac{1}{n}, \frac{1}{n}, \frac{1}{n}, \ldots\rangle$`)`.

2.2 Oblivious leader election for Scenario 2

The main purpose of this subsection is to discuss a randomized leader election protocol for an n-station RN under the assumption that an upper bound u of the number n of stations is known beforehand. However, the actual value of n itself is not known.

Let D_i, $(1 \ge 1)$, be the sequence of probabilities of length i defined as

$$D_i \quad = \quad \langle \frac{1}{2^1}, \frac{1}{2^2}, \ldots, \frac{1}{2^i}\rangle.$$

We propose to investigate the behavior of protocol `Election` when run with the sequence D_i. Can we expect `Election(`D_i`)` to terminate with the election of a leader? The answer is given by the following result.

Lemma 2.2 *For every n, protocol* `Election(`D_i`)` *succeeds in electing a leader with probability at least $\frac{1}{2}$ whenever $i \ge \log n$.*

See [5] for the details of the proof.

Let $D_i^\infty = D_i \cdot D_i \cdot D_i \cdots$ be an infinite sequence, where "$\cdot$" denotes the concatenation of sequences. For

Table 1. *A summary of known leader election protocols*

protocol	scenario	time slots with probability $1-\frac{1}{f}$	time slots average
oblivious	1	$e \ln f$	e
oblivious	2	$\log u \log f$	$O(\log u)$
oblivious	3	$O((\log n)^2 + (\log f)^2)$	$O((\log n)^2)$
oblivious	3	$O(f^\epsilon \log n)$	$O(\log n)$
oblivious	3	$O(\min((\log n)^2 + (\log f)^2, f^\epsilon \log n))$	$O(\log n)$
uniform	3	$\log\log n + o(\log\log n) + O(\log f)$	$\log\log n + o(\log\log n)$
non-uniform	3	$\log\log n + 2.78 \log f + o(\log\log n + \log f)$	$\log\log n + o(\log\log n)$

example, $D_2^\infty = \langle \frac{1}{2}, \frac{1}{4}, \frac{1}{2}, \frac{1}{4}, \frac{1}{2}, \frac{1}{4}, \ldots \rangle$. Suppose that every station knows the upper bound u of the number n of the station. Since $\texttt{Election}(D^\infty_{\lceil \log u \rceil})$ elects a leader with probability at least $\frac{1}{2}$ from Lemma 2.2, t times iteration of $\texttt{Election}(D^\infty_{\lceil \log u \rceil})$ fails to elect a leader with probability $\frac{1}{2^t}$. Also, the t times iteration runs in $t\lceil \log u \rceil$ time slots. Therefore, we have

Lemma 2.3 $\texttt{Election}(D^\infty_{\lceil \log u \rceil})$ *elects a leader in* $\log f \lceil \log u \rceil$ *time slots with probability at least* $1 - \frac{1}{f}$ *for any* $f \geq 1$.

2.3 Oblivious leader election for Scenario 3

Let $V = \langle v(1), v(2), \ldots \rangle$ be a non-decreasing sequence of positive integers such that $1 \leq v_1 \leq v_2 \leq \cdots$ holds. For such sequence V, let $P(V) = D_{v(1)} \cdot D_{v(2)} \cdot D_{v(3)} \cdots$ be the infinite sequence of probabilities. For example, if $V = \langle 1, 2, 3, \ldots \rangle$, then $P(V) = D_1 \cdot D_2 \cdot D_3 \cdots = \langle \frac{1}{2}, \frac{1}{2}, \frac{1}{4}, \frac{1}{2}, \frac{1}{4}, \frac{1}{8}, \ldots \rangle$. We are going to evaluate the performance $\texttt{Election}(P(V))$ for various sequences V.

For a sequence $V = \langle v(1), v(2), \ldots \rangle$, let $l(V)$ denote the minimum integer satisfying $v(l(V)) \geq \lceil \log n \rceil$. Notice that, from Lemma 2.2, each call of $\texttt{Election}(D_{v(l(V))})$, $\texttt{Election}(D_{v(l(V)+1)})$, ..., elects a leader with probability at least $\frac{1}{2}$. Thus, $l(V) + t - 1$ calls $\texttt{Election}(D_{v(1)})$, $\texttt{Election}(D_{v(2)})$, ..., $\texttt{Election}(D_{v(l(V)+t)})$ elect a leader with probability at least $\frac{1}{2^t}$. Further, the $l(V) + t - 1$ calls runs in $v(1) + v(2) \cdots + v(l(V) + t - 1)$ time slots. Consequently, $\texttt{Election}(P(V))$ runs in $v(1) + v(2) \cdots + v(l(V) + \log f - 1)$ time slots with probability $1 - \frac{1}{f}$.

We conclude the following important lemma:

Lemma 2.4 *For any sequence* $V = \langle v(1), v(2), \ldots \rangle$, $\texttt{Election}(P(V))$ *elects a leader, with probability at least* $1 - \frac{1}{f}$ *for any* $f \geq 1$. *in* $v(1) + v(2) \cdots + v(l(V) + \log f - 1)$ *time slots.*

Let $V_1 = \langle 1, 2, 3, \ldots \rangle$ be a sequence of integers. We are going to evaluate the performance of $\texttt{Election}(P(V_1))$ using Lemma 2.4. Recall that

$$P(V_1) = D_1 \cdot D_2 \cdot D_3 \cdots = \langle \frac{1}{2}, \frac{1}{2}, \frac{1}{4}, \frac{1}{2}, \frac{1}{4}, \frac{1}{8}, \ldots \rangle.$$

Since $l(V_1) = \lceil \log n \rceil$, $\texttt{Election}(P(V_1))$ elects a leader with probability $1 - \frac{1}{f}$ in $O(1 + 2 + \cdots + (\log n + \log f - 1)) = O((\log n)^2 + (\log f)^2)$ time slots. Thus, we have the following lemma.

Lemma 2.5 *Protocol* $\texttt{Election}(P(V_1))$ *elects a leader in* $O((\log n)^2 + (\log f)^2)$ *time slots with probability at least* $1 - \frac{1}{f}$ *for any* $f \geq 1$.

For any fixed real number c ($1 < c < 2$), let $V_c = \langle \lceil c^0 \rceil, \lceil c^1 \rceil, \lceil c^2 \rceil, \ldots \rangle$ be a sequence of integers. Clearly, $l(V_c) \leq \lceil \frac{\log\log n}{\log c} \rceil$. Thus, from Lemma 2.4, $\texttt{Election}(P(V_c))$ elects a leader with probability $1 - \frac{1}{f}$ in $O(c^0 + c^1 + \cdots + c^{\lceil \frac{\log\log n}{\log c} \rceil + \log f}) = O(f^{\log c} \log n)$ time slots. Thus we have,

Lemma 2.6 *Oblivious protocol* $\texttt{Election}(P(V_c))$ *(*$1 < c < 2$*) elects a leader in* $O(f^{\log c} \log n)$ *time slots with probability at least* $1 - \frac{1}{f}$ *for any* $f \geq 1$.

For any two sequences $P = \langle p_1, p_2, \ldots \rangle$ and $P' = \langle p'_1, p'_2, \ldots \rangle$, let $P \oplus P' = \langle p_1, p'_1, p_2, p'_2, \ldots \rangle$ denotes the combined sequence of P and P'. We are going to evaluate the performance of $\texttt{Election}(P(V_1) \oplus P(V_c))$.

Let Z be a sequence of probabilities such that $Z = \langle 0, 0, 0, \ldots \rangle$. Clearly, $\texttt{Election}(P(V_1) \oplus Z)$ and $\texttt{Election}(Z \oplus P(V_c))$ run, with probability at least $1 - \frac{1}{f}$, in $O((\log n)^2 + (\log f)^2)$ and $O(f^{\log c} \log n)$ time slots, respectively from Lemmas 2.5 and 2.6. Thus, $\texttt{Election}(P(V_1) \oplus P(V_c))$ runs in $O(\min((\log n)^2 + (\log f)^2, f^{\log c} \log n))$ time slots. Therefore, we have,

Theorem 2.7 $\texttt{Election}(P(V_1) \oplus P(V_c))$ *elects a leader in* $O(\min((\log n)^2 + (\log f)^2, f^{\log c} \log n))$ *time slots with probability at least* $1 - \frac{1}{f}$ *for any* $f \geq 1$.

Note that for a fixed c such that $1 < c < 2$, we have $0 < \log c < 1$. Thus, by choosing small $\epsilon = \log c$, we have,

Corollary 2.8 *With probability at least* $1 - \frac{1}{f}$ *for any* $f \geq 1$, *oblivious protocol* `Election(`$P(V_1) \oplus P(V_c)$`)` *elects a leader in* $O(\min((\log n)^2 + (\log f)^2, f^\epsilon \log n))$ *for any fixed small* $\epsilon > 0$.

3 Uniform leader election protocols

The main purpose of this section is to discuss a uniform leader election protocol that terminates, with probability exceeding $1 - \frac{1}{f}$ for every $f \geq 1$, in $2 \log\log n + o(\log\log n) + O(\log f)$ time slots We begin by presenting a very simple protocol that is the workhorse of all subsequent leader election protocols.

Protocol `Broadcast(`p`)`
    every station broadcasts with probability $\frac{1}{2^p}$;
    if the status of the channel is SINGLE **then**
        the unique station that has transmitted
        becomes the leader.

In outline, our leader election protocol proceeds in three phases. In Phase 1 the calls `Broadcast(`2^{0^2}`)`, `Broadcast(`2^{1^2}`)`, `Broadcast(`2^{2^2}`)`, $\ldots$, `Broadcast(`2^{t^2}`)` are performed until, for the first time, the status of the channel is NULL in `Broadcast(`2^{t^2}`)`. At this point Phase 2 begins. Phase 2 executes a variant of binary search on the interval $[0, 2^{t^2}]$ using the protocol `Broadcast` as follows:

- First, `Broadcast(`$\frac{2^{t^2}}{2}$`)` is executed. If the status of the channel is SINGLE then the unique station that has transmitted becomes the leader.
- If the status of channel is NULL then binary search is performed on the interval $[0, \frac{2^{t^2}}{2}]$, that is, `Broadcast(`$\frac{2^{t^2}}{4}$`)` is executed.
- If the status of channel is COLLISION then binary search is performed on the interval $[\frac{2^{t^2}}{2}, 2^t]$, that is, `Broadcast(`$\frac{3}{4} \cdot 2^{t^2}$`)` is executed.

This procedure is repeated until, at some point, binary search cannot further split an interval. Let u be the integer such that the last call of Phase 2 is `Broadcast(`u`)`. Phase 3 repeats the call `Broadcast(`u`)` until, eventually, the status of the channel is SINGLE, at which point a leader has been elected. It is important to note that the value of u is continuously adjusted in Phase 3 as follows: if the status of the channel is NULL, then it is likely that 2^u is larger than n. Thus, u is decreased by one. By the same reasoning, if the status of the channel is COLLISION, u is increased by one.

With this preamble out of the way, we are now in a position to spell out the details of our uniform leader election protocol.

Protocol `Uniform-election`
Phase 1:
    $i \leftarrow -1$;
    repeat
        $i \leftarrow i + 1$;
        `Broadcast(`2^{i^2}`)`
    until the status of the channel is NULL;
Phase 2:
    $l \leftarrow 0$; $u \leftarrow 2^{i^2}$;
    while $l + 1 < u$ **do**
        $m \leftarrow \lceil \frac{l+u}{2} \rceil$;
        `Broadcast(`m`)`;
        if the status of channel is NULL **then**
            $u \leftarrow m$
        else
            $l \leftarrow m$
    endwhile
Phase 3:
    repeat
        `Broadcast(`u`)`;
        if the status of channel is NULL **then**
            $u \leftarrow \max(u - 1, 0)$
        else
            $u \leftarrow u + 1$
    forever

Phase 1 runs until the status of the channel is NULL in `Broadcast(`2^{t^2}`)`. With high probability, `Broadcast(`2^{t^2}`)`is NULL for the first time if $2^{2^{t^2}} > n > 2^{2^{(t-1)^2}}$. Thus, Phase 1 runs in $t = \sqrt{\log\log n} + O(1)$ time slots. Phase 2 does the binary search on $[0, 2^{t^2}]$. It follows that, Phase 2 performs `Broadcast(`m`)` for $\log 2^{t^2} = \log\log n + O(\sqrt{\log\log n})$ times. Phase 3 runs in $o(\log\log n) + O(\log f)$ time slots with probability $1 - \frac{1}{f}$. Thus, we have proved the following result.

Theorem 3.1 *There exists a uniform leader election protocol that terminates, with probability at least* $1 - \frac{1}{f}$, *in* $\log\log n + o(\log\log n) + O(\log f)$ *time slots, for every* $f \geq 1$.

The readers should refer to [6] for the details.

4 Non-uniform leader election protocol

The main goal of this section is to present a non-uniform leader election protocol for single-hop, single-channel radio networks. The workhorse of the non-uniform leader election protocol is Protocol `Sieve`.

Protocol `Sieve`(p)
every active station broadcasts with probability $\frac{1}{2^{2^p}}$;
if the status of the channel is SINGLE **then**
  the station that has transmitted becomes the leader.
else if the status of the channel is COLLISION
  then the stations that have not transmitted
  become inactive
endif

Using `Sieve`(p), we show a non-uniform leader election protocol terminating in $\log\log n + 2.78\log f + o(\log\log n + \log f)$ time slots for every $f \geq 1$. In outline, our leader election protocol proceeds in three phases. In Phase 1 the calls `Sieve`(0^2), `Sieve`(1^2), `Sieve`(2^2), ..., `Sieve`(t^2) are performed until, for the first time, the status of the channel is NULL in `Sieve`(t^2). At this point Phase 2 begins. In Phase 2 we perform the calls `Sieve`(t^2-1), `Sieve`(t^2-2), ..., `Sieve`(0). Finally, Phase 3 repeats the call `Sieve`(0) until, eventually, the status of the channel is SINGLE, at which point a leader has been elected.

With this preamble out of the way, we are now in a position to spell out the details of our leader election protocol.

Protocol `Non-uniform-election`
initially all the stations are active;
Phase 1:
  for $i \leftarrow 0$ **to** ∞
    `Sieve`(i^2);
    exit for-loop if the status is NULL;
Phase 2:
  $t \leftarrow i^2 - 1$;
  for $i \leftarrow t$ **downto** 0
    `Sieve`(i);
Phase 3:
  repeat
    `Sieve`(0);
  forever

Phase 1 runs until the status of the channel is NULL in `Sieve`(i^2). With high probability, `Sieve`(i^2) is NULL for the first time if $2^{2^{i^2}} > n > 2^{2^{(i-1)^2}}$. Thus, Phase 1 runs in $i = O(\sqrt{\log\log n})$ time slots. Phase 2 performs binary search on $[0, 2^{i^2}]$. It follows that, Phase 2 performs `Broadcast`(m) for $\log 2^{i^2} = \log\log n + O(\sqrt{\log\log n})$ times. Phase 3 runs in $o(\log\log n) + 4\ln f \leq o(\log\log n) + 2.78\ln f$ time slots with probability $1 - \frac{1}{f}$. Thus, we have proved the following result.

Theorem 4.1 *There exists a non-uniform leader election protocol terminating in* $\log\log n + 2.78\log f + o(\log\log n + \log f)$ *time slots with probability at least* $1 - \frac{1}{f}$ *for any* $f \geq 1$.

The reader should refer to [4] for the details.

References

[1] J. Bentley and A. Yao, An almost optimal algorithm for unbounded search, *Information Processing Letters*, 5, (1976), 82–87.

[2] R. M. Metcalfe and D. R. Boggs, Ethernet: distributed packet switching for local computer networks, *Communications of the ACM*, 19, (1976), 395–404.

[3] K.Nakano and S.Olariu, Randomized $O(\log\log n)$-round leader election protocols in radio networks, *Proc. of International Symposium on Algorithms and Computation*, (LNCS 1533), 209–218, 1998.

[4] K. Nakano and S. Olariu, Randomized leader election protocols for ad-hoc networks, *Proc. of Sirocco 7*, June 2000, 253–267.

[5] K. Nakano and S. Olariu, Randomized leader election protocols in radio networks with no collision detection, *Proc. of International Symposium on Algorithms and Computation*, 362–373, 2000.

[6] K. Nakano and S. Olariu, Uniform leader election protocols for radio networks, Proc. of International Conference on Parallel Processing, 240-249, 2001.

[7] D. E. Willard, Log-logarithmic selection resolution protocols in a multiple access channel, *SIAM Journal on Computing*, 15, (1986), 468–477.