Randomized Leader Election Protocols in Radio Networks with no Collision Detection *

Koji Nakano[1] and Stephan Olariu[2]

[1] Department of Electrical and Computer Engineering, Nagoya Institute of Technology, Showa-ku, Nagoya 466-8555, Japan
[2] Department of Computer Science, Old Dominion University, Norfolk, VA 23529, USA

Abstract. The main contribution of this work is to propose energy-efficient randomized leader election protocols for single-hop, single-channel radio networks (RN) that do not have collision detection (CD) capabilities. We first presents a leader election protocol for the case the number n of stations is known beforehand. The protocol runs in $O(\log f)$ time slots with no station being awake for more than $O(\log\log f + \frac{\log f}{\log n})$ time slots with probability at least $1 - \frac{1}{f}$ for any $f \geq 1$. We then present three leader election protocols for the case where n is not known beforehand. The first protocol terminates, with probability exceeding $1 - \frac{1}{f}$, in $O((\log n)^2 + (\log f)^2)$ time slots, with no station being awake for more than $O(\log n + \log f)$ time slots. Clearly, this first protocol terminates in $O((\log n)^2)$ expected time slots. Our second protocol reduces the expected termination time to $O(\log n)$ time slots. This second protocol terminates, with probability exceeding $1 - \frac{1}{f}$ in $O(f^{\frac{3}{5}} \log n)$ time slots. Finally, by combining these two protocols, we obtain a third leader election protocol that terminates in $O(\log n)$ expected time slots. This latter protocol terminates, with with probability exceeding $1 - \frac{1}{f}$, in $O(\min((\log n)^2 + (\log f)^2, f^{\frac{3}{5}} \log n))$ time slots, with no station being awake for more than $O(\log n + \log f)$ time slots.

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*. We assume that the stations are identical and cannot be distinguished by serial or manufacturing number. As customary, time is assumed to be slotted and all the stations have a local clock that keeps synchronous time, perhaps by interfacing with a GPS system. The stations are assumed to have the computing power of a usual laptop computer; in particular, they all run the same protocol and can generate random bits that provide local data on which the stations may perform computations.

* Work supported in part by ONR grant N00014-91-1-0526 and by Grant-in-Aid for Encouragement of Young Scientists (12780213) from Ministry of Education, Science, Sports, and Culture of Japan.

We assume that the stations run on batteries and, therefore, saving battery power is exceedingly important, as recharging batteries may not be possible while on mission. It is well known that a station expends power while its transceiver is active that is, while transmitting or receiving a packet [1–3, 9–11]. Consequently, we are interested in developing protocols that allow stations to power their transceiver off (i.e. go to sleep) to the largest extent possible. Accordingly, we judge the goodness of a protocol by the following two yardsticks:

- the overall number of time slots required by the protocol to terminate, and
- for each individual station the total number of time slots when it has to be awake in order to transmit/receive packets.

The goals of optimizing these parameters are, of course, conflicting. It is relatively straightforward to minimize overall completion time at the expense of energy consumption. Similarly, one can minimize energy consumption at the expense of completion time [11–13]. The challenge is to strike a sensible balance between the two, by designing protocols that take a small number of time slots to terminate while being, at the same time, as energy-efficient as possible.

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 lost. In terms of their collision detection capabilities, the RNs come in two flavors. In the RN with *collision detection* (CD) the status of the channel is:

NULL: if no station transmitted on the channel in the current time slot,
SINGLE: if one station transmitted on the channel in the current time slot,
COLLISION: if two or more stations transmitted in the current time slot.

In the RN with no collision detection (no-CD) the status of a radio channel is:

NOISE: if either no station transmitted or two or more stations transmitted in the current time slot, and
SINGLE: if one station transmitted in the current time slot.

In other words, the RN with no-CD cannot distinguish between no transmissions on the channel and the result of two or more stations transmitting at the same time.

The *leader election* problem asks to designate one of the stations as *leader*. In other words, after performing the leader election protocol, exactly one station learns that it was elected leader, while the remaining stations learn the identity of the leader elected. The leader election problem is fundamental, for many protocols rely directly or indirectly on the presence of a leader in a network [7, 14]. Further, once a leader is available, the RN with CD can be simulated by the RN with no-CD with a constant factor slowdown [4].

It is customary to address the leader election problem on the RN in two different scenarios:

known n: Every station knows in advance the number n of stations;

unknown n: The number n of stations is not known beforehand.

It is intuitively clear that the task of leader election for known n is the easiest.

Several randomized protocols for single-channel RNs have been presented in the literature. Metcalfe and Boggs [5] presented a simple leader election protocol for the RN with no-CD for known n that is guaranteed to terminate in $O(1)$ expected rounds. For unknown n, several protocols have been proposed for the RN with CD and no-CD. Willard [14] showed that the leader election on the RN with CD can be solved in $\log\log n + o(\log\log n)$ expected time slots. Later, Nakano and Olariu [7] presented two leader election protocols for the RN with CD that terminate in $O(\log n)$ time slots with probability at least $1 - \frac{1}{n}$ and in $O(\log\log n)$ time slots with probability at least $1 - \frac{1}{\log n}$. Recently, Nakano and Olariu [8] improved the protocol of [7] showing that the leader election on the RN with CD can be performed 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 every $f \geq 1$. Hayashi *et al.* [4] proposed a leader election protocol for the RN with no-CD that terminates in $O((\log n)^2)$ time slots with probability at least $1 - \frac{1}{n}$.

The main contribution of this work is to propose energy-efficient randomized leader election protocols for single-hop, single-channel radio networks that do not have collision detection capabilities. Consider an arbitrary parameter $f \geq 1$. We begin by showing that a leader can be elected among n identical stations, with probability exceeding $1 - \frac{1}{f}$, in $O(\log f)$ time slots, with no station being awake for more than $O(\log\log f + \frac{\log f}{\log n})$ time slots, provided that the number n of stations is known beforehand.

Turning to the more realistic case where n is not known beforehand, we first discuss a leader election protocol that terminates, with probability exceeding $1 - \frac{1}{f}$, in $O((\log n)^2 + (\log f)^2)$ time slots, with no station being awake for more than $O(\log n + \log f)$ time slots. This first protocol terminates in $O((\log n)^2)$ expected time slots. Next, we propose a second protocol that reduces the expected termination time to $O(\log n)$ time slots. This second protocol terminates, with probability exceeding $1 - \frac{1}{f}$ in $O(f^{\frac{3}{5}} \log n)$ time slots. Finally, by combining the two protocols above, we obtain a third protocol that terminates in $O(\log n)$ expected time slots. This latter protocol terminates, with with probability exceeding $1 - \frac{1}{f}$, in $O(\min((\log n)^2 + (\log f)^2, f^{\frac{3}{5}} \log n))$ time slots, with no station being awake for more than $O(\log n + \log f)$ time slots.

2 Deterministic leader election on RNs with IDs

This section is devoted to proposing a simple deterministic leader election protocol for RNs in which every station has a unique ID. This protocol is a key ingredient in our energy-efficient leader election protocols.

Assume that each of the n, $(n \leq m)$, stations of the RN has a unique ID in the range $[1, m]$ and let S_i, $(1 \leq i \leq m)$, denote the station with ID i. Note that S_i may not exist for some i. We are interested in choosing the station with minimum ID as leader. The following simple protocol performs this task in m time slots. In time slot i, $(1 \leq i \leq m)$, station S_i announces its existence by transmitting on the channel. If the status of the channel is SINGLE then i is the minimum ID and every station terminates the protocol. Clearly, this protocol requires every station to monitor the channel in all the m time slots, and hence, is not energy-efficient. We can reduce the number of awake time slots

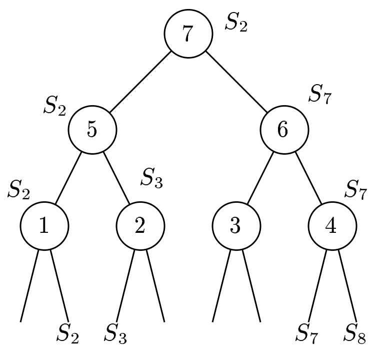

Fig. 1. *Illustrating the protocol for finding station with minimum ID.*

per station by using a binary tree with m leaves as illustrated in Figure 1. The leaves corresponds to the stations. The internal nodes of the binary tree are numbered as indicated in Figure 1. The idea is to traverse the internal nodes in increasing order, identifying for each of them the station with minimum ID in the subtree rooted at that node. The details are spelled out as follows.

Protocol `Election-with-ID`
for $i \leftarrow 1$ **to** $m-1$ **do**
  the minimum ID station in the left subtree of node i sends on the channel;
  the minimum ID station in the right subtree of node i monitors the channel;
  if the status of the channel is SINGLE **then**
    the station with minimum ID in the left subtree becomes
    the station with minimum ID in the subtree rooted at node i;
  else
    the station with minimum ID in the right subtree becomes
    the station with minimum ID in the subtree rooted at node i;
endfor
the station with minimum ID in the tree is declared the leader.

The reader should have no difficulty to confirm that when the protocol `Election-with-ID` terminates the station with minimum ID is elected the leader. Since the tree has exactly $m-1$ internal nodes, protocol `Election-with-ID` always terminates in $m-1$ time slots and each station is awake for at most a number of time slots equal to the height of the binary tree. Thus, we have proved the following result.

Lemma 1. *Assume that each of the n stations of a single-hop, single-channel radio network has a unique ID in the range $[1, m]$, $(n \leq m)$. A leader can be elected among these n-stations in $m - 1$ time slots, with no station being awake for more than $\log m$ time slots.*

3 A refresher of basic probability theory

This section offers a quick review of basic probability theory results that are useful for analyzing the performance of our randomized leader election protocols. For a more detailed discussion of background material we refer the reader to [6].

Throughout, $\Pr[A]$ will denote the probability of event A. For a random variable X, $E[X]$ denotes the expected value of X. Let X be a random variable denoting the number of successes in n independent Bernoulli trials with parameters p and $1 - p$. It is well known that X has a *binomial distribution* and that for every r, $(0 \leq r \leq n)$,

$$\Pr[X = r] \;=\; \binom{n}{r} p^r (1-p)^{n-r}.$$

Further, the expected value of X is given by

$$E[X] = \sum_{r=0}^{n} r \cdot \Pr[X = r] = np.$$

To analyze the tail of the binomial distribution, we shall make use of the following estimate, commonly referred to as *Chernoff bound*:

$$\Pr[X > (1+\delta)E[X]] < \left(\frac{e^{\delta}}{(1+\delta)^{(1+\delta)}}\right)^{E[X]}. \tag{1}$$

For later reference, we state the following result.

Lemma 2. *Let X be a random variable taking on a value smaller than or equal to $T(F)$ with probability at most F, $(0 \leq F \leq 1)$, where T is a non-decreasing function. Then, $E[X] \leq \int_0^1 T(F)dF$.*

4 Randomized leader election for known n

The main goal of this section is to provide leader election protocols for radio networks where the number n of stations in known beforehand. The protocol discussed in Subsection 4.1 is energy-inefficient. In Subsection 4.2 we show how it can be made energy-efficient.

4.1 An energy-inefficient protocol

Let $P = \langle p_1, p_2, p_3, \ldots \rangle$ be a 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 SINGLE. The details are spelled out in the following protocol.

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

Since correctness is easy to see, we now turn to evaluating the number of time slots it takes protocol `Election(`P`)` to terminate. Let X be the random variable denoting the number of stations that transmit in the i-th time slot. Then, the status of the channel is SINGLE with probability

$$\Pr[X = 1] = \binom{n}{1} p_i (1 - p_i)^{n-1}$$

To maximize the probability $\Pr[X = 1]$, we choose $p_i = \frac{1}{n}$ and so

$$\Pr[X = 1] = \left(1 - \frac{1}{n}\right)^{n-1} > \frac{1}{e}.$$

Therefore, if every station knows n, we choose $P = \langle \frac{1}{n}, \frac{1}{n}, \frac{1}{n}, \ldots \rangle$. Thus, each iteration of the **for** loop of `Election(`$P = \langle \frac{1}{n}, \frac{1}{n}, \frac{1}{n}, \ldots \rangle$`)` succeeds in electing a leader with probability at least $\frac{1}{e}$. Suppose that the first t, $(t \geq 0)$, iterations of the **for** loop fail to elect a leader. Since each such trial is independent, the probability of this occurring is at most $(1 - \frac{1}{e})^t \leq e^{-\frac{t}{e}}$. It follows that with probability exceeding $1 - e^{-\frac{t}{e}}$ the protocol elects a leader in at most t time slots. For f satisfying $f = e^{\frac{t}{e}}$, the protocol terminates, with probability exceeding $1 - \frac{1}{f}$, in $e \ln f$ time slots. By Lemma 2, the expected number of time slots taken by the protocol to terminate is bounded by $\int_0^1 e \ln \frac{1}{F} dF = e$. Thus, we have the following result.

Lemma 3. *Protocol* `Election(`$\langle \frac{1}{n}, \frac{1}{n}, \frac{1}{n}, \ldots \rangle$`)` *terminates in at most* e *expected time slots. Also, for every* $f \geq 1$*, the protocol terminates, with probability exceeding* $1 - \frac{1}{f}$*, in* $O(\log f)$ *time slots.*

4.2 An energy-efficient protocol

Note that protocol `Election` is energy-inefficient since with probability exceeding $1 - \frac{1}{f}$, each station has to be awake for $O(\log f)$ time slots to monitor the channel. The main goal of this subsection is to reduce the awake time to $O(\log\log f + \frac{\log f}{\log n})$ time slots.

The idea is simple: in every time slot only the transmitting stations monitor the channel. The others don't. However, if a station never monitors the channel, it cannot know whether or not a leader was elected. The new protocol identifies *candidates* to becoming the leader and selects one of them using protocol `Election-with-ID` discussed in Section 2. Let $Q = \langle p_1, p_2, \ldots p_{|Q|}\rangle$ be a finite sequence of probabilities, where $|Q|$ denotes the number of elements in Q.

Protocol `EPS`(Q)
if $|Q| = 1$ **then** perform `Election`(Q);
else
  for $i \leftarrow 1$ to $|Q|$ do
    every non-candidate station transmits on the channel with probability p_i;
    (Note: only the transmitting stations monitor the channel)
    if the status of the channel is SINGLE **then**
      the unique station that has transmitted becomes a *candidate* with ID i
  endfor
  use protocol `Election-with-ID` to elect a leader from the candidates;
  the leader transmits on the channel to announce its identity and
  every station monitors the channel to learn the identity of the leader; $(\star)$

Note that since only the candidates are awake to run protocol `Election-with-ID`, the non-candidates are not aware of the election of the leader, if any. Thus, the last transmission in $(\star)$ is necessary to ensure the correctness of the protocol.

If $|Q| = 1$, protocol `EPS`(Q) runs in one time slot. Otherwise, the **for** loop takes $|Q|$ time slots, and by Lemma 1, `Election-with-ID` terminates in $|Q| - 1$ time slots. Also, as discussed, one final time slot $(\star)$ is necessary for non-candidate stations to learn the identity of the leader, if any. Thus, for $|Q| \geq 2$, this protocol always takes $2|Q|$ time slots. Notice that, by Lemma 1, each candidate is awake for at most $\log |Q|$ time slots to participate in protocol `Election-with-ID`.

Given an infinite sequence $P = \langle p_1, p_2, p_3, \ldots\rangle$ of probabilities, partition P into subsequences $P_1, P_2, \ldots$ such that $P_i = \langle p_{2^{i-1}}, p_{2^{i-1}+2}, \ldots, p_{2^i-1}\rangle$ for every i, $(i \geq 1)$. Thus, $P_1 = \langle p_1\rangle$, $P_2 = \langle p_2, p_3\rangle$ $P_3 = \langle p_4, p_5, p_6, p_7\rangle$, …, and clearly, $|P_i| = 2^{i-1}$. The following protocol uses `EPS` as a building-block.

Protocol `Election-PS`(P)
for $i \leftarrow 1$ to $+\infty$ do
  perform `EPS`(P_i);
  if a leader was elected **then** the protocol terminates
endfor

Since n is known, the best is to execute `Election-PS`($\langle \frac{1}{n}, \frac{1}{n}, \frac{1}{n}, \ldots, \rangle$). Instead of evaluating the performance of `Election-PS`($\langle \frac{1}{n}, \frac{1}{n}, \frac{1}{n}, \ldots\rangle$), we will discuss the relation between `Election`(P) and `Election-PS`(P) for an arbitrary sequence $P = \langle p_1, p_2, p_3, \ldots\rangle$ of probabilities. We then evaluate the performance of `Election-PS`$(\langle \frac{1}{n}, \frac{1}{n}, \frac{1}{n}, \ldots\rangle)$.

It is clear that for every $T \geq 1$, the following two events have the same probability:

Event1(T) `Election`(P) elects a leader when the stations transmit with probability p_T in the T-th trial, and

Event2(T) `Election-PS`(P) elects the first candidate when the stations transmit with probability p_T in the T-th trial.

Actually, each of the two events occurs if

- for $1 \leq t \leq T-1$ when the stations transmit with probability p_t the status of the channel is NOISE, and
- the status of the channel is SINGLE when the stations transmit with probability p_T.

Observe that if Event1(T) occurs, then `Election`(P) terminates in T time slots. If Event2(T) occurs, the number of time slots taken by `Election`(P) to terminate can be evaluated as follows. Let s be the integer satisfying

$$2^0 + 2^1 + \cdots + 2^{s-1} < T \leq 2^0 + 2^1 + \cdots + 2^s.$$

In this case, `EPS`(P_1), `EPS`(P_2), ..., `EPS`(P_{s-1}) fail to elect a leader, while `EPS`(P_s) succeeds in finding a leader. Now, `Election-PS`(P) takes at most

$$1 + (2^2 + \cdots + 2^{s+1}) < 4T$$

time slots, because `Election-PS`(P_s) terminates in $2|P_s| = 2^{s+1}$ time slots for $s \geq 2$ and `Election-PS`(P_1) takes one time slot. Therefore, if `Election`(P) elects a leader in $g(n, f)$ time slots with probability $1 - \frac{1}{f}$ then `Election-PS`(P) elects a leader in at most $4g(n, f)$ time slots with the same probability. Thus, we have the following result.

Lemma 4. *If protocol* `Election`(P) *terminates with probability at least* $1 - \frac{1}{f}$ *in* $g(n, f)$ *time slots with probability* $1 - \frac{1}{f}$ *for any* $f \geq 1$, *then* `Election-PS`(P) *terminates, with probability at least* $1 - \frac{1}{f}$ *in* $4g(n, f)$ *time slots with probability* $1 - \frac{1}{f}$.

We now turn to the task of evaluating the maximum number of awake time slots over all stations. Suppose that `Election-PS`(P) calls `EPS`(P_1), `EPS`(P_2), ..., `EPS`(P_{s+1}), and `EPS`(P_{s+1}) succeeds in electing a leader. Further, let $T' = 2^0 + 2^1 + \cdots + 2^s + 2^{s+1}$ be the number of trials involved in `Election-PS`(P). In other words, a station transmits with probability p_i, $(1 \leq i \leq T')$, in the i-th trial of `Election-PS`(P). Clearly, every station monitors the channel in one time slot for each `EPS`(P_i), $(1 \leq i \leq s+1)$, to check the existence of the leader. Also, the candidates transmit/monitor the channel at most s times to find the candidate with minimum ID in `EPS`(P_{s+1}). Note that, by assumption, no candidate has been identified in `EPS(`P_1`)`, `EPS(`P_2`)`, ..., `EPS(`P_s`)`. Thus, for these two tasks, every station transmits/monitors the channel for at most $2(s+1) = O(\log T')$ time slots.

Next, we will evaluate the number of awake time slots for T' trials, in each of which the stations transmit/monitor with probability p_i, $(1 \le i \le T')$. Let $\mathrm{SUM}_i(P)$ denote the prefix sum $p_1 + p_2 + \cdots + p_i$ of P. For a particular station, let X be the random variable denoting the number of awake time slots in these T' trials. Clearly, $E[X] = \mathrm{SUM}_{T'}(P) = p_1 + p_2 + \cdots + p_{T'}$ and from the Chernoff bound (1), for every $\delta > 0$, we have

$$\Pr[X > (1+\delta)\mathrm{SUM}_{T'}(P)] < \left(\frac{e^{\delta}}{(1+\delta)^{(1+\delta)}}\right)^{\mathrm{SUM}_{T'}(P)} < \left(\frac{e}{1+\delta}\right)^{(1+\delta)\mathrm{SUM}_{T'}(P)}.$$

Let α be a function satisfying

$$\left(\frac{ex}{\alpha(x)}\right)^{\alpha(x)} = \frac{1}{nf}. \tag{2}$$

By choosing δ to satisfy $(1+\delta)\mathrm{SUM}_{T'}(P) = \alpha(\mathrm{SUM}_{T'}(P))$, we have

$$\Pr[X > \alpha(\mathrm{SUM}_{T'}(P))] < \left(\frac{e\mathrm{SUM}_{T'}(P)}{\alpha(\mathrm{SUM}_{T'}(P))}\right)^{\alpha(\mathrm{SUM}_{T'}(P))} \le \frac{1}{nf}.$$

It follows that with probability at most $\frac{1}{nf}$, a particular station is awake for more than $\alpha(\mathrm{SUM}_{T'}(P))$ time slots in the first T' trials. Hence, with probability at least $1 - \frac{1}{f}$, none of the n stations is awake for more than $\alpha(\mathrm{SUM}_{T'}(P))$ time slots in the first T' trials. Thus, if `Election-PS(`P`)` finds a candidate in T' time slots with probability $1 - \frac{1}{f}$, then with probability at least $1 - \frac{1}{f}$ no station is awake for more than $O(\alpha(\mathrm{SUM}_{T'}(P))$ time slots.

Recall that every station must be awake for $O(\log T')$ time slots to find the minimum ID candidate and to detect the existence of the leader. To summarize our discussion, we state the following result.

Lemma 5. *If* `Election-PS(`P`)` *terminates, with probability* $1-\frac{1}{f}$ *for any* $f \ge 1$, *in* T' *time slots, then with probability* $1 - \frac{1}{f}$ *no station is awake for more than* $O(\log T' + \alpha(\mathrm{SUM}_{T'}(P)))$ *time slots, where* α *is a function satisfying (2).*

Let us now evaluate the number of awake time slots for `Election-PS(`P`)` with $P = \langle \frac{1}{n}, \frac{1}{n}, \frac{1}{n}, \ldots \rangle$. By Lemmas 3 and 4 protocol `Election-PS(`P`)` terminates, with probability $1 - \frac{1}{f}$, in $O(\log f)$ time slots. Hence, we apply $T' = O(\log f)$ to Lemma 5. Since $\mathrm{SUM}_{O(\log f)}(P) = O(\frac{\log f}{n})$ and and $\alpha(\mathrm{SUM}_{O(\log f)}(P)) = O(\frac{\log nf}{\log n}) = O(\frac{\log f}{\log n} + 1)$, no station is awake for more than

$$O(\log T' + \alpha(\mathrm{SUM}_{T'}(P))) = O(\log\log f + \frac{\log f}{\log n})$$

time slots. Thus, we have proved the following result.

Theorem 1. *Protocol* `Election-PS`($\langle \frac{1}{n}, \frac{1}{n}, \frac{1}{n}, \ldots \rangle$) *takes $O(1)$ expected time slots, as well as $O(\log f)$ time slots with no station being awake for more than $O(\log\log f + \frac{\log f}{\log n})$ time slots with probability at least $1 - \frac{1}{f}$ for any $f \geq 1$.*

5 Randomized leader election for unknown n

The purpose of this section is to discuss three randomized leader election protocols for the n-station RN under the assumption that the number n of stations is not know beforehand.

5.1 Electing a leader in $O((\log n)^2 + (\log f)^2)$ time slots

Let U_i $(1 \geq 1)$ be the sequence of probabilities

$$U_i = \langle \frac{1}{2^1}, \frac{1}{2^2}, \ldots, \frac{1}{2^i} \rangle$$

We have the following lemma.

Lemma 6. *Protocol* `Election`(U_i) *succeeds in electing a leader with probability at least $\frac{3}{5}$ whenever $i \geq \log n > 2$.*

Due to the page limitation, we omit the proof.

Let $U = U_1 \cdot U_2 \cdot U_3 \cdots = \langle \frac{1}{2}, \frac{1}{2}, \frac{1}{4}, \frac{1}{2}, \frac{1}{4}, \frac{1}{8}, \ldots \rangle$ be the infinite sequence of probabilities, where "$\cdot$" denotes the concatenation of sequences. We are going to evaluate the performance `Election-PS`(U). From Lemma 4, it is sufficient to evaluate the performance `Election`(U) instead. Let s be an integer satisfying $2^{s-1} < n \leq 2^s$. Suppose that `Election`($U_1 \cdot U_2 \cdots U_{s+t-1}$) for some $t \geq 1$ is executed. Essentially, this contains the execution of `Election`(U_s), `Election`(U_{s+1}), ..., `Election`(U_{s+t-1}), each of them succeeds in electing a leader with probability at least $\frac{3}{5}$ from Lemma 6. Thus, `Election`($U_1 \cdot U_2 \cdots U_{s+t-1}$) fails in electing a leader with probability at most $(\frac{2}{5})^t$. Since $U_1 \cdot U_2 \cdots U_{s+1+t}$ contains $1 + 2 + \cdots + (s + t - 1) = O((s + t)^2)$ trials, `Election`(U) elects a leader in $O((s + t)^2) = O((\log n)^2 + t^2)$ time slots with probability at least $1 - (\frac{2}{5})^t$. Let $f = (\frac{5}{2})^t$. Then, `Election`(U) elects a leader in $O((\log n)^2 + (\log f)^2)$ with probability $1 - \frac{1}{f}$.

Next, we will evaluate the awake time slots of `Election-PS`(U). For this purpose, let $T' = O((\log n)^2 + (\log f)^2)$ and apply Lemma 5 to `Election-PS`(U). Since $\mathrm{SUM}_{T'}(U) = O(\log n + \log f)$, we have $\alpha(\mathrm{SUM}_{T'}(U)) = O(\log n + \log f)$. Thus, we have the following lemma.

Lemma 7. *Protocol* `Election-PS`(U) *elects a leader in $O((\log n)^2 + (\log f)^2)$ time slots with no station is being awake for more than $O(\log n + \log f)$ with probability at least $1 - \frac{1}{f}$ for any $f \geq 1$.*

Lemmas 2 and 7 combined, this protocol runs in $\int_0^1 O((\log n)^2 + (\log \frac{1}{F})^2) dF = O(\log^2 n)$ expected time slots

5.2 Electing a leader in $O(\log n)$ expected time slots

Let $V = U_{2^0} \cdot U_{2^1} \cdot U_{2^2} \cdot U_{2^3} \cdots$ be the infinite sequence of probabilities. We are going to show that `Election(`V`)` elects a leader in $O(\log n)$ expected time slots. It follows that, `Election-PS(`V`)` elects a leader in $O(\log n)$ expected time slots from Lemma 4.

Again, let s be an integer satisfying $2^{s-1} < n \leq 2^s$. Further, let J be the minimum integer satisfying $2^J \geq s$. Clearly, $2^J = O(\log n)$. Essentially, `Election(`V`)` involves `Election(`U_{2^0}`)`, `Election(`U_{2^1}`)`, `Election(`$U_{2^{J-1}}$`)` before executing `Election(`U_{2^J}`)`. Thus, it has spent $2^0 + 2^1 + \cdots + 2^J = 2^J - 1$ time slots before executing `Election(`U_{2^J}`)`. After that, `Election(`U_{2^J}`)`, `Election(`$U_{2^{J+1}}$`)`, ..., are executed, and each of them has ability of electing a leader with probability at least $\frac{3}{5}$ from Lemma 6. Thus, `Election(`$U_{2^{J+k}}$`)` $(k \geq 0)$ is executed if all of the execution of `Election(`U_{2^0}`)`, `Election(`U_{2^1}`)`, `Election(`$U_{2^{J+k-1}}$`)` fails electing a leader, and this probability is at most $(\frac{2}{5})^k$. Therefore, the expected number of time slots that `Election(`V`)` runs is at most

$$2^J - 1 + \sum_{k=0}^{\infty} |U_{2^{J+k}}|(\frac{2}{5})^k = O(\log n) + 2^J \sum_{k=0}^{\infty} (\frac{4}{5})^k = O(\log n).$$

Therefore, we have,

Lemma 8. *Protocol* `Election-PS(`V`)` *elects a leader in* $O(\log n)$ *expected time slots.*

However, with probability $\frac{1}{f}$, `Election-PS(`V`)` runs in $O(f^{\frac{3}{5}} \log n)$ time slots. This can be proved as follows: Since `Election(`$U_{2^{J+k}}$`)` for $k \geq 0$, elects a leader with probability $\frac{3}{5}$, $K = \frac{\log f}{\log \frac{5}{3}}$ calls `Election(`U_{2^J}`)`, `Election(`$U_{2^{J+1}}$`)`, ..., `Election(`$U_{2^{J+K}}$`)` finds a leader with probability at least $1 - (\frac{3}{5})^K = 1 - \frac{1}{f}$. This K calls takes $2^0 + 2^1 + \cdots + 2^{J+K} - 1 = O(f^{\frac{3}{5}} \log n)$ time slots.

5.3 Combining randomized leader election protocols

Let U and V be the sequences of probabilities defined in the previous subsections and u_i and v_i be the i-th probabilities in $U = \langle u_1, u_2, \ldots \rangle$ and $V = \langle v_1, v_2, \ldots, \rangle$, respectively. Let W be the combined sequence of U and V such that $W = \langle u_1, v_1, u_2, v_2, u_3, v_3, \ldots \rangle$. In this subsection, we will evaluate the performance of `Election-PS(`W`)`.

Let $U' = \langle u_1, 0, u_2, 0 \ldots \rangle$ and $V' = \langle 0, v_1, 0, v_2, \ldots \rangle$. Clearly, `Election-PS(`$U'$`)` and `Election-PS(`$V'$`)` run, with probability at least $1 - \frac{1}{f}$, in $O((\log n)^2 + (\log f)^2)$ and $O(f^{\frac{3}{5}} \log n)$ time slots, respectively. Thus, `Election-PS(`W`)` runs in $O(\min((\log n)^2 + (\log f)^2, f^{\frac{3}{5}} \log n))$ time slots.

Next we will evaluate the awake time slots. For this purpose, let $T' = O((\log n)^2 + (\log f)^2)$ and apply Lemma 5 for `Election-PS(`W`)` as follows:

$$\begin{aligned} \mathrm{SUM}_{T'}(W) &< \mathrm{SUM}_{T'}(U) + \mathrm{SUM}_{T'}(V) \\ &= O(\log n + \log f) + O(\log\log n + \log\log f) \\ &= O(\log n + \log f). \end{aligned}$$

Thus, from Lemma 5, `Election-PS(`W`)` runs with stations being awake for at most $O(\log(\min((\log n)^2+(\log f)^2, f^{\frac{3}{5}}\log n))+\alpha(\mathrm{SUM}_{T'}(W)) = O(\log n+\log f)$ time slots with probability at least $1-\frac{1}{f}$. Therefore, we have,

Theorem 2. *Protocol* `Election-PS(`W`)` *elects a leader in* $O(\log n)$ *expected time slots, as well as* $O(\min((\log n)^2 + (\log f)^2, f^{\frac{3}{5}}\log n))$ *time slots with no station being awake for more than* $O(\log n + \log f)$ *time slots with probability at least* $1-\frac{1}{f}$ *for any* $f \geq 1$.

References

1. K. Feher, *Wireless Digital Communications*, Prentice-Hall, Upper Saddle River, NJ, 1995.
2. W. C. Fifer and F. J. Bruno, Low cost packet radio, *Proceedings of the IEEE*, 75, (1987), 33–42.
3. E. P. Harris and K. W. Warren, Low power technologies: a system perspective, *Proc. 3-rd International Workshop on Multimedia Communications*, Princeton, 1996.
4. T. Hayashi, K. Nakano, and S. Olariu, Randomized initialization protocols for Packet Radio Networks, *Proc. 13th International Parallel Processing Symposium*, (1999), 544–548.
5. R. M. Metcalfe and D. R. Boggs, Ethernet: distributed packet switching for local computer networks, *Communications of the ACM*, 19, (1976), 395–404.
6. R. Motwani and P. Raghavan, *Randomized Algorithms*, Cambridge University Press, 1995.
7. K. Nakano and S. Olariu, Randomized $O(\log\log n)$-round leader election protocols in Packet Radio Networks, *Proc. of International Symposium on Algorithms and Computation*, (LNCS 1533), 209–218, 1998.
8. K. Nakano and S. Olariu, Randomized Leader Election Protocols for Ad-hoc Networks to appear in *Proc. of SCIROCCO* (2000).
9. R. A. Powers, Batteries for low-power electronics, *Proceedings of the IEEE*, 83, (1995), 687–693.
10. A. K. Salkintzis and C. Chamzas, An in-band power-saving protocol for mobile data networks, *IEEE Transactions on Communications*, COM-46, (1998), 1194–1205.
11. K. Sivalingam, M. B. Srivastava, and P. Agrawal, Low power link and access protocols for wireless multimedia networks, *Proc. IEEE Vehicular Technology Conference VTC'97*, Phoenix, AZ, May, 1997.
12. M. Stemm, P. Gauthier, and D. Harada, Reducing power consumption on network interfaces in hand-held devices, *Proc. 3-rd International Workshop on Multimedia Communications*, Princeton, 1996.
13. J. E. Wieselthier, G. D. Nguyen, and A. Ephemerides, Multicasting in energy-limited ad-hoc wireless networks, *Proc. MILCOM'98*, 1998.
14. D. E. Willard, Log-logarithmic selection resolution protocols in a multiple access channel, *SIAM Journal on Computing*, 15, (1986), 468–477.