Quasi-Optimal Leader Election Algorithms in Radio Networks with Loglogarithmic Awake Time Slots
Christian Lavault, Jean-François Marckert, Vlady Ravelomanana

To cite this version:
Christian Lavault, Jean-François Marckert, Vlady Ravelomanana. Quasi-Optimal Leader Election Algorithms in Radio Networks with Loglogarithmic Awake Time Slots. 2003, pp.1113-1119. hal-00084617

HAL Id: hal-00084617
https://hal.science/hal-00084617
Submitted on 8 Jul 2006

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L’archive ouverte pluridisciplinaire HAL, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d’enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.
Quasi-Optimal Leader Election Algorithms in Radio Networks with Log-logarithmic Awake Time Slots

Christian Lavault¹, Jean-François Marckert², Vlady Ravelomanana¹

(January 1st, 2003)

ABSTRACT: A radio network (RN) is a distributed system consisting of \(n\) radio stations. We design and analyze two distributed leader election protocols in RN where the number \(n\) of radio stations is unknown. The first algorithm runs under the assumption of limited collision detection, while the second assumes that no collision detection is available. By “limited collision detection”, we mean that if exactly one station sends (broadcasts) a message, then all stations (including the transmitter) that are listening at this moment receive the sent message. By contrast, the second no-collision-detection algorithm assumes that a station cannot simultaneously send and listen signals. Moreover, both protocols allow the stations to keep asleep as long as possible, thus minimizing their awake time slots (such algorithms are called energy-efficient). Both randomized protocols in RN are shown to elect a leader in \(O(\log(n))\) expected time, with no station being awake for more than \(O(\log \log(n))\) time slots. Therefore, a new class of efficient algorithms is set up that match the \(\Omega(\log(n))\) time lower-bound established by Kushilevitz and Mansour in [12].

1 Introduction

E lecting a leader is a fundamental problem in distributed systems and it is studied in a variety of contexts including radio networks [3]. A radio network (RN, for short) can be viewed as a distributed system of \(n\) radio stations with no central controller. The stations are bulk-produced, hand-held devices and are also assumed to be indistinguishable: no identification numbers (or IDs) are available. A large body of research has already focused on finding efficient solutions to elect one station among an \(n\)-station RN under various assumptions (see e.g. [3, 12, 19]). It is also assumed that the stations run on batteries. Therefore, saving battery power is important, since recharging batteries may not be possible in standard working conditions. We are interested in designing power-saving protocols (also called energy-efficient protocols). The present work is motivated by various applications in emerging technologies: from wireless communications, cellular telephony, cellular data, etc., to simple hand-held multimedia services [4].

¹LIPN (CNRS UMR 7030), Université Paris 13, 99, av. J.-B. Clément 93430 Villetaneuse France. E-mail: {lavault,vlad}@lipn.univ-paris13.fr
²LAMA (CNRS UMR 8100), Université de Versailles St-Quentin en Yvelines, 45, av. des Etats-Unis 78035 Versailles France. E-mail: marckert@math.uvsq.fr
The models. As customary, time is assumed to be slotted, stations work synchronously and have no IDs available. No a priori knowledge is assumed on the number $n \geq 2$ of stations involved in the RN: neither a (non-trivial) lower-bound nor an upper-bound on $n$. Awake stations are allowed to communicate globally (i.e. the underlying graph is a clique) by using a unique radio frequency channel with no collision detection (no-CD for short) mechanism. If, during a step, stations may either send (broadcast) a message or listen to the channel, then we talk about weak no-CD RN model. If both operations can be performed simultaneously, then the model is called the strong no-CD RN. Namely, if exactly one station sends, then all stations that listen at this time slot, including the transmitter, receive the message. (In the literature, no-CD RN usually means strong model, see e.g. [12, 15].) Such models feature concrete situations; in particular, the lack of feedback mechanism experiences real-life applications (see e.g. [13]). Usually, the natural noise existing within radio channels makes it impossible to carry out message collision detection. It is thus highly desirable to design protocols that do not depend on the reliability of any collision detection mechanism. When sleeping, any given station remains unable to hear another station, and it may also keep unaware of the election instant time in the protocol. However, stations (awake or asleep) are all required to become eventually aware of the final status of the RN. More precisely, each station may be in two states:

- either awake, i.e. listening and/or broadcasting, according to the respective model (weak or strong no-CD RN),
- or asleep, and thus saving its own battery. When sleeping, a station is “out of reach”: it cannot be waked up by none of its neighbours.

Note also that each broadcast finishes within a rather short lapse of time, and that each awake receiver is able to check if a signal has been sent by exactly one station.

Related works. The RN model considered herein may be regarded as a broadcast network model (see e.g. [3]). In this setting, e.g., Willard’s [19], Greenberg’s et al. [8] (with collision detection) and Kushilevitz and Mansour [12, 14] (no-CD) are among the most popular leader elections protocols. In the model, [13] may serve as a global reference for basic conflict-resolution based protocols. Previous researches on multiple-access channel mainly concern stations that are kept awake during the whole of a protocol in the RN, even when such stations are the “very first losers” of a coin flipping game algorithm [16]. In [10], the authors design an energy-efficient protocol (with $o(\log \log (n))$ energy cost) that approximate $n$ up to a constant factor, but with running time $O(\log^{2+\epsilon} (n))$ in strong no-CD RN. Also, distributional analyses of various randomized election protocols may include [6, 9] for example.

Our results. The first leader election protocol (Algorithm 1) presented in the paper runs in the strong no-CD RN model, while the second one (Algorithm 2) works in the weak no-CD RN model. We design a class of double-loop leader election algorithms that achieve an average $O(\log n)$ running time complexity and an average $O(\log \log n)$ awake time slots for each station in the RN. Indeed, both algorithms match the $\Omega(\log n)$ time lower-bound established in [12] and also allow the stations to keep sleeping most of the time. In other words, each algorithm greatly reduces the total awake time slots.
of the $n$ stations: shrinking from the usual $O(n \log n)$ downto $O(n \log \log n)$, while their expected time complexity still is $O(\log n)$ (with respect to the execution time). Our protocols are thus “energy-efficient” and suitable for hand-held devices working with batteries. Besides, the algorithms use a parameter $\alpha$ which works as a precise and flexible regulator. By tuning the value of $\alpha$, the running time ratio of each protocol to its energy consumption may be adjusted ($\alpha$ serves a “potentiometer”). Furthermore, the design of Algorithms 1 and 2 suggests that within both weak and strong no-CD RN, the mean time complexity of the algorithms only differs of a constant factor. Also, our results improve on [14].

Outline of the paper. In Section 2, we first present Algorithms 1 and 2, which use a simple coin-tossing procedure (rejection algorithms). Section 3 is devoted to the analyses of both algorithms, by means of tight asymptotics techniques. We conclude in Section 4.

2 Algorithms and Results

Both algorithms rely on the intuitive evidence that each station must be awake within a sequence of predetermined time slots. A first naive idea is to have stations using probabilities $1/2, 1/4, \ldots$ to wake up and broadcast. This solution is not correct however, since it is possible that no station ever broadcasts alone.

In order to correct the failure, we have to plan many rounds with predetermined length. Awake time slots are programmed at the end of each such rounds. Thus, we allow all stations to detect the (possible) termination of the session in each round. In the sequel, we let $\alpha > 1$ be the tuning parameter.

2.1 Algorithm 1

(1) \textit{round} ← 1;
(2) \textbf{Repeat}
(3) \textbf{For} $k$ from 1 to $\lceil \alpha^{\textit{round}} \rceil$ \textbf{do} \hspace{1cm} /* probabilistic phase */
(4) Each station wakes up independently with probability $1/2^k$ (to broadcast and listen);
(5) \textbf{If} a unique station broadcasts \textbf{then} it becomes a \textit{candidate} station \textbf{EndIf};
(6) \textbf{EndFor} \hspace{1cm} /* deterministic phase */
(7) At the end of each round, all stations wake up, listen and all \textit{candidate} stations broadcast;
(8) \textbf{If} there is a unique \textit{candidate} \textbf{then} it is \textit{elected} \textbf{EndIf};
(9) \textit{round} ← \textit{round} + 1;
(10) \textbf{until} a station is \textit{elected}

Algorithm 1. Leader election protocol for strong no-CD RN

Given a round $j$ in the outer-loop (repeat-until loop), during the execution time of the inner-loop each station randomly chooses to sleep or to broadcast (and/or to listen) at
each time slot (each station can compute its sequence of awaking times at the beginning of a current round). If a unique station is broadcasting, this station knows the status of the radio channel and it becomes a candidate. At the end of round $j$, every station wakes up and listens to the channel; then the candidates broadcast. If there is a single candidate, it is elected. Otherwise, the next round begins.

Define $q$ as the probability of having an election after $j^*(n) = \lceil \log_\alpha \log_2 n \rceil$ rounds and let $c_q$ be the function defined in inequalities (10) and (11),

$$c_q(\alpha) = \frac{q\alpha^3}{(\alpha - 1)(1 - \alpha(1 - q))}. \quad (1)$$

**Theorem 1** On the average, Algorithm 1 elects a leader in at most $c(q, \alpha) \log_2 n$ time slots, with no station being awake for more than $2\log_\alpha \log_2 (n)(1+o(1))$ mean time slots, where $c_q(\alpha)$ is given in (1) with $q_1 = .6305$.

### 2.2 Algorithm 2

In the case of weak no-CD RN, a potential candidate cannot alone be aware of its status since it cannot broadcast and listen at the same time. So, witnesses are needed to inform the candidates.

1. $\text{round} \leftarrow 1$;
2. \textbf{Repeat}
3. \hspace{1em} \textbf{For} $k$ from 1 to $\lceil \alpha^{\text{round}} \rceil$ \textbf{do} /* probabilistic phase */
4. \hspace{2em} Each station wakes up independently with probability $1/2^k$;
5. \hspace{2em} With probability $1/2$ each awake station decides
6. \hspace{3em} either to broadcast the message $\langle \text{ok} \rangle$ or to listen;
7. \hspace{2em} A listening station that gets this message (from one single sender) becomes a witness;
8. \hspace{1em} \textbf{EndFor} /* deterministic phase */
9. \hspace{1em} At time $\lceil \alpha^{\text{round}} \rceil + 1$, each witness and each station having broadcasted wakes up;
10. \hspace{1em} Each witness broadcasts (forwards) its received message;
11. \hspace{1em} If there is one single witness, the station that sent the (“witness”) message $\langle \text{ok} \rangle$ is elected;
12. \hspace{1em} At time $\lceil \alpha^{\text{round}} \rceil + 2$, all stations are listening;
13. \hspace{2em} \textbf{If} the leader has been elected \textbf{then} the leader broadcasts
14. \hspace{3em} and all stations are aware of the status \textbf{EndIf};
15. \hspace{1em} $\text{round} \leftarrow \text{round} + 1$;
16. \hspace{1em} \textbf{until} a station is elected.

**Algorithm 2. Leader election protocol for weak no-CD RN**

This algorithm is in the same vein as Algorithm 1. Yet, in Algorithm 2 no candidate can listen to its own message. Therefore, to be elected, a candidate needs the help of a
witness. It is important to remark that, in line (7), a station is defined as a witness iff it wakes up exactly when there exists a single broadcasting station. The election thus takes place at the end of the round during which two stations are chosen among \( n \), viz. the single candidate and its corresponding witness.

Some modifications in Algorithm 2 would slightly improve its performances. For example, to avoid possible conflicts, witnesses could be kept asleep till the end of each round and also, the algorithm could prevent any broadcasting station from becoming a witness. In its present form, we have the following result.

**Theorem 2** On the average, Algorithm 2 elects a leader in at most \( c q^2(\alpha) \log_2(n) \) time slots, with no station being awake for more than \( 2.5 \log_\alpha \log_2(n)(1 + o(1)) \) mean time slots, where \( c q^2(\alpha) \) is given by (1), with \( q^2 = .6176 \).

### 3 Analysis

#### 3.1 Technical Lemmas

The following two Lemmas use Mellin transforms \([7, 11]\); they are both at the basis of our analyses.

**Lemma 1** We have

\[
\sum_{k=1}^{r} \frac{n}{2k} \exp \left( -\frac{n}{2k} \right) = \frac{1}{\log 2} + \frac{1}{\log 2} U(\log_2 n) + O \left( \frac{n}{2^r} \right) + O \left( \frac{1}{n} \right),
\]

where

\[
U(z) = \sum_{\ell \in \mathbb{Z} \setminus \{0\}} \Gamma(\chi \ell) e^{-2i\ell \pi z}, \quad \text{with} \quad \chi \ell \equiv \frac{2i\ell \pi}{\log 2}.
\]

The Fourier series \( U(z) \) has mean value 0 and the amplitude of the series does not exceed \( 10^{-6} \). (\( \Gamma(z) \) is the Euler function \( \Gamma(z) = \int_0^\infty e^{-t} t^{z-1} \, dt \).)

**Proof:** Asymptotics on the finite sum in equation (4) is obtained by direct use of Mellin transform asymptotics \([6]\). Periodic fluctuations are occurring under the form of the Fourier series \( U(\log_2 n) \). However, the Fourier coefficients of \( U(z) \) decrease very fast, so that the amplitude of the Fourier series is very tiny, viz. \( |U(z)| \leq 10^{-6} \) (see e.g., \([7]\) or \([11, p. 131]\)). Last, the error term \( O(n/2^r) \) in (4) results from the truncated summation \( \sum_{k=1}^{r} n/2^k e^{-n/2^k} \).

**Lemma 2** Let \( r_1 = r_1(n) \) and \( r_2 = r_2(n) \), such that \( r_1 \to \infty \), while \( n/2^{r_2} \to 0 \) and \( n/2^{r_1} \to \infty \) when \( n \to \infty \). Then, for all positive integer \( m \),

\[
\sum_{k=r_1}^{r_2} \left( \frac{n}{2k} \right)^m \exp \left( -\frac{nm}{2k} \right) = \frac{m!}{m^{m+1} \log 2}
\]

\[
+ \frac{1}{m^m \log 2} U_m(\log_2(n)) + O \left( \frac{2^{r_1 m}}{n^m} \right) + O \left( \frac{n^m}{2^{r_2 m}} \right) + O \left( \frac{1}{n} \right),
\]

(3)
with \( \chi_\ell \equiv \frac{2i\ell\pi}{\log 2} \). For any \( \xi \geq 0 \) and any positive integer \( m \), the above Fourier series

\[
U_m(z) = \sum_{\ell \in \mathbb{Z} \setminus \{0\}} \Gamma(m - 1 + \chi_\ell) \exp(-2i\ell\pi z)
\]

has mean value 0 and the amplitude of the series does not exceed \( 10^{-5} \).

**Proof:** Again, asymptotics on the summation in equation (3) is completed by using the Mellin transform and complex asymptotics [7]. The error terms \( O(2r_1^m/n_m) \) and \( O(n_m/2r_2^m) \) in (3) also result from the “doubly truncated” summation: \( r_1 \leq k \leq r_2 \). □

We also use the following

**Lemma 3** Let \( (X_i)_{i \geq 1} \) and \( (Y_i)_{i \geq 1} \) be two sequences of independent Bernoulli random variables, denoted \( B(P_i) \) and \( B(Q_i) \), respectively, and such that \( P_i \leq Q_i \) for any \( i \). By definition,

\[
\mathbb{P}(X_i = 1) = 1 - \mathbb{P}(X_i = 0) = P_i \quad \text{and} \quad \mathbb{P}(Y_i = 1) = 1 - \mathbb{P}(Y_i = 0) = Q_i.
\]

Let \( H = \inf\{j \mid X_j = 1\} \) and \( K = \inf\{j \mid Y_j = 1\} \), which may be regarded as a first success in each sequence \( X_i \) and \( Y_i \) (resp.). Then, the “stochastic inequality” \( K \leq_S H \) holds. In other words, for any non-negative integer \( k \),

\[
\mathbb{P}(K \leq k) \geq \mathbb{P}(H \leq k).
\]

Moreover, for any non-decreasing function \( f \),

\[
\mathbb{E}(f(K)) \leq \mathbb{E}(f(H)).
\]

The above Lemma is a standard result in probability theory. It can be proven by constructing a probability space \( \Omega \) in which the sequences of r.v. \( (X_i) \) and \( (Y_i) \) “live”: for every \( \omega \), \( X_i(\omega) = 1 \Rightarrow Y_j(\omega) = 1 \). For any \( \omega \in \Omega \), \( K(\omega) \leq H(\omega) \), and the stochastic order is then a simple consequence of this “sure” order on \( \Omega \). Any nondecreasing function \( f \) also satisfies \( f(K(\omega)) \leq f(H(\omega)) \), \( \forall \omega \in \Omega \), and (3) holds.

### 3.2 Analysis of Algorithm 1

Assume that Algorithm 1 is in a given round \( j \) and that \( k \) satisfies \( 1 \leq k \leq \lceil \alpha^j \rceil \). Let \( p_j(n) \) be the probability that one station is elected in round \( j \). In that round, that is for \( k \) ranging from 1 to \( \lceil \alpha^j \rceil \), the stations decide to broadcast with the sequence of probabilities \( (1/2^k)_{1 \leq k \leq \lceil \alpha^j \rceil} \). We have,

\[
p_j = \sum_{k=1}^{\lceil \alpha^j \rceil} \frac{n}{2^k} \left(1 - \frac{1}{2^k}\right)^{n-1} \times \prod_{i=1, i \neq k}^{\lceil \alpha^j \rceil} \left(1 - \frac{n}{2^i} \left(1 - \frac{1}{2^i}\right)^{n-1}\right)
\]

6
\[
\sum_{k=1}^{[\alpha^j]} \frac{n}{2^k} \left(1 - \frac{1}{2^k}\right)^{n-1} \times \frac{1}{\left(1 - \frac{n}{2^j} \left(1 - \frac{1}{2^j}\right)^{n-1}\right)} \times \prod_{i=1}^{[\alpha^j]} \left(1 - \frac{n}{2^i} \left(1 - \frac{1}{2^i}\right)^{n-1}\right)
\]

\[
= \sum_{m=0}^{\infty} \sum_{k=1}^{[\alpha^j]} \left(\frac{n}{2^k} \left(1 - \frac{1}{2^k}\right)^{n-1}\right)^{(m+1)} \times \prod_{i=1}^{[\alpha^j]} \left(1 - \frac{n}{2^i} \left(1 - \frac{1}{2^i}\right)^{n-1}\right).
\] (6)

**Remark 1** Simple considerations show that when \(2^{\alpha^j} \ll n\), the probability \((1 - s_j(n))\) to have an election in the \(j\)-th round is almost 0 for large \(n\). This remark explains the occurrences of the crucial values \(n/2^{\alpha^j}\) and \(j^* = \lceil \log_2 \log_2 n \rceil\) in the analysis.

The following Lemma 4 provides an upper bound on

\[s_j(n) = \prod_{i=1}^{[\alpha^j]} \left(1 - \frac{n}{2^i} \left(1 - \frac{1}{2^i}\right)^{n-1}\right).\] (7)

**Lemma 4** Let \(j\) be increasing integers such that \(j \geq j^*(n)\), then

\[\limsup_n s_j(n) \leq 0.1884.\]

**Proof:** For any given \(i_1\), for all \(i \geq i_1\),

\[
\left(1 - \frac{n}{2^i} \left(1 - \frac{1}{2^i}\right)^n\right) \leq \left(1 - \frac{n}{2^i} \exp \left(-\frac{n}{2^i} \left(1 + \frac{1}{2^i}\right)\right)\right) \leq \left(1 - \frac{n}{2^i} \exp \left(-\frac{n}{2^i} \left(1 + \frac{1}{2^i}\right)\right)\right).
\]

Since \(\alpha^j \to \infty\) and \(n/2^{\alpha^j} \to 0\), \(\alpha^j \gg \log_2 n\), and by choosing \(i_1 = \lceil \frac{1}{2} \log_2 n \rceil\) we obtain

\[s_j(n) \leq \prod_{i=i_1}^{[\alpha^j]} \left(1 - \frac{n}{2^i} \exp \left(-\frac{n}{2^i} \left(1 + \frac{1}{2^i}\right)\right)\right) \leq \exp \left(-\sum_{m \geq 1} \frac{1}{m} \sum_{i=i_1}^{[\alpha^j]} \left(\frac{n}{2^m} \left(1 + \frac{1}{2^m}\right)\right)\right)
\]

\[\leq \leq 0.1883 + O\left(\frac{1}{\sqrt{n}}\right) + O\left(\frac{n}{2^{\alpha^j(n)}}\right).
\]

The value \(\exp \left(-\sum_{m \geq 1} m!/\left(m^{m+2} \log 2\right)\right) = 0.188209 \ldots\) is numerically computed with Maple. The upper bound on \(\limsup_n s_j(n)\) is derived by taking into account the fluctuations of the Fourier series, up to \(e^{10^{-5}}\) in our case, and the Lemma follows. □

Next, the following Lemma 5 provides an upper bound on \(p_j(n)\) (defined in (6)).

**Lemma 5** Let \(j\) be increasing integers such that \(j \geq j^*(n)\), then

\[\limsup_n p_j(n) \leq 0.3694.\]
The value of $\alpha$

It is easily seen that the algorithm and the convergence of the double sum

Remark 2

in (10) (resp.) require conditions $\alpha > 1$ and $\alpha(1 - q_1) < 1$, with $1 - q_1 = 0.3695$ (resp.). The value of $\alpha$ may thus be chosen in the range $1 < \alpha < 2.707\ldots$, so as to achieve a tradeoff between the average execution time of the algorithm and the global awake time. Thus, the minimum value of the constant $c_{q_1}(\alpha)$ is $c_{q_1}(\tilde{\alpha}) \approx 8.837$, with $\tilde{\alpha} = 1.3361\ldots$. 

Proof: By Lemma 3, since $n/2^{\alpha j} \to 0$, we have

$$\sum_{k=1}^{[\alpha j]} \left(\frac{n}{2^k}\right)^{(m+1)} \exp\left(-\frac{(m+1)n}{2^k}\right) \sim \frac{(m+1)!}{(m+1)^{m+1} \log 2} + \frac{1}{(m+1)^{m+1} \log 2} U_m(\log_2(n)), \quad (8)$$

where $U_m(z)$ is defined in (3). Summing on $m$ in equation (3) and using the techniques in Lemma 4 yields the above upper bound on $\limsup p_j(n)$, numerically computed with Maple. □

Proof of Theorem 1: Let $j^* = j^*(n) = \lceil \log_\alpha \log_2(n) \rceil$, which implies that $n/2^{\alpha(j^*+1)} \to 0$ when $n \to \infty$. According to Lemma 3, if $n$ is large enough,

$$1 - p_j \geq 2 \frac{n}{4} \frac{1}{2^{j+1}}, \quad \text{where } q_1 = 1 - 0.3695 = 0.6305. \quad (9)$$

As a consequence, the number of rounds $n_1$ in Algorithm 1 is smaller (with respect to the stochastic order) than $n_1' = j^* + G$, where $G$ is a geometric r.v. with parameter $q_1$. Indeed, let

$$n_1 = \inf\{j \mid \text{the election occurs in round } j\}$$

and let the success probability in the $j$-th round be $P_j = 1 - s_j$ (the successes in different rounds being independent). Then, $P_j \geq Q_j$, where $Q_j = q_1 \mathbb{I}_{j \geq j^*+1}$. Taking $n_1'$ as the first success in a Bernoulli sequence with probability $Q_j$, we obtain $n_1'$ as described above. Indeed, the first $j^*$ trials fail, and afterwards, each trial results in a success with probability $q_1$. The additive number of trials needed follows a geometric distribution $G(q_1)$, and

$$\mathbb{E}(n_1) \leq \mathbb{E}(n_1') = j^* + q_1^{-1} = \log_\alpha \log_2(n) + O(1).$$

Let $T_1 = T_1(n)$ be the time needed to elect a leader in Algorithm 1. Since $n_1'$ is larger than $n_1$ for the stochastic order and $r \mapsto \sum_{j=1}^r [\alpha j]$ is non-decreasing, by Lemma 3,

$$\mathbb{E}(T_1) = \mathbb{E}\left(\sum_{j=1}^{n_1} [\alpha j]\right) \leq \mathbb{E}\left(\sum_{j=1}^{n_1'} [\alpha j]\right) \leq \sum_{k=1}^{+\infty} j^* + k \sum_{j=1}^{n_1'} (1 + \alpha j)q_1(1 - q_1)^{k-1} \quad (10)$$

$$\leq c_{q_1}(\alpha) \log_2(n) + O(\log \log n). \quad (11)$$

Note that, during a round the mean number of awake times for a given station is smaller than 1. Taking into account the large number of rounds, the total number of awake time slots is shown to be smaller than $2n \log_\alpha \log_2(n)(1 + o(1))$. Since $\mathbb{P}(n_1 \leq j^*(1 - \varepsilon)) \to 0$ when $n \to \infty$, the above value is asymptotically tight. □

Remark 2 It is easily seen that the algorithm and the convergence of the double sum in (3) (resp.) require conditions $\alpha > 1$ and $\alpha(1 - q_1) < 1$, with $1 - q_1 = 0.3695$ (resp.).
3.3 Analysis of Algorithm 2

Sketch of proof of theorem 2. As already stated, two awake stations are needed in Algorithm 2: the one is only sending and the other is listening (the witness). The corresponding probability expresses along the same lines as in (6) and, instead of $p_j(n)$, one has now in step $j$,

$$p'_j(n) = \sum_{k=1}^{[\alpha_j]} \frac{1}{2} \binom{n}{2k} \left(1 - \frac{1}{2^k}\right)^{n-2} \frac{1}{\left(1 - \frac{1}{2^k}\right) (1 - \frac{1}{2^k})^{n-2}} \prod_{i=1}^{[\alpha_j]} \left(1 - \frac{1}{2^k} \left(1 - \frac{1}{2^k}\right)^{n-2}\right) (12)$$

The computation is quite similar to the proof of Theorem 1; it uses technical Lemmas as shown in Subsection 3.1. Again, asymptotics on $p'_j(n)$ in equation (12) is completed by use of Mellin transform asymptotics. Periodic fluctuations also occur under the form of a Fourier series, and after some algebra the Theorem follows.

In the case of Algorithm 2,

$$\exp \left(-\sum_{m \geq 1} \frac{m!}{(2^m m^{m+2} \log 2)}\right) = 0.462 \ldots \text{ (instead of } \exp \left(-\sum_{m \geq 1} \frac{m!}{m^{m+2} \log 2}\right) = 0.188 \ldots \text{ in Algorithm 1).}$$

Then, computing $p'_j(n)$ leads to the sum

$$\sum_{m>0} \sum_{k} \left(\frac{1}{2}\right)^m \binom{m}{2k} \left(1 - \frac{1}{2^k}\right)^{n-2} \sim \sum_{m>0} \frac{m!}{2^m m(m+1) \log 2} \sim 0.8274 \ldots$$

Now, the mean number of broadcasting stations is $n/2$ and the mean number of witnesses in round $j$ is

$$\frac{1}{2} \sum_{k=1}^{[\alpha_j]} \frac{1}{2^k} \left(1 - \frac{1}{2^k}\right)^{n-2} = O(1).$$

(Recall that a station becomes a witness if it wakes up exactly when there exists a single sender.)

Thus, the average number of awake time slots per station taking place in a round equals 2 time slots (as in Algorithm 1) plus $1/2 + O(1/n)$, due to the awaking stations appearing in line (9) of Algorithm 2. Therefore, for any station, the expected number of awake time slots is bounded from above by $2.5 \log_{\alpha_0} \log_2(n) (1 + o(1))$.

Note that with $q_2 = 0.6176 \ldots$, $\alpha$ now meets the condition $1 < \alpha < 2.61 \ldots$; and the minimum value of the constant $c_{q_2}(\alpha)$ is $c_{q_2}(\tilde{\alpha}) \simeq 8.96$, with $\tilde{\alpha} = 1.3295 \ldots \square$

Remark 3 Algorithms 1 and 2 can be improved by starting from $k = k_0$, $k_0 > 1$ in line (3). Asymptotically, the running time of the algorithms remains the same, but starting from $k = k_0$ reduces the awake time slots, to $(1 + \epsilon) \log_\alpha \log_2(n)$ for Algorithm 1 and $(1.5 + \epsilon) \log_\alpha \log_2(n)$ for Algorithm 2, respectively (with $\epsilon = 1/2^{k_0-1}$). Yet, this makes the running time longer for small values of $n$. Therefore, the knowledge of any lower bound on $n$ greatly helps.
4 Conclusion

In this paper, we present two new randomized leader election protocols in \(n\)-station RN with no knowledge of \(n\), under the assumption of weak and strong no-CD RN, respectively. The expected \(O(\log(n))\) time complexity of Algorithms 1 and 2 achieves a quasi-optimality (up to a constant factor), with each station keeping awake for \(O(\log \log(n))\) time slots in both algorithms.

Our main contribution is to propose a class of energy-efficient and quasi-optimal leader election protocols for individual clusters of an \(n\)-station RN. This class of double-loop algorithms uses a parameter \(\alpha\) which serves for a time-tuner in adjusting the tradeoff between the average time complexity of algorithms and the awake time slots of the \(n\) stations. (The tradeoff is only obtained with respect to time upper bounds). Next, our analyses provide upper bounds on the current variables. Also, the algorithms presented and the analysis of their performance improve on [14]. Such results pave the way to address the design and analysis of a broad class of energy-efficient protocols in RN: e.g. naming protocols, emulation protocols of single/multi-hop radio networks [1], respectively, etc.

References


