Energy-Efficient Leader Election Protocols for Single-Hop Radio Networks

Marcin Kardas, Marek Klonowski, Dominik Pajak

HAL Id: hal-00996429
https://hal.inria.fr/hal-00996429

Submitted on 26 May 2014

Energy-Efficient Leader Election Protocols for Single-Hop Radio Networks

Marcin Kardas*, Marek Klonowski*, Dominik Pająk[†]
*Faculty of Fundamental Problems of Technology
Wrocław University of Technology, Poland
{firstname.lastname}@pwr.wroc.pl
[†]INRIA, Bordeaux, France
dominik.pajak@inria.fr

Abstract—In this paper we investigate leader election protocols for single-hop radio networks from perspective of energetic complexity. We discuss different models of energy consumption and its relation with time complexity. We also present some results about energy consumption in classic protocols optimal with respect to time complexity – we show that some very basic, intuitive algorithms for simplest models (with known number of stations) do not have to be optimal when energy of stations is restricted. We show that they can be significantly improved by introducing very simple modifications. Our main technical result is however a protocol for solving leader election problem in case of unknown number of stations n, working on expectancy within $O(\log^\epsilon n)$ rounds, with each station transmitting $O(1)$ number of times and no station being awake for more than $O(\log \log \log n)$ rounds.

I. INTRODUCTION

In our paper we investigate *leader election* problem from the perspective of energy consumption, in various types of single hop radio networks with *collision detection*. A radio network is a distributed system without any central control, thus stations after deployment have to execute some procedures necessary for self–organization. A typical and very natural example is the *election of a leader* – the aim is to designate exactly one station as a leader. That is, after the completion of the protocol exactly one station shall have a leader status, while all other stations have non-leader status. Moreover, all stations have to know their status. Except time of execution, saving energy of stations is exceedingly important since in many scenarios replacing batteries is not feasible when stations are already deployed.

Despite there are many papers devoted to the leader election problem (some of them discuss some energetic complexity issues), to the best of our knowledge, many natural questions have not been investigated yet. In particular, it is not clear what is the relation between the execution time of a protocol solving leader election problem and necessary energetic expense of participating stations. Except some particular cases, it is not clear how much energy do we need to choose a leader. In our paper we present some new results that cover some other cases that are important form practical perspective. We also discuss how some asymptotic-optimal protocols can be significantly improved in real-life systems.

A. Model

Browsing the huge number of papers about leader election protocols one can easily notice that many substantially different, yet realistic settings are considered. Moreover, even the terminology is not consistent. In principle, we use the terminology and classification given in [1].

In a model investigated in our paper, radio network is consisted of n identical stations without any serial numbers that could be used for distinguishing them. We assume that stations have an access to stochastically independent sources of random bits. Time is slotted into *slots* and all transmissions occur at slot boundaries. Stations are synchronized as they have an access to a global clock. We assume that stations communicate via single channel. In each slot each station can *listen to the channel*, *transmit* or just have its radio *switched-off*. In each slot the communication channel can be in one of the three states:

> SILENCE – if none of stations transmits,
> SINGLE – if exactly one station transmits,
> COLLISION – if at least two stations transmit.

We assume that station that is transmitting is aware is the transmission is successful (SINGLE) or not (COLLISION). Thus station that is either transmitting or listening in time slot t obtains state of the channel in slot t. We assume that the SILENCE state can be distinguished from the COLLISION state, thus we investigate the *collision-detection* (or CD, for short) model.

In the literature several substantially different scenarios are distinguished:

- the number n of stations is known in advance;
- the number n of stations is unknown, but an upper bound N on n is known in advance;
- the upper bound on n is unknown.

Note, that all results for radio networks can be equivalently expressed for MAC (Multiple Access Channel) as described in [2]. We present however our results for radio networks, since our aim is to design energy-efficient protocols and our motivation comes from systems of strictly restricted devices, like sensor networks.

1) Problem definition: We say that algorithm $\mathcal{A}$ solves leader election problem in our model if at the end of its

execution exactly one station becomes a leader. Moreover each station must know its status (leader or non-leader) and each station must know that the algorithm is finished.

2) Evaluating measures: The most important parameter in the evaluation of protocols for radio networks seems to be the execution time. In some cases, however, more important is the *energetic complexity*. Indeed, it is very often that, for long lasting systems it does not matter for practical reasons if a task is completed in n or $\log \log n$ rounds, when n is of moderate size. More critical may be reduction of energy expense, in order to ensure that the system can work for a long time without re-charging.

The *energy usage* of a particular station is the number of slots, such that station transmits or listens to the channel (i.e., in how many rounds the radio is switched–on). The maximum of the energy usage over all stations is defined as an *energetic complexity* of the protocol. Thus, the energetic complexity of E means, that **all** stations were active (transmitting or listening) for at most E rounds. Clearly, E can be a random variable.

For practical reasons, such approach seems to be more appropriate than, for example, taking into account the average energetic usage over all stations. Indeed, in many applications it is required that all (or almost all) devices have to be working for proper functioning of the network. That is, a lifespan of such systems is determined by the most loaded station.

Let us note that, for example, in papers [3], [4] to the energetic complexity counts only rounds when the station transmits, since at least in some systems transmitting is much more energy consuming than listening that can be considered negligible for the total energy usage. We call such model *Weak Energetic Model*, or WEM for short. To the best of our knowledge, both models, WEM and regular one are justified by practical requirements of different types of systems. We believe that the regular model is more realistic for most of applications coming from sensor networks. The WEM seem to be more adequate for general MAC model when we need to count the number of transmissions, only. For that reason both models are considered in our paper.

B. Related Work

1) Leader Election protocols: Many papers have been devoted to the leader election problem for a single hop radio network. Simple algorithm for finding a leader in case of known number of stations n, is the core idea of the Ethernet Trial protocol published in [5]. In [6] Willard constructed an algorithm choosing the leader with expected running time $\log \log N + O(1)$ in the CD model and unknown number of stations, but with known upper bound N. This result is improved in [7] wherein authors present algorithm for the same settings that is completed in the time $\log \log n + O(\log f)$ with probability $1 - 1/f$. Most of previous results regarding leader election algorithms can be found in the survey [1].

Authors of [8] proved several results for deterministic counterpart of leader election problem, wherein stations have unique labels. They also present randomized protocol for no-CD model with execution time $O(\log n)$ and extremely low energetic complexity $O(\log^* n)$. The authors assume, however, that an exact approximation (up to a constant factor) of the number of stations is known. The authors proved also, that any randomized algorithm solving leader election problem in case, when bound on the number of stations is known, requires $\Omega(\log \log n / \log \log \log n)$ time slots in both CD and no-CD model.

In [9] authors present leader election protocol with expected completion time $O(\log n)$ and energetic complexity $O(\log \log n)$ for unknown n. Leader election protocol with execution time $O(\log^2 n)$ (with high probability) and energetic complexity $O(\log \log n)$ is given in [10]. In [11] a different model is considered, wherein the adversary is capable of jamming some slots. All above results refer to no-CD model.

2) Other Energy-Oriented Results for Radio Networks: Energetic efficiency of algorithms for radio networks is considered in several papers, devoted to initialization protocols [12], size approximation problem [13], alerts for weak devices [14] or routing protocols [15]. Note that most of mentioned papers refer to the model without collision detection. The paper [16] contains a short survey of recent results about to energy-efficient algorithms.

C. Our Results and Organization of the Paper

Below we present leader election algorithm with expected running time $O(\log^\epsilon n)$ and $O(\log \log \log n)$ energetic complexity in regular model and $O(1)$ in WEM model. We prove this property for any fixed $\epsilon > 0$.

In Section III we discuss energetic complexity of the simplest possible model when the number of stations n is known in advance. This part of our contribution is technically much simpler and partially based on simulations. We prove that the classic protocols are extremely efficient in terms of energy usage – i.e. the energetic complexity is a small constant with overwhelming probability in any of considered models. We show however that for moderate (yet very practical) number of stations ($n < 100$) classical protocols can be improved by introducing small changes. More precisely we can lower the maximal energetic effort over all stations up to 30%.

II. Fast Energy-Efficient Algorithm

In this section we will present our main result, which is an algorithm *Fast Energy-Efficient Leader Election* (FEELE for short) for leader election in CD model efficient in terms of time and energy complexity and works in case when neither the exact number of stations nor any approximation is known.

A. Description and Pseudo-code

A typical approach to the leader election problem is to perform a competition (based on some random experiments) between stations until the first one gains exclusive access to the shared channel. This first station that succeeded to transmit becomes a leader. Such protocol is called to be solving a *selection* problem (i.e., the problem of obtaining SINGLE by

any station). Since we demand that every station is aware that procedure is finished then all stations have to listen to the channel in every slot. With the purpose of reducing energetic expense of stations, we designed the competition performed by stations in such a way that it is not necessary for any given station to immediately know the behaviour of the others. That is, we allow stations to work in a somehow asynchronous manner (that is, possibly unsynchronized on a level of the protocol, as we still require per slot synchronization) for a long time, adjusting theirs execution only during special `control` slots.

On a basic level, the protocol consists of iterations during which stations try to reduce the number of participants. In every slot of a given iteration, each station that still takes part in the competition throws a coin and transmits in case of throwing a head. If the transmission is successful, the station considers itself as a leader and waits for the nearest `control` slot to announce its victory. In case the transmission resulted in COLLISION, the transmitting stations stop to actively participate in competition, but only if they are sure there are some other active stations. That is, the colliding stations listens to the communication channel in the next slot and become passive if there is a transmission (SINGLE or COLLISION). However, if the stations can not confirm the existence of other active stations (which does not necessary mean that there are no such stations), they proceed to the next iteration. One can see that, taken into account all stations, they can mismatch on the current number of the iteration. Therefore, starting new iteration is also handled in the nearest `control` slot. The procedure for performing competition slot is presented in Procedure 1 and the Leader Election protocol using it is presented in Protocol 3. In order to simplify the pseudo-codes we used a state-automaton approach. That is, every station can be in one of the following states:

- player – beginning state, station that is actively participating in the competition,
- listener – station that only listens in `control` slots to know whether the leader was elected,
- restart – station waits for the nearest `control` slot to synchronously start the next iteration,
- candidate – station that successfully transmitted in one the slots, potentially a leader,
- stop – after execution each station is in the stop state.

Transitions between states are accomplished in Procedures 1 and 2, with the last one being responsible for `control` slots. The main idea is also presented in Figure 1. The details are presented in the next section, where we prove the following

Theorem 2.1: Algorithm FEELE solves leader election in expected time $O(\log^\epsilon n)$ for any constant $\epsilon > 0$. Any station transmits on expectancy $O(1)$ times and listens on expectancy $O(\log \log \log n)$ times.

In the pseudocode we use function $currentTime()$ which returns a current time slot. We assume that stations are synchronized thus this function returns the same value for all stations. In the pseudocode we defined following `control` procedures arrangement. We say that algorithm FEELE has


Procedure 1 Competition slot

1: **function** COMPETITIONSLOT($phase, p$)
2:     **if not** $transmitted$ **then**
3:         **if** coin(p) **then**
4:             $transmitted \leftarrow$ true
5:             **if** transmit(id)=SINGLE **then**
6:                 $state \leftarrow$ candidate
7:     **else**
8:         **if** listen()$\neq$SILENCE **then**
9:             $state \leftarrow$ listener
10:         **else if** $phase = 1$ **then**
11:             $state \leftarrow$ restart
12:         **else**
13:             $transmitted \leftarrow$ false

Fig. 1. States transition diagram. The dashed arrows correspond to transitions handled in `control` slots.
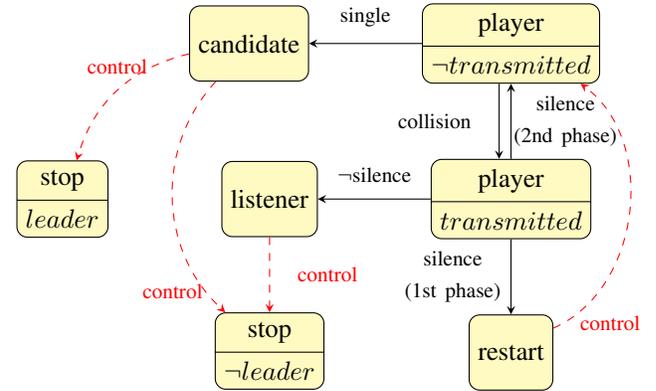
doubly-exponential control slots if procedure `control` is executed in time slots $2^{2^1}, 2^{2^2}, 2^{2^3}, \ldots$ until first `restart`. Then, if `control` with `restart` was in slot s, we execute next `control` in $s + 2^{2^1}, s + 2^{2^2}, s + 2^{2^3}, \ldots$. Note that a station s is listening only in two cases: in one slot after s transmitted (line 8 of procedure CompetitionSlot) and twice in each `control` procedure.

B. Analysis

As it is explained above, our leader election protocol is comprised of several parts. The main part are CompetitionSlots in which we are solving *selection* problem. We will show that a small number of CompetitionSlots will be sufficient to obtain the first SIGNAL. Then, we will enhance the algorithm with so-called doubly-exponential `control` procedures, in order to notify every station that some station transmitted successfully. It may happen that multiple stations transmitted successfully (i.e., we had many SINGLEs). Then, we add the third part called `resolution` to our algorithm FEELE. Procedure `resolution` elects the leader among stations that succeeded to transmit.

First, we will present procedure `control`. This procedure consists of two slots in which all stations are listening. This

Procedure 2 Control($\langle offset, exponent \rangle$)

1: **function** CONTROL
2:    **if** $state = $ candidate **then**
3:       $status \leftarrow$ transmit(id)
4:    **else**
5:       $status \leftarrow$ listen()
6:    **if** $status = $ SINGLE **then**
7:       $leader \leftarrow$ true
8:       $state \leftarrow$ stop
9:    **else if** $status = $ COLLISION **then**
10:      run RESOLUTION protocol
11:      $state \leftarrow$ stop
12:    **if** $phase = 1$ **and** $state \neq$ stop **then**
13:      synchronize()
14:      **if** $state = $ restart **then**
15:         $status \leftarrow$ transmit(beep)
16:         $state \leftarrow$ player
17:      **else**
18:         $status \leftarrow$ listen()
19:      **if** $status \neq$ SILENCE **then**
20:         $iter \leftarrow iter + 1$
21:         $iterSlot \leftarrow 1$
22:         $transmitted \leftarrow false$
23:         **return** $\left\langle offset + 2^{2^{exponent}}, 0 \right\rangle$
   return $\langle offset, exponent + 1 \rangle$

Protocol 3 Fast Energy Efficient Leader Election

1: $state \leftarrow$ player
2: $leader \leftarrow$ false
3: $iter \leftarrow 1$
4: $iterSlot \leftarrow 1$
5: $controlTime \leftarrow \langle offset, exponent \rangle \leftarrow \langle 0, 0 \rangle$
6: **repeat**   ▷ Phase 1
7:    synchronize()
8:    **if** $currentTime() = offset + 2^{2^{exponent}}$ **then**
9:      $controlTime \leftarrow$ Control($controlTime$)
10:    **else**
11:      CompetitionSlot($1, 1 - 2^{-iterSlot^{\alpha + iter}}$)
12:      $iterSlot \leftarrow iterSlot + 1$
13: **until** $state = $ stop **or** $iter > \alpha(\alpha - 1)$
14: **repeat**   ▷ Phase 2
15:    synchronize()
16:    **if** $currentTime() = offset + 2^{2^{exponent}}$ **then**
17:      $controlTime \leftarrow$ Control($controlTime$)
18:    **else**
19:      CompetitionSlot($2, \frac{1}{2}$)
20: **until** $state = stop$

---

procedure will be executed multiple times in our algorithm FEELE. The main purpose of this procedure is to notify stations if some stations succeeded to transmit since the last `control`. If some station transmitted successfully in some CompetitionSlot, it transmits also in the next `control` in which all stations are listening. Procedure `control` is executed doubly-exponential slots as defined above. If multiple stations transmit successfully, we need to resolve the conflict between them.

Our `control` procedure has an additional mechanism to maintain a common counter among all stations. In the algorithm FEELE we allow any station to call `restart`, which ensures, that in next `control` all stations are notified that the counter is incremented. In the counter will keep the number of current iteration which has to be known to all stations. Station that called `restart` cannot participate in next CompetitionSlot but has to wait until the next `control` after which the next iteration will start. We show, however, that overhead can be bounded if number of `restarts` is constant.

Argument of procedure `control` is a pair $\langle offset, exponent \rangle$ such that $offset + 2^{2^{exponent}}$ equals current time slot (time slot in which the procedure was executed). Returned value is also a pair representing the time of the next `control`.

We will use the following variant of the Chernoff bound

Fact 1 (Chernoff bounds): Let $X_1, \ldots, X_n$ be independent Bernoulli random variables such that $X = X_1 + X_2 + \cdots + X_n$, and $\mathbb{E}[X] = \mu$. Then, for $0 < \delta < 1$:

$$\Pr[X \geq (1 + \delta)\mu] \leq e^{-\frac{\delta^2 \mu}{3}},$$

$$\Pr[X \leq (1 - \delta)\mu] \leq e^{-\frac{\delta^2 \mu}{2}}.$$

We will use the following auxiliary lemmas in further analysis of our protocol.

Lemma 2.2: If $\beta \in \mathbb{N}_+$ and $x \geq 2^{2\beta^2}$, then

$$\left(\left\lceil x^{\frac{1}{\beta}} \right\rceil + 1 \right)^{\beta} \leq x + 3\beta x^{\frac{\beta - 1}{\beta}} .$$

Proof: We have

$$\left(\left\lceil x^{\frac{1}{\beta}} \right\rceil + 1 \right)^{\beta} \leq \left(x^{\frac{1}{\beta}} + 2 \right)^{\beta}$$

$$= x + 2\beta x^{\frac{\beta - 1}{\beta}} + \sum_{i=0}^{\beta - 2} \binom{\beta}{i} x^{\frac{i}{\beta}} 2^{\beta - i}.$$

Since $x \geq 2^{2\beta^2}$, then $x^{\frac{1}{\beta}} \geq 2^{2\beta}$ and $x^{\frac{1}{\beta}} \geq \binom{\beta}{i} 2^{\beta - i}$ for any $i = 0, 1, \ldots, \beta$. Thus

$$\left(\left\lceil x^{\frac{1}{\beta}} \right\rceil + 1 \right)^{\beta} \leq x + 2\beta x^{\frac{\beta - 1}{\beta}} + \sum_{i=0}^{\beta - 2} x^{\frac{i+1}{\beta}} \leq x + 3\beta x^{\frac{\beta - 1}{\beta}}.$$

∎

Lemma 2.3: If $\beta, j \in \mathbb{N}_+$, then

$$\frac{i^{\beta + 1}}{\beta + 1} \leq \sum_{j=1}^{i} j^{\beta} \leq \frac{(i + 1)^{\beta + 1}}{\beta + 1} .$$

Proof: We only need to observe, that

$$\int_0^i x^\beta dx \le \sum_{j=1}^i j^\beta \le \int_0^{i+1} x^\beta dx . \qquad \blacksquare$$

Now, we can proceed to analyse the iterations of the first phase of algorithm FEELE. Iteration will be a sequence $(s_1, s_2, s_3, \ldots, s_t)$ of CompetitionSlots such that for $1 \le i < t$ we obtained COLLISION in slot s_i and in slot s_t we obtained SILENCE or SINGLE. Note that a iteration is not necessarily a sequence of consecutive time slots because we may have a control procedure executed in the middle. In this case slots used for control procedure do not belong to the iteration.

Fix any $\alpha \in \mathbb{N}_+$. Let $\mathrm{coin}(p)$ denote function returning *true* with probability p and *false* with probability $1 - p$.

We say that station is active, if it is in player state. In the first phase there are multiple iterations that reduce the number of active stations. We will say, that iteration j is *successful* if number of stations that remained active after this iteration is at most $(3/2)^j 2^{(j+\alpha+1)^{j+\alpha+1}} \log n^{\frac{\alpha}{\alpha+j}}$.

Lemma 2.4: For sufficiently large n, if iterations 1, 2, ..., $j - 1$ were successful, then j-th iteration is successful with probability at least $1 - \frac{1}{n^2}$.

Proof: Note, that at the beginning of the first iteration we have $n = 2^{\log n^{\frac{\alpha}{\alpha}}}$ active stations. Assume, that at the beginning of j-th iteration there are $m = (3/2)^{j-1} 2^{(\alpha+j)^{\alpha+j}} (\log n)^{\frac{\alpha}{\alpha+j-x}}$ active stations for some $x \in (0, 1]$. We want to prove, that at the end of j-th iteration there are at most $(3/2)^j 2^{(\alpha+j+1)^{\alpha+j+1}} \log n^{\frac{\alpha}{\alpha+j}}$ active stations. Let us define random variable $X_i^{(j)}$ denoting number of stations that were active in i-th slot of j-th iteration and did not transmit in this slot. Take a slot

$$i = \Big\lceil \big((\alpha+j)\big((\alpha+j)^{\alpha+j}(\log n)^{\frac{\alpha}{\alpha+j-x}} - (\alpha+j+1)^{\alpha+j+1}(\log n)^{\frac{\alpha}{\alpha+j}} \big)\big)^{\frac{1}{\alpha+j}} \Big\rceil .$$

If $(\alpha+j)^{\alpha+j}(\log n)^{\frac{\alpha}{\alpha+j-x}} \le (\alpha+j+1)^{\alpha+j+1} \log n^{\frac{\alpha}{\alpha+j}}$, then $m \le (3/2)^{j-1} 2^{(\alpha+j+1)^{\alpha+j+1}} \log n^{\frac{\alpha}{\alpha+j}}$ and the iteration is successful because the initial number is smaller than the desired number of stations. In the opposite case we have $i \ge 1$. We want to prove, that $12 \log n \le \mathbb{E}[X_i^{(j)}] \le (3/2)^{j-1} 2^{(j+\alpha+1)^{j+\alpha+1}} \log n^{\frac{\alpha}{\alpha+j}}$. Probability of transmission in i-th round equals $a_i^{(j)} = 1 - \frac{1}{2^{i\alpha+j}}$. Thus

$$\mathbb{E}[X_i^{(j)}] = \frac{m}{\prod_{k=1}^i (1 - a_k^{(j)})} = \frac{m}{2^{\sum_{k=1}^i k^{\alpha+j}}}.$$

By Lemma 2.3, we have

$$\frac{m}{2^{\frac{(i+1)^{\alpha+j}}{\alpha+j}}} \le \mathbb{E}[X_i^{(j)}] \le \frac{m}{2^{\frac{i^{\alpha+j}}{\alpha+j}}} .$$

Thus,

$$\mathbb{E}[X_i^{(j)}] \le \frac{\big(\frac{3}{2}\big)^{j-1} 2^{(\alpha+j)^{\alpha+j}} (\log n)^{\frac{\alpha}{\alpha+j-x}}}{2^{\frac{i^{\alpha+j}}{\alpha+j}}}$$
$$= \Big(\frac{3}{2}\Big)^{j-1} 2^{(\alpha+j)^{\alpha+j} (\log n)^{\frac{\alpha}{\alpha+j-x}} - \frac{i^{\alpha+j}}{\alpha+j}}$$
$$\le \Big(\frac{3}{2}\Big)^{j-1} 2^{(j+\alpha+1)^{j+\alpha+1}} \log n^{\frac{\alpha}{\alpha+j}} .$$

On the other hand, by Lemma 2.2,

$$\mathbb{E}[X_i^{(j)}] \ge \frac{\big(\frac{3}{2}\big)^{j-1} 2^{(\alpha+j)^{\alpha+j}} (\log n)^{\frac{\alpha}{\alpha+j-x}}}{2^{\frac{(i+1)^{\alpha+j}}{\alpha+j}}}$$
$$= \Big(\frac{3}{2}\Big)^{j-1} 2^{(\alpha+j)^{\alpha+j} (\log n)^{\frac{\alpha}{\alpha+j-x}} - \frac{(i+1)^{\alpha+j}}{\alpha+j}}$$
$$\ge \Big(\frac{3}{2}\Big)^{j-1} 2^{(\alpha+j+1)^{\alpha+j+1} (\log n)^{\frac{\alpha}{\alpha+j}} - 3(\alpha+j)^{\alpha+j} (\log n)^{\frac{\alpha}{\alpha+j}}}$$
$$\ge 2^{(\log n)^{1/\alpha}} \ge 12 \log n .$$

These inequalities hold if $(\alpha+j+1)^{\alpha+j+1} - 3(\alpha+j)^{\alpha+j} \ge 1$, but this is true for $\alpha + j \ge 1$. Inequality $\frac{\alpha}{\alpha+j} \ge \frac{1}{\alpha}$ is true, because $j \le \alpha^2 - \alpha$. We also need inequality $2^{(\log n)^{1/\alpha}} \ge 12 \log n$ which is true for sufficiently large n. Since $12 \log n \le \mathbb{E}[X_i^{(j)}]$, then from Chernoff bound

$$\Pr[X_i^{(j)} \le 4 \log n] \le e^{-8/3 \log n} = e^{-8/3 \log e \ln n} \le \frac{1}{n^3} .$$

Thus, with high probability, at least $4 \log n$ stations will participate in slot $i+1$ and since $a_i^{(j)} \ge \frac{1}{2}$ with high probability there will be no SILENCE in slot $i + 1$. If there will be no silence in $i + 1$-st slot, then at most $X_i^{(j)}$ stations will remain active after j-th iteration. Again, from Chernoff bound we have

$$\Pr\Big[X_i^{(j)} \ge \Big(\frac{3}{2}\Big)^j 2^{(j+\alpha+1)^{j+\alpha+1}} \log n^{\frac{\alpha}{\alpha+j}}\Big] \le$$
$$\le \Pr\Big[X_i^{(j)} \ge \frac{3}{2}\mathbb{E}[X_i^{(j)}]\Big] \le \frac{1}{n^2}. \qquad \blacksquare$$

Algorithm FEELE does not stop after the first SINGLE, because all stations need to be notified that leader was elected. Notification is done in the procedure control. But between the first SINGLE and the next control stations are unaware that SINGLE was obtained and will continue executing algorithm. It may happen that more stations will succeed to transmit. In this case we will have multiple stations in candidate state. Thus in the first slot of following control procedure we will obtain COLLISION. We have some set L of stations that transmitted successfully. These stations participate in selection procedure resolution defined in [6]. Stations not belonging to L listen in every slot during the execution of procedure resolution. We will show, that expected cardinality of L is constant.

Lemma 2.5: The expected number of SINGLEs in any iteration of the first phase of algorithm FEELE is at most $\frac{2}{\ln 2} + O\big(\frac{1}{n}\big)$.

Proof: Denote by $q_i^{(j)}$ the probability that in a j-th iteration a station transmitted in slot i and has not transmitted

in slots $1, 2, \ldots, i - 1$. By the definition of the protocol, $q_i^{(j)} = (1 - 2^{-i^{\alpha+j}}) \prod_{k=1}^{i-1} 2^{-k^{\alpha+j}}$ because in any iteration of first phase a station can transmit at most once. Denote $\beta = \alpha + j$. Take any slot i the probability, that we have SINGLE in slot i is at most $n q_i^{(j)} (1 - q_i^{(j)})^{n-1}$. Thus, exactly one station has to transmit in slot i in order to receive SIGNAL. Therefore,

$$\mathbb{E}[|L|] \leq n \sum_{i=1}^{\infty} q_i^{(j)} \left(1 - q_i^{(j)}\right)^{n-1}$$
$$= n/2^n + n \sum_{i=2}^{\infty} q_i^{(j)} \left(1 - q_i^{(j)}\right)^{n-1},$$

because $q_1^{(j)} = 1/2$. Consider the function $f(x) = x(1-x)^{n-1}$ on the interval $[0, 1]$. The function has only one extremal value (maximum) for $x = 1/n$. Thus if i^* yields the maximum value of $q_{i^*}^{(j)} \left(1 - q_{i^*}^{(j)}\right)^{n-1}$ then

$$\mathbb{E}[|L|] \leq q_{i^*}^{(j)} \left(1 - q_{i^*}^{(j)}\right)^{n-1} + \sum_{i=2}^{i^*-1} q_i^{(j)} \left(1 - q_i^{(j)}\right)^{n-1} +$$
$$+ \sum_{i=i^*+1}^{\infty} q_i^{(j)} \left(1 - q_i^{(j)}\right)^{n-1},$$

and both sums are monotonic. By Lemma 2.3, for any i there exists $0 \leq \epsilon_i \leq 1$ such that $q_i^{(j)} = 2^{-\frac{(i-\epsilon_i)^{\beta+1}}{\beta+1}}(1 - 2^{-i^{\beta}})$. Note that $q_{i^*}^{(j)} \left(1 - q_{i^*}^{(j)}\right)^{n-1} \leq \frac{1}{n} \left(1 - \frac{1}{n}\right)^{n-1} \leq \frac{1}{e(n-1)}$. Let $c_n = \frac{n}{2^n} + \frac{1}{e(n-1)}$. Since $1/2 \leq 1 - 2^{-i^{\beta}} \leq 1$, then we have

$$\sum_{i=2}^{i^*-1} q_i^{(j)} \left(1 - q_i^{(j)}\right)^{n-1} \leq$$
$$\leq \int_1^{i^*} 2^{-\frac{x^{\beta+1}}{\beta+1}} \left(1 - 2^{\frac{-x^{\beta+1}}{\beta+1}-1}\right)^{n-1} dx,$$
$$\sum_{i=i^*+1}^{\infty} q_i^{(j)} \left(1 - q_i^{(j)}\right)^{n-1} \leq$$
$$\leq \int_{i^*}^{\infty} 2^{-\frac{x^{\beta+1}}{\beta+1}} \left(1 - 2^{\frac{-x^{\beta+1}}{\beta+1}-1}\right)^{n-1} dx.$$

Thus

$$\mathbb{E}[|L|] \leq c_n + n \int_1^{\infty} 2^{-\frac{x^{\beta+1}}{\beta+1}} \left(1 - 2^{\frac{-x^{\beta+1}}{\beta+1}-1}\right)^{n-1} dx$$
$$= c_n + n \int_{\frac{1}{\beta+1}}^{\infty} 2^{-y}(1 - 2^{-y-1})^{n-1} \frac{dy}{(y(\beta+1))^{\frac{\beta}{\beta+1}}}$$
$$\leq \frac{n}{2^n} + 2n \int_0^{\infty} 2^{-y}(1 - 2^{-y})^{n-1} dy$$
$$= \frac{n}{2^n} + \frac{1}{e(n-1)} + \frac{2}{\ln 2} = \frac{2}{\ln 2} + O\left(\frac{1}{n}\right).$$

Thus $\mathbb{E}[|L|] = O(1)$. ∎

Lemma 2.6: The expected number of SINGLEs in the second phase of algorithm FEELE is $O(1)$.

Proof: Let L_n be a random variable denoting the number of SINGLEs during the second phase of the Protocol. Let A_i be an event that at the beginning there are exactly i slots with no transmission, i.e., the first transmission occurs in $i + 1$-st slot. Let E_n be an event that starting with n active stations the protocol returns to the initial state, that is, in the first slot there is a SILENCE or a COLLISION followed directly by a SILENCE. By analysing two consecutive time slots, we obtain the following recurrence:

$$\mathbb{E}[L_n] = \Pr[E_n]\mathbb{E}[L_n] + \Pr[B_n = 1](1 + \mathbb{E}[L_{n-1}])$$
$$+ \sum_{k=2}^{n} \Pr[B_n = k \wedge B'_{n-k} \neq 0]\mathbb{E}[L_{n-k}|A_0],$$

where B_i and B'_i are binomial random variables, denoting number of transmissions respectively in the first and second slot, in case of i stations participating in a given slot. On the other hand, $\mathbb{E}[L_n]$ can be expressed by the conditional expected value, $\mathbb{E}[L_n|A_0]$, if we consider the number of consecutive SILENCEs occurring at the beginning. Namely,

$$\mathbb{E}[L_n] = \sum_{s=0}^{\infty} \Pr[A_s]\mathbb{E}[L_n|A_0]$$
$$= \mathbb{E}[L_n|A_0] \sum_{s=0}^{\infty} \Pr[A_s] = \mathbb{E}[L_n|A_0].$$

Therefore, we have:

$$\mathbb{E}[L_n] = \frac{\Pr[B_n = 1]}{\Pr[\neg E_n]} + \frac{\Pr[B_n = 1]}{\Pr[\neg E_n]}\mathbb{E}[L_{n-1}]$$
$$+ \sum_{k=2}^{n} \frac{\Pr[B_n = k \wedge B'_{n-k} \neq 0]}{\Pr[\neg E_n]}\mathbb{E}[L_{n-k}]$$
$$= \Pr[B_n = 1|\neg E_n] + \Pr[B_n = 1|\neg E_n]\mathbb{E}[L_{n-1}]$$
$$+ \sum_{k=2}^{n} \Pr[B_n = k \wedge B'_{n-k} \neq 0|\neg E_n]\mathbb{E}[L_{n-k}]$$
$$\leq \Pr[B_n = 1|\neg E_n] + \max_{0 \leq k \leq n-1} \mathbb{E}[L_k],$$

where the last equality comes from the fact that $\neg E_n = (B_n = 1) \vee (B_n = k \wedge B'_{n-k} \neq 0)$. From $\mathbb{E}[L_0] = 0$ and the last inequality, we get by induction that:

$$\mathbb{E}[L_n] \leq \sum_{k=1}^{n} \Pr[B_k = 1|\neg E_k]$$
$$= \sum_{k=1}^{n} \frac{k}{2^k \Pr[\neg E_k]}.$$

One can see that $\Pr[\neg E_k]$ is positive and non-decreasing, thus the last sum is convergent, which ends the proof. Computer simulations shows that $\mathbb{E}[L_n] \approx 2.30$. ∎

Before we bound time and energy complexity of algorithm FEELE we need to show how much time stations will need to wait due to synchronization in control slots after each iteration.

Lemma 2.7: If number of restart in algorithm FEELE is constant, and duration of an iteration is $O(T)$, for some

T, then number of procedures `control` executed in FEELE is $O(\log \log T)$ and total waiting time (calls synchronize() in pseudocode) is $O(T^2)$.

Proof: Each time when `restart` is called, station has to wait, but no longer than T^2, because $2^{2^{\lceil \log \log T \rceil}} \leq 2^{2^{\log \log T+1}} \leq T^2$. Thus, if there is a constant number of `restart`s, then total time complexity is $O(T^2)$. Since procedure `control` is executed in doubly-exponential slots and we perform `restart` a constant number of times, then the procedure `control` is executed in $O(\log \log T)$ time slots. ∎

Lemma 2.8: The run time of the first phase of the algorithm FEELE is $O(\log^{2/\alpha} n)$ with high probability and the maximum number of transmissions by any station is $O(1)$.

Proof: Take $i = \lceil (3 \log n)^{1/\alpha} \rceil$. In every iteration of the first phase probability of transmission in i-th slot is at least $1 - \frac{1}{n^2}$. We have at most n active stations. Probability that any of them is not transmitting is at most

$$\left(1 - \frac{1}{n^3}\right)^n \geq 1 - \frac{1}{n^2},$$

by Bernoulli inequality. Thus, with high probability we obtain SILENCE in slot $i+1$. So in each iteration `restart` is called after $O(\log^{1/\alpha} n)$ steps with probability $1 - 1/n^2$. Since new iteration starts after next `control`, taking into account synchronization, the total duration of an iteration is $O(\log^{2/\alpha} n)$ by Lemma 2.7. We have constant number of iterations in the first phase, thus, with probability at least $1 - 1/n$ the first phase of algorithm FEELE takes $O(\log^{2/\alpha} n)$ time steps. In every iteration any station transmits at most once thus since there is constant number of iterations the maximum number of transmissions is $O(1)$. ∎

If in the first phase a leader is not elected we proceed to the second phase. As we already know with high probability the first phase will significantly reduce number of stations that are in player state. The second phase lasts until a leader is elected among these stations.

Lemma 2.9: Expected time of the second phase of the algorithm FEELE is $O(\log^{2/\alpha} n)$ and expected maximum number of transmissions by any station is $O(1)$.

Proof: Let T denote the time of the second phase until the first occurrence of SINGLE. In algorithm FEELE the second phase always ends in the `control` procedure, thus, by Lemma 2.7, T^2 upper bounds the time of the second phase of algorithm FEELE. Finally, let A be an event that number of stations that remained active after the first phase is $O(2^{(\alpha^2+1)^{\alpha^2+1} \log^{1/\alpha} n})$. In the second phase stations are transmitting with probability $1/2$. If X_i denotes the number of stations active in round i that not transmitted in round $i-1$, then $Pr[\text{SINGLE in round } i] = X_i/2^{X_i}$ and $Pr[\text{SILENCE in round } i] = 1/2^{X_i}$. Consider a sequence of COLLISIONs followed by SILENCE or SINGLE. Expected length of such chain is $O(\log^{1/\alpha} n)$. Since in any slot probability of SINGLE is not smaller then probability of COLLISION then such sequence will end with SINGLE with probability at least $1/2$. Thus the expected number of such sequences is at

most 2. It is easy to see now that $\mathbb{E}[T|A] = O(\log^{1/\alpha} n)$. But we need to prove that $\mathbb{E}[T^2|A] = O(\log^{2/\alpha} n)$. Let $2^{d \log^{1/\alpha} n}$ for some constant d be the number of active stations after the first phase. Probability that a given station not transmitted in rounds $1, 2, \ldots, t$ is $1 - 2^{-t}$. Thus probability, that the exists a station that has not transmitted until $3d \log^{1/\alpha} n$ is

$$\left(1 - 2^{-3d \log^{1/\alpha}}\right)^{2^{d \log^{1/\alpha} n}} \geq 1 - 2^{-2d \log^{1/\alpha} n},$$

by Bernoulli inequality. Thus if C denotes the number of consecutive COLLISIONS in the second phase, then

$$Pr[C \geq 3d \log^{1/\alpha} n|A] \leq 2^{-2d \log^{1/\alpha} n}.$$
$$Pr[C^2 \geq 9d^2 \log^{2/\alpha} n|A] \leq 2^{-2d \log^{1/\alpha} n}.$$

Sequence of COLLISIONS cannot be longer than number of stations thus $C^2 \leq 2^{2d \log^{1/\alpha} n}$. Thus $\mathbb{E}[C^2|A] = O(\log^{2/\alpha} n)$. We know that

$$Pr[\text{There is SILENCE before SINGLE}] \leq \frac{1}{2},$$

We can denote by S_k the event that we had SILENCE k times before SINGLE. If one sequence of COLLISIONs ends with SILENCE we start the next sequence thus

$$\Pr[S_k|A] \leq \frac{1}{2^k},$$

$$\mathbb{E}[T^2|A] \leq \sum_{k=0}^{\infty} \Pr[S_k|A]\, \mathbb{E}[T^2|S_k, A]$$
$$\leq \sum_{k=0}^{\infty} \frac{1}{2^k} k^2 \mathbb{E}[C^2|A] = O(\log^{2/\alpha} n) .$$

On the other hand if A is not true, than we have $O(n)$ active stations. Thus then $\mathbb{E}[T^2|\neg A] = O(\log n)$. From Lemma 2.4 $Pr[A] \geq 1 - \frac{1}{n}$. Thus $\mathbb{E}[T^2] = \mathbb{E}[T^2|A]Pr[A] + \mathbb{E}[T^2|\neg A]Pr[\neg A] = O(\log^{2/\alpha} n)$.

Maximum number of transmissions in the second phase of algorithm FEELE is not bigger then number of SILENCEs. Because only in case of SILENCE in slot $i+1$ stations that transmitted in slot i do not deactivate. But as we already know the expected number of SINGLEs is $O(1)$ and SILENCEs are less probable. Thus the expected number of transmissions by any station is $O(1)$. ∎

Since expected number of stations participating in procedure `resolution` (lines $5, 6$ of procedure `control`) is constant, we can bound the duration of it.

Lemma 2.10: The expected duration of procedure `resolution` is $O(1)$.

Proof: Procedure `resolution` is executed if multiple stations succeed to transmit in phase 1 or 2. Denote by T the duration of `resolution` and by L the set of candidates (stations that succeeded to transmit). By Lemmas 2.5 and 2.6 we know that $\mathbb{E}[|L|] \leq c$ for some constant c. Thus from Chebyshev's inequality $Pr[|L| \geq 2^i] \leq c/2^i$. We have $\mathbb{E}[T] \leq \sum_{i=0}^{\infty} c/2^i \mathbb{E}\left[T|(2^{i+1} > |L| \geq 2^i)\right]$. But Willard's algorithm

from [6] works in expected time $\log \log |L|$. Thus $\mathbb{E}[T] \leq \sum_{i=0}^{\infty} \frac{\log(i+1)c}{2^i} = O(1)$. ∎

We can now finish the proof of Theorem 2.1. Let us remind the statement.

Theorem 2.1: Algorithm FEELE solves leader election in expected time $O(\log^\epsilon n)$ for any constant $\epsilon > 0$. Any station transmits on expectancy $O(1)$ times and listens on expectancy $O(\log \log \log n)$ times.

Proof: Fix any $\epsilon > 0$. Take $\alpha = \left\lceil \sqrt{1/(2\epsilon)} \right\rceil$. We run the presented algorithm FEELE with such α. From Lemmas 2.8 and 2.9 the expected duration of FEELE is $O(\log^{2/\alpha} n) = O(\log^\epsilon n)$. From Lemma 2.10 we know that expected duration of procedure resolution is $O(1)$. Thus total expected time of algorithm FEELE is $O(\log^\epsilon n)$.

Now we want to show that in algorithm FEELE the expected maximum number of transmission is $O(1)$. It follows from Lemmas 2.8, 2.9 and 2.10. From Lemma 2.7 we have that in algorithm FEELE the expected maximum number of listens by any station is $O(\log \log \log n)$.

Now we need to prove, that the algorithms is correct. First notice, that stations perform the same algorithm. All active stations know which iteration ($iter$) they are executing because new iteration is started using restart mechanism. Thus even if some active station never listened in any slot of the iteration it will be notified in second round of the control that new iteration starts.

Also procedure resolution which happens if there is COLLISION in first slot of control is executed by all stations. This procedure performed by small number of stations that transmitted successfully since last control. It is performed until leader is elected and all stations are listening in all slots of this procedure thus all stations are aware that the leader is elected. ∎

III. PROTOCOLS FOR THE MODEL WITH KNOWN n

In this section we investigate simplest model, when the number of stations (i.e., the parameter n) is known. In both cases for models with and without collision detection the optimal strategy for each station (in terms of time complexity) is to transmit with probability $1/n$ till the leader is chosen. This algorithm (see Protocol 4), called *ethernet trail*, described in [5] is a building block for many other algorithms solving various problems. Below we provide formal analysis of the energy complexity of this procedure. As one can expect the energetic complexity is a small constant with overwhelming probability in both models. Nevertheless energetic-efficiency of this algorithm can be improved for networks of small size as shown in Section III. Results presented below are relatively simple when compared to the Section II however we believe that they can be very important from practical perspective.

A. Energetic Complexity of Ethernet Trial

Let E denote the energetic complexity in the regular model and E^* be the energy complexity in the WEM model. One can easy see that E has exactly the same distribution as the

---

Ethernet Trial 4 Time-optimal leader election

1: **function** TIMEOPTIMALLE(n)
2:    $status \leftarrow$ SILENCE
3:    **while** $status \neq$ SINGLE **do**
4:       **if** coin($\frac{1}{n}$) **then**
5:          transmit()
6:       $status \leftarrow$ listen()

---

time of execution of the protocol (denoted below as T.) Thus we instantly get

Fact 2: For any $n, \varepsilon > 0$

$$\Pr[E > \varepsilon] \leq \left(1 - \frac{1}{e}\right)^\varepsilon .$$

Thus, the expected energetic complexity is constant and its distribution is concentrated. One can expect that the energetic complexity of this algorithm in WEM is even better. Before we present the main result, let us prove following lemma. Before we provide a bound for E^*.

Lemma 3.1: Let X be a binomially distributed random variable with parameters $\sqrt{n}$ and $1/n$. Then for any $t > 0$

$$\Pr[X > t] < \left(\frac{e}{t+1}\right)^{t+1} \frac{1}{(\sqrt{n})^t} .$$

Proof:

$$
\begin{aligned}
\Pr[X > t] &= \sum_{i=t+1}^{\sqrt{n}} \binom{\sqrt{n}}{i} \frac{1}{n^i} \left(1 - \frac{1}{n}\right)^{\sqrt{n}-i} \\
&\leq \sum_{i=t+1}^{\sqrt{n}} \binom{\sqrt{n}}{i} \frac{1}{n^i} \leq \sqrt{n} \binom{\sqrt{n}}{t+1} \frac{1}{n^{t+1}} \\
&\leq \sqrt{n} \left(\frac{e\sqrt{n}}{t+1}\right)^{t+1} \frac{1}{n^{t+1}} \\
&< \left(\frac{e}{t+1}\right)^{t+1} \frac{1}{(\sqrt{n})^t} .
\end{aligned}
$$
∎

With the lemma we can prove the following bound for energetic complexity of Ethernet Trial in WEM.

Fact 3: Let E^* be the energetic complexity of the Ethernet Trial algorithm in the Weak Energetic Model. Then

$$\Pr[E^* > \varepsilon] \leq \frac{10}{9} \left(\frac{e}{\varepsilon+1}\right)^{\varepsilon+1} \frac{1}{e^{\frac{\varepsilon-2}{2} \ln n}} + \left(1 - \frac{1}{e}\right)^{\sqrt{n}} .$$

Proof:

$$
\begin{aligned}
\Pr[E^* > \varepsilon] &= \Pr[E^* > \varepsilon | T \leq t] \Pr[T \leq t] \\
&\quad + \Pr[E^* > \varepsilon | T > t] \Pr[T > t] \\
&\leq \Pr[E^* > \varepsilon | T \leq t] + \Pr[T > t] \\
&= \frac{\Pr[E^* > \varepsilon, T \leq t]}{\Pr[T \leq t]} + \Pr[T > t] \\
&\leq \frac{\Pr[\text{There is station transmitting} > \varepsilon \text{ times}]}{\Pr[T \leq t]} \\
&\quad + \Pr[T > t] .
\end{aligned}
$$

Now we apply two observations – namely $\Pr[T > t] \leq (1 - \frac{1}{e})^t$ and $\Pr[T \leq t] > 9/10$ for $n > 10$. Let us also substitute $t = \sqrt{n}$.

$$
\begin{aligned}
\Pr[E^* \leq \frac{10}{9} n \Pr[Bin(\sqrt{n}, \frac{1}{n}) > \varepsilon] + \left(1 - \frac{1}{e}\right)^{\sqrt{n}} \\
\leq \frac{10}{9} \left(\frac{e}{\varepsilon+1}\right)^{\varepsilon+1} \frac{1}{(\sqrt{n})^{\varepsilon-2}} + \left(1 - \frac{1}{e}\right)^{\sqrt{n}} \\
\leq \frac{10}{9} \left(\frac{e}{\varepsilon+1}\right)^{\varepsilon+1} \frac{1}{e^{\frac{\varepsilon-2}{2}\ln n}} + \left(1 - \frac{1}{e}\right)^{\sqrt{n}} . \qquad \blacksquare
\end{aligned}
$$

We proved that probability of any defiation is very small and the distribution of E^* is extremely concentrated. On the other hand we can improve in real–life scenarios with networks of small-size as shown in the next section.

B. Practical Improvements

The Ethernet trial described in the previous subsection guarantees constant and small energetic complexity with high probability. Nevertheless we show below that this protocol can be still improved for networks of small number of stations. We believe that saving around 20% of energy is significant for practical considerations.

The uniformity approach of the time-optimal leader election protocol leads to some stations spending more energy than the others. It becomes even more apparent in case of small sized networks. The fact projects onto total energetic efficiency of the protocol, as we consider maximum energy usage. One way of balancing the energetic load is to introduce some kind of asymmetry, preferring silent stations over "talkative" ones. However, even simple modifications may turn out in much more difficult analysis. Therefore, we present in this subsection results of computer simulations of various modifications to the time-optimal leader election protocol.

1) Suspending strategy: The simplest modification of Protocol 4 lowering energy usage is to disallow stations to transmit more than once. That is, each station that participates in collision turns listen-only mode. Thus, the maximal energetic usage is exactly 1. However, it is possible (especially likely for small networks) that all stations turn into listening mode and the execution of the protocol will last forever. To avoid the problem, instead of permanently excluding colliding stations from competition, we allow them to step aside for few time slots. Protocol 5 shows modified version of Protocol 4, with additional parameter *sleeptime* denoting number of slots to wait after transmission, before returning to competition. Clearly, with *sleeptime* $= 0$ we get the time-optimal protocol as a special case. On the other hand, with greater and greater values of *sleeptime*, the suspending strategy more and more resembles transmit-once strategy, with its benefits and drawbacks.

2) Fade out strategy: Another way of "punishing" stations for collision is to lower their probability of transmission. That is, in the beginning, each station starts with the probability of transmission $p = \frac{1}{n}$, which is optimal in a sense of maximizing

---

Protocol 5 Suspending strategy

```
1: function SUSPENDINGLE(n, sleeptime)
2:     wait ← 0
3:     status ← SILENCE
4:     while status ≠ SINGLE do
5:         if wait = 0 then
6:             if coin(1/n) then
7:                 transmit()
8:                 wait ← sleeptime
9:             else
10:                wait ← wait − 1
11:        status ← listen()
```

Fig. 2. Results of simulations for time-optimal strategy (LE(n)), suspending strategy (SLE(n,t)) and fade-out strategy (FLE(n,f)) for 10 stations. Each row was generated with 10^6 runs.

	$\mathbb{E}[\varepsilon_{max}]$	$\mathrm{Var}[\varepsilon_{max}]$	$\mathbb{E}[T]$	$\mathrm{Var}[T]$
LE(10)	1.228	0.265	2.579	4.072
SLE(10,1)	1.135	0.149	2.594	4.154
SLE(10,2)	1.084	0.090	2.609	4.286
SLE(10,3)	1.055	0.060	2.622	4.409
SLE(10,4)	1.037	0.040	2.634	4.541
SLE(10,5)	1.026	0.028	2.642	4.650
SLE(10,10)	1.005	0.006	2.667	5.098
FLE(10,1)	1.128	0.124	2.601	4.327
FLE(10,2)	1.074	0.070	2.629	4.720
FLE(10,3)	1.042	0.040	2.653	5.132
FLE(10,4)	1.022	0.022	2.668	5.565
FLE(10,5)	1.012	0.012	2.682	6.289

probability of occurrence of SINGLE. Next, every time a given station collides, its probability of transmission is multiplied by *fadeout* factor, $0 \leq fadeout \leq 1$. Similarly as in the previous protocol, the time-optimal and the transmit-once strategies are special cases of fade-out protocol, with *fadeout* $= 1$ and *fadeout* $= 0$, respectively.

Protocol 6 Fade out strategy

```
1: function FADEOUTLE(n, fadeout)
2:     p ← 1/n
3:     status ← SILENCE
4:     while status ≠ SINGLE do
5:         if coin(p) then
6:             transmit()
7:             p ← p/fadeout
8:         status ← listen()
```

IV. CONCLUSIONS AND FUTURE WORK

We believe that many questions are left unanswered - from both theoretical as well as very practical perspective. From our point of view most interesting is the relation between energy necessary for execution of the protocol and its time

Fig. 3. Results of simulations for time-optimal strategy (LE(n)), suspending strategy (SLE(n,t)) and fade-out strategy (FLE(n,f)) for 100 stations. Each row was generated with 10^6 runs.

	$\mathbb{E}[\varepsilon_{max}]$	$\text{Var}[\varepsilon_{max}]$	$\mathbb{E}[T]$	$\text{Var}[T]$
LE(100)	1.040	0.039	2.703	4.604
SLE(100,1)	1.024	0.024	2.703	4.608
SLE(100,2)	1.015	0.015	2.704	4.607
SLE(100,3)	1.010	0.010	2.704	4.610
SLE(100,4)	1.006	0.006	2.704	4.603
SLE(100,5)	1.004	0.004	2.704	4.605
SLE(100,10)	1.000	0.000	2.704	4.611
FLE(100,1)	1.021	0.021	2.704	4.612
FLE(100,2)	1.011	0.011	2.704	4.617
FLE(100,3)	1.005	0.005	2.705	4.626
FLE(100,4)	1.003	0.003	2.705	4.628
FLE(100,5)	1.001	0.001	2.705	4.628

complexity. An interesting question is whether there exists an algorithm for leader election in model without collision detection working in expected time $O(\log \log n)$ and energy $o(\log \log n)$? Many algorithms with expected time $O(\log \log n)$ are presented in literature.

REFERENCES

[1] K. Nakano and S. Olariu, "A survey on leader election protocols for radio networks," in *ISPAN*. IEEE Computer Society, 2002, pp. 71–.

[2] L. Gasieniec, D. R. Kowalski, A. Lingas, and M. Wahlen, "Efficient broadcasting in known geometric radio networks with non-uniform ranges," in *DISC*, ser. Lecture Notes in Computer Science, G. Taubenfeld, Ed., vol. 5218. Springer, 2008, pp. 274–288.

[3] B. S. Chlebus and D. R. Kowalski, "A better wake-up in radio networks," in *PODC*, S. Chaudhuri and S. Kutten, Eds. ACM, 2004, pp. 266–274.

[4] M. Chrobak, L. Gasieniec, and D. R. Kowalski, "The wake-up problem in multihop radio networks," *SIAM J. Comput.*, vol. 36, no. 5, pp. 1453–1471, 2007.

[5] R. Metcalfe and D. Boggs, "Ethernet: Distributed packet switching for local computer networks," *Commun. ACM*, vol. 19, no. 7, pp. 395–404, 1976.

[6] D. E. Willard, "Log-logarithmic selection resolution protocols in a multiple access channel," *SIAM J. Comput.*, vol. 15, no. 2, pp. 468–477, 1986.

[7] K. Nakano and S. Olariu, "A randomized leader election protocol for ad-hoc networks," in *SIROCCO*, M. Flammini, E. Nardelli, G. Proietti, and P. G. Spirakis, Eds. Carleton Scientific, 2000, pp. 253–267.

[8] T. Jurdzinski, M. Kutylowski, and J. Zatopianski, "Efficient algorithms for leader election in radio networks," in *PODC*, A. Ricciardi, Ed. ACM, 2002, pp. 51–57.

[9] C. Lavault, J.-F. Marckert, and V. Ravelomanana, "Quasi-optimal energy-efficient leader election algorithms in radio networks," *Inf. Comput.*, vol. 205, no. 5, pp. 679–693, 2007.

[10] T. Hayashi, K. Nakano, and S. Olariu, "Randomized initialization protocols for packet radio networks," in *IPPS/SPDP*. IEEE Computer Society, 1999, pp. 544–548.

[11] M. Kutylowski and W. Rutkowski, "Adversary immune leader election in ad hoc radio networks," in *ESA*, ser. Lecture Notes in Computer Science, G. D. Battista and U. Zwick, Eds., vol. 2832. Springer, 2003, pp. 397–408.

[12] J. L. Bordim, J. Cui, N. Ishii, and K. Nakano, "Doubly-logarithmic energy-efficient initialization protocols for single-hop radio networks," in *IPDPS*. IEEE Computer Society, 2002.

[13] T. Jurdziński, M. Kutyłowski, and J. Zatopiański, "Energy-efficient size approximation of radio networks with no collision detection," in *Computing and Combinatorics*, ser. Lecture Notes in Computer Science, O. Ibarra and L. Zhang, Eds. Springer Berlin / Heidelberg, 2002, vol. 2387, pp. 77–90.

[14] M. Klonowski, M. Kutylowski, and J. Zatopianski, "Energy efficient alert in single-hop networks of extremely weak devices," in *ALGOSENSORS*, ser. Lecture Notes in Computer Science, S. Dolev, Ed., vol. 5804. Springer, 2009, pp. 139–150.

[15] K. Nakano and S. Olariu, "Energy-efficient randomized routing in radio networks," in *In Proc. of DIALM*, 2000, pp. 35–44.

[16] S. Holzer, Y. A. Pignolet, J. Smula, and R. Wattenhofer, "Monitoring Churn in Wireless Networks," in *Theoretical Computer Science (TCS) Journal*, August 2011.