Efficient Algorithms for Leader Election in Radio Networks

Tomek Jurdziński (TU Chemnitz and Wrocław Univ.)

Mirek Kutyłowski (TU Wrocław and Cryptology Center, AMU Poznań)

Jan Zatopiański (Wrocław Univ.)

Radio network model

- a network consists of *stations* (pocket devices)
- communication between stations via a shared radio channel
- common clock
- messages sent in time slots common to all stations

Applications

- military ...
- rescue operations
- logistics
- new application areas ...

Problems

- stations: *off* and *on*
  stations that are on = *alive stations*
- unknown number of alive stations
- often the stations are indistinguishable
- collisions in communication
  – at least two stations sending at the same time $\Rightarrow$ scrambling
- collision indistinguishable from noise

Complexity measures

time - the total number of time slots used,

energy -

- a station that listens or sends in a time slot is *active*, otherwise inactive
- being active causes the main usage of energy
- energy cost of a station = the number of time slots in which it is active
- energy cost of an execution = the maximal energy cost over all stations

Leader Election Problem - randomized setting

- given an unknown set of active stations, the stations are indistinguishable
- the number of active stations is
  - known
  - known up to a constant factor
  - unknown
- after electing a leader exactly one station should be in the state *leader*, the rest should be in the state *non-leader*

Leader Election Problem - deterministic setting

- the same as randomized - but the stations have unique IDs
- often: the IDs are in the range $1..n$, but not all such IDs are used
- complexity measured in n or the number of active stations

Ethernet solution

repeat until success:

1. each station with probability $\frac{1}{n}$ sends a message and listens
2. if no collision, then the station that has sent is the leader

Properties:

- probability of electing a leader in one trial $\approx \frac{1}{e}$
- within $O(\log n)$ trials a leader should be elected whp
- energy cost $O(\log n)$, equal to time

Tree algorithm

(Nakano, Olariu)

- deterministic, each station has an ID in the range $1..n$
- the number of alive stations unknown
- energy cost is $\log(n)$, time complexity is n

Tree algorithm

- put n ID's (stations) in the leaves of a full binary tree
- elect a leader in each subtree:
  for a subtree S with left subtree L and right subtree R
  1. the leader of L (if exists) sends a message and considers itself as the leader of S
  2. the leader of R listens, if no message received then it considers itself as the leader of S

Properties of the tree algorithm

- average energy cost low, if many stations alive
- the leader has the highest energy cost: always $\log(n)$
- the algorithm inefficient if few stations alive

Can we reduce energy cost??

A combined randomized solution

- $O(\log n)$ "Ethernet steps"
  only stations that send are allowed to listen at a given step
  (check for collisions)
- a station that succeeds to send without collision at step i gets ID equal to i
- run tree algorithm on stations with ID's

Energy cost of the combined algorithm

- $O(\log\log n)$ in the second stage
- $O(1)$ in the first stage, provided that a station may try to send a message at most once once
  (troubles with probability analysis, but works)

may we go below $\log\log n$??
YES!

Slaves

- usually LE algorithms gradually eliminate candidates for leaders,
  the loosers become idle
  the winners have higher energy cost
- *slaves:* a candidate that loose become a slave of the winner
- a candidates that wins not only enslaves the looser but also takes all its slaves
- **the slaves work for their masters**
  goal: more uniform energy cost

Using the slaves

- without slaves: the master has to perform T communication steps (energy cost T)
- with slaves $s_1, \ldots, s_k$:
  - s_1 emulates the first T/k communication steps of the master
  - s_1 informs s_2 of the state of simulation
  - s_2 takes over and simulates the next T/k communication steps of the master
  - ...
- energy cost (as maximum) becomes T/k instead of T

Dense tree algorithm

Assumptions:

- $\Omega(n)$ stations active
- alive stations have unique IDs in the range $1..n$

Result There is a deterministic LE algorithm for this setting with energy cost $O(\log^* n)$ and time $O(n)$.

Idea of the algorithm

- modified tree algorithm
- phase i
  - divide *masters* into groups of size s_i
  - in each group perform tree algorithm with slaves
  - each (rich) master has $\log s_i$ slaves, so energy cost $O(1)$
  - a poor master becomes inactive
- there is $\Omega(s_i)$ rich masters in a group
  $\Rightarrow$ the leader elected in the group gets $\Omega(s_i \cdot \log s_i)$ slaves

Idea of the algorithm

- for the next phase we can set:

$$s_{i+1} = 2^{s_i \cdot \log s_i}$$

- so there are $\log^* n$ phases,
  with energy cost $O(1)$ for each phase
- details: there are enough "rich masters" at each phase

From dense to randomized

Randomized algorithm

1. $O(\log n)$ "Ethernet trials",
   each station participates in at most one trial
2. dense tree algorithm for electing the leader from the stations that have succeeded

Result: randomized algorithm with energy cost $O(\log^* n)$ and time $O(\log n)$.

Technical problems

- “Ethernet trials” are not independent, Bernoulli trials model does not apply
- estimation technique: “Energy-Efficient Size Approximation for Radio Networks with no Collision Detection”, JKZ, COCOON’2002.

Sublogarithmic deterministic solution

Result: a deterministic algorithm for stations with unique IDs in the range $1..n$, with energy cost $O(\log^{\varepsilon} n)$

Sublogarithmic deterministic solution - idea

Construction idea:

- divide IDs into groups of size $k = k(n)$
- in each group
  - all alive stations send a message
  - if no collision, then the station that has succeeded is a leader of the group
  - if collision, then execute the tree algorithm
    note that the leader gets at least one slave!
- choose the leader from the leaders elected in the groups (use slaves if possible!)

Sublogarithmic deterministic solution - remarks

- apply recursively
- for $k(n) = n^{2^{-t}}$ energy cost is $O(\log n^{1/t})$

Lower bounds - time

Assumptions:

- deterministic algorithm
- unique IDs in the range $1..n$
- an arbitrary set of IDs used

Result: each LE algorithm in such a setting requires time $\Omega(n)$.

Lower bounds - energy

Assumptions:

- deterministic algorithm
- unique IDs in the range $1..n$
- an arbitrary set of IDs used

Result: each LE algorithm in such a setting has energy cost

$$\Omega(\log\log n / \log\log\log n).$$

Energy lower bound - proof idea

- we analyze the steps and reduce the set of stations that might be alive
- goal: such stations know nothing about other stations that may be still alive after reduction
- technicalities: weights of stations,
  $s_i =$ the total weight after considering step i

Reductions of step i

- A_i - the set of all stations that might be active at step i after previous reductions, S_i - stations in S_i that would send at step i, R_i - stations in S_i that would only listen.
- reduction (X), if $|S_i \cup R_i| > 2\pi_n(n_i)$
- reduction (Y), otherwise

where

$$\pi_n(m) = m^{1/\log\log n}$$

Reductions of step i

- reduction (X) (set of sender and receivers is small):

$$A_{i+1} := (A_i \setminus (S_i \cup R_i)) \cup \{j\}$$

  where j is the station in $S_i \cup R_i$ with the maximal weight, $n_{i+1} = n_i$, the new weight of j equal to the sum of weights of elements of $S_i \cup R_i$

- reduction (Y):
  A_{i+1} is the bigger of the sets S_i and R_i.
  $n_{i+1} := |A_{i+1}|$

Observations

- many (Y) reductions $\Rightarrow$ high energy cost
- few (Y) reductions $\Rightarrow$ after the last (Y) reduction $n_i \geq \log n$
  $\Rightarrow$ the leader must accumulate the weight up to n_i, but the rate of growth bounded:
  after participating k times in communication step the weight bounded by $(2\pi_n(n_i))^k$

Conclusions and open problems:

- situation in a single hop radio network fairly well recognized, lower and upper bound quite close in many situations
- *multi-hop* radio network?