Adversary Immune Leader Election in Ad Hoc Radio Networks

Mirek Kutyłowski and Wojtek Rutkowski

Wrocław University of Technology

European Symposium on Algorithms 2003

Our model

Single Hop Radio Network

(*RN*, *Ad Hoc network*) :

- $O(N)$ processing units called *stations*
- the stations **are not** numbered 1 through *n* (*initialization problem*)
- a single communication channel,
- messages sent simulateneously collide producing random noise
- stations cannot detect collisions (*no collision detection model – no-CD*),
- discrete, synchronous *time slots*,

Complexity measures

time - the number of time slots

energy cost - the maximal number k such that some station transmits or listens k times during algorithm execution

- relates to battery usage
- communication consumes almost all energy
- battery exaustion is a major issue
- energy required for transmitting and listening of the same magnitude (processor and sensors usage - negligable)
- **extremely important for practical reasons!**

Classic Leader Election Problem

Given a Single Hop Radio network initialize it so that

- *exactly* one station gets the status *leader*
- the other non-active stations receive the status *non-leader*.

Optimize for time and energy costs for each station.

New approach

- an adversary may disturb communication
- design a leader election algorithm that would work anyway

Adversary model

- random transmission errors, or burst errors, or even an adversary knowing the algorithm
- the adversary attempts to cause collisions so that the algorithm:
  - no leader is elected, or
  - more than one leader is elected
- the adversary does not know a secret of legitimite stations
  $\Rightarrow$ the stations can use keyed MAC to prevent faking messages by an adversary
- an adversary cannot use much higher communication resources than other stations

Security of previous solutions

easy to attack by an adversary

- Ethernet-like algorithms
  - energy cost equals execution time,
  - low probability of success,
  - energy cost of stations = adversary
- tree election algorithm
  - a single adversary message – avalanche effect, multiple leaders elected
- other algorithms scenario:
  - a small group of candidates remains, choose a leader from them,
  - the adversary may attack this stage (few messages!)

Main Result

assumptions a single-hop no-CD radio network with $\Theta(N)$ stations sharing a secret key. The stations are not initialized with ID's.

our algorithm leader election with energy cost $O(\sqrt{\log N})$ and time complexity $O(\log^3 N)$,
the outcome might be faulty with probability $O(2^{-\sqrt{\log N}})$ in a presence of an adversary station which has energy cost $O(\log N)$.

Basic tricks

- **cryptographic methods**
  - legitimate stations share a secret
  - messages enciphered and undistinguishable from random noise
- **time windows -** within a group of steps only one used for communication,
  which one is used depends on a pseudo-random value computed from the secret and current time
- **random ID Reassigment -** between phases of the algorithm the stations permute their temporary IDs

Algorithm overview

$v = \Theta(\sqrt{\log N})$

- **Preprocessing -** we choose at random v small groups (each of size at most $O(\log n)$) of (pair of) candidates for the leader
- **Group elections -** v times repeated *group election* phases.

  The first group that succeeds in choosing a group leader "**attacks**" all subsequent group election phases preventing another leader to be chosen.

Preprocessing

- Each station decides randomly to be a sender or receiver.
- For each of $d = v \cdot k$ rounds, a station decides to turn on the radio with probability N^{-1}, and act as sender or receiver.
- If exactly one sends and exactly one receives: the pair gets tempID equal to the step number
  (sender sends a message, receiver confirms, and sender also confirms).
- once a station tries to get tempID, it remains idle for the rest of preprocessing.

Result of Preprocessing

- each of v groups has $\Theta(\log N)$ candidates with high probability
- the adversary may eliminate only a certain fraction of them

Group election phase

Consists of two stages:

- building chains of candidates,
- merging chains.

Id numbers are “rotated” in a psuedo-random way.

Building Chains

- k communication slots - each consisting of 4 windows of size $\Theta(\log^{3/2} N)$
- Current Agent tries to contact the next one
  (introduce, confirm introduce, respond, confirm respond)
- If Current Agent succeeds the next station becomes the Current Agent.
- Adversary can brake a chain.
- If there is no active Agent - the first active pair becomes Current Agent and starts a new chain.

Merging Chains

- Previous stage created chains.
- Current goal: merge chains in a chain covering at least half of the candidates of the phase (more exactly: half of the temporary IDs).
- In a suitable time slot the last agent in a chain informs the current and the next chain about all participants and so does the last agent of the next chain.

Disabling later groups - internal attack

- Successfull chain is blocking the later groups from merging the chain, it acts like an adversary - possible due to the knowledge of exact times of sending messages.
- Enough pairs to act as an adversary without exceeding the energy limit.
- Method of blocking: participating in a special way in creating the chains
- Adversary cannot turn off internal attack - too many stations.

Additional feature

- the algorithm yields a group of $\Omega(\log N)$ active stations which know each other,
- it can be used to choose vice leader at no cost

Open problems

- lower bound?
- energy cost below $O(\sqrt{N})$
  (without an adversary – possible!)
- initialization problem – a new paper in preparation
- other fundamental algorithms?
- multi hop model?