Synchronization problems in multihop wireless networks

Laura Marie Feeney
lmfeeney@sics.se

WWIC 2006, Berne Switzerland

Synchronization problems in multihop wireless networks

Laura Marie Feeney

Motivation

Sensor networks

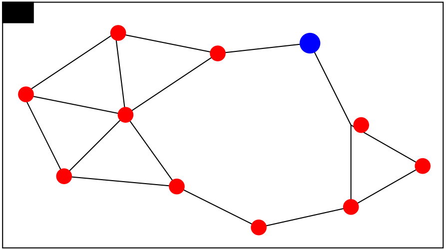
Ad hoc networks

Embracing asynchrony

SICS

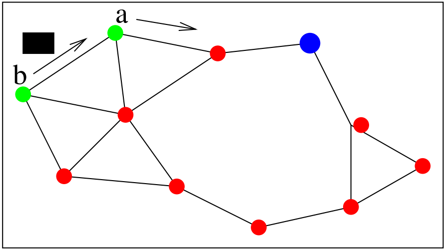
Target tracking

Sensors report to sink



Target tracking

Sensors report to sink

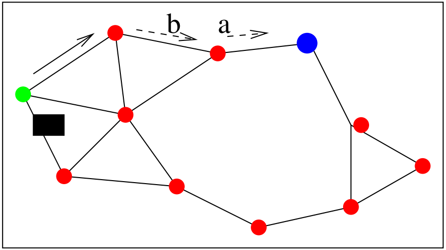


sensors detect target

SICS

Target tracking

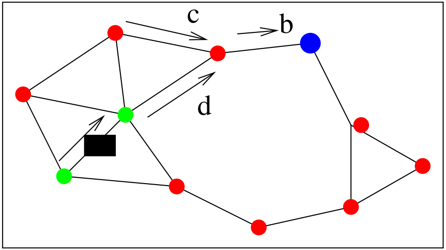
Sensors report to sink



old and new messages in transit

Target tracking

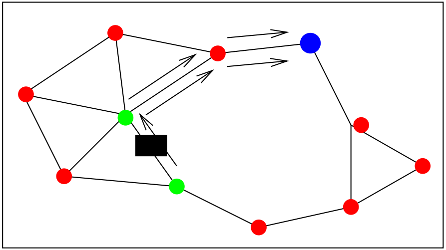
Sensors report to sink



data compression (c-d)?

Target tracking

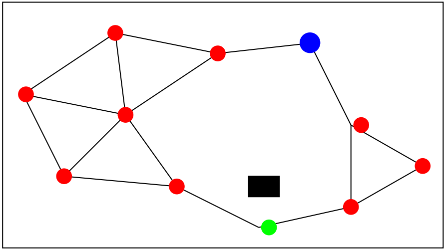
Sensors report to sink



is $t_a > t_b$ or $t_b < t_a$?

Target tracking

Sensors report to sink



calculating velocity of target

Outline

introduction

- motivation

strictest requirements

- sensor networks

relax some requirements

- ad hoc networks

embracing asynchrony

- asynchronous protocols

Motivation

Why should I bother to discuss the problem?

Motivation

Why should I bother to discuss the problem?

(why should you bother to listen?)

Practical problem

Solutions depend on synchronized clocks

- secure protocols
- distributed protocols
    - channel access
    - group communication
- energy management
- sensor data analysis

Easy to get wrong...

Simulators are (necessary) evil

- simulations have poor communication models
  - RSS varies predictably with position ($1/R^{-\alpha}$)
  - RSS doesn't vary over time
- unrealistically reliable communication (esp. broadcast)

Easy to get wrong...

Simulators are (necessary) evil

- common simulation environments are implicitly synchronous
  - nodes turn on at time 0 and have identical clocks
  - periodic events are magically synchronized across the network
  - even under partition and merge...

Easy to get wrong...

Solving the right problem...

- What kind of synchronization is needed?
  - “new regime” - Elson and Römer
- global synchronization
- local (pairwise) synchronization
- on-demand synchronization
- periodicity
- virtual clocks

Hard problem

Assumptions don't hold...

- no infrastructure (e.g. NTP clocks)
- partition and merge
- variable delays, changing routes
- limited resources

Very hard problem

Sensor network context

- centralized structure
- limited resources
- variable connectivity (less mobility?)
- unstable clocks

Very hard problem

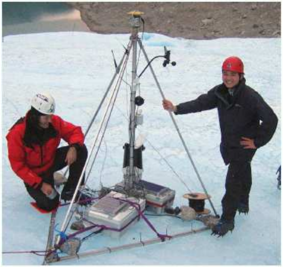
Sensor network context

- centralized structure
- no resources
- variable connectivity (less mobility?)
- unstable clocks

Very hard problem

Sensor network context

- centralized structure
- no resources
- variable connectivity (less mobility?)
- unstable clocks



Very hard problem

Sensor network context

- centralized structure
- no resources
- variable connectivity (less mobility?)
- unstable clocks

Synchronization in sensor networks

Requirements are application-specific

- type of synchronization required
- accuracy

Two drivers

- sensor data analysis
- energy management

Synchronization

global (i.e. UTC) synchronization

- requires consistent access to UTC time
- functions across partition and merge
- federated sensor networks

local synchronization

- sensitive to partition and merge
- sufficient accuracy for many purposes

Synchronization

clock discipline

- clock offset vs clock drift
- continuous correction
- PLL (i.e. NTP) very expensive

Synchronization

clock discipline

- clock offset vs clock drift
- continuous correction
- PLL (i.e. NTP) very expensive

open(?) problem: merging clocks

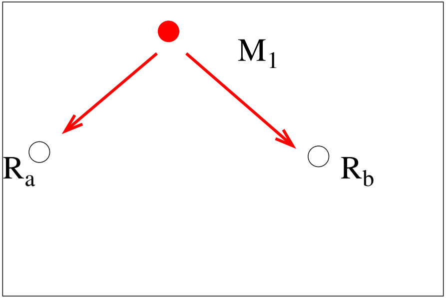
- "easily extended"

Synchronization in sensor networks

- receiver-receiver synchronization
  - Reference Broadcast Synchronization (Elson et.al.)
- sender-receiver synchronization
  - Lightweight Tree-based Synchronization (van Gruenen et.al.)
- on-demand synchronization
  - "post-facto" synchronization (Elson et.al.)

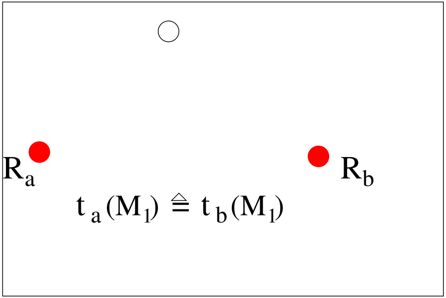
RBS (simplified)

receiver-receiver synchronization



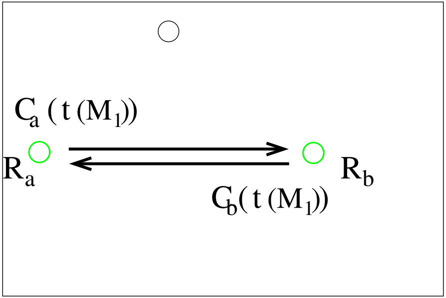
RBS (simplified)

receiver-receiver synchronization



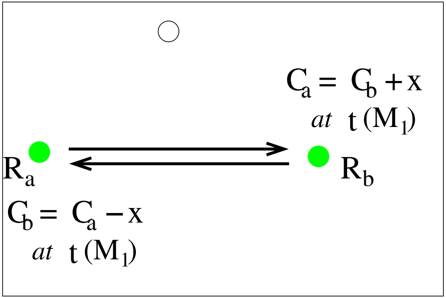
RBS (simplified)

receiver-receiver synchronization



RBS (simplified)

receiver-receiver synchronization

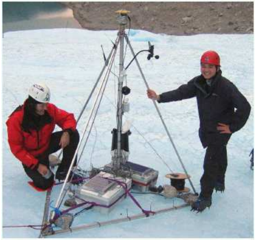


RBS (extensions)

- multiple broadcasts to minimize receiver variation
- statistical analysis to estimate clock skew (!)
- multi-hop synchronization
- on-demand synchronization

state of the art: 1's-10's μsec on mote hardware

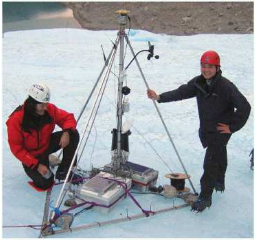
Intermittent connectivity



photo, K.Martinez

eight probes embedded in Norwegian glacier

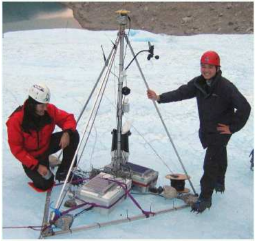
Intermittent connectivity



photo, K.Martinez

intended lifetime ≈ 1 year

Intermittent connectivity



photo, K.Martinez

probes: 6 readings/day (15 sec)

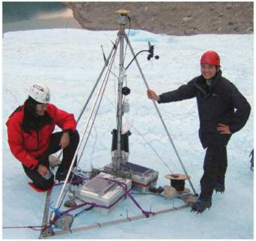
probes: 1 upload/day (3 min)

base: 1 wakeup/day (5-15 min)

base: 1 GPS/week (10 min)

intended lifetime $\approx$ 1 year

Intermittent connectivity



photo, K.Martinez

probes: 6 readings/day (15 sec)

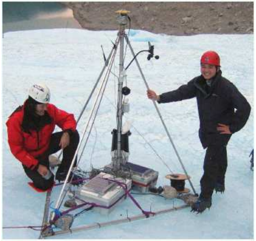
probes: 1 upload/day (3 min)

base: 1 wakeup/day (5-15 min)

base: 1 GPS/week (10 min)

some probes not recovered after base station failure

Intermittent connectivity



photo, K.Martinez

probes: 6 readings/day (15 sec)

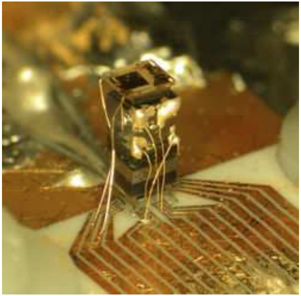
probes: 1 upload/day (3 min)

base: 1 wakeup/day (5-15 min)

base: 1 GPS/week (10 min)

University of Southampton, UK
http://envisense.org/glacsweb/index.html

The future


photo, US NIST

chip-scale atomic clock (1cm, 75mW)
http://tf.nist.gov/ofm/smallclock/

Synchronization in adhoc networks

Relax requirements

- global synchronization
- resource availability

Energy management

- idle interface consumes energy
- maximize time in sleep state

Synchronous protocol

Distributed election of topological cover

- clustering
- dominating-set

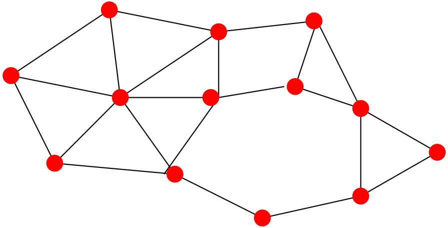
Mimics operation of base station

- clusterhead nodes buffer traffic
- non-clusterhead nodes sleep
- e.g. Chen et.al. 2001

Topology discovery is synchronous...

Example

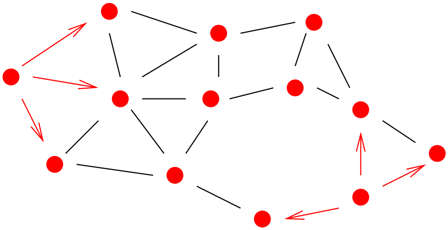
Power save protocol



nominate a “covering set”
(minimum dominating set)

Example

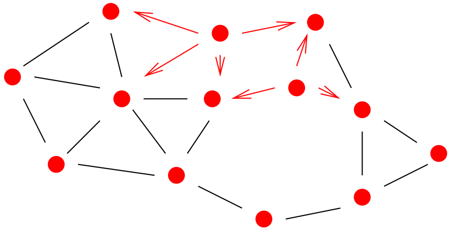
Power save protocol



broadcast messages for topology discovery

Example

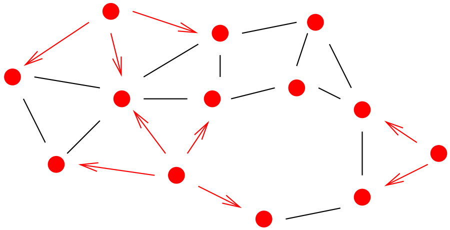
Power save protocol



broadcast messages for topology discovery

Example

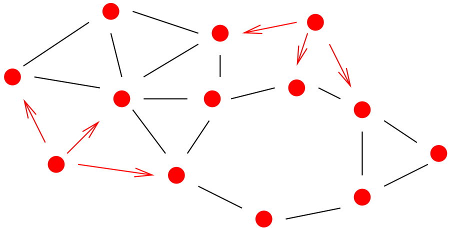
Power save protocol



CSMA prevent many (not all) collisions

Example

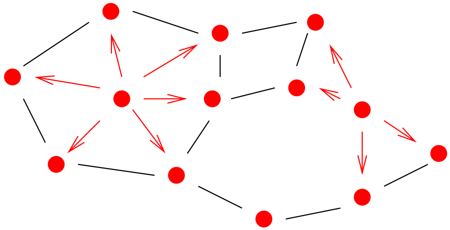
Power save protocol



CSMA prevent many (not all) collisions

Example

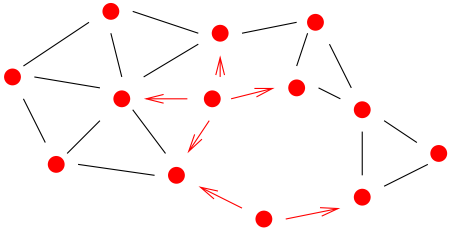
Power save protocol



given 2-hop topology, nodes estimate their suitability

Example

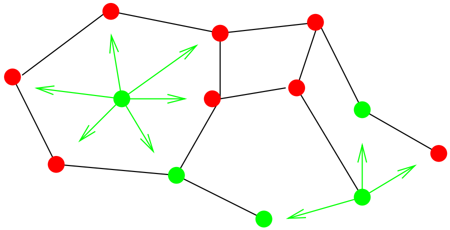
Power save protocol



given 2-hop topology, nodes estimate their suitability

Example

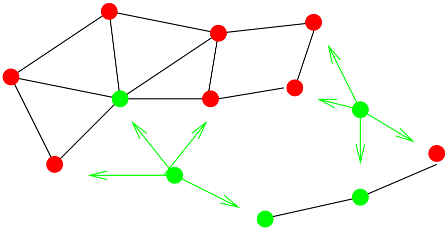
Power save protocol



clusterheads elect themselves
(adaptive backoff)

Example

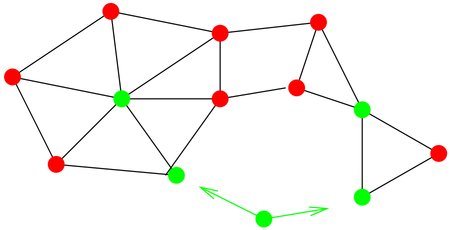
Power save protocol



clusterheads elect themselves
(adaptive backoff)

Example

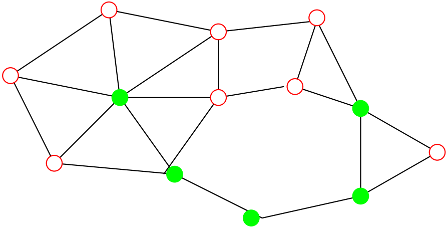
Power save protocol



clusterheads elect themselves (adaptive backoff)

Example

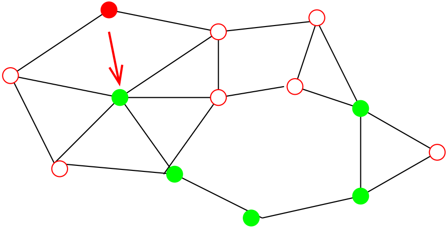
Power save protocol



non-clusterheads periodically wake up

Example

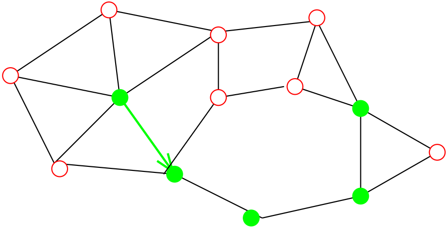
Power save protocol



clusterheads buffer traffic for non-clusterheads

Example

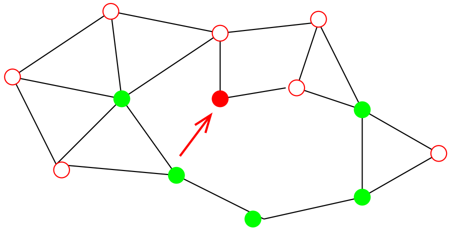
Power save protocol



clusterheads buffer traffic for non-clusterheads

Example

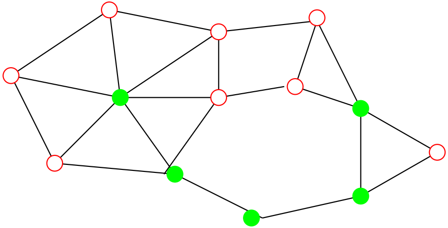
Power save protocol



clusterheads buffer traffic for non-clusterheads

Example

Power save protocol



topology discovery must(?) be synchronous

Synchronization problems in multihop wireless networks

Laura Marie Feeney

Motivation

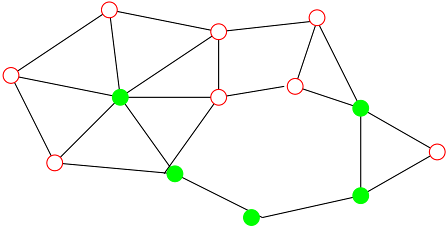
Sensor networks

Ad hoc networks

Embracing asynchrony

Example

Power save protocol



open(?) problem: conditions for asynchronous discovery/election to converge?

Asynchronous protocols

No synchronized sleep/wake

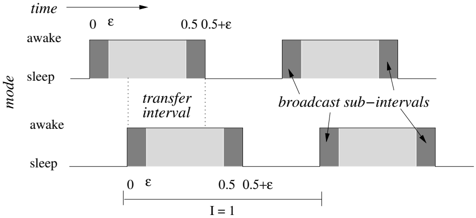
- PHY layer
  - energy sampling
  - preamble sampling (low-traffic networks)
- probabilistic
  - eventually communicate with some neighbor (with high probability) (dense networks)

Asynchronous protocols

No synchronized sleep wake

- asynchronous schedules
  - well-known, periodic wakeup schedule
  - unknown offset to neighbors' schedules
  - deterministic overlap of wake intervals
- use overlapping intervals
  - traffic announcement (wake up message)
  - discovery of offset to neighbor's schedule

Simple deterministic protocol



nodes are awake ϵ more than 50%

guaranteed to overlap with neighbors

Feeney('02), Tseng,Jiang('02,'05)

Complex protocols

More complex periodic schedules have lower duty cycles

- quorum schedule (Tseng, Jiang et.al.)
- block schedule (Zheng et.al.)
- hybrid probabilistic (Hurni et.al.)

Intuition

Nodes discover their neighbors' schedules.

Transmission window when both nodes are awake.

Intuition

Intuitively, some wakeup schedule distributions will be “friendlier” than others.

Intuition

Intuitively, some wakeup schedule distributions will be “friendlier” than others.

- How much friendlier?
- Can we ensure friendliness?

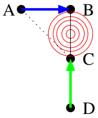
Intuition

Intuitively, some wakeup schedule distributions will be “friendlier” than others.

- How much friendlier?
- Can we ensure friendliness?
- Without solving distributed STDMA...

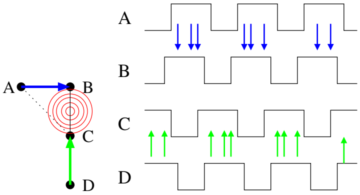
Wakeup schedule distributions

transmit $A \longrightarrow B$ and $D \longrightarrow C$



Wakeup schedule distributions

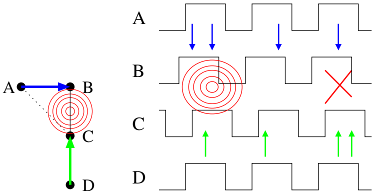
transmit $A \longrightarrow B$ and $D \longrightarrow C$



wakeup distribution reduces contention

Wakeup schedule distributions

transmit $A \longrightarrow B$ and $D \longrightarrow C$

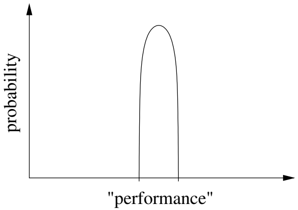


wakeup distribution increases contention

Thought experiment

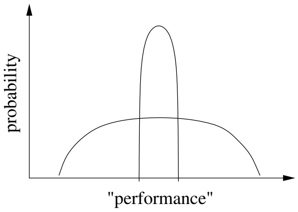
- fix topology/traffic scenario
- measure network "performance" for many wakeup schedule distributions
- performance measurements define probability distribution

Thought experiment



a narrow distribution implies that wakeup schedule distribution has little overall impact

Thought experiment

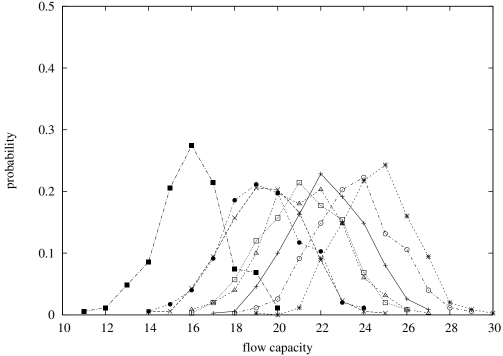


a flat distribution implies high sensitivity
randomized strategies may be useful

Non-thought experiment

For each topology, the flow capacity measurements form a probability distribution, showing sensitivity to wakeup schedule distribution.

Non-thought experiment



8 topologies (large rectangle, 100 nodes)

Non-thought experiment

"Friendly" wakeup schedule distributions provide almost twice as much capacity as "unfriendly" ones.

Friendly distributions

Asynchronous system

- easy to change the distribution
- for the better?
  - randomization?
  - feedback loops?

wakeup overlaps define coarse grain, variable length transmission opportunities (“slots”)

Hypothesis

(quasi-)periodic emergent behavior provides

- soft structure
- (quasi-)predictability

leverage these patterns for higher layer (approximate) traffic management

Biological models?

Question: Do "biological" models apply?

Most "biological" protocols assume lowest level functionality

- e.g. assumes CPU, OS, packet transmission
- "ant-based" ad hoc routing uses packets

Distributed emulation of biological process

Biological models?

A "biological" MAC or L2.5 approach would be quite different.

What does a "biological" PHY look like?

Biological models?

A “biological” MAC or L2.5 approach would be quite different.

What does a “biological” PHY look like?

Very open problem:

- non-packetized PHY/MAC for sensor network?

Thanks!

ACM Workshop on Real-World Wireless Sensor Networks

REALWSN'06

http://www.sics.se/realwsn06/