EXTENDED-DESYNC

A Desynchronized TDMA Protocol for WSNs
- An Energetic and Temporal Analysis

August, 13th – 14th 2009
Fachgespräch *Sensornetze*, TU Hamburg-Harburg

C. Mühlberger
Chair of Computer Science V

Agenda

- Motivation
- DESYNC / EXTENDED-DESYNC
- Energy
- Latency
- Conclusion & Outlook

Motivation

Why and what for such
a MAC Protocol?

C. Mühlberger
Chair of Computer Science V

Motivation

Wireless Sensor Networks

Shared communication medium	Limited energy resources
↓	↓
Collisions	Long network lifetime
↓	↓
MAC protocol	Radio Duty Cycle
↓	↓

DESYNC / EXTENDED-DESYNC

DESYNC / EXTENDED-DESYNC

What's behind and

how does it work?

C. Mühlberger
Chair of Computer Science V

Basic Conditions

Biologically inspired primitive of Desynchronization:

Each oscillator (e.g. periodically transmitting sensor node) tries to maximize its relative time lag to its immediate neighbors.

- Set *N* of nodes with
  - Unique identifier *i*
  - Identical frequency ω ➔ common period $T = \frac{1}{\omega}$
  - *T* must support **at least** *n* participators
    - Single-hop: *n* = |*N*|,
    - Multi-hop: *n* = size of maximum two-hop clique
- Symmetrical links
- Carrier Sense just before any transmission
- Communication range ≈ Interference range

Operating Mode

- Phase $\phi_i \in [0.0, 1.0]$ of node *i*

  (elapsed time since last firing as percentage of period *T*)
- At $\phi_i = 1.0$, node *i* resets $\phi_i = 0.0$ and broadcasts *firing packet*
- Previous phase neighbor *p(i)* (fires just before node *i*)
- Successive phase neighbor *s(i)* (fires just after node *i*)
- With $\phi_{p(i)}$, $\phi_{s(i)}$ and $\alpha \in (0.0, 1.0)$

➔ **Next phase** $\phi_i' = (1-\alpha) \cdot \phi_i + \alpha \cdot \dfrac{\phi_{s(i)} + \phi_{p(i)}}{2}$

➔ Global state $\vec{\phi} = [\phi_1, \phi_2, \dots, \phi_{|N|}]^T$ (generally indeterminable by node *i*)

Desynchrony (stable system state):

Any node has equal temporal distance to its phase neighbors.

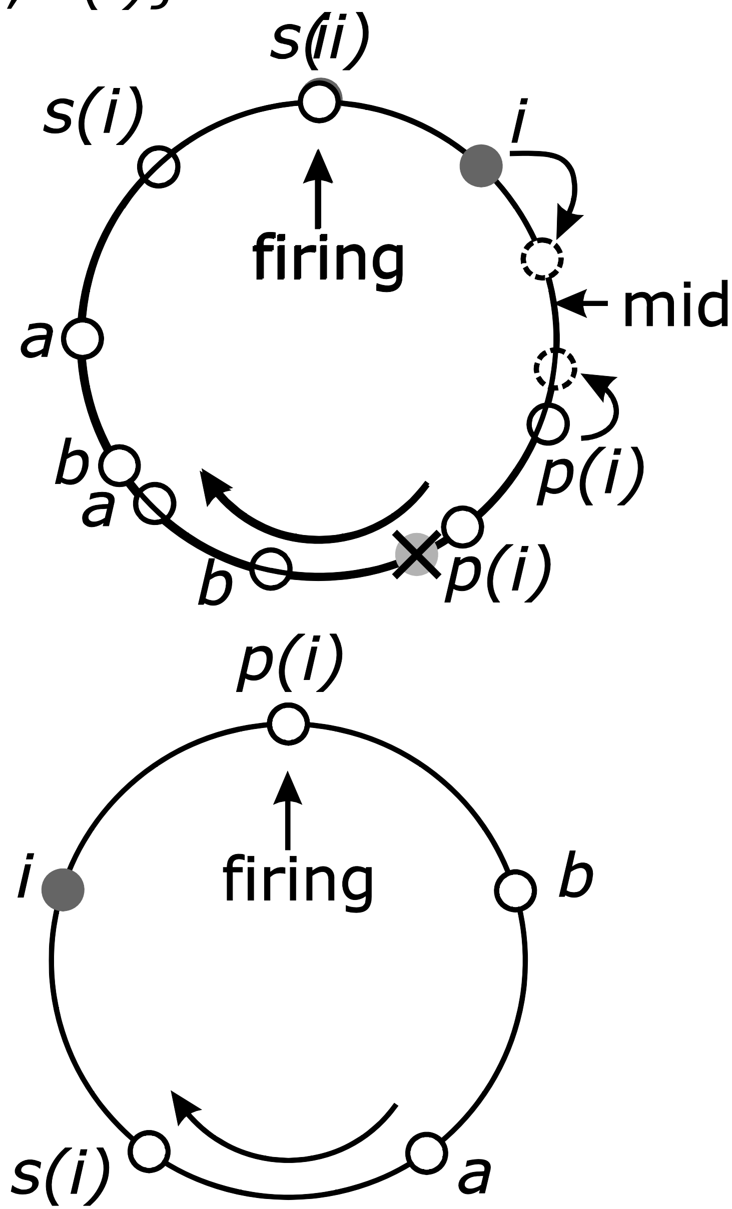
(EXTENDED-)DESYNC converges ➔ desynchrony is demonstrably reachable.

Example

- Single-hop topology, $N=\{a, b, p(i), i, s(i)\}$

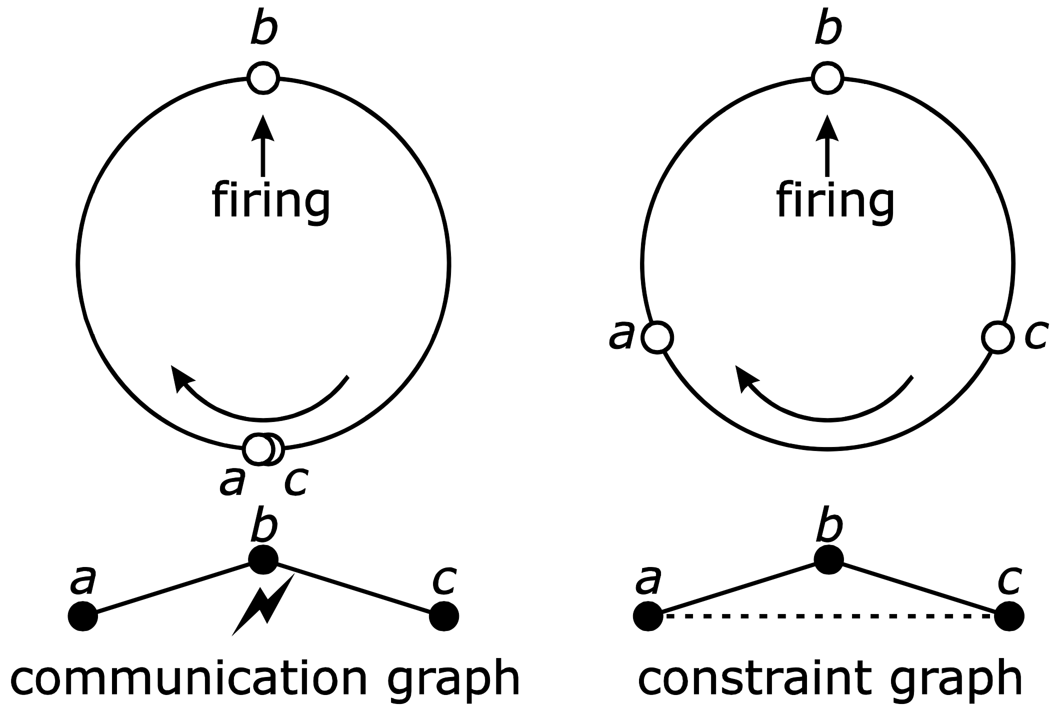
- During settling: $\vec{\phi} = \begin{pmatrix} \phi_a = 0.75 / 0.62 \\ \phi_b = 0.65 / 0.52 \\ \phi_{p(i)} = 0.40 / 0.30 \\ \phi_i = 0.13 / 0.00 \\ \phi_{s(i)} = 0.87 / 0.80 \end{pmatrix}$

- Desynchronized: $\vec{\phi} = \begin{pmatrix} \phi_a = 0.40 \\ \phi_b = 0.20 \\ \phi_{p(i)} = 0.00 \\ \phi_i = 0.80 \\ \phi_{s(i)} = 0.60 \end{pmatrix}$



Multi-hop Considerations

Multi-hop topologies: **Hidden Node Problem:**



Our Solution:

Each node broadcasts its list of one-hop neighbors at its firing
➔ Potential hidden (*two-hop*) neighbors become known

(For deeper insight into EXTENDED-DESYNC, e.g. its adaptability, fault-tolerance and flexibility see C. Mühlberger, R. Kolla *Extended Desynchronization for Multi-Hop Topologies*, TR 460, Institut für Informatik, Universität Würzburg, July 2009)

Energy

How much energy can be saved and in what way?

C. Mühlberger
Chair of Computer Science V

General Acceptance

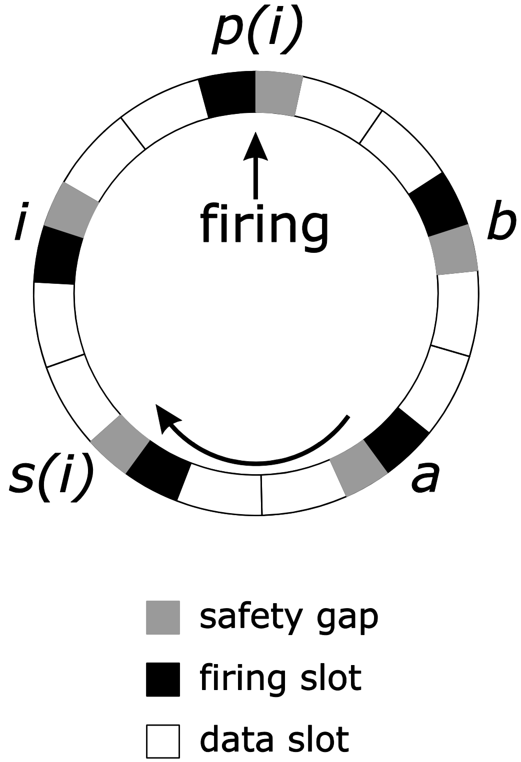
here: optimal desynchronized system

1. *T* into *n* frames *F(i)* of equal length |*F(i)*| = *f*
2. *F(i)* again into *k* slots *F(i, j)*, $1 \leq j \leq k$

   F(i, 1) of length f_f always *firing slot*,

   - to cover Hidden Node Problem, and
   - to stay „up-to-date“,

   remaining ones as *data slots* of length f_k
3. Safety gap σ between any frame, $|\sigma| = \varepsilon \cdot f_f$



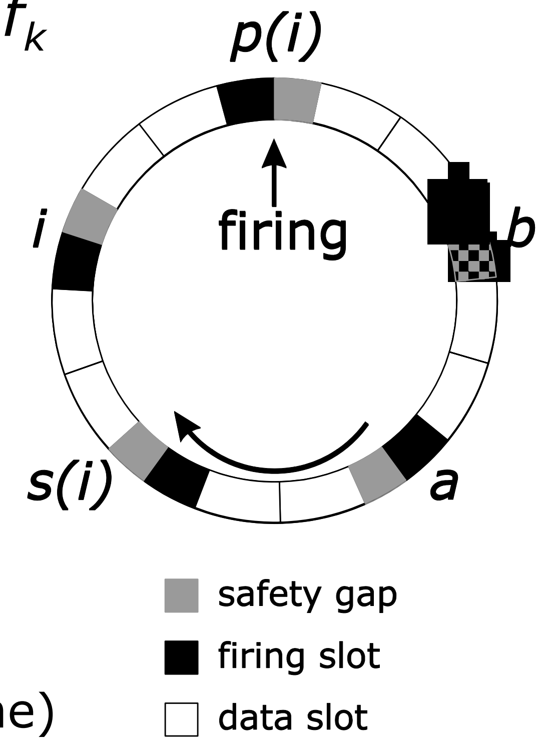
→ Turn off radio unit at unused or unattractive data slots to save energy!

Gain of Energy - I

Gain of energy $\gamma_i = \dfrac{T - \Delta t_{i,RF}}{T}$ at minimal period $T = n \cdot f$

a) Length of firing slot ≠ length of data slot f_k

$$\gamma_i = \frac{(k-1) \cdot f_k}{f}$$ (ratio of data slots per frame)



Just $\eta \le n-1$ firing slot mandatory

(e.g. all one-hop neighbors)

$$\gamma_i^{\eta} = \frac{\left(\frac{n-\eta-1}{n}\right) \cdot (1+\varepsilon) \cdot f_f + (k-1) \cdot f_k}{f}$$

(ratio of data and unimportant firing slots per frame)

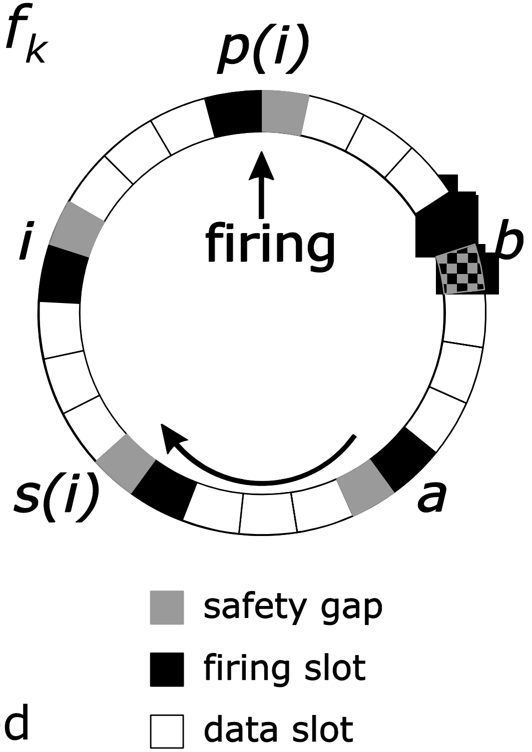
Gain of Energy - II

Gain of energy $\gamma_i = \frac{T - \Delta t_{i,RF}}{T}$ at minimal period $T = n \cdot f$

b) Length of firing slot = length of data slot f_k

$$\gamma_i = \frac{k-1}{k+\varepsilon}$$

(number of data slots divided by total number of slots and safety gap factor)



Just $\eta \leq n-1$ firing slot mandatory

(e.g. all one-hop neighbors)

$$\gamma_i^{\eta} = \frac{k - 1 + (1+\varepsilon)\cdot\left(\frac{n-\eta-1}{n}\right)}{k+\varepsilon}$$

(number of data and unimportant firing slots divided by total number of slots and safety gap factor)

If there are no data slots (*k*=1) to power down radio unit, there is just little energy-saving possible!

Further Savings?

- Power-down radio unit for several periods and keep old slots
  - Clock drift may now cause collisions
  - Additional administrative costs

- Power-down radio unit for several periods but leave network
  - Costly re-joining (completely new frame/slot assignment)

- Pad period *T* out to prolong sleep time
  - Also increases latency

Latency

What's the minimal latency and
it is influenced by what?

C. Mühlberger
Chair of Computer Science V

Observations

- Firing packets contain list of one-hop neighbors

➔ Length of firing slot depends on n ➔ $f_f = \beta \cdot n$

a) Just firing slots, but no data slots

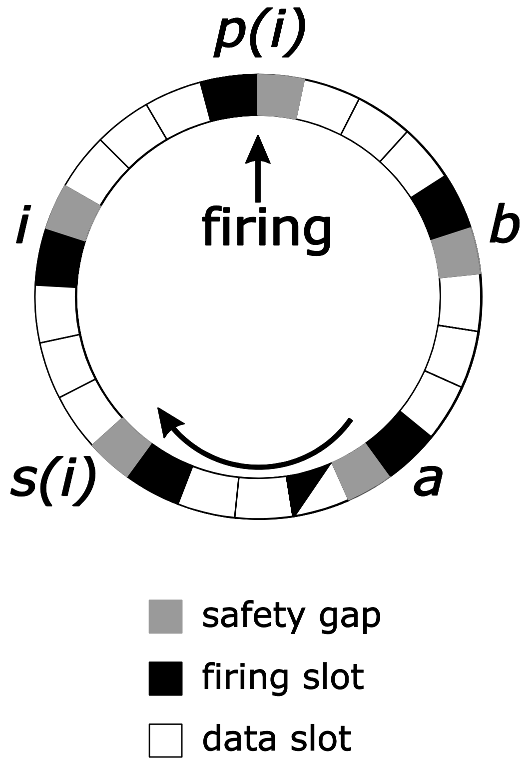
$$T = (1+\varepsilon) \cdot \beta \cdot n^2$$

b) Firing slots and data slots (of length δ)

$$T = (1+\varepsilon) \cdot \beta \cdot n^2 + \delta \cdot n$$

c) Firing slot as **base unit**, i.e. $\delta = \delta_0 \cdot \beta \cdot n$

$$T = (1+\varepsilon+\delta_0) \cdot \beta \cdot n^2$$



Period *T* grows with the square of *n*!

➔ Trade-off between energy savings and latency

Conclusion & Outlook

What was shown and

what is subject to future research?

C. Mühlberger
Chair of Computer Science V

Conlcusion & Outlook

Conclusion:

- EXTENDEND-DESYNC as TDMA MAC protocol:
  - Framework and operating mode
- Energetic and Temporal Analysis:
  - ➔ **Trade-off** Energy-saving ⇔ Latency
  - ➔ Save most energy during data slots
  - ➔ Period *T* strongly depends on *n*

Outlook:

- Balance out some parameters for real-world employments
- Complete, stabilize and strengthen our multi-hop extension
- Integrate further services, e.g. time synchronization, routing

Thank You!

Any questions?