DecTDMA: A Decentralized-TDMA with Link Quality Estimation for WSNs

SSS 2016

Olaf Landsiedel Thomas Petig Elad M. Schiller

Chalmers University of Technology and Gothenburg University

November 19, 2016

Motivation

- Due to IoT we got more interest in WSNs (wireless sensor networks)
- Most approaches do either:
  - centralized TDMA
  - (decentralized) CSMA

What about decentralized TDMA?

Outline

The Problem, The Challenge and Our Approach

Our Contribution

Algorithms

Experimental Results

Conclusions

Outline

The Problem, The Challenge and Our Approach

Our Contribution

Algorithms

Experimental Results

Conclusions

The Challenge

We need to solve:

1. clock synchronization ($\Rightarrow$ time slot alignment),
2. time slot assignment.

The Challenge

Collisions: concurrent transmissions might lead to packet omission.

We do not consider:

- external time references [Herman-Tixeuil ALGOSENSORS'04],
- external location references [Viqar-Welch ALGOSENSORS'09],
- collision detection,
- base stations for scheduling transmissions.

The Challenge

We have to show that communication is possible!

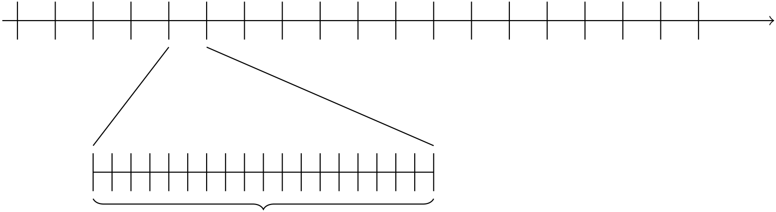
Collisions: concurrent transmissions might lead to packet omission.

We do not consider:

- external time references [Herman-Tixeuil ALGOSENSORS'04],
- external location references [Viqar-Welch ALGOSENSORS'09],
- collision detection,
- base stations for scheduling transmissions.

TDMA Frame

Time is divided into: slots and frames:



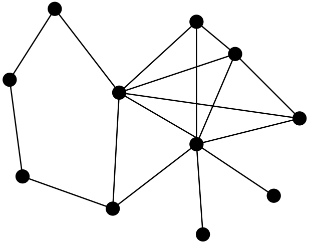
frame: τ slots

A slot can be used for a data packet or a control packet.

A data packet are send on a fixed slot within a frame.

The Problem

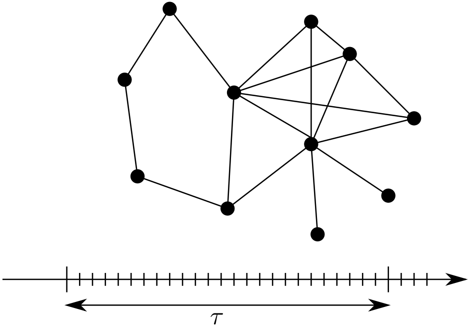
Given a communication graph:



construct a distance-2 coloring for TDMA slot allocation.

The Problem

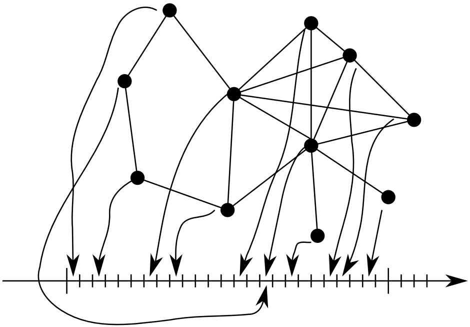
Given a communication graph:



construct a distance-2 coloring for TDMA slot allocation.

The Problem

Given a communication graph:



construct a distance-2 coloring for TDMA slot allocation.

Our Approach

We focus on the implementation of (self-stabilizing) algorithms that their converges considers both:

- clock synchronization, and
- time slot assignment.

Theoretical results can be found in "Self-stabilizing TDMA algorithms for wireless ad-hoc networks without external reference." by Petig, et. al. in MED-HOC-NET'14.

Outline

The Problem, The Challenge and Our Approach

Our Contribution

Algorithms

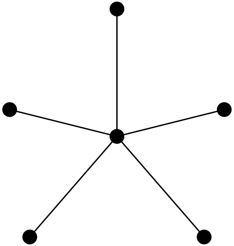
Experimental Results

Conclusions

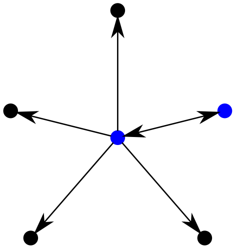
Our Contribution

- Implementation of a Self-Stabilizing algorithm on real hardware.
  - Well, the underlaying system is not self-stabilizing, so the algorithm isn't anymore, either.
- Link-Quality-Estimation (LQE) for acknowledgements.
- Experimental comparisons on the effect of LQE.

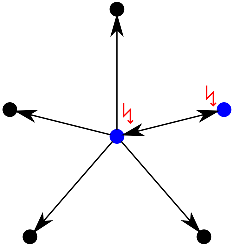
Hidden Neighbors



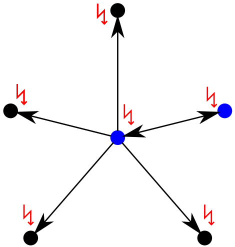
Hidden Neighbors



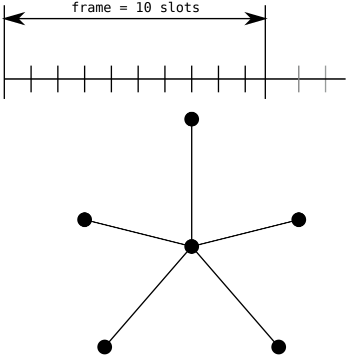
Hidden Neighbors



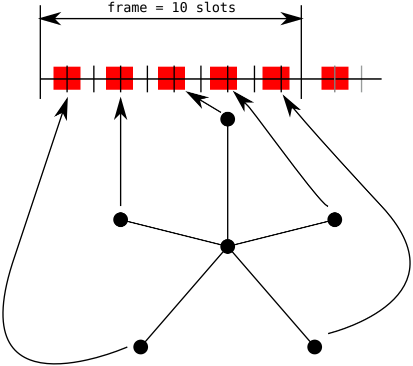
Hidden Neighbors



Hidden Neighbors

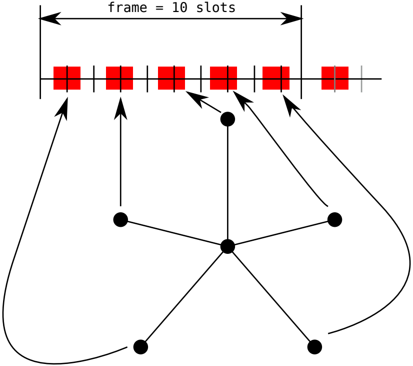


Hidden Neighbors



Hidden Neighbors

The δ leaves can block up to 2δ slots. We consider 2δ as lower bound rather than 2Δ [Busch, et al. Distributed Comp. '08], where $\Delta \in \mathcal{O}(\delta^2)$



Outline

The Problem, The Challenge and Our Approach

Our Contribution

Algorithms

Experimental Results

Conclusions

Algorithm

Upon packet reception:

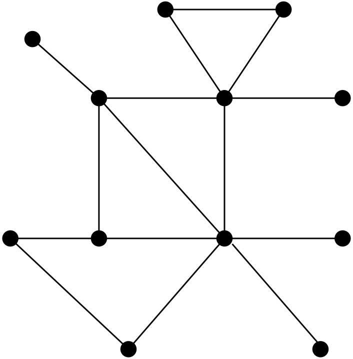
1. if remote clock $\gg$ my clock then adjust and drop timeslot
2. if remote clock $>$ my clock then adjust
3. check acknowledgement (drop timeslot if missing)
4. Merge neighborhood information.

Upon timeslot:

1. If assigned TDMA timeslot then transmit.
2. If randomly chosen free timeslot
    2.1 transmit
    2.2 chose new random timeslot
    2.3 If no TDMA timeslot assigned then take this one

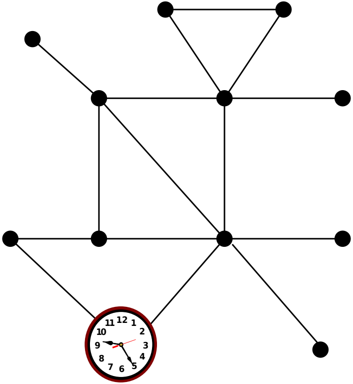
Algorithm

Clock synchronization.



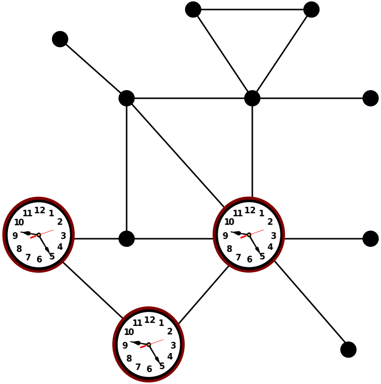
Algorithm

Clock synchronization.



Algorithm

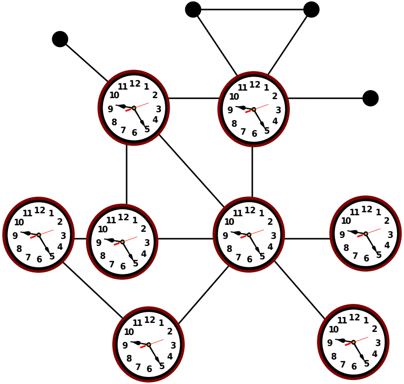
Clock synchronization.



.

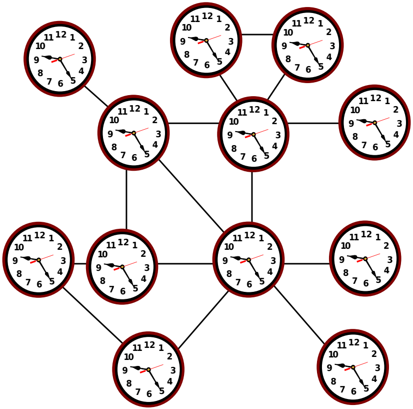
Algorithm

Clock synchronization.



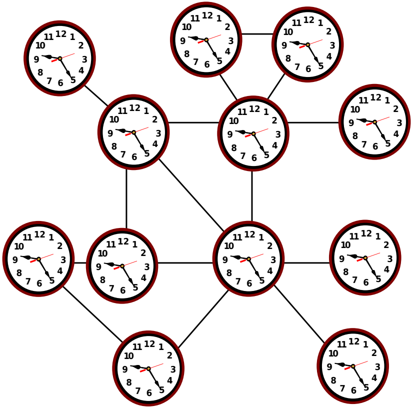
Algorithm

Clock synchronization.



Algorithm

Clock synchronization.



⇒ All clocks are synchronized!.

Algorithm

Upon packet reception:

1. ~~if remote clock $\gg$ my clock then adjust and drop timeslot~~
2. if remote clock $>$ my clock then adjust
3. check acknowledgement (drop timeslot if missing)
4. Merge neighborhood information.

LQE comes in line 3.

LQE

(We received a packet from node *p*.)

1. If ≥ 16 packets from *p* received during last 20 rounds:
    1.1 If < 8 packets acknowledge my transmission:
        1.1.1 Drop timeslot and backoff.

Algorithm

During legal executions:

- TDMA time slots are aligned,
- each node successfully sends data packets once a frame,
- control packets do not collide.

Outline

The Problem, The Challenge and Our Approach

Our Contribution

Algorithms

Experimental Results

Conclusions

TinyOS

Figure: Courtesy of TinyOS Alliance.

Our implementation

- is based on TinyOS 2.1.1,
- is implemented in nesC,
- is targeting the MicaZ and TelosB platform.

But, first: Simulation.

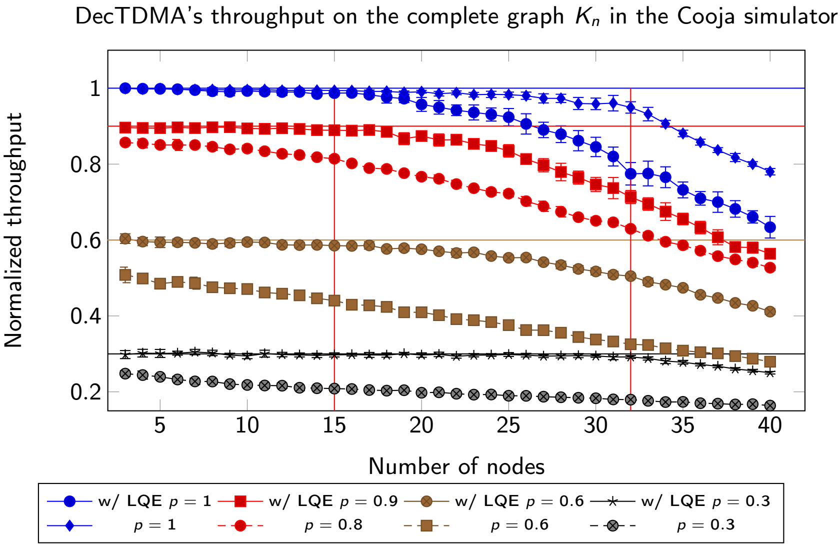
Cooja Simulator

Cooja is

- part of the Contiki toolchain,
- Simulates:
  - Mikrocontroller,
  - Transceiver,
  - Radio channel.

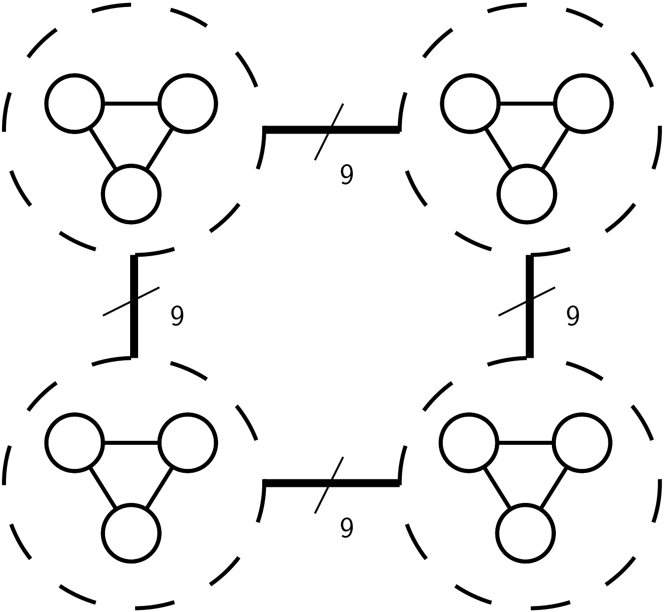
We use the unit disc graph (UDG) model.

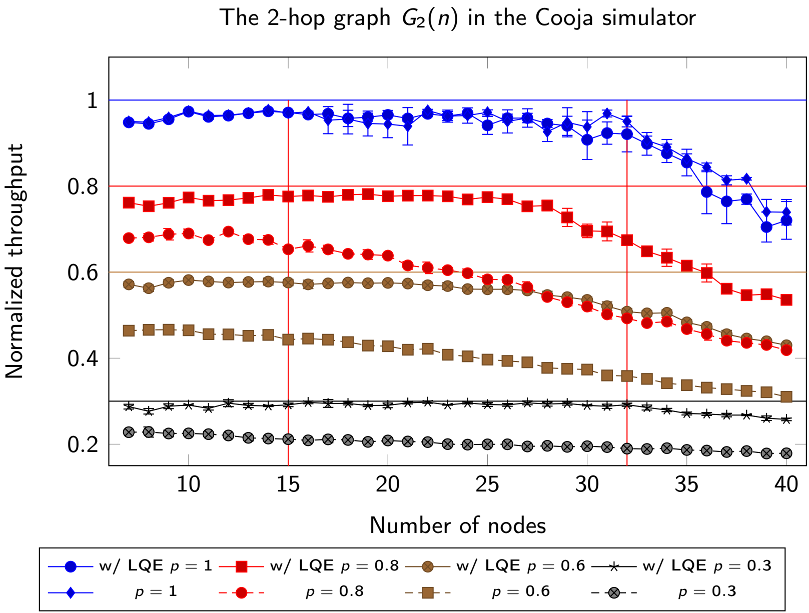
Nodes form a K_n, probability for successful transmission: $p \in [0.2, 1]$



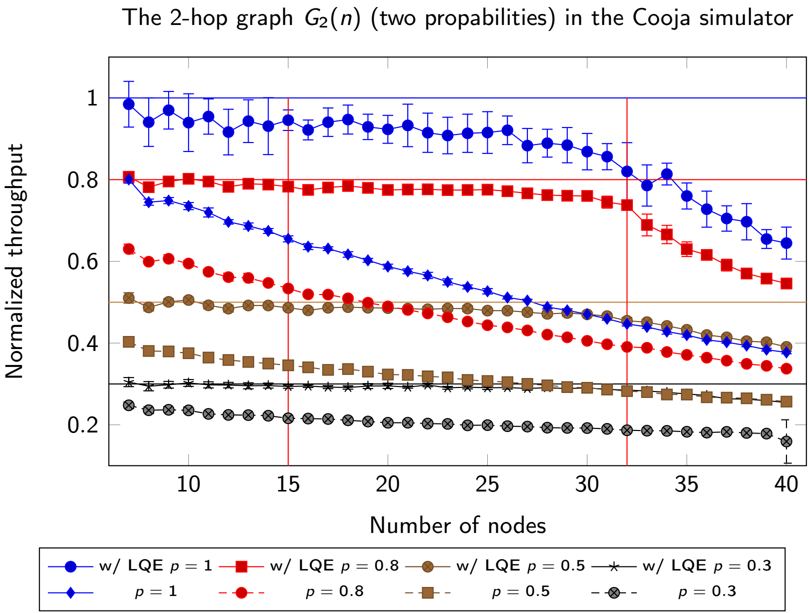
The throughput is normalized by $\#pkts \cdot T^{-1}(n-1)^{-1}2^{-1}n^{-1}$.

Communication graph used in simulation: $G_2(12)$





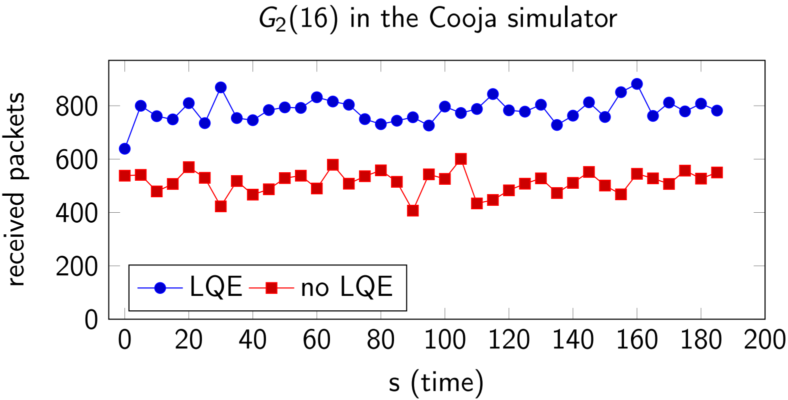
The throughput is normalized by $\#pkts \cdot T^{-1}(0.75n - 1)^{-1}2^{-1}n^{-1}$.



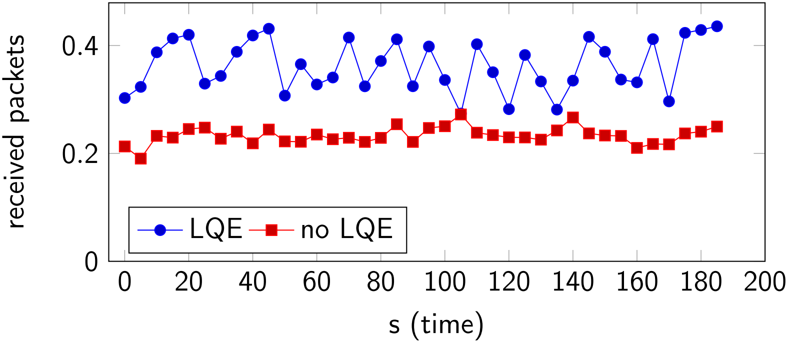
The throughput is normalized by $\#pkts \cdot T^{-1}(0.45n - 1)^{-1}2^{-1}n^{-1}$.

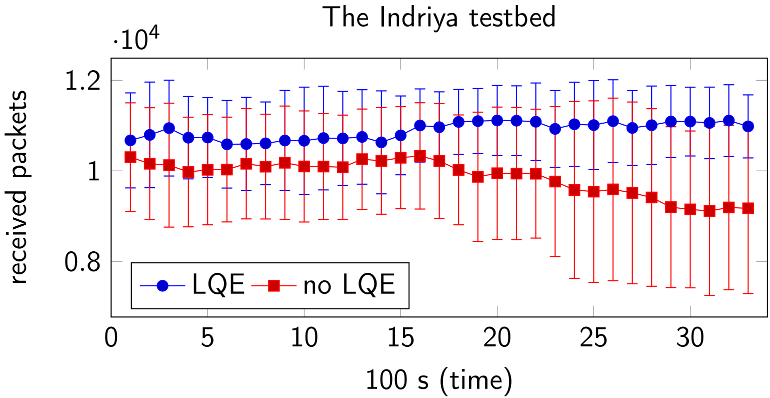
INDRYA Testbed at NSU

- 97 TelosB nodes
  - IEEE 802.11.4 radio
  - 2.4 GHz ISM band
  - 16-bit TI MSP430 microcontroller


$G_2(16)$ in the Cooja simulator
received packets
0
200
400
600
800
0
20
40
60
80
100
120
140
160
180
200
s (time)
LQE
no LQE

$G_2(40)$ in the Cooja simulator





Received packets are accumulated in intervals of 100 s and showed as an average over 10 run each. The error bars represent standard deviation.

Outline

The Problem, The Challenge and Our Approach

Our Contribution

Algorithms

Experimental Results

Conclusions

Conclusions

We do not consider:

- external time reference,
- location reference,
- collision detection,
- base station.

We demonstrate a MAC layer implementation, that allows the integration of high level protocols.

Thank you for your attention!

Contact us:

- Olaf Landsiedel: olafl@chalmers.se
- Thomas Petig: petig@chalmers.se
- Elad M. Schiller: elad.schiller@chalmers.se