DCOSS

ZeroCal: Automatic MAC Protocol Calibration

Andreas Meier <a.meier@tik.ee.ethz.ch>

joint work with

Matthias Woehrle, Marco Zimmerling, Lothar Thiele

Computer Engineering and Networks Laboratory, ETH Zurich

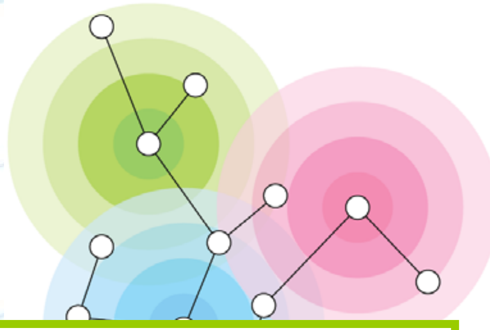
ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

NCCR MICS
National Competence Center in Research
Mobile Information and Communication Systems

FNSNF
Fonds national suisse
Schweizerischer Nationalfonds
Fondo nazionale svizzero
Swiss National Science Foundation

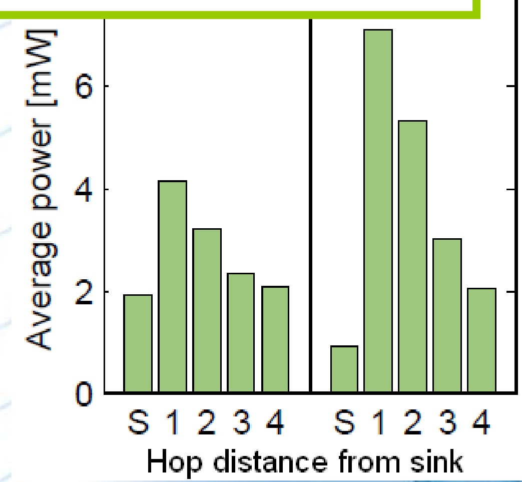
Data gathering in wireless sensor networks

- Sensor nodes sample data regularly
  - The data is sent to a common sink node requiring forwarding



How can we ensure that the configuration of the MAC protocol is optimized?

- T
  ir
  - The energy consumption is not evenly distributed
  - Nodes closest to the sink spend the most energy
  - The energy consumption depends on the MAC parameterization (T_s)
  - Impacts network lifetime

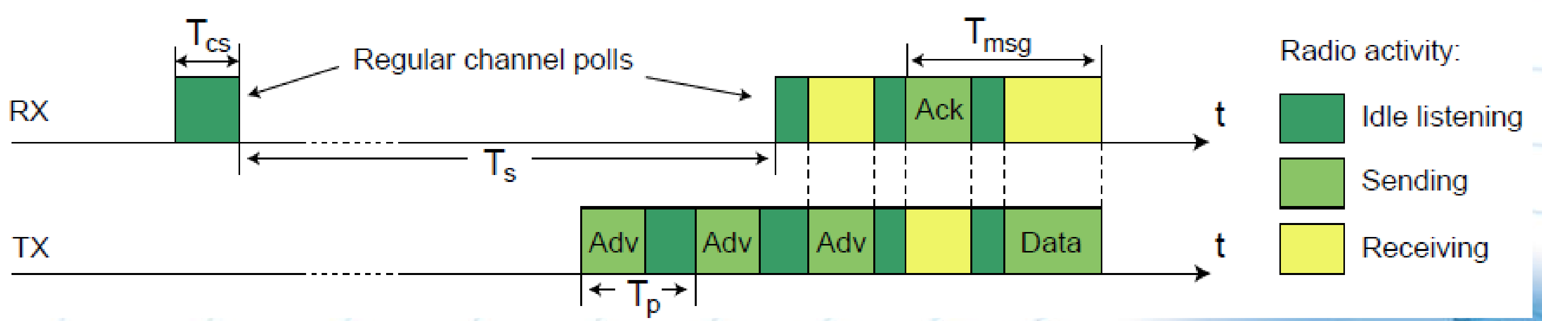


Related work

- Polastre et al. [SenSys'04]
  - Analytical model for adapting B-MAC for a certain traffic
  - Implementation of adaptation service is deferred to upper layer
- Buettner et al. [SenSys'06]
  - Optimize single sender-receiver pair
  - Cannot be used for general networks (e.g. mesh and trees)
- Jurdak et al. [TMC vol. 6, 2007]
  - Energy optimization through greedy local decisions
  - We show that it is beneficial not to make greedy decisions but to consider both parent and child nodes
- Merlin et al. [MASS'08]
  - Control theoretic approach
  - Only suitable for single-hop topologies

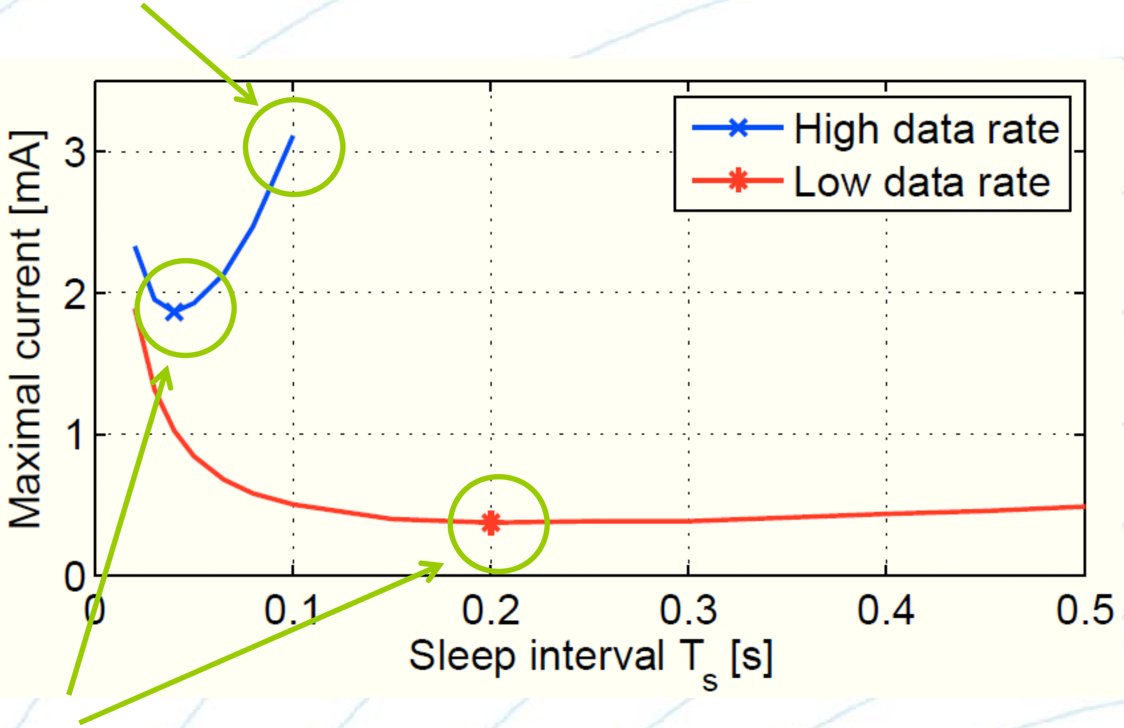
Background: Low-Power Listening MAC protocols (X-MAC)

- Receiver wakes-up every sleep interval T_s
  - The node keeps listening if the channel is busy
- The sender sends a long packet stream
  - The stream ends after $\sim T_s$ or after receiving an acknowledgment
- Increasing a parent node's T_s:
  - Decreases the parent's energy consumption for channel polling
  - Increases the child nodes' transmission energy



MAC parameters determine energy and bandwidth

- MAC parameterization dictates
  - Energy demands
  - Maximum data load



- Optimal MAC parameters
  - Depend on communication rate, traffic and link qualities

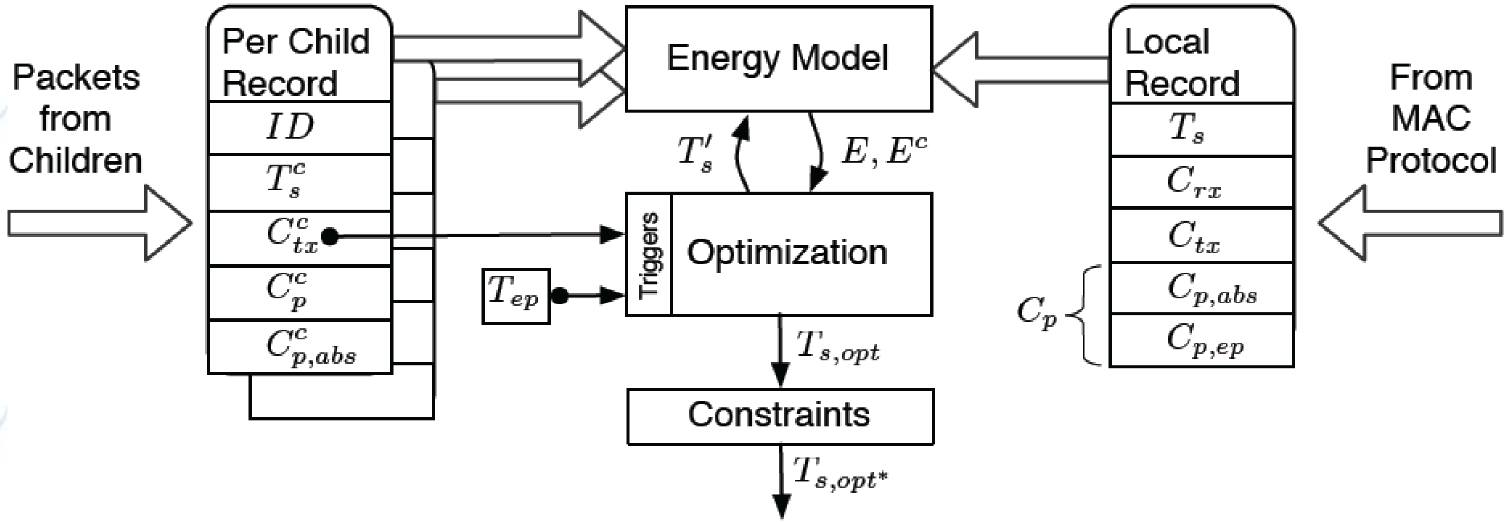
How to set the optimal MAC parameters?

- Traditionally the MAC parameters are set at compile time
  - Requires know-how from the application designer
  - Requires to anticipate the traffic of the network deployment
  - If the traffic is wrongly anticipated:
    - Energy is wasted
    - Packets are lost due to insufficient bandwidth

- ZeroCal: The parameter setting is observed regularly **at runtime** and updated if necessary
  - Every node tries to minimize **both** its own energy and the one of its children
  - This requires on-line **estimation of the energy** spent by the node itself and of if its children

ZeroCal Architecture

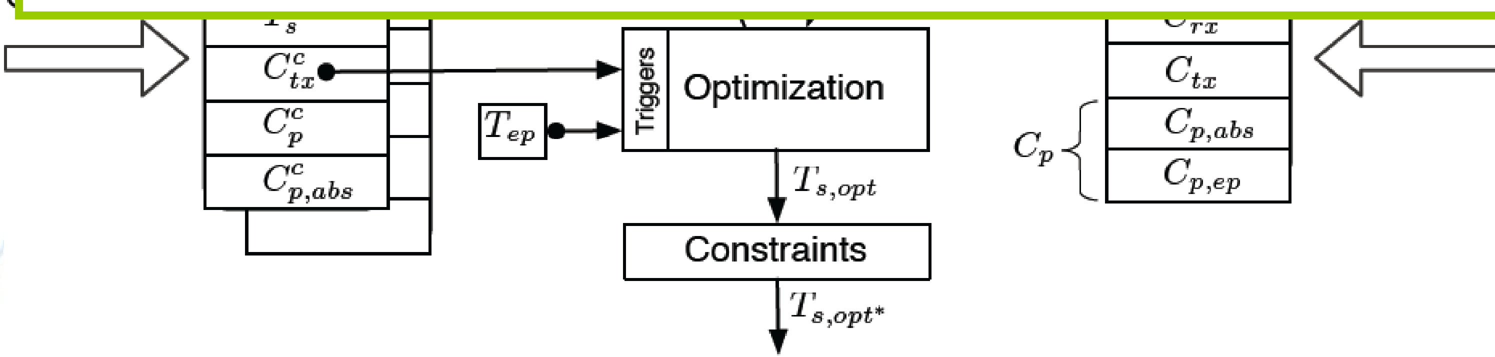
- Keep record of local and child statistics
- Optimization process is performed
  - Traffic based (or periodic if traffic is low)
- Compute a new optimal sleep interval
  - Using an energy model
  - Respect bandwidth and protocol constraints



ZeroCal Architecture

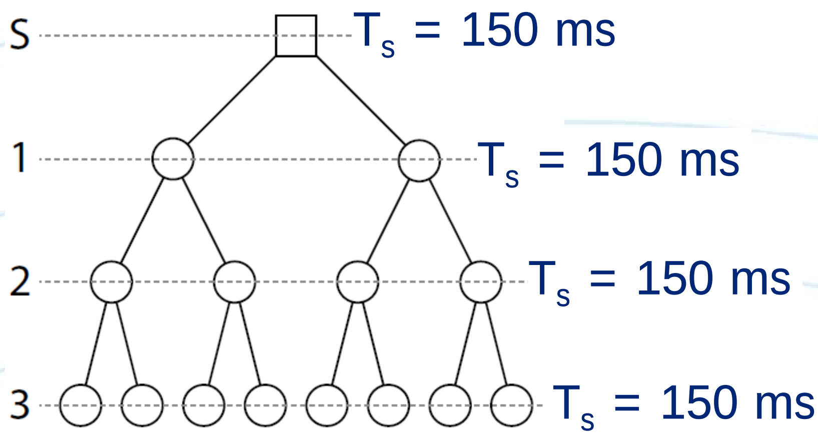
- Keep record of local and child statistics
- Optimization process is performed
  - Traffic based (or periodic if traffic is low)
- Compute a new optimal sleep interval

ZeroCal: Different nodes will have different parameter settings: A parent node will have a shorter T_s than its children

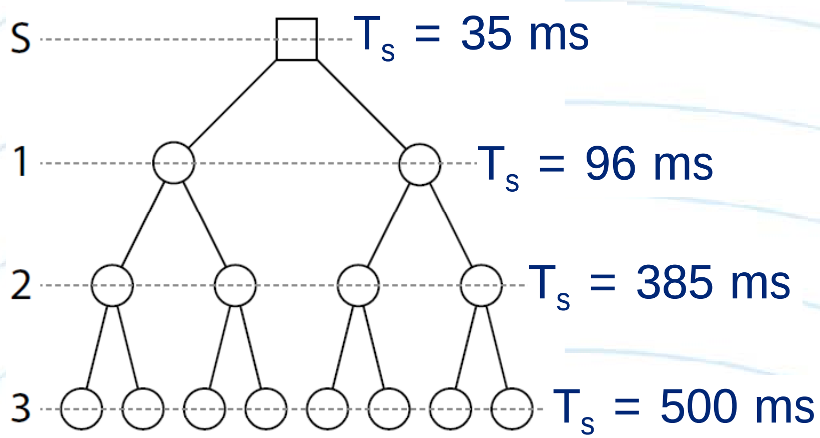


Identical vs. ZeroCals varying parameter setting

- Identical setting
  - Set at development time
  - Same setting for all nodes
  - Does never change



- ZeroCal
  - Different nodes will have different parameters
  - Optimize parameter setting at runtime
  - Adapt to changes in the traffic and environment



Estimate energy consumption with counters

- Energy estimation is based on our previous work
  - Analyzing MAC Protocols for Low Data-Rate Applications [Langendoen et al., ToSN to appear]
- Time for transmitting messages

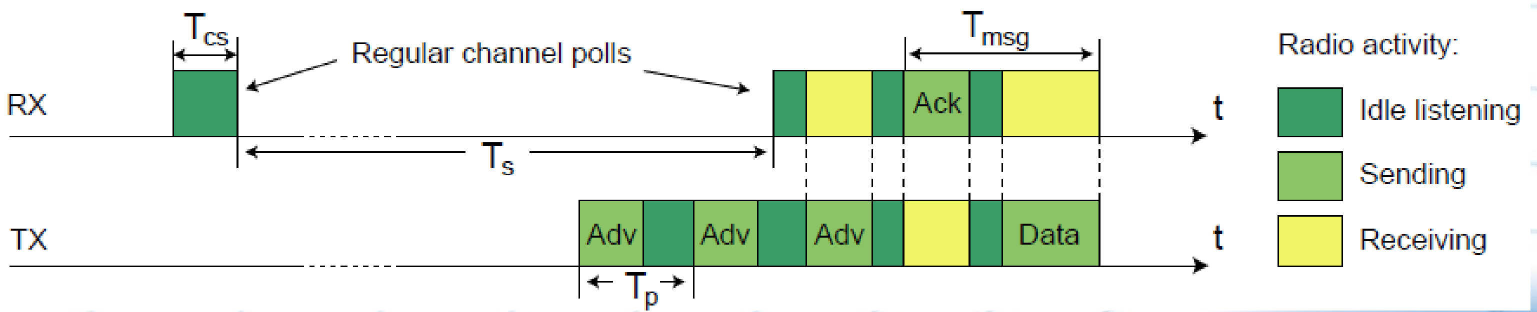
$$T_{tx} = C_p \cdot T_p + C_{tx} \cdot T_{msg}$$

- Time for receiving messages

$$T_{rx} = C_{rx} \cdot T_{msg}$$

- Time for idle listening

$$T_{cp} = (T_{ep} - T_{tx} - T_{rx}) \cdot T_{cs} / (T_s + T_{cs})$$

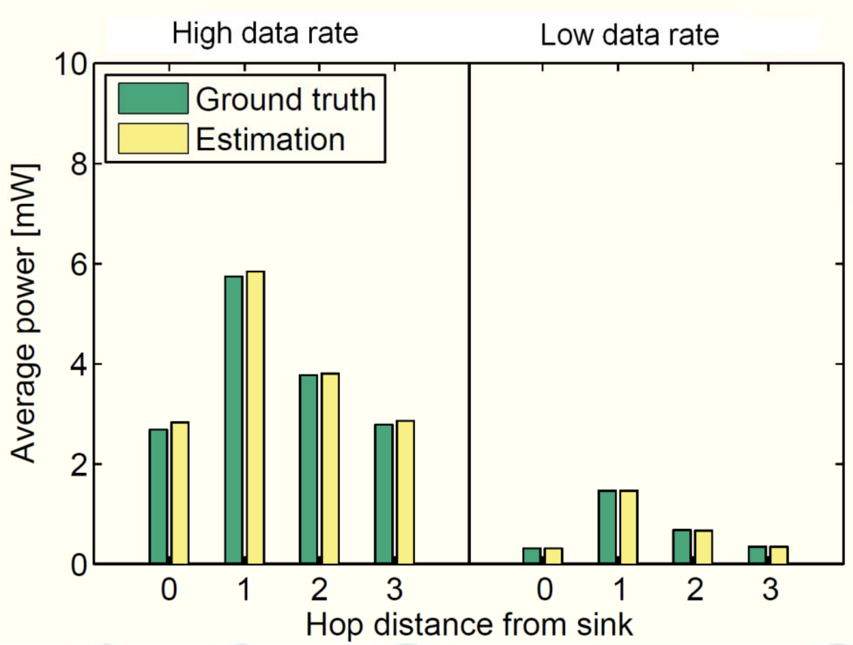


Energy estimation continued

- Calculate energy using the average power consumption for the different states:

$$E = T_{tx} \cdot P_{tx} + T_{rx} \cdot P_{rx} + T_{cp} \cdot P_{cp}$$

- Maximum estimation error:
  - 4.1% for high data rates
  - 1.8% for low data rates



Increasing a parent's sleep interval from T_s to T'_s

- Decreases the parent node's idle-listening time to:

$$T_{cp} = (T_{ep} - T_{tx} - T_{rx}) \cdot T_{cs}/(T'_s + T_{cs})$$

- Increases the child node's transmission time to:

$$T^c_{tx} = C^c_p \cdot T^c_p \cdot T'_s/T_s + C^c_{tx} \cdot T_{msg}$$

There is an optimal sleep interval $T_{s,opt}$:

$$T_{s,opt} = \operatorname*{argmin}_{T'_s \in [T_{s,min}, T_{s,max}]} \max\left[E, \max_{\forall \text{ children } c} (E^c)\right]$$

Two additional constraints for the sleep interval

1. Protocol constraint: The parent's sleep interval must not be longer than that of any child node

$$T_{s,opt*} \leq T_s^c, \quad \forall \text{ children } c$$

2. Bandwidth constraint: A message is sent or received in at most every n-th sleep interval

$$T_{s,opt*} \leq 1/(C_{tx} + C_{rx})/n$$

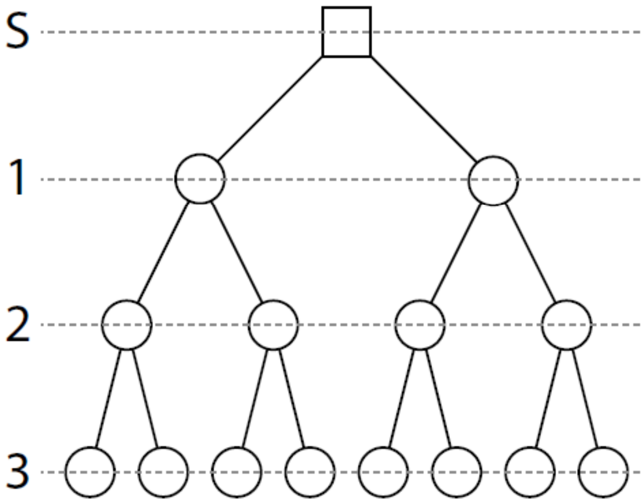
The optimal sleep interval $T_{s,opt*}$ is evaluated if either:

Periodic timeout: $T_{ep} > T_{ep,max}$

Traffic dependent: $C_{rx} > C_{eval}$

Evaluation in simulation

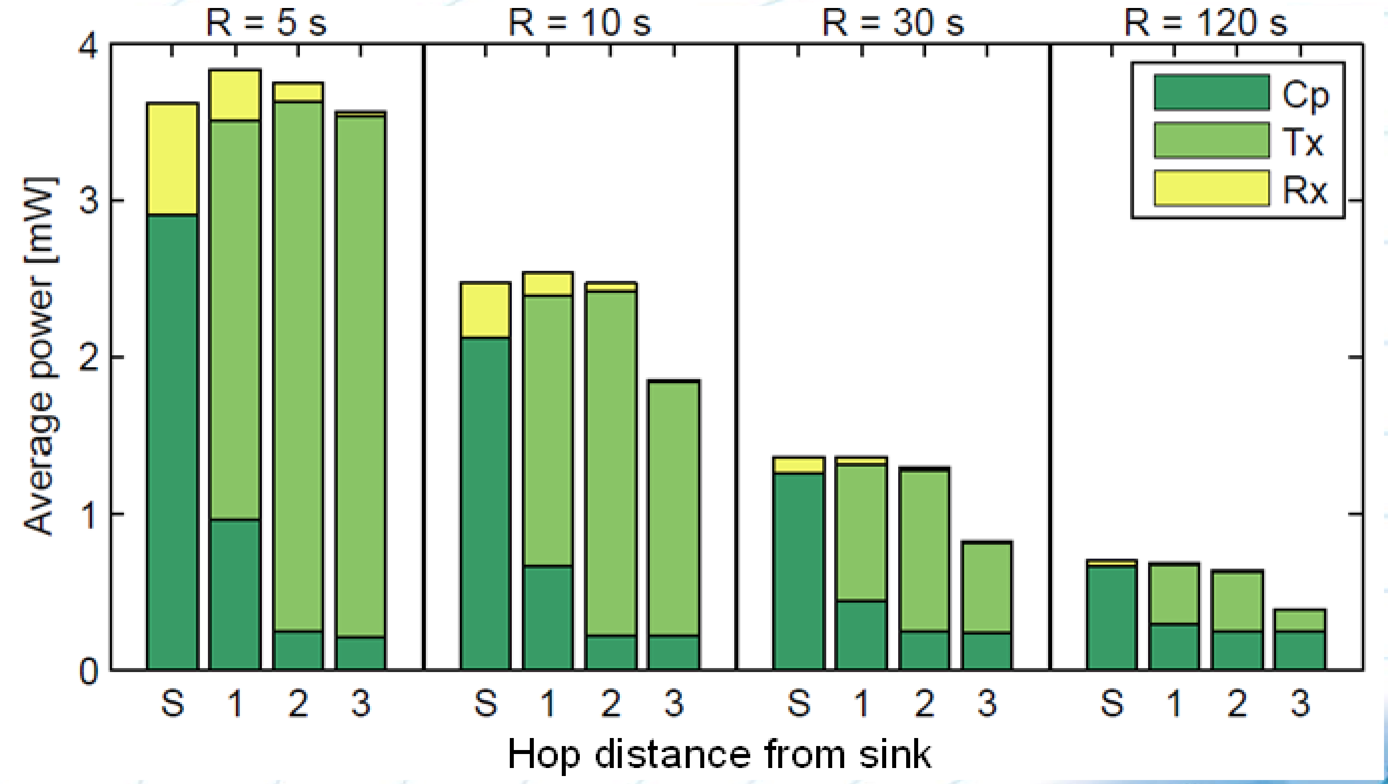
- Castalia simulator
  - X-MAC
- Complete binary tree
  - Height 3 (15 nodes)
  - Root is the sink
- Link Quality: PRR = 0.9
- Data rate: R = [5,120] s



Simulations with large and irregular topologies show the same trend as the binary tree.

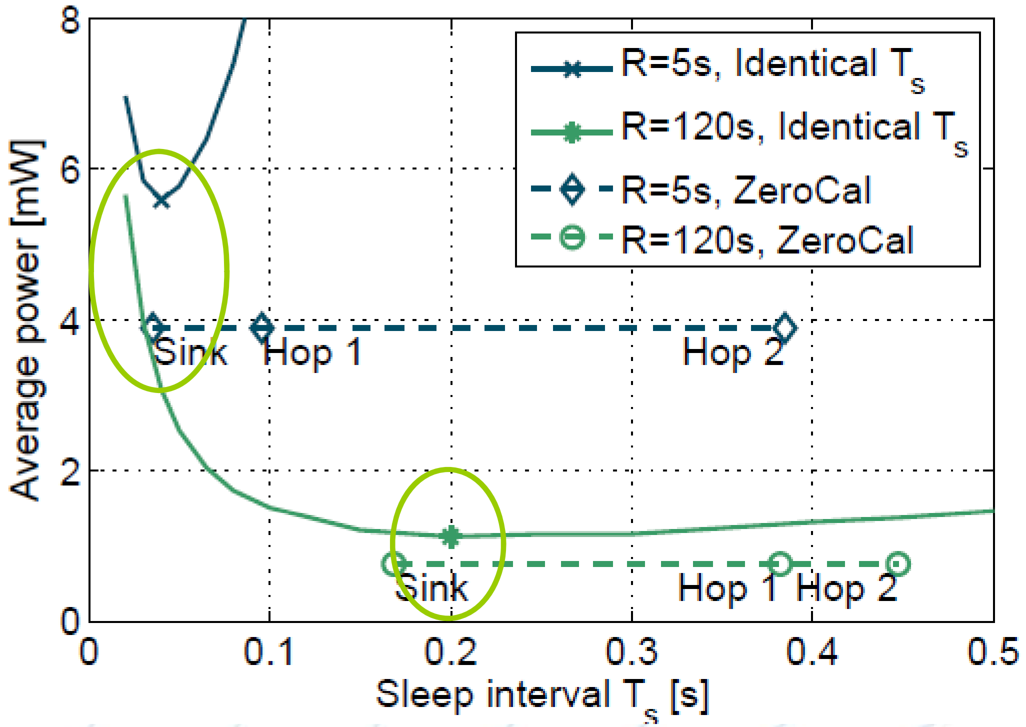
Simulation: Energy consumption using ZeroCal

- Well balanced energy consumption for low (R=120s) and high (R=5s) data rates
- The distribution of the energy consumption changes
  - Channel polling vs. transmission vs. reception



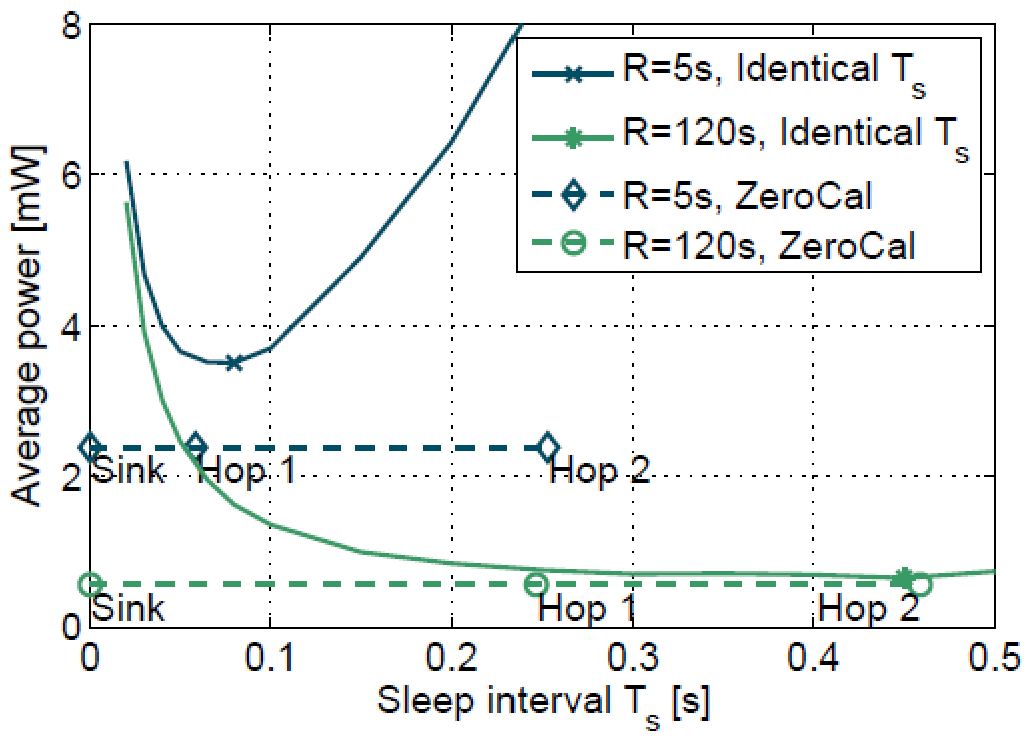
Simulation: ZeroCal vs. identical configuration

- ZeroCal outperforms the ‘hard-to-achieve’ optimal identical configuration by more than 30%
  - For low and for high data rates



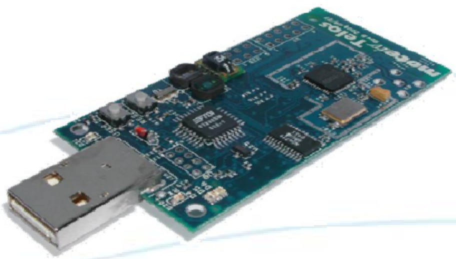
Simulation: Always listening sink

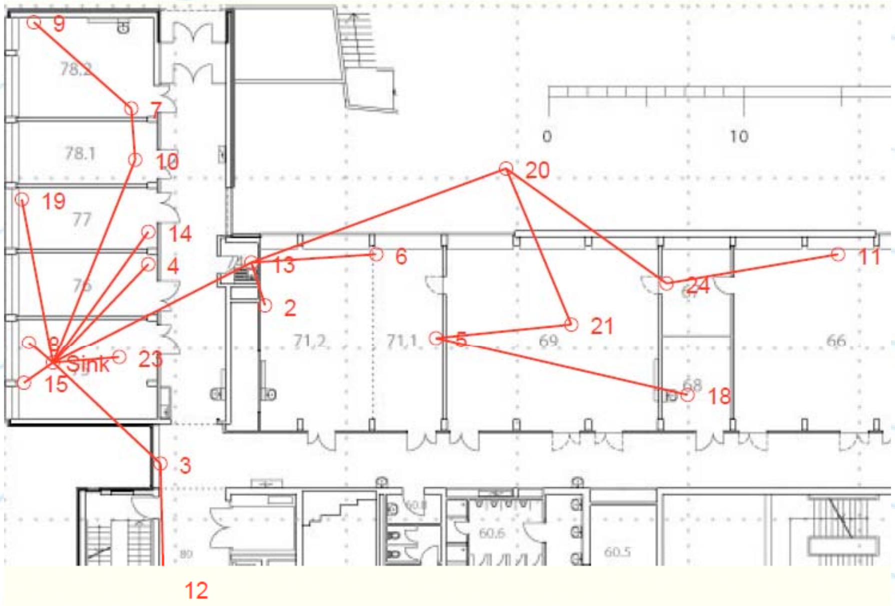
- The sink node has often an unlimited power supply
  - Allows for always listening (T_s=0 s)
- The adaptive duty cycling automatically adapts to this new scenario without any changes in the code



Testbed evaluation

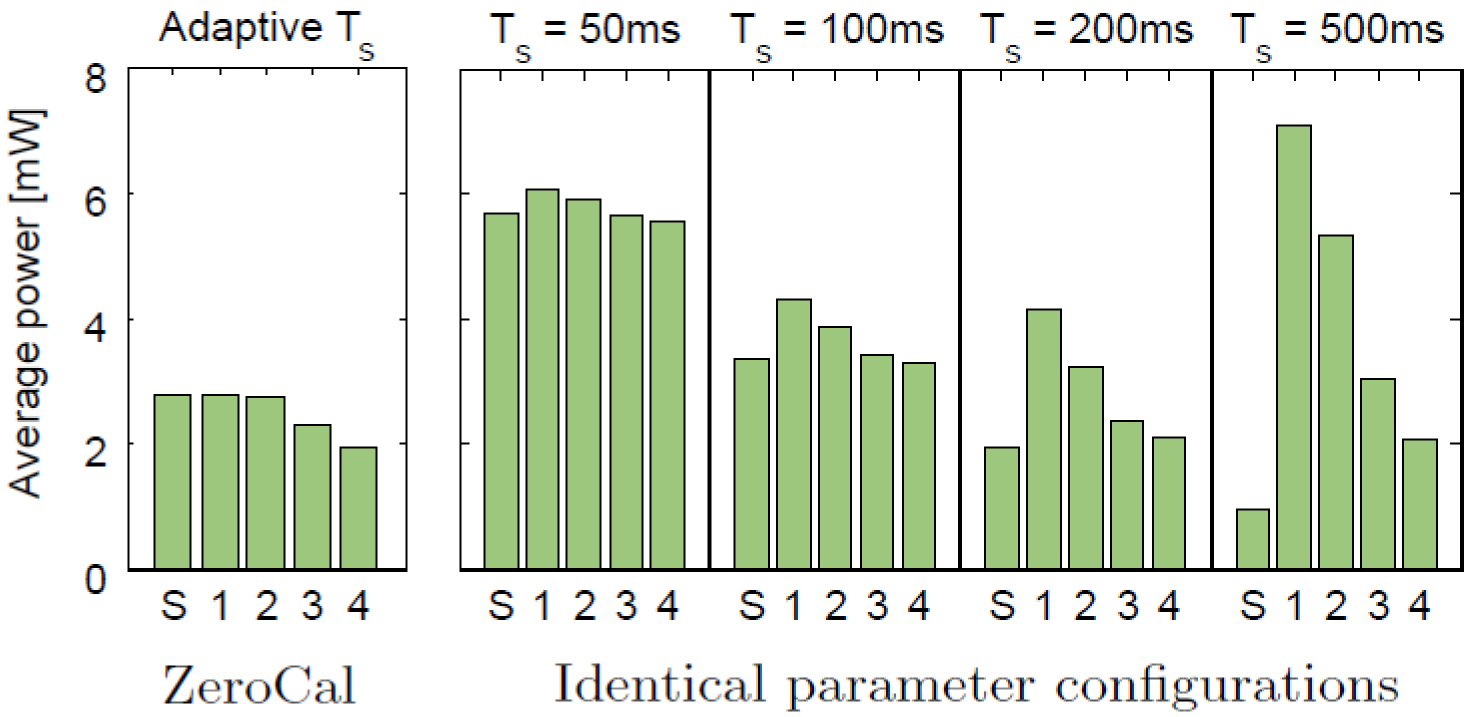
- 21 Tmote sky nodes
- Contiki operating system
  - X-MAC protocol
- Static topology
  - Up to 5 hops
- Data rate:
  - R = {10, 30, 120} s





Testbed: Zerocal vs. identical parameterization

- ZeroCal's energy consumption is
  - Largely reduced compared to the identical parameterization
  - Well balanced among all the hops

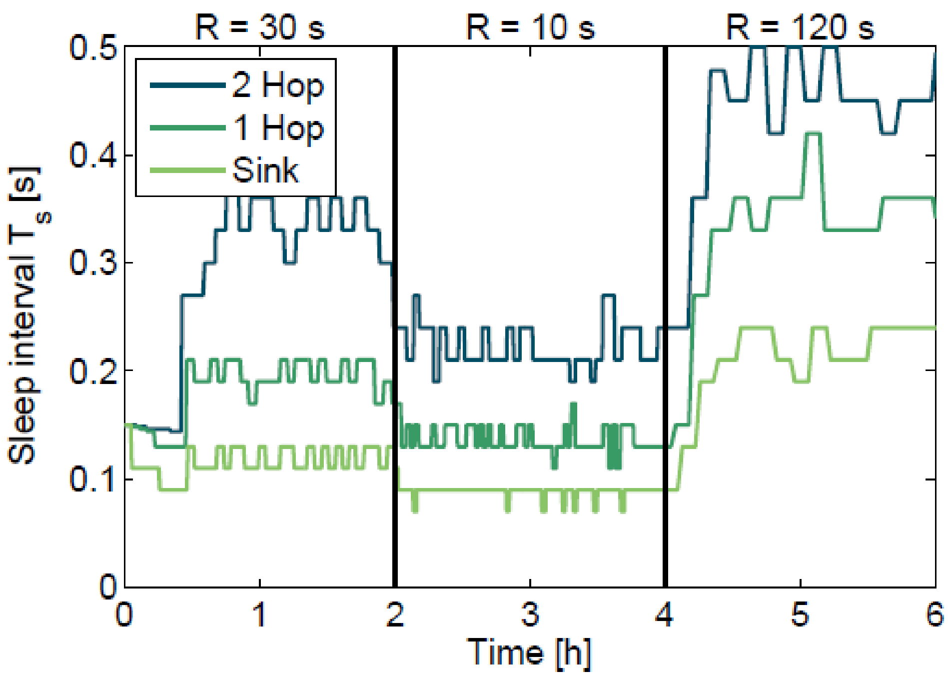


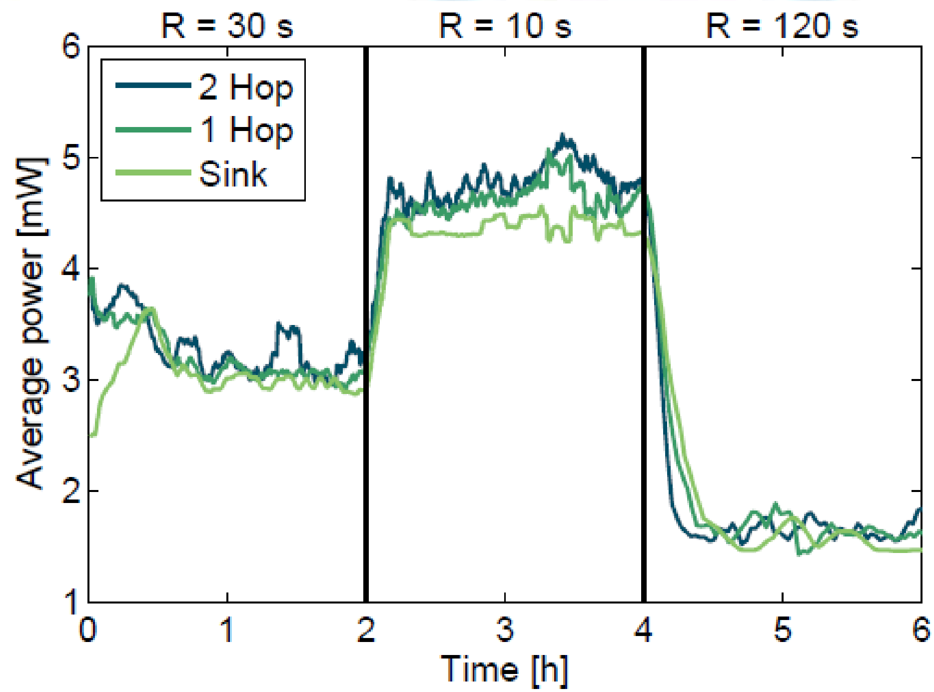
Datarate:

R = 30 s

Testbed: Adaptivity to network dynamics

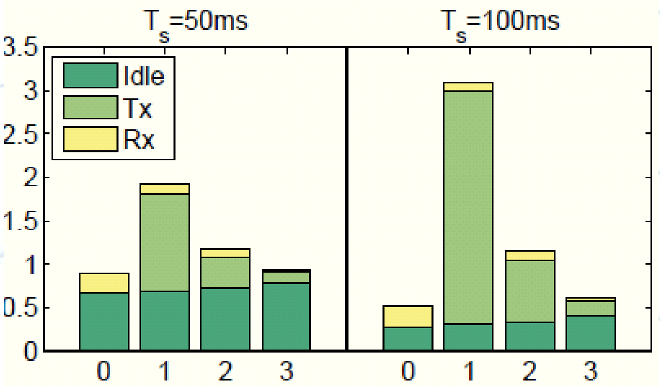
- The data rate R changes every two hours
  - R = [30, 10, 120] seconds
- The duty cycle adapts to the current setting
- The energy consumption is always well balanced





Conclusion

- It is difficult to find an <u>optimal</u> MAC configuration



- We showed an <u>adaptive</u> approach for optimizing the MAC parameters

$$T_{s,opt} = \underset{T'_s \in [T_{s,min}, T_{s,max}]}{\operatorname{argmin}} \max \left[E, \max_{\forall \text{ children } c} (E^c) \right]$$

- ZeroCal outperforms the optimal identic parameter setting.