Title
SYS4: Estimating Clock Uncertainty for Efficient Duty-Cycling in Sensor Networks

Permalink
https://escholarship.org/uc/item/4kv261jf

Authors
Saurabh Ganeriwal
Deepak Ganesan (UMass)
Hohyun Shim
et al.

Publication Date
2005
Estimating Clock Uncertainty for Efficient Duty-Cycling in Sensor Networks

Saurabh Ganeriwal, Deepak Ganesan, Hohyun Sim, Mani B. Srivastava, Mark Hansen
Networked Embedded Systems Lab – http://nesl.ee.ucla.edu

Introduction: Time Uncertainty Problem

Asynchronous approach (BMAC)

- Choose preamble such that receiver will wake up during the preamble transmission time.
- 11.5% duty cycle → 250 bytes or preamble, 2.2% duty cycle → 1212 bytes of preamble.
- The overhead of this 1212 byte preamble is too high for our 29 byte payload.

Synchronous approach (SMAC, TMAC)

- Time synchronized duty cycling of nodes.
- Available approaches propose to resynchronize after every minute.

Problem Description: Limitation of existing approaches to estimate clock uncertainty

- Two main drawbacks
  - Requires manual configuration for every new sensor network deployment.
  - Synchronization period is chosen to tackle the worst case, resulting in wasted energy.

Proposed Solution: Rate-adaptive long term time synchronization protocol

Synchronize a pair of nodes

- Fixed Preamble mode
  - BMAC uses a preamble of x bytes irrespective of duty cycle.
  - RATS maintains the time uncertainty within (x-4)*byte time.
- Variable Preamble mode
  - BMAC decides preamble on the fly!
  - Uses RATS to estimate the current time uncertainty between the nodes.

Uncertainty-driven duty cycling

RATS + BMAC = UBMAC

Performance Comparison

For an error bound of 90μs

<table>
<thead>
<tr>
<th>Hardware specs</th>
<th>FTSP</th>
<th>RATS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Drift</td>
<td>Resync period</td>
<td>Drift Estimate</td>
</tr>
<tr>
<td>5μs/s</td>
<td>18s</td>
<td>2μs/s</td>
</tr>
</tbody>
</table>

For a scenario of 1 event / day

Energy gain of 2800x due to time sync

<table>
<thead>
<tr>
<th>Hardware specs</th>
<th>FTSP</th>
<th>RATS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Overhead Net gain</td>
<td>Overhead Net gain</td>
<td>Overhead Net gain</td>
</tr>
<tr>
<td>~4800x</td>
<td>NA</td>
<td>~1920x</td>
</tr>
</tbody>
</table>

Evaluation of UBMAC on motes

Snapshot of energy consumption

- Energy gains of UBMAC increase with higher duty cycle
- For lower latency requirements, energy gains can be up to two orders of magnitude.

Packet loss rates

<table>
<thead>
<tr>
<th></th>
<th>First Child</th>
<th>Second Child</th>
<th>Always-on</th>
</tr>
</thead>
<tbody>
<tr>
<td>Indoor</td>
<td>2.2%</td>
<td>1.95%</td>
<td>1.92%</td>
</tr>
<tr>
<td>Outdoor</td>
<td>2.45%</td>
<td>3.1%</td>
<td>3.45%</td>
</tr>
</tbody>
</table>

- Loss rates are comparable.
- No bulk packet losses.