Tight Bounds on Delay-Sensitive Aggregation
Yvonne Anne Pignolet, Stefan Schmid, Roger Wattenhofer / TIK, ETH Zurich

Introduction
We investigate the fundamental trade-off between communication cost and delay cost arising in networks. We consider an optimization problem where nodes are organized in a tree topology. The nodes seek to minimize the time until the root is informed about their states and to use as few transmissions as possible at the same time. We derive an upper bound on the competitive ratio of $O\left(\min(h,c)\right)$ where $h$ is the tree's height, and $c$ is the transmission cost per edge. Moreover, we prove that this upper bound is tight in the sense that any oblivious algorithm has a ratio of at least $\Omega\left(\min(h,c)\right)$.

Environmental Monitoring
Goal: Up-to-date information

Recharging of sensor nodes can be tedious or even impossible. Unfortunately the radio communication system uses a large amount of energy and we need algorithms with as few transmissions as possible.

Algorithm
Balance delay cost and total communication cost

AGG: forward msg $m$ as soon as delay($m,t$) > $c$.

Definition: Oblivious algorithm
Decision (transmit/wait) of node $v$ depends on
• # events currently at node $v$
• Information stored in messages

Decision of node $v$ does NOT depend on
• History (messages forwarded earlier)
• $v$'s location in the aggregation network

perfect for sensor nodes!

Results
• competitive ratio of AGG: $O\left(\min(c,h)\right)$
• competitive ratio of best oblivious algo: $\Omega\left(\min(c,h)\right)$

Upper bound: We compare the number of messages sent by the AGG and the optimal offline algorithm as well as the accumulated delay cost. In short, merging less than AGG, bad for OPT since that results in too many transmissions, and merging more is bad because the delay increases to much.

Lower bound: Assume that an event occurs at all the leaves at the same time. In this case the messages are forwarded separately to the root by any oblivious algorithms and the total cost amounts to $2ch$. The minimal cost of the optimal offline algorithm is at least $2ch$ (communication cost) plus $h/2$ (delay cost).

Reference