On Decentralized In-Network Aggregation in Real-World Scenarios with Crowd Mobility

M. Gregorczyk, T. Pazurkiewicz, K. Iwanicki

University of Warsaw

DCOSS 2014, Marina Del Rey, CA, USA, May 26th, 2014

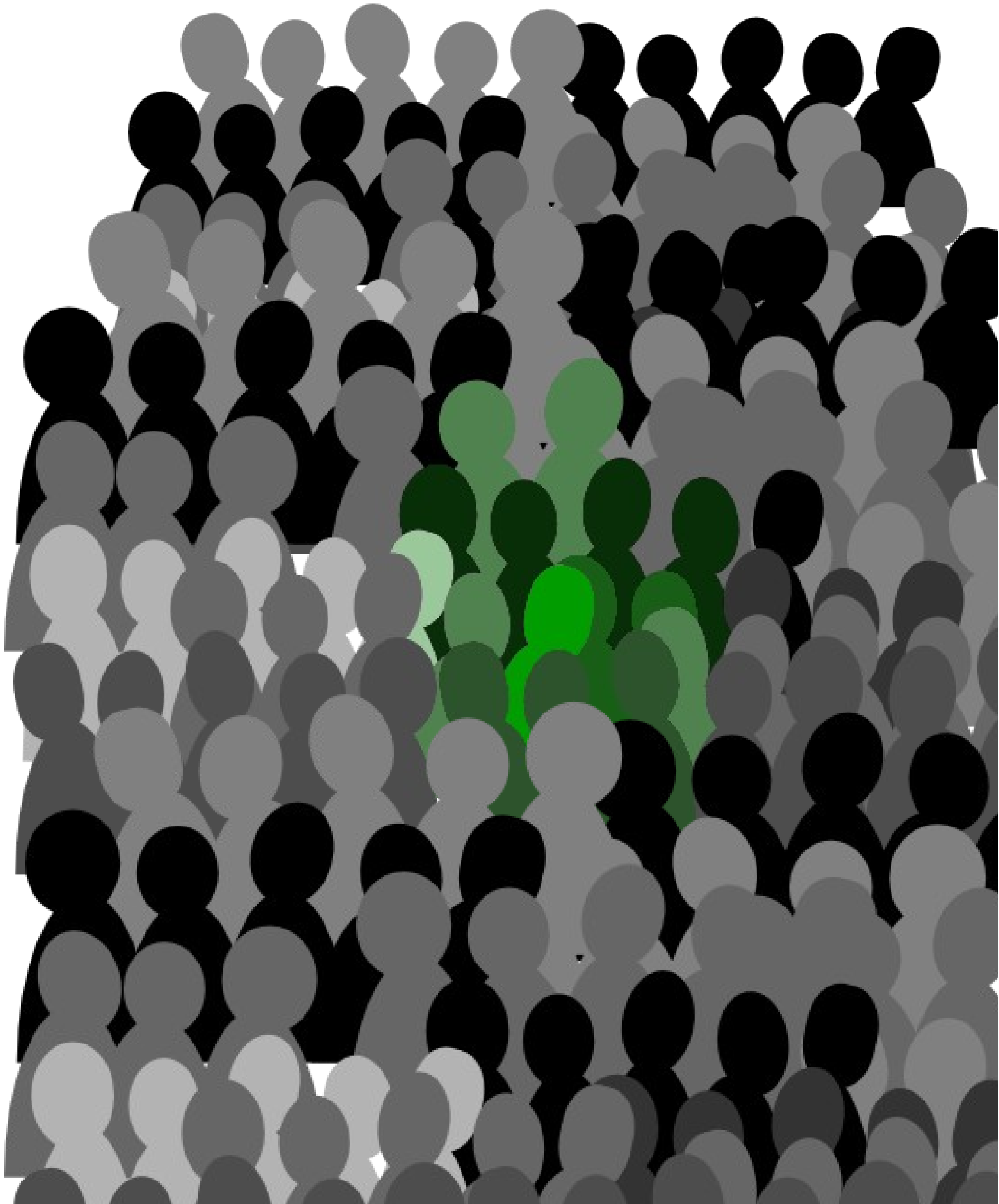
Monitoring Crowds

- Our interest:
  - utilize sensing, processing, and communication capabilities of low-power wearable devices
  - to monitor the behavior of crowds from the inside.
- Envisioned effect:
  - Deeper understanding of crowd behavior
  - More informed planning (e.g., transportation, infrastructure)
  - Ability to manage and control crowds in real time

Image source: http://en.wikipedia.org/wiki/File:Crowd_in_street.jpg

In-Network Aggregation

- Problem: data deluge.
- One of the solutions: decentralized in-network aggregation:
  - Each node senses its surroundings.
  - It communicates its observations via low-power radios to other nearby nodes.
  - The nodes collaboratively aggregate the readings to reduce the traffic volume to an external monitoring site.
- We target basic aggregates:
  - AVG, COUNT, MAX, MIN, SUM

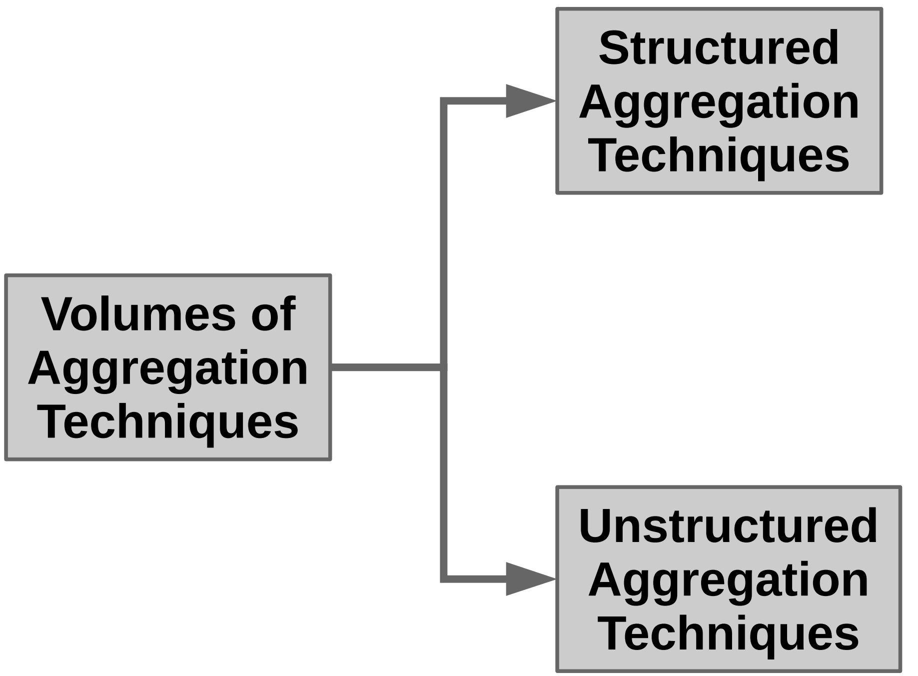


adapted from image: http://www.clker.com/clipart-smaller-crowd-rdc.html

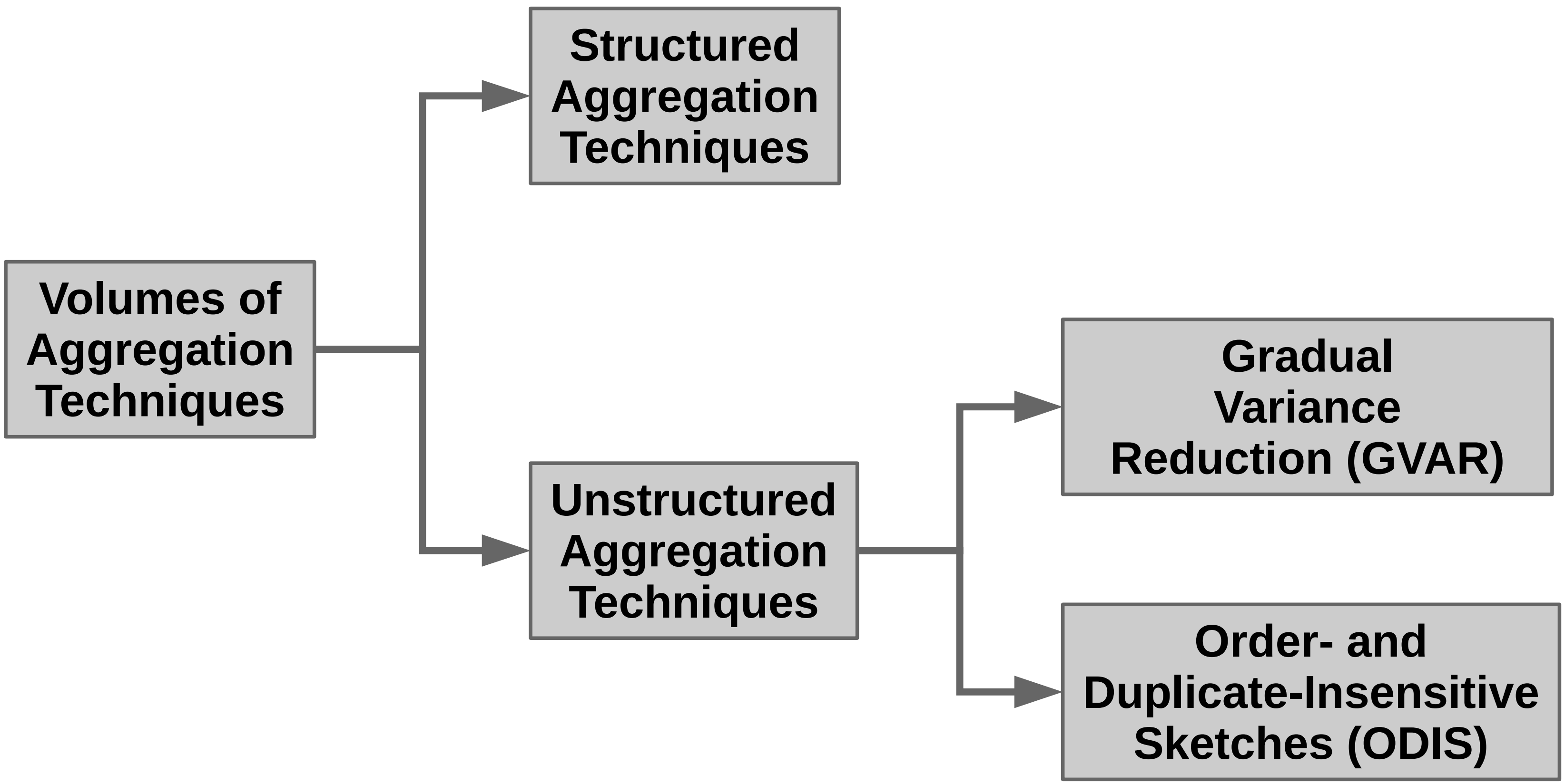
Aggregation in Sensornets

Volumes of Aggregation Techniques

Aggregation in Sensornets

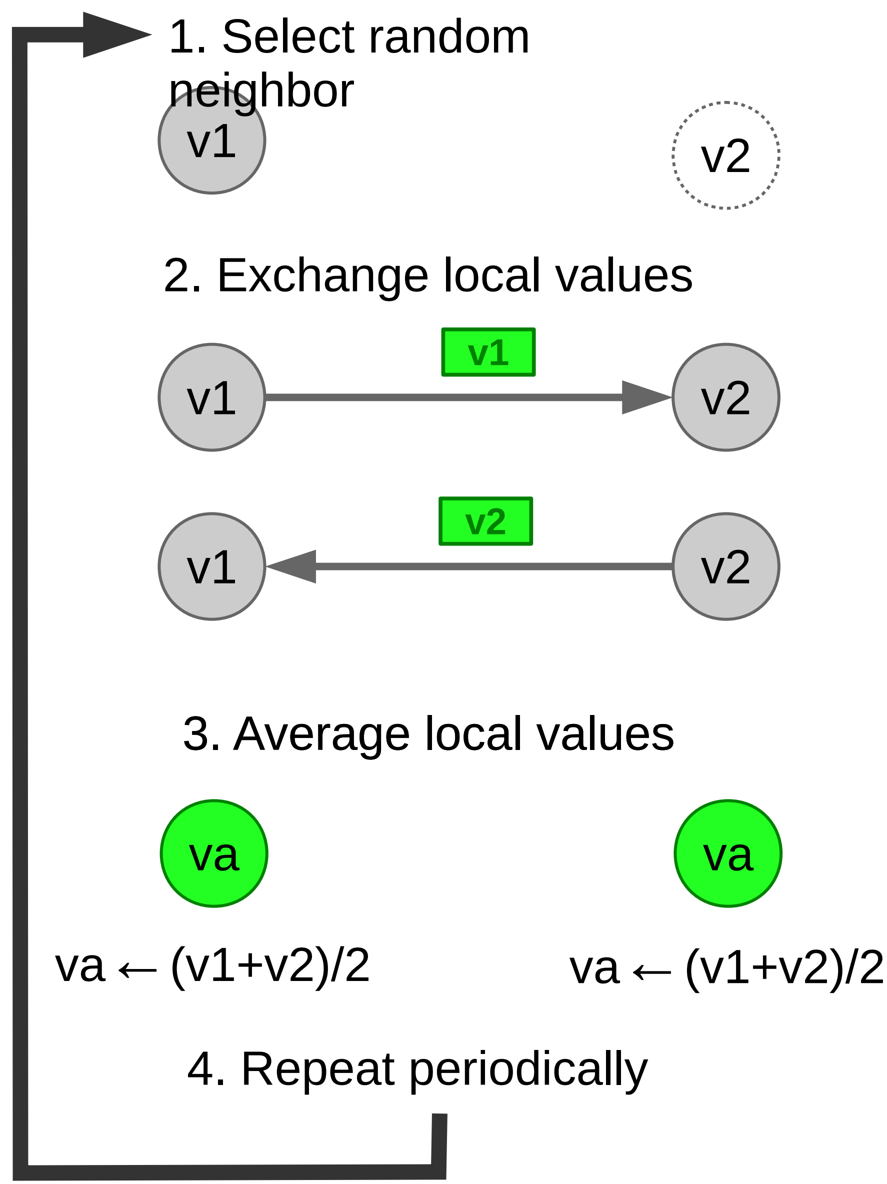


Aggregation in Sensornets



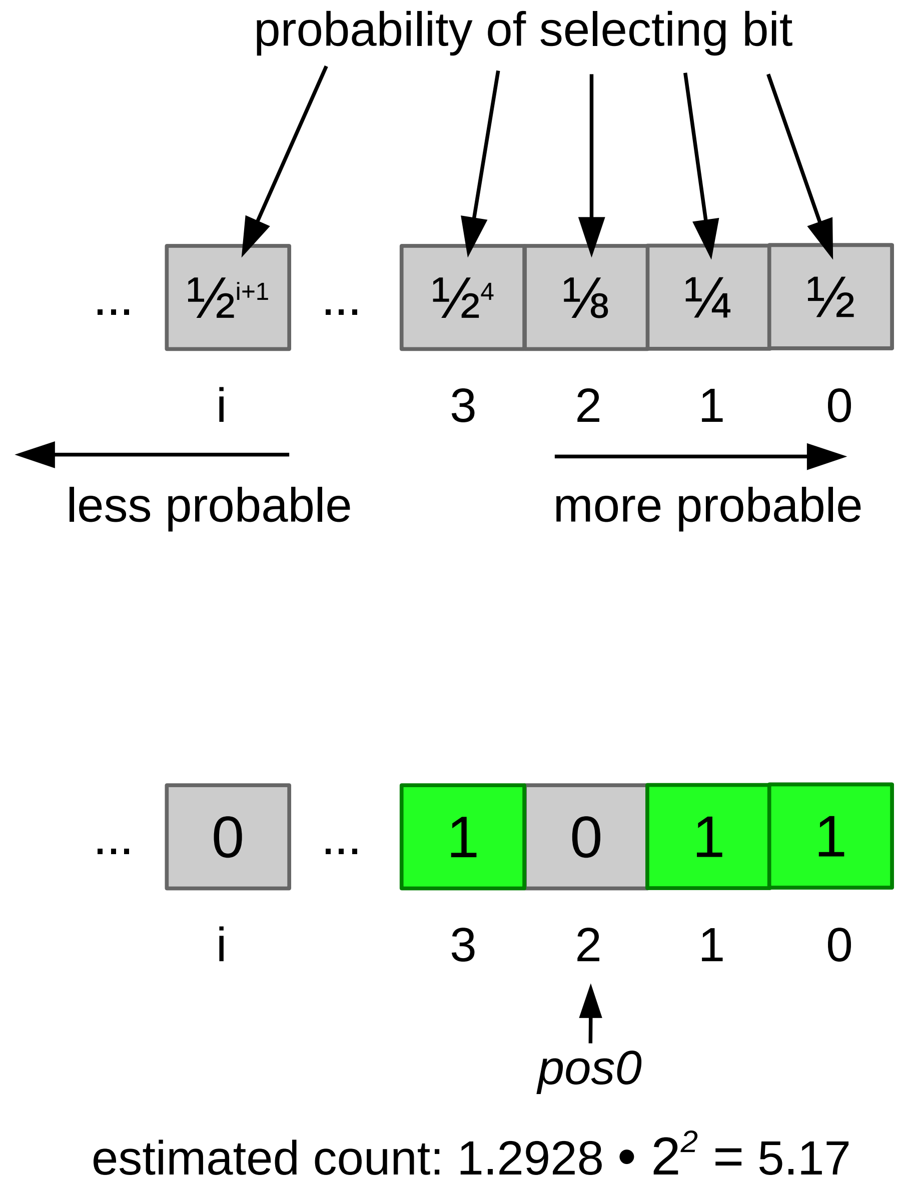
Gradual Variance Reduction (GVAR)

- To compute a global average:
  - Each node periodically selects another neighbor at random.
  - It exchanges its local value with the neighbor's local value.
  - Both nodes set their local values to the average of two values.
  - Over time, the local node values converge to the global average.
- To count the number of nodes:
  - One node sets its initial local value to 1.
  - Others set their values to 0.



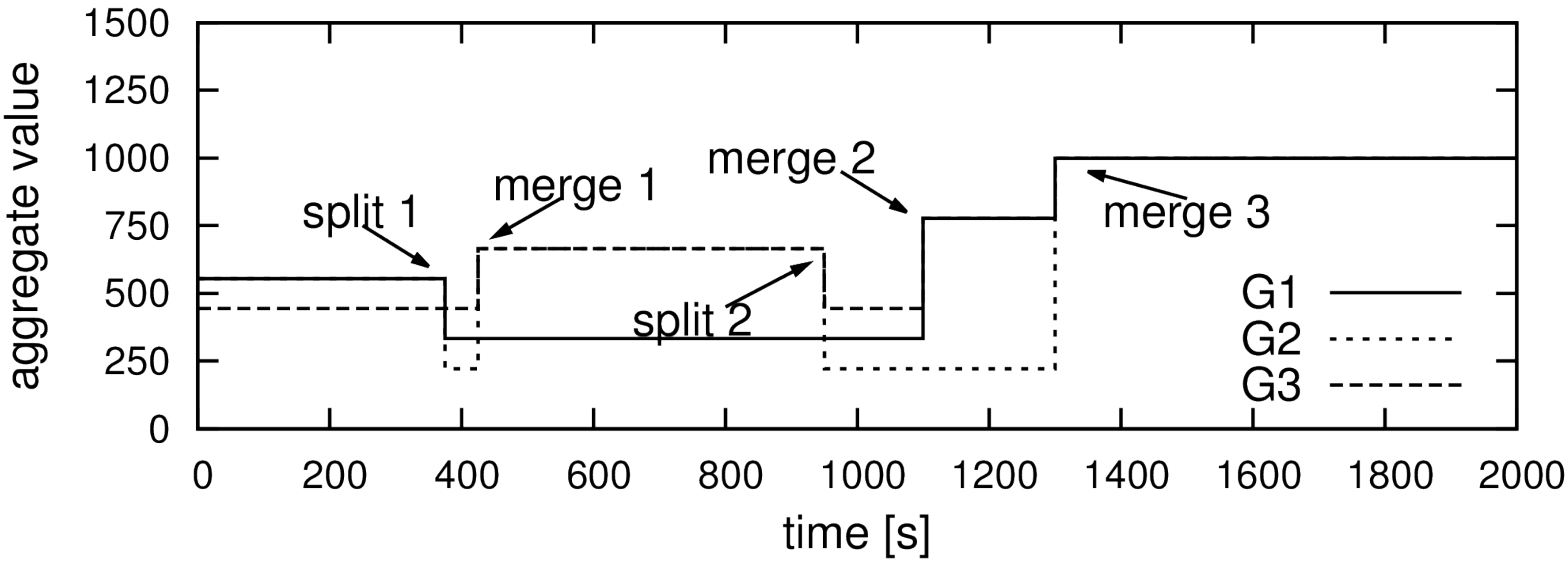
Order- & Duplicate-Insensitive Sketches (ODIS)

- To count the number of nodes:
  - All nodes maintain local 16-bit bitmasks (initially all zeroes).
  - Each node sets one bit in the bitmask with the index drawn from a geometric distribution.
  - Repeatedly exchanges its bitmask with its neighbors OR-ing the received bitmasks with its own.
  - When all bitmasks have converged, the number of nodes is estimated as:
    - $1.2928 \cdot 2^{pos0}$, where *pos0* is the position of the least significant 0.



Base Simulated Scenario

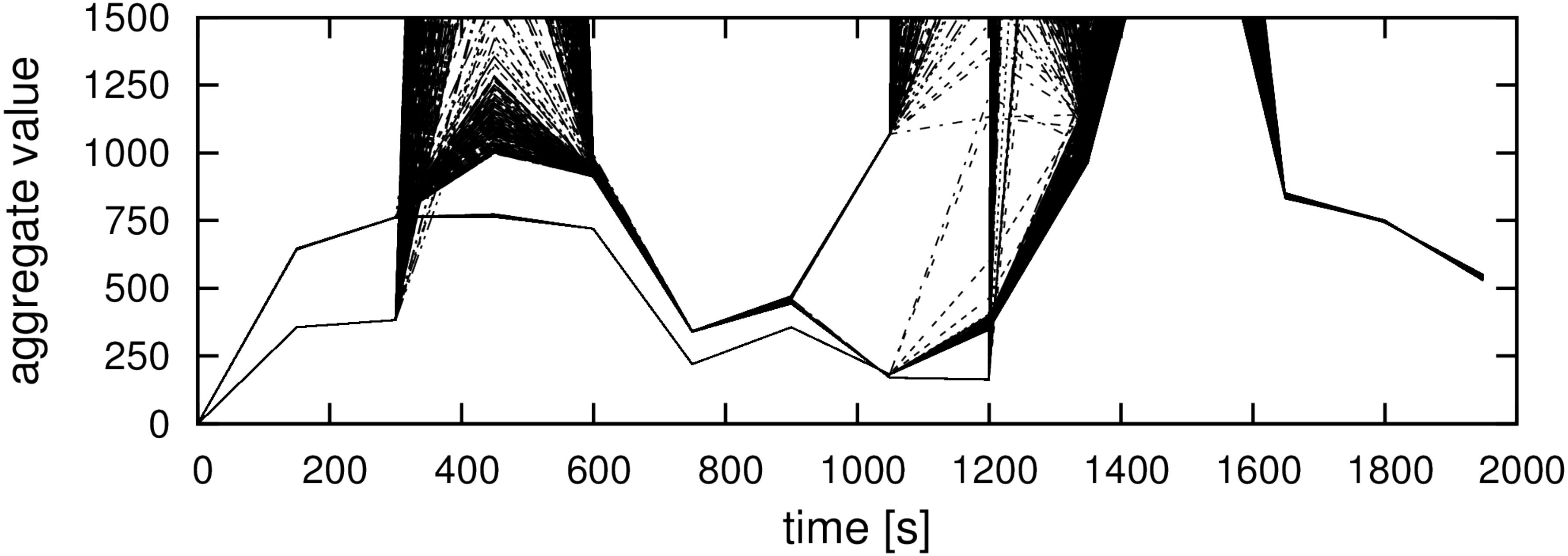
Done in OMNeT++ with MiXiM extensions for wireless sensor networks.



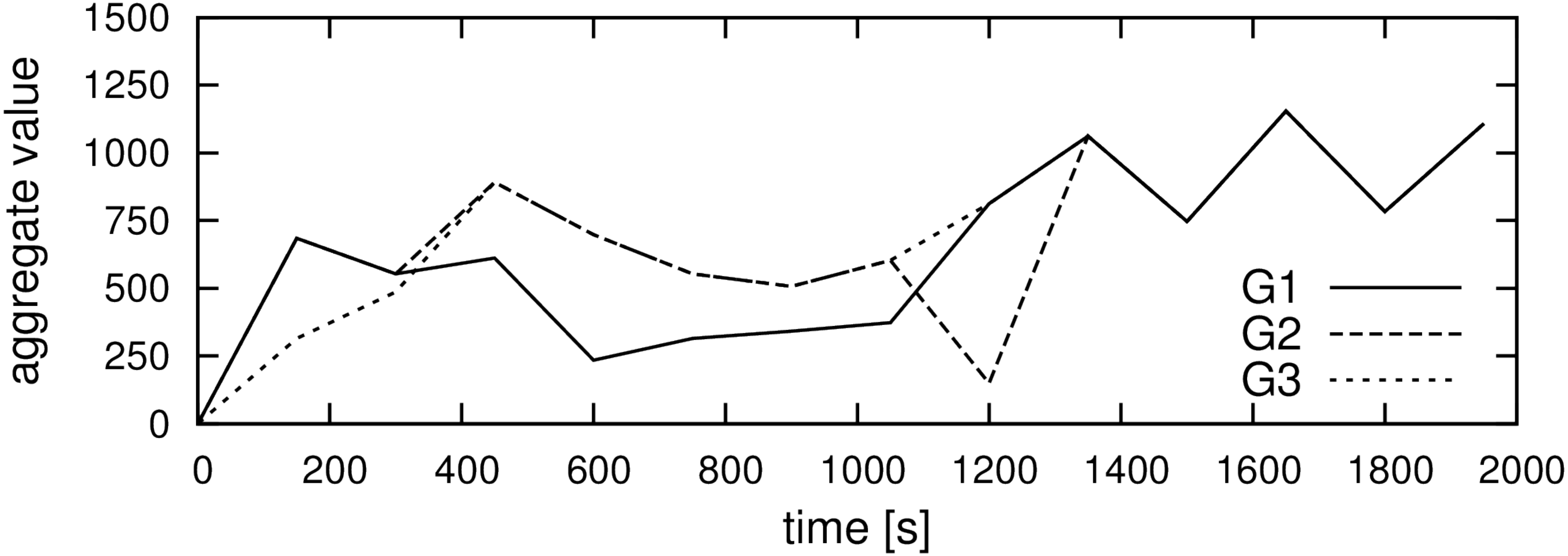
G1 = 333 nodes; G2 = 222 nodes; G3 = 444 nodes

a node's aggregate = COUNT of nodes in the node's connected component.

GVAR Results

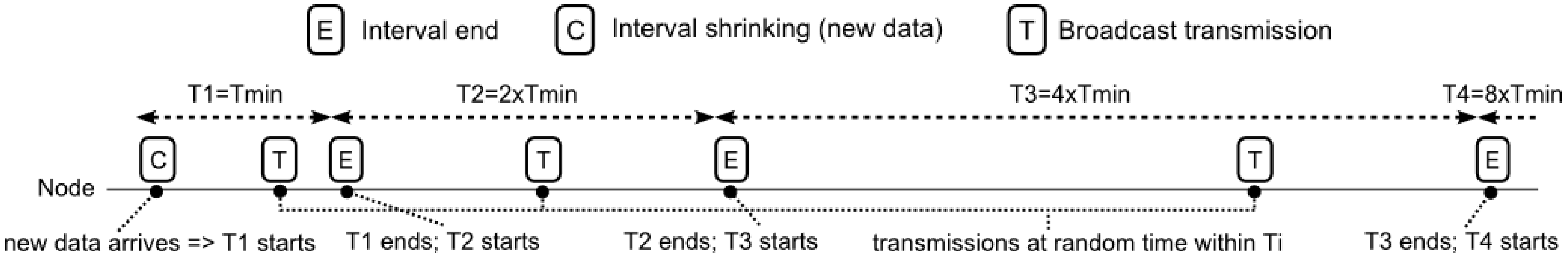


ODIS Results



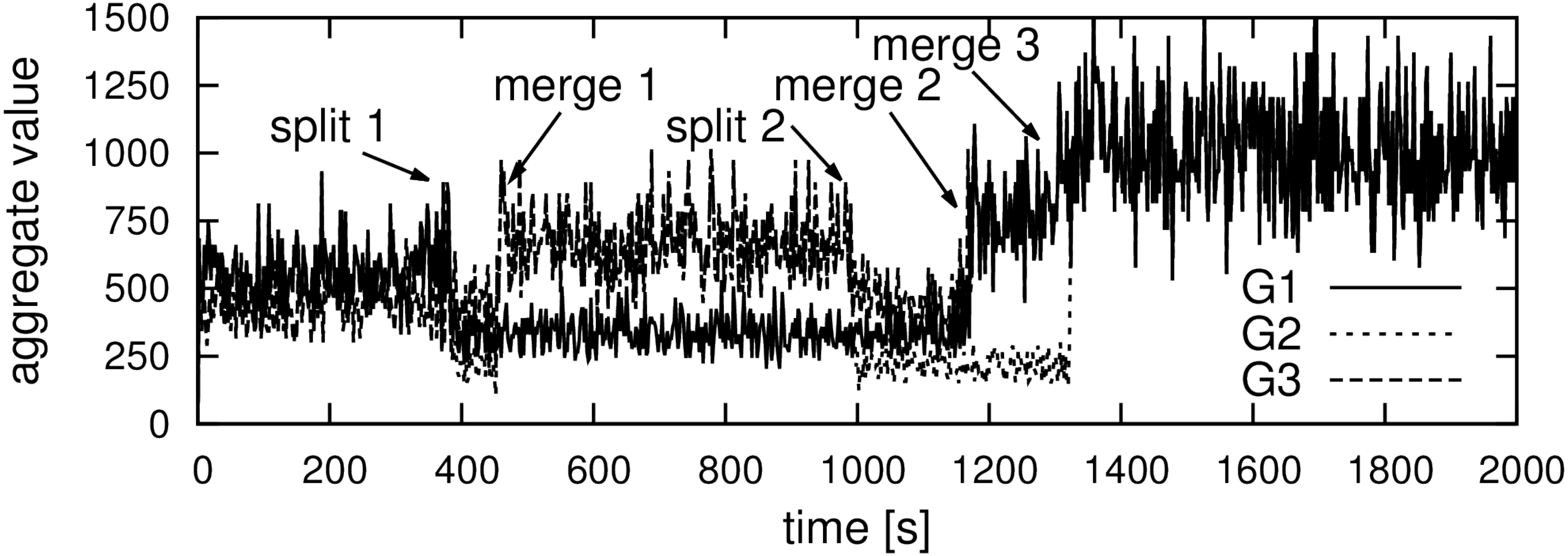
Improving Communication

- To improve communication we adapt the Trickle algorithm for code propagation to ODIS aggregate computation.
  - Normally, broadcast your bitmask randomly within every T_{max} time units.
  - But, when your bitmask changes significantly shrink the interval to T_{min}.
  - Each subsequent interval doubles up to T_{max}.
  - Suppressing broadcasts when several similar bitmasks are received.



Effect: When the local bitmast quickly compute aggregates while minimizing traffic when the system is quiescent.

ODIS with New Communication

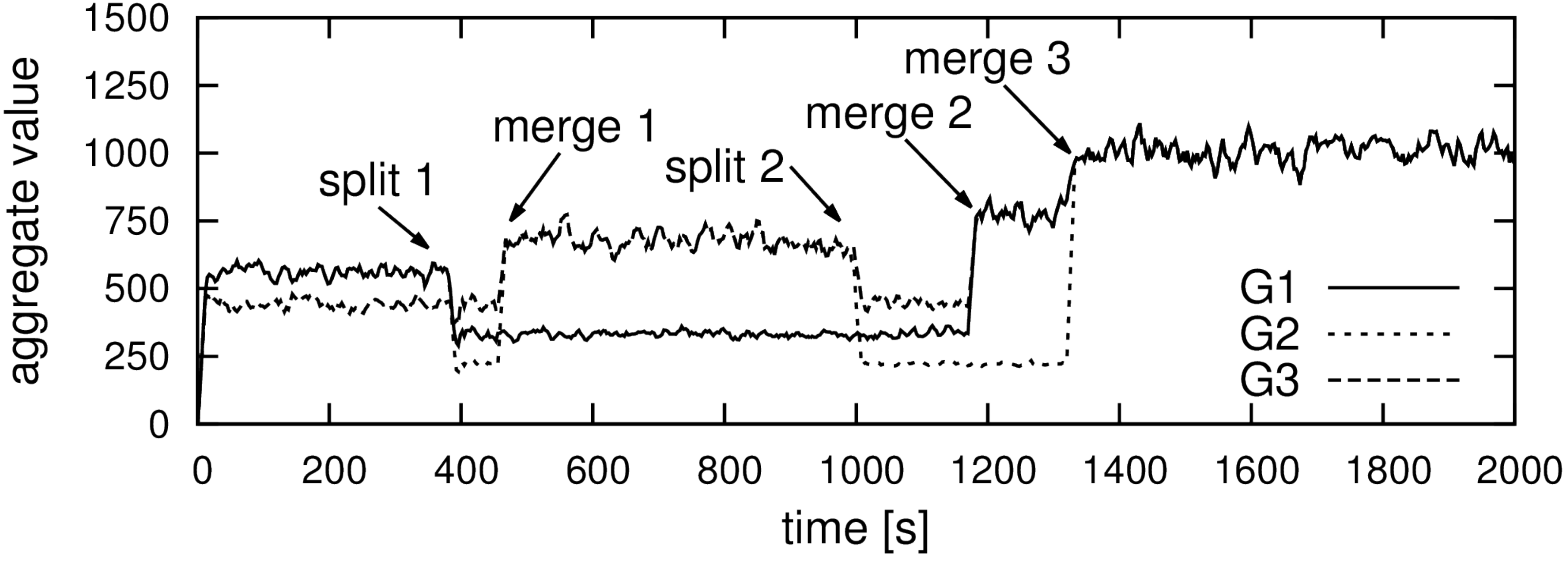


Improving Accuracy

- Use many instances of sketches: **smoothing**.
- Use more efficient sketches:
  - parameterless sketches:
    - can be used out-of-the-box, but
    - are not the most efficient ones (wrt. error / #bits).
  - parametrized sketches:
    - are very efficient, but
    - their accuracy depends on the final result.
  - Solution: **pipelining** a parameterless sketch with a parametrized one.

Effect: The accuracy improves for the same number of bits.

ODIS with Improved Accuracy

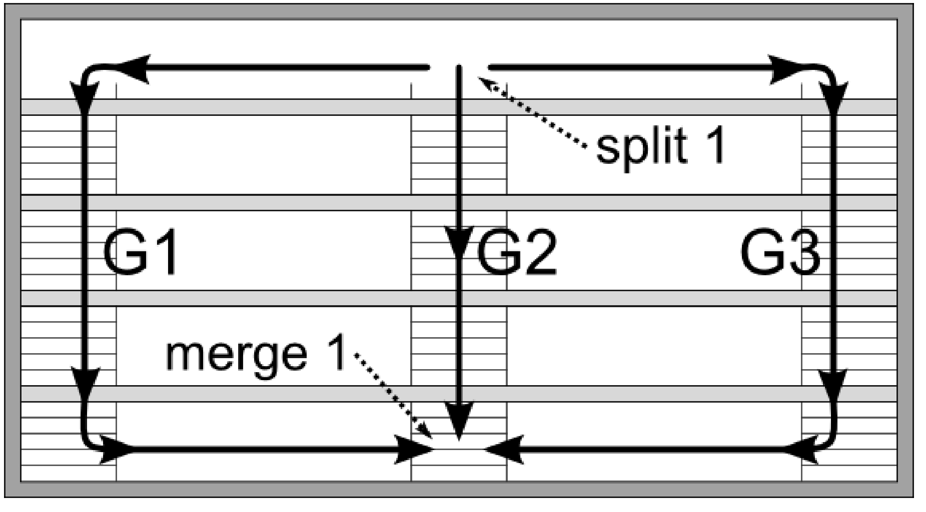


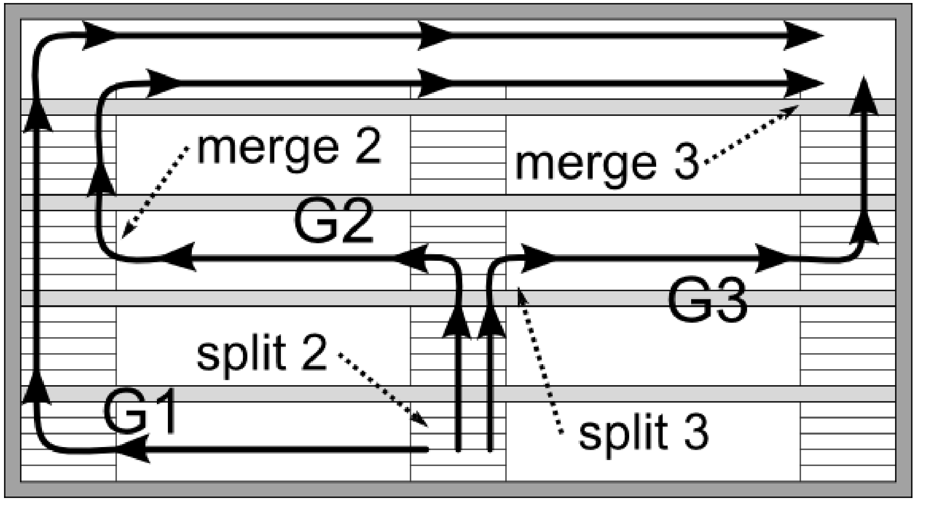
Back to Real-World Experiments

- We implemented the algorithms as an aggregation service for TinyOS.
- We conducted several real-world deployments of the service.
  - Up to 177 nodes.
- Mostly on eZ430 Chronos smart watches.

Image source: http://electronicdesign.com/content/14978/59382_fig_01.jpg

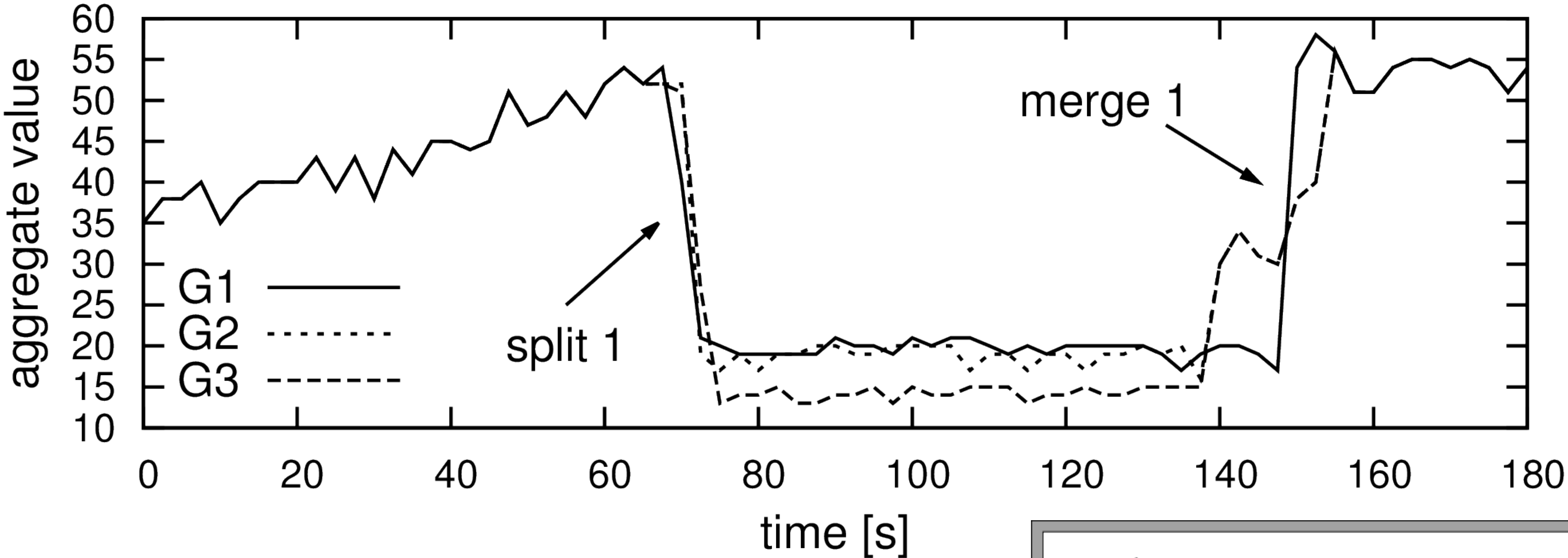
Sample Scenarios



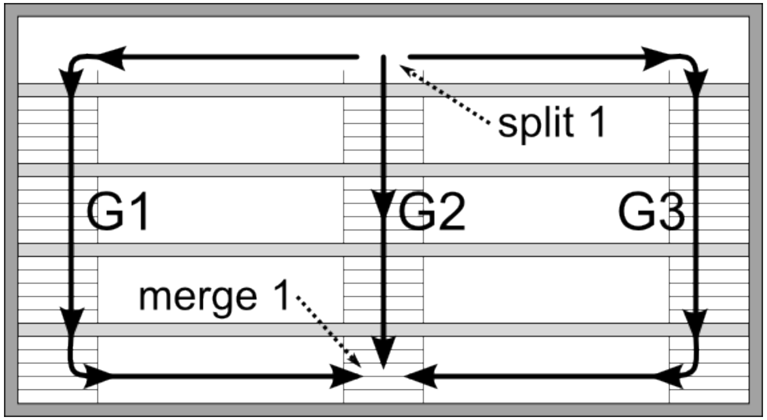


G1 = 20 nodes; G2 = 19 nodes; G3 = 15 nodes

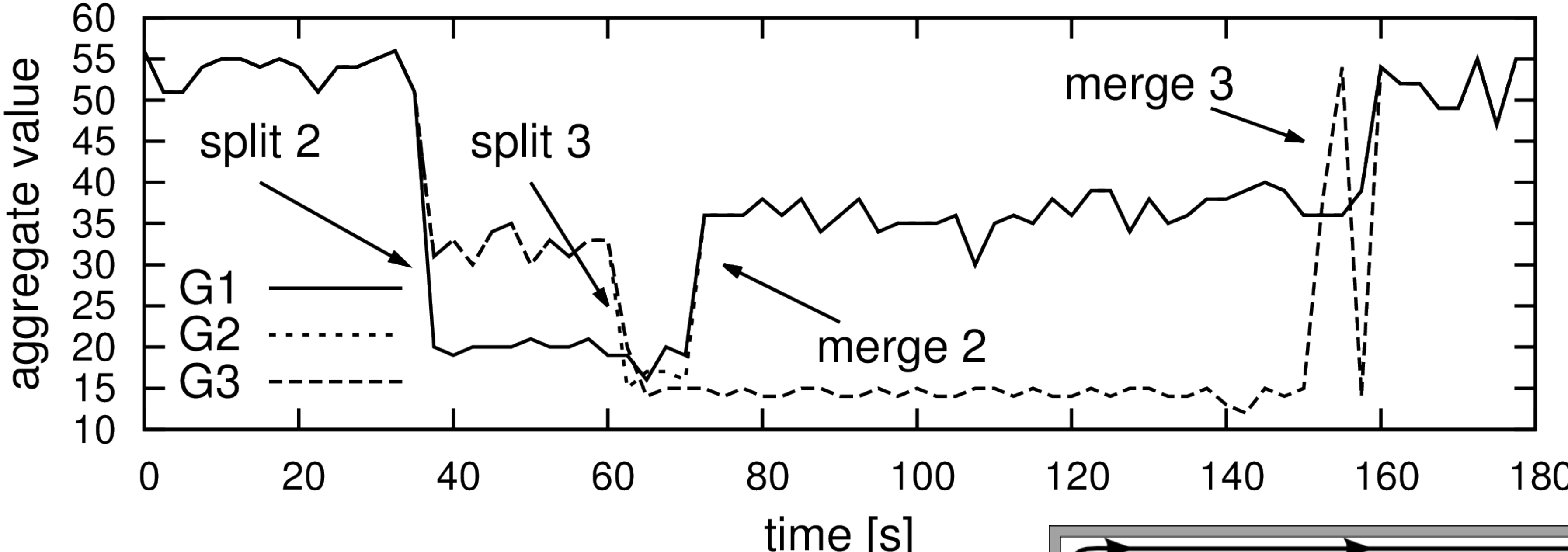
Results (Scenario 1)

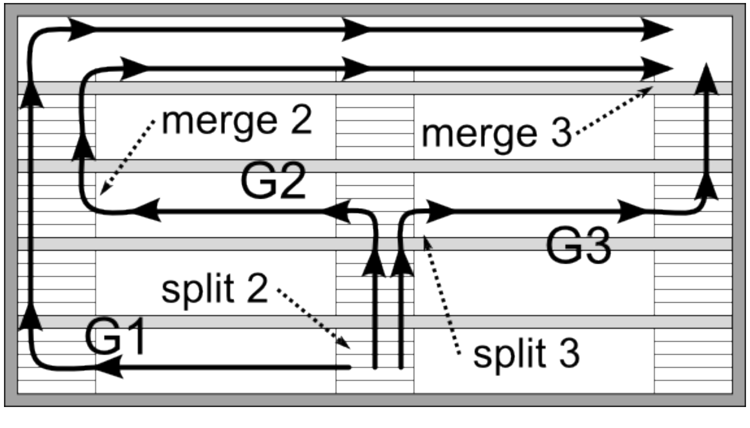


G1 = 20 nodes;   G2 = 19 nodes;   G3 = 15 nodes



Results (Scenario 2)





G1 = 20 nodes;   G2 = 19 nodes;   G3 = 15 nodes

Conclusions

- To be applied in real-world crowd-monitoring scenarios, decentralized in-network aggregation algorithms for sensornets require considerable adaptation.
- Applications have to be prepared that the aggregates they see may exhibit errors.
- We may need to revisit some of their assumptions.
- (Conducting real-world crowd-monitoring deployments is challenging.)

Thank You

Questions?

Supported by the (Polish) National Science Centre (NCN) within
the SONATA programme under grant no. DEC-2012/05/D/ST6/03582.

NATIONAL SCIENCE CENTRE