LMU LUDWIG-MAXIMILIANS-UNIVERSITY MUNICH
DEPARTMENT INSTITUTE FOR INFORMATICS
DATABASE SYSTEMS GROUP

Continuous Quantile Query Processing in Wireless Sensor Networks

Johannes Niedermayer, Mario Nascimento,
Matthias Renz, Peer Kröger, Hans-Peter Kriegel

Ludwig-Maximilians-Universität München
Munich, Germany

University of Alberta, Edmonton, Canada

Outline

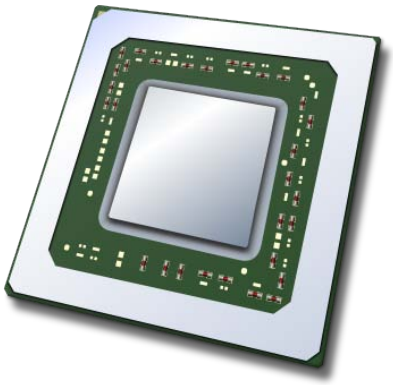
- Problem Definition
  - What is a Wireless Sensor Network?
  - Quantile Queries in Wireless Sensor Networks
- Related Work
- Algorithms
  - Preliminaries
  - The Continuous *b*-ary Search
  - Interval-based Quantiles
- Experimental Evaluation
- Conclusions

Problem Definition

- What is a Wireless Sensor Network (WSN)?
  - A WSN is a *large-scale*, *wireless*, *ad-hoc*, *multi-hop*, *unpartitioned* network of *homogeneous*, *small*, *static* nodes [RM04]
  - Nodes consist of
    - Sensors, (temperature, humdity, radiation, …)
    - Microprocessor
    - Memory
    - Radio transceiver

Problem Definition

- (Historical) Deficiencies of WSNs



Limited processing power

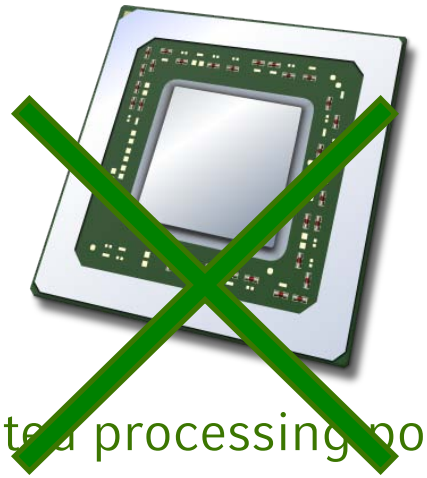
Limited storage capacity

Limited energy supply

Problem Definition

- (Historical) Deficiencies of WSNs

Limited storage capacity

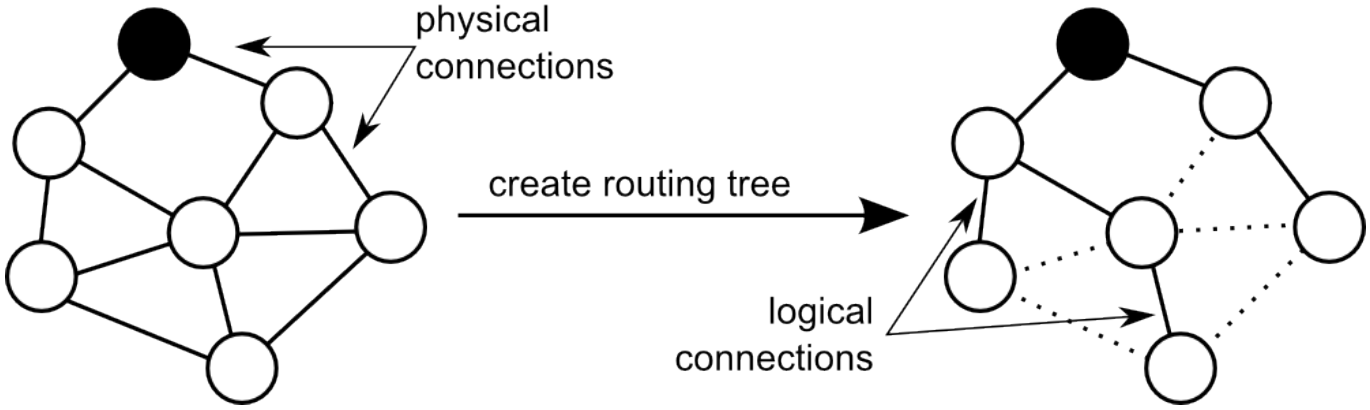


Limited processing power

Limited energy supply

Problem Definition

- Quantile Queries in WSNs
  - Given a set of nodes *N* producing one measurement each and an input parameter $\varphi \in]0,1[$, a φ-quantile query returns the measurement with rank $k = \lceil \varphi |N| \rceil$.
  - Special case: median ($\varphi=0.5$)
  - Snapshot Query: quantile is computed once
  - **Continuous Query: quantile is computed periodically**
  - We concentrate on quantile queries in hierarchical WSNs:

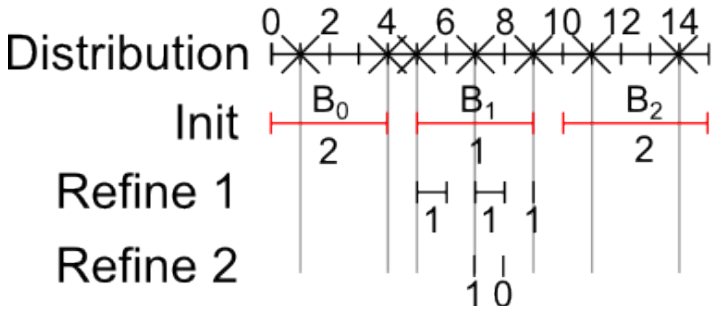


Related Work

- Approximate Solutions
  - Returns the φ-quantile with rank difference at most $\varepsilon|N|$.
  - Examples: Greenwald-Khanna, Q-Digest
- Probabilistic Solutions
  - Return the φ-quantile within given error bounds with probability $1-\delta$.
  - Examples: sampling, layered architectures
- Exact Solutions
  - Return the exact φ-quantile.
  - Can be made probabilistic by using a layered architecture
  - Examples: sending all values, binary search, b-ary search, sampling-based b-ary search

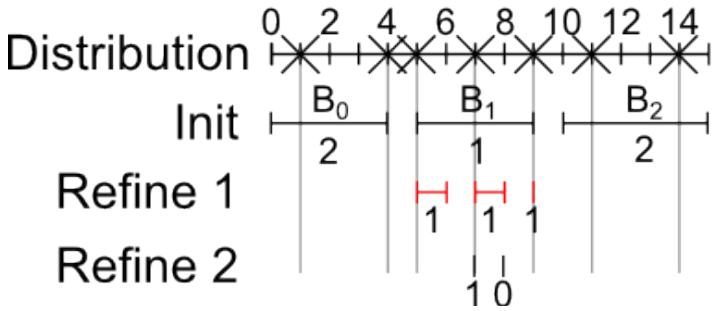
Preliminaries
Bucket-based Queries

- The *b*-ary Search [CCR06, LCLL08, NNRKAK13]
  - Split the intervall of possible values in *b* subintervals, count measurements in these subintervals
  - Refine the interval that contains the *k*-th value by splitting it in *b* subintervals
  - Continue until the length of a subinterval becomes < 1



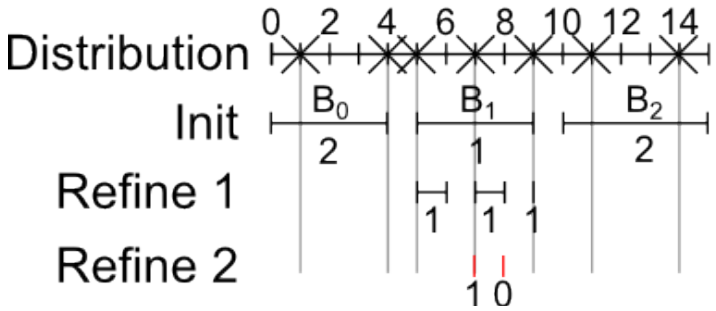
Preliminaries
Bucket-based Queries

- The *b*-ary Search [CCR06, LCLL08, NNRKAK13]
  - Split the intervall of possible values in *b* subintervals, count measurements in these subintervals
  - Refine the interval that contains the *k*-th value by splitting it in *b* subintervals
  - Continue until the length of a subinterval becomes < 1



Preliminaries
Bucket-based Queries

- The *b*-ary Search [CCR06, LCLL08, NNRKAK13]
  - Split the intervall of possible values in *b* subintervals, count measurements in these subintervals
  - Refine the interval that contains the *k*-th value by splitting it in *b* subintervals
  - Continue until the length of a subinterval becomes < 1



Preliminaries
Bucket-based Queries

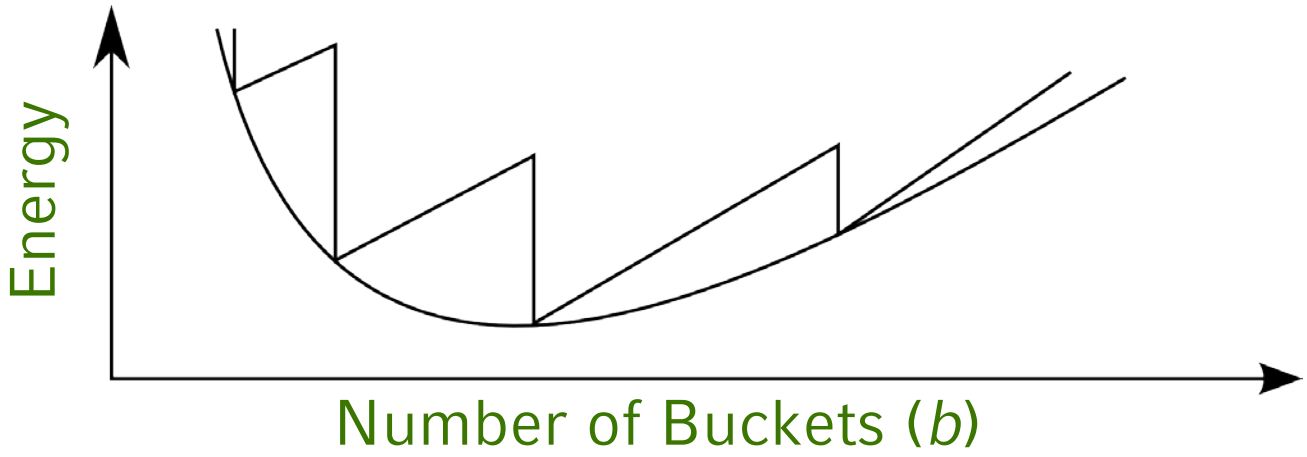
- Perforance Considerations
  - The number of iterations of the *b*-ary search is generally logarithmic in the number of possible values *r*:

    $$O(log(r))$$

  - By varying the number of buckets, we can optimize the energy consumption of the query by optimizing the number of *iterations*
  - Reasoning:
    - Less buckets → higher number of iterations
    - More buckets → higher cost for a single iteration
  - Goal: Find the optimal number of buckets

Preliminaries
Bucket-based Queries

- Cost Model
  - In [NNRKAK13] we developed a cost model for approaches based on a *b*-ary search
  - The cost model describes the cost of a query as the maximum cost for a single node, based on:
    - The number of refinement iterations
    - Message sizes (requests, responses) including protocol information



Preliminaries
POS

- POS Initialization [CCR06]
  - Central computation of $v_0{}^k$ at the root node (TAG)
  - Additionally the root node initializes three variables

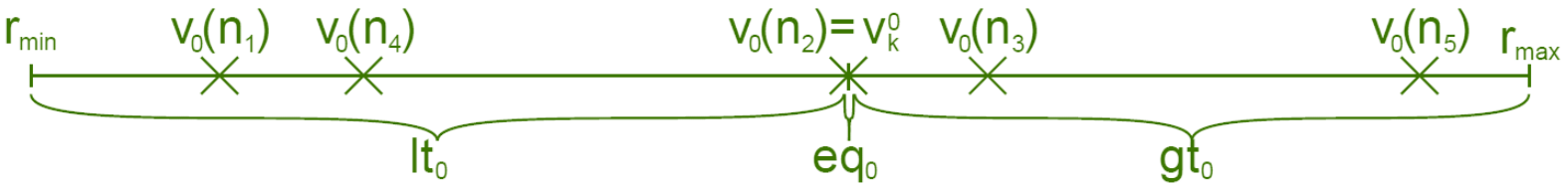
$$e = |\{v(n_i)|n_i \in N \wedge v(n_i) = v_k^0\}| = 1$$

$$l = |\{v(n_i)|n_i \in N \wedge v(n_i) < v_k^0\}| = 2$$

$$g = |N| - l - e = 2$$

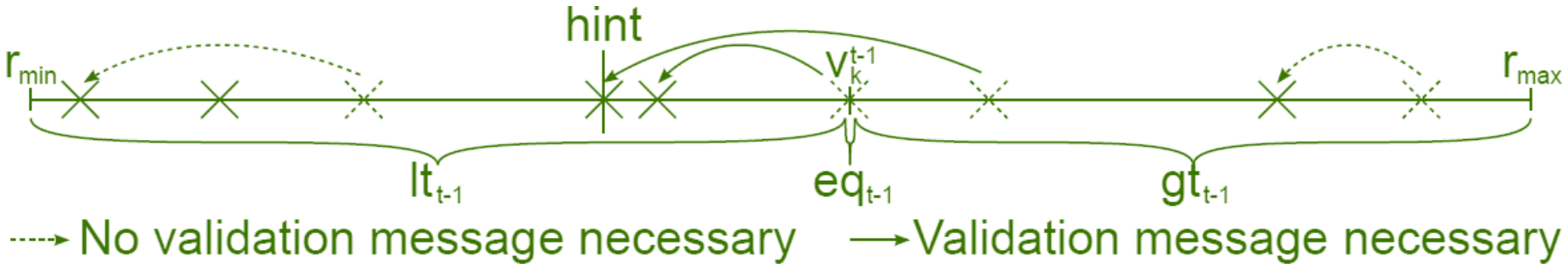
  - $v_0{}^k$ is broadcasted into the tree, i.e. each node can decide in which of the following intervals its value falls:

$$lt =]-\infty, v_k^{t-1}[; eq = [v_k^{t-1}, v_k^{t-1}]; gt =]v_k^{t-1}, \infty[$$



Preliminaries
POS

- POS Update [CCR06]
  - A validation message $(out_{gt}, in_{gt}, out_{lt}, in_{lt}, hint)$ is sent if a node's value jumped from one interval to another, e.g. $lt_{t-1} \rightarrow gt_t$ or $eq_{t-1} \rightarrow lt_t$
  - Based on this information, the root node updates g, l, and e
  - Refinement:
    - Binary search in $]v_k^{t-1}, hint]$ or $[hint, v_k^{t-1}[$
    - Broadcast v_k^t if necessary



Continuous Queries:
Approach 1: Continuous b-ary Search

- Approach 1: Extend POS to a *b*-ary search
  - Perform a *b*-ary search instead of a binary search
  - Reduces the number of refinement iterations compared to a binary search.
  - Especially with large headers, more buckets lead to a lower energy consumption
  - Uses our cost model and optimizations for quantile queries from [NNRKAK13]

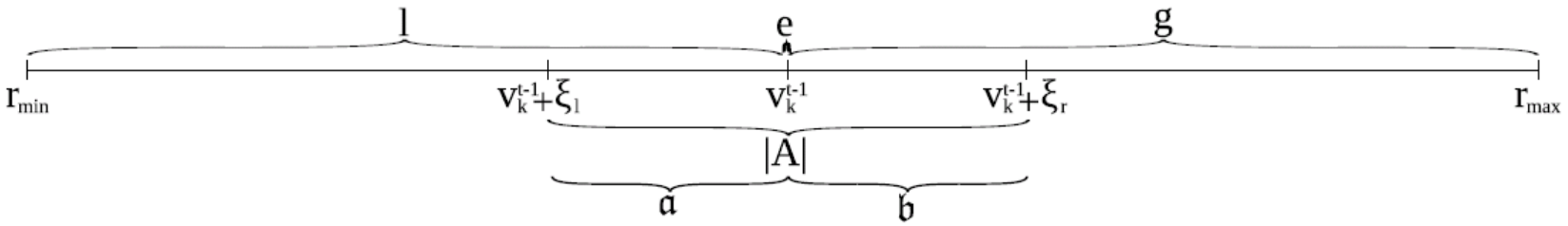
Continuous Queries:
Approach 2: Interval-based Quantiles

- Approach 2: Interval-based Quantiles (IQ)
  - Takes the idea of reducing update messages and refinement messages to an extreme
  - Due to the overhead for transmitting protocol information (e.g. headers), it is usually better to transmit more data in a single message
    → Strictly avoid refinements
    → Strictly avoid multiple refinement iterations

Continuous Queries:
Approach 2: Interval-based Quantiles

- IQ Initialization:
  - Equivalent to POS
  - By using TAG or bucket-based solutions
  - We just need to gather the information for initializing *l*,*g*, and *e*

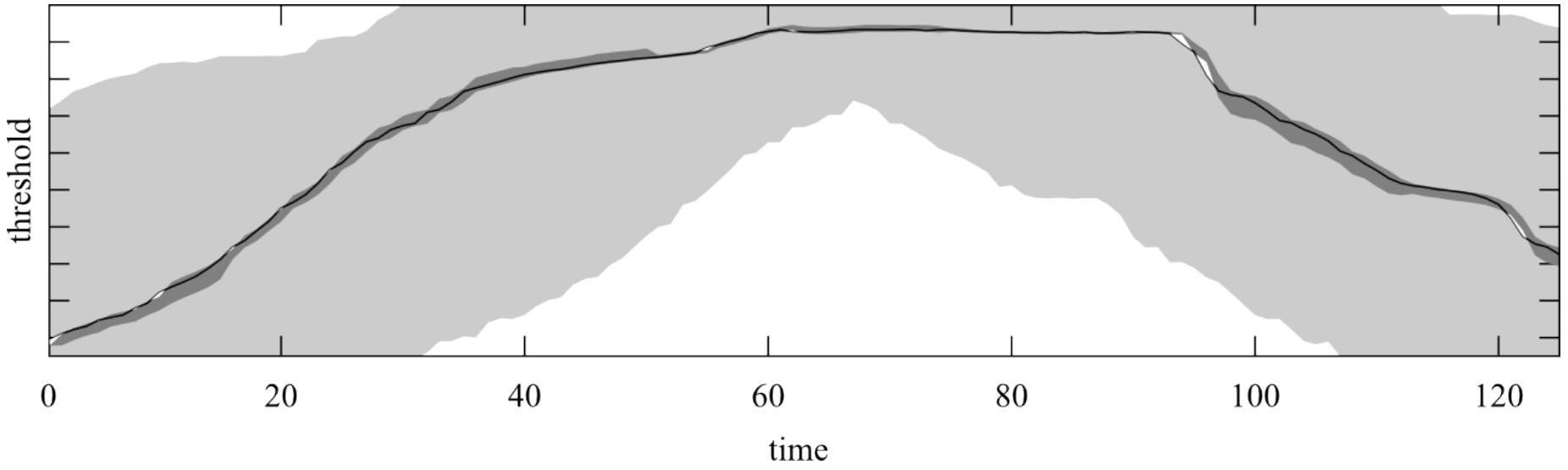
- IQ Update:
  - Send values in $[v_k^{t-1}+\xi_l, v_k^{t-1}+\xi_r]$ directly
  - ξ_l and ξ_r are calculated by taking the most *m* recent quantiles into account



Continuous Queries:
Approach 2: Interval-based Quantiles

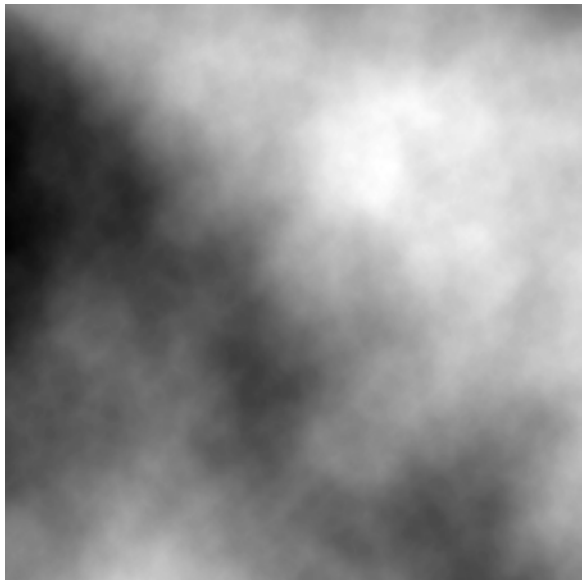
- Refinement: no binary search, instead transmit …
  - … the p largest values in $]-\infty, v_k^{t-1}]$ if $v_k^t < v_k^{t-1}$
  - … the p smallest values in $]v_k^{t-1}, \infty[$ if $v_k^t > v_k^{t-1}$

  p can be calculated based on the information known at the root node



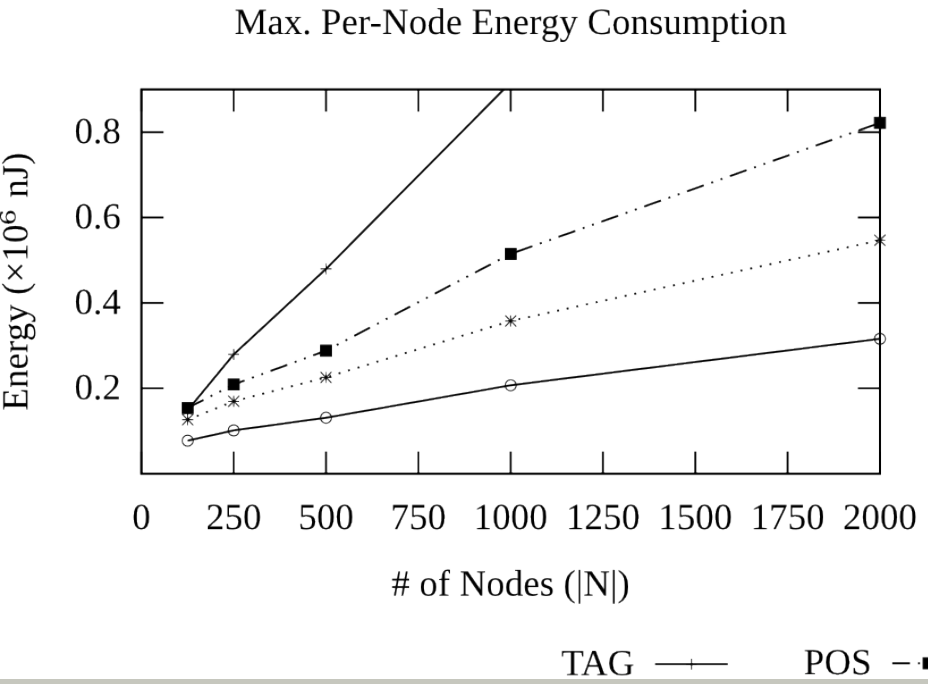
Simulations
Setup

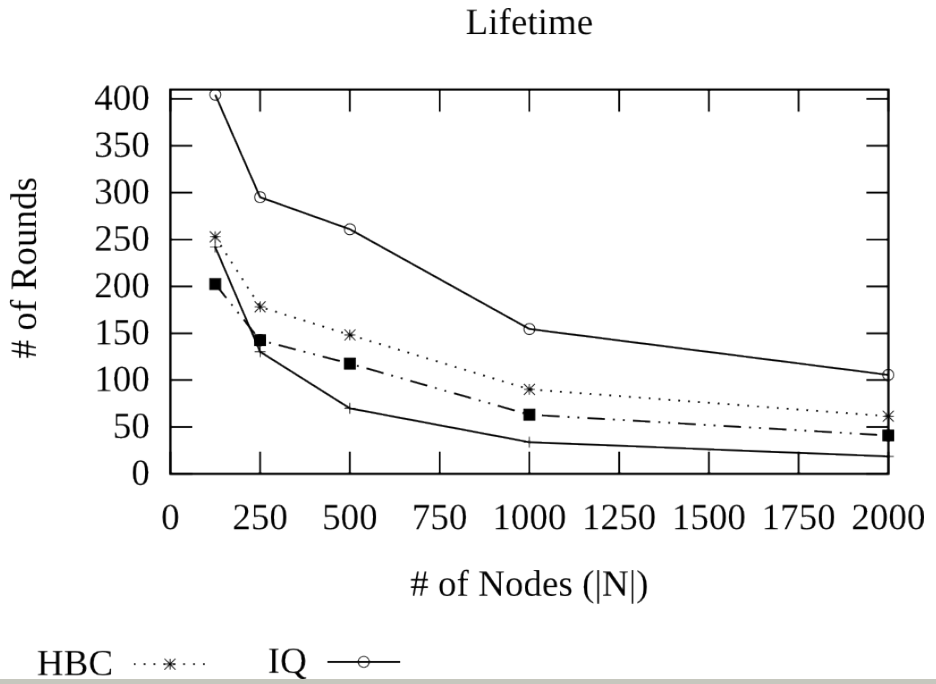
- Setup
  - HBC and IQ were compared to
    - TAG [MFHH02]
    - POS [CCR06]
    - …
  - Topology: Shortest Path Tree
  - Datasets:
    - Synthetic data (sinusoid function with noise, locally correlated values)
    - Real (air pressure)
  - Variables: |N|, period, …



Simulations
Synthetic Data (|N|)

- Synthetic data: Varying the number of nodes |N|
  - The larger the network, the more energy is consumed by hot-spot nodes
  - As the network contains more nodes, more values have to be transmitted and received

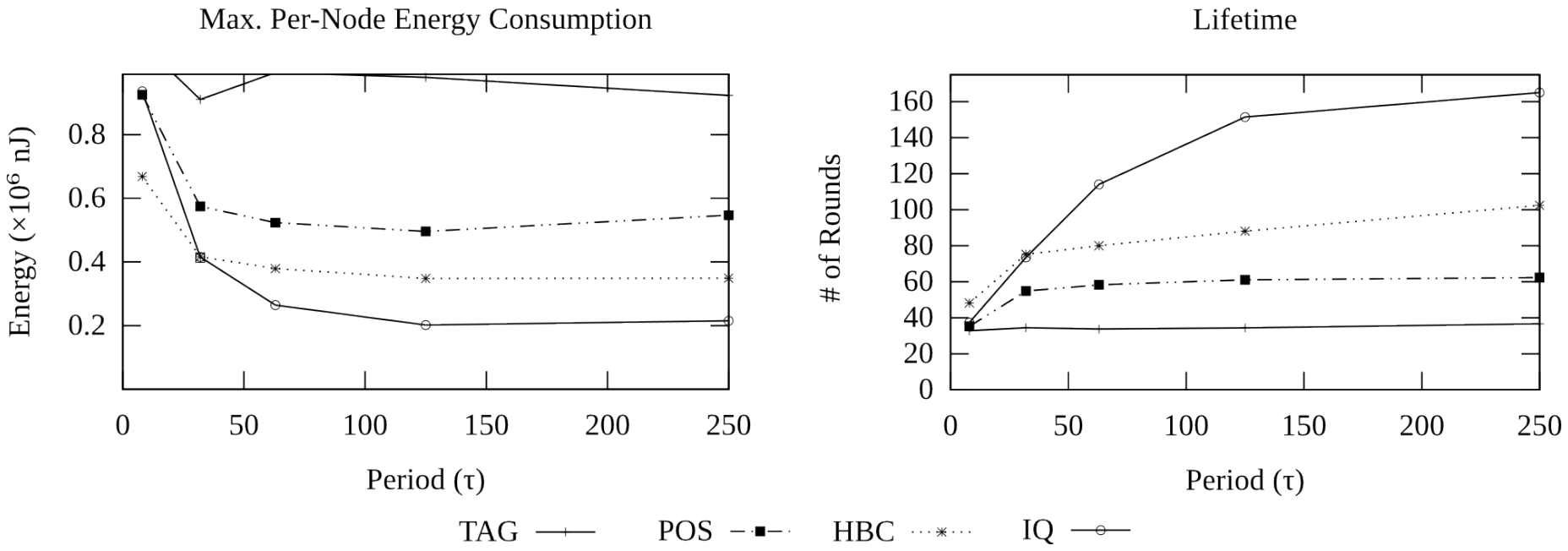




TAG —+— POS —■— HBC ···✳··· IQ —○—

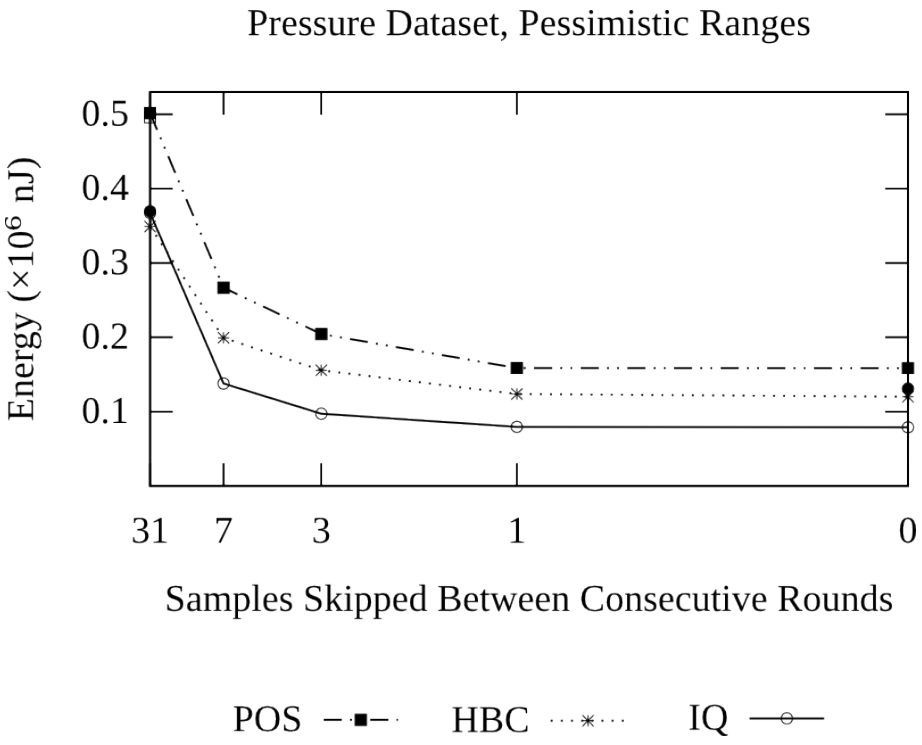
Simulations
Synthetic Data (Period)

- Synthetic data: Varying the period of the underyling periodic function
  - Static data leads to low energy consumption for most approaches.
  - HBC consumes less energy than its binary counterpart POS, but more energy than IQ for relatively static data



Simulations
Real Data

- Real data (air pressure):
  - 1022 nodes
  - We skipped measurements of the data traces to simulate different sampling rates
  - The lower the sampling rate (the more samples skipped between rounds), the more energy is consumed



Conclusions

- We compared two approaches for continuous quantile queries in wireless sensor networks: bucket-based approaches and interval-based quantiles.

- If there exists temporal correlation between values, the heuristic solution achieves gut results, often better than bucket-based solutions.

Thank you!

Thank you!

References

- [CCR06]: L. P. Cox, M. Castro, and A. Rowstron. Pos: A practical order statistics service for wireless sensor networks. In Proc. of ICDCS, pages 52-64, 2006.
- [LCLL08]: K. Liu, L. Chen, M. Li, and Y. Liu. Continuous answering holistic queries over sensor networks. In Proc. of IPDPS, pages 1-11, 2008.
- [NNRKAK13]: J. Niedermayer, M. A. Nascimento, M. Renz, P. Kröger, K. Ammar, and H.-P. Kriegel. Cost-based quantile query processing in wireless sensor networks. In Proc. MDM, 2013.
- [RM04]: K. Römer and F. Mattern. The design space of wireless sensor networks. IEEE Wireless Communications, 11(6):54-61, 2004.
- [MFHH02]: S. Madden, M. J. Franklin, J. Hellerstein, and W. Hong. Tag: a tiny aggregation service for ad-hoc sensor networks. SIGOPS Oper. Syst. Rev., 36(SI):131-146, 2002.

Problem Definition

- Why Quantiles?

The median and quantiles in general are less prone to outliers than for example the mean