MINT Views: Materialized In-Network Top-k Views in Sensor Networks

Demetrios Zeinalipour-Yazti (Uni. of Cyprus)

Panayiotis Andreou (Uni. of Cyprus)

Panos Chrysanthis (Uni. of Pittsburgh, USA)

George Samaras (Uni. of Cyprus)

http://www.cs.ucy.ac.cy/~dzeina/

Wireless Sensor Networks

- Resource constrained devices utilized for **monitoring** and **understanding** the physical world at a high fidelity.
- Applications have already emerged in:
  - Environmental and habitant monitoring
  - Seismic and Structural monitoring
  - Understanding Animal Migrations & Interactions between species.

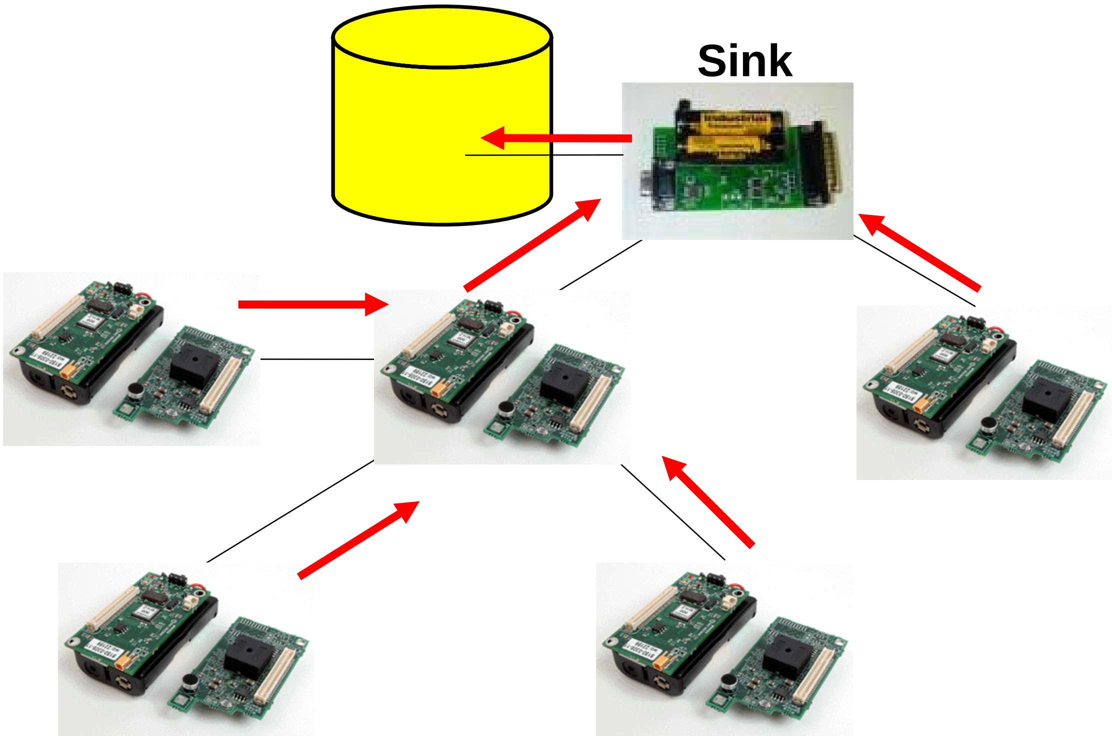
Great Duck Island – Maine (Temperature, Humidity etc).

Golden Gate – SF, Vibration and Displacement of the bridge structure

Zebranet (Kenya) GPS trajectory

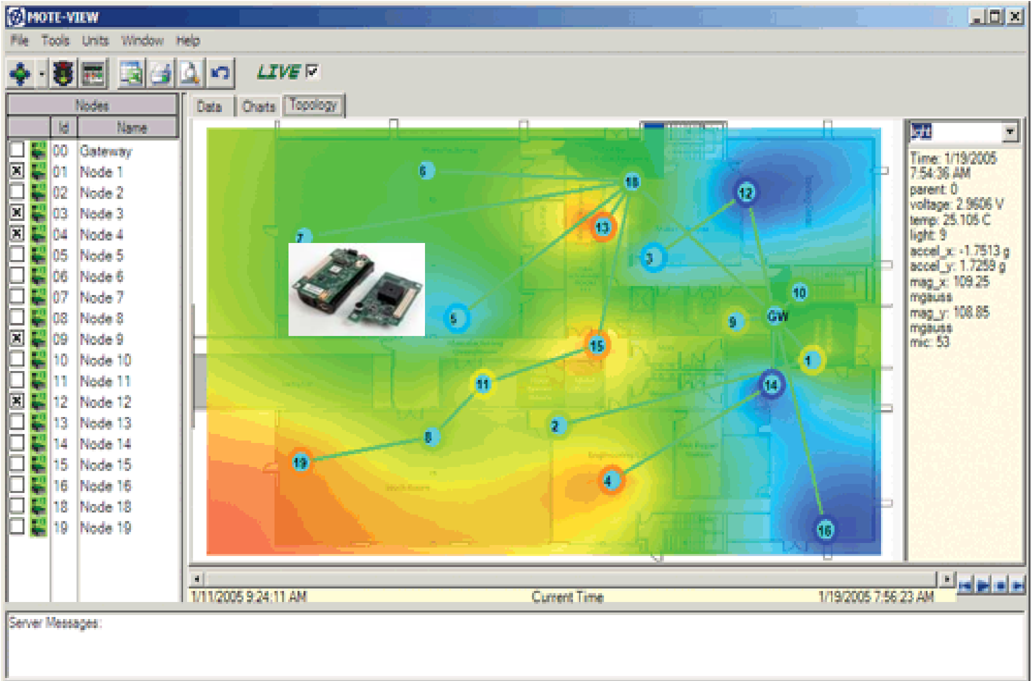
Wireless Sensor Networks

- **Distributed Sensing** of the environment.
- **Hierarchical Transfer** of readings to the sink.



Coarse Data Acquisition

- *Out-of-Network computation (at the sink)*
  - *No in-network Aggregation*
  - *No in-Network Filtering*
- *Example:* Crossbow's Moteview software



Drawback

Retrieving every single reading

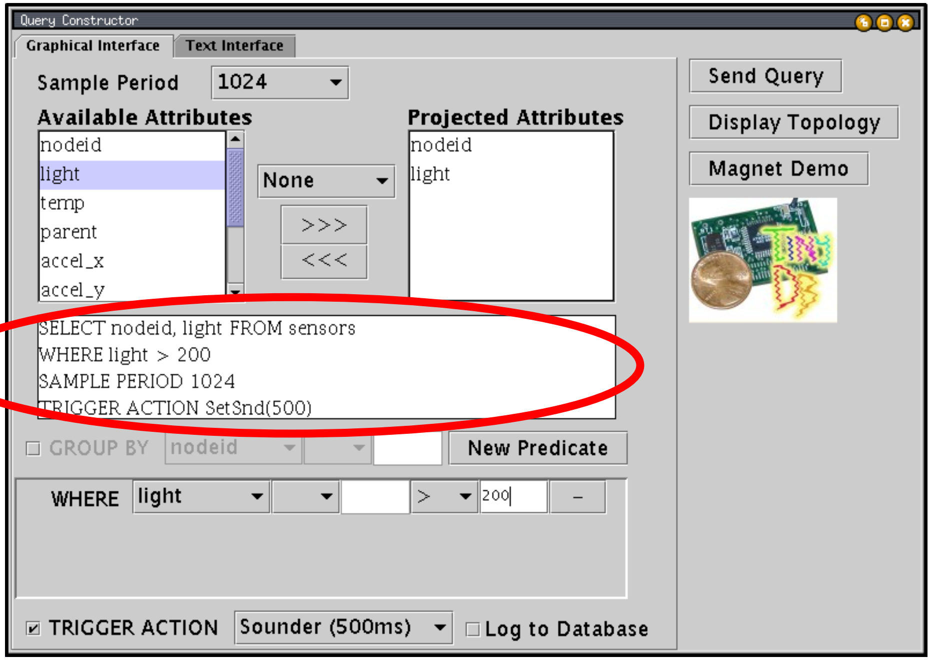
➔ too coarse and too energy demanding

Available at: http://www.xbow.com/

In-Network Computation

- *In-Network Aggregation*
- *In-Network Filtering (i.e., WHERE clause)*

Example: TinyDB: A Declarative Interface for Data Acquisition in Sensor Networks.



Drawback

The Answer set might be very large (e.g. temp>70)

Available at: http://telegraph.cs.berkeley.edu/tinydb/

Top-K Queries

- ***Example:***

  SELECT **TOP-K** room, Avg(temp)

  FROM sensors

  GROUP BY room

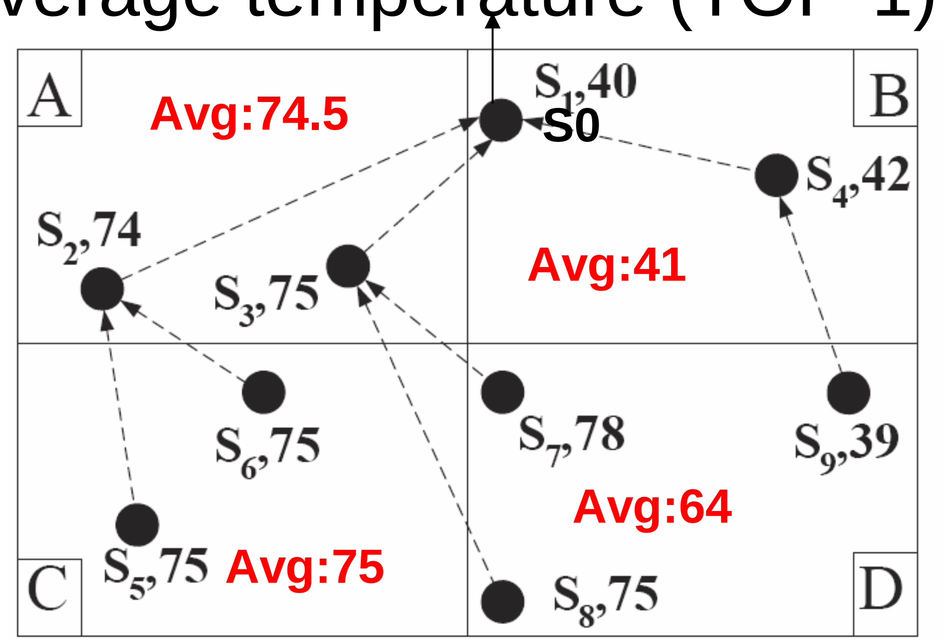
  EPOCH DURATION 1 min

- **Goal:** Trade the number of answers with the execution cost, i.e.,
  - Return less results (K<<n tuples)
  - Minimize retrieval **cost** (i.e., disk I/Os, network I/Os, CPU etc).

Centralized Top-K Pruning

Example: Four rooms {A,B,C,D}, 9 sensors {**s1,…,s9**}

Query (Q): Find the room with the highest average temperature (TOP-1)



A 4-room environment monitored by 9 sensors

Answer

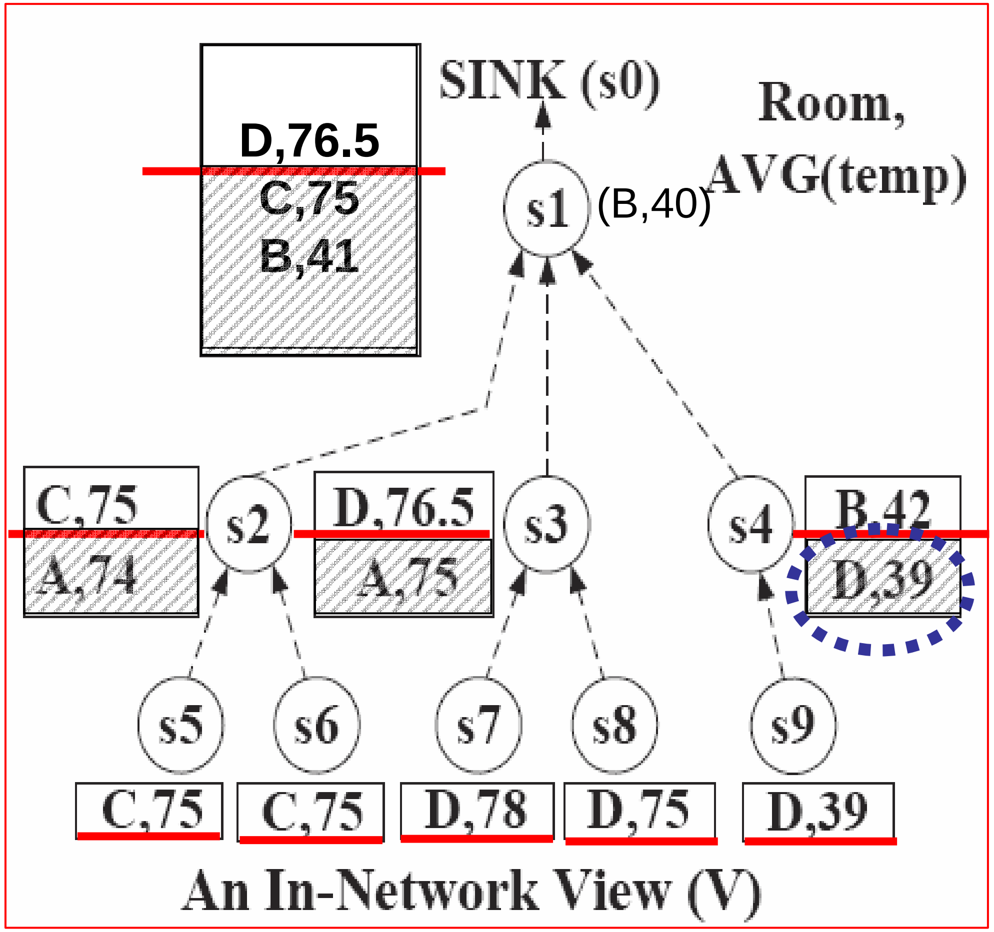
C, 75F

A, 74,5F
D, 64F
B, 41F

Drawback: No energy savings!

Naïve In-Network Top-K Pruning

Each node eliminates any tuple with a score lower than its top-1 result.



Drawback:

We received a **incorrect answer** i.e. (D,76.5) instead of (C,75).

This happens because we eliminated (D,39) that would have changed the result.

Our Approach

- ***Design and Implement framework that enables:***
  - *In-Network Aggregation*
  - *In-Network Filtering (i.e., WHERE clause)*
  - ***In-Network Top-K Pruning***

- **Problem - Challenges**
  - Determine Correct Top-K Results
  - Continuous Top-K Execution
  - Energy Constraints

Presentation Outline

- Introduction and Motivation
- **Materialized In-Network Top-K Views**
  - **Construction Phase**
  - **Pruning Phase**
  - **Maintenance Phase**
- Experimentation
- Conclusions & Future Work

MINT Framework

- The MINT Framework works in three phases:

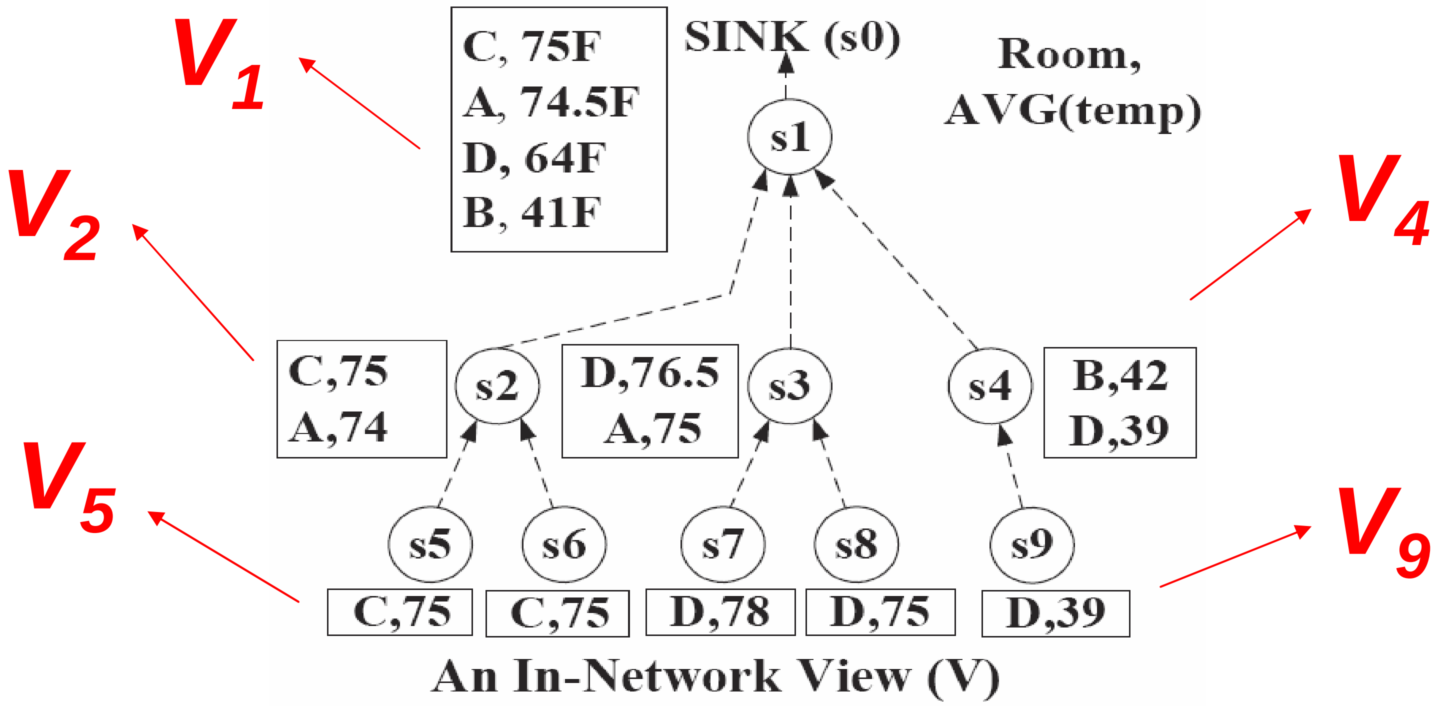
A) **Creation Phase:** Executed during the first acquisition of readings which results in ***n*** distributed views , V_i ***(i<=n)***

B) **Pruning Phase:** Each sensor s_i locally prunes V_i and generates V_i' ($\subseteq V_i$).

C) **Update Phase:** Executed once per **epoch**, during which s_i updates its parent with V_i'.

MINT: Creation Phase

- **Execute** Q locally
- **Aggregate** the result with the Query Answers from children.
- **This generates a local View V_i .**



MINT: Pruning Phase

- Each sensor node $\boldsymbol{s_i}$ now locally prunes $\boldsymbol{V_i}$ and generates $\boldsymbol{V_i'} \boldsymbol{(\subseteq V_i)}$.

- ***Problem: Each s_i needs to know which tuples will be required by its parent.***
  - Recall the elimination of (D,39) that lead to wrong answer at the sink.

MINT: Pruning Phase

- **Pruning Algorithm Outline**
  - **Bounding Step:** Locally Bound (above) each tuple in V_i with its maximum possible value.
  - **Elimination Step:** Prune away any tuple in V_i that can not be among the K highest-ranked answers.

MINT: Pruning Phase

Bounding Step

- s_i maintains a list of (room,sum) tuples.
- s_i knows some meta-information about the network, e.g.,
  - **γ1** = «max possible temperature» = **120**, and
  - **γ2** = «sensors in each room» = **5.**
- **sum'** is an upper bound for **sum**

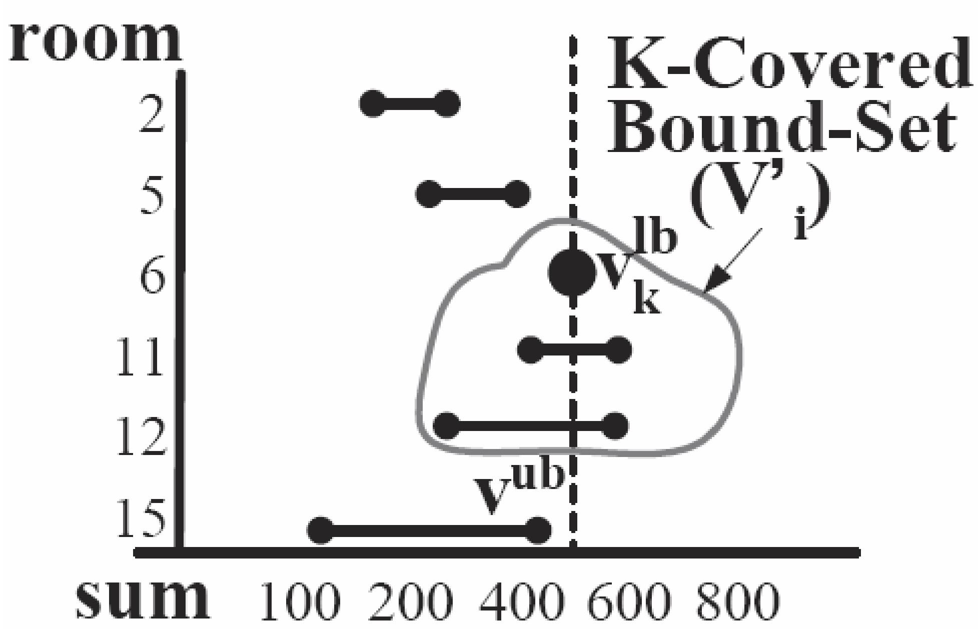
V_i

room	sum	count	sum'
2	200	4	320
5	270	4	390
6	500	5	500
11	460	4	580
12	290	3	530
15	130	2	490

sum'=sum+(γ2-count)*γ1

MINT: Pruning Phase

room	sum	sum'
2	200	320
5	270	390
6	500	500
11	460	580
12	290	530
15	130	490



Elimination Step

- Prune-away any tuple outside the K-covered-bound set.
- **K-covered Bound-set (V_i'):** Includes all the tuples which have an **upper bound (v^{ub})** greater or equal to the **kth highest lower bound (v_k^{lb})**, **i.e., $v^{ub} >= v_k^{lb}$**

The running time of the pruning algorithm is O(|*Vi*|)

MINT: Update Phase

- We assumed so far that each s_i is **state-less** (it does not remember the V_i of the previous time chronon).

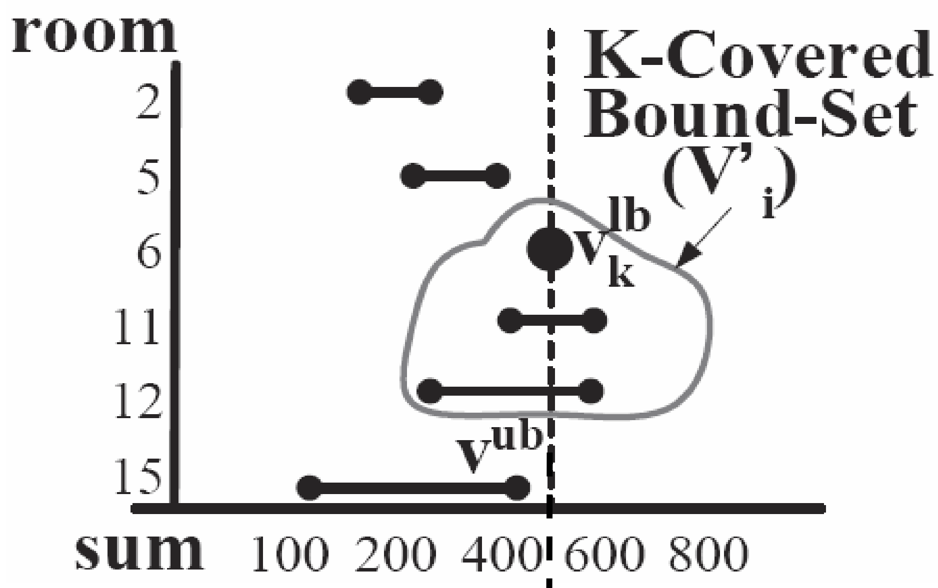
- This defines an **INT View** that is appropriate for devices with limited SRAM or FLASH storage.

- Now assume that we have adequate space to store the V_i of the previous chronon, as PV_i.

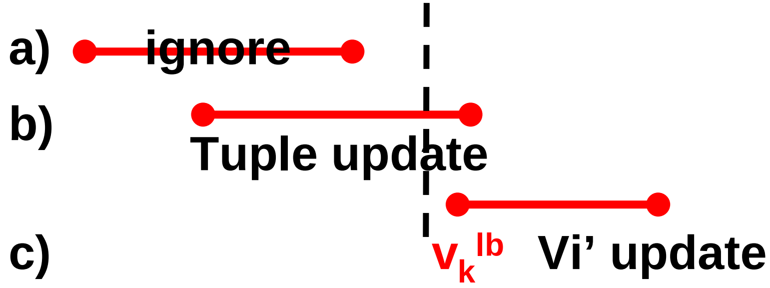
MINT: Update Phase

- Core idea of the update phase:

"Utilize PV_i in order to update the parent's View"



Update cases:



Presentation Outline

- Introduction and Motivation
- Materialized In-Network Top-K Views
  - Construction Phase
  - Pruning Phase
  - Maintenance Phase
- **Experimentation**
- Conclusions & Future Work

MINT Views: Experimentation

- **Datasets:**

1. **Great Duck Island (GDI):**
   - 14 sensors deployed on the Great Duck Island (Maine) in 2002.
   - Sensors: Temperature, Light, Humidity, Voltage...
2. **Washington State Climate (Atmomon):**
   - 32 sensors deployed in Washington and Oregon for 208 days in (2003-2004).
   - Sensors: Temperature and Wind speed.

MINT Views: Experimentation

- **Sensing Device**
  - We utilize the energy model of Crossbow's TELOSB Sensor (250Kbps, RF On: 23mA)
  - Trace-driven experimentation using **Energy = Volts x Amperes x Seconds.**

- **Query:**

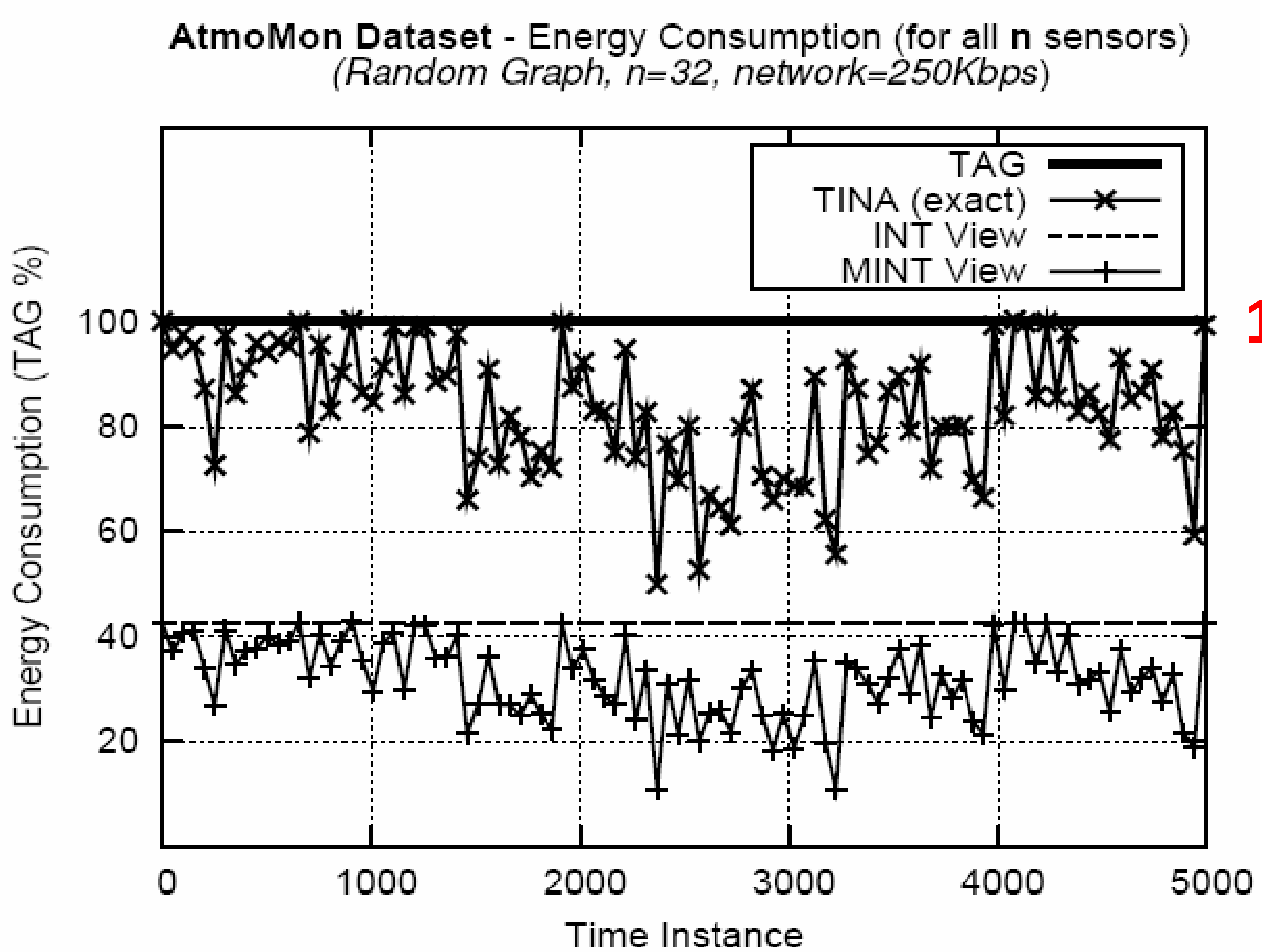
  SELECT TOP-K area, Avg(temp)

  FROM sensors GROUP BY area

  EPOCH DURATION 1 min

Energy Consumption

Atmomon Dataset



100%

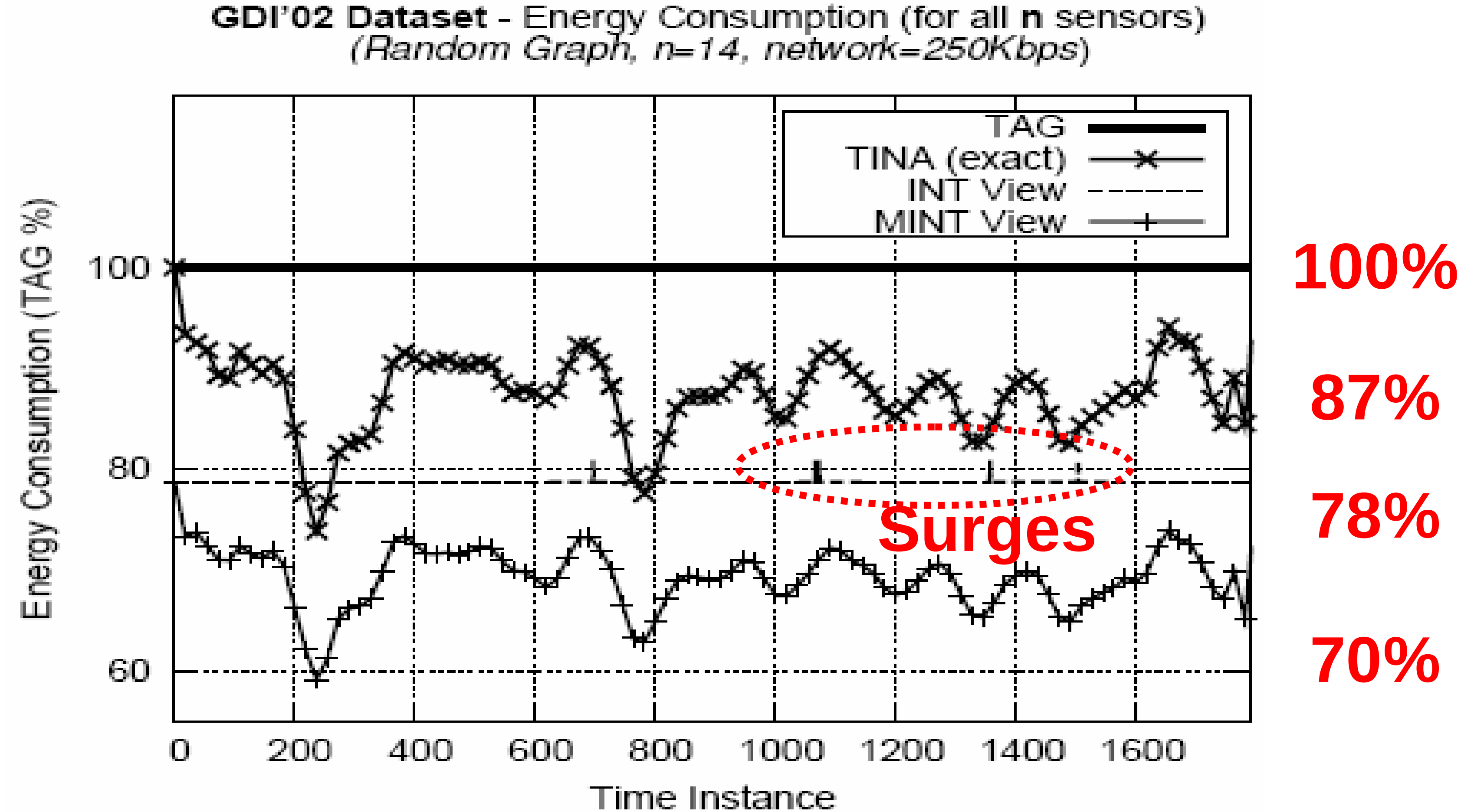
83%

42%

32%

Energy Consumption

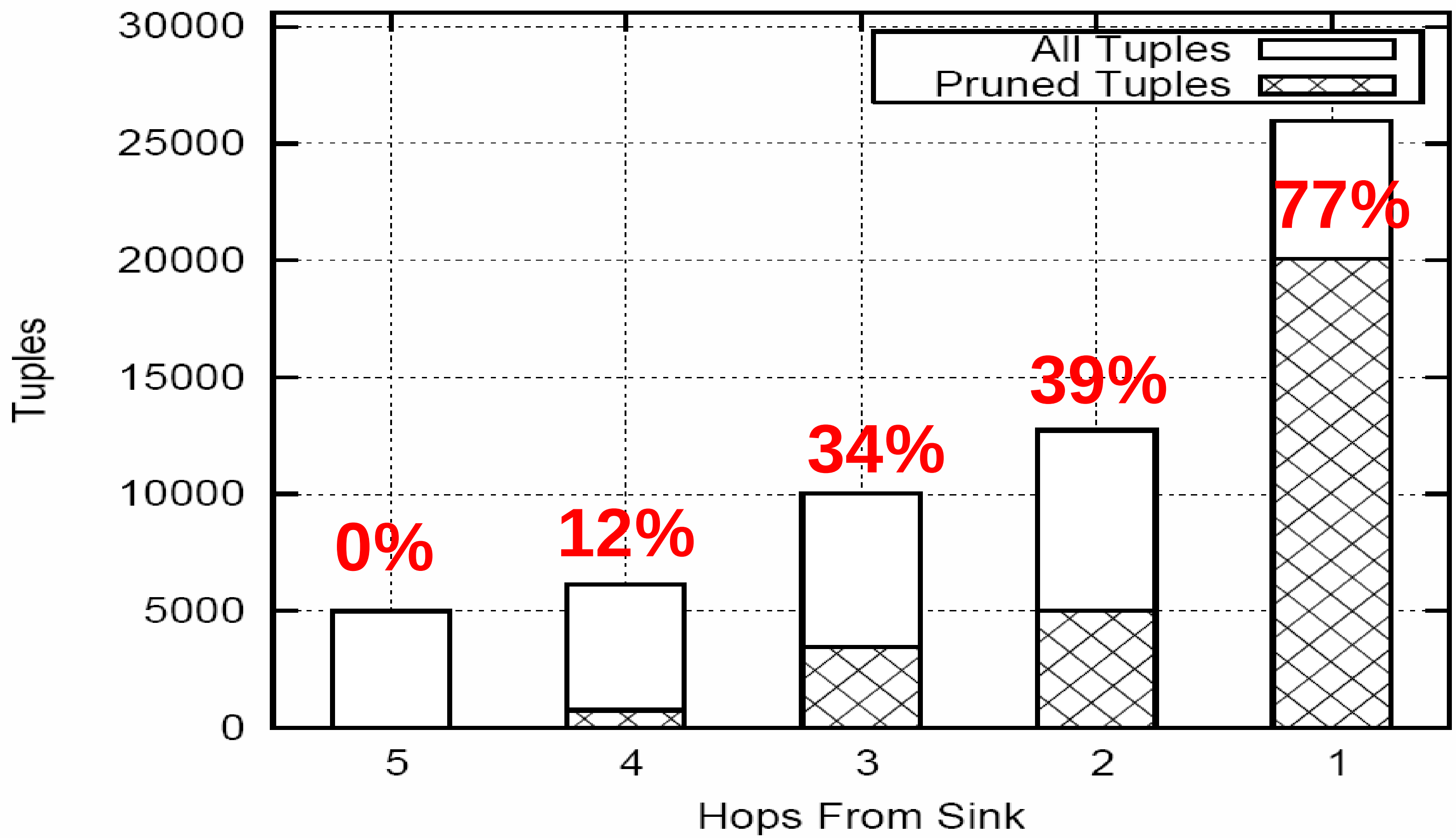
Great Duck Island Dataset



Top-k Pruning is less efficient for shallow query acquisition trees (depth=3 with 14 nodes).

Pruning Magnitude (at each level)

Atmomon Dataset



- MINT eliminates 48% of the tuples.(29K / 60K).
- Nodes closer to sink eliminate more tuples.

Presentation Outline

- Introduction and Motivation
- Materialized In-Network Top-K Views
  - Construction Phase
  - Pruning Phase
  - Maintenance Phase
- Experimentation
- **Conclusions & Future Work**

Conclusions

- We have presented MINT, a new framework for the execution of continuous queries in WSN.
- We devised efficient Construction, Pruning and Maintenance for such In-Network Views.
- Experimentation reveals that MINT can be the premise for energy efficiency in WSN.

Future Work

- We are currently implementing a **nesC prototype** of the MINT View Framework.

- **Deferred View Maintenance:** instead of updating the view on each change, propagate changes **periodically** (after a certain number of changes or randomly).

MINT Views: Materialized In-Network Top-k Views in Sensor Networks

Thank you!

Questions?

This presentation is available at:
http://www.cs.ucy.ac.cy/~dzeina/talks.html

Related Publications available at:
http://www.cs.ucy.ac.cy/~dzeina/publications.html