Building Efficient Wireless Sensor Networks with Low-Level Naming

John Heidemann et al.
USC/ISI

Key Contributions

- Exploiting application-specific naming and in-network processing for building efficient scalable wireless sensor networks
- First software architecture implemented in an operational, multi-application sensor-network

Outline

- Motivation
- Related Work
- Software Architecture
- Implementation Testbed
- Results
- Discussions

Motivation (or why <u>not</u> IP naming?)

- Traditional IP-based naming
  - Hierarchical Addressing
  - Binding At Runtime
    - Communication Overhead
  - High Bandwidth, Small Delay
- New class of distributed systems with unique requirements
  - b/w, energy constraints
  - unpredictable node and packet losses
  - Communication rather than computation is bottleneck

Motivation (or why attribute-based naming?)

- **Low-Level communication using attribute**
  - External to network topology and relevant to the application
  - Data-centric communication primitives
  - *Avoids overhead of binding and discovery*
  - *Exploits knowledge of sensor data types*
- **In-network processing**
  - Process and filter data at node
  - *Avoids Communication overhead (aggregation, nested queries etc)*
  - *Exploits knowledge about sensor applications*

Motivating Example

- Wireless monitoring system with light+motion sensors:
  - Idle state: audio sensors off, light sensors periodically monitor
  - Queries can be on either audio or light
  - Queries diffuse into n/w, handled geographically
  - Inter-sensor interaction can be pushed into n/w

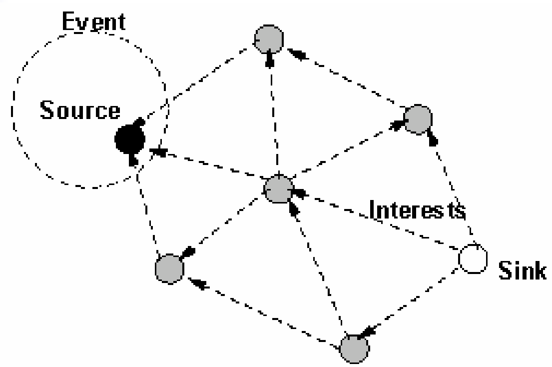
Related Work

- Internet ad hoc routing
- Jini →resource discovery
- The Piconet
- SPIN
- LEACH
- DataSpace
- COUGAR
- Declarative Routing
- Attribute-based name system

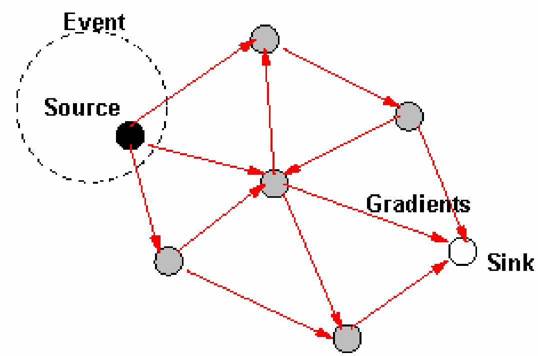
System Architecture Description

- Data managed as list of *attribute-value-operation* tuples
- *Matching rules* to identify either data arrival at destination, or filter processing
- Directed Diffusion as task-specific publish/subscribe distribution mechanism

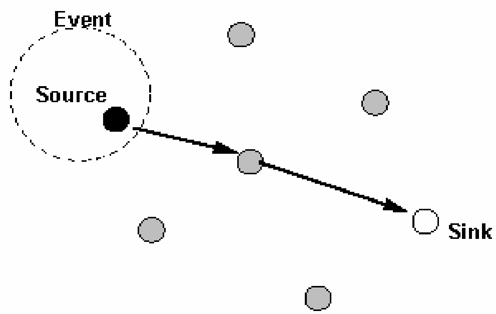
Summarizing Directed Diffusion



(a) Interest propagation



(b) Initial gradients set



(c) Data delivery along reinforced path

Attribute Matching

- Attributes have unique keys (domain)
- Have well-defined data format (even structures)
- Pattern matching done by operator fields
- Operators can be arithmetic or logic
- Can QL be shown to be complete for any domain?
  - Rectangulation of region, etc.

Matching Algorithm

Given two attribute sets A and B

For each attribute a in A where a.op is a formal {

Matched=false

For each attribute b in B where a.key=b.key and b.op is an actual:

if a.val compares with b.val using a.op then matched=true

if not matched, return false

}

Return true

Filters

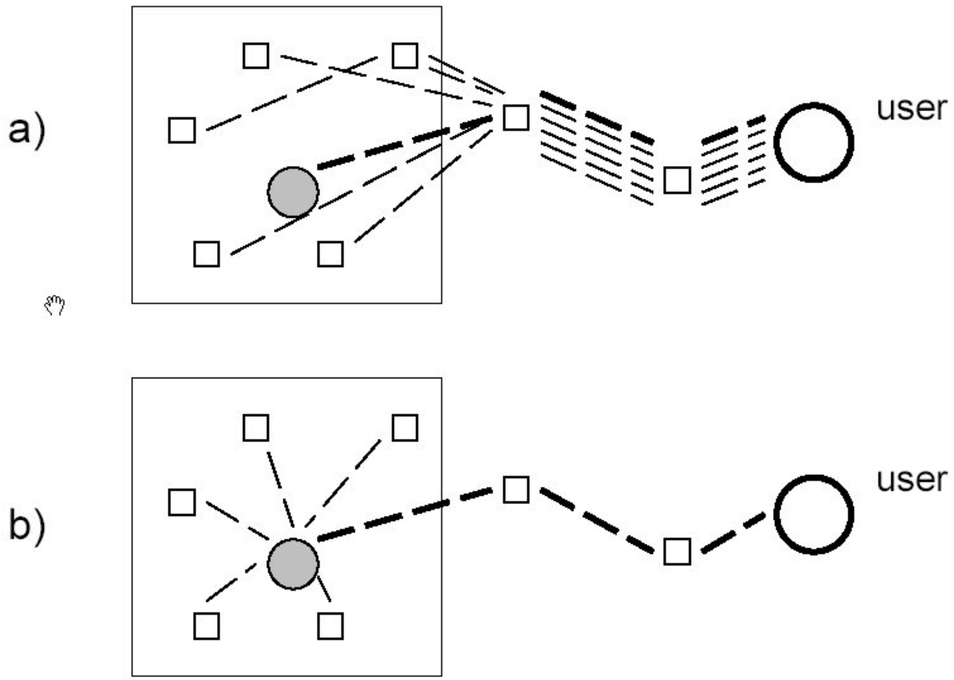
- Unique to system
- Application specific: have access to all data *and* state
- Register for handling data types, distributed as mobile code packages
- Can modify/extend/suppress/delete data and state
- For ex: generate confidence metrics about multiple sensors from multiple sampled data about number of 4-legged animals

In-network data aggregation

- Binary/region/application-specific aggregation
- *Opportunistic* aggregation:
  - Sensor selection and tasking through app-level attributes
  - Data cached as it traverses
  - App filters act on data

Nested Queries

- Goal: Reduce work (duty cycle) of multi-modal sensors by leveraging proximity and optimizing correlation triggers:
  - Ex: accelerometer triggering GPS receiver, traffic triggered n/w imager, motion sensor triggering steerable camera, etc.
  - Can be done both by source and in-network node processor
  - From paper:



Nested Queries

- Create sub-task at triggered sensor that constantly monitors nested query for events from initial sensors
- In case of multiple triggered sensors:
  - Random-delays+election mechanism
  - Use weighted distance by leveraging location (external frame of reference) to find best nodes

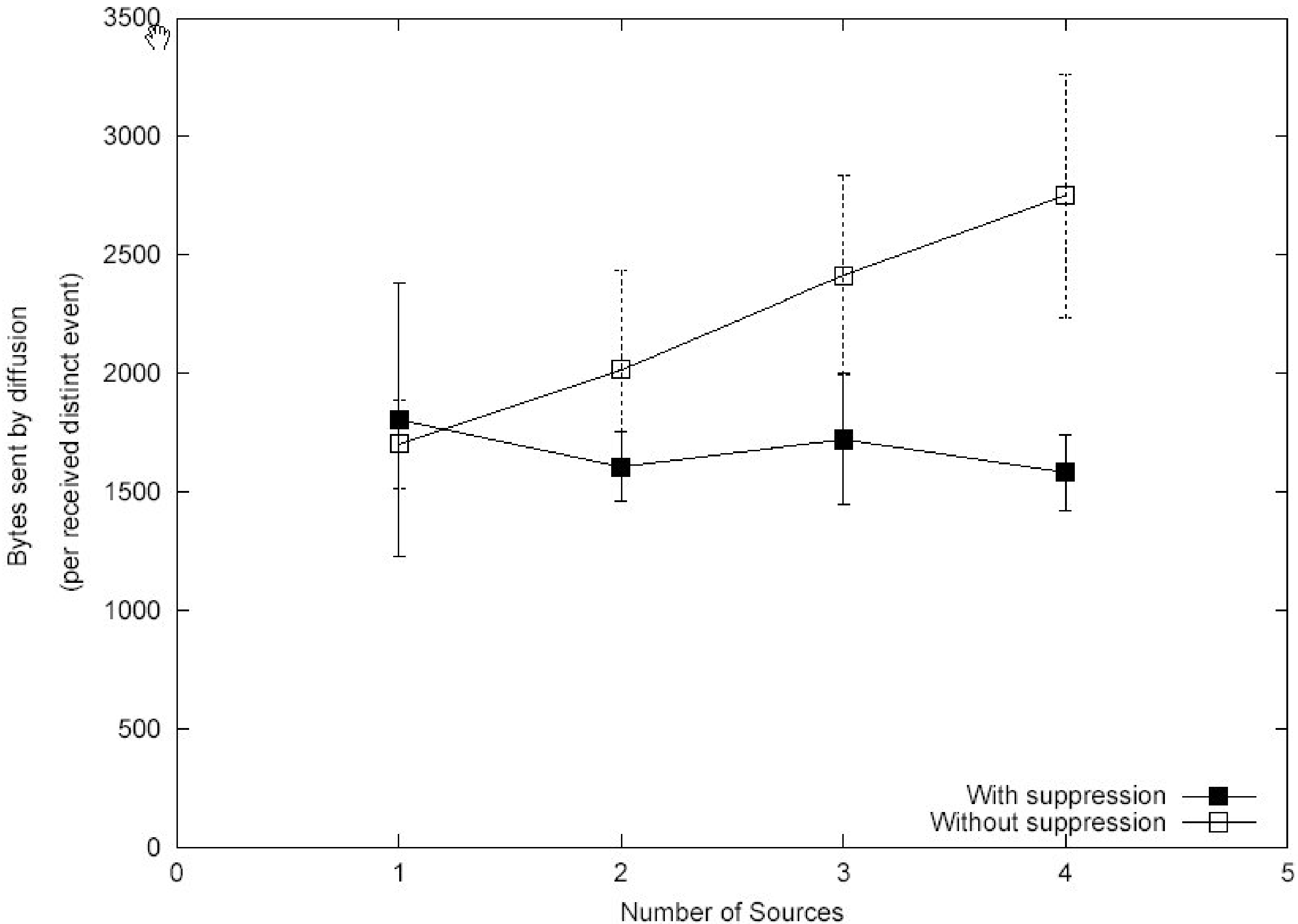
Experimental Testbed

- Quantifies benefits of aggregation, nested queries, performance of matching algorithms
- 14 PC/104 nodes, 13kb/s Radiometrix RPC modems
- 1 sink, multiple sources, ~ 4 hops apart

Measurements Notes

- MAC completely dominates energy measurements
- Don't have TDMA=>approximate energy consumed:
  - P=d*p_l*t_l+p_r*t_r+p_s*t_s
  - 1:3:40 time ratios, ~1:1.5:2 power ratios
  - Max. of 10% duty cycle for realistic benefits

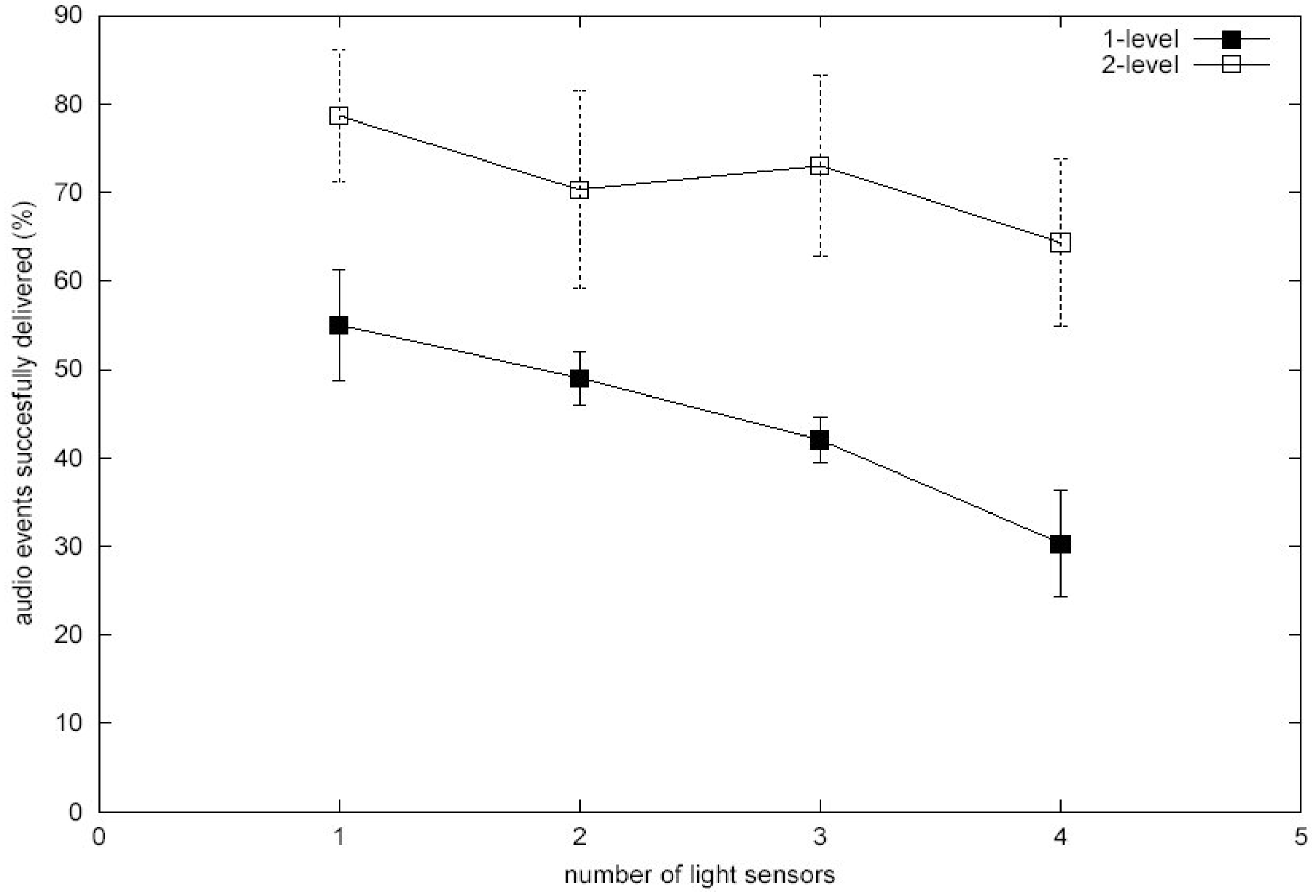
Energy graph



Notes for this experiment

- Measures bytes/event
- Unsophisticated MAC:
  - No RTS/CTS, no ARQ
  - Message fragmented into 27-byte units
  - Painfully obvious at high congestion levels
- Good back-of-envelope verification
- Upto 40% reduction for 4 sources
- Discrepancy from prev. simulation attributable to higher exploratory message %
- Delay possible killer

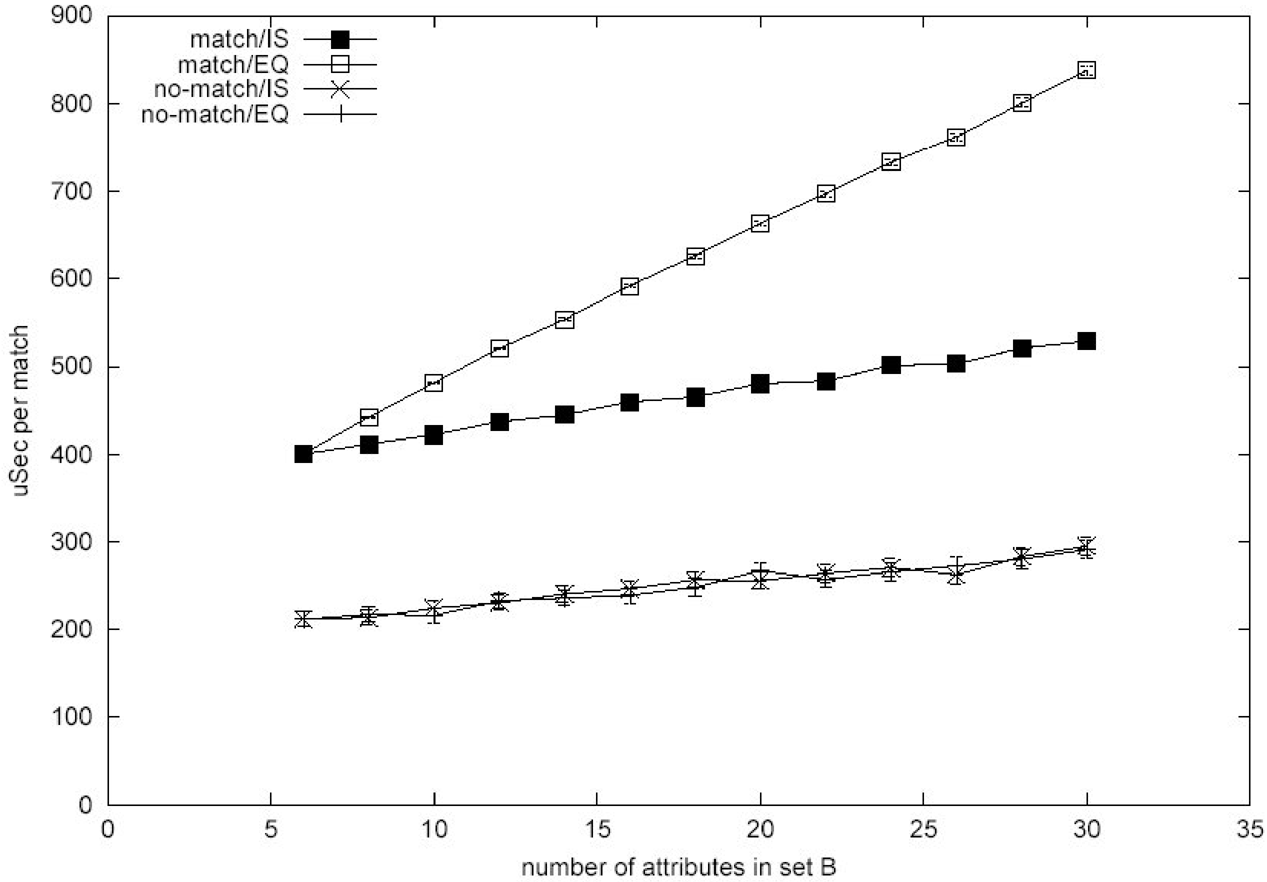
Nested Query Performance



Experiment notes

- Light state changes @1min, signal @30Hz
- Heavy congestion and loss
- Missing events=>increased latency
- Much sharper drop-off for 1-level queries
- Only 1 triggered sensor, effects more radical for more

Cost of matching



Matching Cost notes

- Cost of matching linear with no. of elements
- Incremental costs insensitive to attribute type
- Match/EQ more expensive than Match/IS
- Several optimizations possible:
  - Separate actuals from formals
  - Order them in decreasing order of rejection probability

Lessons

- Results:
  - Minimal CPU overhead of matching
  - Nested queries and filters useful
  - Low-level naming and app-specific filtering broadly a success
- Unexpected:
  - Asymmetric links: DD degrades
  - Intermittent conectivity: multi-path dissemination?
- Other future work:
  - Heavy congestion=>sense-nets must adapt to uneven densities
  - Understand tradeoffs between overhead & reliability, effects of various free parameters (i.e., parameterize), etc.
  - Include more feedback and congestion control into loop
  - Energy-aware MAC protocols absolutely must

Problems

- Scalability
- Data Reliability
  - No.of events delivered in a given time?
- Expressiveness of QL/matching
  - Will dictate what queries can be handled
- Effects of delay

Routing and Data Dissemination - Synthesizing Ideas

- Proactive Dissemination
  - Two-Tier Data Dissemination
  - SPIN (?)
  - Rumor Routing
- On-Demand Dissemination
  - Directed Diffusion
- Each suitable in a different setting
  - Either for long-standing or one-shot queries