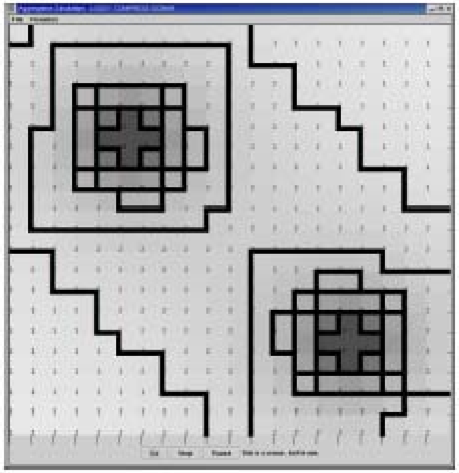
Information management (1)

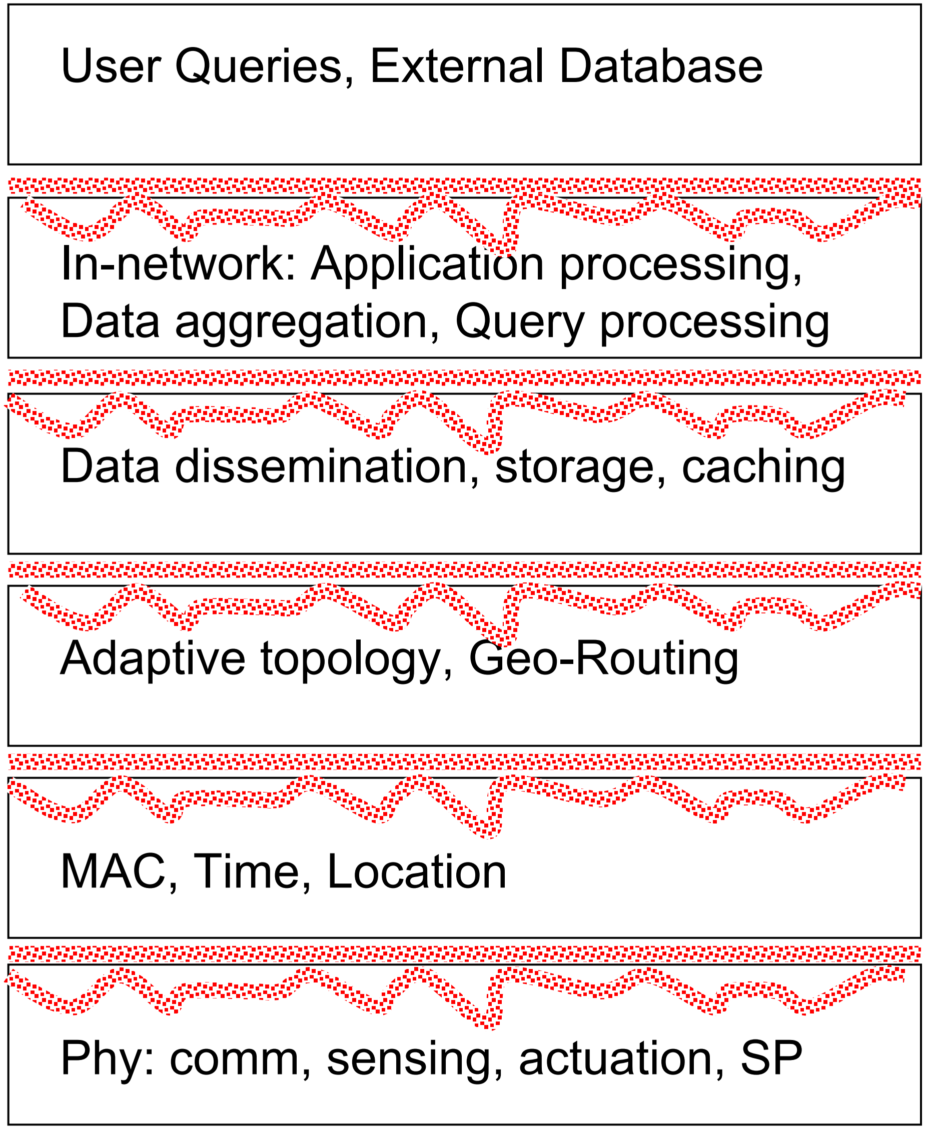
Feng Zhao



Papers

- Querying and in-network aggregation
  - Madden et al., “TAG: a Tiny AGgregation Service for Ad Hoc Sensor Networks.”
  - Hellerstein, “Beyond Averages: Towards Sophisticated Sensing with Queries.”
- Data-centric storage and access
  - Ratnasamy et al., “GHT: A Geographic Hash Table for Data-Centric Storage.”

Sensor Network Abstraction



User Queries, External Database

In-network: Application processing, Data aggregation, Query processing

Data dissemination, storage, caching

Adaptive topology, Geo-Routing

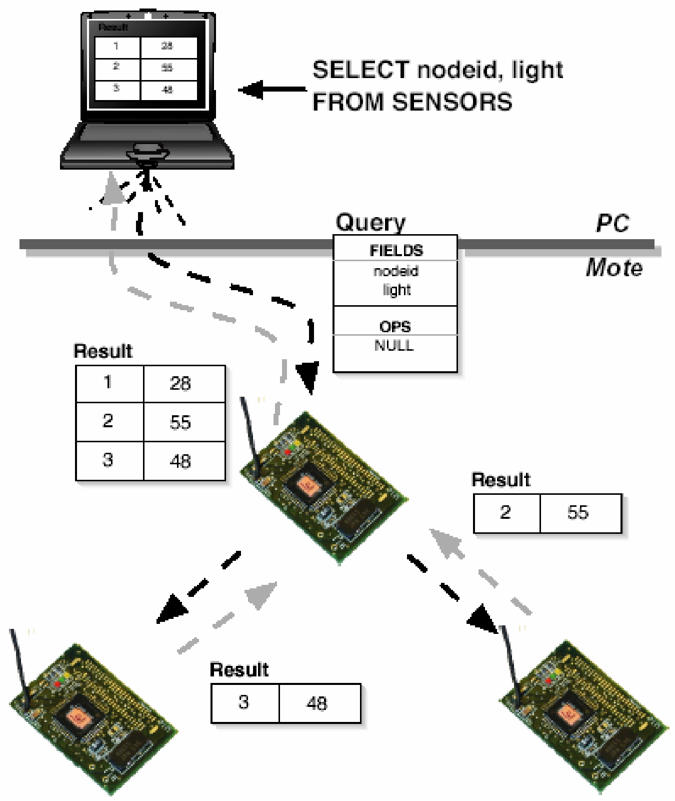
MAC, Time, Location

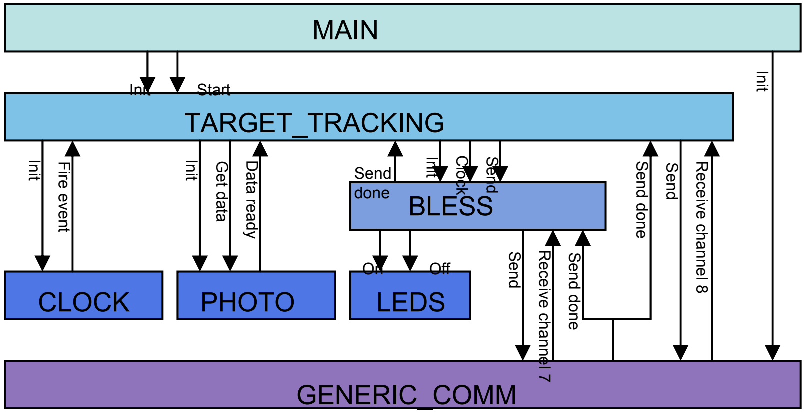
Phy: comm, sensing, actuation, SP

Figure source: Estrin et al.

- Characteristics: distributed, resource-constrained, failure prone
  - How does an application user specify a problem?
  - How does a system developer model the capabilities of the system?
  - What features should a lower layer expose to higher layers?
- From data storage point of view: think of a sensor net as a distributed database
  - How is data stored after sensing?
  - What is the interface to the network?
  - How does an external query find the data in an efficient manner?

TinyDB meets TinyOS



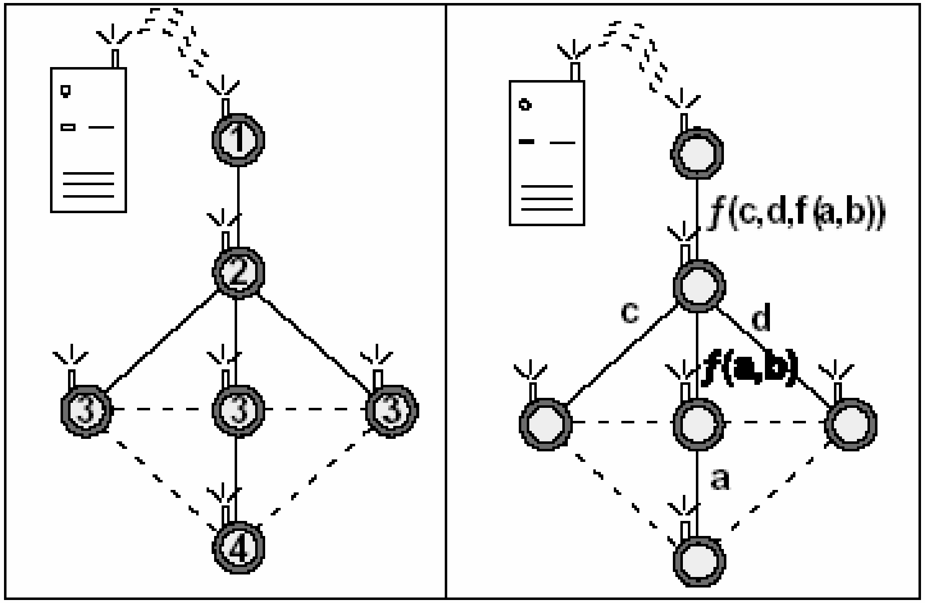


parc
Palo Alto Research Center

Querying sensor networks: TAG and Tiny DB

- SQL-like declarative interface

  -
    ```
    SELECT max(temp), room FROM sensors
      WHERE floor =6
      GROUP BY room
      HAVING max(temp) > threshold
      EPOCH DURATION 30sec
    ```
- In-network aggregation to reduce message count
  - See example on the right
- Manage transmission and aggregation:
  - A hierarchical routing tree
  - Epoch-based time division
    - » Time synch is important
  - Repair routing

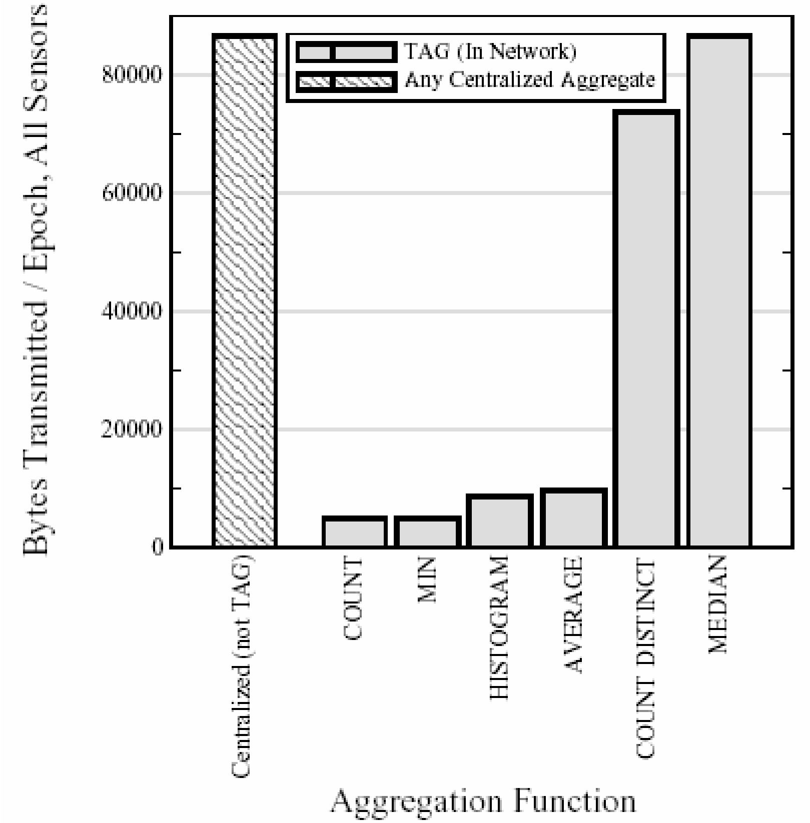


Server-based TinyDB

Left: centralized, requiring 16 message to be sent. Right: in-network aggregation, requiring 6 messages to be sent

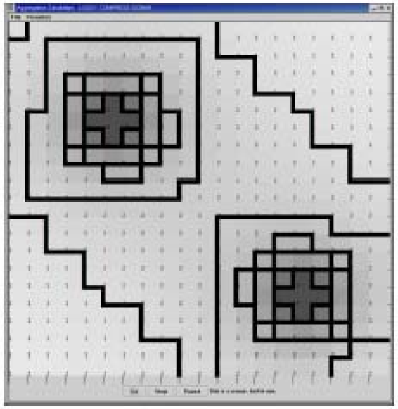
Aggregates and their efficiency in TAG

	MAX, MIN	COUNT, SUM	AVERAGE	MEDIAN	COUNT DISTINCT[4]	HISTOGRAM[5]
Duplicate Sensitive	No	Yes	Yes	Yes	No	Yes
Exemplary (E), Summary (S)	E	S	S	E	S	S
Monotonic	Yes	Yes	No	No	Yes	No
Partial State	Distributive	Distributive	Algebraic	Holistic	Unique	Content-Sensitive



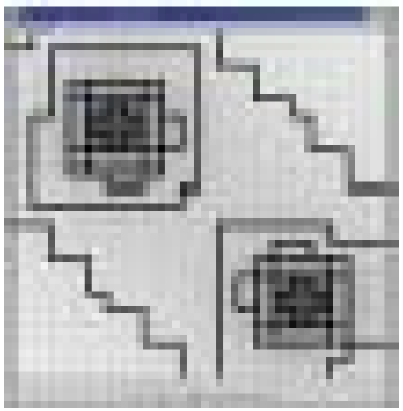
Some can be decentralized better than others!

Extend TAG beyond averages to spatio-temporal aggregates



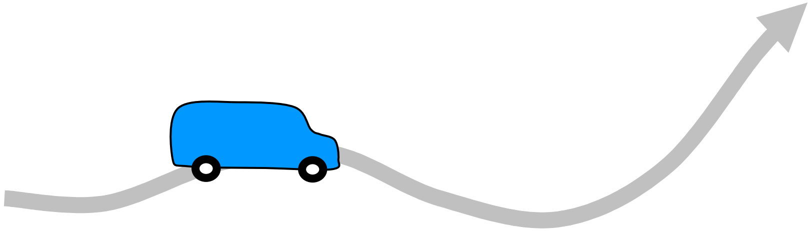
Geometry

Example: isobar mapping



Resolution

Example: wavelet compression



Time (Example: tracking)

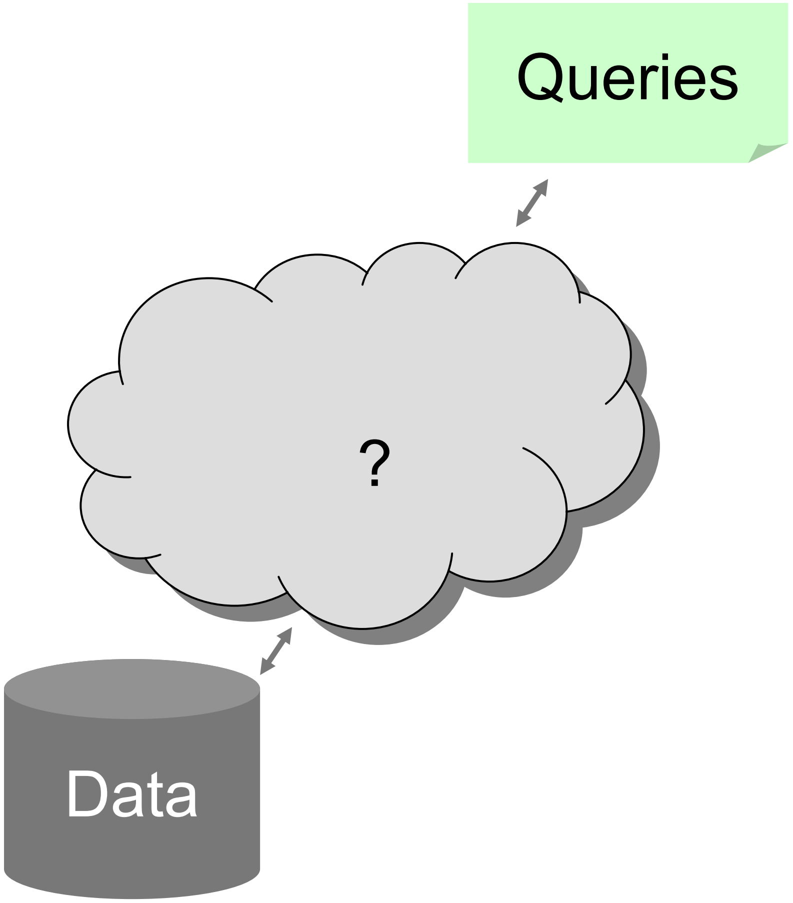
– Who triggers the aggregation?

– Interface between querying and tracking?

parc
Palo Alto Research Center

Data Centric Storage

- TinyDB routing tree or fixed storage point supports only limited access patterns
- Need more general indexing scheme
  - Scalable
  - Load balance
  - Tolerant to failures and changes
    - Persistent
    - Consistent



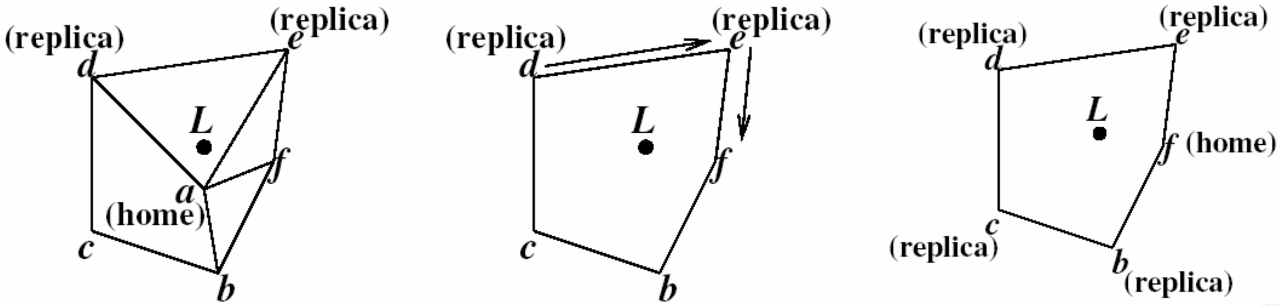
parc
Palo Alto Research Center

Geographic Hash Table

- Data is indexed by geographic coordinates
  - Data centric: data is named by physical attributes external to the nodes or network topology
  - Geographic hashing
    - » Key of a key-data pair hashed to geo location (for both PUT and GET)
    - » Load balanced
  - Storage localization:
    - » GPSR geographic routing (discussed earlier) to find storage node, defined as the node nearest to the geo location

Locate storage node using GPSR

- Home node: closest to the geo location
- Replica nodes: those along the perimeter enclosing the geo location
- Perimeter Refresh Protocol to ensure persistency and consistency
  - Node may fail or move
  - Home node periodically sends out refresh packet to the geo location. This updates the home node when necessary
  - Time-out mechanism to deal with home node failure


(replica)
d
(replica)
e
L
a
(home)
f
c
b
(replica)
d
(replica)
e
L
f
c
b
(replica)
d
(replica)
e
L
f (home)
c
(replica)
b
(replica)