Outline

- Ganesan, D., Greenstein, B., Estrin, D., Heidemann, J., and Govindan, R. **Multiresolution storage and search in sensor networks**. Trans. Storage
  - Storing sensed data can allow for temporal queries
    - E.g., what was the average temperature in the past month?
  - Challenge:
    - Location of storage
    - Location where query is processed
    - Managing the storage capacity - every storage eventually becomes full

Solution space

- Centralized storage and querying:
  - All sensors send their sensed value to a central storage
  - All queries are routed to this central storage
  - Concerns: high energy consumption for sensors near central storage
  - Works for scenarios where sensing is rare, sensed values are small and system is small (2-3 hops - storage)
- Local storage and Geographical search
  - All sensed values are stored locally
  - All queries are routed to nodes that store data
    - Use geographical information to limit coverage area
  - Spatio-temporal data need significant processing
  - Storage limitations on sensor

Local storage with distributed indexing

- Local storage but distribute an index of where objects are stored
  - Geographic hashing and structured replication
  - Each sensor hashes sensed data's name and stores the sensed value in the node that is responsible for the particular hash value
  - Query need not visit more than a single node

Design goals

- Energy efficient
- Long term storage
- Multi-resolution storage
  - Users can search for low-resolution data from a large region
  - Compressed low-resolution sensor data from a large number of nodes can often be sufficient for spatio-temporal querying to obtain statistical estimates of a large body of data
- Balanced, distributed data storage
- Robustness to failure
- Graceful data aging
- Exploiting correlations in sensor data

MultiResolution summarization

- Temporal summarization:
  - Each sensor looks at the sensed values (over time) and creates a summary
  - Compressed using wavelet coding
- Spatial summarization:
  - Create hierarchical grid-based overlay
  - Spatio-temporal wavelet compression to summar at each level
  - Especially useful in dense deployments (spatial redundancy)

Distributed Quad Trees

- Distributed QT to partition the world into quadrants
- Drill down querying from top to down
- Networked data aging
  - Lower levels (raw data) age fastest
  - Higher levels (summaries) age slowest (but also less precise)
  - Look at:
    - Distributed storage resources in the network
    - Storage requirements of different summaries
    - Incremental query benefit obtained by storing summary

Aging problem

- Communication overhead
  - Amount of data sent to higher level for summary
- Query quality across levels
- Storage overhead
- User specified aging function