CMPE 259
Wireless Sensor Networks: Data Aggregation

Xin Li
What is data aggregation?

- Aggregate data as it flows through the network
  - Concatenation
    - Raw data
    - Application independent
  - Fusion
    - Statistical computation
    - e.g. MAX/MIN/SUM/Moment
  - Reduction
    - Duplication of the same event
    - Temporal/spatial similar numerical value
Why data aggregation?

- Energy! Energy! Energy!

- Assumptions:
  1. Computation consumes less energy than communication.
  2. Computation is not the bottleneck.
  3. The data is able to be aggregated.

- Does these assumption always holds?
  - CV/VR/AR computational intensive tasks
  - Encrypted/compressed data
Why data aggregation?

- Bandwidth
  - CSMA
    - Less collision -> less power consumption
  - TDMA
    - Feasible scheduling
- Distributed information processing?
  - Load balance
- Privacy preserving?
What data to aggregate?

- Periodic
  - continues real-time monitoring
    - e.g. environmental monitoring, energy monitoring

- Sporadic
  - dynamic event detection
Where to aggregate?

- Aggregation Tree parent node [TinyDB]
  - Minimal spanning tree
  - Not robust
Where to aggregate?

- Cluster head [Cougar]
  - Hierarchical
Where to aggregate?

- Multiple neighbor nodes [Directed Diffusion]
  - Robust
  - But duplications.
Aggregate structure

- Static structure
  - Routing on a pre-computed structure
  - Suitable for unchanging traffic pattern
  - Inappropriate for dynamic event
Aggregation structure

- Dynamic structure
  - Create a structure dynamically
  - Optimization for a subset of nodes
  - High control overhead for dynamic events
Aggregation structure

• Structure-free
  • Improve aggregation without any structure
  • Suitable for dynamic event scenarios
  • No guarantee of aggregation for all packets
When to aggregate?
(periodic timing models)

- Periodic Simple Aggregation
  - each node wait a pre-defined period of time, aggregate all data item received, and send out a single packet containing the result

- Periodic Per-hop Aggregation
  - similarly to periodic simple, but transmits the aggregated data as soon as it hears from all its children

- Periodic Per-hop Adjusted Aggregation
  - nodes adjust their timeout based on their position in the data collection tree.
Performance metrics

- Energy efficiency
- Latency
- Communication cost
- Data accuracy
<table>
<thead>
<tr>
<th>Isoline</th>
<th>data type</th>
<th>aggregation</th>
<th>Structure</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>periodic data</td>
<td>Reduction</td>
<td>Structure-free</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>ToD</th>
<th>data type</th>
<th>aggregation</th>
<th>Structure</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>dynamic event</td>
<td>Fusion</td>
<td>Hybrid</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Sparse</th>
<th>data type</th>
<th>aggregation</th>
<th>Structure</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>sparse event</td>
<td>Not specified</td>
<td>Dynamic tree</td>
</tr>
</tbody>
</table>
Efficient Continuous Mapping in Sensor Networks Using Isolines [isoline aggregation]

• Basic idea
  • Spatial correlation of data
  • Group nodes that report similar readings into isoclusters

• Key Operation: Isoline detection
Isoline detection

- Isolines
  - lines which pass through our network and have the same value.
  - Detection by local comparison with neighbor readings
  - Only node detecting isoline reports to sink
Continuous monitoring

• Real-time data

• temporal correlation
  ‣ If the isoline doesn't change or there is no nearby isoline, there is no report.
Simulation Setup

- Temperature continuous monitoring
  - 16*16 nodes grid, 400 m$^2$
  - CDMA: 40m transmission range
  - Reality is simulated by a matrix of 80*40 points
  - Initial centered at 45 degrees, Aggregated at interval of 10 degrees

- Comparison alternatives
  - No aggregation
  - No aggregation optimized: temporal data aggregation
  - polygon aggregation
Two scenarios

• Hotspot

• Front moving
Simulation result

- Hotspot
  - No aggregation and isoline aggregation send similar amount of data
  - Expensive initial data collection
- Polygon aggregation sends more data
  - Aggregation happens down in the collection tree

<table>
<thead>
<tr>
<th></th>
<th>Similarity</th>
<th>KBytes sent</th>
</tr>
</thead>
<tbody>
<tr>
<td>No Agg.</td>
<td>98.7 (sd 0.09)</td>
<td>180.0 (sd 5.4)</td>
</tr>
<tr>
<td>No Agg opt.</td>
<td>98.9 (sd 0.09)</td>
<td>21.1 (sd 0.4)</td>
</tr>
<tr>
<td>Polygons</td>
<td>98.1 (sd 0.49)</td>
<td>62.9 (sd 4.6)</td>
</tr>
<tr>
<td>Isolines</td>
<td>97.0 (sd 0.36)</td>
<td>15.3 (sd 1.2)</td>
</tr>
</tbody>
</table>
Simulation result

- Moving front
  - All nodes will eventually change value.
  - Error due to packet loss.

<table>
<thead>
<tr>
<th></th>
<th>Similarity</th>
<th>KBytes sent</th>
</tr>
</thead>
<tbody>
<tr>
<td>No Agg.</td>
<td>93.2 (sd 1.72)</td>
<td>177.1 (sd 5.9)</td>
</tr>
<tr>
<td>No Agg opt.</td>
<td>89.3 (sd 0.70)</td>
<td>62.1 (sd 2.3)</td>
</tr>
<tr>
<td>Polygons</td>
<td>82.4 (sd 2.93)</td>
<td>77.0 (sd 3.6)</td>
</tr>
<tr>
<td>Isolines</td>
<td>96.7 (sd 0.50)</td>
<td>55.8 (sd 3.1)</td>
</tr>
</tbody>
</table>
Simulation result

• Moving front
Conclusion

• Isolines are an effective method of aggregating information

• What if…
  ‣ Sparse deployment
  ‣ Week spatial correlation
Scalable Data Aggregation for Dynamic Events in Sensor Networks

• Target: rare dynamic events

• Related work
  
  › Statistic Structure
    
    - Suitable for unchanging traffic pattern;
  
  › Dynamic Structure
    
    - High control overhead for dynamic events
  
  › Structure-Free
    
    - Suitable for dynamic event scenarios;
    - Not scalable
Approach:
Tree on Directed Acyclic Graph

- Combine benefits of structured and structure-free approaches
- Two-stage method
  - Structure-free data aggregation: early aggregation
  - Packet forwarding on an implicit structure: scalability
ToD - Tree on DAG

- One-Dimension illustration

- Definition
  - Cell: Cell size is the maximum diameter of events
  - F-cluster: First-level Cluster. Composed of multiple cells
  - S-cluster: Second-level Cluster. Composed of multiple cells
    - Interleaved with F-clusters
ToD - Tree on DAG

F-cluster-head

S-cluster-head

S-cluster
Dynamic Forwarding

- Rule 0: forward packets to F-cluster-head by structure-free data aggregation protocol [Infocom ’06]

- Rule 1: event spans two cells, forward to sink

- Rule 2: event spans one cell, forward to S-cluster-head
Two-Dimension ToD Construction

F-Clusters

Cells

S-Clusters
Experimental Results

- Evaluated Protocols
  - ToD
  - Data Aware Anycast (DAA) (includes RW)
  - Shortest Path Tree (SPT)
  - SPT with Delay (SPT-D)

- Testbed Configuration
  - 105 Mica2-based motes
  - 15 * 7 grid network
  - TX Range: 2 grid-neighbor (max 12 neighbors)

- Evaluated Metric
  - Normalized Number of Transmissions
    \[
    \frac{\text{Number of Total Transmissions}}{\text{Number of Contributing Sources}}
    \]

- Parameters
  - Maximum Delay
    - ToD, DAA, SPT-D
  - Event Size
Experiment Results - Delay

- All nodes are sources
- Data rate: 0.1 pkt/s
- Data payload: 20 bytes
- 2 F-clusters in ToD

Key observations
- ToD performs better than DAA
- SPT-D is sensitive to the delay
Experiment Results – Event Size

- 12 ~ 78 sources
- Data rate: 0.1 pkt/s
- Data payload: 20 bytes
- SPT-D delay: 6s

Key observations
- ToD performs best
- High variation of SPT-D: Long stretch problem
Conclusion

• Structure-Free Aggregation

• Dynamic Forwarding on ToD for Scalability

• Efficient Aggregation without overhead of structure computation and maintenance
Sparse Data Aggregation in Sensor Networks

- Problem
  - Aggregate data from a sparse set of nodes.
    - Events are rare.
    - e.g. anomaly detection
  - No global information on where all these nodes are located.

- Goals:
  - Autonomously discover each other in a distributed fashion.
  - Ad hoc Aggregation structure
Network setup

- Sensor nodes are uniformly deployed inside a regular region.
- The boundary of the field is known and connected to high-speed network.
Tree-based Sparse data aggregation

- Each hot-node has a unique priority number
- Base station node has the highest priority
- The hot nodes with data participate in the tree formation protocol composed of two sub-protocols:
  - The probe protocol: node ID + node priority number
  - The recall protocol: parent node ID.
- The hot nodes tries to find the nearest hot node with highest priority.
- Nearly optimal
Tree formation

Assume that priority $p > k > q$

Probe
Recall

Routing

- The aggregation tree is a logical structure: each node $p$ knows its parent $q$ in the tree.
- Routing can be done by several choices:
  - Send a packet from $p$ to $q$ along $p$’s trail to the junction node $w$, then along $q$’s trail to $q$.
  - Use some network-specific point-to-point routing mechanism.
  - Multi-path depend on the importance of the message.
Probabilistic Aggregation

- Exponential distribution

$$f(x; \lambda) = \begin{cases} 
\lambda e^{-\lambda x}, & \text{if } x \geq 0 \\
0, & \text{if } x < 0
\end{cases}$$

$$F(x; \lambda) = \begin{cases} 
1 - e^{-\lambda x}, & \text{if } x \geq 0 \\
0, & \text{if } x < 0
\end{cases}$$

- $E[x] = 1/\mu$

- $\text{Var}[x] = 1/\mu^2$
Probabilistic aggregation

**Theorem 6.2.** If $x_1, x_2, \ldots, x_n$ are independent exponential random variables, where $x_i$ has parameter $\lambda_i$, then

$$\min(x_1, x_2, \ldots, x_n)$$

is an exponential random variable with parameter $\sum_{i=1}^{n} \lambda_i$.

- $E[\min(x_1, x_2, \ldots, x_n)] = \frac{1}{\sum_{i=1}^{n} \lambda_i}$
- Robust to data loss/duplication
Simulation Setup

• Alternatives for comparison
  • Pull: query and answer (shortest path)
  • Push: nodes themselves report (shortest path)

• Communication cost
  • Proportional to Euclidian distance
  • Sparse aggregation: tree build + data transmission
  • Pull: Query + data transmission (w or w/o aggregation)
  • Push: data transmission (no aggregation)
Simulation

• Regular 100*100 grid

• separation ratio: ratio of the diameter of the hot nodes and the shortest distance between hot nodes and boundary

• Without in-network aggregation: better for all parameter settings.
Simulation

- With in-network “pull” aggregation: better when hot nodes are sparse.
- With in-network “push” aggregation: better unless the hot nodes are too few and their separation is small

Conclusion

• Distributed Tree-based sparse data aggregation

• Communication is more efficient compared to the “pull” approach without in-network aggregation and the “push” approach with in-network aggregation.

• Probabilistic aggregation
Limitations

- Not all boundary nodes are directed connected
- Grid deployment
- Timing is not discussed

\( q \) doesn’t know whether it needs to wait for data from other nodes or not.
## Recap

<table>
<thead>
<tr>
<th>Structure</th>
<th>data type</th>
<th>aggregation</th>
<th>Structure</th>
</tr>
</thead>
<tbody>
<tr>
<td>Isoline</td>
<td>periodic data</td>
<td>Reduction</td>
<td>Structure-free</td>
</tr>
<tr>
<td>ToD</td>
<td>dynamic event</td>
<td>Fusion</td>
<td>Hybrid</td>
</tr>
<tr>
<td>Sparse</td>
<td>sparse event</td>
<td>Not specified</td>
<td>Dynamic tree</td>
</tr>
</tbody>
</table>
Thanks!