Event Query Processing Based on Data-Centric Storage in Wireless Sensor Networks

Longjiang Guo, Yingshu Li, and Jianzhong Li
Harbin Institute of Technology, China
Georgia State University, USA
Outline

- Why? Motivation
- What? Preliminaries
- How? Event Formulation
- How to store? Event Storage
- How to query? Event Query Processing
- Conclusion
Motivation

- A popular application of sensor networks is *event monitoring*.
- In such applications, the observers may not be interested in the sensors or the raw data from the sensors, but more interested in the *events*. 
Motivation (cont.)

- “What is the temperature sensed by the 10th sensor?”
- “Which events did happen in region $R$ from 10pm to 11pm?”
- “Where did earthquake happen from 3am to 5am?”

Event queries: more complex queries with multiple attributes and complex predicates!
Preliminaries

- **Observation attribute** $A_i$ ($i=1\ldots n$) refers to an attribute a sensor observes.

- **Event type** refers to a set of pre-defined observation attributes and the corresponding predicates defined on the attributes.

  - $ET_i(A_{i1}, A_{i2}, \ldots, A_{ij}) = P_{i1} \land P_{i2} \land \ldots \land P_{ij}$, where $A_{ij}$ is an observation attribute, and $P_{ij}$ is the predicate defined on $A_{ij}$.

  - $fire = ET(Tmpt, Smk, Hudit)$
    
    $= (Tmpt>t1) \land (Smk>s2) \land (Hudit<h3)$.
Preliminaries (cont.)

- **Observation zone:** each observation zone can be assigned an ID.
Preliminaries (cont.)

- **Event observation node**
  - If we want to monitor an event $ET_i(A_{i1}, A_{i2}, \ldots, A_{ij})$, we need some nodes to observe the attributes $A_{i1}, A_{i2}, \ldots, A_{ij}$. These nodes are called the observation nodes of $ET_i$. 

![Event observation diagram](Image)
Event Formulation

Assumptions

– Each node knows which observation zone it belongs to.

– Up to $W$ types of events $ET_1, ET_2, ..., ET_w$ in all the observation zones are monitored.
Event Formulation (cont.)

- Within each observation zone, every node $S_{ij}$ broadcasts a message ($ET_i$, $ID$, $pos.$).
Event Formulation (cont.)

- **Event fusion node**
  - In order to capture the event of type $ET_i$ and uniformly consume the energy among nodes $S_{i1}, S_{i2}, ..., S_{iki}$ for each observation zone, a node $S_{ij}$ is selected in cyclic as the event fusion node for $ET_i$. 
return Red-ET₁ Green-ET₂ Blue-ET₃
Event Storage

- **Event storage node:** is used to store all the events of the same event type within the whole sensor network.

- Based on the positions of the event storage nodes, we can classify all the event storage schemes into three categories:
  - External Storage (ES)
  - Local Storage (LS)
  - Data-Centric Storage (DCS)
External Storage
Local Storage

- The event storage node is the same as the event fusion node.

- Event information is stored locally at an event fusion node.
Data Centric Storage

\[
\text{Hash(“fire”)} = (11,28)
\]

Sink

\[
(11, 28)
\]

\[
(“fire”, \text{info})
\]

\[
\text{Hash(“fire”)} = (11,28)
\]

\[
(“fire”, \text{info})
\]
Center Mapping-DCS

- The general idea of CM-DCS is:
  - Events of the same type are hashed to a particular observation zone lying at the center of the network.
  - A node in the observation zone nearest to the center of the network serves as an event storage node for an event type.
CM-DCS (cont.)

- **Theorem 1:** If the observation nodes are distributed uniformly in the observation zone, the event storage nodes should be located near the center of the sensor networks so that the event transmission energy consumption can be minimized.
central region

\[ ET_1 \rightarrow Z_6; \ ET_2 \rightarrow Z_7; \ ET_3 \rightarrow Z_{10}; \ ET_4 \rightarrow Z_{11} \]
Storage at Event Storage Node

- **Time-stamped vector-based storage strategy**
  - \( B(ET_i) = \{ (l, t) \mid l = (j_1, j_2, \ldots, j_{m \times n}) \} \)
  - **Event insertion**: upon receiving an event \( e(ET_i, g, t) \), set the \( g^{th} \) bit of \( l \) to 1.
  - **Event deletion**: if the number of the vectors has reached the upper bound, the oldest (wrt \( t \)) vector is deleted.
Query Processing Based on CM-DCS

- **EP-CM-DCS** is composed of four phases.
  - Phase 1: Deciding the routing destination.
  - Phase 2: Routing query $Q_{\{[t_1, t_2], ET_i\}}$ from the sink to the event storage node.
  - Phase 3: Answering query $Q$.
  - Phase 4: Routing the query result back to the sink from event storage node.
Query Processing Based on Local Storage

- EP-LS is composed of three phases.
  - Phase 1: Query dissemination.
  - Phase 2: Collection of children’s IDs.
  - Phase 3: Combination of query results.
  - Phase 4: Routing the query result back to the sink from event storage node.
Choosing the Correct Storage Scheme

- **Observation 1**: if $N_q > N \cdot \text{prob}$, we should adopt the external storage strategy. Otherwise, we should adopt CM-DCS.

- **Observation 2**: if $W \cdot N \cdot \text{prob} < N_q \left(4\sqrt{2N} \left(1+3N/(2XY) \pi R^2\right)-9\right)$, i.e. the node density $\rho$ and $N_q$ increase, we should adopt CM-DCS. Otherwise, if $\rho$ is a constant and $N$ increases, we should adopt the local storage strategy.

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$N_q$</td>
<td>Number of queries in each round</td>
</tr>
<tr>
<td>$N$</td>
<td>Number of sensor nodes</td>
</tr>
<tr>
<td>$\text{prob}$</td>
<td>Event occurring probability</td>
</tr>
<tr>
<td>$W$</td>
<td>Number of monitored event types</td>
</tr>
<tr>
<td>$\rho$</td>
<td>Node density</td>
</tr>
<tr>
<td>$X/Y$</td>
<td>Length of the monitored area on X/Y axis</td>
</tr>
</tbody>
</table>
Simulation

- Simulation software: *ns*-2
- PC: P4 CPU, 512MB RAM
- MAC protocol: 802.15.4
- The cost of answering a query: total amount of energy spent on communication.
Simulation Parameters Used in *ns*-2

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>Radio range</td>
<td>10m</td>
</tr>
<tr>
<td>Sleeping power in watts</td>
<td>0.035W</td>
</tr>
<tr>
<td>Transmitting power in watts</td>
<td>0.66W</td>
</tr>
<tr>
<td>Initial energy per node in joules</td>
<td>10000J</td>
</tr>
<tr>
<td>Receiving power in watts</td>
<td>0.4W</td>
</tr>
<tr>
<td>Number of bit vectors saved at a node</td>
<td>50</td>
</tr>
</tbody>
</table>
# Topology

Table 2. Sensor network field size and the number of nodes when node density varies

<table>
<thead>
<tr>
<th>Sensor network field size</th>
<th>Number of nodes ($N$)</th>
<th>Node density ($\rho$)</th>
</tr>
</thead>
<tbody>
<tr>
<td>64 x 64</td>
<td>100</td>
<td>8 nodes / 314 m$^2$</td>
</tr>
<tr>
<td>90 x 90</td>
<td>250</td>
<td>10 nodes / 314 m$^2$</td>
</tr>
<tr>
<td>110 x 110</td>
<td>460</td>
<td>12 nodes / 314 m$^2$</td>
</tr>
<tr>
<td>127 x 127</td>
<td>720</td>
<td>14 nodes / 314 m$^2$</td>
</tr>
<tr>
<td>142 x 142</td>
<td>1020</td>
<td>16 nodes / 314 m$^2$</td>
</tr>
</tbody>
</table>
Effect of node density on energy consumption

The sink locates close to (0,0), $prob=0.5$, $W=4$, $N_q=5$.

Effect of $N$ on energy consumption

Node density is fixed at 8nodes/314m$^2$, $prob=0.5$, $W=4$, $N_q=5$. 
Effect of $W$ on energy consumption

The monitor area is 127*127, $N=720$, $prob=0.5$, $N_q=5$.

Effect of $N_q$ on energy consumption

The monitor area is 127*127, $N=720$, $prob=0.5$, $W=4$. 
Effect of \(prob\) on energy consumption

The monitor area is 127*127, \(N=720\), \(W=4\), \(N_q=5\).
Conclusion

- Storage scheme: data centric storage.

- How to describe the event more efficiently?
- How to answer a query in a timely manner?
- How to better balance the energy consumptions of sensors?
Thanks