Nearly Constant Approximation for Data Aggregation Scheduling in Wireless Sensor Networks

Scott C.H. Huang, Peng-Jun Wan, Chinh T. Vu, Yingshu Li, and Frances Yao

05-08-2007
Outline

• Why? Motivation

• What? Preliminaries

• How? Scheme

• Conclusion
Motivation
Monitoring Applications of WSNs
User Interest

• Users expect the sensed data to be gathered quickly.

• Data aggregation is a popular operation.
  – E.g. Max, Min, Sum, Avg, and etc.
  – In-network data processing.
  – Try to reduce latency by designing a reasonable schedule.
Challenges

• Collisions waste scarce resources of WSNs and incur more latency.
• Aggregation tree based schedule
• Data aggregation scheduling is NP-hard
  – Previous schedule latency bound: \((\Delta - 1)R\).
  – Improvement: \(23R + \Delta - 18\).
Preliminaries
Maximal Independent Set (MIS) is a maximal set of pair-wise non-adjacent nodes.
Network Model

- **Unit Disk Graphs (UDG):** intersection graphs of circles of unit radius
- In a UDG, a node is adjacent to at most 5 MIS nodes.
Collision-free Aggregation

- Given a UDG $G = (V; E)$ with a base station $b \in V$. Consider two subsets $A, B \subset V$ with $B \subset A$. If all nodes in $A-B$ transmit in one slot simultaneously and the data can be received collision-free by some nodes in $B$, then data are *aggregated* from $A$ to $B$ in one time slot.
Collision-free Aggregation Schedule

- Aggregation schedule: a sequence of senders \( \{S_1, \ldots, S_l\} \) where \( S_i \subseteq V \) satisfying the data aggregation property. 
- \( l \) is the data aggregation latency.
Scheme
Approximation Algorithm

1. Data Aggregation Tree Construction
   - Construct a BFS tree
   - Find MIS nodes layer by layer
   - Interconnect MIS nodes

2. Data Aggregation Scheduling
   - First-fit scheduling
   - Schedule design
Data Aggregation Tree Construction

The original network

The BFS tree
Data Aggregation Tree Construction (cont.)

1. Aggregation link
2. BFS link
3. Network link
Data Aggregation Scheduling

First-fit scheduling

Input:
S={1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11}
Aggregation Tree T

Output:
X1={9, 8, 4}
X2={10, 6, 2}
X3={11, 3}
X4={7}
X5={5}
X6={1}
Data Aggregation Scheduling (cont.)

<table>
<thead>
<tr>
<th>Time slot</th>
<th>Senders</th>
<th>Receivers</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>10, 8, 4, 3</td>
<td>9, 5, 2, 1</td>
</tr>
<tr>
<td>2</td>
<td>11, 6</td>
<td>9, 5</td>
</tr>
<tr>
<td>3</td>
<td>9</td>
<td>7</td>
</tr>
<tr>
<td>4</td>
<td>7</td>
<td>5</td>
</tr>
<tr>
<td>5</td>
<td>5</td>
<td>2</td>
</tr>
<tr>
<td>6</td>
<td>2</td>
<td>1</td>
</tr>
</tbody>
</table>

<Data symbol>: Data
Performance Ratio

- The performance ratio of our algorithm is $23R + \Delta - 18$, which is an improvement over the previous best algorithm with performance ratio of $(\Delta-1)R$.
  - $\Delta$ contributes to an additive factor instead of a multiplicative one
  - A nearly constant approximation
  - A significantly less latency bound especially for dense networks.
Simulation Results

![Aggregation times when the transmission range fixed to 30](image)

- **Our algorithm**
- **Chen et al.**

**X-axis:** Number of nodes
**Y-axis:** Times
Simulation Results (cont.)

![Graph showing time vs. transmission range with two lines: one for "Our algorithm" and another for "Chen et al." with different slopes and time values at various transmission ranges.](image)
THANK YOU