CORD: Energy-efficient Reliable Bulk Data Dissemination in Sensor Networks

報告者: 郭逸旻
出處: INFOCOM 2008
Outline

- Introduction
- Related work
- CORD design
- Protocol evaluation
- Conclusion
INTRODUCTION

- Reprogramming problem?
  - Updating their software.
- Reliable bulk data dissemination protocol
  1. These protocols is that the data object is propagated from the sink to the rest of the network in a neighborhood by neighborhood fashion.
     Ex: Deluge, MNP, MOAP
2. In contrast, in the second category of protocols is divided into two distinct phases.
   - First phase: object is reliably propagated from the sink to the core nodes.
   - Second phase: core nodes disseminate the object to their neighboring non-core nodes in parallel.
   Ex: CORD, Sprinker
INTRODUCTION

• CORD (COre based Reliable Dissemination)
  – Construct a core for data dissemination.
  – Core node and non-core node.
  – Two-phase core-based approach.
  – Use coordinated sleep schedule to reduce energy consumption.

• CORD contribution
  – Combine sleep schedule with two-phase approach
  – In experiment, the energy consumption of CORD is 30–60% of that of Deluge.
RELATED WORK

• **Data Dissemination in Sensor Networks**
  – The protocols were developed for supporting network reprogramming in multi-hop networks
  – SPIN (Sensor Protocols for Information via Negotiation) for three-phase handshaking
  – Deluge and MNP compare MOAP
  – Deluge compare MNP

• **Connected Dominating Set (CDS)**
  – Subset of nodes in a network are selected as a backbone for routing, or as cluster-heads for data aggregation and forwarding.
  – CORD adapts Cheng’s single leader algorithm
RELATED WORK

SPIN-BC: A three-stage handshake protocol for broadcast media
CORD DESIGN

• **Core Construction**
  – Cheng’s algorithm
  – Link Quality
    • Two nodes are considered connected only when the link quality between them is above a threshold, $Q_{th}$.
    • Link Quality Indicator (LQI) as a metric of the link quality
  – **Establishing Coordinated Schedules**
    • modified Cheng’s algorithm to integrate core construction with the establishment of coordinated node schedules.
CORD DESIGN

Cheng’s algorithm
Cord Design

- Establishing Coordinated Schedules
  - The sink initiates core construction by starting the schedule and sending a CLAIM message.
  - Nodes that receive the CLAIM message update their effective degrees.
  - If a node has a good link, it selects sink as its parent and initiates its own repeating schedule. Then broadcast COMPETE message.
  - Node responds with a SUBSCRIBE message to the competitor.
  - A node that receives SUBSCRIBE messages become a core node, otherwise it becomes a non-core node.
Coordinated Node Sleep Scheduling

- In protocols, we use a pipelined data dissemination approach.
  - Nodes that transmit data simultaneously should be at least three hops apart to ensure that transfers of different pages do not interfere with each other.

![Diagram of pipelined data dissemination](image)

Fig. 1. Pipelined data dissemination
CORD DESIGN

- **Coordinated Node Sleep Scheduling**
  - We refer to these consecutive slots in a node’s schedule
    - P-slot: parent slot
    - C-slot: child slot
    - Q-Slot: quiescent slot
  - Different schedule in different nodes.
    - Core node: C–P–Q schedule
    - Non-core node: C–Q–Q schedule
CORD DESIGN

- **Coordinated Node Sleep Scheduling**
  - Each node synchronizes the boundaries of its time slots with its parent in the core at the time of core construction.
  - A node’s C-slot coincides with the P-slot of its parent in the core.
  - The sink marks its first slot as a P-slot. Nodes that receive advertisements or data from their parents assign the current slot to be a C-slot.
CORD DESIGN
Cord Design

- **Two-phase Data Dissemination**
  - In the first phase:
    - pages of the object are propagated through the core in a pipelined fashion.
    - The non-core nodes passively participate by listening to communications between core nodes.
  - In the second phase:
    - non-core nodes make requests to their local core nodes for missing data packets.
CORD using the nesC programming language on the TinyOS platform.

*Evaluation Metrics & Methodology*

- Use table I to compare the energy consumption.

### Table I

<table>
<thead>
<tr>
<th>Component</th>
<th>Current (mA)</th>
</tr>
</thead>
<tbody>
<tr>
<td>CPU current in active state</td>
<td>1.8</td>
</tr>
<tr>
<td>CPU current in sleep state</td>
<td>5.1</td>
</tr>
<tr>
<td>Radio current in receive state</td>
<td>23</td>
</tr>
<tr>
<td>Radio current in transmit state</td>
<td>21</td>
</tr>
<tr>
<td>Radio current in sleep state</td>
<td>1</td>
</tr>
<tr>
<td>External EEPROM current in write state</td>
<td>20</td>
</tr>
<tr>
<td>External EEPROM current in read state</td>
<td>4</td>
</tr>
<tr>
<td>External EEPROM current in sleep state</td>
<td>2</td>
</tr>
</tbody>
</table>

*TelosB Current Specification*
TESTBED DESCRIPTION AND RESULTS

- Indoor
- Outdoor

Fig. 3. Indoor TelosB network testbed including the core structure from one experiment (nodes in circles are core nodes)
PROTOCOL EVALUATION

- Testbed Description and Results

<table>
<thead>
<tr>
<th></th>
<th>Latency (sec)</th>
<th>Node Uptime (sec)</th>
<th>Number Packet Transmissions</th>
<th>Node Energy (mAh)</th>
</tr>
</thead>
<tbody>
<tr>
<td>CORD</td>
<td>243 ± 14.7</td>
<td>76.3 ± 5.55</td>
<td>261 ± 32.6</td>
<td>0.52</td>
</tr>
<tr>
<td>Deluge</td>
<td>226 ± 17.3</td>
<td>226 ± 17.3</td>
<td>331 ± 21.5</td>
<td>1.56</td>
</tr>
</tbody>
</table>

**TABLE II**

Average Object Delivery Latency and Energy expenditure per node for indoor experiments (confidence intervals are shown with 90% confidence level)
PROTOCOL EVALUATION

- Testbed Description and Results

<table>
<thead>
<tr>
<th></th>
<th>Latency (sec)</th>
<th>Node Uptime (sec)</th>
<th>Number Packet Transmissions</th>
<th>Node Energy (mAh)</th>
</tr>
</thead>
<tbody>
<tr>
<td>CORD</td>
<td>301 ± 10.0</td>
<td>95.9 ± 5.56</td>
<td>398 ± 78.7</td>
<td>0.66</td>
</tr>
<tr>
<td>Deluge</td>
<td>313 ± 10.1</td>
<td>313 ± 10.1</td>
<td>483 ± 5.91</td>
<td>2.15</td>
</tr>
</tbody>
</table>

**TABLE III**

*Average Object Delivery Latency and Energy expenditure per node for outdoor experiments (confidence intervals are shown with 90% confidence level)*
PROTOCOL EVALUATION

Fig. 8. Individual node uptime for CORD in one experiment on outdoor 3x11 TelosB network
Simulation Results

- grid network is denoted by $m \times n - s$

<table>
<thead>
<tr>
<th>Network topology</th>
<th>5x20 Grid</th>
</tr>
</thead>
<tbody>
<tr>
<td>Spacing</td>
<td>9 meters</td>
</tr>
<tr>
<td>Transmission power level</td>
<td>medium (0dBm)</td>
</tr>
<tr>
<td>Object size</td>
<td>10 pages</td>
</tr>
<tr>
<td>Page size ($K$)</td>
<td>128 packets</td>
</tr>
<tr>
<td>Packet payload size</td>
<td>23 bytes</td>
</tr>
<tr>
<td>Slot length ($L$)</td>
<td>6 seconds</td>
</tr>
</tbody>
</table>

**TABLE IV**

**DEFAULT PARAMETER SETTINGS FOR THE SIMULATION EXPERIMENTS**
PROTOCOL EVALUATION

- Simulation Results
  - Effect of Network Size:
Simulation Results

Effect of Data Object Size

- Fig. 15. Latency for various object sizes (5x20-9 network)

- Fig. 16. Energy consumption for various object sizes (5x20-9 network)
Simulation Results
  Effect of Network Density

Fig. 17. Latency for various network densities (5x20 network)

Fig. 18. Energy consumption for various network densities (5x20 network)
CONCLUSIONS

- CORD differs from previously proposed protocols in its aggressive use of sleep scheduling in conjunction with a two-phase core-based pipelined object propagation approach.
- The energy consumption for large object dissemination in CORD is 30%–60% of that of Deluge.