CS649
Sensor Networks
Lecture 27: Reliable Broadcast

Andreas Terzis
http://hinrg.cs.jhu.edu/wn06/

Thanks to Jonathan Hui
Background

- Retasking is essential
  - Often learn about the environment after deployment (sensing data, network characteristics, etc.)
- Retasking over the network is crucial
  - Embedded nature of sensor networks
  - Network scales reaching thousands of nodes
  - A necessity in debugging and testing cycle
Methods for retasking

<table>
<thead>
<tr>
<th>Method</th>
<th>Flexibility</th>
<th>Cost</th>
<th>Frequency</th>
</tr>
</thead>
<tbody>
<tr>
<td>Configuration Parameters</td>
<td>Low</td>
<td>Low</td>
<td>High</td>
</tr>
<tr>
<td>Scripts/ByteCode</td>
<td>Med</td>
<td>Med</td>
<td>Med</td>
</tr>
<tr>
<td><strong>New Binary</strong></td>
<td><strong>Very High</strong></td>
<td><strong>Very High</strong></td>
<td><strong>Very Low</strong></td>
</tr>
</tbody>
</table>

Consider retasking using a new binary
Challenges in network reprogramming

- **Goal**
  - Reliably disseminate large objects (i.e. size >> RAM) over a multi-hop sensor network from few to many nodes.

- **Issues**
  - Constrained storage hierarchy
    - Packet (32 bytes) << RAM (4K) << program (10K) < external flash (512K)
  - 100% reliability
  - Rapid propagation
  - Eventual consistency
  - Scalability (network size and density)
Deluge Protocol Overview

- A General Protocol for Bulk Data Dissemination

- State-machine with strictly local rules
  - Nodes advertise, request data, and broadcast
Deluge Protocol Overview

- A General Protocol for Bulk Data Dissemination

- State-machine with strictly local rules
  - Nodes advertise, request data, and broadcast
Deluge Protocol Overview

- A General Protocol for Bulk Data Dissemination

- State-machine with strictly local rules
  - Nodes advertise, request data, and broadcast
Deluge Protocol Overview

- A General Protocol for Bulk Data Dissemination

- State-machine with strictly local rules
  - Nodes advertise, request data, and broadcast
Deluge Protocol Overview

- A General Protocol for Bulk Data Dissemination

- State-machine with strictly local rules
  - Nodes advertise, request data, and broadcast
Deluge Protocol Overview

- A General Protocol for Bulk Data Dissemination

- State-machine with strictly local rules
  - Nodes advertise, request data, and broadcast
Properties

• Simple
• Storage hierarchy
• High reliability
• Spatial multiplexing
• Epidemic -> eventual consistency
• No neighbor state management
• Density-aware
• Robust to asymmetric links
Data Representation

- Monotone increasing version number

- Object divided into contiguous pages, each consisting of $N$ packets.
  - CRC at page and packet level

- Reduced RAM requirements for maintaining state about which packets are needed

- Allows for spatial multiplexing
Spatial Multiplexing

- Propagate in “waves”
- Exploit limited range of radio to allow for concurrent broadcasts.
- Reduced completion time \( o(d + S_{obj}) \) vs. \( o(d \times S_{obj}) \)
Maintain

- Advertise
  - Version and fraction of image complete
    - Nodes request pages in sequential order

- Use Trickle (Levis et. al., NSDI’04)
  - Manage advertisement broadcasts
  - Uses suppression to decrease advertisement rate as neighbors increase
  - *Bounds advertisement rate independent of node density.*
Maintain

- Advertise
  - Version and fraction of image complete
    - Nodes request pages in sequential order
  - Use Trickle (Levis et. al., NSDI’04)
    - Bounds advertisement rate independent of node density.

- Transition to:
  - Transmit on receiving a request
  - Request on receiving an advertisement with newer data (e.g. from a node with a larger fraction of the complete image)
    - Unless a request or data packet was recently overheard
    - Exploit Trickle suppression to minimize set of senders
Request

• Transmit a request
  – After random backoff
  – Suppress if
    • any similar requests or data packets are overheard during backoff period
  – Minimize senders by unicasting requests to the node that advertised

• Transition to **Maintain**
  – After receiving all packets of a page
  – After $k$ requests to protect against asymmetric links
Transmit

- Transmit all requested packets
  - May receive requests when transmitting
- Round-robin schedule to provide fairness
- Transition to Maintain when all requested packets are transmitted
Experimental Methodology

- **Real-world deployment** (77 Mica2-dots indoors)
  - Propagation Time
  - Suppression Mechanisms
  - Received Data Redundancy

- **Simulation** (TOSSIM, up to 800 nodes, square grid)
  - Propagation Time while varying: Network Diameter, Object Size, and Density.

- **Parameters**
  - Page size: about 1KB/page
  - Advertisement rate: 0.5 msgs/second
Real-World Deployment
Real World: Completion Time

Overall Completion Time vs. Object Size

- No Pipeline (projected)
- Observed
- Optimal (projected)

One-ninth effective channel bandwidth (ECB)
Real World: Effect of Suppression

Transmitted Requests (20 page object)

Observed

Nodes (%)

Less than 10 requests transmitted for majority of nodes

No Suppression, No Packet Loss
Real World: Data Redundancy

Received Data Ratio
(20 page object)

Nodes (%)

Received Data Ratio (multiple of minimum required)

Less than 5 times the required data received for all nodes
Completion Time

- Varying network diameter
- Constant density & object size

Spatial multiplexing improves performance
Completion Time

- Varying object size
- Constant density & diameter

Time linear with object size
Completion Time

- Varying density
- Constant diameter & object size

Time increases with density
Propagation Dynamics

15 ft Spacing, 5 Pages

10 ft Spacing, 5 Pages
Hidden Terminal Problem

Node Near Edge

Node In Center

Spring 2006  CS 649