Efficient Sensor Networks for Smart Environments

Huang Lee
Wireless Sensor Networks Lab
Stanford University
Wireless Sensor Networks

- Wireless Sensor Networks for Smart Environment Applications
  - Fire alarm systems
  - Smart surveillance systems
  - Intelligent control systems

- Features
  - Battery-powered sensors
  - Hub (Base station)
    - Ac-powered
    - Computational power

- Operations
  - Event-driven operation
  - Time-driven operation
Event-Driven Operation: Motivation

- Sensing and actuation application
  - For example, fire alarm sprinkler systems
  - Detect and notify **rare but urgent** events
  - Receive **delay sensitive** actuation commands
Time-Driven Operation: Motivation

- Periodic data gathering
  - Nodes report data via the data aggregation tree
  - One of the most common operations
  - Take 30%-60% of energy consumption of a node
## Network Operations

<table>
<thead>
<tr>
<th></th>
<th>Event-driven</th>
<th>Time-driven</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Application</strong></td>
<td>Sensing and actuation</td>
<td>Periodic data gathering</td>
</tr>
<tr>
<td><strong>Topology</strong></td>
<td>Communication path</td>
<td>Data collection tree</td>
</tr>
<tr>
<td><strong>Direction</strong></td>
<td>Two-way traffic</td>
<td>One-way traffic</td>
</tr>
<tr>
<td><strong>Traffic</strong></td>
<td>Bursty</td>
<td>Periodic</td>
</tr>
<tr>
<td><strong>Feature</strong></td>
<td>Delay sensitive</td>
<td>Energy hungry</td>
</tr>
<tr>
<td><strong>Issue</strong></td>
<td>How to optimize the trade-off between the delay and power?</td>
<td>How to optimize the energy (lifetime) for data collection?</td>
</tr>
</tbody>
</table>
Outline

- Event-driven operation
  - Energy-efficient wake-up scheduling
  - Efficient algorithm to double the performance of delay and power trade-off

- Time-driven operation
  - Energy-efficient data aggregation
  - Near lifetime-optimal and practical algorithm

- Intelligent light control
  - Light control based on camera sensor observations
Network Topology

- Stationary
- Synchronized
- Dense network
- Levels (L1 – L4) are assigned in breadth first order
Outline

- Event-driven operation
- Time-driven operation
- Intelligent light control
Outline

- Event-driven operation
  - Wake-up scheduling
- Time-driven operation
- Intelligent light control
Wake-up Scheduling

- Receiver: Wakeup periodically to check for any activity
  - Sample RSSI (received signal power)
  - If RSSI > Threshold stay in active mode to receive the message
  - Otherwise switch to low-power mode
- Sender: Awaken the receiver and send the message
  - Send a tone (at correct time) to awaken receiver
  - Send the message after the receiver is awakened

**Trade-off between the delay and power**
Traffic Model

- Two communication paths
  - Forward direction (downlink)
    - Commands/Queries
  - Backward direction (uplink)
    - Event notifications, Alarms

- Messages arrive at random time asynchronously

- Delay: time between generation of a message till its delivery at the destination
  - Worst delay depends on the maximum number of hops
Wakeup Patterns – Fully Synchronized

- Delay due to wakeup over multiple hops
- Simple but worst performance!

[ W. Ye, J. Heidmann, and D. Estrin 2002 ]
Wakeup Patterns – Ladder pattern

- Forward delay is significantly improved
- Backward delay is not so good

[ G. Lu, B. Krishnamachari, and C. Raghavendra 2004 ]
Delay and Power Trade-off Curve

- How to improve the performance?

Performance bound (e.g., Ladder pattern)
Multi-Parent Method – Basic Idea

**Parent Wakeup Pattern**
- Parent 1 Wakeup Pattern
- Parent 2 Wakeup Pattern
- What the child sees

**Single-parent Case**
- Base Station
- Parent
- Child

**Multi-parent Case**
- Parent 1 - Mother
- Parent 2 - Father
- Child
Delay and Power Trade-off Curve

- Most of the benefit is from one parent to two parents
- How to assign the (two) parents?
Graph Coloring Problem

- Color all nodes in the network either red or blue

- Each node has at least one red parent and one blue parent

- Balance the number of red and blue parents of each node
  - Have balanced number of parents to select
  - Easier to solve
Graph Coloring Algorithm – Formulation

- Define a variable $c_i$ for each node $i$

$$b_6 = c_1 + c_2 + c_3$$
$$b_7 = c_2 + c_3 + c_4 + c_5$$

$$c_i = \begin{cases} 
+1 & \text{node } i \text{ blue} \\
-1 & \text{node } i \text{ red} 
\end{cases}$$

$$b_i = \sum_{j=1}^{N} p_{ij} c_j$$

- $p_{ij} = 1$ if node $j$ is a parent of node $i$, otherwise it is 0!

- If the parent colors of node $i$ are balanced, $b_i$ will be close to 0

Event-Driven Time-Driven Light Control
Graph Coloring Algorithm

- Minimize the weighted sum of square of $b_i$s

\[
\begin{align*}
\text{minimize} \quad & \sum_{i=1}^{N'} w_i b_i^r = c^T P^T WPc \\
\text{subject to} \quad & c_i^r = \backslash, i = \backslash, \ldots, N
\end{align*}
\]

- Heuristic:
  - Approximate the problem with relaxation
  - Refine the solution based on the value of $c_i$
  - Assign weights to node and adaptively change the weights
  - Move the node to higher levels (e.g., level 2 $\to$ level 3)

![Graph Coloring Diagram]
Graph Coloring Algorithm - Results

![Graph Coloring Algorithm Results](image-url)
Event-Driven Operation Summary

- Wake-up is an energy consuming process for many delay-sensitive applications

- Multi-parent method
  - Significantly improves the performance
  - Partitioning of the nodes (NP-complete)

- Heuristic coloring algorithm
  - Efficiently assigning the colors (at network initialization)

- Joint design of MAC and topology
Outline

- Event-driven operation
- Time-driven operation
  - Periodic data aggregation
- Intelligent light control
Data Aggregation Tree

- Build tree for data aggregation
  - The energy cost (load) depends on the number of children
  - Balance the load among nodes

Different trees gives different performance

- Data centric routing
  [B. Krishnamachari, D. Estrin, and S. Wicker 2002]

- Maximize network lifetime
  [Y. Wu, S. Fahmy, and N. B. Shroff 2008]
Proposed Data Aggregation Trees

- Balance the load over time
  - Example (left node): First cycle 2 children, second and third cycles 1 child, last three cycles no child
  - On average $= \frac{2+1+1}{6} = \frac{2}{3}$ child

- How to construct the set of trees to maximize the lifetime?
Data Aggregation Models

• Before we talk about tree construction…

✓ All data can be combined
  • For example, network supervision
    ➢ An algorithm using a set of trees is proposed


✓ No data can be combined
  ➢ An algorithm based on DAG (Directed Acyclic Graph) structures is proposed

Data Aggregation and Network Models

- Nodes are divided into groups
- Relay Nodes
  - May be used if it's more energy-efficient
- Example
  - Base Stations: A, B
  - Relay node: D
  - Source nodes in 3 groups: (C,G,H), (E,F,I), (J)

✓ Only data from same group of nodes are combined
  - Most general (e.g., temperature in each room)
  - Considered in the tree construction (routing)
Flow Calculation - Principles

- One unit of data from each sensor
- Flow conservation
  - All the data should be routed to the base stations
- Data fusion
  - Relay B: \(0.6 < 0.2 \text{ (C)} + 0.6 \text{ (D)}\)
  - For the flows from the same group, only the “maximum flow” has impact on energy consumption: \(\text{max}(0.2, 0.6)\)
  - Effective flows \(x_{ij}\) on each link: Sum of the “maximum flow” from each group
Flow Calculation - Formulation

- Formulated as a convex optimization problem
- Optimize the network lifetime

\[
\text{minimize} \quad q
\]
\[
\text{subject to} \quad \sum_{j \in R_n} f_{nj}^n = 1, \quad \forall n \in S,
\]
\[
\sum_{j \in S_n} f_{jn}^n = 0, \quad \forall n \in S,
\]
\[
\sum_{j \in R_i} f_{ij}^n = \sum_{j \in S_i} f_{ji}^n, \quad \forall n \in S, \quad \forall i \in \mathcal{N} \setminus \{n\},
\]
\[
x_{ij} \geq \sum_{k=1}^{g} \max_{n \in G_k} (f_{ij}^n), \quad \forall n \in S, \quad \forall i \in \mathcal{N}, \quad \forall j \in R_i,
\]
\[
\sum_{j \in R_i} e_{ij}^t x_{ij} + \sum_{j \in S_i} e_{ji}^r x_{ji} \leq q, \quad \forall i \in \mathcal{N},
\]
\[
f_{ij}^n \geq 0, \quad \forall n \in S, \quad \forall i \in \mathcal{N}, \quad \forall j \in R_i,
\]

Flow equations

Energy equation

- $e_{ij}^t : tx$ energy
- $e_{ij}^r : rx$ energy
Flow Calculation

- We calculate the optimal flows and their directions on all the links
- Flows from different group are jointly optimized
Multi-tree Construction

- In each cycle, we construct one multi-tree
- The energy consumption of nodes depending on the TX/RX pairs
  - Node G and H exchange roles in different cycles

Flow Calculation → Multi-tree Construction → Scheduling

<table>
<thead>
<tr>
<th>Multi-tree 1 in cycle 1</th>
<th>Multi-tree 2 in cycle 2</th>
</tr>
</thead>
</table>

Base Stations

Event-Driven, Time-Driven, Light Control
Goal: make the average link usage in all cycles close to the optimal flow
- Construct multi-trees one by one (iteration)
  - Given one multi-tree, how to construct the second multi-tree?

Optimal flows  First multi-tree  Weights  Costs  Second multi-tree

- Event-Driven
- Time-Driven
- Light Control
Multi-tree Construction – Find the Weights

- Given \( m - 1 \) multi-trees, find the weights for the \( m^{th} \) multi-tree
  - Count the number of use of each link for all the \( m - 1 \) multi-trees
  - Find the weight via convex optimization formulation
  - Change the weight to cost for each link (e.g., \( 1 - w_{ij} \))

\[
\begin{align*}
\text{minimize} & \quad \sum_{k=1}^{g} \sum_{\forall j \in R_i, \forall i \in \mathcal{N}} \left\| \frac{1}{m} \left( w_{ij} (k) + \sum_{p=1}^{m-1} z_{ij} (k, p) \right) - \max_{n \in G_k} f_{ij}^n \right\|_2^2 \\
& \quad + \sum_{\forall j \in R_i, \forall i \in \mathcal{N}} \left\| \sum_{k=1}^{g} \frac{1}{m} \left( w_{ij} (k) + \sum_{p=1}^{m-1} z_{ij} (k, p) \right) - \hat{x}_{ij} \right\|_2^2 \\
\text{subject to} & \quad \sum_{j \in R_i} w_{ij} (k) = 1, \; k = 1, \ldots, g, \; \forall i \in \mathcal{S} \cap G_k, \\
& \quad \sum_{j \in R_i} w_{ij} (k) = \sum_{j \in S_i} w_{ji} (k), \; k = 1, \ldots, g, \; \forall i \in \mathcal{N} \setminus (\mathcal{S} \cap G_k), \\
& \quad w_{ij} (k) + w_{ji} (k) \leq 1, \; j \in R_i \text{ and } i \in R_j, \; k = 1, \ldots, g, \\
& \quad 0 \leq w_{ij} (k) \leq 1, \; \forall i \in \mathcal{N}, \forall j \in R_i, \; k = 1, \ldots, g,
\end{align*}
\]
Multi-tree Construction – Algorithm

- Given the **cost** on each link, we construct multi-tree
- Each multi-tree contains several Steiner trees
  - A tree only connects some nodes with minimum cost
- The Takahashi and Matsuyama Algorithm

### Flow Calculation → Multi-tree Construction → Scheduling

- **Base station:** A
- **Relay nodes:** B, D
- **Source nodes:** C, E, F

- **Step 1**
- **Step 2**
- **Step 3**
- **Step 4**
Greedy algorithm is used to calculate a schedule with no collision and with multiple retries over different frequencies.
Performance Analysis

Number of nodes = 45, 10% relay nodes, 3 groups

- Ratio of the real lifetime to the theoretical optimal value
- Multi-tree structure gives much better performance
Performance Analysis

Number of nodes = 40, No relay node, All data are combine

- Obtaining near-optimal performance with around 35 multi-trees
Performance Analysis

- Obtaining near-optimal performance with around 35 multi-trees
Performance Analysis

Number of nodes = 45, 10% relay nodes, 3 groups

- NP-complete problem solved by heuristic algorithm
- Obtain 5%-to-optimal performance with around 100 multi-trees
Time-driven Operation Summary

- Data gathering is one of the most common tasks
  - Very energy-consuming

- Near energy-optimal and practical data fusion algorithm
  - Data aggregation models
  - Flow calculation / Multi-tree construction / Scheduling

- By changing the aggregation trees over time, we can balance the traffic load and achieve near-optimal lifetime

- Joint design of MAC and routing
Outline

- Event-driven operation
- Time-driven operation
- Intelligent light control
Light Control Scenario

- Wireless nodes are deployed over a floor
- Light control based on camera observations

- Goal:
  - Comfort light setting
  - Energy efficiency

- Camera sensors:
  - Rich information
  - Multiple functions

- Only data from the same room can be combined
Light Control System

**Camera nodes**
- Making observations
- Local processing (Region of interest)

**Base stations**
- Occupancy sensing
- Activity analysis
- Optimization

**Actuators**
- Light control

- Camera sensors regularly send processed observations to the base stations
- Base stations send commands to controllers when the light setting has to be changed

Event-Driven | Time-Driven | Light Control
---|---|---

Data Collection

Wake-up Scheduling

Time
Vision Analysis

- Occupancy Sensing

- Activity Analysis
  - Walking
  - Sitting
  - Lying

- Base station obtains the user location and activity

[A. Ercan, A. El Gamal, and L. Guibas 2003]
[D. Yang, H. Gonzalez-Banos, and L. Guibas 2003]
[W. Chen and H. Aghajan 2009]
Optimization Formulation

- Energy vs. Satisfaction
- Utility Functions
  - Walking, sitting, lying
  - Obtained from learning
    [V. Singhvi, C. E. Dept, J. H. Garrett 2005]
- Intensity control
- Switch control
Simulations: Intensity Control

Event-Driven
Time-Driven
Light Control
Simulations: Switch Control
Light Control Summary

- A camera sensor network for light control

- Operations
  - Data collection cycle
  - Wake-up scheduling cycle

- Energy and satisfaction Trade-off
  - Utility functions
  - Intensity and switch control

- Simulations with artificial lights
Conclusions

- Wireless sensor networks for smart environment
- Event-driven operation
  - Multi-parent method in wake-up scheduling
  - Improve the energy and delay trade-off
  - Efficient graph coloring algorithm
- Time-driven operation
  - Data aggregation models
  - Near lifetime-optimal and practical data aggregation
  - Dynamic data aggregation trees
- Light control using camera sensors