Survey on Data Dissemination to Mobile sinks in Wireless Sensor Networks

Presented by S.C.Chuang, MNET
2005/08/25
Outline

- Introduction
- Different Approaches of Data Dissemination
  - TTDD
  - SEAD
- Issues Discussion
- References
Introduction

- Wireless sensor networks
  - a large number of sensors (stationary)
  - a few data collectors, named “sinks”
  - Restriction: energy
  - sources have to build and maintain multihop routes toward sinks to report sensory data
Introduction
TTDD
A Two-tier Data Dissemination Model

Mobicom 2002
TTDD - basic idea

Assumption:
Stimulus/sink: mobile, multiple
Sensor node: stationary, with GPS
TTDD - basic idea

TTDD Protocol:
1. Grid construction
2. Two-Tier query and data forwarding
3. Grid maintenance
TTDD - Grid construction
TTDD - Grid construction

- Source send "Data Announcement" using greedy geographical forwarding (Source location, Lp, Grid lifetime)
- "Data Announcement" will stop at a node that is closer to Lp than all its neighbors
TTDD - Grid construction
TTDD - Grid construction
TTDD - Two-Tier query and data forwarding

Immediate dissemination node

Primary Agent
TTDD - Two-Tier query and data forwarding
TTDD - Two-Tier query and data forwarding
TTDD - Two-Tier query and data forwarding
TTDD - Two-Tier query and data forwarding
TTDD - Two-Tier query and data forwarding
TTDD - Two-Tier query and data forwarding
TTDD - Two-Tier query and data forwarding
TTDD - Grid maintenance

Two way to maintain the grid

- “Grid lifetime” in data announcement
- Dissemination node fails
  - “upstream information duplication” mechanism
TTDD - Grid maintenance
TTDD - Grid maintenance
Advantage
- Construct & maintain a grid structure with low overhead
- Can effectively deliver data from multiple sources to multiple mobile sinks

Disadvantage
- Each source must constructs & maintains a grid structure
- Is it suitable for “moving target”?
SEAD
a Scalable Energy-efficient Asynchronous Dissemination protocol

SEAD
a Scalable Energy-efficient Asynchronous Dissemination protocol

- When sinks are mobile in sensor networks
  - Communication consists of three main parts
    - Building the dissemination tree (d-tree)
    - Disseminating data
    - Maintaining linkage to mobile sinks
SEAD
a Scalable Energy-efficient Asynchronous Dissemination protocol
SEAD
a Scalable Energy-efficient Asynchronous Dissemination protocol

Overview of the algorithm
- One source generates the sensory update traffic possibly on behalf of a group of local sensors
- The data updates are disseminated along a tree to the mobile sinks in an asynchronous manner
- Each branch of the tree may have its own update rate
  - depending on the desired refresh rate of the downstream observers
SEAD
a Scalable Energy-efficient Asynchronous Dissemination protocol

- When a mobile sink wants to join the d-tree
  - It selects one of its neighboring sensor nodes to send a *join query* to the source of the tree
  - The selected node is called the sink’s *access node*

- The access node
  - is used to represent the moving sink when the optimal d-tree is built
  - Amortize the overhead in the presence of mobility
SEAD
a Scalable Energy-efficient Asynchronous Dissemination protocol

- The tree delivers data to the fixed access node. In turn, the access node delivers the data to the sink without exporting the sink’s location information to the rest of the tree.
- The tree is updated only when the access node changes.
- As the sink moves, no new access node is chosen until the hop count between the access node and the sink exceeds a threshold.
  - Trade-off between path delay and energy spent on reconstructing the tree.
SEAD
a Scalable Energy-efficient Asynchronous Dissemination protocol

- Source data is replicated at selected nodes between the source and sinks.
- We define a *replica* as a sensor node that stores a copy of the source data.
  - It temporarily stores the latest data incoming from the source and asynchronously disseminates it to others along the tree.
SEAD
a Scalable Energy-efficient Asynchronous Dissemination protocol

- **Subscription Query**
  - Sink directs a join query to source via its access node

- **Gate replica search**
  - A gate replica is determined, which serves as the grafting point from which a branch to the new access point is extended

- **Replica placement**
  - Locally readjusts the tree in the neighborhood of the gate replica to further reduce communication energy

- **D-tree management**
  - The constructed tree is managed to accommodate mobile sinks or defective regions such as a group of congested or failed nodes
Mobile sinks beacon periodically to determine their neighbors

A mobile sink $B_i$ selects the nearest of its adjacent nodes as the access node $A_i$
- $B_i$ sends a join query to a source via $A_i$
- The join query message contains the location of the $A_i$
- and the sink’s desired updated rate $R_i$

The access node directly sends the join query to the source via the underlying routing protocol
SEAD - Gate Replica Search

- Communication cost

\[ \text{Energy cost}(a, b) \propto d(a, b) P_{ab}. \]

\text{Pab} : \text{the packet sending rate}
\[ K(r) - K(c) = \]
\[ R_d(d(r, A_t)) + \sum_{m \in E_r} ||R_t - Q_m^{p(m)}|| d(p(m), m) \]
\[ - R_d(d(c, A_t)) - \sum_{m \in E_c} ||R_t - Q_m^{p(m)}|| d(p(m), m) \]
\[ = R_d(d(r, A_t)) - R_d(d(c, A_t)) - ||R_t - Q^{p(c)}_t|| d(p(c), c) \]
\[ = R_d(d(r, A_t)) - R_d(d(c, A_t)) - ||R_t - Q^{r}_c|| d(r, c) \]
Locally adjusts the tree around the gate replica to produce an optimal d-tree.

There are two ways to connect the access node to the gate replica:

- **Non-replica mode**
  - Connect it as a child of the gate replica.
  - Adds no replicas to the tree.

- **Junction mode**
  - Create a child for the gate replica to feed the access node and some of the gate replica’s original children.
  - The new child replica is called a junction replica.
The replica placement phase
- compares the non-replica mode cost $U_{\text{non_replica}}$ to a junction replica mode cost $U_{\text{jreplica}}$
- Selects the better option so that the access node joins the tree in a way that minimizes the energy cost
The gate replica $g$ calculates the cost of the non-replica mode $U_{\text{non-replica}}(c)$ for each child $c \in C(g)$.
The gate replica $g$ finds the neighbor node $n$ among its adjacent nodes within a single hop range. Then it calculates the energy cost $U_{ireplica}(c)$ for each child $c$.

$$U_{ireplica}(c) = \min_{n \in W} \left\{ d(g, n) \max(R_t, Q_c^g) + d(n, A_t)R_t + d(n, c)Q_c^g \right\}$$
SEAD - D-tree management

- Sink mobility

(a) initial access node setup
SEAD - D-tree management

(b) path A to B setup
SEAD - D-tree management

(c) path B to C setup
(d) case 1: shorter path setup
SEAD - D-tree management

(e) case 2: new access node create
Evaluation

- Energy consumed for data packets
Evaluation

- Average delay
Evaluation - Sink mobility

- Energy consumption for the number of sinks

![Graph showing energy consumption for different numbers of sinks in a network with 400 nodes, 2000m x 2000m area, and speed 10m/sec. The graph plots average communication energy (J/node) against the number of sinks.]
Evaluation - Sink mobility

- Energy consumption with different sink speeds
Evaluation - End-to-end delay

- Average delay with different number of sinks
Evaluation - End-to-end delay

- Average delay with different sink speeds
Advantage

- Saves energy consumption in both building the d-tree and maintaining linkage to mobile sinks
- Strikes a balance between end-to-end delay and power consumption that favors power savings over delay minimization

Disadvantage

How do access nodes know where the sources is
THANK YOU