Routing Techniques in Wireless Sensor Networks

Dr. Ayşegül Tüysüz Erman
aysegul.erman@isikun.edu.tr

Computer Science and Engineering
Işık University
2012
Outline

- Research Goal
- Routing challenges and design issues
- Classification of routing protocols
- Routing Protocols in WSNs
- Conclusions
Research Goal

• The research goal of routing protocols is
  – To develop energy efficient routing strategies to transfer sensor data from nodes to sinks for the purpose of maximizing the lifetime of WSNs. → General purpose!
  – Different applications may have different requirements
Data Routing Methods

- Application-specific
- Time-driven: Periodic monitoring
- Event-driven: Respond to sudden changes
- Query-driven: Respond to queries
- Hybrid
Routing challenges and design issues

- Node deployment (random, manual)
- Node/link heterogeneity
- Connectivity
- Fault tolerance
  - Some sensors may fail due to lack of power, physical damage, or environmental interference
  - Adjust transmission power, change sensing rate, reroute packets through regions with more power
- Network dynamics
  - Mobile node
  - Mobile events, e.g., target tracking
Classification of routing protocols(1)

- Number of path
  - Uni-path and Multi-path
- Network topology
  - Flat and hierarchical
- Relationship between routing establishment and data transmission
  - Proactive, reactive and hybrid
- Geographical position information in nodes
  - Location-based and not location-based
- Using data style to identify nodes
  - Data-based and not data-based
Classification of routing protocols(2)

- Addressing nodes
  - Address-based and not address-based
- QoS guarantee
  - QoS-Supported not QoS-Supported
- Aggregating during data transmission
  - Data aggregation and no data aggregation
- Designating routing by source nodes
  - Source and no source
- Relationship between routing establishment and query action
  - Query-driven and not query-driven
Routing Protocols in WSNs

- I. Flat
- II. Hierarchical
- III. Location-based
  - Routing to mobile sink
I. Flat routing
Flooding

- Sends a copy of the data to all the neighbors
- Whenever a node receives some data, it sends that data to all its neighbors.
- Lots of redundancy
- 100% Coverage
• Flooding/Gossiping
  - Too much waste
  - Implosion & Overlap
  - Use in a limited scope, if necessary

• Data-centric routing
  - No globally unique ID
  - Naming based on data attributes
  - SPIN, Directed diffusion, ...
SPIN (Sensor Protocols for Information via Negotiation)

Fig. 3. SPIN protocol. Node A starts by advertising its data to node B (a). Node B responds by sending a request to node A (b). After receiving the requested data (c), node B then sends out advertisements to its neighbors (d), who in turn send requests back to B (e–f).
SPIN

• Pros
  – Each node only needs to know its one-hop neighbors
  – Significantly reduce energy consumption compared to flooding

• Cons
  – Data advertisement cannot guarantee the delivery of data
    • If the node interested in the data are far from the source, data will not be delivered
    • Not good for applications requiring reliable data delivery, e.g., intrusion detection
Direct Diffusion: Motivation

• Properties of Sensor Networks
  – Data centric
  – No central authority
  – Resource constrained
  – Nodes are tied to physical locations
  – Nodes may not know the topology
  – Nodes are generally stationary
Directed Diffusion: Main Features

• Data centric
  – Individual nodes are unimportant
• Request driven
  – Sinks place requests as interests
  – Sources satisfying the interest can be found
  – Intermediate nodes route data toward sinks
• Localized repair and reinforcement
• Multi-path delivery for multiple sources, sinks, and queries
Directed Diffusion: Motivating Example

- Sensor nodes are monitoring animals
- Users are interested in receiving data for all 4-legged creatures seen in a rectangle
- Users specify the data rate
Directed Diffusion: Interest and Event Naming

- **Query/interest:**
  - Type=four-legged animal
  - Interval=20ms (event data rate)
  - Duration=10 seconds (time to cache)
  - Rect=[-100, 100, 200, 400]

- **Reply:**
  - Type=four-legged animal
  - Instance = elephant
  - Location = [125, 220]
  - Intensity = 0.6
  - Confidence = 0.85
  - Timestamp = 01:20:40

- **Attribute-Value pairs, no advanced naming scheme**
Directed Diffusion: Interest Propagation

- Flood interest
- Constrained or Directional flooding based on location is possible
- Directional propagation based on previously cached data
Directed Diffusion: Data Propagation

- Multipath routing
  - Consider each gradient’s link quality
Directed Diffusion: Reinforcement

- Reinforce one of the neighbor after receiving initial data.
  - Neighbor who consistently performs better than others
  - Neighbor from whom most events received
Directed Diffusion: Negative Reinforcement

- Explicitly degrade the path by re-sending *interest* with lower data rate.
- Time out: Without periodic reinforcement, a gradient will be torn down.
Directed Diffusion: Summary of the protocol

(a) Interest propagation
(b) Initial gradients set up
(c) Data delivery along reinforced path
Directed Diffusion: Pros & Cons

• Different from SPIN in terms of on-demand data querying mechanism
  – Sink floods interests only if necessary
    • A lot of energy savings
  – In SPIN, sensors advertise the availability of data
• Pros
  – Data centric: All communications are neighbor to neighbor with no need for a node addressing mechanism
  – Each node can do aggregation & caching
• Cons
  – On-demand, query-driven: Inappropriate for applications requiring continuous data delivery, e.g., environmental monitoring
  – Attribute-based naming scheme is application dependent
    • For each application it should be defined a priori
    • Extra processing overhead at sensor nodes
Extension of Directed Diffusion*

• One-phase pull
  – Propagate interest
  – A receiving node pick the link that delivered the interest first
  – Assumes the link bidirectionality

• Push diffusion
  – Sink does not flood interest
  – Source detecting events disseminate exploratory data across the network
  – Sink having corresponding interest reinforces one of the paths
Rumor Routing

• Variation of directed diffusion
  – Don’t flood interests (or queries)
  – Flood events when the number of events is small but the number of queries large
  – Route the query to the nodes that have observed a particular event
  – Long-lived packets, called agents, flood events through the network
  – When a node detects an event, it adds the event to its events table, and generates an agent
  – Agents travel the network to propagate info about local events
    • An agent is associated with TTL (Time-To-Live)
Rumor Routing

- When a node generates a query, a node knowing the route to a corresponding event can respond by looking up its events table
  - No need for query flooding
  - Only one path between the source and sink
  - Rumor routing works well only when the number of events is small
  - Cost of maintaining a large number of agents and large event tables will be prohibitive
  - Heuristic for defining the route of an event agent highly affects the performance of next-hop selection
Gradient-Based Routing (GBR)

* 

- Variation of directed diffusion
- Each node memorizes the number of hops when the interest is diffused
- Each node computes its height, i.e., the minimum number of hops to BS
- Difference between a node’s height and its neighbor’s is the gradient on the link
- Forward a packet on a link with the largest gradient
- Data aggregation
  - When multiple paths pass through a node, the node can combine data
- Traffic spreading
  - Uniformly divide traffic over the network to increase network lifetime
    - Stochastic scheme: Randomly pick a gradient when two or more next hops have the same gradient
    - Energy-based scheme: A node increases its height when its energy drops below a certain threshold
    - Stream-based scheme: New streams are not routed through nodes that are part of the path for other streams
- Outperforms directed diffusion in terms of total energy
II. Hierarchical Routing
LEACH (Low Energy Clustering Hierarchy)

- Cluster-based protocol
- Each node randomly decides to become a cluster head (CH)
- CH chooses the code to be used in its cluster
  - CDMA between clusters
- CH broadcasts Adv; Each node decides to which cluster it belongs based on the received signal strength of Adv
- CH creates a transmission schedule for TDMA in the cluster
- Nodes can sleep when it’s not their turn to transmit
- CH compresses data received from the nodes in the cluster and sends the aggregated data to BS
- CH is rotated randomly
LEACH

- Pros
  - Distributed, no global knowledge required
  - Energy saving due to aggregation by CHs

- Shortcomings
  - LEACH assumes all nodes can transmit with enough power to reach BS if necessary (e.g., elected as CHs)
  - Each node should support both TDMA & CDMA

- Extension of LEACH
  - High level negotiation, similar to SPIN
  - Only data providing new info is transmitted to BS
# Comparison between SPIN, LEACH & Directed Diffusion

<table>
<thead>
<tr>
<th></th>
<th>SPIN</th>
<th>LEACH</th>
<th>Directed Diffusion</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Optimal Route</strong></td>
<td>No</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td><strong>Network Lifetime</strong></td>
<td>Good</td>
<td>Very good</td>
<td>Good</td>
</tr>
<tr>
<td><strong>Resource Awareness</strong></td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td><strong>Use of meta-data</strong></td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
</tr>
</tbody>
</table>
TEEN (Threshold sensitive Energy Efficient Network protocol)

• Reactive, event-driven protocol for time-critical applications
  – A node senses the environment continuously, but turns radio on and xmit only if the sensor value changes drastically
  – No periodic xmission
    • Don’t wait until the next period to xmit critical data
    • Save energy if data is not critical

• CH sends its members a hard & a soft threshold
  – Hard threshold: A member only sends data to CH only if data values are in the range of interest
  – Soft threshold: A member only sends data if its value changes by at least the soft threshold
  – Every node in a cluster takes turns to become the CH for a time interval called cluster period

• Hierarchical clustering
Multi-level hierarchical clustering in TEEN & APTEEN

Fig. 8. Hierarchical clustering in TEEN and APTEEN.
TEEN

• Good for time-critical applications
• Energy saving
  – Less energy than proactive approaches
  – Soft threshold can be adapted
  – Hard threshold could also be adapted depending on applications
• Inappropriate for periodic monitoring, e.g., habitat monitoring 😞
• Ambiguity between packet loss and unimportant data (indicating no drastic change)
APTEEN (Adaptive Threshold sensitive Energy Efficient Network protocol) *

- Extends TEEN to support both periodic sensing & reacting to time critical events
- Unlike TEEN, a node must sample & transmit a data if it has not sent data for a time period equal to CT (count time) specified by CH
- Compared to LEACH, TEEN & APTEEN consumes less energy (TEEN consumes the least)
  - Network lifetime: TEEN ≥ APTEEN ≥ LEACH
- Drawbacks of TEEN & APTEEN
  - Overhead & complexity of forming clusters in multiple levels and implementing threshold-based functions
III. Location-based routing protocols
Location based Routing

- Each sensor nodes knows its position/location
- Why?
- In a sensor network the information, and where the information is located, is more important than the node that sent the information.
- If the sensor nodes on the path of transmission die or lose their energy of transmission/receiving then the message can be sent via another intuitive (i.e. around a fire) route.
GPSR – Greedy Perimeter Stateless Routing

- It uses the positions of sensor nodes and a packet’s destination to make packet forwarding decisions.
- Greedy forwarding decisions are based on the position of the neighbors, the direction to which the packet has to be sent, and the location of the destination.
Greedy-Face-Greedy (GFG)

- Routes along the face of a planar sub-graph using the right-hand rule
  - Gabriel graph
  - the relative neighbor graph
GAF (Geographic Adaptive Fidelity)

- Energy-aware location-based protocol mainly designed for MANET
- Each node knows its location via GPS
  - Associate itself with a point in the virtual grid
  - Nodes associated with the same point on the grid are considered equivalent in terms of the cost of packet routing
  - Node 1 can reach any of nodes 2, 3 & 4 $\rightarrow$ 2, 3, 4 are equivalent; Any of the two can sleep without affecting routing fidelity

Fig. 11. Example of virtual grid in GAF.
GAF

- Three states
  - Discovery: Determine neighbors in a grid
  - Active
  - Sleep
- Each node in the grid estimates its time of leaving the grid and sends it to its neighbors
  - The sleeping neighbors adjust their sleeping time to keep the routing fidelity
GEAR (Geographic and Energy Aware Routing)

- Restrict the number of interest floods in directed diffusion
  - Consider only a certain region of the network rather than flooding the entire network
- Each node keeps an estimated cost & a learning cost of reaching the sink through its neighbors
- Estimated cost = f(residual energy, distance to the destination)
- Learned cost is propagated one hop back every time a packet reaches the sink
  - Route setup for the next packet can be adjusted
GEAR

• Phase 1: Forwarding packets towards the region
  – Forward a packet to the neighbor minimizing
    the cost function $f$
    • Forward data to the neighbor which is closest to the
      sink and has the highest level of remaining energy
  – If all neighbors are further than itself, there is a
    hole → Pick one of the neighbors based on the
    learned cost
GEAR

• Phase 2: Forwarding the packet within the target region
  − Apply either recursive forwarding
    • Divide the region into four subareas and send four copies of the packet
    • Repeat this until regions with only one node are left
  − Alternatively apply restricted flooding
    • Apply when the node density is low

• GEAR successfully delivers significantly more packets than GPSR (Greedy Perimeter Stateless Routing)
Ill-a. Routing to Mobile Sink
Rendezvous-based Protocols

- Sensor node
- Rendezvous node
- Sink node
- Source node

(a) (b) (c) (d) (e) (f)
TTDD (Two-Tier Data Dissemination)

Dissemination Node

Data Announcement

Immediate Dissemination Node

Sink

Query
TTDD

- Two-tier forwarding model
  - Source proactively builds a grid structure
  - **Localize** impact of sink mobility on data forwarding
  - A small set of sensor node maintains forwarding state
HexDD (Hexagonal Cell-based Data Dissemination)
## Summary

<table>
<thead>
<tr>
<th>Routing protocol</th>
<th>Data-centric</th>
<th>Hierarchical</th>
<th>Location-based</th>
<th>QoS</th>
<th>Network-flow</th>
<th>Data aggregation</th>
</tr>
</thead>
<tbody>
<tr>
<td>SPIN</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td>Directed Diffusion</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td>Rumor Routing</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td>Shah et al.</td>
<td>✓</td>
<td></td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>GBR</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td>CADR</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>COUGAR</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ACQUIRE</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Fe et al.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td>LEACH</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td>TEEN&amp;APTEEN</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td>PEGASIS</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Younis et al.</td>
<td>✓</td>
<td></td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Subramanian et al.</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MECN&amp;SMECN</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>GAF</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>GEAR</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Chang et al.</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Kalpakis et al.</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Akkaya et al.</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SAR</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td>SPEED</td>
<td></td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Conclusions

• There is no single, best routing protocol that outperforms all others or suitable for all applications

• Application specific needs are important in the design space
  - Energy-efficiency
  - Latency
  - Realiability (~ packet delivery ratio)