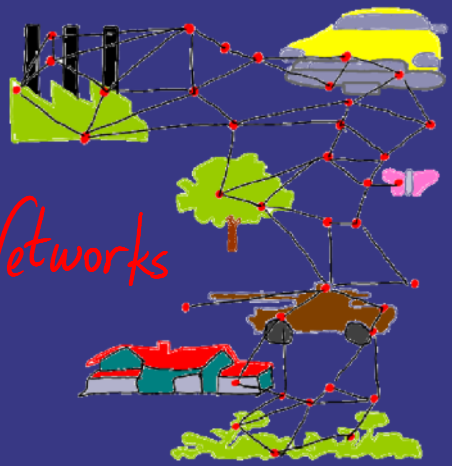
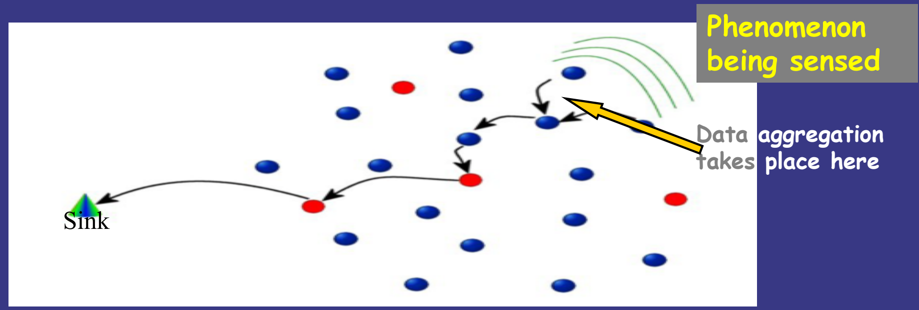
CmpE 58C Wireless Sensor Networks



Network Layer

Routing



Multihop routing is common due to limited transmission range

Some routing issues in WSNs →

- Low node mobility
- Power aware
- Irregular topology
- MAC aware
- Limited buffer space

Why can't we use conventional routing algorithms here?

- A sensor node is not an identity (address)
  - Content based and data centric
    - ✓ Where are nodes whose temperatures will exceed more than 10 degrees for next 10 minutes?
    - ✓ Tell me the location of the object (with interest specification) every 100ms for 2 minutes.

Why can't we use conventional routing algorithms here?

- Multiple sensors collaborate to achieve one goal.
- Intermediate nodes can perform data aggregation and caching in addition to routing.
  - ✓ where, when, how?

Why can't we use conventional routing algorithms here?

- Not node-to-node packet switching, but node-to-node data propagation.
- High level tasks are needed:
  - ✓ At what speed and in what direction was that elephant traveling?
  - ✓ Is it the time to order more inventory?

Challenges

- Energy-limited nodes
- Computation
  - *Aggregate data*
  - *Suppress redundant routing information*
- Communication
  - Bandwidth-limited
  - Energy-intensive

Goal: Minimize energy dissipation

Challenges

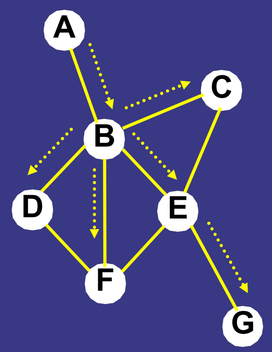
- Scalability: ad-hoc deployment in large scale
  - Fully distributed w/o global knowledge
  - Large numbers of sources and sinks
- Robustness: unexpected sensor node failures
- Dynamically Change: no *a-priori* knowledge
  - sink mobility
  - target moving

Challenges

- Topology or geographical issue
- Time: out-of-date data is not valuable
- Value of data is a function of time, location, and its real sensor data.
- Is there a need for some general techniques for different sensor applications?
  - Small-chip based sensor nodes
  - Large sensors
  - Moving sensors, e.g., robotics

What Should be the Optimum "Ideal" Routing Protocol for WSNs

- Shortest-path routes
- Avoids overlap
- Minimum energy consumption
- Needs global topology information



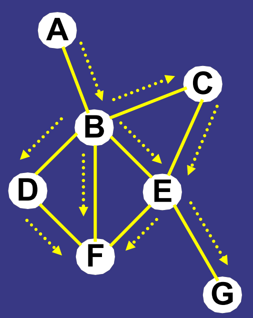
Taxonomy of Routing Protocols for Wireless Sensor Networks

K. Akkaya and M. Younis, "A Survey on Routing Protocols for Wireless Sensor Networks," AdHoc Networks Journal, 2005

1. DATA CENTRIC PROTOCOLS
   Flooding, Gossiping, SPIN, Directed Diffusion, SAR (Sequential Assignment Routing), Rumor Routing, Constrained Anisotropic Diffused Routing, COUGAR, ACQUIRE
2. HIERARCHICAL PROTOCOLS
   LEACH, PEGASIS, TEEN (Threshold Sensitive Energy Efficient Sensor Network Protocol), APTEEN
3. LOCATION BASED (GEOGRAPHIC) PROTOCOLS
   MECN, SMECN (Small Minimum Energy Com Netw), GAF (Geographic Adaptive Fidelity), GEAR, Distributed Topology/Geographic Routing Algorithm (PRADA)

Conventional Approach: FLOODING

- Broadcast data to all neighbor nodes



Gossiping

- Sends data to one randomly selected neighbor

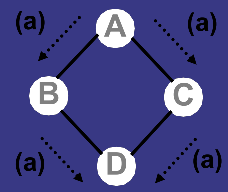
Problems of Flooding and Gossiping

- Although these techniques are simple and reactive, they have some disadvantages including:
  - Implosion

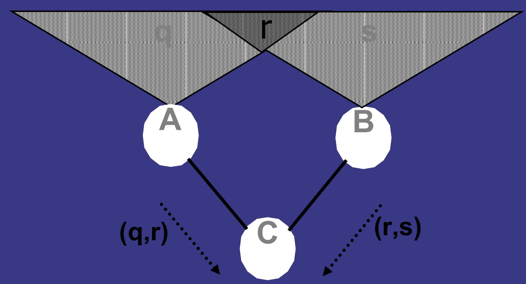
    (NOTE: Gossiping avoids this by selecting only one node; but this causes delays to propagate the data through the network)
  - Overlap
  - Resource Blindness
  - Power (Energy) Inefficient

Problems of Flooding and Gossiping

Implosion



Data Overlap



Resource Blindness

No knowledge about the available power of resources

Gossiping

- Uses randomization to save energy

  Selects a single node at random and sends the data to it
- Avoids implosions
- Distributes information slowly
- Energy dissipates slowly

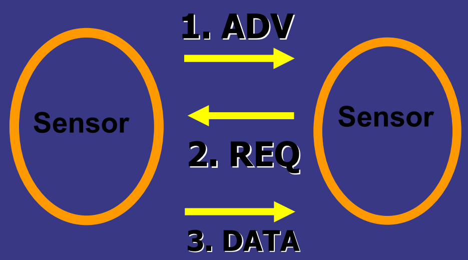
SPIN: Sensor Protocol for Information via Negotiation

W.R. Heinzelman, J. Kulik, and H. Balakrishnan, "Adaptive Protocols for Information Dissemination in Wireless Sensor Networks", *Proc. ACM MobiCom'99,* pp. 174-185, 1999

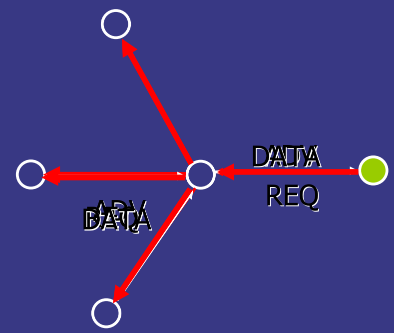
- Uses three types of messages: ADV, REQ, and DATA.
- When a sensor node has something new, it broadcasts an advertisement (ADV) packet that contains the new data, i.e., the meta data.
- Interested nodes send a request (REQ) packet.
- Data are sent to the nodes that request by DATA packets.
- This will be repeated until all nodes will get a copy.

SPIN

- Good for disseminating information to all sensor nodes.
- SPIN is based on data-centric routing where the sensors broadcast an advertisement for the available data and wait for a request from interested sinks



SPIN



Family of SPIN Protocols

- **SPIN-PP** – For point-to-point communication
- **SPIN-EC** – Similar to SPIN-PP but with energy conservation heuristics added to it
- **SPIN-BC** – Designed for broadcast networks. Nodes set random timers after receiving ADV and before sending REQ to wait for someone else to send the REQ
- **SPIN-RL** – Similar to SPIN-BC but with added reliability. Each node keeps track of whether it receives requested data within the time limit, if not, data is re-requested

SPIN-PP Protocol

- SPIN-PP
  - 3-stage handshake protocol
  - Advantages
    - Simple
    - Implosion avoidance
    - Minimal start-up cost
  - Disadvantages
    - Cannot isolate the nodes that do not want to receive the information.
    - Consumes unnecessary power.

SPIN-EC

- Spin-EC
  - SPIN-PP + low-energy threshold
  - Modifies behavior based on current energy resources

SPIN-EC

- Adds a simple energy conservation heuristic
- When energy is plentiful, SPIN-EC behaves like SPIN-PP
- When energy approaches a low-energy threshold, SPIN-EC node reduces its participation in the protocol (DORMANT)
  - participate in a stage of protocol only if the node believes that it can complete all the remaining stages

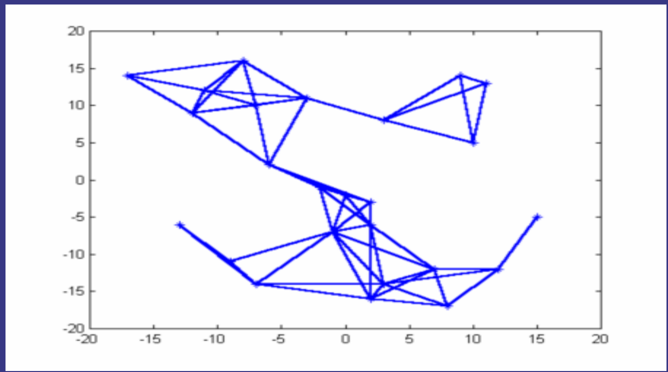
SPIN- Conclusions

- Flooding converges first
  - No delays
- SPIN-PP
  - Reduces energy by 70%
  - No redundant DATA messages
- SPIN-EC distributes
  - 10% more data per unit energy than SPIN-PP
  - 60% more data per unit energy than flooding

SPIN-Conclusions

- Energy – More efficient than flooding
- Latency – Converges quickly
- Scalability – Local interactions only
- Robust – Immune to node failures

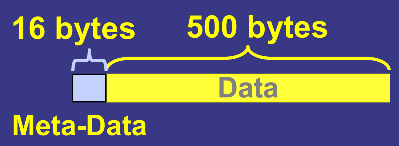
Test Network



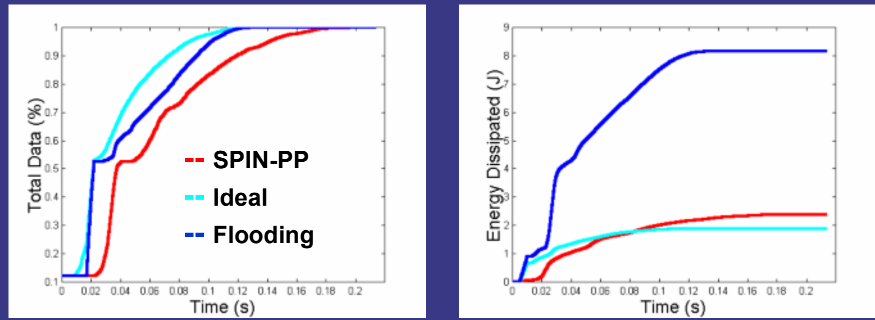
25 Nodes

59 Edges

Average degree = 4.7 neighbors
Network diameter = 8 hops
Antenna reach = 10 meters

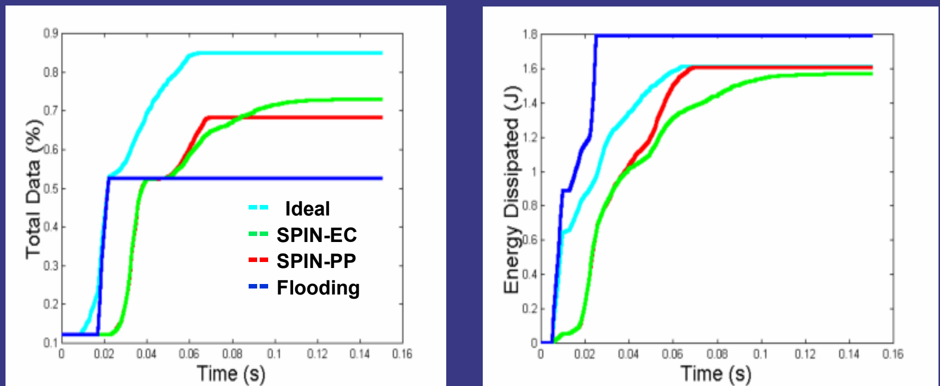


Unlimited Energy Simulations



- Flooding converges first
  - No queuing delays
- SPIN-PP
  - Reduces energy by 70%
  - No redundant DATA messages

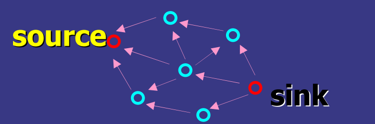
Limited Energy Simulations



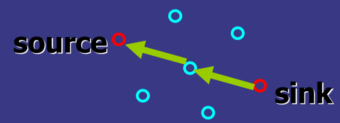
- SPIN-EC distributes additional 20% data

Directed Diffusion Routing Algorithm

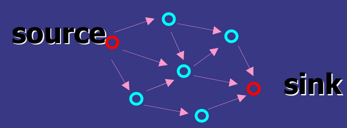
C. Intanagonwiwat, *et al.*,
"Directed Diffusion: A Scalable and Robust Communication Paradigm for Sensor Networks", IEEE/ACM Transactions on Networking, 2003



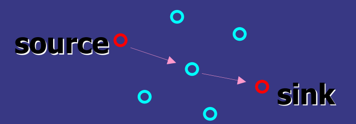
Interest Propagation



Reinforcement



Gradient Setup



Data Delivery

Directed Diffusion Overview

- Data-centric vs. address centric architecture
  - Individual network address is not critical; Data is important and is accessed as needed
- User can pose a specific task, that could be executed by sensor nodes
- Concept of Named Data: (Attribute, Value) pair
- Sink node requests data by sending "interests" for data
- Interests are propagated through the network, setting up gradients in the network, designed to "draw" data
- Data matching the interest is then transmitted towards the sink, over single or multiple paths (obtained by the gradients)
- The sink can then reinforce some of these paths to optimize

Basic Principles Used for Directed Diffusion Algorithm

- DATA CENTRIC ROUTING scheme !!!!
- Very large number of sensors → impossible to assign specific IDs.
- Without a unique identifier, gathering data may become a challenge.
- To overcome this challenge, some routing protocols gather/route data based on <u>the description of the data, i.e., data-centric.</u>

What is DATA CENTRIC?

- Data-Centric
  - Sensor node does not need an identity!!!
    - What is the temp at node #27 ?
  - Data are named by attributes
    - Where are the nodes whose temperature recently exceeded 30 degrees ?
    - How many pedestrians do you observe in region X?
    - Tell me in what direction that vehicle in region Y is moving?
- Application-Specific
  - Nodes can perform application specific <u>data aggregation, caching, and forwarding</u>

What is Data-Centric Routing?

- Requires attribute based naming where the users are more interested in querying an attribute of the phenomenon, rather than querying an individual node.
- Example:

  "the areas where the temperature is over 70F"

  is a more common query than

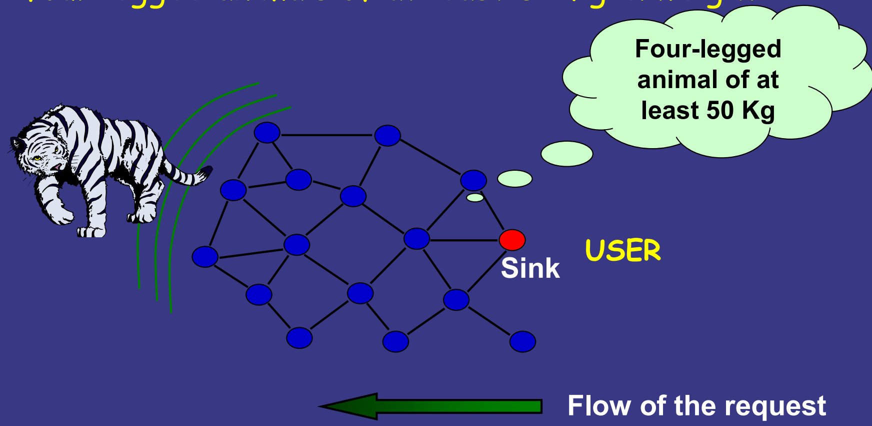
  "the temperature read by a certain node (e.g., #27)".

Assigning Sensing Tasks to Sensor Nodes

- This can be done in two ways:
  - INTEREST DISSEMINATION
  - DATA ADVERTISEMENT

Interest Dissemination

The sink or controlling nodes broadcast the nature of the interest,
e.g. "four legged animals of at least 50 Kg in weight"

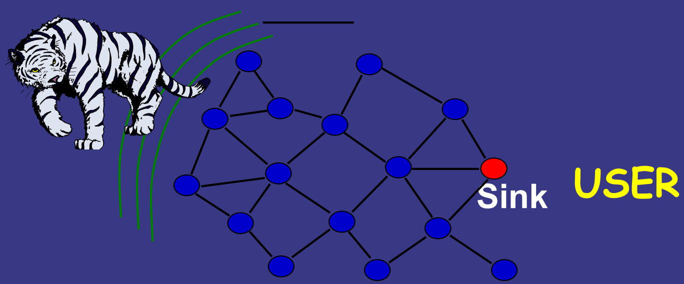


Advertisement of Available Data

Sensor nodes broadcast an advertisement of available sensed data and wait for a request from the interested sinks

Tiger, tiger, burning bright,
In the forest of the night,
What immortal hand or eye
Could frame thy fearful symmetry?

Flow of the advertisement



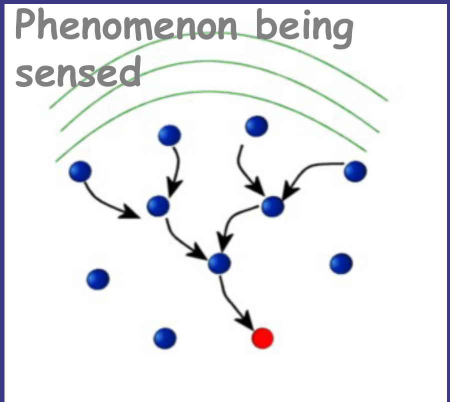
Data Aggregation

- To solve the problem of implosion in WSNs.
- When more than one node senses the same phenomenon, packets carrying the same information may arrive at the same node

NOTE:

- Different from "duplicate packets" in conventional ad hoc networks.
- Here → a high level interpretation of the data in the packets is that determines if the packets are the "same."
- Even for the case when the packets are deemed to be different, they could still be aggregated into a single packet before the relaying process continues.
- In this regard data aggregation can be considered as data fusion.

Data Aggregation



- Data coming from multiple sensor nodes are aggregated, if they have about the same attributes of the phenomenon being sensed, when they reach a common routing or relaying node on their way to the sink.
- In this view the routing mechanism in a sensor network can be considered as a form of reverse multicast tree.

Metrics for Data Aggregation

- *Accuracy*: Difference between value(s) the sink obtains from aggregated packets and from the actual value (obtained in case no aggregation/no faults occur)
- *Completeness*: Percentage of all readings included in computing the final aggregate at the sink
- *Latency*
- *Message overhead*

How to Express Aggregation Request?

- One option: Use database abstraction of WSN
- Aggregation is requested by appropriate SQL clauses

```
SELECT {agg(expr), attributes} FROM sensors
WHERE {selectionPredicates}
GROUP BY {attributes}
HAVING {havingPredicates}
EPOCH DURATION i
```


How to express aggregation request?

- Agg(expr): actual aggregation function, e.g., AVG(temperature)
- WHERE: filter on value before entering aggregation process
  - Usually evaluated locally on an observing node
- GROUP BY: partition into subsets, filtered by HAVING
  - GROUP BY floor HAVING floor > 5

Aggregation Operations – Categories

- Duplicate Sensitive, e.g., median, sum, histograms
- Duplicate Insensitive: e.g., maximum or minimum
- Summary or examplary

Placement of Aggregation Points

- Convergecast trees provide natural aggregation points
- But: what are *good* aggregation points?
  - Ideally: choose tree structure such that the size of the aggregated data to be communicated is minimized
  - Figuratively: long trunks, bushy at the leaves
  - In fact: again a Steiner tree problem in disguise

Placement of Aggregation Points

- Good aggregation tree structure can be obtained by slightly modifying Takahashi-Matsuyama heuristic
- Result: no simple rule guarantees an optimal aggregation structure
- Can be regarded as optimization problem as well

Elements of Directed Diffusion

- Naming Scheme
  - *Data* is named using attribute-value pairs
- Interests
  - A node requests data by sending *interests for named data*
- Gradients
  - *Gradients* is set up within the network designed to "draw" events, i.e. *data matching the interest.*
- Reinforcement
  - Sink *reinforces* particular neighbors to draw higher quality (higher data rate) events

Naming Scheme

- Data generated by sensor nodes is NAMED by ATTRIBUTE-VALUE pairs
- In order to create a query, an interest is defined using a list of attribute-value pairs such as name of *objects, interval, duration, geographical area, etc.*
- An arbitrary sensor node (usually the SINK) uses attribute-value pairs (interests) for the data and queries the sensors in an on-demand basis.

Naming Scheme

Request:

Interest (Task) Description

- Example (Animal Tracking Task):

  Type = four legged animal (detect animal location)

  Interval = 30 s (send back events every 30 s)

  Duration = 1h (.. for the next hour)

  Rec = [-100,100,200,400] (from sensors within the rectangle)

Naming Scheme

- The data sent in response to interests are also named similarly.

Example: REPLY

Sensor detecting the animal generates the following data:

  Type – four legged animal (type of animal seen)

  Instance= elephant (instance of this type)

  Location = (125,220) (node location)

  Intensity = 0.6 (signal amplitude measure)

  Confidence = 0.85 (confidence in the match)

  Timestamp= 01:20:40 (event generation time)

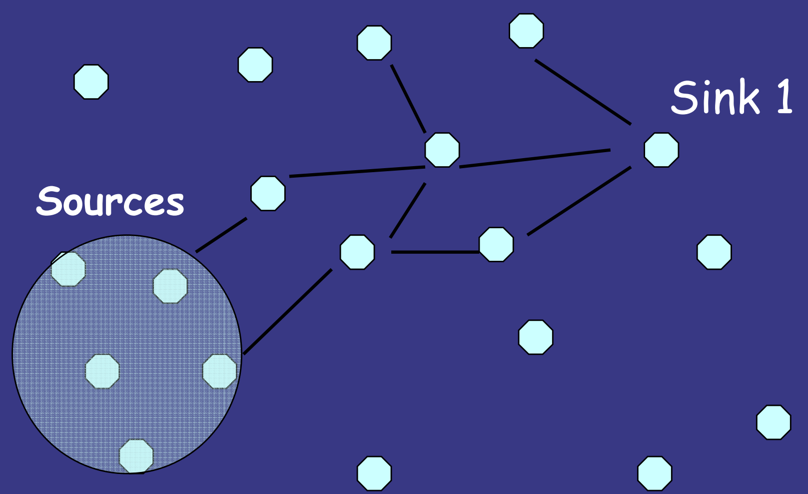
Interests

- The sink periodically broadcasts an interest to sensor nodes to query information from a particular area in the field.
- As the interest propagates, data may be *locally transformed* (e.g., *aggregated*) at each node, or be *cached*.
- Every node maintains an interest cache
  - Each item corresponds to a distinct interest
  - Interest aggregation: identical type, completely overlap rectangle attribute
- Each entry in the cache has several fields
  - Timestamp: last received matching interest
  - Several gradients: data rate, duration, direction
- Other nodes may express *interests* based on these attributes

Interests

- When a node receives an interest, it:
  - Checks cache to see if an entry is present.
  - If no entry, creates an entry with a single gradient to neighbor who sent this interest
  - Gradient specifies the direction and data rate.
- Resend interest to a subset of its neighbors
  - This is essentially flooding-based approach
  - Other probabilistic, location-based and other intelligent forwarding approaches possible
- Similar to multicast tree formation, at sink instead of at source

Interest Propagation



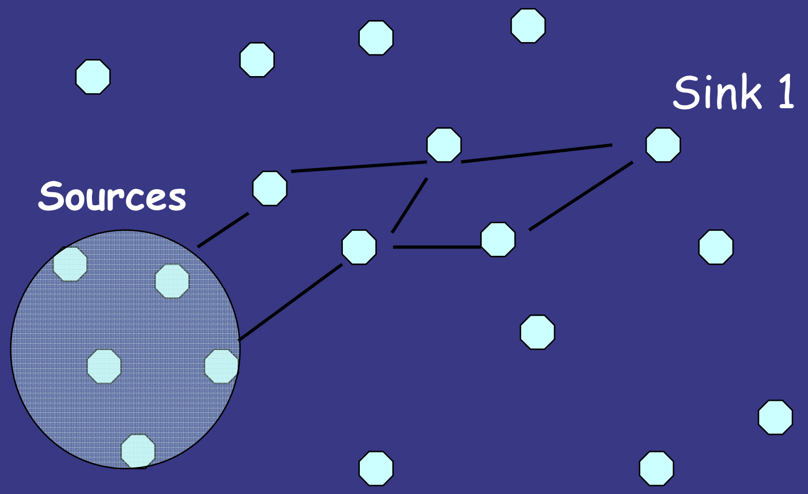
GRADIENT SETUP

- When a sensor node detects a target, it:
  - Searches interest cache for matching entry
  - If found, computes highest requested event rate among its gradients

Setting Up Gradients

- The sensor nodes send GRADIENT SETUP replies back towards to the sink.
- Each sensor on the path compares the interests with the gradients and updates its gradient fields
- Each sensor then forwards these gradients to the next neighbors.

Gradient Setup



Local Rules for Propagating Interests

- Just flood interest
- More sophisticated techniques possible:
  - Directional interest propagation based on cached aggregate information

    "I recently heard about suspicious activity from neighbor A, so let me try sending this interest for recent intrusions to that neighbor"

Local Rules for Establishing Gradients

- Highest gradient towards neighbor who first sends interest
- Others possible e.g.,
  - towards neighbor with highest remaining energy (energy gradients)
  - probabilistic gradients.

Gradient Reinforcement

- All gradients end up at the sink (destination/user).
- Sink selects the best path based on the contents of the collected gradient packets and the application requirements; i.o.w., sink picks a suitable set of neighbor(s) (best link, low delay, etc.)
- As a result, sink unicasts the reinforcement packet to the next hop indicating the selected path based on the gradient packet
- Each intermediate sensor forwards the reinforcement packet to its next hop based on the same principle
- At the end, the data path from source to destination will be established.

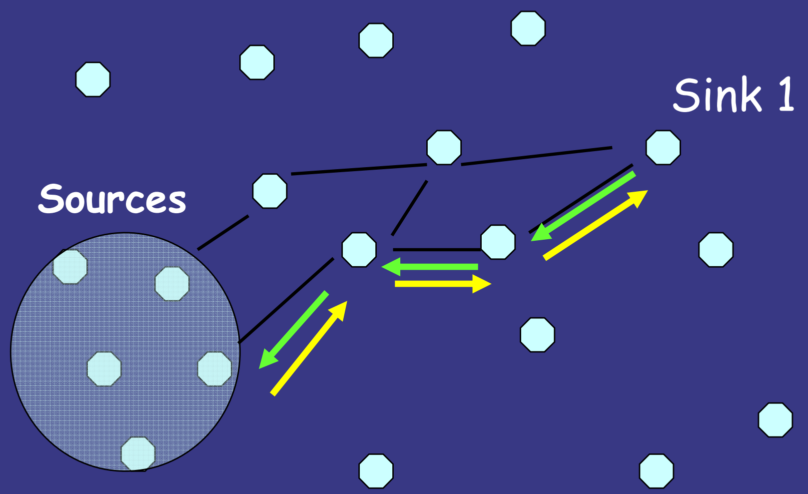
Gradient Reinforcement

- We can define several criteria for selecting which path is reinforced
  - amount of data received from neighbor
  - loss rates
  - observed delay variance

Data Propagation

- Data will be sent based on the data rate from sources towards the sink based on the established data path.
- Each intermediate node forwards the data to its next hop neighbor.

Reinforcement and Data Propagation

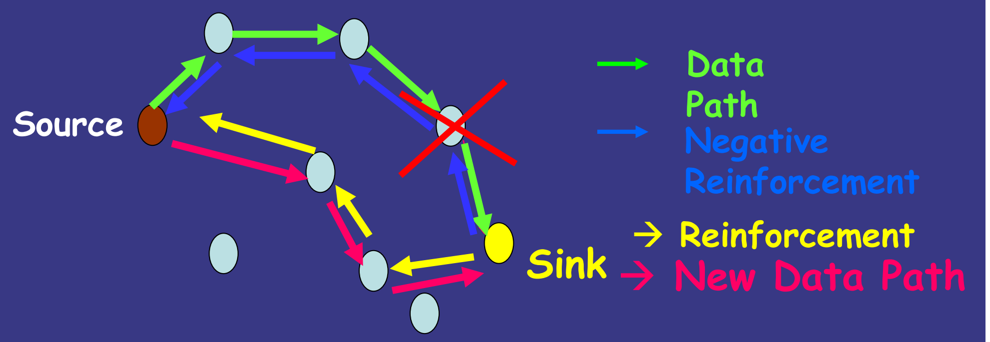


Data Transmission Choices

- Different local data forwarding rules can result in different kinds of transmission
  - single path delivery
  - multi-path delivery, with traffic on each link proportional to its gradient
  - delivery from single source to multiple sinks
  - delivery from multiple sources to multiple sinks

Negative Reinforcement

- Time out
- Explicitly degrade the path by re-sending interest with lower data rate.



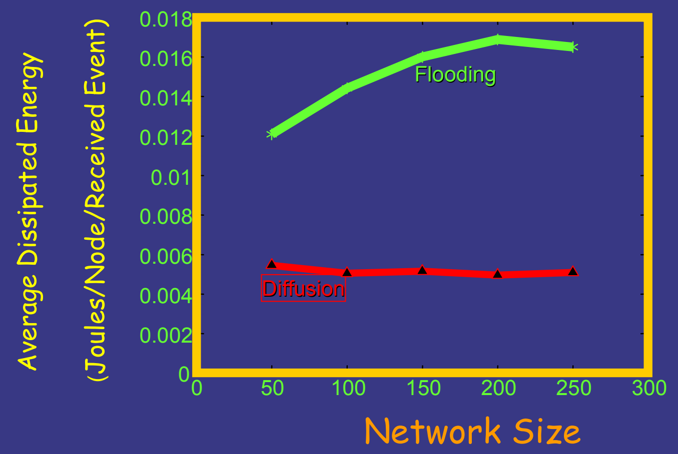
Simulation

- Simulator: *ns-2*
- Network Size: 50-250 Nodes
- Total area for 50 nodes 160m x 160m
- Transmission Range: 40m
- Constant Density: 1.95×10^{-3} nodes/m^2 (9.8 nodes in radius)
- MAC: Modified Contention-based MAC
- Energy Model: Mimic a realistic sensor radio
  - 660 mW in transmission, 395 mW in reception, and 35 mw in idle

Performance Metrics

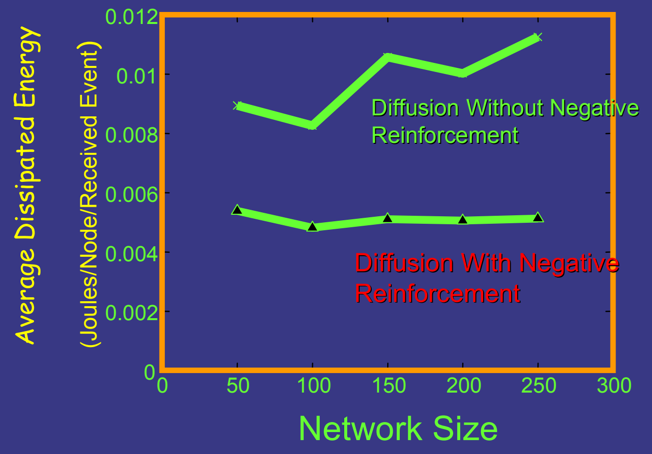
- Average Dissipated Energy
  - Ratio of total dissipated energy per node in the network to the number of distinct events seen by sinks.
- Average Delay
  - Average one-way latency observed between transmitting an event and receiving it at each sink.
- Event Delivery Ratio
  - Ratio of the number of distinct events received to number originally sent.

Average Dissipated Energy (Sensor Radio Energy Model)



Diffusion can outperform flooding.

Impact of Negative Reinforcement



Reducing high-rate paths in steady state is critical

Directed Diffusion - Extensions

- Two-Phase Pull suffers from interest flooding problems
- Push Diffusion - Data Advertisement by the sources
  - Sink sends reinforcement packet.

Directed Diffusion vs SPIN

- In DD → Sink queries sensors if a specific data is available by flooding some interests.
- In SPIN → Sensors advertise the availability of data allowing sinks to query that data.

Directed Diffusion
Advantages

- DD is data centric → no need for a node addressing mechanism.
  - Each node is assumed to do aggregation, caching and sensing.
- DD is energy efficient since it is on demand and no need to maintain global network topology.

Directed Diffusion

Disadvantages

- Not generally applicable since it is based on a query driven data delivery model.
- For DYNAMIC applications needing continuous data delivery (e.g., environmental monitoring) → DD is not a good choice.
- Naming schemes are application dependent and each time must be defined a-priori.
- Matching process for data and queries cause some overhead at sensors.