Wireless Sensor Networks (WSNs)

Technological Revolution

1. Computer Networking — 1990
   - LAN
   - Internet
2. Wireless Communications — 2000
   - GSM/UMTS
   - WLAN
3. Wireless Sensing Technologies — 2010
   - MEMS Technology
   - WSNs

Applications for Wireless Sensor Networks

- **Military Applications**

  (monitoring friendly forces, monitoring equipment, battlefield surveillance, reconnaissance of opposing forces and terrain)

- **Environmental Monitoring**

  (flood/forest fire detection, space exploration, biological attack detection)

- **Commercial Applications**

  (home/office smart environments, health applications. environmental control in buildings)

- **Tracking**

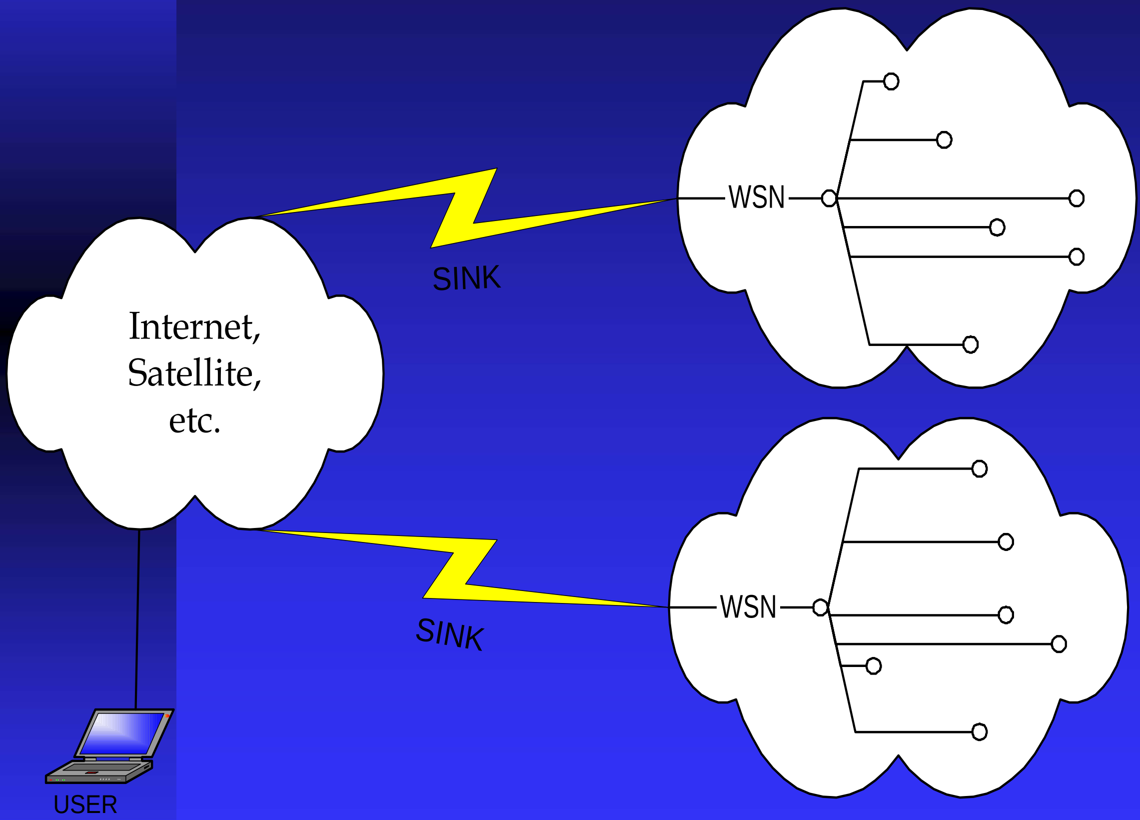
  (targeting in intelligent ammunition, tracking of doctors and patients inside a hospital)

Application Examples

WSN Model Terminology

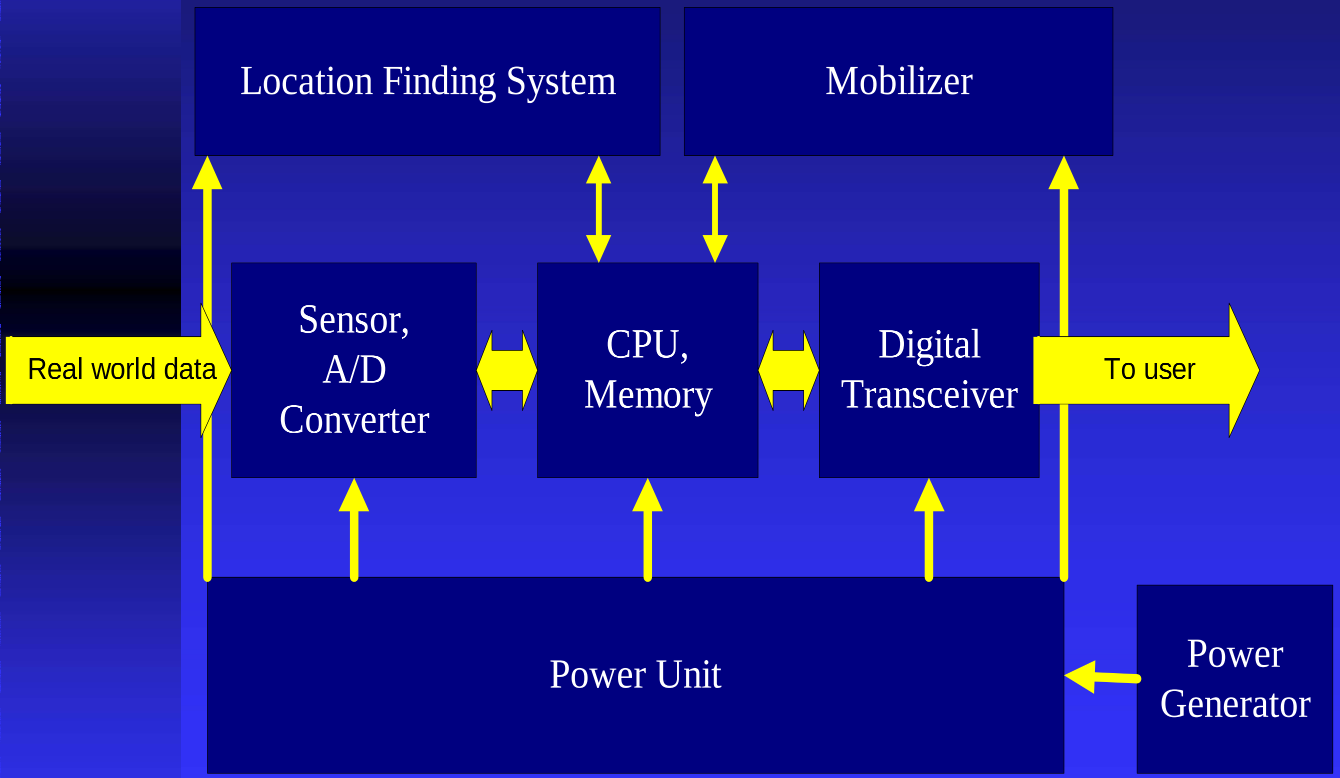
1. **Sensors**
   - Make discrete, local samples (*measurements*) of the phenomenon
   - Communicate over wireless medium, forming a wireless sensor network
   - *Disseminate information* about the phenomenon to the observer
2. **Observer**
   - Is interested in *measuring/ monitoring* the behaviour of a phenomenon
   - Accepts measurements under specific performance requirements (accuracy or delay*)*
3. **Phenomenon**
   - Entity of interest to the observer

System Architecture



- Cheap, low-power, tiny sensors used in **thousands**
- Communication with the use of miniaturized **wireless transceivers**
- **Data aggregation** during data propagation or at the sink
- **Unattended** operation of the sensor network
- Sink transmits data to the end-user at the **other end** of the world

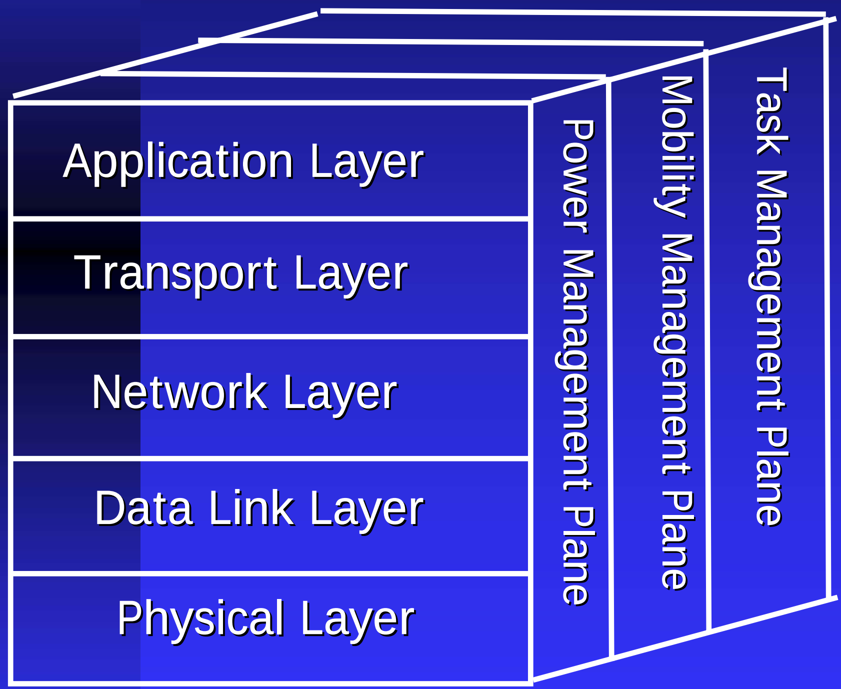
Sensors Hardware Platform



Node characteristics

- Tiny size
- Low power
- Low bit rate
- High densities
- Low cost (dispensable)
- Autonomous
- Adaptive

Communication Architecture



- **Cross-layer design** of protocol stack
- **Integration** of routing functionality and power awareness (energy-aware routing)
- **Integration** of routing functionality and data transport (aggregation)
- **Inclusion** of mobility as a network control primitive
- Promotes cooperative efforts (task management plane)

WSNs vs. MANETs

WSNs and MANETs are equivalent networks build for different purposes!

Similarities

- Data communication over wireless medium
- Ad-hoc network topology
- Power and bandwidth are scarce resources

WSNs vs. MANETs

<u>Differences</u>

- WSNs are **deployed** and **owned** by a single user
- Sensor nodes are **extremely cheap, tiny** devices, not like ad-hoc network nodes (PDAs, laptops, etc.)
- No general purpose communication network, but a **data-gathering, surveillance** network
- Number of nodes **several orders of magnitude** higher than MANETs
- **Energy and bandwidth conservation** is a primary concern in WSN protocol design

WSNs vs. MANETs Comparison Summary

Features	MANET	WSN
Multi-hop routing protocols applicable	Yes	Yes
Ad-hoc deployment (unattended operation)	Yes	Yes
Extreme power constraints for nodes operation	No	Yes
Low-cost nodes of tiny size	No	Yes
Robust to node failures (self-healing)	Yes	Yes

WSNs vs. MANETs Comparison Summary

Features	**MANET**	**WSN**
General purpose communication network	Yes	No
Node density	<100	<1000
Mobility of nodes	Yes	Yes
In-network data processing	No	Yes
Unique global IP addresses	Yes	No

Sensor Network Protocols Design Challenges

- **Energy depletion** is the main resource bottleneck
  - Reduce each sensor's **active duty cycle**
  - **Minimize data communication** over wireless channel
    - Use computation to reduce data size (data aggregation)
    - Communicate only network state summaries instead of actual data
  - Maximize total network lifetime
    - Minimum energy routing

Sensor Network Protocols Design Challenges

- **Robustness** to dynamic environment
  - Network should be *self-configuring*
  - Network should be *self-healing*
  - Network should be *adaptive* (measure and act)
- **Scalable to thousands** of nodes
  - Organize network in a *hierarchical* manner (possibly with the use of clustering)
  - Use only *localized algorithms*; with localized interactions between nodes

Sensor Network Protocols Design Characteristics

- Data-centric operation
  - Focus on **application data**, not individual nodes: information gathering is the purpose of sensor networks

Traditional networks:

"What is the temperature **at sensor #27** ? "

Sensor Networks:

"**Where are the nodes** whose temperatures recently exceeded 30 degrees? "

Sensor Network Protocols Design Characteristics

- Application-specific design
  - WSN networks can be **tailored to** the sensing task at hand
  - Intermediate nodes can perform application-specific **data aggregation** and **caching**
- Low energy expenditure at nodes
  - Use of low duty-cycled sensors
  - Coordinate groups of sensors to fall to the sleep stated

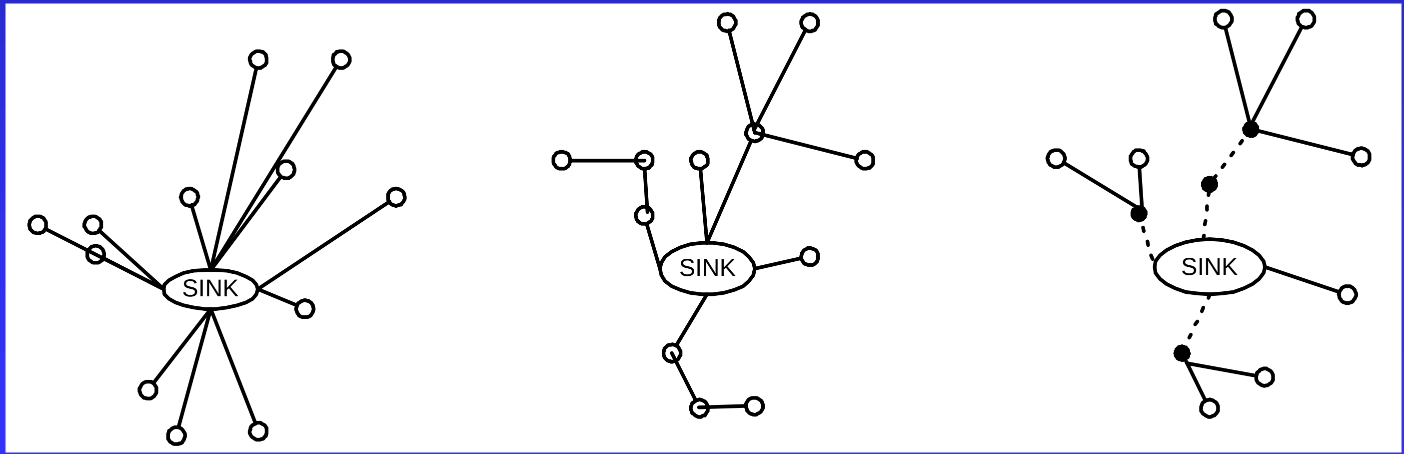
Classification of Routing Protocols

- According to route discovery
  1. Proactive
  2. Reactive
  3. Hybrid

- According to location awareness
  1. Location aware routing
  2. Location-less routing

Classification of Routing Protocols (cont'd)

- According to nodes' participating style
  1. Direct communication
  2. Flat routing
  3. Clustering routing protocols



Sensor Network Communication Protocols

- Proposed Sensor Network **Performance Metrics**
  - Energy efficiency/system lifetime
  - Latency
  - Accuracy
  - Fault-tolerance
  - Scalability

SPAN

Problem: Need to minimize the energy consumption of wireless nodes in a wireless ad hoc network!

IDEA:

Leverage the time the network interface of a node remains **idle** to **power-down** the radio of the node.

SPAN

Desired Characteristics

1. As many nodes as possible should be in sleep mode
2. Forwarding of packets should occur with minimal additional delays
3. Awake nodes should provide as much total capacity as original network
4. Distributed algorithm for so that nodes make *local* decisions

SPAN

Routing layer	GPSR	AODV	DSR
	Span		
MAC/Phy layer	802.11, HIPERLAN/2		

- Span is a power-saving protocol that operates **between** the routing layer and the MAC layer.

SPAN

Operation of SPAN

- Certain nodes are elected as **'coordinators'** to participate in the backbone network. Coordinators stay **always-on** to provide global connectivity of the network. The rest of nodes remain in **power-save mode** and periodically check to change status
- Coordinators are rotated among nodes
- Attempt to minimize the number of coordinators
- Distributed coordinators election process

SPAN

- Span is proactive: each node *periodically broadcasts* HELLO messages:
  1. the node's status
  2. its current coordinators
  3. its current neighbors
- From the HELLO messages each node *builds*
  1. a list of own neighbors and coordinators
  2. for each neighbor: a list of its neighbors and coordinators

SPAN

- Coordinator announcement

  Regular nodes **periodically** wake up and decide to become *coordinators* or not based on a coordinator eligibility rule

Coordinator eligibility rule

- A non-coordinator node should become a coordinator if it discovers, using only information gathered from local broadcast messages, that two of its neighbors cannot reach each other either directly or via one or two coordinators

SPAN

- Contention resolution

  What happens if two nodes decide to become coordinators at the same time?

- Introduce a **randomized backoff delay** at each node, based on
  - Nodes with **roughly equal remaining energy**

    N_i: number of neighbors at node i

    C_i: number of additional pairs of nodes to be connected if i became a coordinator

    $$0 \leq C_i \leq (N_i \text{ ov. } 2)$$

  - Define as **utility** of a node i: C_i / (N_i ov. 2)

SPAN

- Contention resolution

Nodes with **higher C_i** should volunteer **more quickly** than ones with smaller C_i

$$delay = \left(\left(1 - \frac{C_i}{\binom{N_i}{2}}\right) + R\right) \times N_i \times T.$$

the delay for each node is randomly chosen over an interval proportional to N_i x T

R picked uniformly at random from interval (0,1]

SPAN

- Contention resolution
  - Nodes with **unequal remaining energy**

    E_r: amount of remaining energy at a node

    E_m: maximum amount of energy available

Fairness rule

A node with **larger E_r/E_m** should become coordinator **more quickly**

$$delay = \left(\left(1 - \frac{E_r}{E_m} \right) + \left(1 - \frac{C_i}{\binom{N_i}{2}} \right) + R \right) \times N_i \times T.$$

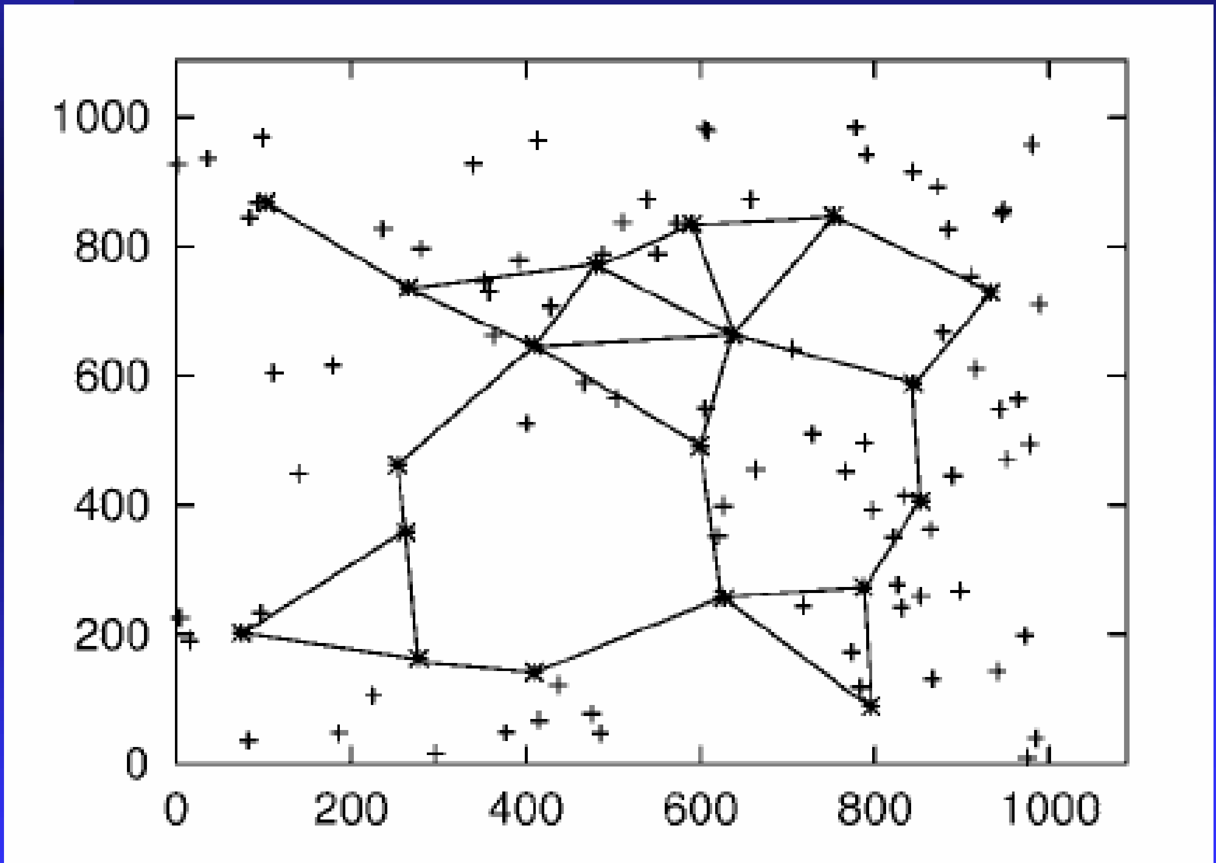
SPAN

- Coordinators withdrawal
  - Each coordinator **periodically** checks if it should withdraw as a coordinator
  - Rule to withdraw: every pair of its neighbors should be able to reach each other either **directly** or via **one or two** other coordinators
  - To rotate coordinators among all nodes fairly: use of **tentative coordinators**
  - **Tentative coordinators:** provide the chance for non-coordinators to become coordinators
  - Coordinators **stay tentative** for W_T amount of time

    $W_T = 3 \times N_i \times T$ (max. delay for cont. resolution)
  - After W_T, the tentative bit is removed

SPAN

- Illustration of SPAN alg. at some arbitrary moment



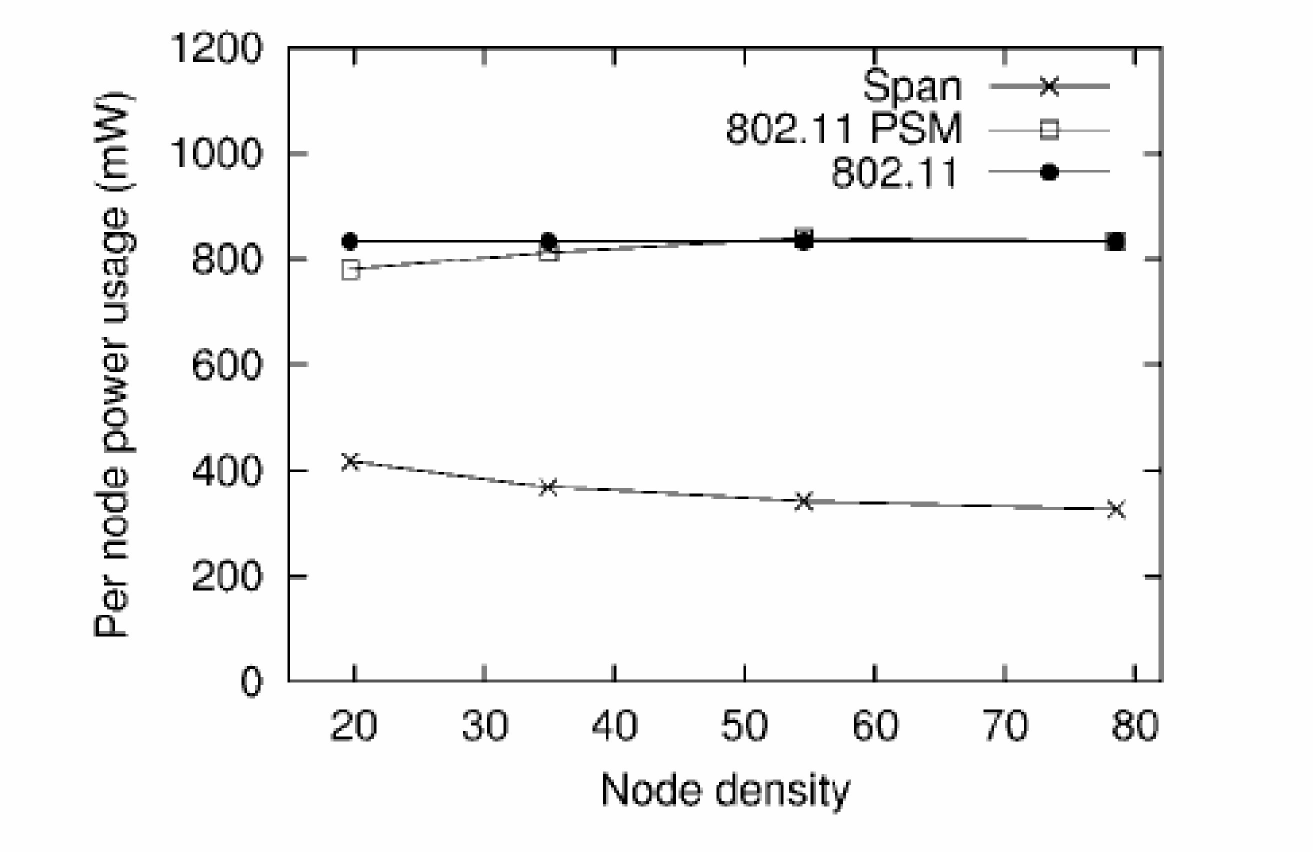
+: non-coordinator nodes

*: coordinator nodes

Solid lines: connect neighboring coordinators

SPAN

- Energy consumption characteristics



per-node power usage in networks running Span, 802.11 PSM, and 802.11

SPAN

- Pros
  - Achieves high energy-savings, even with regular ad hoc routing protocols
  - Slow increase of energy savings with higher network densities due to periodicity
  - Low latency, low throughput degradation
- Cons
  - Can not be applied to sensor networks, because sensing nodes may not be powered up or down
  - High communication overhead

LEACH

- Low Energy Adaptive Clustering Hierarchy
  - A **clustering-based** protocol utilizing randomized rotation of local cluster base stations (cluster-heads) to **evenly distribute the energy load** among the sensors in the network
  - LEACH makes the following assumptions:
    1. The base station is **fixed** and located **far** from the sensors
    2. All nodes in the network are **homogeneous** and **energy-constrained**

LEACH

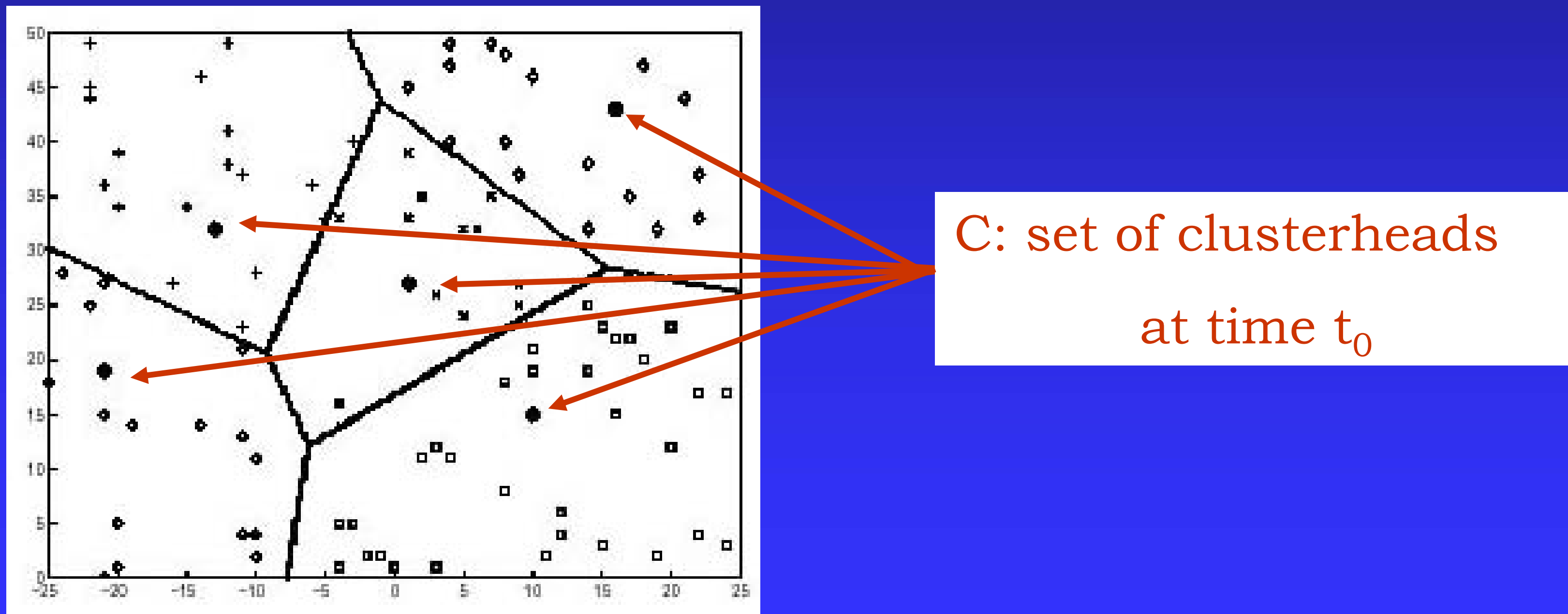
- Key features of LEACH:
  - **Localized** coordination and control for cluster set-up and operation
  - **Randomized rotation** of the cluster "base stations" or "cluster-heads" and the corresponding clusters
  - **Local compression** to reduce global communication

LEACH

- Protocol description
  - Nodes organize themselves into **local clusters**, with one node acting as **local base station** or "**cluster-head**"
  - **Randomized rotation** of high-energy cluster-head position so as not to 'drain' the energy of a single node
  - Election of clusterheads at any given time with a **certain probability**
  - Sensors choose their preferred clusterhead to belong to, based on the **minimum required energy** to communicate with
  - Clusterheads **create schedules** for the nodes in their cluster, so that plain nodes can **power-down** when they are not scheduled to transmit
  - Clusterheads **aggregate data** from sensors in cluster and transmit **compressed data** to the base station

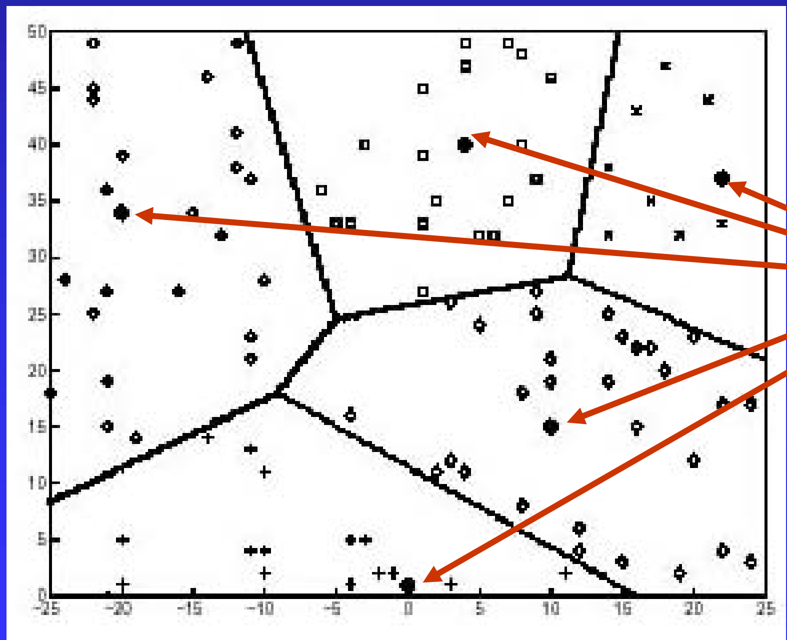
LEACH

- LEACH operates in consecutive rounds
- **Clusterheads are elected new** at each round of operation



LEACH

- New set of clusterheads C` for the next round



C`: set of clusterheads at time $t_0 + \delta_0$

LEACH

- Phases of operation

1. **Advertisement Phase**
    - Clusterheads are elected in this phase
    - Election is based on **P (percentage of clusterheads for the network**) and the number of times the node has been a clusterhead so far
    - Node n chooses a **random number** between 0 and 1 and if this number is less than a threshold T(n), the node becomes clusterhead in this round
    - Clusterheads broadcast advertisement messages using **CSMA MAC protocol** using the same energy
    - Receiving nodes decide which clusterehad to belong to based on the received advertisement signal strength
2. **Cluster Set-up Phase**
    - Nodes inform the clusterheads that they want to join their cluster
    - Again a **CSMA MAC protocol** is used

LEACH

- Phases of operation

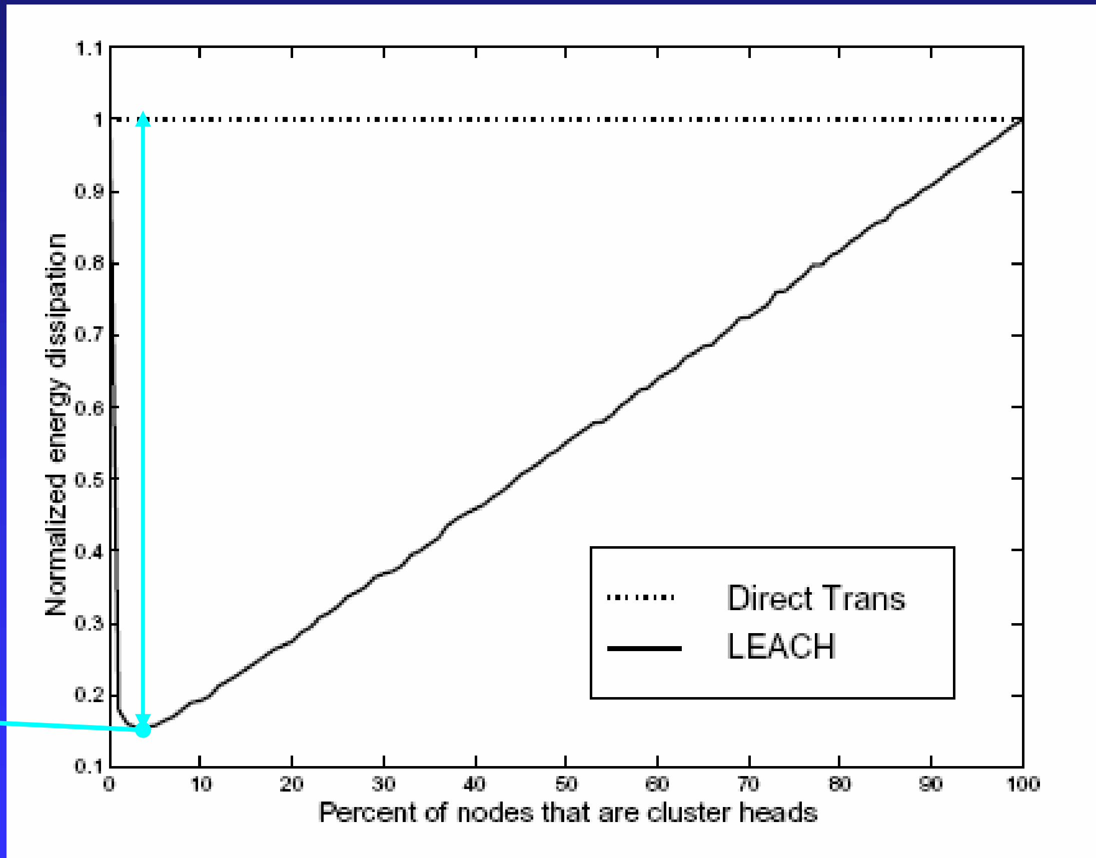
3. **Schedule Creation Phase**
   - Clusterheads receive all messages from nodes to be included in cluster
   - Based on the number of nodes in the cluster, **clusterhead creates TDMA schedule**
   - Schedule is **broadcast** to all cluster nodes
4. **Data Transmission Phase**
   - Assuming nodes have data to send, they wait for their **allocated time to send data** to the clusterhead
   - The rest of the time they **power down** their radio to conserve energy
   - Clusterhead performs **data fusion** so as to send compressed data to the sink
   - This final transmission is a high-energy data transmission

LEACH

Normalized total system energy dissipated versus the percent of nodes that are cluster-heads.

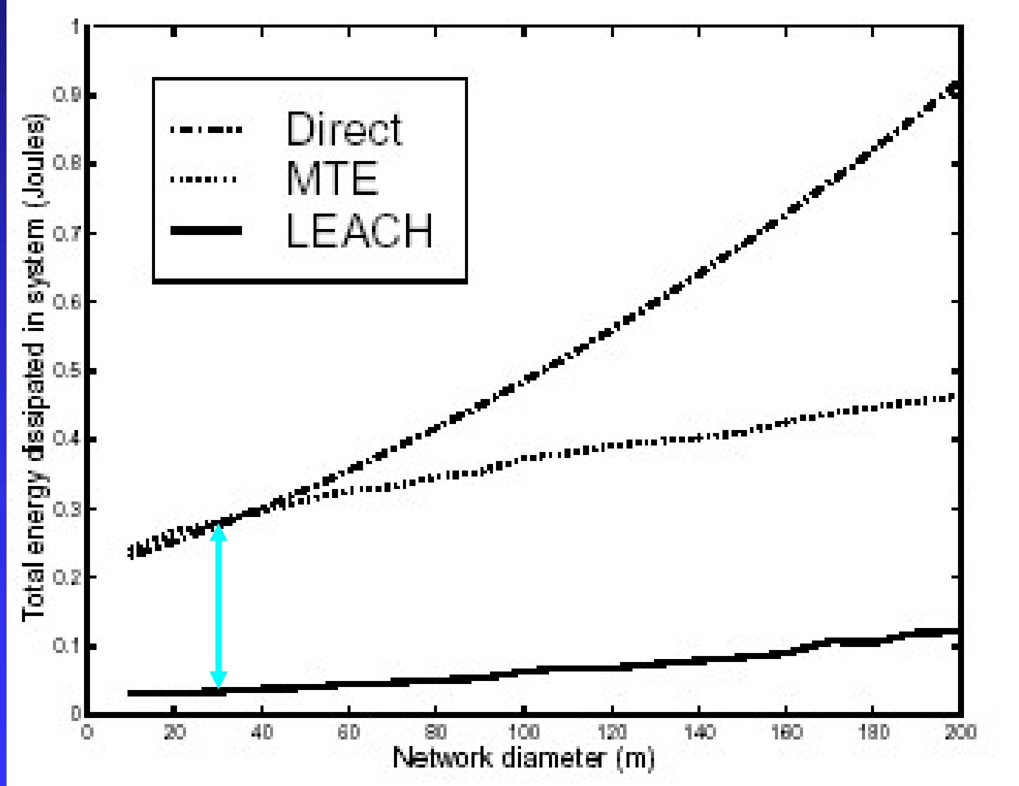
Over a factor of 7 for reduction in energy dissipation when optimal number of clusterheads

Optimal point of LEACH operation



LEACH

- Total system energy dissipated using direct communication, MTE and LEACH for a 100-node random network



Up to 8x reduction in energy dissipation between LEACH and conventional routing protocols

LEACH

- LEACH's strengths
  - Localised coordination of clusters
  - Randomized rotation of the clusterheads
  - Scalable due to clustering hierarchy
  - Energy-efficient due to the combination of data compression and routing

LEACH

- LEACH's weaknesses
  - Presence of a hot spot can deplete the power of nodes in its vicinity very quickly
  - Some sensors may not be able to power down due to their assigned tasks

SPIN

- Adaptive Protocols for Information Dissemination in Wireless Sensor Networks
  - Family of adaptive protocols called SPIN for efficient dissemination of information in energy-constrained wireless sensor network

- <u>SPIN characteristics</u>
  - Introduction of *high-level data descriptors* (use of *meta-data*)
  - Use of *meta-data negotiation* to eliminate transmission of redundant information
  - Nodes base communication decisions upon *application-specific knowledge* and *knowledge of the resources* that are available to them

SPIN

- Analysis of problems characterizing conventional protocols for data dissemination in a sensor network:
  1. Implosion
  2. Overlap
  3. Resource blindness

- SPIN solutions:
  1. Negotiation
  2. Resource adaptation

SPIN

- Implosion problem
- Overlap problem

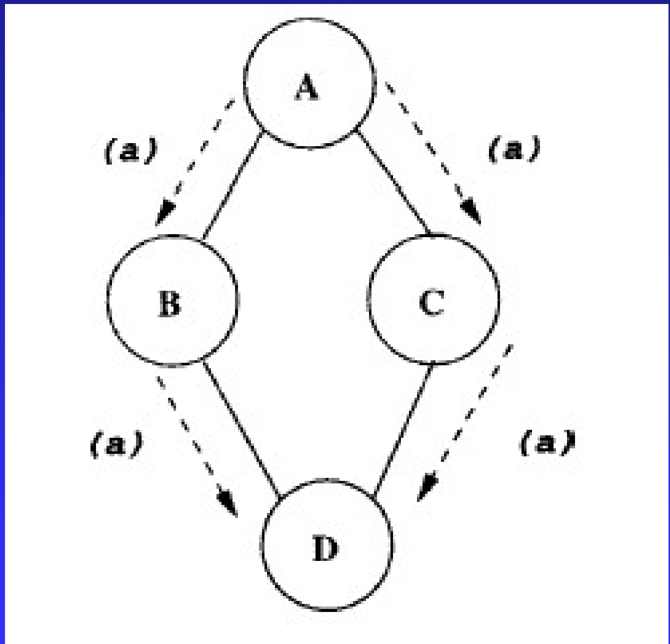


Figure 1

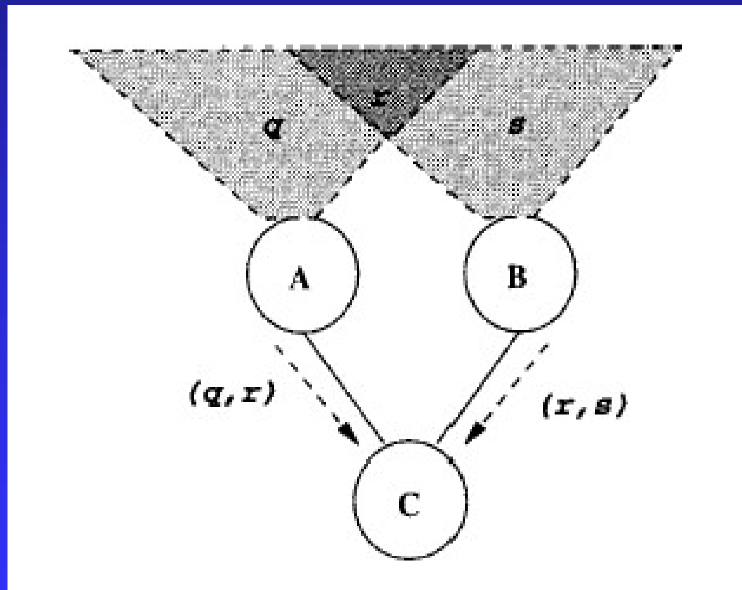


Figure 2

SPIN: Sensor Protocol for Information via Negotiation

- **Two basic ideas:**
  1. sensor applications need to communicate with each other **about** the data that they already have and the data they still need to obtain
  2. nodes in a network must **monitor** and **adapt** to changes in their own energy resources to extend the operating lifetime of the system

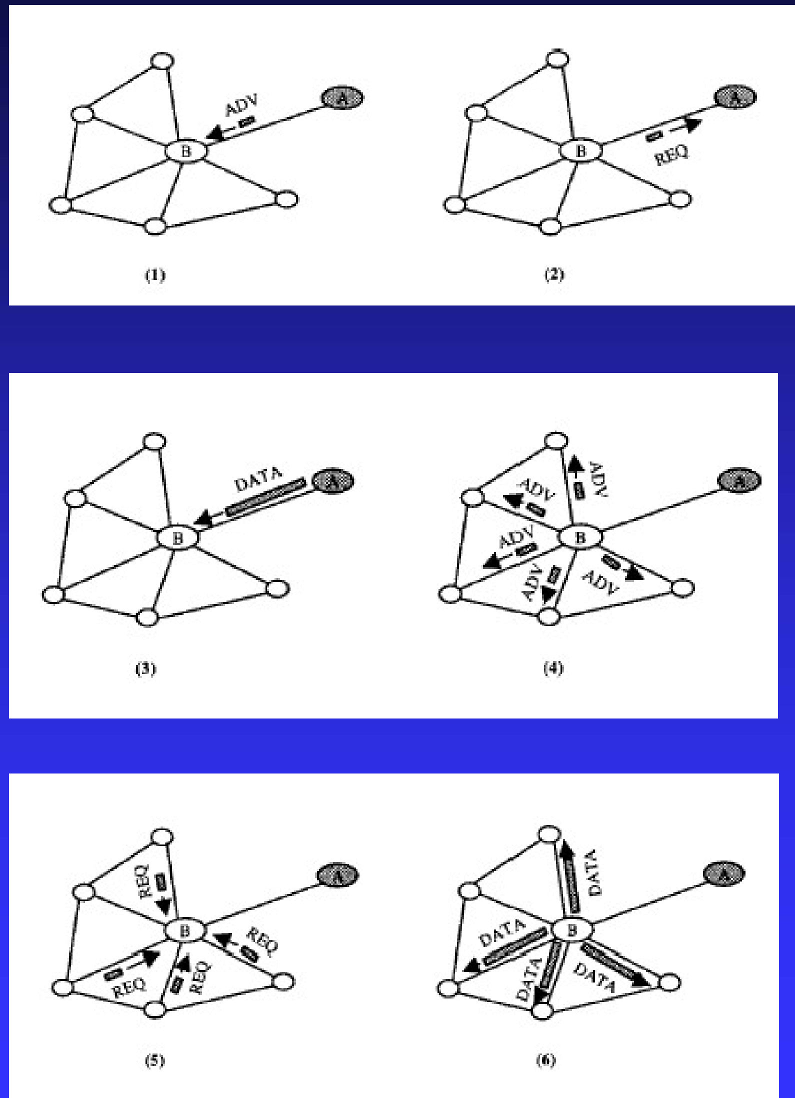
- **Meta-data:**

If x is the meta-data descriptor for sensor data X, then **size of x < size of X** for SPIN to be efficient

SPIN

- SPIN messages:
  1. **ADV**: New Data Advertisement (*meta-data*)
     Nodes that have data to share send advertisement messages containing meta data
  2. **REQ**: Request for Data (*meta-data*)
     Nodes wishing to receive some data, send request messages to inform the source node
  3. **DATA**: Data message (*data*)
     This message type contains actual sensor data with a meta-data header

SPIN-1: A 3-stage Handshake Protocol



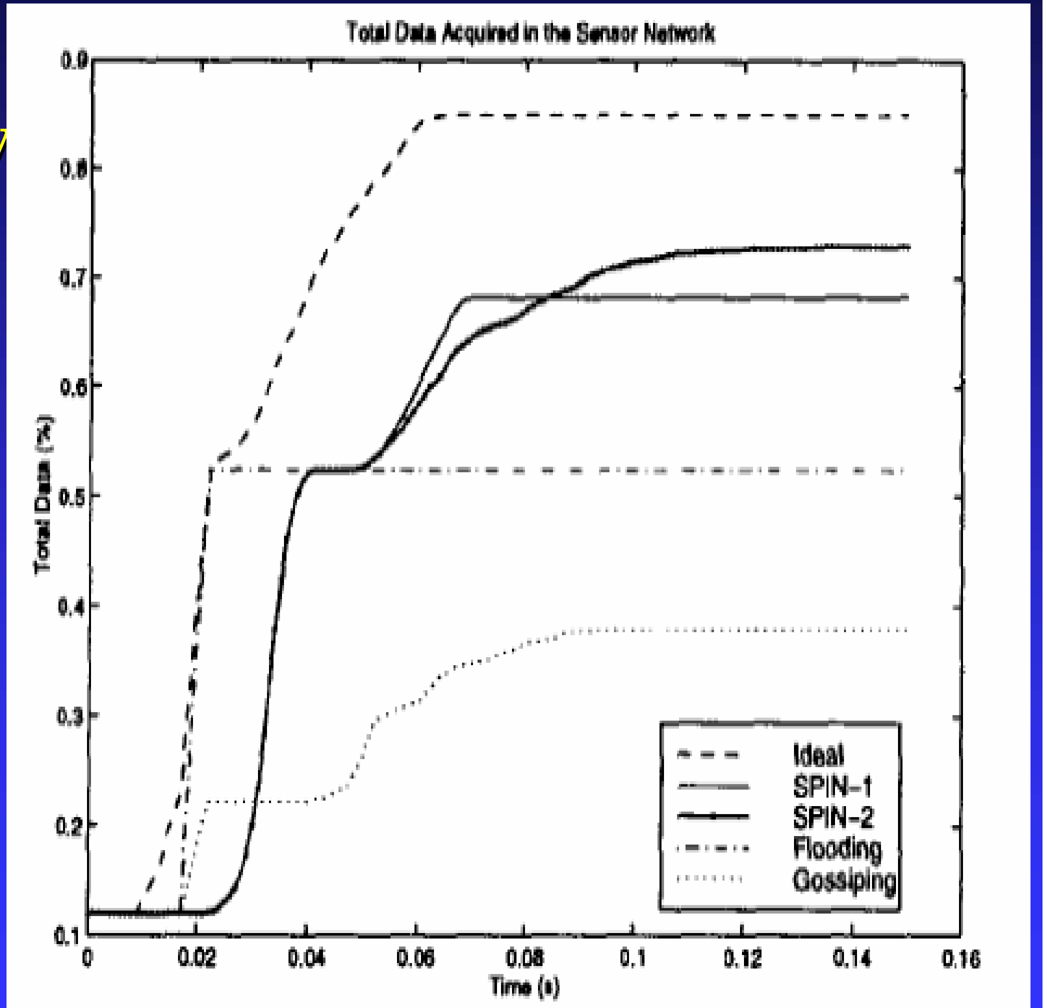
1. ADV stage
   - New Data Ad
   - Check for Data
   - Data Request

2. REQ stage
   - Data Transmission
   - Data Fusion
   - New Data Ad

3. DATA stage
   - Data Request
   - Data Transmission

SPIN: Limited-energy simulations

- Determine how effectively each protocol uses its available energy
  - SPIN-1 distributes 68%
  - **SPIN-2 is able to distribute 73%**
  - the ideal protocol distributes 85%
  - flooding distributes 53%
  - gossiping distributes only 38%



SPIN

- Overall assessment
  - Focus on **efficient dissemination** of sensor data to data sinks and **energy conservation** at the sensors
  - Employs two key innovations: *negotiation and resource-adaptation*
  - Introduces *meta-data* as descriptors for negotiations
  - Each sensor has a resource manager for monitoring resources
  - Exchanging meta-data is **more efficient** than exchanging data
  - Polling the resource manager allows for **extensive energy savings** of sensors

Directed Diffusion for WSN

- Motivation for algorithm design
  1. **Robustness** of communication
  2. **Scaling** for high nubmers of nodes
  3. **Energy efficienct** network operation

- Example of operation:
  - “How many pedestrians do you observe in the geographical region X?”
  - “In what direction is that vehicle in region Y moving?”

Directed Diffusion for WSN

- Example of operation:
  - The operator's query will be transformed into an **interest that is diffused** toward nodes in regions X or Y (broadcast, geographical routing)
  - Nodes **activate their sensors** which begin collecting information about pedestrians
  - Information returns **along the reverse path** of interest propagation
  - Intermediate nodes **might aggregate the data**

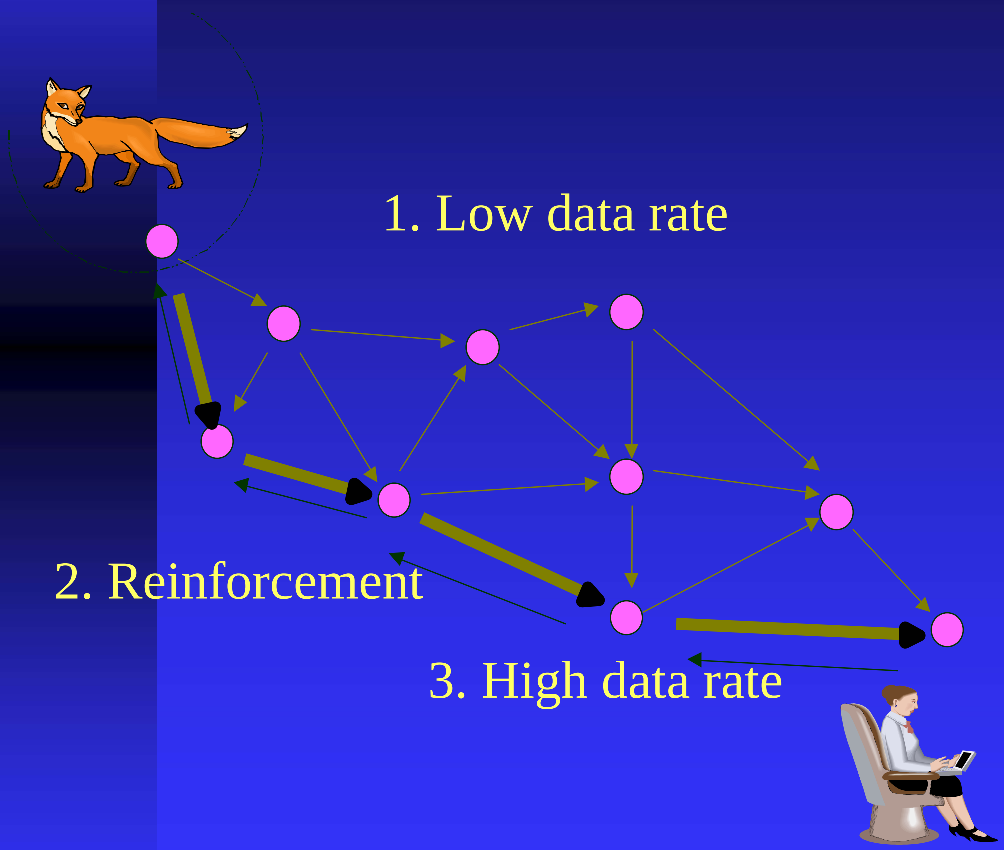
Directed Diffusion for WSN

- Directed Diffusion elements:
  - Algorithm based on
    - Interests
    - Data messages
    - Gradients
    - Reinforcements
  - Sinks request data by sending *interest messages*
  - Each interest contains *a description of a sensing task* for acquiring data
  - Data is a *collection of events* or *processed information* of a physical phenomenon

Directed Diffusion for WSN

- Directed Diffusion elements:
  - Data is named using *attribute-value pairs*
  - The interest dissemination sets up *gradients within the network* designed to “draw” events
  - *A gradient direction state* is created in each node that receives an interest
  - Events *start flowing toward* the originators of interests along multiple gradient paths
  - The sensor network *reinforces one* or *a small number* of these paths

Directed Diffusion for WSN



- Key features
  1. Interests dissemination
  2. Gradients setup
  3. Reinforcement of one or more gradient paths

Directed Diffusion for WSN

- Naming for a vehicle tracking example

Interest Naming

{type = wheeled vehicle;

interval = 20 ms;

duration = 10 s;

rect = [-100, 100, 200, 400] }

Data Naming

{type = wheeled vehicle;
interval = truck;
location = [125; 220];
intensity = 0:6;
confidence = 0:85;
timestamp = 01 : 20 : 40}

Directed Diffusion for WSN

- An example of path Reinforcement
  - initial interest: { type = wheeled vehicle; interval = 1 s; rect = [-100, 200, 200, 400]; timestamp = 01 : 20 : 40; expiresAt = 01 : 30 : 40}
  - A possible rule: **Reinforce any neighbor from which a node receives a previously unseen event**
  - the sink resends the original interest: { type = wheeled vehicles; interval = 10 ms; rect = [-100, 200, 200, 400]; timestamp = 01 : 22 : 35; expiresAt = 01 : 30 : 40}

Directed Diffusion for WSN

- Differences w.r.t. IP-based networks
  - diffusion is data-centric
  - all communication in diffusion is neighbor-to-neighbor (not end-to-end)
  - sensor nodes do not need to have globally unique identifiers (no IP address required)
  - every node can cache, aggregate, and more generally, process messages (no servers for performing such tasks)

Directed Diffusion for WSN

- Directed Diffusion characteristics
  - All communication is for *named data*
  - Data is named by *attribute-value* pairs
  - Intermediate nodes may *aggregate data*
  - Thus achieving significant energy-savings
  - Propagation and aggregation procedures are based on local information, gained by *localized interactions*

DD is capable of realizing **robust**, **multi-path, energy-efficient data delivery** in WSNs