Lecture

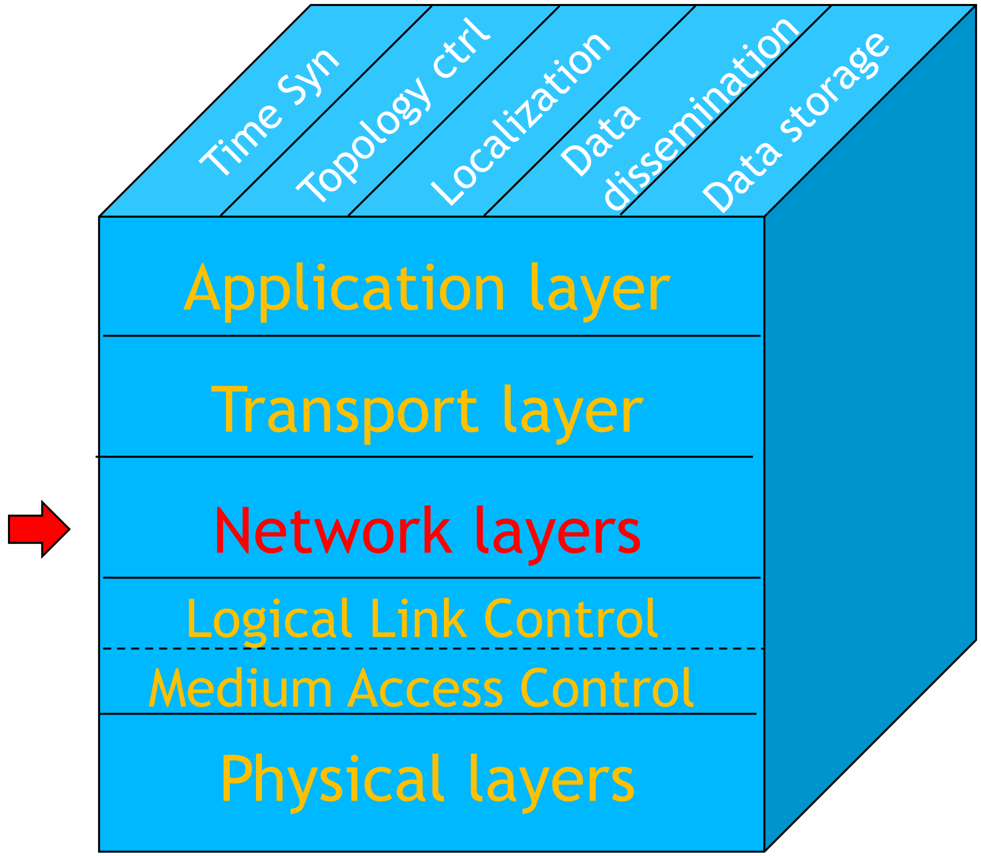
WSN Networking
- Data Centric and Content Based Networking

Qi Zhang

(Email: qz@eng.au.dk; Edison 314)

AARHUS SCHOOL OF ENGINEERING

Relevant topics in WSN



Outline

- **Some basics of Routing in WSNs**
- Data centric networking
  - Motivation and basics
  - SPIN
  - Directed Diffusion
- Data Aggregation

Goal of Routing Protocols in WSN

- Distributed
- Efficient (low overhead)
- Self-configuring
- Resilient (to changing network topology)

ID-centric routing

- Given: a network/a graph
  - Each node has a unique identifier (ID)
- Goal: Derive a mechanism that allows a packet sent from an arbitrary node to arrive at some arbitrary destination node
  - The routing & forwarding problem
  - Routing: Construct data structures (e.g., tables) that contain information how a given destination can be reached
  - Forwarding: Consult these data structures to forward a given packet to its next hop
- Challenges
  - Nodes may move around, neighborhood relations change
  - Optimization metrics may be more complicated than "smallest hop count" – e.g., energy efficiency

WSN routing protocols

- Because of challenges, standard routing approaches are not really applicable
  - Too big an overhead, too slow in reacting to changes
  - Examples: Dijkstra’s link state algorithm; Bellman-Ford distance vector algorithm
- Simple solution: Flooding
  - Does not need any information (routing tables) – simple
  - Packets are usually delivered to destination
  - But: overhead is prohibitive

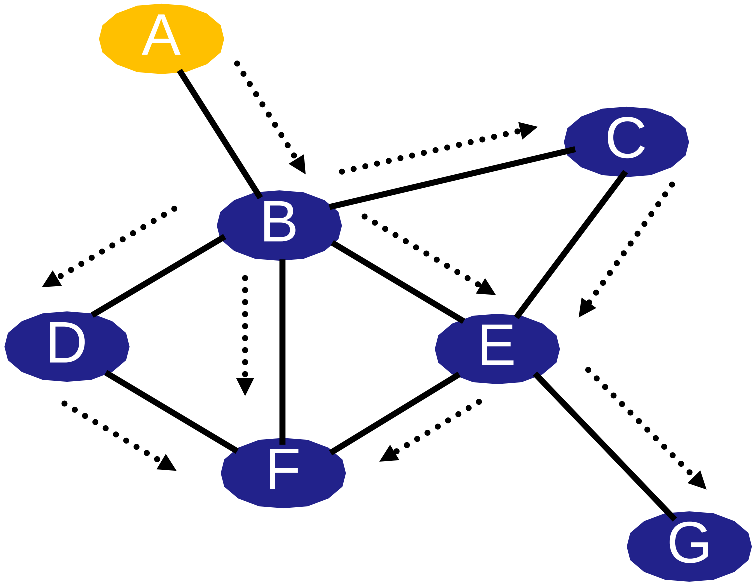
! Usually not acceptable, either

Need specific ***WSN routing protocols***

AARHUS SCHOOL OF ENGINEERING

Flooding and Gossiping

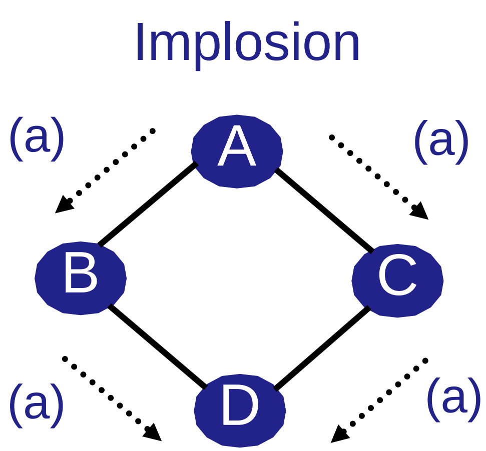
- Flooding
  - Sensor broadcasts every packet it receives
    - Only packets not seen earlier are forwarded
  - Relay of packet til the destination or maximum number of hops
  - No topology maintenance

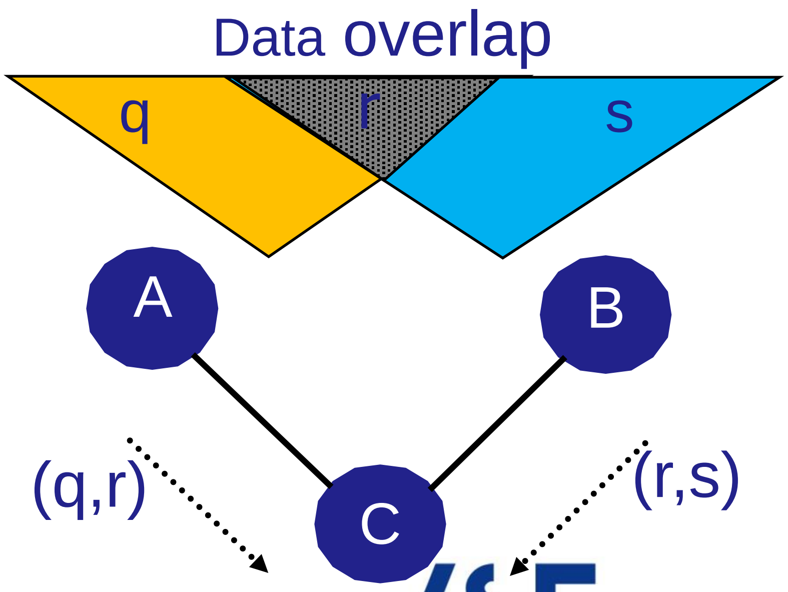


- "Gossip"
  - Enhanced version of flooding
  - Each node forwards a message with some probability
  - Randomized forwarding

Problems of Flooding and Gossiping

- Simple and reactive
- BUT have drawbacks:
  - Implosion
    - (NOTE: Gossiping avoids this by selecting only one node; but this causes delays to propagate the data through the network)
  - Overlap
  - Resource blindness
  - Energy inefficient





AARHUS SCHOOL OF ENGINEERING

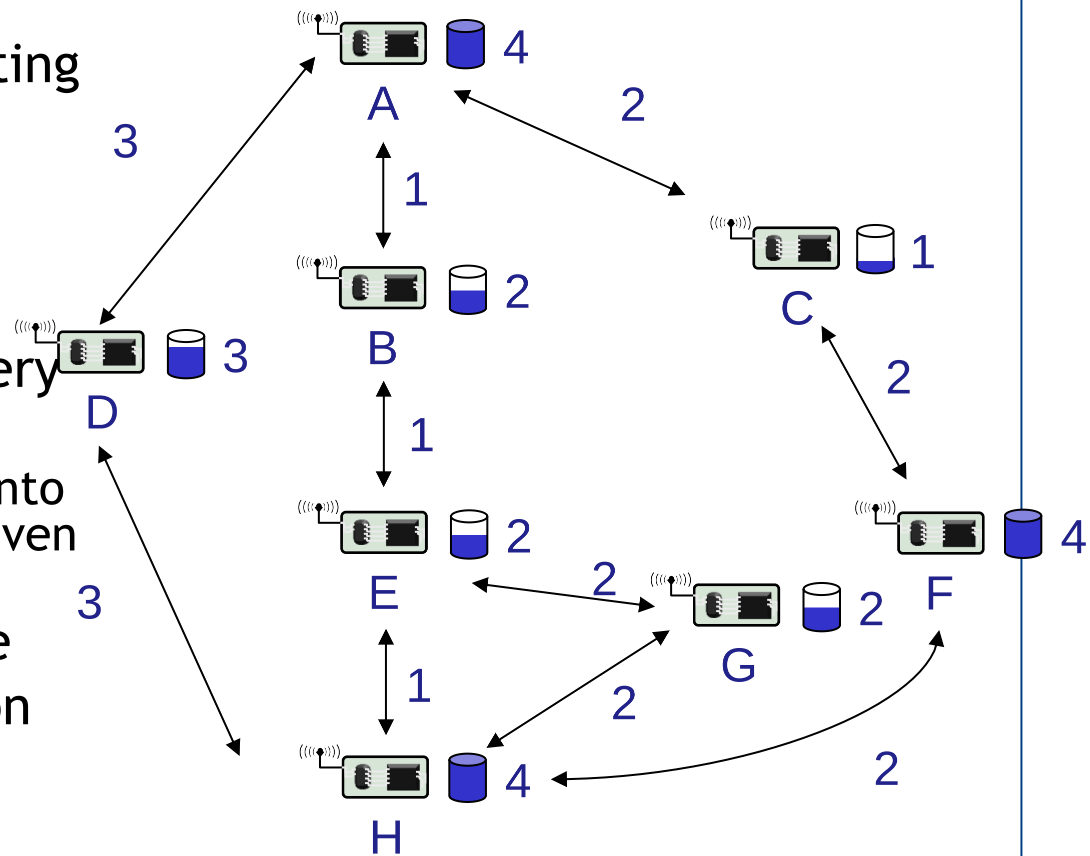
Energy-efficient unicast

- Particularly interesting performance metric: Energy efficiency
  - Assign cost value reflecting energy consumption
  - Pick an algorithm to compute the least-cost paths
- How do you define an energy-efficient path
  - E.g. Minimize energy per packet
- Standard traditional routing uses minimum hop count as metrics, in WSN proper link/path metrics are needed

Basic options for path metrics (Find route A->H)

- Minimize energy/bit, Example: A-B-E-H
- Minimum battery cost routing
  - Path metric: Sum of reciprocal battery levels

    Example: A-D-H
- Conditional max-min battery capacity routing
  - Only take battery level into account when below a given level
- Maximize network lifetime
- Minimum total transmission power
- Etc.



Example: Send data from node A to node H

Wireless multicast advantage

- Broad-/Multicasting in wireless is unlike broad-/multicasting in a wired medium
  - Wires: locally distributing a packet to *n* neighbors: *n* times the cost of a unicast packet
  - Wireless: sending to *n* neighbors can incur costs
    - As high as sending to a single neighbor – if receive costs are neglected completely
    - As high as sending once, receiving *n* times – if receives are tuned to the right moment
    - As high as sending *n* unicast packets – if the MAC protocol does not support local multicast

If local multicast is cheaper than repeated unicasts, then ***wireless multicast advantage*** is present

AARHUS SCHOOL OF ENGINEERING

Broadcast incremental power (BIP)

- How to broadcast, using the wireless multicast advantage?
  - Goal: use as little transmission power as possible
- Idea: Use a minimum-spanning-tree-type construction

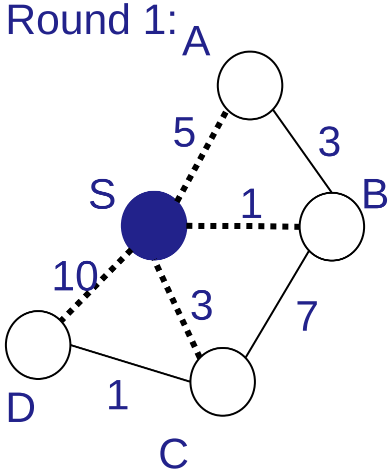
- Note: Once a node transmits at a given power level & reaches some neighbors, it becomes *cheaper* to reach *additional* neighbors

BIP

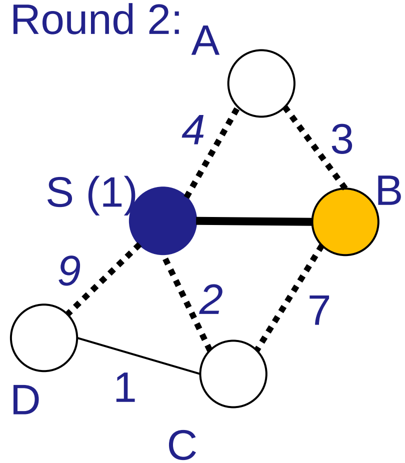
- Basic algorithm
  - One node is added to the tree per round
  - Each “as-yet-not-added” node maintain a “candidate edge”
  - In each round, the node with the lowest-cost candidate edge is chosen.
  - The candidate edge weight computation: the additional cost is the actual edge weight subtracts the currently used transmission power
- From BIP to multicast incremental power (MIP):
  - Start with broadcast tree construction, then prune unnecessary edges out of the tree

BIP - Example

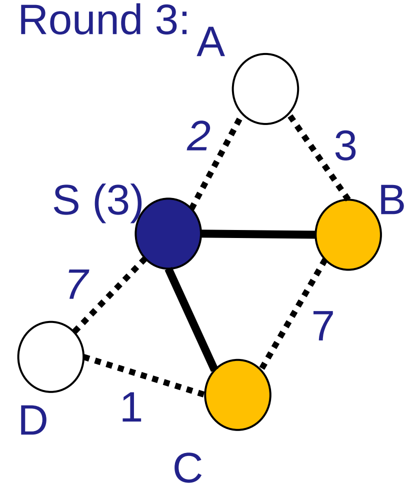
Round 1:



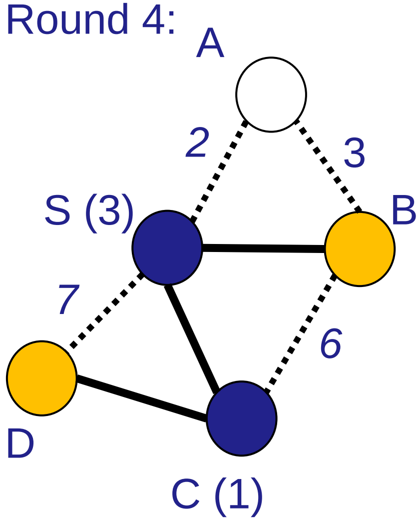
Round 2:



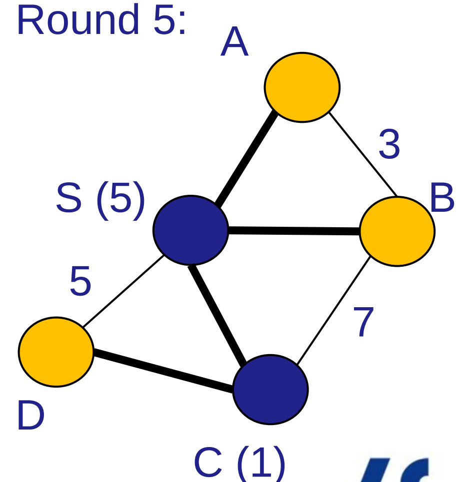
Round 3:



Round 4:

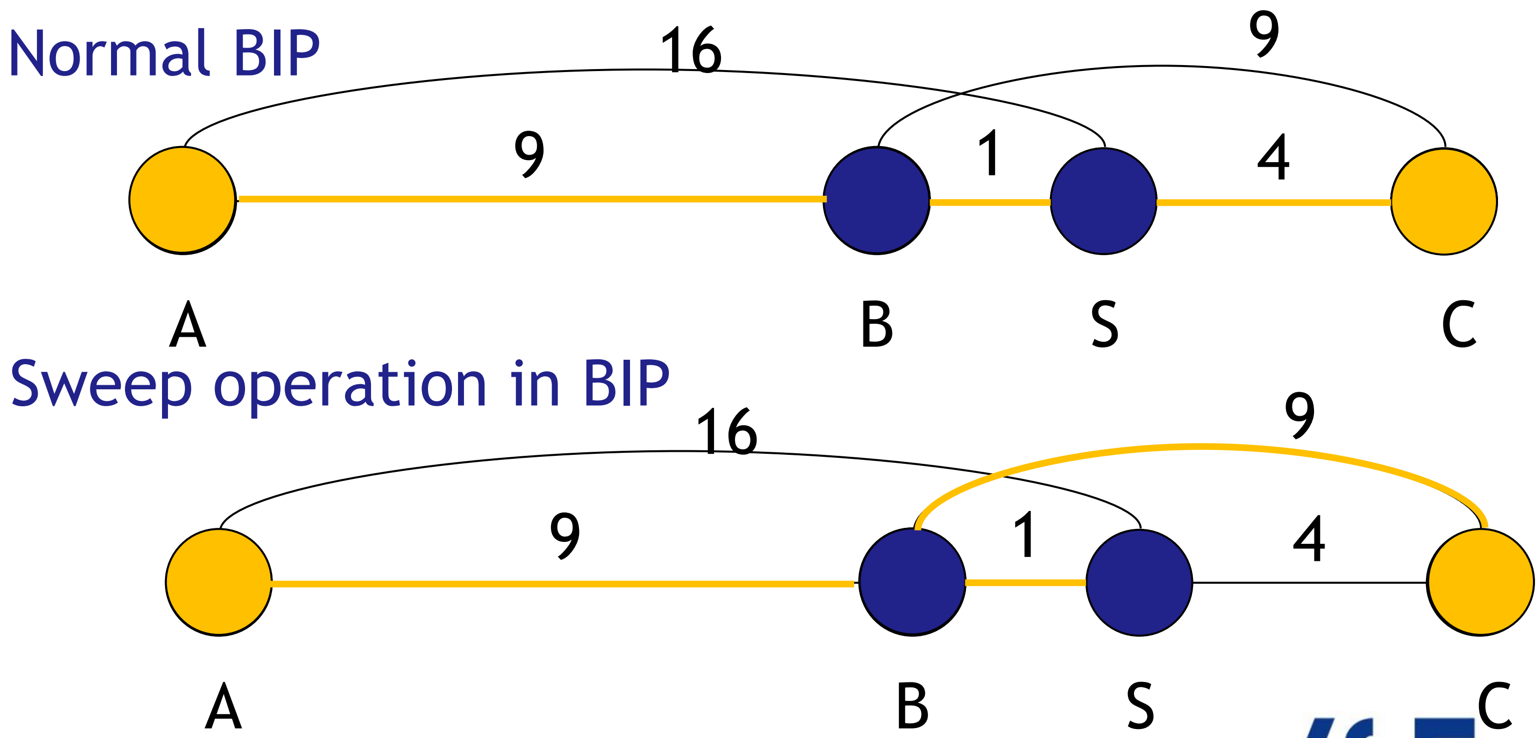


Round 5:



Example of opportunity for "Sweeping" in BIP

- A=(0,0) B=(3,0) S=(4,0) D=(6,0)
- Transmission cost is assumed to be proportional to the square of the distance (i.e. path loss exponent of 2)



Outline

- Some basics of Routing in WSNs
- **Data centric networking**
  - **Motivation and basics**
  - SPIN
  - Directed Diffusion
- Data Aggregation

Motivation of Data Centric Networking

- In WSN, meaningful data traffic is generated due to
    1. The sensors’ response to a query from the user (i.e., sink or BS)
    2. Sensors actively reporting a detected event
- Very large number of sensors → traditionally *impossible to assign specific IDs.*
    - Note: with IPv6, ID assignment is not an issue anymore.
- To overcome this challenge, some routing protocols gather/route data based on the *description of the data, i.e. data centric*

What is DATA CENTRIC?

- Data-Centric
  - Sensor node does not need an identity!!!
    - Sink is not interested in the node ID, e.g. sink doesn’t usually ask “What is the temp at node #27 ?”
  - Data are named by attributes
    - Where are the nodes whose temperature have 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?

What is Data-Centric Networking?

- Naming the data by attributes because the users are more interested in querying an attribute of the phenomenon rather than querying an individual node.
- Construct routes based on the *interest* of query message

- Example:

  "the areas where the temperature is over 70C"

  is a more common query than

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

AARHUS SCHOOL OF ENGINEERING

Outline

- Some basics of Routing in WSNs
- **Data centric networking**
  - Motivation and basics
  - **One-shot interactions: SPIN**
  - Repeated interactions: Directed Diffusion
- Data Aggregation

SPIN:
Sensor Protocol for Information via Negotiation

- Uses three types of messages: ADV, REQ, and DATA.

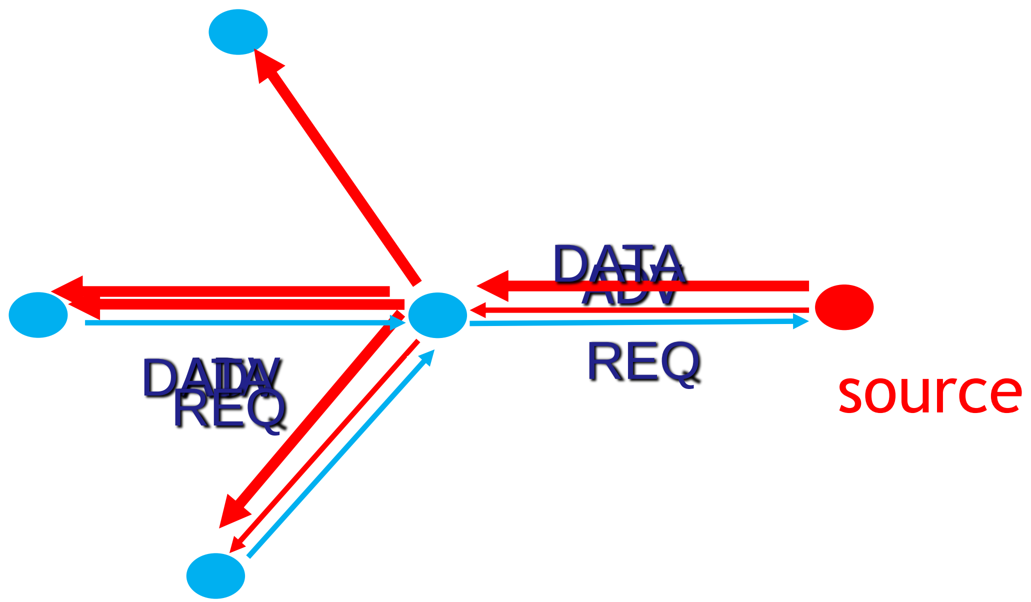
- ADV – data advertisement
  - Node that has data to share can advertise this by transmitting an ADV with meta-data attached (i.e., the name of the data)
- REQ – request for data
  - Node sends a request when it wishes to receive some actual data
- DATA – data message
  - Contains actual sensor data with a meta-data header
  - Usually much bigger than ADV or REQ messages

Basic procedure in SPIN

- When a sensor node has something new, it broadcasts an advertisement (ADV) packet that contains the new data, i.e., the meta data.
  - New data: either by local measurement or from some other nodes
- If meta data is as yet unknown, the interested nodes send a request (REQ) packet.
- DATA packets is sent to the nodes that send REQ 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



Family of SPIN Protocols

- SPIN-PP
  - For point-to-point transmission media
- SPIN-EC
  - Similar to SPIN-PP but with energy conservation heuristics added to it
- SPIN-BC
  - Designed for broadcast transmission media. 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-EC (Energy conservation)

- Basically Spin-EC= SPIN-PP + low-energy threshold
  - Modifies behavior based on current energy resources
- Adds simple energy-conservation heuristic to SPIN-PP
- When energy is plentiful, SPIN-EC behaves like SPIN-PP
- Reduce participation of node when approaching low-energy-threshold
  - When node receives new data, it only initiates protocol if it can participate in all three stages with all neighbor nodes
  - When node receives ADV, it does not request the data
- Node still exhausts energy below threshold by receiving ADV messages

AARHUS SCHOOL OF ENGINEERING

SPIN-RL Protocol

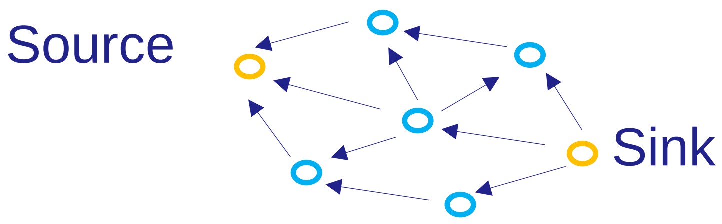
- SPIN-RL = SPIN-BC+ reliable transmission
- Needs knowledge about single-hop network neighbors
- Adaptation for lossy networks
  - Compensate lost ADV messages by re-advertising periodically
  - Compensate lost REQ/DATA by re-requesting after fixed time (i.e., if a node sends DATA to other nodes, it needs to wait for some period of time before responding to any further request of the same piece of data)
- Adaptation for mobile networks
  - Topology changes trigger updates to the neighbor lists of nodes
  - When a node's neighbor list is changed, the node re-advertises all its data

Outline

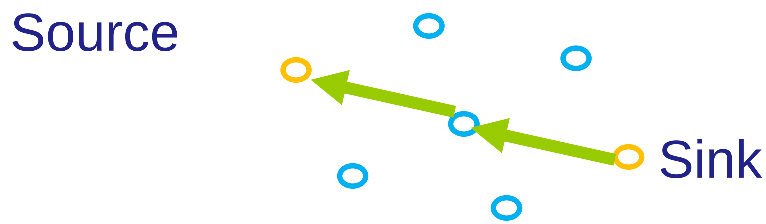
- Some basics of Routing in WSNs
- **Data centric networking**
  - Motivation
  - One-shot interactions: SPIN
  - **Repeated interactions: Directed Diffusion**
- Data Aggregation

Directed Diffusion- Two phase pull

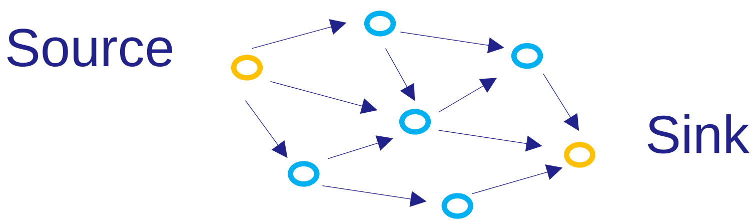
- One possible realization of publish/subscribe for WSN
- The service pattern is subscription to data sources that will publish data at a selectable data rate over a selectable duration.



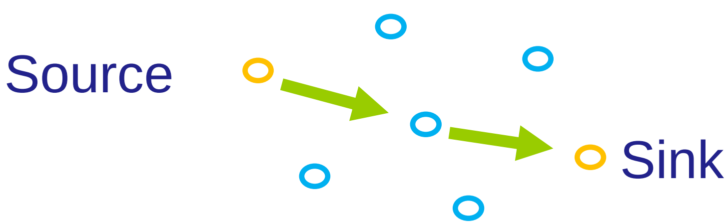
1- Interest Propagation



3- Reinforcement



2- Gradient Establishment



4- Data Delivery

AARHUS SCHOOL OF ENGINEERING

Basic Concepts in Directed Diffusion

- Naming Scheme
  - *Data* is named using attribute-value pairs
- An *interest* message is a query which specifies what a user wants.
  - Interest: a description of the sensing task
- An *event* is a short description of the sensed phenomenon.
- *Gradient* is a direction state created in each node that receives an *interest*.
  - The gradient direction is set toward the neighboring node from which the *interest* is received.
- Reinforcement
  - Sink reinforces particular neighbors to draw higher quality (higher data rate) events

Naming scheme

- Request: *Interest* (Task) Description (Using attribute-value pairs)
  - 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)
- Data Reply in response to interests (Using attribute-value pairs).
  - 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)

How *interests* are diffused through the sensor network?

- The sink periodically broadcasts an *interest* to sensor nodes to query information from a particular area in the field.
  - This initial interest contains a *much larger interval* attribute.
  - The initial interest is called *exploratory* which tries to determine if there indeed are any sensor nodes that detect the specific type.
  - The interest is *soft state* that will be periodically refreshed by the sink. The sink simply resends the same interest with a monotonically increasing timestamp attribute.

How *interests* are diffused through the sensor network?

- 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 *interest entry* in the cache has several fields, e.g.
  - A timestamp field: It indicates the timestamp of the last received matching interest.
  - Several gradient fields, up to one per neighbor:
    - Each gradient contains a data rate field requested by the specified neighbor

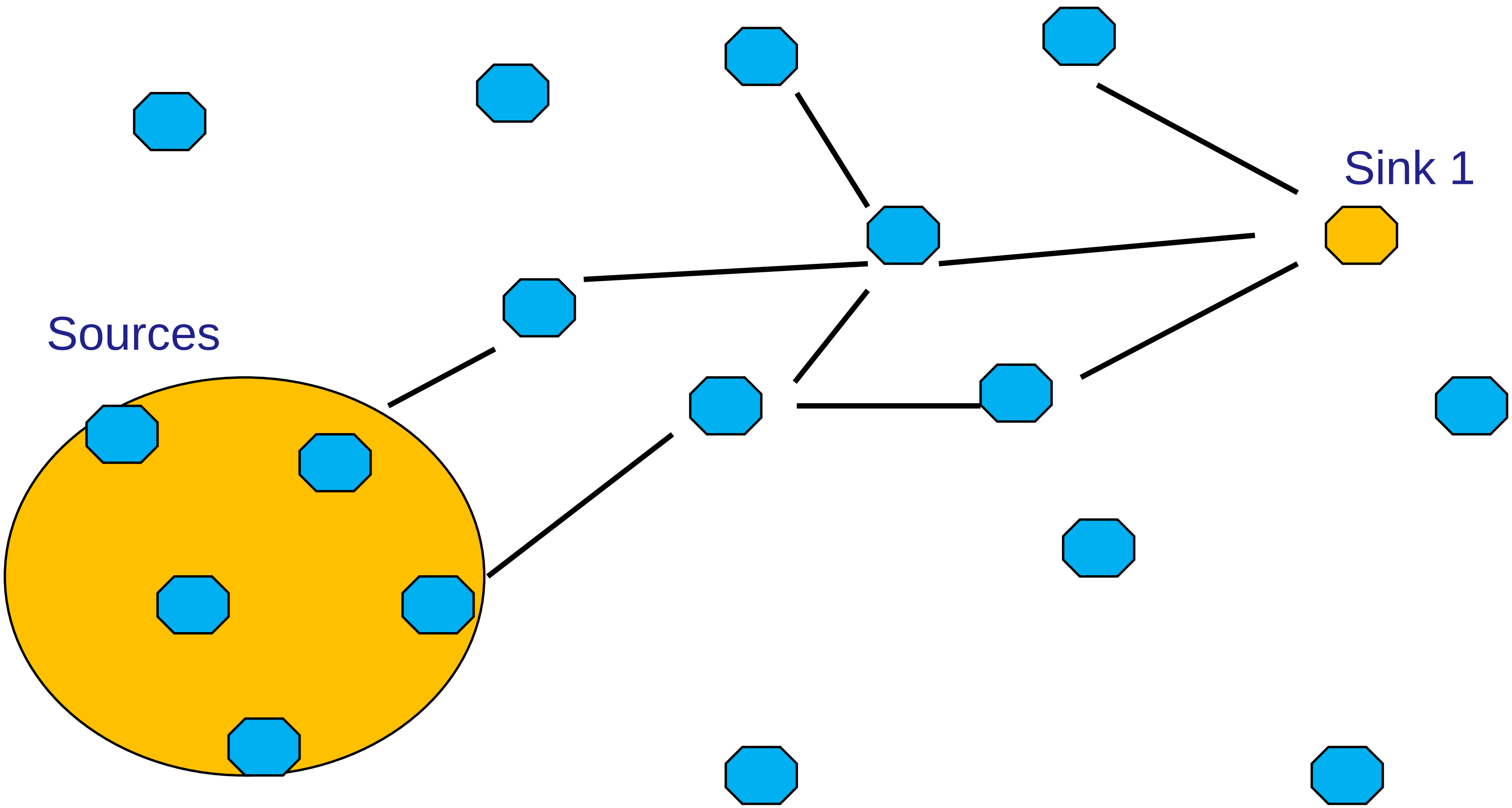
Interest Propagation and Gradient Establishment

- 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 toward the neighbor who sent this interest
    - Gradient specifies the direction in which to send events and data rate, i.e., Gradient: a weighted reverse link
  - If there is an interest entry in the cache, aggregation on the interests and update of the gradient fields
- The interest entry is removed from the cache when it has expired.

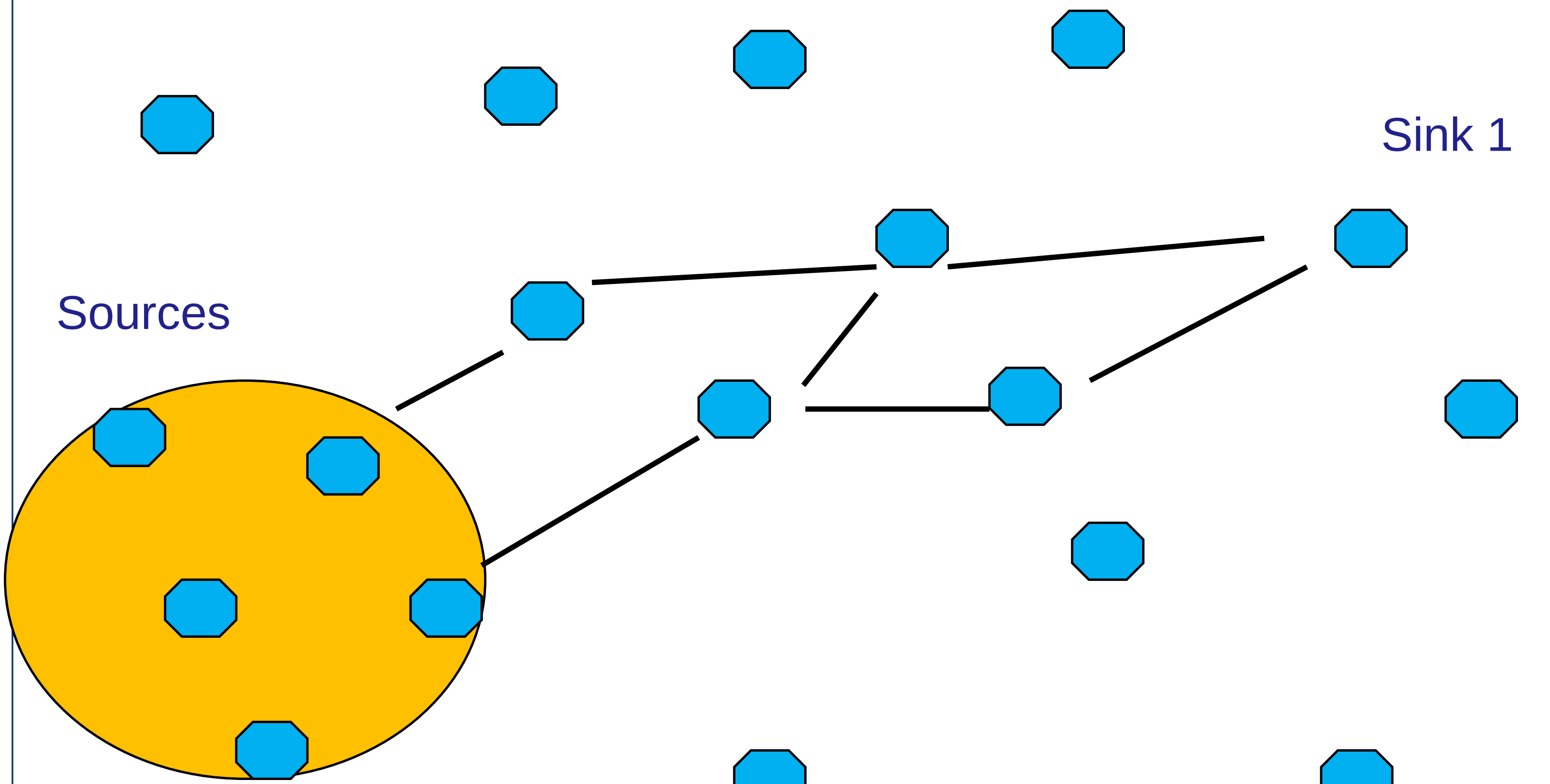
Cont…

- Re-send interest to a subset of its neighbors
  - This is essentially flooding-based approach
  - Other probabilistic, location-based and other intelligent forwarding approaches possible
  - A node may suppress a received interest if it recently has re-sent a matching interest.

Interest Propagation



Gradient Establishment



Local Rules

- Local Rules for Propagating Interests
  - Just flood interest
  - More sophisticated techniques possible:
    - Directional interest propagation based on cached aggregate information

- 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.

AARHUS SCHOOL OF ENGINEERING

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;
  - 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
- Each intermediate node forwards the reinforcement packet to its next hop based on the same principle;
  - each selected neighbor forwards this new interest to a subset of its neighbors by selecting a smaller set of paths
- 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

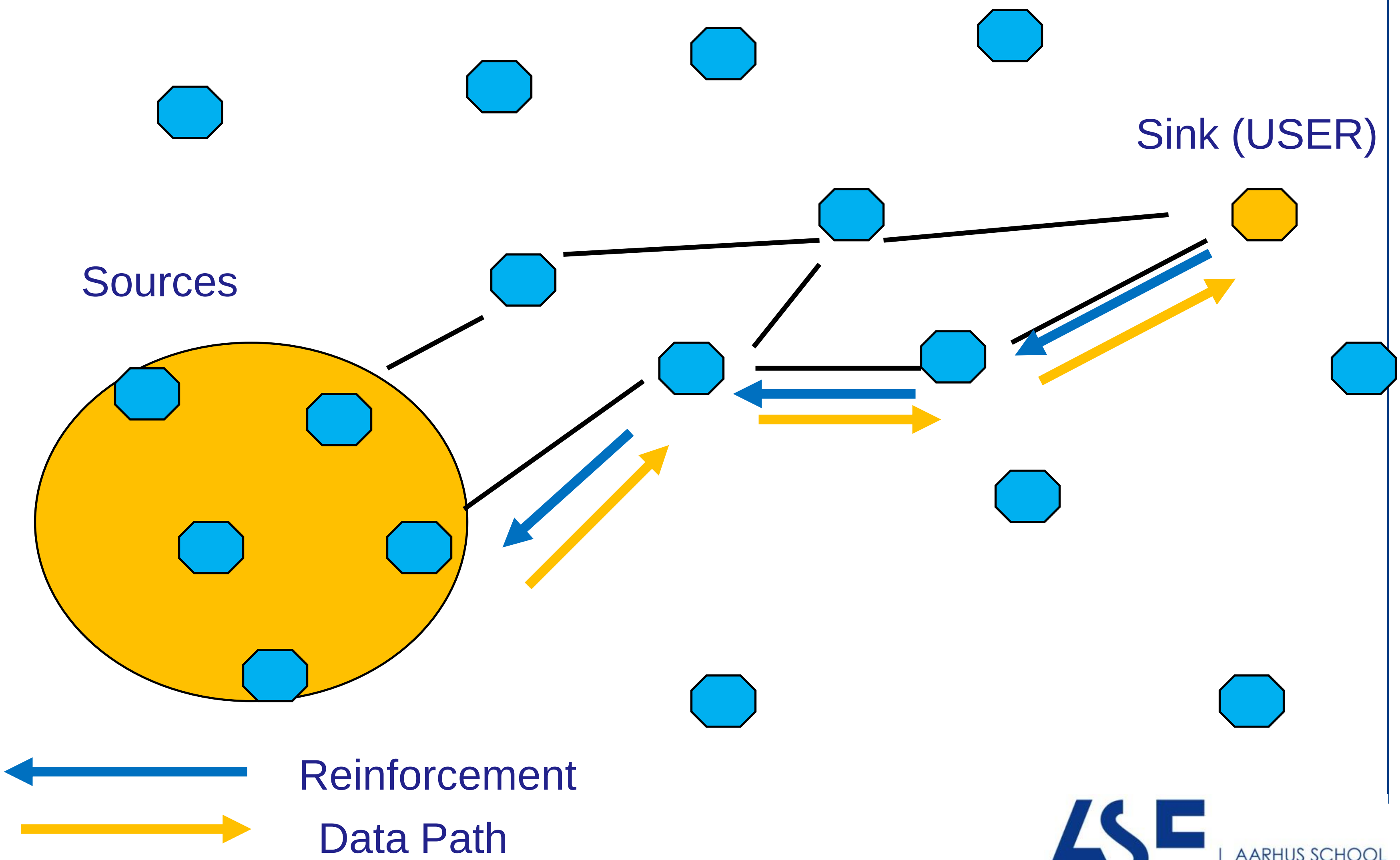
AARHUS SCHOOL OF ENGINEERING

Data Propagation

- Data will be sent based on the data rate from sources towards the sink based on the established data path.
  - If all gradients have a data rate > = the rate of incoming events
    - The node may simply send the received data message to the appropriate neighbors.
  - Else
    - the node may down convert the data message to the appropriate gradient which has the lower data rate

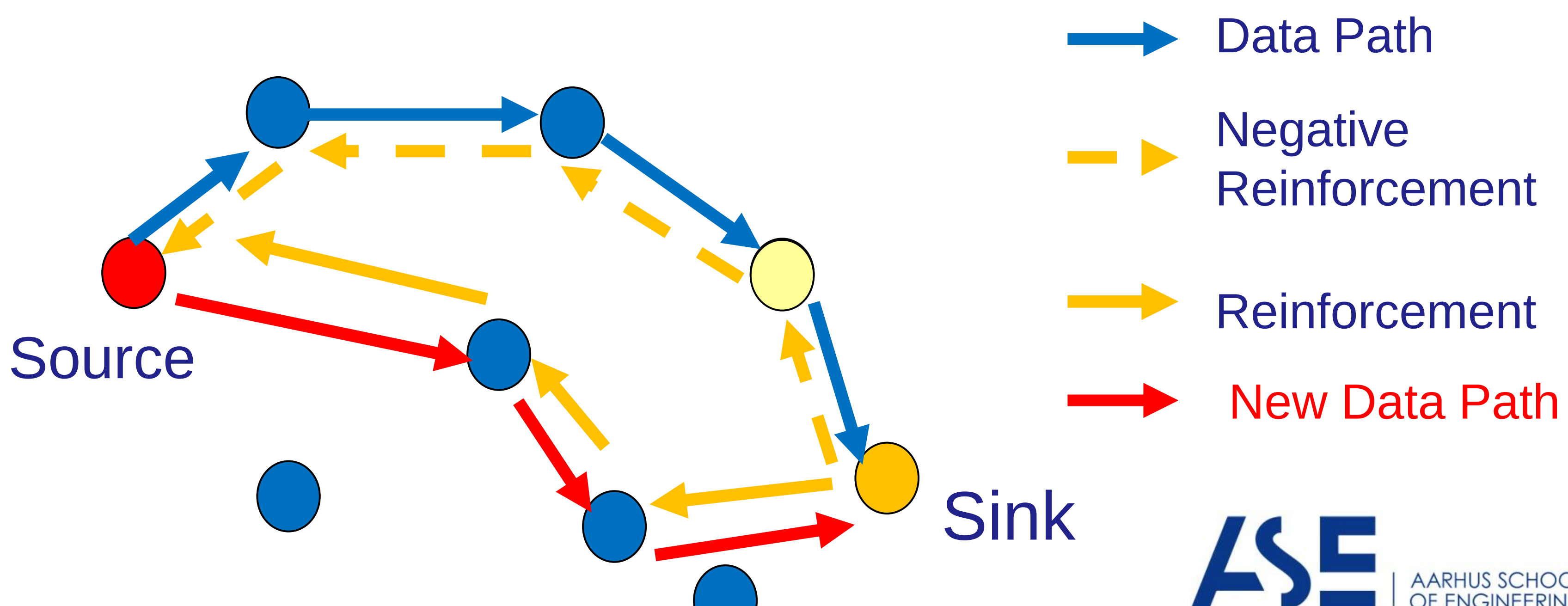
- Each intermediate node forwards the data to its next hop neighbor.

Reinforcement and Data Propagation



Negative Reinforcement

- Two mechanisms:
  - Soft state (implicit approach): Time out all data gradients in the network unless they are explicitly reinforced.
  - Explicitly degrade the path by re-sending a negative reinforcement message which is an interest with lower data rate.



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

Directed Diffusion vs. SPIN

- In directed diffusion:
  - 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.

Outline

- Some basics of Routing in WSNs
- Data centric networking
  - Motivation
  - SPIN
  - Directed Diffusion
- **Data Aggregation**

ASE | AARHUS SCHOOL OF ENGINEERING

Data aggregation – Motivation

- Redundancy and correlation
  - A certain degree of overlap and redundancy is created as measured sensor data is often generated by nearby nodes
  - Measured data can be expected to be highly correlated allowing further improvements of the information quality by using data fusion approaches (possibly exploiting further available meta information)

- Energy constraints and network congestion
  - Data transmission in sensor networks is much more energy expensive compared to local computation efforts
  - The reduced number of transmitted messages towards the base station helps reducing network congestion (especially near the base station)

Data aggregation - Motivation

- Impossible to continuously collect raw sensor data
  - Limited memory and bandwidth
  - Information overload
  - => Data aggregation in storage.

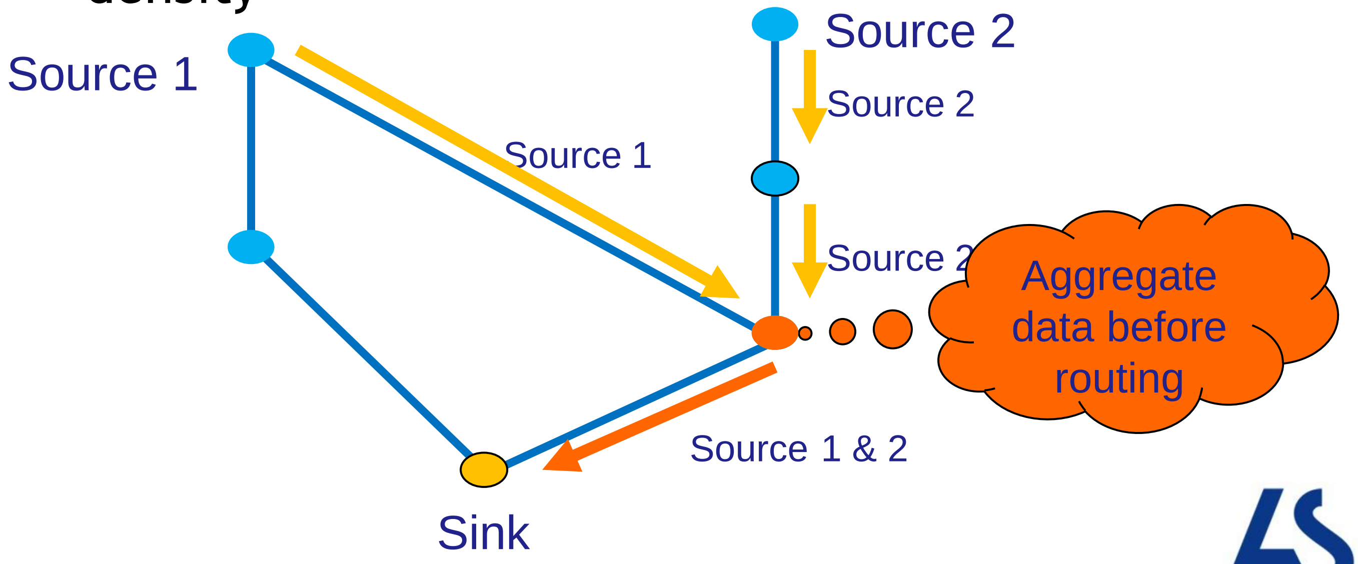
- Individual sensor readings are of limited use:
  - User is interested in collective information rather than individual sensor data
  - Exploit in-network processing

- Goal:
  - To save energy and increase network lifetime by combining multiple sensor data

AARHUS SCHOOL OF ENGINEERING

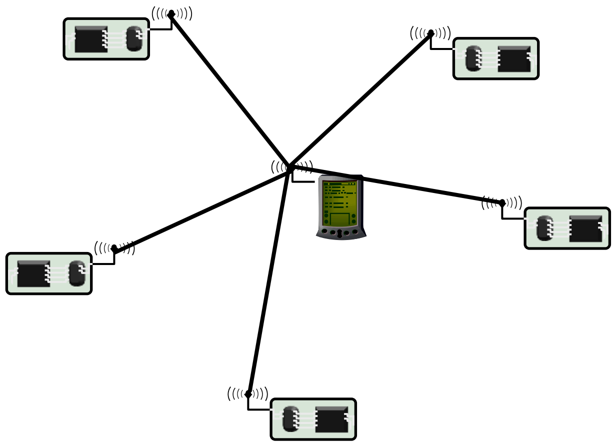
Data aggregation

- Data Aggregation:
  - Process of combining data or information to estimate or predict events, i.e., computing a smaller representation of the messages that is equivalent in its content to all the individual messages.
- Idea:
  - Take advantage of the routing hierarchy and high network density

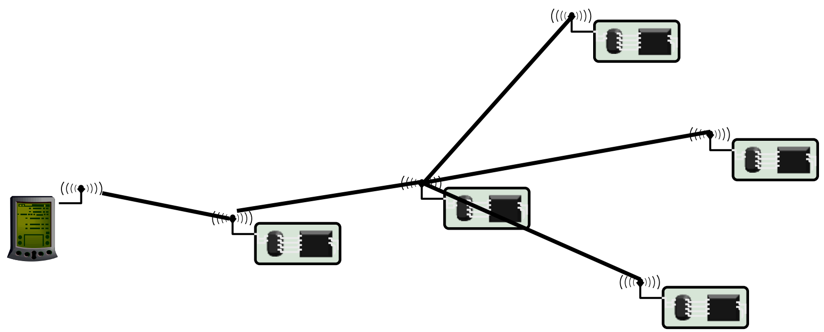


Data aggregation

- The benefit of aggregation depending on the location of data sources and sinks
  - aggregation can be *useful* or *pointless*



Pointless case



Useful case

Metrics for the efficacy of 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 that are included in computing the final aggregate at the sink
- ***Latency***
- ***Message overhead***

AARHUS SCHOOL OF ENGINEERING

Data Aggregation Components

DATA STORAGE
Store sensor data in a memory efficient way, while preserving the accuracy of the information.

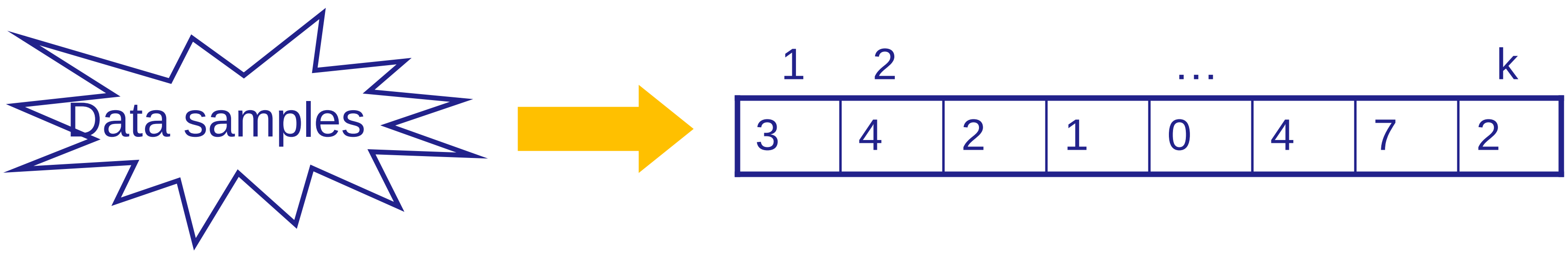
AGGREGATION FUNCTIONS

AGGREGATION PATHS
Which are the optimal aggregation points on the path from sensors to sink?
Which is the most suitable path from source to sink to favor data aggregation?

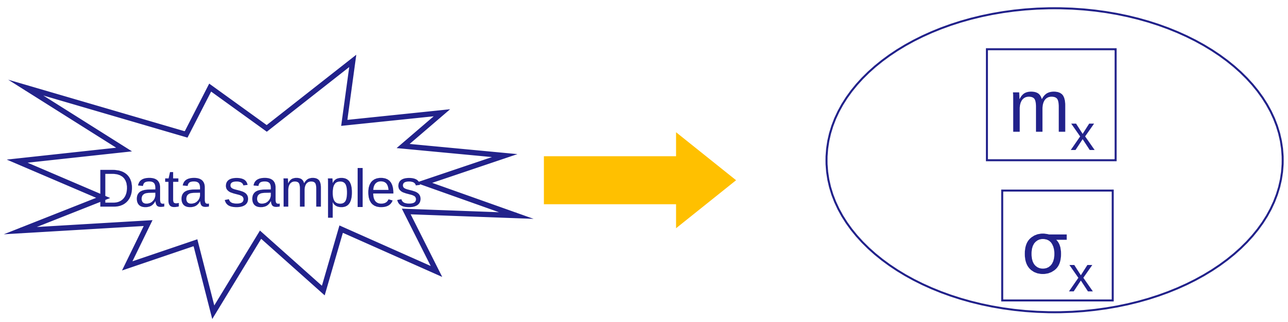
Data Storage Representations

- Data can be represented with different degree of accuracy.

Store individual data → histograms and lists



Store only mean and variance or other statistical representations



AARHUS SCHOOL OF ENGINEERING

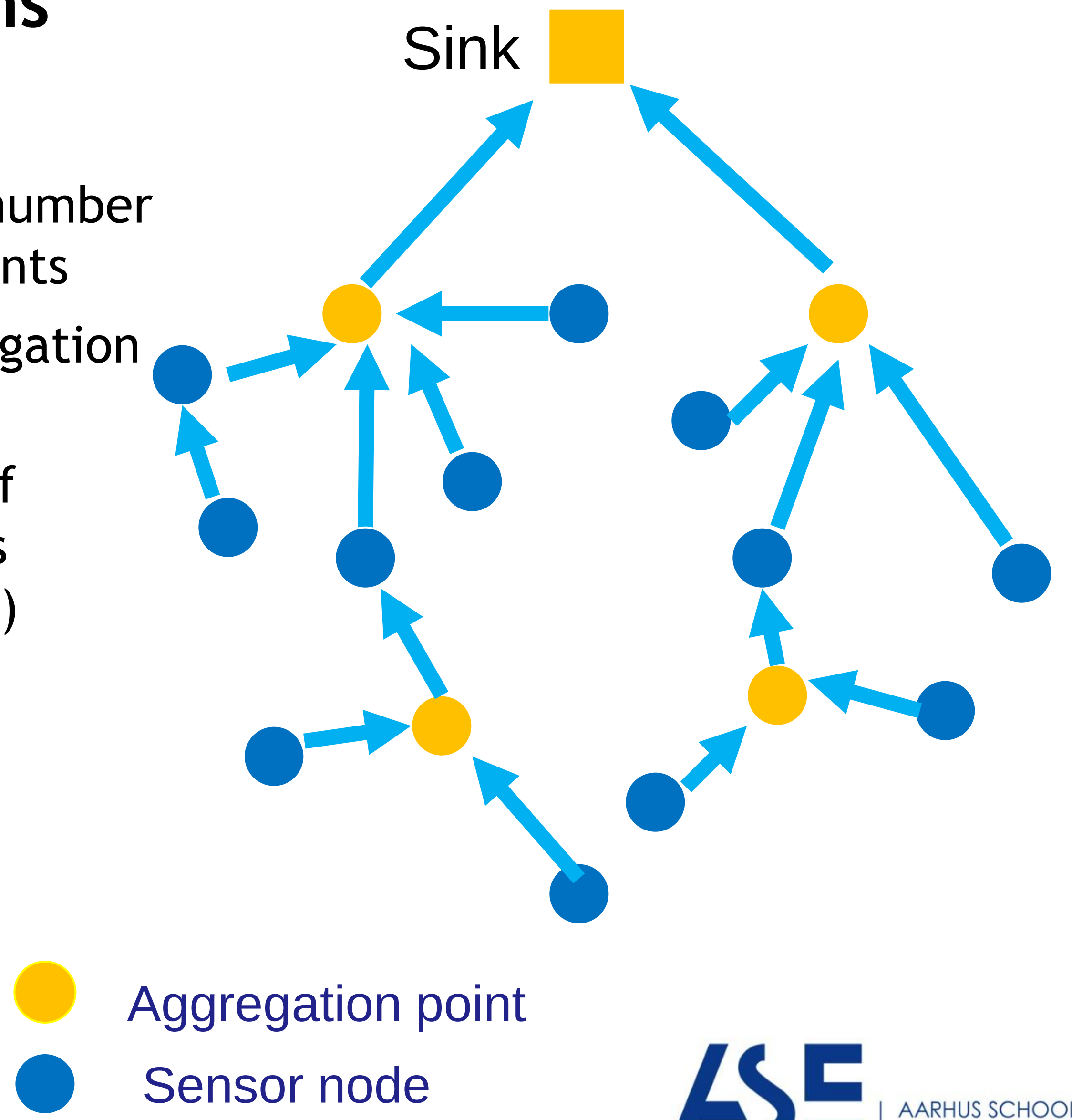
Aggregation Functions

- Very simple functions
  - Average, max, min, median
  - Suppression of duplicates
- More sophisticated functions
  - Exploit spatial and temporal correlation
  - Signal processing (convolution, filtering, etc.)

ASE | AARHUS SCHOOL OF ENGINEERING

Aggregation Paths

- Challenges:
  - Find the optimal number of aggregation points
  - Selection of aggregation points
  - Dynamic change of aggregation points (energy efficiency)



Challenges in Aggregation

- Inherently unreliable environment, certain information unavailable or expensive to obtain
  - How many nodes are present?
  - How many nodes are supposed to respond?
  - What is the error distribution (in particular, what about malicious nodes?)
  - Trying to build an infrastructure to remove all uncertainty from the application may not be feasible

AARHUS SCHOOL OF ENGINEERING

Optimal Data Aggregation

- Optimal Data Aggregation is *NP-Hard*
- Sub-optimal Algorithms:
  - Opportunistic
    - Just aggregate when possible
  - Center at Nearest Source (CNS)
    - The nearest source acts as the aggregation point
  - Shortest Paths Tree (SPT)
    - Sources send using shortest path if able to aggregate
  - Greedy Incremental Tree (GIT)
    - Recursively select the closest source to the tree
  - Etc.

Data Aggregation Pros & Cons

- Pros:
  - In network processing reduces traffic load by aggregating data in route
    - Energy and memory efficiency
  - Scalable to large numbers of both sinks and sensors

- Cons:
  - Requires careful design to tradeoff accuracy and storage and message size
  - Incur information loss, making robust estimation more difficult:
    - E.g. a single outlier reading can damage MAX/MIN aggregates

Summary

- Data centric routing is a preferred routing alternative in WSN.

- Providing robust aggregation solution with resource constrained sensor node is challenging; however, aggregation or other advanced in-network processing is a key enabler for efficient networking

AARHUS SCHOOL OF ENGINEERING