5. Network layer

Network layer protocols, routing and routing protocols

Wireless sensor networks

Martin Úbl
ublm@kiv.zcu.cz

2023/24

<DCSE> DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

Network layer

Basics

- network layer – ISO/OSI layer 3
- we assume that:
  - data link layer performs its purpose
  - we have a list of neighbors (links, clusters, ...)
  - we are able to transmit and receive data, presumably through a queue
- network layer performs:
  - organizing nodes to a logical structure (logical / virtual topology)
  - routing packets between network segments
  - upper layer queries

Network layer

Basics

- WSN data transfer use cases mostly include:
  - gathering sensor data – routing from end node to base station
  - event-driven signaling – routing from end node to base station
  - collective queries – scattering query and gathering aggregated information from all end nodes to base station
- don't forget, that the actual data are not the only packets transfered
  - control packets of all kinds gets transfered between peer nodes, or between end node and base station
- to satisfy the three most frequent use cases, we need **data-centric protocols**

Network layer

Basics

- **data-centric protocols**
- assumes that there are many *data sources* and one or more *data sinks*
- *router* is a node that routes packets on L3
- *data sink* may be a base station, edge node, ...
  - basically any node that gathers the data
- all network nodes are *data sources*
- all network nodes are (potentially) *routers*
- **convergecast**
  - type of transmission, that sends the data from the data source to a data sink
  - potentially: from all data sources to a data sink
  - may be seen as a constrained variant of multicast

Network layer

Basics

- **hirearchical protocols**
- may base on data-centric protocols
  - data-centric protocols often forms a partially-connected mesh
  - e.g., to become hierarchical, one such approach performs mesh pruning to obtain a spanning tree with data sink as a root
- establishes a tree topology
- may work very well on partially-connected mesh physical topology
- may perform data "filtering" and pruning during operation

Network layer

Basics

- **geographical routing**
- makes use of a location when routing
- requires one of the following:
  - node localization techniques
  - signal strength assessment
  - a composite metric for distance evaluation
- performs well in large-scale networks
- often contains more data sinks

Network layer

Basics

- other types
  - QoS-based protocols
  - application-specific protocols
    - e.g., when there are many data sinks on all edges of the network, ...
  - physical topology-specific protocols
    - when there is no other option, e.g., when having the linear physical topology
  - etc.

Network layer

Data-centric protocols

- data-centric protocols examples:
  - flooding
  - directed diffusion
  - gossiping (rumoring)
  - gradient-based routing
  - SPIN-based procotols
  - etc.
- each of the above have some characteristic principle we will mention

Network layer

Requirements

- basic requirements on L3 protocols:
  - conserve power
  - respond to topology change / dynamic reconfiguration
  - data aggregation
  - scalability
  - addresation
  - robustness

Network layer

Requirements

- *addresation*
- does every node need an unique address?
- in a normal network, we have BOOTP, DHCP, ...
- we don’t have any of the above in WSN
- although we may implement one, it poses high overhead
- different techniques
  - local addressing
  - retain L2 (MAC, HW) address
  - do not use address at all – use a protocol that does not need an address

Network layer

Requirements

- *scalability*
- prefering scalable protocol with distributed approach
- nodes should not know the whole topology
- this allows for a great scalability
- this requirement is crucial, as the WSN may contain hundreds of nodes
- we may also want to add additional nodes during WSN operation

Network layer

Requirements

- *robustness*
- nodes may fail – network must adapt
- every node must be able to function as a router
- no network should depend on a single node
  - if the node fails, the network must reorganize
- no network should depend on a single packet transmission
  - if the packet gets lost, we must be able to retransmit

Network layer

Requirements

- *topology*
- depends on the protocol
- deterministic topology $\rightarrow$ more effective routing
- non-deterministic topology $\rightarrow$ usually faster adaptation
- we should not be using absolute data
  - when possible, use relative – relative location, relative distance, ...
- routing protocols must consider a neighborhood discovery phase
- sometimes nodes may migrate / fail
- routing protocol must adapt

Network layer

Flooding

- **flooding**
- simple data-centric protocol
- every packet has time-to-live (TTL)
  - TTL decreases with each hop
- every packet might have a unique identifier (some variants does not have it)
  - may be a pair of source address and sequence number
- the source node broadcasts the packet to all neighbors
- every node decreases the TTL and broadcasts the incoming packet to all neighboring nodes except the source
- forwarding stops when:
  - TTL reaches 0
  - packet reaches the destination node

Network layer

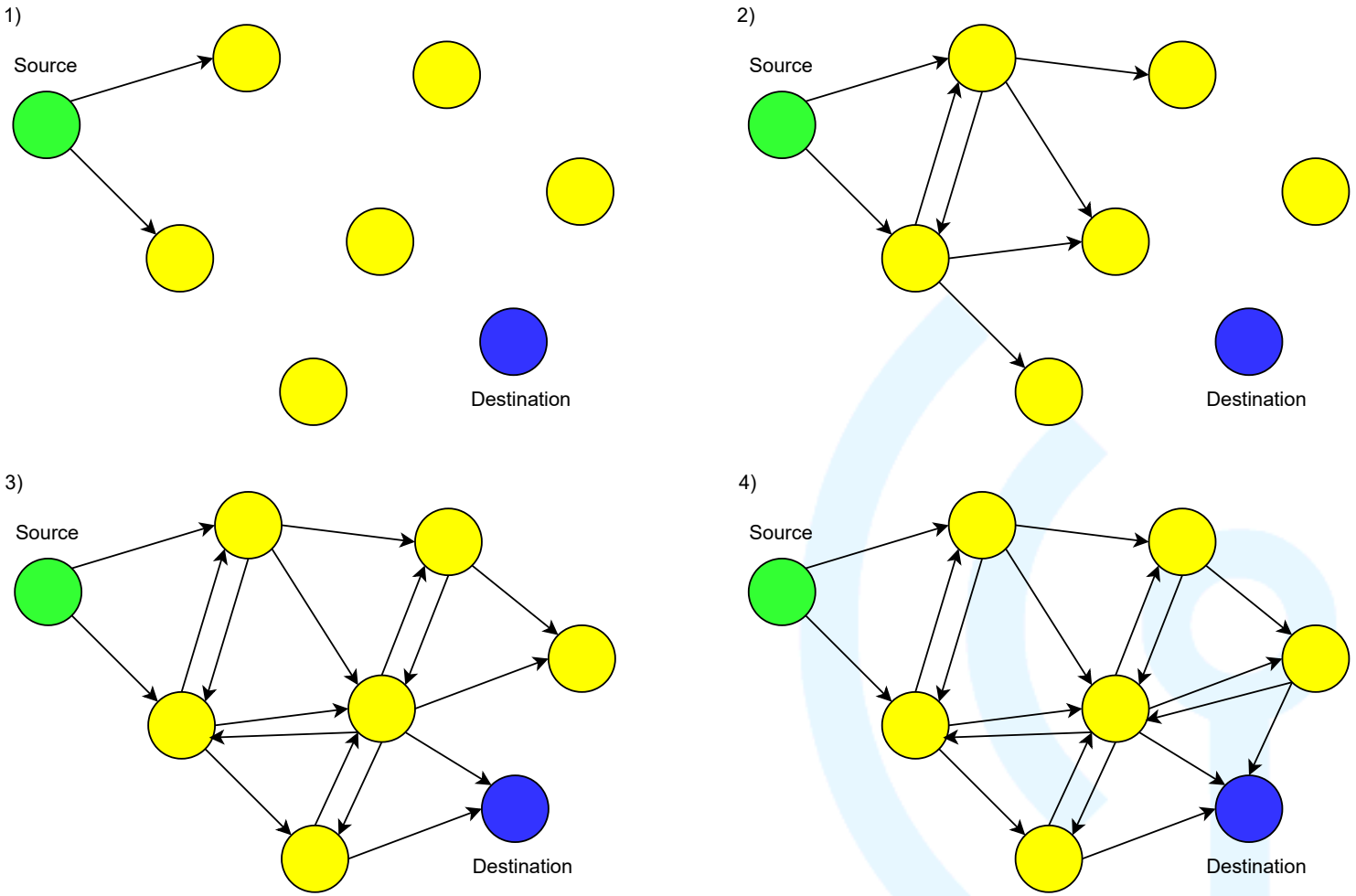
Flooding

- flooding
- is a reactive protocol
  - does not maintain a routing table
- does not require neighbor discovery – relies on L2 information/broadcast
- not an energy-aware protocol
- packet reaches destination from many sides
- implosion problem (packets get multiplied during transfer)
- overlapping problem (duplicate packets have a different source – cannot be distinguished without addressing)

Network layer

Flooding

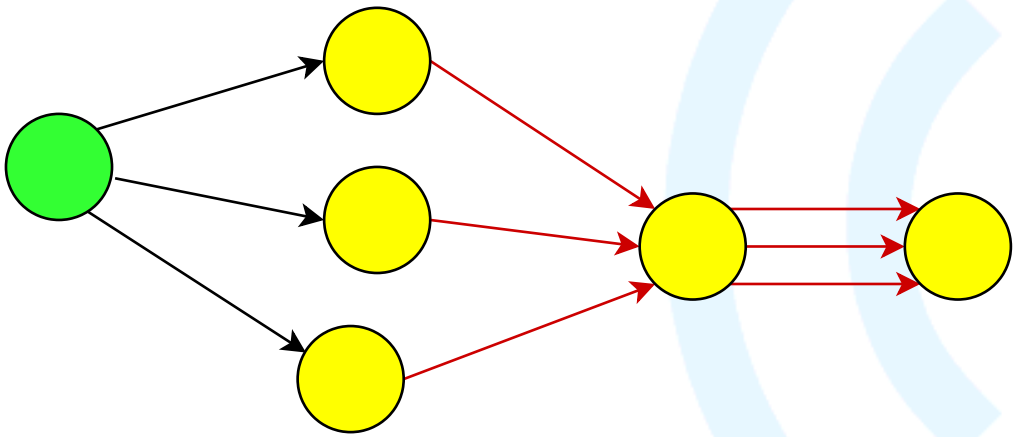
- example of flooding



Network layer

Flooding

- implosion and overlapping problem
- packet gets duplicated, each duplicate arrives from a different node
- the targeting node (router) has no means of differentiating between them
- must forward all of them



Network layer

Flooding

- flooding does not choose an optimal route
- well... one of the routes is the shortest one
- prone to power drain attacks
- amplifies traffic on certain topologies
- can be useful either way
  - often used as an initialization phase of another protocol
  - when the actual neighborhood is unknown

Network layer

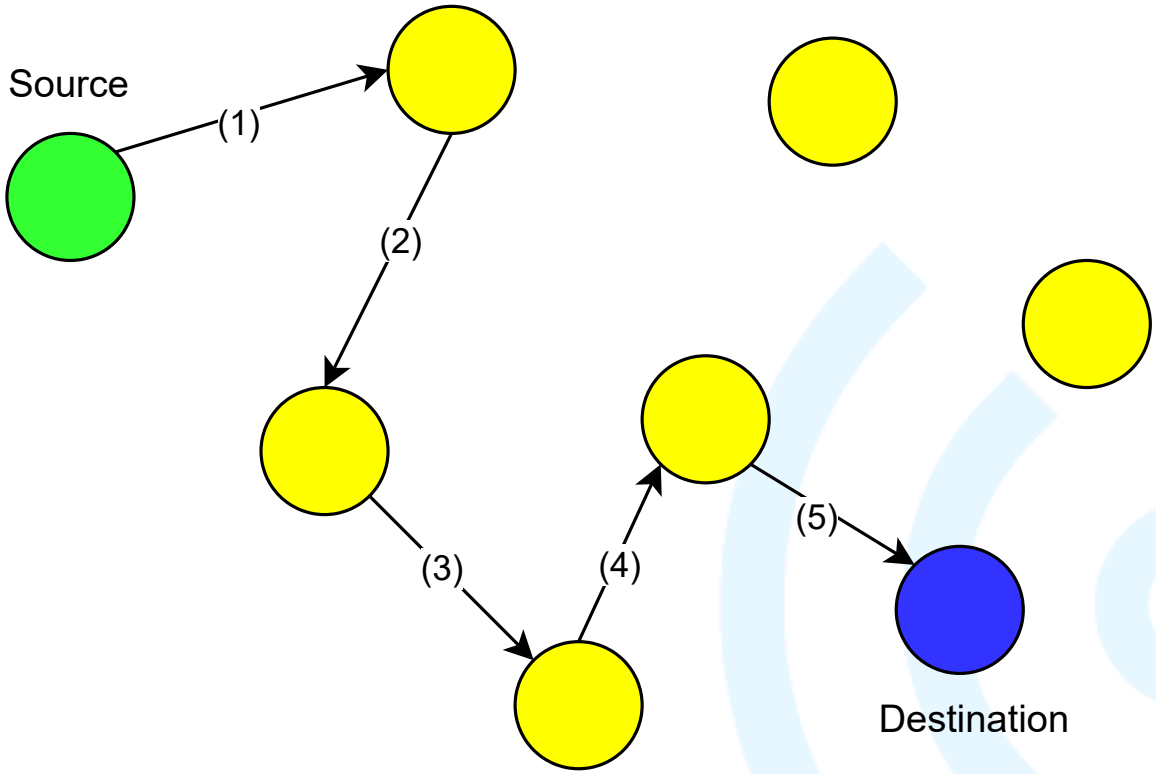
Gossiping

- **Gossiping** or **rumoring** (gossip or rumor routing)
- based on flooding
- instead of all neighbors, the node selects one at random
  - excluding the source, as in flooding
- may seem utterly nondeterministic, however, it performs well on networks with low traffic
- protocol either does not use TTL, or uses much higher value
- very simple and convenient for mesh topologies

Network layer

Gossiping

- example of gossiping



Network layer

Gossiping

- nothing guarantees optimal route
- nothing guarantees delivery
- statistically very successful
  - eventually reaches the target
- reasonably power-conserving
- no implosion/overlapping
- no need for packet de-duplication

Network layer

Gossiping

- gossiping serves as a base for more protocols
- very simple to implement
- some added heuristics for data sink routing (convergecast)
  - introduces routing table with heuristically obtained "metric"
- some considered two neighbors instead of just one
  - convenient for sparsely-connected meshes
- and more

Network layer

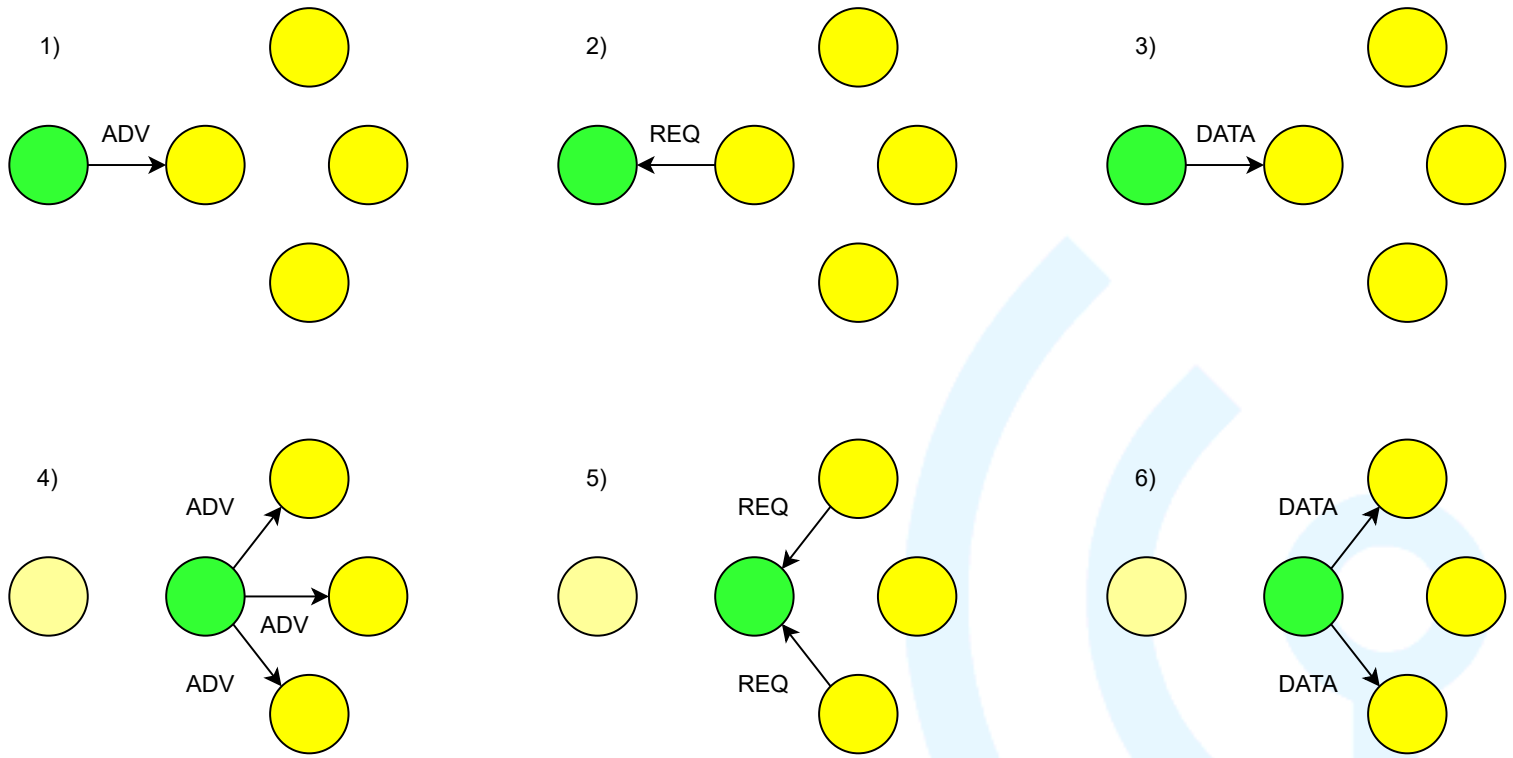
SPIN

- **Sensor Protocol for Information via Negotiation** (SPIN)
- data-centric group of protocols
- routes from data sources to data sinks
- three packet types:
  - ADV – advertisement of meta-data
  - REQ – request for data
  - DATA – actual data
- more of a base technique, than a protocol by itself

Network layer

SPIN

- SPIN example exchange



Network layer

SPIN

- SPIN-PP
- PP stands for Point-to-Point
- ADV is sent via broadcast
- REQ and DATA sent via point-to-point (unicast)
- node may choose not to respond to ADV
  - for example when it already possesses the data (identified by meta-data sent in ADV)
  - or when the memory is full, or the node is busy

Network layer

SPIN

- SPIN-EC
- EC stands for Energy-Conserving
- modification of SPIN-PP
- the node does not respond to ADV, if it detects the energy is below a certain threshold
- may conserve energy by not participating in packet routing

Network layer

SPIN

- SPIN-BC
- BC stands for BroadCast
- ADV and DATA packets are sent via broadcast
- if at least one node responds with REQ, the data are sent
  - nodes randomize delays before REQ is sent
  - senses carrier wave
  - if any node transmits REQ, it detects it and switches to receiving mode
- convenient, when there are many nodes potentially sending REQ
  - and thus, in SPIN-PP, the message would be retransmitted many times

Network layer

SPIN

- SPIN-RL
- RL stands for ReLiable
- if the node, that sent REQ did not receive the DATA to a given timeout, it sends REQ to the originating node once more

Network layer

Directed diffusion

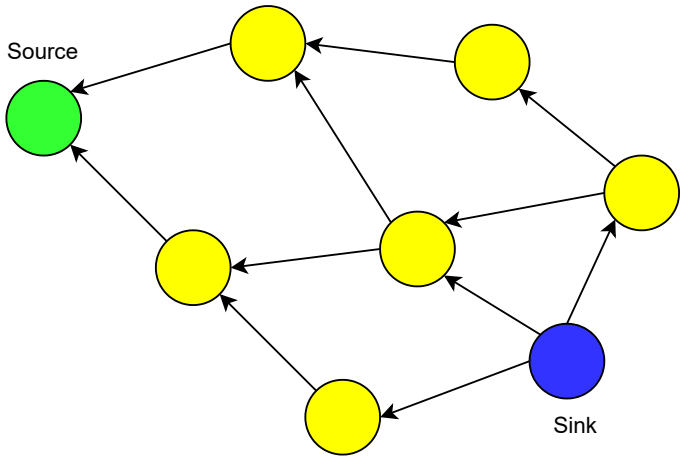
- **directed diffusion**
- a base station (or user) requests a specific information from a node
- four protocol states:
  1. interest propagation
  2. gradient setup
  3. reinforcement
  4. data delivery
- broadcasts periodically
- uses flooding for initial interest propagation

Network layer

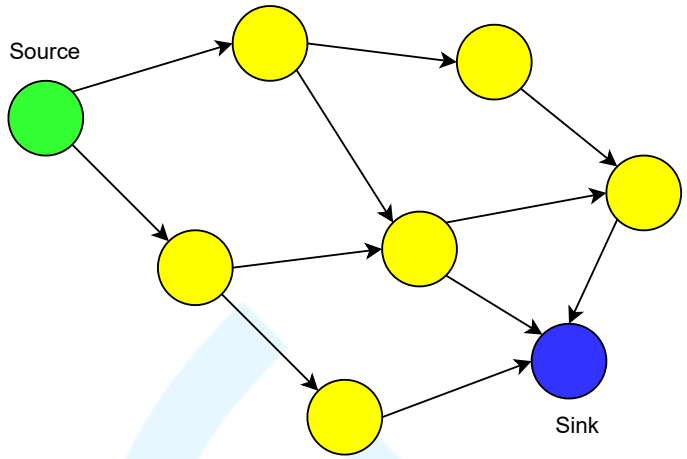
Directed diffusion

- example directed diffusion setup flow

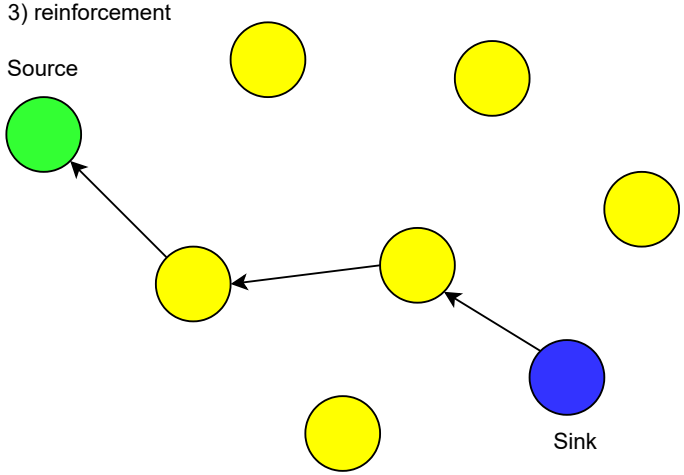
1) Interest propagation



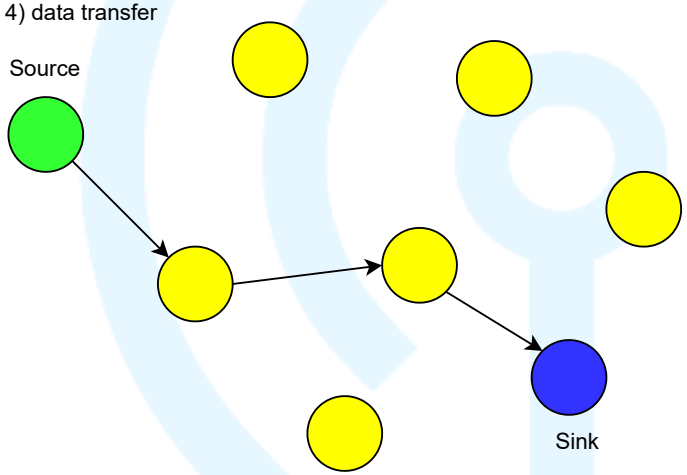
2) gradient setup



3) reinforcement



4) data transfer



Network layer

Directed diffusion

- gradient is built using many strategies
  - first node that sent the "interest message"
  - highest energy node
  - lowest delay node
  - etc.
- each node stores for each message:
  - timestamp
  - gradient (source node)
  - interval
  - validity period

Network layer

Directed diffusion

- multiple paths
- selection by reinforcements
- reducing weight by negative reinforcements
  - e.g., when a removable loop is detected
  - or when some node fails and the path is no longer valid
- positive and negative reinforcements allows for topology change

Network layer

Hierarchical protocols

- **hierarchical protocols**
- nodes are clustered by protocol-specific strategies
- establishes a tree topology
- tree can be seen even on a cluster level
  - clusters are connected trough nodes (cluster heads)
- node types:
  - root – usually a base station, data sink node
  - cluster head – a cluster controller, connects with parent cluster
  - regular node – any other node that is not a root, nor a cluster head

Network layer

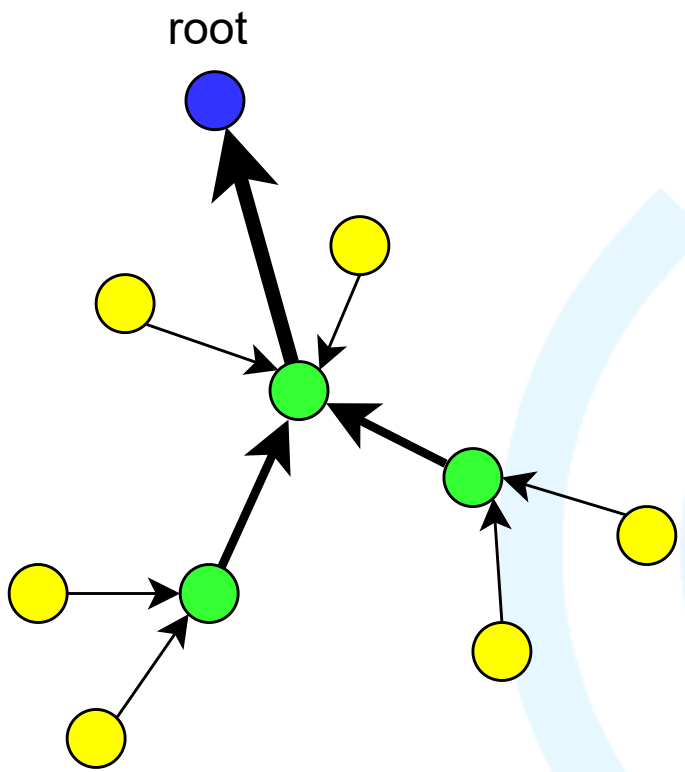
Hierarchical protocols

- *cluster head*
  - maintains the cluster
  - passes packets between clusters
  - is either elected or pre-configured
  - **aggregates data**
- *root*
  - maintains the network
  - very often a *data sink*
  - very often pre-configured
  - we often assume, that the root has unlimited energy

Network layer

Hierarchical protocols

- An example of hierarchical network
  - blue – root, green – cluster heads, yellow – regular nodes



Network layer

Hierarchical protocols

- there are several widely used protocols
  - LEACH
  - PEGASIS
  - TEEN, APTEEN

Network layer

LEACH

- Low-Energy Adaptive Clustering Hierarchy (LEACH)
- hierarchical protocol
- network segments elect cluster heads
- cluster heads communicate directly with base station (root)
- to balance energy drain, cluster heads are re-elected based on battery state

Network layer

LEACH

- operation runs in *rounds*
- each round has *phases*:
  1. cluster head election (optional)
  2. setup
  3. steady state
- round durations are application-specific
- often changes on a hourly basis or longer

Network layer

LEACH

- *cluster head election*
- node can become a cluster head only once per P rounds
  - P is also a number of cluster heads in the whole network
- every node calculates a probability of becoming a cluster head
- $T(n) = \frac{\frac{1}{p}}{1-\frac{1}{p}(r\%p)}$
- then the node generates a random number; if it is greater than the $T(n)$, the node declares itself a cluster head candidate
- each candidate broadcasts an advertisement message
- if there are more cluster head candidates, they vote

Network layer

LEACH

- *setup phase*
- each node, that is not a cluster head, sends a cluster membership message to the cluster head
- at the end of the setup phase, cluster head establishes a schedule
- schedule – a TDMA schedule for sending messages from node to cluster head and vice-versa
- thus, LEACH requires a synchronous TDMA L2 protocol

Network layer

LEACH

- *stead state phase*
- nodes respect the TDMA schedule
- nodes send their data to the cluster head
- cluster head aggregates, compresses and filters data
- cluster head sends the data to root (base station)
- after the round ends, re-election happens

Network layer

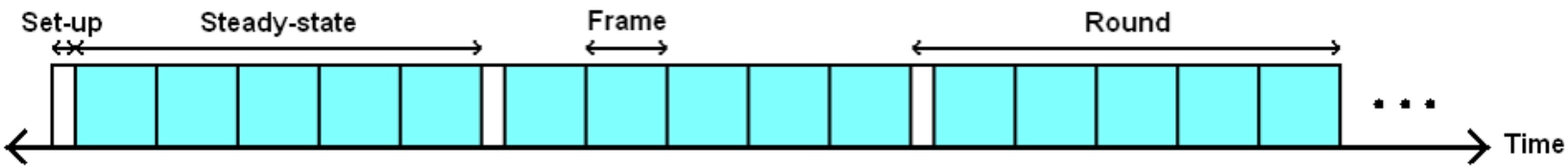
LEACH

- advatages:
  - power conserving
  - balanced battery usage
  - clustering with aggregation reduces traffic
- disadvantages:
  - each node must become cluster head each P rounds regardless of battery state
  - uneven distribution
  - may not be efficient for certain physical topologies

Network layer

LEACH

- a time diagram of LEACH phases



Network layer

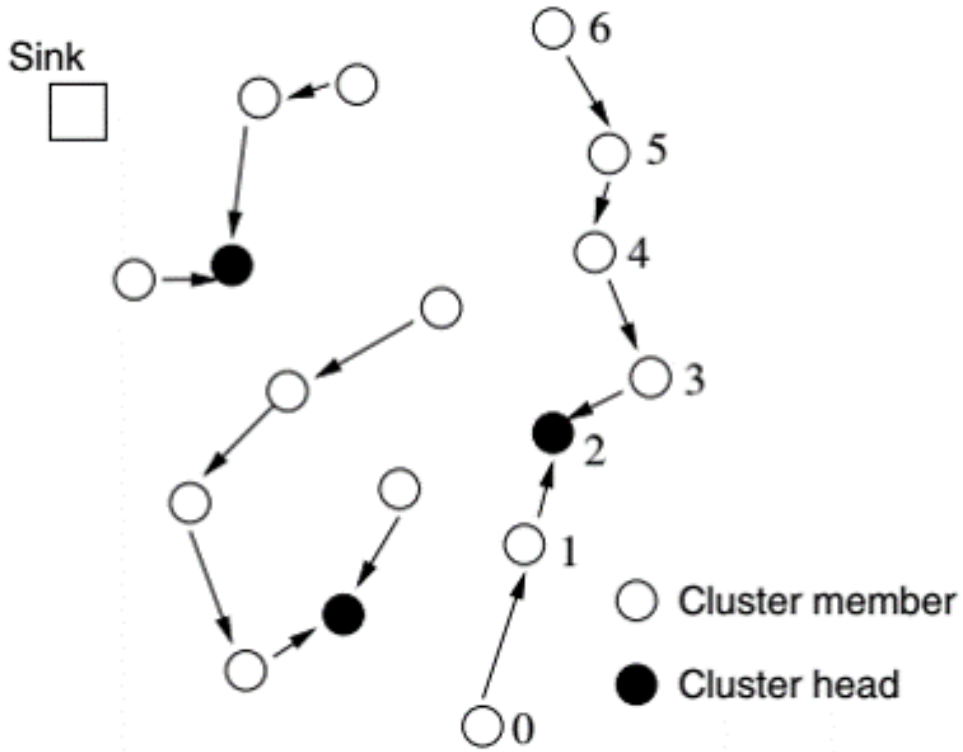
PEGASIS

- **Power-Efficient Gathering in Sensor Information Systems** (PEGASIS)
- instead of a star sub-topology in each cluster, it forms a linear chain
  - nodes do not send data directly to cluster head
  - nodes pass their data linearly, ends up in cluster head
- linear chain is bi-directional
- data are propagated from the furthest node to the nearest
- up to 300 % more effective than LEACH

Network layer

PEGASIS

- an example of PEGASIS node organization and topology



Network layer

TEEN

- **Threshold-sensitive Energy Efficient Network** (TEEN)
- multi-level hierarchy
- core principle: do not transmit data change, if the change is not significant
- two thresholds:
  - hard threshold T_H
  - soft threshold T_S
- if the measured value is larger than T_H, data are sent to cluster head
- if the measured value difference is larger than T_S, data are sent to cluster head
- data aggregation

Network layer

APTEEN

- **Adaptive TEEN** (APTEEN)
- cluster now has a transmission token
- thresholds are not fixed
- periodically transmits all data (refresh round)
- works well in stationary networks with steady measured quantities

Network layer

Geographical routing

- **Geographical routing**
- routing based on real location
- problem: localization
- we do not use GPS (absolute location), as it drains too much power
- instead, we often use signal strength indicators (relative location)
  - e.g., RSSI (Receive Signal Strength Indicator), etc.
- more data sinks
- different techniques

Network layer

Geographical routing

- **greedy forwarding**
  - basically a location-based heuristic
  - each routing step tries to bring the data closer to the target based solely on local information
  - may fail in a dead end if the local information contains noisy information
- **face routing**
  - recovery from dead end (greedy)
  - routing the packet alongside the interior of the faces
  - then, greedy forwarding starts again

Network layer

Geographical routing

- geographical routing suits well on a large-area applications
- e.g., monitoring of an ecosystem on a small island
- practically, they are often application-specific