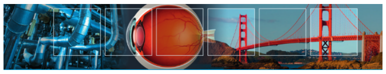
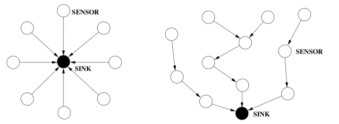
Chapter 7: Network Layer



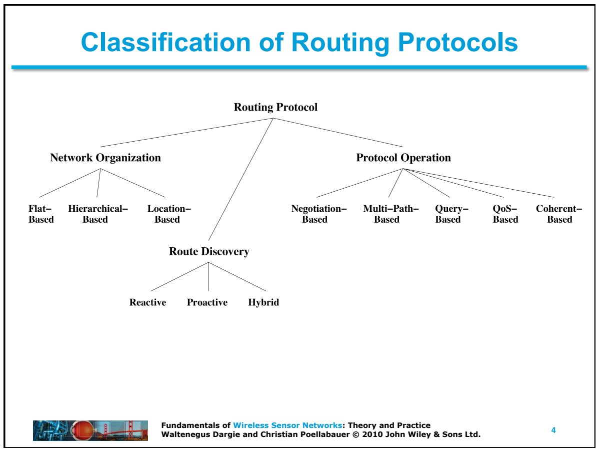
Chapter 7: Roadmap

- Routing basics
- Data-centric routing
- Proactive routing
- On-demand routing
- Hierarchical routing
- Location-based routing
- QoS-based routing

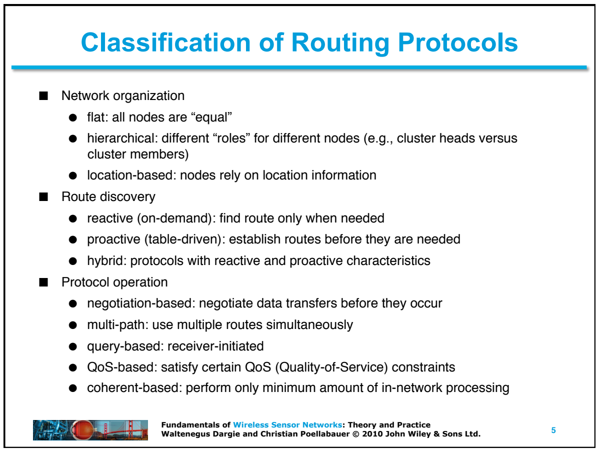
Network Layer Responsibility



- Direct communication model: every sensors communicates directly (single-hop) with the sink device (base station)
  - often not feasible due to lack of energy, scale of network, lack of unobstructed communication links between sensors and sink
- Multi-hop communication model: sensors cooperate in propagating sensor data towards the sink
- Routing is a key responsibility of the network layer
- Routing protocol is responsible for finding and maintaining path from sensor to sink

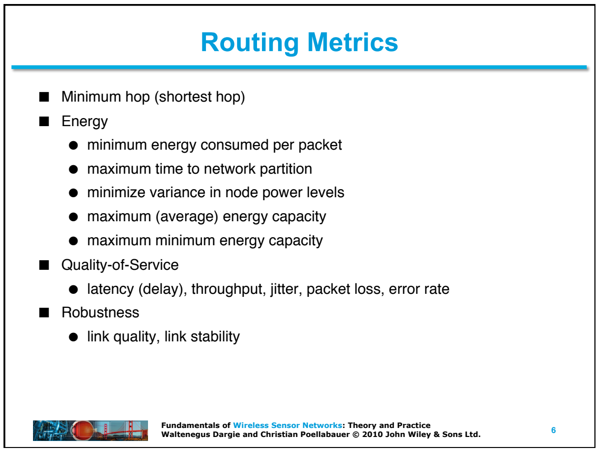


Classification of Routing Protocols



Classification of Routing Protocols

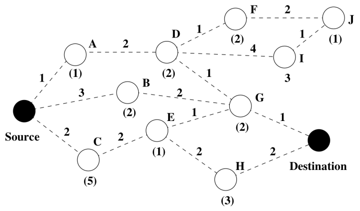
- Network organization
  - flat: all nodes are "equal"
  - hierarchical: different "roles" for different nodes (e.g., cluster heads versus cluster members)
  - location-based: nodes rely on location information
- Route discovery
  - reactive (on-demand): find route only when needed
  - proactive (table-driven): establish routes before they are needed
  - hybrid: protocols with reactive and proactive characteristics
- Protocol operation
  - negotiation-based: negotiate data transfers before they occur
  - multi-path: use multiple routes simultaneously
  - query-based: receiver-initiated
  - QoS-based: satisfy certain QoS (Quality-of-Service) constraints
  - coherent-based: perform only minimum amount of in-network processing



Routing Metrics

- Minimum hop (shortest hop)
- Energy
  - minimum energy consumed per packet
  - maximum time to network partition
  - minimize variance in node power levels
  - maximum (average) energy capacity
  - maximum minimum energy capacity
- Quality-of-Service
  - latency (delay), throughput, jitter, packet loss, error rate
- Robustness
  - link quality, link stability

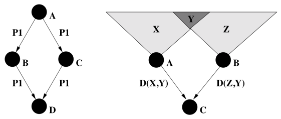
Routing Metrics (Example)



- Numbers along links: cost for transmission over link
- Numbers in parentheses: remaining energy capacity

Flooding

- Every sensor node (re-)broadcasts sensor data to all of its neighbors
- Simple and reliable technique
- Incurs large traffic overhead (maximum-hop counts and sequence numbers can be used to limit broadcasts and eliminate duplicates)
- However, flooding faces three more challenges:
  - Implosion: nodes will re-broadcast even when neighbors already have a copy
  - Overlap: sensor data contains redundant information
  - Resource blindness: resource constraints are ignored



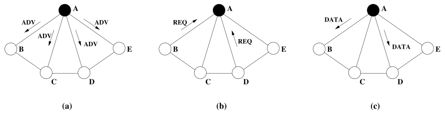
SPIN Family of Protocols

- Sensor Protocols for Information via Negotiation (SPIN)
- Example of data-centric routing
- Uses negotiations to address all problems of flooding
  - implosion: nodes negotiate before data transmission
  - overlap: nodes negotiate before data transmission
  - resource blindness: resource manager keeps track of actual resource consumption and adapts routing and communication behavior

- SPIN uses meta-data to succinctly and completely describe sensor data
- Requirements:
  - if *x* describes the meta-data for some sensor data *X*, the size of *x* (in bytes) must be less than the size of *X*
  - if two pieces of sensor data differ, their meta-data representations should differ too

SPIN-PP

- Optimized for networks using point-to-point transmission media (two nodes communicate exclusively with each other without interference)
- Flooding occurs using a 3-way handshake:
  - node A advertises newly available data using ADV (advertisement) message (containing data's meta-data)
  - neighbor nodes (B, C, D, E) check if advertised data needed and if so, respond with REQ (request for data) message
  - A sends DATA message to nodes that responded (i.e., B and D)

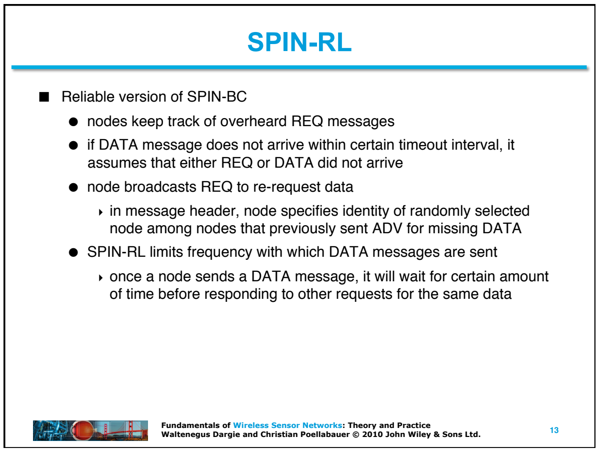


SPIN-PP

- Only nodes needed the advertised data respond
- Nodes can aggregate received data with their own and advertise aggregate
- Simple protocol
- Nodes need to know their single-hop neighbors only
- Lost ADV messages: periodically re-advertise
- Lost REQ/DATA messages: re-request data of interest if data does not arrive within certain timeout interval
- Protocol could also use explicit acknowledgments (e.g., REQ can contain list of data a node wants or does not want and using this list, a node can identify whether previous advertisements were received successfully)

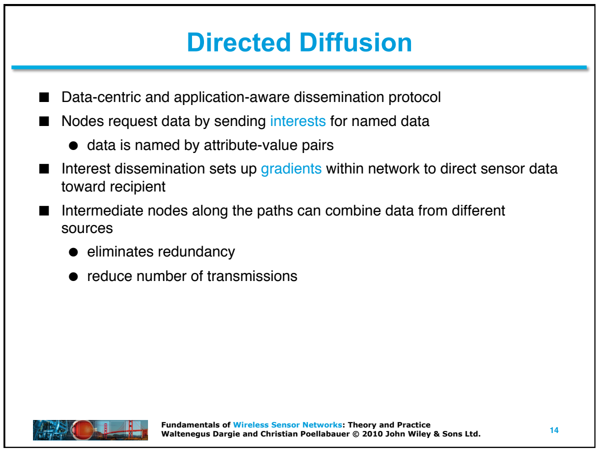
SPIN-EC and SPIN-BC

- SPIN-EC:
  - adds simple heuristic to protocol to add energy conservation
  - as long as energy sufficient, node participates in 3-way handshake
  - nodes does not participate if it believes that this will reduce its energy below a certain low-energy threshold
    - node replies to ADV only if sufficient energy for transmitting REQ and receiving DATA
    - node initiates handshake only if it has sufficient energy to send DATA to all neighbors
- SPIN-BB:
  - uses one-to-many communications (broadcast)
  - receiver node waits for a random time interval before issuing REQ; if other node's REQ overheard, the receiver node cancels timer and does not send its own REQ
  - advertiser broadcasts DATA only once (ignore duplicate REQs)



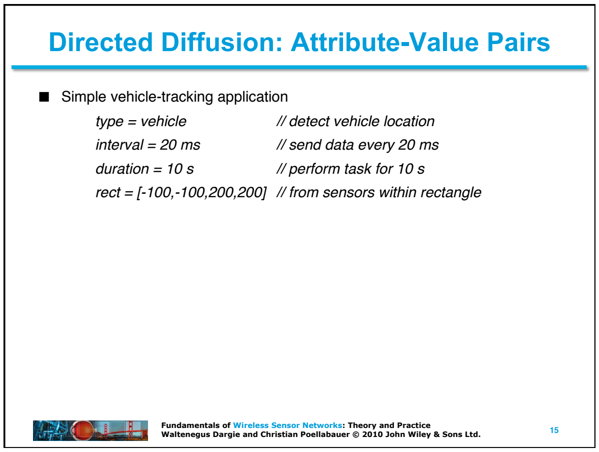
SPIN-RL

- Reliable version of SPIN-BC
  - nodes keep track of overheard REQ messages
  - if DATA message does not arrive within certain timeout interval, it assumes that either REQ or DATA did not arrive
  - node broadcasts REQ to re-request data
    - in message header, node specifies identity of randomly selected node among nodes that previously sent ADV for missing DATA
  - SPIN-RL limits frequency with which DATA messages are sent
    - once a node sends a DATA message, it will wait for certain amount of time before responding to other requests for the same data



Directed Diffusion

- Data-centric and application-aware dissemination protocol
- Nodes request data by sending interests for named data
  - data is named by attribute-value pairs
- Interest dissemination sets up gradients within network to direct sensor data toward recipient
- Intermediate nodes along the paths can combine data from different sources
  - eliminates redundancy
  - reduce number of transmissions

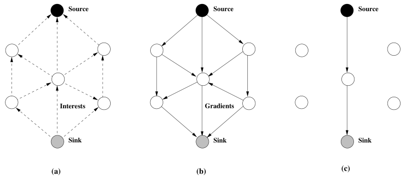


Directed Diffusion: Attribute-Value Pairs

- Simple vehicle-tracking application

```
type = vehicle               // detect vehicle location
interval = 20 ms             // send data every 20 ms
duration = 10 s              // perform task for 10 s
rect = [-100,-100,200,200]   // from sensors within rectangle
```

Directed Diffusion



- Interested node (sink) periodically sends interest to neighbors, which re-broadcast these messages (a)
- Each node establishes a gradient towards the sink, which is a reply link toward the neighbor from which the interest was received (b)
- Source can send data towards sink using these gradients (c)

Directed Diffusion

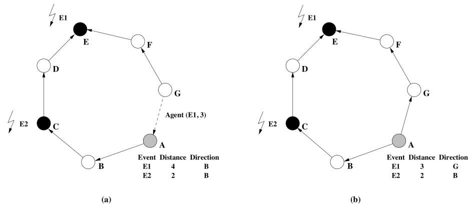
- Reinforcement:
  - multiple paths from source to sink may exist
  - sink can reinforce one particular neighbor based on some data-driven local rule
    - example: sink could reinforce neighbor from which it has received a previously unseen event
    - sink resends original interest message to that neighbor, which in turn reinforces one or more of its neighbors based on its own local rule
- Comparison to SPIN:
  - interests (queries) are issued on demand by the sink (instead of source advertisements)
    - query-based approach may not be best solutions for networks with need for continuous data transfers (e.g., environmental monitoring)
  - communication is all neighbor-to-neighbor (removing the need for addressing schemes and allowing nodes to perform aggregation and caching)

Rumor Routing

- Attempt to combine characteristics of event flooding (classic flooding) and query flooding (directed diffusion)
- Nodes maintain two data structures:
  - list of neighbors
  - event table that contains forwarding information to all known events
    - once an event is witnessed, it is added to the table (including a distance of 0) and an agent is generated with certain probability
    - agent is a long-lived packet traveling the network to propagate information about this new event (and other events encountered along its route)
    - once agent arrives at a node, the node can update its table

Rumor Routing: Example

- Node E reports new event E1 using agent
- A receives agent via G and updates its table reflecting shorter distance to E1 via G



Rumor Routing: Queries

- When node wants to issue query for certain event:
  - check if route towards event is in table
    - if yes, forward query to next-hop neighbor indicated in table
    - if no, pass query to randomly selected neighbor
  - process is continued on each forwarding node and query collects list of recently seen nodes (to avoid revisiting them)
  - TTL (time-to-live) counter is used to limit traveled distance
  - if no response received after certain timeout value (i.e., the query did not reach the target), a different technique (classic flooding of the query) can be used
- Summary: strikes balance between query and event flooding
  - query flooding may be too expensive with large query/event ratio
  - event flooding may be too expensive when number of events large
  - rumor routing propagates queries directly to nodes that seen event of interest
  - latency secondary concern in rumor routing
  - table can grow large with large number of events

Gradient-Based Routing (GBR)

- Another variant of directed diffusion
- Gradient is determined on the basis of the number of hops to the sink
  - a sink floods interest message
  - each interest message records number of hops traveled
  - therefore, a node can determine its distance (called height) to sink
  - gradient is difference between a node's height and height of neighbor
  - packet is forwarded on the link with the largest gradient
- Nodes can establish a Data Combining Entity (DCE), which is responsible for data compaction
- Nodes can use a traffic spreading technique to balance traffic over network
  - stochastic scheme: node selects next hop randomly when there are two or more with same gradient
  - energy-based scheme: node increases its height when its energy level is low
  - stream-based scheme: new streams are diverted away from nodes that already serve other streams (e.g., by reporting increased heights)

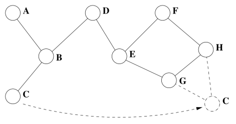
Proactive Routing

- Routes are established before they are actually needed
- Also called table-driven routing
- Main advantage: routes can be used immediately when needed (table look up for next-hop neighbor)
- Main disadvantages:
  - establishing and maintaining routes that are infrequently (or never) needed
  - routing tables can become very large
  - stale information in tables can lead to routing errors

Destination-Sequenced Distance Vector

- DSDV is a modified version of class Distributed Bellman-Ford algorithm
- Concept of distance-vector algorithms:
  - node *i* maintains a list of distances $\{d_{ij}^{x}\}$ for each destination *x* via node *j*
  - node *i* selects *k* as next-hop if $d_{ik}^{x} = min\ \{d_{ij}^{x}\}$
  - distances stored in routing table, along with sequence number for each entry (assigned by destination node)
    - purpose of sequence number is to identify stale routes
  - nodes broadcast their routing tables to their neighbors periodically and whenever significant information is available
    - full dump packet: contains entire routing table
    - incremental packet: contains only changed table entries
    - a receiving node updates its table if the received information has a more recent sequence number or if sequence number is identical, but new route is shorter

DSDV: Example



Destination	NextHop	Distance
A	B	2
B	B	1
C	B	2
D	D	0
E	E	1
F	E	2
G	E	2
H	E	3

Node D's Table Before C's Move

Destination	NextHop	Distance
A	B	2
B	B	1
C	E	3
D	D	0
E	E	1
F	E	2
G	E	2
H	E	3

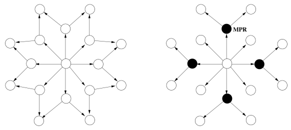
Node D's Table After C's Move

Optimized Link State Routing

- OLSR is a protocol based on the link state algorithm
  - nodes periodically broadcast topological information updates to all other nodes in the network
  - allows nodes to obtain complete topological map and to compute paths
- Nodes use neighbor sensing using HELLO messages:
  - allows nodes to identify neighbors and to detect changes in neighborhood
  - HELLO message contains node’s identity (address) and list of all known neighbors
  - if node A can receive node B’s HELLO messages, but is not in node B’s list of known neighbors, the link B ➔ A is asymmetric (otherwise symmetric)
  - instead of flooding HELLO messages to all nodes in network, OLSR uses multipoint relays (MPRs)

Optimized Link State Routing

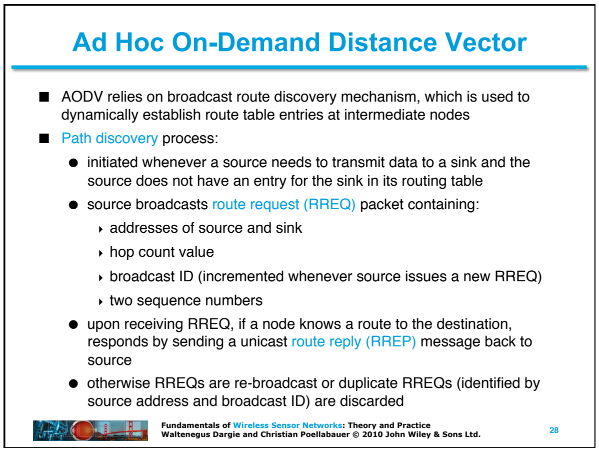
- Multipoint relays:
  - a node selects a set of symmetric neighbors as MPRs (MPR selector set); each node can use different algorithm/heuristic for the selection process
    - example: node determines its 2-hop neighbors (using HELLO messages) and selects minimum set of 1-hop neighbors to reach all 2-hop neighbors
  - only MPRs forward messages to other nodes
  - HELLO messages contain addresses of MPRs, i.e., a node advertises its reachability via MPRs only instead of all neighbors



(a) (b)

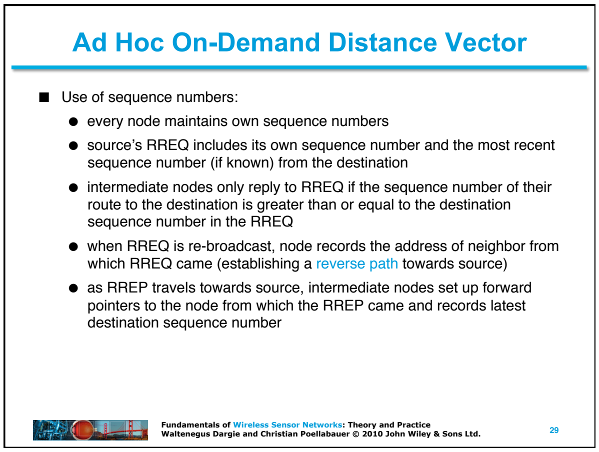
On-Demand (Reactive) Routing

- Routes are not established until actually needed
- Instead, a source node (knowing the identity/address of a destination node) initiates a route discovery process which completes when at least one route has been found or all possible paths have been examined
- A newly discovered route is then maintained until it either breaks or is no longer needed by the source



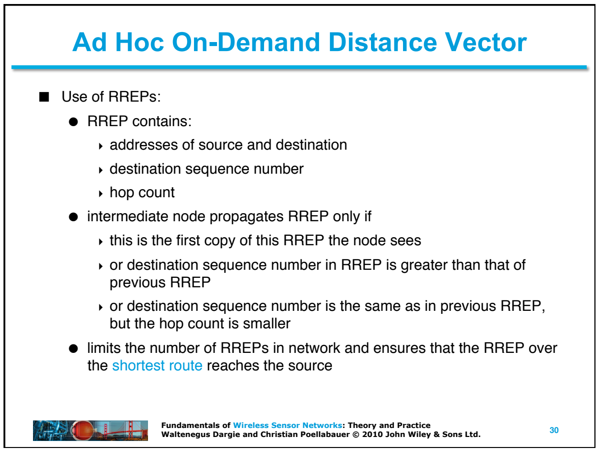
Ad Hoc On-Demand Distance Vector

- AODV relies on broadcast route discovery mechanism, which is used to dynamically establish route table entries at intermediate nodes
- Path discovery process:
  - initiated whenever a source needs to transmit data to a sink and the source does not have an entry for the sink in its routing table
  - source broadcasts route request (RREQ) packet containing:
    - addresses of source and sink
    - hop count value
    - broadcast ID (incremented whenever source issues a new RREQ)
    - two sequence numbers
  - upon receiving RREQ, if a node knows a route to the destination, responds by sending a unicast route reply (RREP) message back to source
  - otherwise RREQs are re-broadcast or duplicate RREQs (identified by source address and broadcast ID) are discarded



Ad Hoc On-Demand Distance Vector

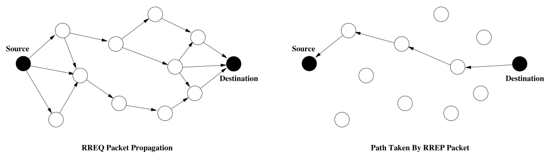
- Use of sequence numbers:
  - every node maintains own sequence numbers
  - source's RREQ includes its own sequence number and the most recent sequence number (if known) from the destination
  - intermediate nodes only reply to RREQ if the sequence number of their route to the destination is greater than or equal to the destination sequence number in the RREQ
  - when RREQ is re-broadcast, node records the address of neighbor from which RREQ came (establishing a reverse path towards source)
  - as RREP travels towards source, intermediate nodes set up forward pointers to the node from which the RREP came and records latest destination sequence number



Ad Hoc On-Demand Distance Vector

- Use of RREPs:
  - RREP contains:
    - addresses of source and destination
    - destination sequence number
    - hop count
  - intermediate node propagates RREP only if
    - this is the first copy of this RREP the node sees
    - or destination sequence number in RREP is greater than that of previous RREP
    - or destination sequence number is the same as in previous RREP, but the hop count is smaller
  - limits the number of RREPs in network and ensures that the RREP over the shortest route reaches the source

Ad Hoc On-Demand Distance Vector

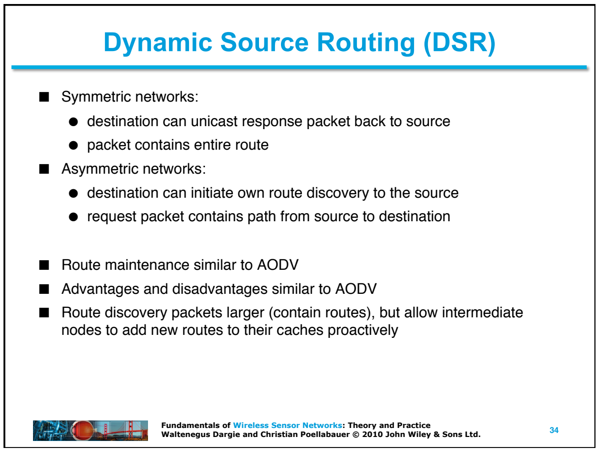


Ad Hoc On-Demand Distance Vector

- Routes expire after certain amount of time
- Link state monitored using HELLO messages among neighbors
  - when a link/route breaks, the node noticing this issues a route error (RRER) packet towards the source (which can re-initiate route discovery)
- Summary:
  - routes are established only when needed
  - no need for route table updates and exchanges for unused routes
  - periodic HELLO messages
  - initial transmission delay if route must be discovered first
  - RREP uses reverse path of RREQ, i.e., AODV assumes symmetric links

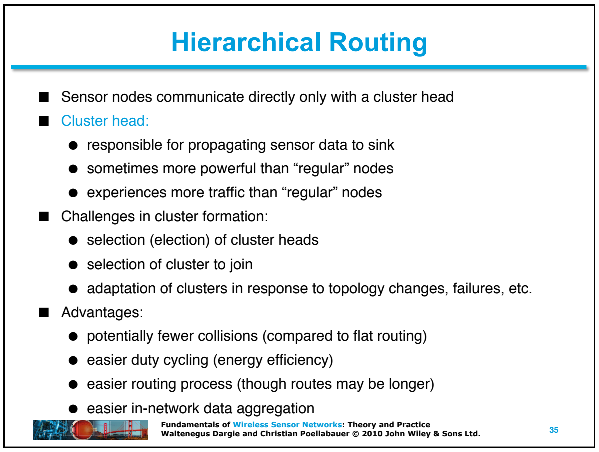
Dynamic Source Routing (DSR)

- Route discovery and maintenance similar to AODV
- Each node maintains route cache with entries that are continuously updated when node learns new routes
- Similar to AODV, source first inspects cache and initiates route discovery if no route found
- RREQ contains:
  - addresses of source and destination
  - unique request ID
  - list of visited nodes
- Each forwarding node inserts its own address into list of visited nodes
- When node finds its own address in RREQ, packet is discarded
- Nodes keep cache of recently forwarded RREQs (sender address and request ID) to identify duplicates (which are discarded)
- Once RREQ reaches destination, it has recorded the entire route



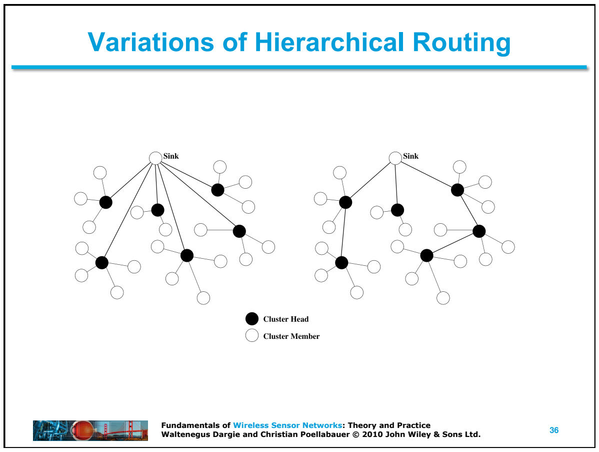
Dynamic Source Routing (DSR)

- Symmetric networks:
  - destination can unicast response packet back to source
  - packet contains entire route
- Asymmetric networks:
  - destination can initiate own route discovery to the source
  - request packet contains path from source to destination
- Route maintenance similar to AODV
- Advantages and disadvantages similar to AODV
- Route discovery packets larger (contain routes), but allow intermediate nodes to add new routes to their caches proactively



Hierarchical Routing

- Sensor nodes communicate directly only with a cluster head
- Cluster head:
  - responsible for propagating sensor data to sink
  - sometimes more powerful than “regular” nodes
  - experiences more traffic than “regular” nodes
- Challenges in cluster formation:
  - selection (election) of cluster heads
  - selection of cluster to join
  - adaptation of clusters in response to topology changes, failures, etc.
- Advantages:
  - potentially fewer collisions (compared to flat routing)
  - easier duty cycling (energy efficiency)
  - easier routing process (though routes may be longer)
  - easier in-network data aggregation

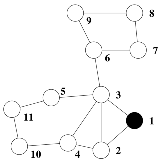


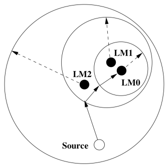
Variations of Hierarchical Routing

Landmark Routing

- Landmark: node to which its neighbors within certain number of hops know a route to that node
- Example:
  - nodes 2..6 have routes to node 1
  - node 1 is a landmark “visible” to all nodes within 2-hop distance, i.e., node 1 is a landmark of radius 2
- Node *i* for which nodes within *n* hops have routes toward is is a landmark of radius *n*
- Hierarchy of landmarks: packet can be forwarded toward a destination by choosing an appropriate sequence of landmarks

Landmark Routing

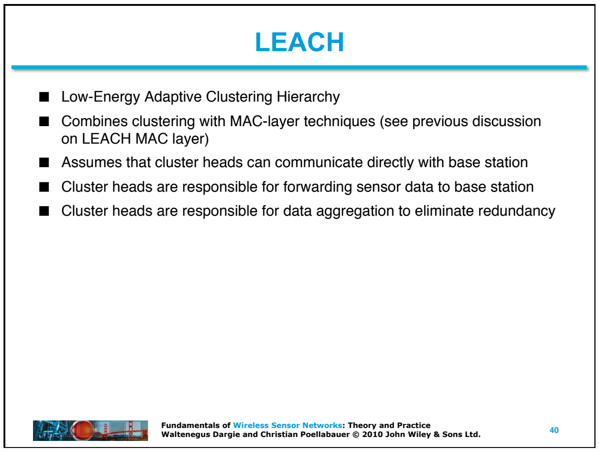




- Example:
  - landmark address of node LM0: LM2.LM1.LM0
  - source has entry for LM2: packet travels towards LM2
  - once packet reaches node within range of LM1: packet travels towards LM1
  - once packet reaches node within range of LM0: packet travels to LM0

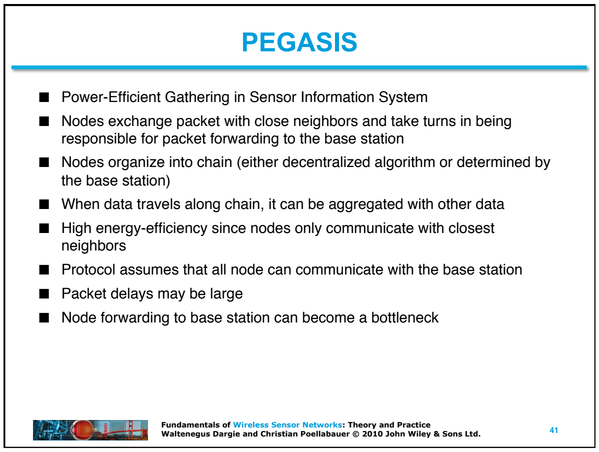
LANMAR Protocol

- Combines landmark routing with Fisheye State Routing
- Uses landmarks to establish two-tiered logical hierarchy
  - each landmark is a cluster head of a logical subnet
- Fisheye State Routing
  - link state routing protocol
  - frequency of route updates depends on the distance (routes within fisheye scope are more accurate than others)
- Routing updates are only exchanged with nodes in immediate neighborhood and with landmark nodes (update frequency to all other nodes is zero)
- Data propagation:
  - if destination is known, forward data directly to destination
  - otherwise, forward towards the landmark corresponding to the destination’s logical subnet
  - once packet enters the scope of the destination, it is forwarded directly to the destination



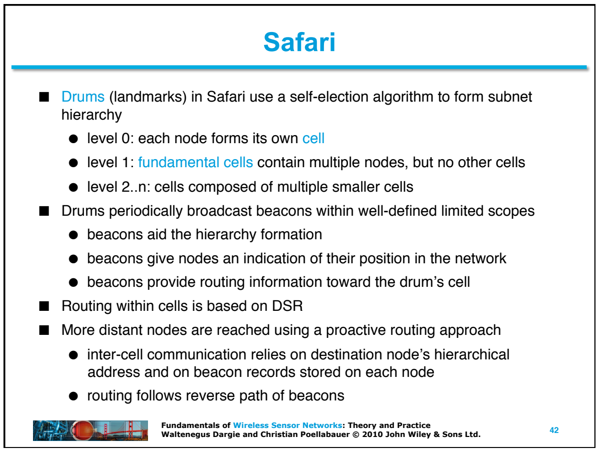
LEACH

- Low-Energy Adaptive Clustering Hierarchy
- Combines clustering with MAC-layer techniques (see previous discussion on LEACH MAC layer)
- Assumes that cluster heads can communicate directly with base station
- Cluster heads are responsible for forwarding sensor data to base station
- Cluster heads are responsible for data aggregation to eliminate redundancy



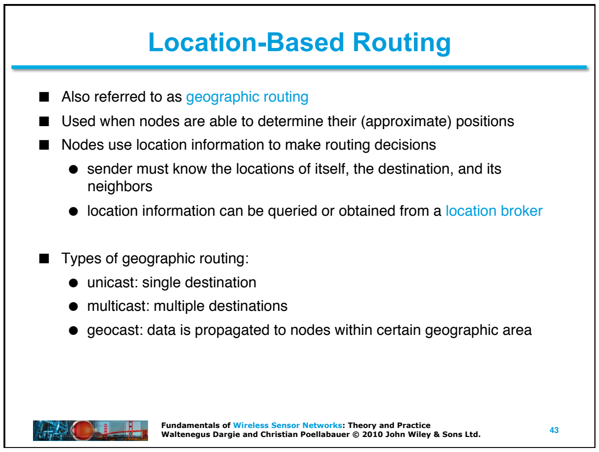
PEGASIS

- Power-Efficient Gathering in Sensor Information System
- Nodes exchange packet with close neighbors and take turns in being responsible for packet forwarding to the base station
- Nodes organize into chain (either decentralized algorithm or determined by the base station)
- When data travels along chain, it can be aggregated with other data
- High energy-efficiency since nodes only communicate with closest neighbors
- Protocol assumes that all node can communicate with the base station
- Packet delays may be large
- Node forwarding to base station can become a bottleneck



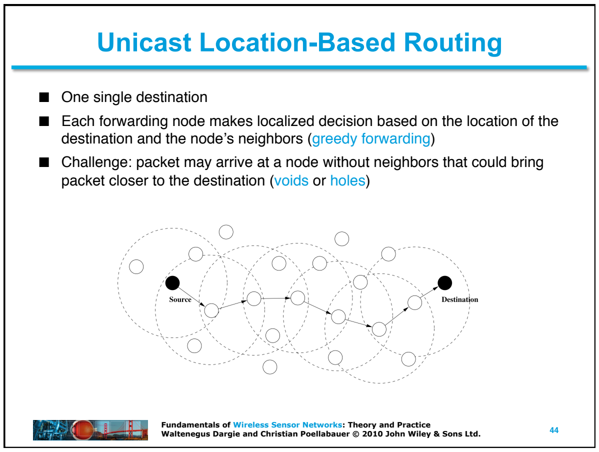
Safari

- Drums (landmarks) in Safari use a self-election algorithm to form subnet hierarchy
  - level 0: each node forms its own cell
  - level 1: fundamental cells contain multiple nodes, but no other cells
  - level 2..n: cells composed of multiple smaller cells
- Drums periodically broadcast beacons within well-defined limited scopes
  - beacons aid the hierarchy formation
  - beacons give nodes an indication of their position in the network
  - beacons provide routing information toward the drum’s cell
- Routing within cells is based on DSR
- More distant nodes are reached using a proactive routing approach
  - inter-cell communication relies on destination node’s hierarchical address and on beacon records stored on each node
  - routing follows reverse path of beacons



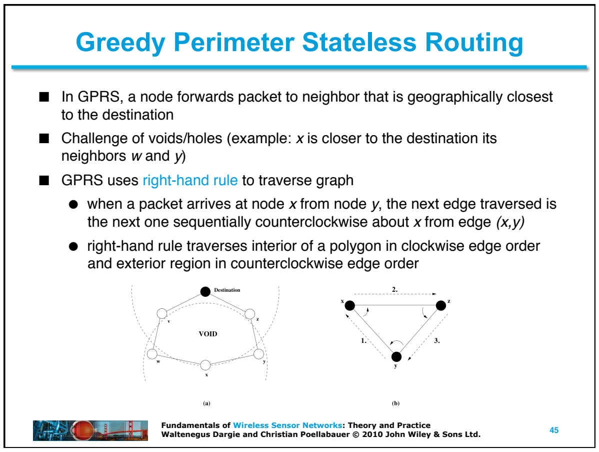
Location-Based Routing

- Also referred to as geographic routing
- Used when nodes are able to determine their (approximate) positions
- Nodes use location information to make routing decisions
  - sender must know the locations of itself, the destination, and its neighbors
  - location information can be queried or obtained from a location broker
- Types of geographic routing:
  - unicast: single destination
  - multicast: multiple destinations
  - geocast: data is propagated to nodes within certain geographic area



Unicast Location-Based Routing

- One single destination
- Each forwarding node makes localized decision based on the location of the destination and the node's neighbors (greedy forwarding)
- Challenge: packet may arrive at a node without neighbors that could bring packet closer to the destination (voids or holes)



Greedy Perimeter Stateless Routing

- In GPRS, a node forwards packet to neighbor that is geographically closest to the destination
- Challenge of voids/holes (example: *x* is closer to the destination its neighbors *w* and *y*)
- GPRS uses right-hand rule to traverse graph
  - when a packet arrives at node *x* from node *y*, the next edge traversed is the next one sequentially counterclockwise about *x* from edge *(x,y)*
  - right-hand rule traverses interior of a polygon in clockwise edge order and exterior region in counterclockwise edge order

(a)

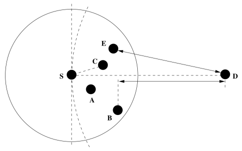
(b)

Greedy Perimeter Stateless Routing

- Sequence of edges traversed according to rule is called perimeter
- In non-planar graphs, right-hand rule may take tour of edges that does not trace the boundary of a closed polygon
- To obtain planar graphs, crossing edges must be removed (without partitioning the network)
  - Relative Neighborhood Graph (RNG)
  - Gabriel Graph (GG)
- Example: RNG
  - lune between *u* and *v* (intersection of radio ranges) must be empty of any witness node *w* such that edge *(u,v)* can be included in the RNG (i.e., if this region is nonempty, the link *(u,v)* will be removed)

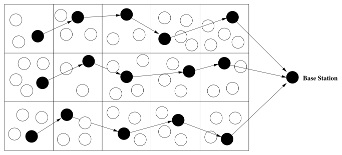
Forwarding Strategies

- Greedy: minimize distance to destination in each hop
- Nearest with Forwarding Progress (NFP): nearest of all neighbors that make positive progress (in terms of geographic distance) toward destination
- Most Forwarding Progress within Radius (MFR): neighbor that makes greatest positive progress (progress is distance between source and its neighbor node projected onto a line drawn from source to destination)
- Compass Routing: neighbor with smallest angle between a line drawn from source to the neighbor and the line connecting source and destination



Geographic Adaptive Fidelity

- GAF was primarily designed for networks with mobile nodes
- Network region is divided into virtual grid
- In each cell, only one node is a forwarder at any given time (all other nodes can sleep)
- Assumption: all nodes within a cell can communicate with all nodes within all adjacent cells



Geographic Adaptive Fidelity

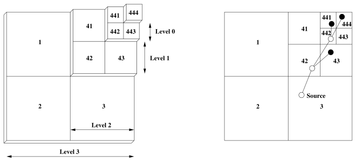
- Discovery state:
  - nodes are initially in the discovery state, where they listen for messages from other nodes within cell
  - each node uses timer; when timer expires, node broadcasts discovery message and enters active state
- Active state:
  - node periodically re-broadcasts discovery messages
  - node sets another timer to re-enter discovery state after timer expiration
- Sleep state:
  - entered (from both discovery or active states) whenever node determines that some other node is forwarder
  - nodes periodically re-enter discovery state
- Forwarder node:
  - determined using negotiation process
  - nodes in active state win over nodes in discovery state
  - node IDs used to break ties

Multicast Location-Based Routing

- Multicast: deliver the same packet to multiple receivers
- Simple solution 1: deliver copies of same packet to each individual receiver via unicast routing
  - resource-inefficient
- Simple solution 2: flood the entire network
  - resource-inefficient
- Concerned with efficiently delivering the same packet to receivers, i.e., minimize the number of links the packet has to travel
- Common approach: multicast tree rooted at source and destinations are leaf nodes

Scalable Position-Based Multicast

- SPBM relies on group management scheme to maintain list of all destinations for a packet
- Packet header carries group membership information instead of all destination's addresses to ensure efficiency
- Network is represented as a quad-tree with pre-defined number of levels *L*
- Example: *L=3* (levels *0..L-1*)
- All nodes within level-0 square are in radio range of each other



Scalable Position-Based Multicast

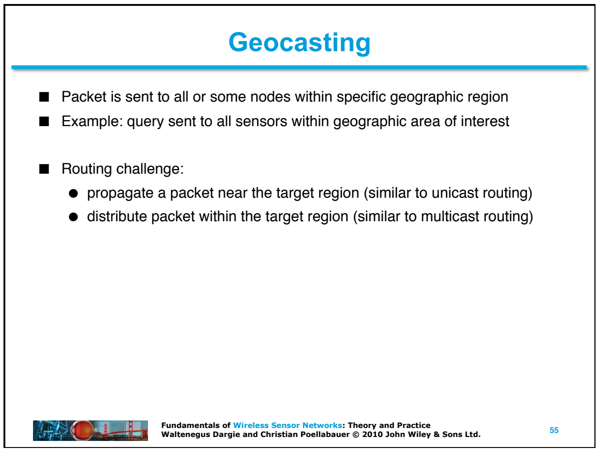
- Nodes maintain two tables:
  - global member table
    - containing entries for the three neighboring squares for each level
    - each entry contains square's identifier and list of nodes in square
  - local member table
    - containing all members of the node's level-0 neighbors
    - each entry contains node IDs and membership information of nodes
- Membership information indicates the multicast groups to which node belongs and is encoded as vector (each bit represents a multicast group)
  - 10100010 for square 41 in the global member table indicates that there exist nodes in square 41 that belong to multicast groups 2, 6, and 8
  - 00000001 for node 14 in the local member table indicates that node 14 is a member of multicast group 1
- Local member table is broadcast periodically within node's level-0 square
- Randomly chosen node in each level-0 square periodically disseminates its global member table to all nodes in its level-1 square (process repeated at each level)

Scalable Position-Based Multicast

- SPBM uses greedy forwarding to choose next-hop neighbors
- SPBM uses splitting to send copies of the same packet into different "directions"
  - splitting is based on a heuristic that provides a tradeoff between total number of nodes forwarding the packet and the optimality of the individual routes toward the destinations
  - once a forwarding node finds a multicast member in its local member table, it forwards packet directly to that node
- Similar to GPSR, SPBM uses perimeter routing mode to address voids

Other Multicast Routing Protocols

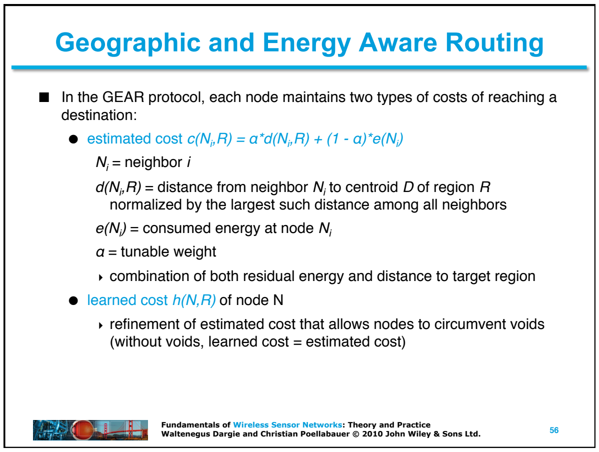
- Geographic Multicast Routing (GMR)
  - heuristic neighbor selection scheme with low computational overheads
  - results in routest based on a cost over progress metric (ratio of the number of selected forwarding nodes over the progress made toward all destinations)
- Receiver Based Multicast (RBMulticast)
  - sender can transmit packet without specifying next-hop node
  - network is divided into multicast regions
  - RBMulticast does not use membership tables (stateless)
  - each multicast region is represented by virtual node and each forwarding node replicates packet for each region that contains at least one multicast member; packet is then sent toward virtual node
  - relies on MAC layer that ensures that neighbor closest to location of virtual node takes responsibility for forwarding the packet (nodes contend for channel access and nodes closer to destination are more likely to win)



Geocasting

- Packet is sent to all or some nodes within specific geographic region
- Example: query sent to all sensors within geographic area of interest
- Routing challenge:
  - propagate a packet near the target region (similar to unicast routing)
  - distribute packet within the target region (similar to multicast routing)

Fundamentals of Wireless Sensor Networks: Theory and Practice
Waltenegus Dargie and Christian Poellabauer © 2010 John Wiley & Sons Ltd.



Geographic and Energy Aware Routing

- In the GEAR protocol, each node maintains two types of costs of reaching a destination:
  - estimated cost $c(N_i,R) = \alpha*d(N_i,R) + (1 - \alpha)*e(N_i)$

    N_i = neighbor i

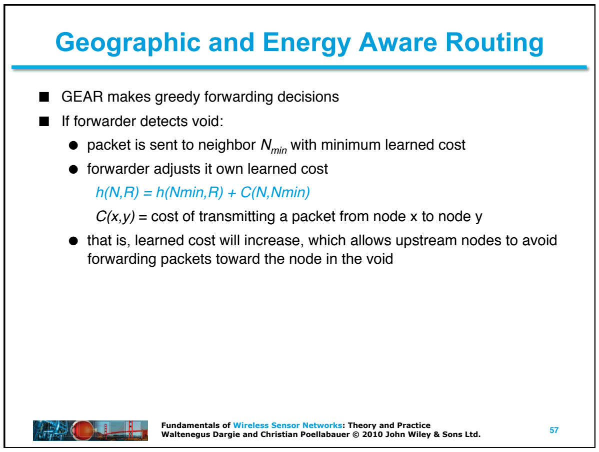
    $d(N_i,R)$ = distance from neighbor N_i to centroid D of region R normalized by the largest such distance among all neighbors

    $e(N_i)$ = consumed energy at node N_i

    α = tunable weight

    - combination of both residual energy and distance to target region
  - learned cost $h(N,R)$ of node N
    - refinement of estimated cost that allows nodes to circumvent voids (without voids, learned cost = estimated cost)

Fundamentals of Wireless Sensor Networks: Theory and Practice
Waltenegus Dargie and Christian Poellabauer © 2010 John Wiley & Sons Ltd.



Geographic and Energy Aware Routing

- GEAR makes greedy forwarding decisions
- If forwarder detects void:
  - packet is sent to neighbor N_{min} with minimum learned cost
  - forwarder adjusts it own learned cost

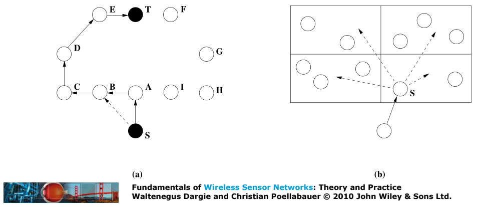
    $h(N,R) = h(Nmin,R) + C(N,Nmin)$

    $C(x,y)$ = cost of transmitting a packet from node x to node y
  - that is, learned cost will increase, which allows upstream nodes to avoid forwarding packets toward the node in the void

Fundamentals of Wireless Sensor Networks: Theory and Practice
Waltenegus Dargie and Christian Poellabauer © 2010 John Wiley & Sons Ltd.

Geographic and Energy Aware Routing

- Example: S(source) sending a packet to centroid of target region (T)
- Learned (estimated) costs are √5 for neighbors B and I and 2 for neighbor A
- Packet goes from S to A, which finds itself in hole
- A forwards packet to B and computes own cost h(A,T) = h(B,T) + C(A,B) (assuming cost (A,B)=1, the new cost will be √5+1)
- Next time, S will directly select B instead of A



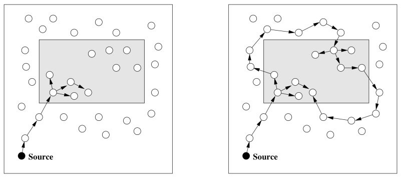
Geographic and Energy Aware Routing

- Once packet reaches target region R, simple flooding with duplicate suppression could be used
- GEAR relies on Recursive Geographic Forwarding
  - nodes within target region create four copies of packet bound to four smaller subregions
  - in each subregion, GEAR repeats the forwarding and splitting process until node is reached that is the only one within the current subregion

Geographic-Forwarding-Perimeter-Geocast

- GFPG uses greedy forwarding to propagate packet toward target region (destination is center of the region)
- Perimeter routing is used when voids are reached
- Within target region, flooding can be used, but delivery to all nodes would not be guaranteed
- Instead, GFPG uses a combination of geocast and perimeter routing
  - once a packet reaches target region, it is flooded too all nodes
  - further, region border nodes (nodes with at least one neighbor outside the region) send packet to neighbor nodes outside region
  - nodes outside the region use right-hand rule to forward packet to neighbors in the planar graph; packet travels around the face until it enters the region again

Geographic-Forwarding-Perimeter-Geocast

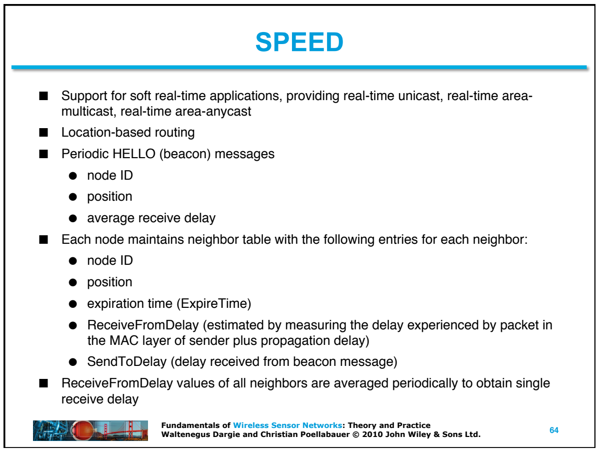


QoS-Based Routing Protocols

- Protocols that explicitly address one or more Quality-of-Service (QoS) metrics
- Examples of QoS metrics:
  - low (end-to-end) latency/delay
  - low jitter
  - low energy consumption
  - low bandwidth requirements
  - high reliability

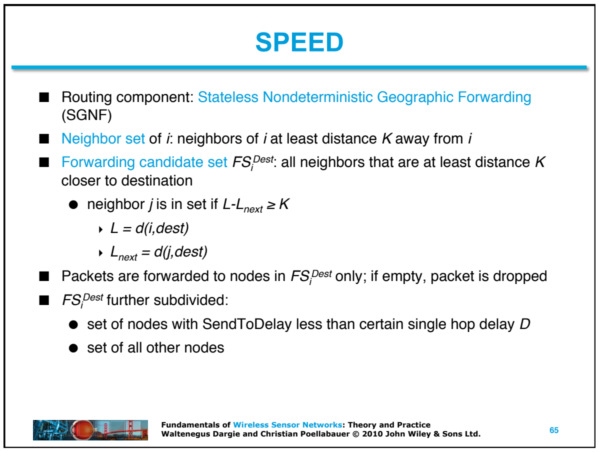
Sequential Assignment Routing

- Example of multipath routing
- SAR creates multiple trees, each rooted at a 1-hop neighbor of the sink, to establish multiple paths from each node to the sink
- Tree grows outward from the sink, while avoiding nodes with low QoS
- Additive QoS metric (higher values indicate lower QoS)
- Nodes are likely to be part of multiple trees (routes)
- Route is chosen based on the QoS metric, energy (how many packets can be transmitted without energy depletion?), and priority level of packet
- Goal of SAR is to minimize the average weighted QoS metric over the lifetime of the network
- Multipath routing helps with fault tolerance and quick recovery from broken routes
- Establishing and maintaining all trees is expensive



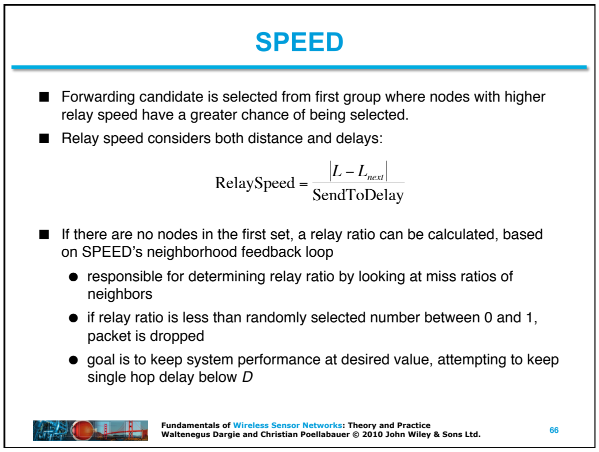
SPEED

- Support for soft real-time applications, providing real-time unicast, real-time area-multicast, real-time area-anycast
- Location-based routing
- Periodic HELLO (beacon) messages
  - node ID
  - position
  - average receive delay
- Each node maintains neighbor table with the following entries for each neighbor:
  - node ID
  - position
  - expiration time (ExpireTime)
  - ReceiveFromDelay (estimated by measuring the delay experienced by packet in the MAC layer of sender plus propagation delay)
  - SendToDelay (delay received from beacon message)
- ReceiveFromDelay values of all neighbors are averaged periodically to obtain single receive delay



SPEED

- Routing component: Stateless Nondeterministic Geographic Forwarding (SGNF)
- Neighbor set of *i*: neighbors of *i* at least distance *K* away from *i*
- Forwarding candidate set FS_i^{Dest}: all neighbors that are at least distance *K* closer to destination
  - neighbor *j* is in set if $L-L_{next} \geq K$
    - $L = d(i,dest)$
    - $L_{next} = d(j,dest)$
- Packets are forwarded to nodes in FS_i^{Dest} only; if empty, packet is dropped
- FS_i^{Dest} further subdivided:
  - set of nodes with SendToDelay less than certain single hop delay *D*
  - set of all other nodes



SPEED

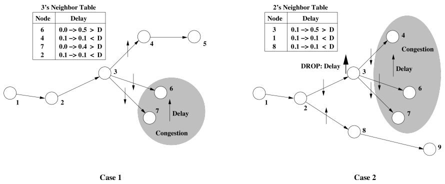
- Forwarding candidate is selected from first group where nodes with higher relay speed have a greater chance of being selected.
- Relay speed considers both distance and delays:

$$\text{RelaySpeed} = \frac{|L - L_{next}|}{\text{SendToDelay}}$$

- If there are no nodes in the first set, a relay ratio can be calculated, based on SPEED's neighborhood feedback loop
  - responsible for determining relay ratio by looking at miss ratios of neighbors
  - if relay ratio is less than randomly selected number between 0 and 1, packet is dropped
  - goal is to keep system performance at desired value, attempting to keep single hop delay below *D*

SPEED

- Back-pressure rerouting protocol of SPEED is responsible for
  - preventing voids that occur when nodes fail to find next hop node
  - reducing congestion using a feedback approach



3's Neighbor Table

Node	Delay
6	0.0 –> 0.5 > D
4	0.1 –> 0.1 < D
7	0.0 –> 0.4 > D
2	0.1 –> 0.1 < D

2's Neighbor Table

Node	Delay
3	0.1 –> 0.5 > D
1	0.1 –> 0.1 < D
8	0.1 –> 0.1 < D

Case 1

Case 2

Multipath Multi-SPEED

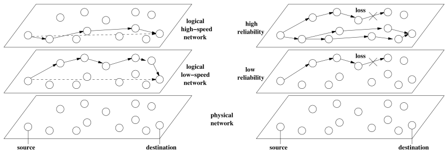
- MMSPEED provides QoS differentiation in terms of timeliness and reliability, while also minimizing protocol overhead
- Makes localized routing decision without a priori route discovery or global network state updates
- Offers multiple delivery speed options
- Can be considered as virtual overlay of multiple SPEED layers
- Each layer l has SetSpeedl associated, which is a pre-specified lower-bound speed, i.e., when a node computes the relay speed for each neighbor, it chooses a forwarding node whose relay speed is at least the desired SetSpeed value

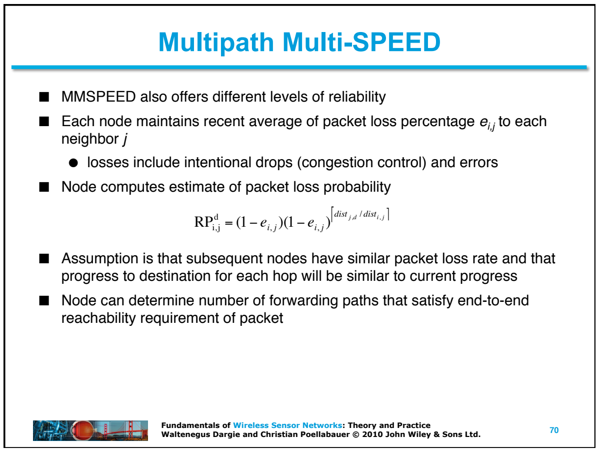
$$\text{ReqSpeed(x)} = \frac{dist_{s,d}(x)}{deadline(x)}$$

$$\text{SetSpeed}_l = \min_{j=1}^{L}\{\text{SetSpeed}_j \mid \text{SetSpeed}_j \geq \text{ReqSpeed(x)}\}$$

Multipath Multi-SPEED

- Layer selected at one node can differ from layer selected at other node (e.g., packet can be boosted by using a higher layer)
- Necessary to determine remaining time to deadline, requiring synchronized clocks in the network
- MMSPEED measures elapsed time at each node and piggybacks time onto packet
- MMSPEED also offers different levels of reliability



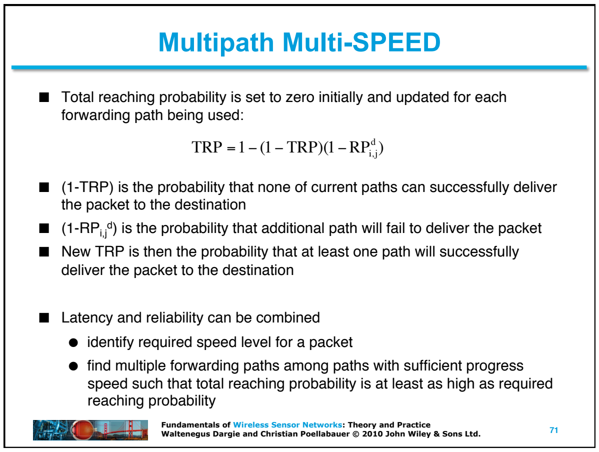


Multipath Multi-SPEED

- MMSPEED also offers different levels of reliability
- Each node maintains recent average of packet loss percentage $e_{i,j}$ to each neighbor *j*
  - losses include intentional drops (congestion control) and errors
- Node computes estimate of packet loss probability

$$\mathrm{RP}_{\mathrm{i,j}}^{\mathrm{d}} = (1 - e_{i,j})(1 - e_{i,j})^{\lceil dist_{j,d} / dist_{i,j} \rceil}$$

- Assumption is that subsequent nodes have similar packet loss rate and that progress to destination for each hop will be similar to current progress
- Node can determine number of forwarding paths that satisfy end-to-end reachability requirement of packet



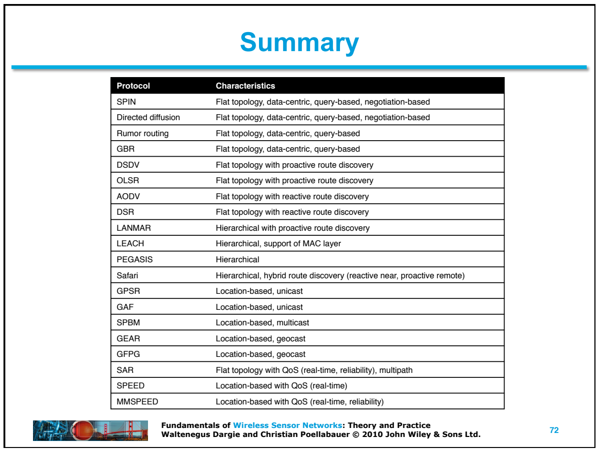
Multipath Multi-SPEED

- Total reaching probability is set to zero initially and updated for each forwarding path being used:

$$\mathrm{TRP} = 1 - (1 - \mathrm{TRP})(1 - \mathrm{RP}_{\mathrm{i,j}}^{\mathrm{d}})$$

- (1-TRP) is the probability that none of current paths can successfully deliver the packet to the destination
- (1-$\mathrm{RP}_{\mathrm{i,j}}^{\mathrm{d}}$) is the probability that additional path will fail to deliver the packet
- New TRP is then the probability that at least one path will successfully deliver the packet to the destination

- Latency and reliability can be combined
  - identify required speed level for a packet
  - find multiple forwarding paths among paths with sufficient progress speed such that total reaching probability is at least as high as required reaching probability



Summary

Protocol	Characteristics
SPIN	Flat topology, data-centric, query-based, negotiation-based
Directed diffusion	Flat topology, data-centric, query-based, negotiation-based
Rumor routing	Flat topology, data-centric, query-based
GBR	Flat topology, data-centric, query-based
DSDV	Flat topology with proactive route discovery
OLSR	Flat topology with proactive route discovery
AODV	Flat topology with reactive route discovery
DSR	Flat topology with reactive route discovery
LANMAR	Hierarchical with proactive route discovery
LEACH	Hierarchical, support of MAC layer
PEGASIS	Hierarchical
Safari	Hierarchical, hybrid route discovery (reactive near, proactive remote)
GPSR	Location-based, unicast
GAF	Location-based, unicast
SPBM	Location-based, multicast
GEAR	Location-based, geocast
GFPG	Location-based, geocast
SAR	Flat topology with QoS (real-time, reliability), multipath
SPEED	Location-based with QoS (real-time)
MMSPEED	Location-based with QoS (real-time, reliability)