Freie Universität Berlin

Embedded Internet and the Internet of Things WS 12/13

5. Network Layer

Prof. Dr. Mesut Güneş
Distributed, embedded Systems (DES)
Institute of Computer Science
Freie Universität Berlin

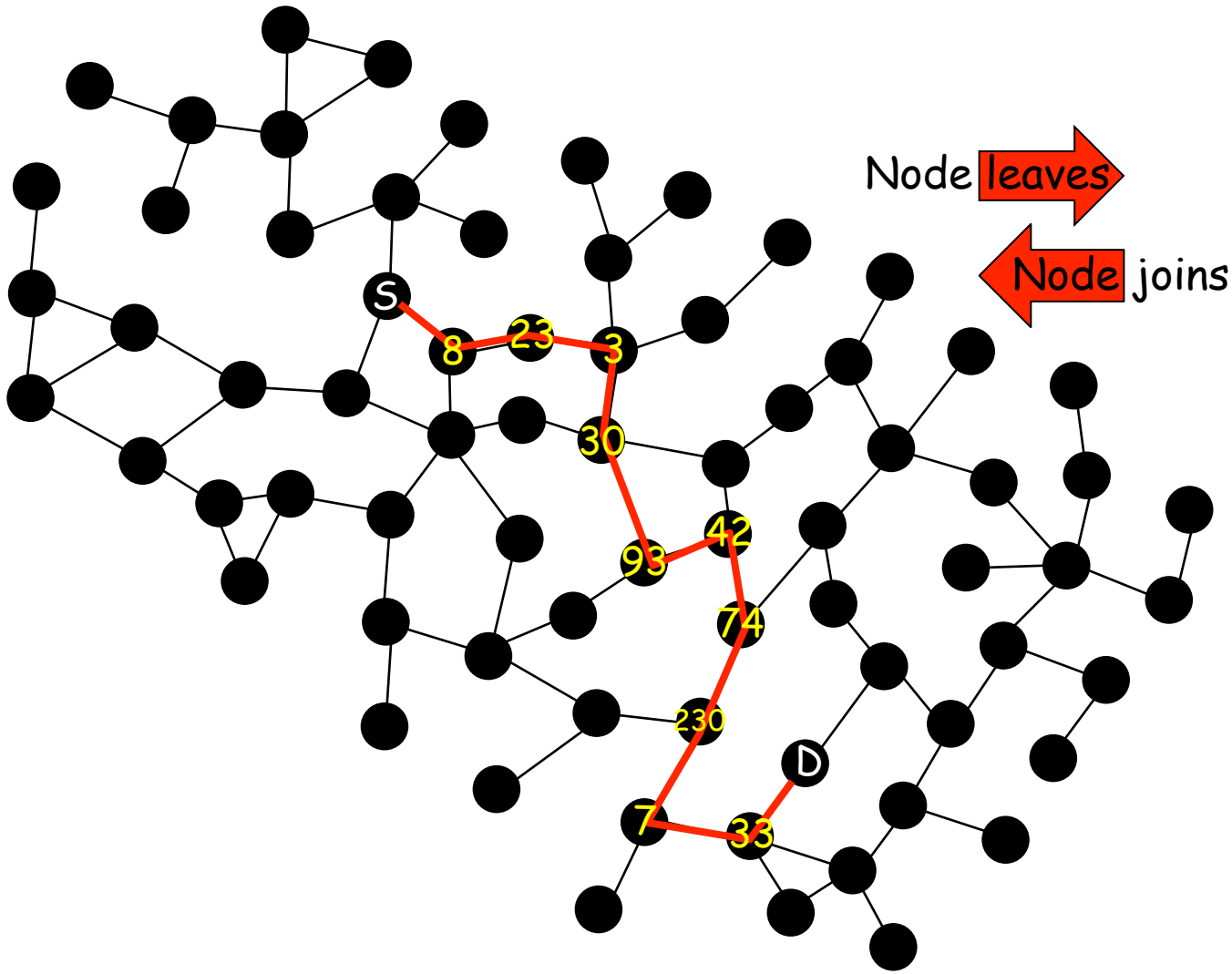
Overview

- Motivation & challenges for routing
- Link and routing metrics
- Probabilistic routing
- Content based routing / Data centric routing
- Geographical routing
- RPL

Motivation

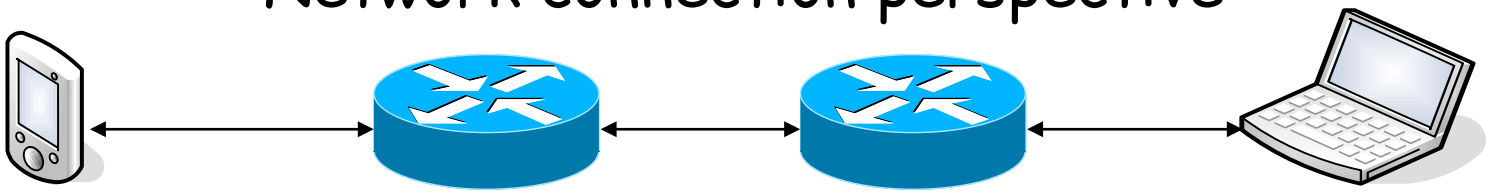
Objectives of the network layer

- Routing
- Addressing

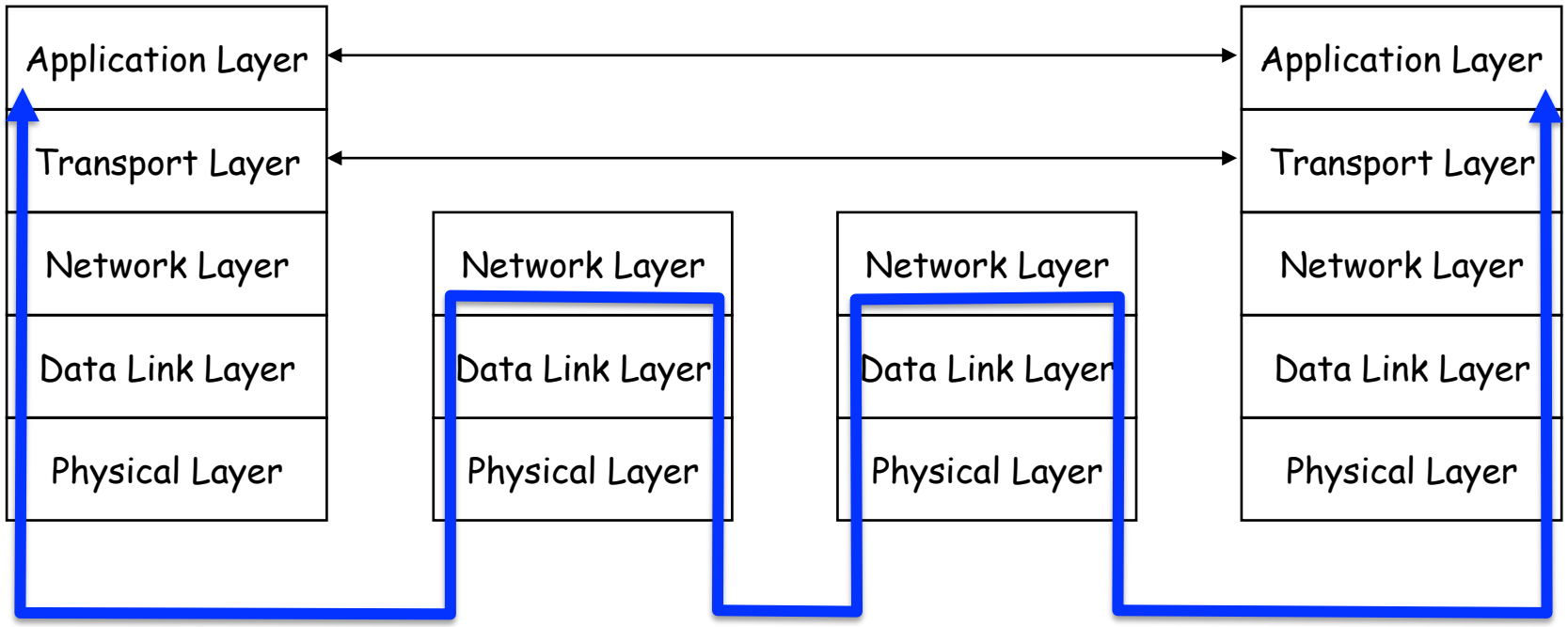


Internet architecture

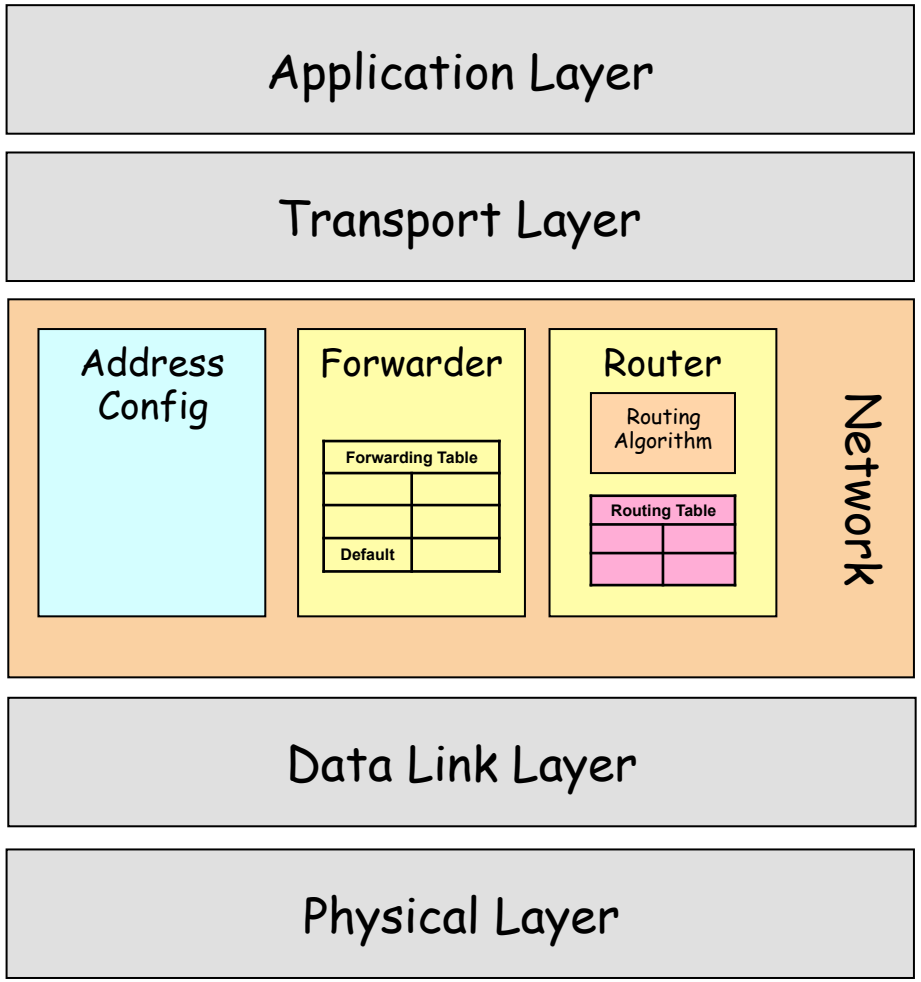
Network connection perspective



Network stack perspective

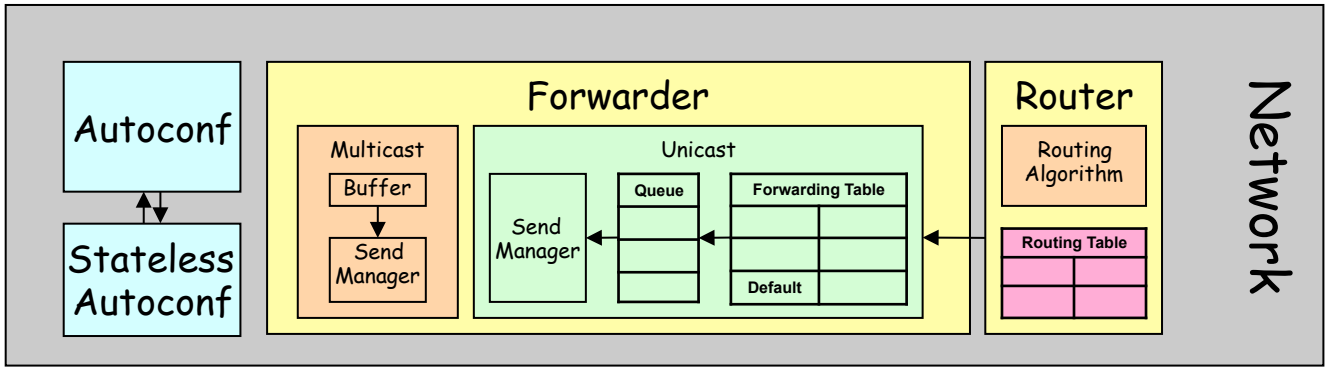


Architecture of a node



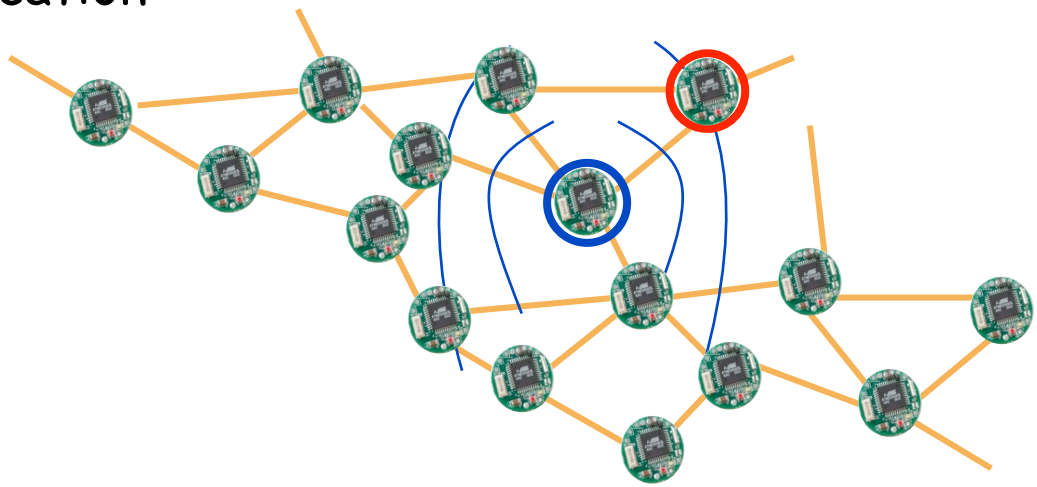
Architecture of a node

- Router
  - Populate the routing table with candidates
  - Manages entries in the forwarding table (FT)
- Forwarder
  - Receive datagram on interface, lookup next hop in FT, request transmission to that
  - Default route
- Address configuration
  - Manage own ID



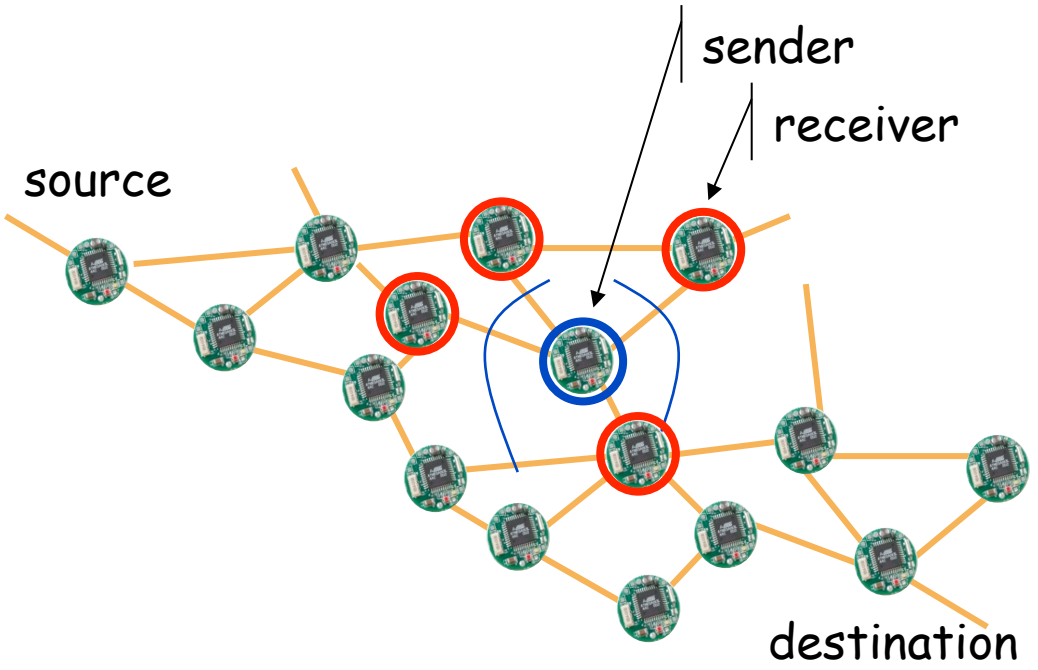
The basic communication primitive

- Transmit a packet
- Received by a set of nodes
  - Dynamically determined
  - Depends on physical environment at the time and what other communication is on-going
  - And further constraints by the link layer
- Each selects whether to retransmit
  - Potentially after modification
- And if so, when

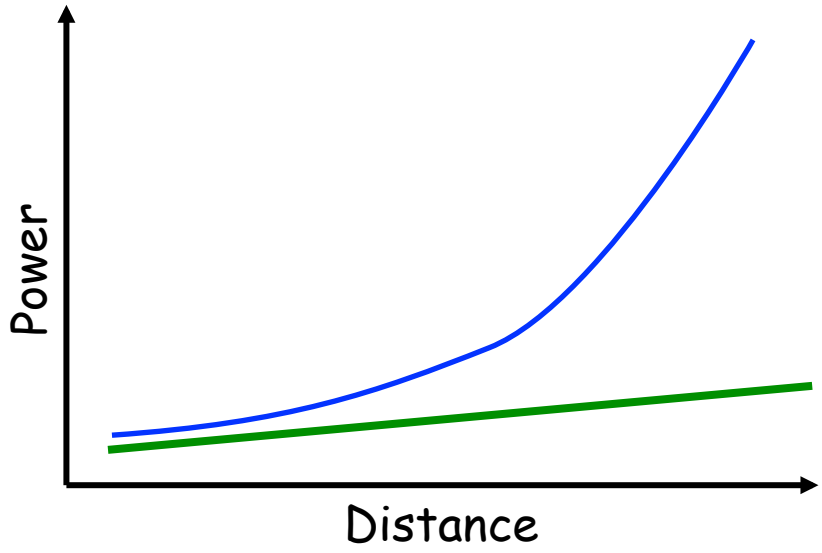


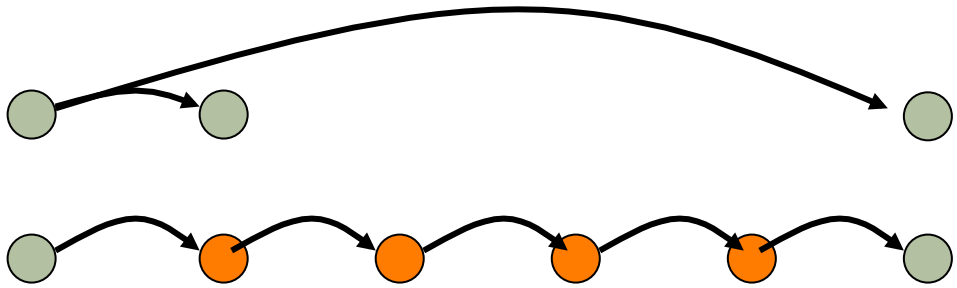
Wireless multi-hop communication

- Upon each transmission, one of the recipients retransmits -> forwards
- Determined by
  - source
  - destination
  - sender
  - receiver
  - etc.



Why multi-hop communication?

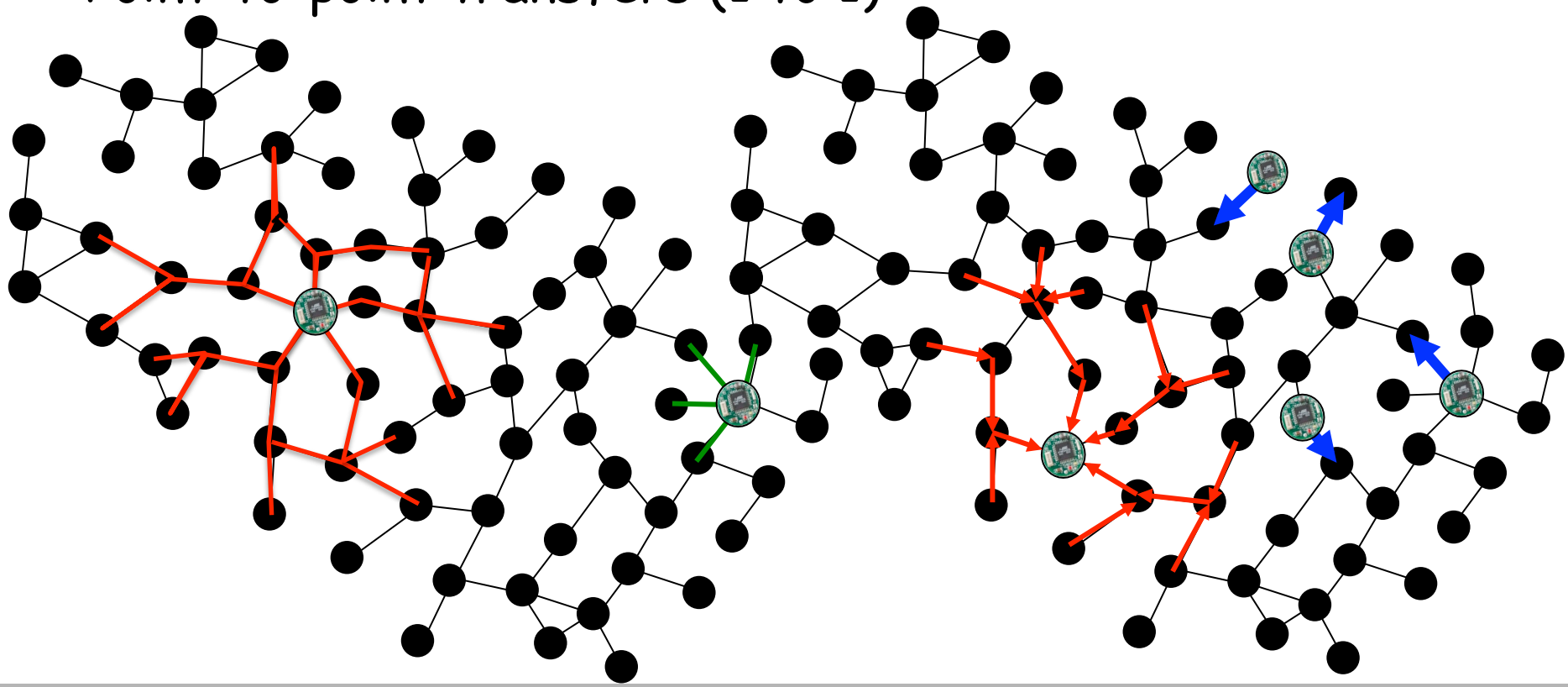




- Power!
  - to transmit d grows as d^3 or worse
  - to route distance d grows linearly
- Bandwidth (spatial multiplexing)
  - With n nodes in a single cell, each gets at most 1/n bandwidth
  - Many small cells -> many simultaneous transmissions.
- Reliability (spatial diversity)
  - Individual links experience interference, obstacles, and multipath effects
  - Even short-range "wireless wires" require human nurturing
    - IRDA, Bluetooth, WiFi, Cell
  - Provides spatial diversity and receiver diversity rather than antenna diversity
  - Protocol level reliability

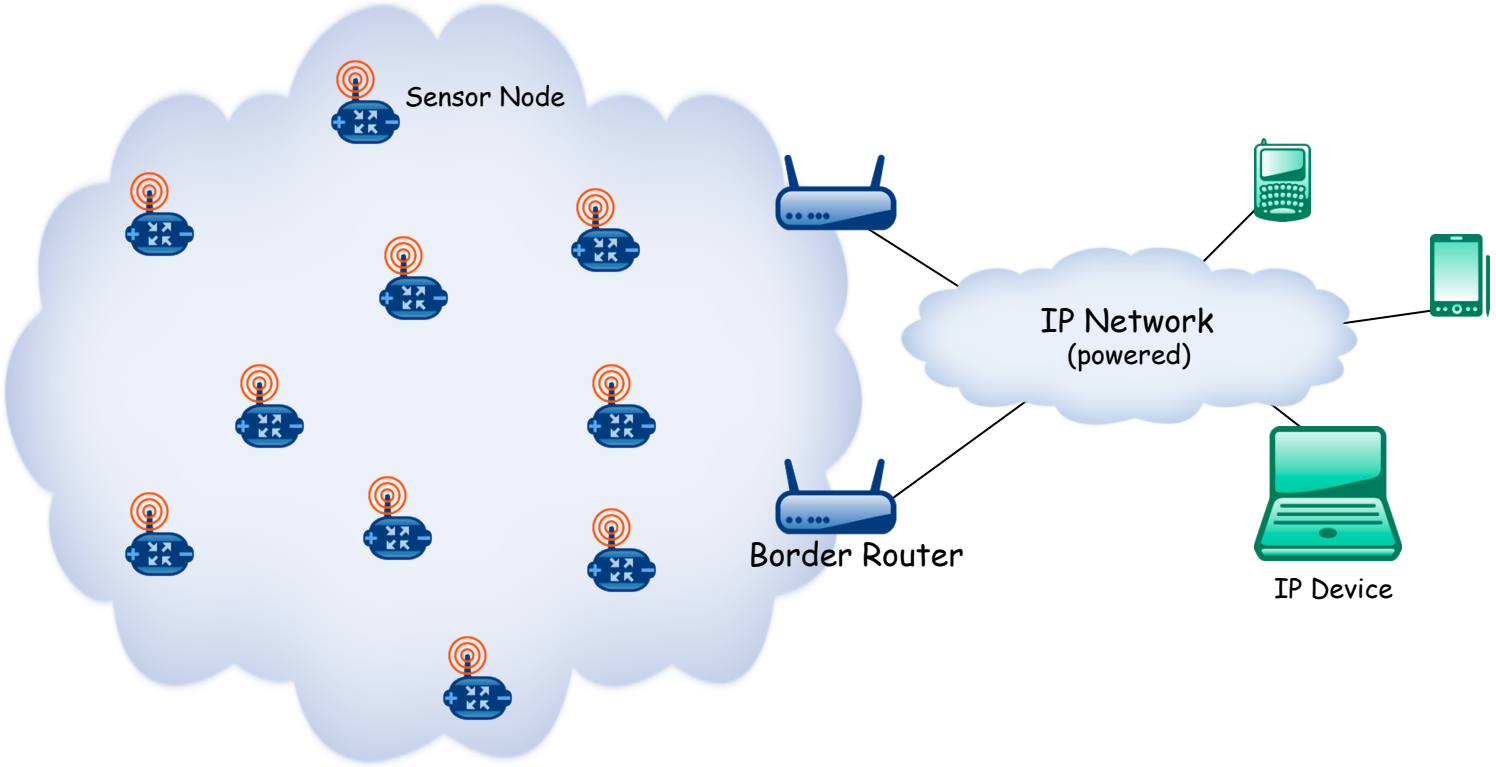
WSN communication requirements

- Dissemination (1 to many)
- Local neighbor communication (1 to few)
- Data collection and aggregation (many to 1)
- Point-to-point transfers (1 to 1)



Embedded network organization

IP based sensor network

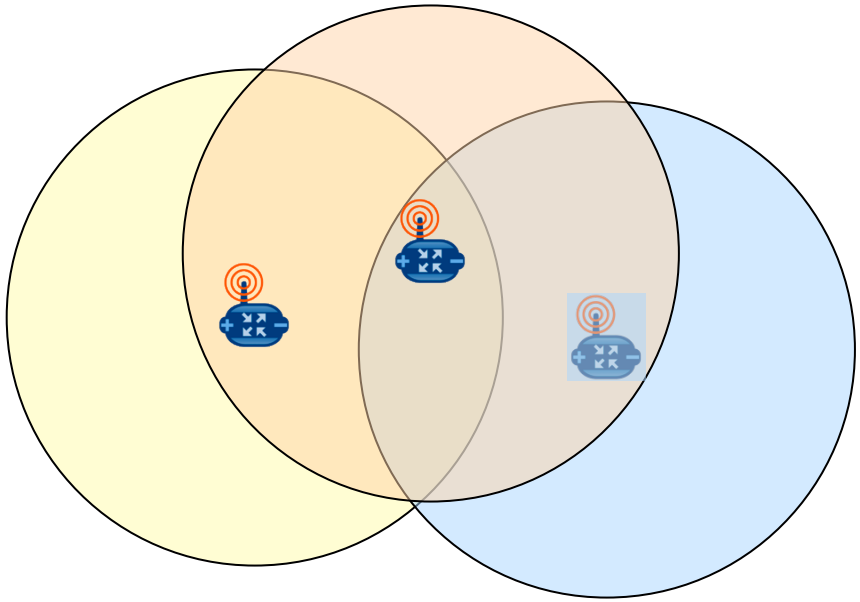


Network architecture: What forms the IP link?

- *IP Protocols assume certain link properties*

- Many assume a full-broadcast domain
  - "Everyone can communicate with each other"
  - Reflexive and transitive reachability
  - Not ad-hoc, wireless networks

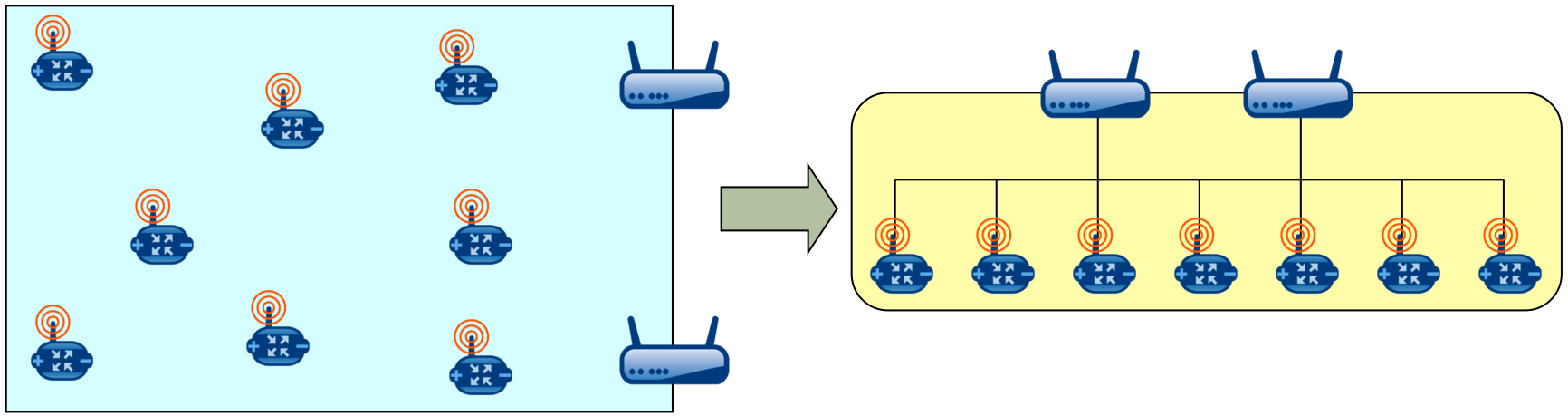
- Examples
  - IPv6 Neighbor Discovery
  - IPv6 Address Auto-configuration
  - ICMPv6 Redirect
  - DHCPv6



Network architecture: PAN = IP Link

Personal Area Network (PAN)

Single IP Link



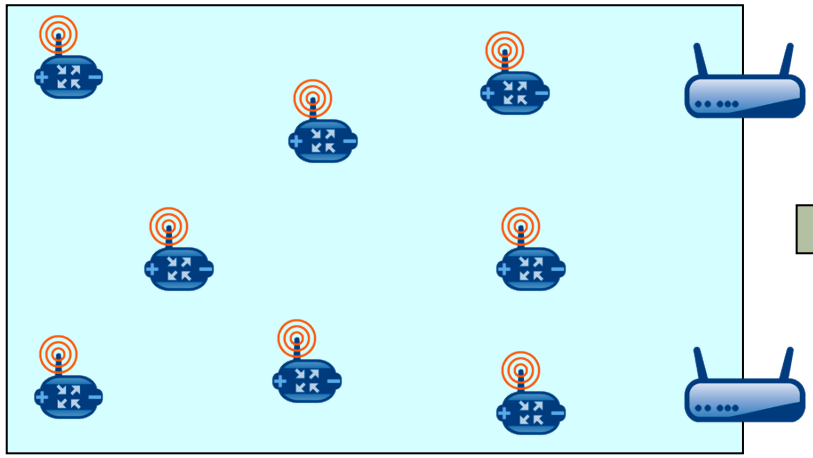
- PAN <-> IPv6 Link-Local Scope
- Emulate reflexive and transitive reachability
- Conceivable to run existing IP-based protocols unmodified

- No IP-level visibility into wireless topology
- Must define subnetwork functionality

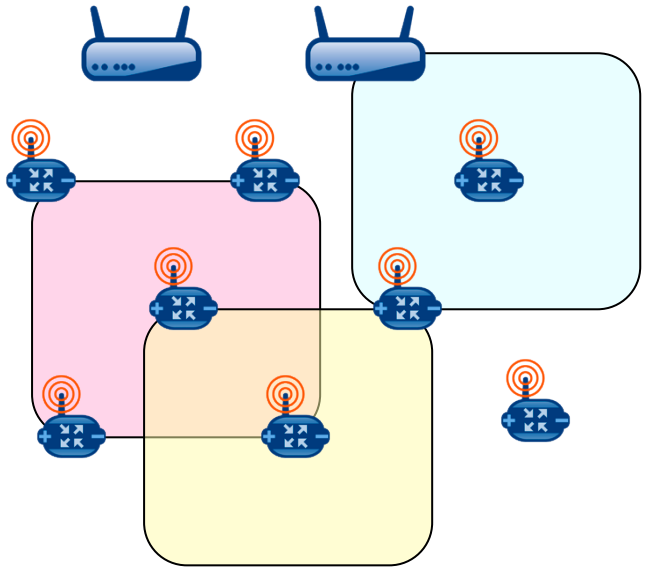
mesh-under

Network architecture: Local Link = IP Link

Personal Area Network (PAN)



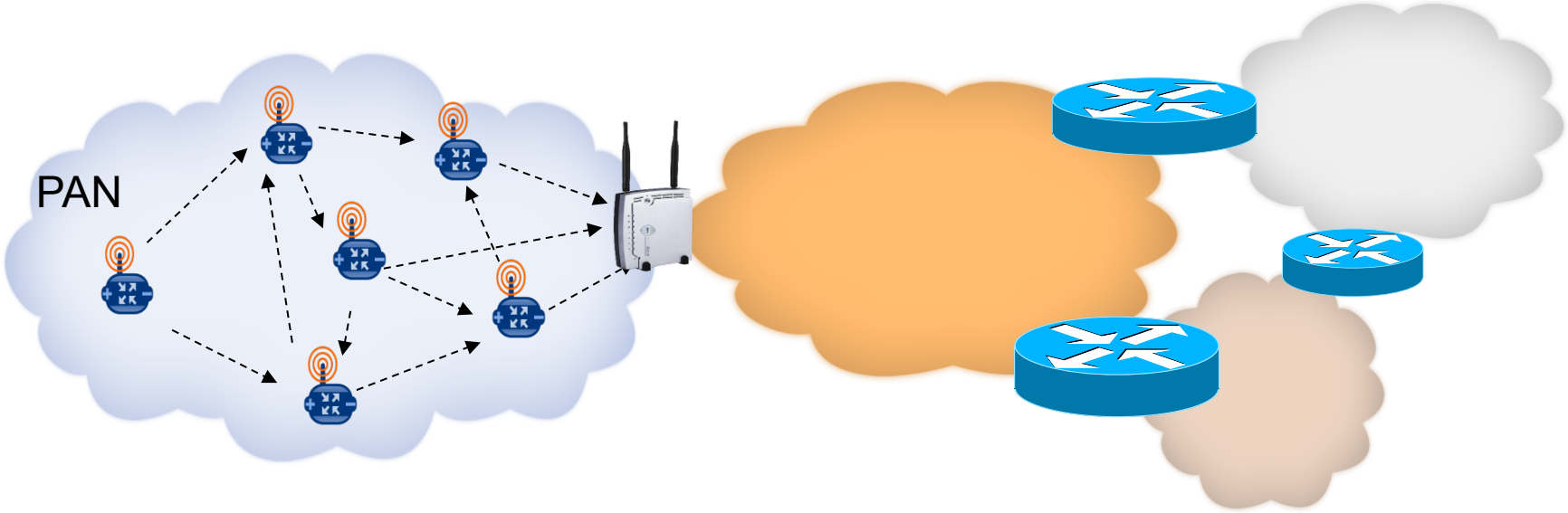
Multiple IP Links



- Radio Range <-> IPv6 Link-Local Scope
- IP-level visibility into link topology
- Routing metrics across other link technologies
- Utilize functionality defined by IP
- Non-reflexive and non-transitive reachability

route-over

Multi-hop communication



- Short-range radios & Obstructions -> multi-hop communication is often required
  - i.e. Routing and Forwarding
  - That is what IP does!
- Mesh-under: multi-hop communication at the link layer
  - Still needs routing to other links or other PANs
- Route-over: IP routing within the PAN
- 6LoWPAN supports both

IP-based multi-hop

- IP has always done "multi-hop"
  - Routers connect sub-networks to one another
  - The sub-networks may be the same or different physical links
- Routers utilize routing tables to determine which node represents the "next hop" toward the destination
- Routing protocols establish and maintain proper routing tables
  - Routers exchange messages with neighboring routers
  - Different routing protocols are used in different situations
  - RIP, OSPF, IGP, BGP, AODV, OLSR, DSR, DYMO, …
- IP routing over IEEE 802.15.4 links does not require additional header information at 6LoWPAN layer
- Vast body of tools to support IP routing
  - Diagnosis, visibility, tracing, management
  - These need to be reinvented for meshing
- IP is widely used in isolated networks too
  - Broad suite of security and management tools

Terminology

- There is no single-hop routing!
- IP routing allows hosts in one network (IP Link) to communicate with hosts in another
- **Topology Formation**: determining the connectivity graph, i.e., the network
- **Routing**: Protocols and process for establishing what paths are used in communicating over that graph and setting up tables
- **Forwarding**: process of receiving messages, looking up the next hop, and transmitting them
- **Meshing**: some combination of formation, routing, and forwarding that occurs at the link layer (L2) transparently to the network layer

Meshing vs routing

- Conventional IP link is a full broadcast domain
  - Routing connects links (i.e, networks)
- Many IP links have evolved from a broadcast domain to a link layer "mesh" with emulated broadcast
  - Ethernet -> switched ethernet
  - IEEE 802.11 -> IEEE 802.11s
- Utilize high bandwidth on powered links to maintain the illusion of a broadcast domain
- IEEE 802.15.4 networks are limited in bandwidth and power so the emulation is quite visible.
- Routing at two different layers may be in conflict
- On-going IETF work in ROLL working group
  - Routing Over Low-Power and Lossy networks (RPL)

Classical view of routing

- Connectivity between nodes defines the *network graph*
  - Topology formation
- A **routing algorithm** determines the sub-graph that is used for communication between nodes.
  - Route formation, path selection
- Packets are **forwarded** from source to destination over the routing sub-graph
  - At each node in the path, determine the recipient of the next hop
- The selection at each hop is made based on the information at hand
  - Sender address, current address, destination address, information in the packet, information on the node
  - Table-driven, source based, algorithmic, etc.
  - Who knows the route?
  - Do you determine it as you go?

Basic routing approaches

- Link state
  - Nodes shout (send) and listen (receive) to determine neighbor connectivity
  - Each floods this information throughout (Link State Advertisement) so every node has a map (Link state data base) of the network
  - Any node can determine the path or the next hop
  - Management protocol deals with changes in connectivity
  - Classic Example: OSPF

Basic routing approaches

- Distance vector
  - Nodes maintain routing information about “distance” and “direction” to destinations
  - Choose next hop by comparing the cost of routing through neighbors
    - Cost(dest D, neighbor b) = linkCost(b) + pathCost(b,D)
  - Management propagates routing information
    - Sequence numbers, etc.
  - Classic Example: RIP

What is different in WSN?

- There is no *a priori* network graph
  - It is discovered by sending packets and seeing who receives them
  - The link relationship is not binary
    - pairs of nodes communicate with some probability that is determined by many of factors
  - It is not static

- The embedding of the "network" in space is important
  - Need to get information to travel between particular physical places
  - But the "communication range" is not a simple function of distance

What is different in WSN?

- Addressing & Naming
  - Address -> ID for a node?
  - Name -> ID for content?

- Addressing & Naming
  - Flat EUID?
  - Hierarchical IP?
  - Topologically meaningful?
  - Spatially meaningful?
  - Based on content?

Wireless routing protocols

http://datatracker.ietf.org/wg/manet/charter/

- Many wireless protocols in the IP context have been developed in the IETF MANET (Mobile Ad Hoc Networking) working group in the context of IEEE 802.11 links carrying traditional TCP/IP traffic
  - AODV Ad hoc on-demand distance vector
  - OLSR Optimized link state routing
  - DSDV Destination sequenced distance vector
  - DSR Dynamic source routing
  - TDRPF Topology dissem. based on reverse-path forwarding
  - ARA Ant routing algorithm

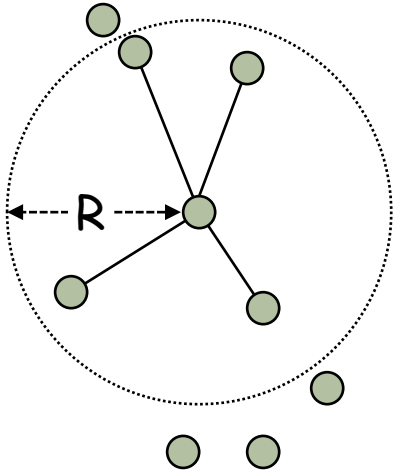
- Assume a fairly "classic" view of connectivity
  - Naïve radio

Link and routing metrics

Link and routing metrics

- Much of the "paper protocols" define connectivity graph with unit disk model

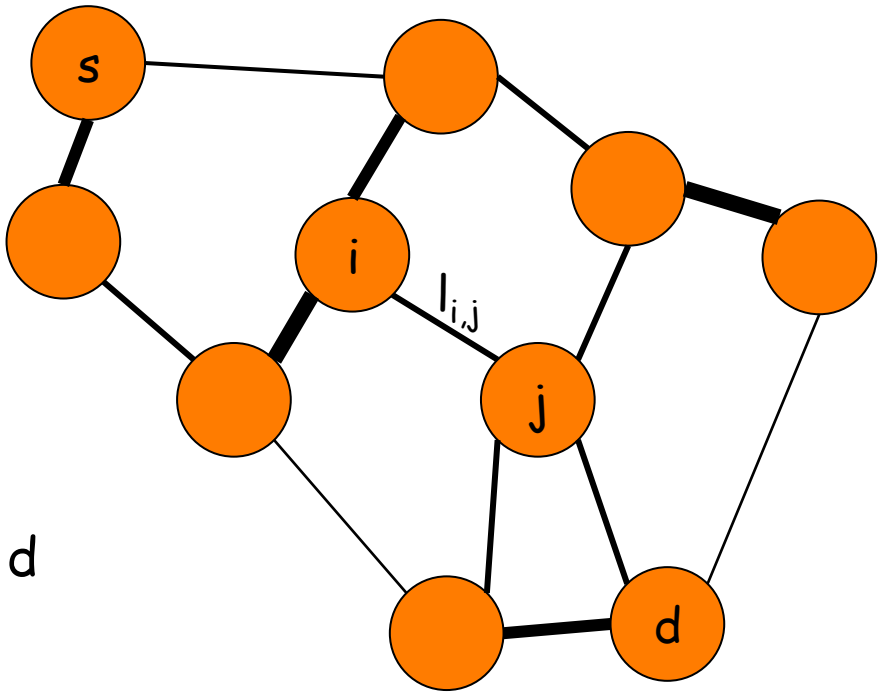
  Link(A,B) iff dist(A,B) ≤ R



- OK for rough calculations, but not for protocol design
  - Nearby nodes may not be able to communicate
  - Far away nodes may be able to communicate
  - Nodes that communicated in the past may not be able to communicate in the future
  - Nodes may have intermittent communication depending on external factors

Link and routing metrics

- To asses the "quality" of a link or a route
- Link metric $l_{i,j}$
  - Quality of the "link" between nodes i and j
- Routing metric
  - Quality of the "route" between source s and destination d
- Challenge
  - Links vary continuously
  - Topology may change continuously



Link and routing metrics

Link metrics

Binary link metric

- Simple metric
- Link is either perfect or not available

$$l_{A,B} = \begin{cases} 1 & \text{if } A \text{ and } B \text{ can communicate} \\ 0 & \text{else} \end{cases}$$

- Pro
  - Bidirectional metric
- Challenge
  - Requires a rule to decide A and B can communicate

Packet delivery ratio

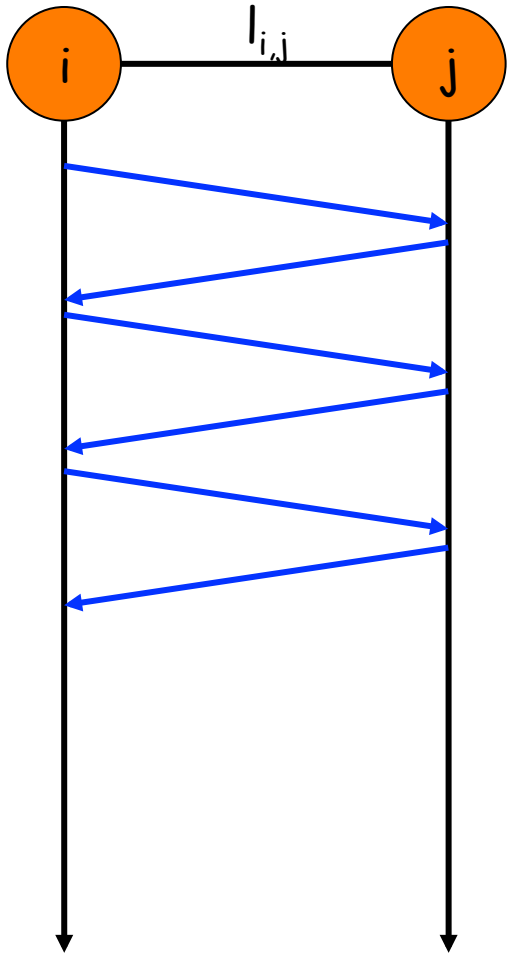
- Packet delivery ratio (PDR)
- Sender transmits n measurement packets, r of them are received correctly at receiver

$$l_{A,B} = PDR_{A,B} = \frac{r}{n}$$

- Pro
  - For a source A all neighbors can compute $PDR_{A,i}$
- Challenges
  - Time window
  - Asymmetry -> $PDR_{A,B} \neq PDR_{B,A}$

Round trip time (RTT)

- Time as link metric?
- Pro
  - Considers inherently used bit rate
- Challenge
  - Time interval
  - Technology may use different bit rates for
  - Clock accuracy



Expected transmission count (ETX)

- Metric for bidirectional links
- Estimates the expected number of transmissions for a successful transmission

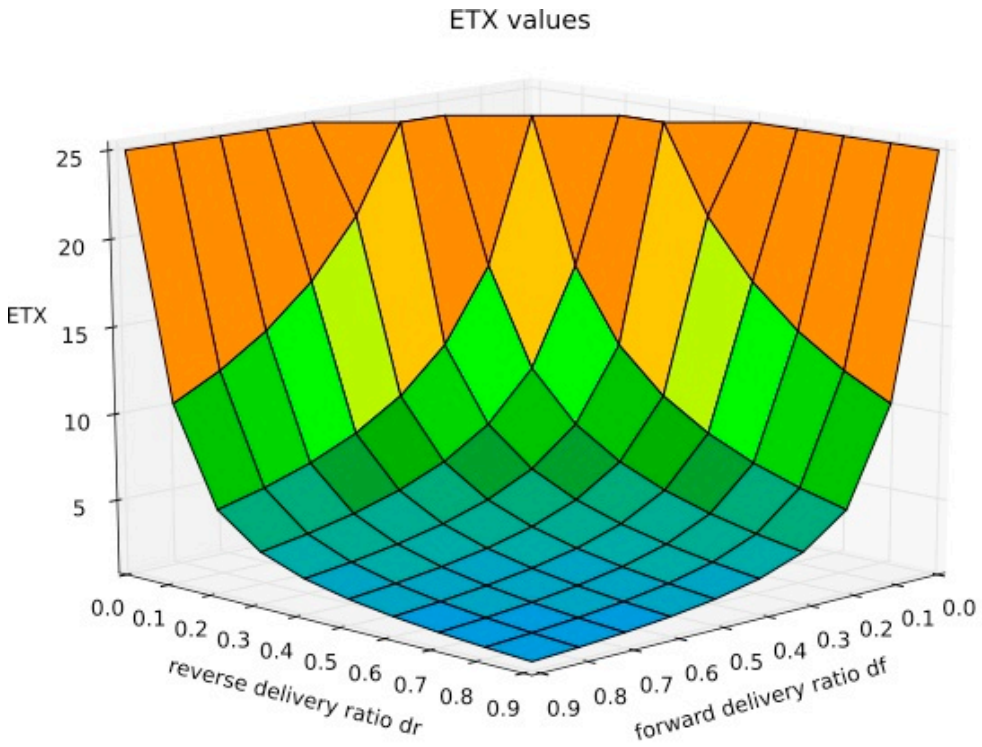
$$l_{A,B} = ETX_{A,B} = \frac{1}{p_f \cdot p_r} \qquad p_f, p_r \in (0,1]$$

Probability for successful transmission from A to B

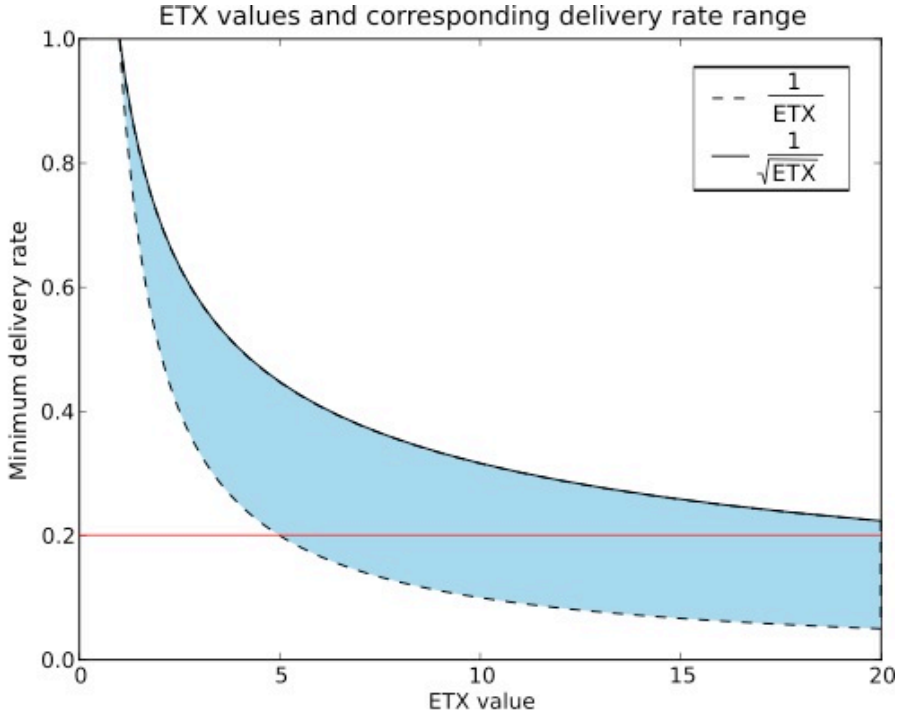
Probability for successful transmission from B to A

Expected transmission count (ETX)

[Juraschek]

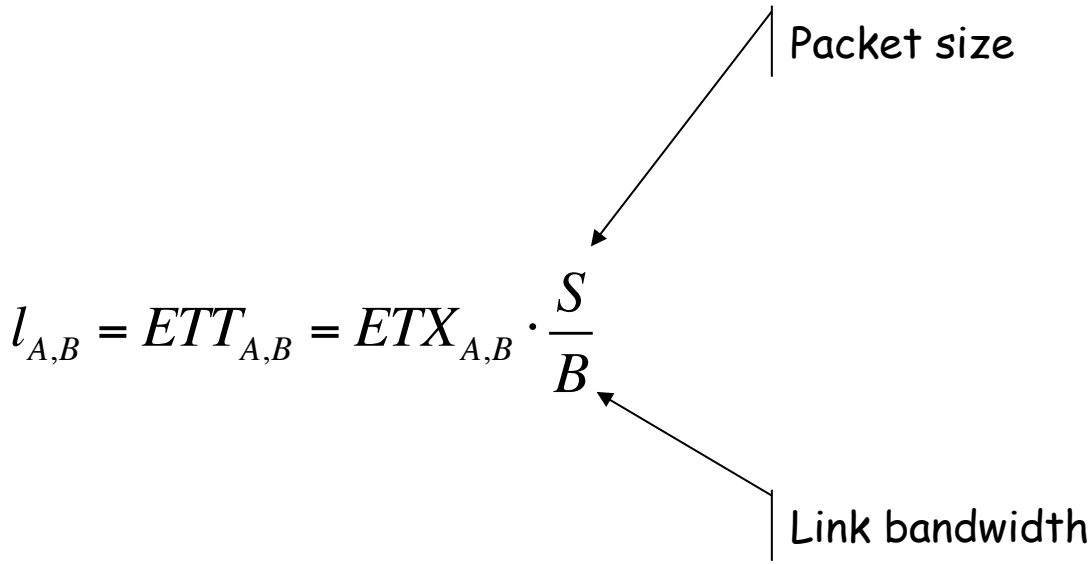


(a) ETX values for reverse and forward delivery rates



(b) Range of minimum delivery rates for particular ETX values

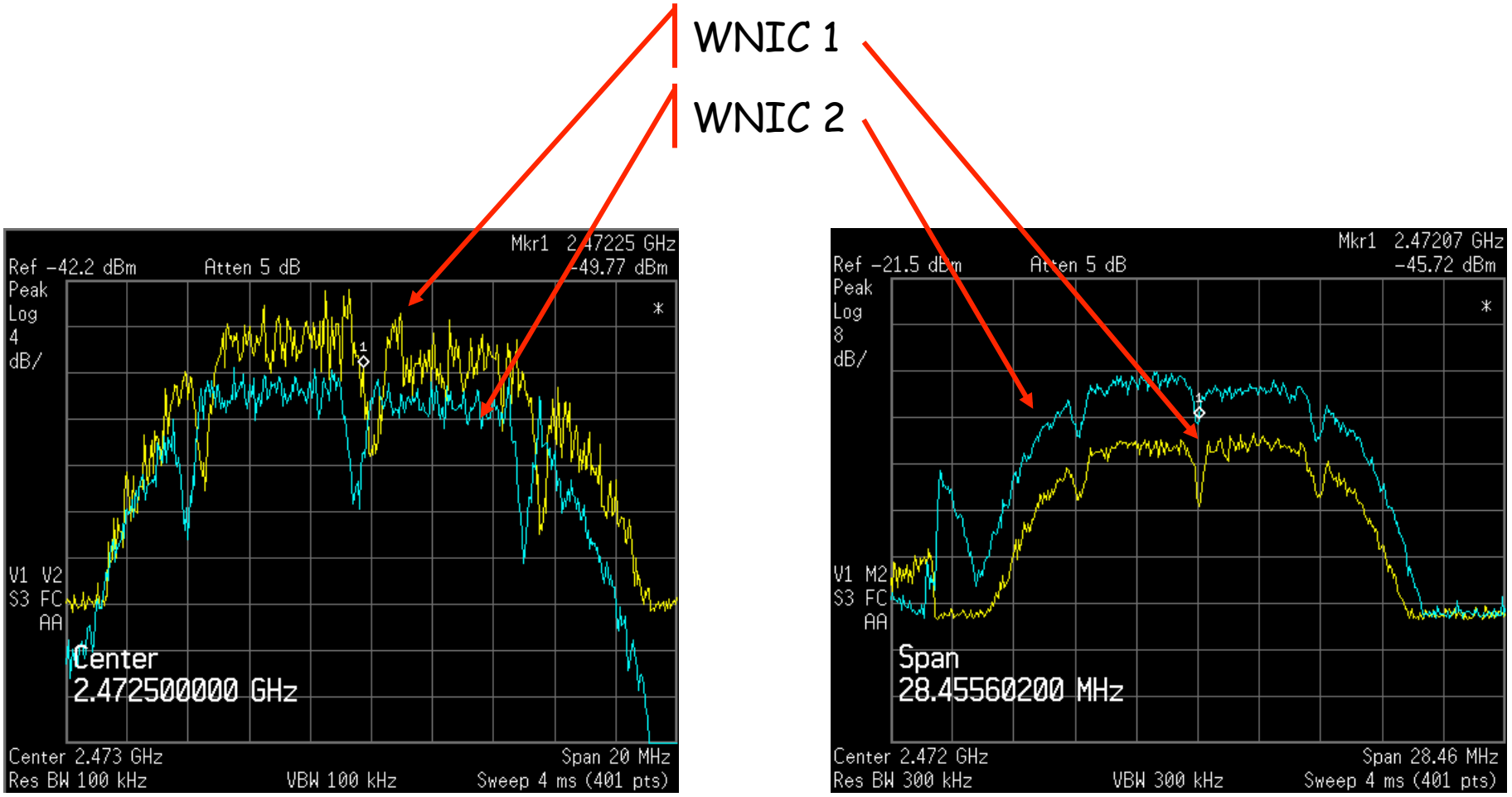
Expected transmission time (ETT)



RSSI

- Receive Signal Strength Indicator (RSSI) specifies the signal strength of a received packet
- The RSSI is computed at the interface of the receiver and does not include any information from the sender
- The RSSI value is typically specified in dBm and maps the signal strength from mW

RSSI

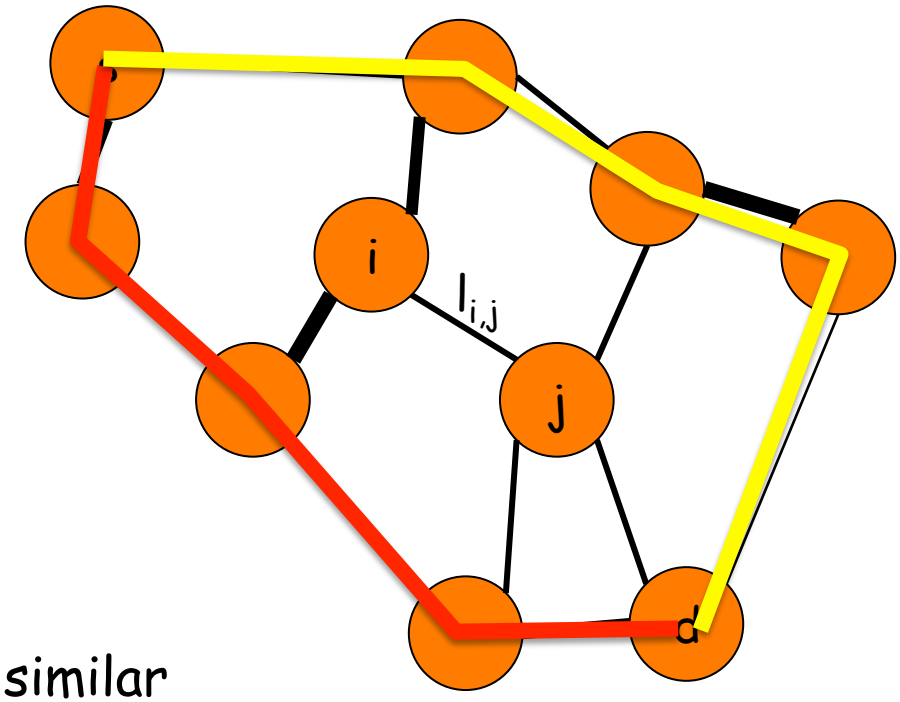

WNIC 1
WNIC 2
Ref -42.2 dBm
Atten 5 dB
Mkr1 2.47225 GHz
-49.77 dBm
Peak
Log
4
dB/
V1 V2
S3 FC
AA
Center
2.472500000 GHz
Center 2.473 GHz
Res BW 100 kHz
VBW 100 kHz
Span 20 MHz
Sweep 4 ms (401 pts)
Ref -21.5 dBm
Atten 5 dB
Mkr1 2.47207 GHz
-45.72 dBm
Peak
Log
8
dB/
V1 M2
S3 FC
AA
Span
28.45560200 MHz
Center 2.472 GHz
Res BW 300 kHz
VBW 300 kHz
Span 28.46 MHz
Sweep 4 ms (401 pts)

Link and routing metrics

Routing metrics

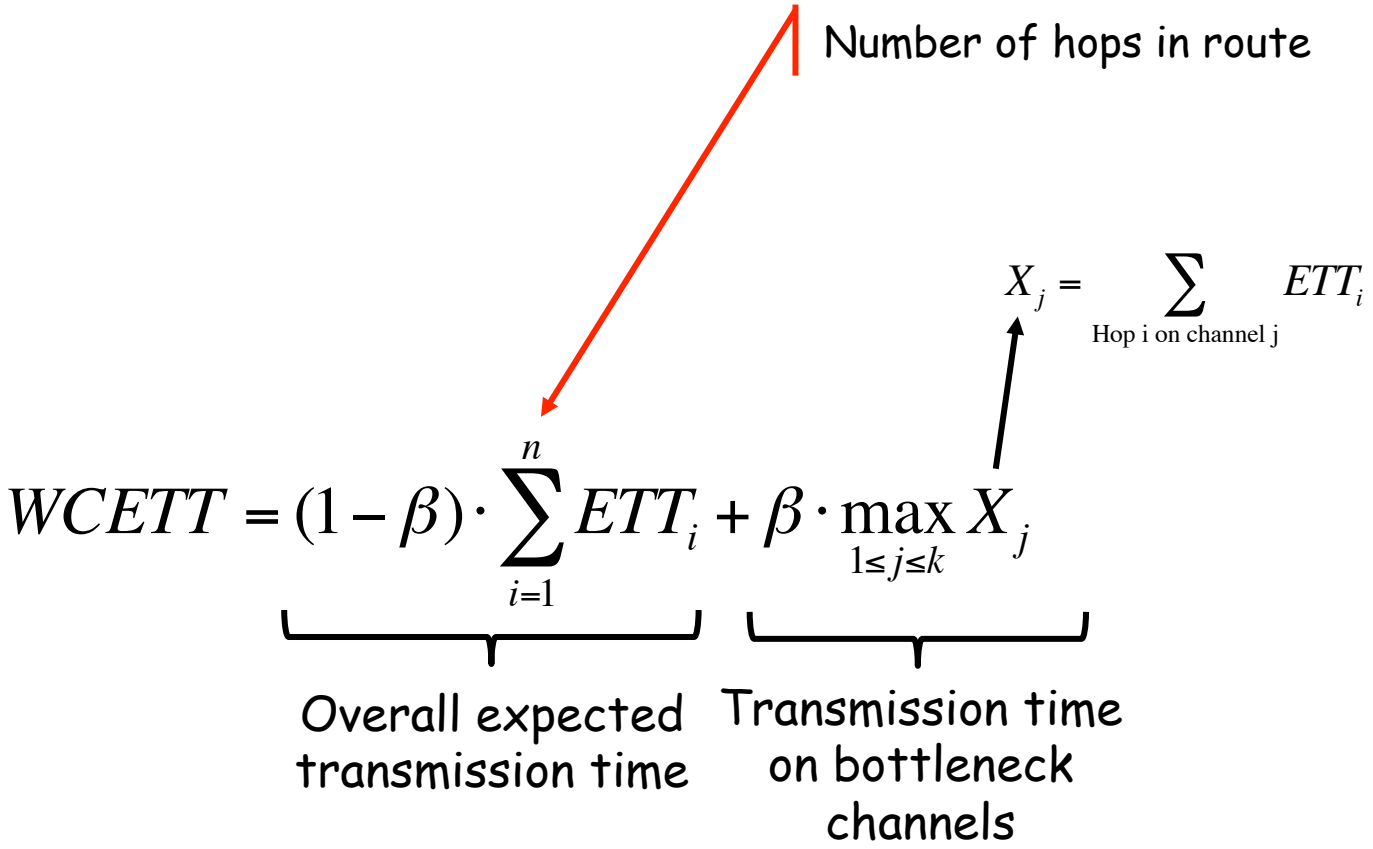
Hop count

- The hop count = the number of hops between a source and destination node
- As routing metric
  - the minimum hop count is used



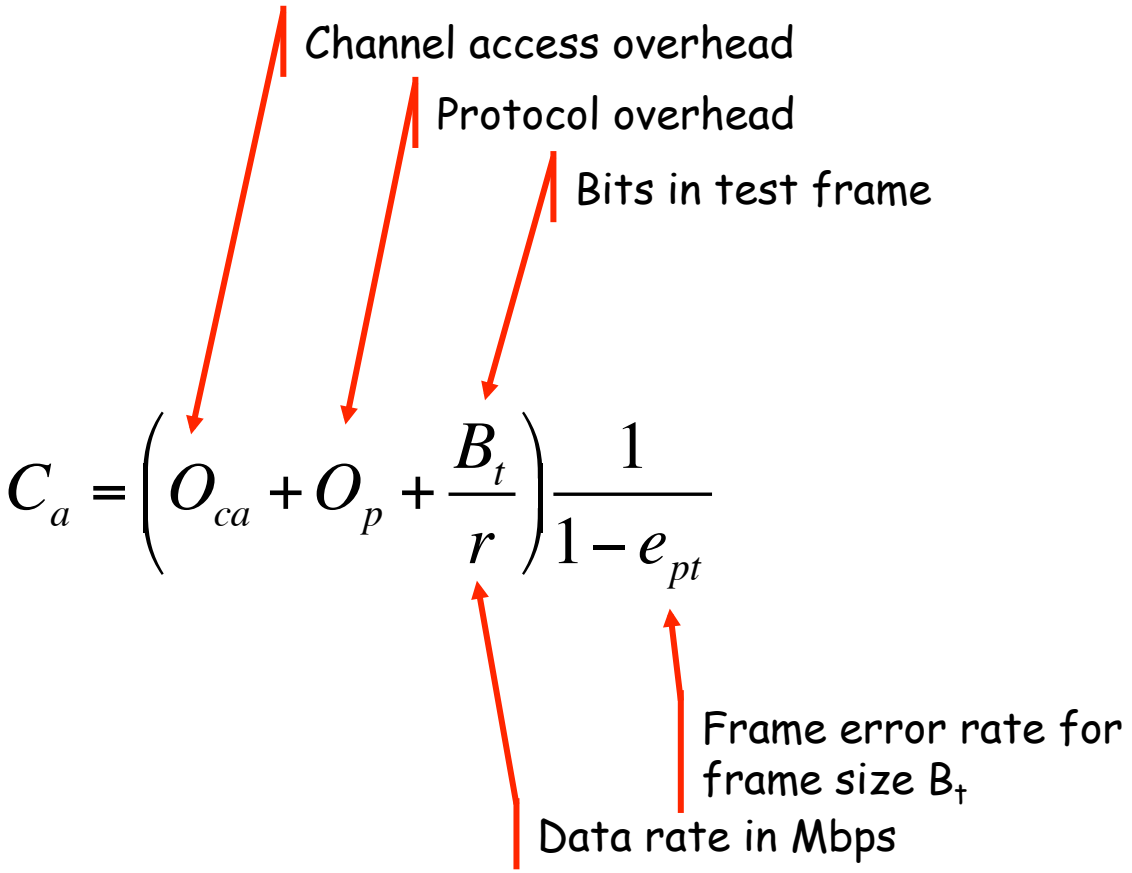
- Pro
  - Very simple
  - Ignores link quality
  - Results "good" if all links behave similar
- Contra
  - Ignores link quality
  - Results "bad" if links vary highly

Weighted cumulative expected transmission time (WCETT)



Airtime cost routing metric

- Default routing metric in IEEE 802.11s



Summary: Link and routing metrics

- Many "paper" metrics proposed
- From link metrics to routing metrics
  - Additive
  - Multiplicative

- Open research subject
  - How to come from a "good" link metric to a "good" routing metric

Probabilistic routing

Probabilistic routing

- Flooding
- Gossiping
- Rumor routing
- Ant routing algorithm

Probabilistic routing

Flooding

Flooding

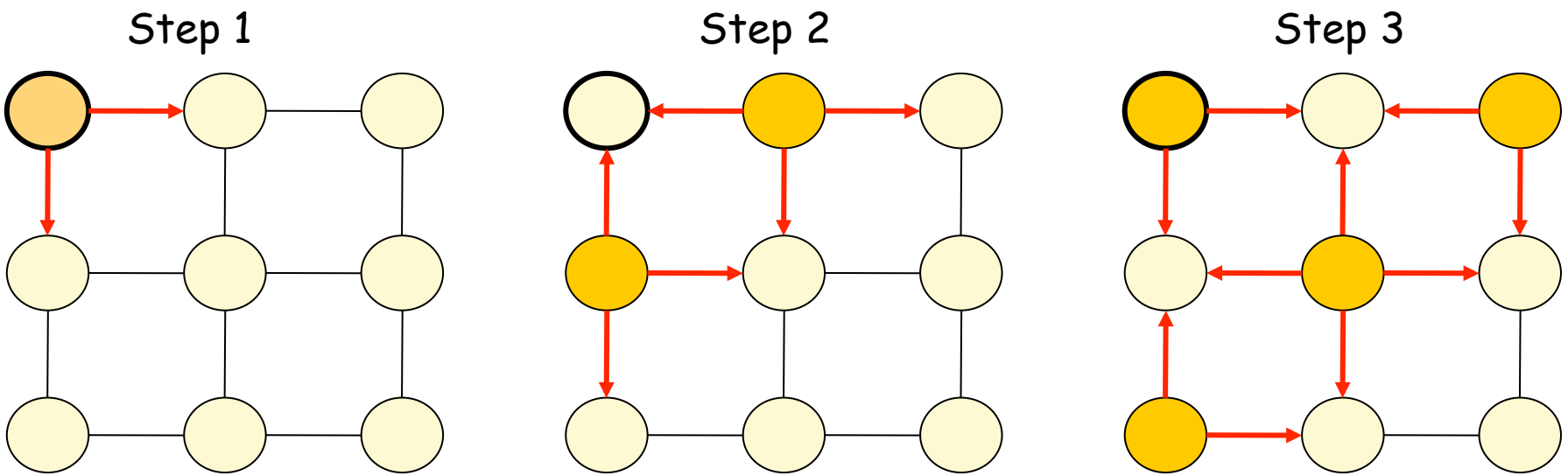
- Root broadcasts a "new" message to local neighborhood
- Each node performs a simple rule

```
if ("new" incoming msg) {
      take local action;
      retransmit modified msg;
}
```

- No underlying routing structure required
  - The connectivity over physical space determines it

Flooding

- Approach
  - Sender node broadcasts packet to all neighbors
    - Broadcast-Address on MAC layer
  - All nodes, that received the packet forward the packet (broadcast again)
- Properties
  - Distributed and self-organizing
  - No routing tables required
  - High network load



Flooding

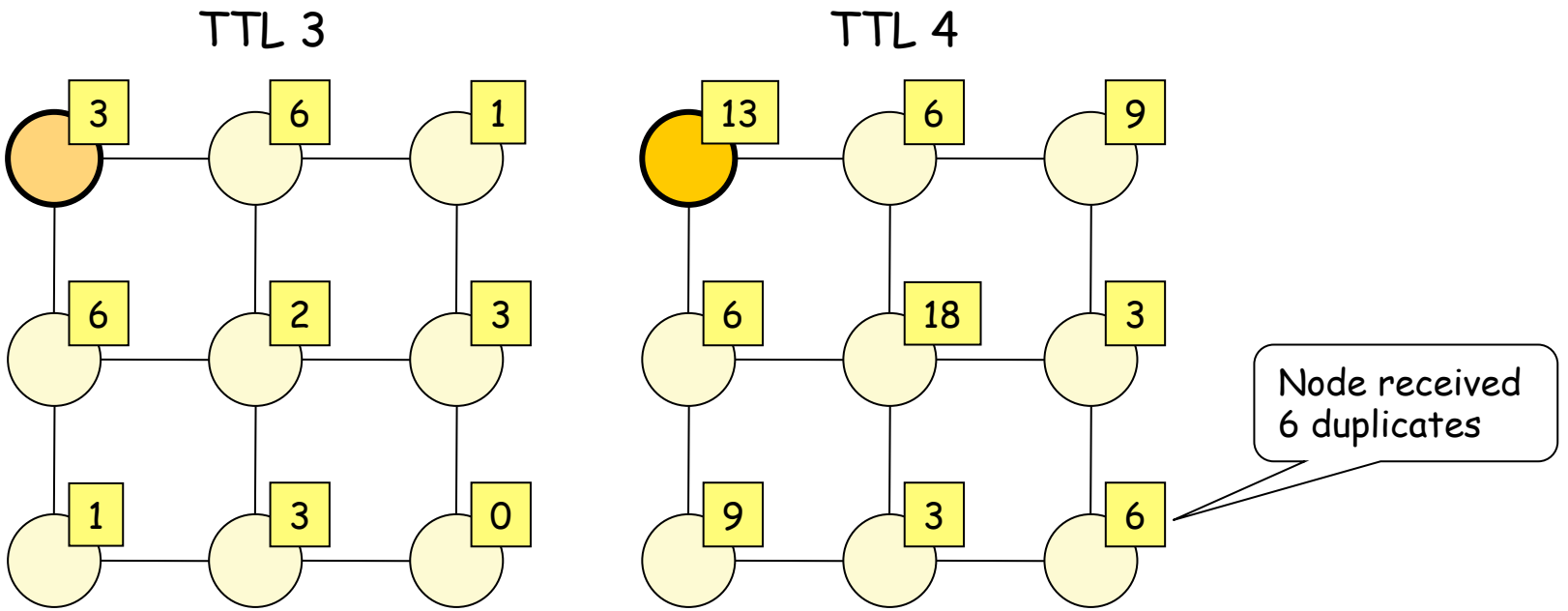
- Advantages
  - No route discovery required
  - No route maintenance required
  - No routing table or similar required
- Disadvantages
  - Implosion: packets are retransmitted many times
  - Not adapted to limited resources (i.e., bandwidth, energy, ...)
    - High network load -> many collisions -> energy waste
  - No termination rule: packets may loop forever
- ... many variations and improvements

Flooding: Improvement

- Goal
  - A node forwards a packet only once
- Requirements
  - Nodes need knowledge about forwarded packet -> state
  - Unique ID of packets -> ?
- Applicability for WSN
  - Requirements are not easy to fulfill for distributed systems
    - How to recognize duplicates?
    - Limited memory available -> Structure of state?
  - Probabilistic approaches for duplicate recognition

Flooding: Limiting the TTL

- Approach
  - Limit the forwarding by TTL -> Drop packet if it reaches the max. hop count
- Pro
  - Reduction of the network load -> reduction of collisions
- Contra
  - No reliability -> which nodes receive a packet?



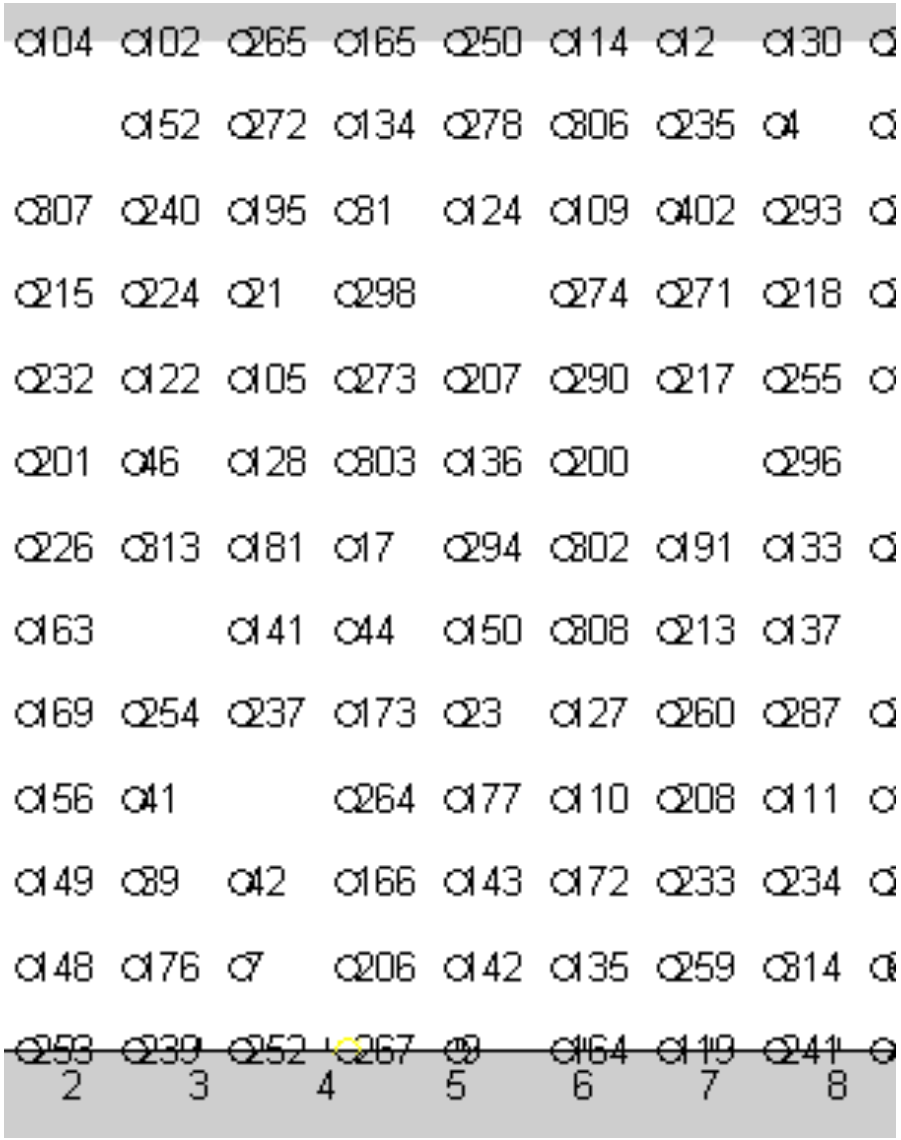
Flooding

- Route free dissemination is extremely useful in its own right
  - Disseminate information
    - Router advertisements, solicitations, …
  - Network-wide discovery
  - Join, …

- It is also the network primitive that most "ad hoc" protocols use to determine a route
  - Flood from source till destination is reached
  - Each node records the source of the flood packet
    - This is the parent in the "routing tree"
  - Reverse the links to form the path back

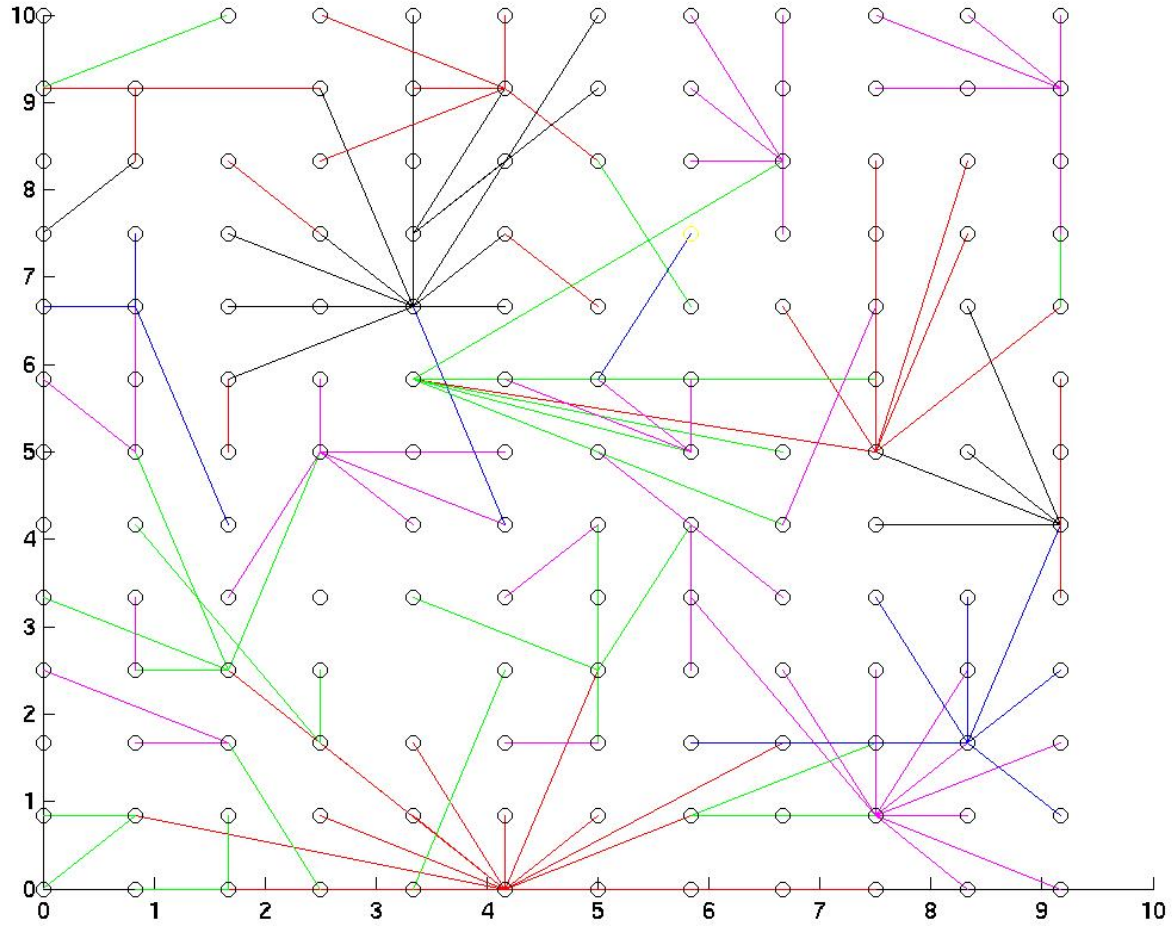
Flood Dynamics

[Ganesan]



- Experimental Setup
  - 13x13 grid of nodes
  - separation 2ft
  - flat open surface
  - Identical length antennas, pointing vertically upwards.
  - Fresh batteries on all nodes
  - Identical orientation of all nodes
  - The region was clean of external noise sources.
- Range of signal strength settings
- Log many runs

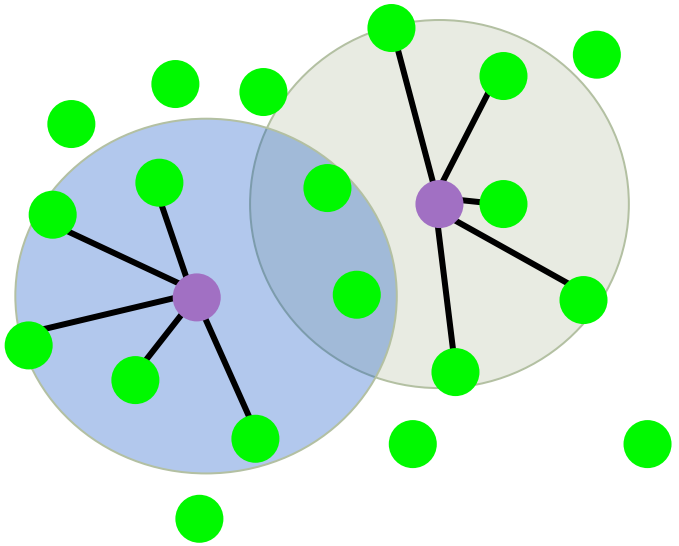
Final Tree

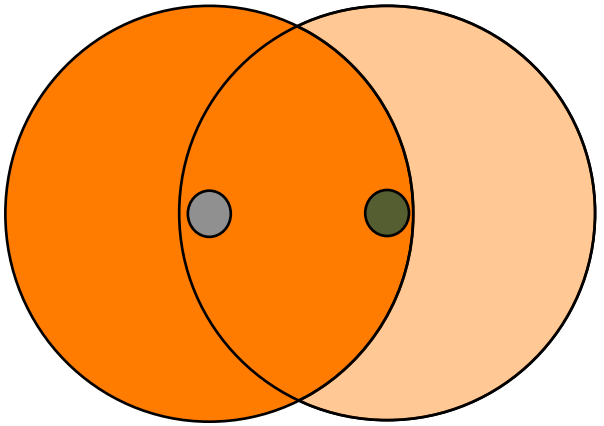


Factors

[Ni]

- Long asymmetric links are common
  - Many children
- Nodes out of range may have overlapping cells
  - Hidden terminal effect
- Collisions -> these nodes hear neither ‘parent’
  - Become stragglers
- As the tree propagates
  - folds back on itself
  - rebounds from the edge
  - picking up these stragglers.
- Redundancy
  - Geometric overlap -> <41% additional area





Probabilistic routing

Gossiping

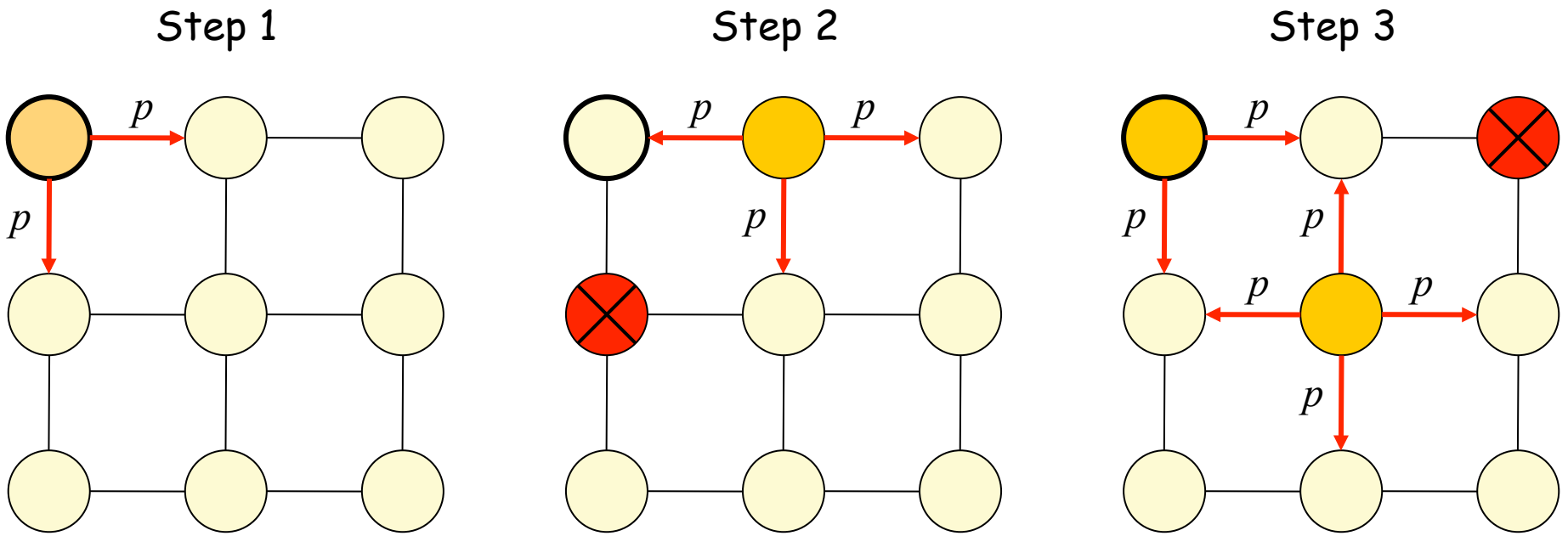
Gossip & Rumor

Gossiping

- Basic idea
  - Similar to the circulation of rumors
    - Tell a story to a limited number of people -> everybody in the neighborhood knows
  - Application to routing
    - Forward packets probabilistically to some neighbor nodes
- Approach
  - A node forwards packets with probability p
    - Broadcast on MAC layer -> all 1-hop neighbors receive packet
    - Node does not forward packet with probability (1-p)
  - Parameter p
    - Typically: $0.65 \leq p \leq 0.75$
    - With $p < 0.65$ high probability that gossiping dies -> packet does not reach destination

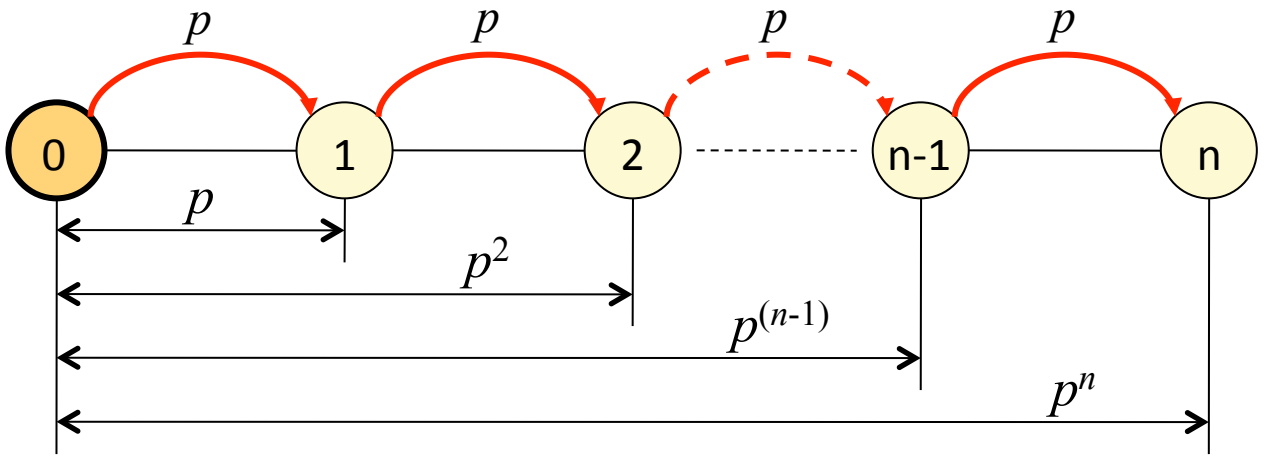
Gossiping: Example

- A node forwards packet with probability p



Gossiping

- Pro
  - No implosion of packets in the network
  - Reduced network load compared to flooding
- Contra
  - Long transmission time in the network
  - Packet may be transmitted over a long path
  - No reliability
    - Only some of the nodes receive the packet -> here it is a "feature"



Gossiping: Variants

- Gossip0(p)
  - All nodes gossip with p
- Gossip1(k,p)
  - First k hops flood than gossip with p
- Gossip2(k, p1, n, p2)
  - First k hops flood than gossip with p1. If fewer than n neighbors gossip with p2
- Gossip3(k, p, m)
  - First k hops flood than gossip with p. If you do not hear at least m copies gossip
- Gossip4()
  - Gossip in your zone.
- Gossip5(k, p1, n, p2, m) = Gossip2() + Gossip3()
- Gossip6(k, p, d) = Gossip3() + Dynamic adaption of p
  - p = p/(d+1) // d = number of heard copies
- Gossip7()
  - Count heard copies of msg
- Gossip8(pmin, pmax)
  - Dynamic adaption of p
- Gossip9()
  - Dynamic adaption of p based on number of neighbor estimation

Gossiping: Variants


[Haas]

- Idea of Gossip1(k, p)
  - In the first k hops Flooding, afterwards Gossiping with p
- Parameters
  - Gossip1(k, 1) -> Flooding
  - Gossip1(0, p) -> Traditional Gossiping

- Evaluation
  - Assumptions
    - Regular network topology, middle size network (~ 1000 nodes)
    - Perfect MAC protocol (no packet loss, no collisions)
  - Results
    - Parameter p=0.72 and k=4
    - Almost all nodes receive the packet

Gossiping

[Haas]

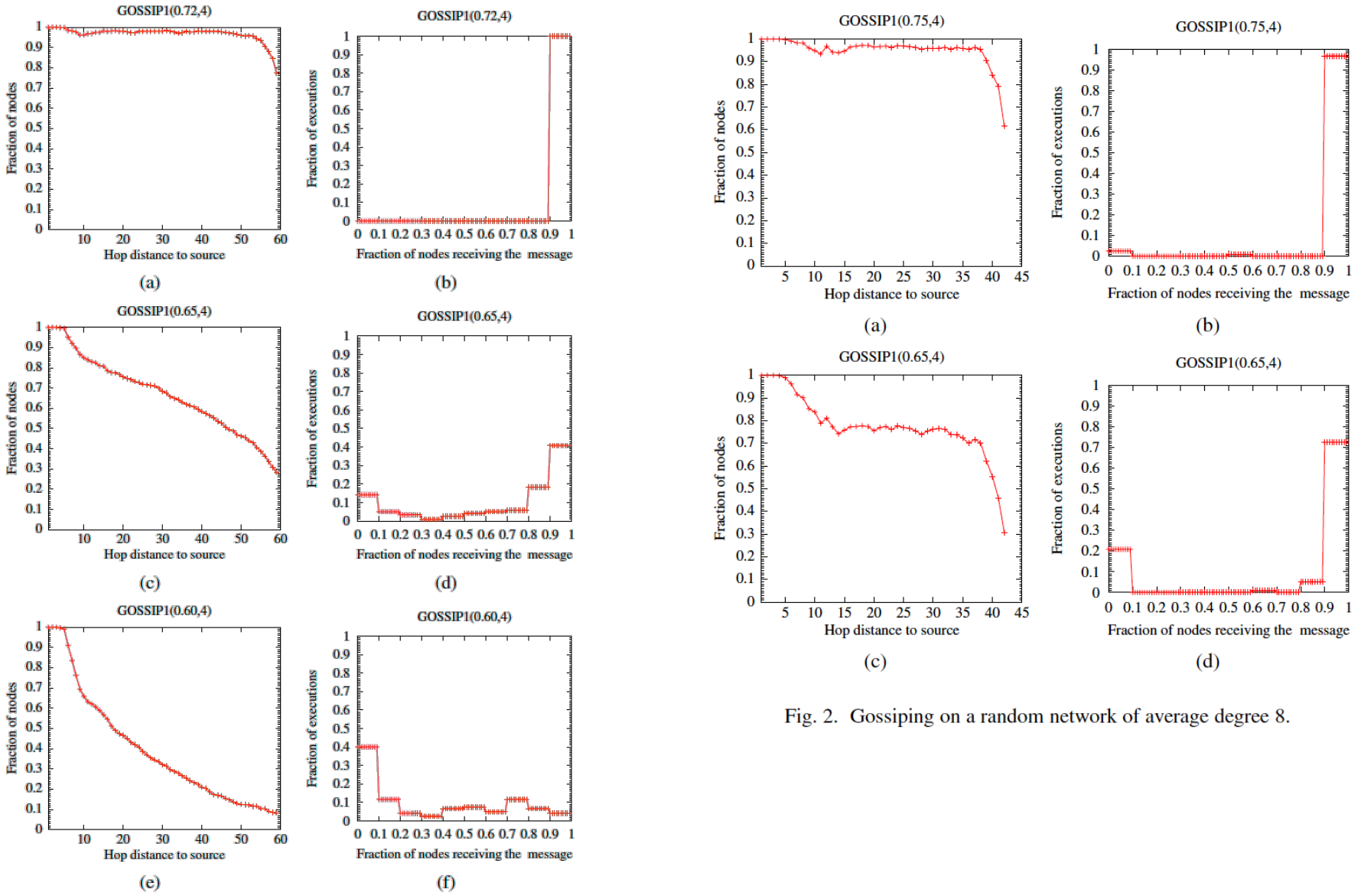
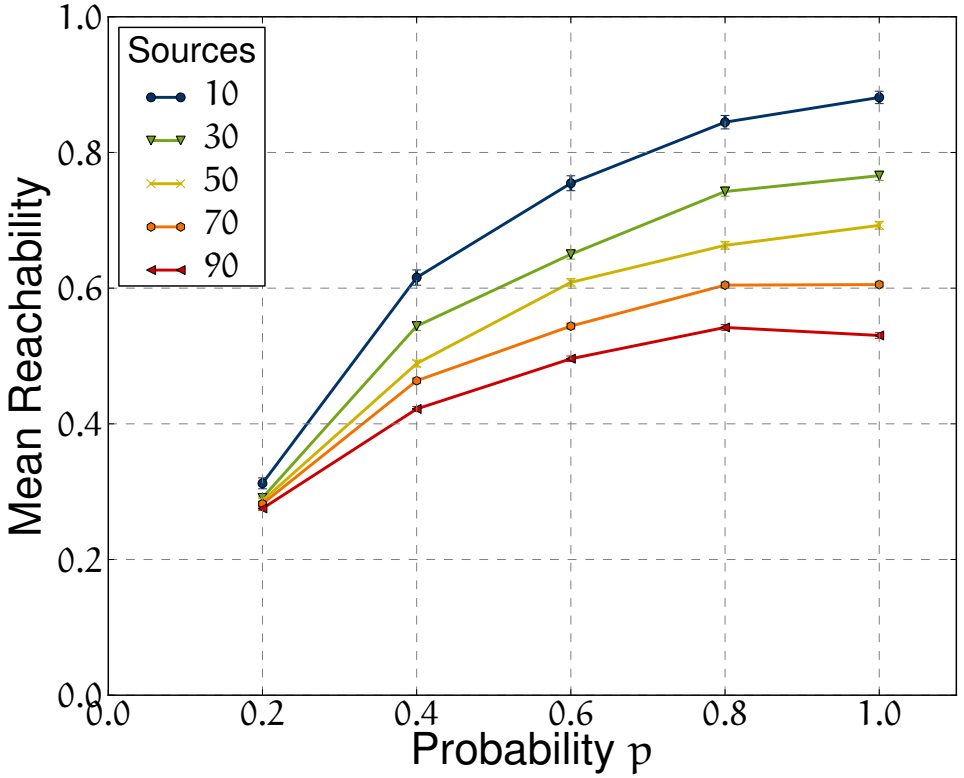


Fig. 2. Gossiping on a random network of average degree 8.

Gossip

[Blywis]

- Experiments on the DES-Testbed, FU Berlin
- Reachability for different
  - numbers of parallel sources
  - probabilities p



DES-Testbed, FU Berlin

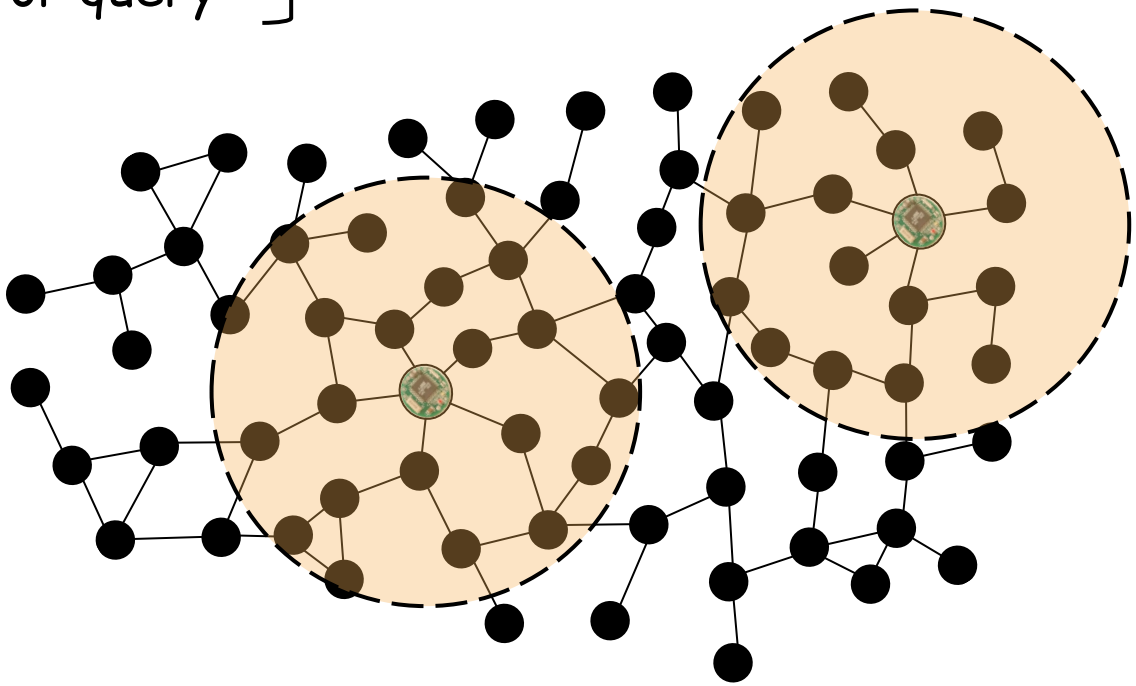
Probabilistic routing

Rumor routing

Rumor routing

- Goal: Inform a node (the sink) about an event
  - Event sink: A node that wants to be informed about an event
  - Event source: A node that observes events
- Classical approach
  - Flooding of event
  - Flooding or query

  } High overhead



Rumor routing

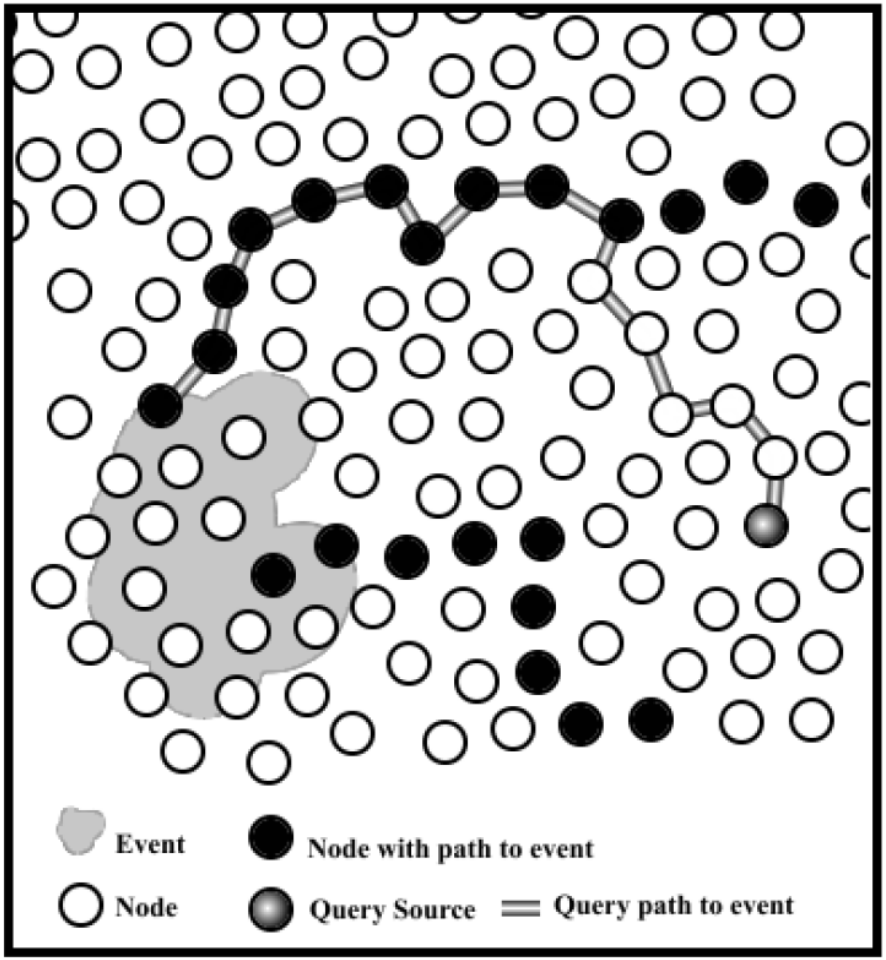
- General
  - Entity based addressing and routing
  - Based on the idea of Random Walks
  - Analogy to the dissemination of rumors
- Assumptions
  - Suboptimal path is better than Flooding
  - Transmission of few packets
- Goal: Transport of queries and replies (Event Notification)
  - Event is abstract, identifies set of sensor values, etc.
  - Event is local phenomena, i.e., temperature in a room
- Properties
  - Initiative may start from event source or sink
  - Compromise between flooding of events and flooding of queries

Rumor routing

- Idea: Interpret observed event as an agent, that walks through the network
  - Random Walk
    - Choose randomly direction and speed
  - Simplest form of Random Walk
    - Forward packet to any one of the neighbors
    - Packets are transmitted per unicast (MAC addresses are used)
  - Variant
    - Forward several copies of a packet to some randomly chosen neighbors
- Agents
  - Nodes create agents when they detect an event
    - Agents are generated with some probability
    - Agents walk through the network and generate path information (event path)
    - Dissemination of the agents is limited
  - Biological analogy -> ants

Rumor routing: Basic idea

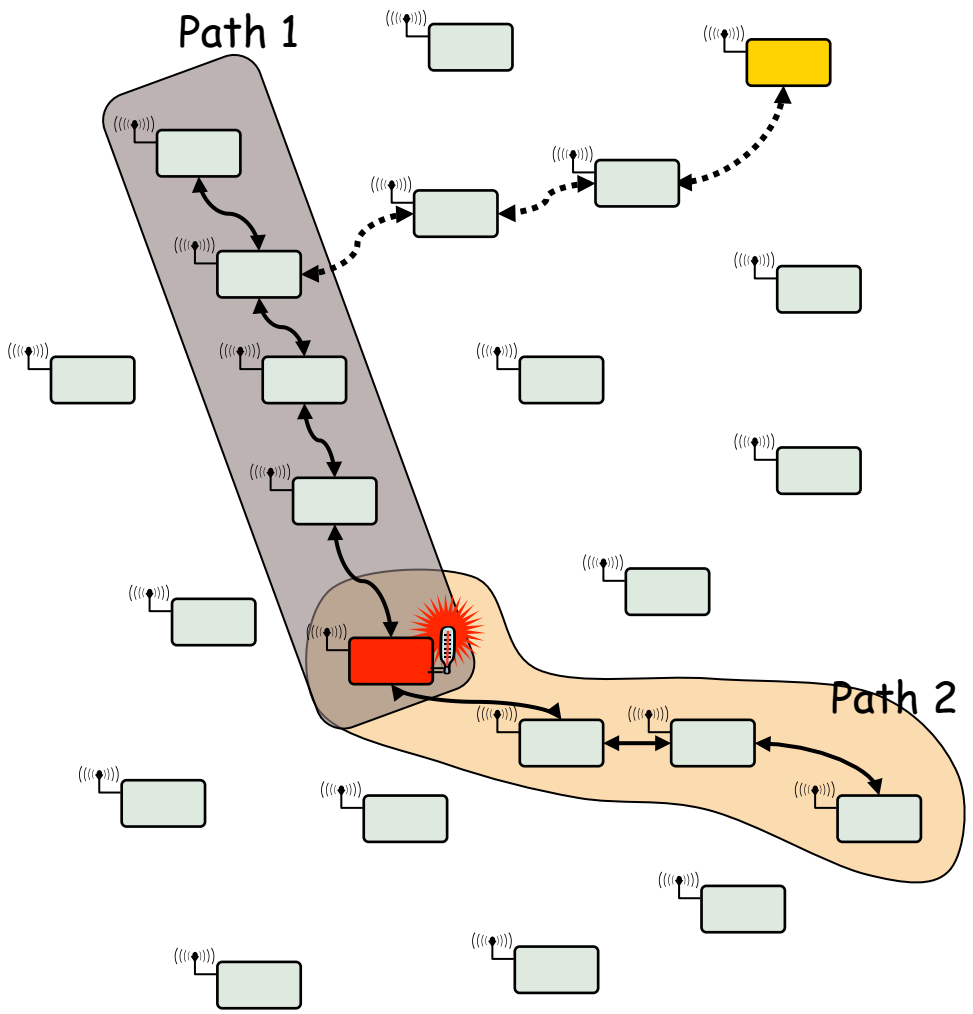
- Rumor routing approach
  - Event agents create route information about events
  - A sink which is interested in an event sends a search agent on several paths
  - Query is sent on a random walk until it finds the event path
  - Search agents are forwarded until they cross an event-path
  - Thus, the route to an event is discovered
- Setting path is important



Query is originated from the query source and searches for a path to the event. As soon as it finds a node on the path, it is routed directly to the event.

Rumor routing: Basic idea

- Node observes an event
  - Sends two agents
  - Agents install routing information about the events
    - Path 1 and Path 2
- Sink sends an query agent
  - Search agent travels across the event-path
    - Path 1 in the example
  - Path to the event is known



Rumor routing: Algorithm overview

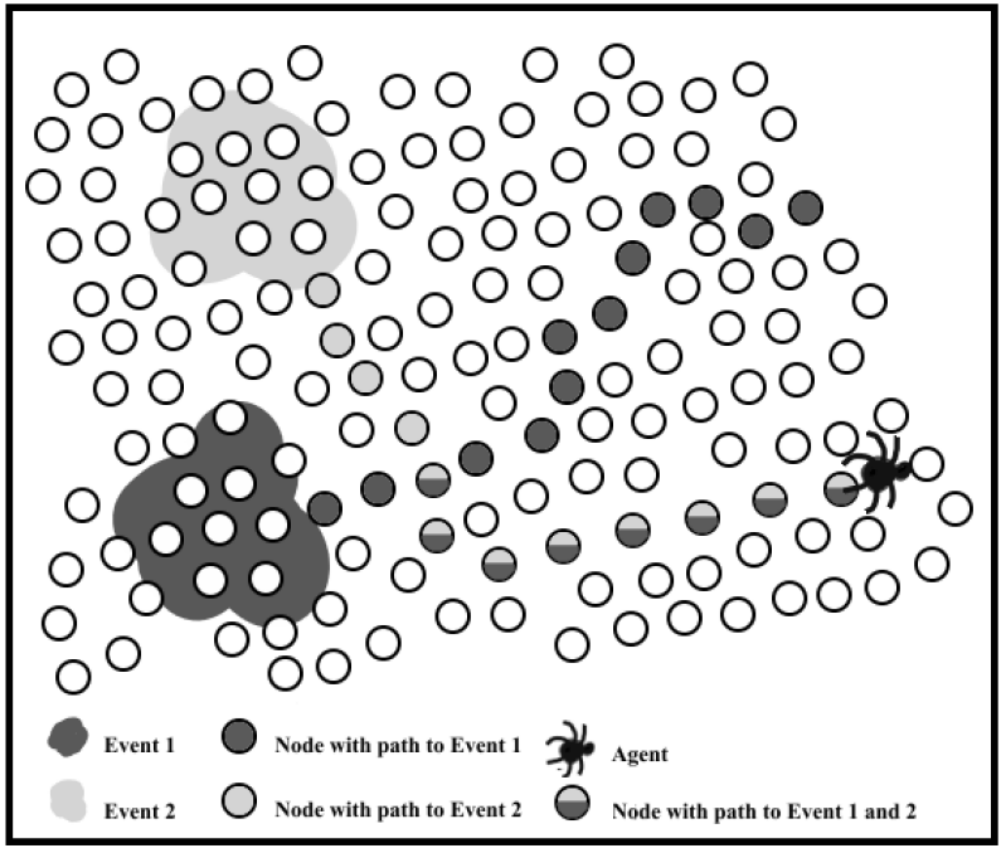
- Each node maintains a list of neighbors and its event table.
- An agent is a long-lived packet, propagating information about local events to distant nodes.

- Any node can generate a query
  - Query is forwarded in a random direction to find the path
  - Query is forwarded until its TTL expires or finds the path

- If query dies, the node can act …
  - retransmit
  - give up
  - flood

Rumor routing: Agents

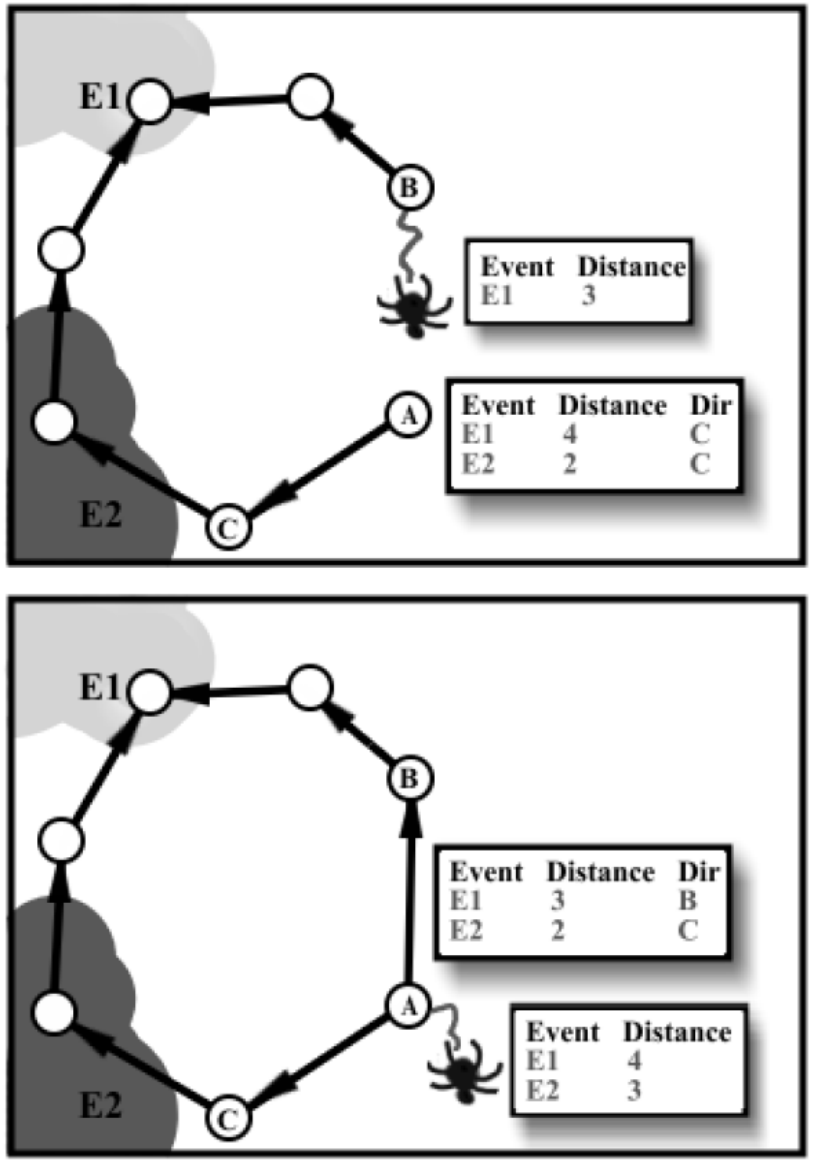
- Set up path by random walk
- Aggregate paths



When an agent propagating the path to Event 2 comes across a path to Event 1, it begins to propagate the aggregate path to both.

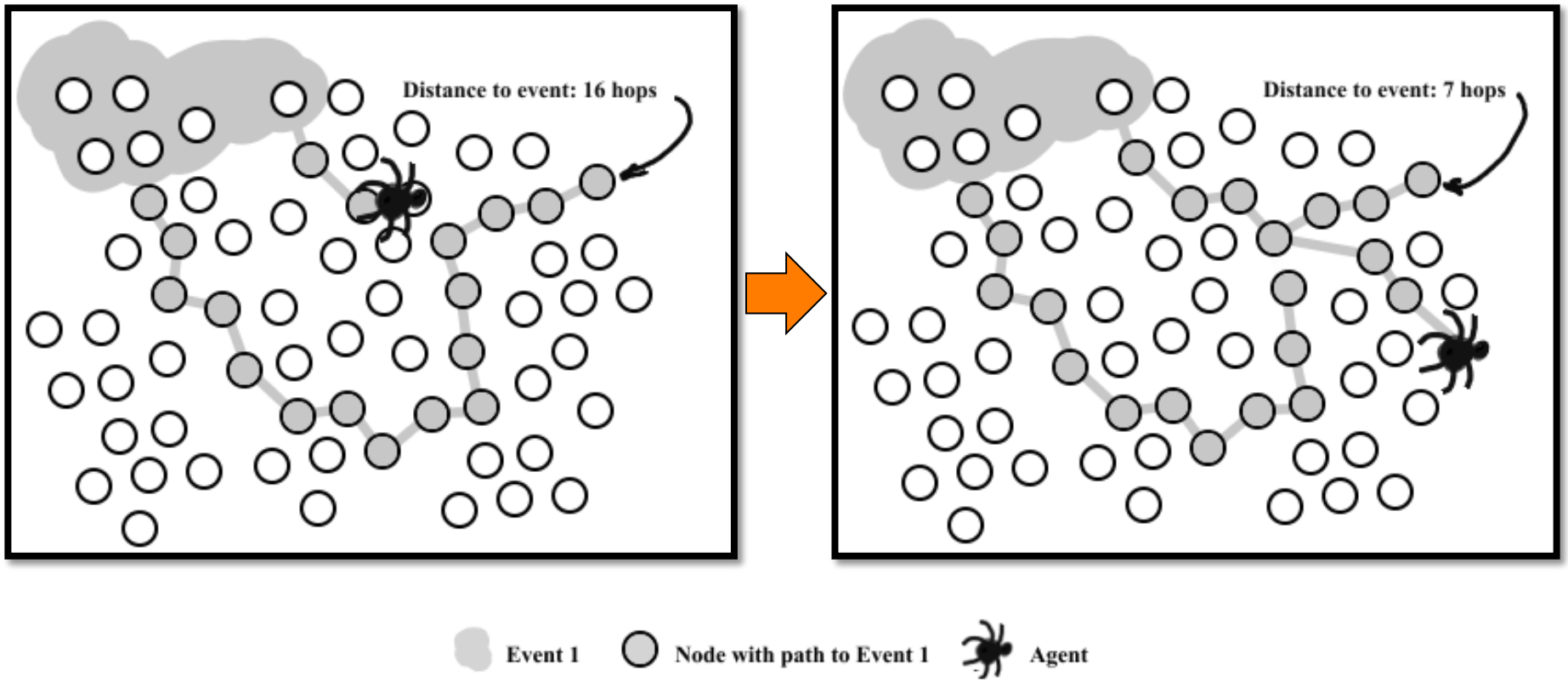
Rumor routing: Agents

- Each agent carries a list of all the events it encounters with the number of hops to that event.
- Each node among the path maintains a table of events, the number of hops to that event and the next node in the path toward the event.
- Agent syncs info with node's table to learn and improve path info.



Rumor routing: Agents

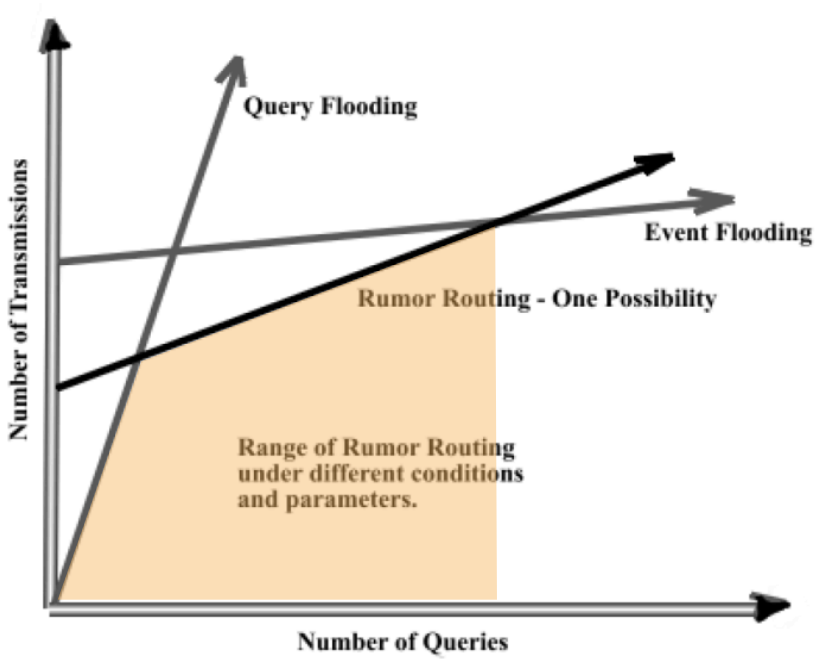
Optimize the paths in the network



Rumor routing: Evaluation

- Comparison of Flooding with Rumor Routing
  - Metric: Nr of transmissions as metric for energy consumption
- Trade-off between
  - Discover shortest path by Flooding

- When is Rumor Routing better?
  - Flood queries
    - When many events and few queries
  - Flood events
    - Install gradients to event source
    - Cheap queries possible
  - Rumor Routing
    - For cases between these



Rumor routing: Evaluation

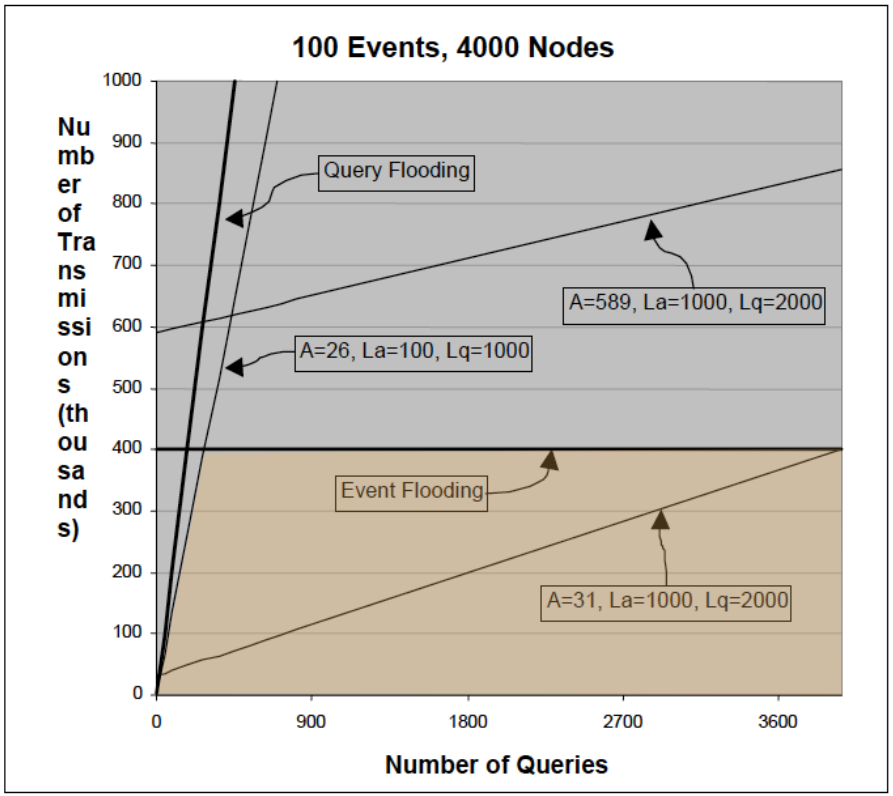


Figure 6: Some possible configurations of Rumor Routing. Although certain parameters result in costs greater than the flooding alternatives, others allow lower total cost for up to 36 queries per event with 98.1% delivery rate.

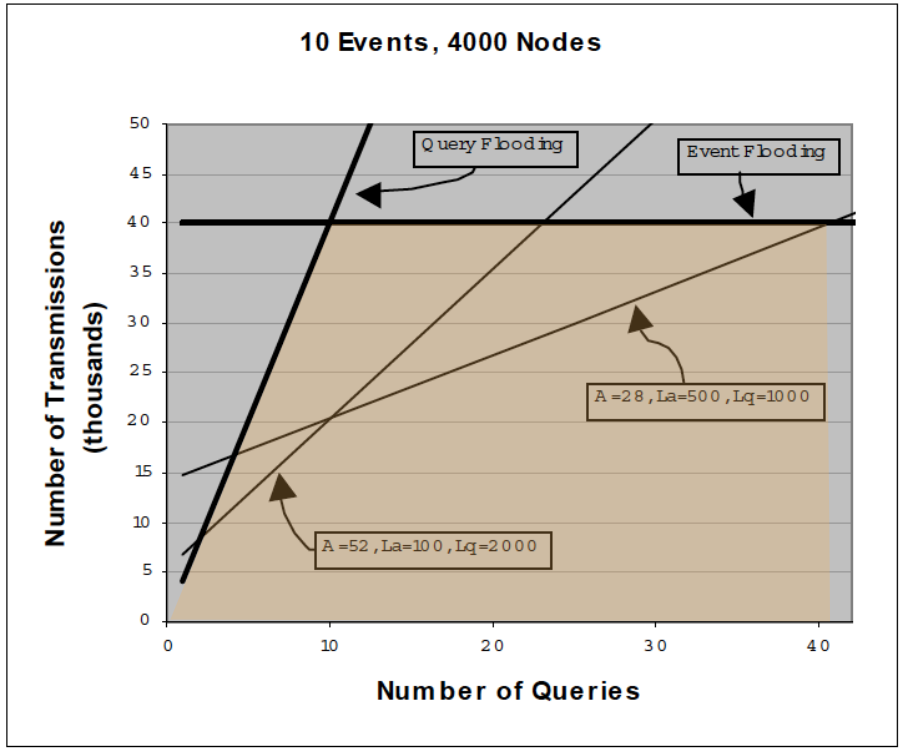


Figure 7: If the number of queries per event is less than ten, a smaller setup cost is better than a smaller per-query delivery cost. If, however, we want to deliver more queries (up to 40), a larger investment in path building yields better results. Delivery is guaranteed, as undelivered queries are flooded.

A Number of agents
La Agent TTL
Lq Query TTL

Rumor routing: Evaluation

- Effect of event distribution
- Same set of parameter values
- 100 randomly generated event, node, and query
- CDF of the delivery rate

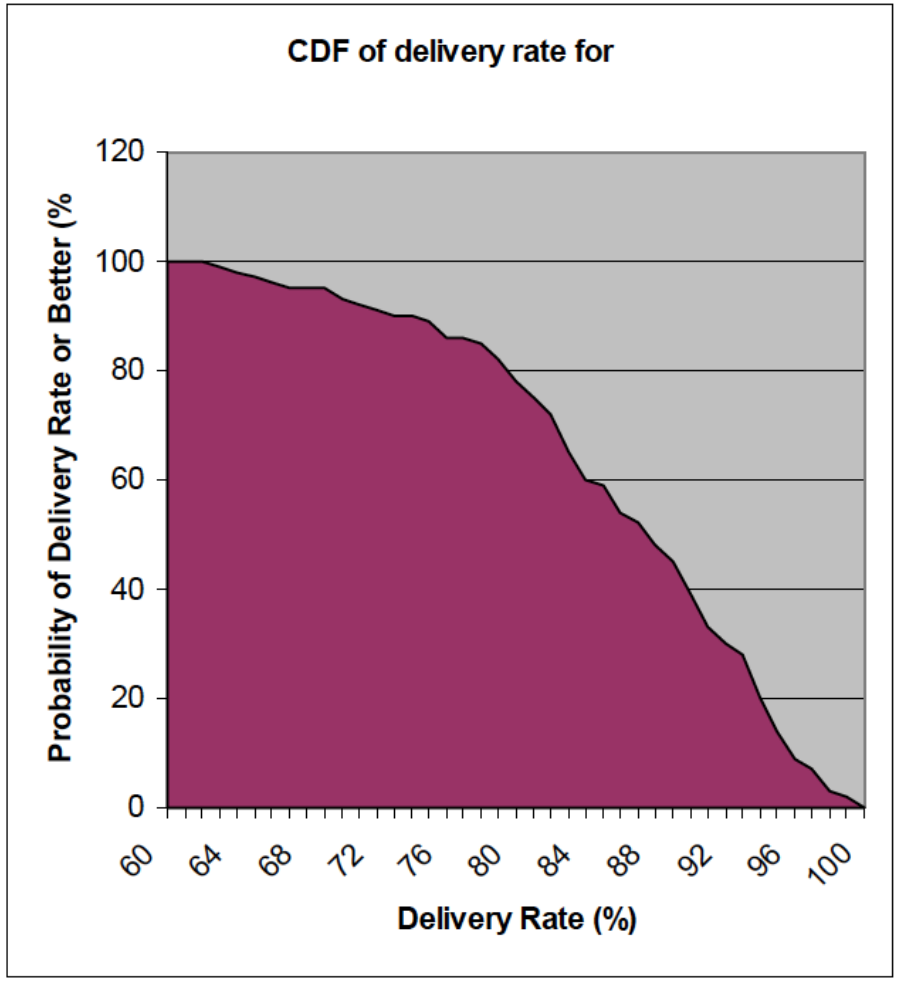


Figure 8: The probability that Rumor Routing successfully delivers at least that number of queries for any event/node/query distribution.

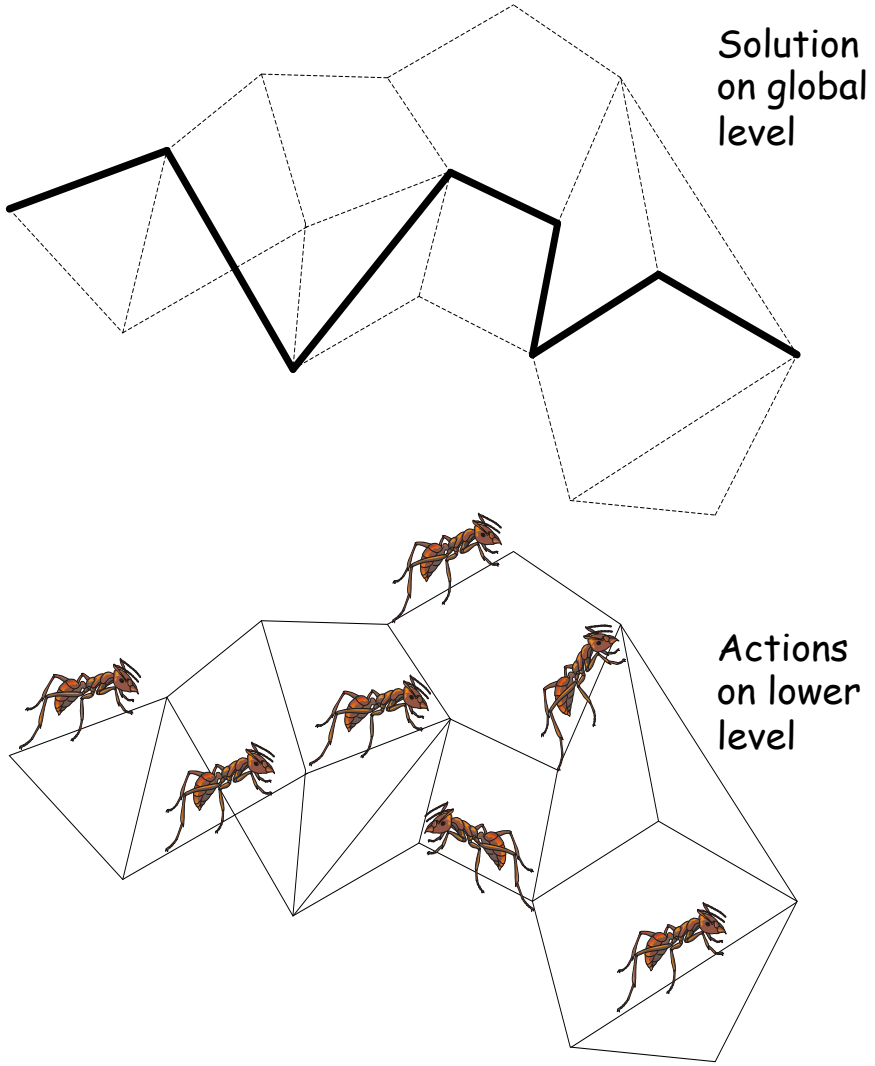
Probabilistic routing

Ant Routing Algorithm

Swarm intelligence

- Swarm intelligence
  - Algorithms based on collective behavior of social animals and insects
  - An individual does not have
    - high sophisticated instruments
    - knowledge about the "big problem"
  - The collective is self organized, i.e. no central control instance
    - Only local information is used ➡ Stigmergy
    - Positive and negative feedback
    - Random fluctuation
  - Applied to many mathematical problems as heuristic

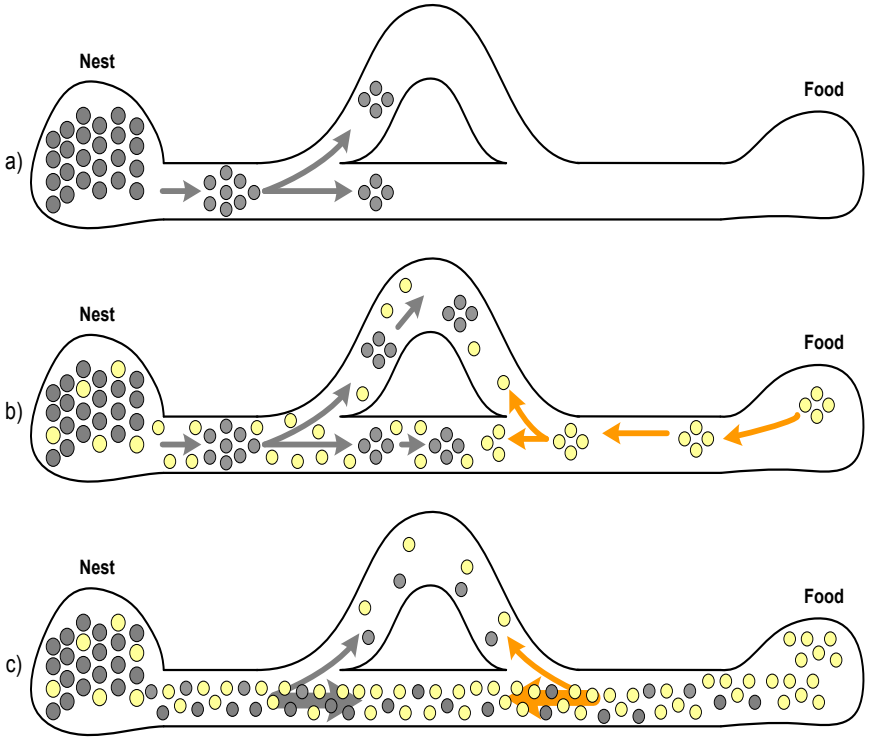
- Ant algorithms
  - Subset of swarm intelligence which model the behavior of ant colonies, e.g. nest building, brood feeding
  - Here: Foraging of ant colonies is used as heuristic



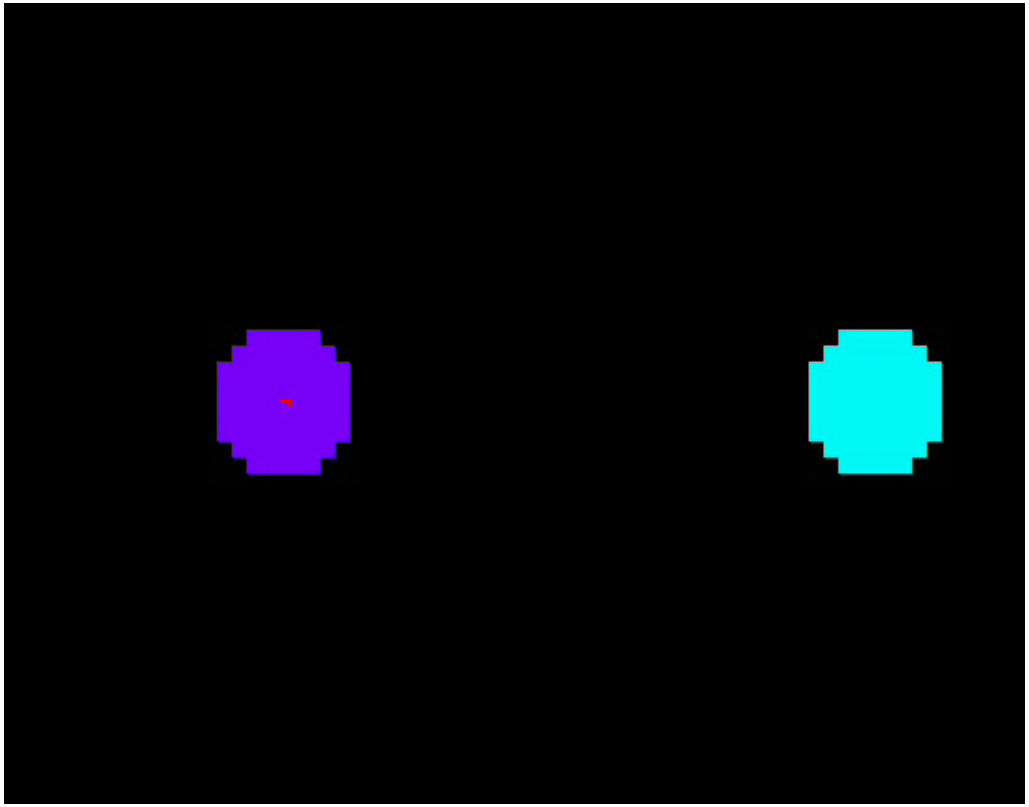
Foraging of ant colonies

- Ants deposit pheromone while walking
- The pheromone concentration on a link changes over time
  - increases by link usage
  - decreases by time (i.e. by evaporation)
- The pheromone concentration on a link is an indication of its usage
- Ants prefer links with higher pheromone concentration
- After a while, nearly all ants take the shortest path.

The principle of finding the shortest path in a graph by using the food searching behavior of ants.



Foraging of ant colonies



The ant algorithm

- Let G=(V,E) be a connected graph with n=|V| nodes
- An ant travels from source node s to destination node d
- Each edge e(i,j) of the graph has variables $\varphi^i_{d,j}$ denoting pheromone concentration for selecting this edge when traveling to d
- An ant residing on node i selects node j as next node with probability

$$p^i_{d,j} = \begin{cases} \dfrac{\varphi^i_{d,j}}{\sum_{k \in N_i} \varphi^i_{d,k}} & \text{if } j \in N_i \\ 0 & \text{otherwise} \end{cases} \qquad N_i := \text{ set of neighbors of } i$$

- The pheromone concentration is modified according to

$$\varphi^i_{d,j} \leftarrow \begin{cases} (1-q)\cdot\varphi^i_{d,j} + \Delta\varphi & \text{if } j \text{ is selected as next node} \\ (1-q)\cdot\varphi^i_{d,j} & \text{otherwise} \end{cases}$$

The Ant Routing Algorithm (ARA)

- Design goals
  - Small routing overhead
  - Simple implementation

- ARA is an on-demand routing algorithm consisting of three phases
  - Route discovery
  - Route maintenance
  - Route failure handling

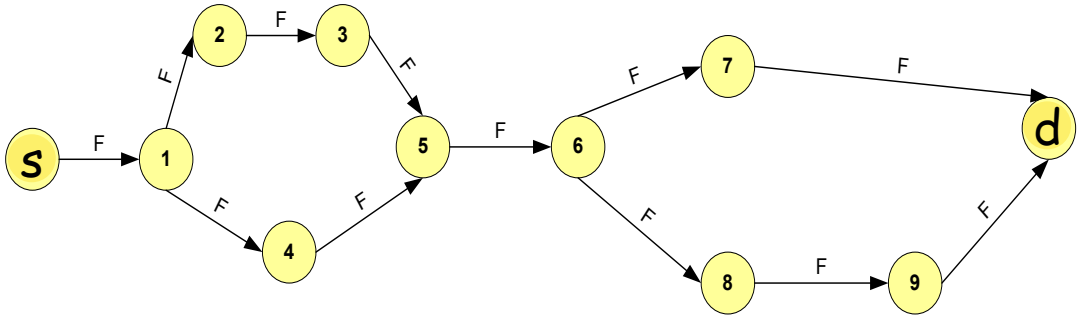
- Small routing tables consisting of

```
(Destination, Next-Hop 1, Pheromone Value 1)
(Destination, Next-Hop 2, Pheromone Value 2)
```

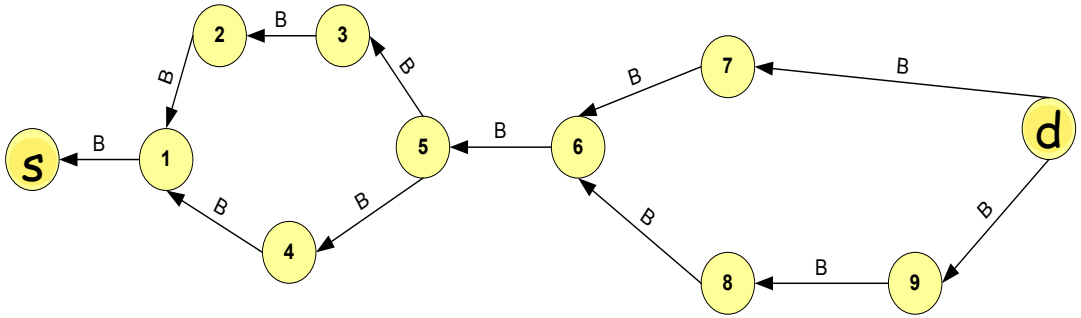
The Ant Routing Algorithm (ARA)

Phase 1: Route Discovery

- Forward Ant Messages (FANT) and Backward Ant Messages (BANT) are used to discover routes in the network
  - A FANT establishes the pheromone track back to the source node (s)
  - A BANT establishes the pheromone track towards the destination node (d)



a)



b)

The Ant Routing Algorithm (ARA)

Phase 2: Route Maintenance

- Data packets modify the pheromone concentration similarly to FANT and BANT messages, i.e., a route which is regularly used tends to stay "strong".

- When a node i relays a data packet towards the destination d via a neighbor node j, it changes its routing table as follows:
  $(d, j, \varphi^i_{d,j}) \Rightarrow (d, j, \varphi^i_{d,j} + \Delta\varphi)$.
  The path to the destination is reinforced

- The next hop j performs $(s, i, \varphi^j_{s,i}) \Rightarrow (s, i, \varphi^j_{s,i} + \Delta\varphi)$.
  Thus the path to the source node is also reinforced.

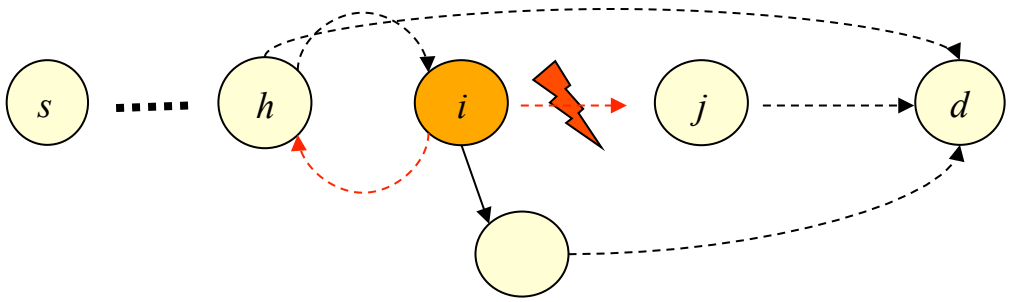
- Pheromone values are continuously decreased, i.e., a route which is not well used will "disappear".
  ⇒ This prevents negative impact of cached routing table entries.

The Ant Routing Algorithm (ARA)

Phase 3: Route Failure Handling

- If a route does not exist any more (due to topology changes or due to insufficient level of pheromone)
- i deactivates the link to j and searches an alternative path to d
- If no alternative exists, the previous node h is informed
- The previous node h deactivates the link to i
  - If h has an alternative path to d, it tries to transport over this path
- i and h may try only one alternative path

➡ Local failure handling



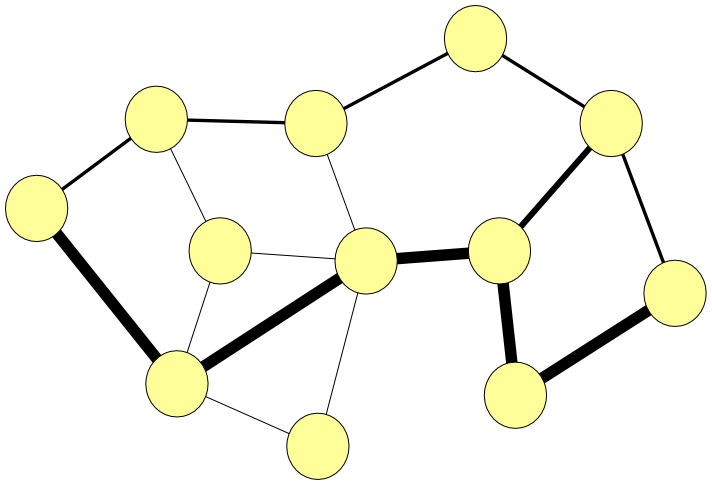
The Ant Routing Algorithm (ARA)

- Selection of next hop (probabilistic vs. dynamic routing)
  - Dynamic routing

$$p_{d,j}^{i} = \begin{cases} 1 & \text{if } \varphi_{d,j}^{i} = \max_{k \in N_i} \{\varphi_{d,k}^{i}\} \\ 0 & \text{otherwise} \end{cases}$$

  - Probabilistic routing

$$p_{d,j}^{i} = \begin{cases} \dfrac{\varphi_{d,j}^{i}}{\sum_{k \in N_i} \varphi_{d,k}^{i}} & \text{if } j \in N_i \\ 0 & \text{otherwise} \end{cases}$$



- Prioritized packets
  - FANT, BANT, packets with loop and error flags are prioritized.

The Ant Routing Algorithm (ARA)

- Evaporation of pheromone trails
  - Continual decrease of pheromone values

$$\varphi^{i}_{d,j}(t) = \varphi^{i}_{d,j}(t_{ta}) \cdot e^{t_{la}-t}$$

when t_{la}=last access to routing table for node i

- Flooding of the BANTs
  - BANTS are also flooded, instead of sending back via unicast
    - ➡ better exploitation of multi-path routing property

The Ant Routing Algorithm (ARA)

- Extension of ARA
  - Integration of other parameters γ into the routing decision
    - Energy level
    - Bandwidth
  - Open questions are
    - How to collect and distribute the information?
    - Integration into route discovery and route maintenance?

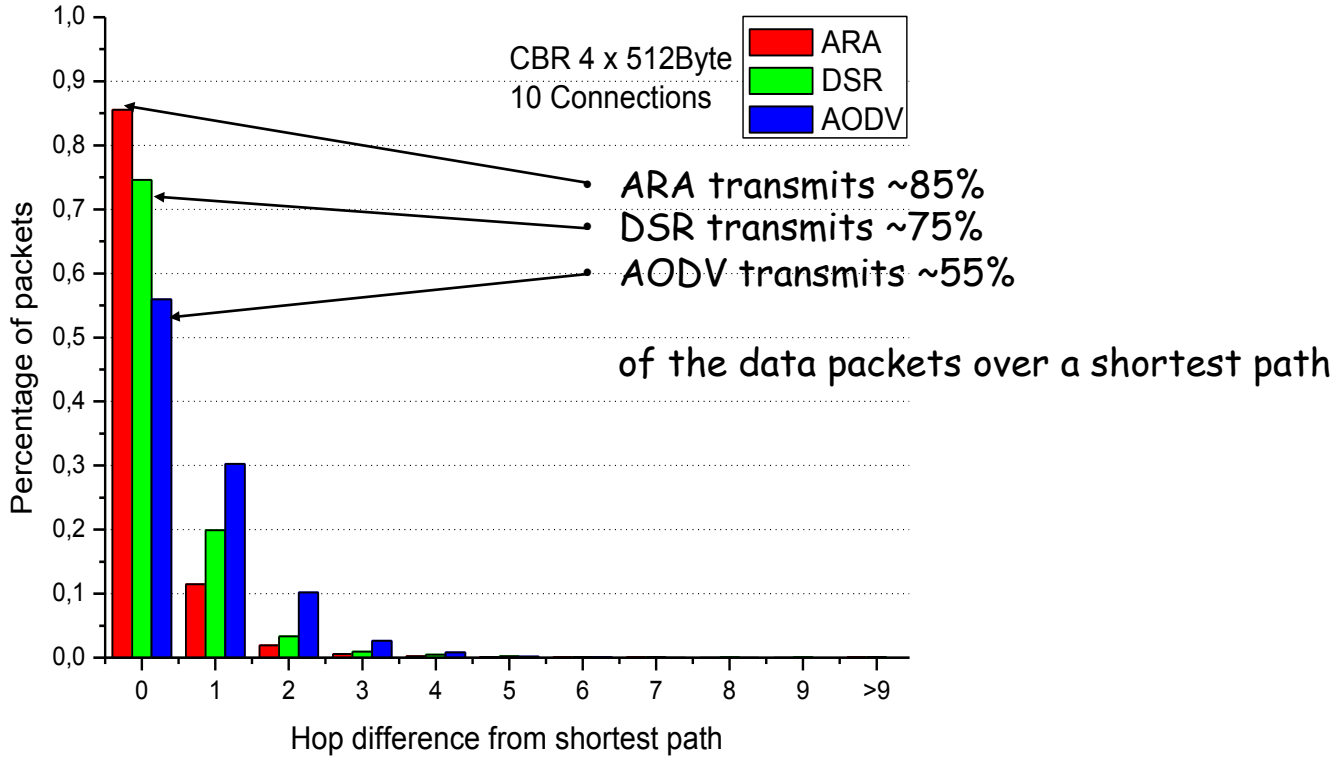
$$p_{i,j} = \begin{cases} \dfrac{\left[\varphi_{i,j}\right]^{\alpha}\left[\gamma_{i,j}\right]^{\beta}}{\sum_{k\in N_i}\left[\varphi_{i,k}\right]^{\alpha}\left[\gamma_{i,k}\right]^{\beta}} & \text{if } j\in N_i \\ 0 & \text{otherwise} \end{cases}$$

The Ant Routing Algorithm (ARA)

- **Distributed operation:** Each node controls the pheromone concentration independently.
- **Loop-free:** The nodes register the unique sequence number of route finding packets, FANT and BANT, so they do not generate loops.
- **On-demand operation:** A route finding process is only initiated, if a sender demands for it.
- **Local:** The routing table and the statistic information block of a node are local and are not transmitted to any other node.
- **Multi-path routing:** Each node maintains several paths to a destination. The choice of a particular route depends on the current state.
- **Sleep mode:** In the sleep mode a node snoops. Only packets which are destined to it are processed: This may be used to save power.

Simulation environment and results

Path optimality, ns-2, 10 CBR Connections, 4 Packets/s with 512 Byte



Content based routing

Basic principles

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

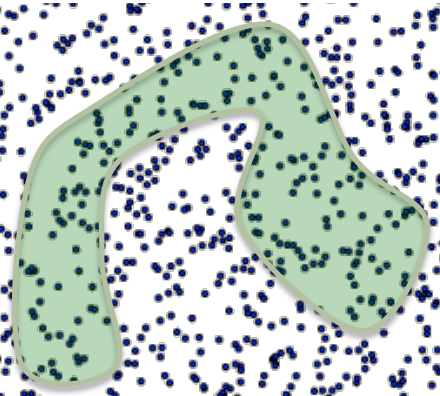
What is data centric?

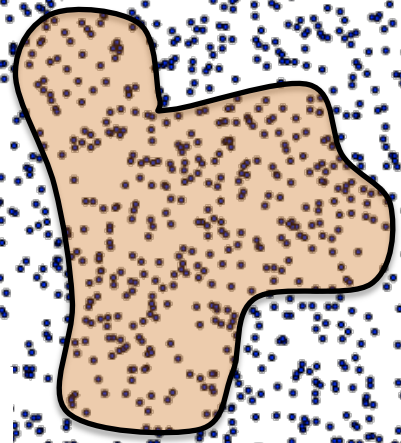
Sensor node does not need an identity

- What is the temperature at node #27?
- Data are named by attributes

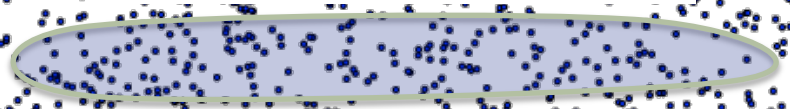
Where are the nodes whose temperature recently exceeded 30° ?

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 routing?

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

- Example:

  "the areas where the temperature is over 24°C"

  is a more common query than

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

Content based routing

Directed Diffusion

Elements of Directed Diffusion

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

Naming scheme

- Data generated by sensor nodes is named by attribute-value pairs

- In order to create a query, an interest is defined using a list of attribute-value pairs such as:
  - name of objects
  - interval
  - duration
  - geographical area
  - etc.

- An arbitrary sensor node (usually the sink) uses attribute-value pairs (interests) for the data and queries the sensors in an on-demand basis

Naming scheme

Interest (Task) Description

- Example: (Animal Tracking Task)
  - Type = four legged animal (detect animal location)
  - Interval = 30 s (send back events every 30 s)
  - Duration = 1h (for the next hour)
  - Rec = [-100,100,200,400] (sensors within the rectangle)

Naming scheme

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

- Example: Reply

- Sensor detecting the animal sends the following:
  - Type = four legged animal (type of animal seen)
  - Instance = elephant (instance of this type)
  - Location = (125,220) (node location)
  - Intensity = 0.6 (signal amplitude measure)
  - Confidence = 0.85 (confidence in the match)
  - Timestamp = 01:20:40 (event generation time)

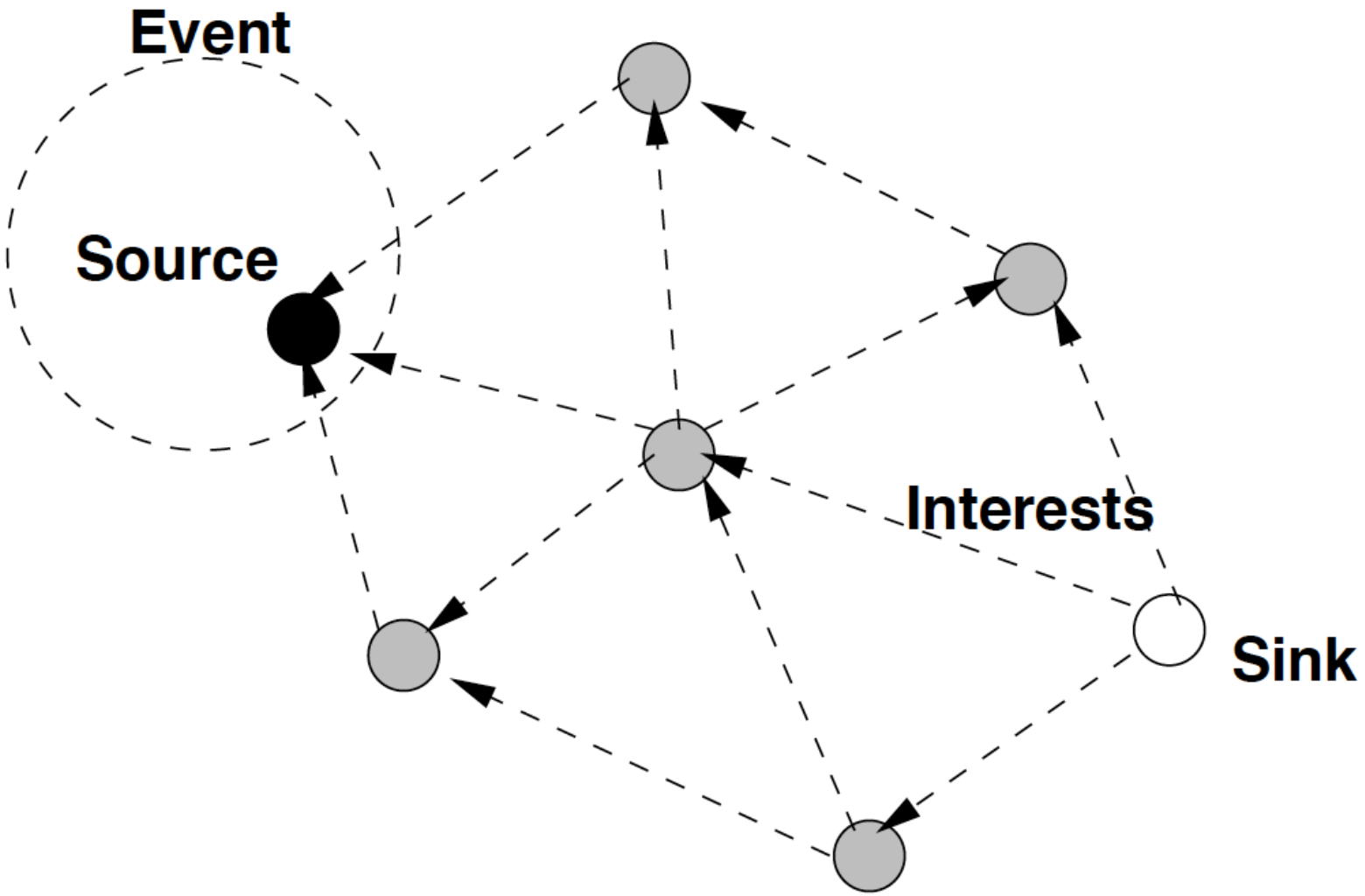
Interests

- The sink periodically broadcasts an interest to sensor nodes to query information from a particular area in the field
- As the interest propagates, data may be locally transformed (e.g., aggregated) at each node, or be cached
- Every node maintains an interest cache
  - Each item corresponds to a distinct interest
  - Interest aggregation: identical type, completely overlap rectangle attribute

Interests

- Each entry in the cache has several fields
  - Timestamp: last received matching interest
  - Several gradients: data rate, duration, direction
- Other nodes may express interests based on these attributes
- When a node receives an interest, it:
  - Checks cache to see if an entry is present
  - If no entry, creates an entry with a single gradient to neighbor who sent this interest
  - Gradient specifies the direction and data rate

Interest propagation

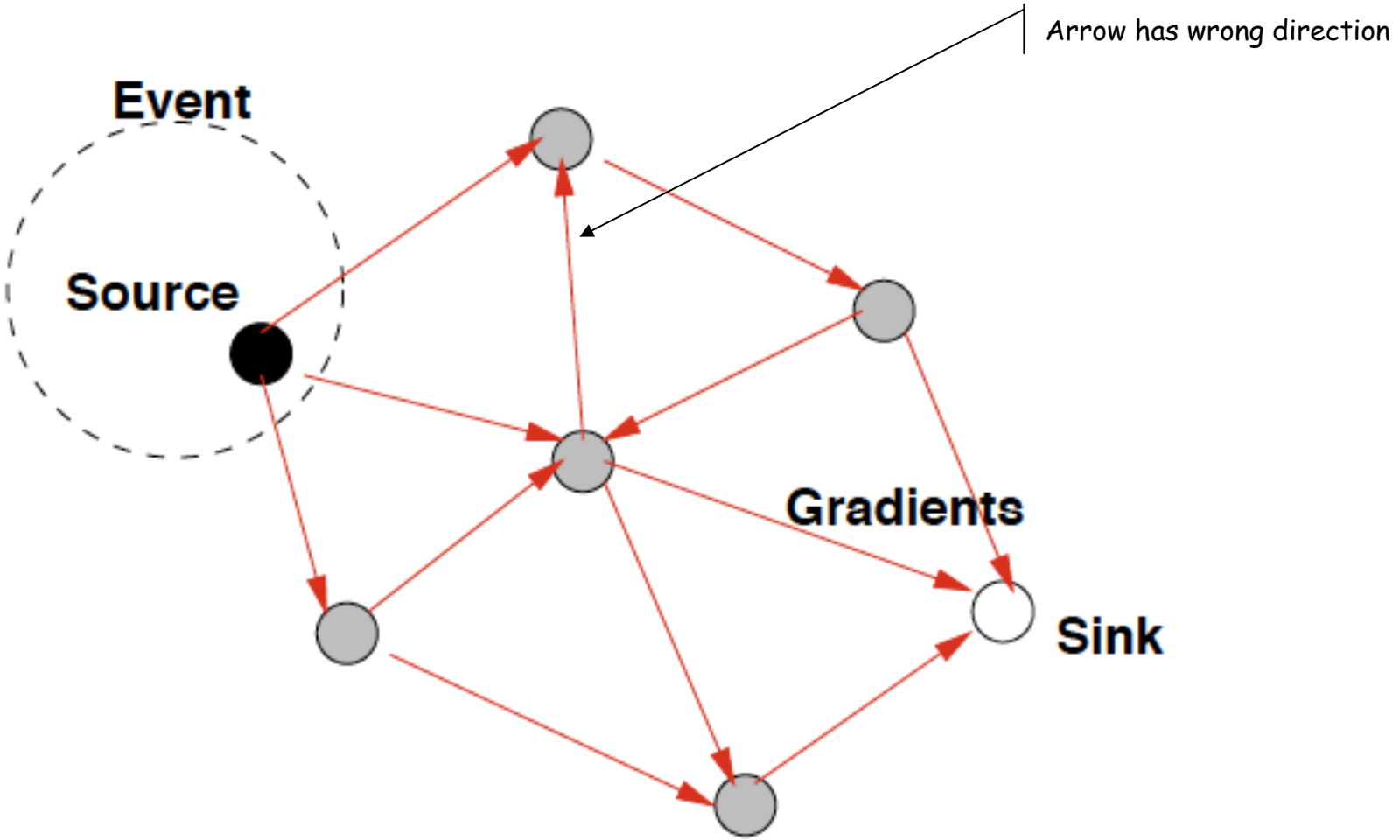


Gradient setup

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

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

Gradient setup



Local rules for propagating interests

- Just flood interest
- More sophisticated techniques possible:
  - Directional interest propagation based on cached aggregate information
  - "I recently heard about suspicious activity from neighbor A, so let me try sending this interest for recent intrusions to that neighbor"
- Highest gradient towards neighbor who first sends interest
- Others possible, e.g.,
  - towards neighbor with highest remaining energy (energy gradients)
  - probabilistic gradients

Gradient reinforcement

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

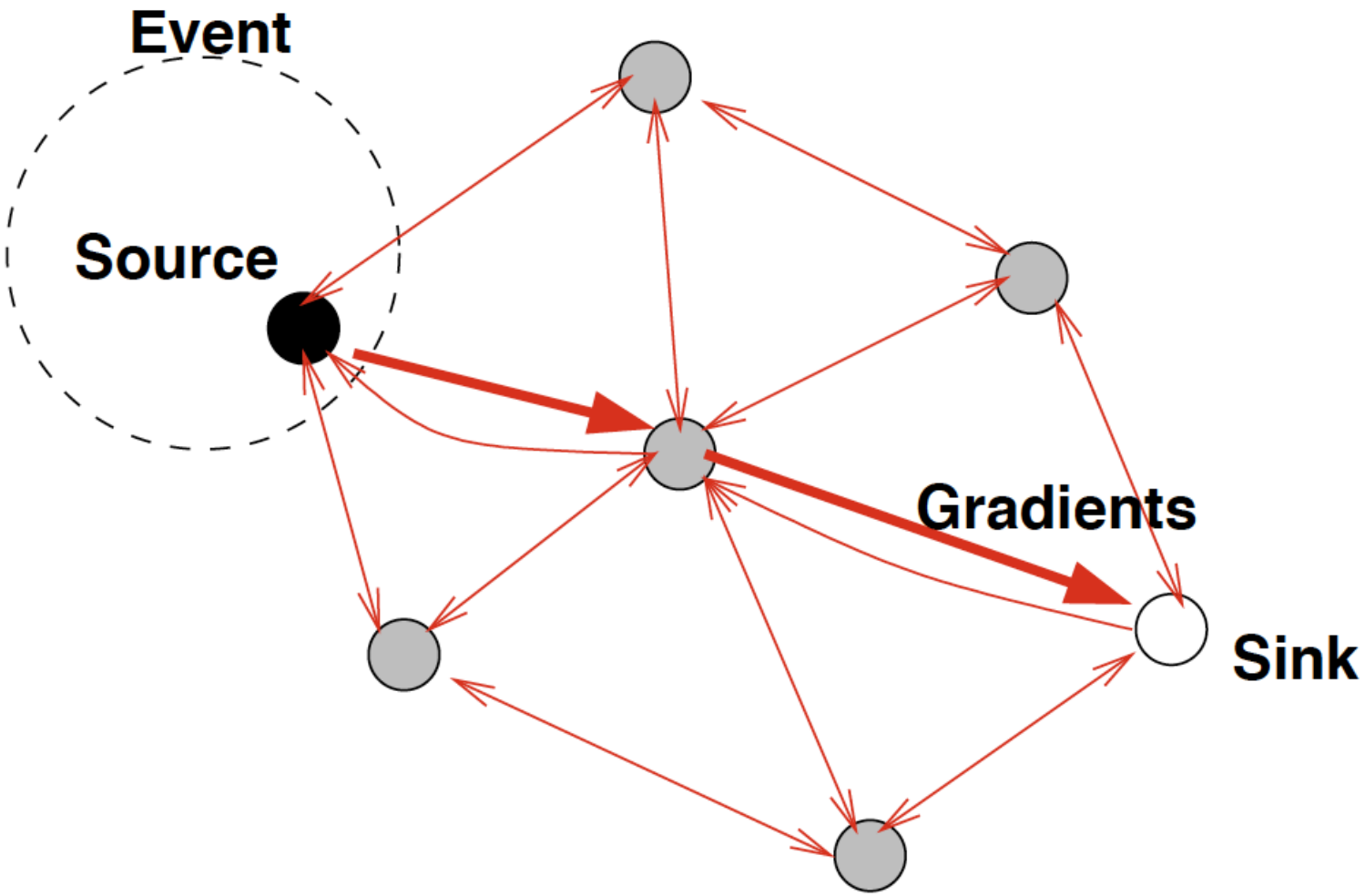
Gradient reinforcement

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

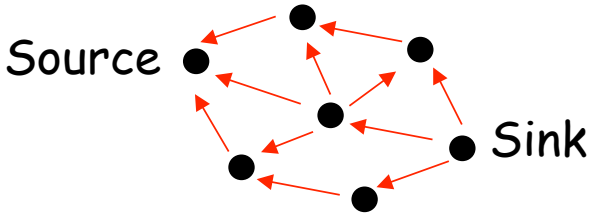
Data propagation

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

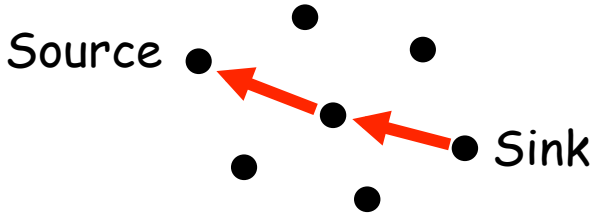
Reinforcement and data propagation



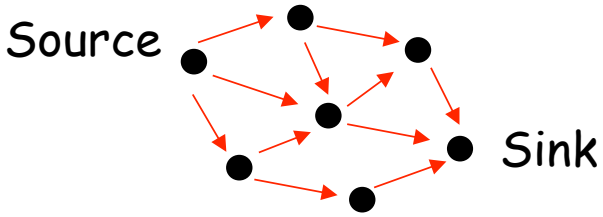
Directed Diffusion: Overview



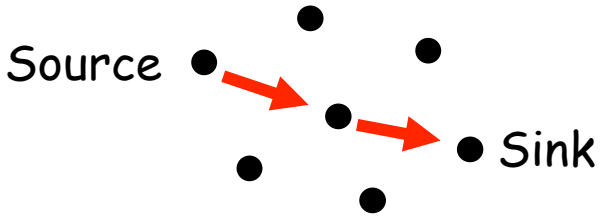
Interest Propagation



Reinforcement



Gradient Setup



Data Delivery

Negative reinforcement

- Time out
- Explicitly degrade the path by re-sending 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

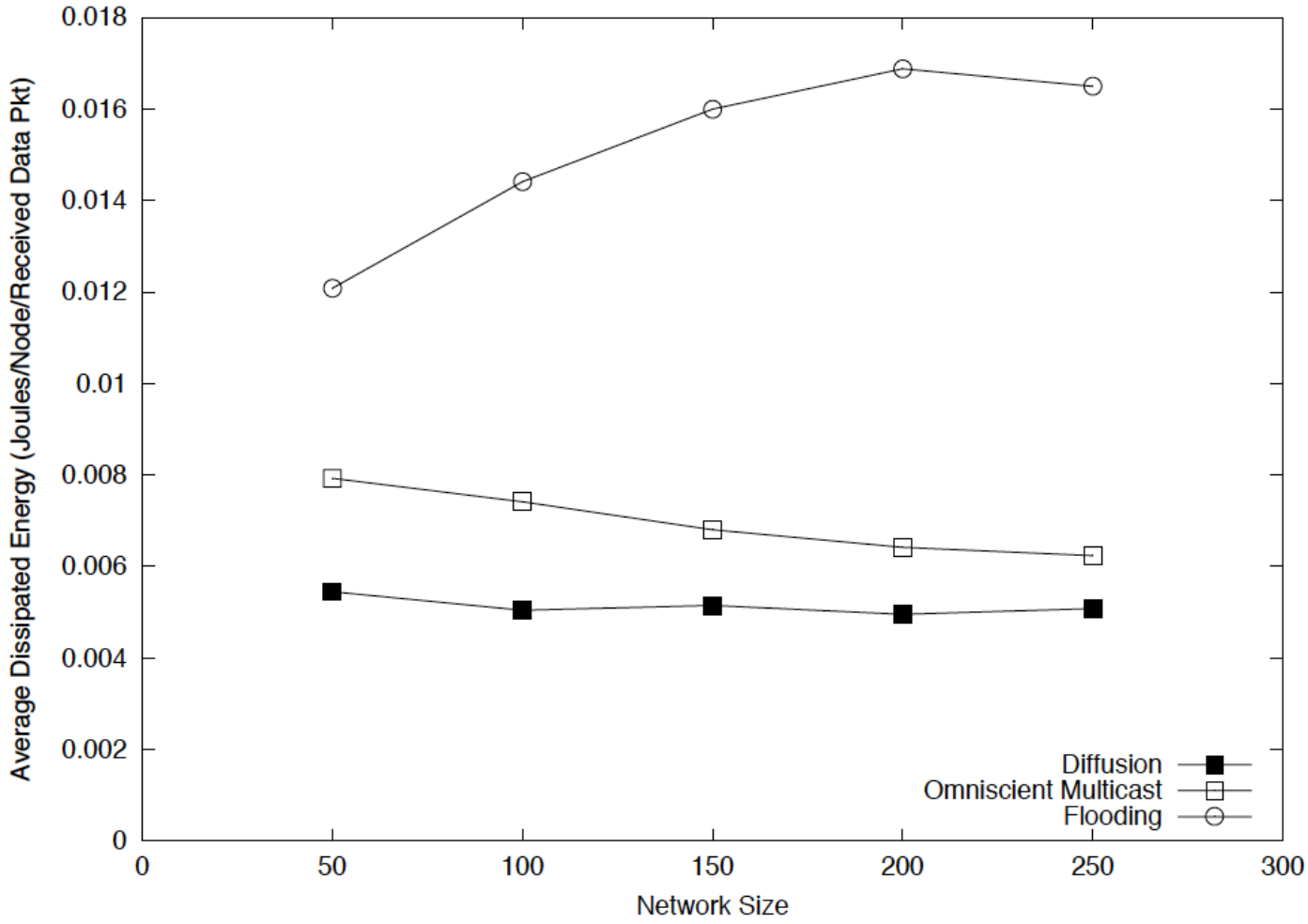
Simulation

- Simulator: ns-2
- Network Size: 50-250 Nodes
- Total area for 50 nodes 160m x 160m
- Transmission Range: 40m
- Constant Density: 1.95x10-3 nodes/m2 (9.8 nodes in radius)
- MAC: Modified Contention-based MAC
- Energy Model: Mimic a realistic sensor radio
  - 660 mW in transmission, 395 mW in reception, and 35 mw in idle

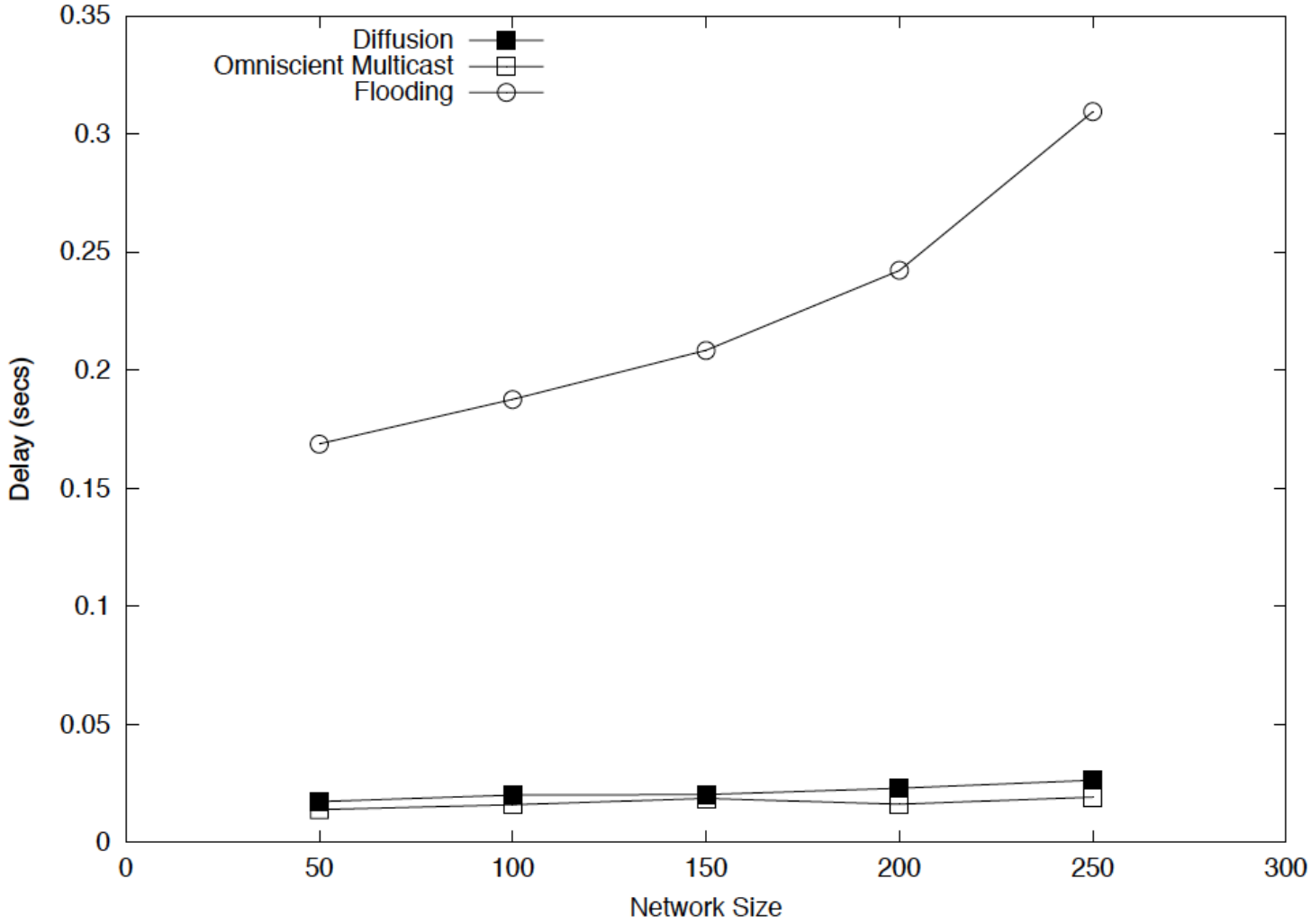
Performance metrics

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

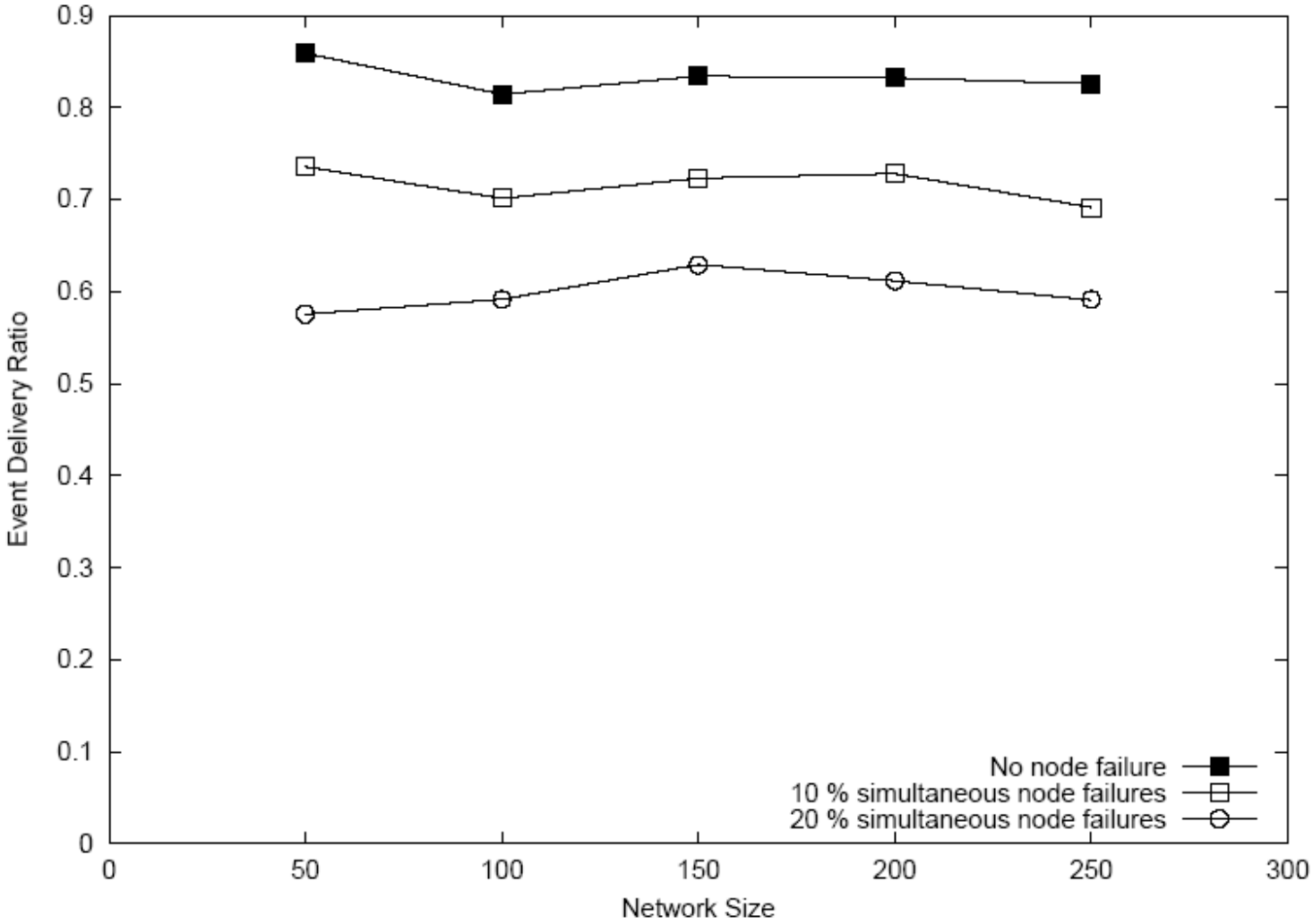
Average dissipated energy



Average delay



Event delivery ratio



Directed Diffusion: Conclusions

Advantages

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

Disadvantages

- Not generally applicable since it is query driven
- Not suitable for applications needing continuous data delivery, e.g., monitoring
- Naming scheme is application dependent -> find an appropriate one
- Matching process for data and queries cause some overhead at sensors

Comparison of data centric routing algorithms

	Data Efficiency	Energy Efficiency	State Complexity
Flooding	Fastest	Low b/c Implosion	None
Gossiping	Slowest	Low Random walk	Small, random selection
SPIN	Very Fast	Higher than above, SPIN-EC close to ideal	Data- neighbor pairs
Directed Diffusion	Quite Fast	Higher than global flooding + strong aggregation	Complex: Neighbor × Interest

Comparison of content vs. address based routing

	Address based	Content based
Approach	Path is determined by addresses	Destination is determined by content
Assumptions	Globally unique addresses	Pre definied format with semantics
Routing method	Proactive or reactive routing	(probabilistic) Flooding or registration based reverse routing
Pro	Low delay in connection establishment and data transmission	No addresses required
Contra	Globally unique addresses required	High effort for an single transmission

Geographical routing

Geographic routing

- Idea: Positions of nodes can be used to simplify routing

- Three categories
  - All nodes know their positions
    - Every node need GPS or similar
  - Some nodes know their positions
    - Some nodes need GPS
    - Other nodes can estimate their positions
  - No node knows its position
    - No GPS data available
    - Relative coordination system possible

- Here: We assume node positions are available!

Geographical routing

- Motivation
  - In many applications querying (nodes or data) of a region
  - Data needs to be enriched with location information
  - Position information of nodes can support routing decisions
    - May simplify routing at all

- Advantage
  - Position could be equivalent to addresses
    - Implicit order of addresses
    - Suitable for greedy methods

Distance based greedy routing

[Takagi]

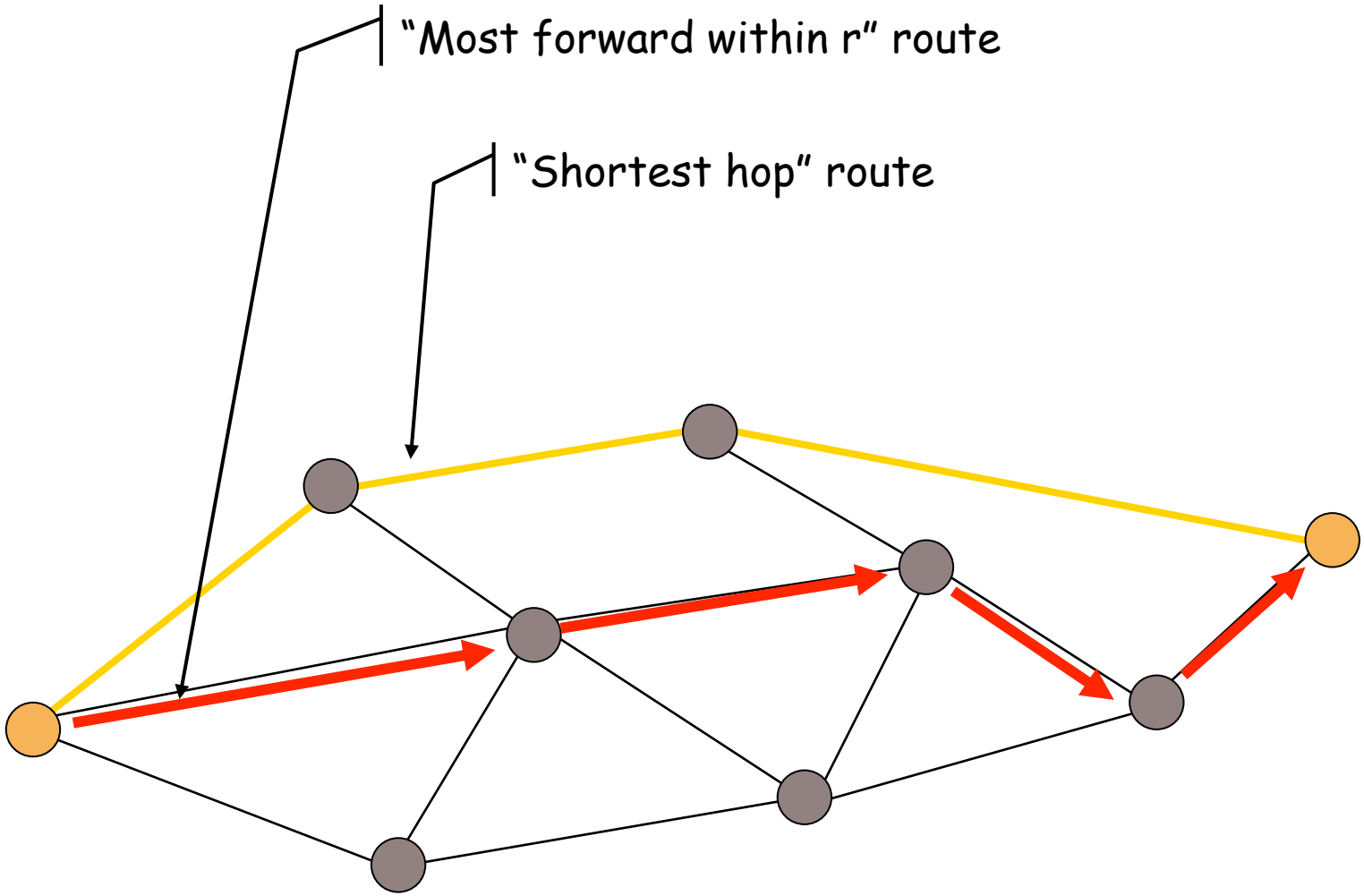
- Objective
  - Transmit packet to a known position
- Assumption
  - Every node knows its position -> GPS or similar

- Approach: "most forward within r"
  - r communication radius -> neighborhood
  - Forward packet to node that is closest to the destination

$$nexthop(v,dst) = \min_{u \in neighbor(v)} \{dist(u,dst)\}$$

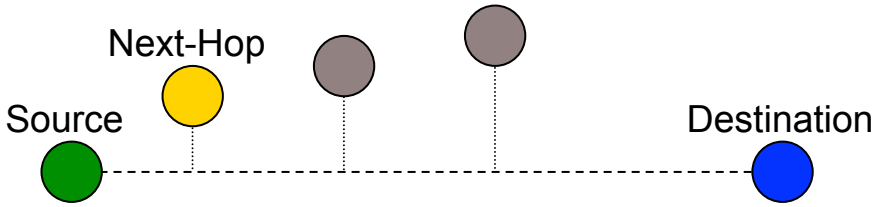
- Properties
  - \+ Loop free
  - – Does not consider topology
    - Does not select the shortest hop
  - – Nodes at the border of communication range are favored
    - Instable paths

Example: Most forward within r

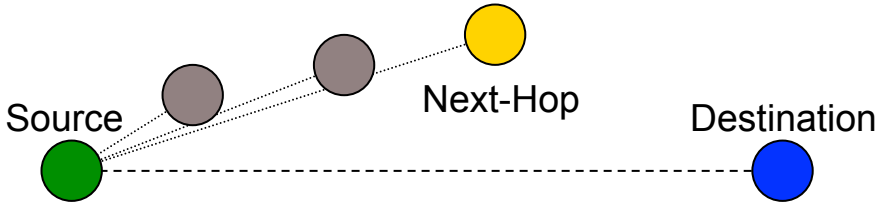


Direction based greedy approach

- Idea: Select node that is closest to the ideal direction to the destination
  - ideal direction = line between source and destination
- Two metrics
  - Minimum distance to connection line



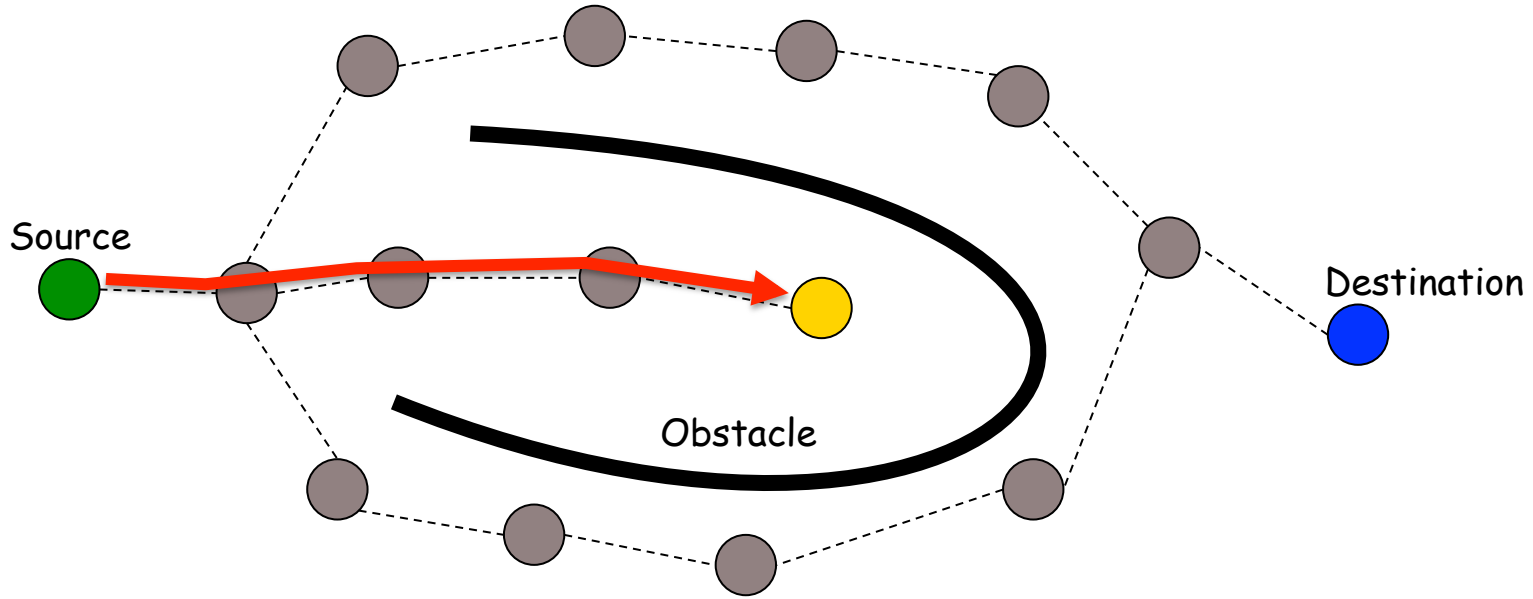
  - Minimum angle to connection line



- Problem
  - Not loop free

Problem of dead-ends

- Problem
  - Distance and direction based greedy approaches may end up in a dead-end



  - Solutions?

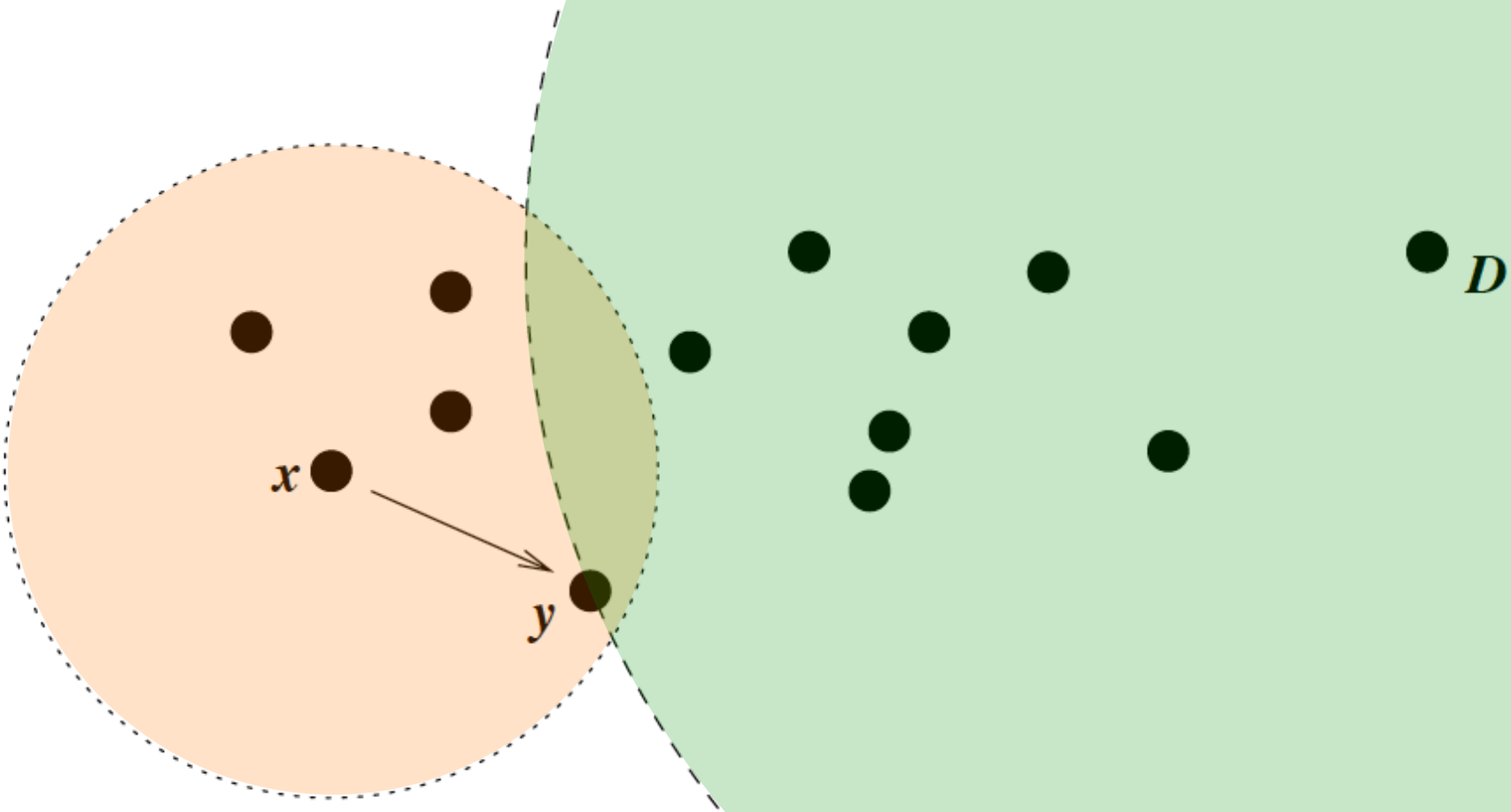
Geographic routing

GPSR

Greedy Perimeter Stateless Routing (GPSR)

- Approach: Two forwarding methods
  - Greedy forwarding
    - Next hop is the neighbor geographically closest to the packet's destinations
    - Only local information is used
  - Perimeter forwarding
    - Traversing a graph (or labyrinth)
- GPSR switches between both forwarding methods

GPSR: Greedy forwarding



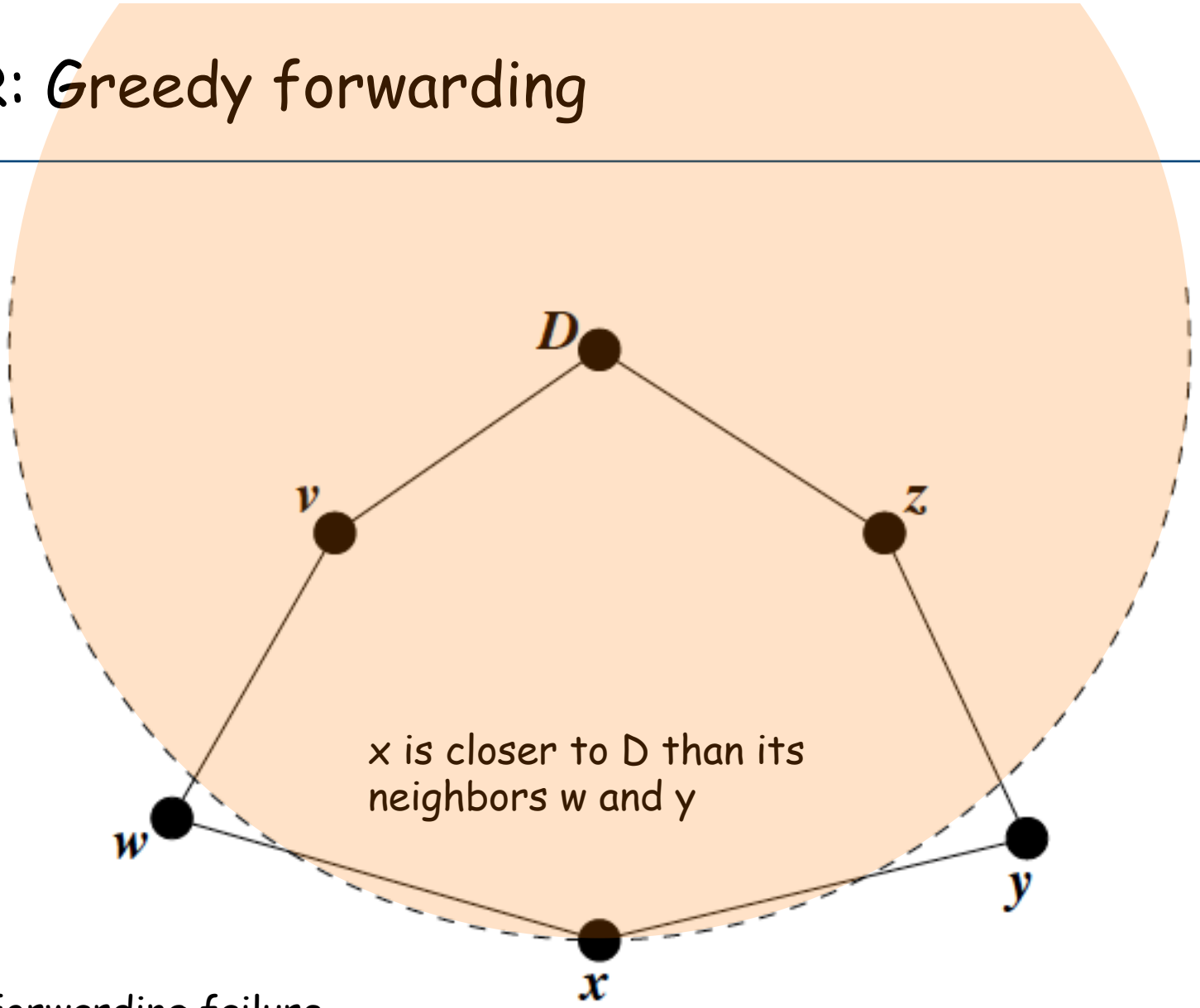
Greedy forwarding example. y is x's closest neighbor to D.

GPSR: Greedy forwarding

- Advantage
  - Only local knowledge of the forwarding node's immediate neighbors is needed.
  - The state required is negligible, and dependent on the density of nodes in the wireless network, not the total number of destinations in the network.

- Disadvantage
  - No node to forward geographically always!

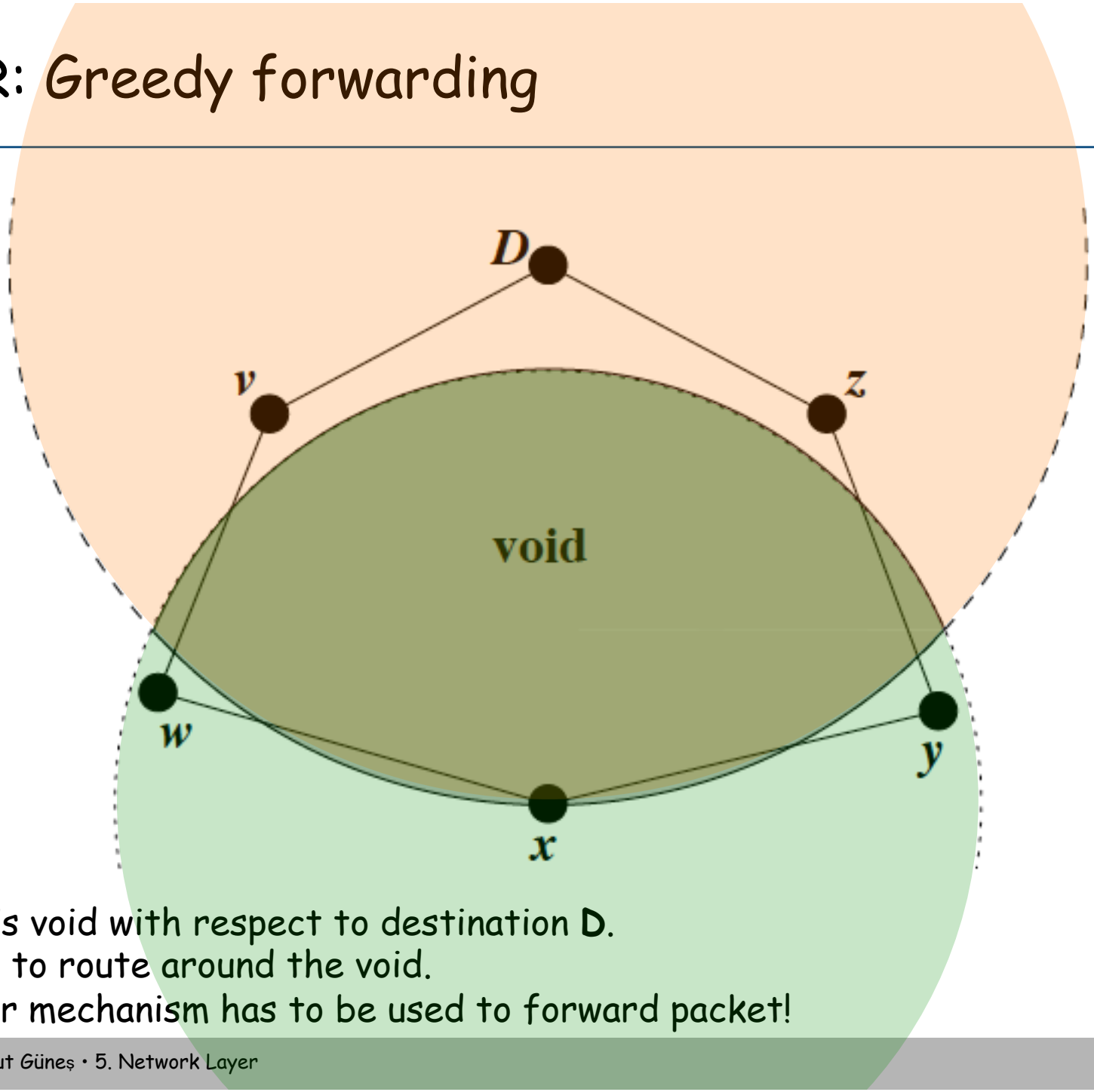
GPSR: Greedy forwarding



Greedy forwarding failure

- x is a local maximum in its geographic proximity to D
- w and y are farther from D

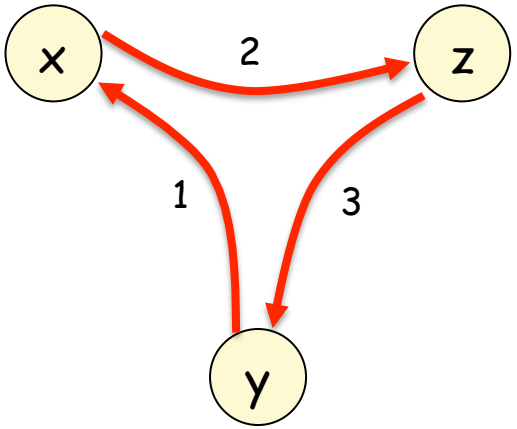
GPSR: Greedy forwarding

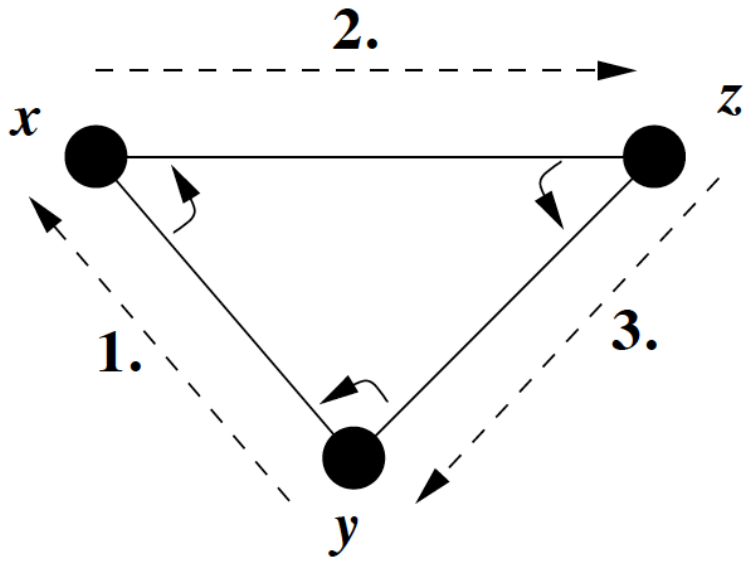


Node **x**'s void with respect to destination **D**.
x seeks to route around the void.
-> Other mechanism has to be used to forward packet!

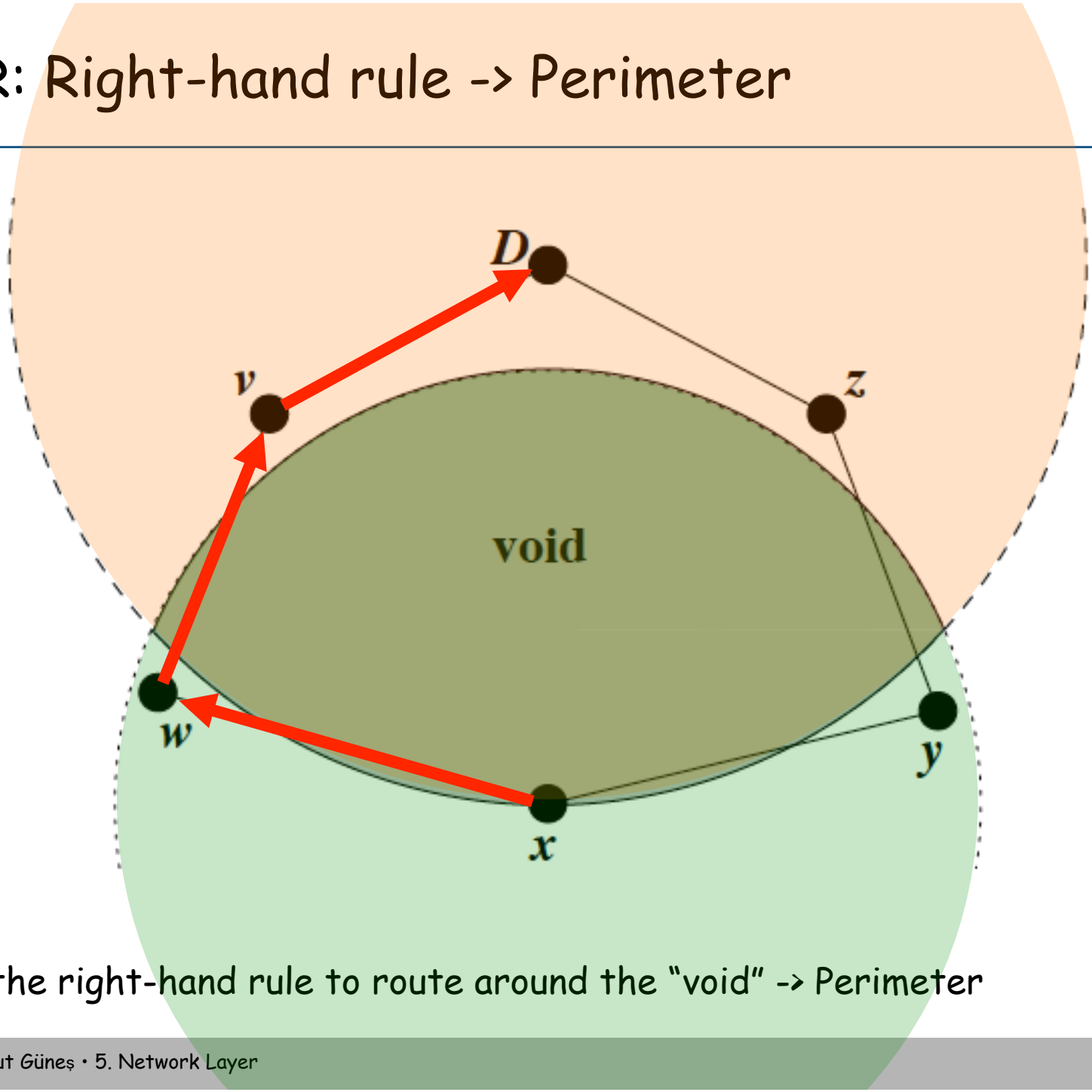
GPSR: Right-hand rule

- The right-hand rule (interior of the triangle).
  - x receives a packet from y, and forwards it to its first neighbor counterclockwise about itself, z.
- The right-hand rule traverses the interior of a closed polygonal region (a face) in clockwise edge order
- In example: y -> x -> z -> y
- The rule traverses an exterior region, in this case, the region outside the same triangle, in counterclockwise edge order.



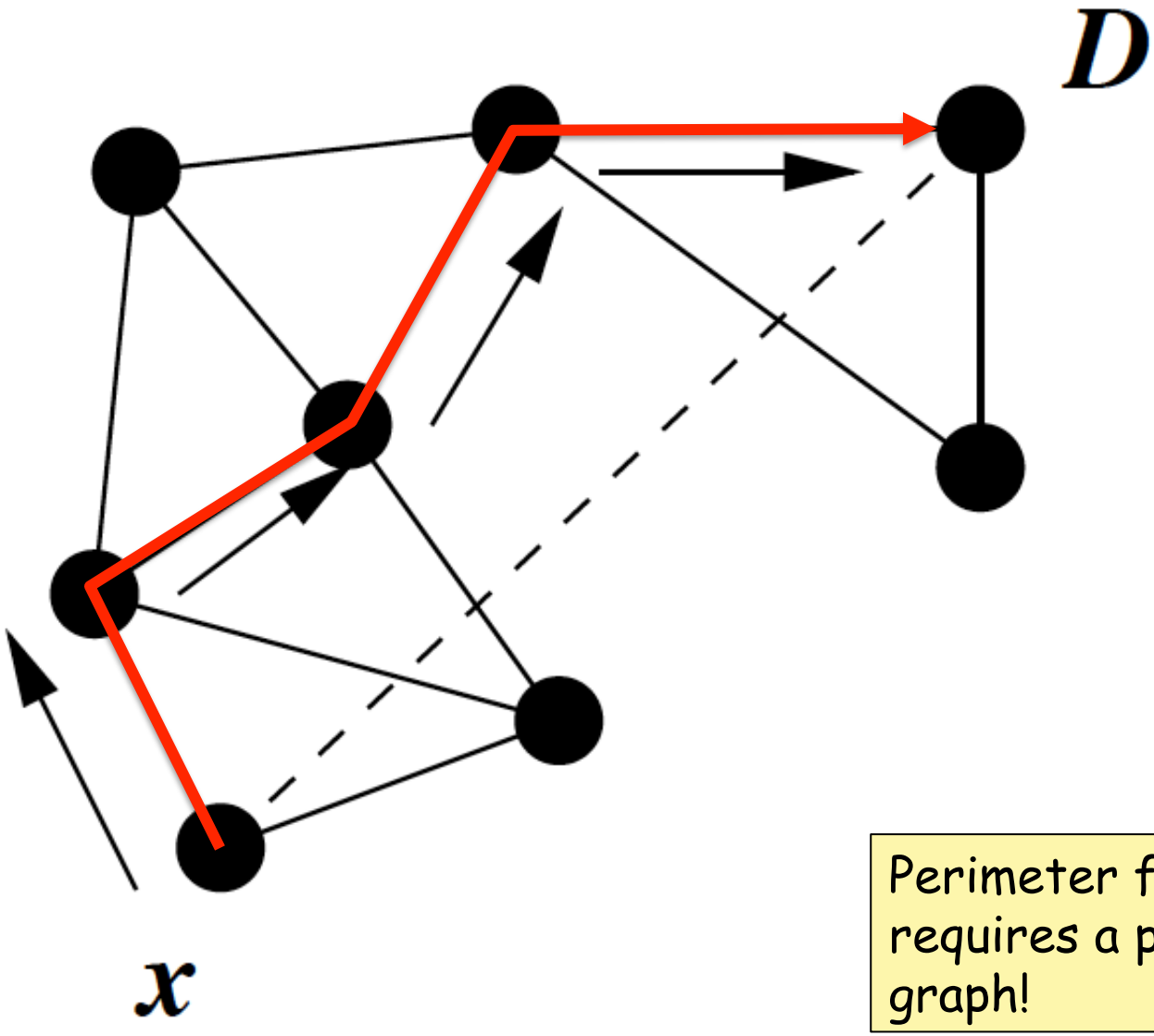


GPSR: Right-hand rule -> Perimeter



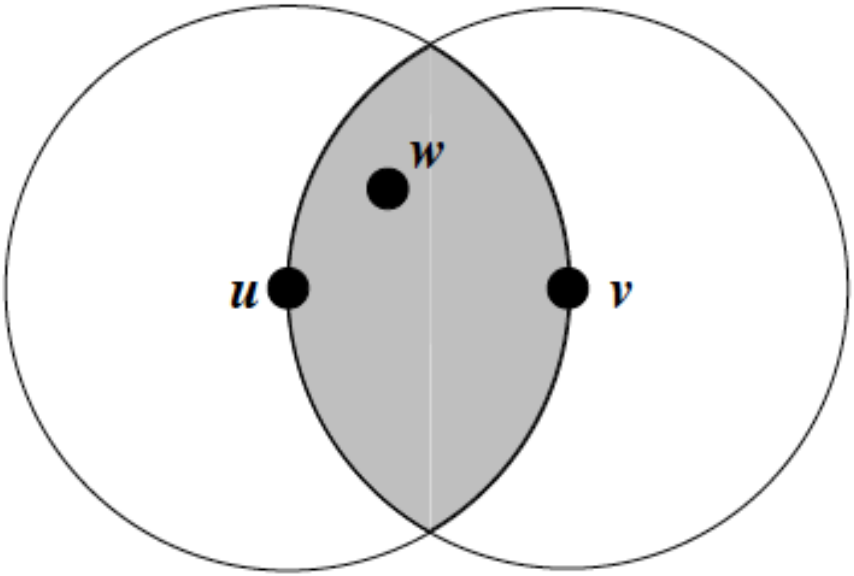
Use of the right-hand rule to route around the "void" -> Perimeter

GPSR: Perimeter forwarding



Perimeter forwarding requires a planar graph!

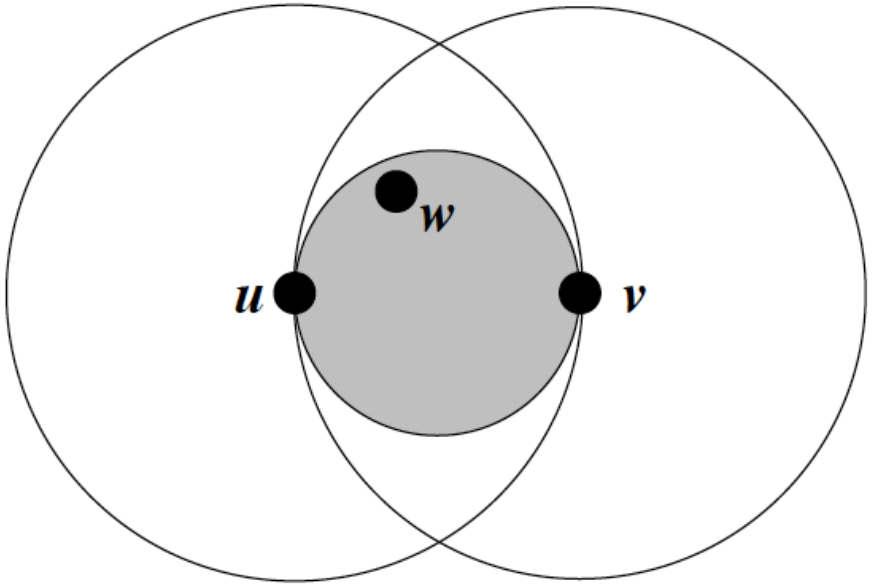
GPSR: Planar graph -> Relative neighborhood graph



An edge (u, v) exists between vertices u and v if the distance between them, $d(u, v)$, is less than or equal to the distance between every *other* vertex w, and whichever of u and v is farther from w. In equational form:

$$\forall w \neq u, v : d(u, v) \leq \max[d(u, w), d(v, w)]$$

GPSR: Planar graph -> Gabriel graph

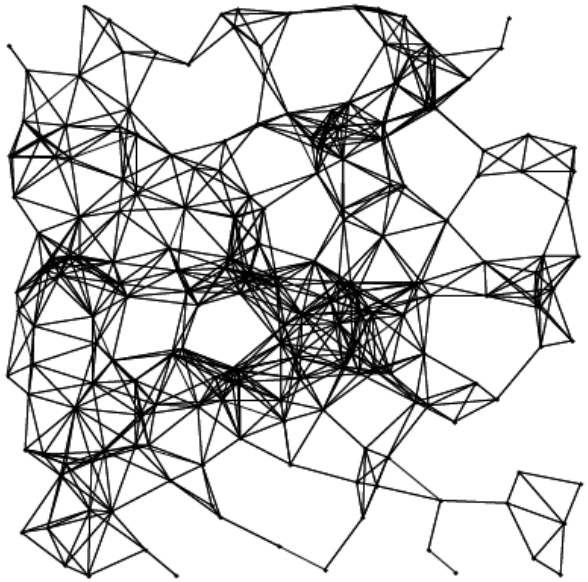


An edge (u, v) exists between vertices u and v if no other vertex w is present within the circle whose diameter is $\overline{uv}$. In equational form:

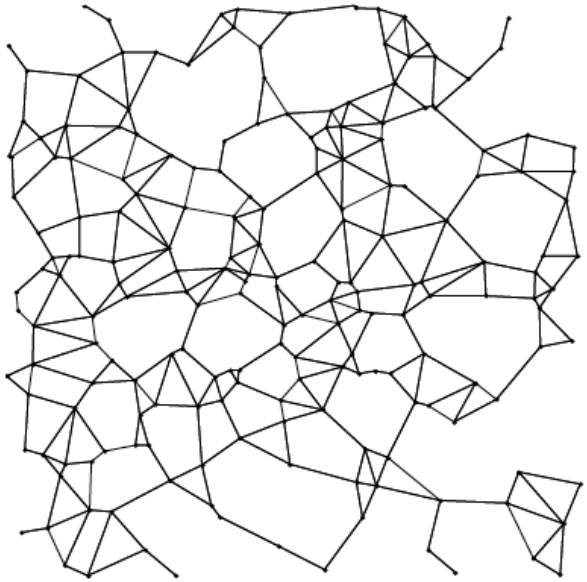
$$\forall w \neq u, v : d^2(u, v) < [d^2(u, w) + d^2(v, w)]$$

GPSR: Planar graph -> Example

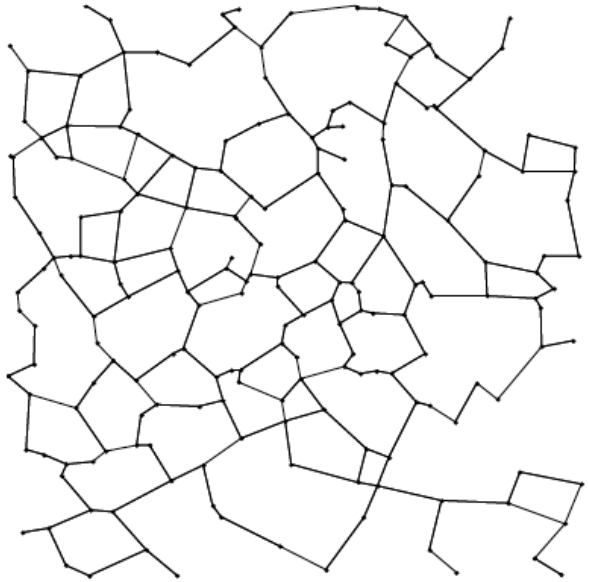
Full graph



Gabriel graph



Relative neighborhood graph



200 nodes, uniformly randomly placed on a 2000 x 2000 meter region, with a radio range of 250 m

GPSR

- Full GPSR algorithm combines both methods:
  - greedy forwarding on the full network graph
  - perimeter forwarding on the planarized network graph where greedy forwarding is not possible

- Packet header fields used in perimeter mode

Field	Function
D	Destination Location
L_p	Location Packet Entered Perimeter Mode
L_f	Point on $\overline{xV}$ Packet Entered Current Face
e_0	First Edge Traversed on Current Face
M	Packet Mode: Greedy or Perimeter

GPSR

- GPSR packet headers include a flag -> indicating …
  - greedy mode
  - perimeter mode
  - initially all packets are marked as greedy mode
- The packet source includes the geographic location of the destination.
  - Only the source sets the location destination field; it is left unchanged as the packet is forwarded through the network
- Upon receiving a greedy-mode packet for forwarding …
  - node searches its neighbor table for the neighbor geographically closest to the packet's destination
  - if the neighbor is closer to the destination, the node forwards the packet to that neighbor
  - when no neighbor is closer, the node marks the packet into perimeter mode

GPSR

- GPSR forwards perimeter-mode packets using a simple planar graph traversal.
  - when a packet enters perimeter mode at node x bound for node D, GPSR forwards it on progressively closer faces of the planar graph, each of which is crossed by the line x-D.
  - A planar graph has two types of faces. Interior faces are the closed polygonal regions bounded by the graph's edges.
- When a packet enters perimeter mode, GPSR records in the packet the location **Lp**, the site where greedy forwarding failed.
- Upon receiving a perimeter-mode packet for forwarding, GPSR first compares the location **Lp** in a perimeter-mode packet with the forwarding node's location.
- GPSR returns a packet to greedy mode if the distance from the forwarding node to D is less than that from Lp to D

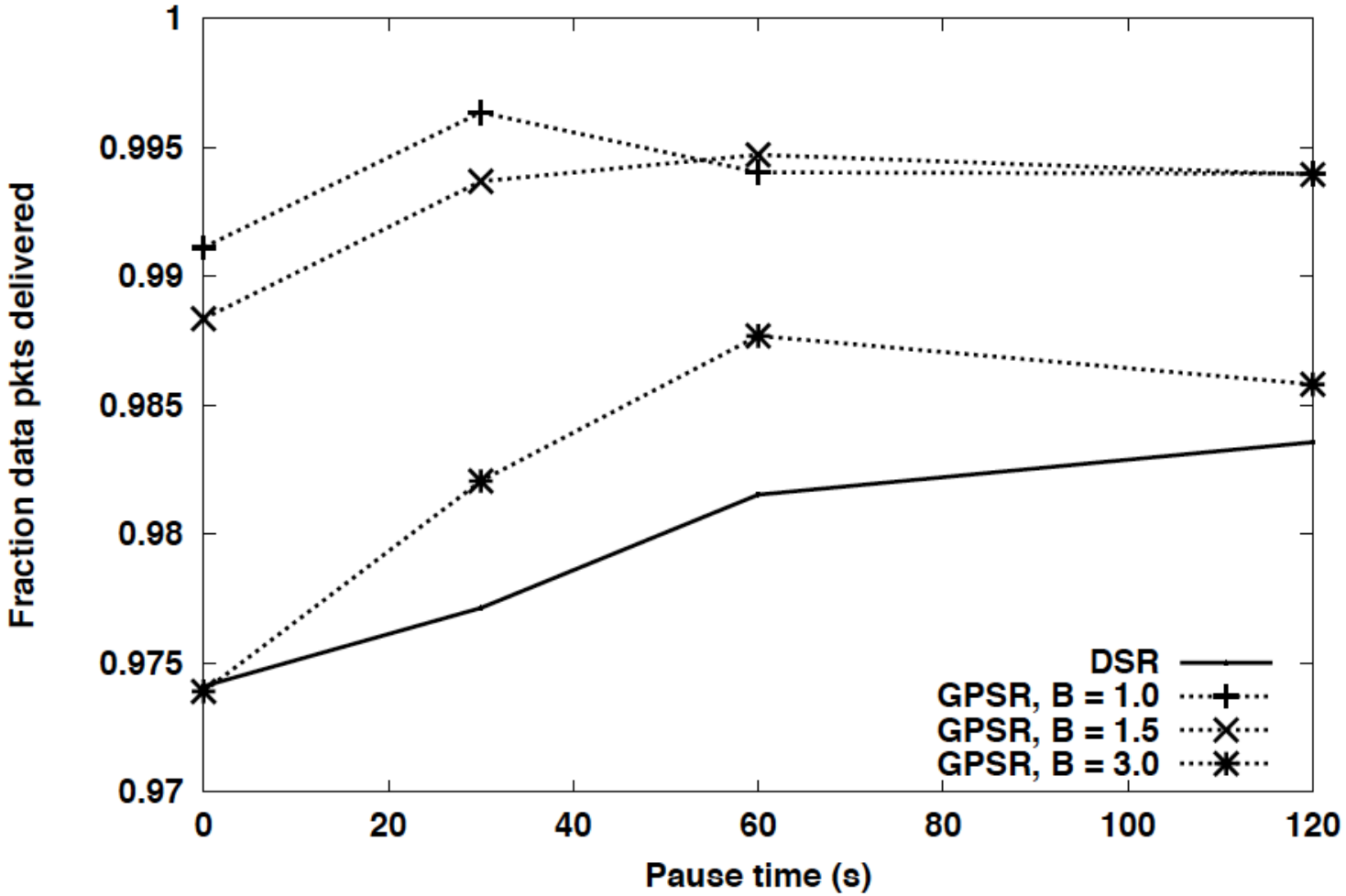
GPSR: Performance

- ns-2 simulations
- IEEE 802.11 MAC layer
- Random waypoint mobility model
- Considered topologies

Nodes	Region	Density	CBR Flows
50	1500 m × 300 m	1 node / 9000 m^2	30
112	2250 m × 450 m	1 node / 9000 m^2	30
200	3000 m × 600 m	1 node / 9000 m^2	30

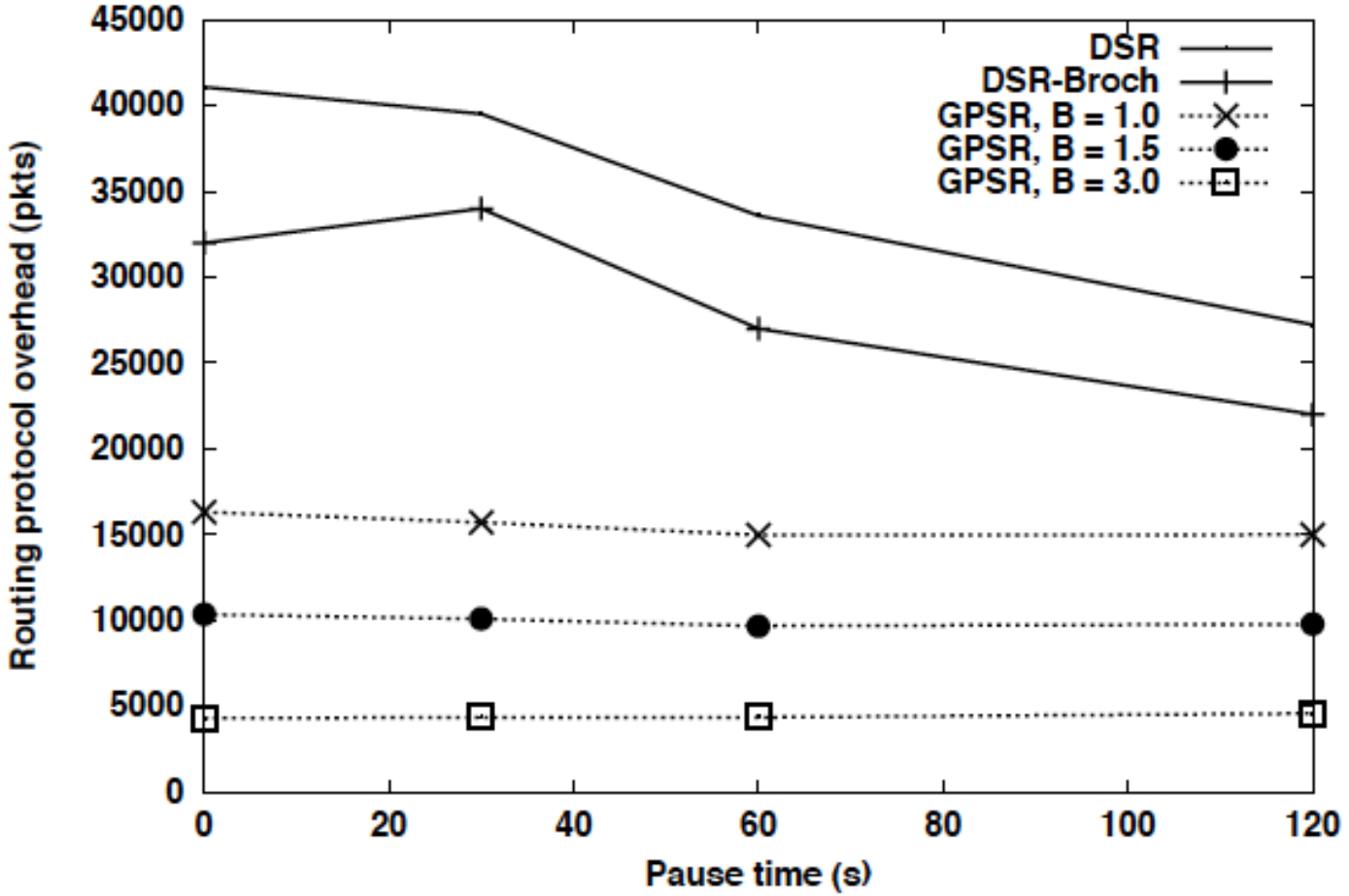
- Comparison to DSR

GPSR: Performance



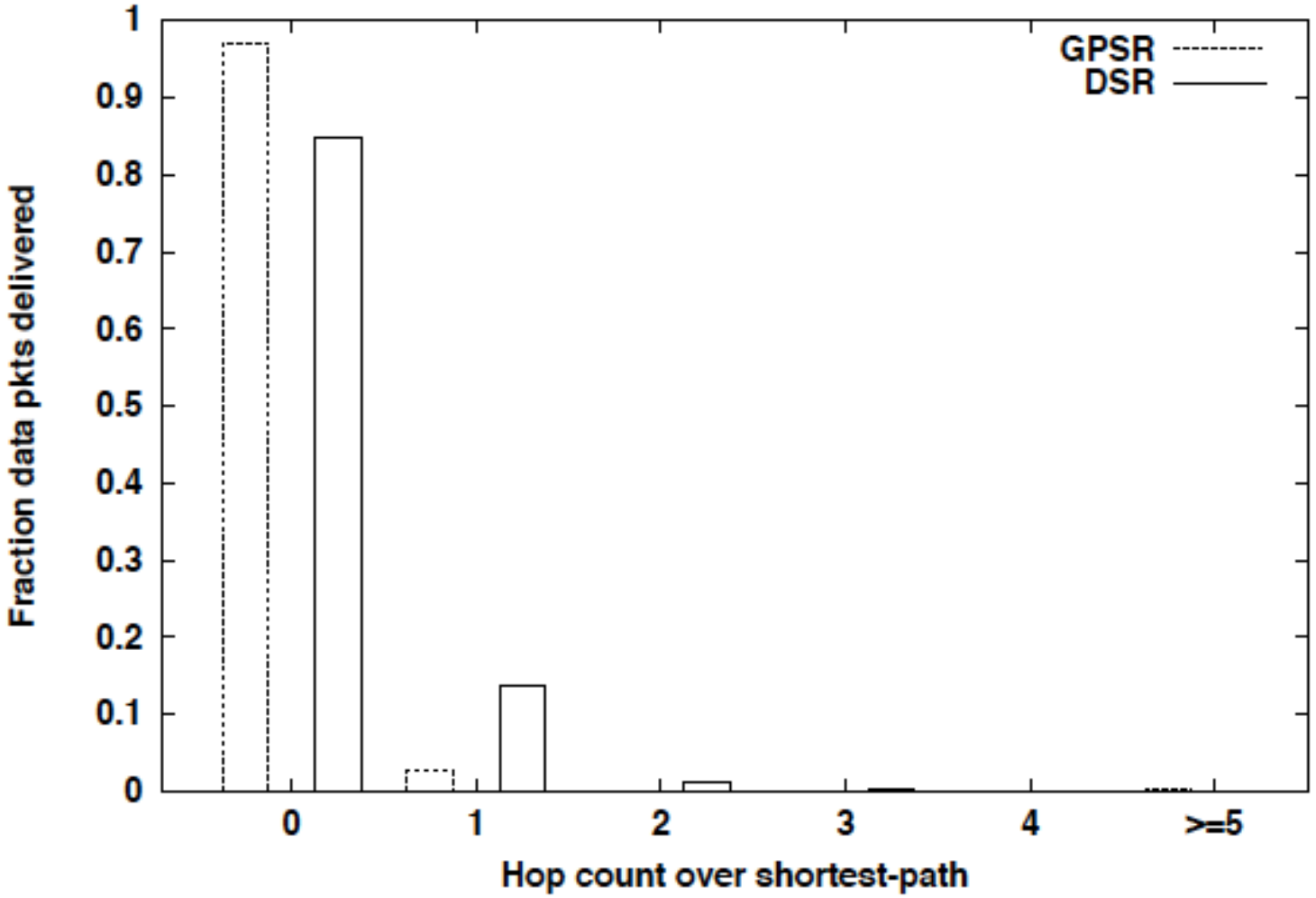
Packet Delivery Success Rate. GPSR with varying beacon intervals, **B**, compared with DSR. 50 nodes

GPSR: Performance



Routing Protocol Overhead. Total routing protocol packets sent network-wide during the simulation for GPSR with varying beacon intervals, **B**, compared with DSR. 50 nodes.

GPSR: Performance



Path length beyond optimal for GPSR's and DSR's successfully delivered packets. 50 nodes.

[RFC6550]

RPL

What is RPL?

- Routing Protocol for Low-power and Lossy Networks (RPL)
  - Open routing protocol for IP based smart object networks
  - IPv6-based routing solution
  - Standardized by the IETF

- IETF Working Group ROLL
  - Routing Over Low power and Lossy Networks (ROLL)
  - Low-power and Lossy Network (LLN)
  - Started in 2008

- Goal
  - Solution for many application areas of LLN

What is RPL?

- RPL was designed with the objective to meet the requirements of:
  - [RFC5867]
    Building Automation Routing Requirements in Low-Power and Lossy Networks
  - [RFC5826]
    Home Automation Routing Requirements in Low-Power and Lossy Networks
  - [RFC5673]
    Industrial Routing Requirements in Low-Power and Lossy Networks
  - [RFC5548]
    Routing Requirements for Urban Low-Power and Lossy Networks

Why standardization?

- Disadvantages of proprietary solutions
  - Lifetime of solutions depend on the strategy of the companies
  - Reduced or no compatibility and interoperability
    - "no solution is an island"
    - Gateway solutions to bridge different approaches are expensive
  - Communication of different (unknown) objects should be possible

- IP guarantees interoperability …
  - different link layer technologies
  - with existing networks

- Routing protocols for IP exist
  - But, are they usable in sensor networks and similar?

Why standardization?

- Routing approaches for the Internet not usable -> Too much overhead
  - Distance vector routing (DV)
  - Link state routing (LS)

  } Static metrics are used
- Requirements from sensor networks
  - Adaptive -> Routing metrics?
  - Constraint-based routing
    - Exclude nodes based on "constraints", e.g., battery state, link quality, …
  - Traffic patterns
    - Multipoint-to-Point (MP2P) -> nodes to sink
    - Point-to-Multipoint (P2MP) -> sink to nodes
    - Point-to-Point (P2P) -> node to node
    - Parallel paths (load balancing)
  - Configuration and management
    - Objective: as little as possible

RPL routing metrics: Categories

- Routing metrics may be categorized according to the following characteristics:
  - Link versus node metric
  - Qualitative versus quantitative
  - Dynamic versus static

- Metric vs. Constraints
  - Metric is used to compute the shortest path
  - Constraint exclude nodes (links)

RPL routing metrics: Examples

- Node State and Attributes (NSA)
  - 1-Bit Flag for aggregation -> node can act as traffic aggregator
  - 1-Bit Flag for node workload -> node is overloaded and may not be able to process traffic
- Node Energy
  - Node power mode -> powered, battery, scavenger
  - Estimated remaining lifetime
  - Details -> not specified
- Hop-Count
- Throughput
- Latency
- Link Reliability (ETX)

[RFC6550]

RPL

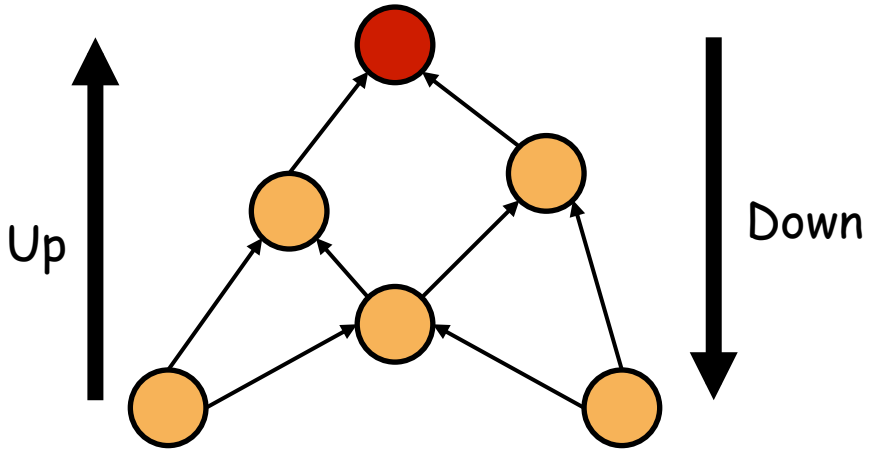
Basics of the protocol

Basic principles

- In a network multiple instances of RPL can be run concurrently
  - Each instance may have different constraints or performance criteria
- RPL separates packet processing and forwarding from the routing optimization
  - minimizing energy
  - minimizing latency
  - satisfying constraints
  - RFC6550 specifies only the operation of RPL
    - Other documents describe the remaining issues!

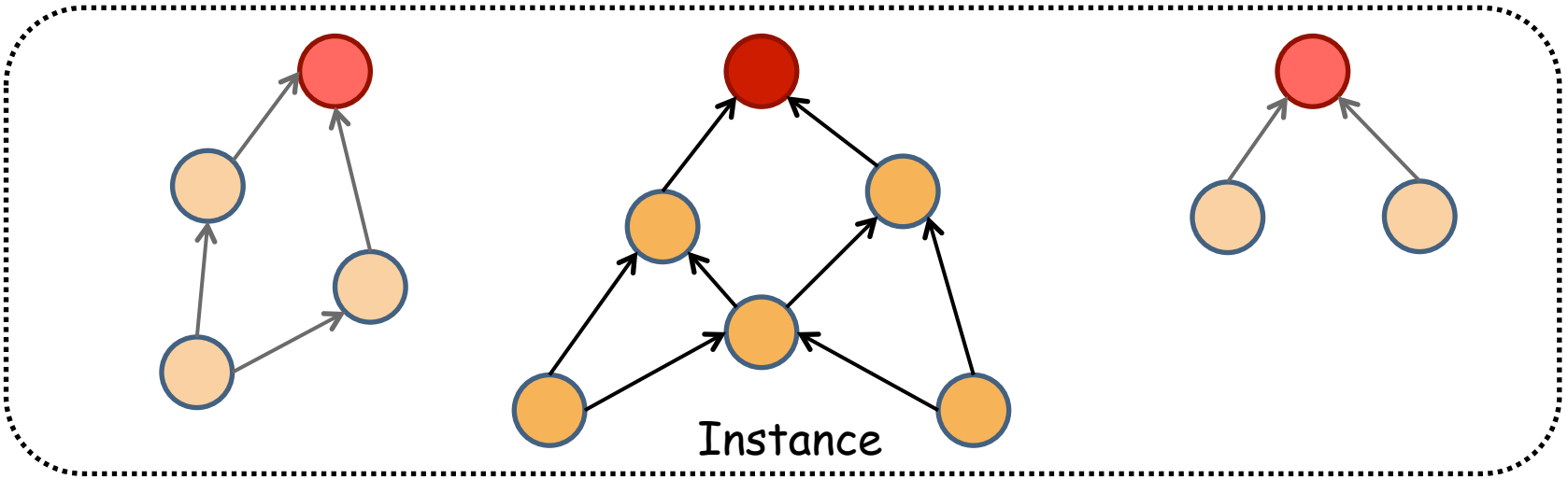
Basic principles

- Assumption: most of the traffic is bound by few nodes
  - Point-to-Multipoint (P2MP) -> sink to nodes
  - Multipoint-to-Point (MP2P) -> nodes to sink
- Approach: Create Directed Acyclic Graphs (DAG)
  - Up: direction from leaf nodes to DAG-Root for MP2P
  - Down: direction from root to leaf nodes for P2MP
  - Point-to-Point (P2P) uses both directions
    - First "up" to common parent, then "down" to destination



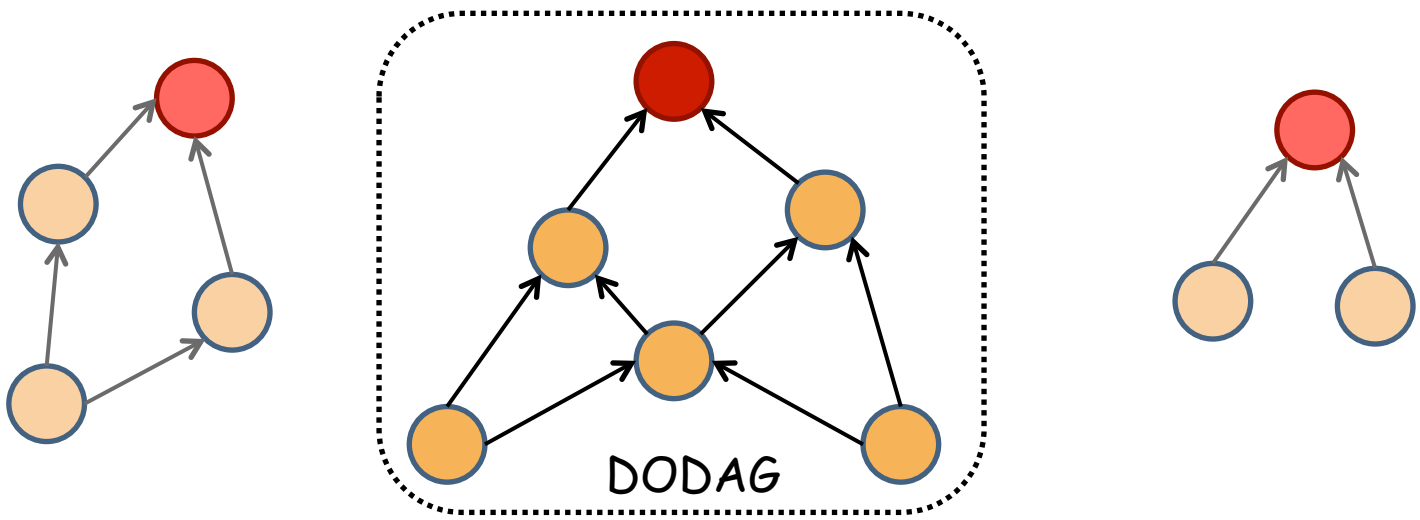
RPL Instance

- Contains one ore more routing topologies
- RPL instance may comprise
  - a single DODAG with a single root
  - multiple uncoordinated DODAGs with independent roots
  - a single DODAG with a virtual root that coordinates LLN sinks over a backbone network.



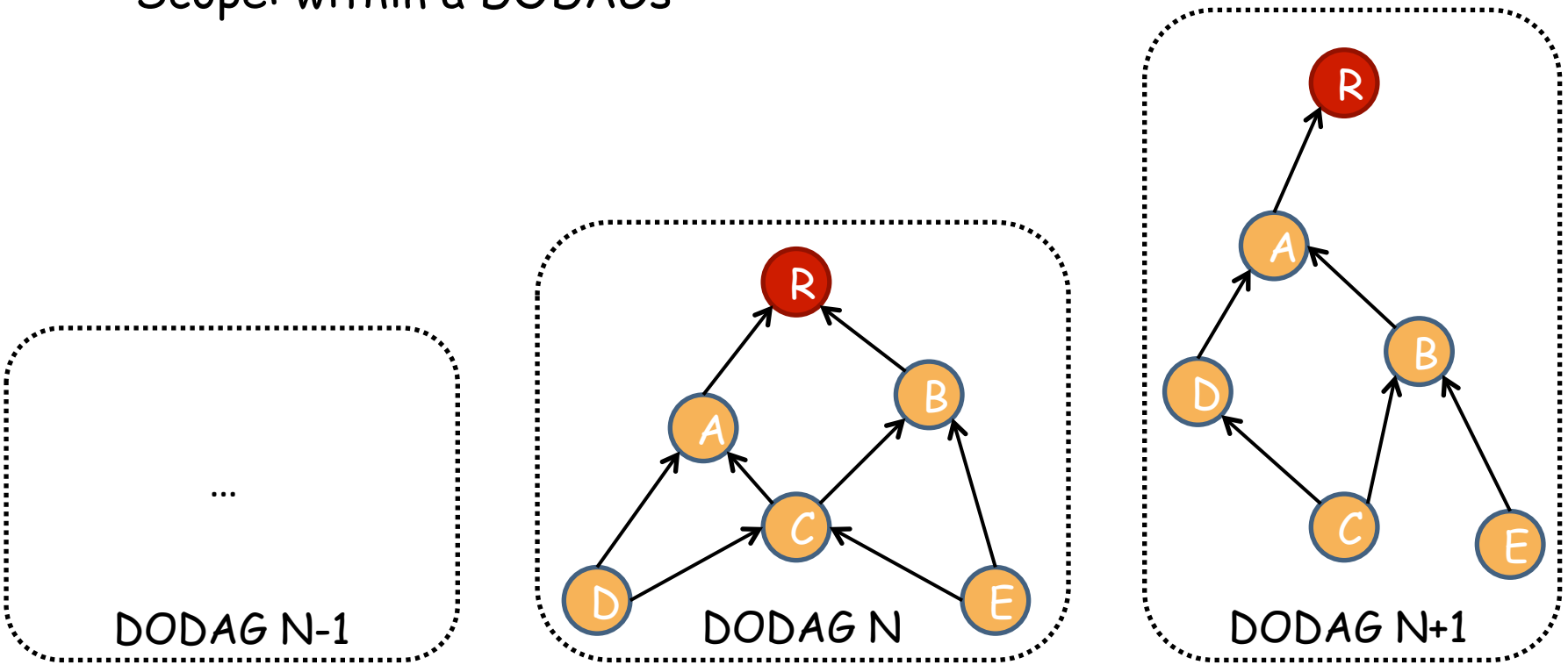
DODAG

- Destination Oriented DAG (DODAG)
  - A DAG rooted at a single destination
  - Scope: within a RPL instance
  - DODAG ID identifies a DODAG in a RPL instance
- A node can join a DODAG in a instance
  - Multi topology routing possible by joining into different instances



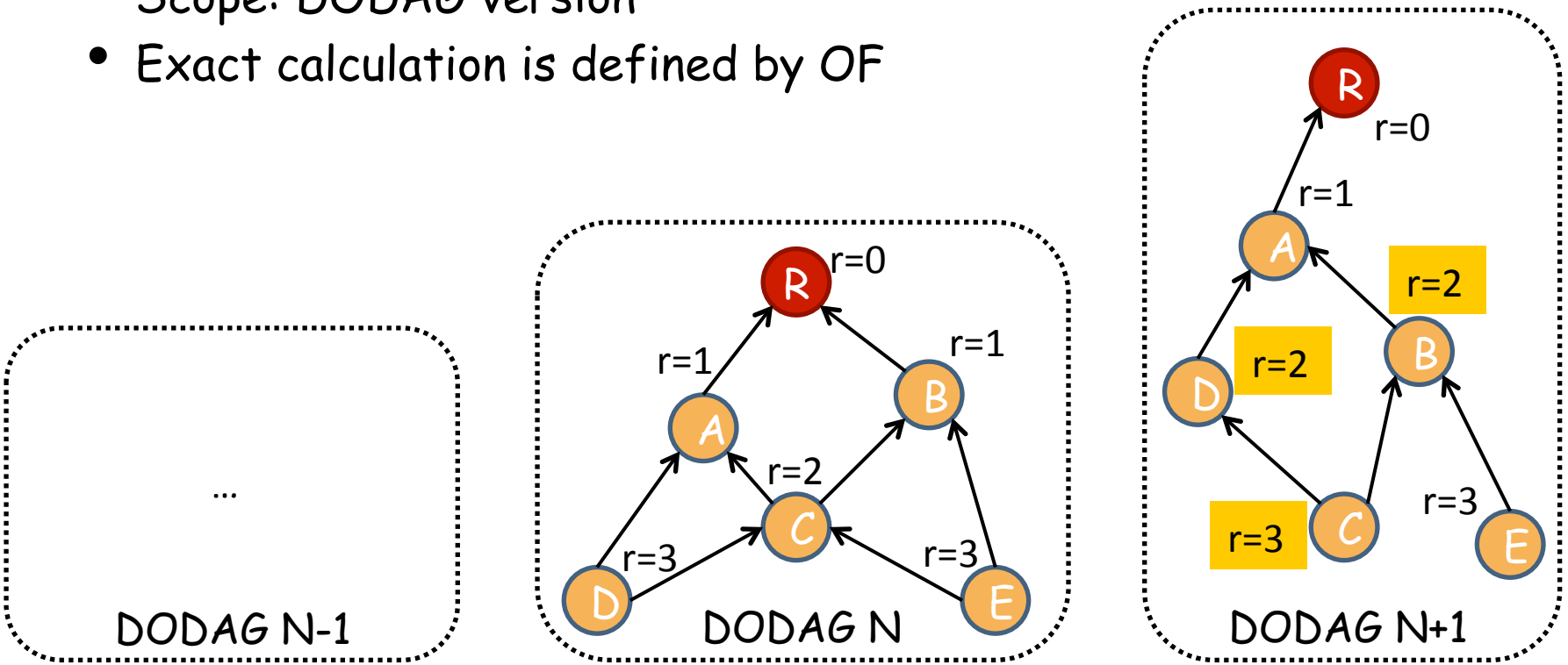
DODAG Version

- Controlled by the root (R)
  - Used for repairing of non-optimal DODAGs due to network dynamic
  - May change topology (as in N->N+1)
  - Scope: within a DODAGs



Node Rank

- Scalar value
  - Denotes the relative position of a node in DODAG from the root
  - Grows monotone down
    - Is used for loop detection in DODAG creation
  - Scope: DODAG version
  - Exact calculation is defined by OF



Objective Function (OF)

[Thu11]
[RFC6550]

- Objective Function (OF)
  - Abstract defined path metric
  - Consist of metrics and constraints
  - Example: Maximize ETX metric on the path, but exclude all battery driven nodes
- Defines the translation of metrics and constraints into Node Rank
  - OF defines how nodes select and optimize routes
  - Thus, which nodes can be selected as parent
  - Also, choice of one or several parent nodes
  - Scope: RPL instance

-> Is used to create DODAGs

  - Each DODAG version consists of the network topology in respect to OF

Control messages

- RPL defines new ICMPv6 format and three types:
  - DODAG Information Solicitation (DIS)
    Solicit DODAG information from a node
  - DODAG Information Object (DIO)
    The DODAG Information Object carries information that allows a node to discover a RPL Instance, learn its configuration parameters, select a DODAG parent set, and maintain the DODAG.
  - Destination Advertisement Object (DAO)
    is used to propagate destination information upwards along the DODAG.
  - For details see Section 6 of [RFC6550]

DODAG construction

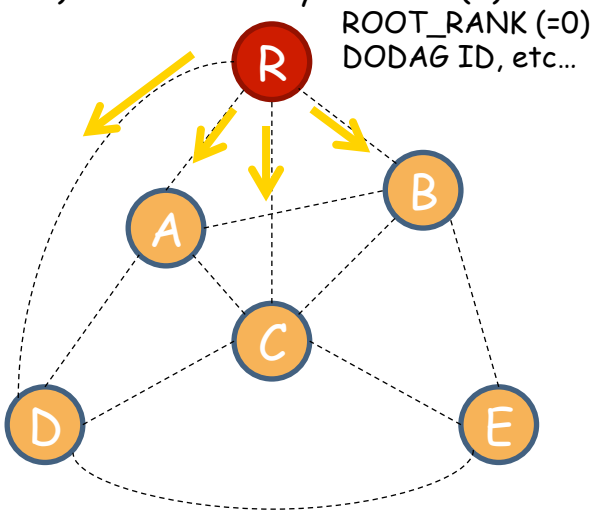
- Distributed algorithm to construct a DODAG
  - Some nodes are configured to be DODAG roots, with associated DODAG configurations.
  - Nodes advertise their presence, affiliation with a DODAG, routing cost, and related metrics by sending link-local multicast DIO messages to all-RPL-nodes.
  - Nodes listen for DIOs and use their information to
    - join a new DODAG -> selecting DODAG parents
    - maintain an existing DODAG -> i.r. to OF and Rank of neighbors
  - Nodes provision routing table entries, for the destinations specified by the DIO message, via their DODAG parents in the DODAG Version.
    - Nodes that join a DODAG can provision one or more DODAG parents as the next-hop for the default route

DODAG construction example: Assumptions

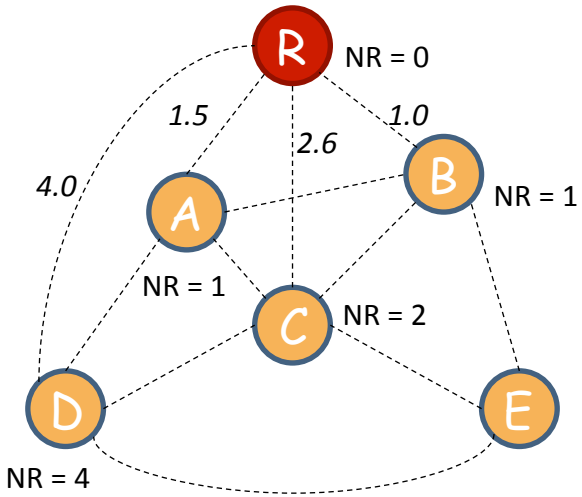
- NodeRank
  - Integer part of the minimum path cost to root
- Path cost (to root)
  - Sum of link costs on the path
- Link cost
  - Symmetric
  - Constant during DODAG construction
- Parent selection
  - Select two nodes
    - Minimize the path costs
  - Nodes are allowed to decrease their NodeRank
    - Prevents some problems
- DIOs contain
  - Own NodeRank
  - Own path cost
    - Path cost over the best parent node
- DIOs are sent
  - initially from the root
  - If NodeRank or path cost change

DODAG construction step by step

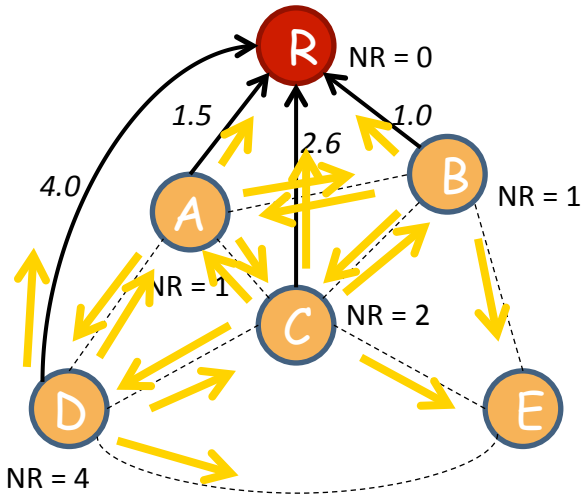
1.) Initialization by the root (R)



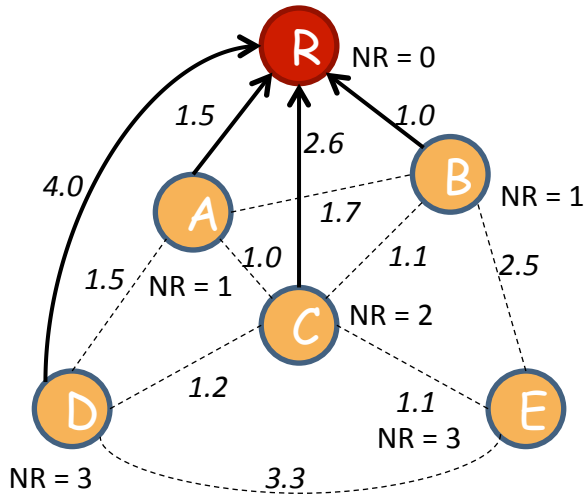
2.) Link costs and NodeRanks



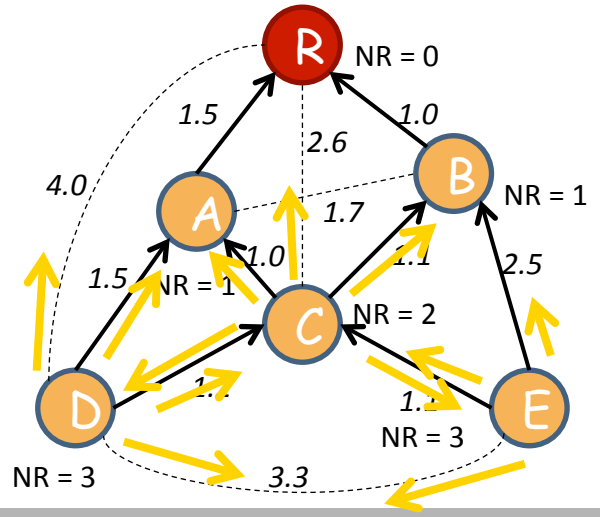
3.) Parent selection and DIO multicast



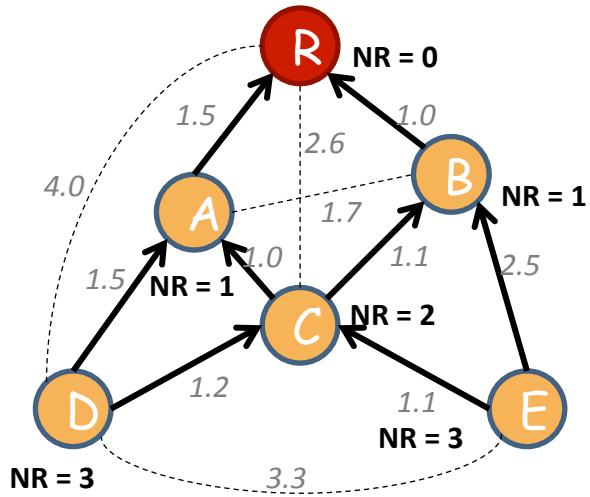
4.) Link costs and NodeRanks



5.) Parent selection and DIO



6.) Link costs and NodeRanks (no changes)



DODAG construction step by step

- Step 1
  - Root announces only ROOT_RANK
  - Other logical roots would use the same value
- Step 2
  - Random waiting time possible
  - Node D could have received DIOs from A and C -> R not a parent selected
- Step 4
  - Node D
    - Path cost(R) = 4.0
    - Path cost(A) = 3.0
    - Path cost(C) = 3.8

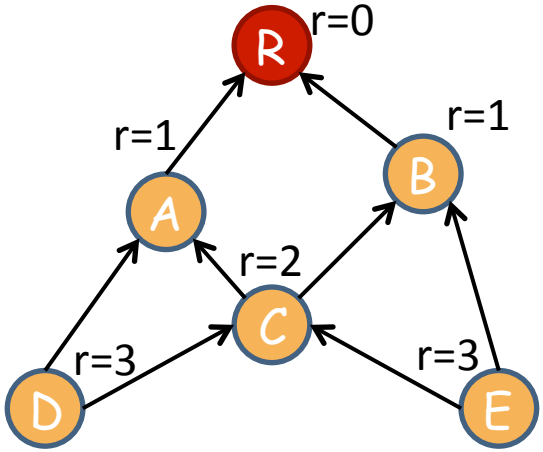
    -> min = 3.0 -> NodeRank = 3
  - Node C
    - Path cost(R) = 2.6
    - Path cost(A) = 2.5
    - Path cost(B) = 2.1
- Step 5
  - Path costs of nodes C, D, E change
  - NodeRank of D changes
  - Node E
    - Path cost(B) = 3.5
    - Path cost(C) = 3.7

    -> own path cost = 3.5

  -> new DIO multicast
  - B and A cannot select second parent
    - None of nodes has a suitable NodeRank (smaller than 1)
- Step 6
  - Node E has to send again DIO
    - Path cost(C) = 3.2
    - -> own path cost = 3.2
  - Step not shown since no topology change

DODAG construction

- Distance vector protocol
  - Announcement of the best path to the root
  - Parent nodes are selected in way that they minimize path costs
    - Selection of parent set and a default parent
    - Depending on the OF selection of more default parent nodes possible
- Problems to be considered
  - Loops
  - Count-to-infinity



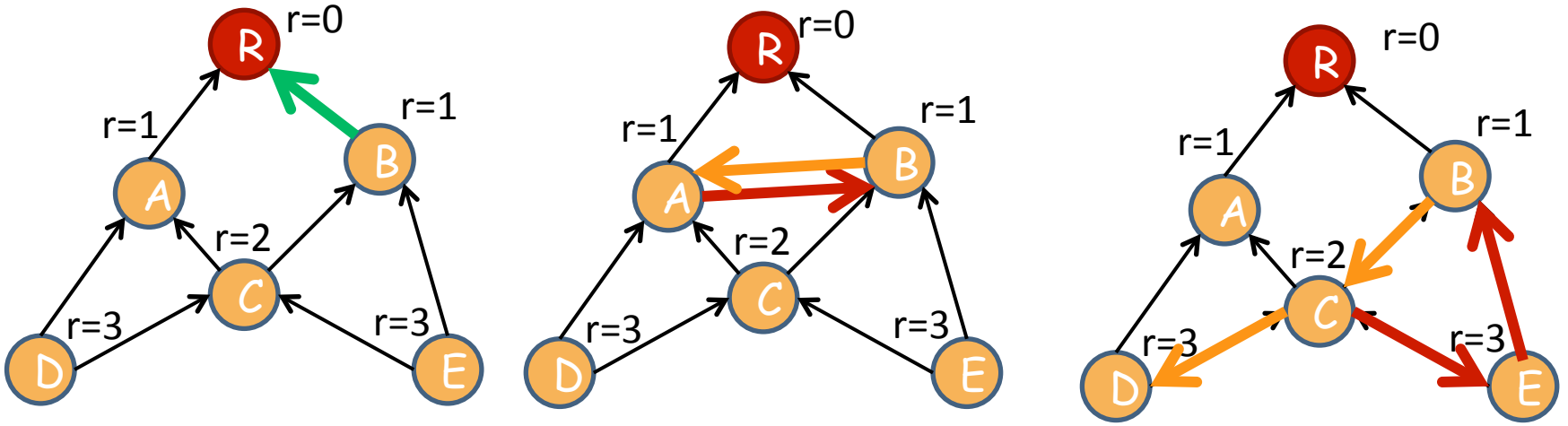
Maintenance of downwards routes

- Required to provide Point-2-Multipoint and Point-2-Point traffic
- Two modes available
  - Storing: Node stores routing table, forwarding based on stored information
  - Non-storing: No routing tables stored, forwarding based on source routing

- P2P packets travel Up toward a DODAG root then Down to the final destination
  - Non-Storing case: The packet travels all the way to a DODAG root before traveling Down.
  - Storing case: The packet may be directed Down towards the destination by a common ancestor of the source and the destination prior to reaching a DODAG root.

Node Rank relationships

- Node Rank is defined by OF
  - Rank(R) < Rank(B) -> B can select R as parent without loop risk
    - R is closer to the root than B -> cannot be successor of B
  - Rank(A) = Rank(B) -> similar Node Rank -> similar distance to root
    - Loop may occur if A routes also through a sibling node
  - Rank(C) > Rank(B) -> C has higher distance to root. Loop is created if node C is also a successor of B, e.g., over B

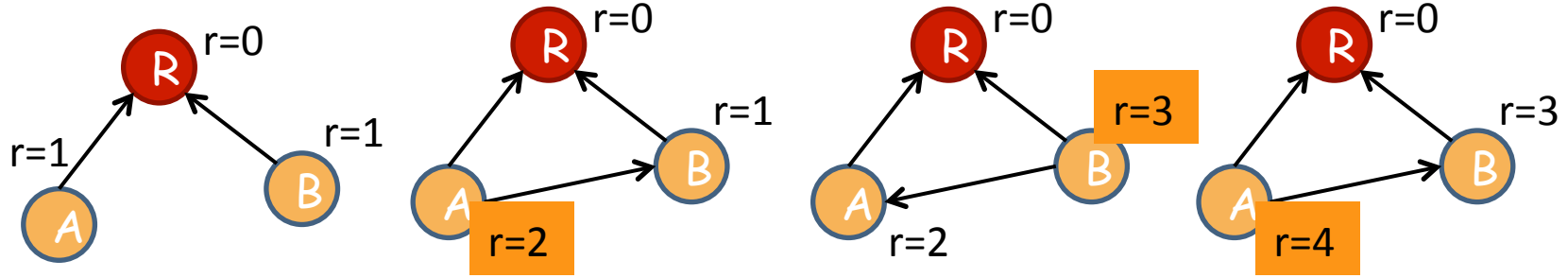


Node Rank / Greediness and Count-to-infinity

- It may be favorable to have a high Node Rank
  - Larger set of potential parent nodes
    - Improved robustness
  - Smaller number of nodes can select this node as parent (Loop prevention) -> energy save, less packet forwarding
- Problem of greediness -> example

  1: Node A leaves DODAG and joins with higher Rank

  2: Node B does the same -> goto 1:



Security

- RPL supports three security modes:
  - Unsecured:
    - Basic operation without security sections in the RPL messages
  - Preinstalled:
    - RPL with secure messages.
    - Nodes need preinstalled keys to join a RPL instance as host or router
    - Provides message confidentiality, integrity, and authenticity
  - Authenticated:
    - As Preinstalled mode, but …
    - A node can join only as host
    - To join as router, node must obtain a second key from a key authority.
- For details see [RFC6550]

RPL

Discussion

Discussion


[Clausen]

- Assumptions about the traffic
  - MP2P is predominant
  - P2MP is rare
    - Communication to actors is not considered
  - P2P is "somewhat esoteric"
    - Routes are not always optimal, in the Non-Storing mode packets are routed always over the root!
- Are assumptions valid for all scenarios / applications?
- Generation of "down" routes with DAOs results in high signaling overhead

Discussion

- Fragmentation on the link layer
  - IEEE 802.15.4 limits the frame size to 127 bytes
  - RPL designed for IPv6
    - Deviating from the specified L3 MTU of 1280 bytes
    - Header of 40 octets can be compressed to 2 octets
  - So 127 octets minus
    - 25 octets L2 frame overhead
    - 21 octets link layer security
    - 2 octets compressed IPv6 header

  -> Results in 79 octets for L3 payload
- Is this enough?
  - RPL control traffic is transmitted with ICMPv6 (header 4 octets)
  - DIOs can grow up to 80 octets

    -> It is not enough, thus fragmentation is required
  - If source routing is used then L3 header grows in O(# hops)
- Notice: Loss of a fragment means the loss of the packet!
  - And: the network is defined as "lossy"

Discussion

- Aggregation of addresses
  - In storing mode an RPL router has to store routing entries for all destinations in its sub-DODAG
    - Realized with prefixes -> assumes that addresses are aggregable
  - Change of parent is expected behavior -> Change point of attachment towards the DODAG root
- Change of parent results in …
  - Change of address of the router and
  - Change of addresses in the sub-DODAG… (!)
- Usefulness of RPL without aggregation of L3 addresses limited
  - Routing tables can grow "very large"
    - In comparison to the Internet "small"
    - But, memory is "low" in these networks
- Thus, leading simply to a Tree-Topology instead a DODAG
  - Parent selection restricted

Discussion

- Bidirectionality hypothesis
  - Parents are selected based on received DIOs from them
    - No verification that router can reach parent
  - Basic use of link is "Upwards"

- Symmetry of wireless links is problem
  - Unidirectional links and link asymmetry is very common
  - May result in that router does not have connection to its parent

Summary

- Discussed approaches
  - Probabilistic routing
    - Flooding, Gossiping
    - Rumor Routing
  - Content based routing
    - Directed Diffusion
  - Geographical routing
    - Greedy Strategy with respect to distance / direction
    - Greedy Perimeter State Routing
  - RPL
    - Step towards a standard for routing in LLN

Literature

[Juraschek] Juraschek, F., M. Günes, M. Philipp, B. Blywis, "State-of-the-art of distributed channel assignment", no. TR-B-11-01: Freie Universität Berlin, FB Mathematik und Informatik, Jan, 2011

[Ganesan] Ganesan, Krishnamachari, Woo, Culler, Estrin and Wicker, "Complex Behavior at Scale: An Experimental Study of Low-Power Wireless Sensor Networks", UCLA Computer Science Technical Report UCLA/CSD-TR 02-0013, 2002

[Ni] Ni, S.Y., Tseng, Y.C., Chen, Y.S., Sheu, J.P.: "The broadcast storm problem in a mobile ad hoc network", MobiCom'99, 1999

[Blywis] Bastian Blywis, Philipp Reinecke, Mesut Günes, Katinka Wolter, "Gossip Routing, Percolation, and Restart in Wireless Multi-Hop Networks", IEEE Wireless Communications and Networking Conference (2012)

[Akkaya] Kemal Akkaya, Mohamed Younis, "A survey on routing protocols for wireless sensor networks", Ad Hoc Networks, 2005

[AlKK04] J. Al-Karaki & A. Kamal, "Routing techniques in wireless sensor networks: A Survey", in IEEE Wireless Communications, 2004, 11, p. 6-28

[Braginsky] D. Braginsky & D. Estrin, "Rumor routing algorithm for sensor networks", Proc. of the 1st ACM international workshop on Wireless sensor networks and applications, ACM Press, 2002, 22-31

[Haas] Z. J. Haas, J. Y. Halpern, L. Li, "Gossip-Based Ad Hoc Routing", Proc. IEEE Infocom, New York, USA, Juni 2002

[Günes] Mesut Günes, Udo Sorges, Imed Bouazizi, "ARA - The Ant-Colony Based Routing Algorithm for MANETs", In Stephan Olariu, editor, Proceedings of the 2002 ICPP Workshop on Ad Hoc Networks (IWAHN 2002), pages 79-85. IEEE, August, 2002.

Literature

[Intanagonwiwat] C. Intanagonwiwat, R. Govindan, D. Estrin, "Directed Diffusion: A Scalable and Robust Communication Paradigm for Sensor Networks", Proc. ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom), USA, August 2000

[Intanagonwiwat03] C. Intanagonwiwat, R. Govindan, D. Estrin, "Directed Diffusion for Wireless Sensor Networking", IEEE/ACM Transactions on Networking, Vol. 11, No. 1, Februar 2003

[Takagi] H. Takagi & L. Kleinrock, "Optimal Transmission Ranges for Randomly Distributed Packet Radio Terminals", IEEE Transactions on Communications, 1984, 32, 246-257

[Karp] B. Karp & H. T. Kung, "GPSR: Greedy perimeter stateless routing for wireless networks", MobiCom 2000, Proceedings of the 6th annual international conference on Mobile computing and networking, ACM Press, 2000, 243-254

[Kuhn] F. Kuhn, R. Wattenhofer, A. Zollinger, "Worst-Case optimal and average-case efficient geometric ad-hoc routing", Proceedings of the 4th ACM international symposium on Mobile ad hoc networking & computing, ACM, 2003, S. 267-278

[RFC6550] Winter et al., "RPL: IPv6 Routing Protocol for Low power and Lossy Networks", Internet-Draft, March 2012, http://tools.ietf.org/html/rfc6550.

[Thubert] B. Thubert, Ed., "RPL Objective Function 0 draft-ietf-roll-of0-20", Internet-Draft, September 2011, Expires 08. March 2012, Work in Progress.

[Clausen] T. Clausen, U. Herberg, M. Philipp, "A Critical Evaluation of the „IPv6 Routing Protocol for Low Power and Lossy Networks" (RPL)", May 2011, Rapport de recherche, INRIA.