Technische Universität Berlin

Networked Embedded Systems [NES]

Ad Hoc Routing

Course Overview

- Sensor networks
  - Principles and applications
- Wireless communications
  - Concepts of modulation and encoding on the physical layer
- Wireless access
  - Typical medium access protocols for low-power sensor nodes
- Design and architecture of embedded systems
  - Architecture of embedded systems, programming paradigms
- **Routing**
  - **Ad hoc routing and data centric communication**
- Clustering
  - Clustering algorithms, guaranteed connectivity
- Time synchronization
  - Clock vs time synchronization, distributed algorithms
- Localization
  - Ranging techniques, localization algorithms

Ad Hoc Routing

Characteristics of an Ideal Routing Protocol

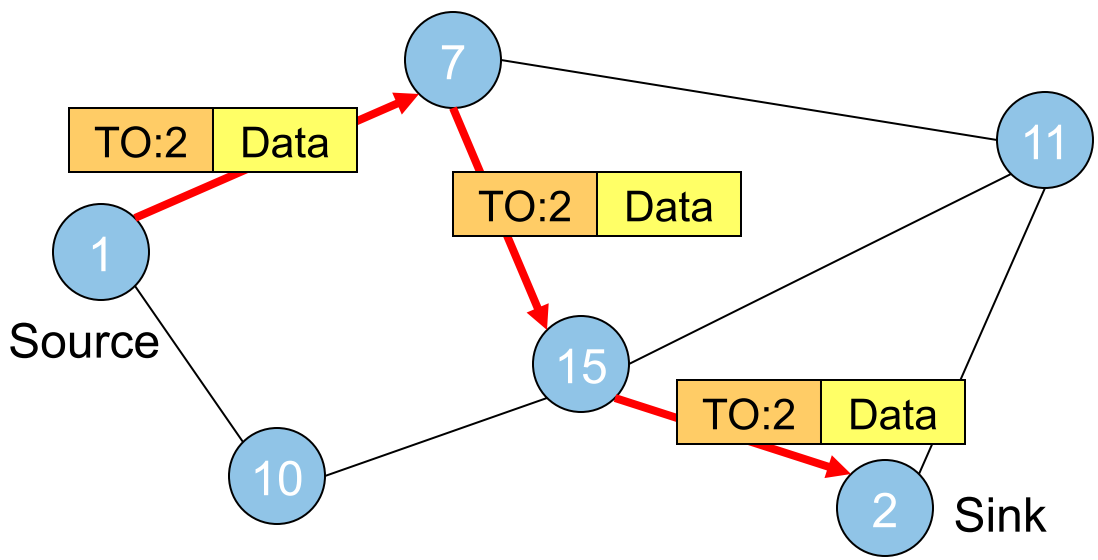
- Requirements
  - Fully distributed (scalability)
  - Adaptive (topology changes)
  - Minimum number of nodes involved for route computation
  - Localized state (reduced global state)
  - Loop-free, free from stale routes
  - Limited number of broadcasts (collision avoidance)
  - Quick and stable convergence
  - Optimal resource utilization (bandwidth, processing, memory, battery)
  - Localized updates
  - Provision of QoS as demanded by the applications

- Typical problems (wireless networking)
  - Node mobility
  - Unreliable radio communication
  - Limited energy resources

Address-based routing vs. data-centric forwarding

- Address-based routing
  - Directed towards a well-specified **particular destination** (sink)
  - Support for unicast, multicast, and broadcast messages

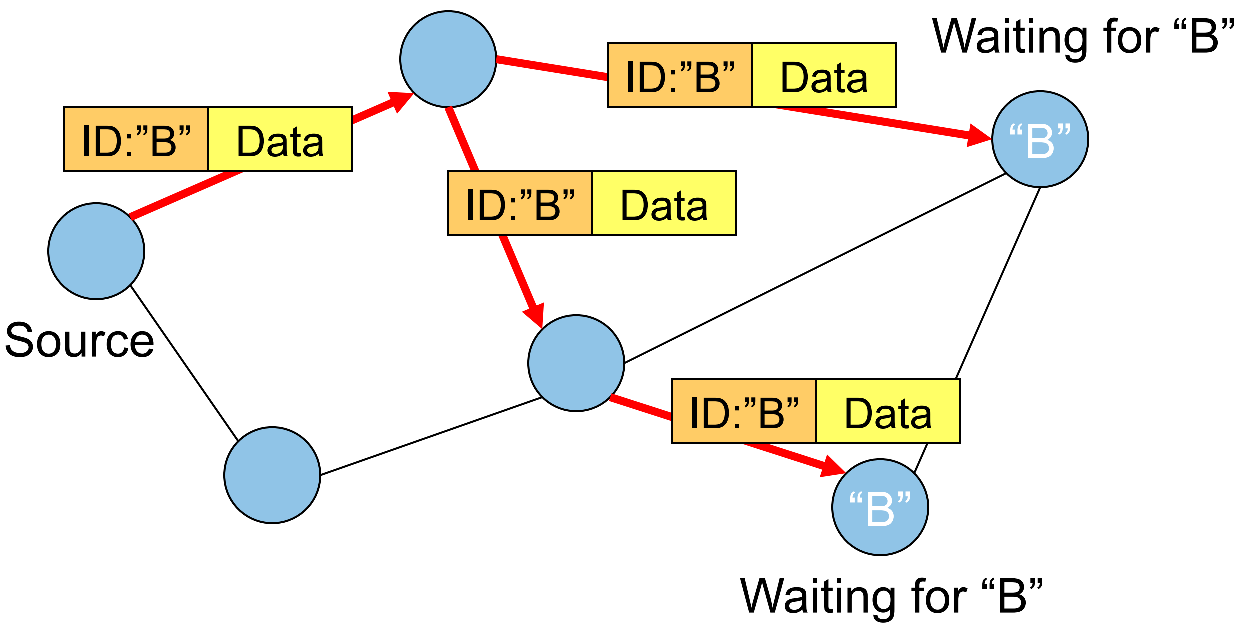
  - → Topic of this chapter



Address-based routing vs. data-centric forwarding

- Data-centric forwarding
  - Forwarding of messages to all / some **appropriate nodes**
  - Routing decisions according to the "data"

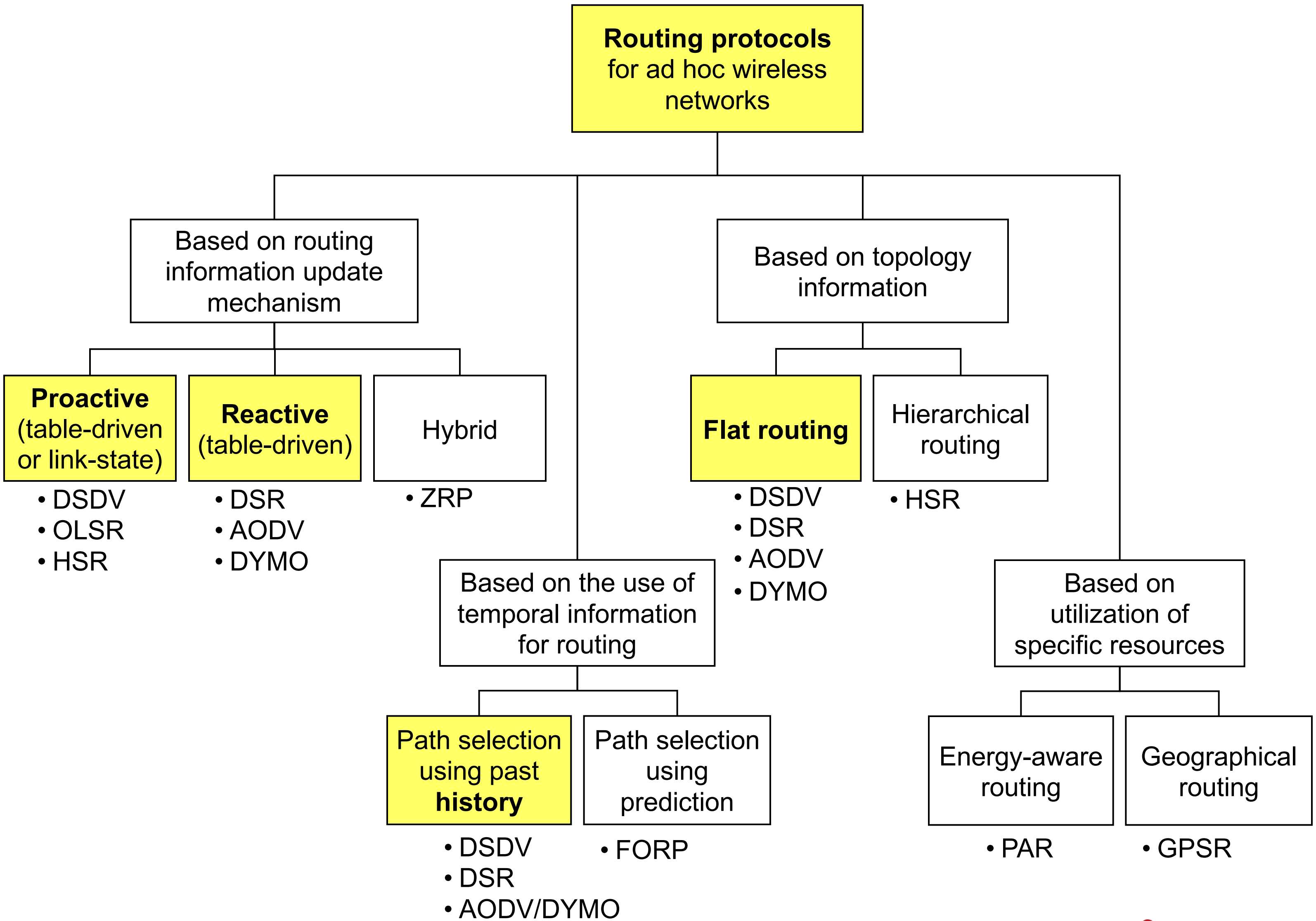
  - → Topic of the next chapter



Address-based routing vs. data-centric forwarding

	Address-based routing	Data-centric forwarding
Routing approach	Identification of a path according to the destination address of the data message	Determination of the destination of a data message according to the content of the packet
Prerequisites	Network-wide unique addresses	Pre-defined message types and semantics
Routing techniques	Proactive routing (continuous state maintenance) or reactive routing (on-demand path finding)	(probabilistic) flooding schemes or interest-based reverse routing
Advantages	Usually low delays in connection setup and data dissemination	No address information required and simplified self-management and redundancy
Disadvantages	Network-wide unique address identifiers required	Increased overhead for single transmissions

Classification of Ad Hoc Routing Protocols



Classification of Ad Hoc Routing Protocols

- Routing information update mechanism
  - ***Proactive*** or table-driven routing protocols
  - ***Reactive*** or on-demand routing protocols
  - ***Hybrid*** routing protocols

- Use of temporal information for routing
  - Routing protocols using ***past temporal information***
  - Routing protocols that use ***future temporal information***

- Routing topology
  - ***Flat topology*** routing protocols
  - ***Hierarchical*** topology routing protocols

- Utilization of specific resources
  - ***Power-aware*** routing
  - ***Geographical*** information assisted routing

DSDV

Proactive Protocols – DSDV

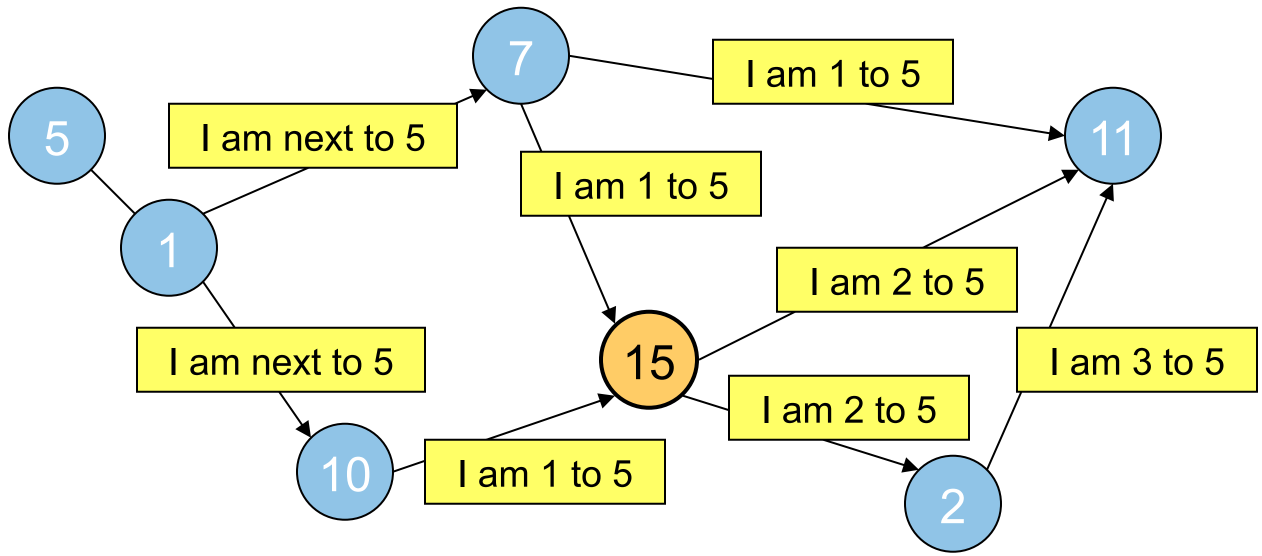
- Idea: Start from a +/- standard routing protocol, adapt it
- Adapted distance vector: ***Destination Sequence Distance Vector (DSDV)***
  - Based on distributed Bellman Ford procedure
  - Add ***aging*** information to route information propagated by distance vector exchanges; helps to avoid routing loops
  - Periodically send full route updates
  - On topology change, send incremental route updates
  - Unstable route updates are delayed
  - … + some smaller changes

DSDV

- Setup: exchange of routing tables

Dest	Next	Dist	Seq
7	7	1	12
1	7	2	26
5	7	3	26
…	…	…	…

Routing table at node 15



- Errors: update messages are created by the end of the broken link with the broken link's weight assigned to infinity (∞) and with a new sequence number greater than the stored number for this destination

DSDV

- Advantages
  - Availability of routes to all destinations at all times implies much less delay in route setup
  - Incremental updates with sequence number tags allows to adapt existing wired network protocols

- Disadvantages
  - Updates due to broken links lead to a heavy control overhead during high mobility
  - Even a small network with high mobility or a large network with low mobility can completely choke the available bandwidth
  - → exhaustive control overhead proportional to the number of nodes
  - → not scalable in ad hoc wireless networks
  - To obtain information about a particular destination node, a node has to wait for a table update message initiated by the same destination node
  - → delayed updates
  - → could result in stale routing information

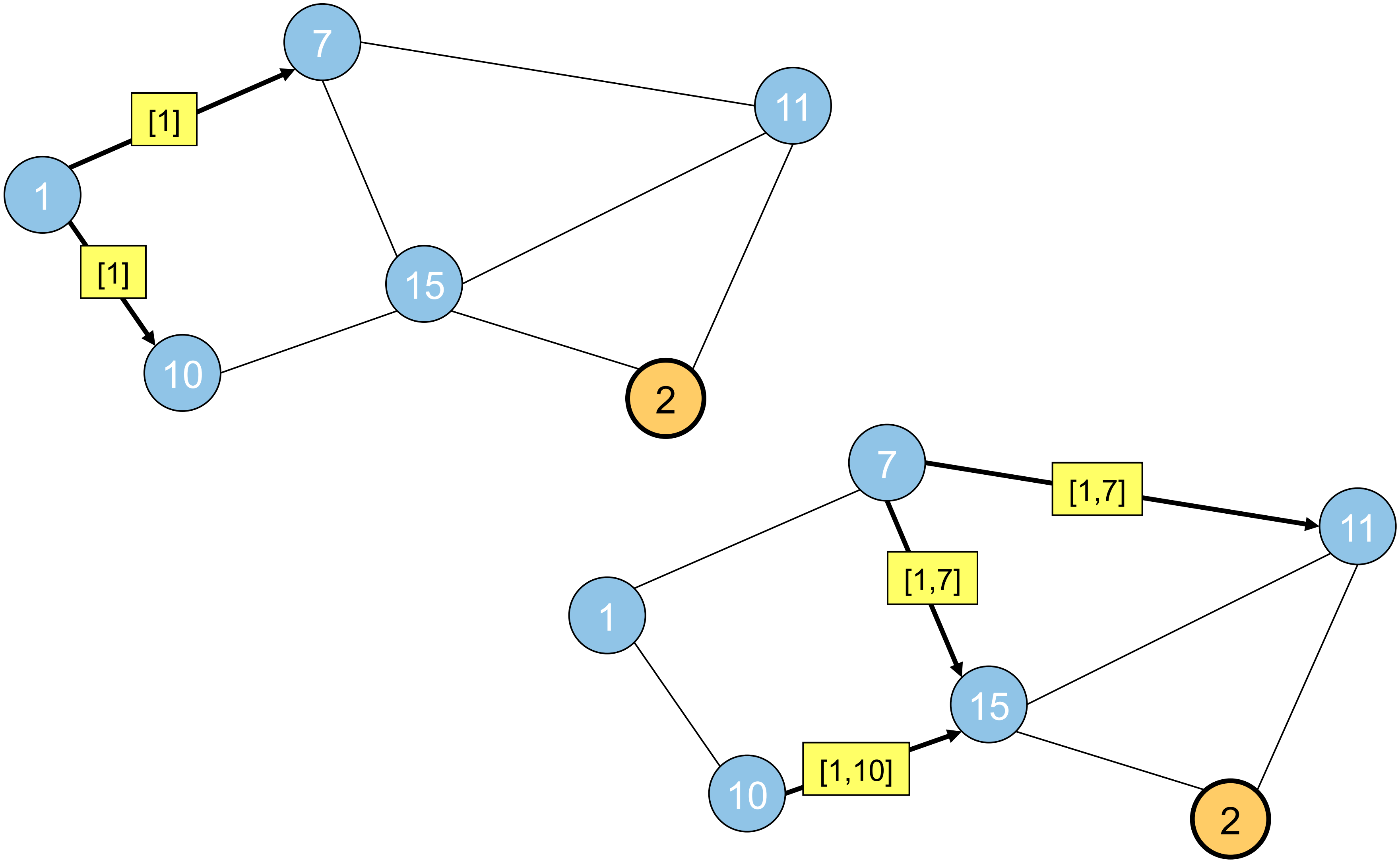
DSR

Reactive Protocols – DSR

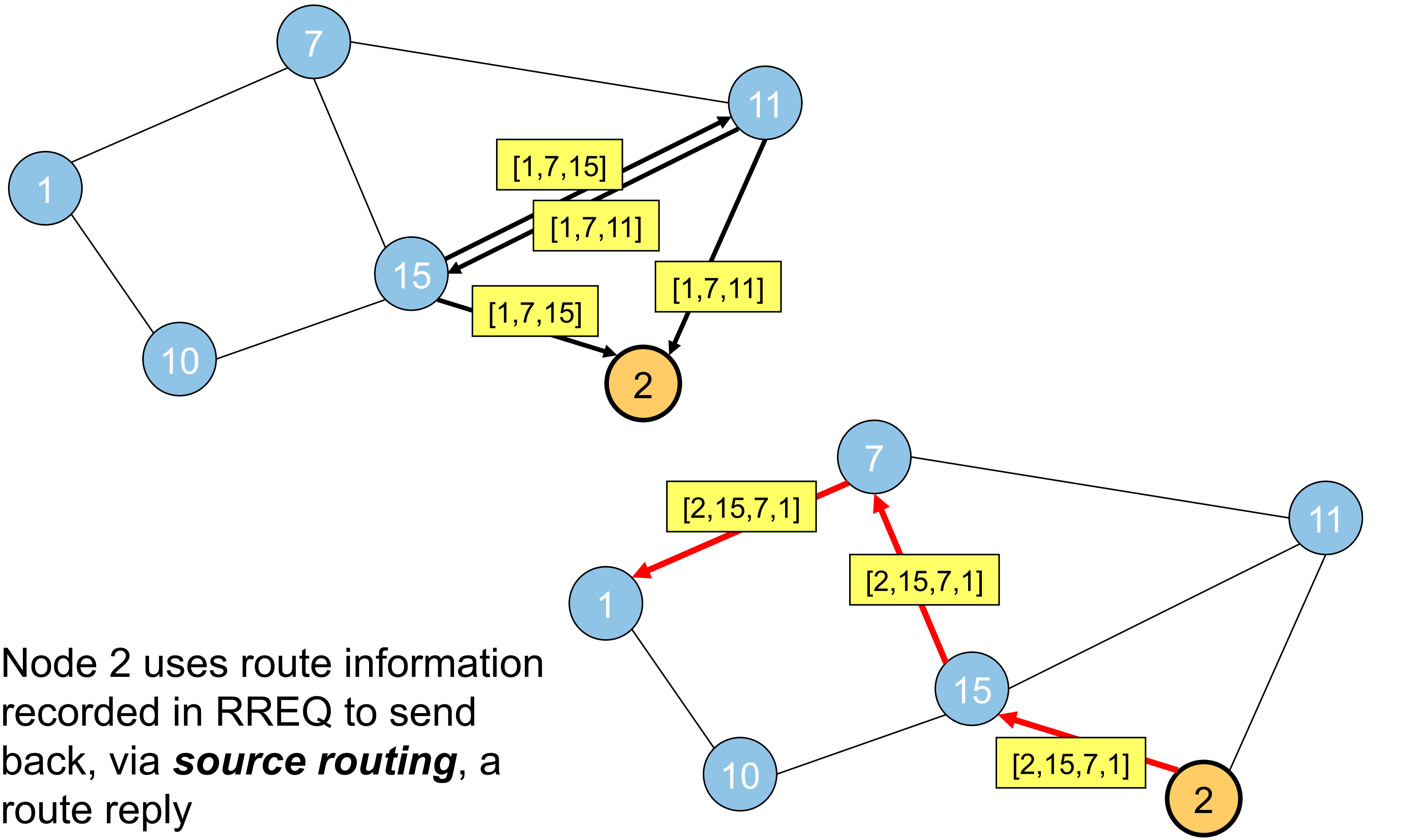
- In a reactive protocol, how to forward a packet to destination?
  - Initially, no information about next hop is available at all
  - One (and only?) possible recourse: Send packet to ***all*** neighbors – flood the network
  - Hope: At some point, packet will reach destination and an answer is sent back – use this answer for ***backward learning*** the route from destination to source

- Practically: ***Dynamic Source Routing (DSR)***
  - Use separate ***route request/route reply*** packets to discover route
    - Data packets only sent once route has been established
    - Discovery packets smaller than data packets
  - Store routing information in the discovery packets

DSR Route Discovery Procedure



DSR Route Discovery Procedure



Node 2 uses route information recorded in RREQ to send back, via ***source routing***, a route reply

DSR

- Route cache
  - Used to store all possible information extracted from the source route contained in a data packet
  - Used to optimized the route construction phase
  - → Problem: stale route caches

- Optimizations
  - Many nodes might know an answer – reply storms
  - → Exponential backoff to avoid frequent RouteRequest packets
  - Piggy-backing data packets on the RouteRequest

- Route maintenance
  - If a link breaks, a RouteError message is sent towards the source
  - Route construction is re-initiated

DSR

- Advantages
  - Reactive approach eliminating the need to periodically flood the network with table update messages
  - Less storage and maintenance requirements
  - Connection performs well in static and low-mobility environments

- Disadvantages
  - Connection setup delay is higher than in table-driven approaches
  - Does not locally repair broken links
  - Stale route information may result in inconsistencies
  - Performance degrades with increasing mobility
  - Routing overhead is directly proportional to the path length

AODV

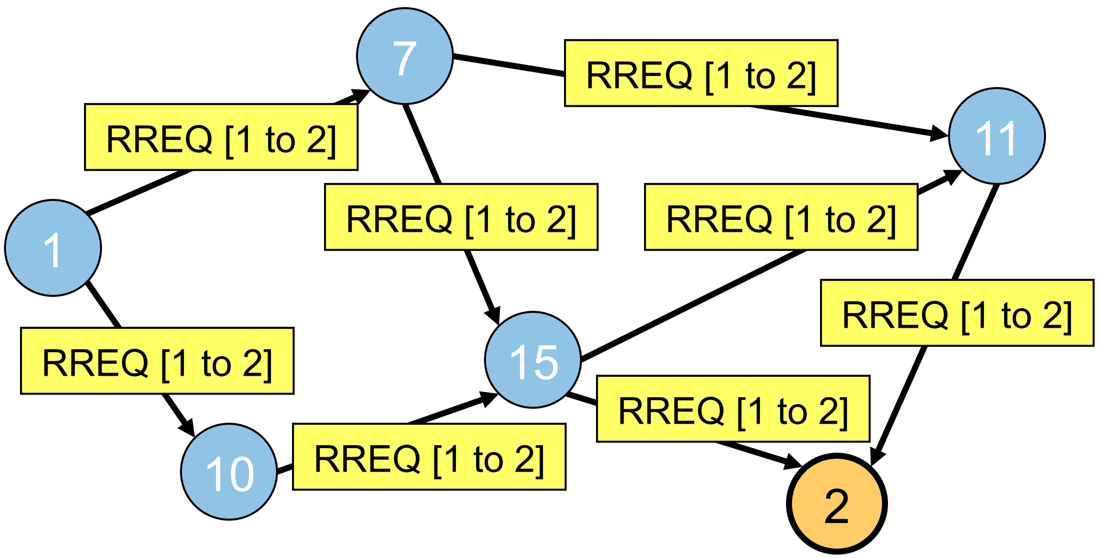
Reactive protocols – AODV

- ***Ad hoc On Demand Distance Vector*** routing (AODV)
  - Very popular routing protocol
  - Essentially same basic idea as DSR for discovery procedure
  - Nodes maintain routing tables instead of source routing
  - Sequence numbers added to handle stale caches
  - Nodes remember from where a packet came and populate routing tables with that information

- Protocol behavior
  - ***RouteRequests*** are flooded though the network
  - Flooding is stopped at the destination or if an intermediate node has a valid route to the destination
  - If a RouteRequest is received multiple times, the duplicates are discarded
  - ***RouteReplies*** are sent back to update the path information

AODV – route setup

- RouteRequests (RREQ) are flooded through the entire network (limited by a TTL describing the maximum network diameter)

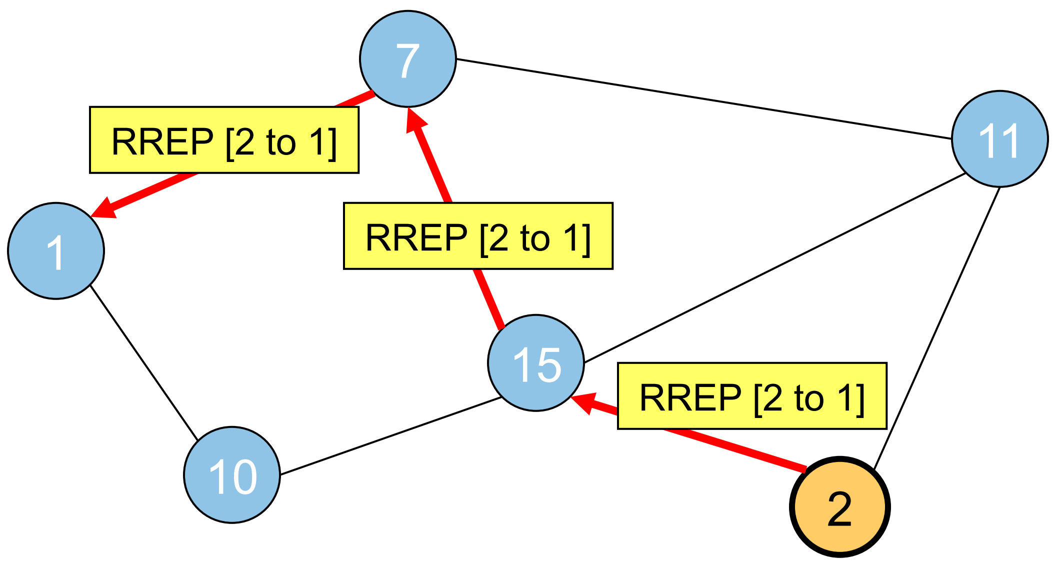


Node	Dest	Next	Dist
7	1	1	1
11	1	7	2
15	1	7	2
…	…	…	…
2	1	15	3

Routing tables after flooding the RREQ [1 to 2]

AODV – route setup

- The RouteReply (REP) is unicasted towards the source

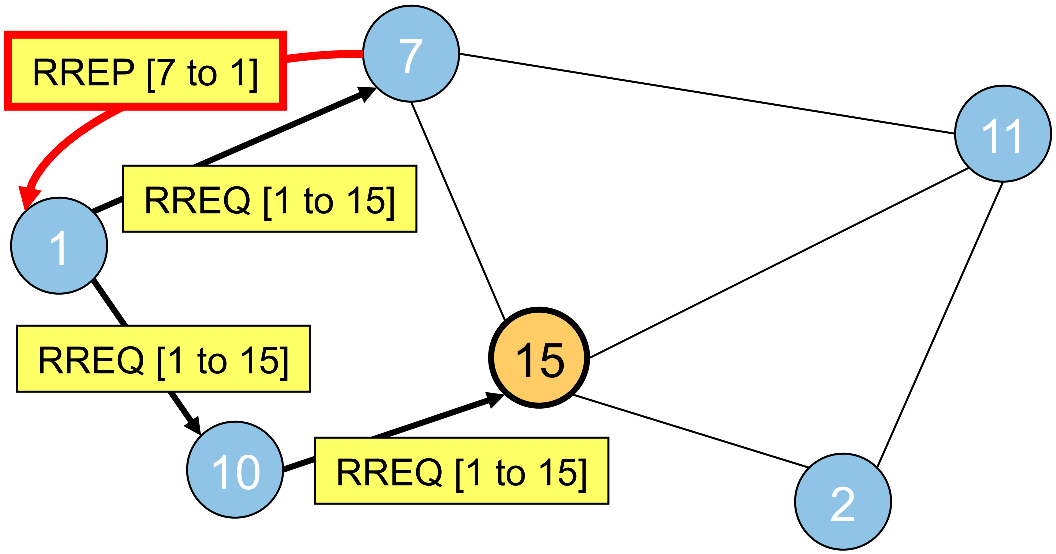


Node	Dest	Next	Dist
7	1	1	1
7	2	15	2
11	1	7	2
15	1	7	2
15	2	2	1
…	…	…	…
2	1	15	3

Routing tables after sending the RREP [2 to 1]

AODV – route setup

- Abbreviated route setup – intermediate nodes are allowed to answer to a RREQ **on behalf** of the final destination

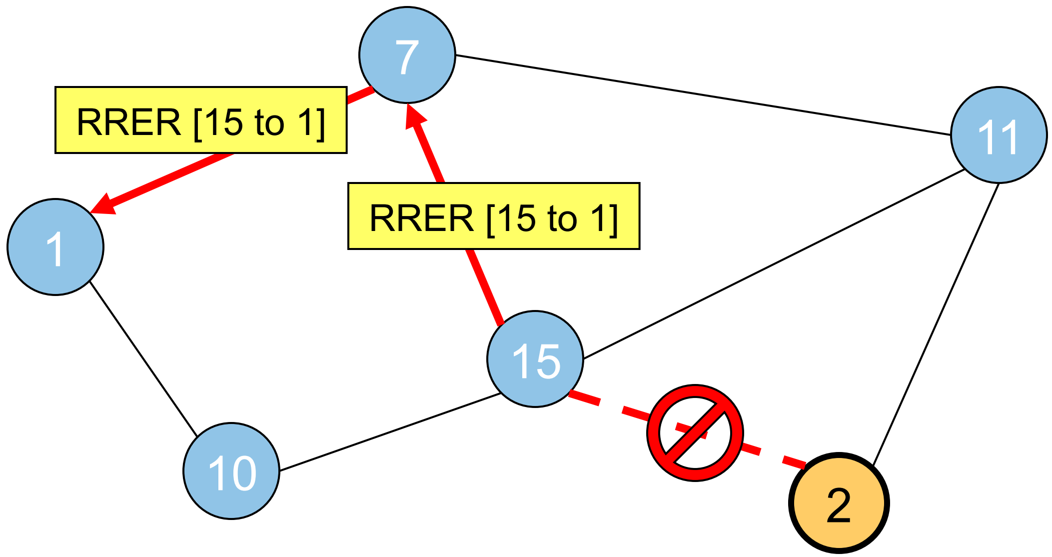


Node	Dest	Next	Dist
7	15	1	1
…	…	…	…

Routing tables before flooding the RREQ [1 to 15]

AODV – route maintenance

- Broken links are announced by RouteError (RERR) messages with the hop count set to infinity



AODV

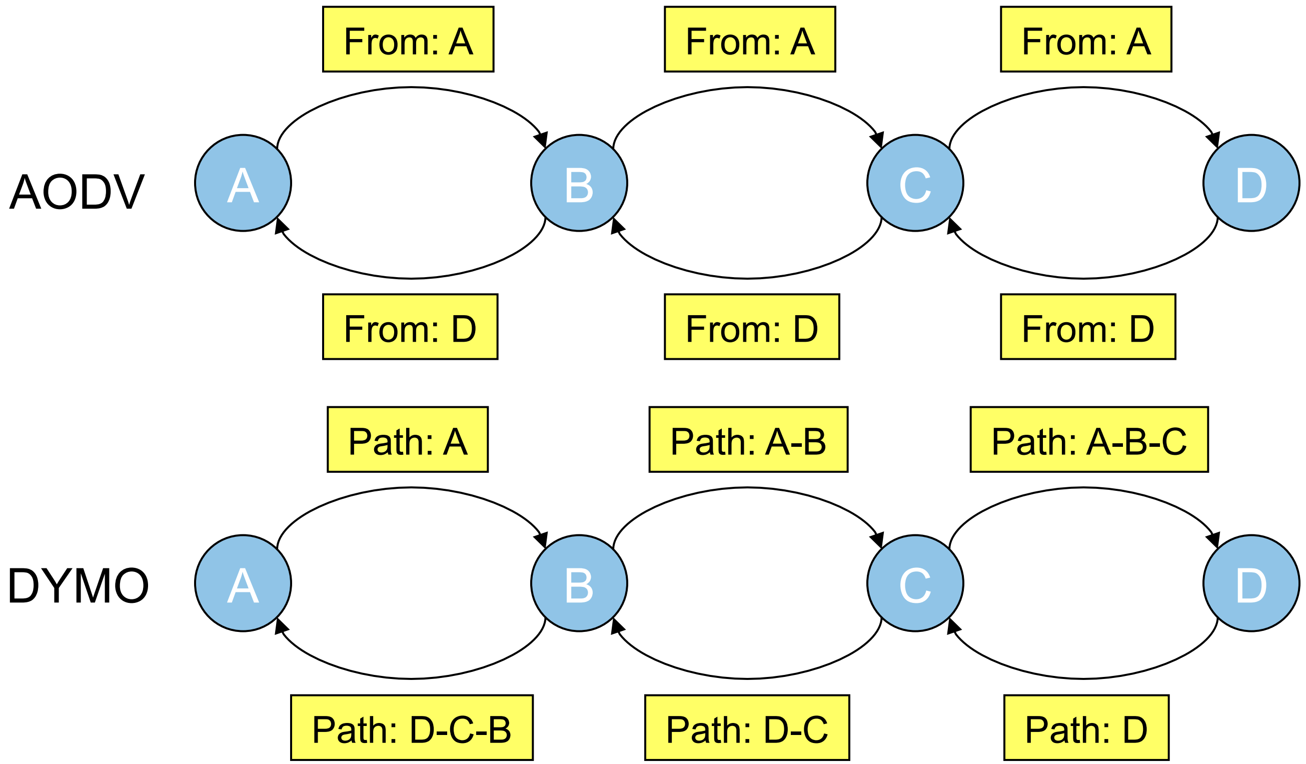
- Advantages
  - On-demand route establishment
  - Destination sequence numbers to find the latest route to the destination
  - Less connection setup delay (compared to DSR)

- Disadvantages
  - Intermediate nodes can lead to inconsistent routes if the source sequence number is very old and the intermediate nodes have higher but not the latest destination sequence number
  - Control overhead due to multiple RouteReply packets in response to a single RouteRequest
  - Periodic beaconing leads to unnecessary bandwidth consumption

Reactive protocols – DYMO

Technische Universität Berlin

- **Dynamic MANET On Demand (DYMO)**
  - Successor of AODV
  - Reduced overhead in route setup and route maintenance



Telecommunication Networks Group

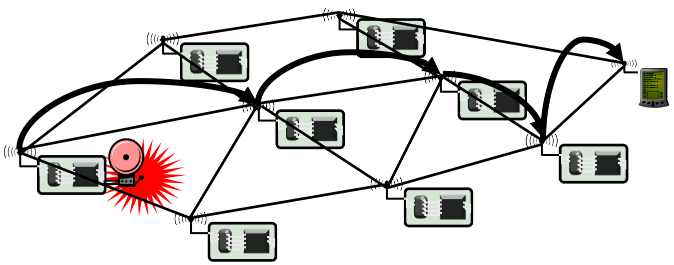
Geo Routing

Geographic Routing

- Routing tables contain information to which next hop a packet should be forwarded
  - Explicitly constructed
- Alternative: Implicitly *infer* this information from physical placement of nodes
  - Position of current node, current neighbors, destination known – send to a neighbor in the right direction as next hop
  - ***Geographic routing***
- Options
  - Send to any node in a given area – ***geocasting***
  - Use position information to aid in routing – ***position-based routing***
    - Might need a ***location service*** to map node ID to node position

Basics of Position-based Routing

- "Most forward progress within range r" strategy
  - Send to that neighbor that realizes the most forward progress towards destination
  - NOT: farthest away from sender!
  - Also known as **greedy routing**

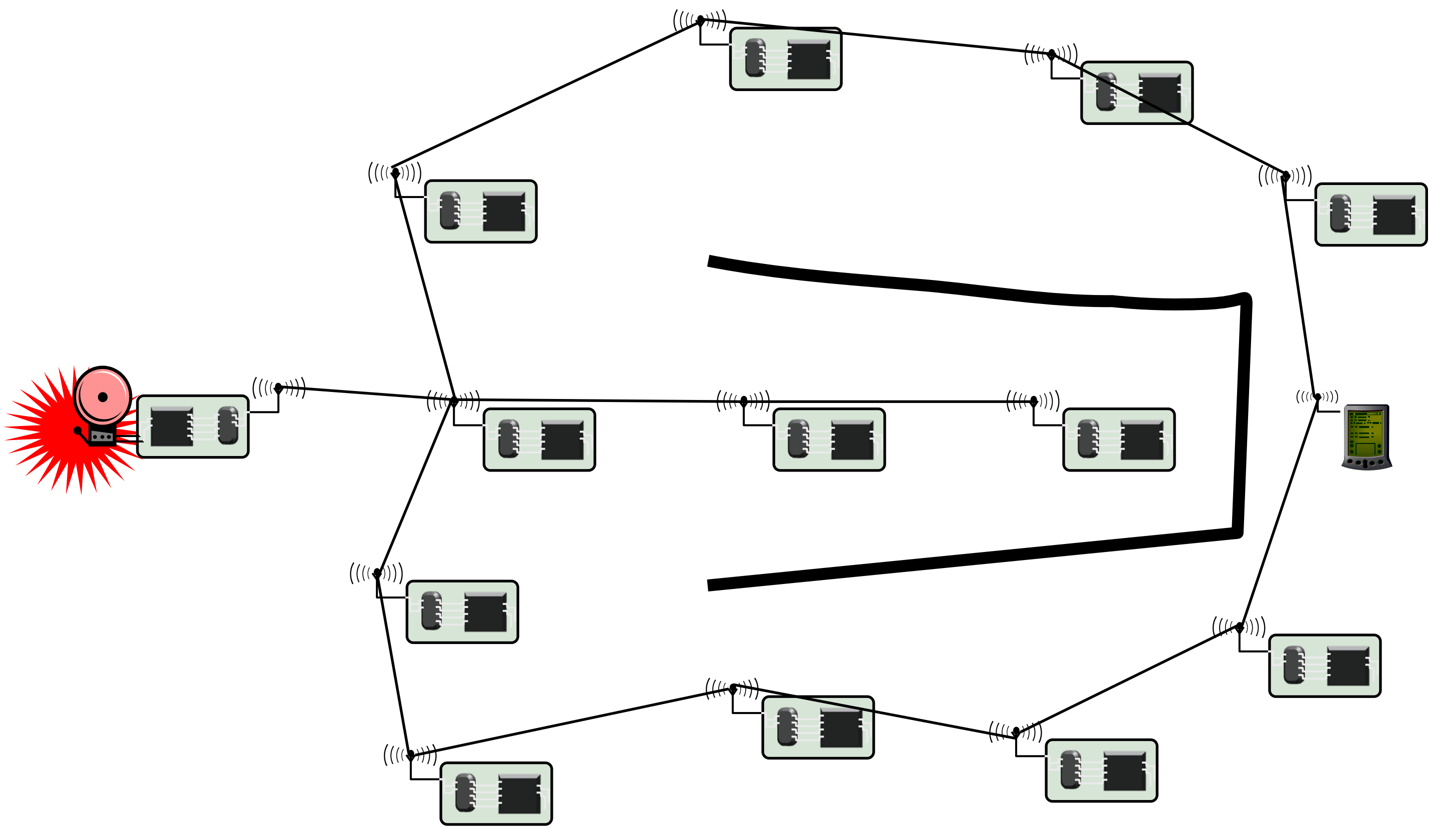


- Nearest node with (any) forward progress
  - Idea: Minimize transmission power

- Directional routing
  - Choose next hop that is angularly closest to destination
  - Choose next hop that is closest to the connecting line to destination
  - Problem: Might result in loops!

Problem: Dead Ends

- Simple strategies might send a packet into a dead end

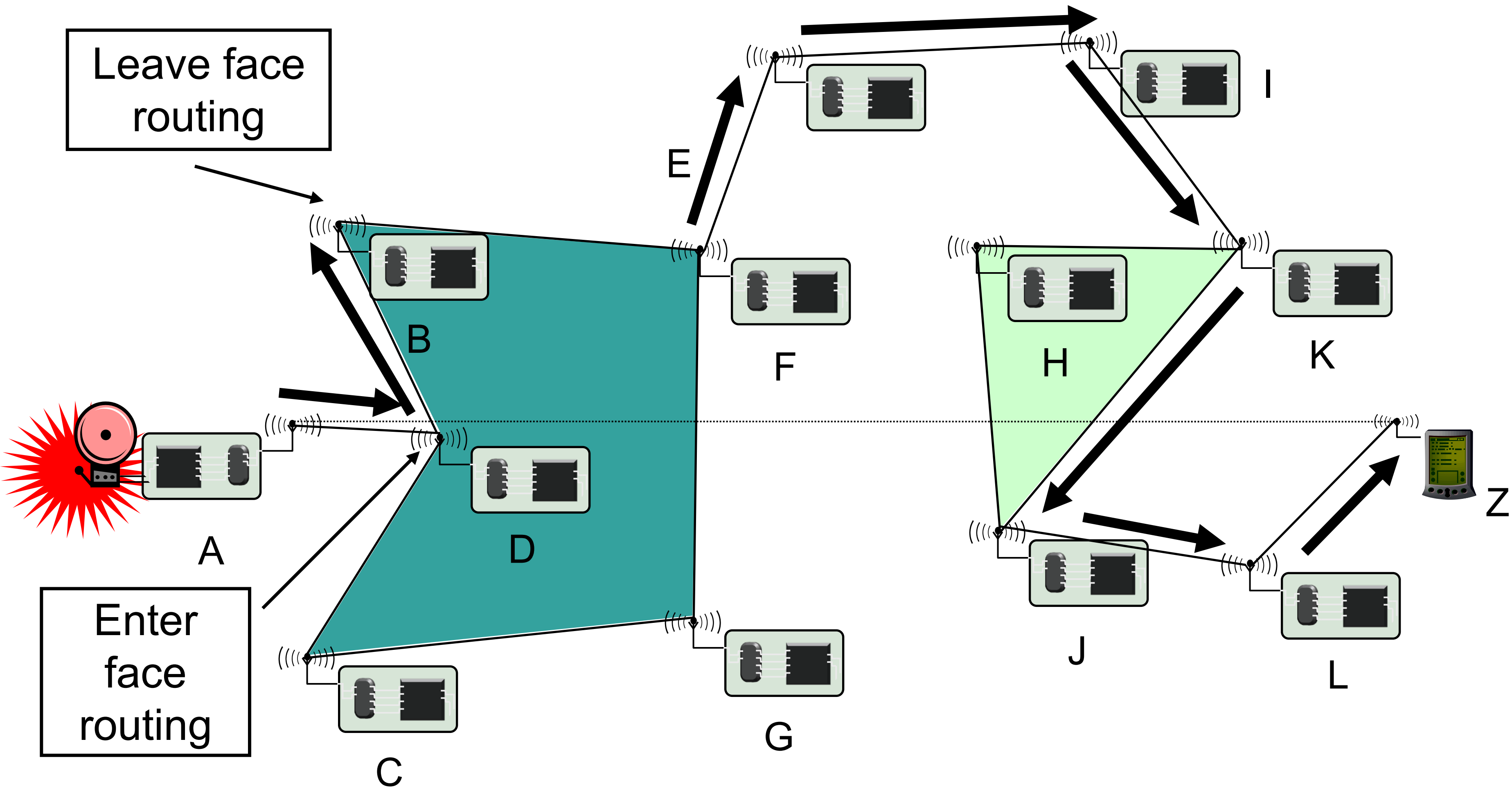


Right Hand Rule to Leave Dead Ends – GPSR

- Basic idea to get out of a dead end: Put right hand to the wall, follow the wall
  - Does not work if on some inner wall – will walk in circles
  - Need some additional rules to detect such circles
- **Greedy Perimeter Stateless Routing (GPSR)**
  - Earlier versions: Compass Routing II, face-2 routing
  - Use greedy, "most forward progress" routing as long as possible
  - If no progress possible: Switch to "face" routing
    - Face: largest possible region of the plane that is not cut by any edge of the graph; can be exterior or interior
    - Send packet around the face using right-hand rule
    - Use position where face was entered and destination position to determine when face can be left again, switch back to greedy routing
  - Requires: planar graph! (topology control can ensure that)

GPSR – Example

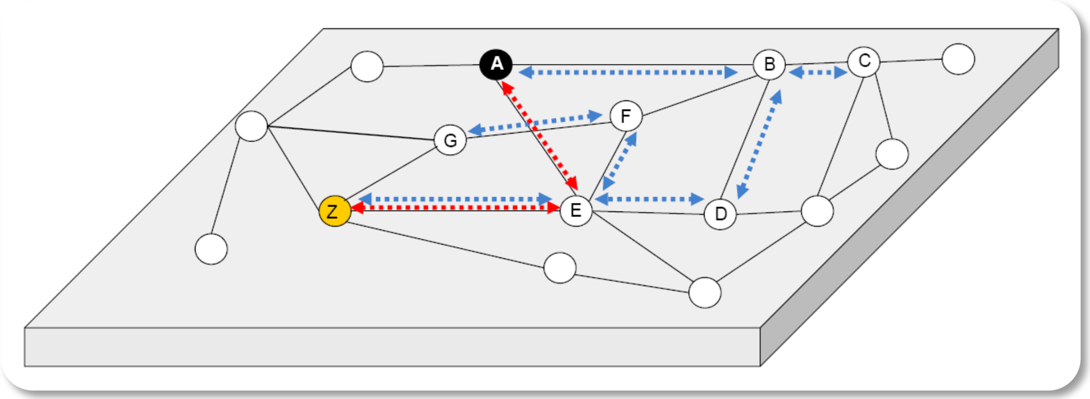
- Route packet from node A to node Z

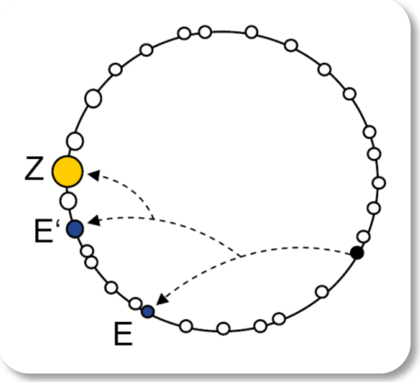


Virtual coordinate based routing

Using virtual coordinates in WSNs

- **Virtual ring routing (VRR)**
  - Combined overlay + underlay routing
  - Virtual addressing
  - Small routing tables

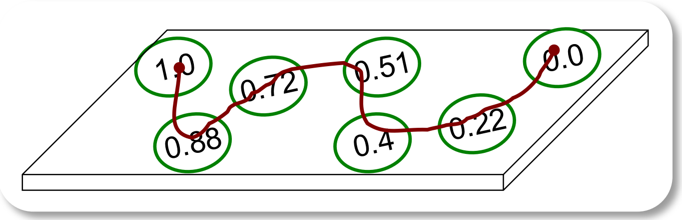




- Problems
  - Does not ensure shortest path routing
  - Problems with system dynamics, e.g. frequent node failures
  - Overlay addressing does not take the physical network structure into account

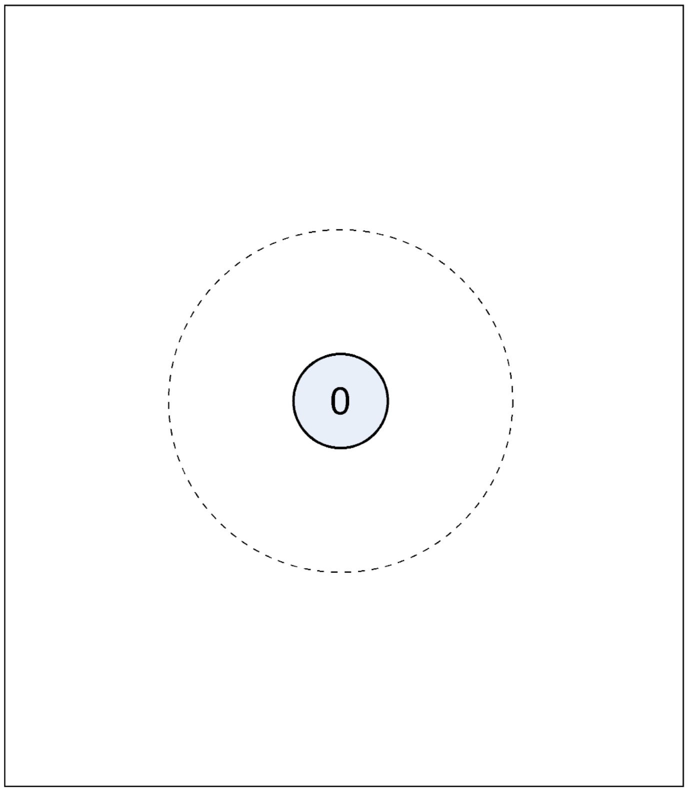
VCP – Virtual Cord Protocol

- **Virtual Cord Protocol (VCP)**
- Cord setup
  - Assignment of virtual coordinates (or positions)
  - Initial start node “S” and the range “[S, E]” are pre-defined
  - Local “hello”s are used to exchange neighborhood information
- Routing
  - Greedy along the cord
  - Exploiting neighborhood information



- Data storage
  - Using application-specific hash functions to uniformly distribute data over the cord
  - Replication either on the cord or within the local neighborhood

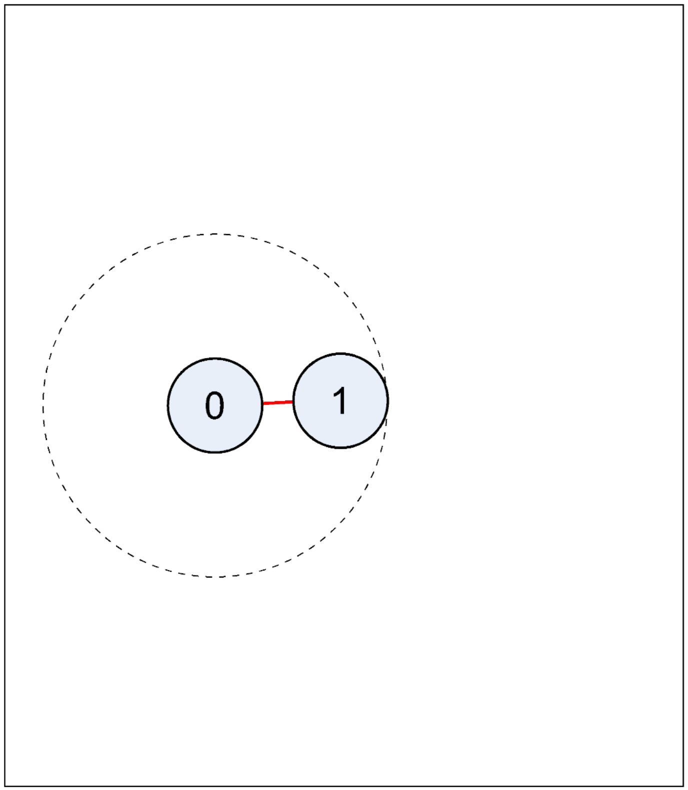
VCP Join



Requirements

- One node must be pre-programmed to initiate the process, i.e. its virtual address is $S = 0.0$
- All nodes periodically (every T_h) broadcast hello messages to maintain neighborhood information and to join new nodes

VCP Join

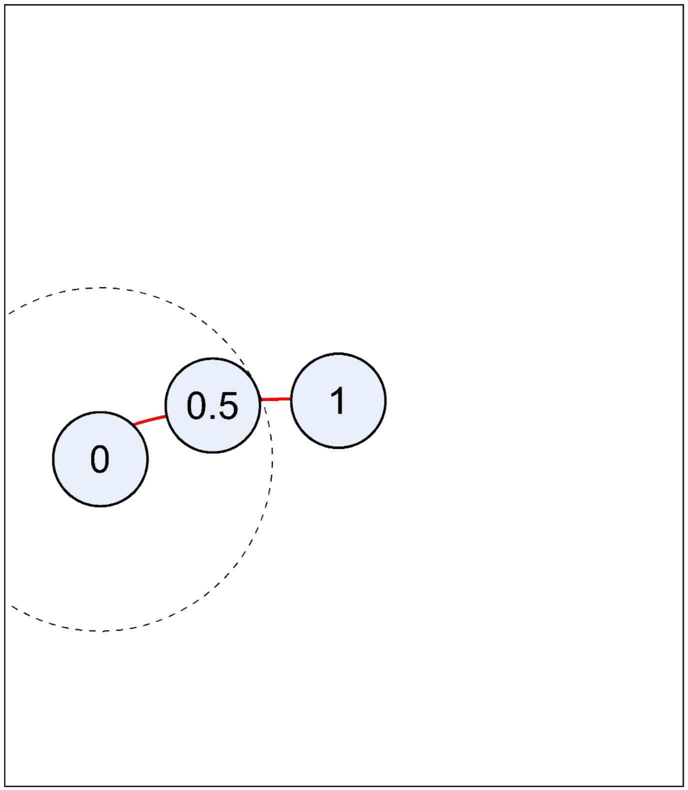


Algorithm 2 SetMyPosition()

Require: Neighbor information stored in set N

for $\forall N_i \in N$ **do**
  if Position$(N_i) == S)$ **then**
    if Successor$(N_i) < S)$ **then**
      $P_{temp} \leftarrow E)$
    else if Successor$(N_i) == E$ **then**
      $P_{temp} \leftarrow (S + E)/2$
    else
      $P_{temp} \leftarrow$ Successor$(N_i) - I \times ($Successor$(N_i) -$ Position$(N_i))$
    end if
    SendNewPositionToNeighbor(N_i, P_{temp})
  else if Position$(N_i) == E$ **then**

VCP Join

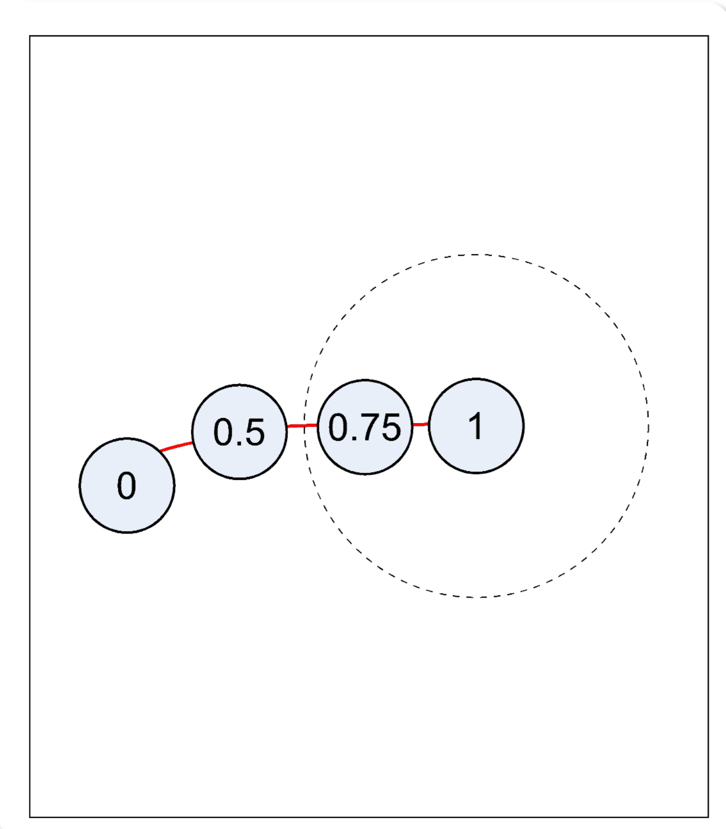


Algorithm 2 SetMyPosition()

Require: Neighbor information stored in set N

for $\forall N_i \in N$ **do**
    if Position$(N_i) == S)$ **then**
        if Successor$(N_i) < S)$ **then**
            $P_{temp} \leftarrow E)$
        else if Successor$(N_i) == E$ **then**
            $P_{temp} \leftarrow (S + E)/2$
        else
            $P_{temp} \leftarrow$ Successor$(N_i) - I \times ($Successor$(N_i) -$ Position$(N_i))$
        end if
        SendNewPositionToNeighbor(N_i, P_{temp})
    else if Position$(N_i) == E$ **then**

VCP Join



else if Position(N_i) $==$ E **then**

  if Successor(N_i) $==$ S **then**

    $P_{temp} \leftarrow (S + E)/2$

  else

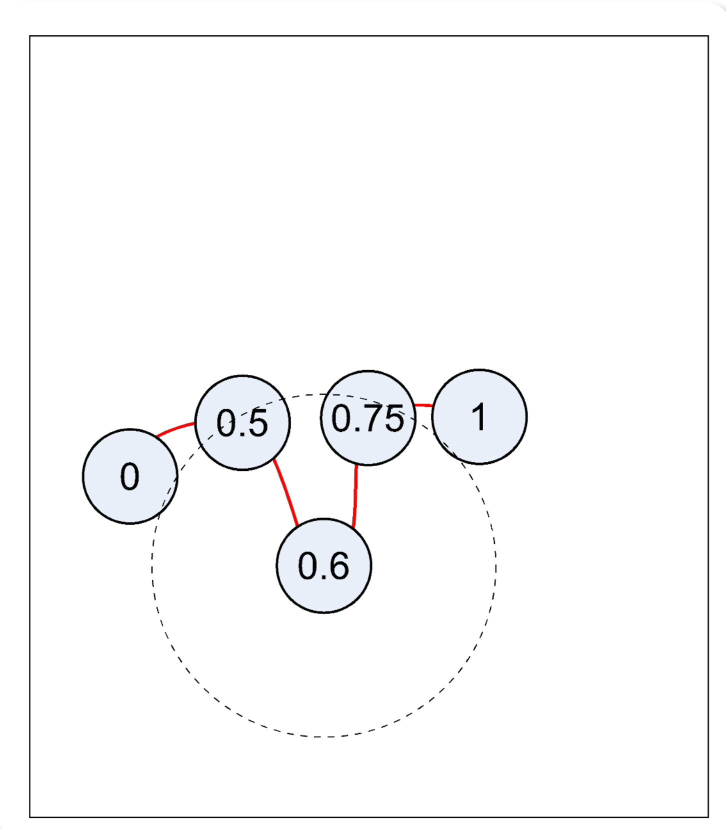
    $P_{temp} \leftarrow \text{Predecessor}(N_i) - I \times (\text{Predecessor}(N_i) - \text{Position}(N_i))$

  end if

  SendNewPositionToNeighbor(N_i, P_{temp})

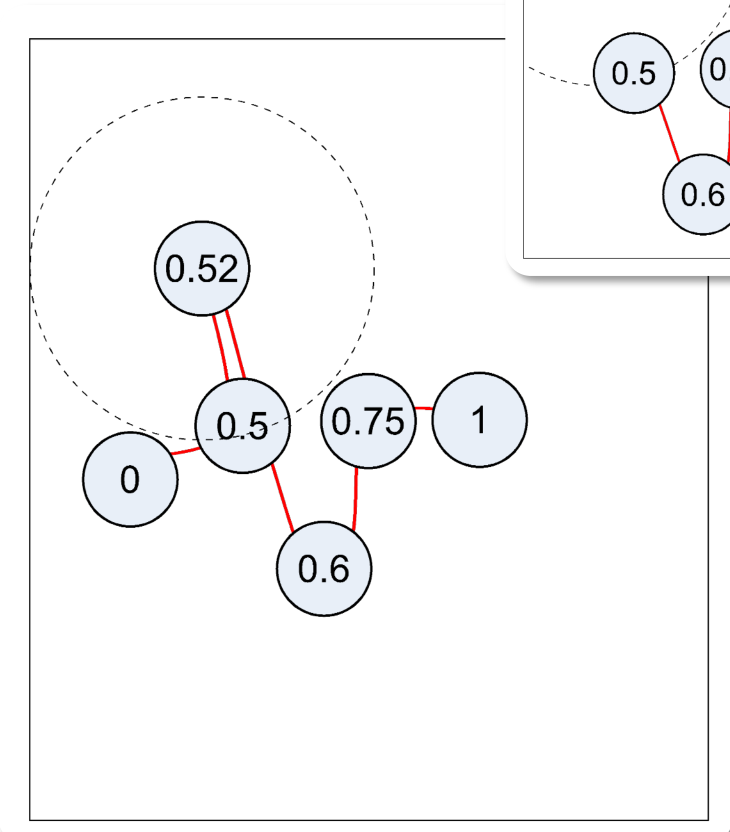
else

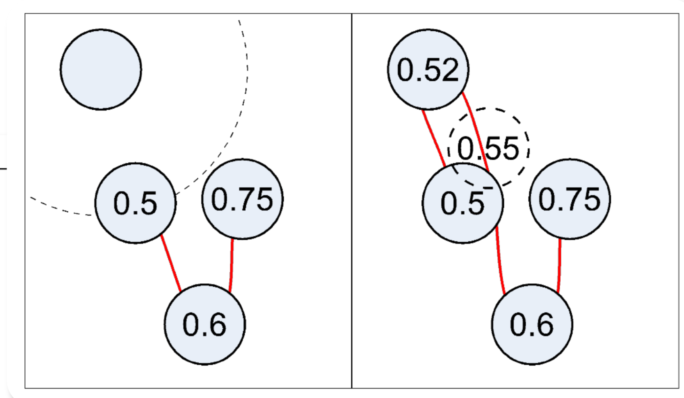
VCP Join




```
    else
        found ← 0
        for ∀N_j ∈ N : i ≠ j do
            if Predecessor(N_i) == Position(N_j) then
                found ← 1
                P_temp ← (Position(N_i) + Position(N_j))/2
                temporally store positions of N_i and N_j
                SendBlockReq(N_j, P_temp)
            end if
        end for
        if found == 0 then
            if (Time() − OldVTime) > T_vps then
                OldVtime ← Time()
                temporally store position of N_i
                SendCreatVirtualNode(N_i)
            end if
        end if
    end if
end for
```

VCP Join



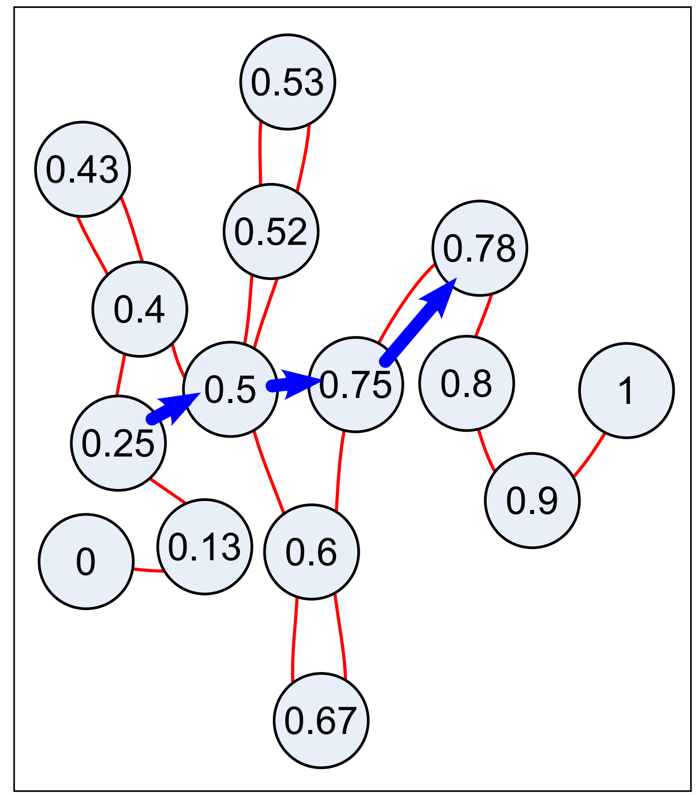



```
$i \neq j$ do
r($N_i$) == Position($N_j$) then
(Position($N_i$) + Position($N_j$))/2
        temporally store positions of $N_i$ and $N_j$
        SendBlockReq($N_j$, $P_{temp}$)
      end if
    end for
    if found == 0 then
      if (Time() − OldVTime) > $T_{vps}$ then
        OldVtime ← Time()
        temporally store position of $N_i$
        SendCreatVirtualNode($N_i$)
      end if
    end if
  end if
end for
```

VCP Routing

- Greedy forwarding along the Cord
  - Always guarantees reachability for any destination
- Speedup by exploiting local short-cuts

Node 0.25	Neighbors
Successor 0.4	0.0
Predecessor 0.13	0.13
	0.4
	0.5 (0.55)



Parameters regulating the Join

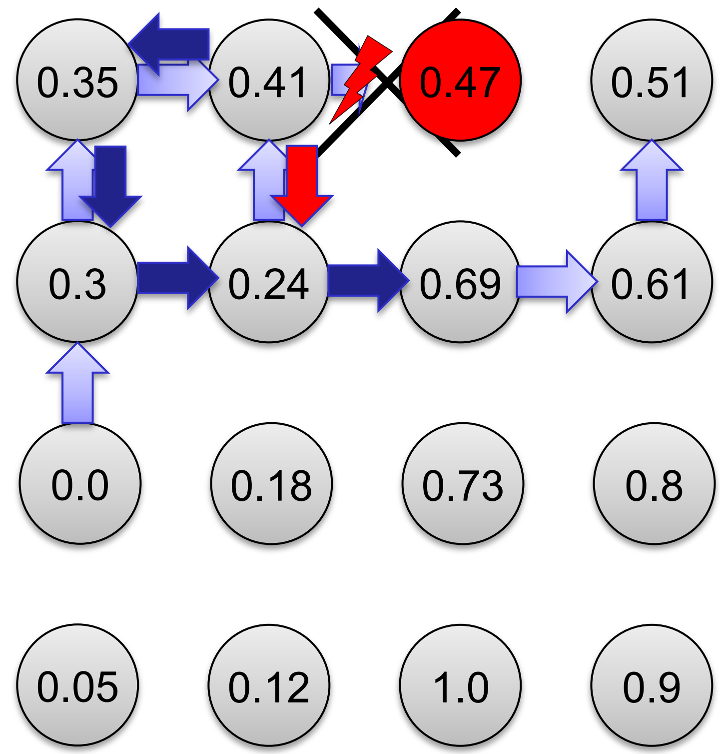
Parameter, Default value	Description
Start $S = 0.0$	Lowest position on the cord
End $E = 1.0$	Highest position on the cord
Position $P = -1.0$	Current position on the cord (uninitialized: $P = -1.0$)
HelloPeriod $T_h = 1s$	Interval for the hello messages
SetPosDelay $T_{ps} = 1s$	Time before re-requesting a new position
SetVPosDelay $T_{vps} = 1s$	Time before requesting a virtual position
BlockDelay $T_b = 1s$	Blocking period to prevent assigning the same position twice
Interval $I = 0.1$	Interval for calculating regular cord positions
VirtInterval $I_v = 0.9$	Interval for calculating virtual node positions

Data Management

- **Data storage (or “push”)**
  - Using an application-dependent hash function to evenly distribute the workload among all the nodes on the Cord
  - Storage at the node with a virtual coordinate closest to the hash value
- **Data retrieval (or “pull”)**
  - Identification of the location using the same hash function
- **Replication**
  - Neighbors on the Cord: simplified replication mechanism
  - Within the vicinity of the node: improved reliability

Failure Management

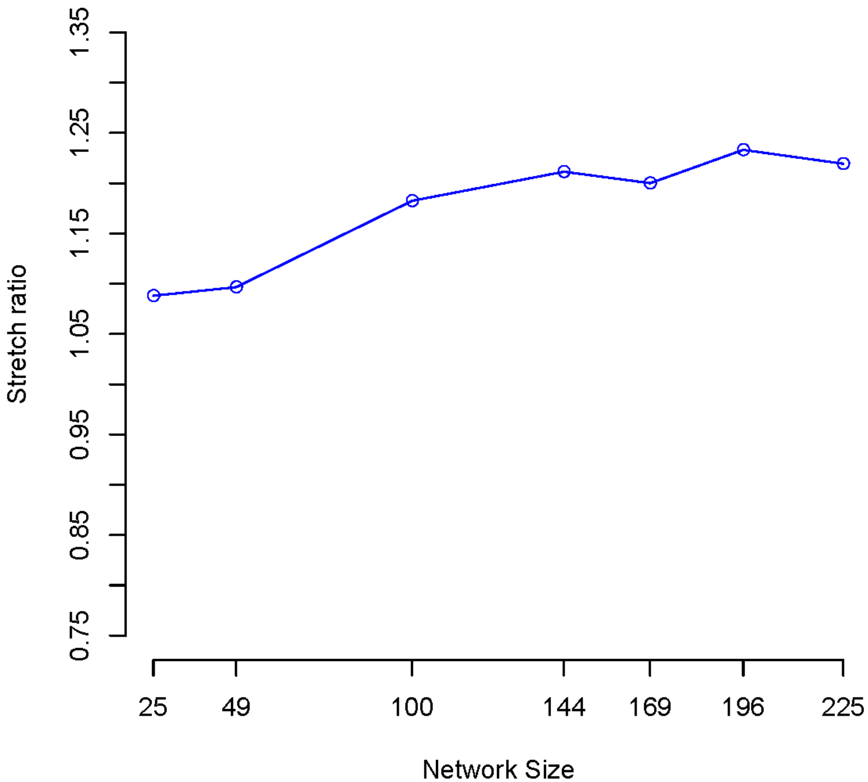
- Routing from 0.0 to 0.51
- At 0.41, a **route error** is discovered
- Node 0.41 knows that the destination 0.51 was a direct neighbor of 0.47, thus, it generates a **NPB (no path back) message**
- The NPB message travels backwards until another possible path to 0.51 is available
- Node 0.3 selects the second best path (via node 0.24) and forwards the NPB message
- Node 0.24 forwards the message (again) to 0.41, which detects that there is no alternative path to 0.51 and generates a **NP (no path) message**
- The NP message arrives at 0.24, which in turn forwards it as a NPB message along its second choice path to 0.69
- Node 0.69 has an available path to 0.51, transforms the NP message back into a normal data message and forwards it via 0.61 to 0.51



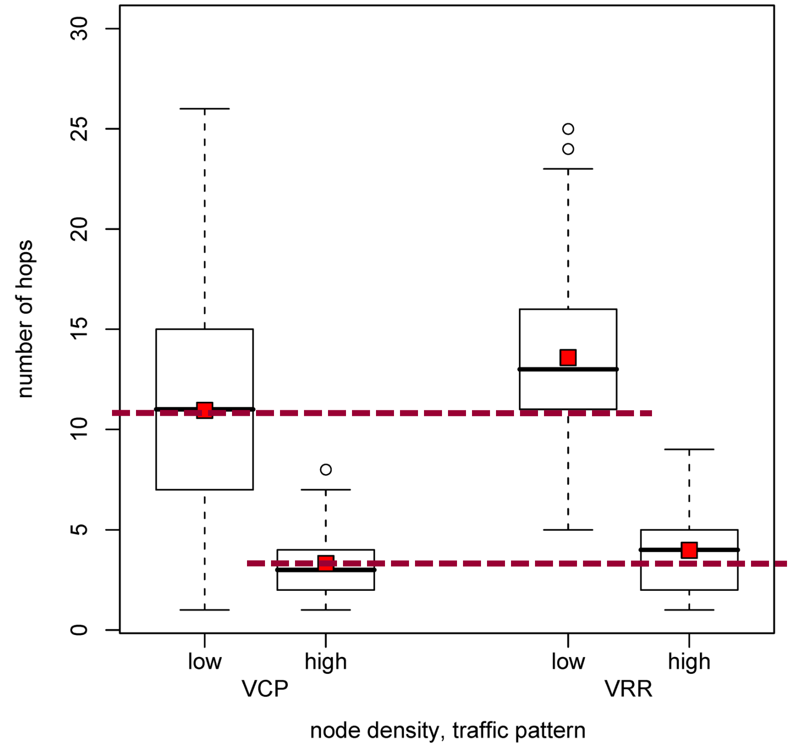
Path Length

- Grid scenario, no node failures

Stretch ratio: VCP path vs. shortest path



Path length: VCP vs. VRR



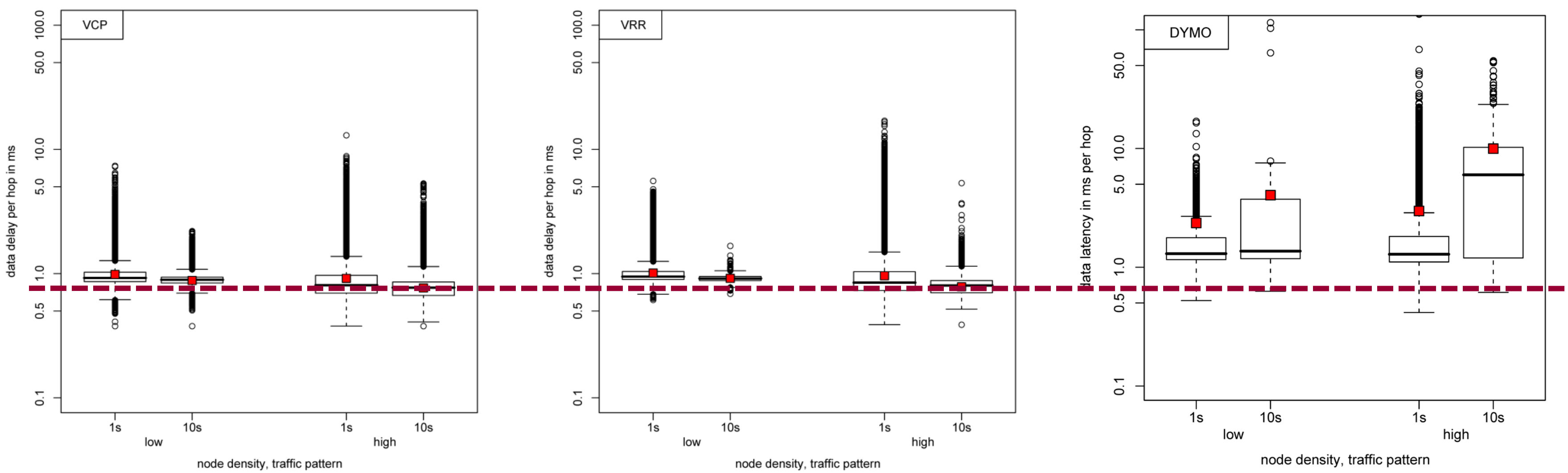
Technische Universität Berlin

Telecommunication Networks Group

Transmission Latency

- Grid scenario, no node failures

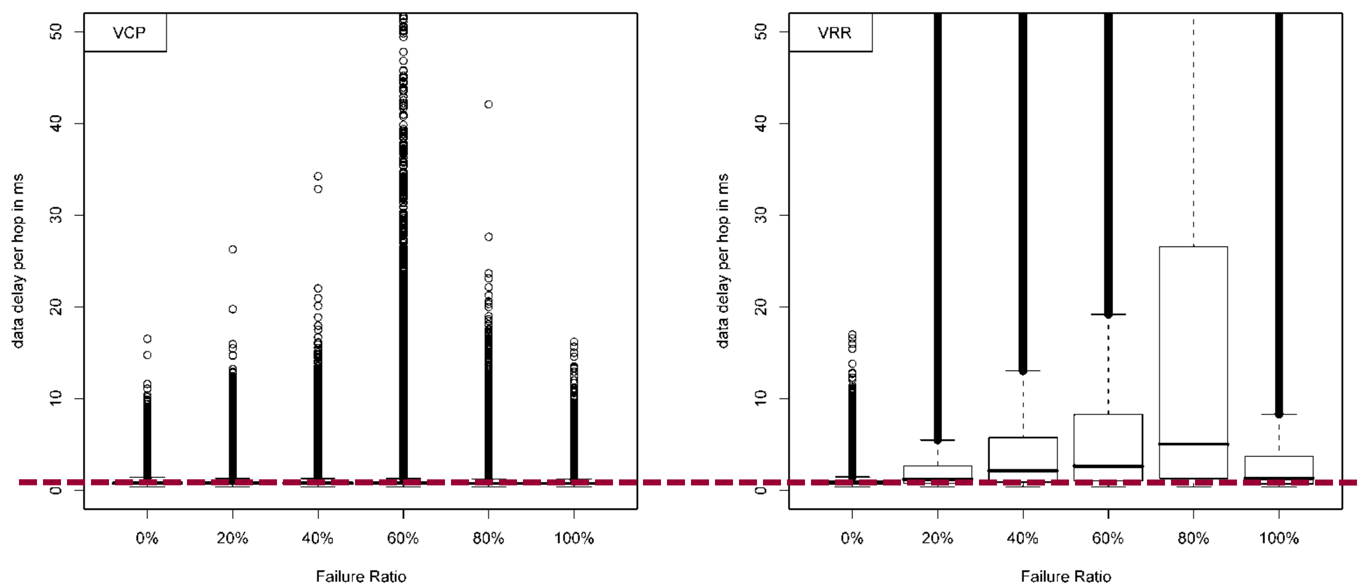
Per-hop-delay in ms



Transmission Latency

- Grid scenario, CBR, frequent node failures

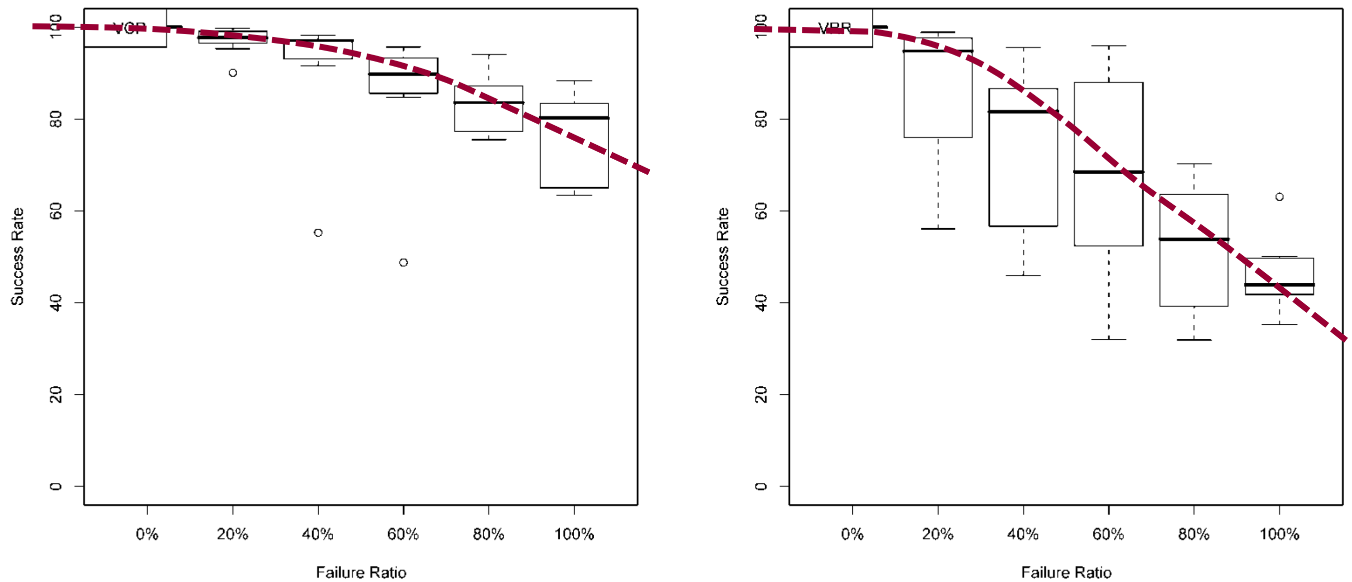
Per-hop-delay in ms



Success Rate

- Grid scenario, CBR, frequent node failures

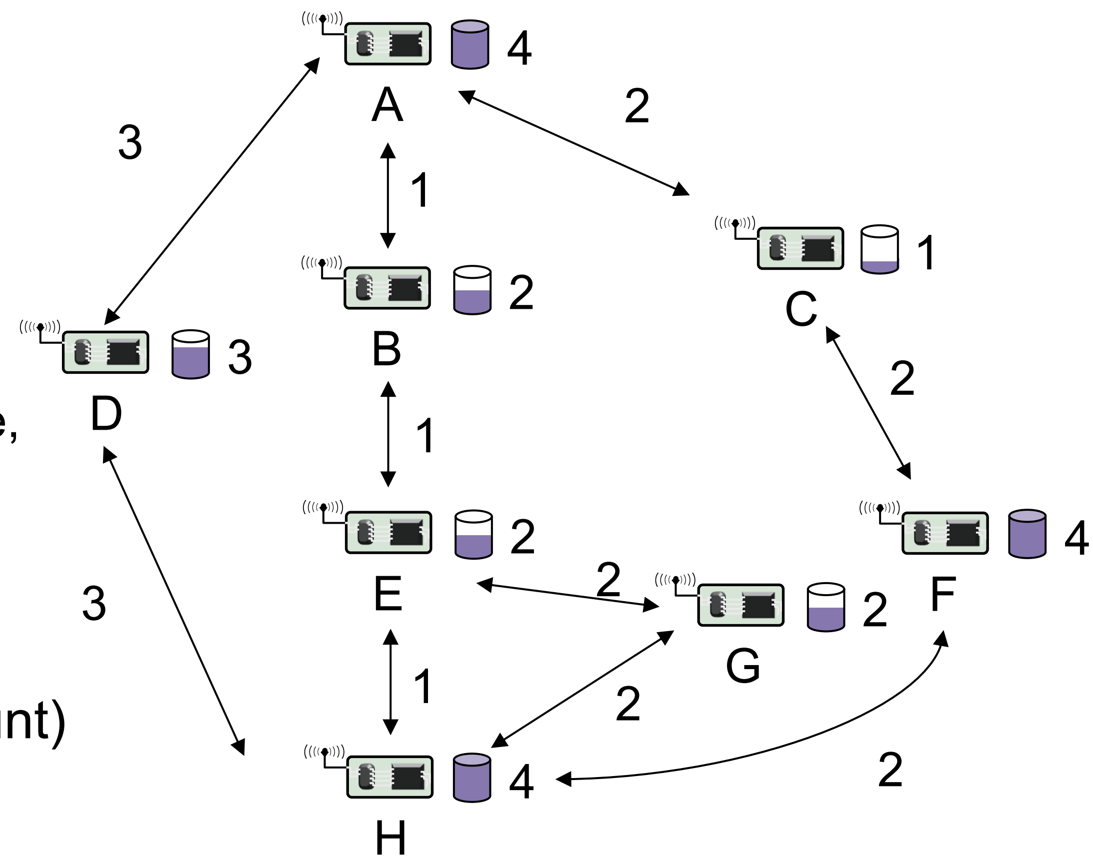
Ratio of successful transmissions



Complementary Routing Approaches

Energy-Aware Routing Protocols

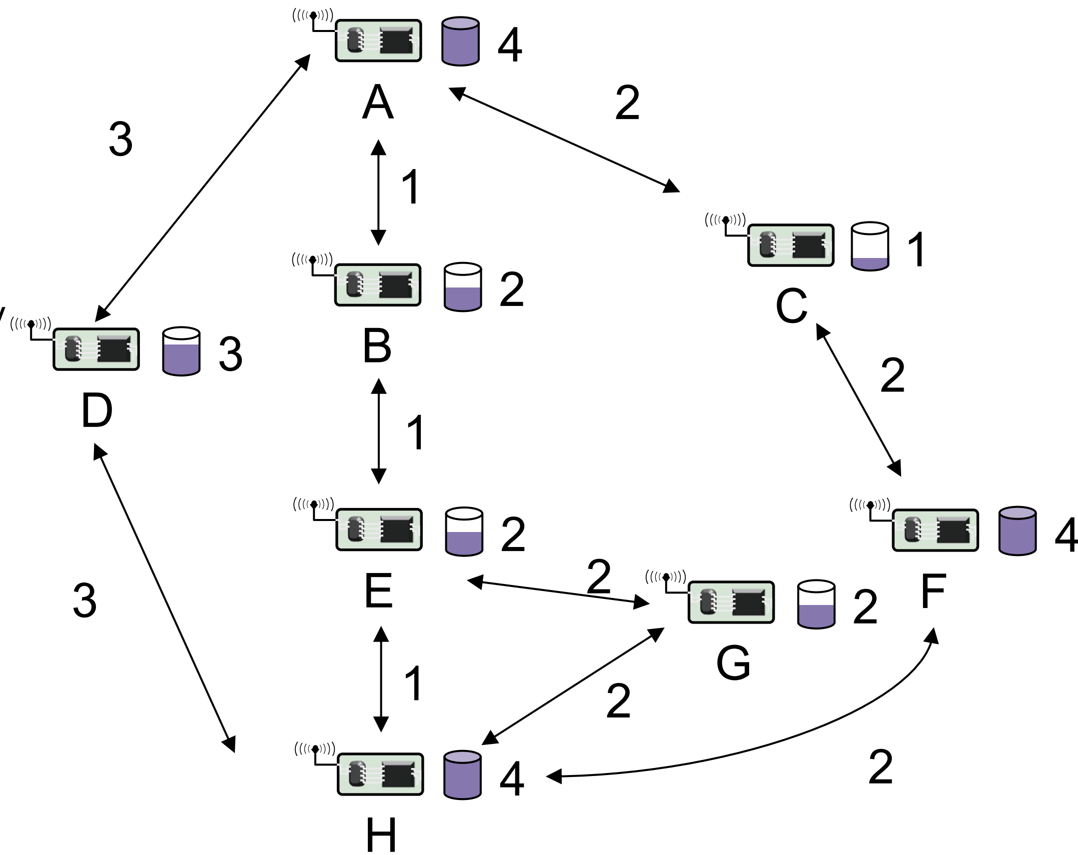
- Particularly interesting performance metric: Energy efficiency
- Goals
  - Minimize energy/bit
    - Example: A-B-E-H
  - Maximize network lifetime
    - Time until first node failure, loss of coverage, partitioning
- Seems trivial – use proper link/path metrics (not hop count) and standard routing



Example: Send data from node A to node H

Basic Options for Path Metrics

- Maximum total available battery capacity
  - Path metric: Sum of battery levels
  - Example: A-C-F-H
- Minimum battery cost routing
  - Path metric: Sum of reciprocal battery levels
  - Example: A-D-H
- Conditional max-min battery capacity routing
  - Only take battery level into account when below a given level
- Minimize variance in power levels
- Minimum total transmission power



Optimized route stability

- Route-Lifetime Assessment Based Routing (RABR)
  - Frequent link failures due to node mobility → reduced throughput
  - Or, improved stability of routes → reduced overhead for retransmissions
  - Based on a new measure → ***link affinity*** a_{nm} between nodes *n*, *m* (please note: a_{nm} is a time!)

$S_{nm(current)}$	– current signal strength
S_{thresh}	– given threshold for the signal strength
$\delta S_{nm(avg)}$	– average of the ***rate*** of change of signal strength

$$a_{nm} = \begin{cases} high & \text{if } \delta S_{nm(avg)} > 0 \\ \dfrac{S_{thresh} - S_{nm(current)}}{\delta S_{nm(avg)}} & \text{otherwise} \end{cases}$$

  - Optimized path metric, based on weakest link → ***path affinity*** $p_{x0,\, x1,\, \ldots\, xl}$

$$p_{x0,\, x1, \ldots,\, xl} = \min_{0 \le i < l}(a_{xi,\, xi+1})$$

Optimized route stability

- Dynamic power adjustment for data transmissions
  - based on the link affinity
  - Periodic (τ) exchange of Hello packets with constant power
  - Received signal strength is measured as S_H

  - Derive relative transmit strength for sending during next τ

$$S_{t,t+\tau} = \begin{cases} S_H - (S_H - S_{thresh})\dfrac{\tau}{a} & \text{if moving father and } \tau < a \\ S_H & \text{if moving closer and } \tau > a \\ S_{thresh} & \text{otherwise} \end{cases}$$

  - Calculate adjusted transmission power

$$P_{t,t+\tau} = P_T \frac{S_{thresh}}{S_{t,t+\tau}}$$

Summary (What do I need to know)

- Concepts of ad hoc routing
  - Proactive / reactive
- Protocols
  - DSDV
  - DSR
  - AODV / DYMO
- Geo routing
  - GPSR
- Virtual coordinate based routing
  - VRR
  - VCP
- Complementary routing approaches
  - Energy-aware routing
  - Link stability