Self-Organization in Autonomous Sensor/Actuator Networks [SelfOrg]

Dr.-Ing. Falko Dressler

Computer Networks and Communication Systems
Department of Computer Sciences
University of Erlangen-Nürnberg

http://www7.informatik.uni-erlangen.de/~dressler/
dressler@informatik.uni-erlangen.de

Overview

- **Self-Organization**
  Introduction; system management and control; principles and characteristics; natural self-organization; methods and techniques

- **Networking Aspects: Ad Hoc and Sensor Networks**
  Ad hoc and sensor networks; self-organization in sensor networks; evaluation criteria; medium access control; ad hoc routing; data-centric networking; clustering

- **Coordination and Control: Sensor and Actor Networks**
  Sensor and actor networks; coordination and synchronization; in-network operation and control; task and resource allocation

- **Bio-inspired Networking**
  Swarm intelligence; artificial immune system; cellular signaling pathways

Ad Hoc Routing

- Overview routing protocols
- Proactive and reactive ad hoc routing
- Geographical / energy-aware routing
- Route stability
- Dynamic address allocation

Characteristics of an Ideal Routing Protocol

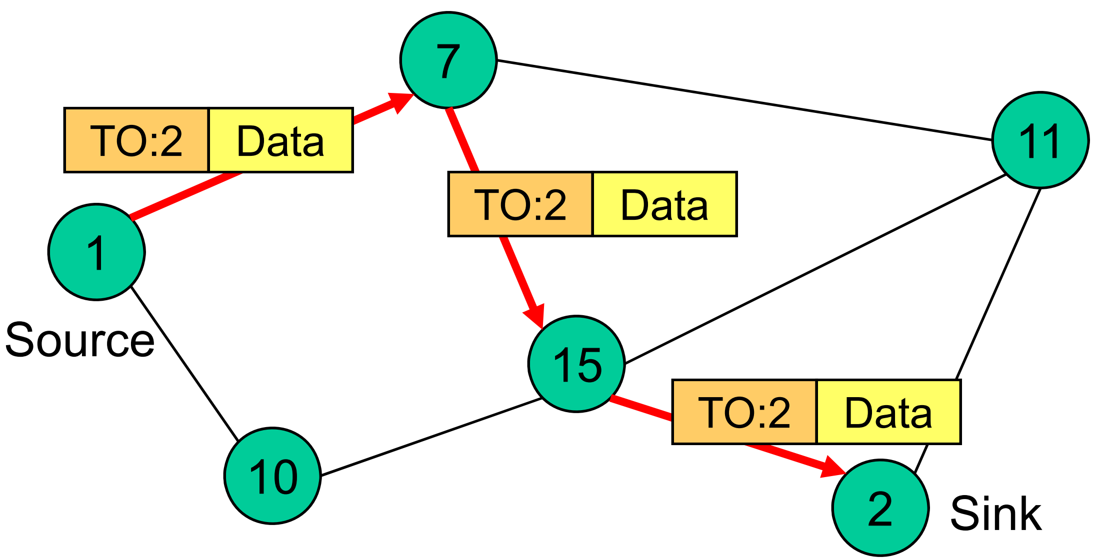
- Requirements
  - Fully distributed (scalability)
  - Adaptive (topology changes)
  - Minimum number of nodes involved for route computation
  - Localized state (reduced global state)
  - Loop-free, free from stale routes
  - Limited number of broadcasts (collision avoidance)
  - Quick and stable convergence
  - 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

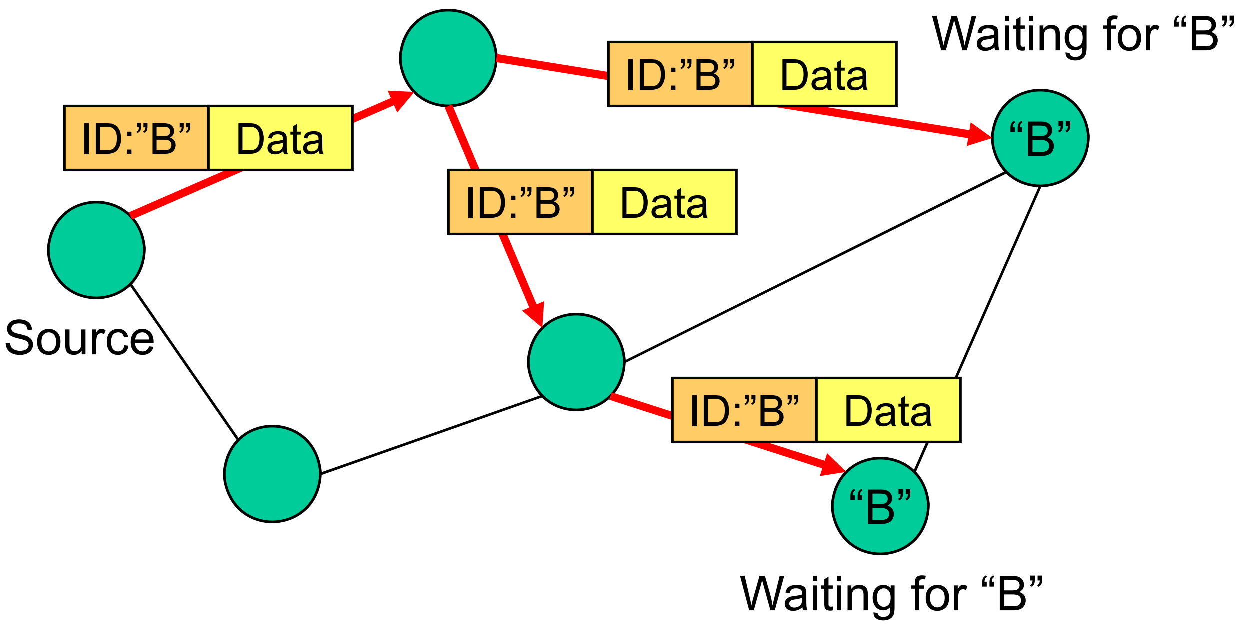
→ Topic of this chapter



Address-based routing vs. data-centric forwarding

- Data-centric forwarding
  - Forwarding of messages to all / some ***appropriate*** nodes
  - Routing decisions according to the “data”, i.e. encoding rules are needed

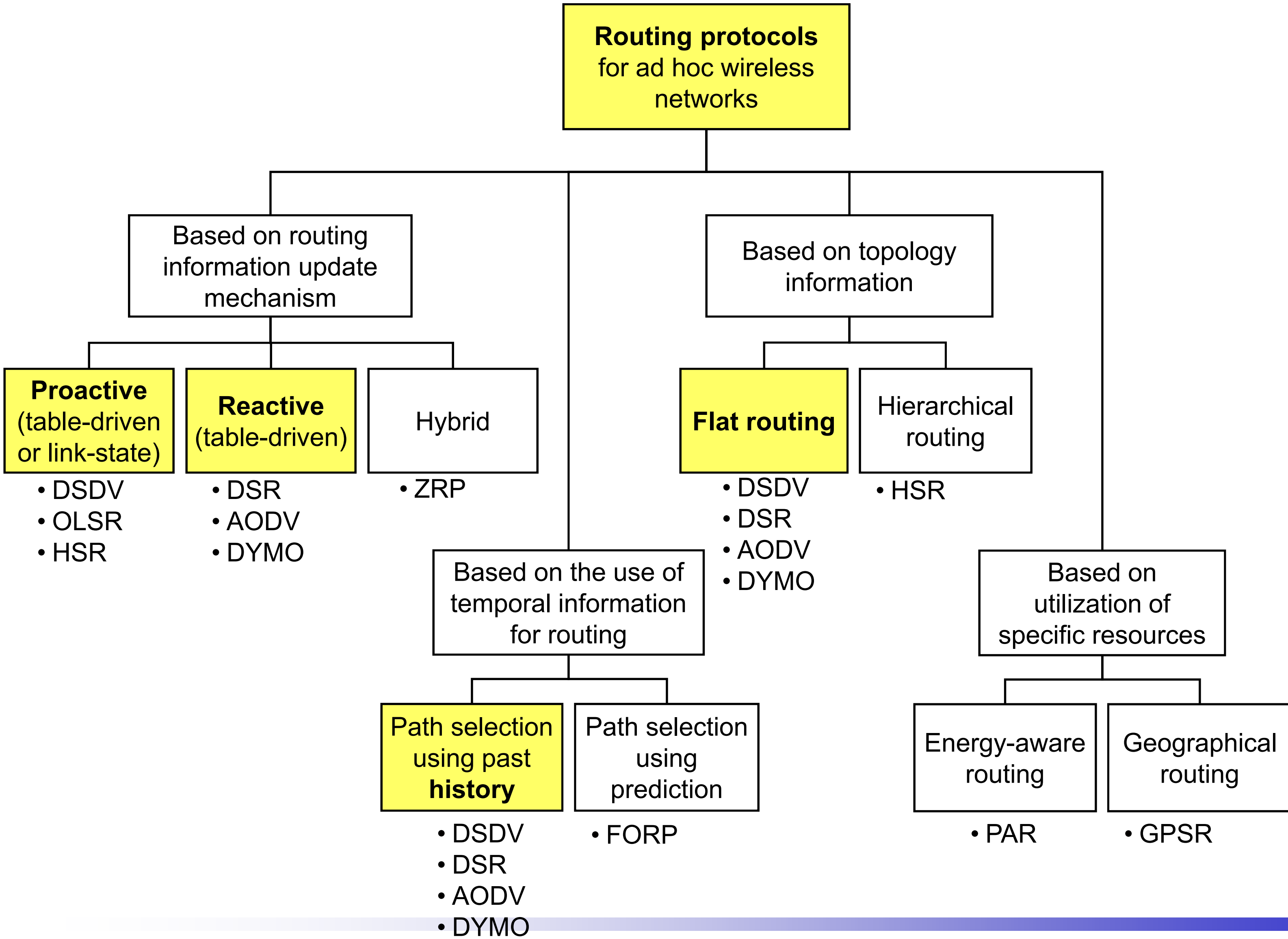
→ Topic of the next chapter



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

Proactive Protocols – DSDV

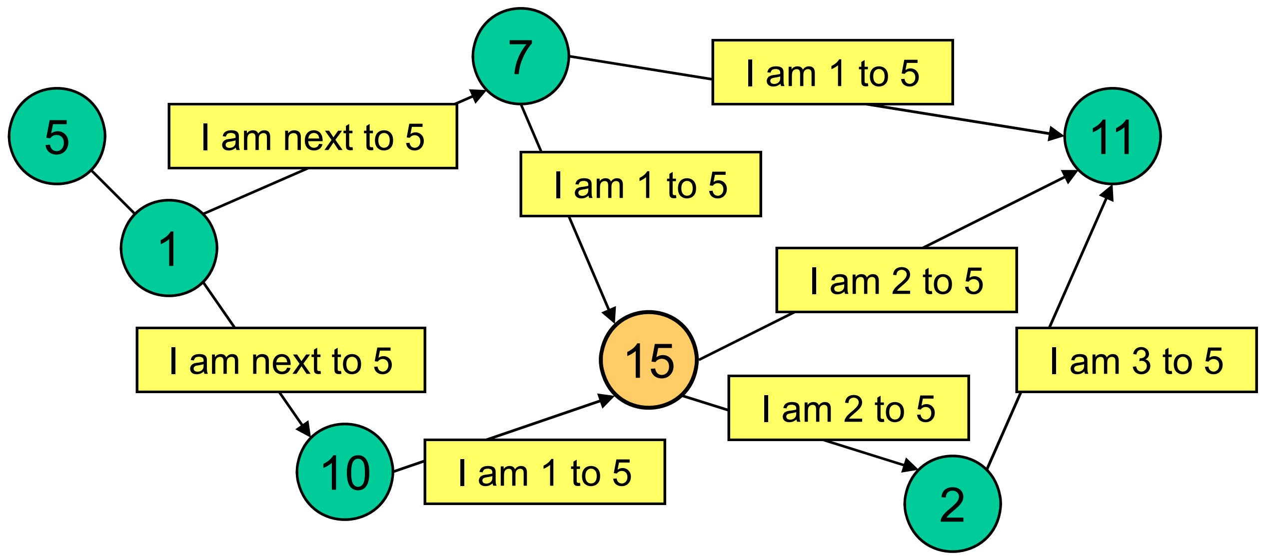
- Idea: Start from a +/- standard routing protocol, adapt it
- Adapted distance vector: ***Destination Sequence Distance Vector (DSDV)***
  - Based on distributed Bellman Ford procedure
  - Add ***aging*** information to route information propagated by distance vector exchanges; helps to avoid routing 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

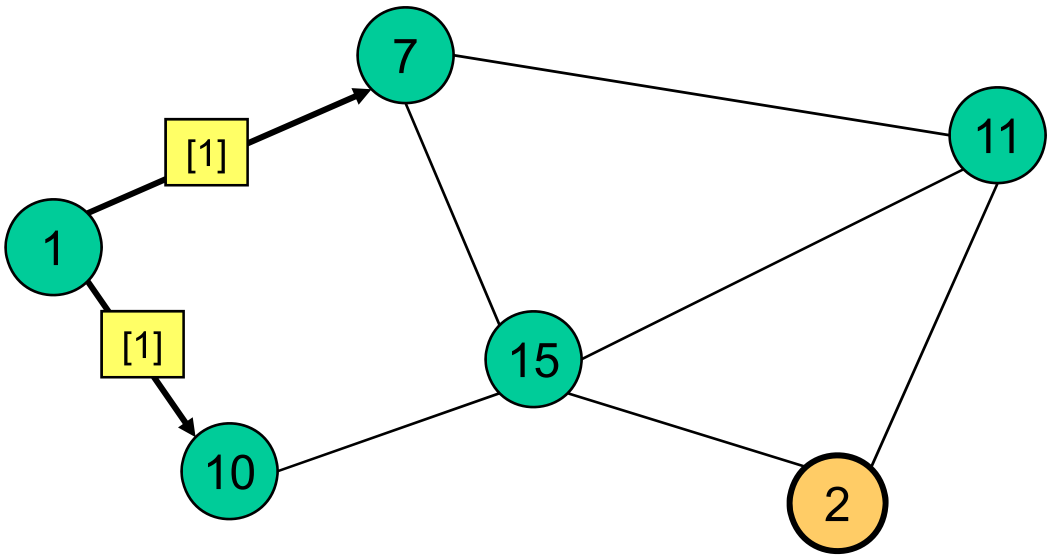
    → could result in stale routing information

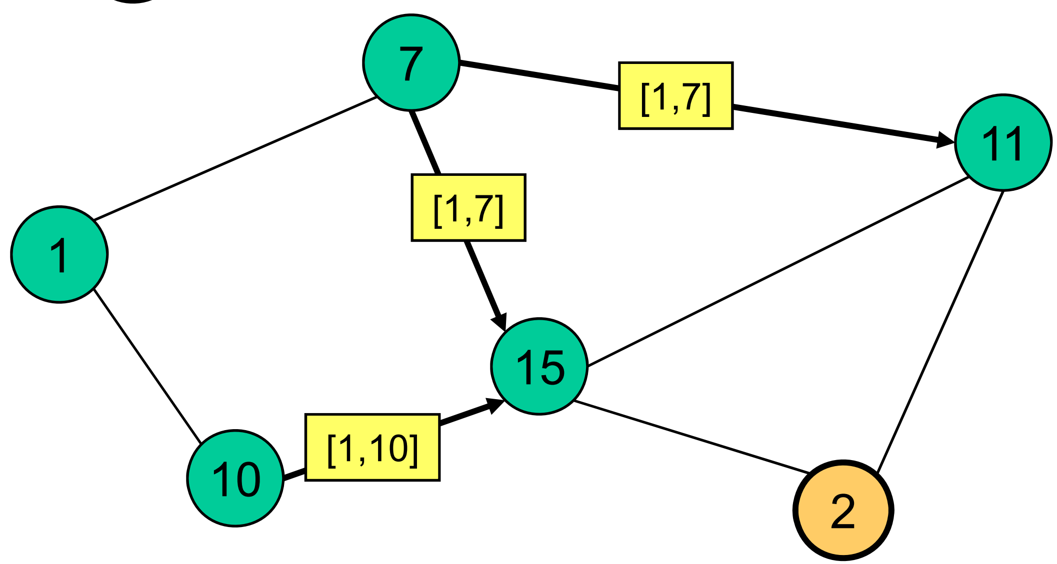
Reactive Protocols – DSR

- 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

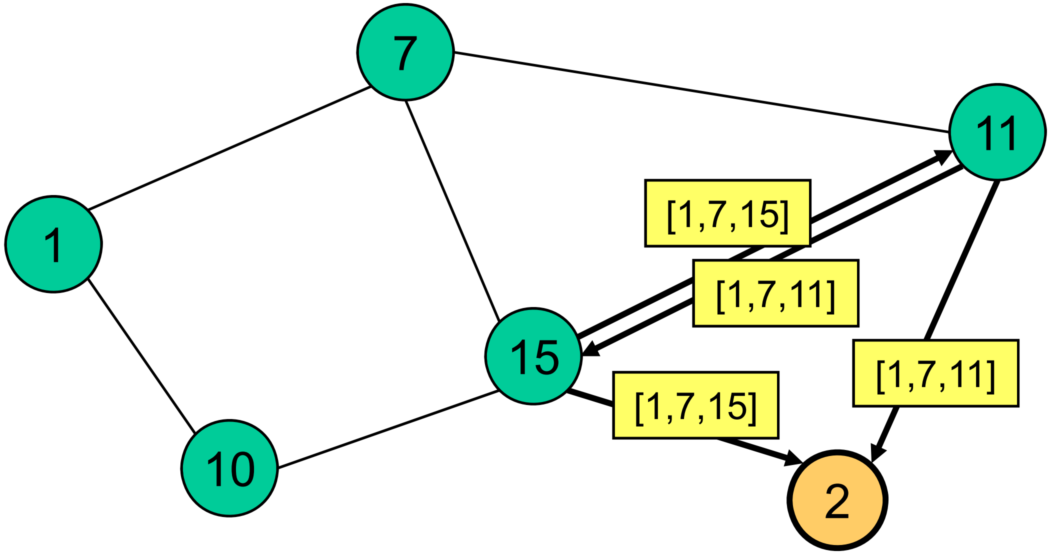
- Practically: ***Dynamic Source Routing (DSR)***
  - Use separate ***route request/route reply*** packets to discover route
    - Data packets only sent once route has been established
    - Discovery packets smaller than data packets
  - Store routing information in the discovery packets

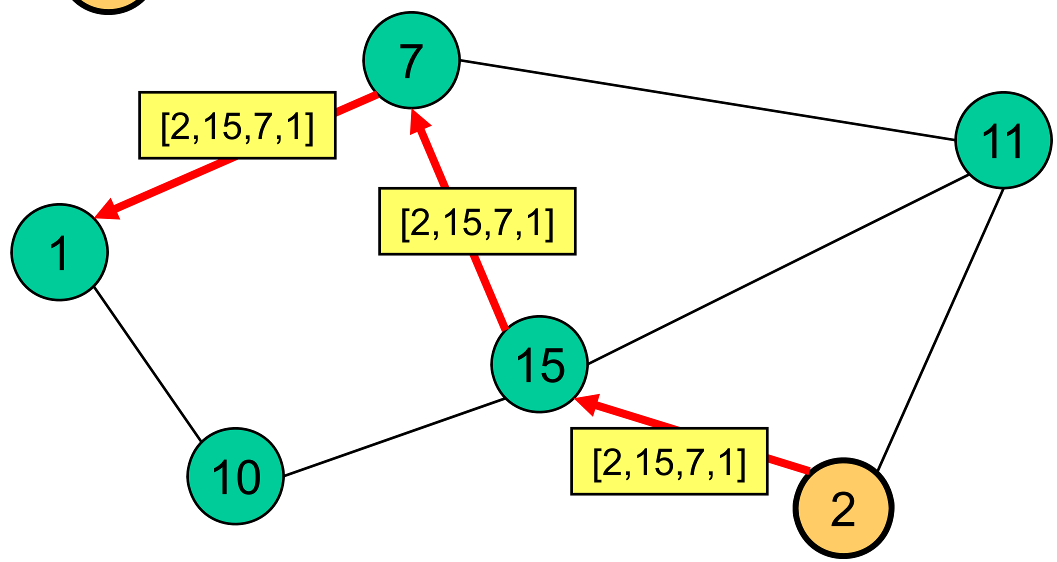
DSR Route Discovery Procedure





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

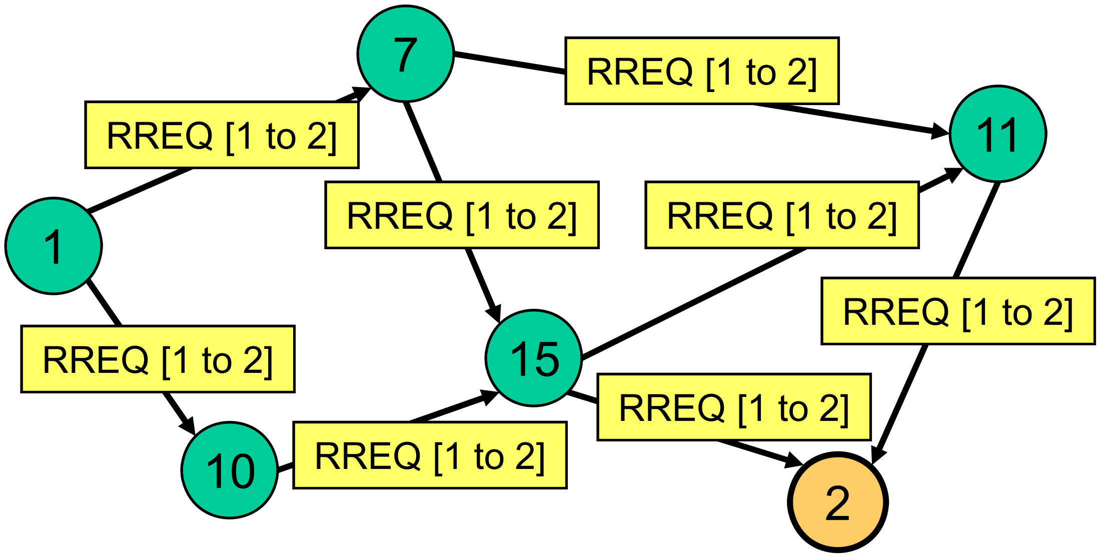
- Disadvantages
  - Connection setup delay is higher than in table-driven approaches
  - Does not locally repair broken links
  - Stale route information may result in inconsistencies
  - Performance degrades with increasing mobility
  - Routing overhead is directly proportional to the path length

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)

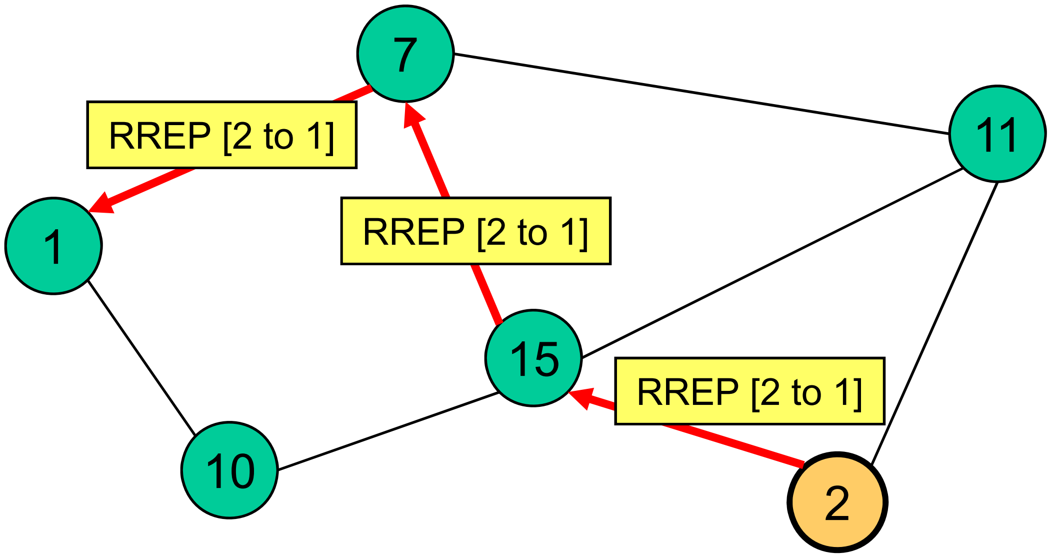


Node	Dest	Next	Dist
7	1	1	1
11	1	7	2
15	1	7	2
…	…	…	…
2	1	15	3

Routing tables after flooding the RREQ [1 to 2]

AODV – route setup

- The RouteReply (REP) is unicasted towards the source

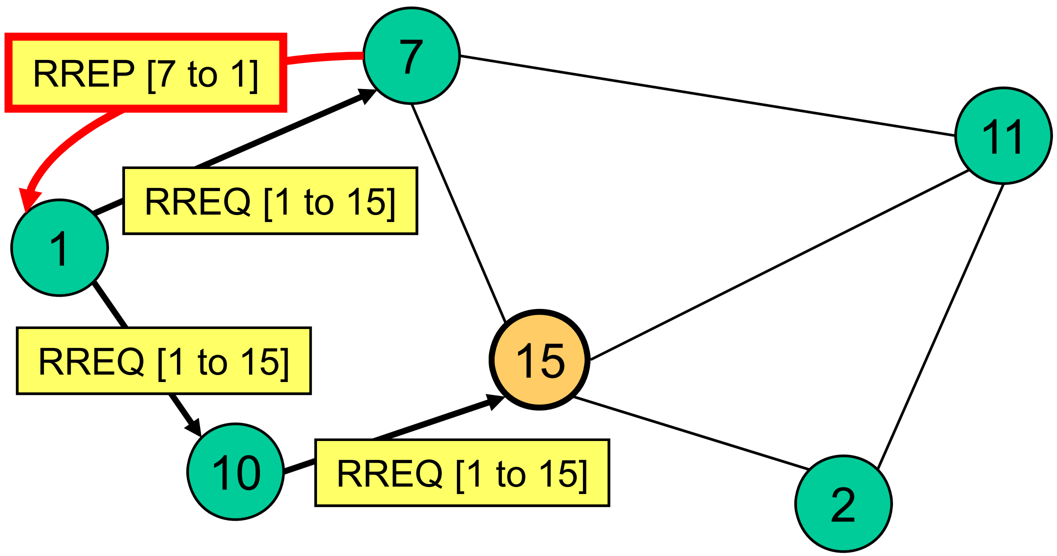


Node	Dest	Next	Dist
7	1	1	1
7	2	15	2
11	1	7	2
15	1	7	2
15	2	2	1
…	…	…	…
2	1	15	3

Routing tables after sending the RREP [2 to 1]

AODV – route setup

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

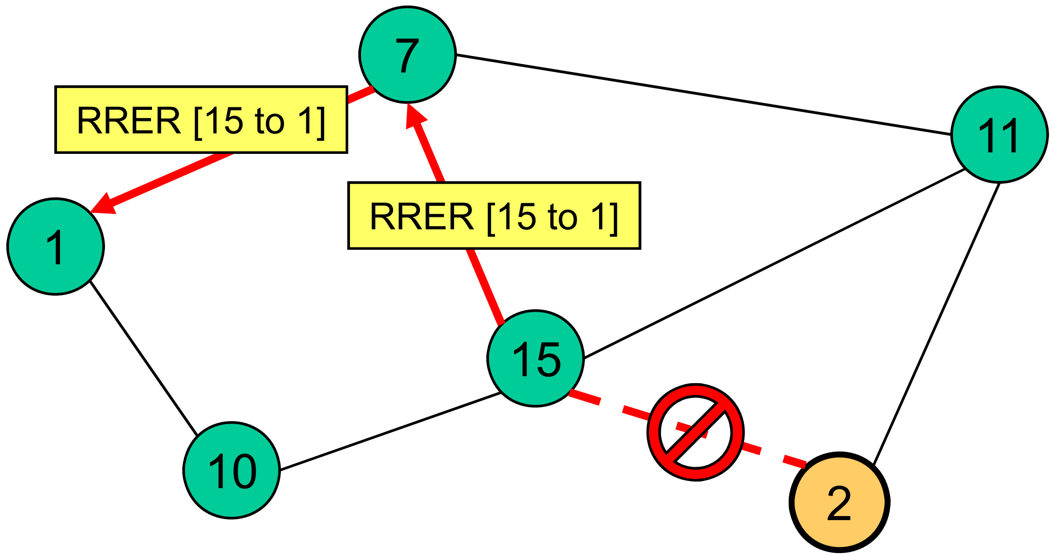


Node	Dest	Next	Dist
7	15	1	1
…	…	…	…

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



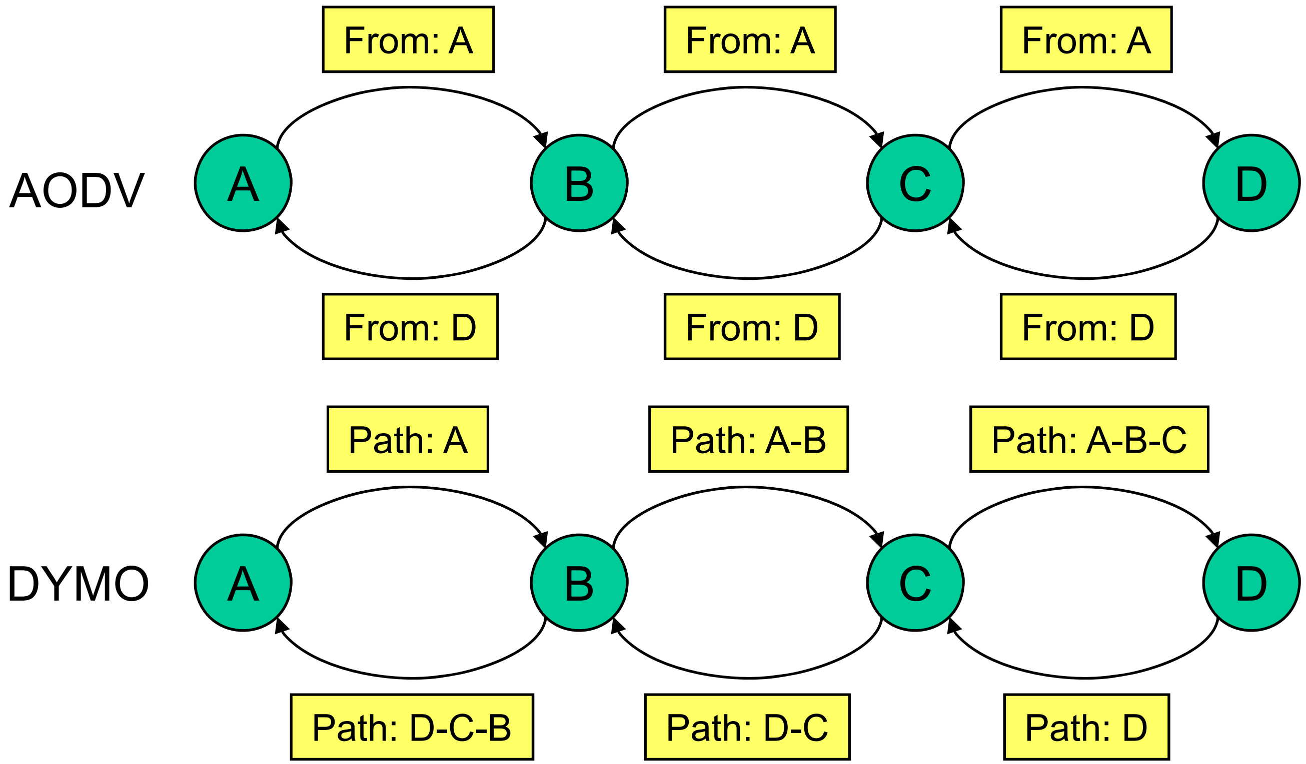
AODV

- 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

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

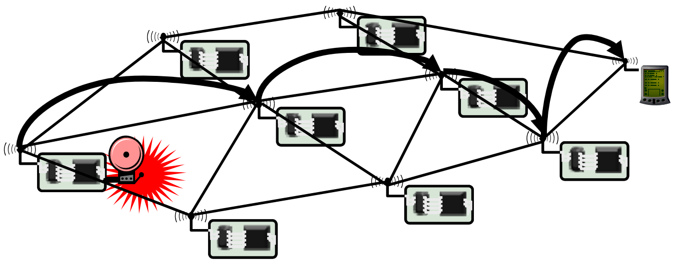


Geographic Routing

- 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

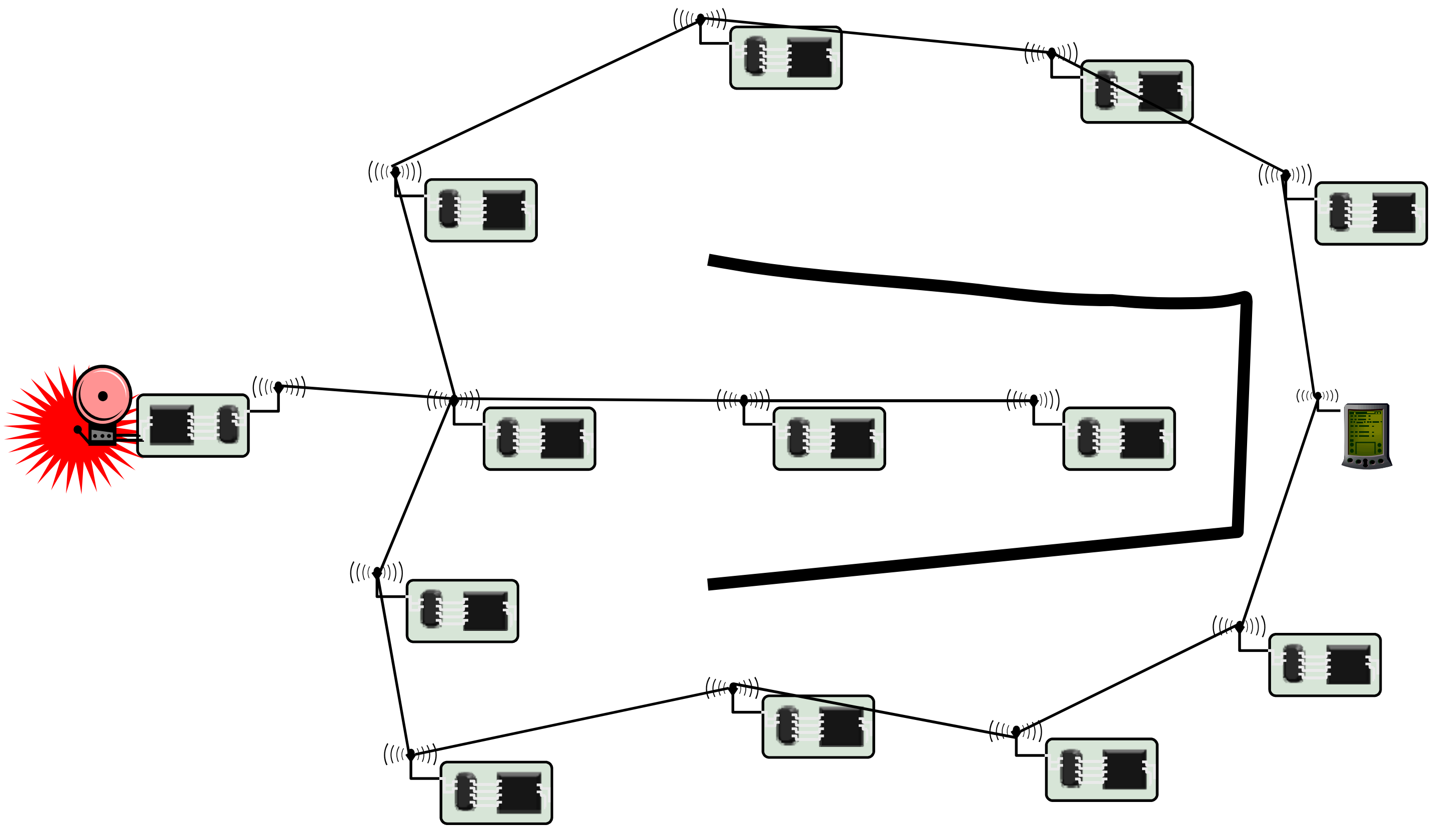
- “Most forward within range r” strategy
  - Send to that neighbor that realizes the most forward progress towards destination
  - NOT: farthest away from sender!



- 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

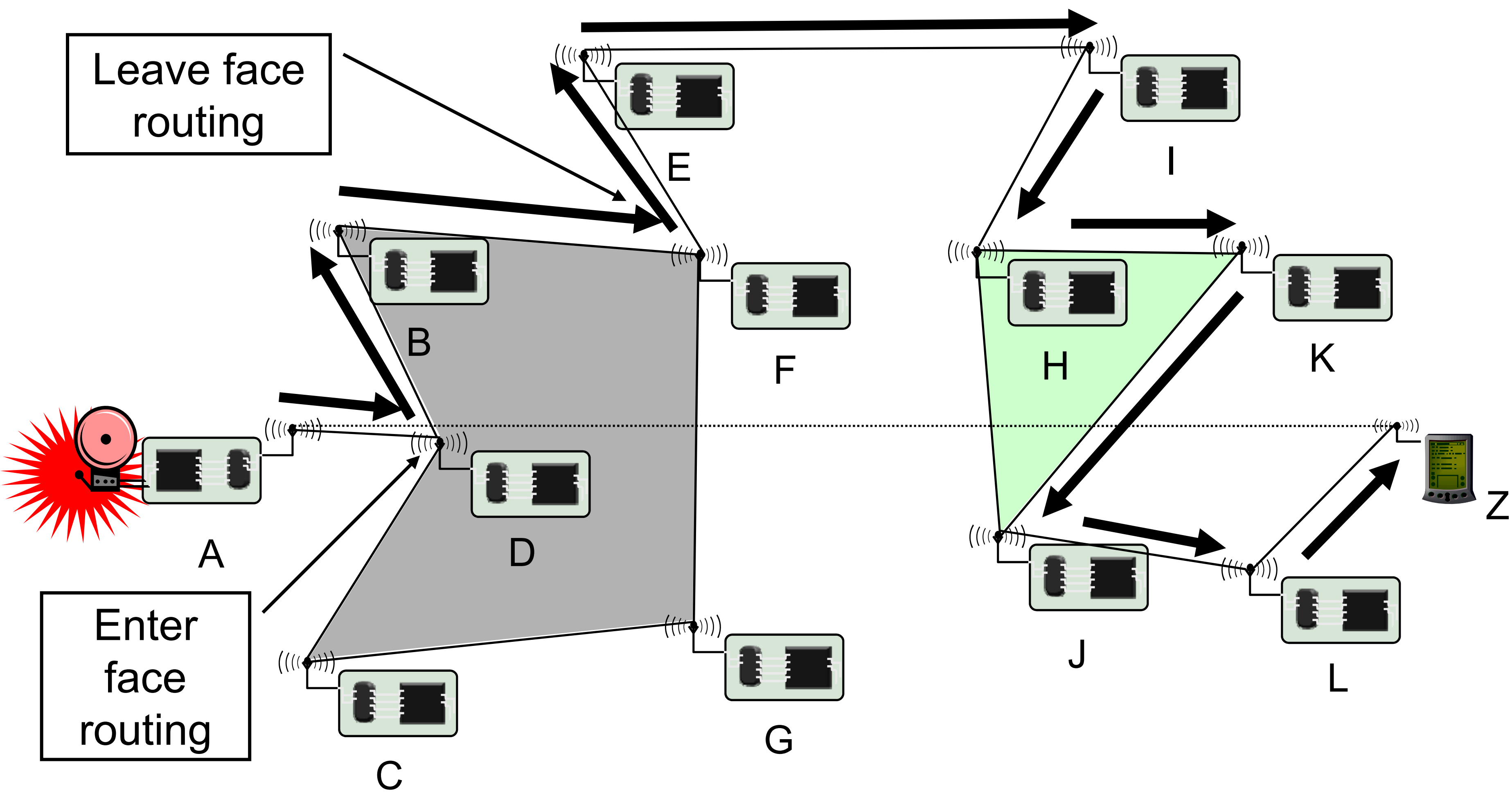


Right Hand Rule to Leave Dead Ends – GPSR

- 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
- ***Geometric Perimeter State Routing*** (***GPSR***)
  - Earlier versions: Compass Routing II, face-2 routing
  - Use greedy, "most forward" 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

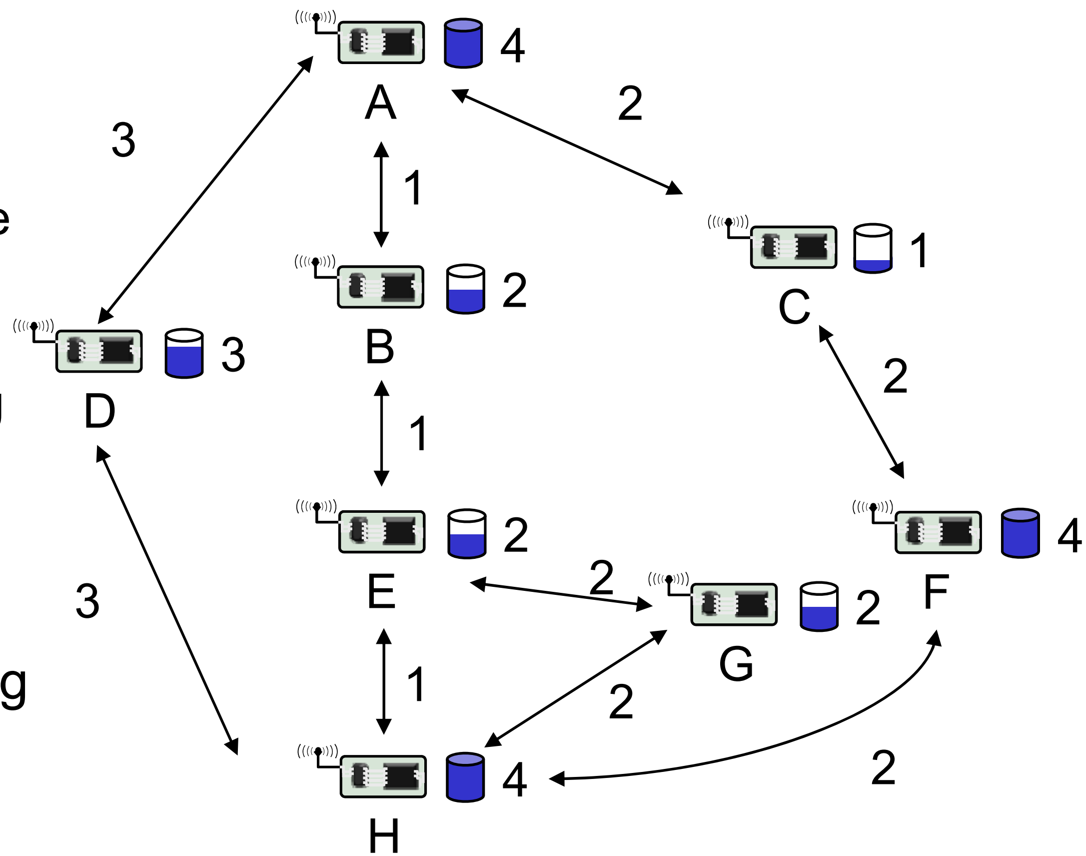


Energy-Aware Routing Protocols

- 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

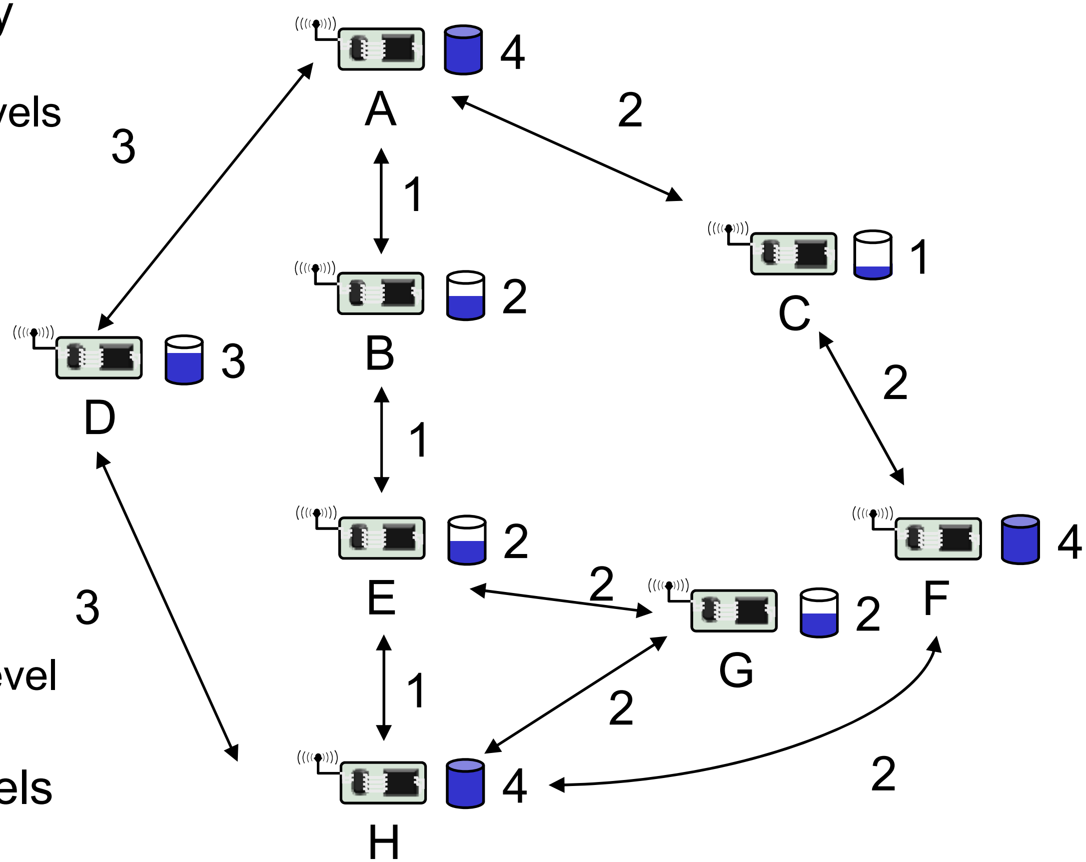
- Seems trivial – use proper link/path metrics (not hop count) and standard routing



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



A Non-trivial Path Metric

- Previous path metrics do not perform particularly well
- One non-trivial link weight: $w_{ij} = e_{ij}(\lambda^{\alpha_i} - 1)$
  - w_{ij} weight for link node i to node j
  - e_{ij} required energy, λ some constant, α_i fraction of battery of node i already used up
- Path metric: Sum of link weights
  - Use path with smallest metric
- Properties: Many messages can be send, high network lifetime
  - With admission control, even a competitive ratio logarithmic in network size can be shown

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} (please not, a_{nm} is a time!)

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

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

  - Optimized path according to the weakest link → ***path affinity*** $p_{x0,\, x1,\, \ldots\, xl}$

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

Optimized route stability

- Dynamic power adjustment based on the link affinity
  - Periodic exchange of Hello packets with constant power
  - Calculation of the signal strength of the transmitter ($S_{t,t+\tau}$)

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

  - Calculation of the adjusted transmission power ($P_{t,t+\tau}$)

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

Dynamic Address Allocation

Dynamic Address Allocation

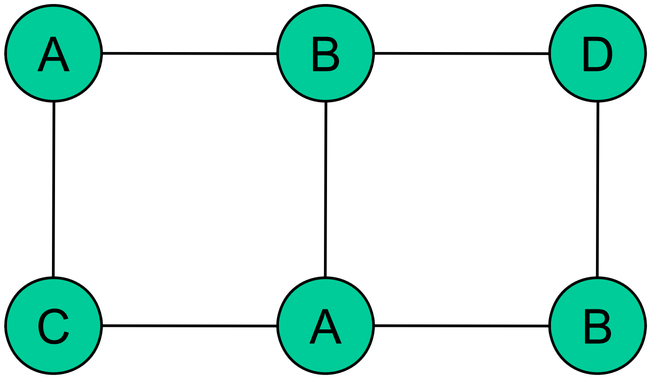
- Remember: ad hoc routing depends on unique address identifiers
  - Manufacturer set-up, e.g. Ethernet addresses (also used for Bluetooth)
    - Requires world-wide accepted standards
  - Application-based pre-programming of addresses
    - What about node replacements?
    - Pre-programming hundreds of nodes with node-specific software?
- Further arguments for dynamic address allocation
  - Privacy reasons → nodes should not be trackable
  - Heterogeneity of hardware modules prevents address pre-programming
  - Some applications do not need unique address identifiers except for maintenance and software updates

Dynamic Address Allocation

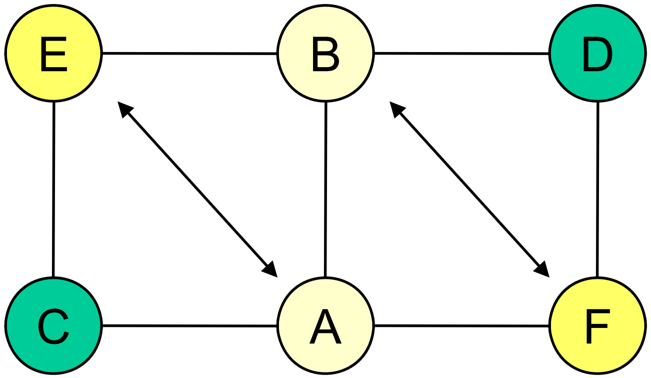
- Centralized approaches
  - DHCP and similar solutions
  - Rely on a central “address database”
  - Scalability to hundreds of nodes, each requesting an address?
- Decentralized approaches
  - Typically based on random address assignment + duplicate address detection (DAD)
- Leader-based approaches
  - Similar to centralized approaches but using dynamic leader election
- Best effort approaches
  - Explicitly allow duplicate addresses but use a weak DAD to enforce routing towards the “right” destination

PDAD

- Passive Duplicate Address Detection
  - Based on the original DAD algorithm
  - Uniqueness is identified by passively observing the network traffic (PDAD need to operate in cooperation with the routing protocol)



(a) Random address allocation



(b) Stepwise reinforcement after detecting address conflicts

Summary (what do I need to know)

- ***Address-based ad hoc routing vs. data-centric forwarding***
  - Main ideas and differences
  - Address-based path determination vs. geographical routing

- ***Proactive and reactive ad hoc routing***
  - Main ideas, commonalities, and differences
  - Working behavior of selected routing protocols (DSDV, DSR, AODV, DYMO)
  - Applicability in sensor networks
  - Primary measures: ***scalability*** and ***performance***

- ***Optimized route stability***
  - Link and path affinity
  - Transmission power approximation according to estimated node mobility

References

- S. Agarwal, A. Ahuja, J. P. Singh, and R. Shorey, "Route-Lifetime Assessment Based Routing (RABR) Protocol for Mobile Ad-Hoc Networks," Proceedings of IEEE International Conference on Communications (ICC 2000), vol. 3, New Orleans, LA, USA, June 2000, pp. 1697-1701.
- K. Akkaya and M. Younis, "A Survey of Routing Protocols in Wireless Sensor Networks," *Elsevier Ad Hoc Network Journal, vol. 3 (3), pp. 325-349, 2005.*
- I. Dietrich, C. Sommer, and F. Dressler, "Simulating DYMO in OMNeT++," University of Erlangen, Dept. of Computer Science 7, Technical Report 01/07, April 2007.
- D. B. Johnson and D. A. Maltz, "Dynamic Source Routing in Ad Hoc Wireless Networks," in *Mobile Computing, vol. 353, Imielinski and Korth, Eds.: Kluwer Academic Publishers, 1996, pp. 152-181.*
- M. Mauve and J. Widmer, "A Survey on Position-Based Routing in Mobile Ad-Hoc Networks," *IEEE Network, vol. 15 (6), pp. 30-39, 2001.*
- K. Paul, S. Bandyopadhyay, A. Mukherjee, and D. Saha, "Communication-Aware Mobile Hosts in Ad-hoc Wireless Network," Proceedings of IEEE International Conference on Personal Wireless Communication (ICPWC 1999), Jaipur, India, February 1999, pp. 83-87.
- C. E. Perkins and P. Bhagwat, "Highly Dynamic Destination-Sequenced Distance-Vector Routing (DSDV) for Mobile Computers," *Computer Communications Review, pp. 234-244, 1994.*
- C. Perkins and E. Royer, "Ad hoc On-Demand Distance Vector Routing," Proceedings of 2nd IEEE Workshop on Mobile Computing Systems and Applications, New Orleans, LA, February 1999, pp. 90-100.