# Wireless Networks for Mobile Applications

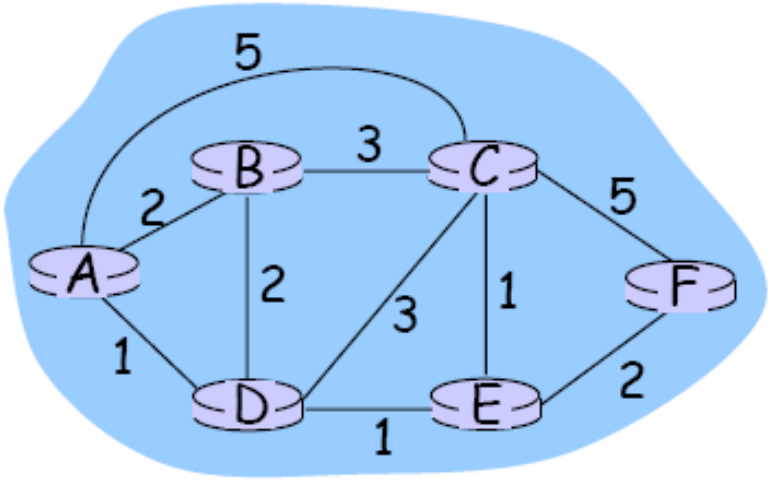
Prof. Claudio Palazzi
cpalazzi@math.unipd.it

Università degli Studi di Padova

# Routing

**Routing protocol**

Goal: determine "good" path (sequence of routers) thru network from source to dest.




Graph abstraction for routing algorithms:

- graph nodes are routers
- graph edges are physical links
  - link cost: delay, $ cost, or congestion level

- "good" path:
  - typically means minimum cost path
  - other def's possible

# Routing Algorithm Classification

## Global or decentralized information?

Global:

- all routers have complete topology, link cost info
- "link state" algorithms

Decentralized:

- router knows physically-connected neighbors, link costs to neighbors
- iterative process of computation, exchange of info with neighbors
- "distance vector" algorithms

## Static or dynamic?

Static:

- routes change slowly over time

Dynamic:

- routes change more quickly
  - periodic update
  - in response to link cost changes

# Example of a Linkstate Algorithm

## Dijkstra's algorithm

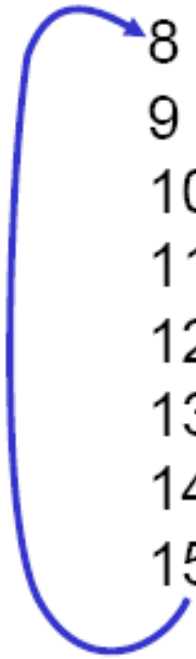
- net topology, link costs known to all nodes
  - accomplished via "link state broadcast"
  - all nodes have same info
- computes least cost paths from one node ('source") to all other nodes
  - gives forwarding table for that node
- iterative: after k iterations, know least cost path to k dest.'s

## Notation:

- c(x,y): link cost from node x to y; = ∞ if not direct neighbors
- D(v): current value of cost of path from source to dest. v
- p(v): predecessor node along path from source to v
- N': set of nodes whose least cost path definitively known

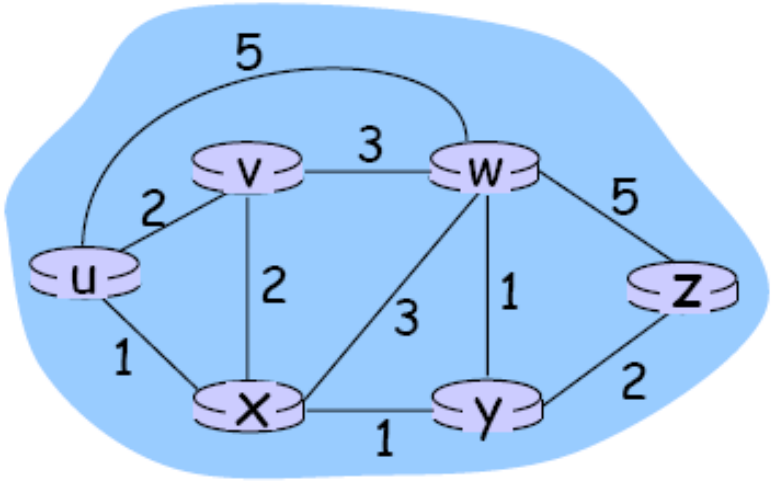
# Dijkstra Algorithm

```
1  Initialization:
2    N' = {u}
3    for all nodes v
4      if v adjacent to u
5          then D(v) = c(u,v)
6      else D(v) = ∞
7
8  Loop
9     find w not in N' such that D(w) is a minimum
10    add w to N'
11    update D(v) for all v adjacent to w and not in N' :
12       D(v) = min( D(v), D(w) + c(w,v) )
13    /* new cost to v is either old cost to v or known
14     shortest path cost to w plus cost from w to v */
15 until all nodes in N'
```



# Dijkstra Algorithm: Example

| Step | N' | D(v),p(v) | D(w),p(w) | D(x),p(x) | D(y),p(y) | D(z),p(z) |
|---|---|---|---|---|---|---|
| 0 | u | 2,u | 5,u | 1,u | ∞ | ∞ |
| 1 | ux | 2,u | 4,x | | 2,x | ∞ |
| 2 | uxy | 2,u | 3,y | | | 4,y |
| 3 | uxyv | | 3,y | | | 4,y |
| 4 | uxyvw | | | | | 4,y |
| 5 | uxyvwz | | | | | |



# Infrastructure Networks vs. MANETs

- Infrastructure-based wireless network …
  - Access points or base stations define cells or service areas
  - Routing is relatively simple -- there is just a single hop from the access point to the wireless node
- Ad hoc wireless network …
  - There is no pre-defined or static network structure imposed by infrastructure
  - Wireless nodes are not necessarily all adjacent, so a node may need to forward data for other nodes (i.e., to participate in routing)

# Wireline Networks vs. MANETs

- Wireline networks …
  - Symmetric links, usually with respect to both capacity and quality
  - Limited planned redundancy for reliability and load sharing
  - Planned links, typically of uniformly high quality, in a fixed topology
- Ad hoc wireless networks …
  - Asymmetric links
  - High degree of random redundancy in connectivity between wireless nodes
  - Unplanned, dynamic links with quality that may vary greatly due to interference, signal, etc.

# Inadequacy of Traditional Routing Algorithms in MANETs

- Dynamic of the topology
  - frequent changes of connections, connection quality, participants
- Limited performance of mobile systems
  - periodic updates of routing tables need energy without contributing to the transmission of user data, sleep modes difficult to realize
  - limited bandwidth of the system is reduced even more due to the exchange of routing information
  - links can be asymmetric, i.e., they can have a direction dependent transmission quality
- Problem
  - protocols have been designed for fixed networks with infrequent changes and typically assume symmetric links

# Inadequacy of Traditional Routing Algorithms in MANETs

- Traditional routing algorithms are likely to be …
  - Inefficient due to slow convergence times (e.g., using distance vector algorithms as in RIP)
  - Non-functional due to large amounts of data or inability to deal with asymmetric links (e.g., link state algorithms as in OSPF)
- MANET routing must rely on data link information, not just network layer updates
  - Link layer determines connectivity and quality of links

# Inadequacy of Traditional Routing Algorithms in MANETs

- Centralized approaches are too slow and not robust enough for MANET
- All (or almost all) nodes in a MANET may be routers
- Long-lived circuits cannot be used in MANETs
- Path length (hop count) may not be the best metric for routing in MANETs

# Goals of a Good Unicast Routing Protocol

- Minimal control overhead
- Minimal processing overhead
- Multi-hop path routing capability
- Dynamic topology maintenance
- No loops
- Self-starting

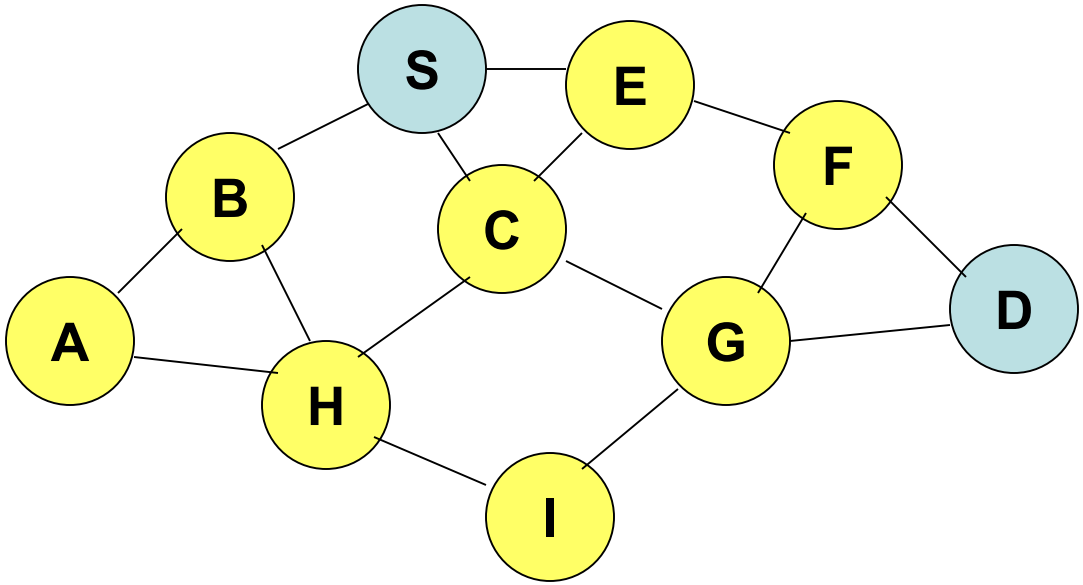
# Routing in Ad Hoc Wireless Networks: in Summary

- Routing in mobile ad hoc networks (MANETs) is a tough problem:
  - Unreliable wireless medium
  - Mobile nodes
  - No central authority

- Other issues
  - Traffic patterns application specific
  - Energy constraints

# Example

**ad hoc wireless network**




Nodes have unique identifiers

Routing problem – find path between S and D

# Classification of routing protocols

- Table-driven (**proactive**)
  - Up-to-date routing information maintained
  - Routing overhead independent of route usage
- Source-initiated (demand-driven / reactive / **on demand**)
  - Routes maintained only for routes in use
  - Explicit route discovery mechanism
- Hybrid Protocols
  - Combination of proactive and reactive

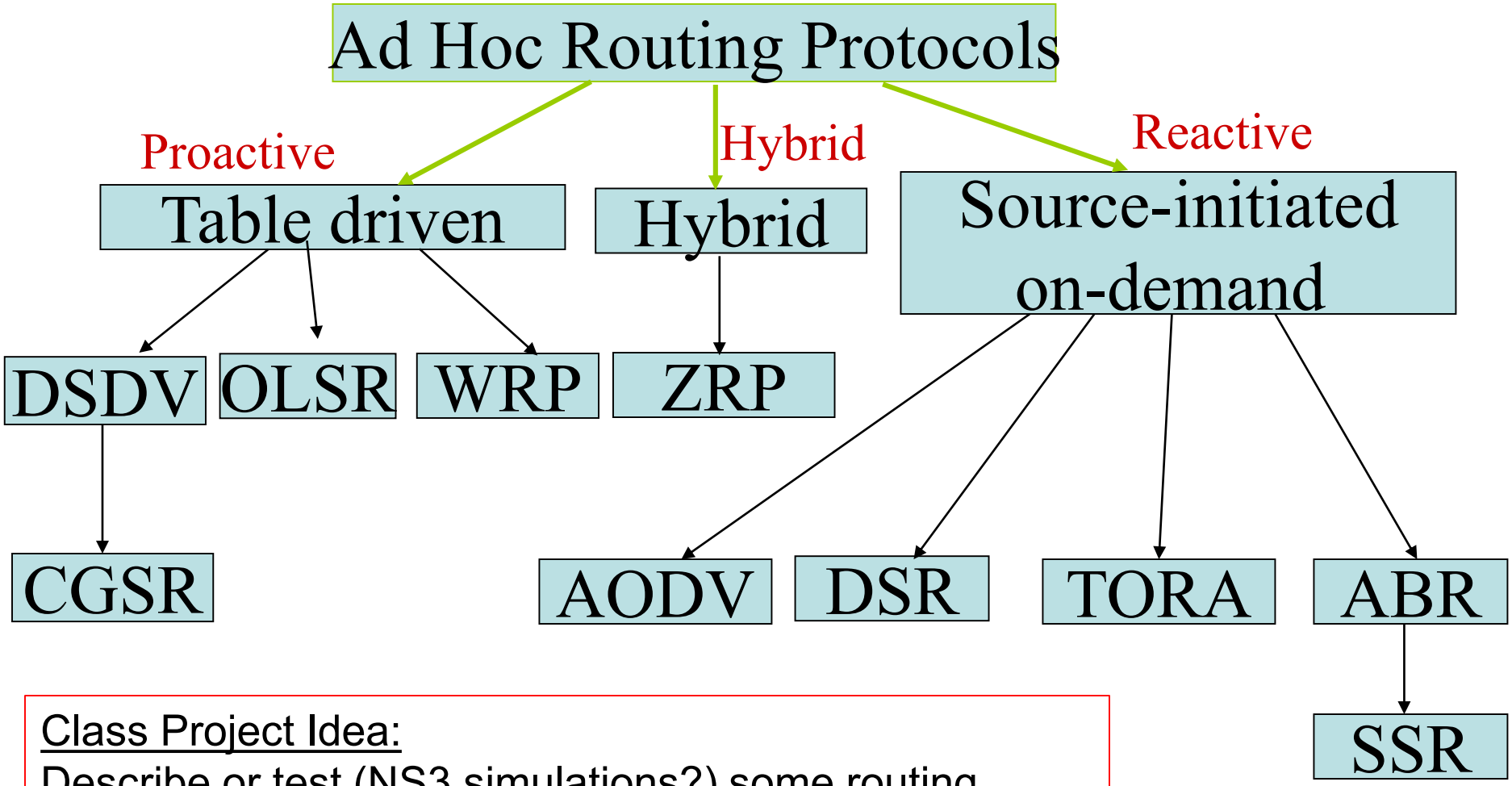
# Proactive Approach

- Based on traditional distance-vector and link-state protocols
- Each node maintains route to each other network edge
- Periodic and/or event triggered routing update exchange
- Higher overhead in most scenarios because of continuous updates
  - Mobility results in significant updates – some of the updates may never be used, if the ad hoc network moves faster than the route requests
- Low latency of packet forwarding as the route is known (compared to reactive approach)
- Longer route convergence time
- Examples: DSDV, WRP, TBRPF, OLSR

# Reactive Approach

- Source builds routing on demand by "flooding"
- Maintain only active routes
- Route Discovery cycle
- Typically, less control overhead, better scaling properties
- Drawback: route acquisition latency or long delay in finding the route
  - May not be suitable for real-time traffic
- Example: AODV, DSR

# Classification (cont.)




Class Project Idea:
Describe or test (NS3 simulations?) some routing protocols for MANET, or VANET, or FANET, or Space…

# Table-driven (Proactive) Routing Protocols

- Each node maintains an updated routing table
  - Contains routes to all nodes in the network
- Changes to network topology are immediately propagated
  - Tables need to be consistent
- Protocols differ in mechanisms used to propagate topology information

# Destination Sequenced Distance Vector (DSDV)

- Based on Bellman-Ford algorithm
- Enhanced with sequence number to avoid loops
  - Fresher routes have higher sequence numbers
- Optimizations added to reduce routing overheads
  - incremental data exchange, delayed exchange of updates

# DSDV Routing

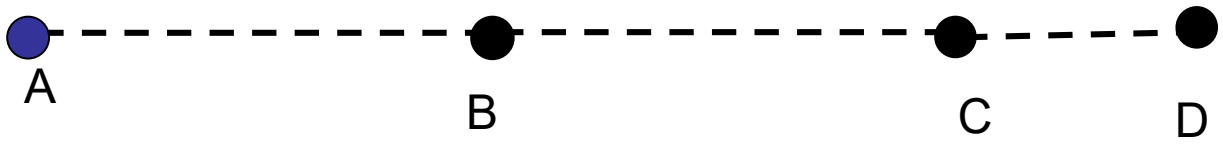
- Packets transmitted according to the routing table
- Each node maintains routing table with entry for each node in the network

  <dest_addr, dest_seqn#, next-hop, hop_count, install_time>
- Each node maintains its own sequence number
  - Updates at each change in neighborhood information
  - Used for freedom from loops
  - To distinguish stale routes from new ones

# DSDV Example

Routing Table of Node A

| Destination | Next | Metric | Seq. Nr |
|---|---|---|---|
| A | A | 0 | A-550 |
| B | B | 1 | B-102 |
| C | B | 2 | C-588 |
| D | B | 3 | D-312 |




Route information is exchanged periodically

# DSDV Routing

- Each node periodically transmits updates to keep table consistency
  - Includes its own sequence number #, route table updates

    <dest_addr, dest_seq#, hop-count>
- Nodes also send routing table updates for important link changes (i.e. link breaks)
- When two routes to a destination received from two different neighbors
  - Choose the one with the greatest destination sequence number
  - If equal, choose the smallest hop-count

# DSDV Routing

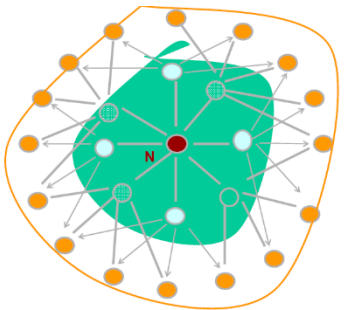
- Routing table updates create lots of control traffic
- DSDV addresses this problem by using two types of routing update packets
  1. Full Dumps
     - Carry all routing table information (Several NPDUs)
     - Transmitted relatively infrequently
  2. Incremental Updates
     - Carry only information changed since last full dump
     - Fits within one network protocol data unit (NPDU)
     - When updates can no longer fit in one NPDU, send full dump

# Optimized Link State Routing Protocol (OLSR)

- The Optimized Link State Routing Protocol (OLSR)
  - Based on the link state algorithm
  - All links with neighboring MHs are declared and are flooded in the entire network
  - Minimizes flooding of this control traffic by using only the selected MHs, called multipoint relays
  - Only normal periodic control messages sent
  - Beneficial for the traffic patterns with a large subset of MHs are communicating with each other
  - Good for large and dense networks
  - An in-order delivery of its messages is not needed as each control message contains a sequence number

# Multipoint Relays (for Broadcast)

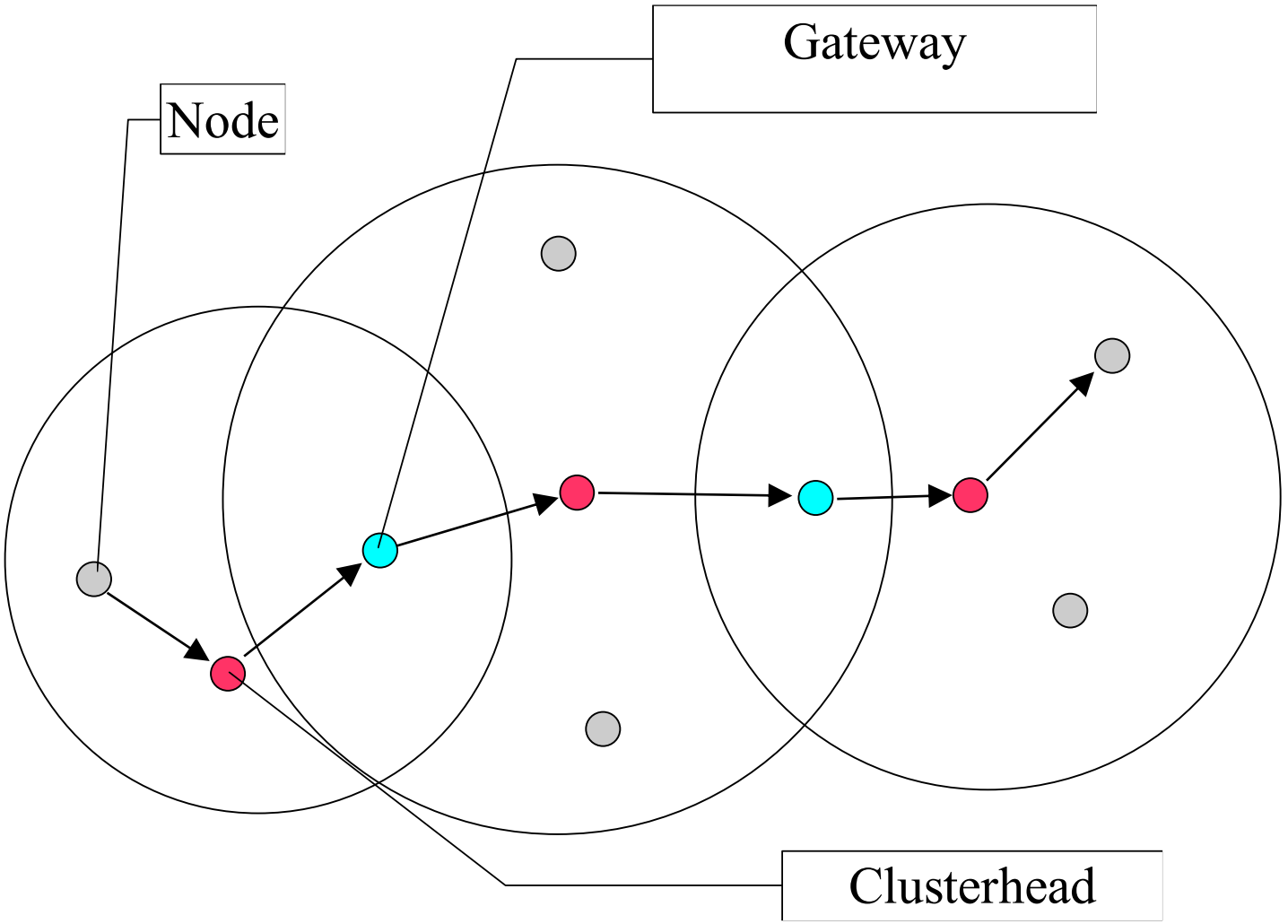
- Multipoint Relays
  - Minimize the flooding of broadcast packets in the network by reducing duplicate retransmissions in the same region
  - Each MH selects a set of neighboring MHs, to retransmit its packets and is called the multipoint relays (MPRs)
  - This set can change over time and is indicated by the selector nodes in their hello messages
  - Each node selects MPR among its one hop bi-directional link neighbors to all other nodes that are two hops away



# Clusterhead Gateway Switch Routing (CGSR)

- Nodes organized into hierarchy of clusters.
- Each node has a clusterhead, selected using an election.
- Nodes send packet through clusterheads.
- Clusterheads communicate amongst themselves using DSDV.
  - Two clusters are connected through a gateway node

# CSGR



# Source-Initiated On-Demand Routing

- Create routes only when needed
  - Maintained until route is broken or the source does not need it anymore
- Routes found using a “route discovery” process
  - flooding
- Route maintenance procedure used to repair routes

# Ad Hoc On-Demand Distance Vector Routing (AODV)

- Now RFC 3561, based on DSDV
- Destination sequence numbers provide loop freedom
- Routing table exchanges only happen along a given route
- Source sends Route Request Packet (RREQ) when a route has to be found
- Route discovery floods all nodes with requests
  - A node responds the first time it receives a request.
  - Replies only if it has contact with the destination, or a valid route to the destination
- Route Reply Packet (RREP) is sent back by destination
- Route Error messages update routes

# AODV

- If routes are not used they expire and get discarded
  - Reduces stale routes
  - Does not require explicit route maintenance
- Minimizes the number of active routes between source and destination
- Can identify multiple routes between a source and a destination but implement only one of them

# AODV Properties

1. AODV discovers routes as and when necessary
   - Does not maintain routes from every node to every other
2. Routes are maintained just as long as necessary
3. Every node maintains its monotonically increasing sequence number -> increases every time the node notices change in the neighborhood topology

# AODV Properties

- AODV utilizes routing tables to store routing information
  1. A Routing table for unicast routes
  2. A Routing table for multicast routes

- The route table stores: <destination addr, next-hop addr, destination sequence number, life_time>

- For each destination, a node maintains a list of precursor nodes, to route through them
  - Precusor nodes help in route maintenance (more later)

- Life-time updated every time the route is used
  - If route not used within its life time -> it expires

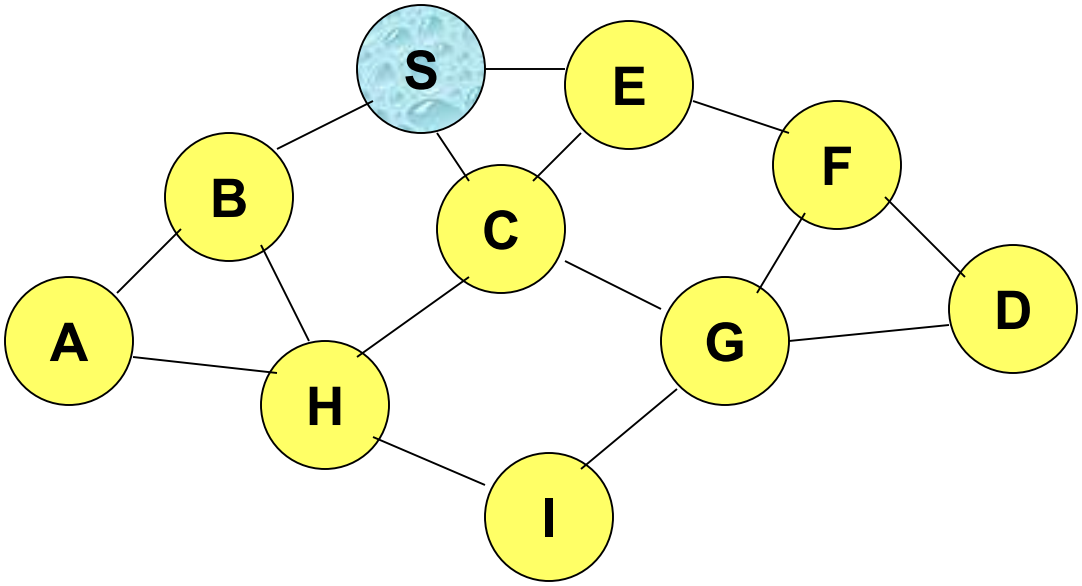
# AODV Route Discovery

- When a node wishes to send a packet to some destination –
  - It checks its routing table to determine if it has a current route to the destination
    - If Yes, forwards the packet to next hop node
    - If No, it initiates a route discovery process
- Route discovery process begins with the creation of a Route Request (RREQ) packet -> source node creates it
- The packet contains – source node's IP address, source node's current sequence number, destination IP address, destination sequence number

# AODV Route Discovery

- Packet also contains broadcast ID number
  - Broadcast ID gets incremented each time a source node uses RREQ
  - Broadcast ID and source IP address form a unique identifier for the RREQ
- Broadcasting is done via Flooding

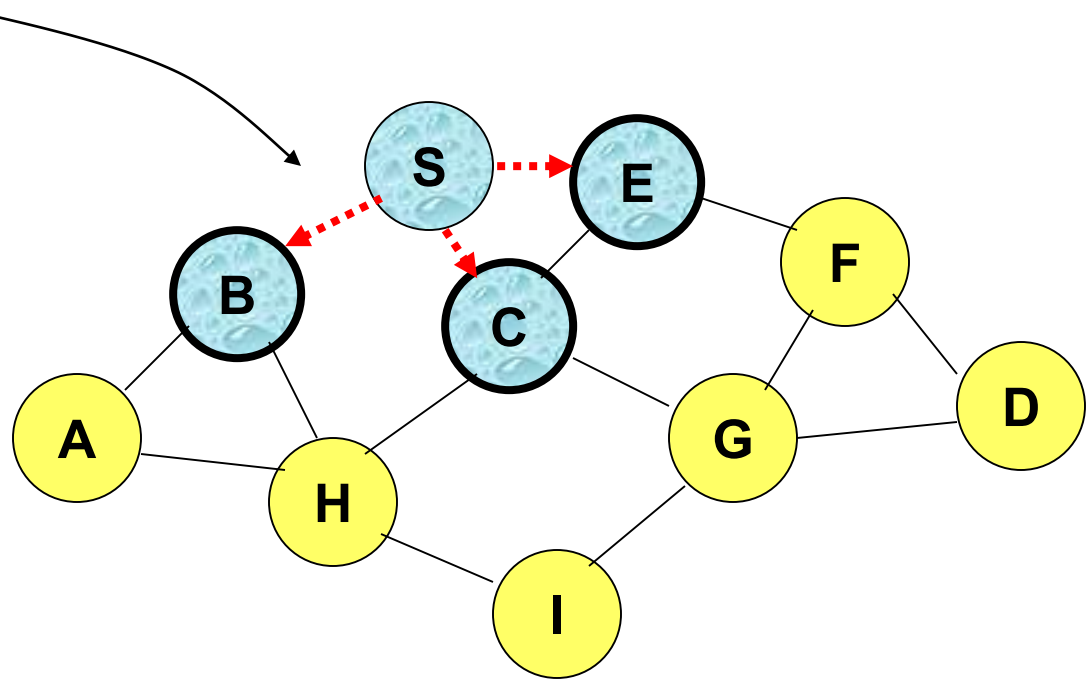
# Route Requests in AODV




**Represents a node that has received RREQ for D from S**

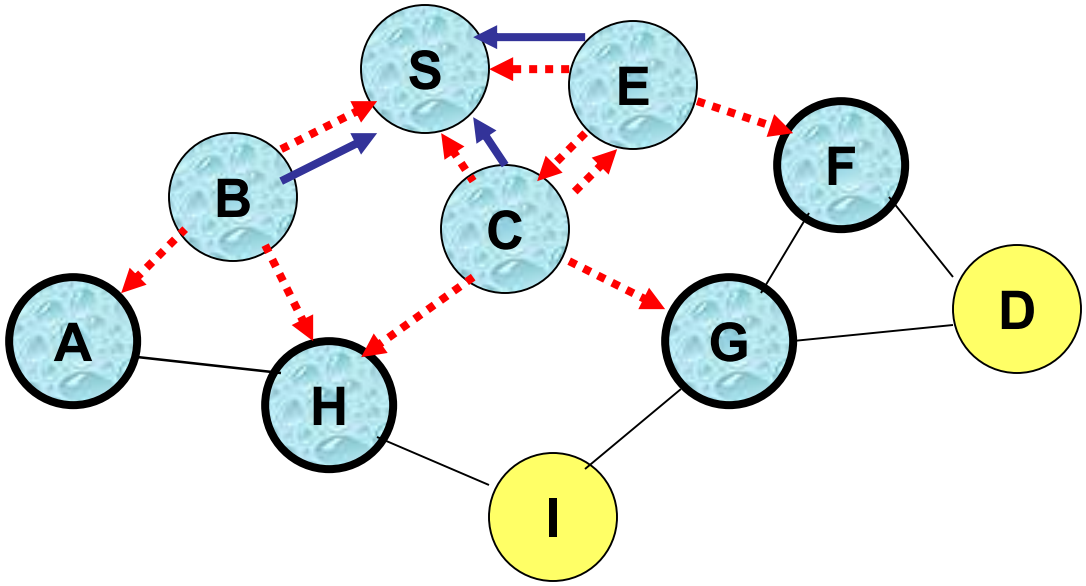
# Route Requests in AODV

**Broadcast transmission**



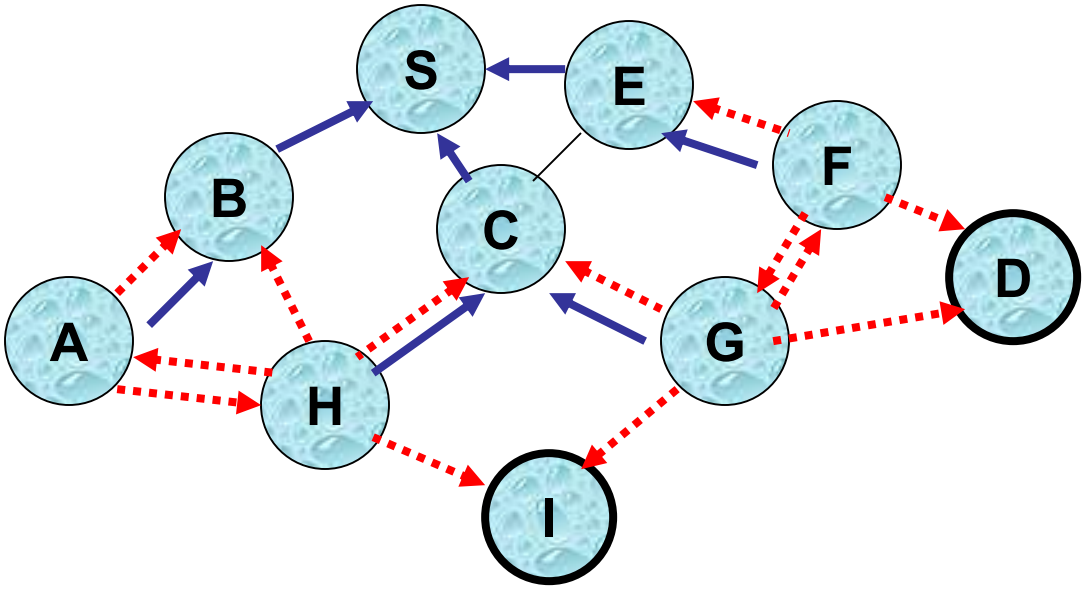

**Represents transmission of RREQ**

# Route Requests in AODV



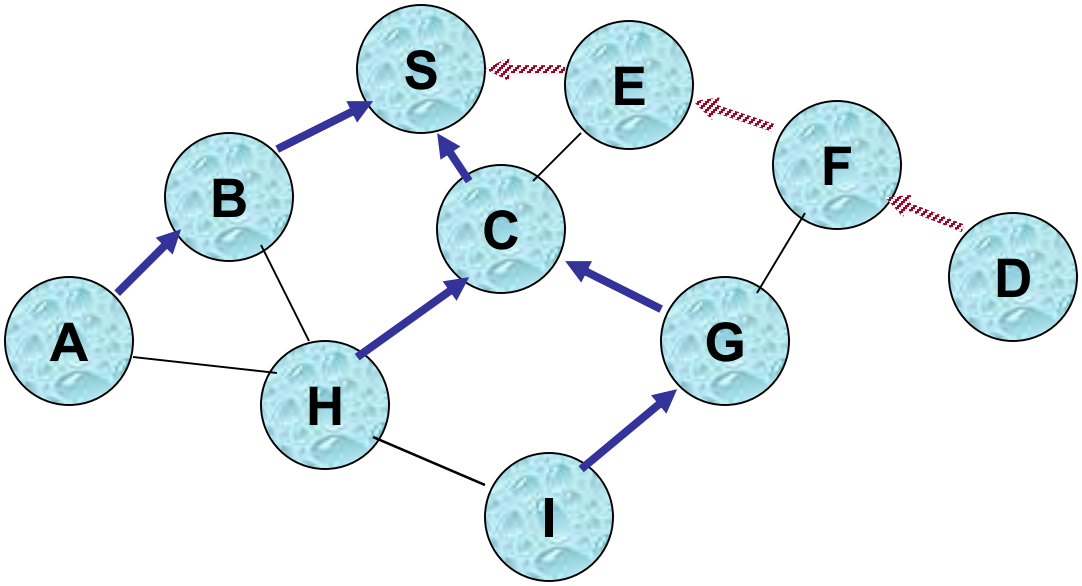

Represents links on Reverse Path

# Reverse Path Setup in AODV




- **Node C receives RREQ from G and H, but does not forward it again, because node C has already forwarded RREQ once**

# Route Reply in AODV




Represents links on path taken by RREP

# Route Discovery

- Once an intermediate node receives a RREQ, the node sets up a reverse route entry for the source node in its route table
  - Reverse route entry consists of <Source IP address, Source seq. number, number of hops to source node, IP address of node from which RREQ was received>
  - Using the reverse route a node can send a RREP (Route Reply packet) to the source
    - Reverse route entry also contains – life time field
- RREQ reaches destination -> In order to respond to RREQ a node should have in its route table:
  1. Unexpired entry for the destination
  2. Seq. number of destination at least as great as in RREQ (for loop prevention)

# Route Discovery

- RREQ reaches destination (contd.)
  - If both conditions are met & the IP address of the destination matches with that in RREQ → the node responds to RREQ by sending a RREP back using unicasting and not flooding to the source using reverse path
  - If conditions are not satisfied, then node increments the hop count in RREQ and broadcasts to its neighbors
- Ultimately the RREQ will make to the destination

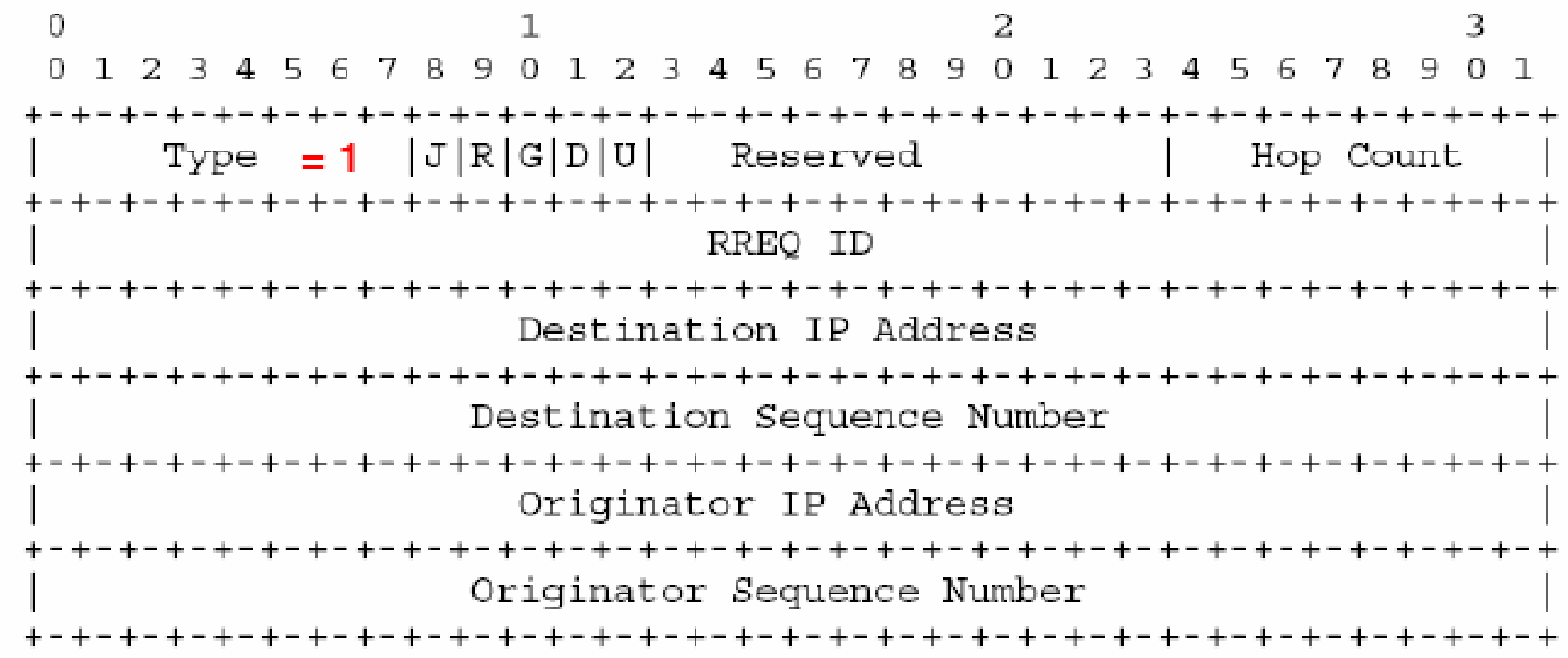
# Route Discovery

- An intermediate node (not the destination) may also send a Route Reply (RREP) provided that it knows a more recent path than the one previously known to sender S

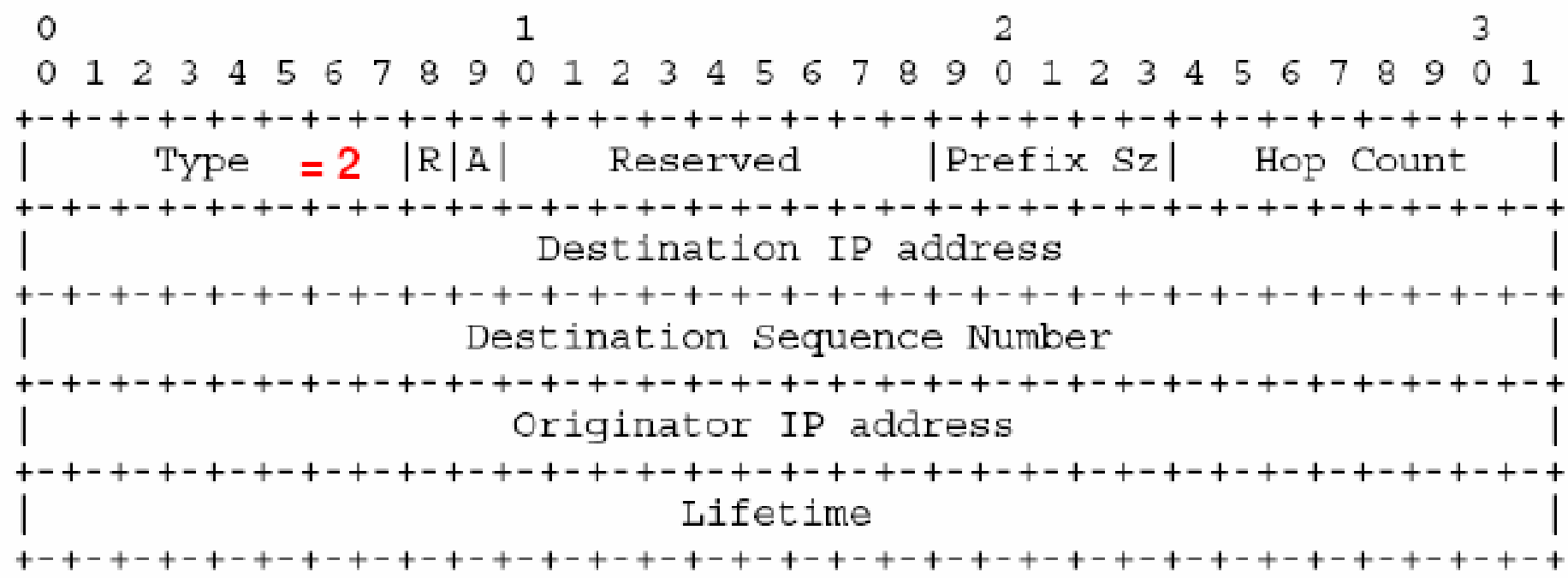
- To determine whether the path known to an intermediate node is more recent, *destination sequence numbers* are used

- The likelihood that an intermediate node will send a Route Reply when using AODV not as high as DSR (Later)
  - A new Route Request by node S for a destination is assigned a higher destination sequence number. An intermediate node which knows a route, but with a smaller sequence number, cannot send Route Reply

# RREQ Message



# RREP Message



# Route Discovery - Timeouts

- A routing table entry maintaining a reverse path is purged after a timeout interval
  - timeout should be long enough to allow RREP to come back

- A routing table entry maintaining a forward path is purged if *not used* for a *active_route_timeout* interval
  - if no is data being sent using a particular routing table entry, that entry will be deleted from the routing table (even if the route may actually still be valid)

# Link Failure Detection

- *Hello* messages: Neighboring nodes periodically exchange hello message
- Absence of hello message is used as an indication of link failure
- Alternatively, failure to receive several MAC-level acknowledgements may be used as an indication of link failure

# AODV: Optimization

- Route Requests are initially sent with small Time-to-Live (TTL) field, to limit their propagation
  - DSR also includes a similar optimization
- If no Route Reply is received, then larger TTL tried
- Expanding Ring Search
  - Prevents flooding of network during route discovery
  - Control Time to Live (TTL) of RREQ to search incrementally larger areas of network
  - Advantages: Less overhead when successful
  - Disadvantages: Longer delay if route not found immediately

# DSDV vs. AODV

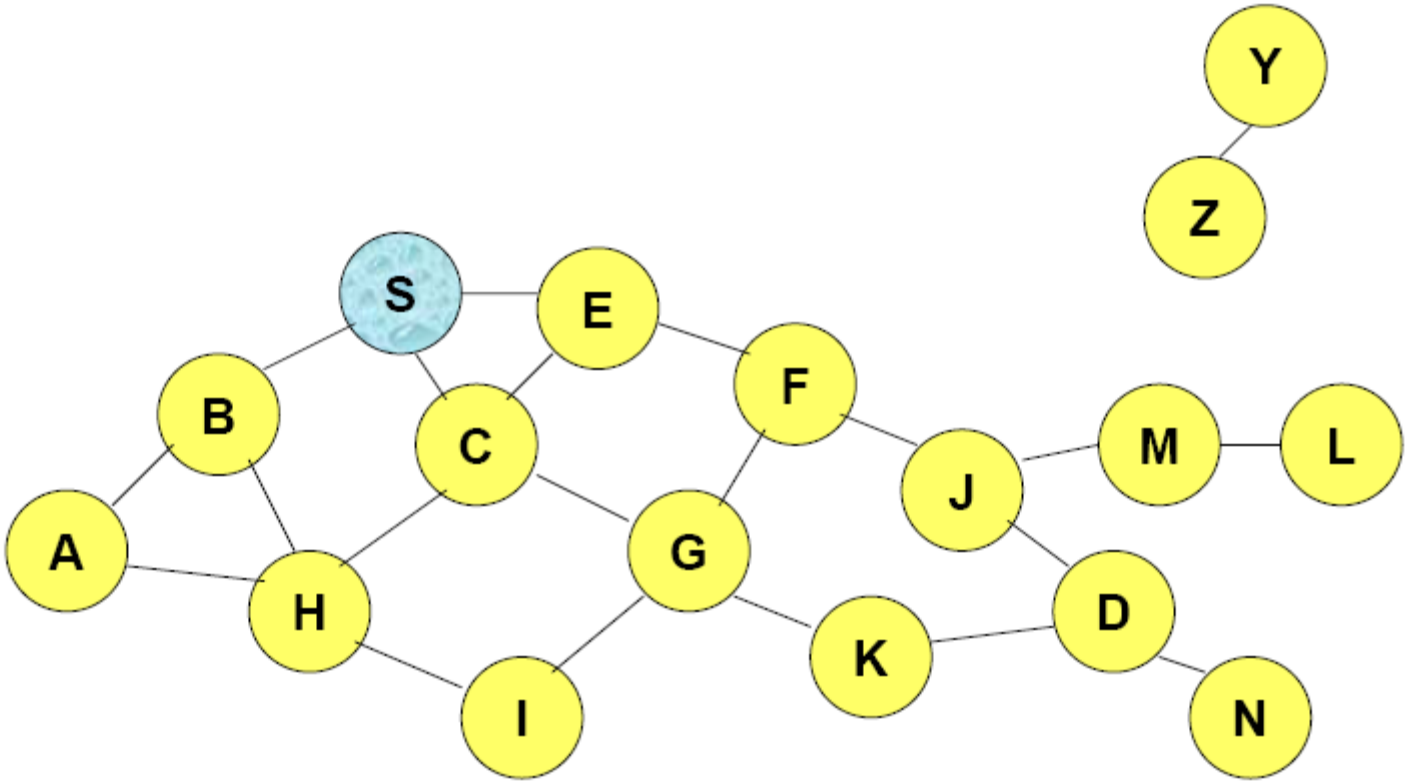
- DSDV broadcasts every change in the network to every node
- When two neighbors enter communication range of each other
  - This results in a network wide broadcast
- Similarly when two nodes drift apart from each other's range -> link breakage
  - Also results in a network wide broadcast
- Local movements have global effects

- In AODV such broadcasts are not necessary
- If a link breakage does not affect on going transmission -> no global broadcast occurs
- Only affected nodes are informed
- Local movements of nodes have local effects
- AODV reduces the network wide broadcasts to the extent possible
- Significant reduction in control overhead as compared to DSDV

# Flooding for Control Packet Delivery

- Sender S broadcasts a control packet P to all its neighbors
- Each node receiving P forwards P to its neighbors
- Sequence numbers help to avoid the possibility of forwarding the same packet more than once
- Packet P reaches destination D provided that D is reachable from sender S
- Node D does not forward the packet

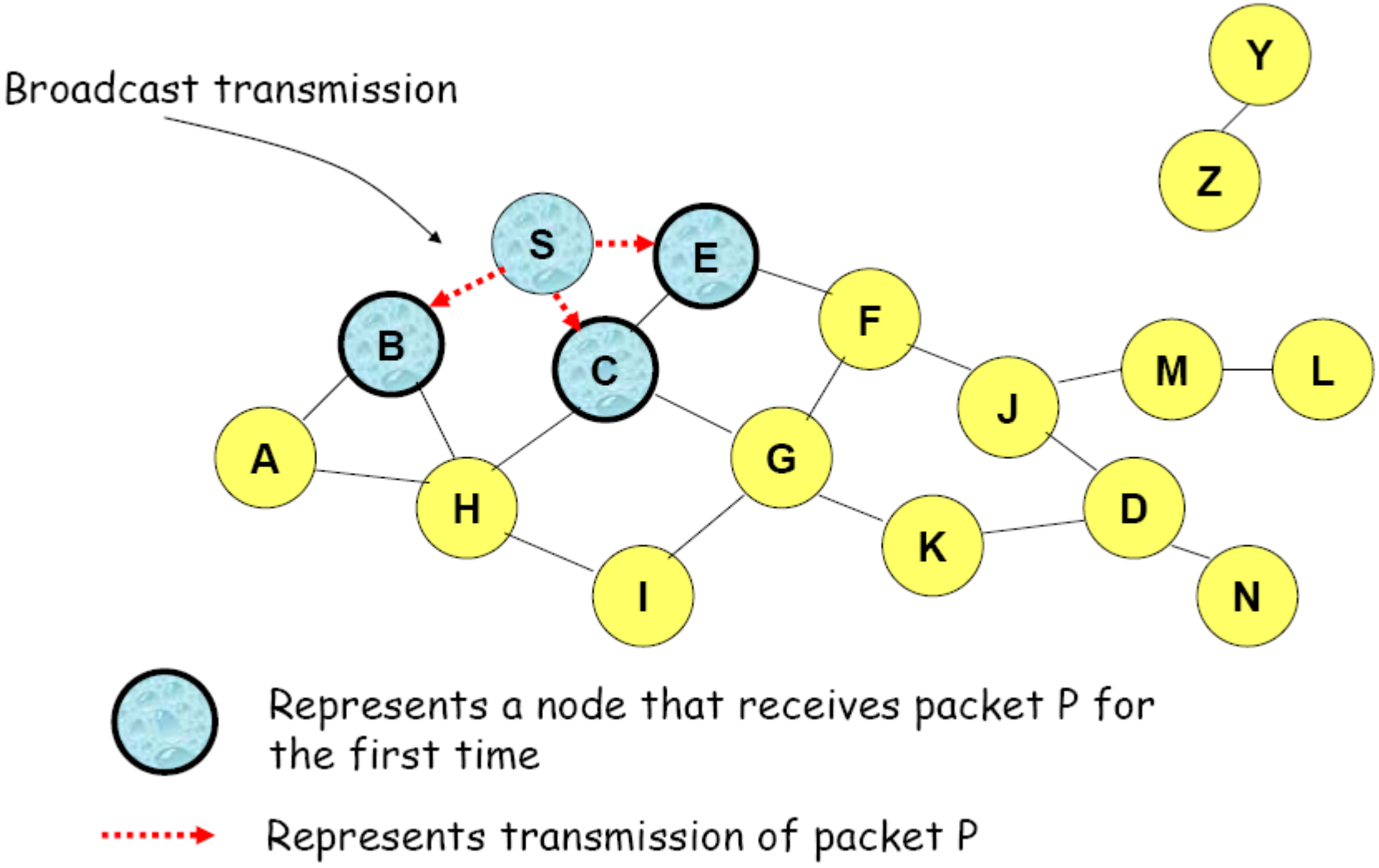
# Flooding for Control Packet Delivery: Example



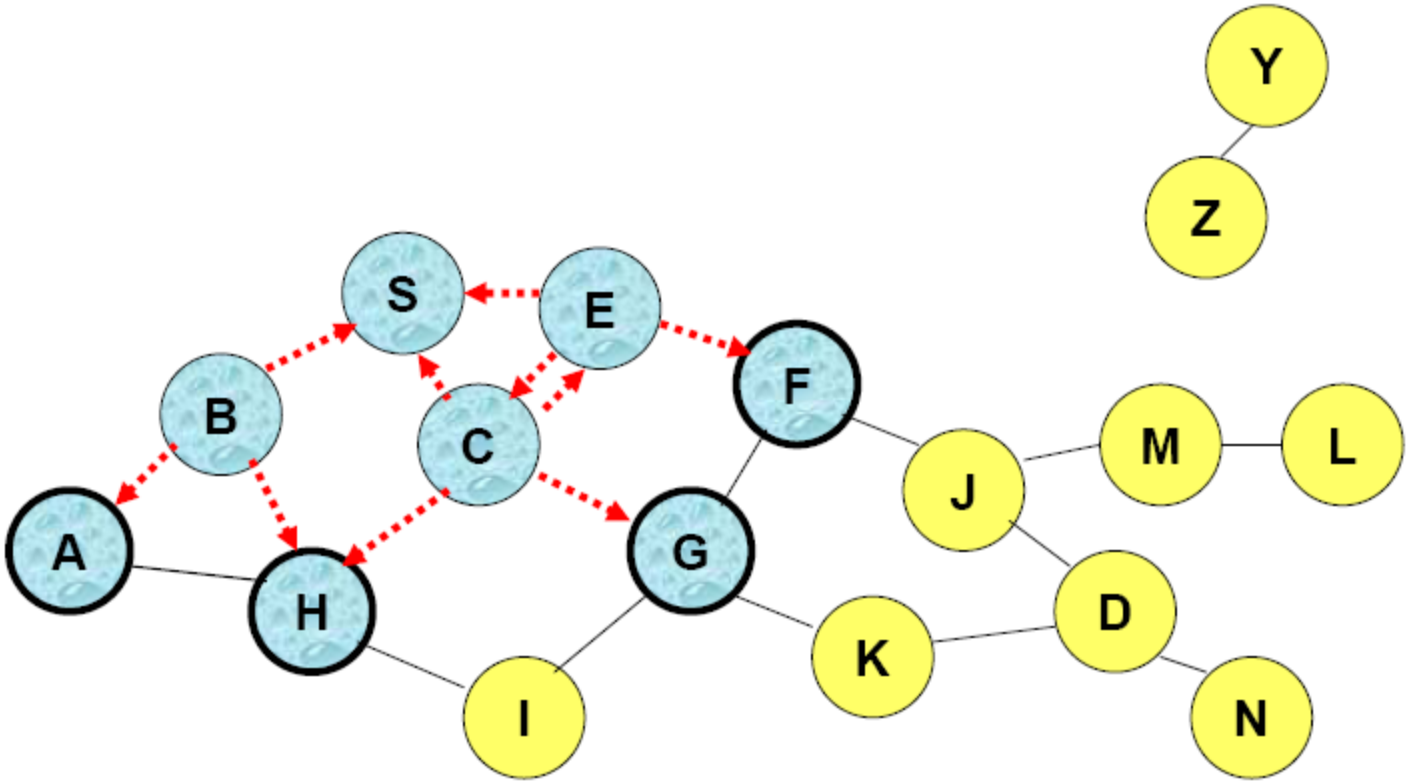

Represents a node that has received packet P

Represents that connected nodes are within each other's transmission range

# Flooding for Control Packet Delivery: Example

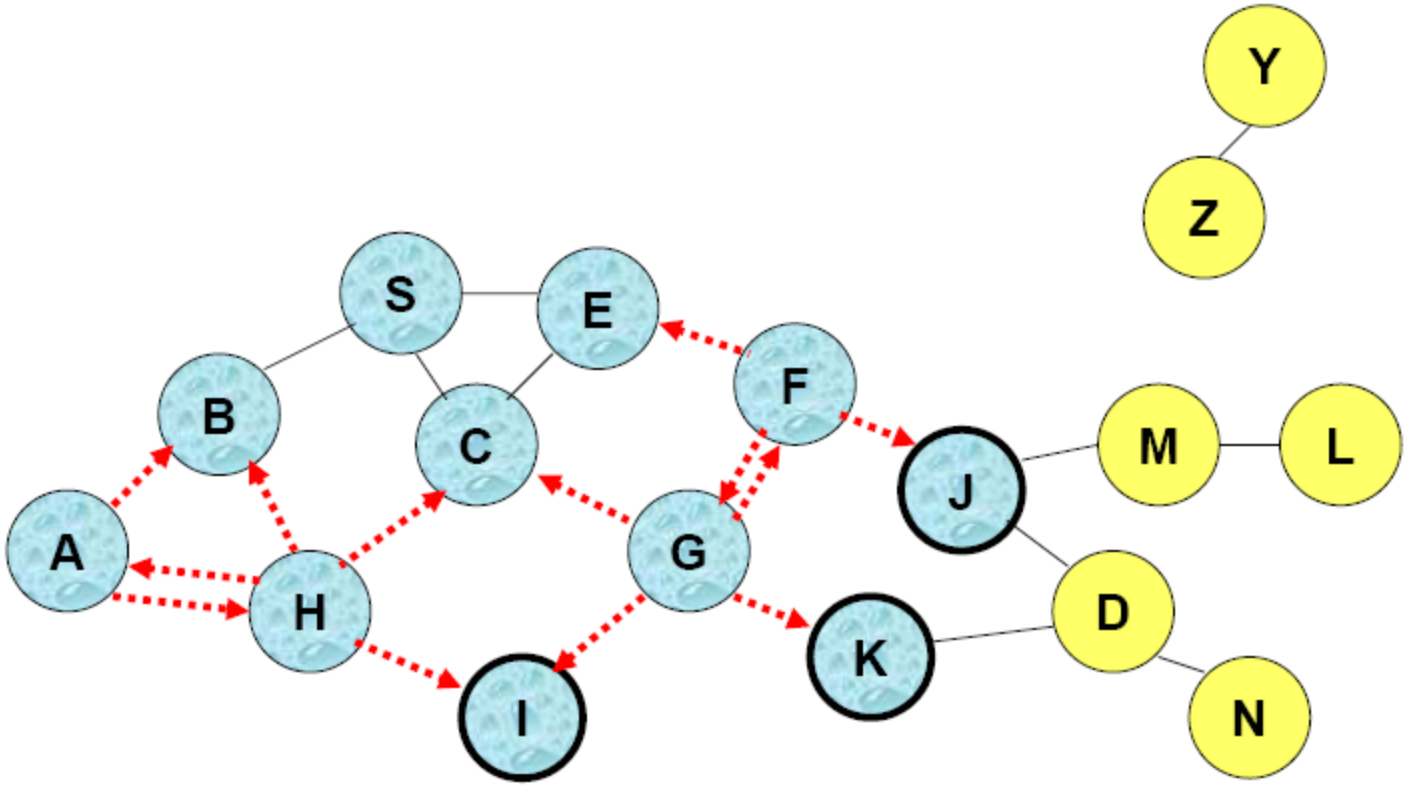


# Flooding for Control Packet Delivery: Example



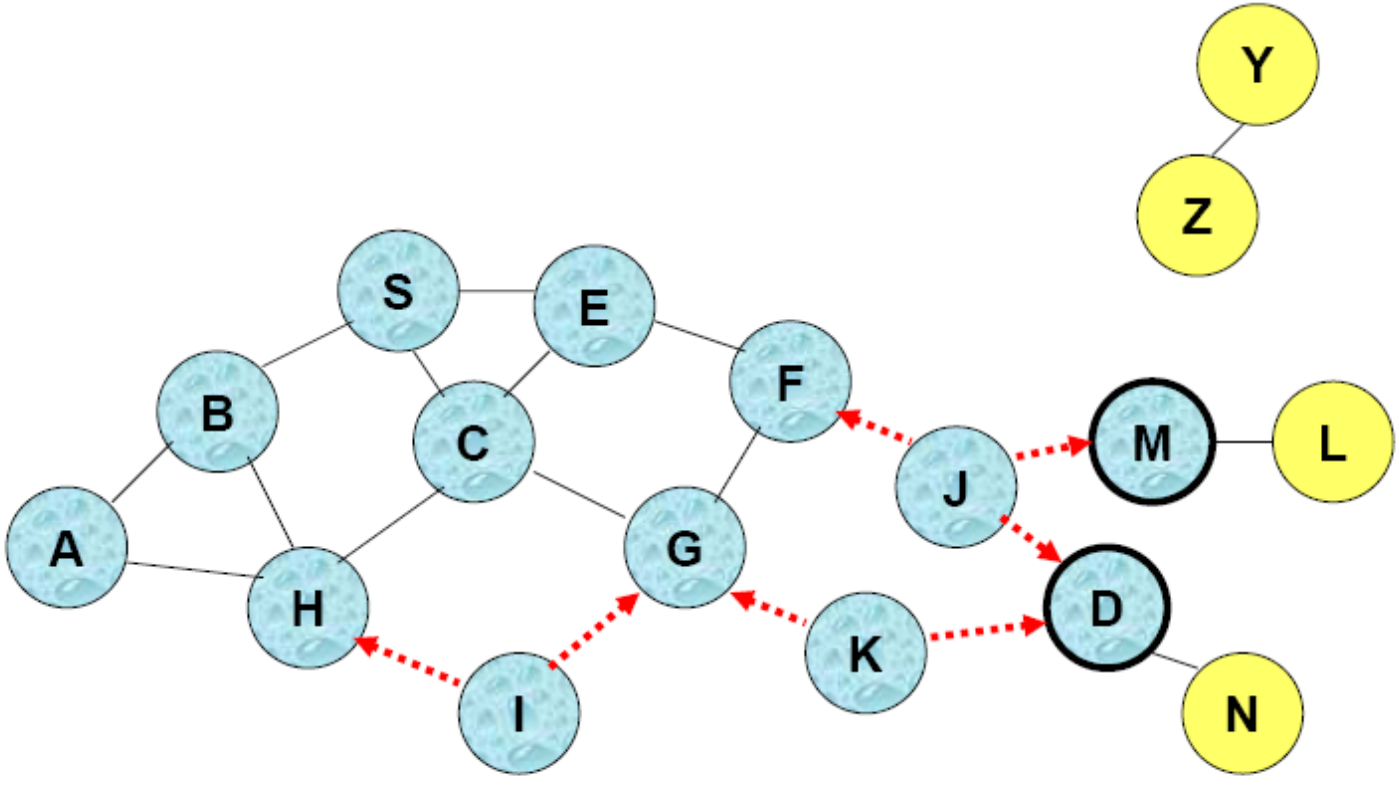

- Node H receives packet P from two neighbors: potential for collision

# Flooding for Control Packet Delivery: Example



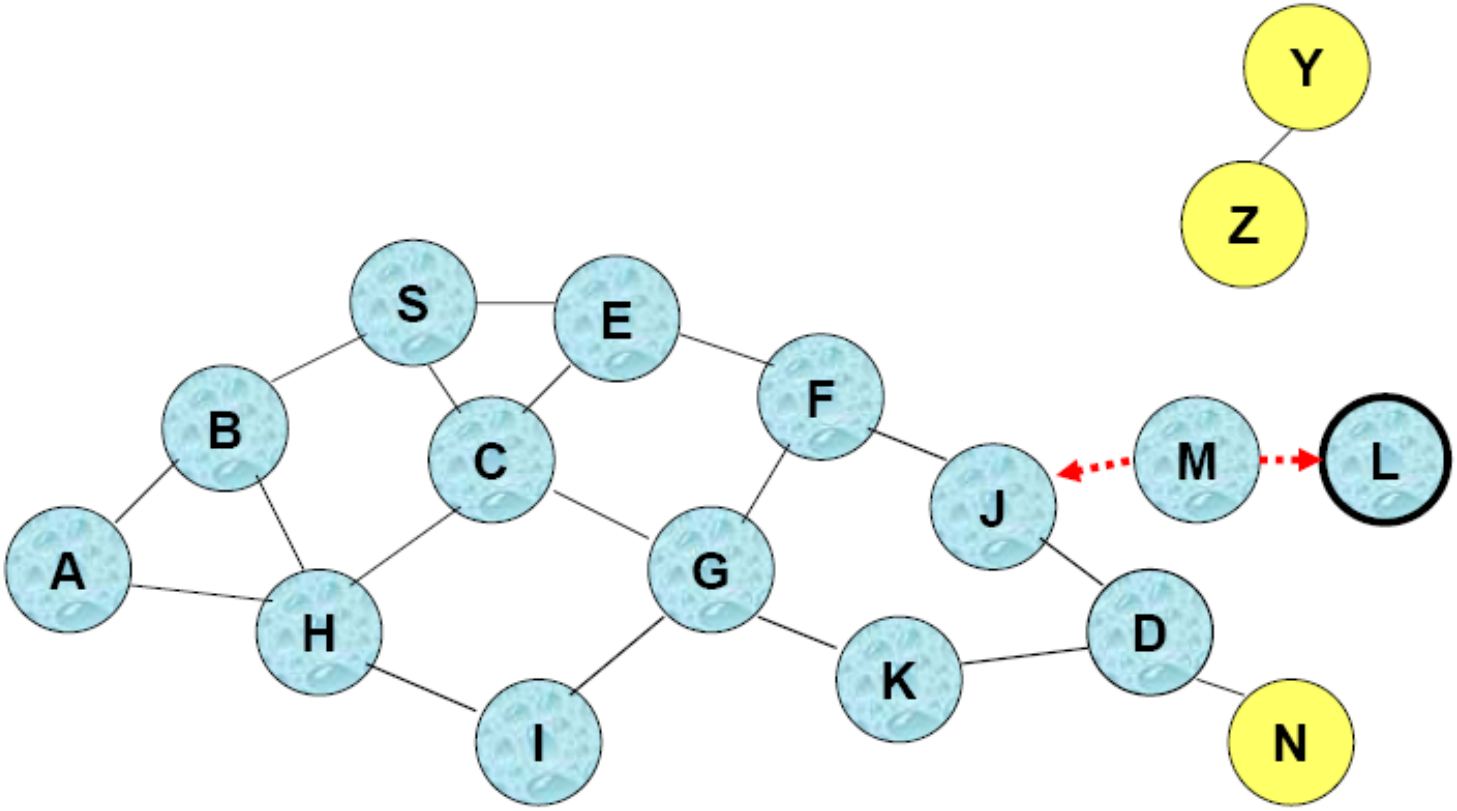

- Node C receives packet P from G and H, but does not forward it again, because node C has already forwarded packet P once

# Flooding for Control Packet Delivery: Example



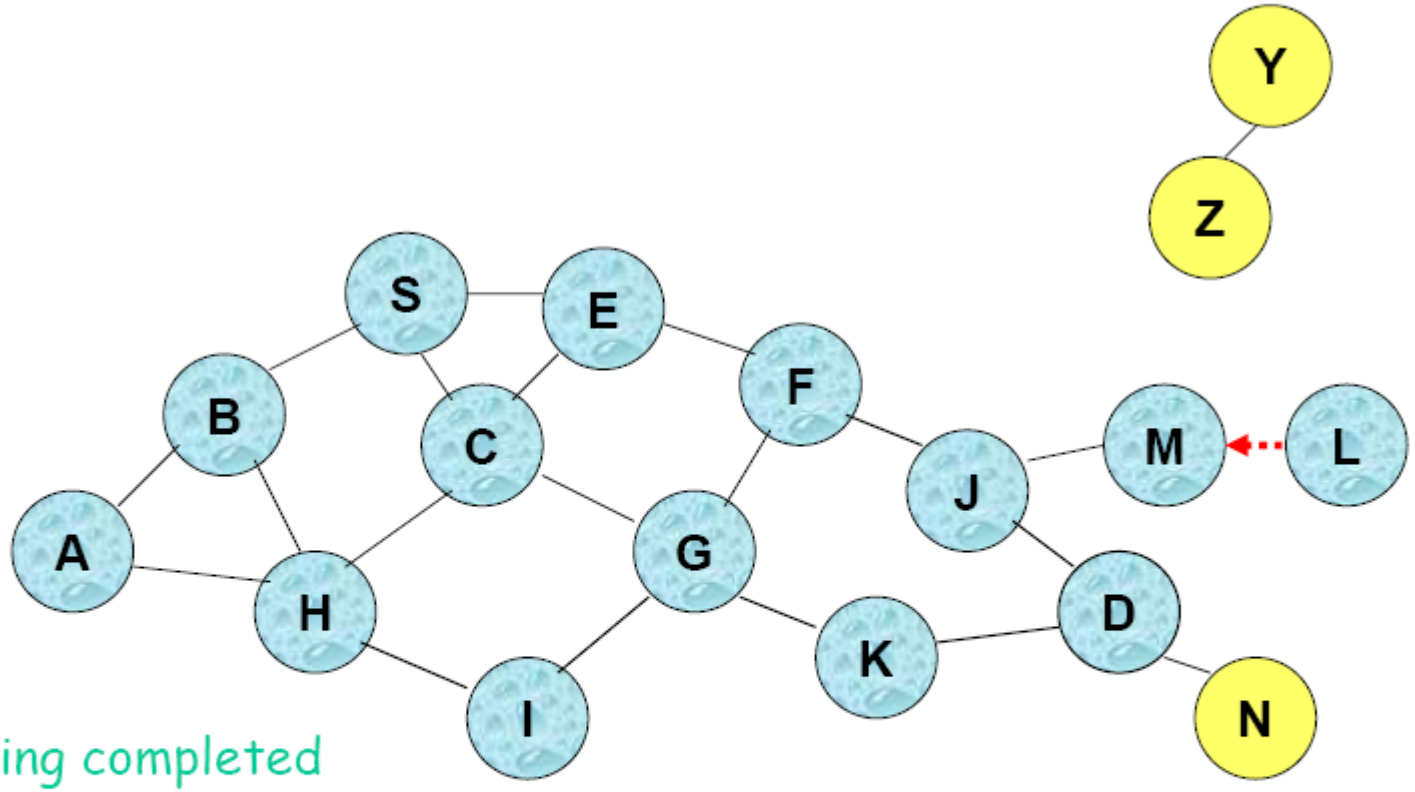

- Nodes J and K both broadcast packet P to node D
- Since nodes J and K are hidden from each other, their transmissions may collide
  - => Packet P may not be delivered to node D at all, despite the use of flooding

# Flooding for Control Packet Delivery: Example



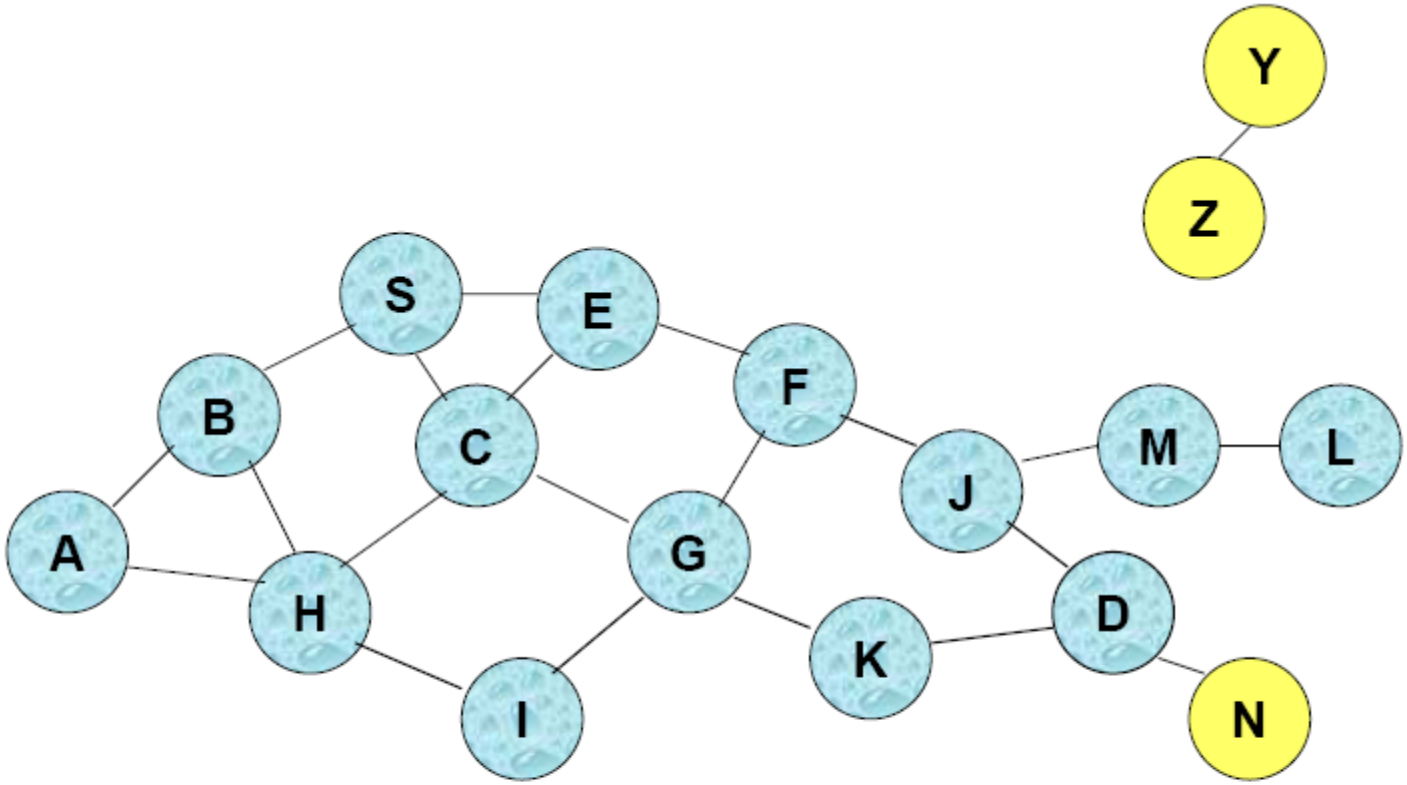

- Node D does not forward packet P, because node D is the intended destination of packet P

# Flooding for Control Packet Delivery: Example




- Flooding completed
- Nodes unreachable from S do not receive packet P (e.g., node Z)
- Nodes for which paths go through the destination D also do not receive packet P (example: node N)

# Flooding for Control Packet Delivery: Example




- Flooding may deliver packets to too many nodes (in the worst case, all nodes reachable from sender may receive the packet)

# Flooding - Advantages

- Simplicity
- May be more efficient than other protocols when rate of information transmission is low enough that the overhead of explicit route discovery and maintenance incurred by other protocols is relatively higher
  - this scenario may occur, for instance, when nodes transmit small data packets relatively infrequently, and many topology changes occur between consecutive packet transmissions
- Potentially higher reliability of data delivery
  - Because packets may be delivered to the destination on multiple paths

# Flooding - Disadvantages

- Potentially, very high overhead
  - Data packets may be delivered to too many nodes who do not need to receive them
- Potentially lower reliability of data delivery
  - Flooding uses broadcasting -- hard to implement reliable broadcast delivery without significantly increasing overhead
    - Broadcasting in IEEE 802.11 MAC is unreliable
  - In this example, nodes J and K may transmit to node D simultaneously, resulting in loss of the packet
    - in this case, destination would not receive the packet at all

# Dynamic Source Routing (DSR)

- Similar to AODV in route discovery
- Full source-route is aggregated in RREQ and sent back in RREP
- Each data packet has full source route

- Route table overhead only at source node
- However, overhead with each data packet

# Dynamic Source Routing (DSR)

- On Demand Protocol
- Nodes maintain route caches
- Route discovery
  - Sender initiates request
  - Intermediate nodes add their address onto request
  - When request reaches destination, it includes the full path.

# DSR vs. AODV

- DSR includes source routes in packet headers
- Resulting large headers can sometimes degrade performance
  - particularly when data contents of a packet are small
- AODV attempts to improve on DSR by maintaining routing tables at the nodes, so that data packets do not have to contain routes
- AODV retains the desirable feature of DSR that routes are maintained only between nodes which need to communicate
- DSR route cache entries do not have lifetimes (at present, only proposed); AODV route table entries do have lifetimes

# Associativity-Based Routing

- Defines metric "Degree of Association Stability"
  - This metric used instead of shortest hop
- Nodes with less mobility/better links have higher stability value

- DSR-like protocol is used for routing

# Signal Stability Routing

- Signal strength of links is used as metric
- DSR-like routing is used
- RREQ is forwarded only if packet is received over a link with good signal strength

# Other metrics

- Expected Transmission Time (ETT) metric
  - Easier to compute, and more useful than signal strength
- Weighted Cumulative Expected Transmission Time
  - Better for multi-radio, and asymmetric rate links

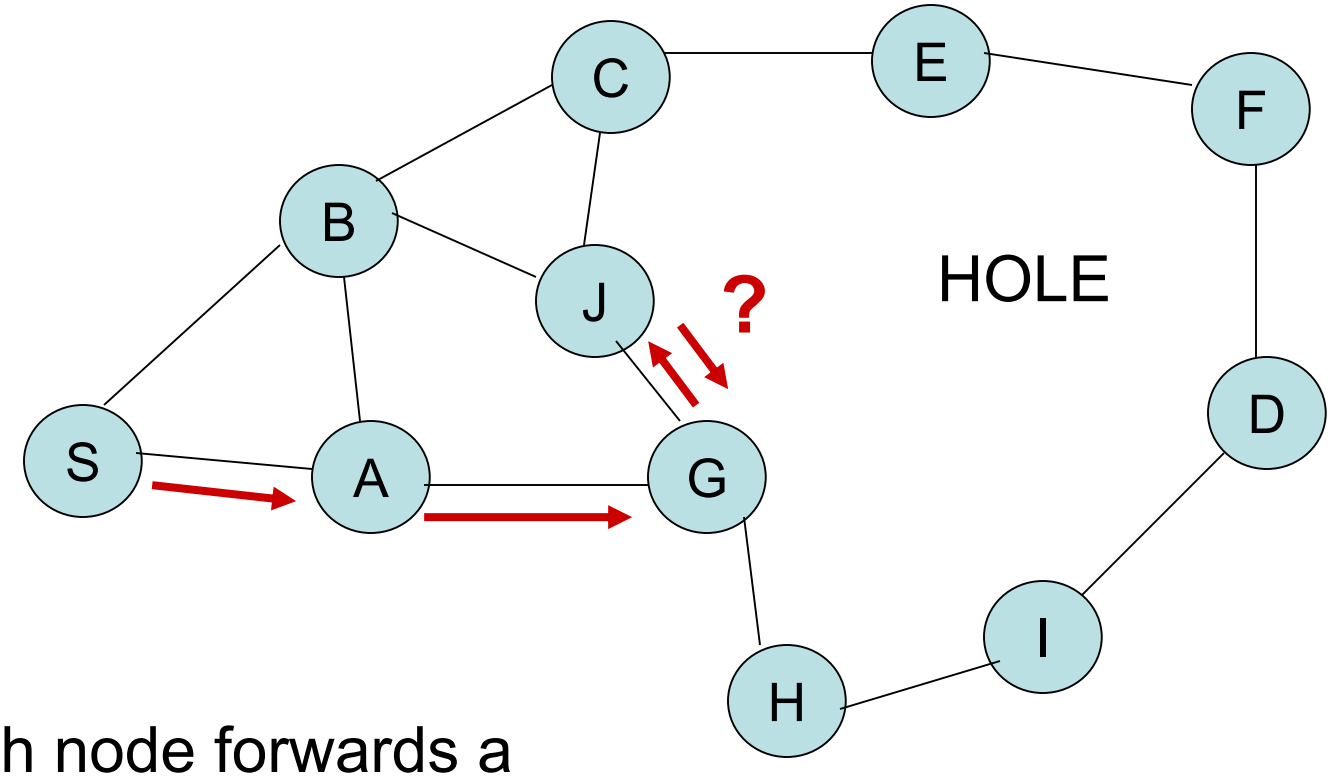
# Other routing protocols

- Geographic Routing Protocols
  - Location Aided Routing (LAR)
  - Distance Routing Effect Algorithm for Mobility (DREAM)
  - Greedy Perimeter Stateless Routing (GPSR)
- Hybrid Routing Protocols
  - Zone Routing Protocol (ZRP)

# Discussion

- Proactive routing protocols suitable for high traffic load, low mobility
- On-demand routing protocols suitable for low traffic load and/or moderate mobility
- With high mobility, flooding of data packets may be the only option

# Problem: Locating and Bypassing Routing Holes in Sensor Networks

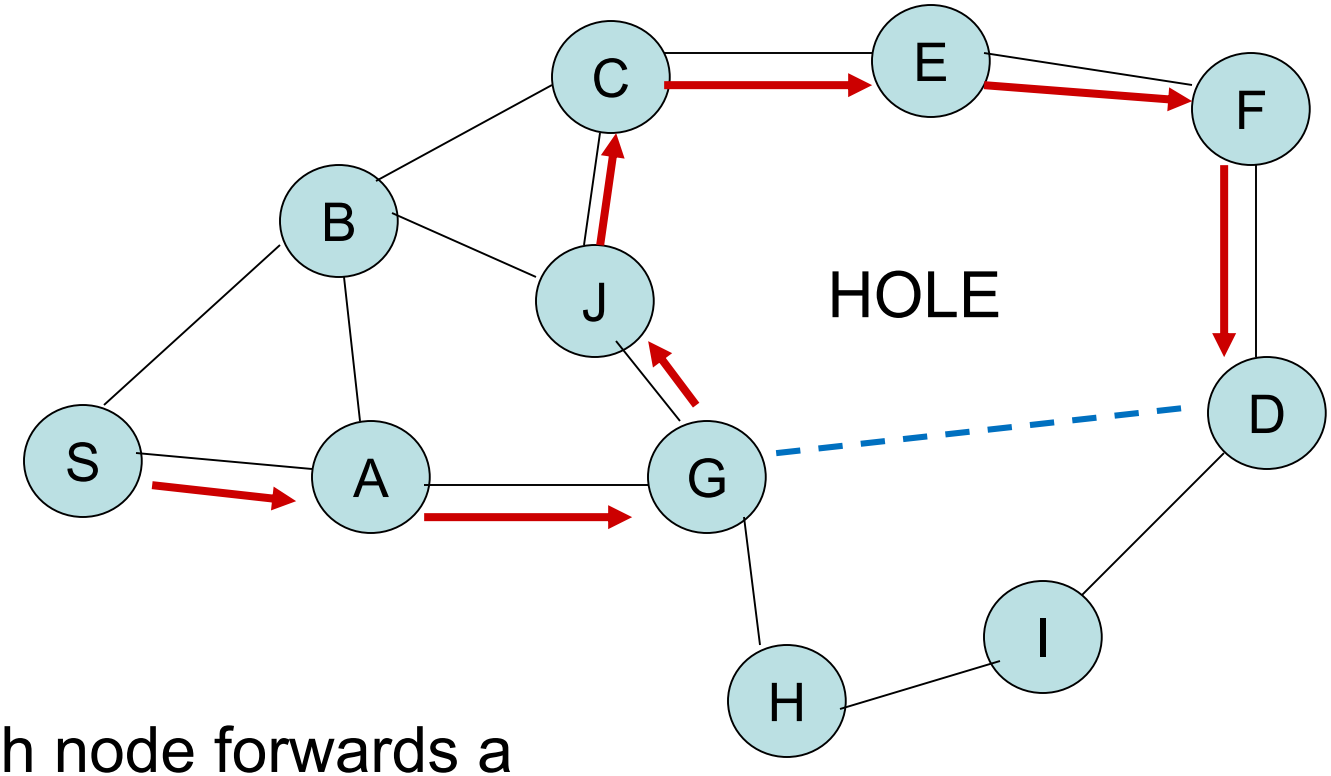



Each node forwards a packet to its neighbor closest to the destination

# Greedy Perimeter Stateless Routing (GPSR)

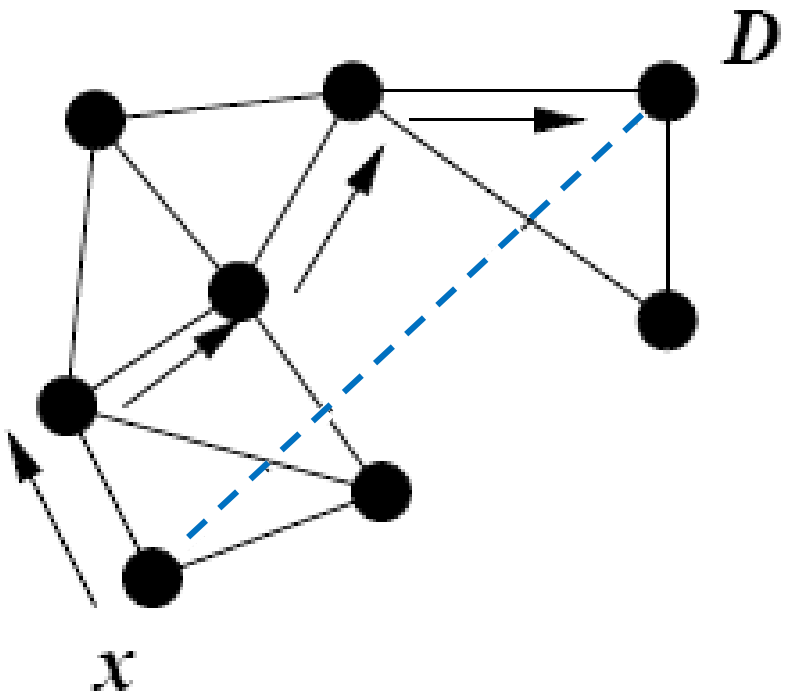
- Location of the destination node is assumed to be known
- Each node knows location of its neighbors
- Each node forwards a packet to its neighbor closest to the destination
- If routing holes are found, uses perimeter routing (right-hand rule)

# Problem: Locating and Bypassing Routing Holes in Sensor Networks



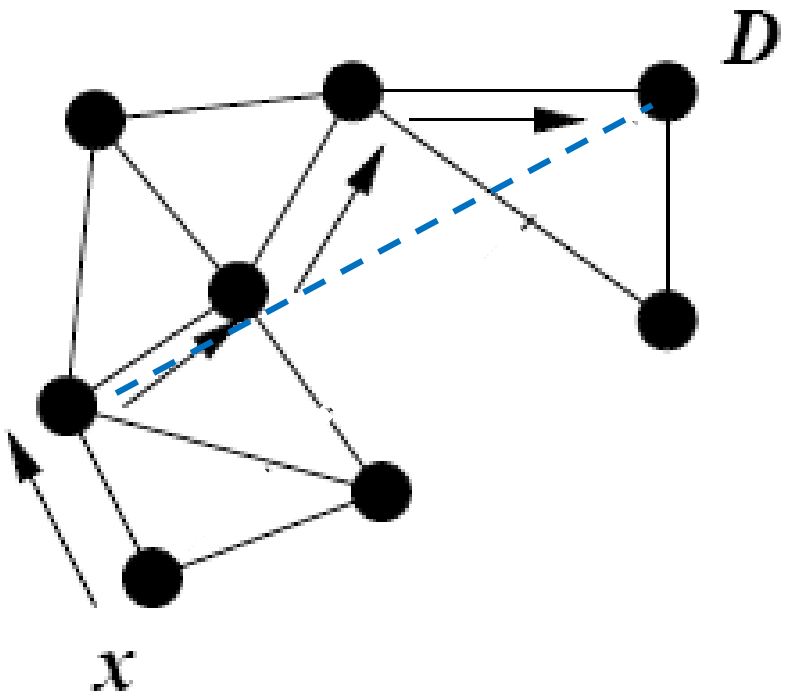

Each node forwards a packet to its neighbor closest to the destination

# Example of Right Hand Rule



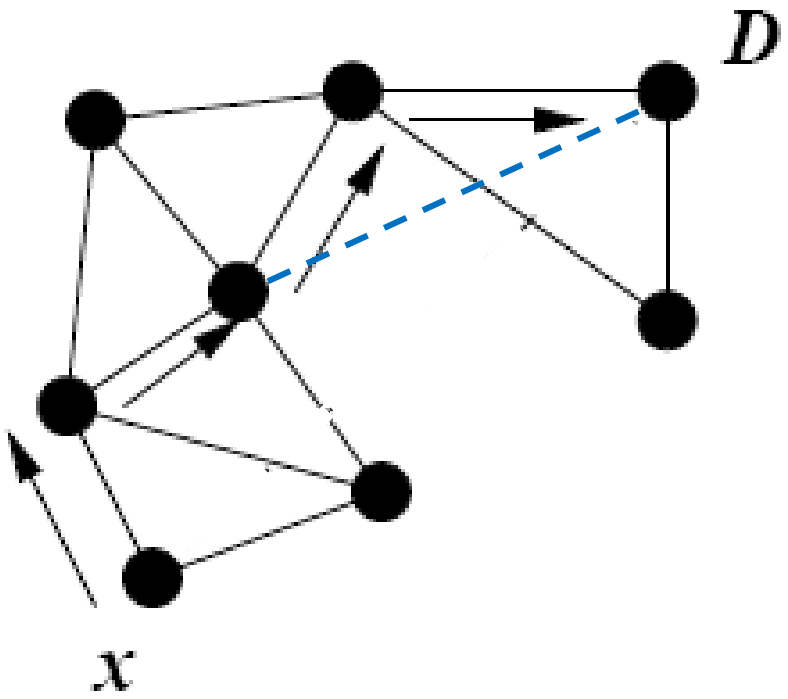

Each node forwards to the first node reachable counterclockwise with respect to a line to the destination

# Example of Right Hand Rule




Each node forwards to the first node reachable counterclockwise with respect to a line to the destination

# Example of Right Hand Rule




Each node forwards to the first node reachable counterclockwise with respect to a line to the destination