Overview of Unicast Routing Protocols for Multihop Wireless Networks

Prof. Alhussein Abouzeid

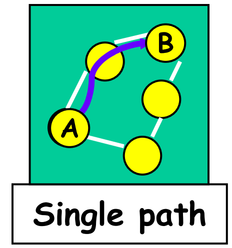
Rensselaer Polytechnic Institute

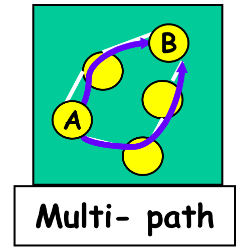
Acknowledgment: DSR & AODV Slides are based in part on Nitin Vaidya's tutorial slides

Overview of Unicast Routing Protocols for Multihop Wireless Networks

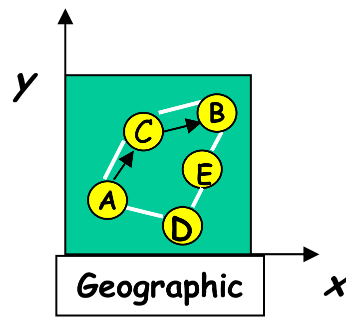
Classes of Routing Protocols

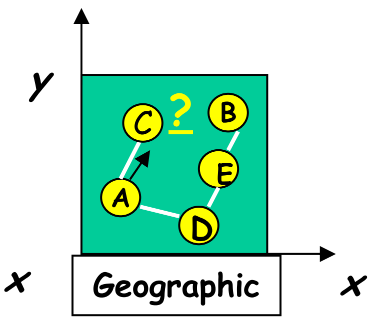
- Single Path vs. Multi-Path
  - Shortest delay
  - Largest available bandwidth
  - Largest residual power
  - Large expected lifetime



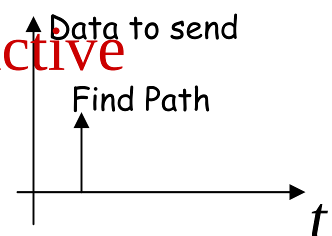


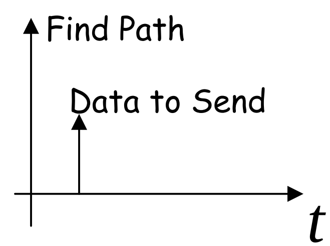
- Unicast vs. Multicast
- Statefull vs. “Stateless” (GPS)



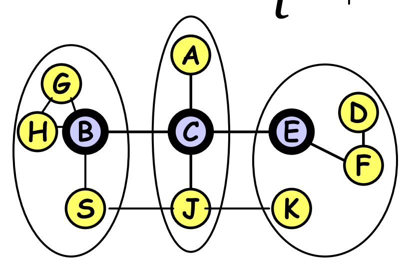


- On-Demand (Reactive) vs. Proactive





- Flat vs. Hierarchical



Routing: from wired to wireless

- Ad-hoc networks have dynamic time-dependent topology
  - Arcs (Edges) added/deleted
  - Nodes (Vertices) addition/deletion
  - Bi-directional or uni-directional links

- Wireless medium is different from wireless
  - It is inherently a broadcast medium
  - Fading, shadowing cause burst errors and/or intermittent connnectivity

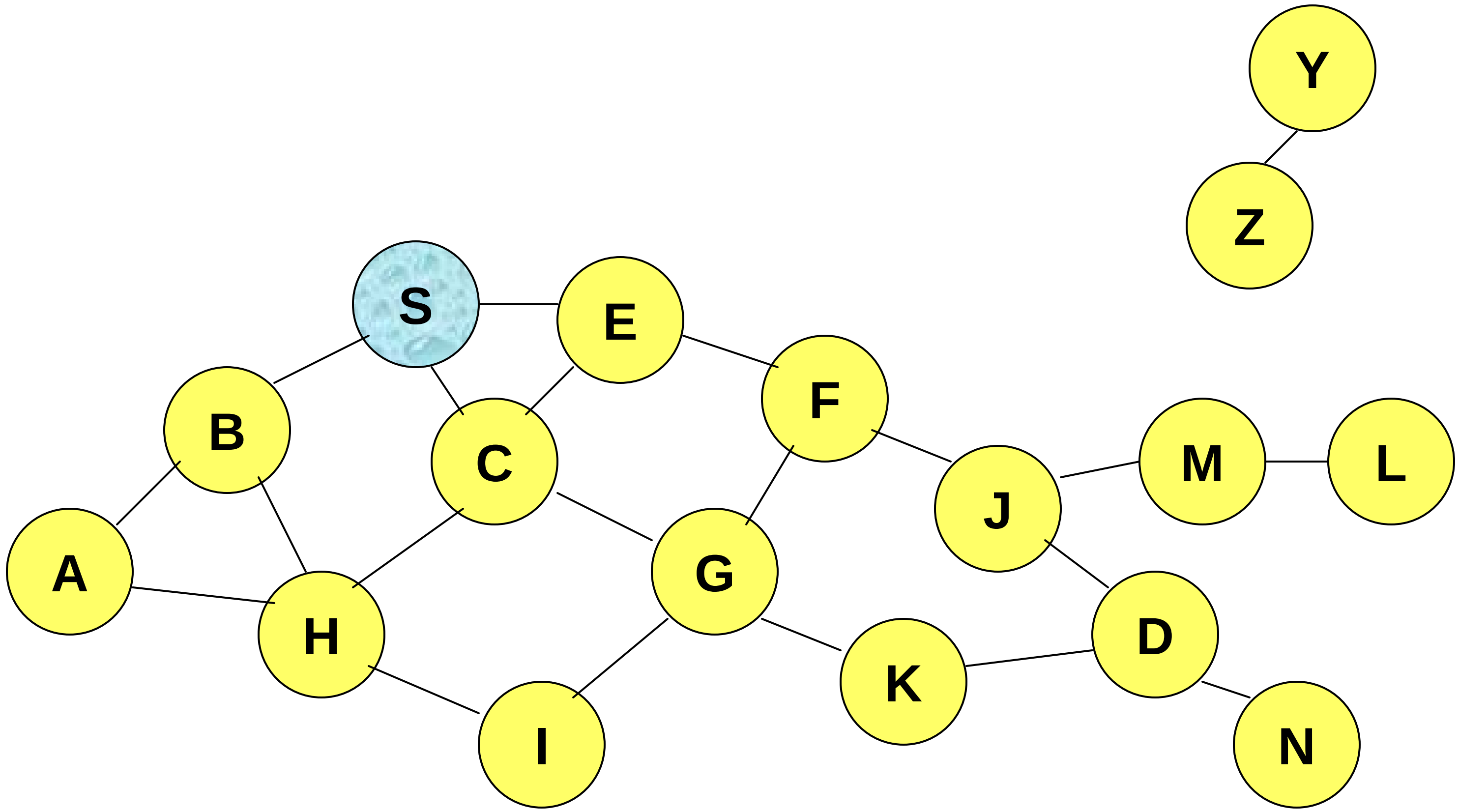
➔Need to update (rediscover) the network topology

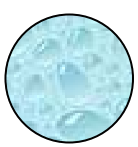
Flooding

Flooding for Data Delivery

- Sender S broadcasts data packet P to all its neighbors
- Each node receiving P forwards P to its neighbors
- Sequence numbers used 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 Data Delivery-a wireless example

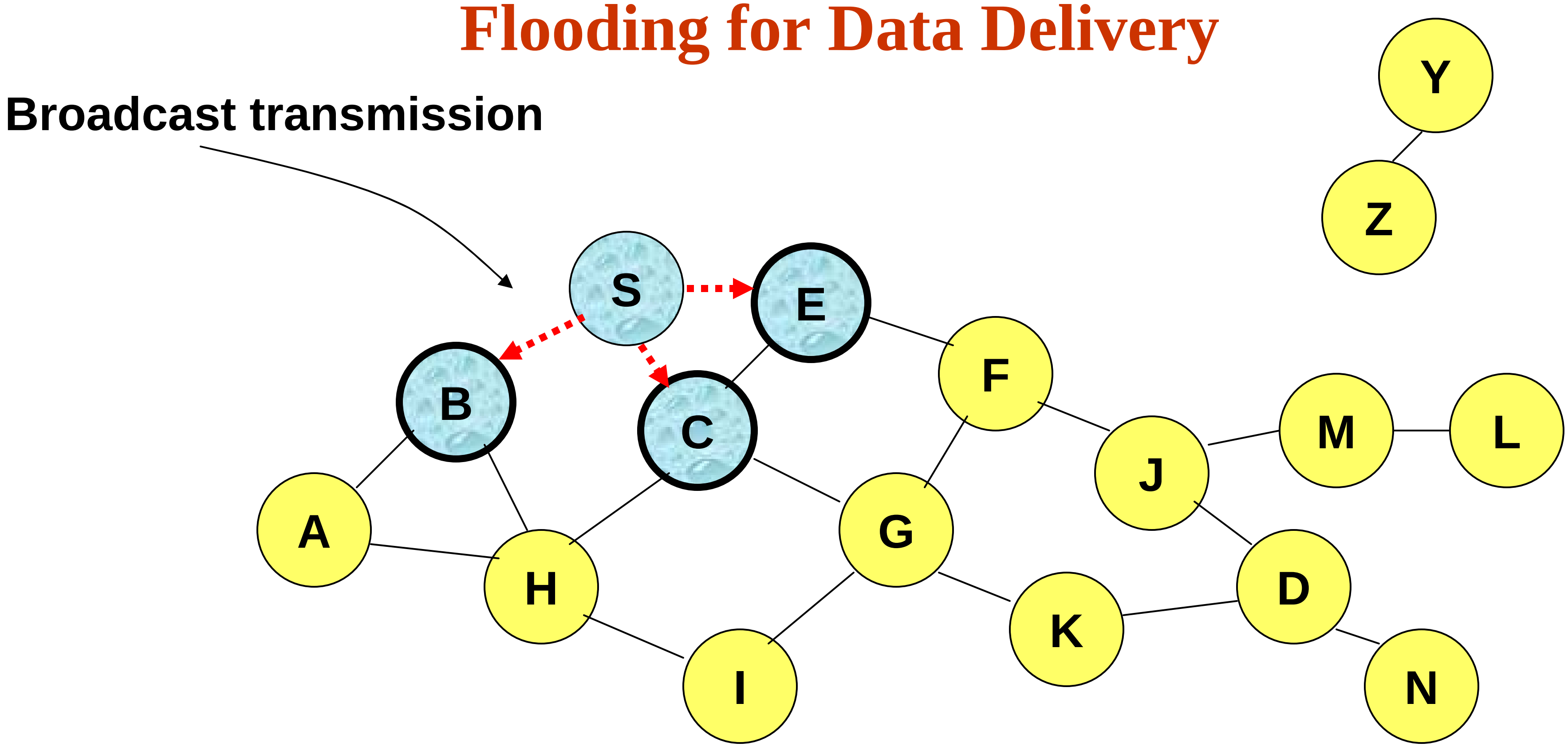




Represents a node that has received packet P

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

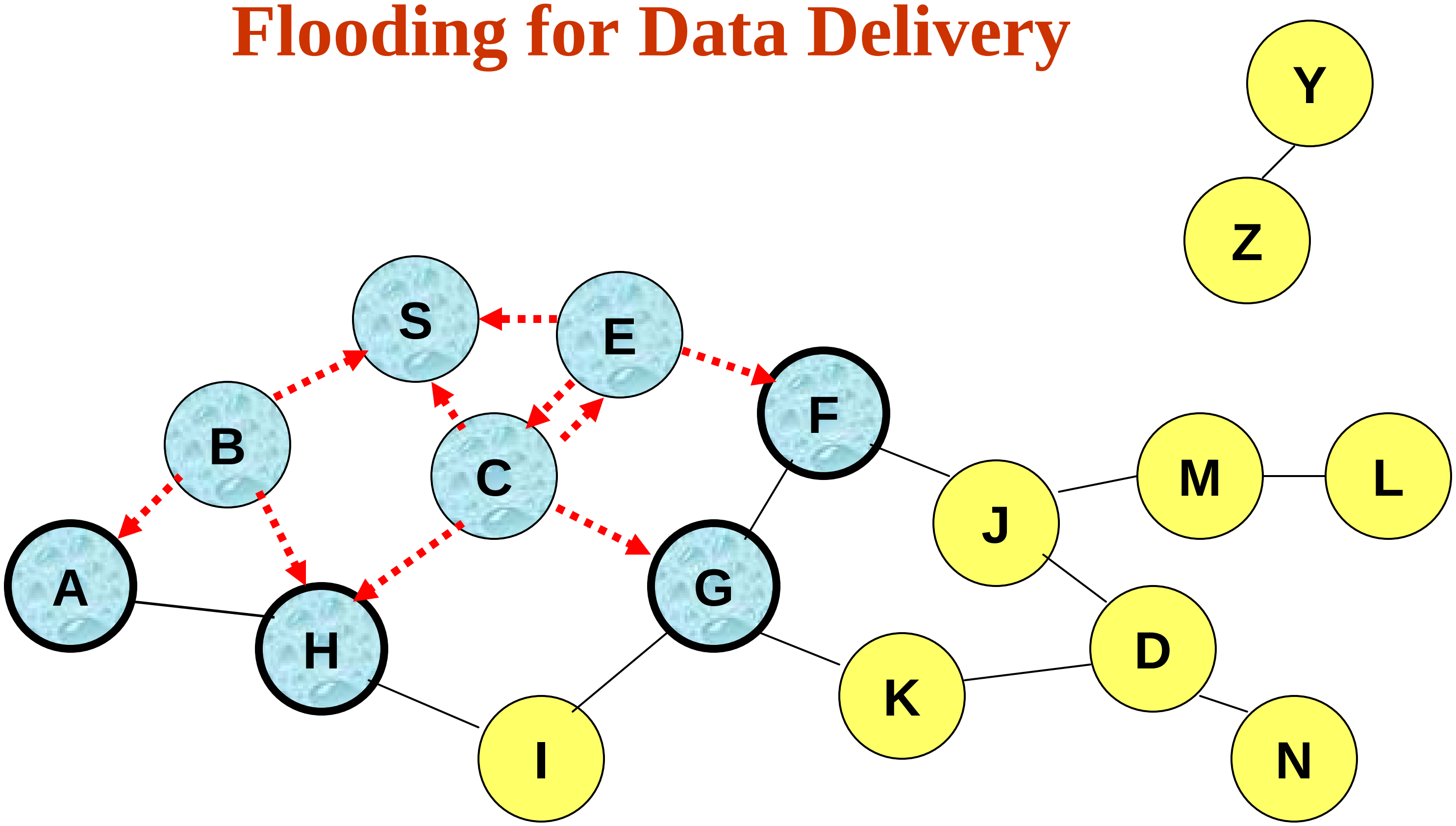
Flooding for Data Delivery



Represents a node that receives packet P for the first time

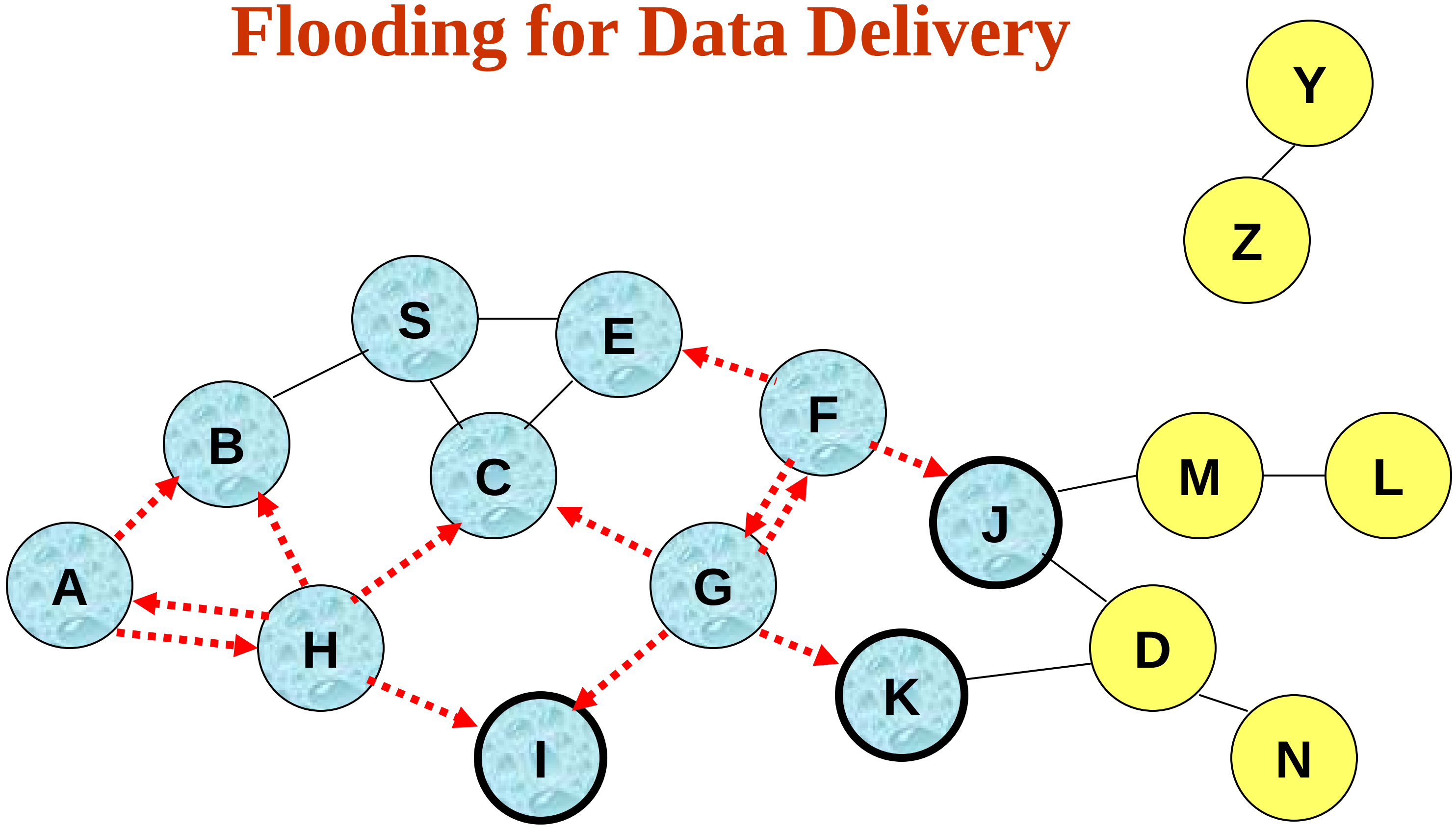
Represents transmission of packet P

Flooding for Data Delivery



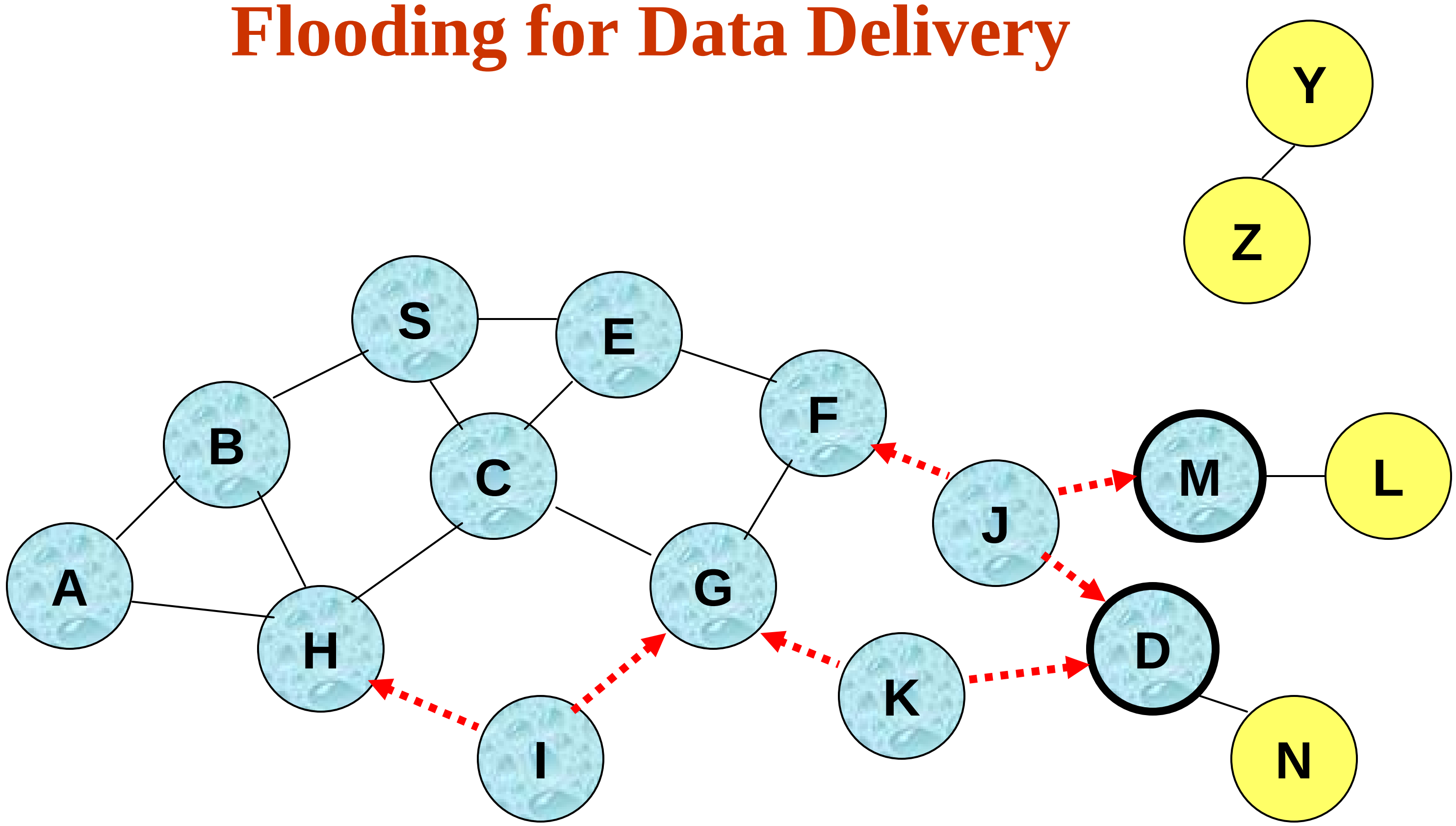
- **Node H receives packet P from two neighbors: potential for collision**

Flooding for Data Delivery



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

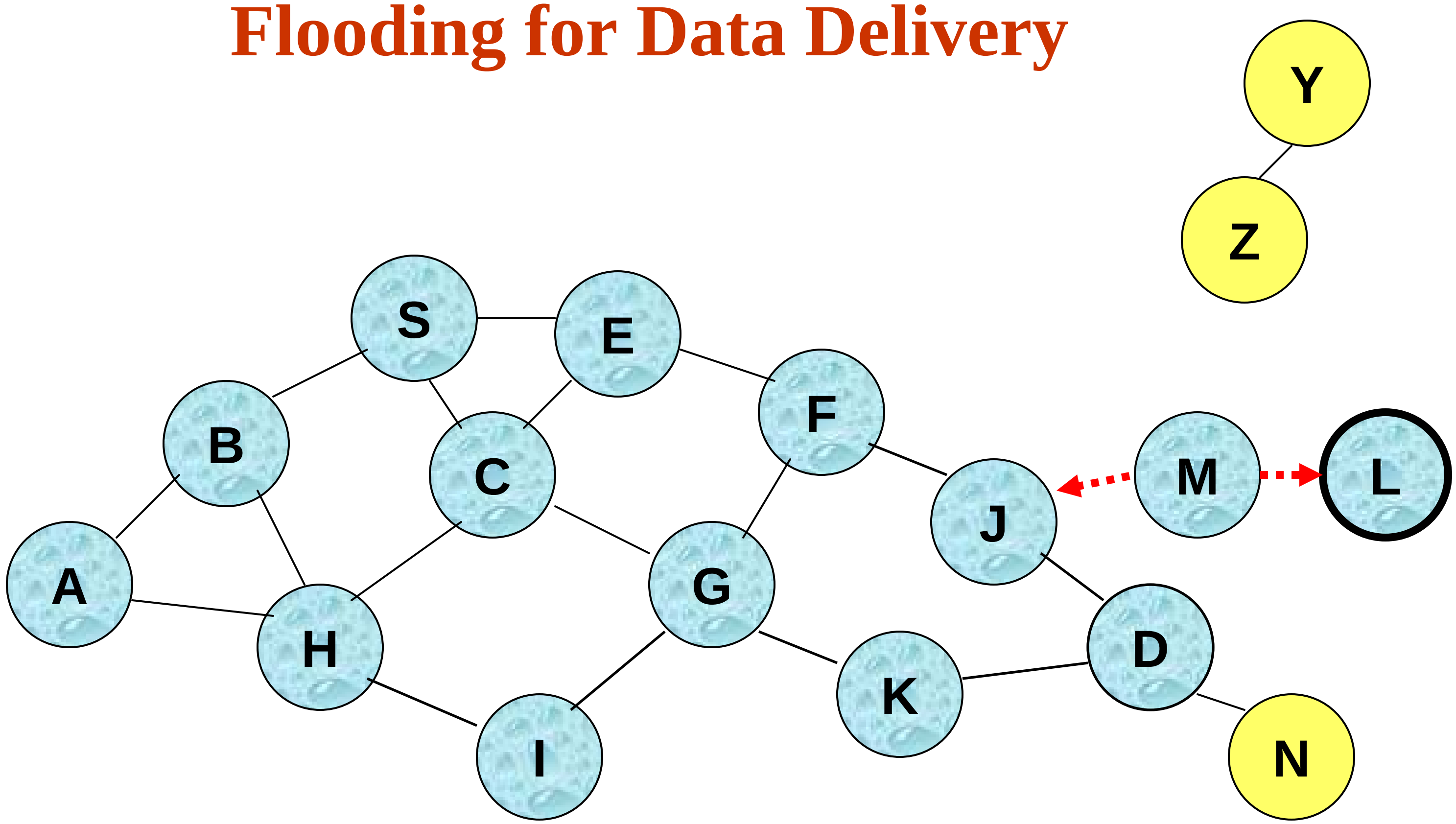
Flooding for Data Delivery



- **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**

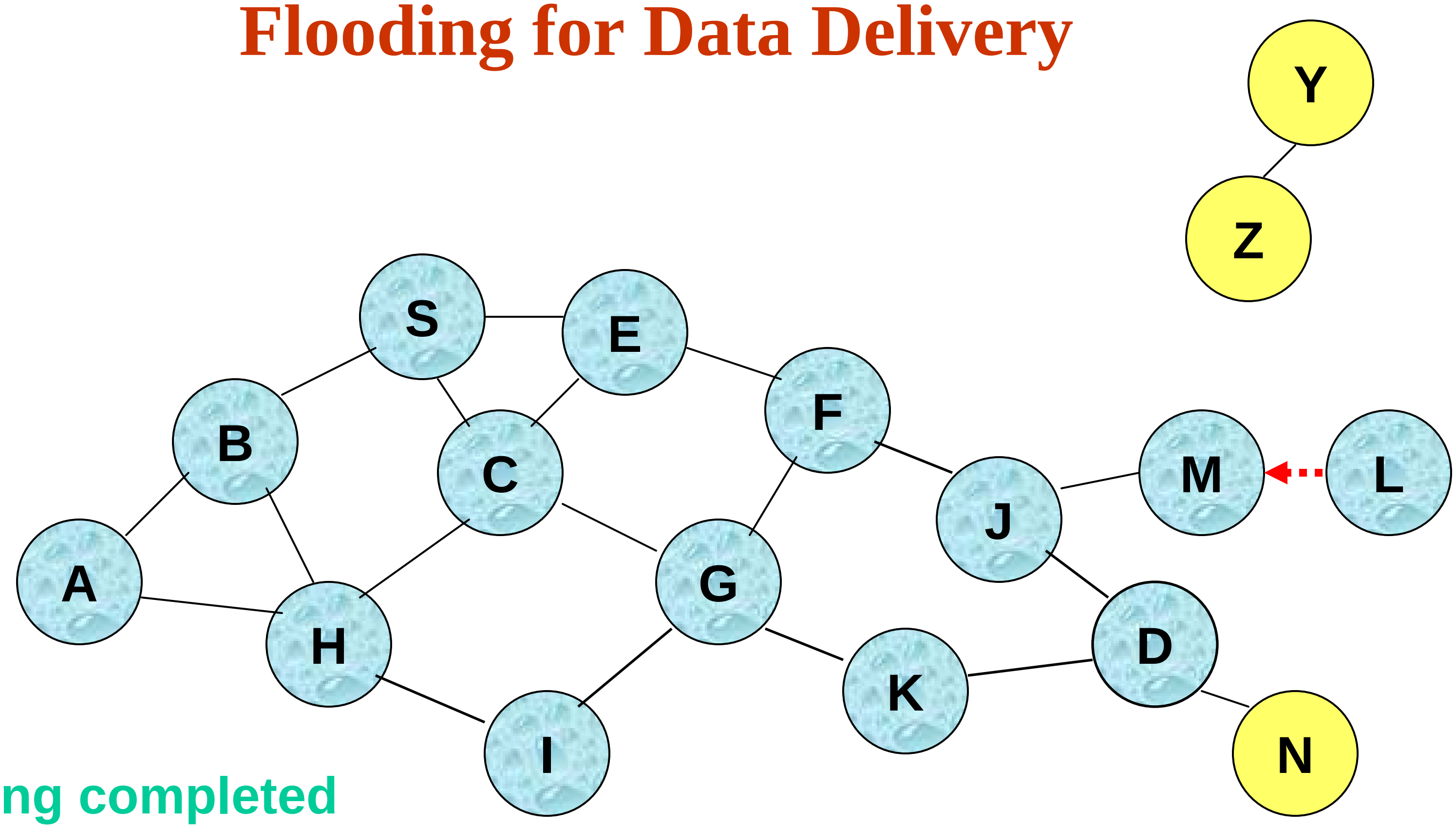
  => Packet P may not be delivered to node D at all, despite the use of flooding

Flooding for Data Delivery



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

Flooding for Data Delivery

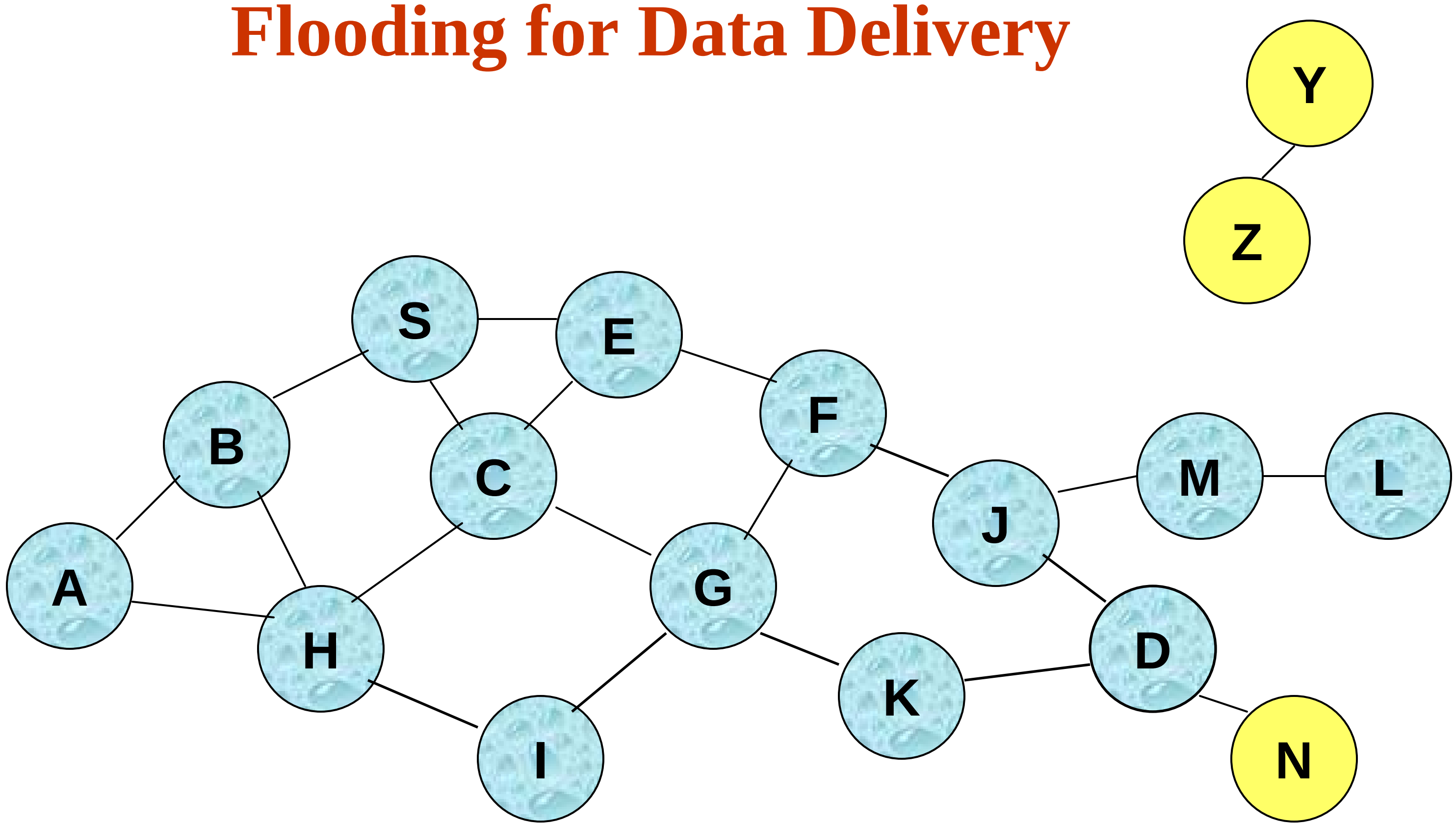


- Flooding completed

- Nodes unreachable from S do not receive packet P (e.g., node Z)

- Nodes for which all paths from S go through the destination D also do not receive packet P (example: node N)

Flooding for Data Delivery



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

Flooding for Data Delivery: Advantages

- Simplicity
- Efficient for:
  - Low information exchange rate
  - High mobility

    In these cases, the overhead of explicit route discovery/maintenance incurred by other protocols may be relatively higher e.g. nodes transmit small data packets relatively infrequently, and 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 for Data Delivery: 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 (or higher delay)
  - Flooding uses broadcasting -- hard to implement reliable broadcast delivery without significantly increasing overhead
    - Broadcasting in IEEE 802.11 MAC is unreliable
  - In our 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

Flooding of Control Packets

- Many protocols perform (potentially *limited*) flooding of control packets, instead of data packets

- The control packets are used to discover routes

- Discovered routes are subsequently used to send data packet(s)

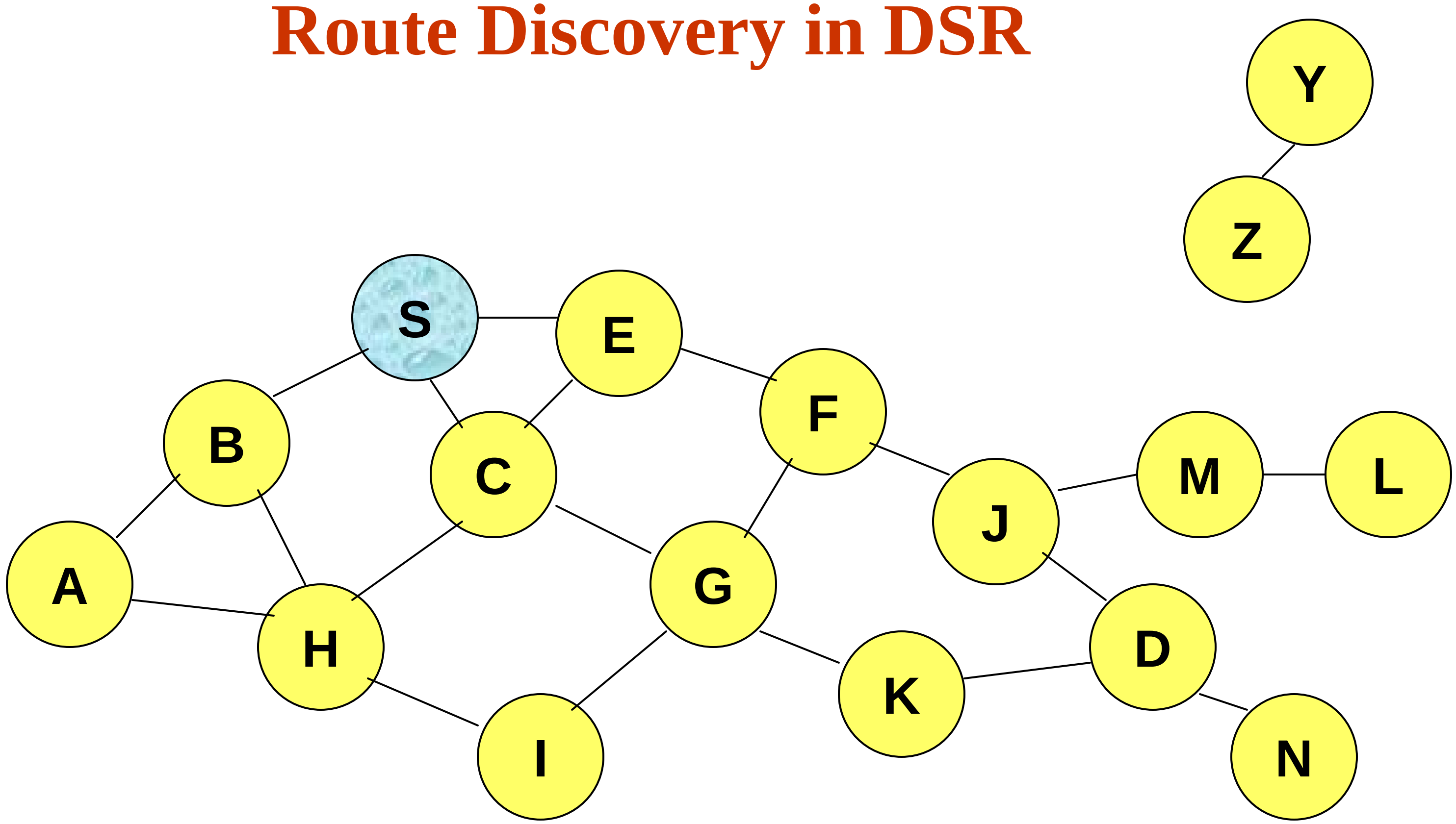
- Overhead of control packet flooding is amortized over data packets transmitted between consecutive control packet floods

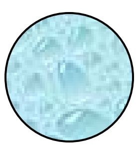
Dynamic Source Routing (DSR) [Johnson96]

Dynamic Source Routing (DSR) [Johnson96]

- When node S wants to send a packet to node D, but does not know a route to D, node S initiates a route discovery
- Source node S floods Route Request (RREQ)
- Each node appends own identifier when forwarding RREQ

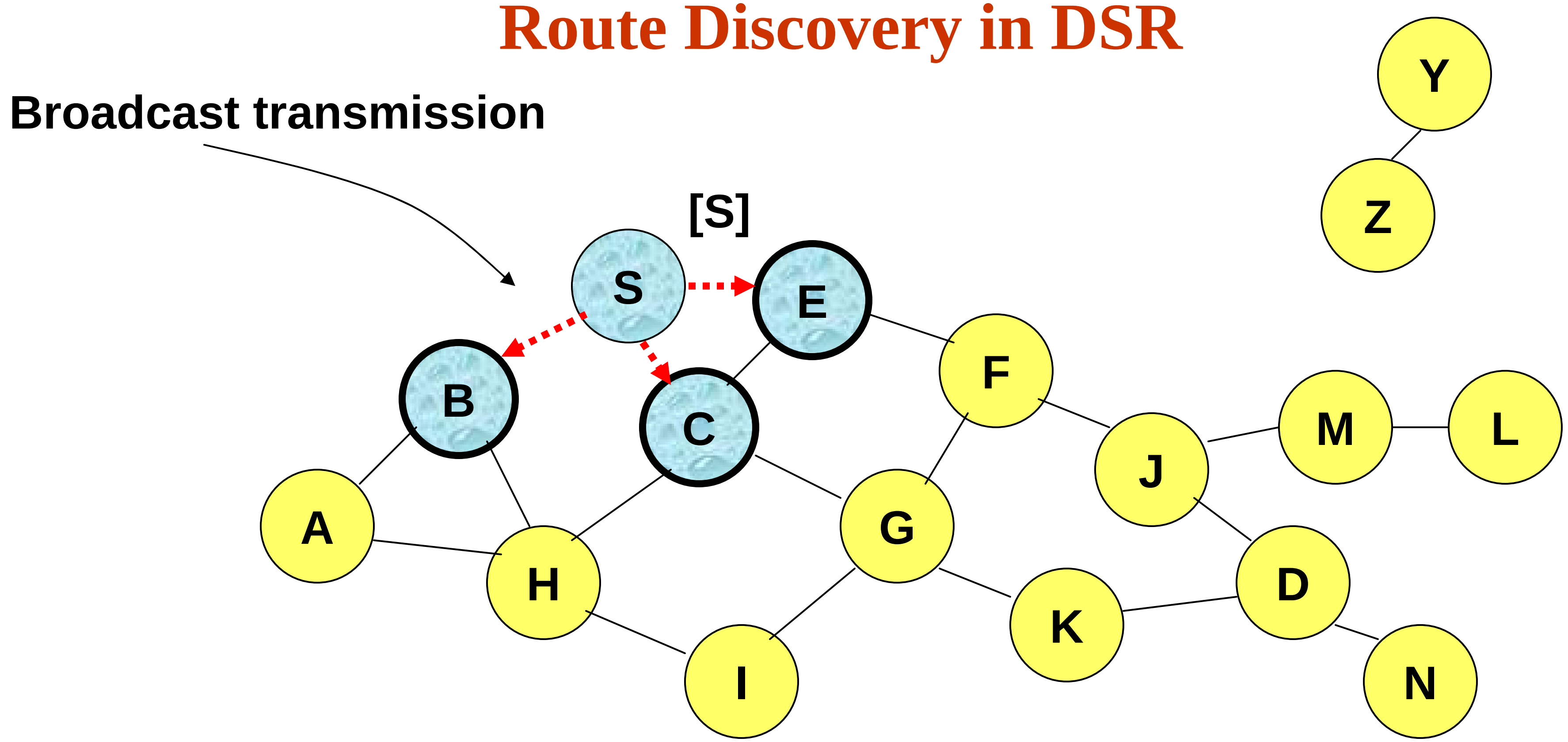
Route Discovery in DSR





Represents a node that has received RREQ for D from S

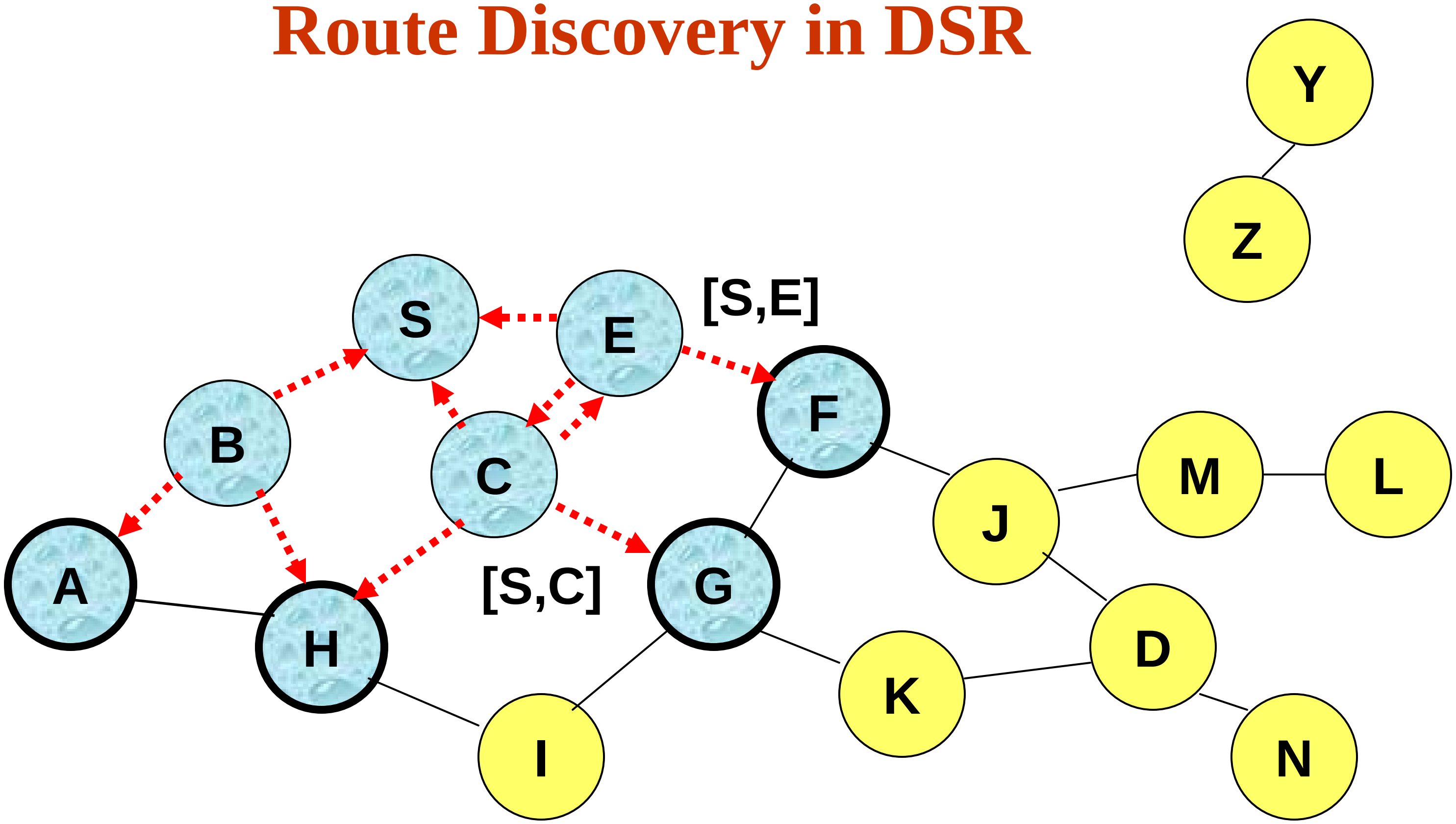
Route Discovery in DSR



Represents transmission of RREQ

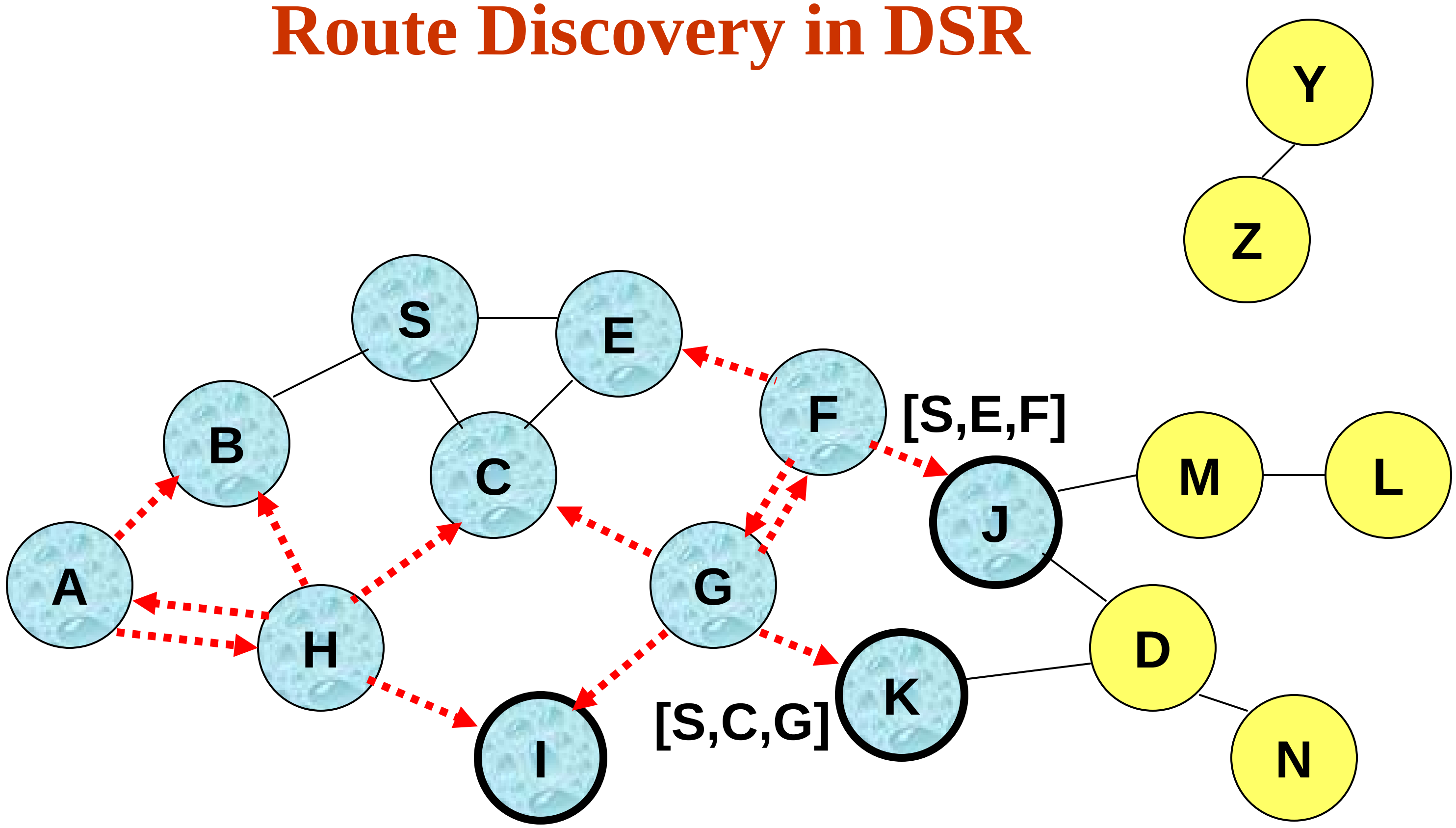
[X,Y] Represents list of identifiers appended to RREQ

Route Discovery in DSR



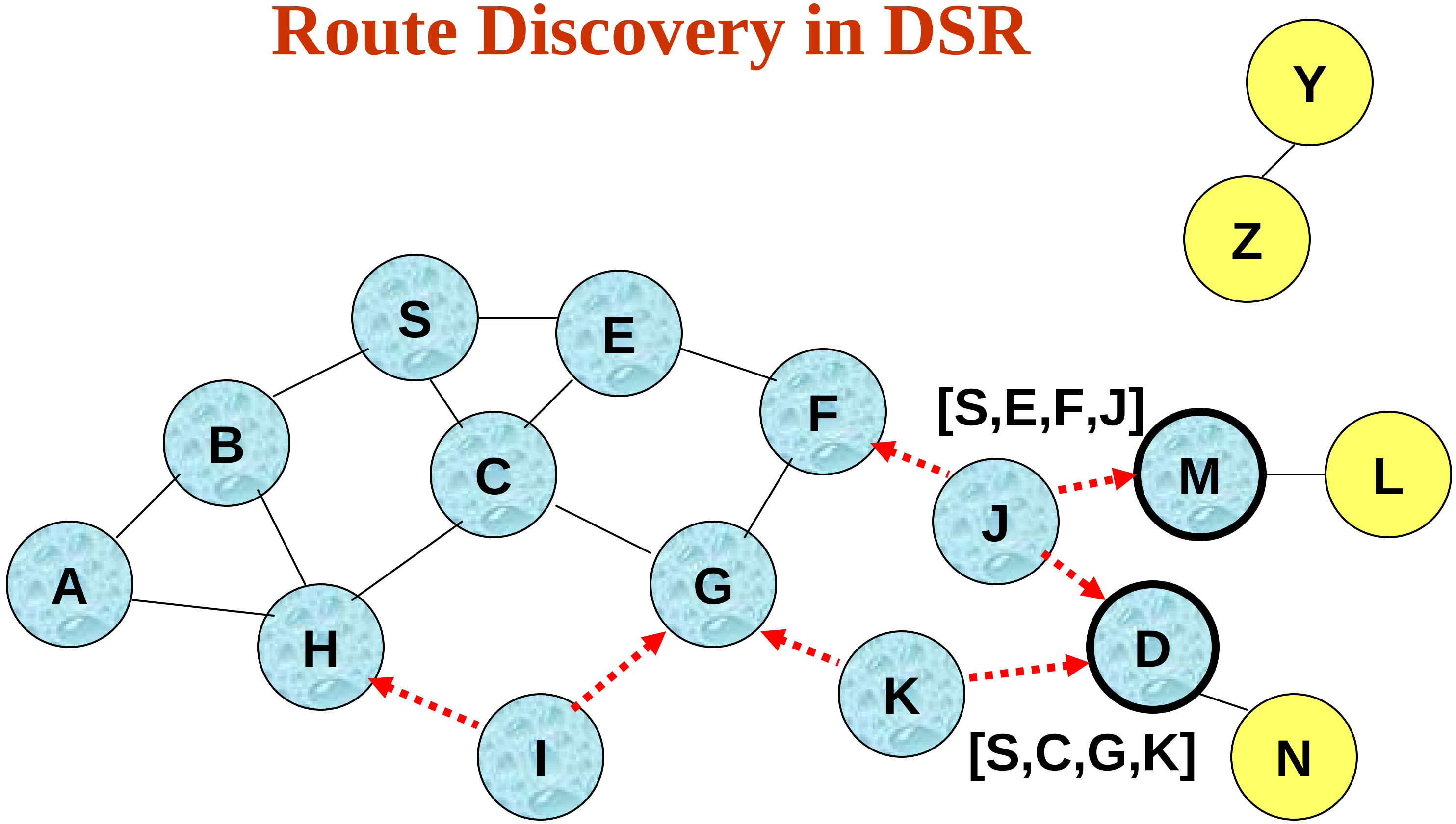
- **Node H receives packet RREQ from two neighbors: potential for collision**

Route Discovery in DSR



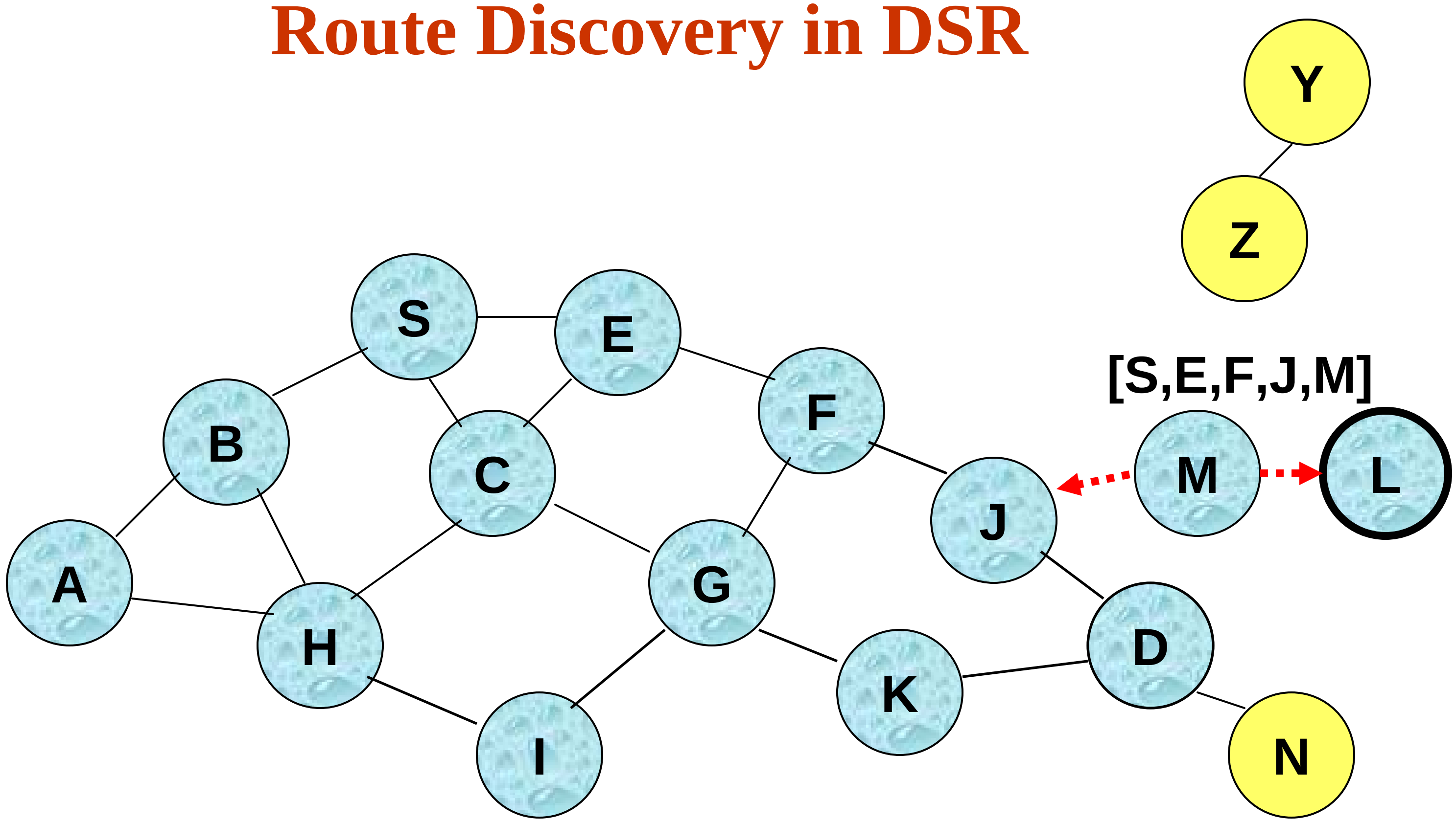
- Node C receives RREQ from G and H, but does not forward it again, because node C has already forwarded RREQ once

Route Discovery in DSR



- Nodes J and K both broadcast RREQ to node D
- Since nodes J and K are hidden from each other, their transmissions may collide

Route Discovery in DSR

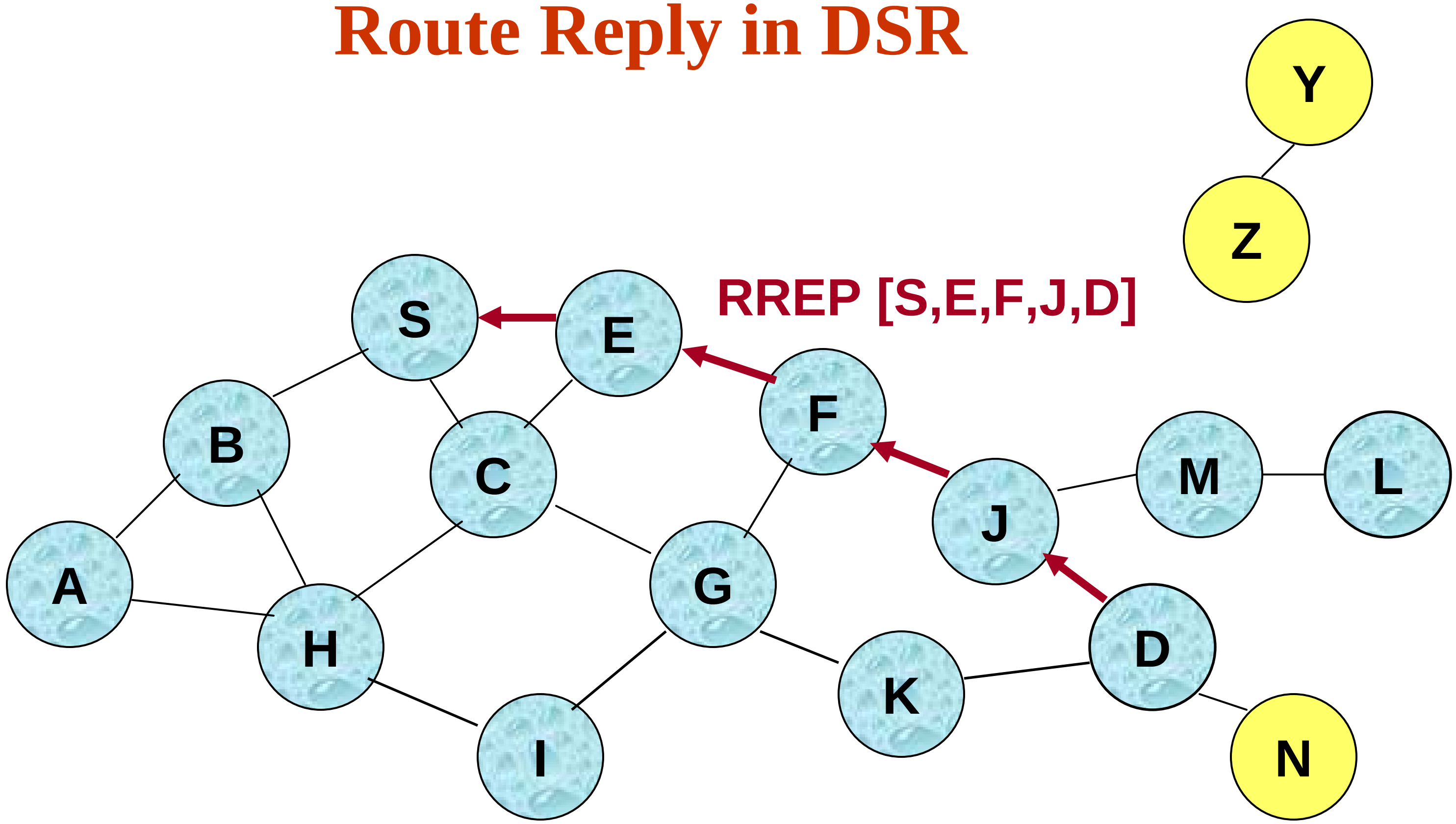


• Node D **does not forward** RREQ, because node D is the **intended target** of the route discovery

Route Discovery in DSR

- Destination D on receiving the first RREQ, sends a Route Reply (RREP)
- RREP is sent on a route obtained by reversing the route appended to received RREQ
- RREP packet contains the route from S to D that was discovered using the RREQ packet

Route Reply in DSR

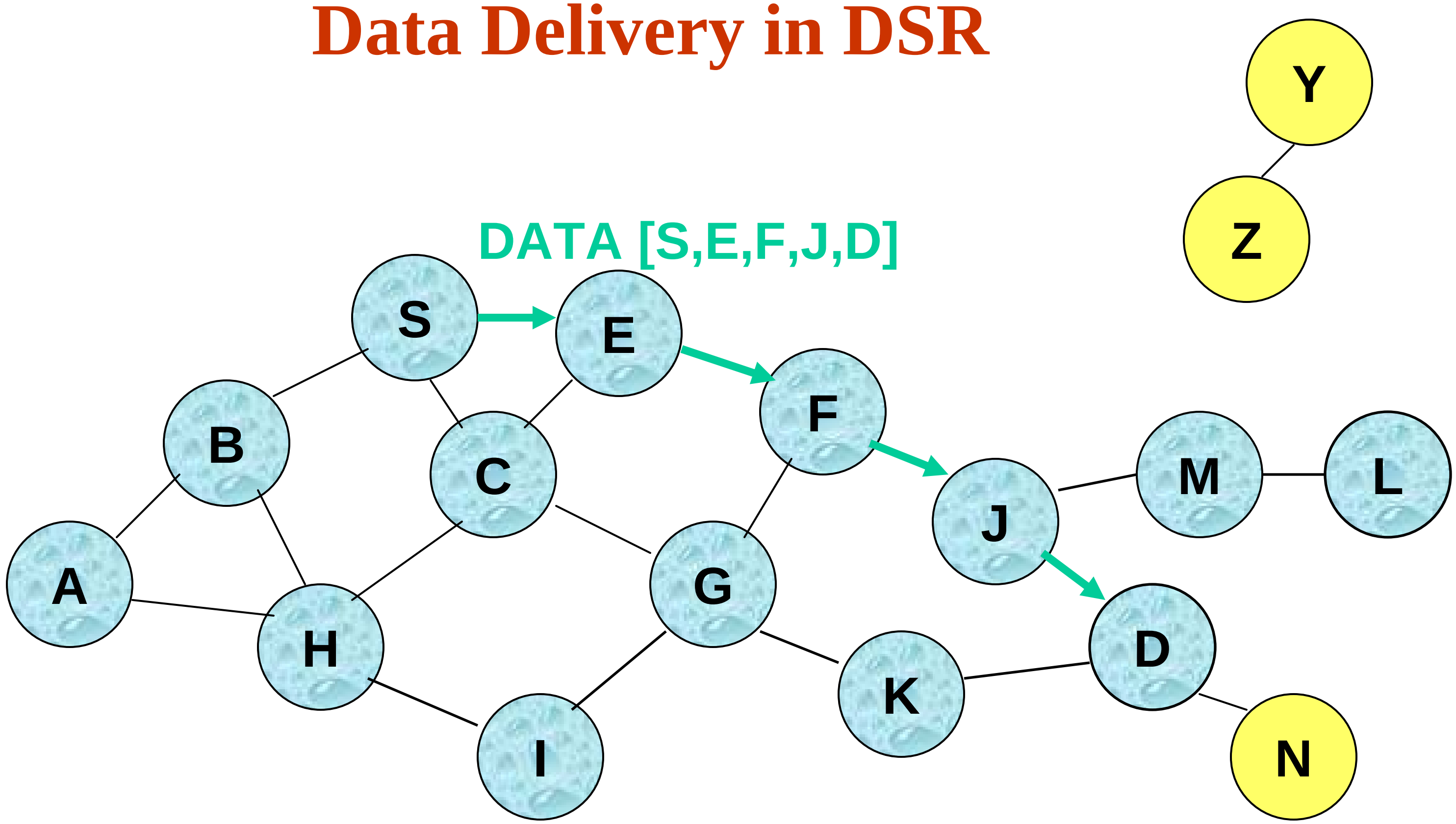


Represents RREP control message

Dynamic Source Routing (DSR)

- Node S on receiving RREP, caches the route included in the RREP
- When node S sends a data packet to D, the entire route is included in the packet header
  - hence the name source routing
- Intermediate nodes use the source route included in a packet to determine to whom a packet should be forwarded

Data Delivery in DSR



Packet header size grows with route length

When to Perform a Route Discovery?

- When node S wants to send data to node D (i.e. *on-demand*), but does not know a valid route to node D

DSR Optimization: Route Caching

- Each node caches a new route it learns by *any means*
- When node S finds route [S,E,F,J,D] to node D, node S also learns route [S,E,F] to node F
- When node K receives Route Request [S,C,G] destined for node D, node K learns route [K,G,C,S] to node S
- When node F forwards Route Reply [S,E,F,J,D], node F learns route [F,J,D] to node D
- When node E forwards Data [S,E,F,J,D] it learns route [E,F,J,D] to node D
- A node may also learn a route when it overhears Data packets!

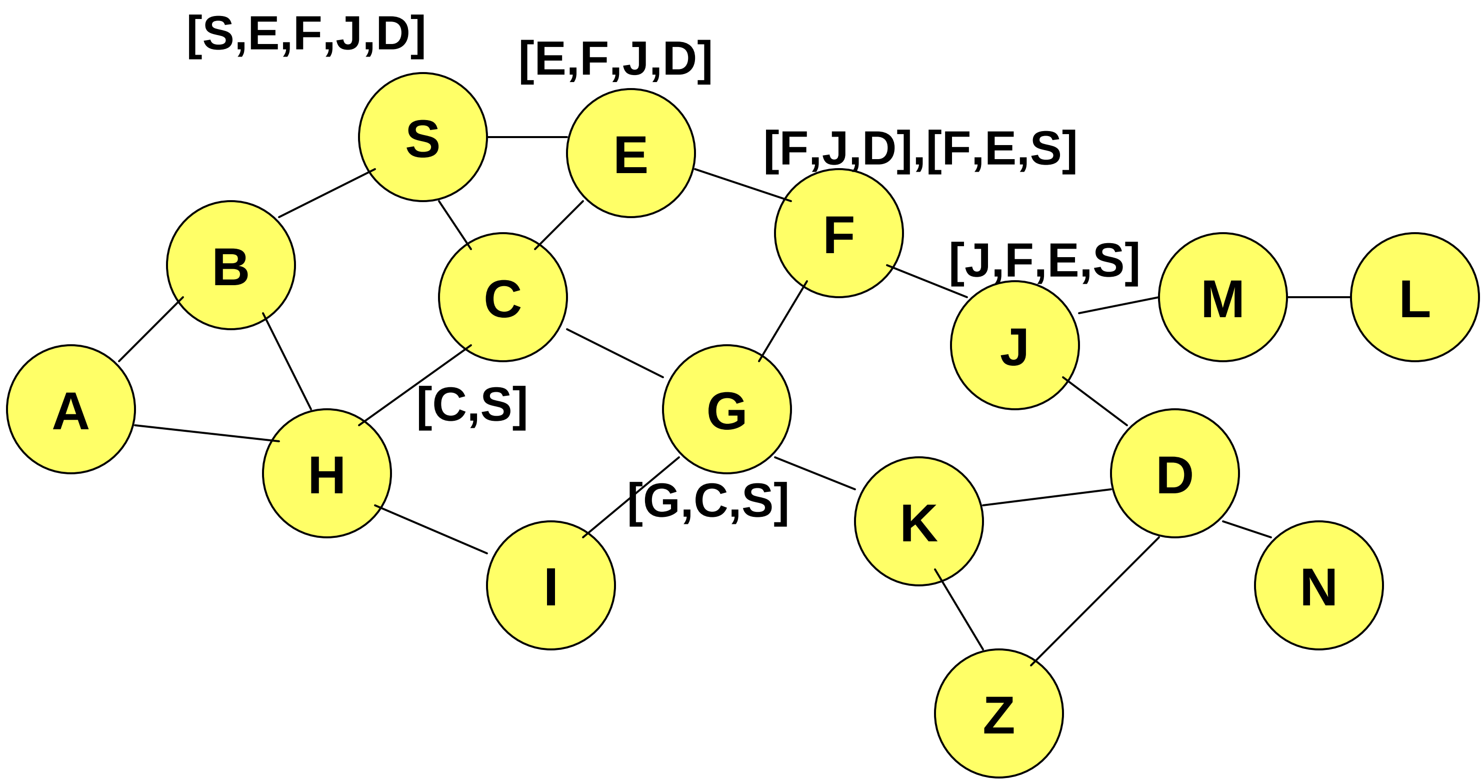
Use of Route Caching

- When node S learns that a route to node D is broken, it uses another route from its local cache, if such a route to D exists in its cache. Otherwise, node S initiates route discovery by sending a route request

- Node X on receiving a Route Request for some node D can send a Route Reply if node X knows a route to node D

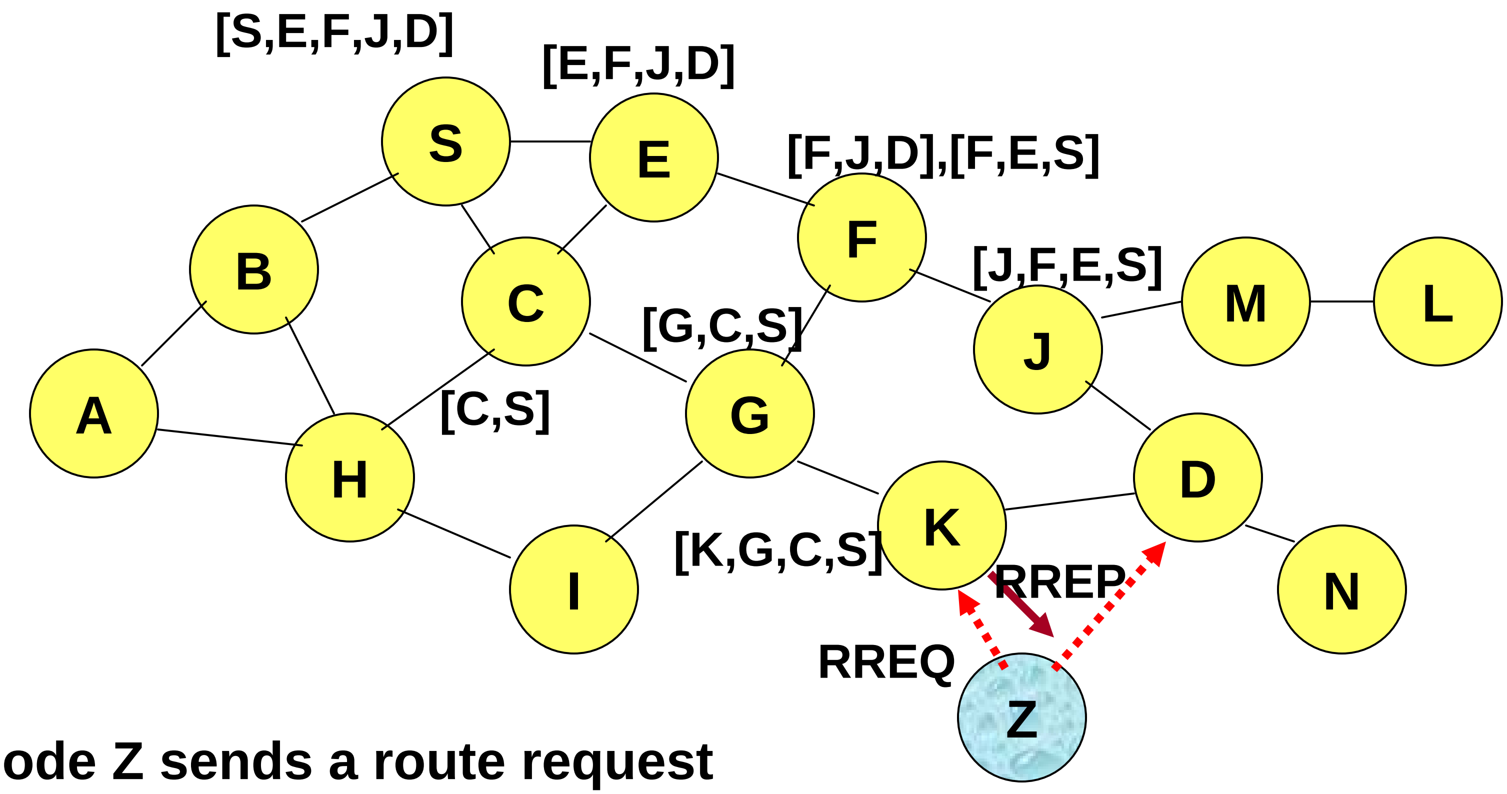
- Use of route cache
  - can speed up route discovery
  - can reduce propagation of route requests

Use of Route Caching



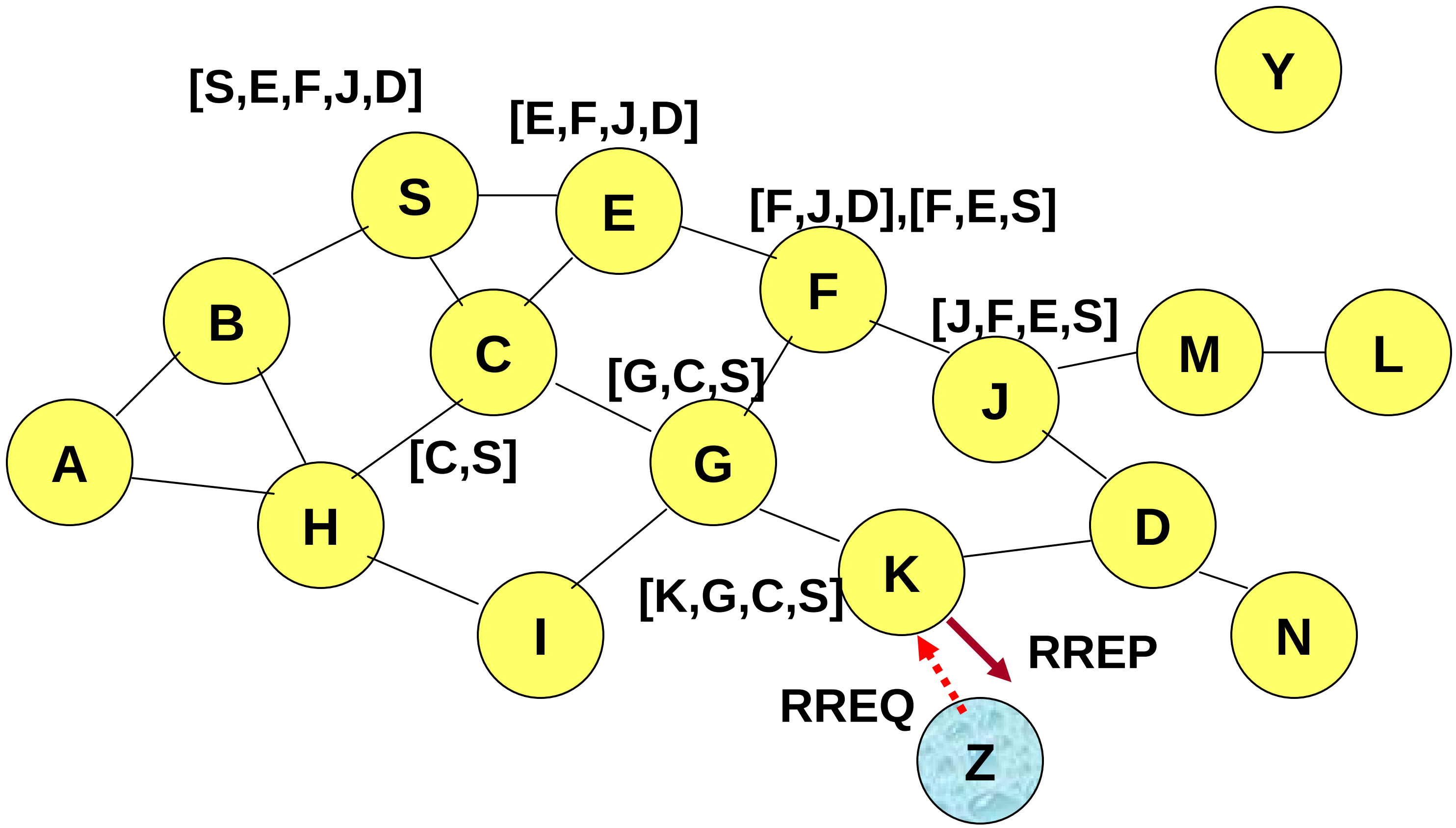
[P,Q,R] Represents cached route at a node

Use of Route Caching:
Can Speed up Route Discovery



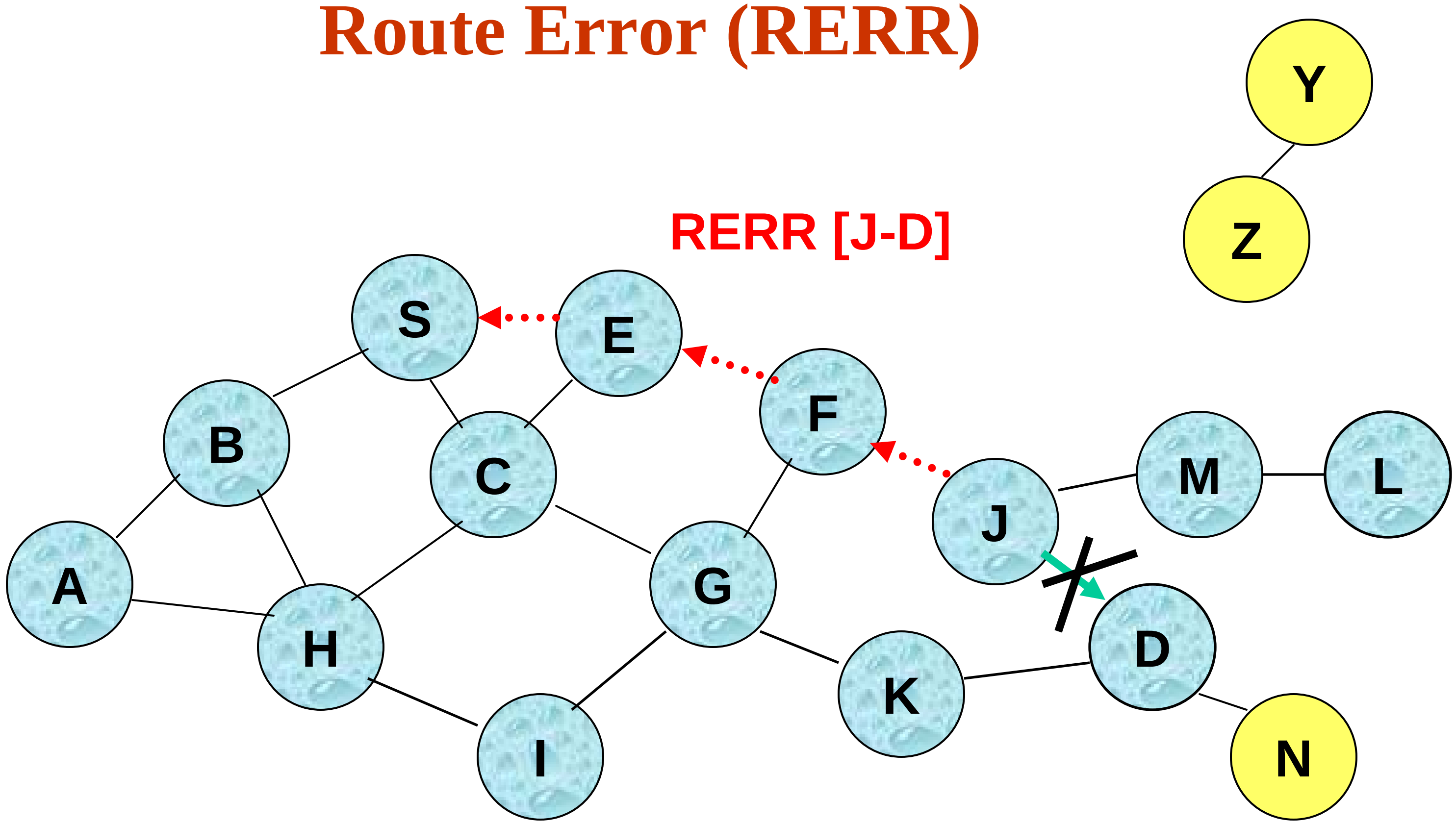
When node Z sends a route request for node C, node K sends back a route reply [Z,K,G,C] to node Z using a locally cached route

Use of Route Caching:
Can Reduce Propagation of Route Requests



Assume that there is no link between D and Z.
Route Reply (RREP) from node K limits flooding of RREQ.
In general, the reduction may be less dramatic.

Route Error (RERR)



J sends a route error to S along route J-F-E-S when its attempt to forward the data packet S (with route SEFJD) on J-D fails

Nodes hearing RERR update their route cache to remove link J-D

Route Caching:Disadvantages

- Stale caches can adversely affect performance
- With passage of time and host mobility, cached routes may become invalid
- A sender host may try several stale routes (obtained from local cache, or replied from cache by other nodes), before finding a good route
- (An illustration of the adverse impact on TCP can be found in [Holland99])

Dynamic Source Routing: Advantages

- Routes maintained only between nodes who need to communicate (ie. *on-demand*)
  - reduces overhead of route maintenance
- Route caching can further reduce route discovery overhead
- A single route discovery may yield many routes to the destination, due to multiple intermediate nodes replying from local caches

Dynamic Source Routing: Disadvantages

- Packet header size grows with route length due to source routing
- Flood of route requests may potentially reach all nodes in the network
- Care must be taken to avoid collisions between route requests propagated by neighboring nodes
  - insertion of random delays before forwarding RREQ

Dynamic Source Routing: Disadvantages (cache)

- Increased contention if too many route replies come back due to nodes replying using their local cache
  - Route Reply *Storm* problem
  - Reply storm may be eased by preventing a node from sending RREP if it hears another RREP with a shorter route

- An intermediate node may send Route Reply using a stale cached route, thus polluting other caches

- This problem can be eased if some mechanism to purge (potentially) invalid cached routes is incorporated.

- For some proposals for cache invalidation, see [Hu00Mobicom]
  - Static timeouts
  - Adaptive timeouts based on link stability

Detour: Next-hop Routing

Next-hop Routing

- Each node maintains for each destination a preferred neighbor (i.e. a next hop)
- Packet contains a destination node identifier in the header
- Each node forwards the packet to the preferred neighbor corresponding to the destination
- Routing table construction, maintenance and update mechanism differs from one routing protocol to another
- The objective is to route packets along an optimal path, by implementing the distributed version of the shortest-path problem

Traditional Next-Hop Routing Methods

1. Link-State (e.g. used in the Internet OSPF protocol)
   - Closer to the centralized version of the shortest-path computation (Djikstra's method)
   - Each node maintains a view of the network topology
   - Each link is assigned a cost
   - Each node periodically broadcasts its outgoing link costs (e.g. by flooding)
   - Each node updates link information + runs a shortest-path algorithm to choose its next hop for each destination
   - Some info may be inconsistent (long prop. Delays, partitioned network, etc.)→may result in formation of routing loops, but these loops are temporary [McQuillan80] (max loop existence time=time it takes a packet to traverse the diameter of the network)

See CCN slides for a detailed description and examples

Next-Hop Routing Methods (Cont'd)

2. Distance-Vector
   - Applies the Distributed Bellman-Ford (DBF) algorithm [Bertsekas87]
   - For each destination x, each node i maintains a set of distances (d) for each path along every node j of its neighbors: $d_{ij}(x)$
   - When node i receives a packet intended to node k, it sends it along the node k which satisfies $d_{ik}(x)=\min_j d_{ij}(x)$
   - This succession leads to x along the shortest path
   - Each node monitors the cost of its outgoing links
   - Each node periodically broadcasts, to its neighbors, its current estimate of the shortest distance to every other node in the network
   - Thus the name "distance" (to the destination) and "vector" (next hop)

See CCN slides for a detailed description and examples

Link-State vs. Distance Vector

- router knows global view of the network
- Computationally inefficient
- Storage Space inefficient
- Short-lived loops

- router knows physically-connected neighbors, link costs to neighbors
- iterative process of computation, exchange of partial info with neighbors
- Can result in formation of long-lived and short-lived loops→can be removed by "internodal coordination protocol," (difficult to enforce in an ad-hoc network)

Ad Hoc On-Demand Distance Vector Routing (AODV) [Perkins99Wmcsa]

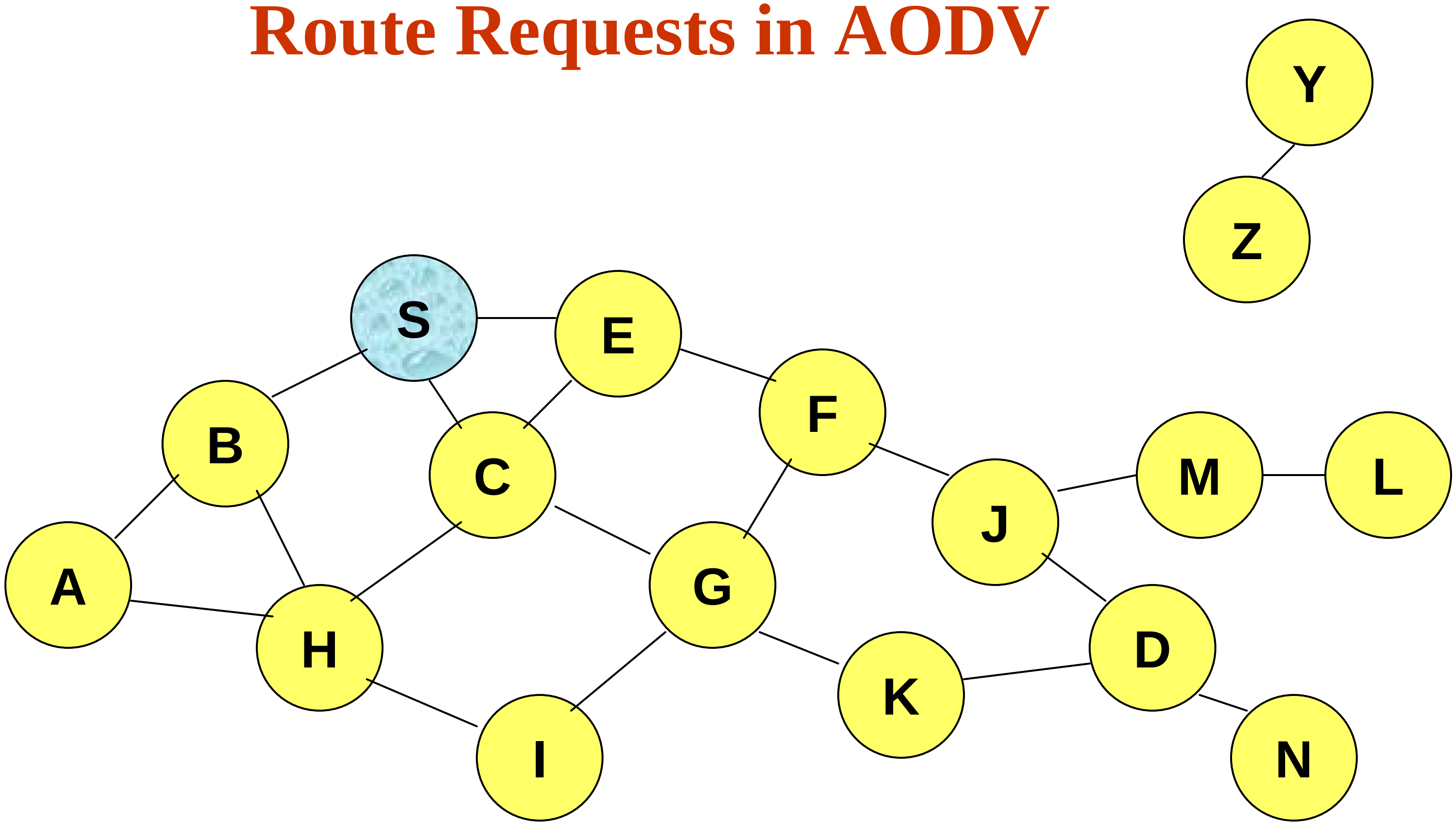
Ad Hoc On-Demand Distance Vector Routing (AODV) [Perkins99Wmcsa]

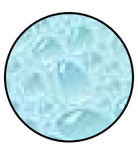
- DSR includes source routes in packet headers
- Resulting large headers can sometimes degrade performance
  - particularly when data contents of a packet are small
- AODV attempts to improve on DSR by maintaining routing tables at the nodes, so that data packets do not have to contain routes
- AODV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

AODV

- Route Requests (RREQ) are forwarded in a manner similar to DSR
- When a node re-broadcasts a Route Request, it sets up a reverse path pointing towards the source
  - AODV assumes symmetric (bi-directional) links
- When the intended destination receives a Route Request, it replies by sending a Route Reply
- Route Reply travels along the reverse path set-up when Route Request is forwarded

Route Requests in AODV

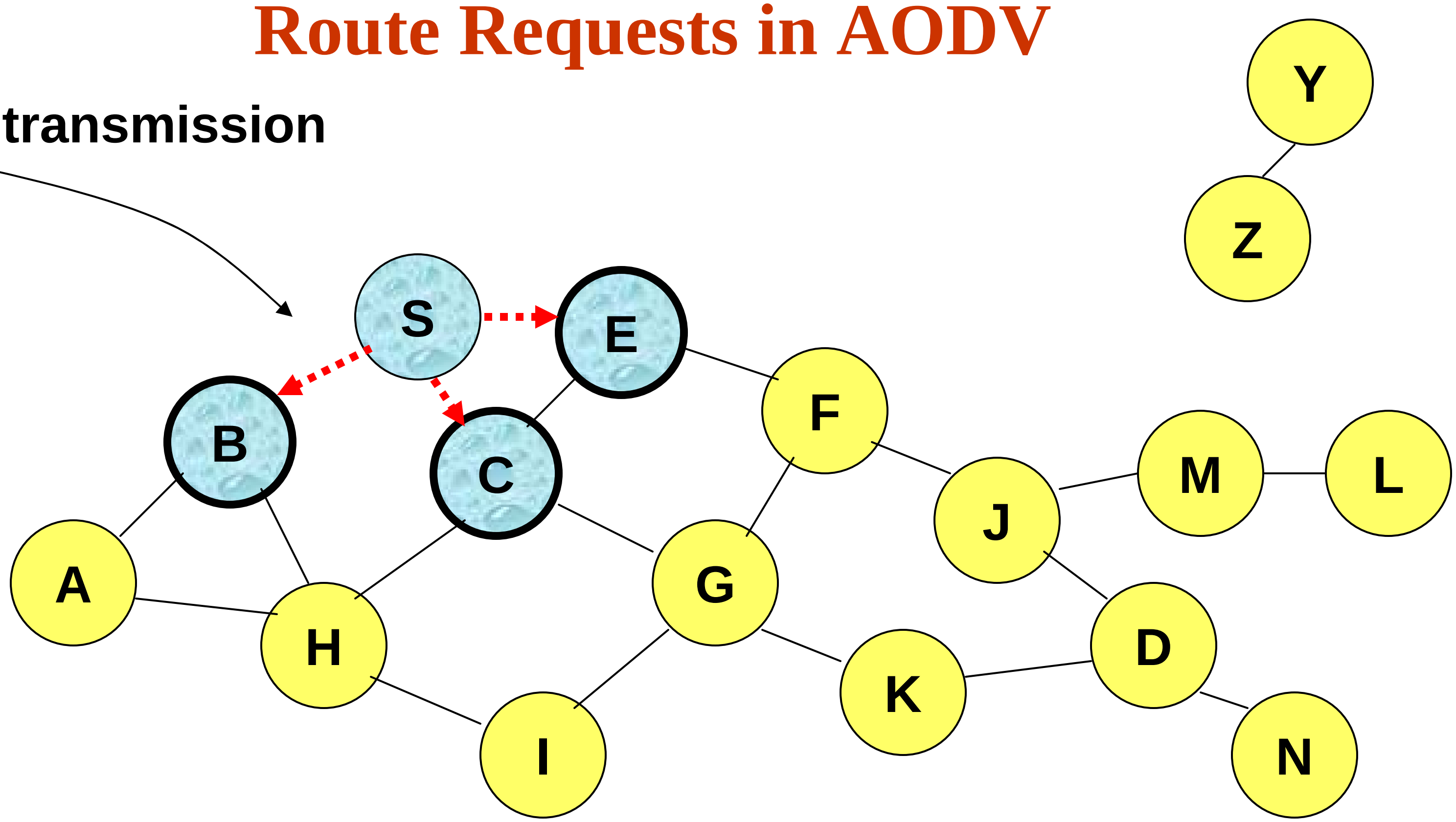




Represents a node that has received RREQ for D from S

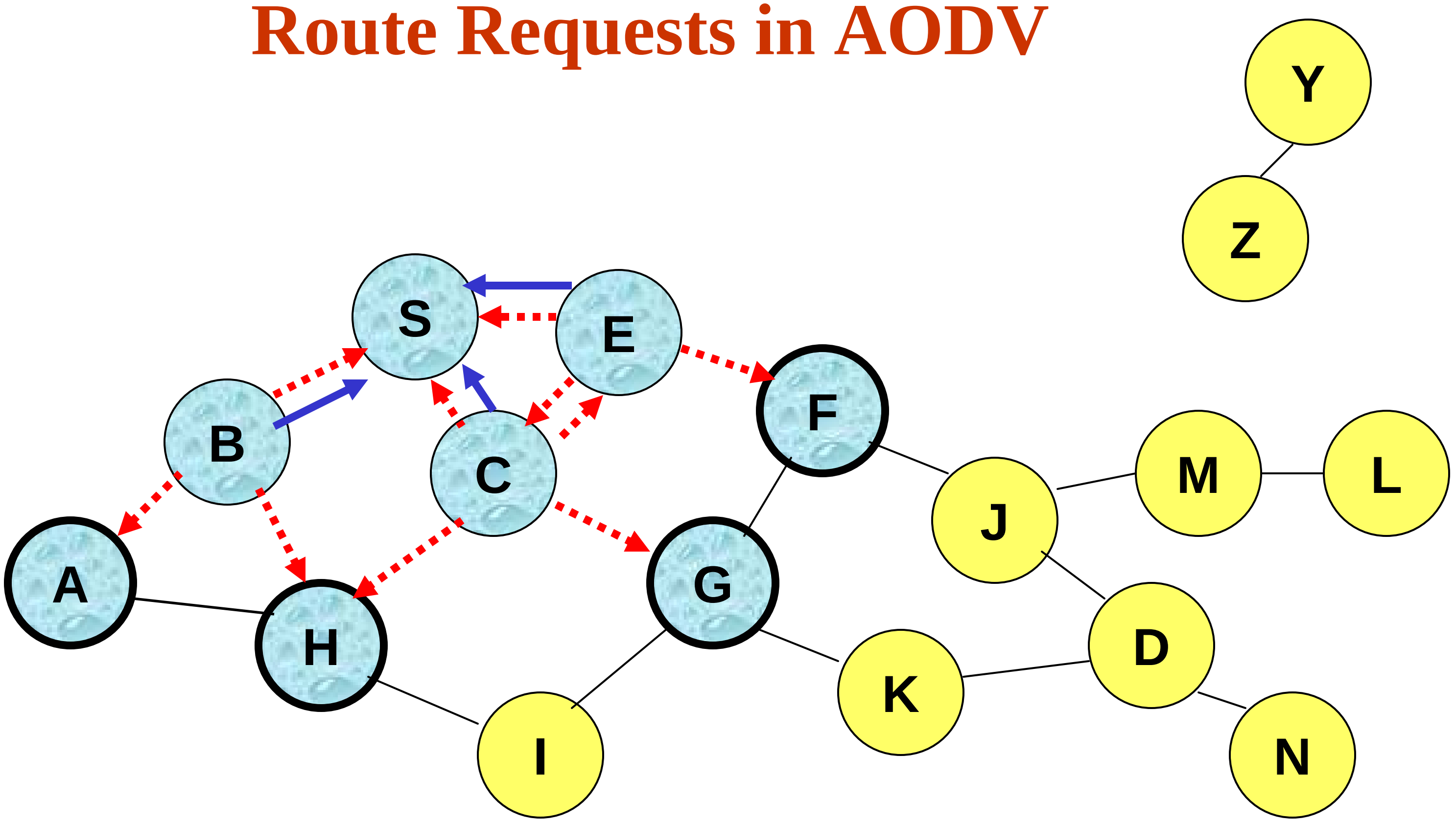
Route Requests in AODV

Broadcast transmission



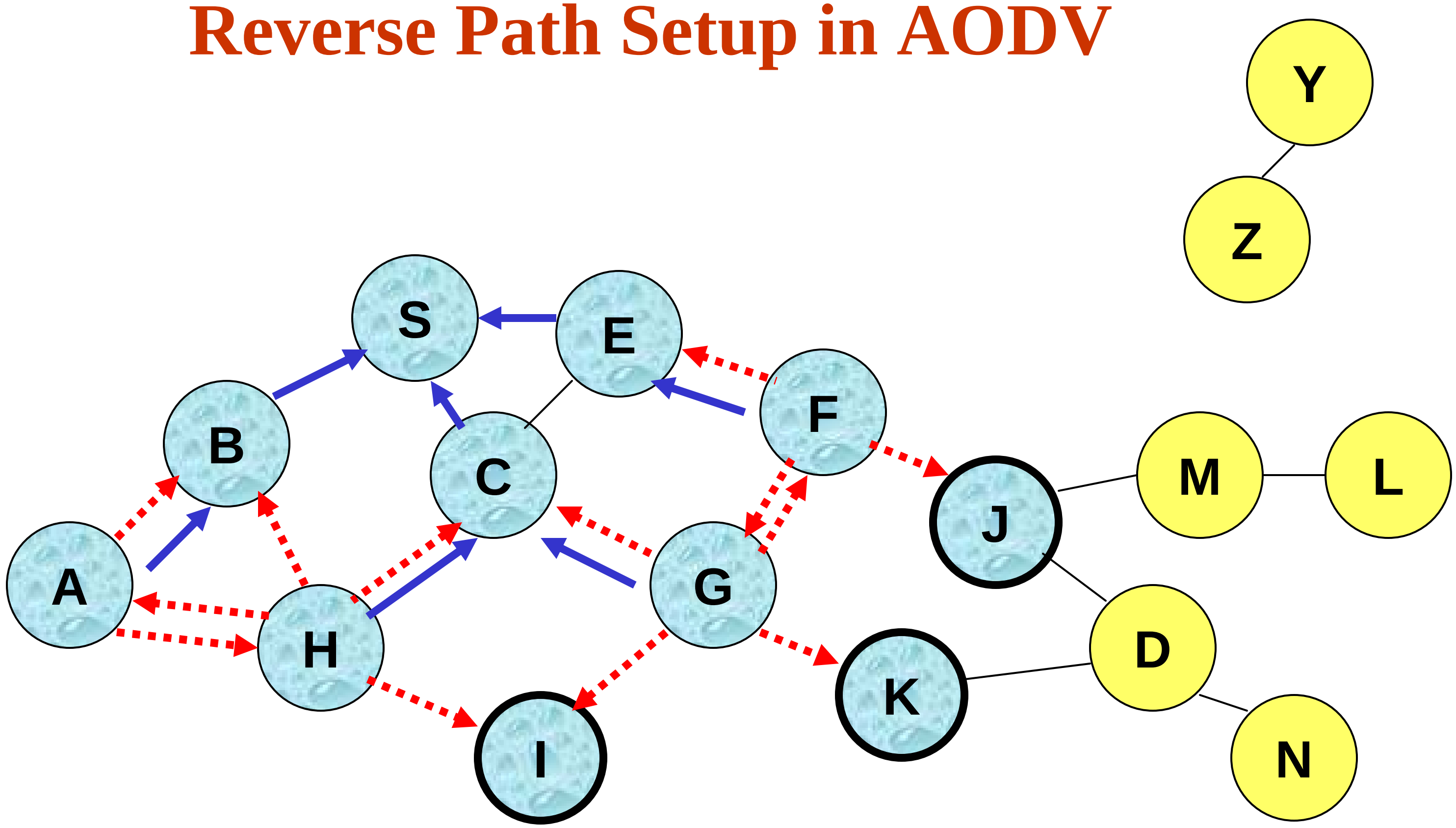
Represents transmission of RREQ

Route Requests in AODV



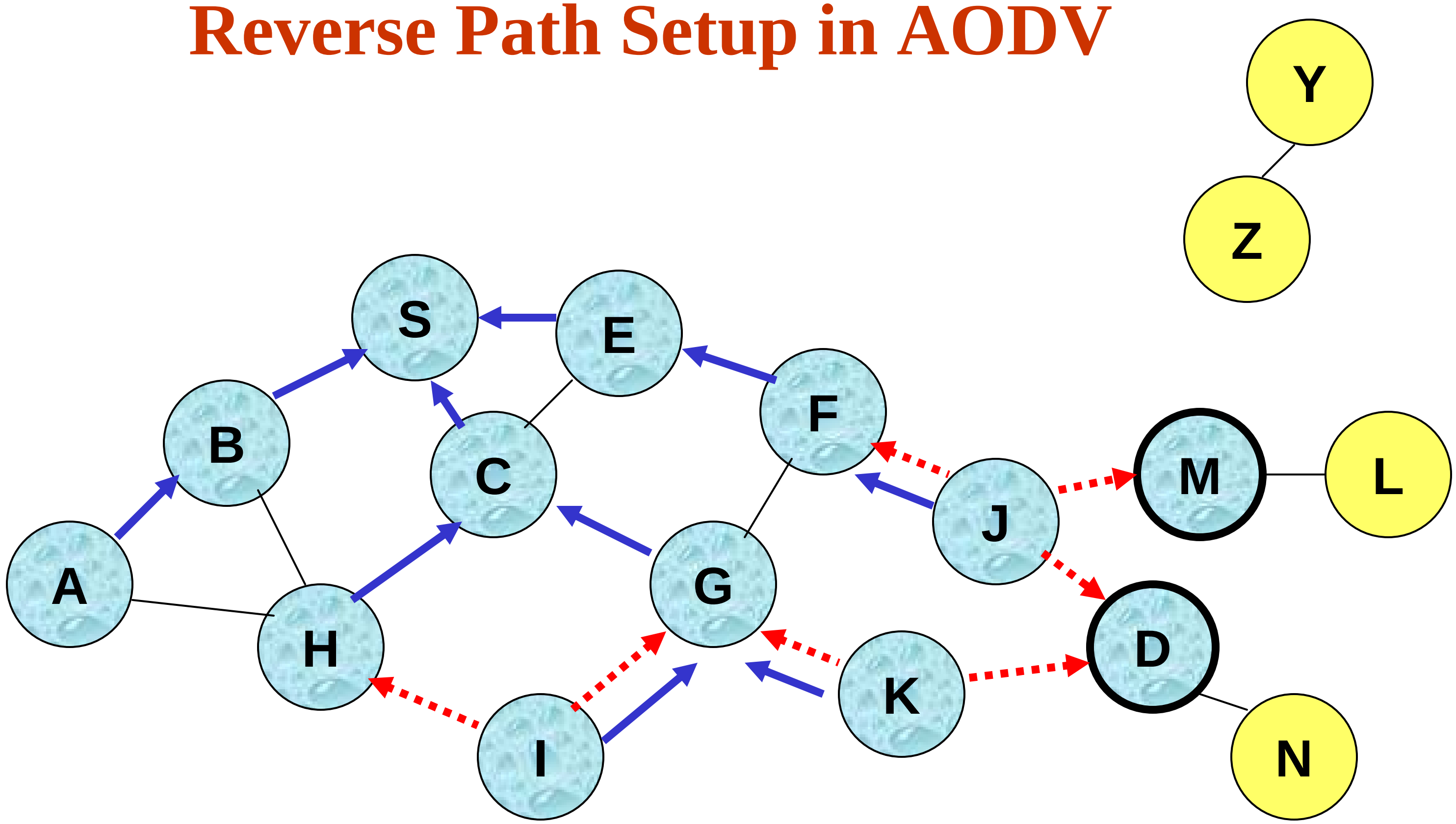
Represents links on Reverse Path

Reverse Path Setup in AODV

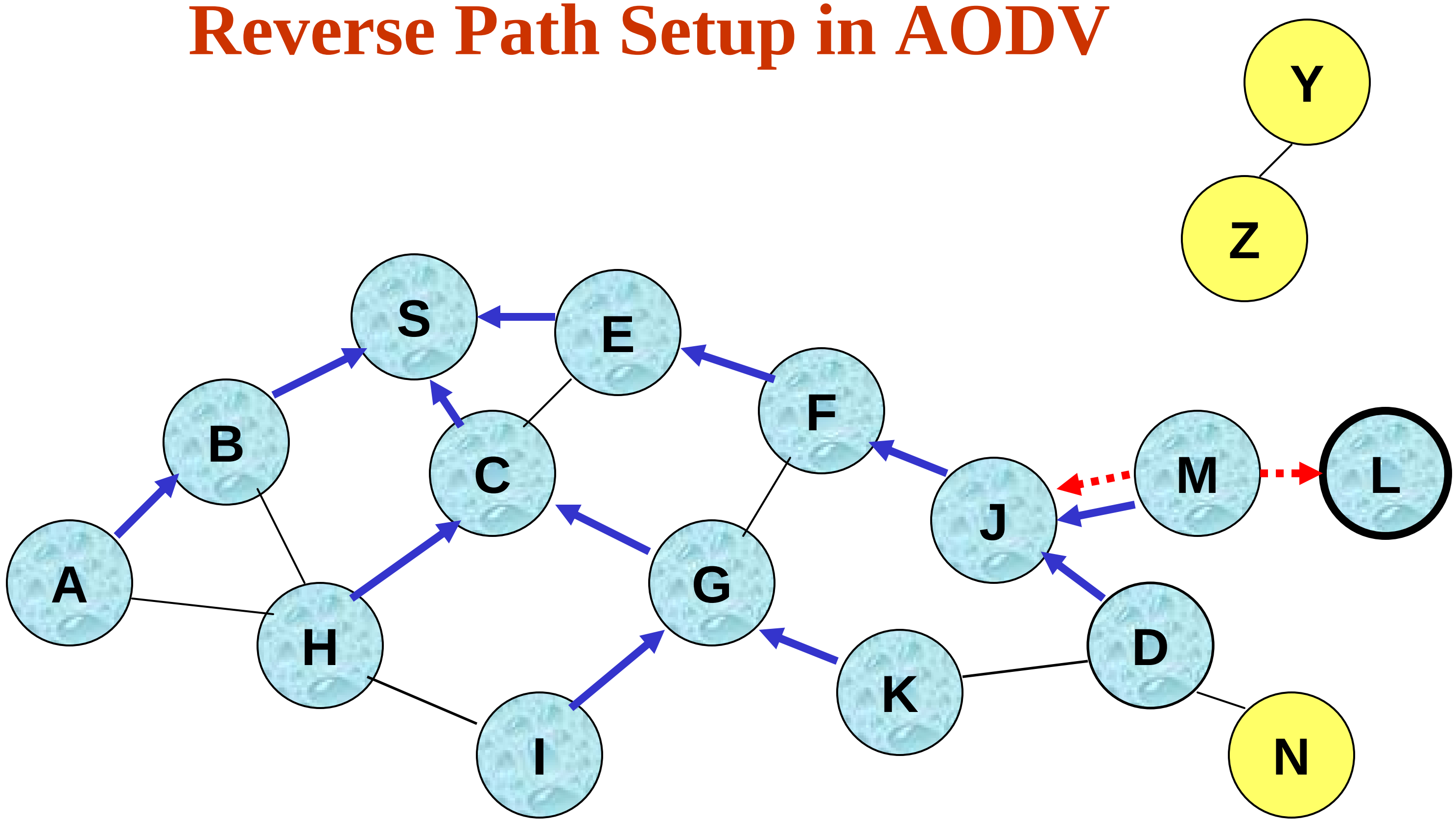


- Node C receives RREQ from G and H, but does not forward it again, because node C has already forwarded RREQ once

Reverse Path Setup in AODV

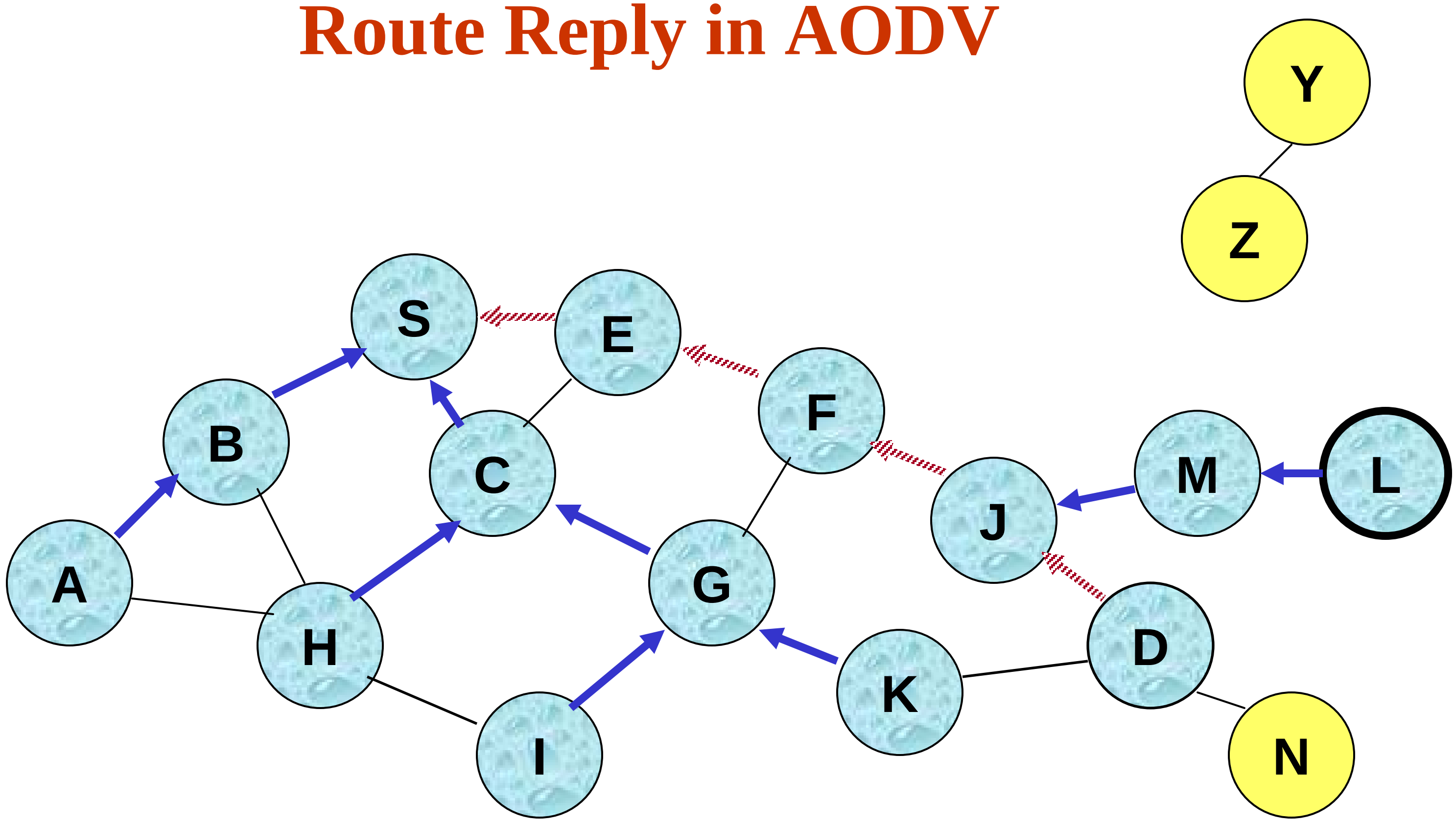


Reverse Path Setup in AODV



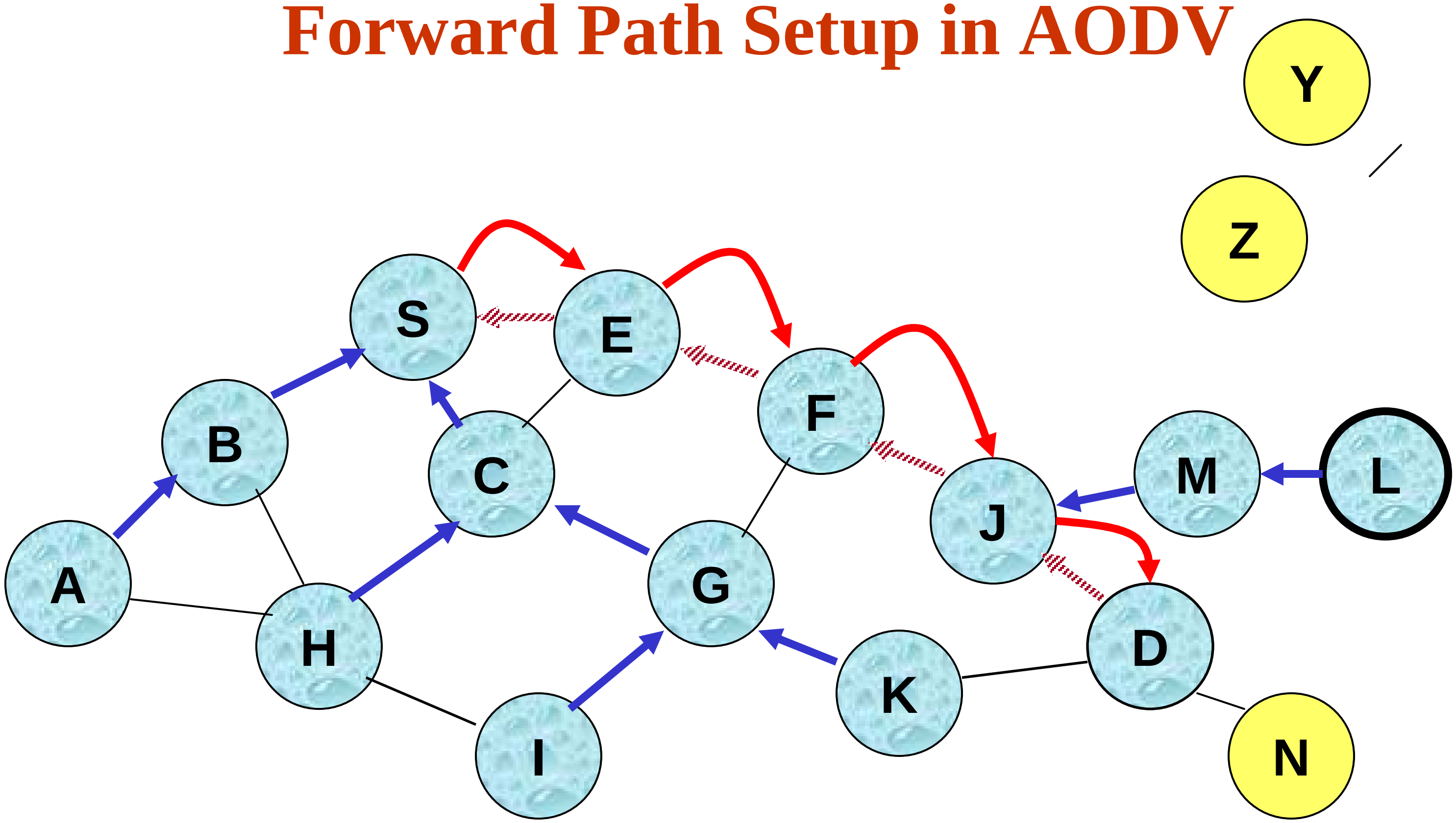
- Node D **does not forward** RREQ, because node D is the **intended target** of the RREQ

Route Reply in AODV



Represents links on path taken by RREP

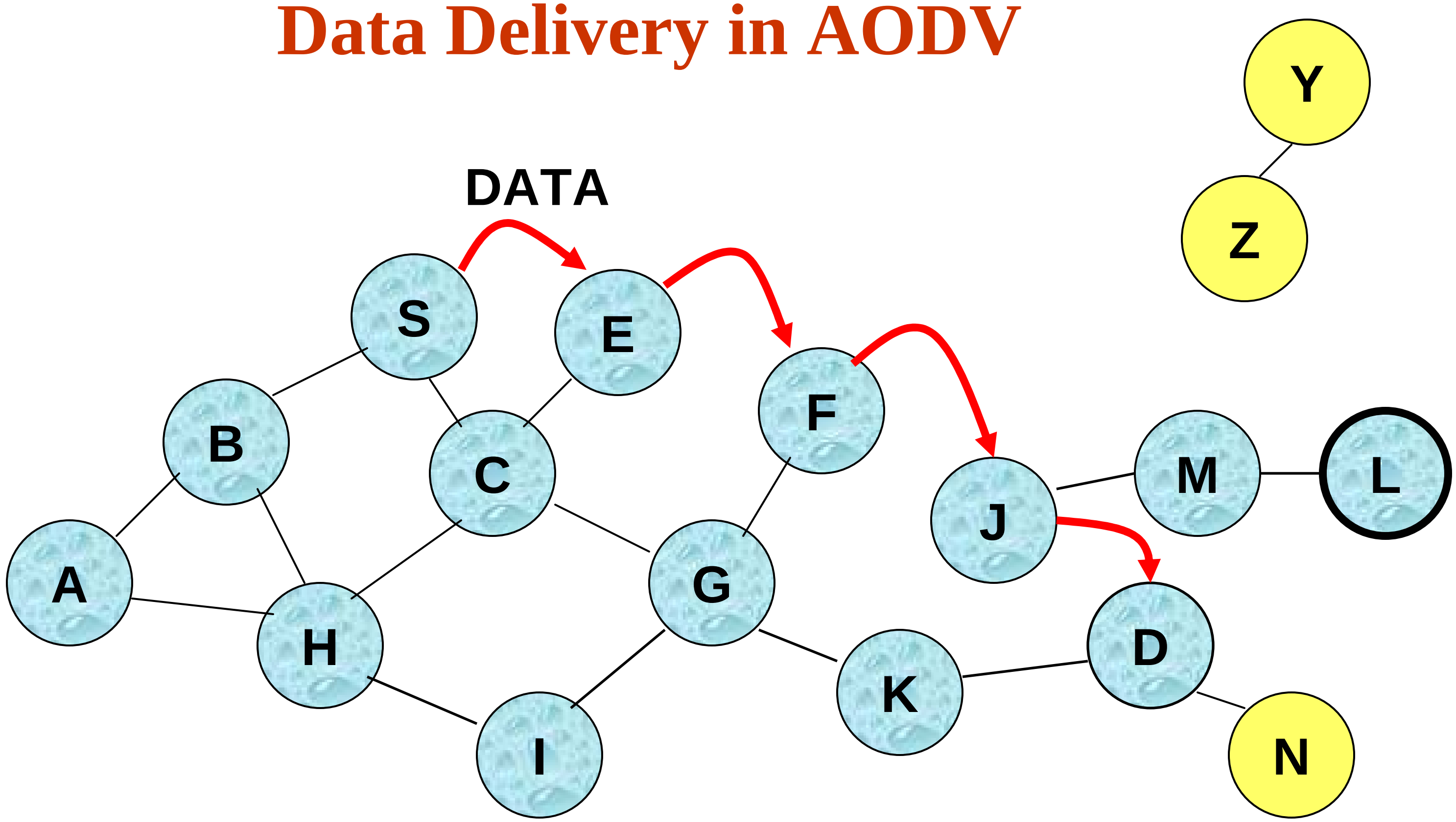
Forward Path Setup in AODV



Forward "next-hops" are setup when RREP travels along the reverse path – there is no actual path stored

Represents a next-hop entry on the forward path to D

Data Delivery in AODV



Routing table entries used to forward data packet.

Route is *not* included in packet header.

Destination Sequence Numbers in AODV

- 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
  - 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

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 data is 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 Reporting

- A neighbor of node X is considered active for a routing table entry if the neighbor sent a packet within *active_route_timeout* interval which was forwarded using that entry

- When the next hop link in a routing table entry breaks, all active neighbors are informed

- Link failures are propagated by means of Route Error messages, which also update destination sequence numbers

Route Error

- When node X is unable to forward packet P (from node S to node D) on link (X,Y), it generates a RERR message
- Node X increments the destination sequence number for D cached at node X
- The incremented sequence number *N* is included in the RERR
- When node S receives the RERR, it initiates a new route discovery for D using destination sequence number at least as large as *N*
- When node D receives the route request with destination sequence number N, node D will set its sequence number to N, unless it is already larger than N

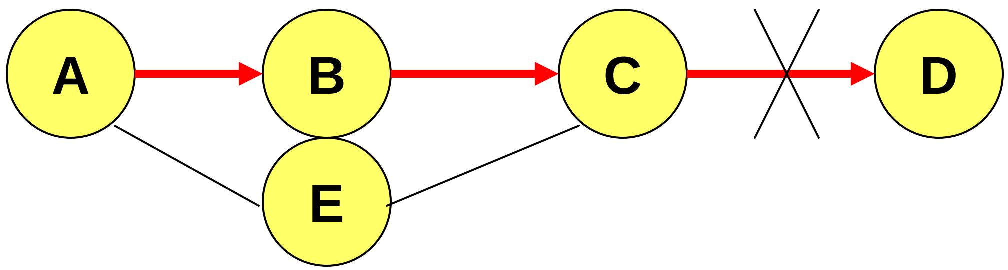
Link Failure Detection

- *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 acknowledgement may be used as an indication of link failure

Why Sequence Numbers in AODV

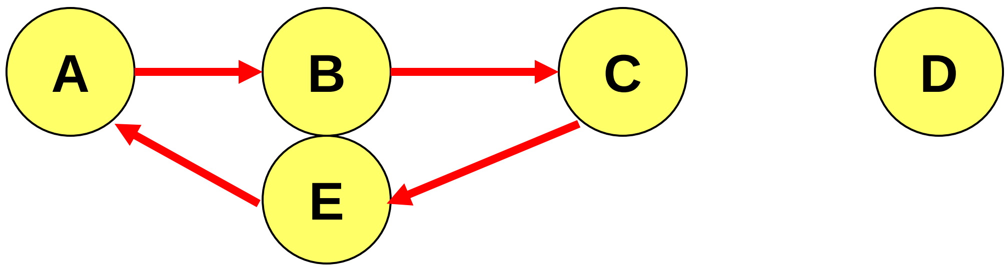
- To avoid using old/broken routes
  - To determine which route is newer

- To prevent formation of loops



  - Assume that A does not know about failure of link C-D because RERR sent by C is lost
  - Now C performs a route discovery for D (in aodv, it increments the dest seq no for D). Node A receives the RREQ (say, via path C-E-A)
  - Node A will reply since A knows a route to D via node B (it would have not replied in AODV since its dest. Sequ. No. is less than the one used in rreq)
  - Results in a loop (C-E-A-B-C)

Why Sequence Numbers in AODV



– Loop C-E-A-B-C

Optimization: Expanding Ring Search [Perkins00]

- 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

Summary: AODV

- Routes need not be included in packet headers
- Nodes maintain routing tables containing entries only for routes that are in active use
- At most one next-hop per destination maintained at each node
  - DSR may maintain several routes for a single destination
- Unused routes expire even if topology does not change

So far ...

- All nodes had identical responsibilities
- Some schemes propose giving special responsibilities to a subset of nodes
  - Even if all nodes are physically identical
- Core-based schemes are examples of such schemes