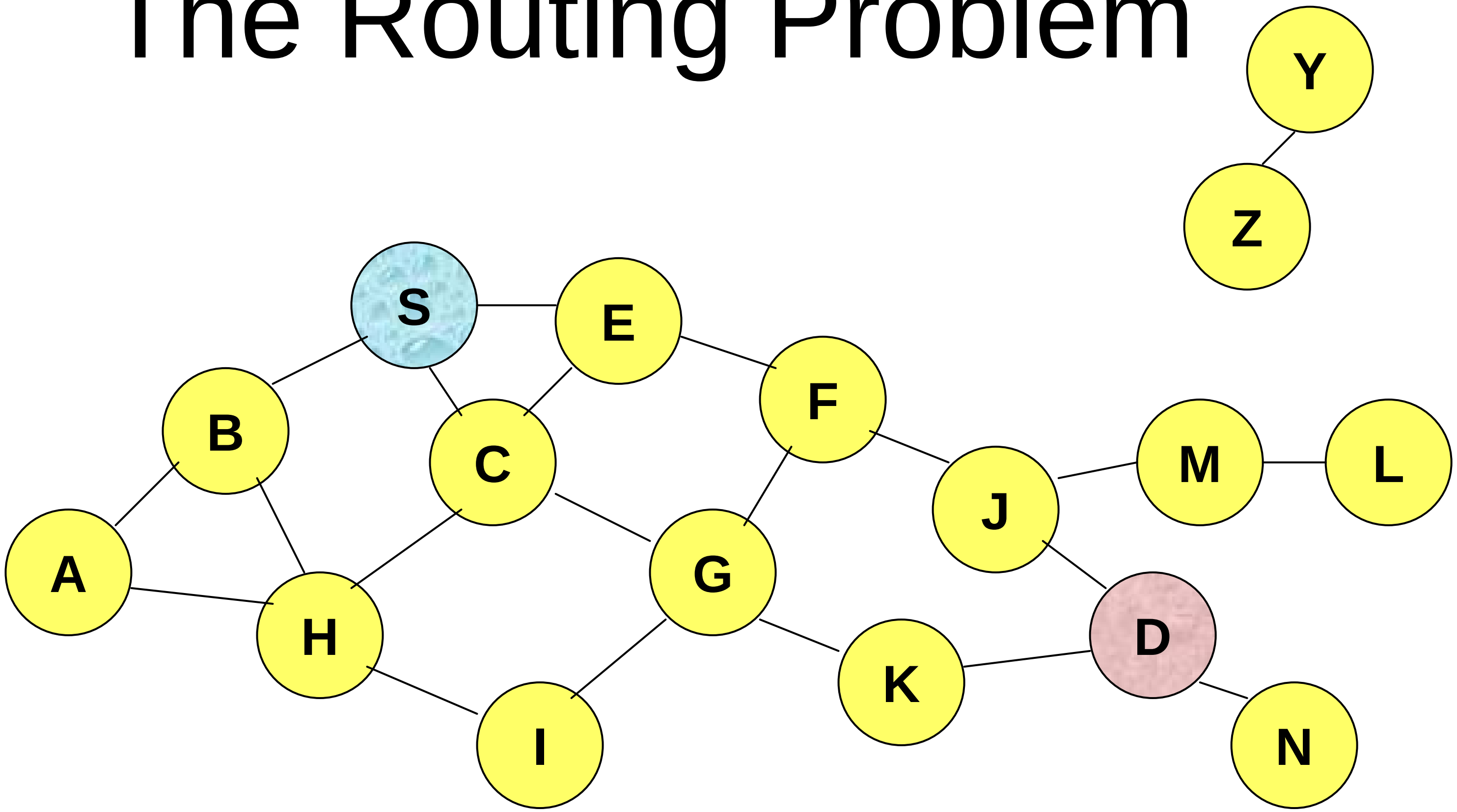
Lecture 3

Routing in Mobile Ad Hoc Networks

The Routing Problem



How to find a suitable path from source S to destination D?

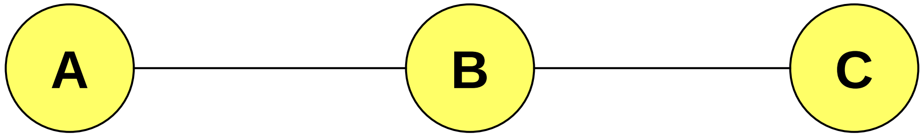
Motivation

- *Q: Why is routing in mobile ad hoc networks different?*
- Topology changes rapidly when terminals move fast.
- New performance criteria may be used
  - route stability despite mobility
  - energy consumption

Distance Vector Routing

- A protocol widely used in fixed network.
- Protocol:
  - A node broadcast its current estimate of its distance to its neighbors.
    - e.g. B sends A its current estimate of distance to C
  - Each neighbor then adds one to this distance. If this new distance is less than its current estimate, it adopts this new distance.
    - e.g. $d_{AC} = d_{BC} + 1$

Example



$\hat{d}_{AC} = \infty$

$\hat{d}_{BC} = 1$

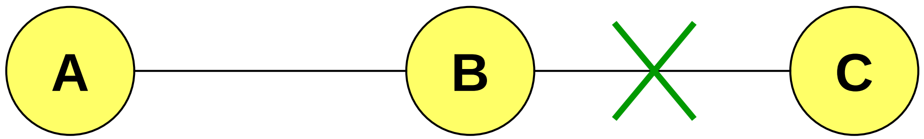
C is 1 hop away from B.

$\hat{d}_{AC} = 2$

$\hat{d}_{AC} = \hat{d}_{BC} + 1$

This new distance is adopted since 2 < ∞.

Counting-to-Infinity Problem



$\hat{d}_{AC} = 2$ $\qquad$ $\hat{d}_{BC} = \infty$

B cannot reach C now.

Distance to C via B is 2.

$\hat{d}_{BC} = 3$

$\hat{d}_{AC} = 4$

New distance adopted since it comes from B.

$\hat{d}_{BC} = 5$

……

Since links are unreliable in mobile ad hoc networks, this routing algorithm is obviously not a good choice.

Routing Protocols for Mobile Ad Hoc Networks

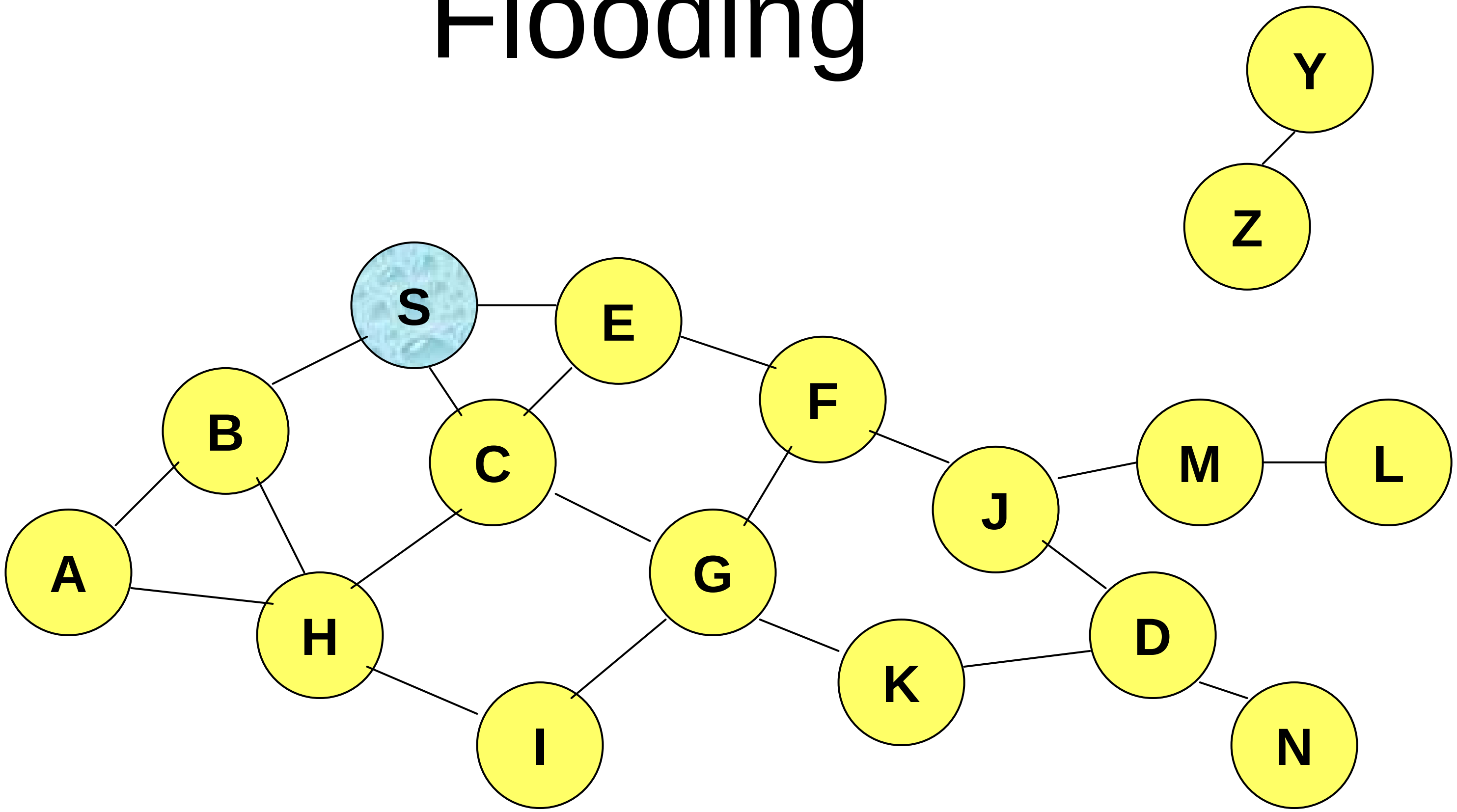
- A naive approach
  - Flooding

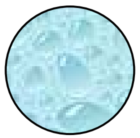
- Two popular protocols:
  - Dynamic Source Routing (DSR)
  - Adhoc On-demand Distance Vector (AODV) Routing

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

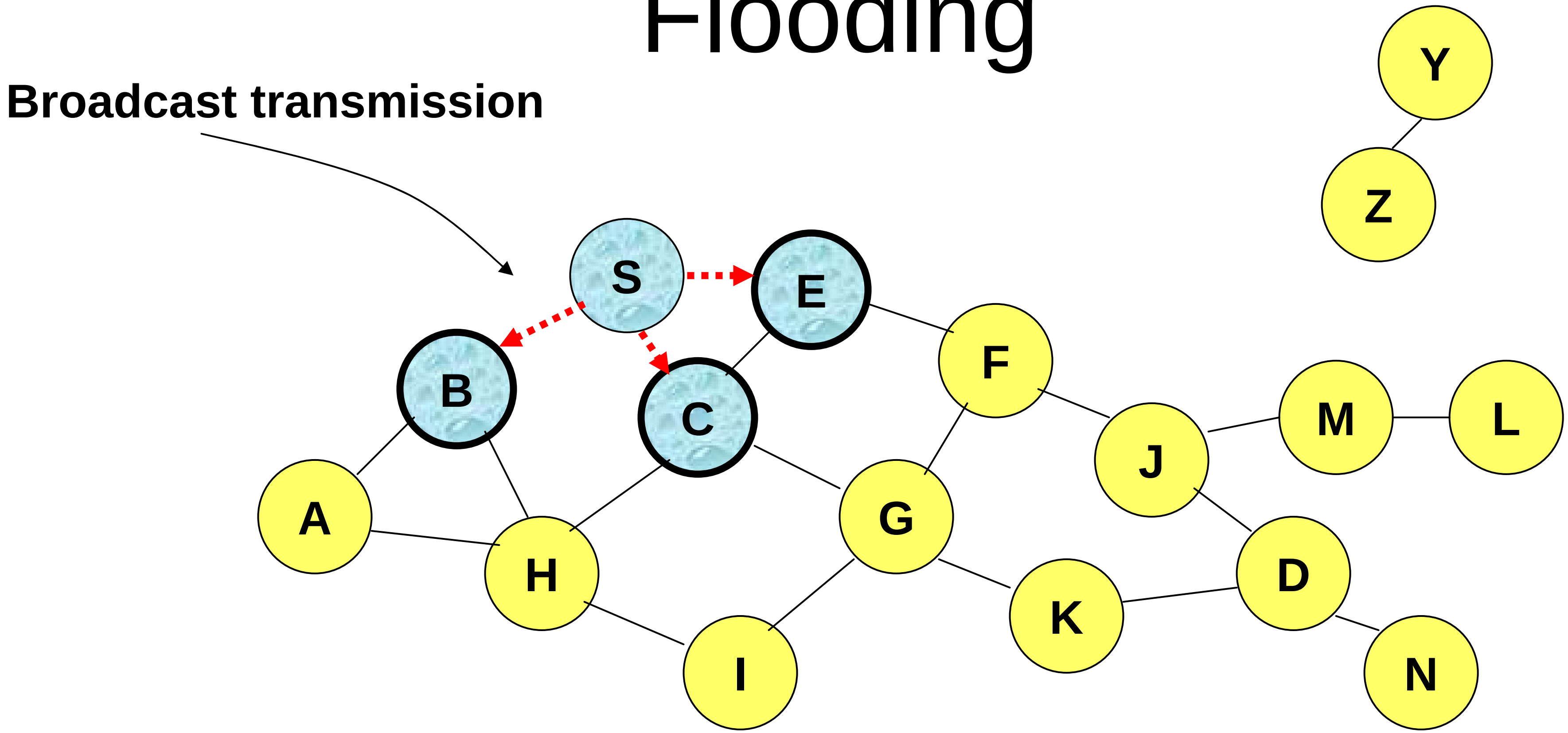




Represents a node that has received packet P

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

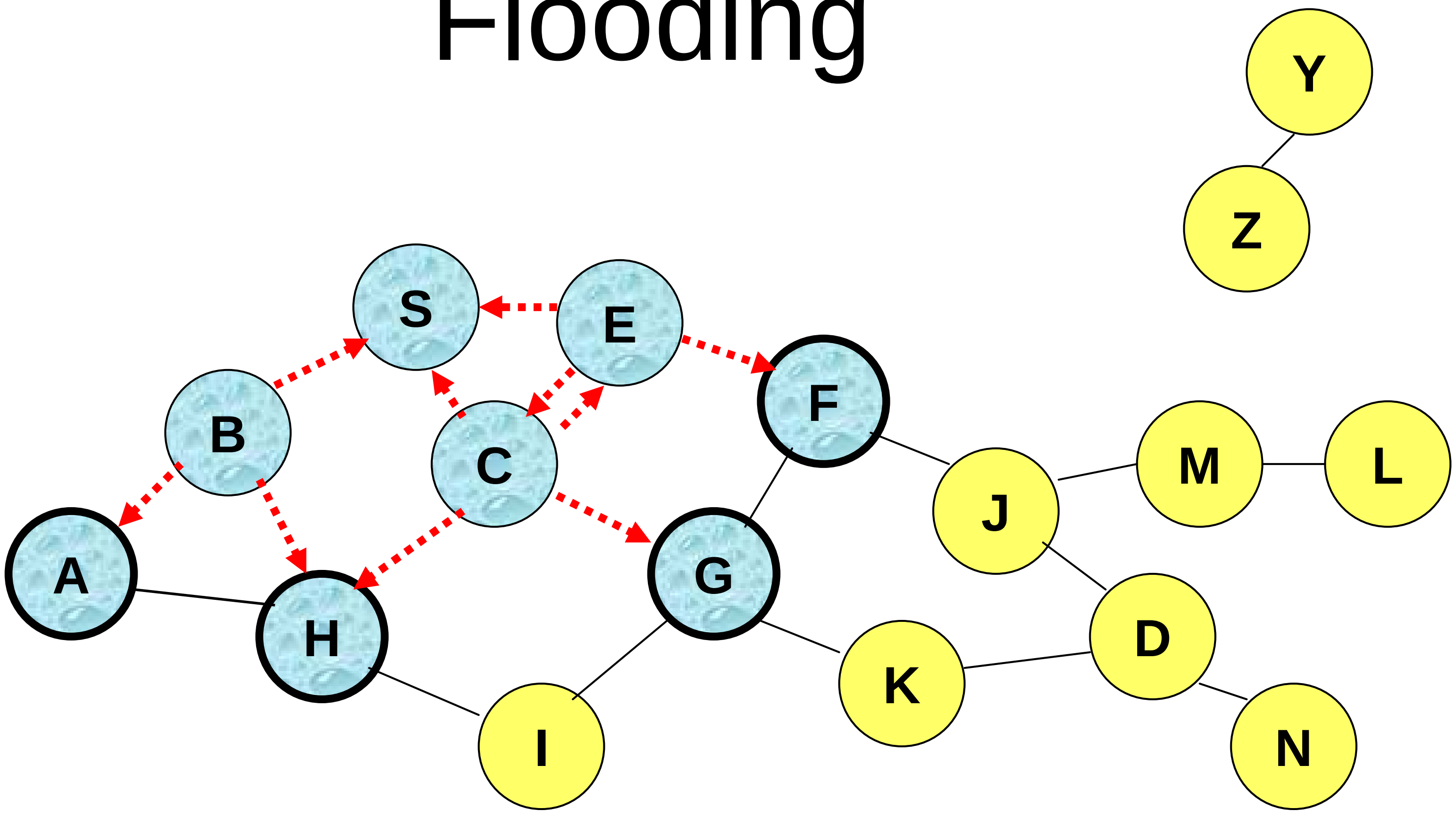
Flooding



Represents a node that receives packet P for the first time

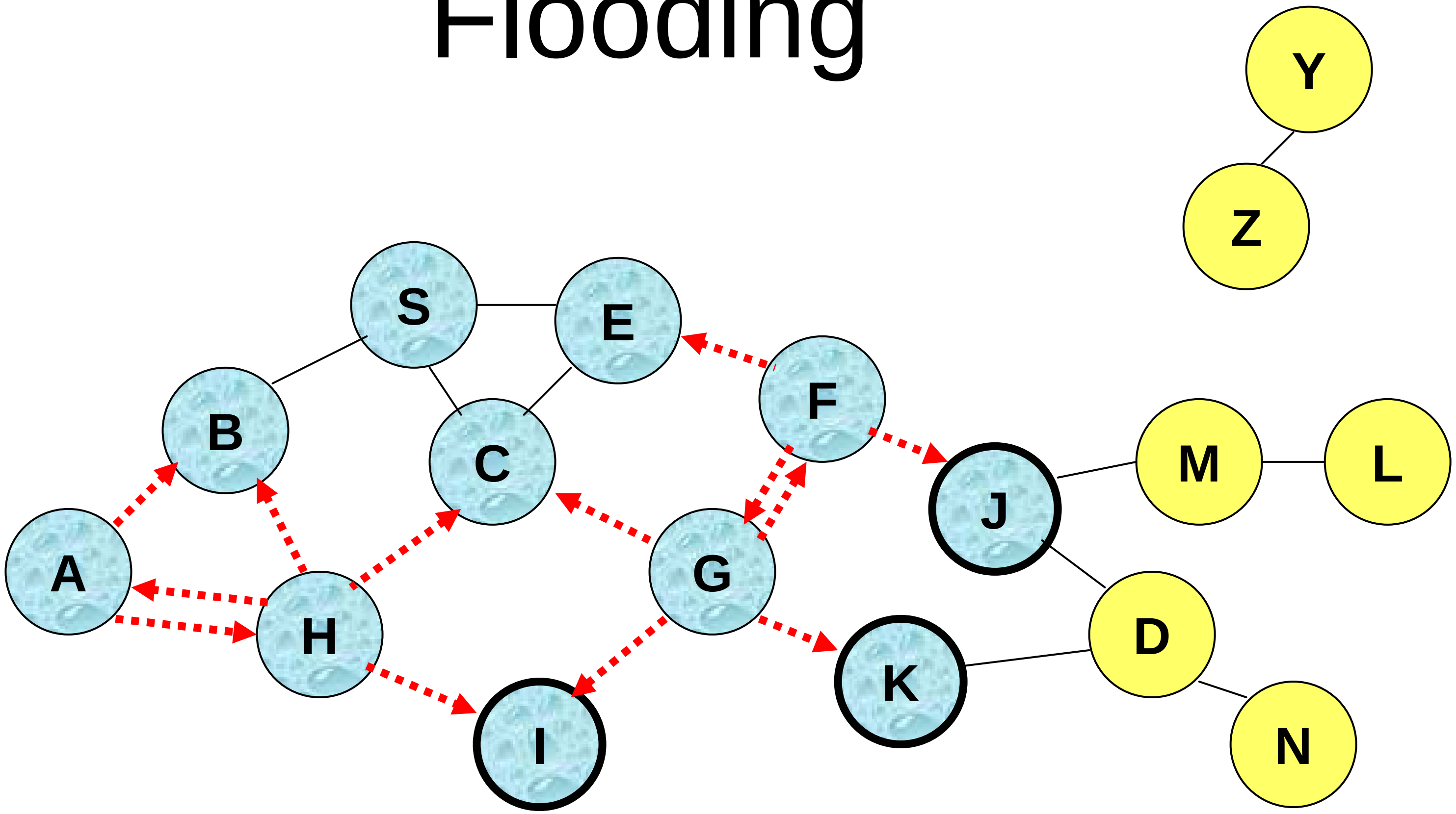
Represents transmission of packet P

Flooding



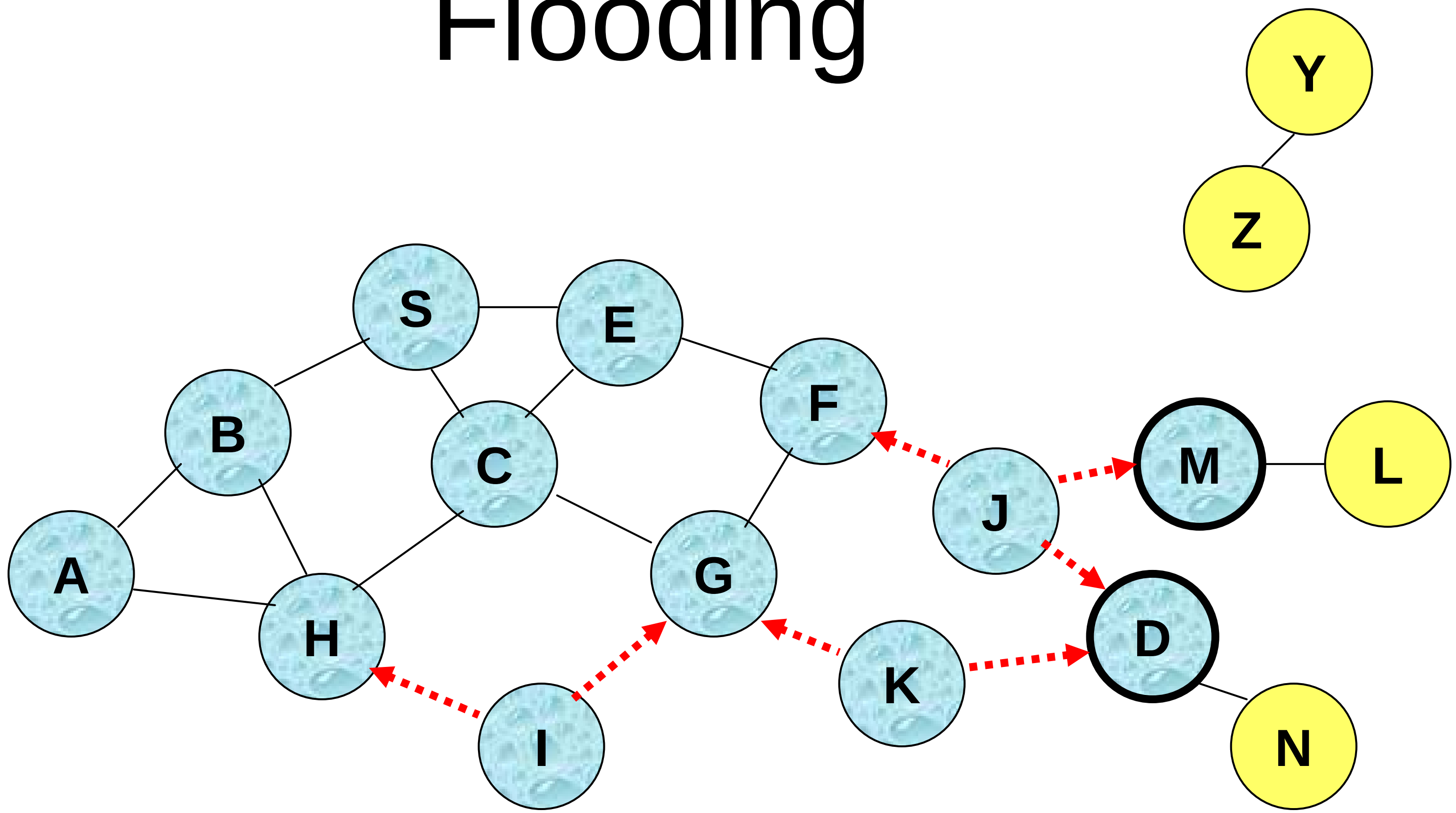
- **Node H receives packet P from two neighbors:**
  potential for collision

Flooding



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

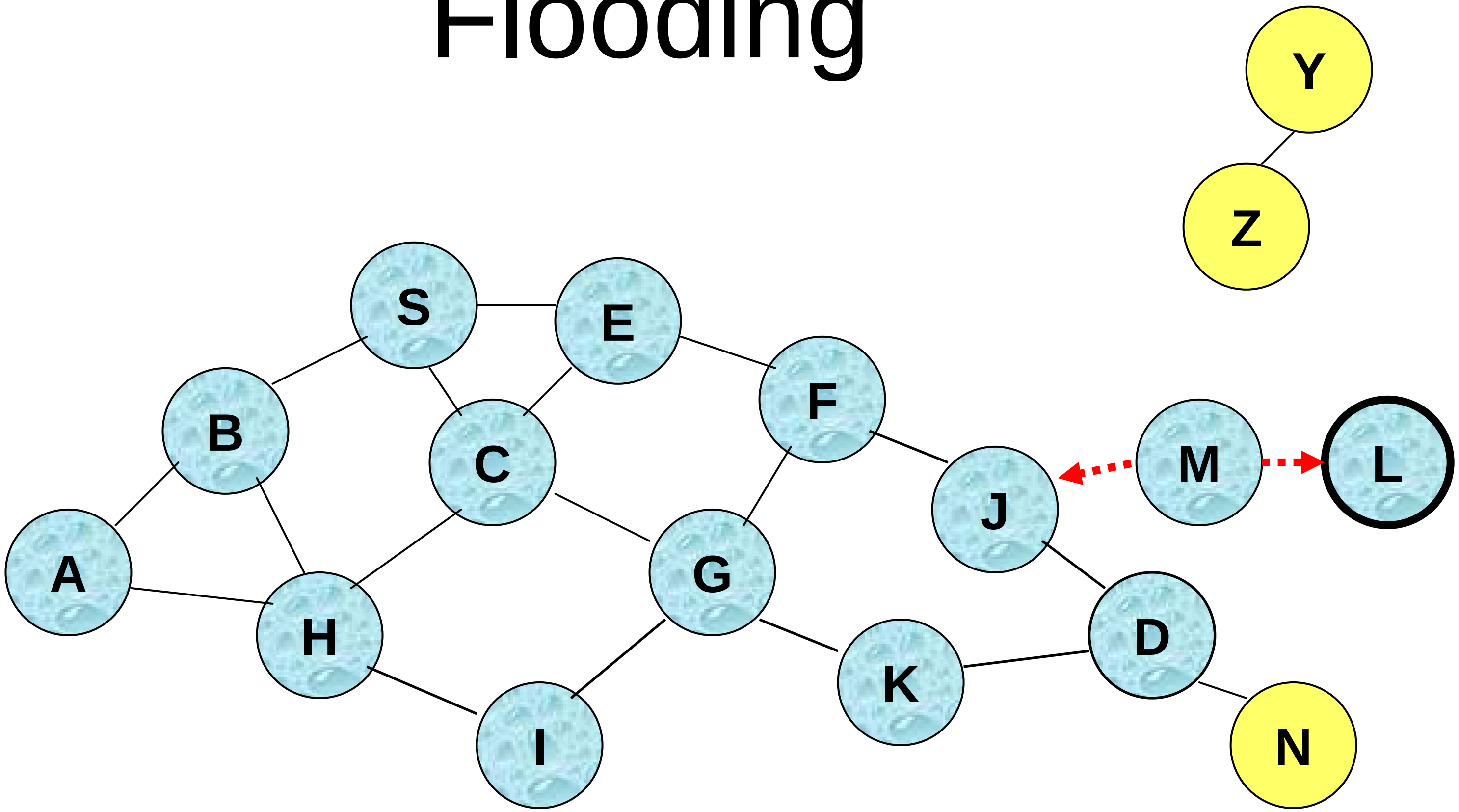
Flooding



- **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**

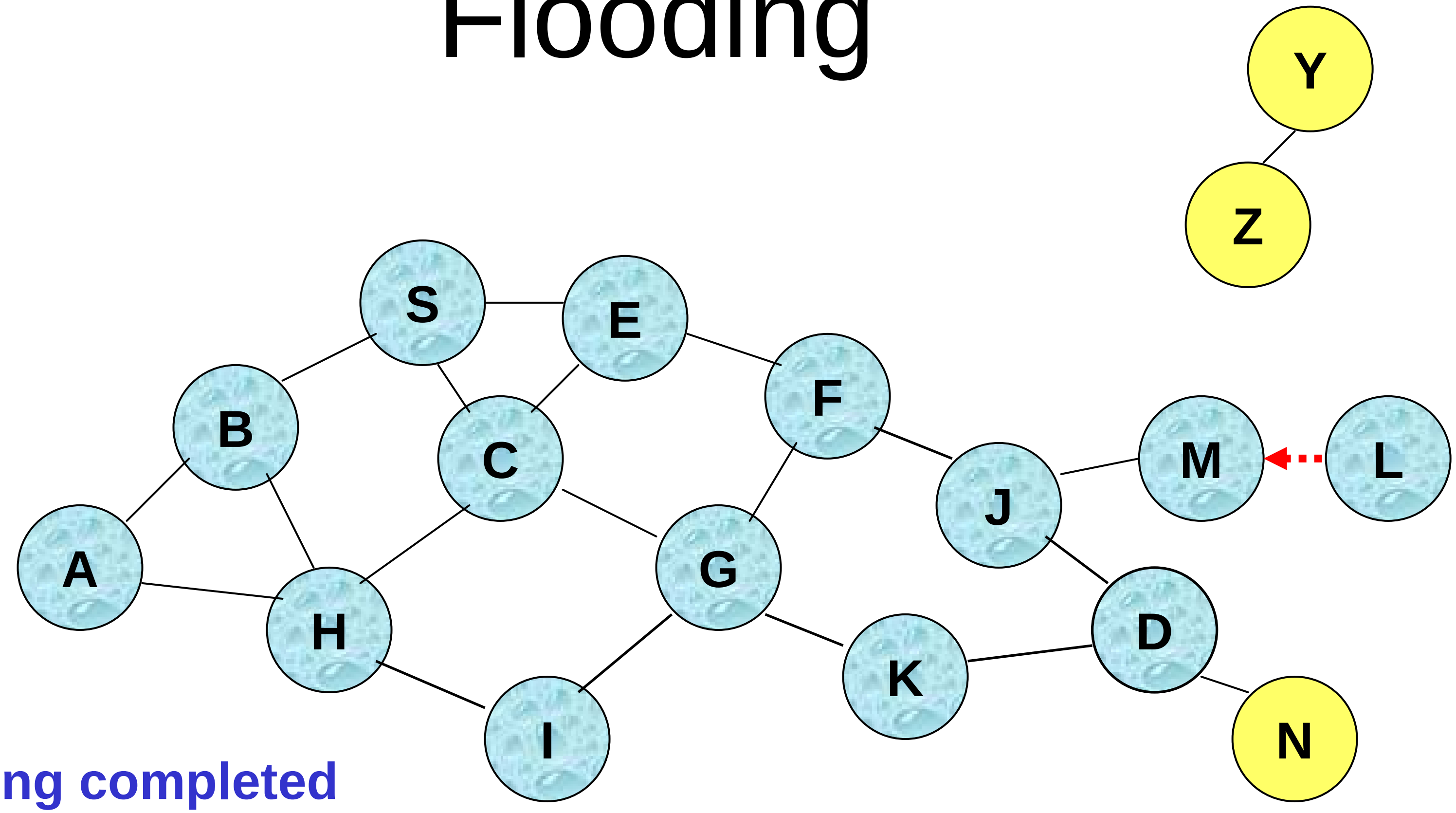
  => **Packet P may not be delivered to node D at all, despite the use of flooding**

Flooding



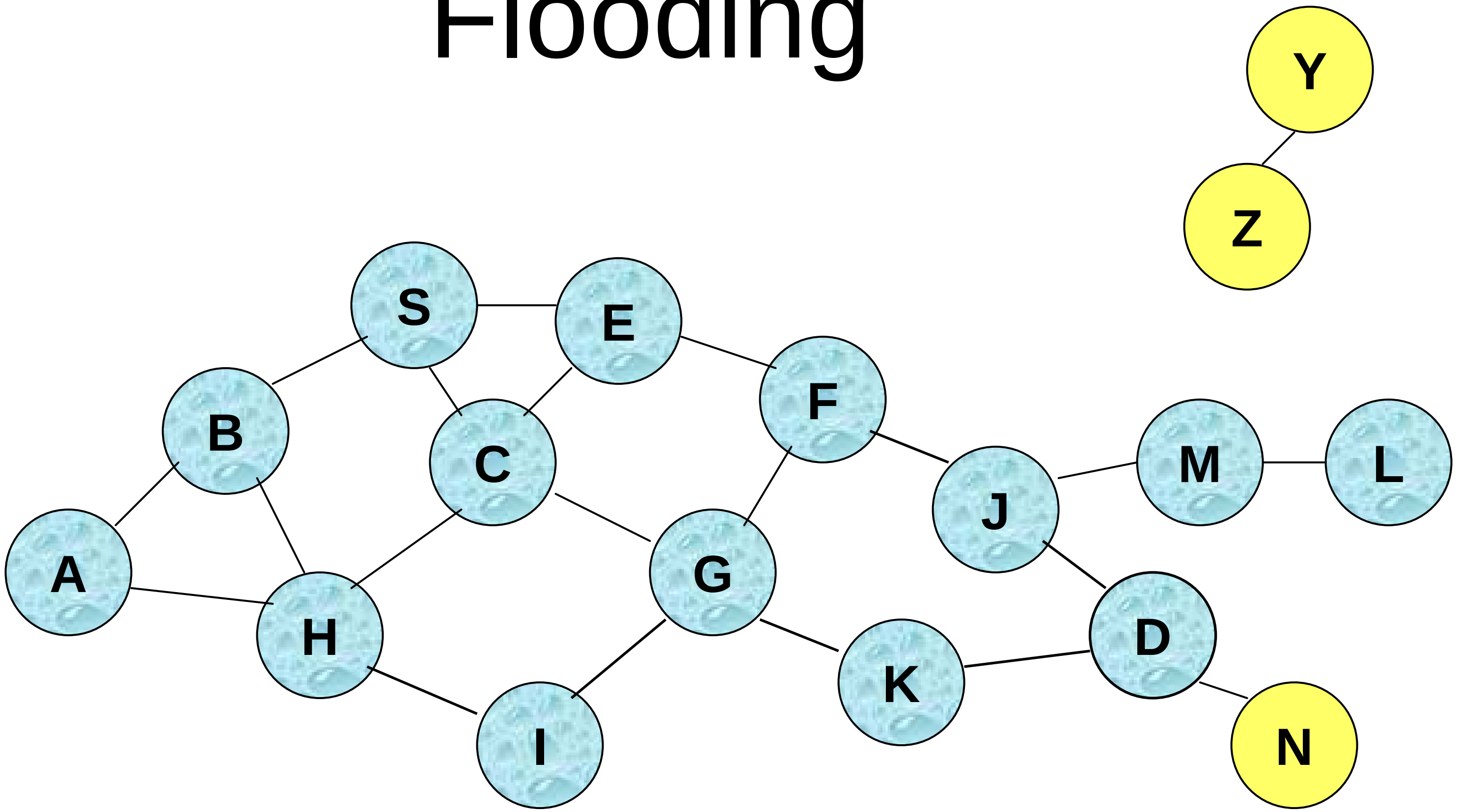
- **Node D does not forward packet P, because node D is the intended destination of packet P**

Flooding



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

Flooding



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

Flooding: Advantages

- Simplicity
- Efficient when rate of information transmission is low enough that the overhead of explicit route discovery/maintenance is relatively higher
  - Example: when nodes transmit small data packets relatively infrequently, and many topology changes occur between consecutive packet transmissions
- Potentially higher reliability of data delivery
  - 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
  - hard to implement reliable broadcast delivery without significantly increasing overhead
    - e.g. broadcasting in IEEE 802.11 MAC is unreliable

Flooding of Control Packets

- Many protocols perform flooding of control packets, instead of data packets

- The control packets are used to discover routes

- Discovered routes are subsequently used to send data packets

Routing Protocols: Classifications

- Proactive protocols
  - Determine routes independent of traffic pattern
  - Traditional link-state and distance-vector routing protocols are proactive
    - these protocols are used in wired networks
- Reactive protocols
  - Maintain routes only if needed
  - e.g. DSR, AODV.

Trade-Off

- Latency of route discovery
  - Proactive protocols have lower latency since routes are maintained at all times
  - Reactive protocols have higher latency because a route from X to Y will be found only when X attempts to send to Y
- Overhead of route discovery/maintenance
  - Proactive protocols may have higher overhead due to continuous route updating
  - Reactive protocols may have lower overhead since routes are determined only if needed

Algorithm 1

Dynamic Source Routing

Dynamic Source Routing (DSR)

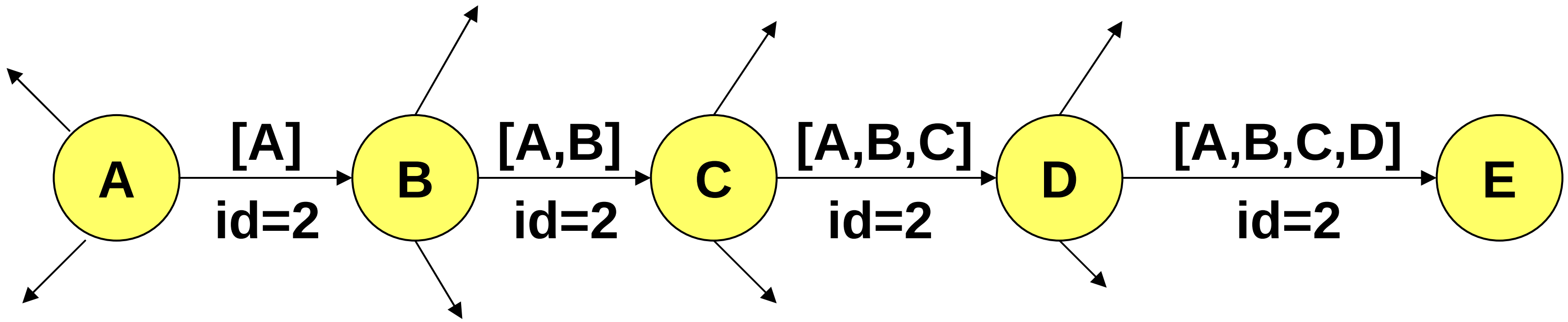
Two mechanisms :

- Route discovery is initiated when S wants to send a packet to D, but does not know a route to D.
- Route maintenance is initiated when the network topology has changed such that a link along the current route from S to D no longer exists.

Route Discovery

- The source S floods Route Request (RREQ)
  - Each RREQ identifies the source and destination, and contains a unique request ID, determined by the source.
- Each node appends its own identifier when forwarding RREQ.

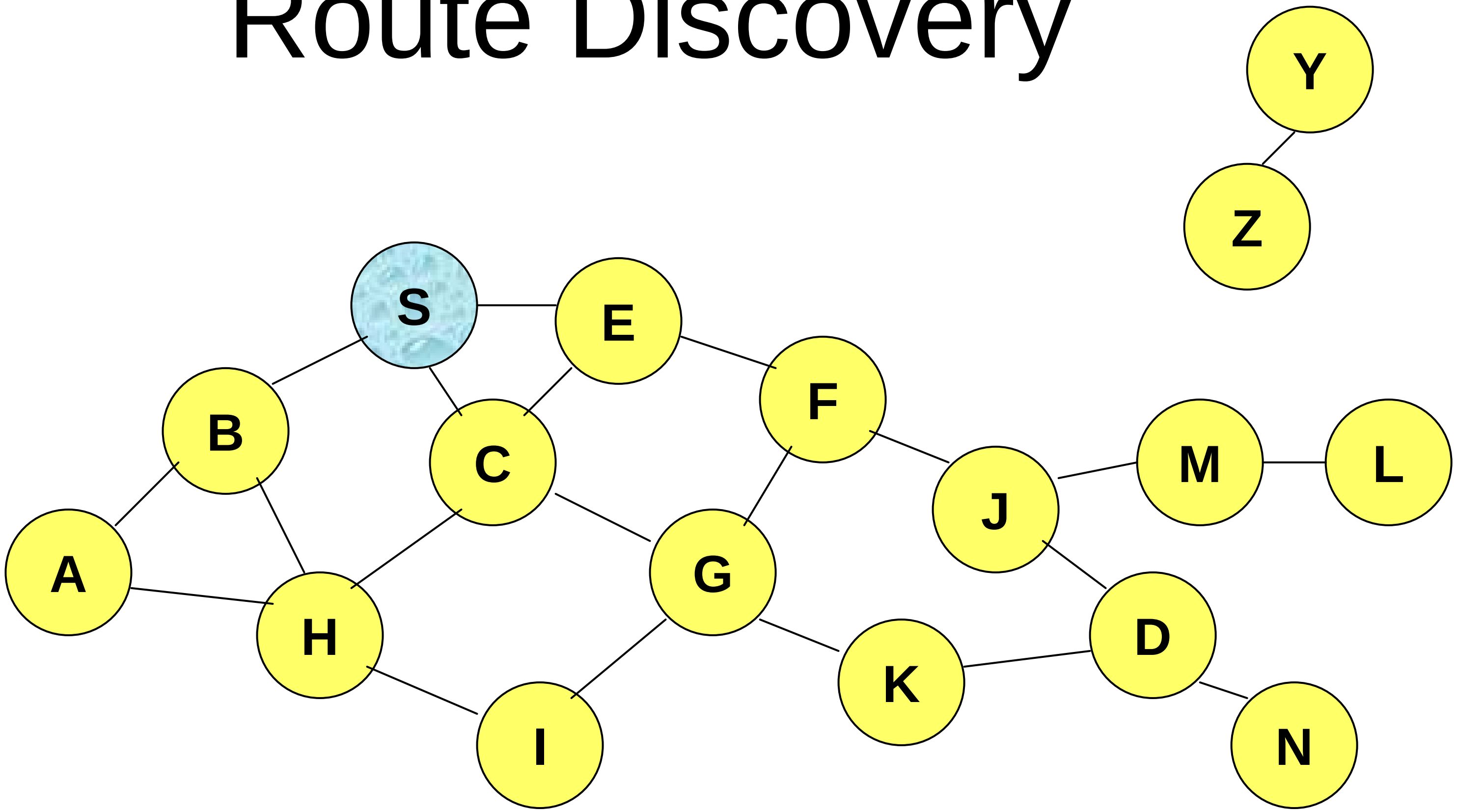
Route Discovery Example

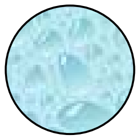


A is the source, E is the destination.

Note that request id does not change when the RREQ propagates.

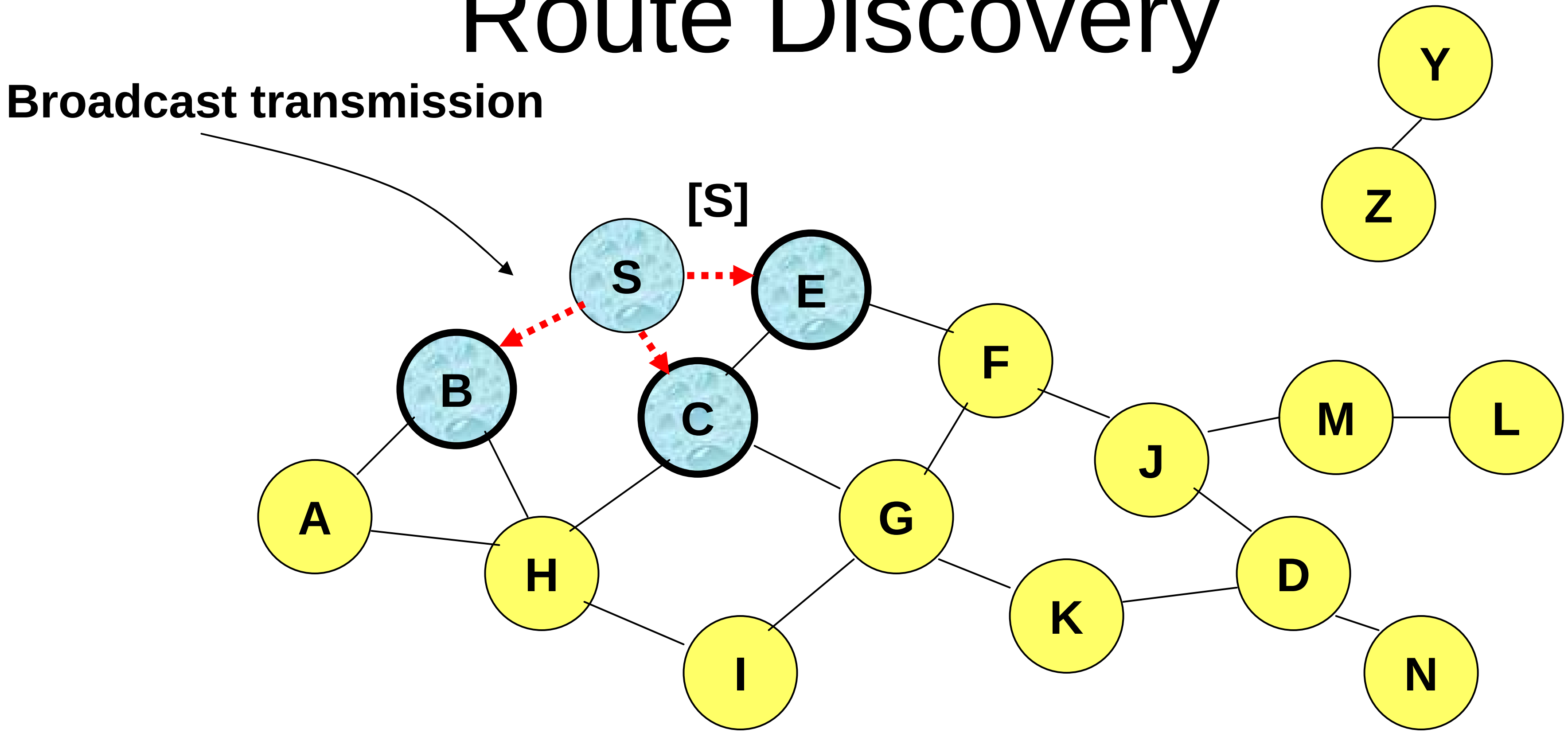
Route Discovery





Represents a node that has received RREQ for D from S

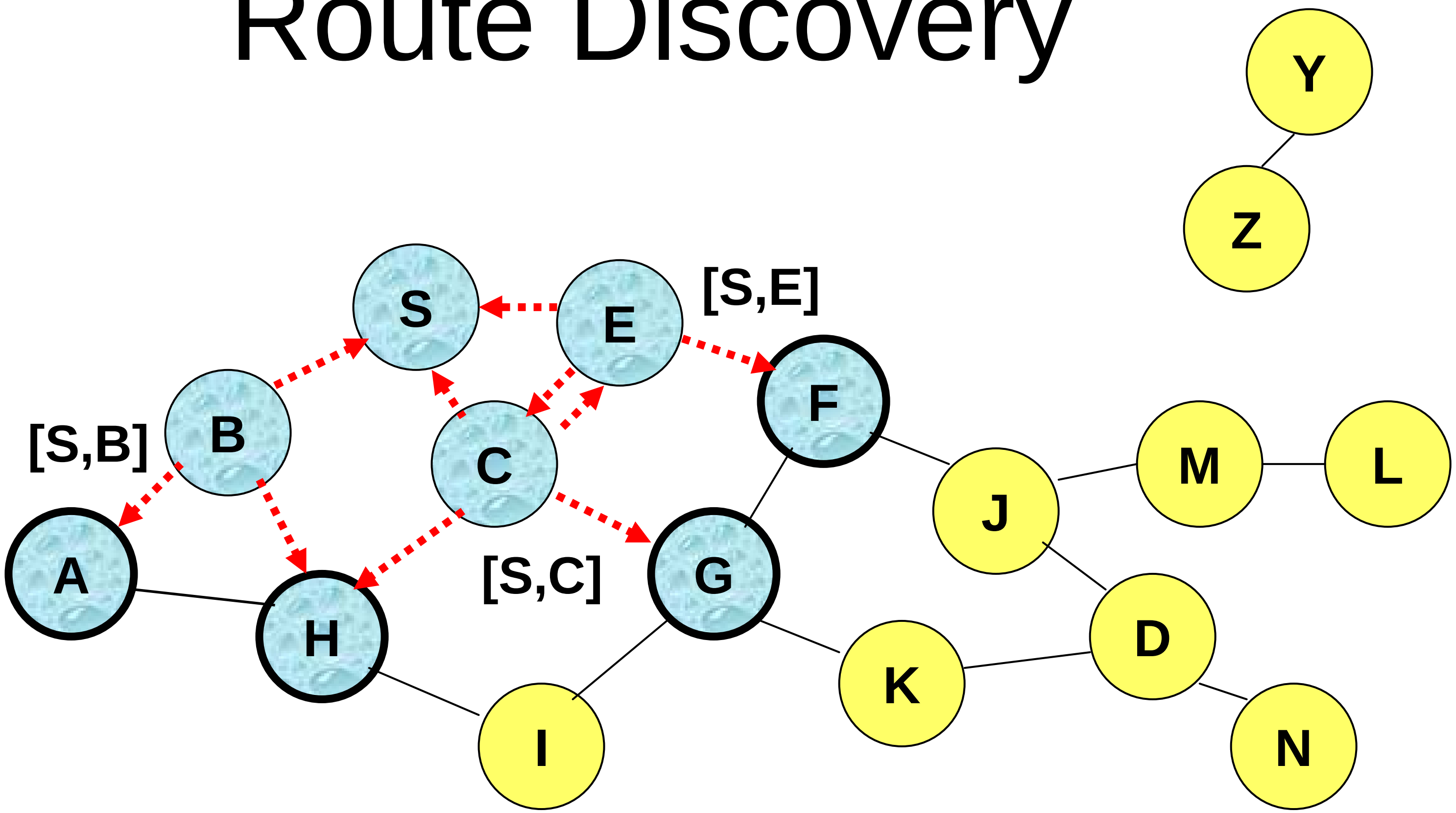
Route Discovery



Represents transmission of RREQ

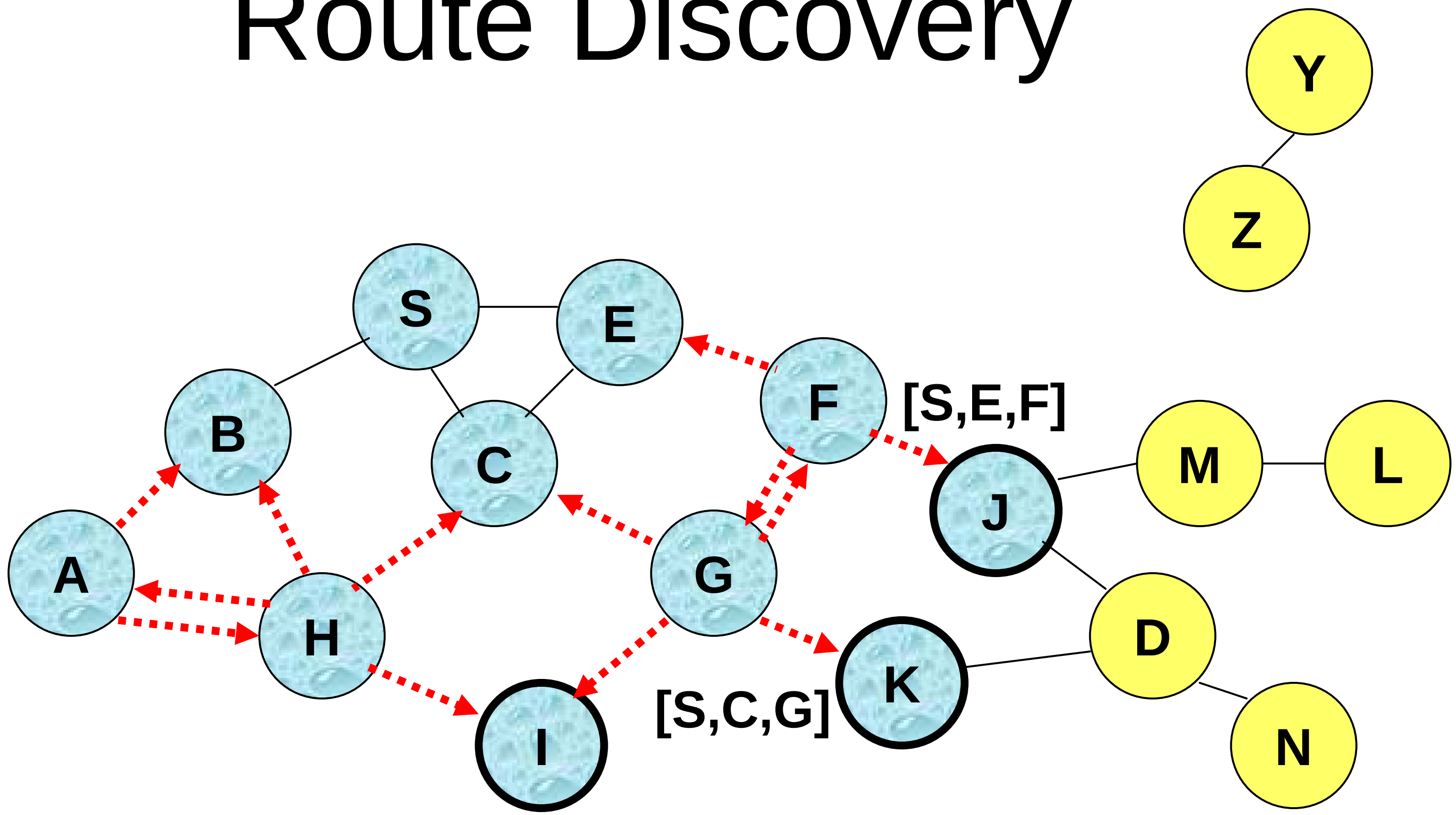
[X,Y] Represents list of identifiers appended to RREQ

Route Discovery



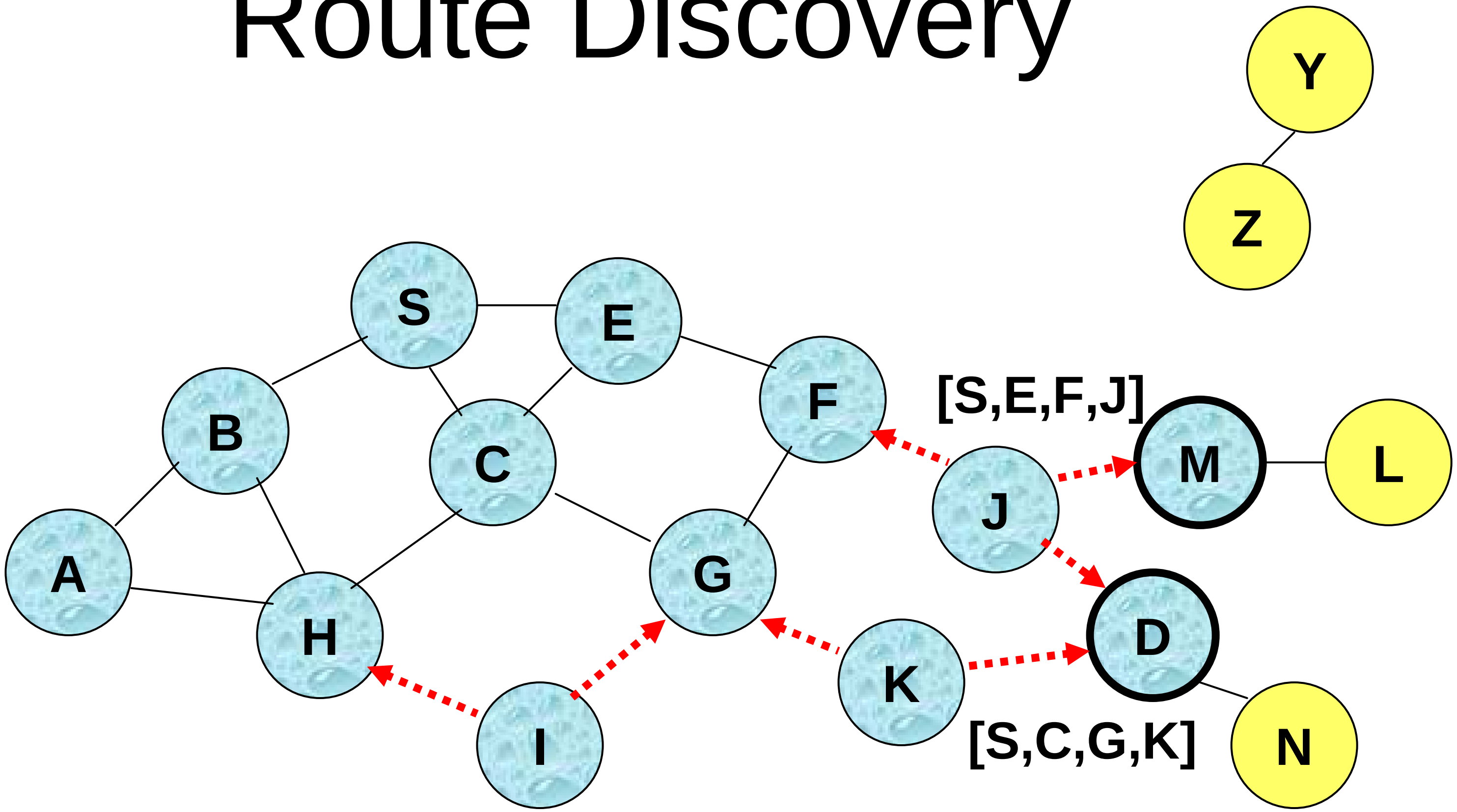
- **H receives packet RREQ from two neighbors: potential for collision**

Route Discovery



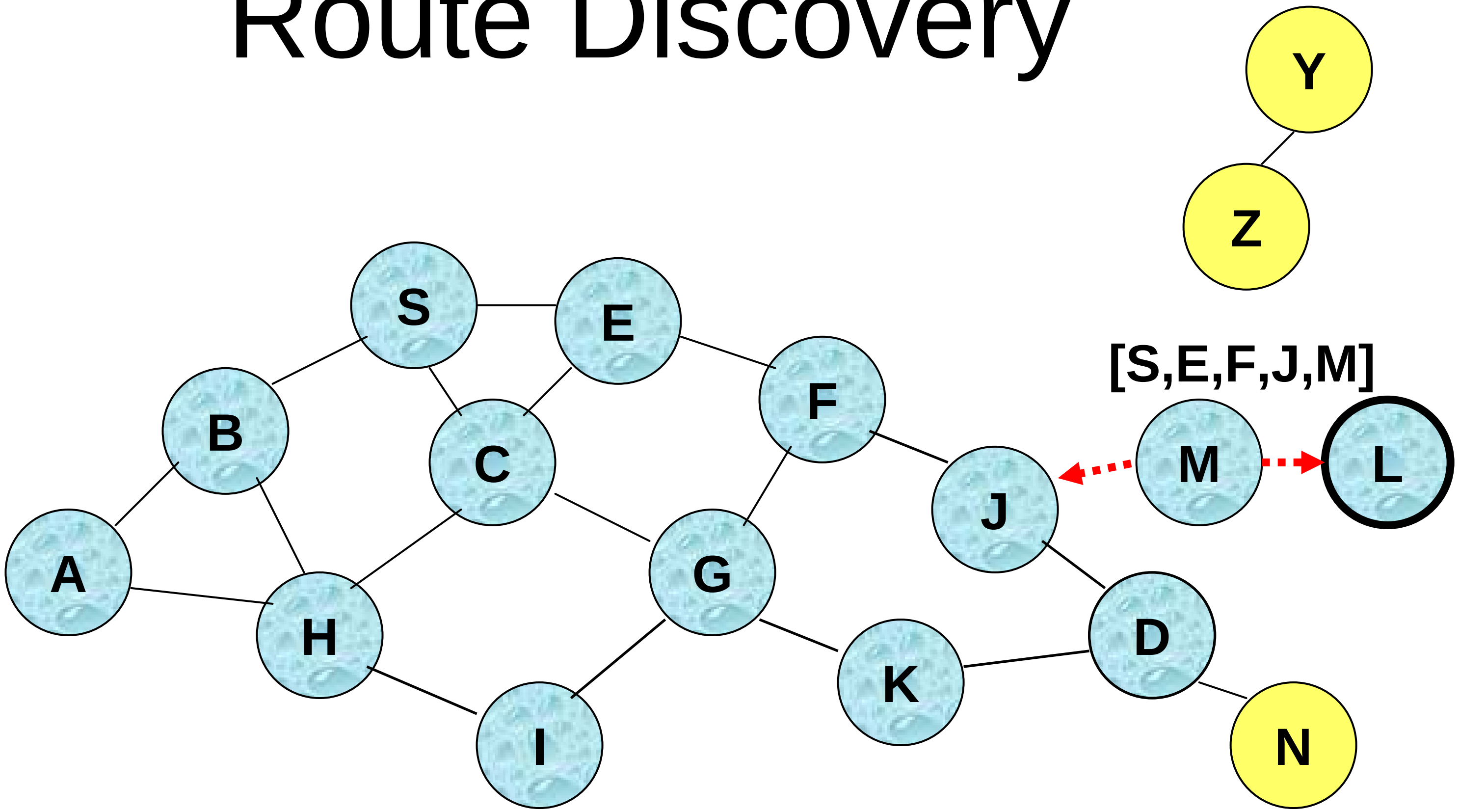
- C receives RREQ from G and H, but does not forward it again, because C has already forwarded RREQ once (i.e. same request id)

Route Discovery



- Both J and K broadcast RREQ to D
- Since J and K are hidden from each other, their transmissions may collide

Route Discovery

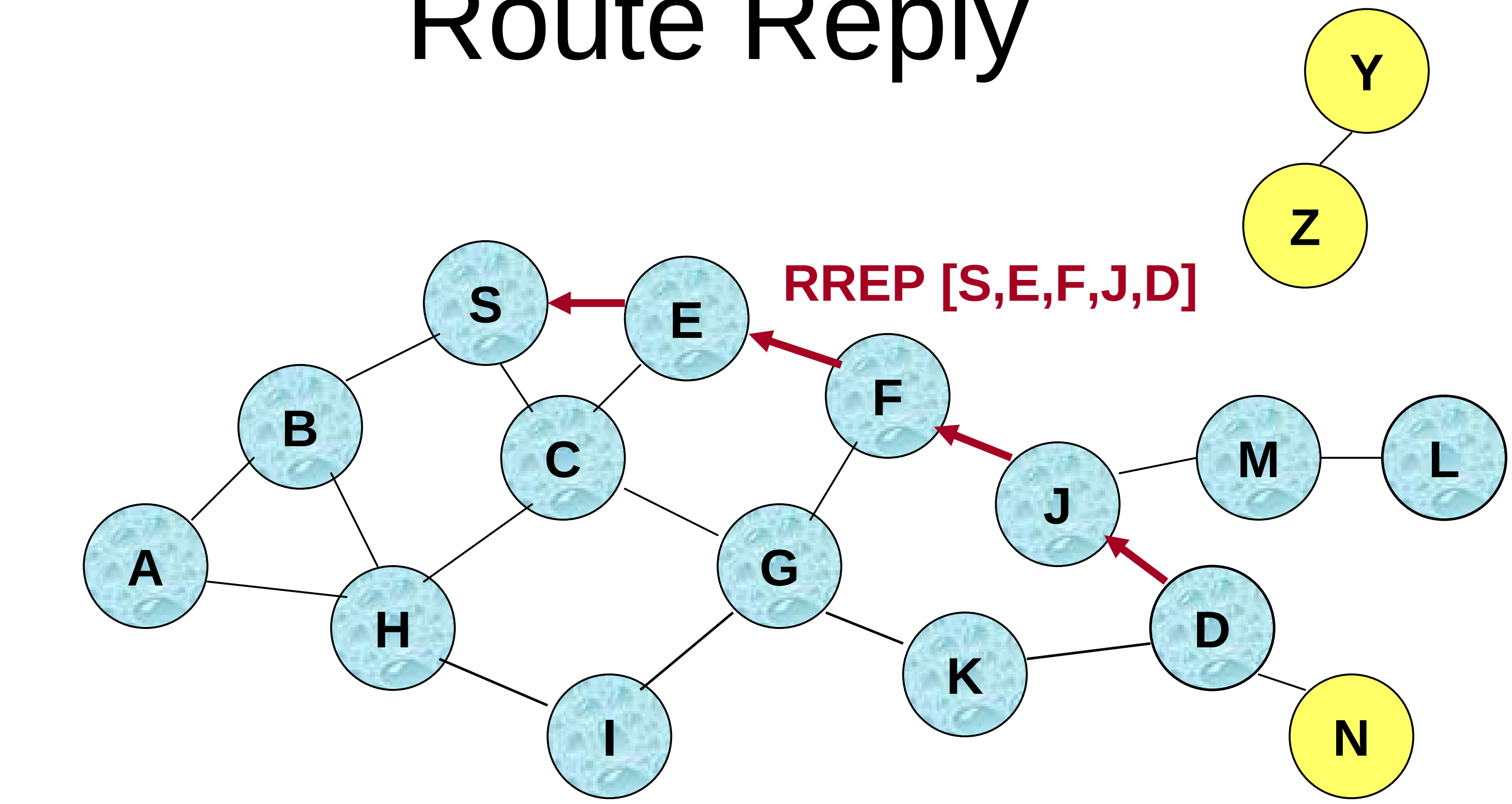


- D **does not forward** RREQ, because it is the **intended target** of the route discovery

Route Discovery

- Destination D on receiving the first RREQ, sends a Route Reply (RREP)
- If the links are bi-directional, RREP is sent on a route obtained by reversing the route appended to received RREQ
  - RREP includes the route from S to D on which RREQ was received by D

Route Reply



Represents RREP control message

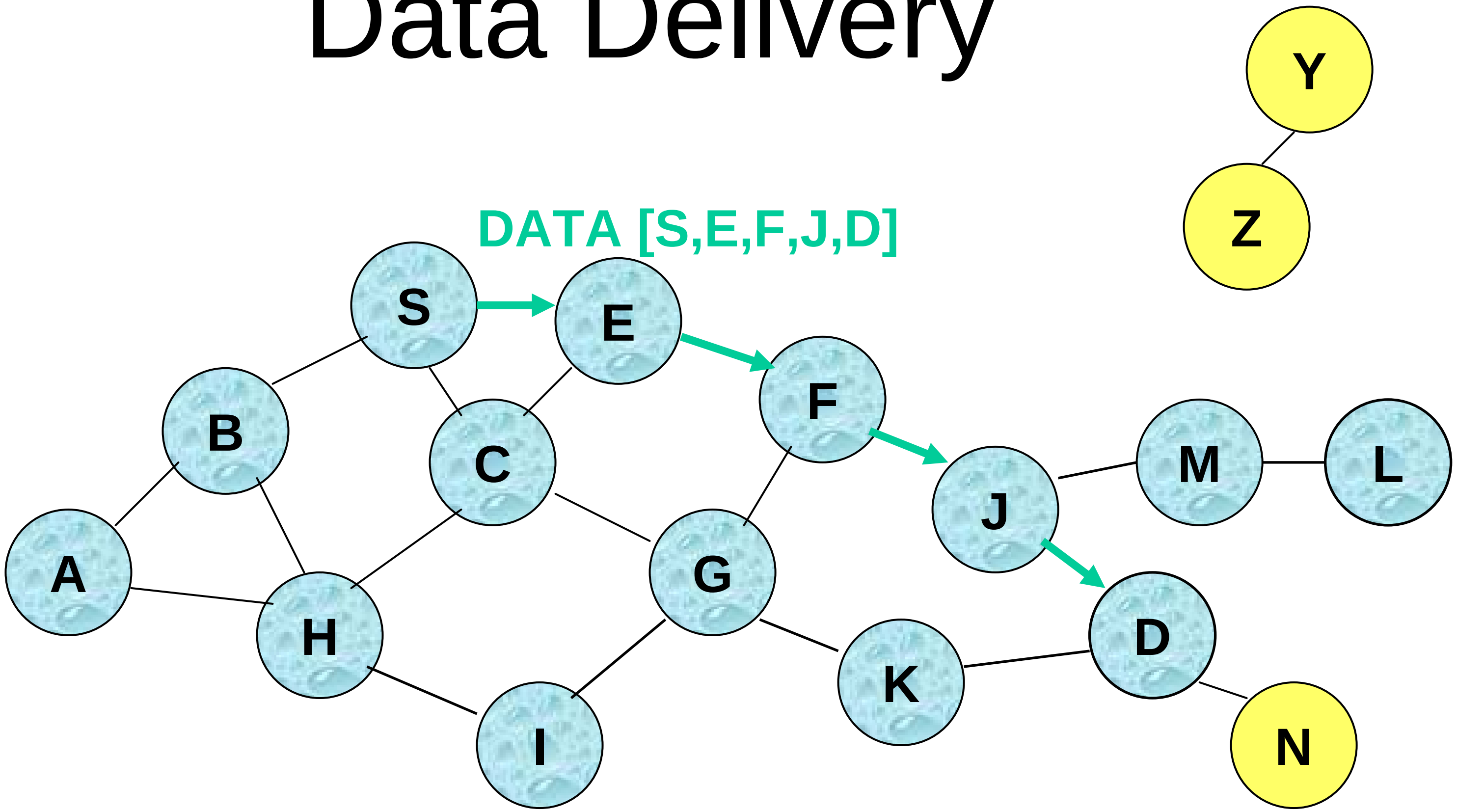
Unidirectional Links

- DSR still works if unidirectional links are allowed.
- RREP may need a route discovery for S from D
  - Unless D already knows a route to node S
  - If a route discovery is initiated by D for a route to S, then the RREP is piggybacked on the RREQ from D.
- However, if 802.11 MAC is used to send data, then links have to be bi-directional
  - since ACK is used

Data Delivery

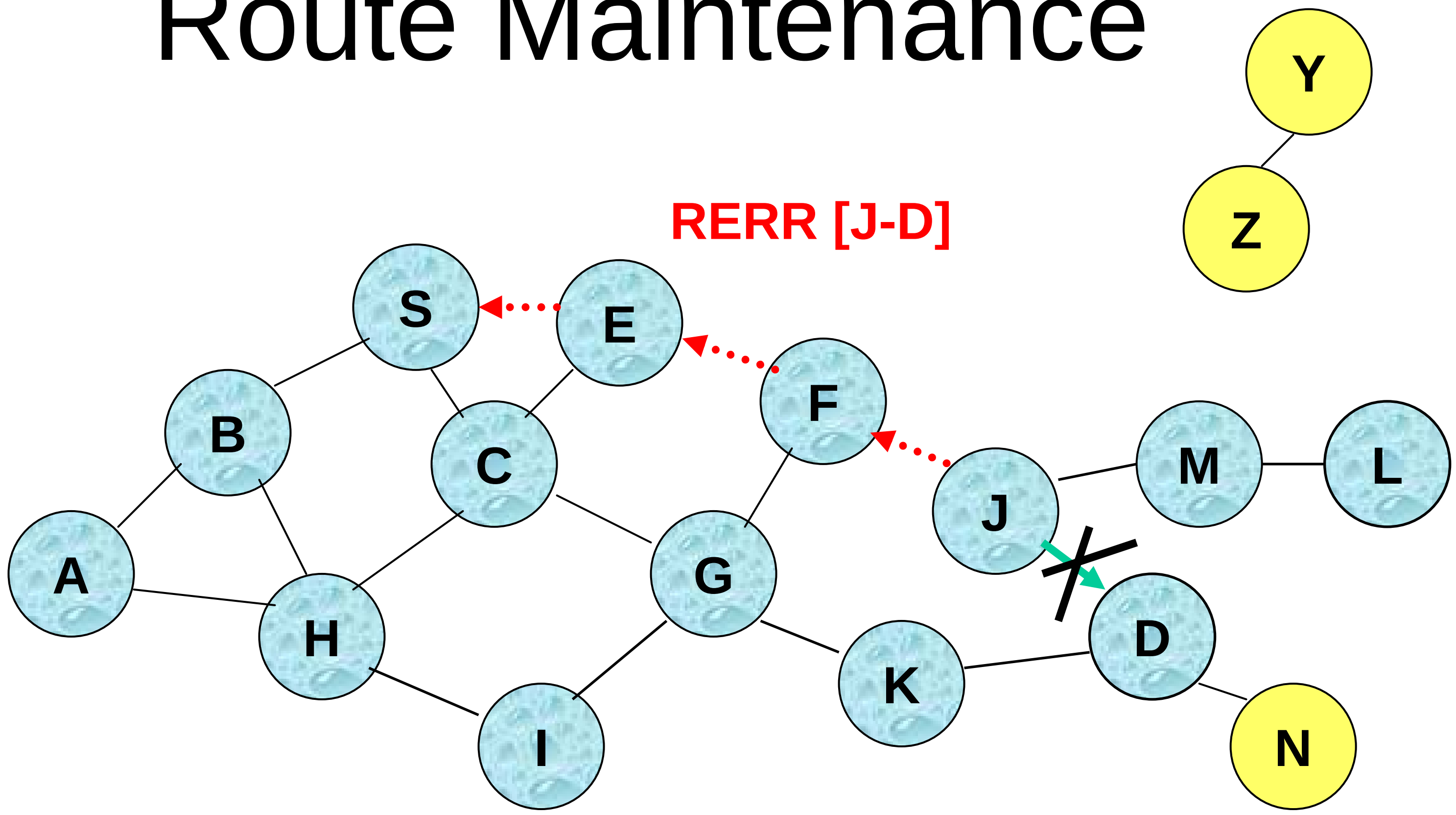
- S on receiving RREP, caches the route included in the RREP
- When S sends a data packet to D, the entire route is included in the packet header
  - hence the name source routing
- Intermediate nodes use the source route included in a packet to determine to whom a packet should be forwarded

Data Delivery



Packet header size grows with route length

Route Maintenance



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

Link Failure Detection

- Q: *How can J know that link J-D is broken?*
  - No ACK received from MAC protocol (such as 802.11).
  - If mechanism not available in MAC layer, J may set a bit in the packet's header to request that a DSR-specific ACK be returned by D.
    - this ACK may return from a different path if links are unidirectional.

Additional Route Discovery Features

- Route Discovery
  - Route Caching
  - Preventing RREP storms

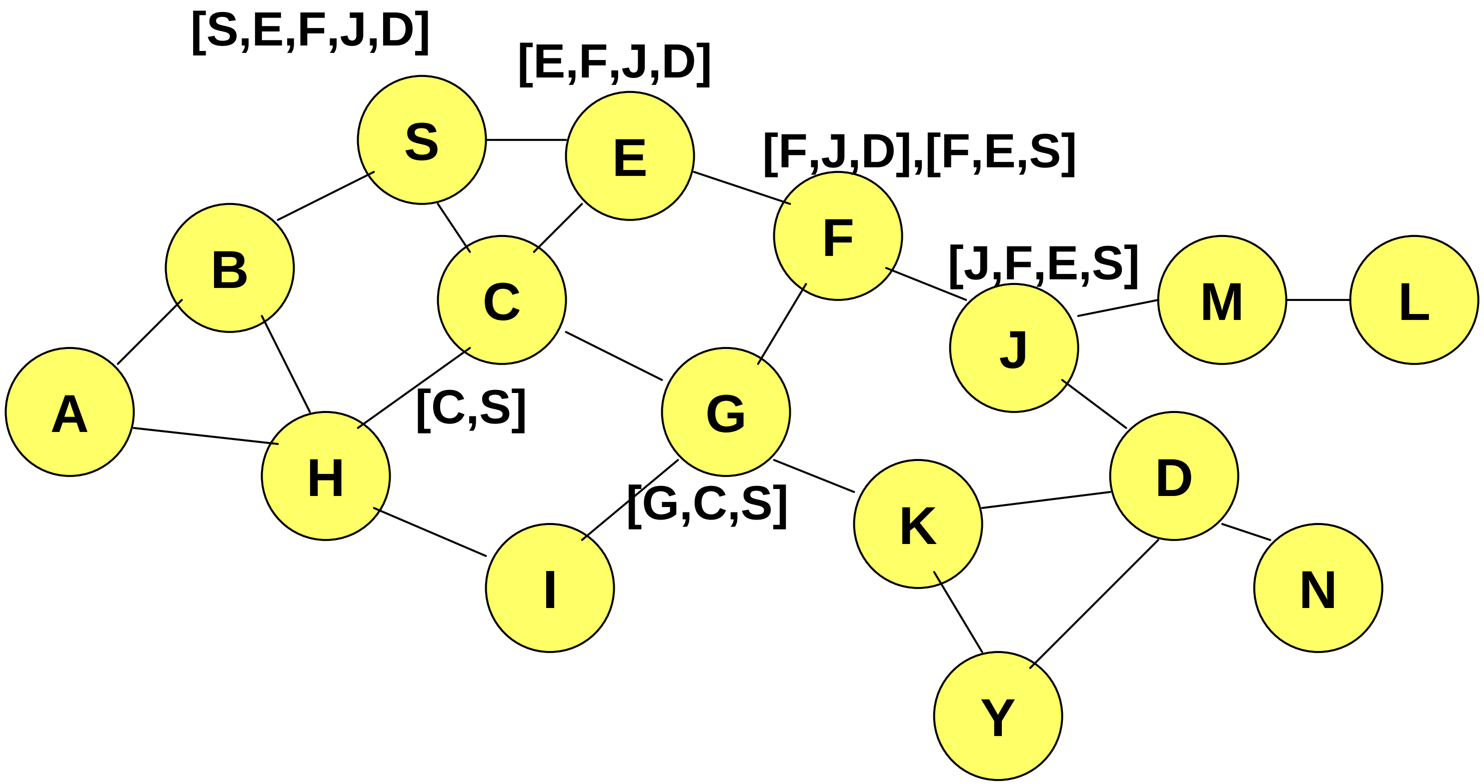
Route Caching

- Each node caches a new route it learns *by any means.*
- Examples:
  - When S finds route [S,E,F,J,D] to D, S also learns route [S,E,F] to F
  - When K receives RREQ [S,C,G] destined for D, K learns route [K,G,C,S] to S (if bi-directional links)
  - When F forwards RREP [S,E,F,J,D], F learns route [F,J,D] to D

Use of Route Caching

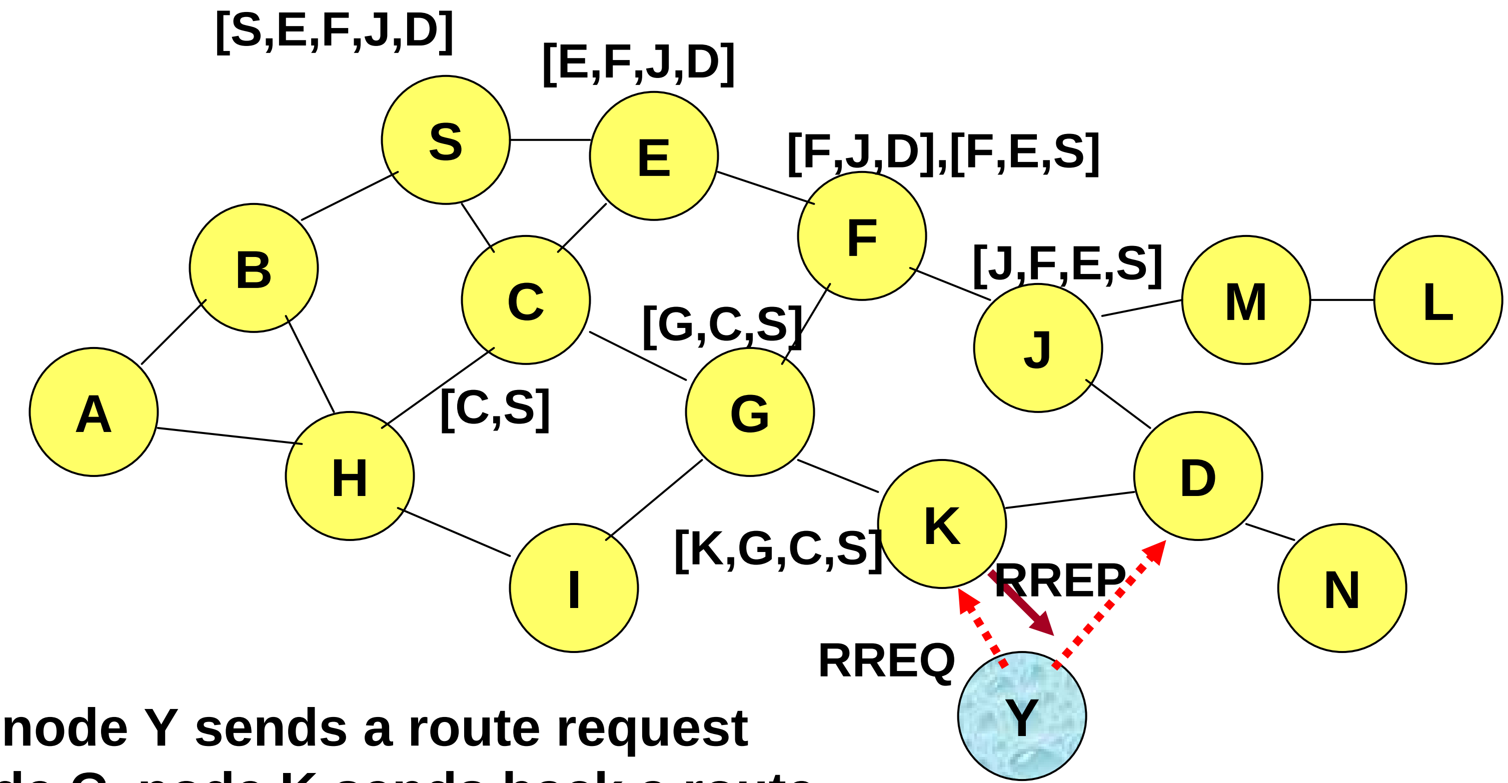
- RREP can be sent by intermediate nodes
  - e.g. X on receiving a RREQ for D can send a RREP if X knows a route to D
- Use of route cache
  - can speed up route discovery
  - can reduce propagation of RREQ

Use of Route Caching



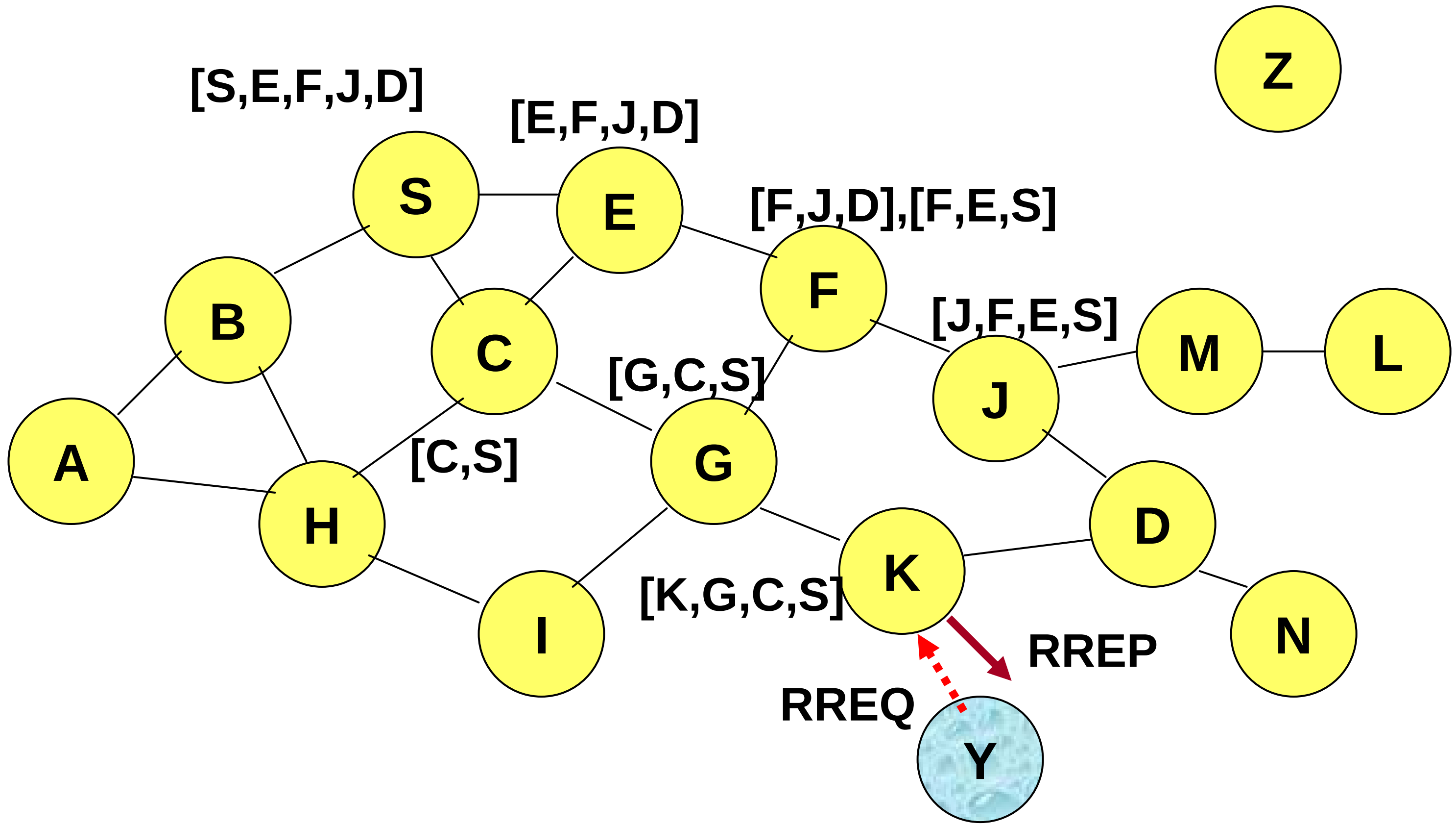
[P,Q,R] Represents cached route at a node

Speed up Route Discovery



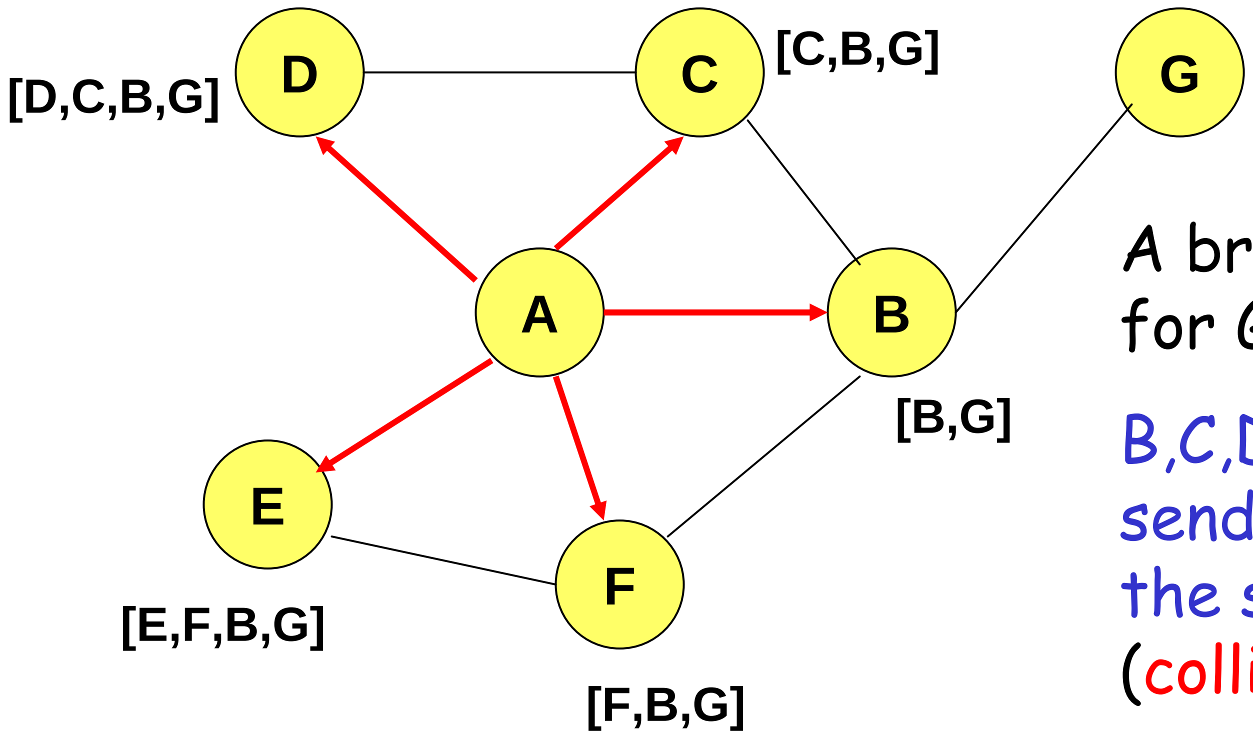
When node Y sends a route request for node C, node K sends back a route reply [Y,K,G,C] to node Y using a locally cached route

Reduce Propagation of RREQ



Assume that there is no link between D and Y.
Route Reply (RREP) from node K limits flooding of RREQ.

RREP Storms



A broadcasts RREQ for G.

B,C,D,E, and F all send RREP at about the same time (collisions!)

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

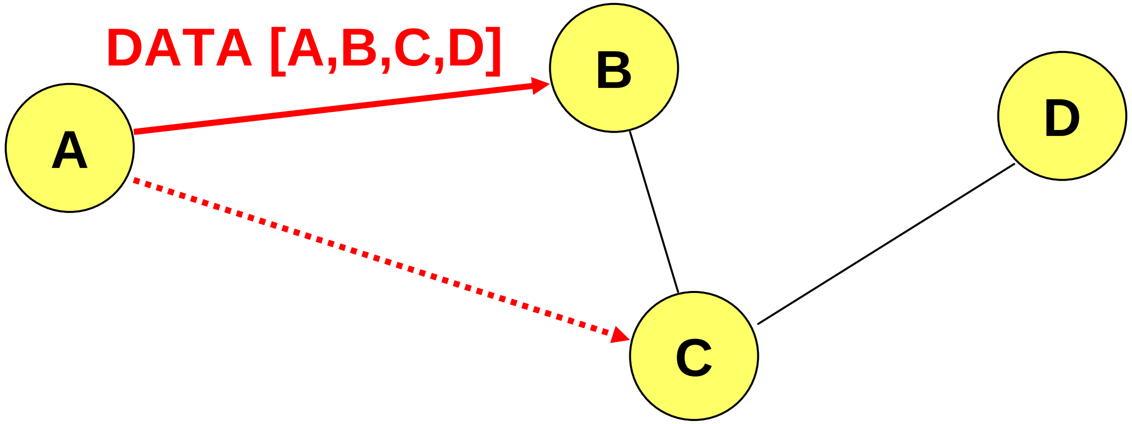
Preventing RREP Storms

- A node should delay sending RREP for a random period $d = H \times (h - 1 + r)$ and listens to the channel.
  - H is a small constant delay (at least twice the maximum wireless link propagation delay).
  - h is the no. of hops from this node to the destination (e.g. $h = 1$ for B, $h = 2$ for F)
  - r is a random number between 0 and 1 (randomizes the transmission time)
- If a node hears data packet from source to destination, it does *not* send its RREP.

Additional Route Maintenance Features

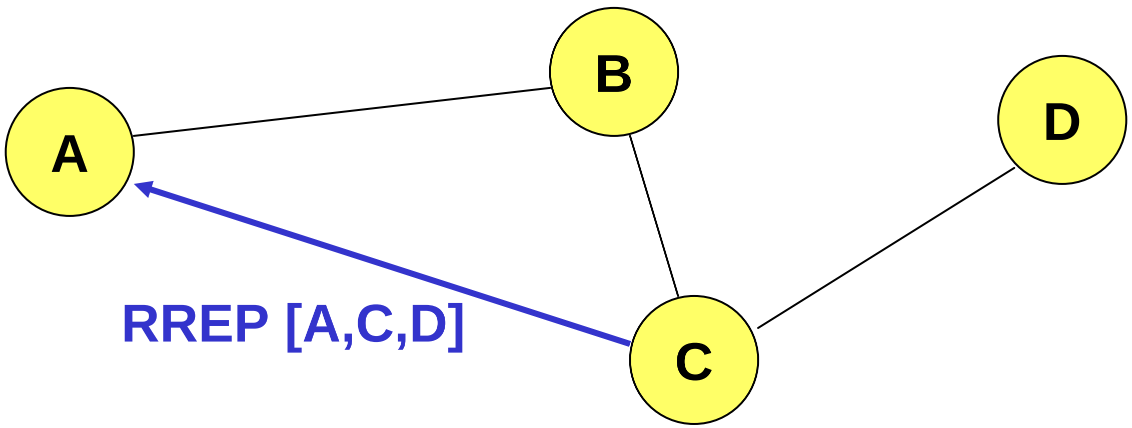
- Route Maintenance
  - Automatic route shortening
  - Increased spreading of REER messages

Automatic Route Shortening



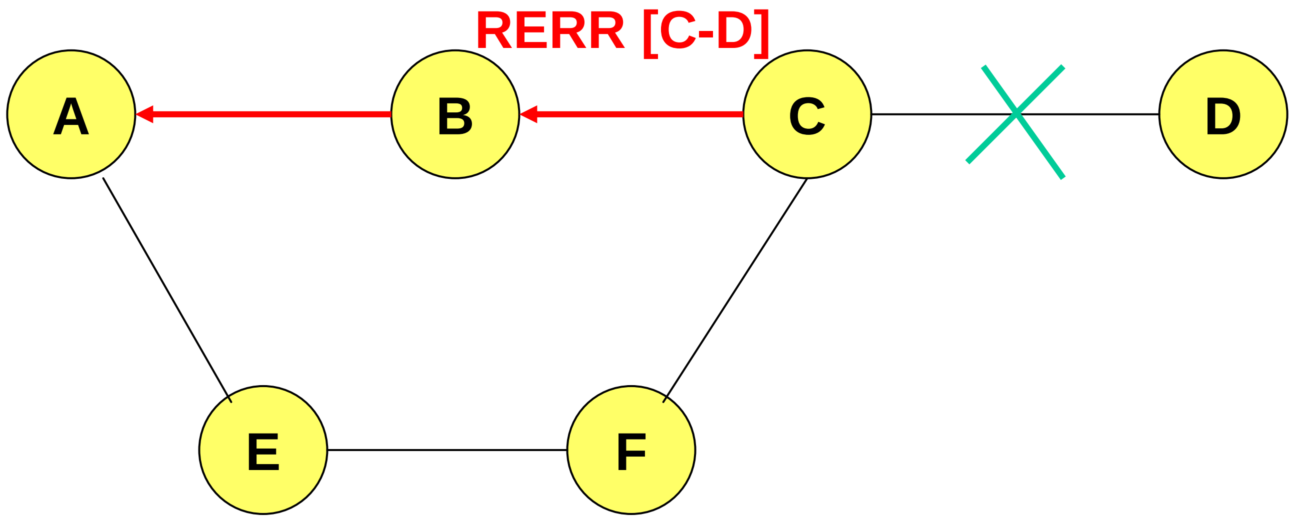
When C overhears a data packet being transmitted from A to B for later forwarding to C, it can infer that B is not needed.

Automatic Route Shortening



C returns a gratuitous RREP to A, which gives a shorter route.

Increased Spreading of RERR

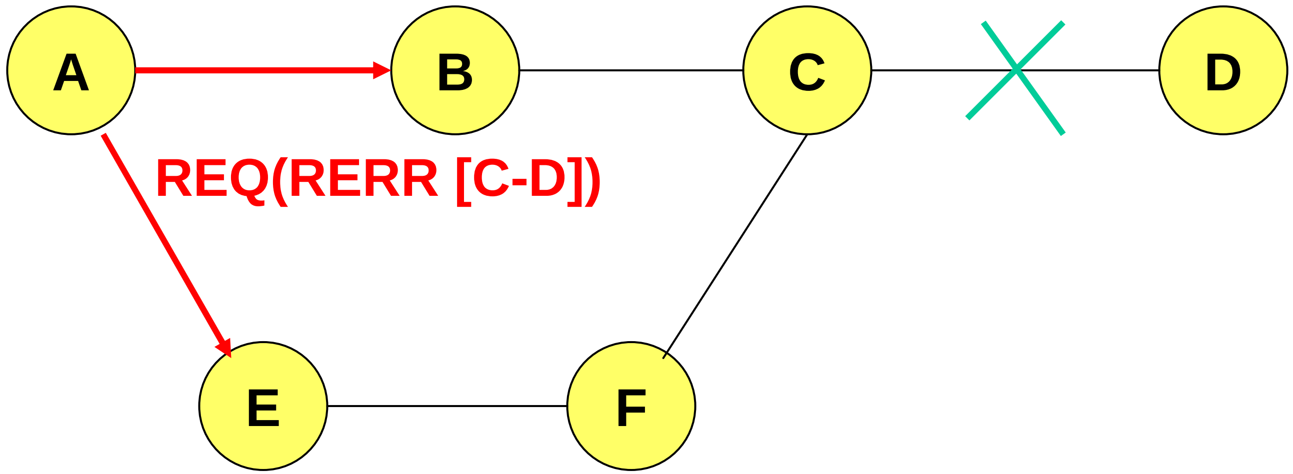


A learns that link from C to D is broken.

Then A initiates a Route Discovery by sending RREQ.

E may return RREP[A,E,F,C,D], which is a stale route.

Increased Spreading of RERR



A piggybacks a copy of RERR to the REQ, ensuring that any RREP in response does not contain a route that assumes the existence of link C-D.

DSR: Advantages

- Periodic transmission of control packets is not required
  - control packet overhead goes to zero when all nodes are stationary and all routes have already been found.
- Support uni-directional links

DSR: Disadvantages

- Packet header size grows with route length due to source routing
- RREQ may reach all nodes in the network
  - large control overhead
- An intermediate node may send RREP using a stale cached route, thus polluting other caches
  - adversely affect performance

Algorithm 2

Ad Hoc On-demand Distance Vector Routing

Ad Hoc On-Demand Distance Vector Routing (AODV)

- DSR includes source routes in packet headers
  - 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
- AODV maintains routes only between nodes which need to communicate
  - also a reactive protocol.

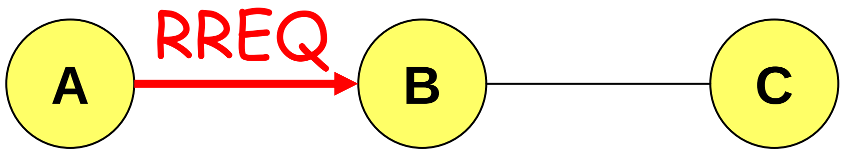
AODV Route Discovery

- RREQ are forwarded in a manner similar to DSR.
- Each RREQ identifies the source and destination, and contains a unique request ID, determined by the source.
- In addition, each RREQ also contains
  - the current sequence number of the source
  - the last known sequence number of the destination.

Reverse Path Setup

- When a node forwards a RREQ, it sets up a reverse route entry for the source in its routing table.
  - AODV assumes bi-directional links
- This reverse route entry contains
  - source node
  - sequence number
  - number of hops
  - neighbor from which the RREQ was received

Example: Reverse Route Entry at Node B

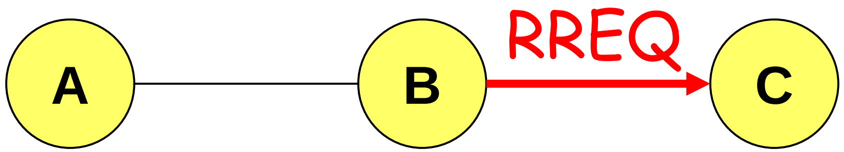


Source ID	Sequence Number	Number of hops	Neighbor ID
A	1	1	A

source ID

source sequence no.

Example: Reverse Route Entry at Node C

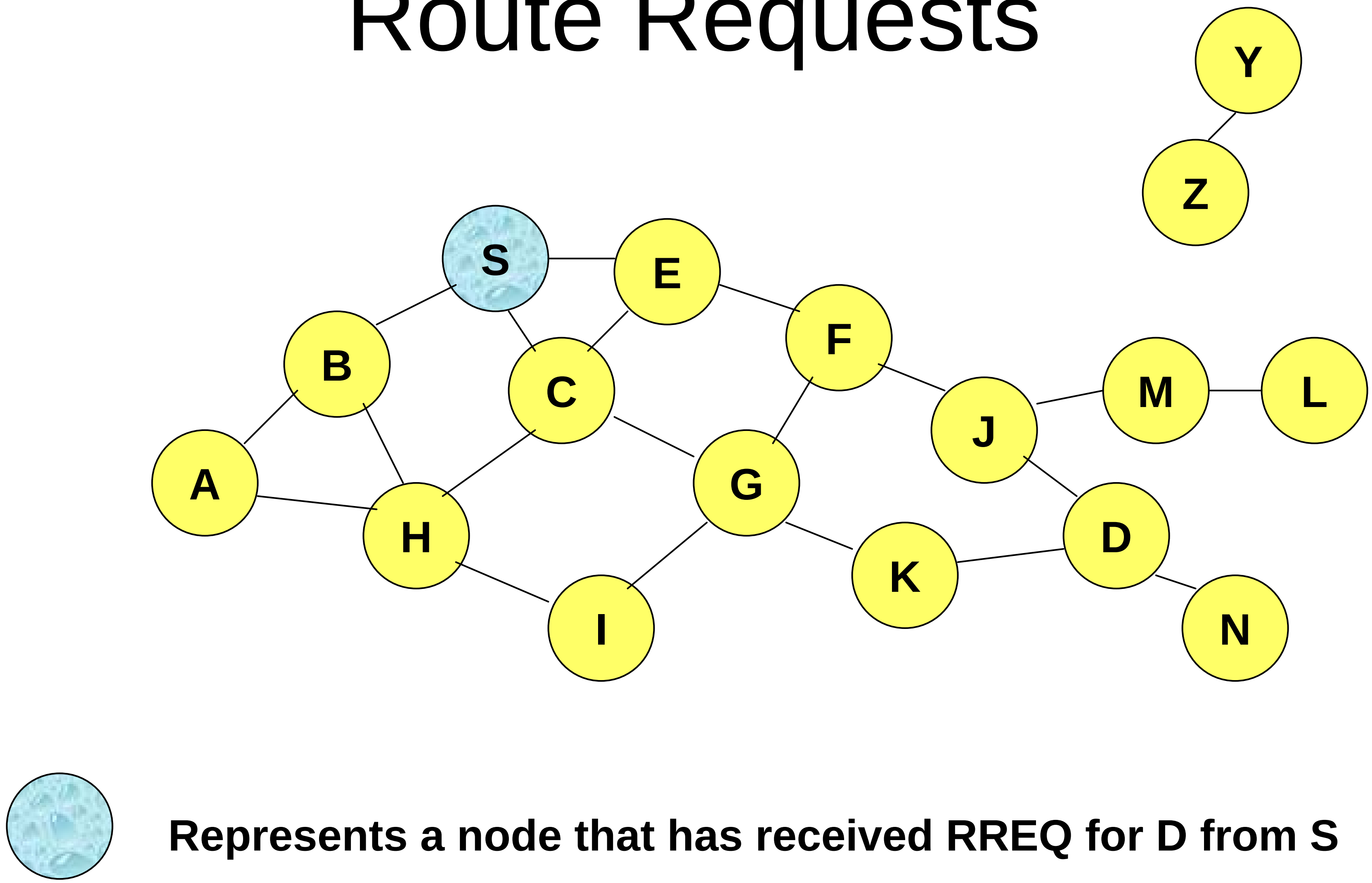


Source ID	Sequence Number	Number of hops	Neighbor ID
A	1	2	B

RREP by Destination

- When the intended destination receives a RREQ, it sends a RREP to the source.
- RREP travels along the reverse path set-up when RREQ is forwarded.

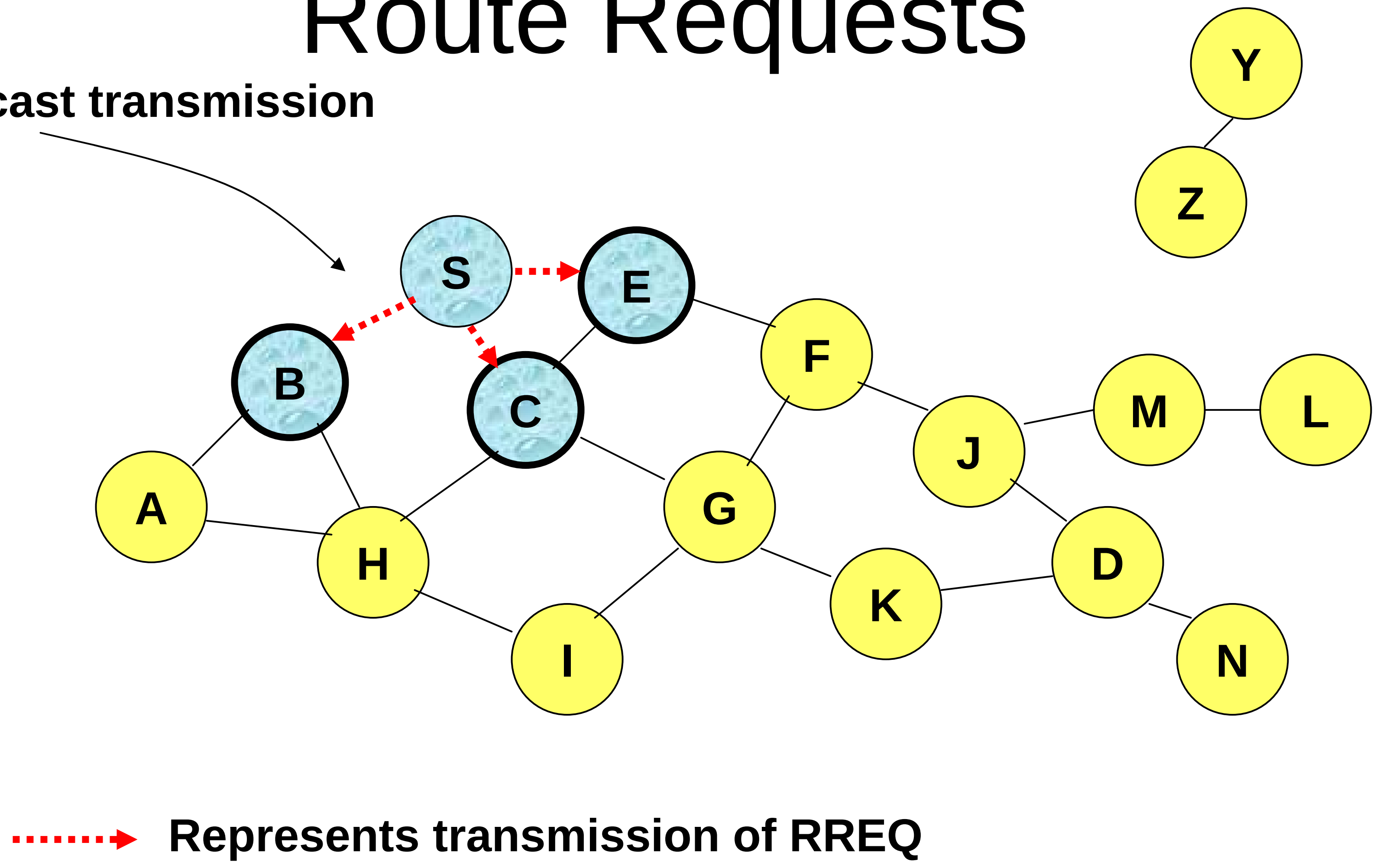
Route Requests



Represents a node that has received RREQ for D from S

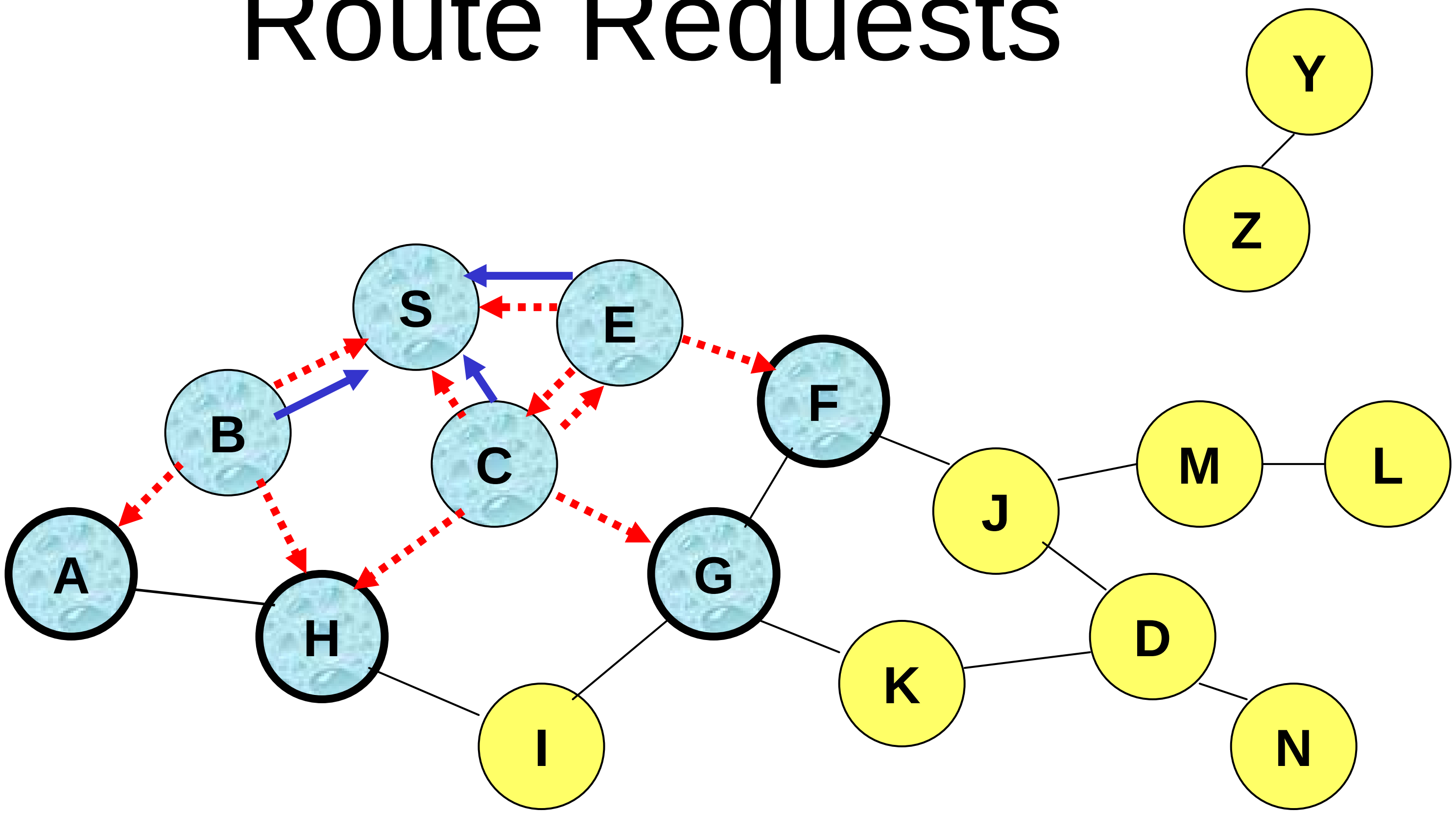
Route Requests

Broadcast transmission



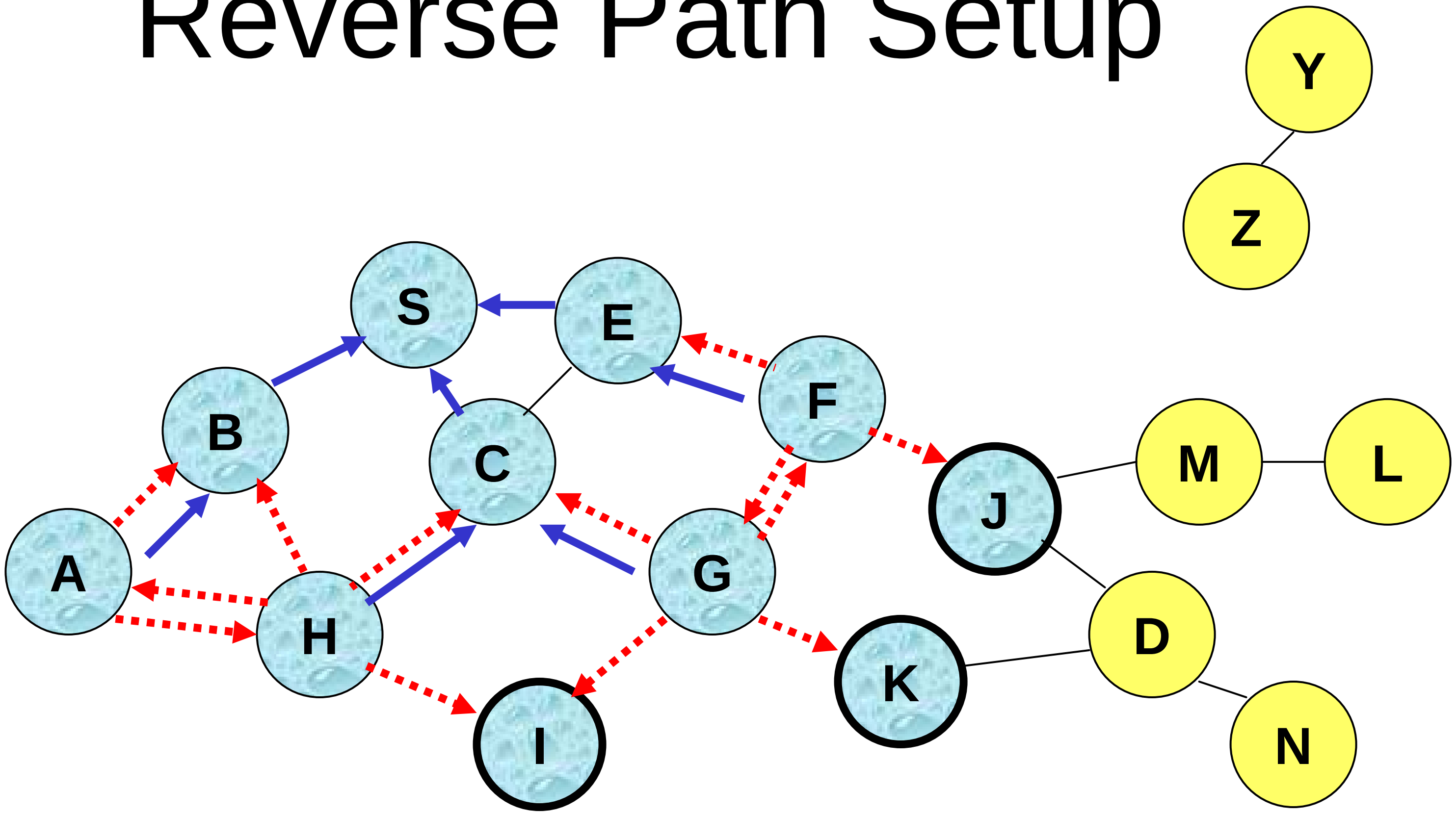
Represents transmission of RREQ

Route Requests



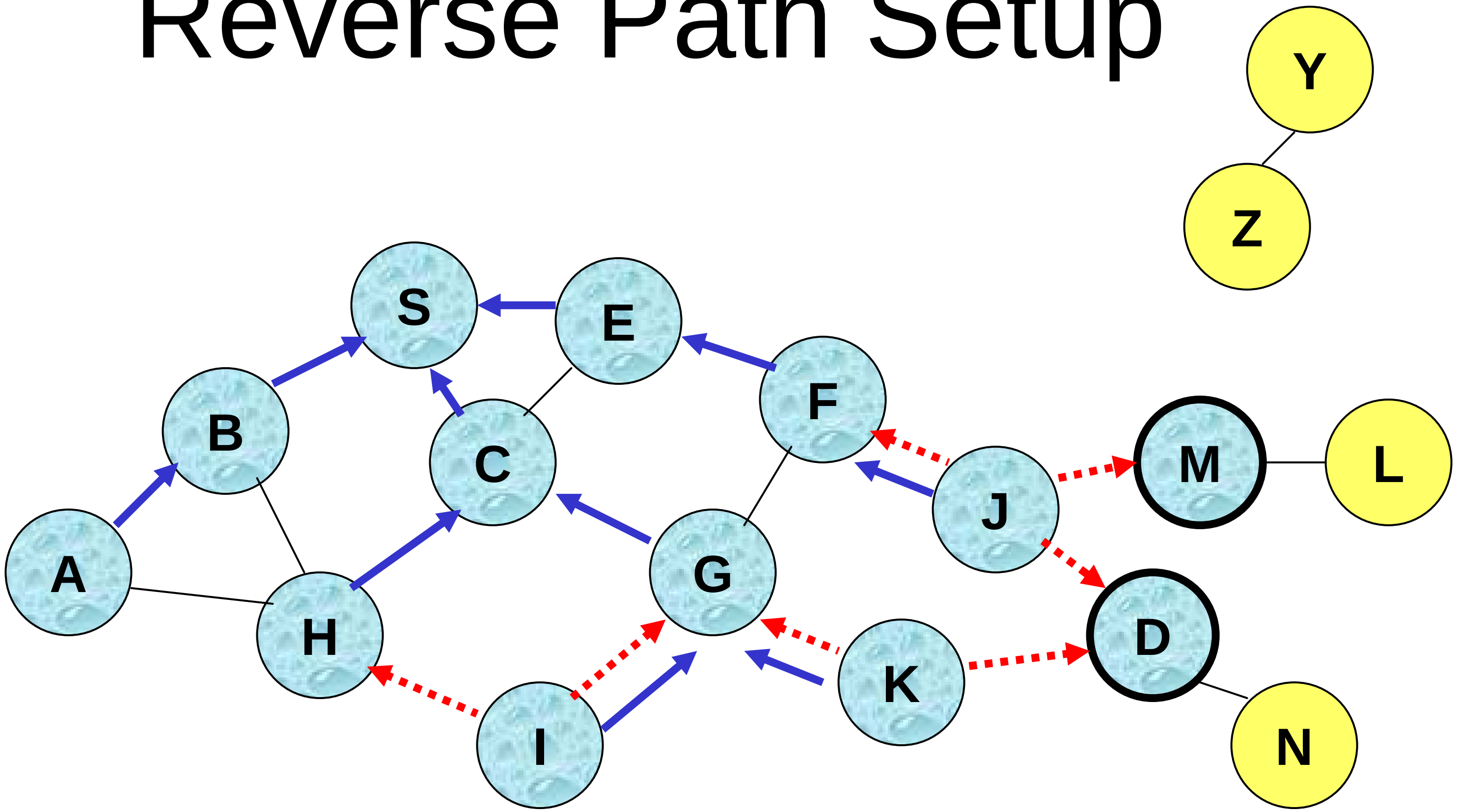
Represents links on Reverse Path

Reverse Path Setup

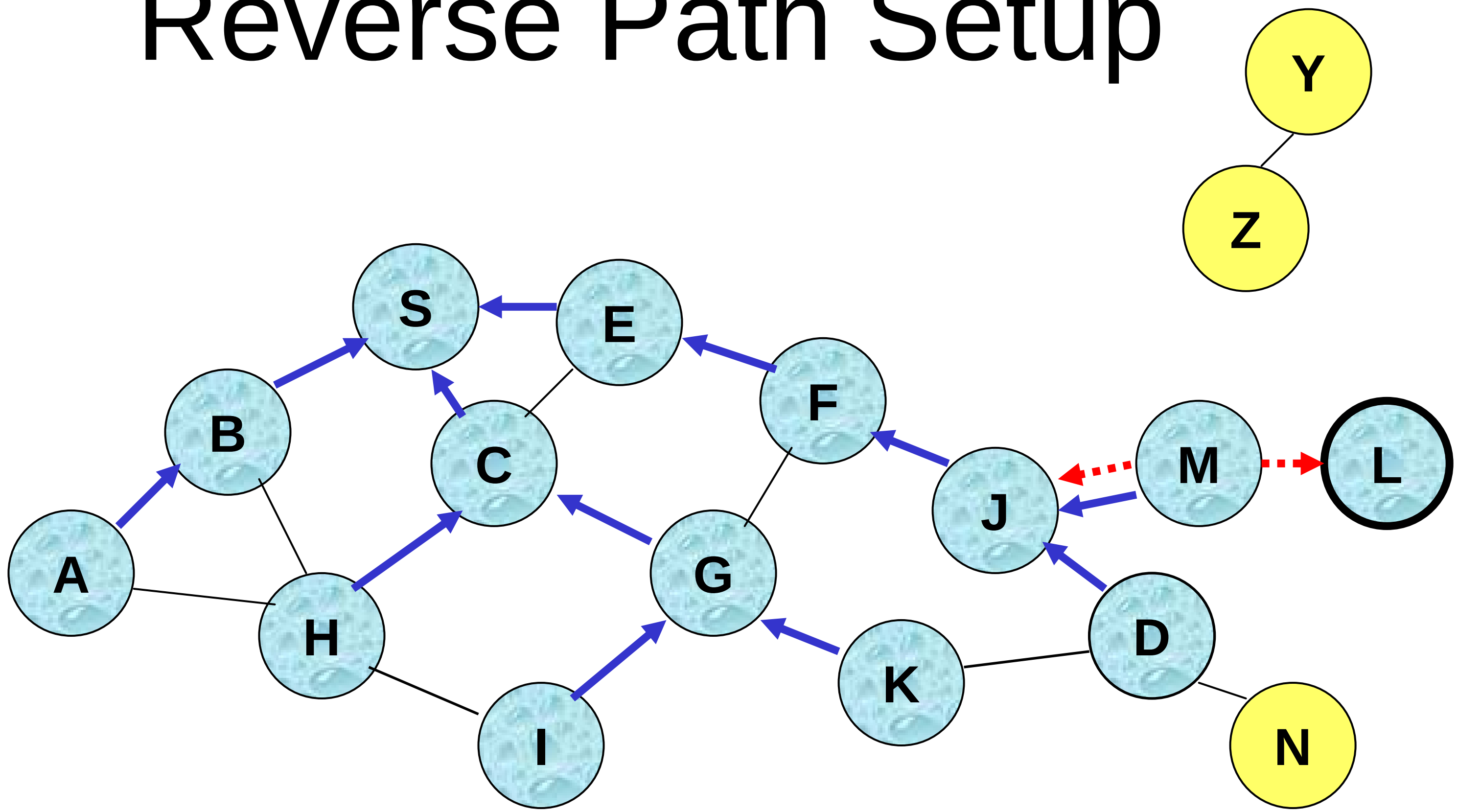


- **Node C receives RREQ from G and H, but does not forward it again, because node C has already forwarded RREQ once (i.e. same request ID).**

Reverse Path Setup

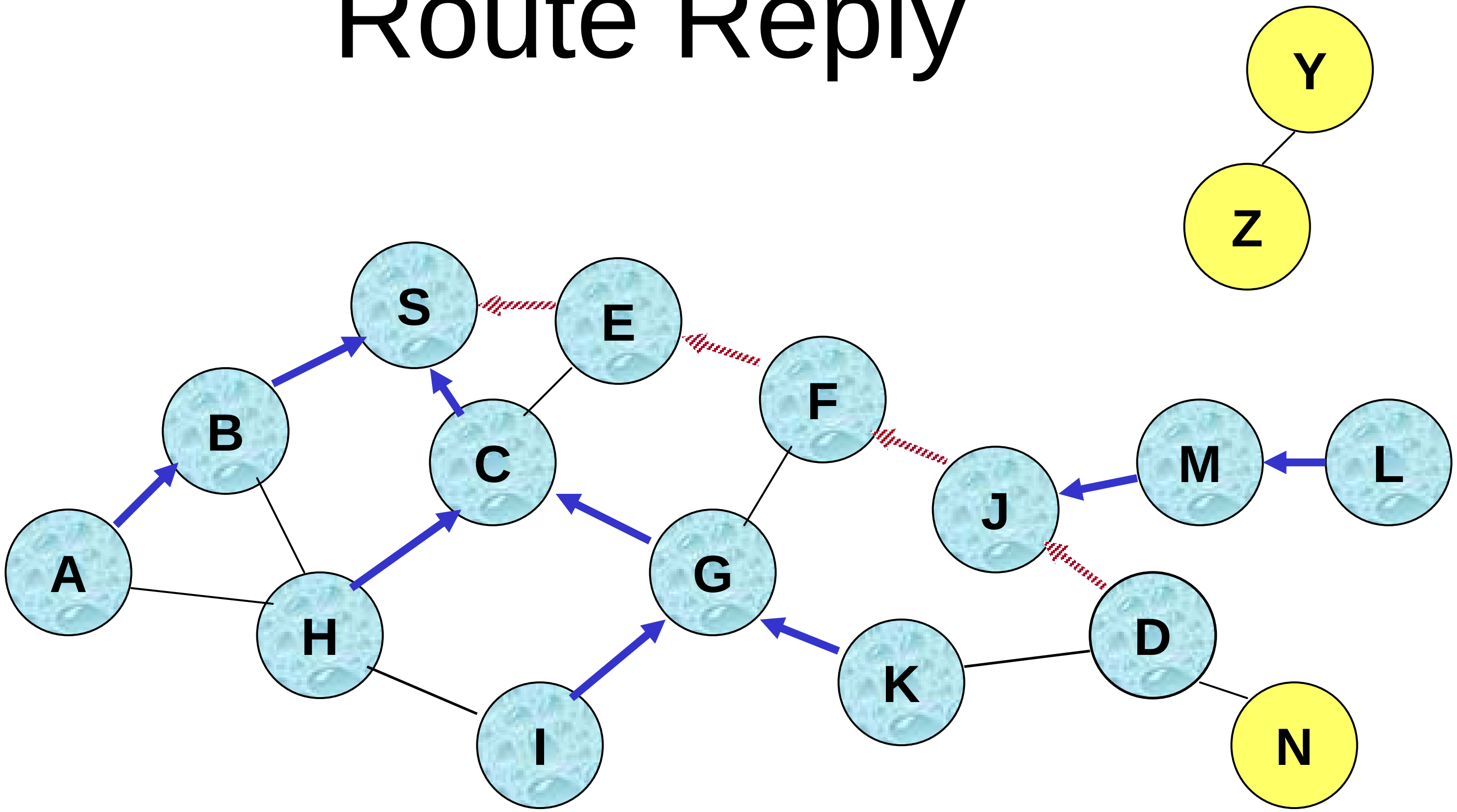


Reverse Path Setup



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

Route Reply

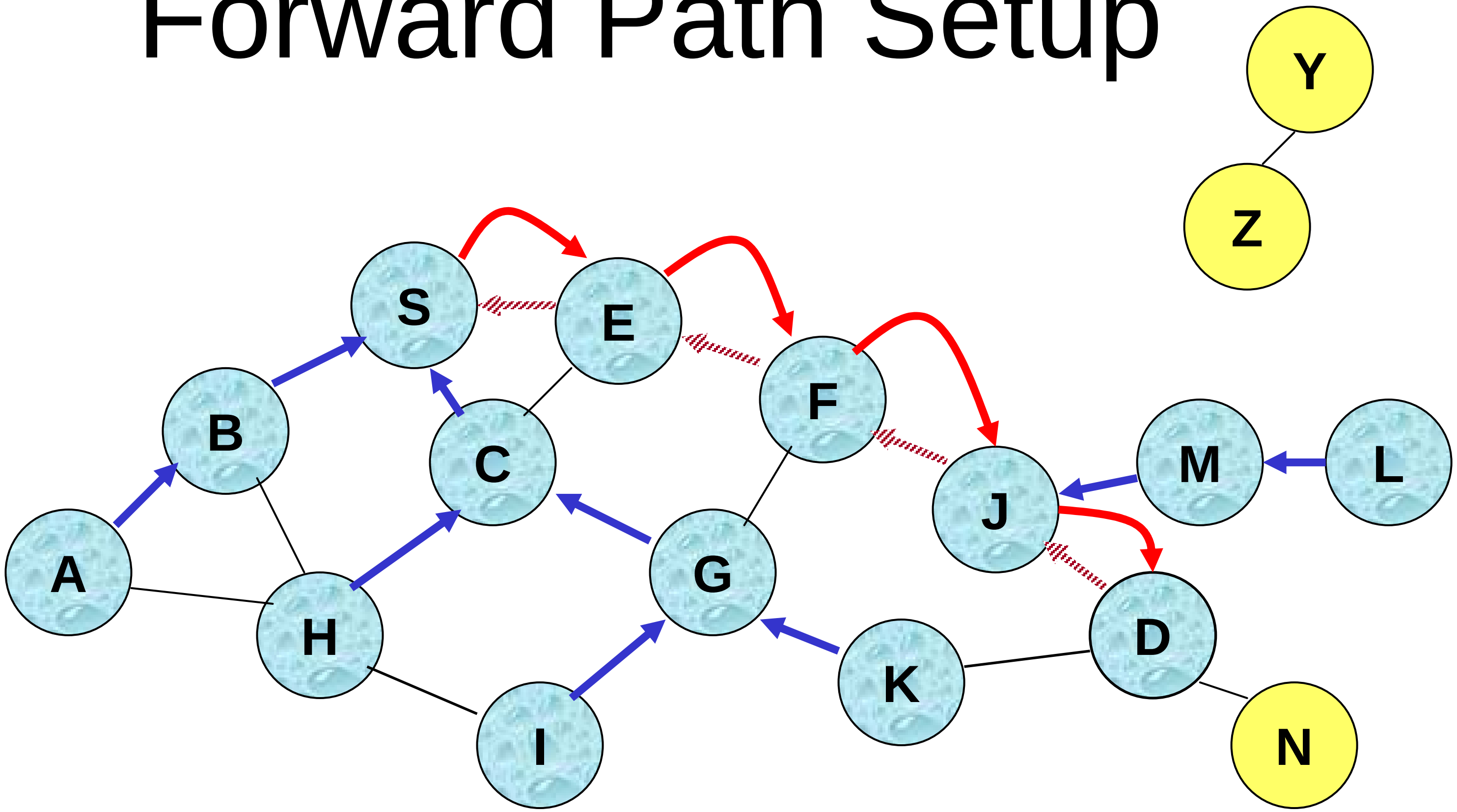


Represents links on path taken by RREP

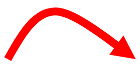
RREP by Intermediate Nodes

- An intermediate node (not the destination) may also send a RREP provided that it knows a more recent path than the one previously known to the source.
  - To determine whether the path is more recent, *destination sequence number* is used.
  - a path is more recent if the sequence no. in the routing table is larger than the destination sequence no. specified in RREQ.

Forward Path Setup



Forward links are setup when RREP travels along the reverse path



Represents a link on the forward path

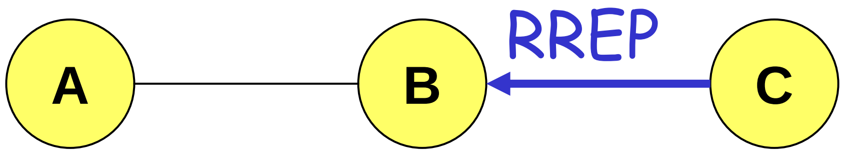
RREP by Destination

- If destination sends the RREP, the RREP message contains
  - the destination's current sequence no.
  - a hop count of zero
  - the length of time this route is valid

RREP by Intermediate Nodes

- If an intermediate node sends the RREP, the RREP message contains
  - its record of the destination sequence number
  - a hop count equal to its distance from the destination
  - the amount of time for which its route table entry for the destination will still be valid

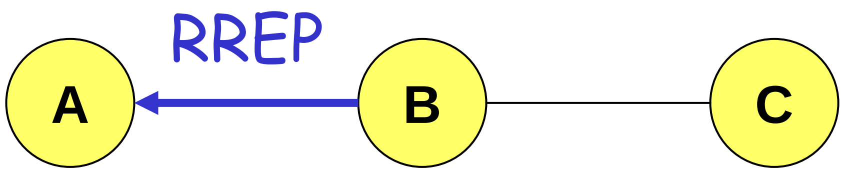
Example: Forward Route Entry at Node B



Destination ID	Sequence Number	Number of hops	Neighbor ID
C	6	1	C

destination sequence no.

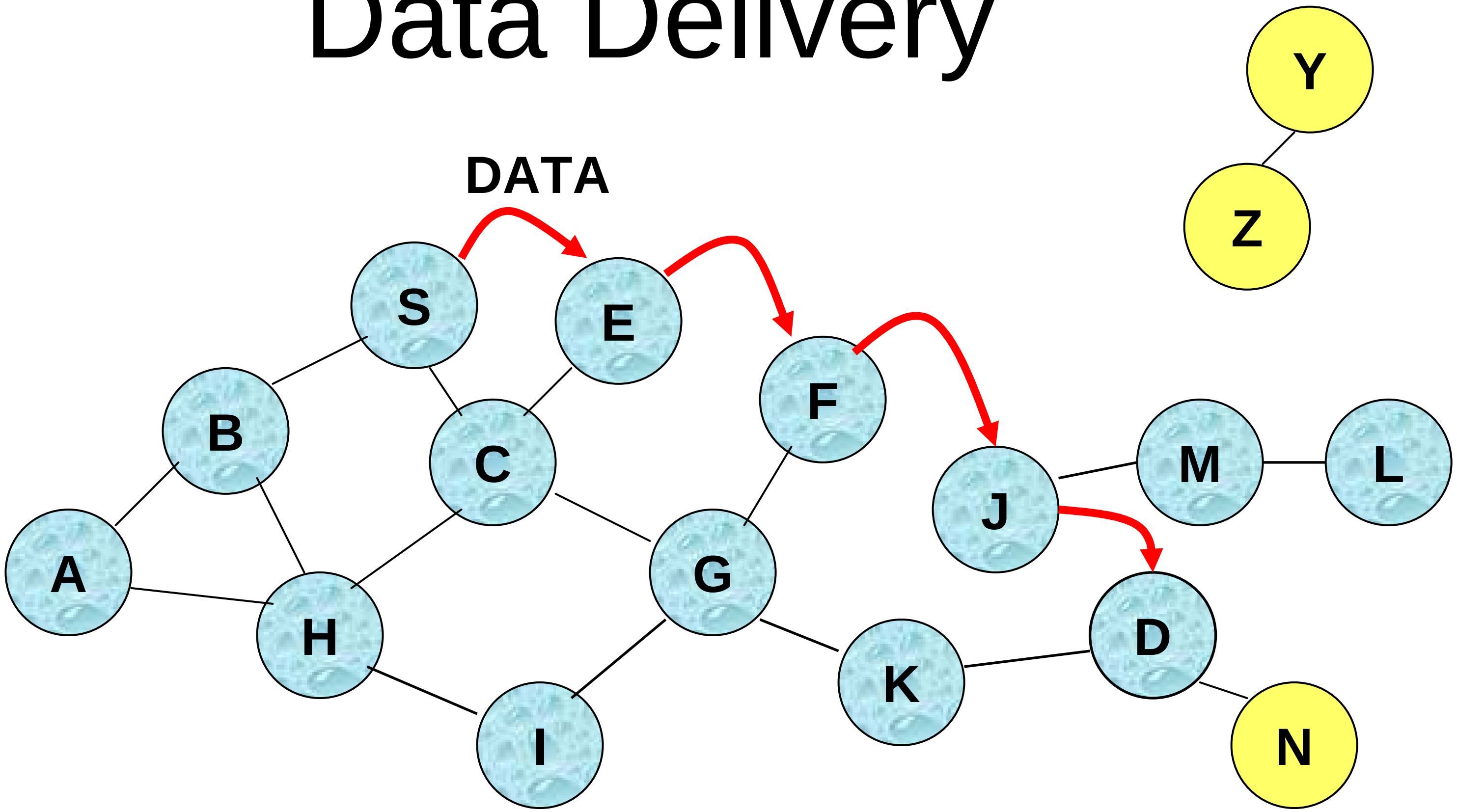
Example: Forward Route Entry at Node A



Destination ID	Sequence Number	Number of hops	Neighbor ID
C	6	2	B

destination
sequence no.

Data Delivery



Forward route entry in routing tables are used to forward data packet. Route is *not* included in packet header.

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 an *active_route_timeout* interval
  - if no 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)

Route Maintenance

- When X is unable to forward packet P (from S to D) on link (X,Y), it sends a RERR message to S

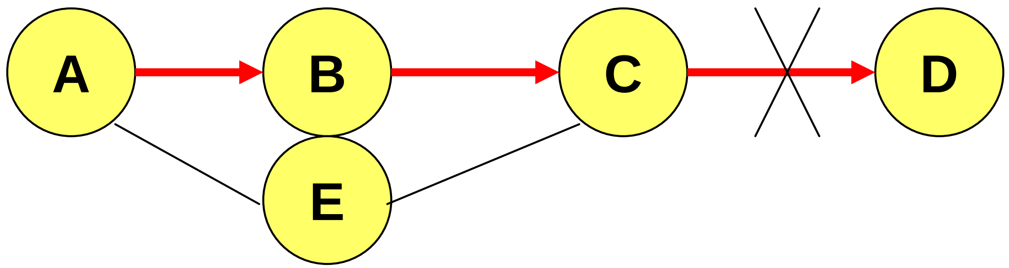
- When S receives the REER, it can reinitiate route discovery if the route is still needed.

Link Failure Detection

- *Hello* messages: Neighboring nodes periodically exchange hello messages
- 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 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. Node A receives the RREQ (say, via path C-E-A)
  - A will reply since A knows a route to D via node B
  - Results in a loop (for instance, C-E-A-B-C)

Expanding Ring Search

- RREQs are initially sent with small Time-to-Live (TTL) field, to limit their propagation
  - DSR also includes a similar optimization

- If no RREP is received, then increase TTL and try again

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

References

- C. E. Perkins, *Ad hoc networking*, Addison Wesley, 2001.
- D. B. Johnson and D. A. Maltz, “Dynamic source routing in ad hoc wireless networks,” *Mobile Computing*, Kluwer Academic Publishers, edited by T. Imielinski and H. F. Korth, pp. 153-181, 1996.
- C. Perkins and E. Royer, “Ad hoc on-demand distance vector routing,” *Proc. 2nd IEEE Workshop on Mobile Computing Systems and Applications*, pp. 90-100, Feb. 1999.
- Presentation slides of Prof. Nitin H. Vaidya on mobile ad hoc networks.