Dynamic Source Routing (DSR) Protocol

CS: 647
Advanced Topics in Wireless Networks

Drs. Baruch Awerbuch & Amitabh Mishra
Department of Computer Science
Johns Hopkins

Reading

- Chapter 5 – Ad Hoc Networking, Perkins, Addison Wesley, 2001

Outline

- Introduction
- Route Discovery
- Route Cache
- Route Maintenance
  - Preventing Route Reply Storms
  - Route Request hop limits
  - Packet Salvaging
  - Automatic Route Shortening
  - Increased spreading of Route Error messages
- Summary

Dynamic Source Routing (DSR) - Introduction

- Reactive or On Demand
- Developed at CMU in 1996
- Route discovery cycle used for route finding – on Demand
- Maintenance of active routes
- No periodic activity of any kind – Hello Messages in AODV
- Utilizes source routing (entire route is part of the header)
- Use of caches to store routes
- Supports unidirectional links -> Asymmetric routes are supported

Dynamic Source Routing (DSR)

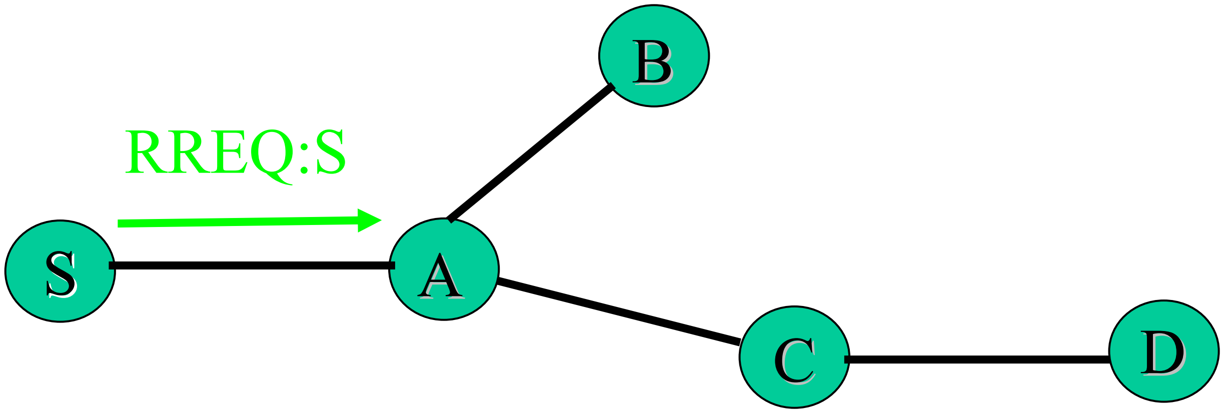
- When node S wants to send a packet to node D, but does not know a route to D, node S initiates a route discovery
- Source node S floods Route Request (RREQ)
- Each RREQ, has sender's address, destination's address, and a unique Request ID determined by the sender
- Each node appends own identifier when forwarding RREQ

Outline

- Introduction
- Route Discovery
- Route Cache
- Route Maintenance
  - Preventing Route Reply Storms
  - Route Request hop limits
  - Packet Salvaging
  - Automatic Route Shortening
  - Increased spreading of Route Error messages
- Summary

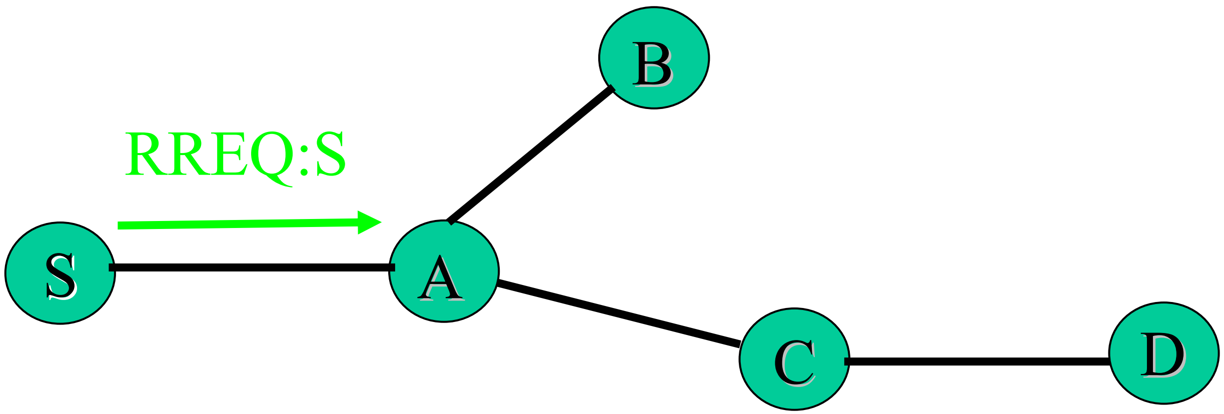
Route Discovery in DSR - Example 1

DSR - Route Discovery



1. Node S needs a route to D
2. Broadcasts RREQ packet

DSR - Route Discovery



1. Node S needs a route to D
2. Broadcasts RREQ packet
3. Node A receives packet, has no route to D
   - Rebroadcasts packet after adding its address to source route

Route Discovery - Node Actions

- Upon receiving a RREQ, the node takes the following actions:

1. The node is the Target (Destination)
   - Returns a Route Reply (RREP) message to the sender
   - Copies the accumulated route record from RREQ into RREP
   - Sender upon receiving RREP, caches the route in its route cache for subsequent routing

Route Discovery - Node Actions

2. The node is the intermediate node

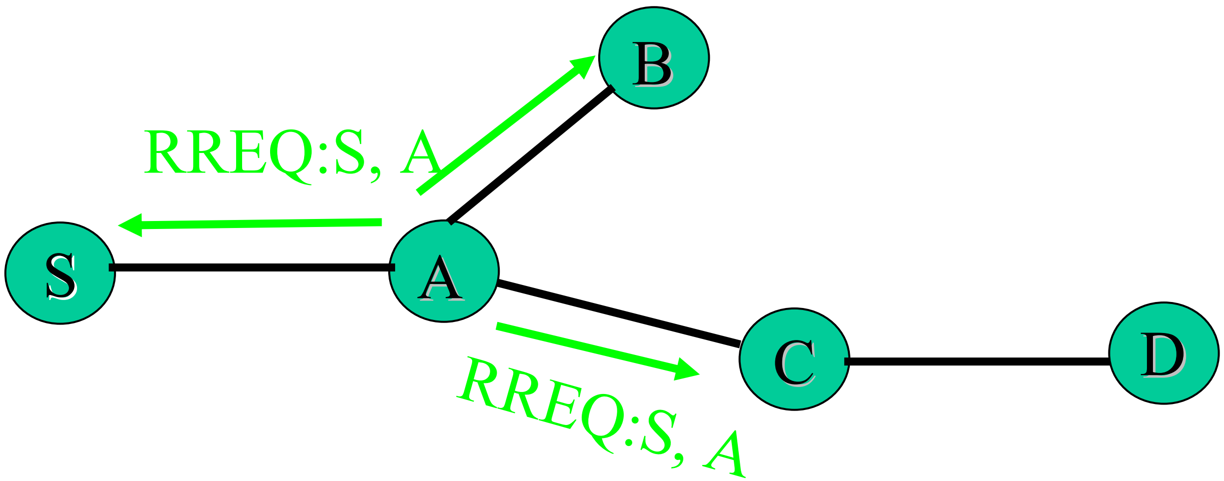
- The node discards this message, if
  1. The message has the same ID i.e. has seen it before

  OR

  2. Finds its own address in the route record

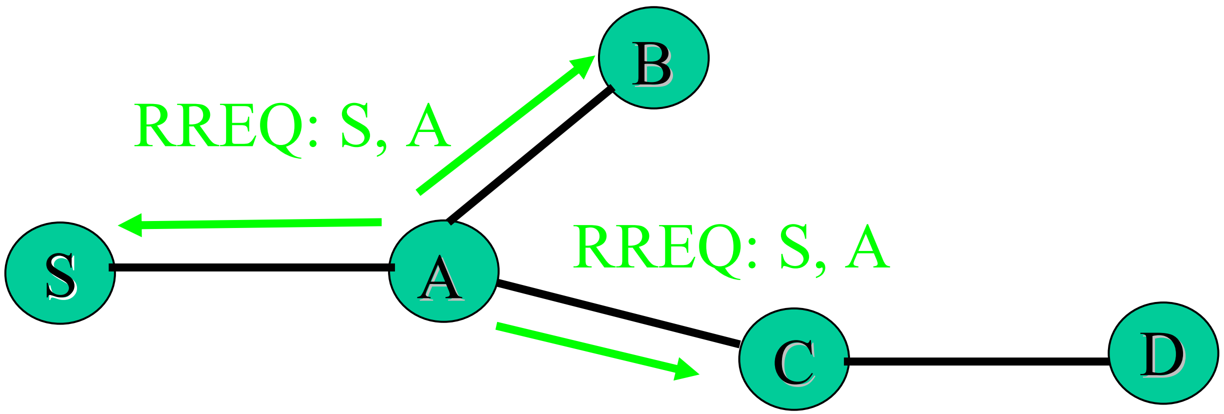
- If Not, The node appends its own address to the route record in the ROUTE REQUEST message
  1. Propagates the message to the next hop neighbors

DSR - Route Discovery



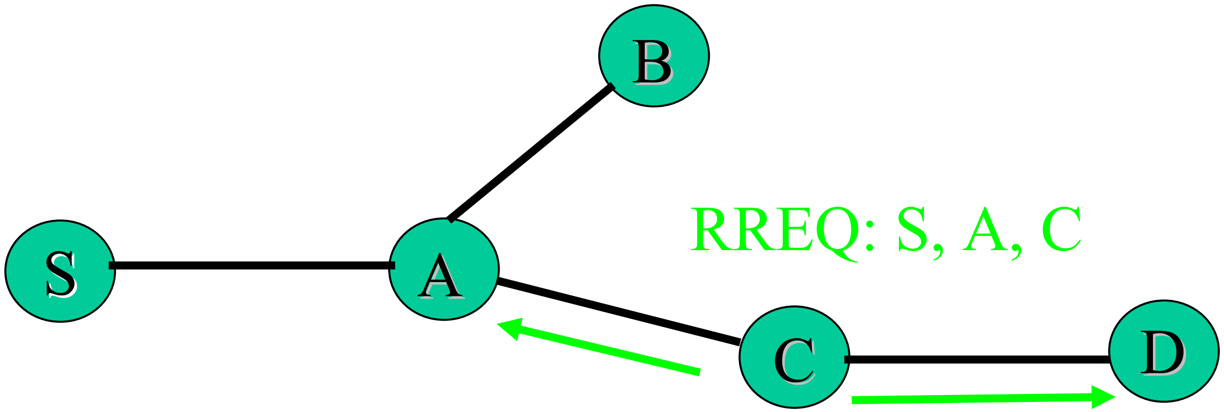
1. Node S needs a route to D
2. Broadcasts RREQ packet
3. Node A receives packet, has no route to D
   - Rebroadcasts packet after adding its address to source route

DSR - Route Discovery



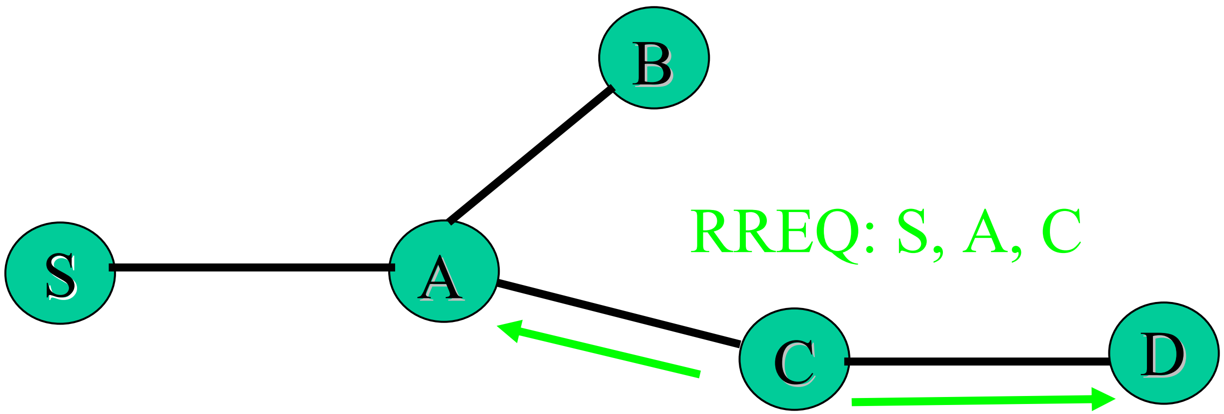
4. Node C receives RREQ, has no route to D
   - Rebroadcasts packet after adding its address to source route

DSR - Route Discovery



4. Node C receives RREQ, has no route to D
   - Rebroadcasts packet after adding its address to source route

DSR - Route Discovery



4. Node C receives RREQ, has no route to D
   - Rebroadcasts packet after adding its address to source route
5. Node D receives RREQ, unicasts RREP to C
   - Puts D in RREP source route

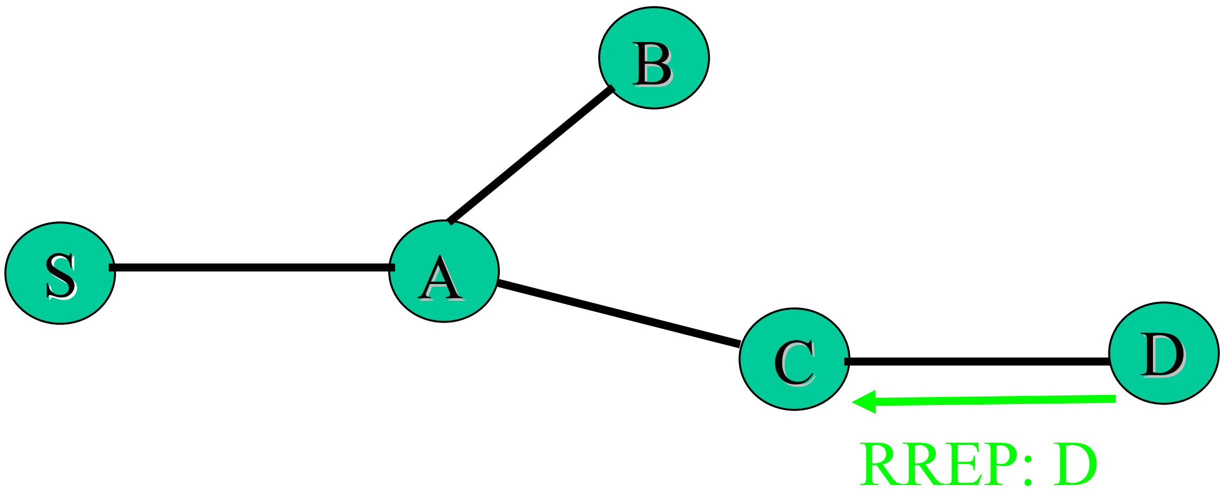
Route Reply in DSR

- Route Reply can be sent by reversing the route in Route Request (RREQ) only if links are guaranteed to be bi-directional
  - One way to ensure this is to check, if the received RREQ was on a link that is known to be bi-directional, e.g.

- If IEEE 802.11 MAC is used to send data, then links have to be bi-directional (since Ack is used)

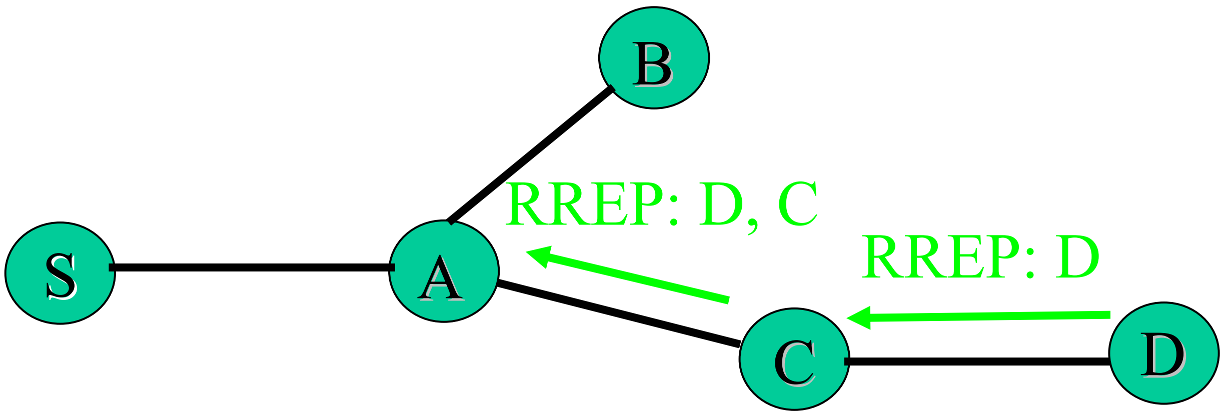
- If unidirectional (asymmetric) links are allowed, then RREP may need a route discovery for S from node D
  - Route discovery not needed -> If node D already knows a route to node S
  - If a route discovery is initiated by D for a route to S, then the Route Reply is piggybacked on the Route Request from D.

DSR - Route Reply



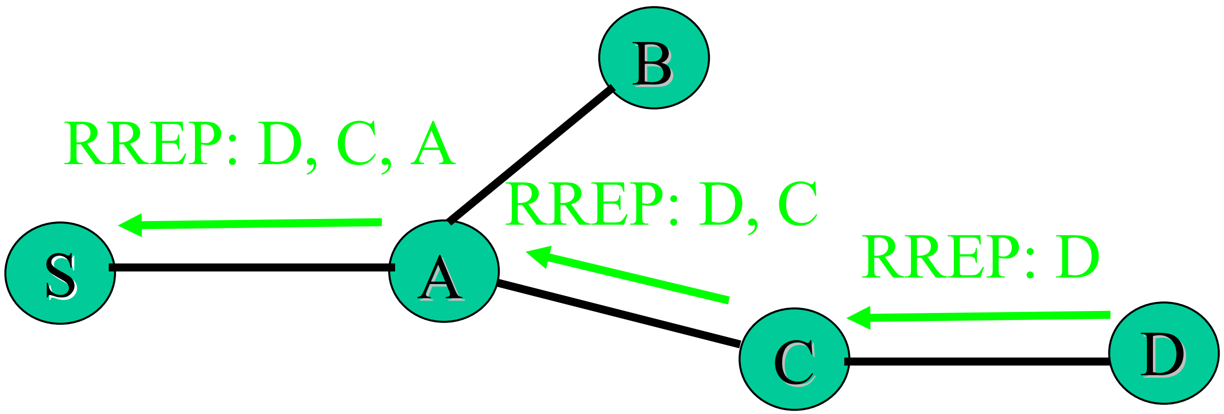
4. Node C receives RREQ, has no route to D
   - Rebroadcasts packet after adding its address to source route
5. Node D receives RREQ, unicasts RREP to C
   - Puts D in RREP source route

DSR - Route Reply



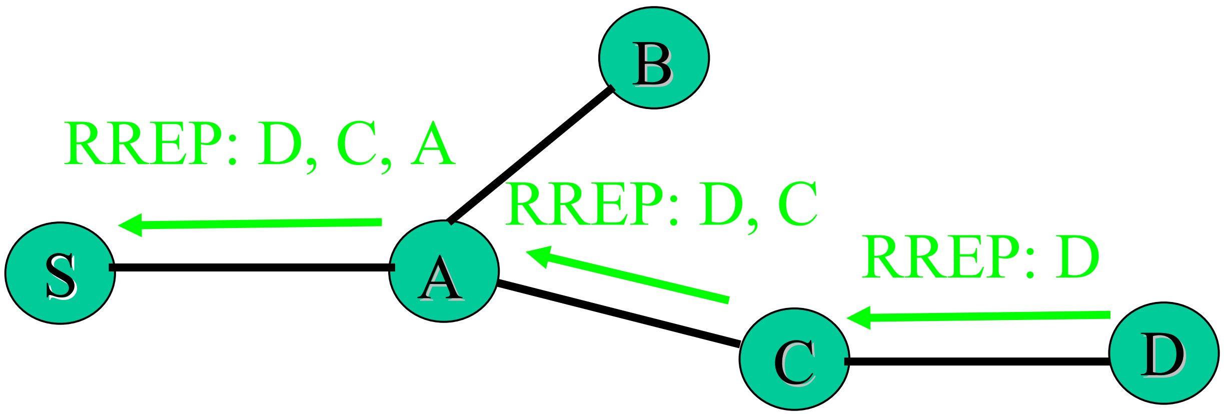
6. Node C receives RREP
   - Adds its address to source route
   - Unicasts to A

DSR - Route Reply



7. Node A receives RREP
   - Adds its address to source route
   - Unicasts to S

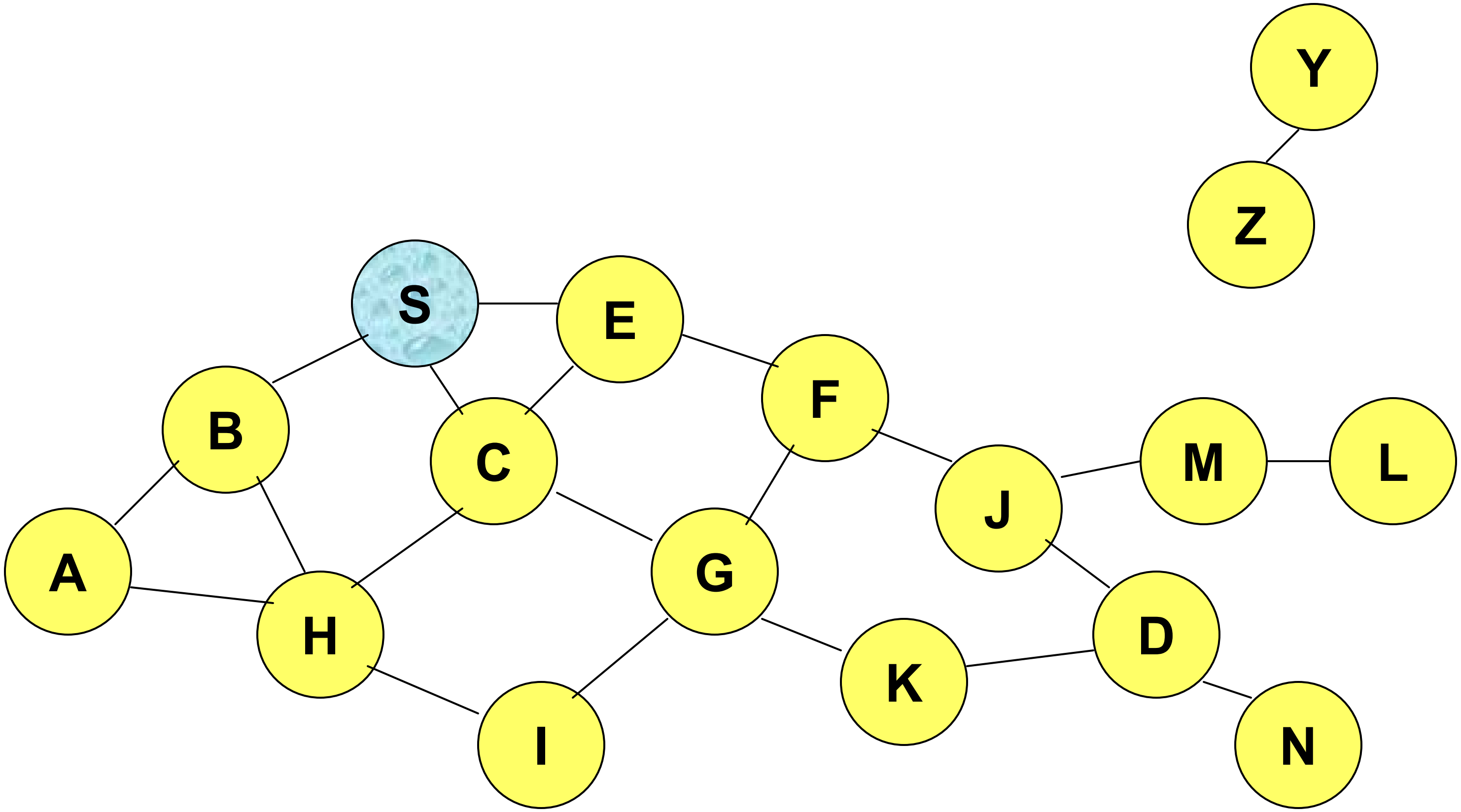
DSR - Route Reply

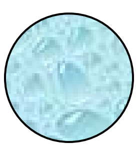


7. Node S receives RREP
   - Uses route for data packet transmissions

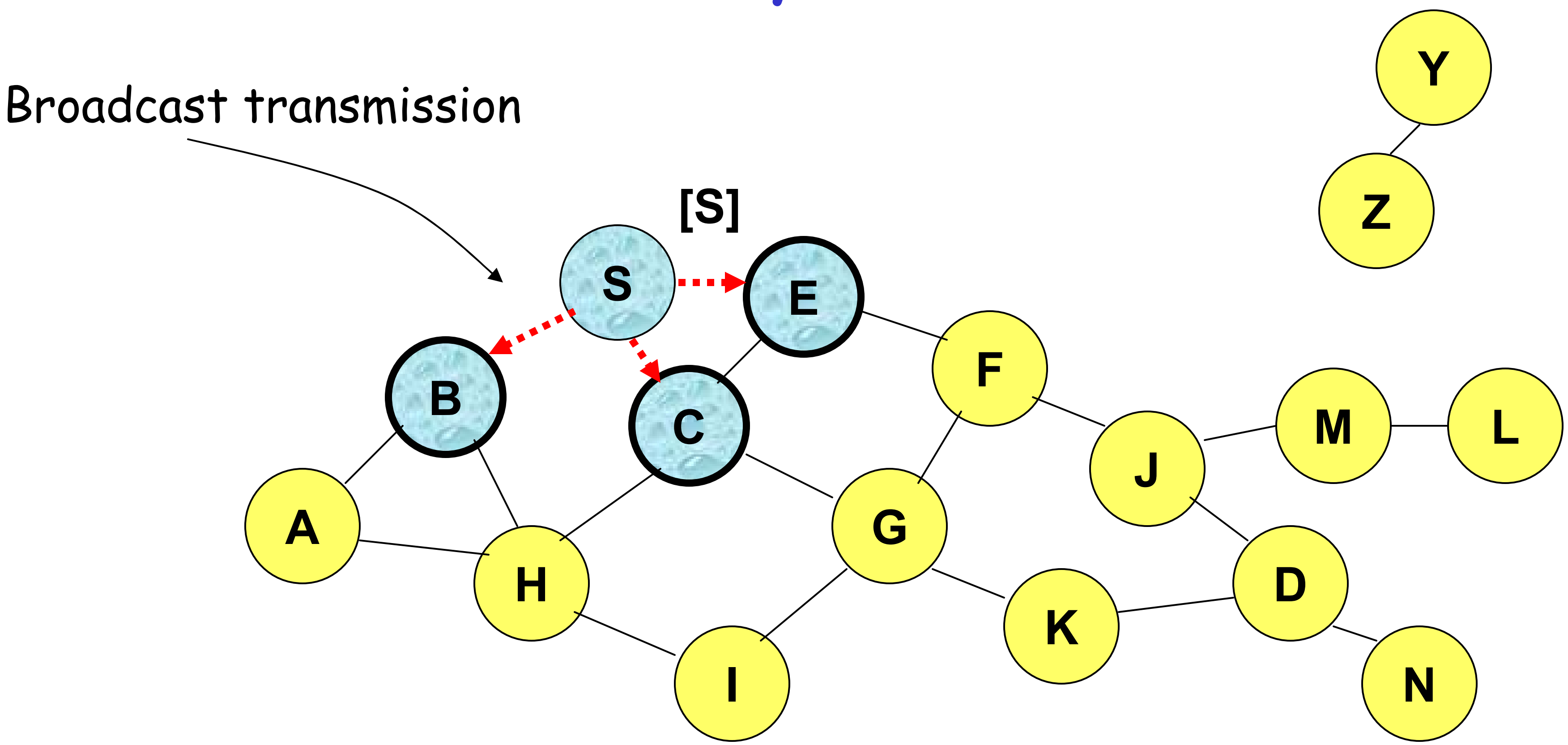
Route Discovery in DSR - Example 2

Route Discovery in DSR



 Represents a node that has received RREQ for D from S

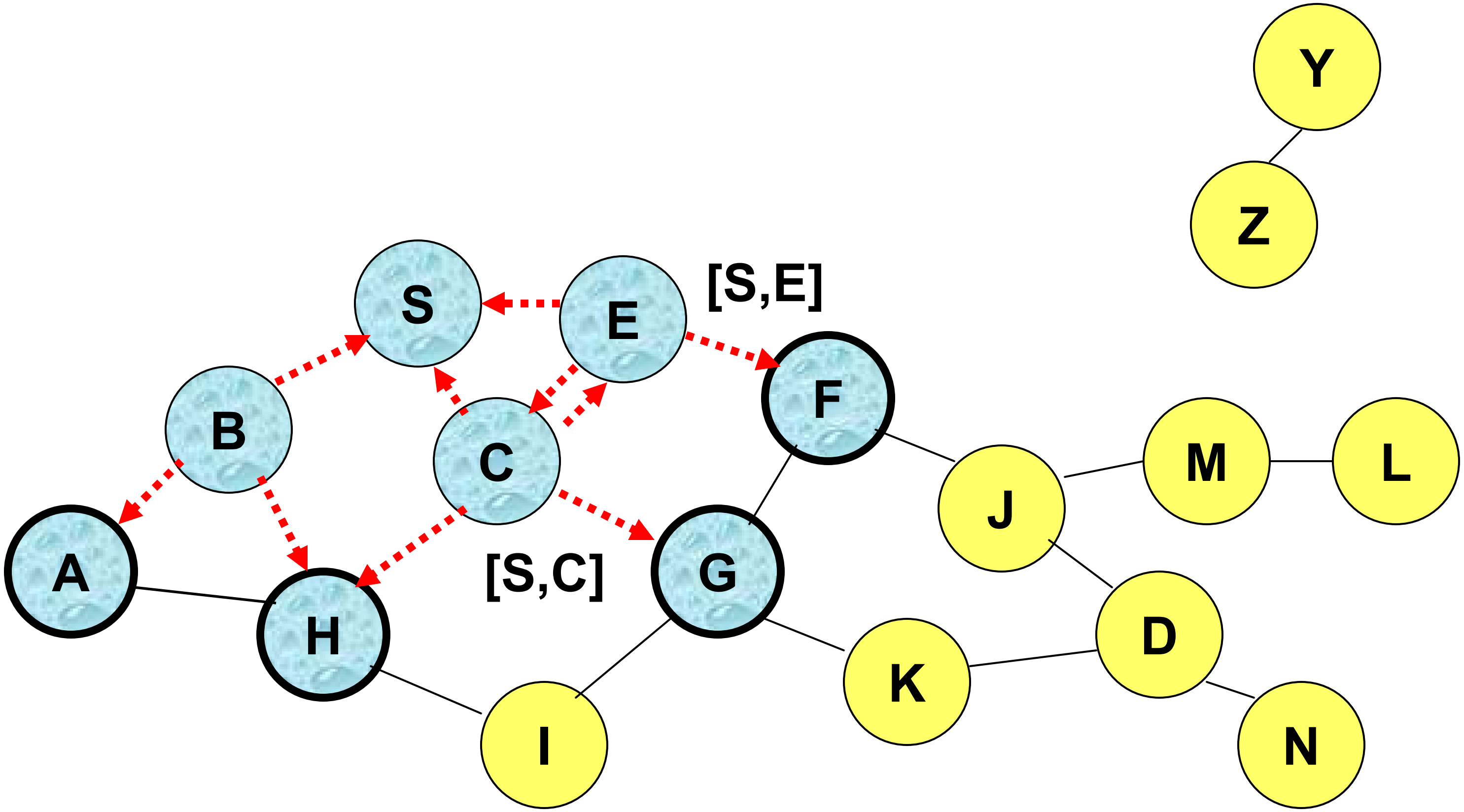
Route Discovery in DSR



Represents transmission of RREQ

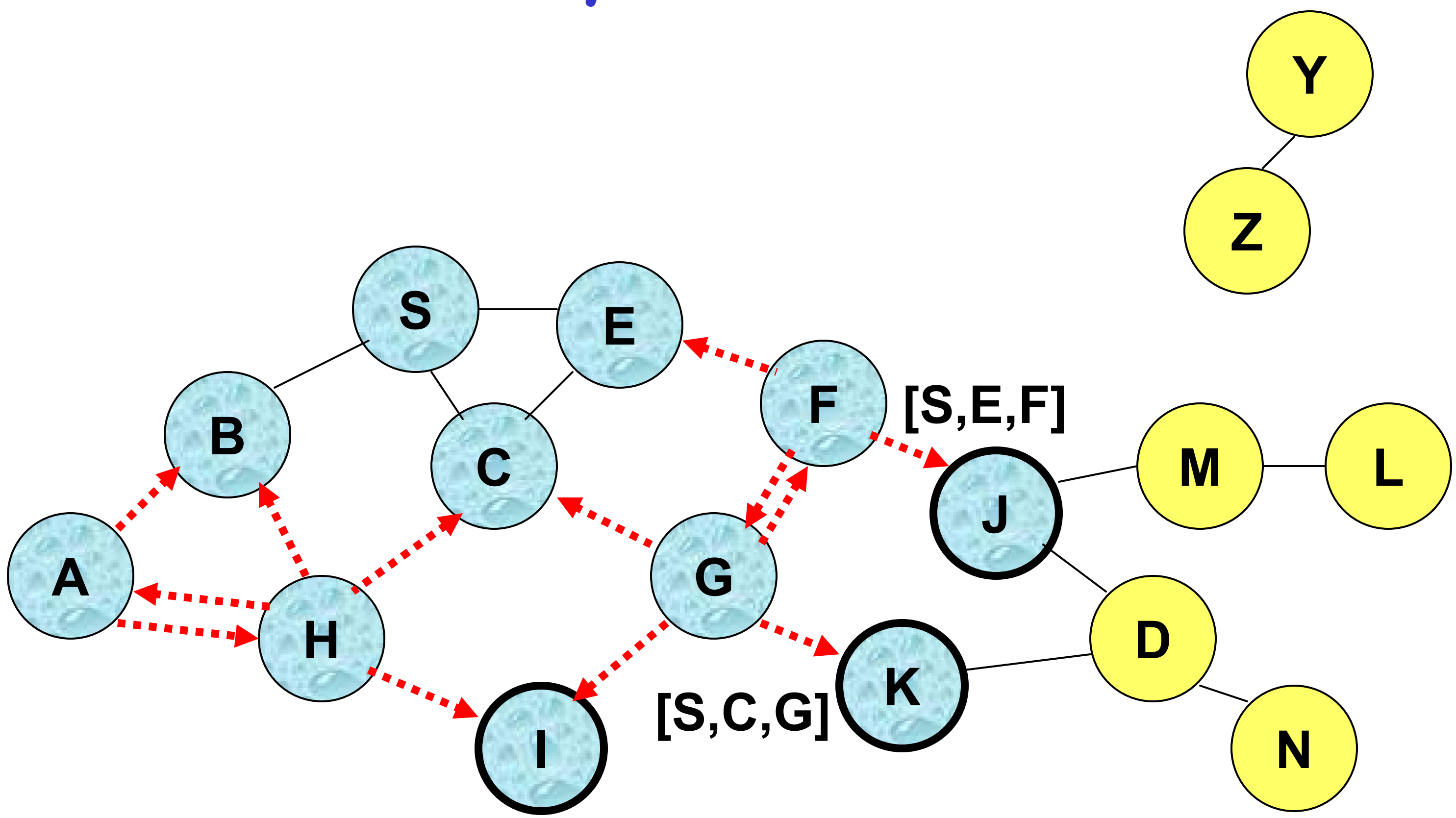
[X,Y] Represents list of identifiers appended to RREQ

Route Discovery in DSR



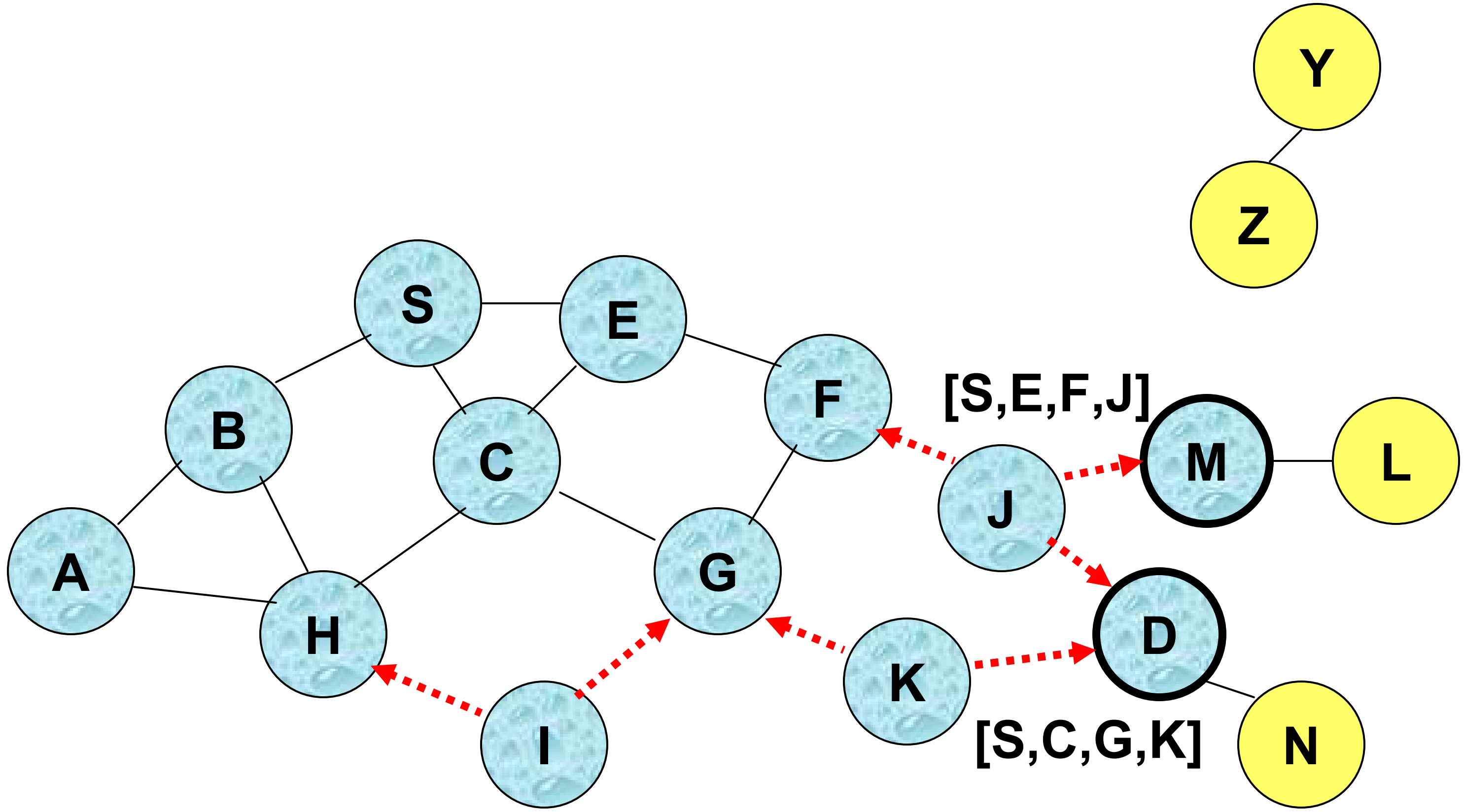
- Node H receives packet RREQ from two neighbors: potential for collision

Route Discovery in DSR



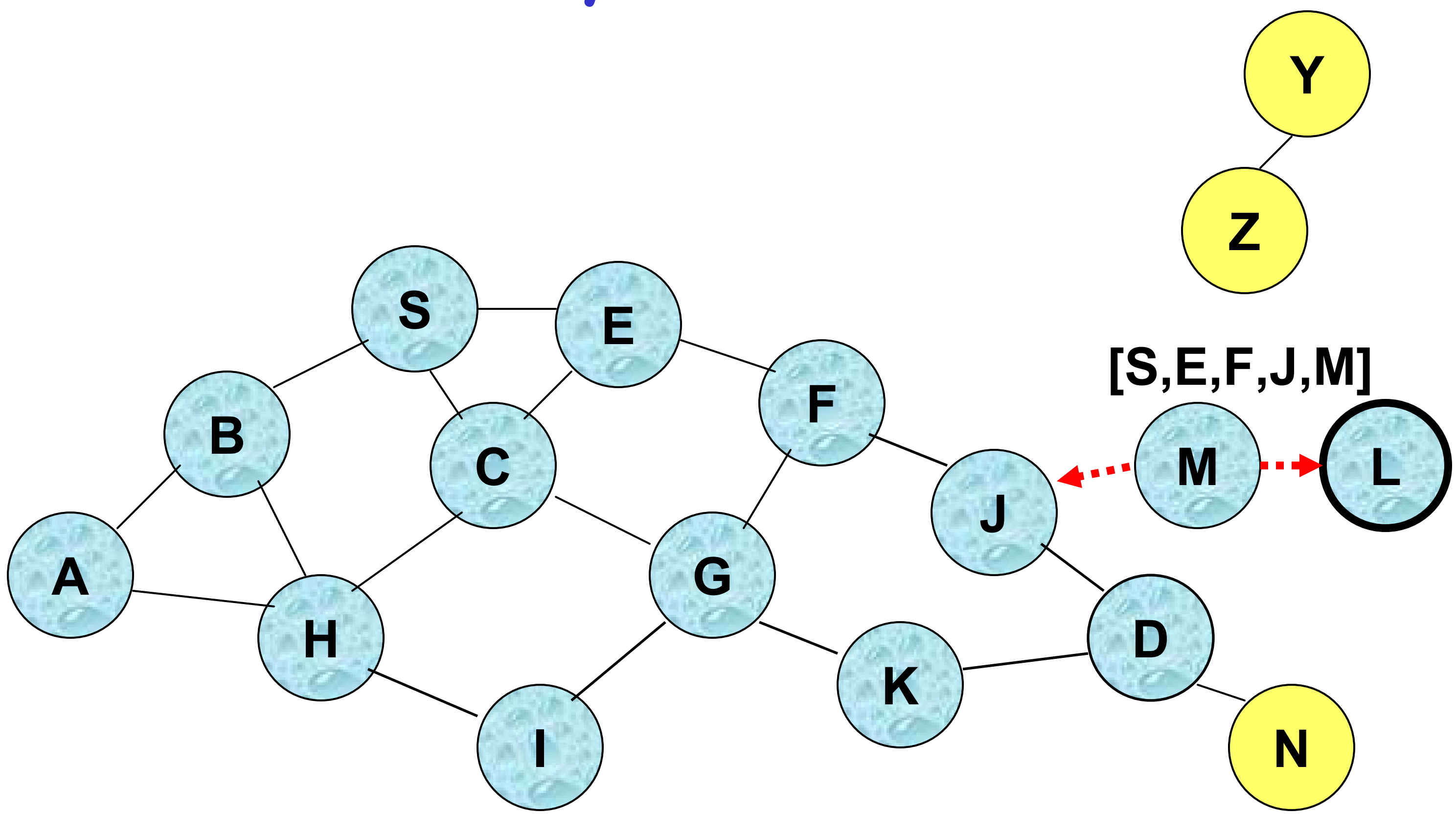
- Node C receives RREQ from G and H, but does not forward it again, because node C has already forwarded RREQ once

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

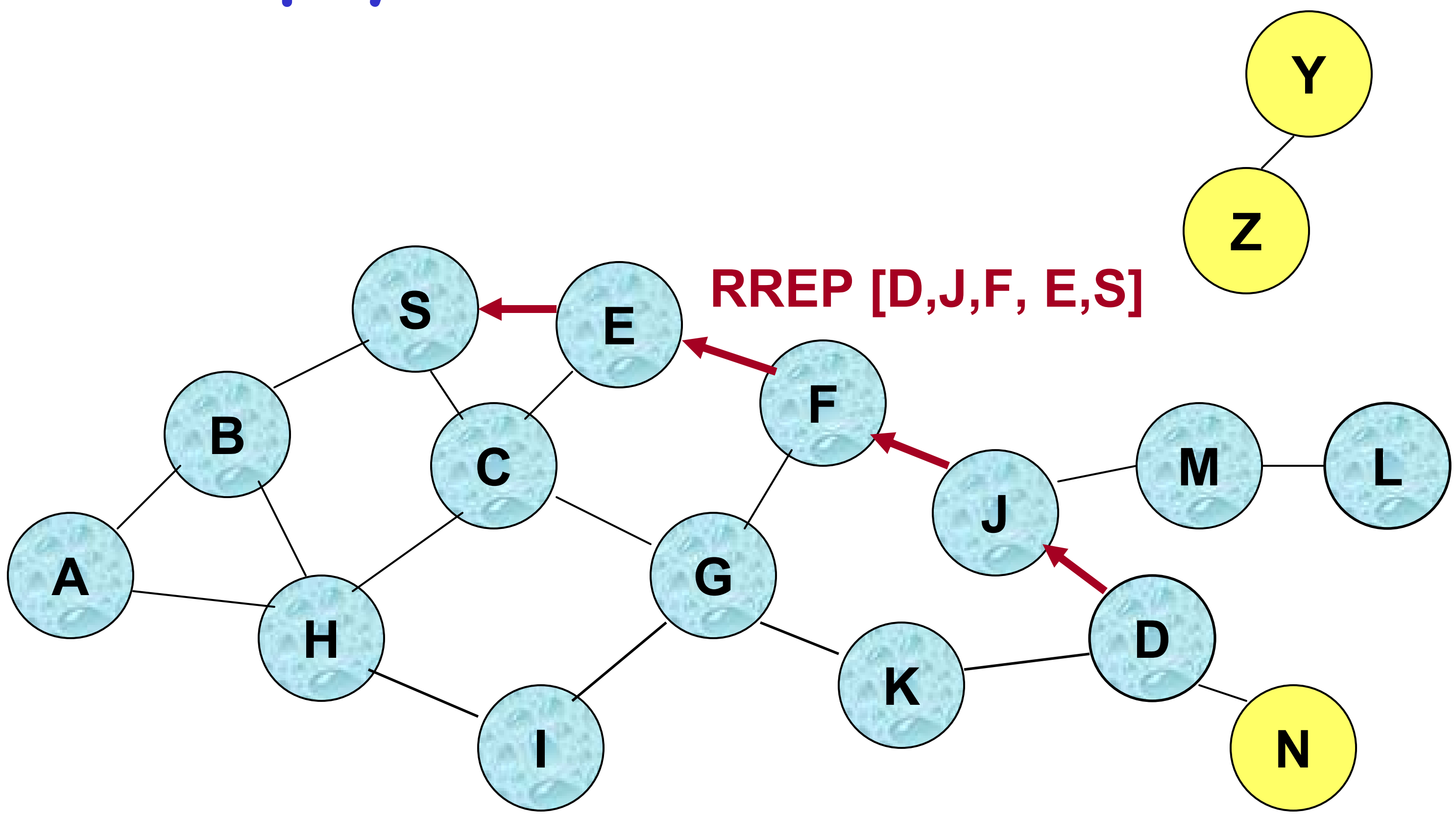
Route Discovery - Send Buffer

- During Route Discovery, the sending node saves a copy of the message in the send buffer
- Send buffer has a copy of every packet that cannot be transmitted by this node due to lack of a route
- Each packet is time stamped and discarded after a specified time out period, if it cannot be forwarded
- For packets waiting in the send buffer, the node should occasionally initiate a new route discovery

Route Discovery - Send Buffer

- New Route Discovery rate for the same destination node should be limited if the node is currently unreachable
  - Results in wastage of wireless bandwidth due to a large number of RREQs destined for the same destination -> High overhead
  - To reduce the overhead, the node goes into exponential back-off for the new route discovery of the same target
  - Packets are buffered that are received during the back-off

Route Reply in DSR



Represents RREP control message

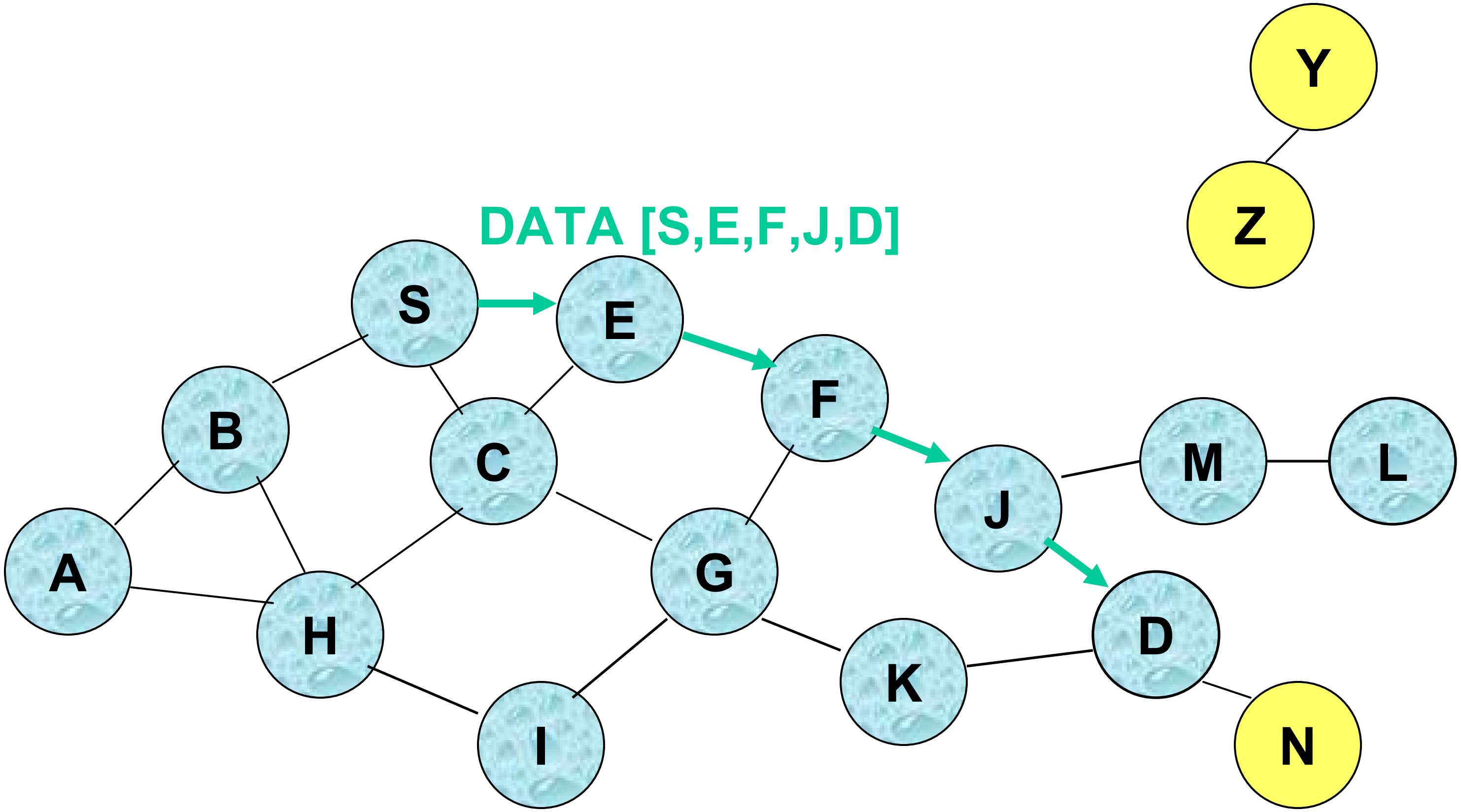
Dynamic Source Routing (DSR)

- 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

Outline

- Introduction
- Route Discovery
- Route Cache
- Route Maintenance
  - Preventing Route Reply Storms
  - Route Request hop limits
  - Packet Salvaging
  - Automatic Route Shortening
  - Increased spreading of Route Error messages

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 RREP [D,J,F, E,S], 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

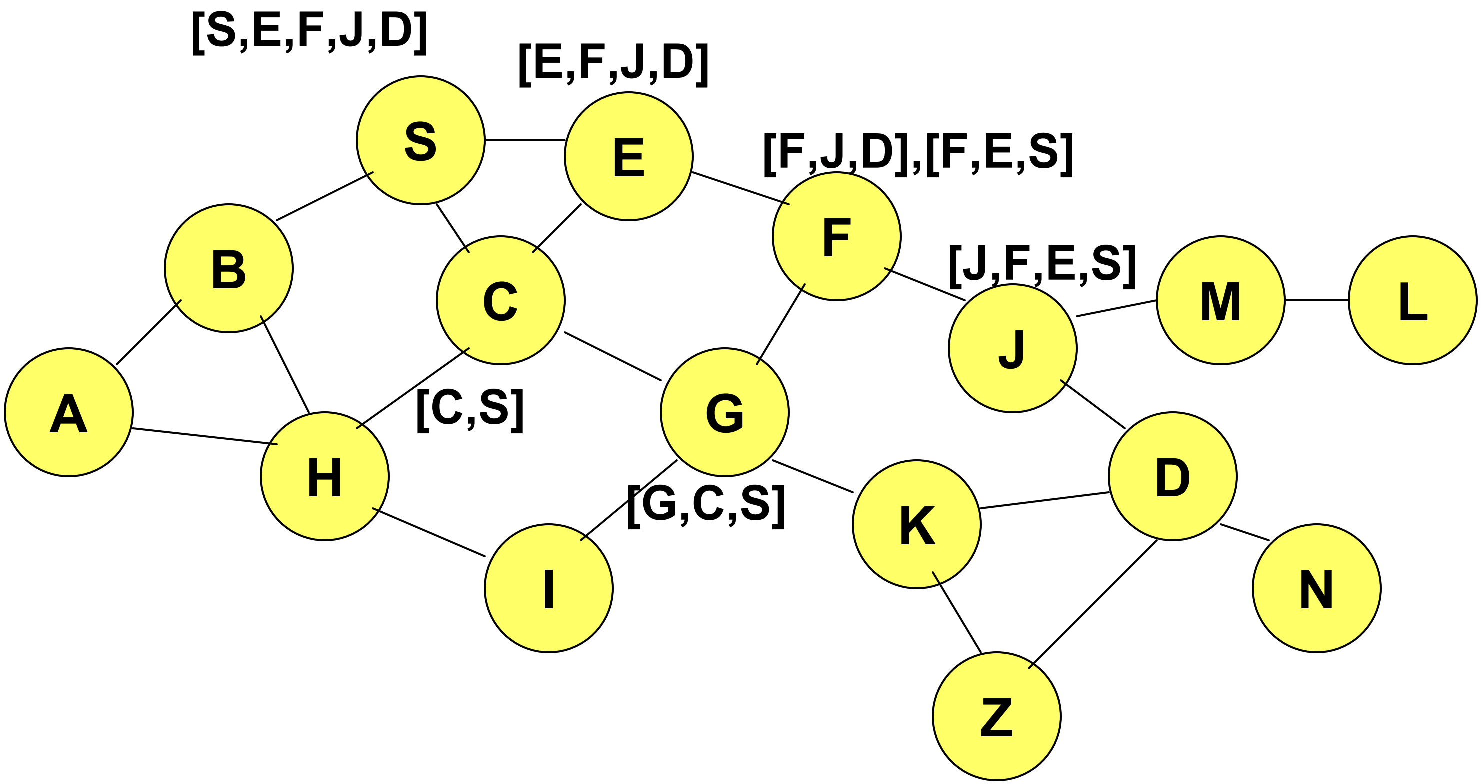
Use of Route Caching

- When node S learns that a route to node D is broken,
  - Can use 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

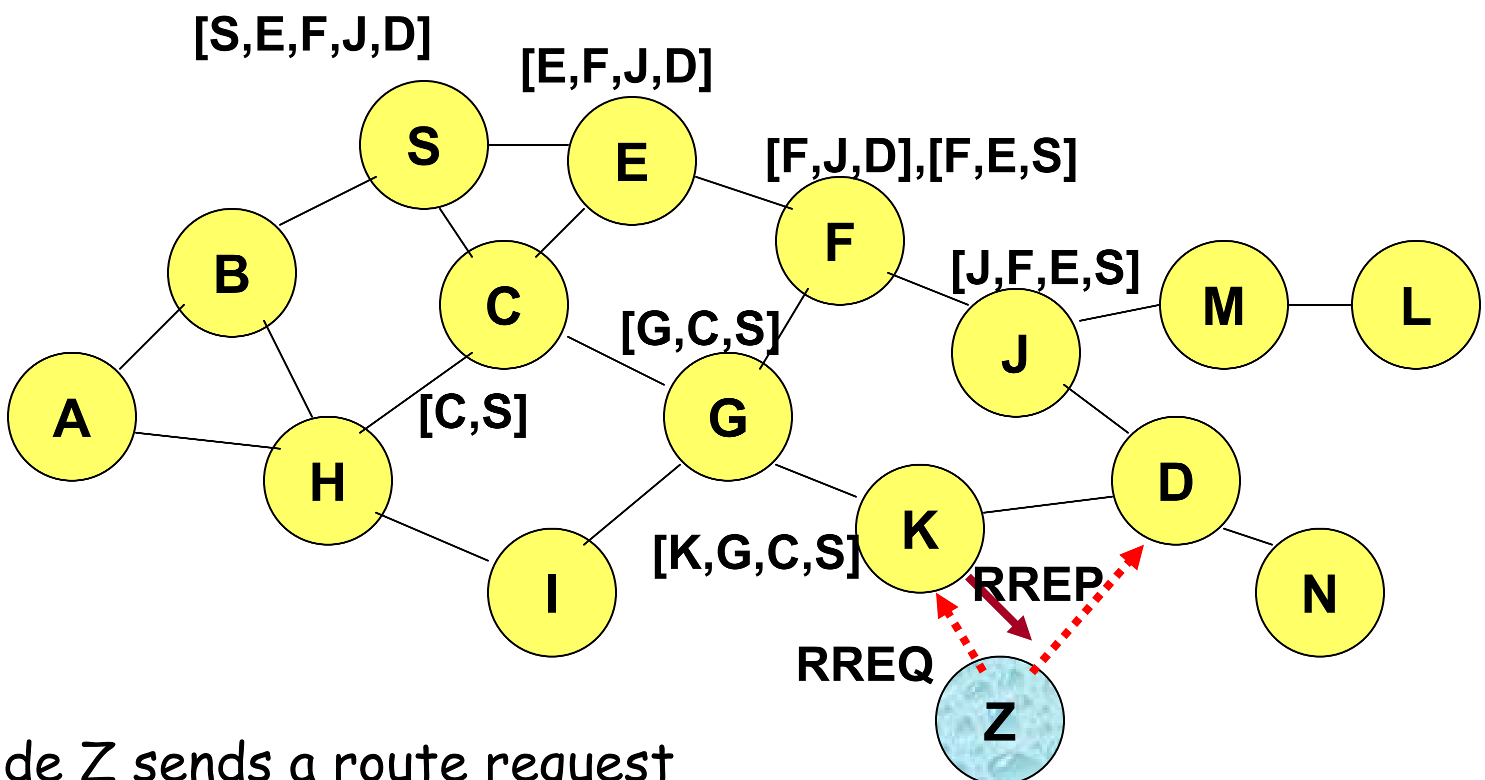
- Use of route cache
  - can speed up route discovery
  - can reduce propagation of route requests

Use of Route Caching - Example



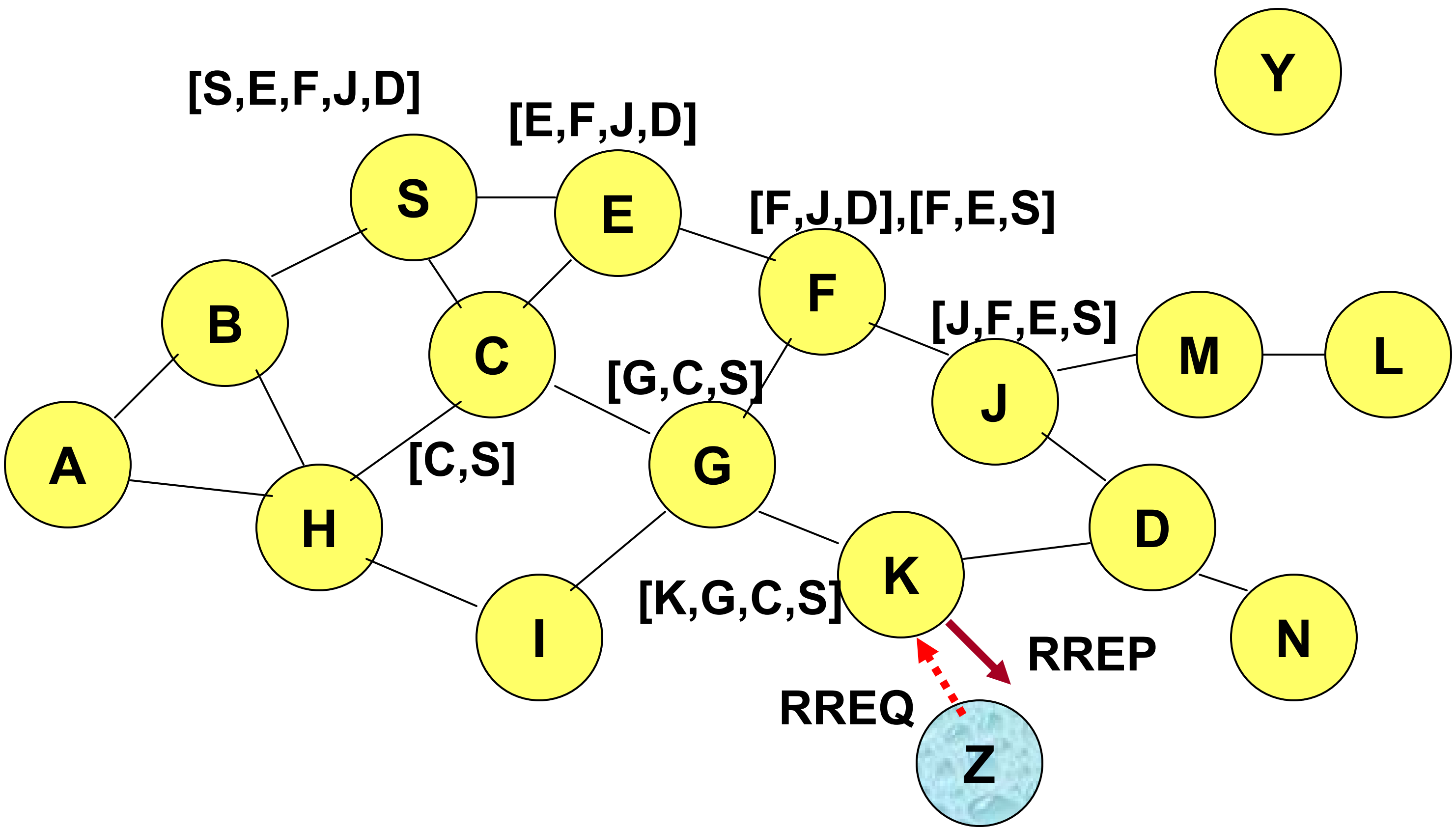
[P,Q,R] Represents cached route at a node
(DSR maintains the cached routes in a tree format)

Use of Route Caching: Benefits - (1) Speed up of Route Discovery



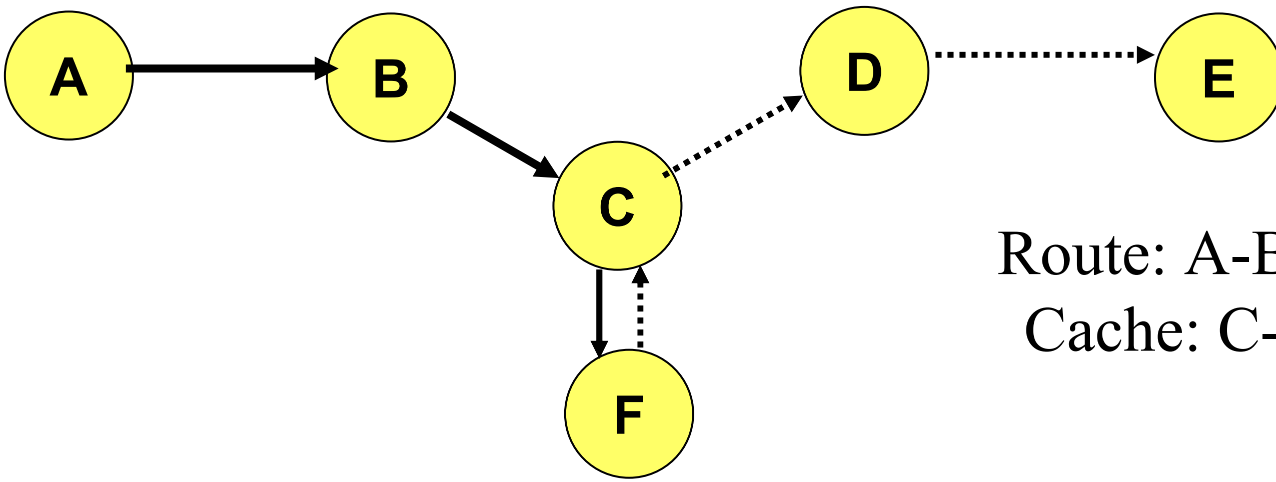
When node Z sends a route request for node C, node K sends back a route reply [Z,K,G,C] to node Z using a locally cached route

Use of Route Caching: Benefits - (2) Reduction in Propagation of Route Requests



- Route Reply (RREP) from node K limits flooding of RREQ.

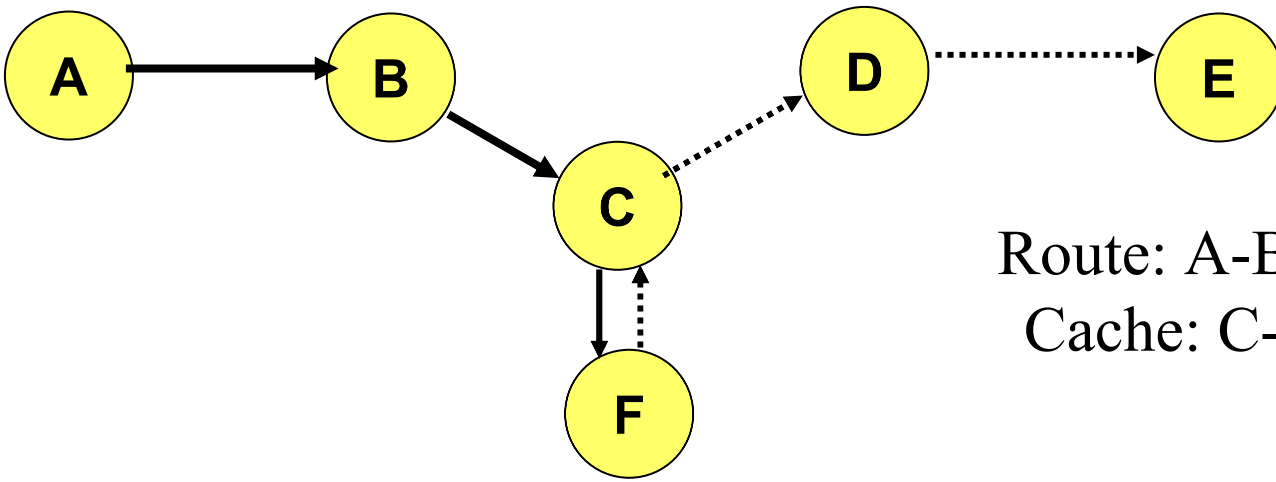
Duplication of Route Hops



Route: A-B-C-F
Cache: C-D-E

- Example: A needs a route to E
- A route reply that is generated from the route cache must avoid duplication of hops, e.g.
- Consider a route request for node E that has been received by F
- F has a route C-D-E in its route cache from itself to E

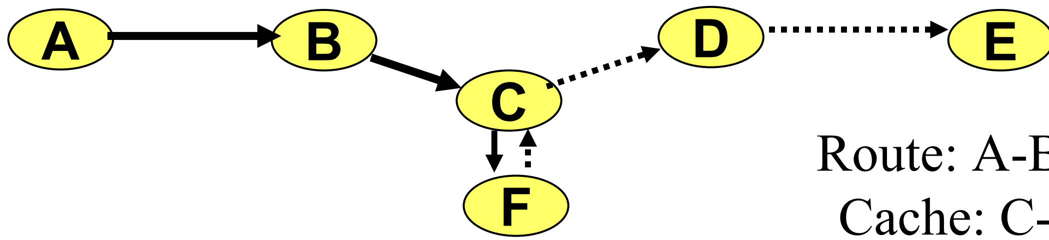
Duplication of Route Hops



Route: A-B-C-F
Cache: C-D-E

- The concatenation of the accumulated route from RREQ (A-B-C-F) and route in the route cache (C-D-E) has a duplicate node C -> in passing from C to F, and back to C
- This must be avoided by editing the route (A-B-C-F-C-D-E) by F e.g. A-B-C-D-E

Duplication of Route Hops



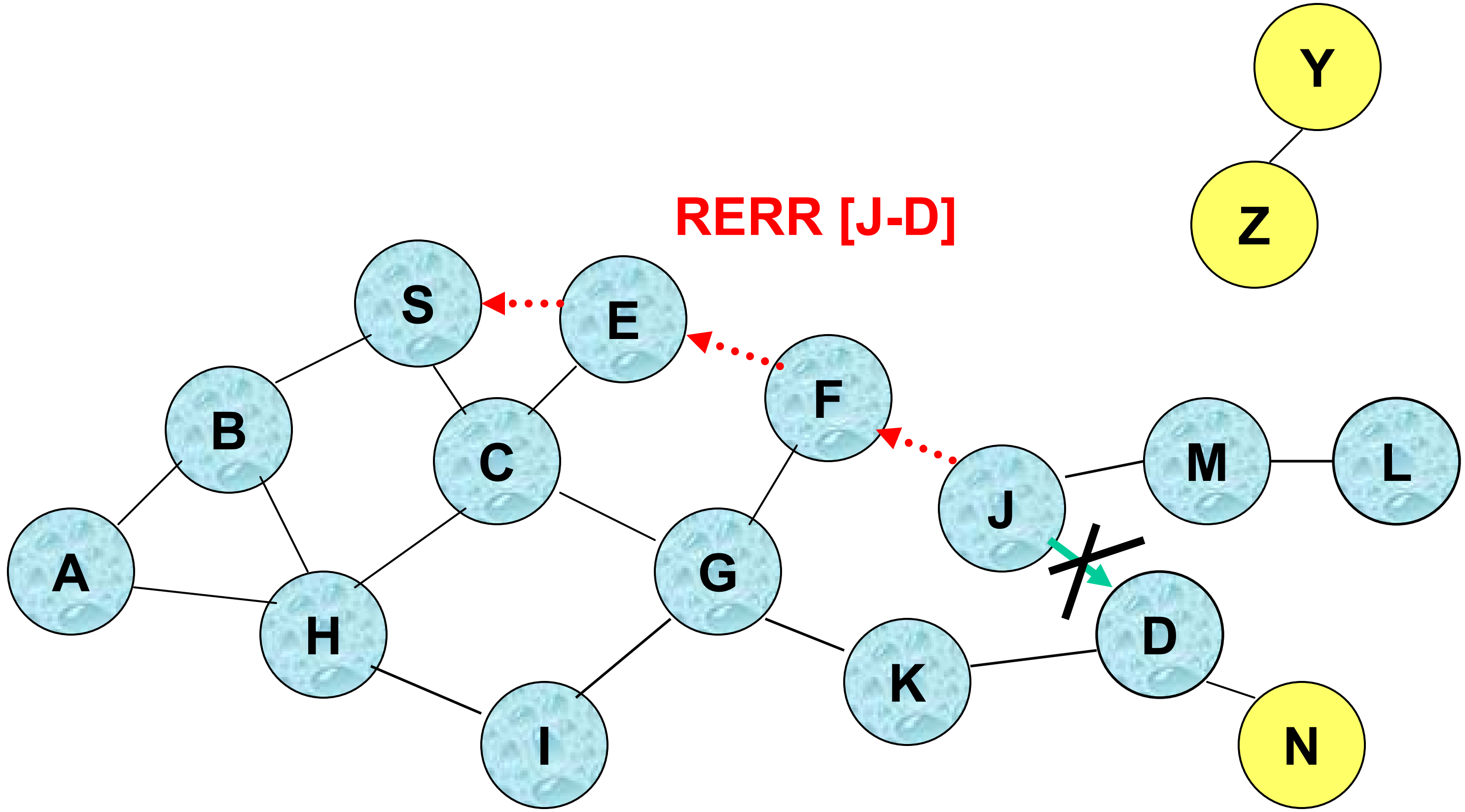
Route: A-B-C-F
Cache: C-D-E

- Notice that F returned the route to E in route reply from its cache even though its not on the path between A-E i.e. (A-B-C-D-E)
- DSR route discovery does not allow nodes like F to reply to RREQ

Outline

- Introduction
- Route Discovery
- Route Cache
- Route Maintenance
  - Preventing Route Reply Storms
  - Route Request hop limits
  - Packet Salvaging
  - Automatic Route Shortening
  - Increased spreading of Route Error messages
- Summary

Route Error (RERR) - Example

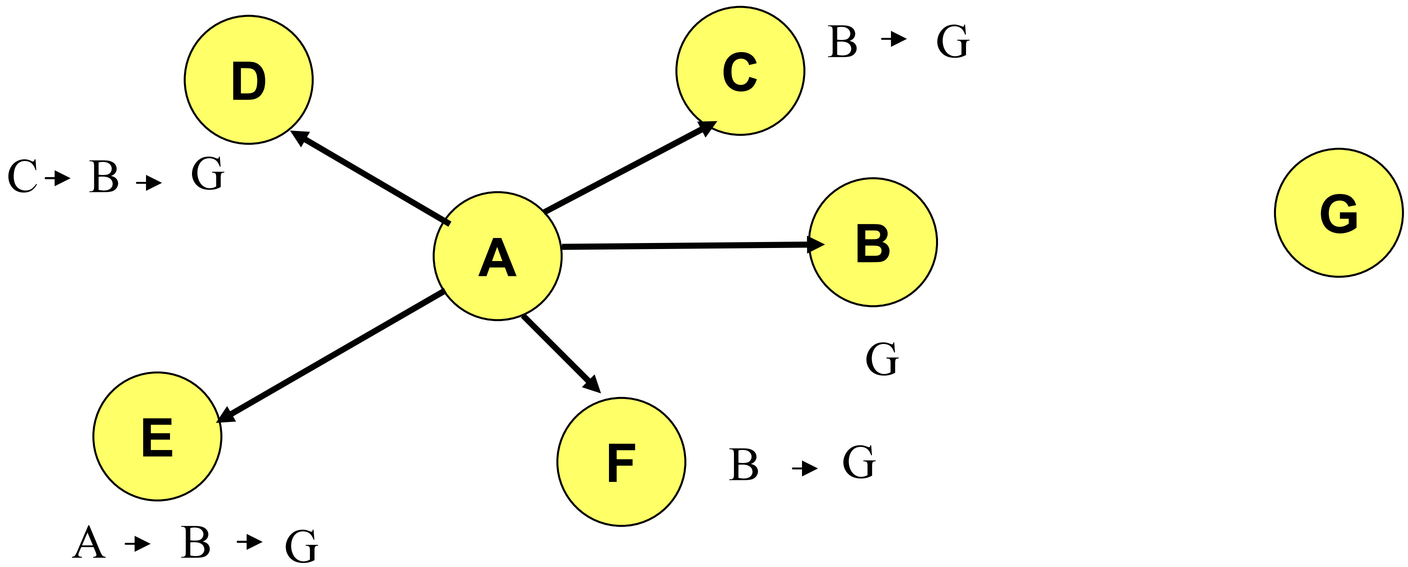


- Consider link between J and D fails
- J sends a route error to S along route J-F-E-S when its attempt to forward the data packet S (with route SEFJD) on J-D fails
- Nodes hearing RERR update their route cache to remove link J-D

Outline

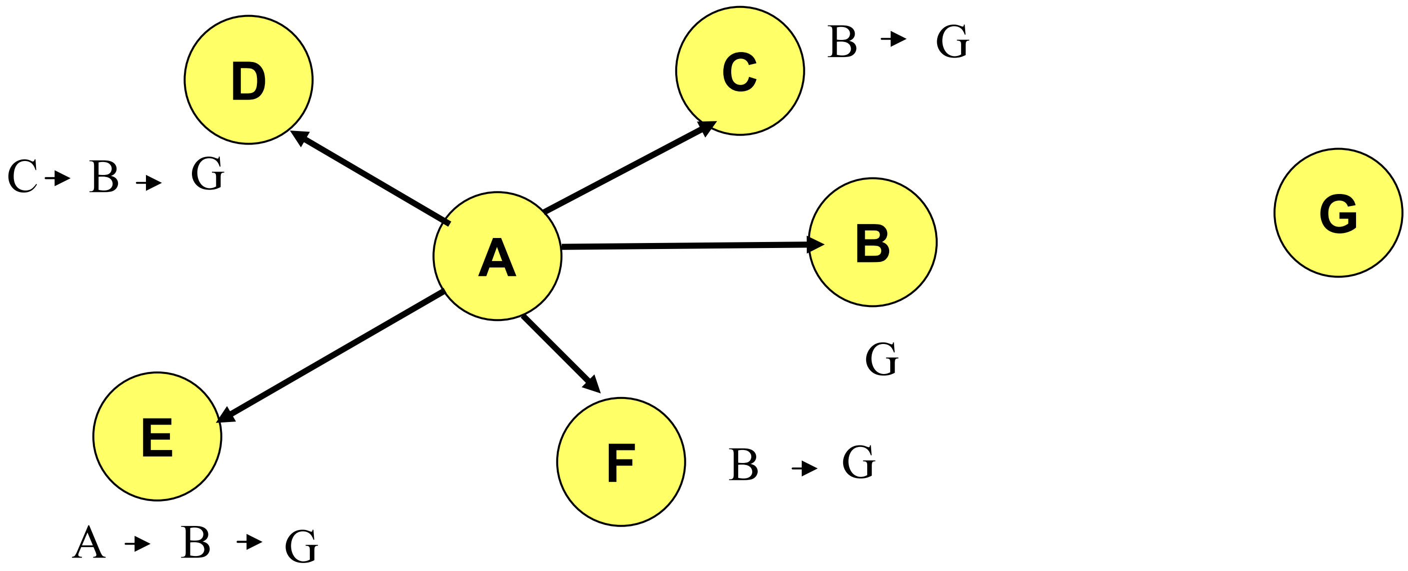
- Introduction
- Route Discovery
- Route Cache
- Route Maintenance
  - Preventing Route Reply Storms
  - Route Request hop limits
  - Packet Salvaging
  - Automatic Route Shortening
  - Increased spreading of Route Error messages
- Summary

Route Reply Storms



- Using route cache nodes can reply to RREQ, if they have the route
- If lots of node reply at the same time, it can cause route reply storm
- In the Figure nodes B, C, D, E, F all have A's route request to destination G

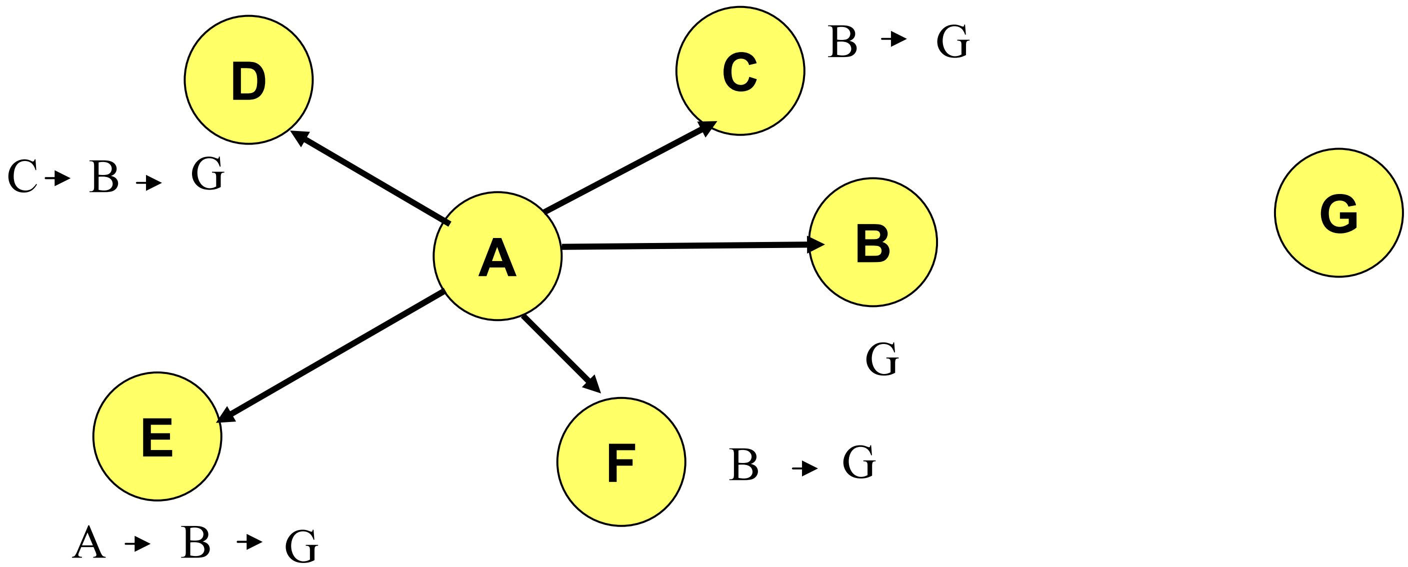
Route Reply Storms



- When A sends the RREQ, B, C, D, E, F can respond at the same time using their route caches -> because they all received the RREQ at the same time

- Simultaneous replies from B, C, D, E, F can cause collision at A (route reply storm)

Route Reply Storms



- Simultaneous replies from B, C, D, E, F can also cause local congestion at A
- Also each node may reply with a different route length, e.g. 1 hop (G) , 2 hops (B-G) , and 3 (C-B-G)
- Solution to Reply Storm – each node should randomly delay sending the route reply

Outline

- Introduction
- Route Discovery
- Route Cache
- Route Maintenance
  - Preventing Route Reply Storms
  - Route Request hop limits
  - Packet Salvaging
  - Automatic Route Shortening
  - Increased spreading of Route Error messages
- Summary

Route Request - Hop Limits

- Each RREQ message contains a field called hop-limit
- Hop limit controls the propagation of RREQ to the number of hops i.e. how many intermediate nodes are allowed to forward the RREQ
- Each receiving node decrements the hop-limit by 1 before forwarding.
- RREQ is not forwarded & is discarded by node when this limit becomes zero even before reaching the destination

Route Request - Hop Limits

- A RREQ with hop-limit zero will determine that the target is the one hop neighbor
- It also likely that this one hop neighbor has the source route in its cache
- If no RREP is received within a timeout period, a new RREQ is sent by the sender with no hop-limit
- Variations of this theme are sending RREQ with hop-limits of 0, 2, 4 etc. -> Similar to ring search of AODV
- This process increases the latency

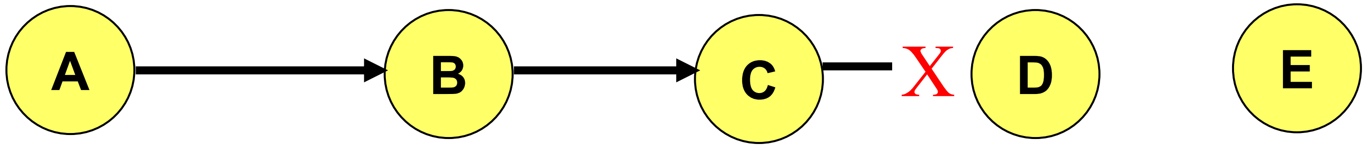
Outline

- Introduction
- Route Discovery
- Route Cache
- Route Maintenance
  - Preventing Route Reply Storms
  - Route Request hop limits
  - Packet Salvaging
  - Automatic Route Shortening
  - Increased spreading of Route Error messages
- Summary

Route Maintenance - Packet Salvaging

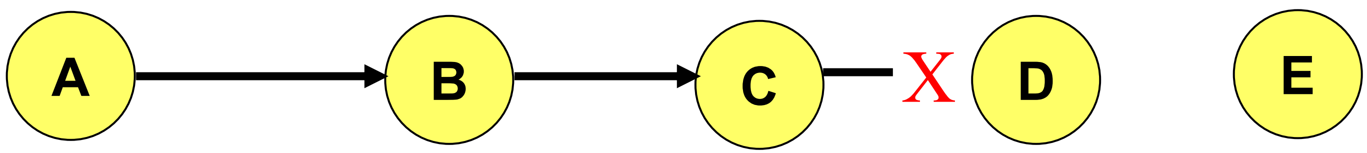
- When a node discovers that it cannot forward a data packet because the next-hop link is broken, it generates RERR
  - Sends RERR upstream
  - Searches its own cache to find an alternate route from itself to destination to forward this packet
  - If route is found, the node modifies the route as per the route cache and forwards to the next hop node

Route Maintenance - Packet Salvaging - Example



- In this example C is not able to forward the packet to D and E
  - C Generates RERR
  - Examines its route cache for an alternate path to E
  - If found, modifies the source route in the packet
  - Forwards the packet
  - Otherwise packet is dropped

Route Maintenance - Packet Salvaging - Example

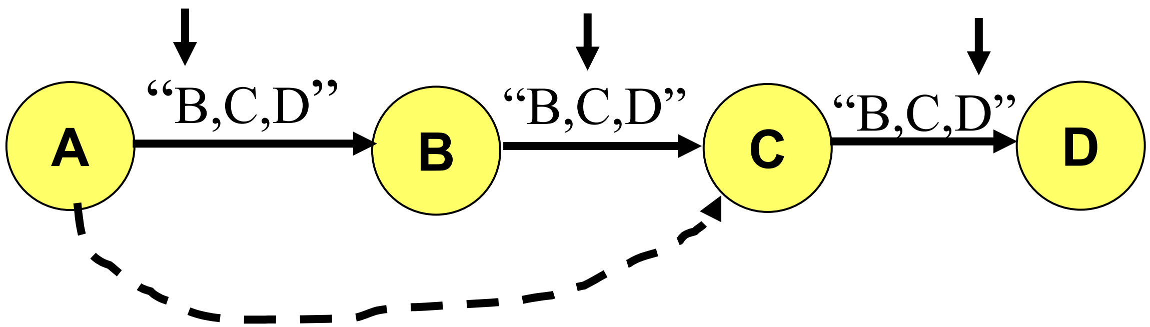


- When a packet is salvaged – its marked as "Salvaged"
- A Salvaged packet is salvaged only one time to avoid routing loops when salvaged at multiple locations
- A recommended strategy for salvaging is
  - Breakdown the address into two parts – prefix address (hops that are used until now) and suffix address (address from the route cache)
  - This strategy avoids backtracking from the current node to an already traversed node

Outline

- Introduction
- Route Discovery
- Route Cache
- Route Maintenance
  - Preventing Route Reply Storms
  - Route Request hop limits
  - Packet Salvaging
  - Automatic Route Shortening
  - Increased spreading of Route Error messages
- Summary

Route Maintenance - Route Shortening



- Routes may be shortened if one of intermediate nodes become unnecessary
- The vertical arrow shows the one –hop destination e.g. B, C, D, with arrow on B means B is the destination
- If C overhears that A is forwarding a packet to B that is destined to C, then
- C sends a "Gratuitous" message (Its RREP message) to original sender A.
- The RREP informs A to route packets as A-C-D instead of A-B-C-D

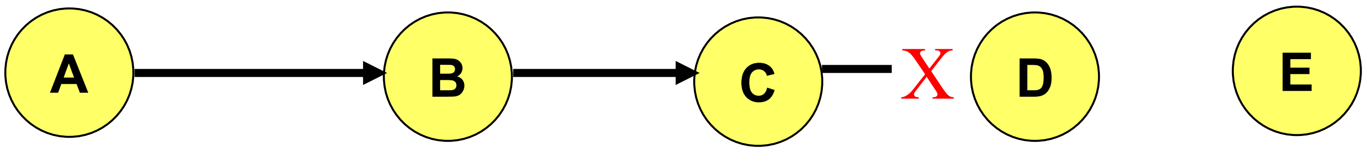
Outline

- Introduction
- Route Discovery
- Route Cache
- Route Maintenance
  - Preventing Route Reply Storms
  - Route Request hop limits
  - Packet Salvaging
  - Automatic Route Shortening
  - Increased spreading of Route Error messages
- Summary

Route Maintenance - Spreading of Route Error Message

- When a source node receives an RERR in response to a data packet that it forwarded
  - It piggybacks this RERR on a new RREQ that it forwards to its neighbors
  - Neighbors get aware of the RERR and update their route caches
  - This helps in reductions in getting the stale routes in RREP sent by the neighbors

Route Maintenance - Spreading of Route Error Message - Example

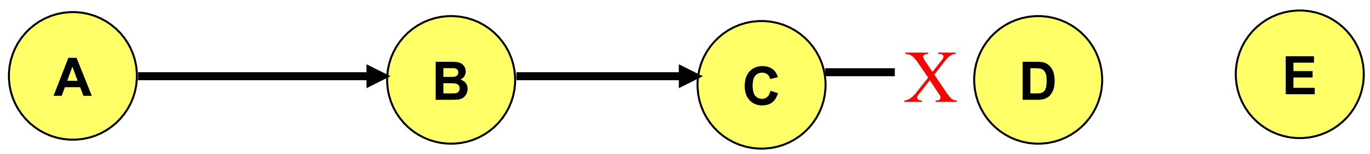


- Node A learns from RERR generated by C that link between C-D is broken
- A removes the link from its own route cache
- A - initiates a new route discovery (assuming it does not have another route to E) and piggybacks a copy of RERR message on RREQ
- This ensures that every node becomes aware of this link being broken and they update their route cache
- This also ensures nodes do not get replies with old routing information

Outline

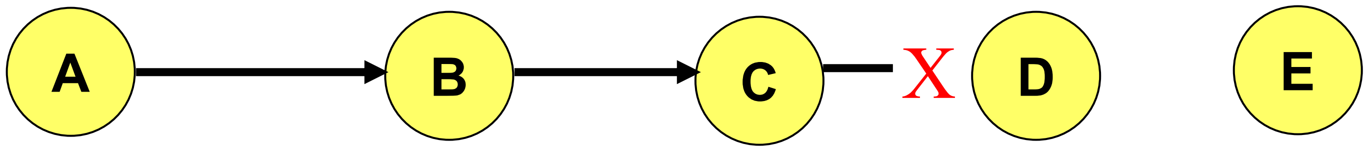
- Introduction
- Route Discovery
- Route Cache
- Route Maintenance
  - Preventing Route Reply Storms
  - Route Request hop limits
  - Packet Salvaging
  - Automatic Route Shortening
  - Increased spreading of Route Error messages
  - Caching negative information
- Summary

Route Maintenance - Caching Negative Information



- In certain situations, caching of negative information can help DSR. For example,
  - If A knows that link C-D is broken, it can keep this information in its routing cache for a specified time (using a timer) , e.g. by making the distance to routes through C as infinity
  - A will not use this path in response to any RREP it receives for subsequent RREQs
  - After the expiration of timer, the link can be added again in the route cache with correct hop counts, if link is repaired

Route Maintenance - Caching Negative Information



- Consider a case where link quality is varying with respect to time i.e. its in fade for some time. For example,
  - Assume that link C-D is in fade, i.e. its healthy for an interval and broken for another interval.
  - By keeping the information that the link is broken, the node can prevent the addition of this link in its route cache when it becomes healthy again
  - It can keep this information in its routing cache for a specified times (using a timer) till the link become normal
  - After the expiration of timer, the link can be added again in the route cache with correct hop counts
  - This mechanism prevents oscillations in the route cache

Route Caching: Beware!

- 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

Dynamic Source Routing: Advantages

- Routes maintained only between nodes who need to communicate
  - 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 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
- Increased contention if too many route replies come back due to nodes replying using their local cache
  - Route Reply *Storm* problem

Dynamic Source Routing: Disadvantages

- 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

AODV Vs DSR

- 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

AODV and DSR Differences

- DSR route cache entries do not have lifetimes (at present, only proposed); AODV route table entries do have lifetimes

Summary

- In this lecture we discussed the Dynamic Source Routing Protocol (DSR)
- We discussed
  - Route Discovery
    - Source Routing (Accumulation of routes in the packet header)
    - Role of route cache in speeding up the route discovery and reducing the propagation of RREQs/RREPs
  - Route Maintenance
    - Route cache and the role it plays in route maintenance e.g. how to prevent route reply storms, packet salvaging etc.
  - Comparisons with AODV