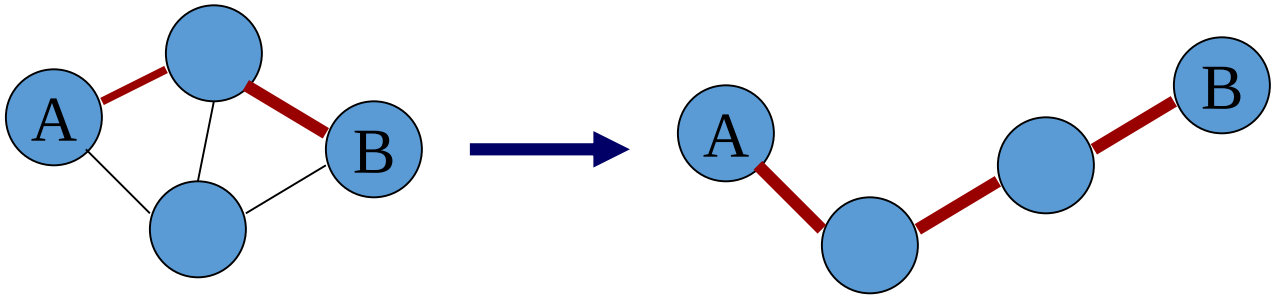
Routing in Ad-hoc Network

Mobile Ad Hoc Networks (MANET)

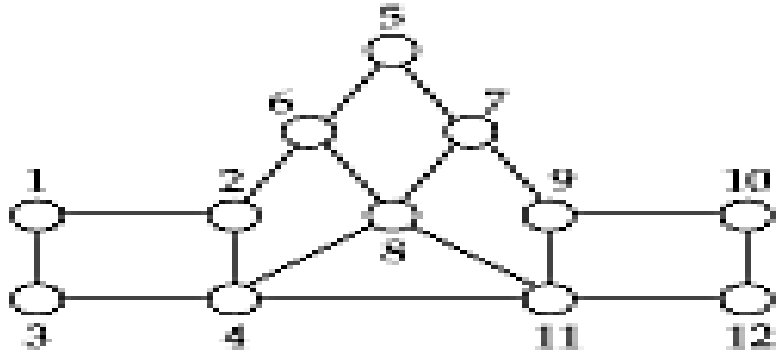
- Host movement frequent
- Topology change frequent



- No cellular infrastructure. Multi-hop wireless links
- Data must be routed via intermediate nodes

Traditional Routing

- A *routing protocol* sets up a *routing table* in *routers*
- A node makes a *local* choice depending on *global* topology



ROUTING TABLE AT 1

Destination	Next hop
1	—
2	2□
3	3□
4	3□
5	2□
6	2

Destination	Next hop
7	2
8□	2□
9□	2□
10□	2□
11□	3□
12	3

Routing in MANET

- Finding a path from a source to a destination
- Issues
  - Frequent route changes
    - amount of data transferred between route changes may be much smaller than traditional networks
  - Route changes may be related to host movement
  - Low bandwidth links

- Goal of routing protocols
  - decrease routing-related overhead
  - find short routes
  - find “stable” routes (despite mobility)

Routing Protocols

- Proactive protocols
  - Traditional distributed shortest-path protocols
  - Maintain routes between every host pair at all times (O(N) state per node, N = #nodes)
  - Low latency, suitable for real-time traffic
  - Based on periodic updates; High routing overhead
  - Example: DSDV (destination sequenced distance vector)

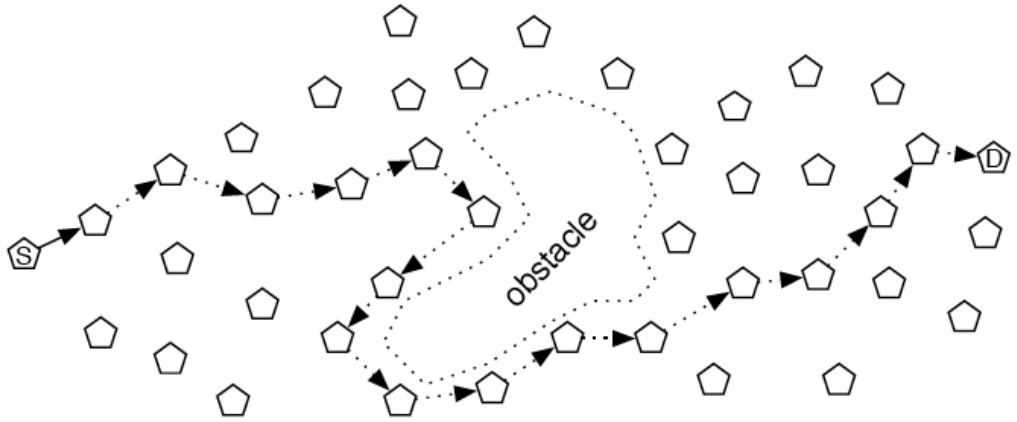
- Reactive protocols
  - Determine route if and when needed
  - Source initiates route discovery
  - Example: DSR (dynamic source routing)
  - Saves energy and bandwidth during inactivity
  - Can be bursty -> congestion during high activity
  - Significant delay might occur as a result of route discovery
  - Good for light loads, collapse in large loads

Routing Protocols

- Hybrid protocols
  - Adaptive; Combination of proactive and reactive
  - Proactive for neighborhood, Reactive for far away (ZRP (zone routing protocol))
  - Proactive for long distance, Reactive for neighborhood (Safari)
  - Attempts to strike balance between the two
- Hierarchical Routing
  - Nodes are organized in clusters
  - Cluster head “controls” cluster
  - Trade off
    - Overhead and confusion for leader election
    - Scalability: intra-cluster vs intercluster
  - One or Multiple levels of hierarchy

Routing Protocols

- Geographical Routing
  - Nodes know their geo coordinates (GPS)
  - Route to move packet closer to end point
  - Protocols DREAM, GPSR, LAR
  - Propagate geo info by flooding (decrease frequency for long distances)



Dynamic Routing: a new approach

- DART [Ericsson et al.], L+ [Morris et al]
- Goal: can we enforce address aggregation
- But: nodes are moving
- Then: address should change
- General Idea
  - Separation of identity and address
    - Identity is who you are
    - Address is where you are
  - Rule for enforcing “structure” in addresses:
    - nearby nodes should have nearby addresses
  - Using the Rule, we can “aggregate” information

Dynamic Routing: a new approach

- Consequences:
  - Routing is simplified: address tell me where you are
  - Nodes with similar addresses are “near” each other
- Challenges:
  - Address allocation: When I move, change my address
  - ID to Address mapping: Given an ID, find the address

Reactive Routing

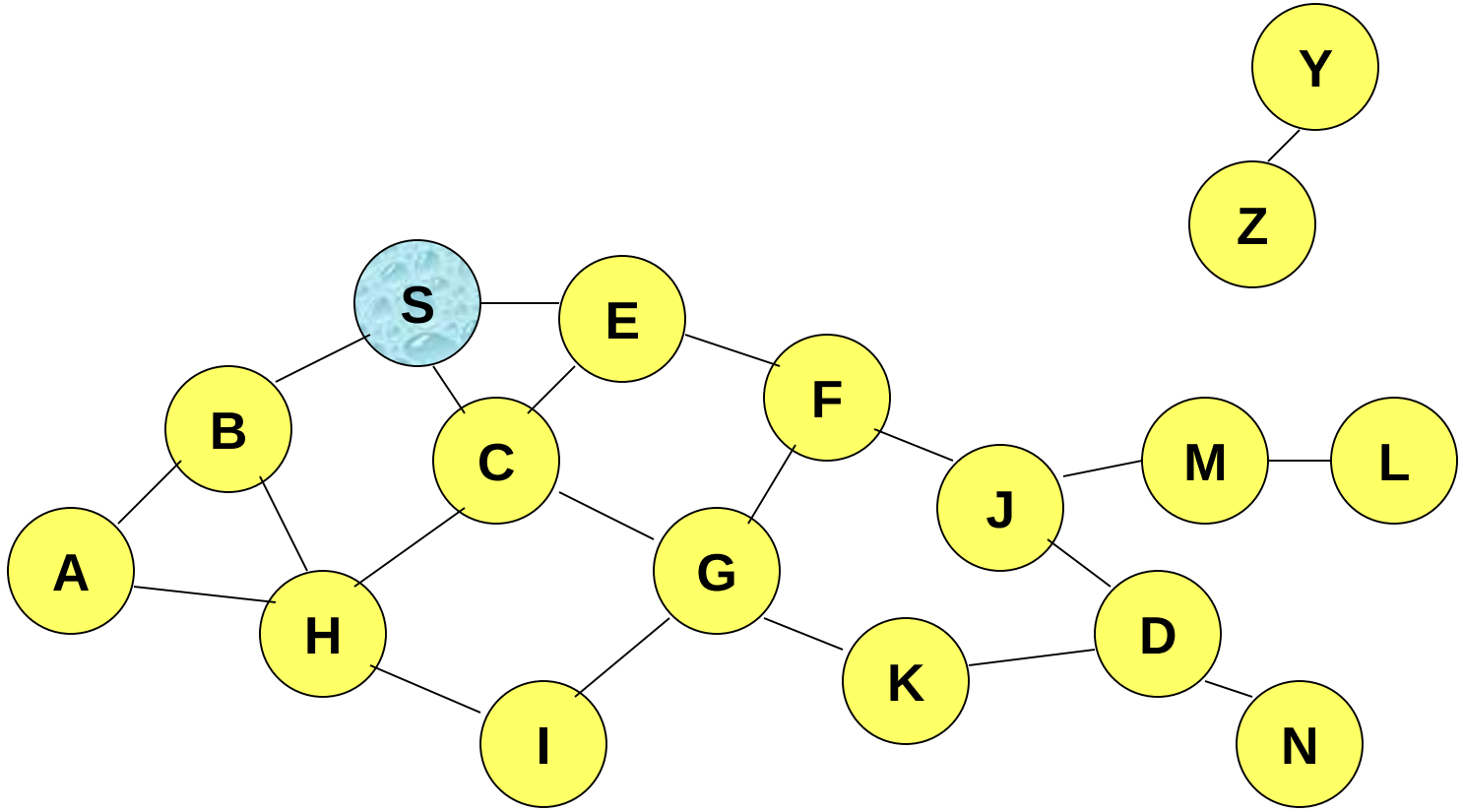
Dynamic Source Routing (DSR) [Johnson96]

- Route discovery
  - Undertaken when source needs a route to a destination
- Route maintenance
  - Used when link breaks, rendering specified path unusable

Route Discovery in DSR

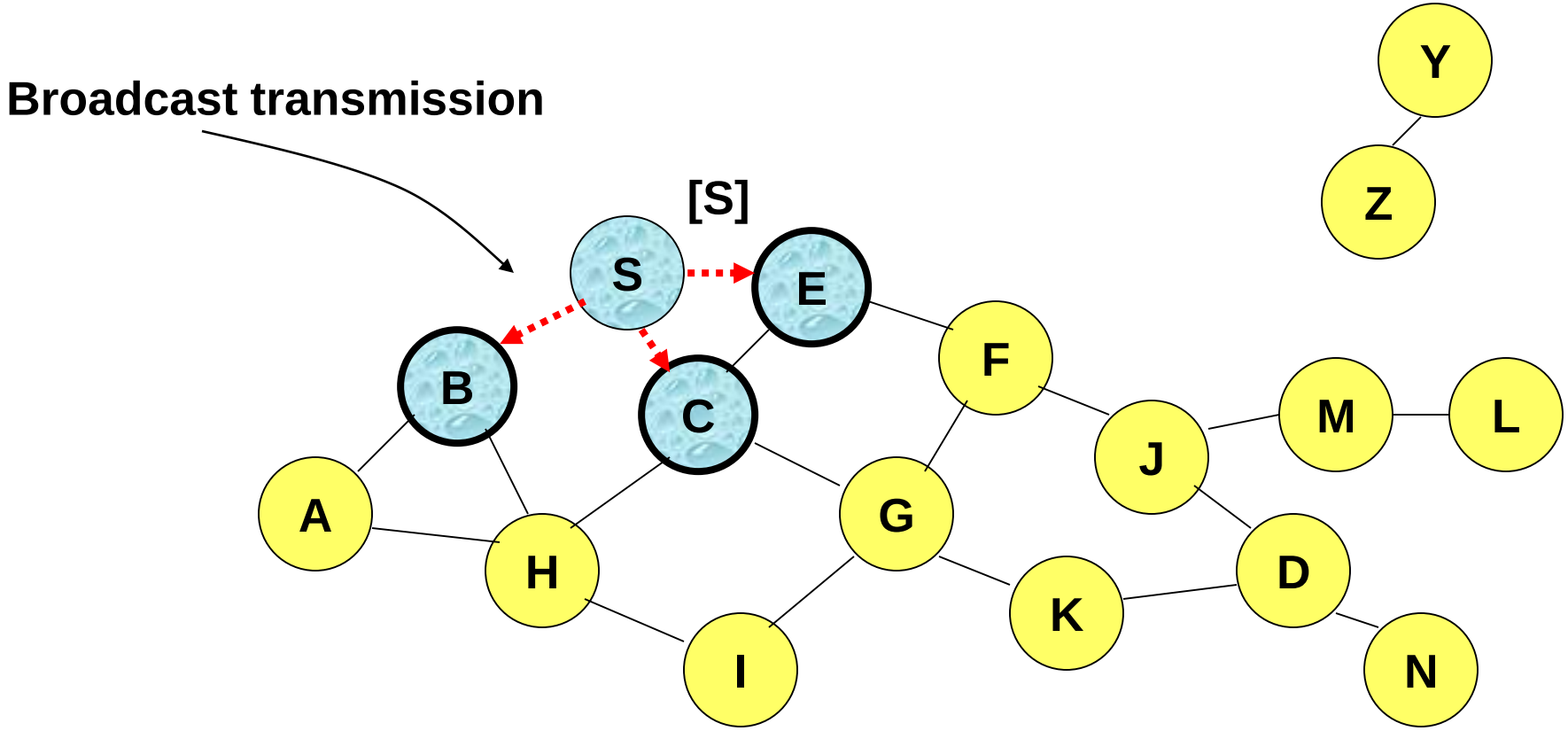
- When node S wants to send a packet to node D, but does not know a route to D, node S initiates a route discovery
- Source node S floods Route Request (RREQ)
  - broadcasts packet which contains source address, destination address, request id and path.
- Each node *appends own identifier* when forwarding RREQ

Route Discovery in DSR



Represents a node that has received RREQ for D from S

Route Discovery in DSR



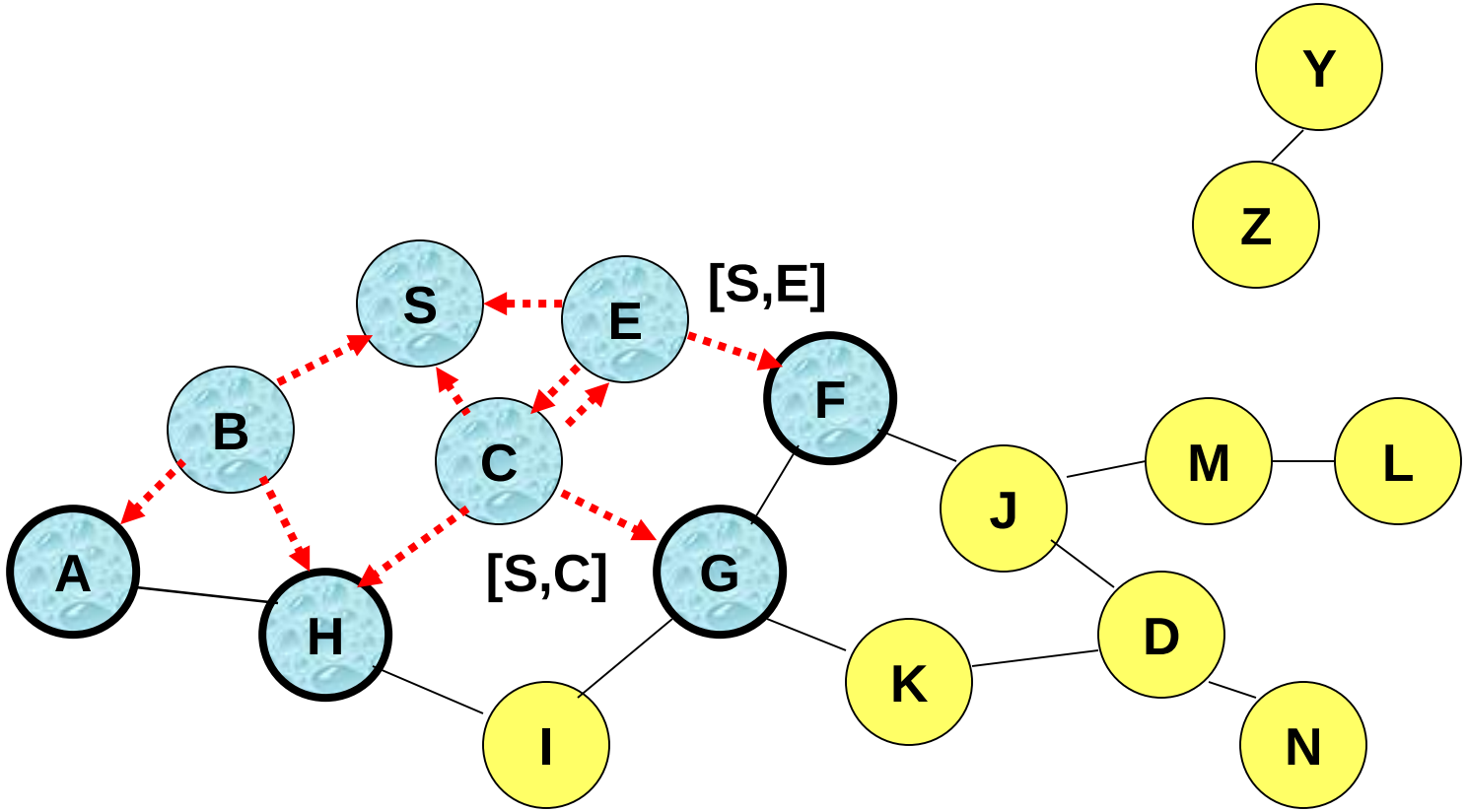
Represents transmission of RREQ

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

Route Discovery in DSR

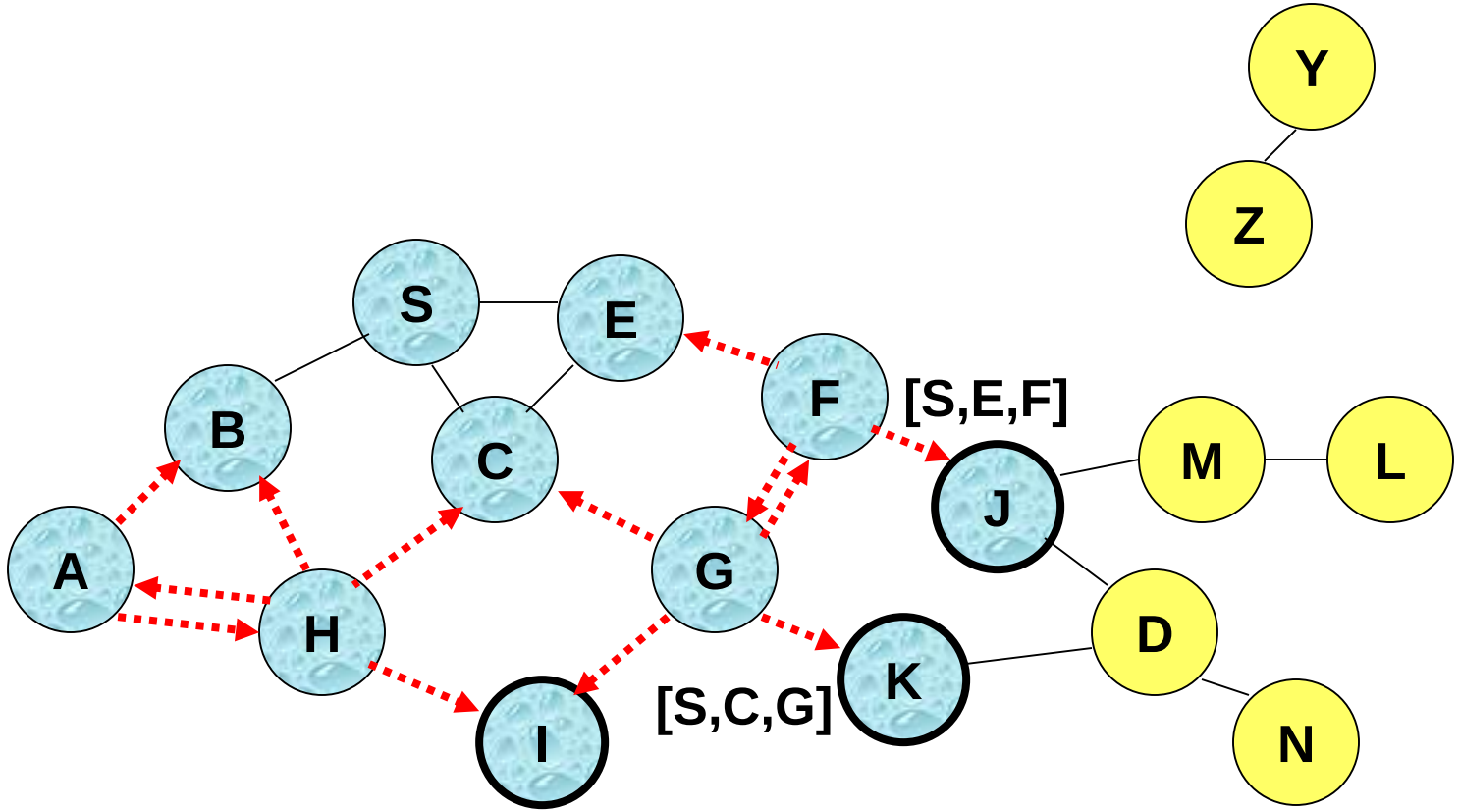
- If a node saw the packet before, discards it.
- Otherwise, the route looks up its route caches to look for a route to destination
- If not found, it appends its address into the packet, rebroadcasts
- If it finds a route in its route cache, sends a route reply packet, which is sent to the source by route cache or the route discovery

Route Discovery in DSR



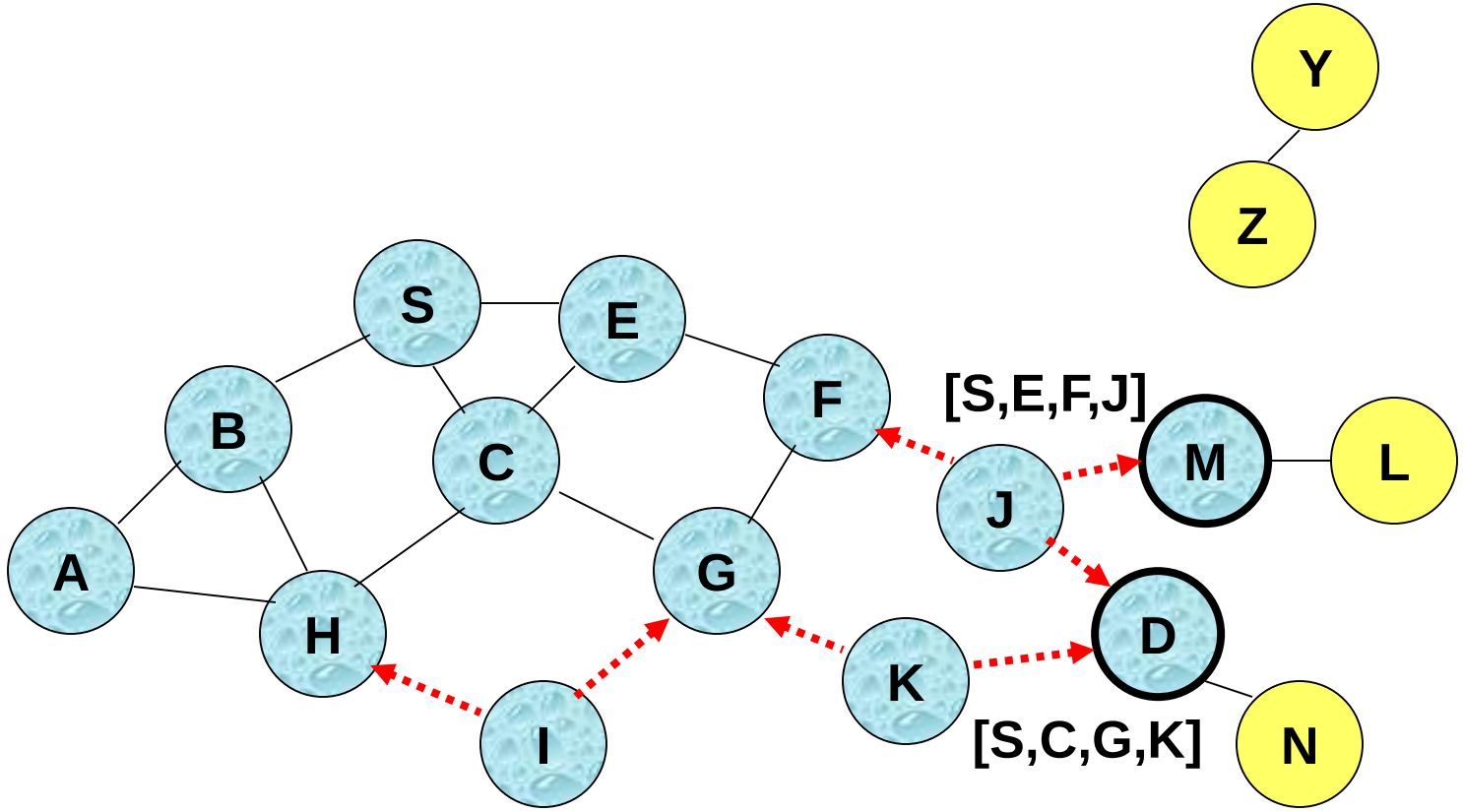
- **Node H receives packet RREQ from two neighbors:**
  potential for collision

Route Discovery in DSR



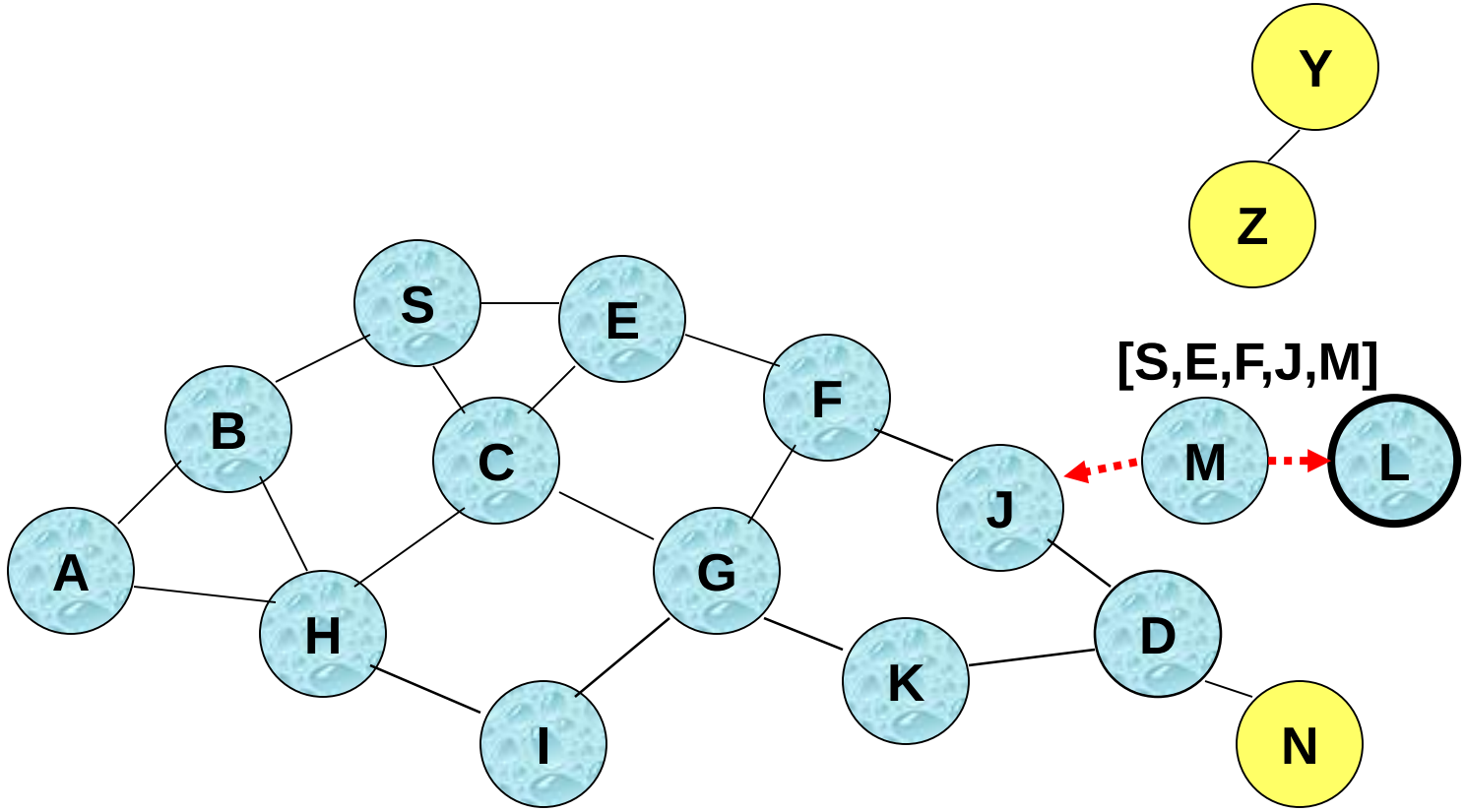
- **Node C receives RREQ from G and H, but does not forward it again, because node C has already forwarded RREQ once**

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

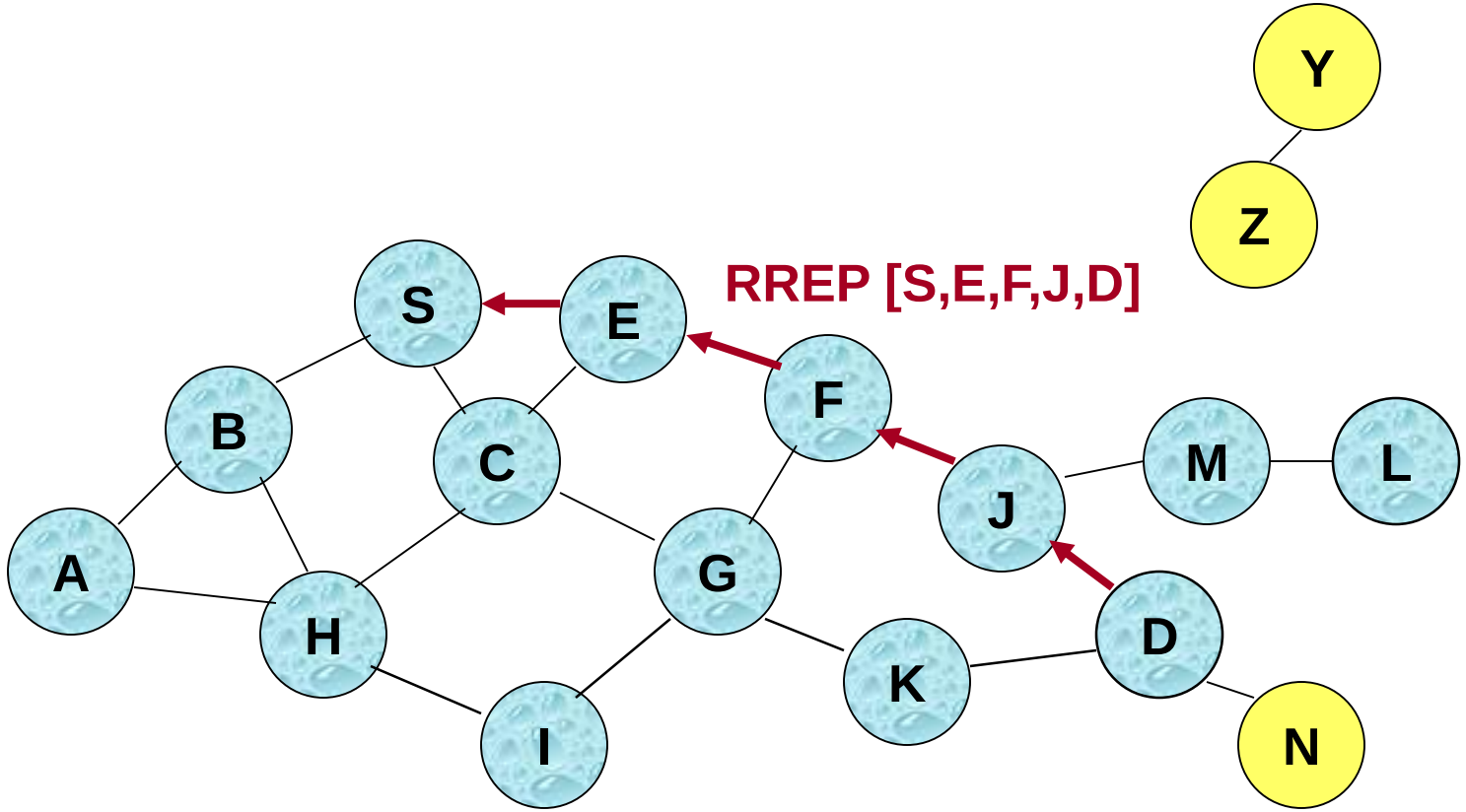


- Node D **does not forward** RREQ, because node D is the **intended target** of the route discovery

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 includes the route from S to D on which RREQ was received by node D

Route Reply in DSR

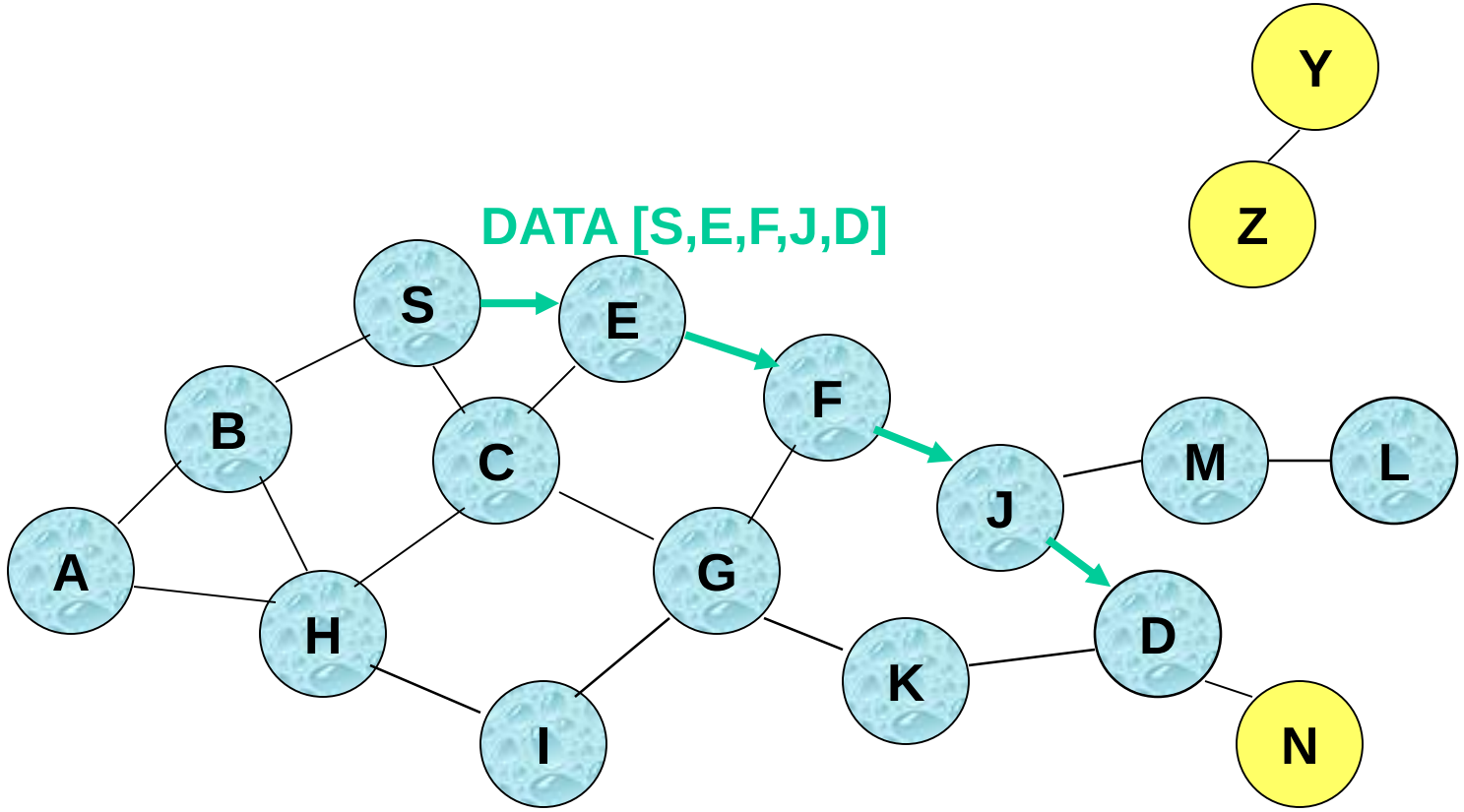


Represents RREP control message

Dynamic Source Routing (DSR)

- Node S on receiving RREP, caches the route included in the RREP
- When S wants to send a packet, it first checks its route cache
- If route exists, S constructs a source route in the packet's header
  - hence the name source routing
- If route expires or does not exist, sender initiates the Route Discovery Mechanism
- 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

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, node K learns route [K,G,C,S] to node S
- When node F forwards Route Reply RREP [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
- Problem: Stale caches may increase overheads

Route Maintenance

- Two types of packets used: Route Error Packet and Acknowledgement
- If transmission error is detected at data link layer, Route Error Packet is generated and sent to the original sender of the packet.
- The node removes the hop in error from its route cache when a Route Error packet is received
- ACKs are used to verify the correction of the route links.

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
- Potential 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
- Stale caches will lead to increased overhead

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

AODV

- Reactive or on Demand
- Uses bi-directional links
- Route discovery cycle used for route finding
- Maintenance of active routes
- Provides unicast and multicast communication

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 (RREP)
- Route Reply travels along the reverse path set-up when Route Request is forwarded

AODV Properties

- AODV discovers routes as and when necessary
  - Does not maintain routes from every node to every other
- Routes are maintained just as long as necessary
- Every node maintains its monotonically increasing sequence number -> increases every time the node notices change in the neighborhood topology
- AODV utilizes routing tables to store routing information
  - A Routing table for unicast routes
  - A Routing table for multicast routes

AODV Properties

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

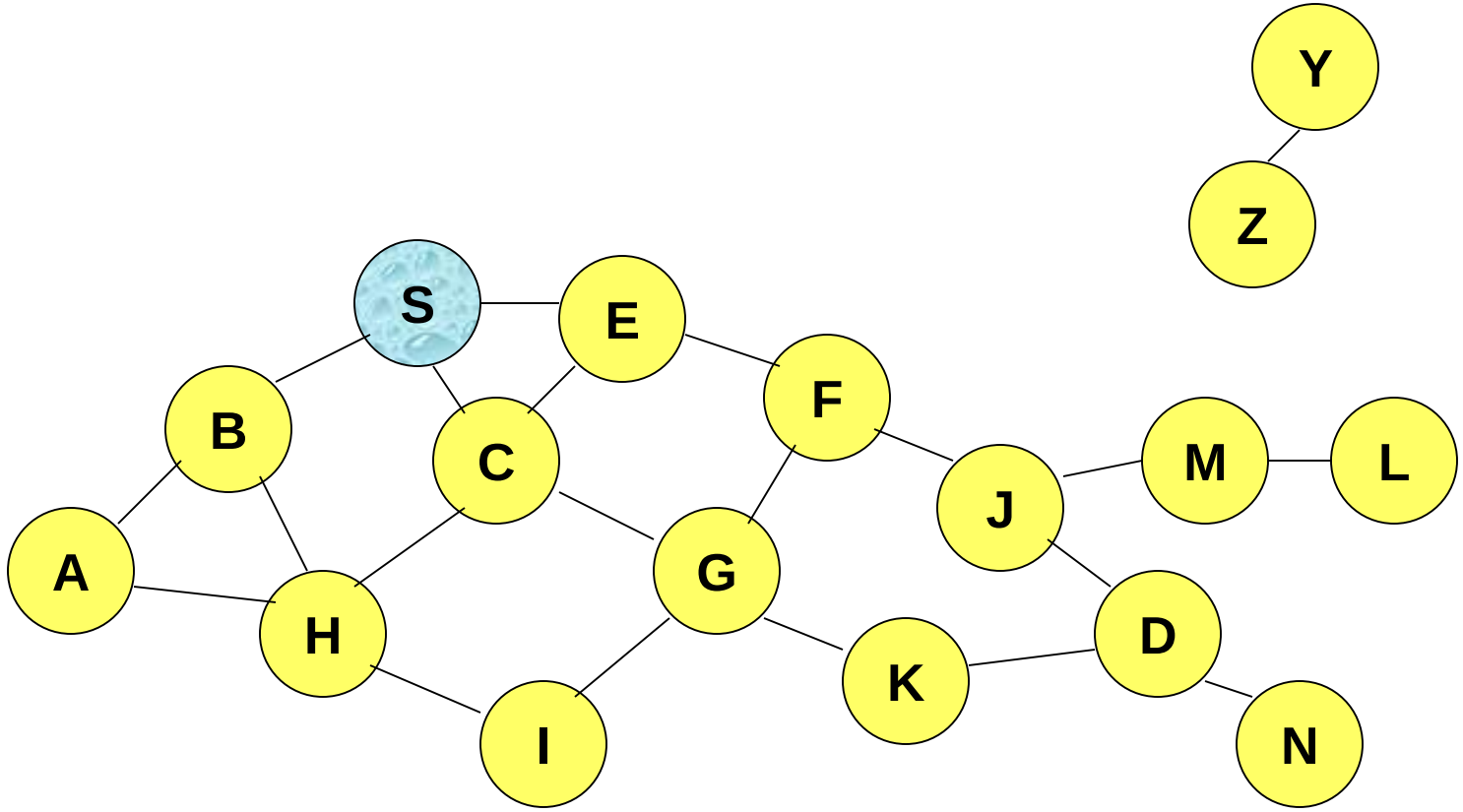
AODV Route Discovery

- 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
- 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

Fields in the RREQ packets

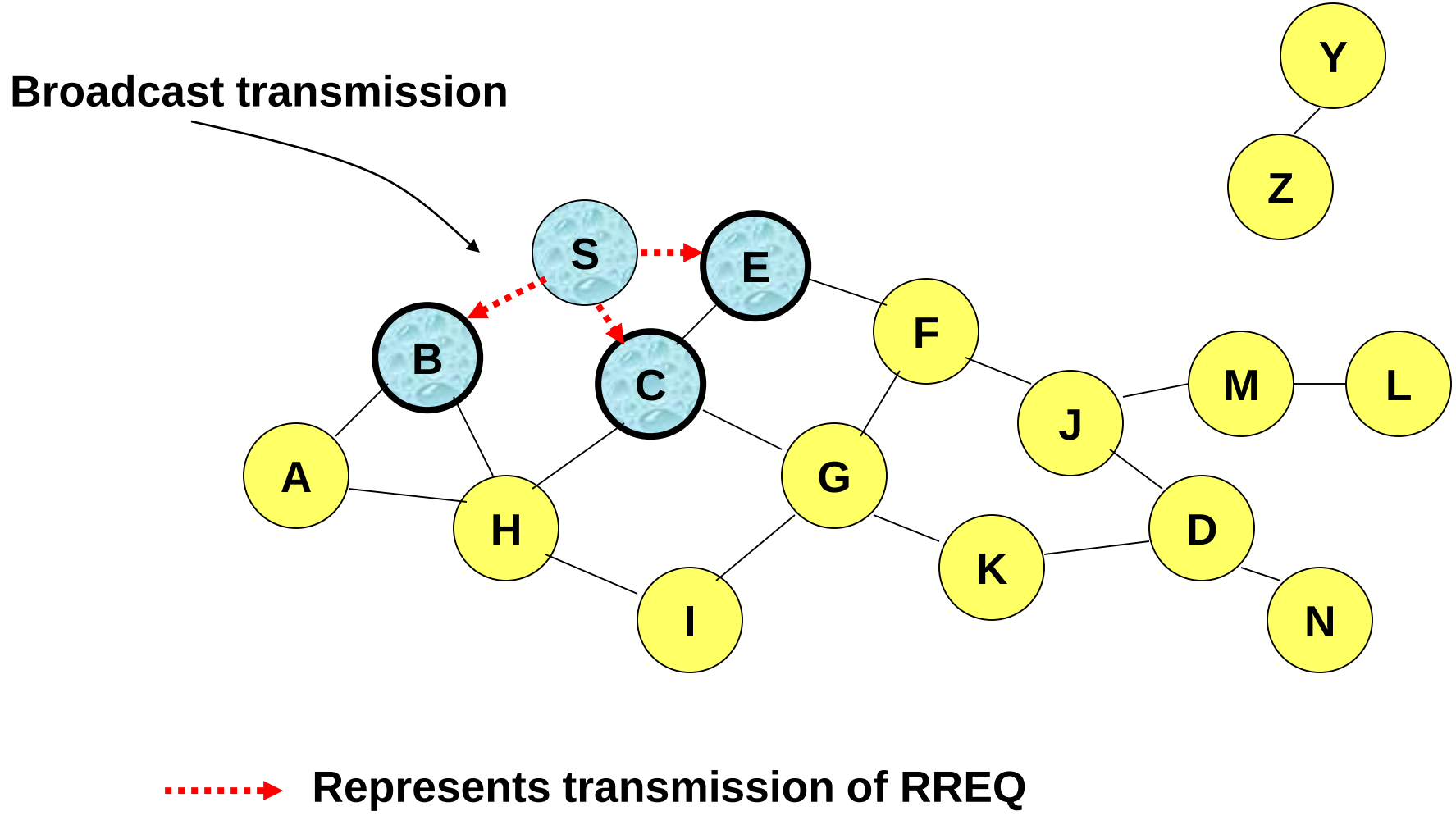
- Source Sequence Number is the current sequence number to be used for route entries pointing to (and generated by) the source of the route request.
- Destination Sequence Number is the last sequence number received in the past by the source for any route towards the destination.
- Broadcast ID is a sequence number uniquely identifying the particular RREQ when taken in conjunction with the source node's IP address.

Route Requests in AODV

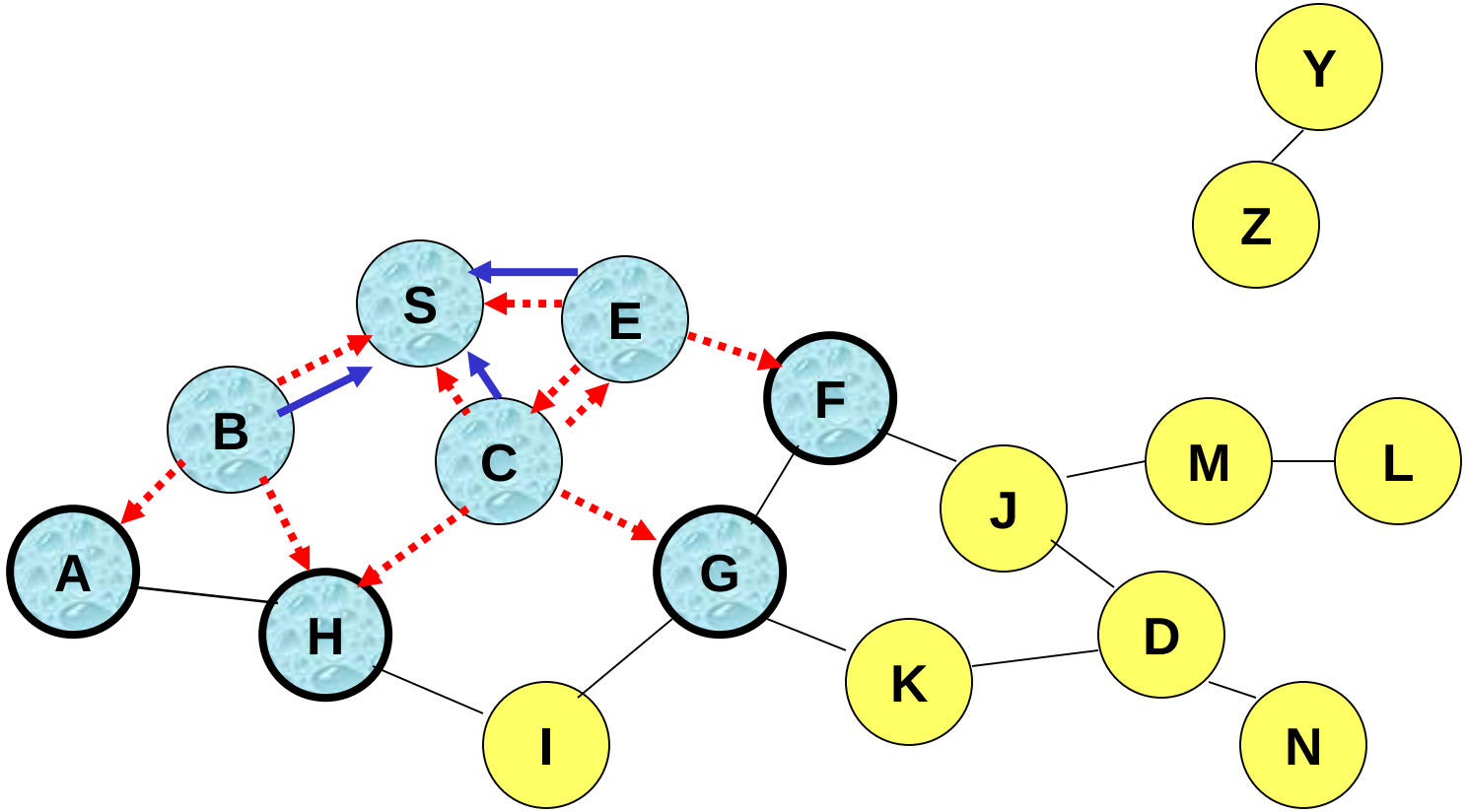


Represents a node that has received RREQ for D from S

Route Requests in AODV



Route Requests in AODV

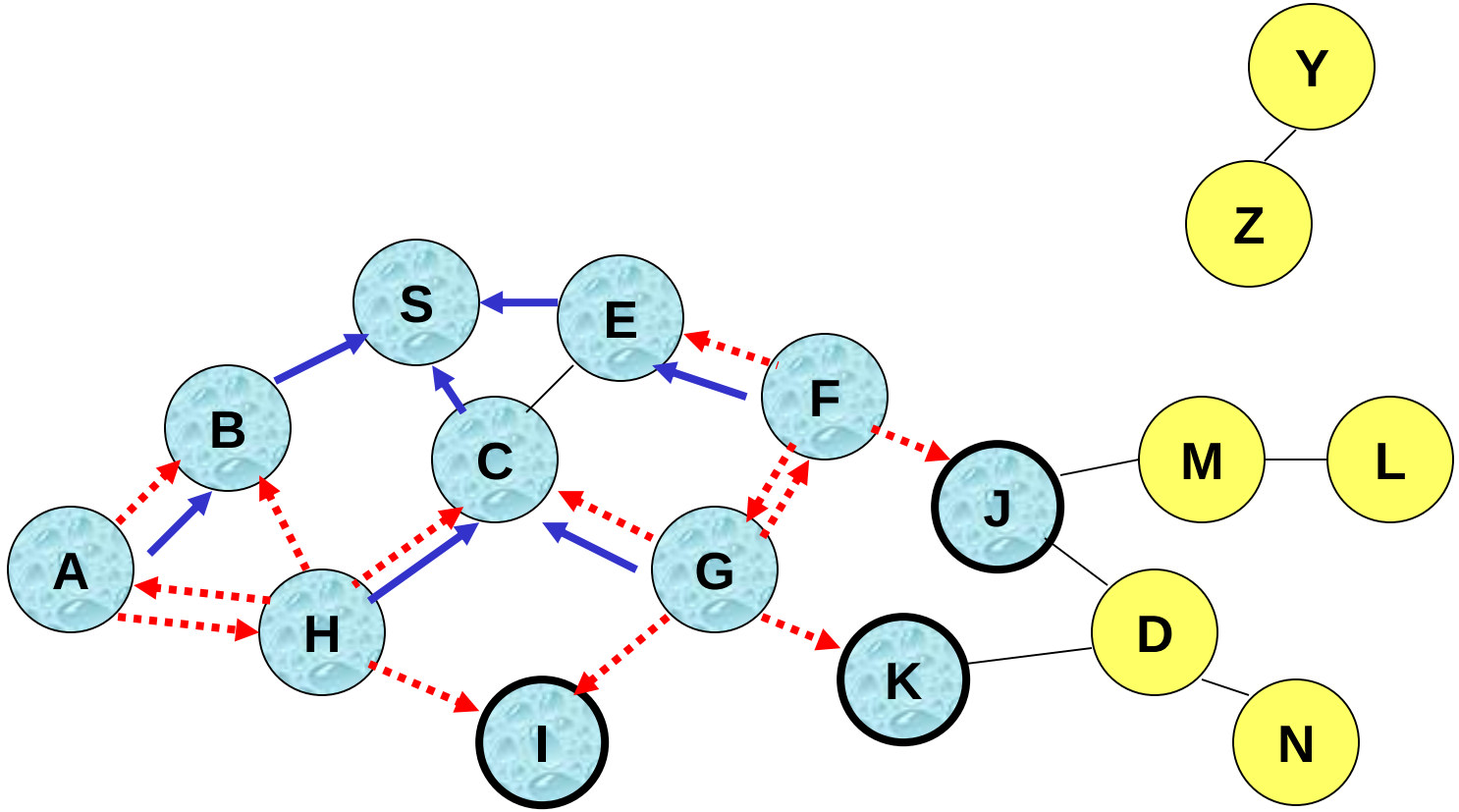


Represents links on Reverse Path

RREQ Propagation

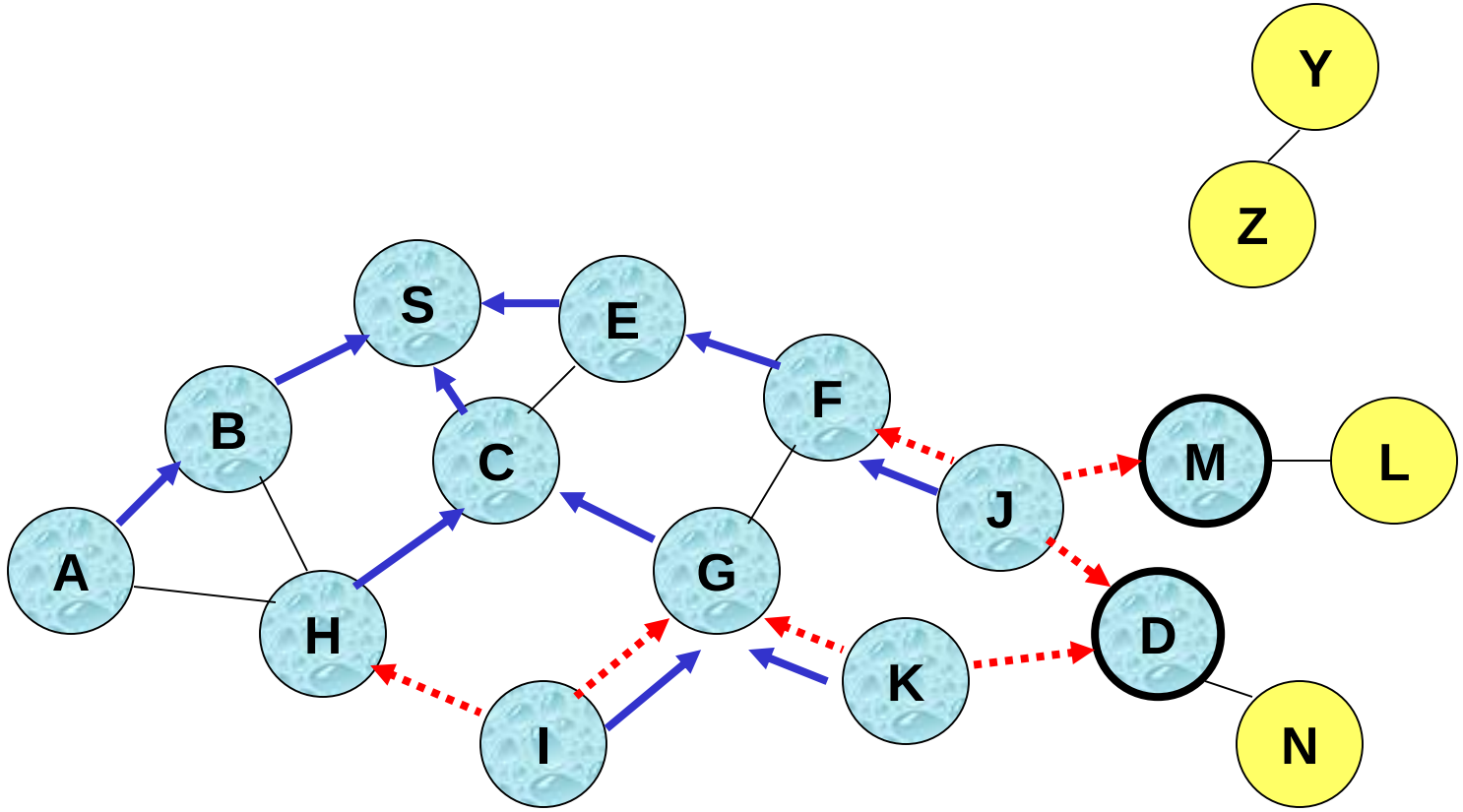
- Once an intermediate node receives a RREQ, the node sets up a reverse route entry for the source node in its route table
- Reverse route entry consists of <Source IP address, Source seq. number, number of hops to source node, IP address of node from which RREQ was received>

Reverse Path Setup in AODV

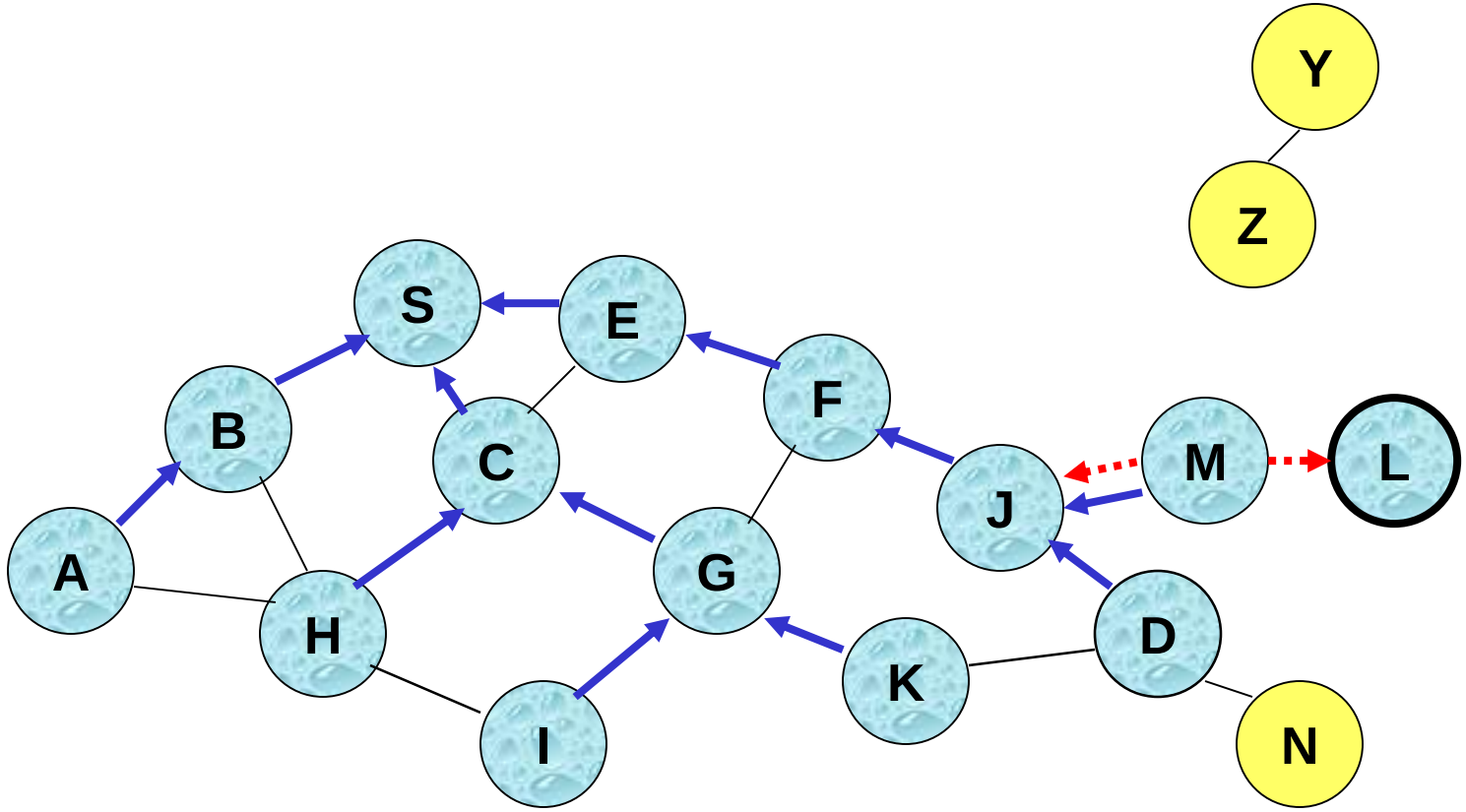


- **Node C receives RREQ from G and H, but does not forward it again, because node C has already forwarded RREQ once**

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**

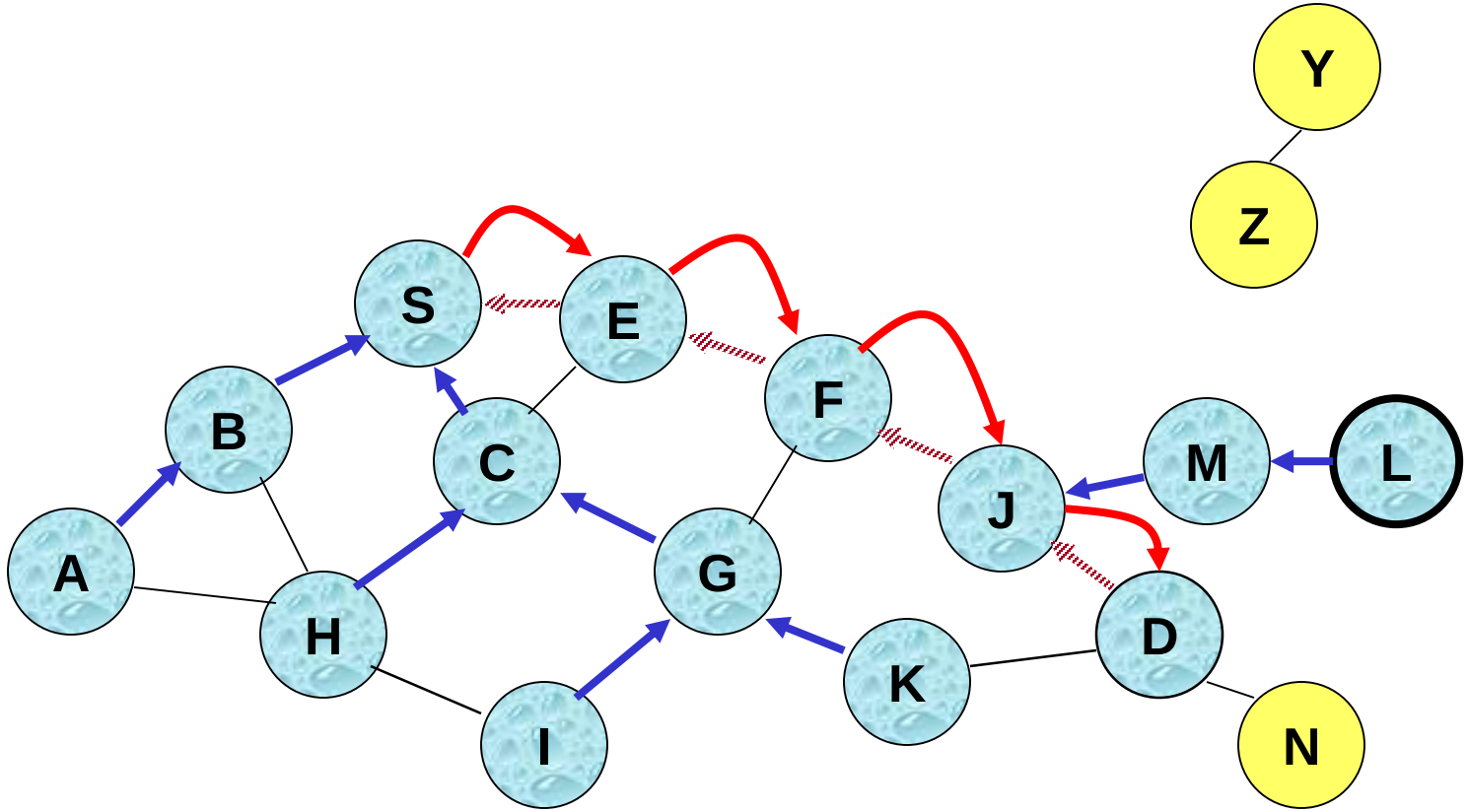
Propagation of RREP

- Using the reverse route a node can send a RREP (Route Reply packet) to the source
  - Reverse route entry also contains – life time field
- In order to respond to RREQ a node should have in its route table:
  - Unexpired entry for the destination
  - Seq. number of destination at least as great as in RREQ (for loop prevention)
- RREQ reaches destination -> RREP is sent back using unicasting (not flooding) to the source using reverse path
- An intermediate node may also send a Route Reply (RREP) provided that it knows a more recent path than the one previously known to sender
- Intermediate nodes that forward the RREP, also record the next hop to destination

Forward Path Setup

- When a node determines that it has a current route to respond to RREQ i.e. has a path to destination – It creates RREP (Route Reply)
- RREP contains <IP address of source and destination>
  - If RREP is being sent by destination, it will also contain the <current sqn # of destination, hop-count=0, life-time>
  - If RREP is sent by an intermediate node, it will contain its record of the <destination sequence number, hop-count=its distance to destination, its value of the life-time>
- When an intermediate node receives the RREP, it sets up a forward path entry to the destination in its route table
  - Forward path entry contains <IP Address of destination, IP address of node from which the entry arrived, hop-count to destination, life-time>
  - To obtain its distance to destination i.e. hop-count, a node increments the distance by 1
- After processing the RREP, the node forwards it towards the source

Forward Path Setup in AODV



Forward links are setup when RREP travels along the reverse path

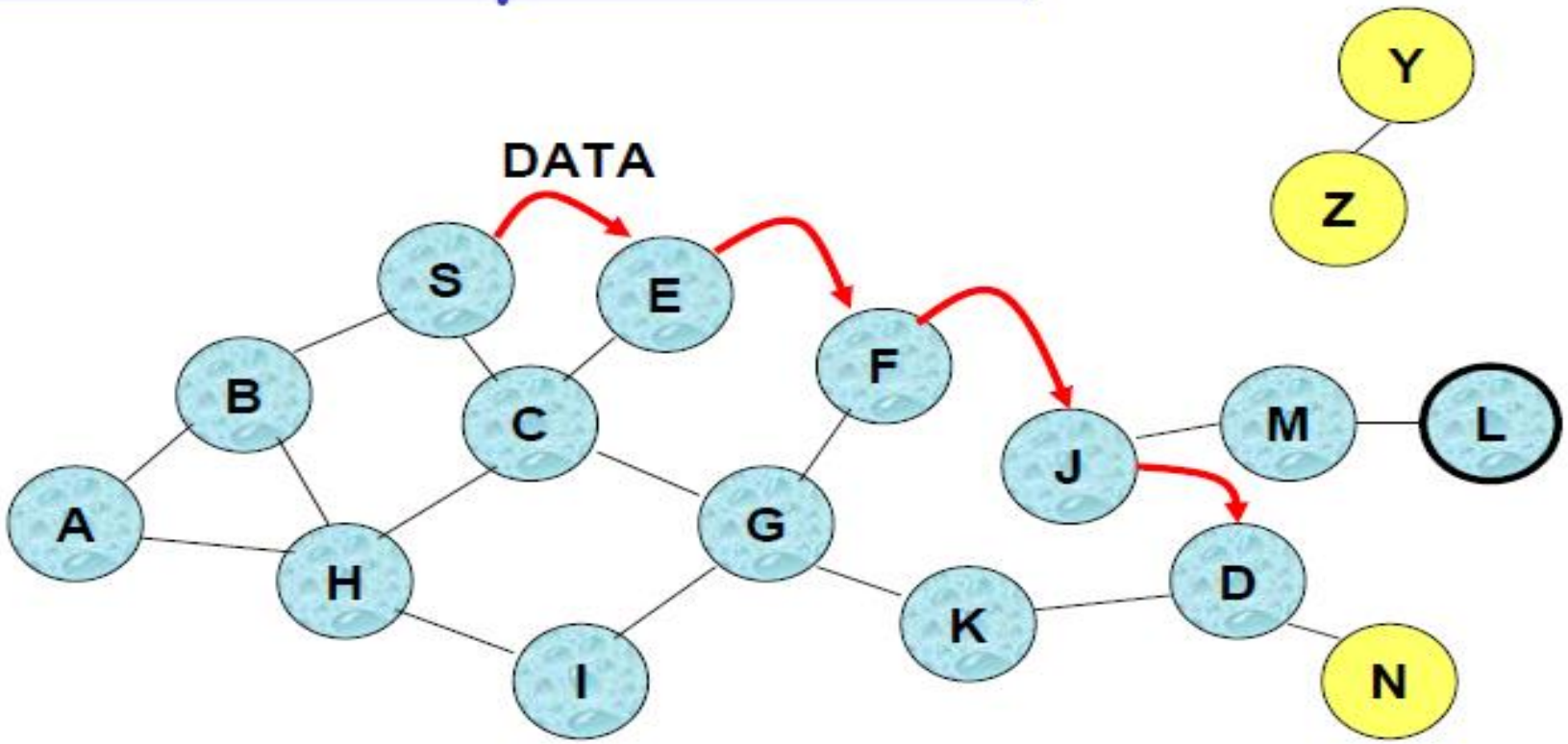
Represents a link on the forward path

Receipt of Multiple RREP

- A node may receive multiple RREP for a given destination from more than one neighbor
  - The node only forwards the first RREP it receives
  - May forward another RREP if that has greater destination sequence number or a smaller hop count
  - Rest are discarded -> reduces the number of RREP propagating towards the source
- Source can begin data transmission upon receiving the first RREP

Data Delivery in AODV

- Routing table entries used to forward data packet.
- Route is not included in packet header.



Discarding old paths

- 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)

Discarding old paths

- Whenever routes are not used -> get expired ->Discarded
  - Reduces stale routes
  - Reduces need for route maintenance
- Minimizes number of active routes between an active source and destination
- Can determine multiple routes between a source and a destination, but implements only a single route, because
  - Difficult to manage multiple routes between same source/destination pair
  - If one route breaks, its difficult to know whether other route is available
  - Lot of book-keeping involved

Link Failure

- 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

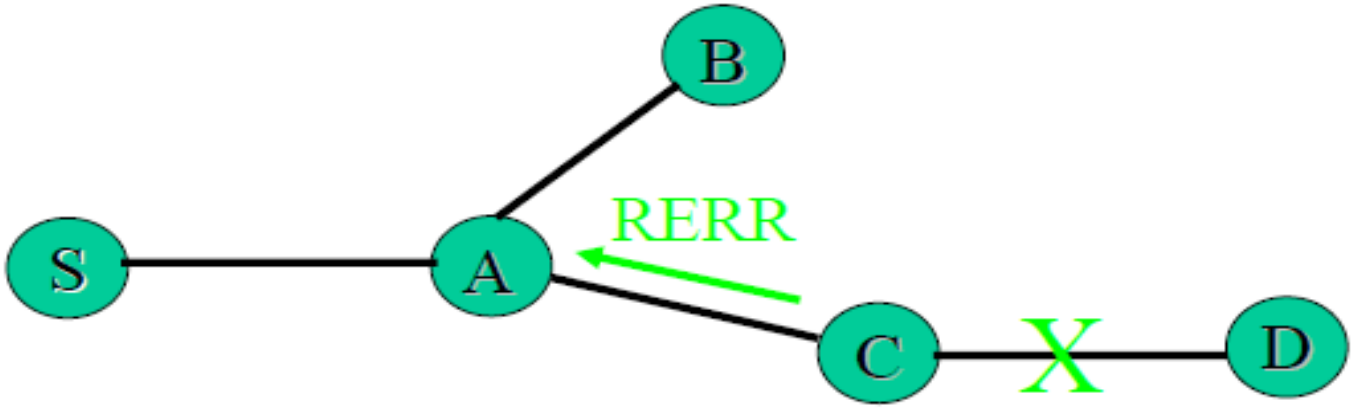
- If a source node moves, a new route discovery process is initiated
- If intermediate nodes or the destination move
  - The next hop links break resulting in link failures
  - Routing tables are updated for the link failures
  - All active neighbors are informed by RERR message

Route Maintenance - RERR

- RERR is initiated by the node upstream (closer to the source) of the break
  - It is propagated to all the affected destinations
  - RERR lists all the nodes affected by the link failure -> Nodes that were using the link to route messages (precursor nodes)
  - When a node receives an RERR, it marks its route to the destination as invalid -> Setting distance to the destination as infinity in the route table
- When a source node receives an RERR, it can reinitiate the route discovery

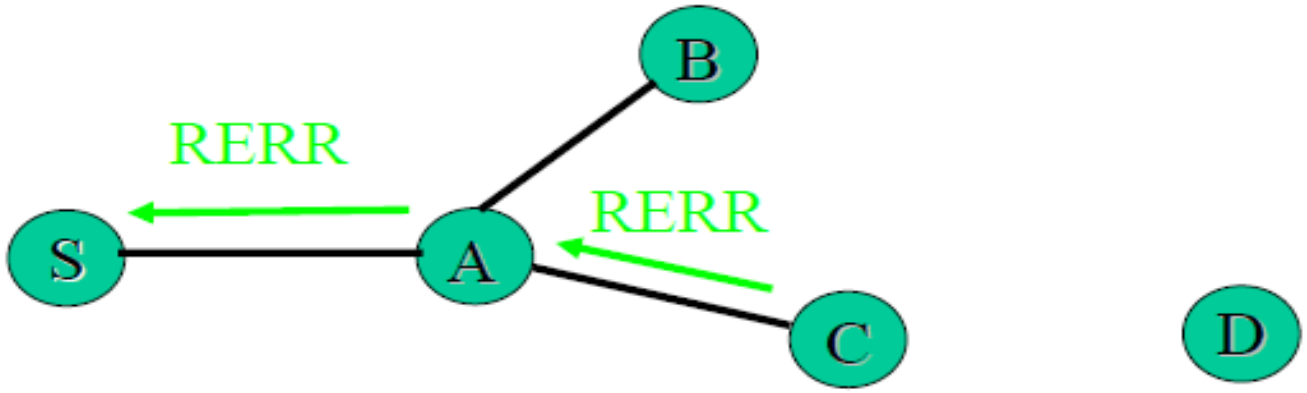
AODV – Route Maintenance- Example

1. Link between C and D breaks
2. Node C invalidates route to D in route table
3. Node C creates Route Error message
    1. Lists all destinations that are now unreachable
    2. Sends to upstream neighbors



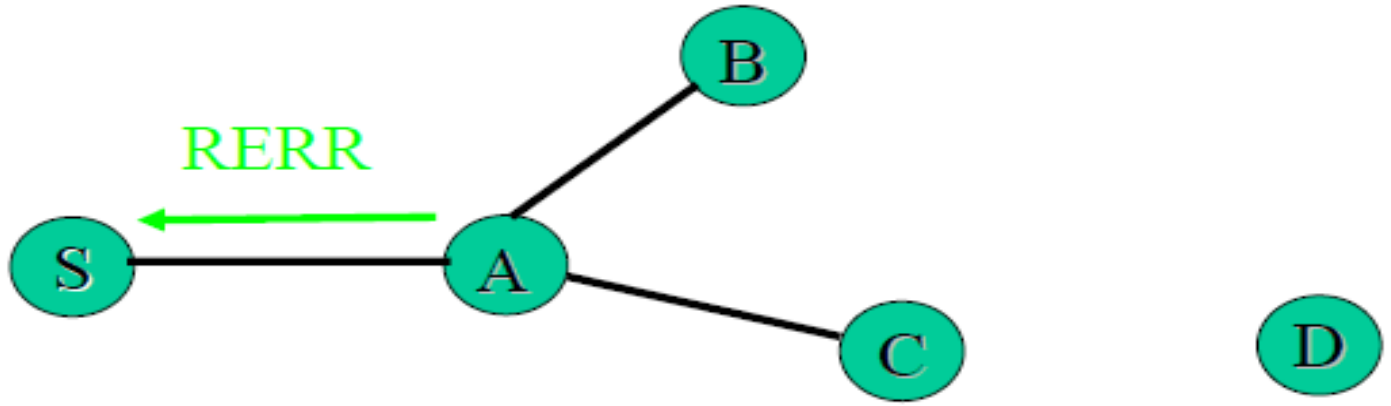
AODV – Route Maintenance- Example

- Node A receives RERR
  - Checks whether C is its next hop on route to D
  - Deletes route to D (makes distance -> infinity)
  - Forwards RERR to S



AODV – Route Maintenance- Example

- Node S receives RERR
  - Checks whether A is its next hop on route to D
  - Deletes route to D
  - Rediscovers route if still needed



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

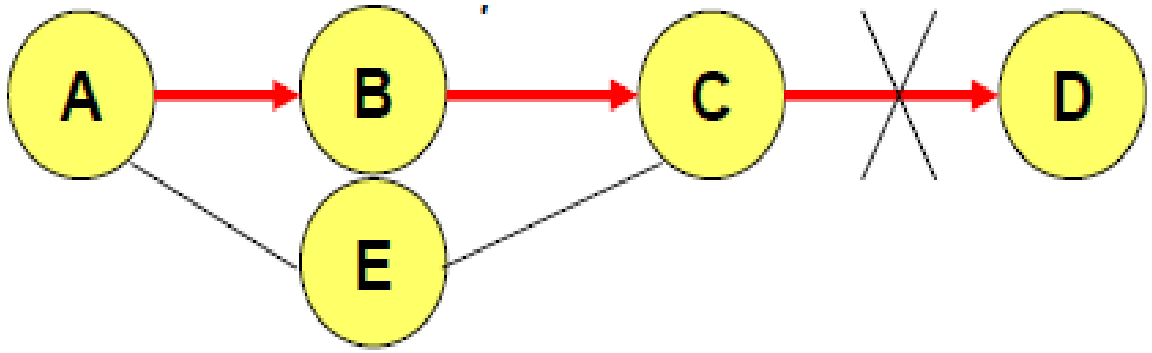
- A neighbor of node X is considered active for a routing table entry if the neighbor sent a packet within *active_route_timeout* interval which was forwarded using that entry

- 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 acknowledgements may be used as an indication of link failure

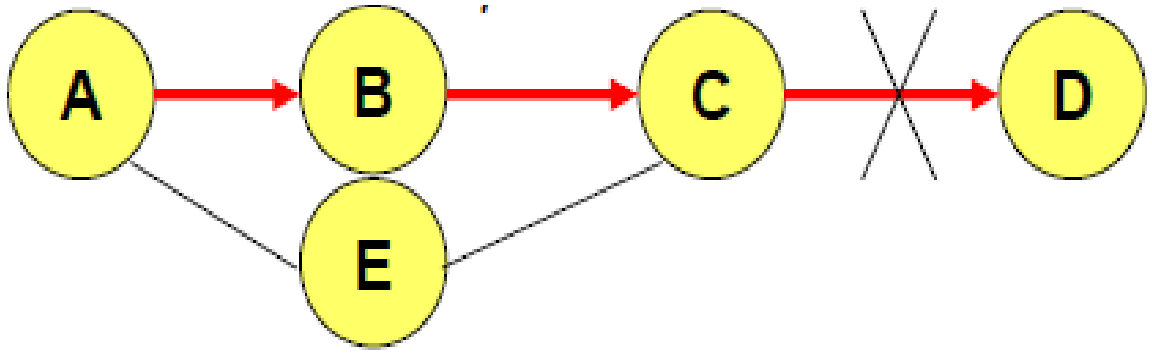
Why Sequence Numbers in AODV

- To avoid using old/broken routes
  - To determine which route is newer
- To prevent formation of loops
  - A had a route to D initially
  - 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)
  - Node A will reply since A knows a route to D via node B
  - Results in a loop (for instance, C-E-A-B-C)



Why Sequence Numbers in AODV

- But because of usage of sequence number, A will not use the route A-B-C, because the sequence numbers will be lower than what A receives from C



AODV: Summary

- 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

- Sequence numbers are used to avoid old/broken routes
- Sequence numbers prevent formation of routing loops

- Unused routes expire even if topology does not change

Location-Aided Routing (LAR) [Ko98Mobicom]

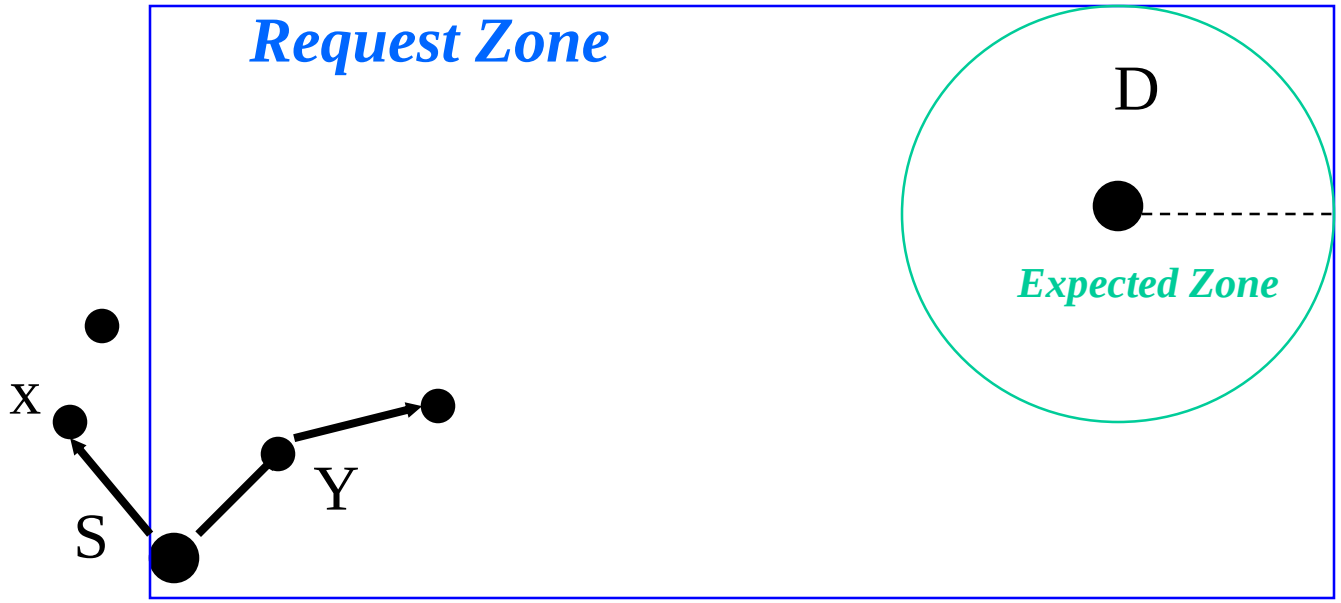
- Exploits location information to limit scope of route request flood
  - Location information may be obtained using GPS

- *Expected Zone* is determined as a region that is expected to hold the current location of the destination
  - Expected region determined based on potentially old location information, and knowledge of the destination's speed

- Route requests limited to a *Request Zone* that contains the Expected Zone and location of the sender node

Request Zone

- Define a Request Zone
- LAR is same as flooding, except that only nodes in request zone forward route request
- Smallest rectangle including S and expected zone for D



Location Aided Routing (LAR)

- Advantages
  - reduces the scope of route request flood
  - reduces overhead of route discovery
- Disadvantages
  - Nodes need to know their physical locations
  - Does not take into account possible existence of obstructions for radio transmissions

Proactive Routing Protocols

Destination-Sequenced Distance-Vector (DSDV) [Perkins94Sigcomm]

- Based on Bellman Ford algorithm
  - Exchange of routing tables
  - Routing table: the way to the destination, cost
- Every node knows “where” everybody else is
  - Thus routing table O(N)
- Each node advertises its position
  - Sequence number to avoid loops
  - Maintain fresh routes

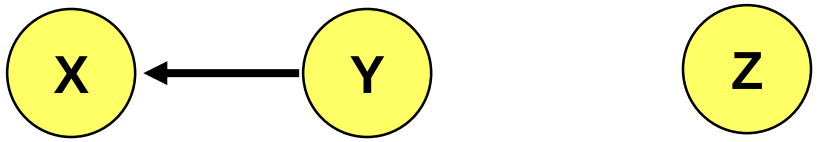
Destination-Sequenced Distance-Vector (DSDV) [Perkins94Sigcomm]

- Each node maintains a routing table which stores
  - next hop, cost metric towards each destination
  - a sequence number that is created by the destination itself
- Each node periodically forwards routing table to neighbors
  - Each node increments and appends its sequence number when sending its local routing table
- Each route is tagged with a sequence number; routes with greater sequence numbers are preferred

- Each node advertises a monotonically increasing even sequence number for itself
- When a node decides that a route is broken, it increments the sequence number of the route and advertises it with infinite metric
- Destination advertises new sequence number

Destination-Sequenced Distance-Vector (DSDV)

- When X receives information from Y about a route to Z
  - Let destination sequence number for Z at X be S(X), S(Y) is sent from Y



  - If S(X) > S(Y), then X ignores the routing information received from Y
  - If S(X) = S(Y), and cost of going through Y is smaller than the route known to X, then X sets Y as the next hop to Z
  - If S(X) < S(Y), then X sets Y as the next hop to Z, and S(X) is updated to equal S(Y)

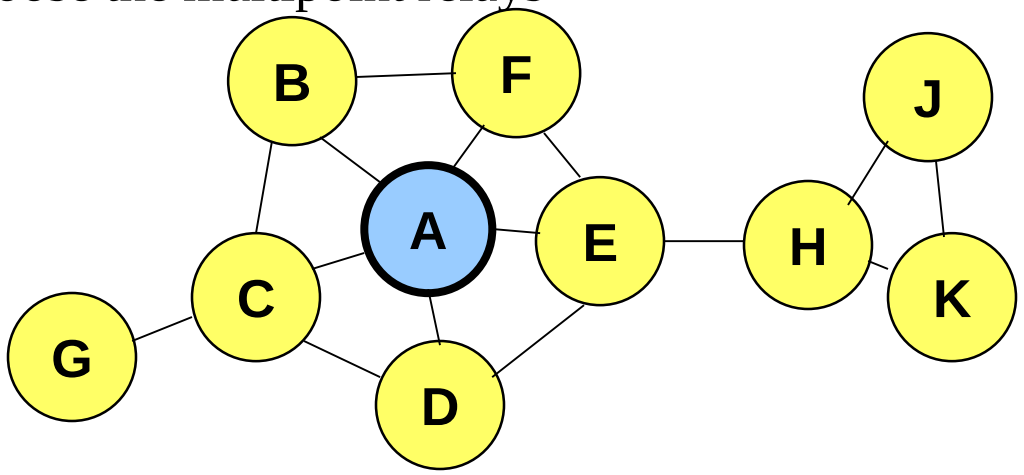
Destination-Sequenced Distance-Vector (DSDV)

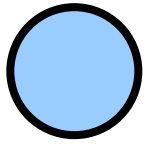
- Routes are broadcast from the “receiver”
  - Nodes announce their presence: advertisements
- Each broadcast has
  - Destination address: originator
  - No of hops
  - Sequence number of broadcast
- The route with the most recent sequence is used

Optimized Link State Routing (OLSR)

[Jacquet00ietf]

- Nodes C and E are multipoint relays of node A
  - Multipoint relays of A are its neighbors such that each two-hop neighbor of A is a one-hop neighbor of one multipoint relay of A
  - Nodes exchange neighbor lists to know their 2-hop neighbors and choose the multipoint relays

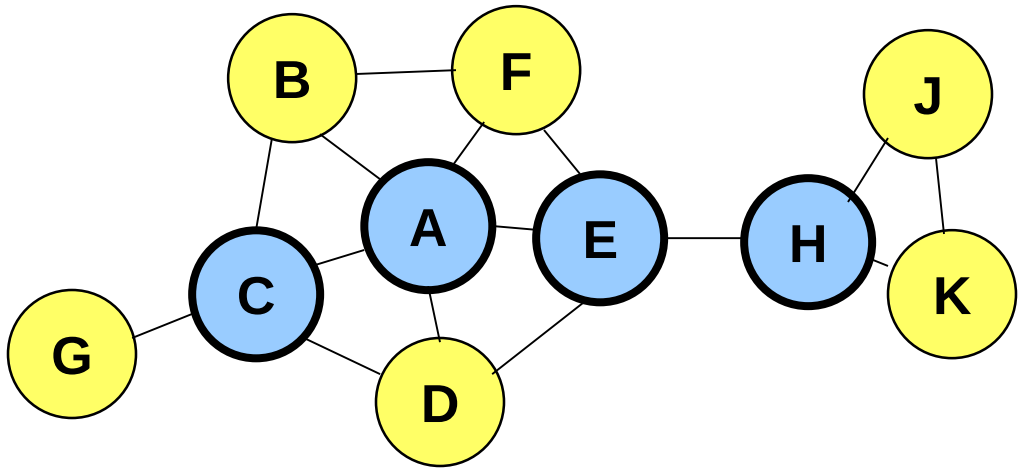




Node that has broadcast state information from A

Optimized Link State Routing (OLSR)

- Nodes C and E forward information received from A
- Nodes E and K are multipoint relays for node H
- Node K forwards information received from H



Node that has broadcast state information from A

Hybrid Routing Protocols

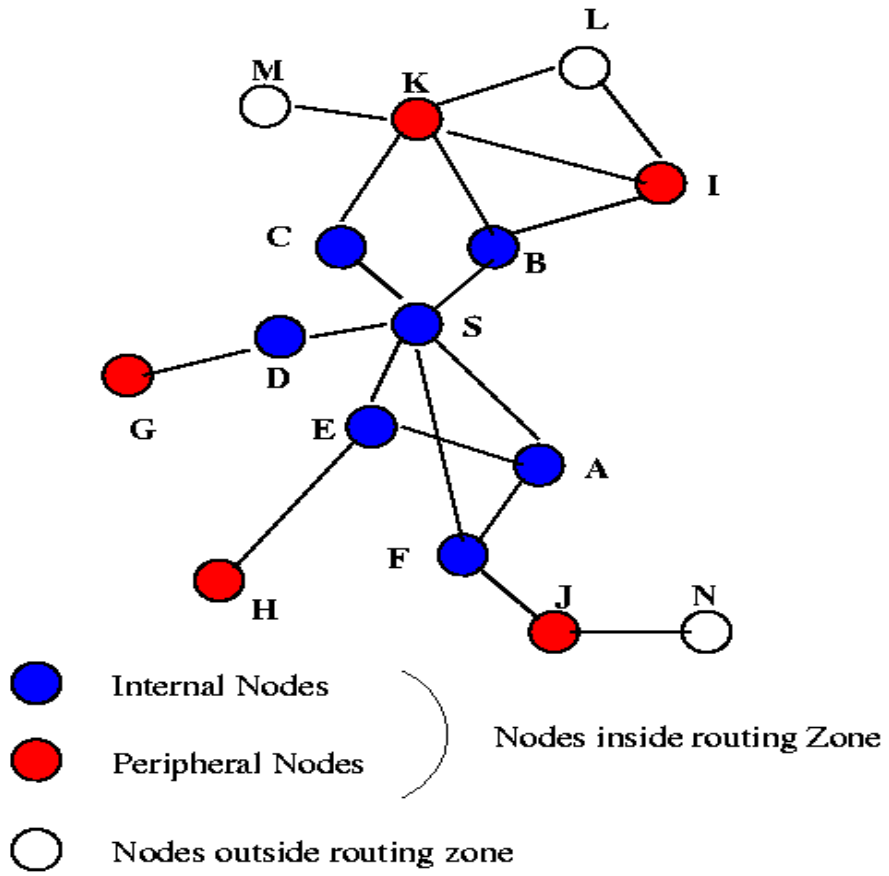
Zone Routing Protocol (ZRP) [Haas98]

- ZRP combines proactive and reactive approaches

- All nodes within hop distance at most *d* from a node X are said to be in the routing zone of node X
- All nodes at hop distance exactly *d* are said to be peripheral nodes of node X's routing zone

- Intra-zone routing: Proactively maintain routes to all nodes within the source node's own zone.
- Inter-zone routing: Use an on-demand protocol (similar to DSR or AODV) to determine routes to outside zone.

Zone Routing Protocol (ZRP)



Radius of routing zone = 2