Unicast Routing Protocols for Ad Hoc Networks

Introduction

* Characteristics
* No fixed infrastructure (a set of nodes)
  - » Multiple Hops to reach destinations
  - » Route changes because of node movements

* Radio used for communication
  - » Variable transmission range
  - » Broadcast nature of radio
  - » Interference , fading etc.

Introduction Cont'd

* Many Variations in mobility patterns
  - » Almost fixed (sensors, actuators)
  - » Highly mobile (vehicles)
  - » Discrete movements
  - » Continuous movements

* Mobility Characteristics
  - » Speed
  - » Geographical location

Applications

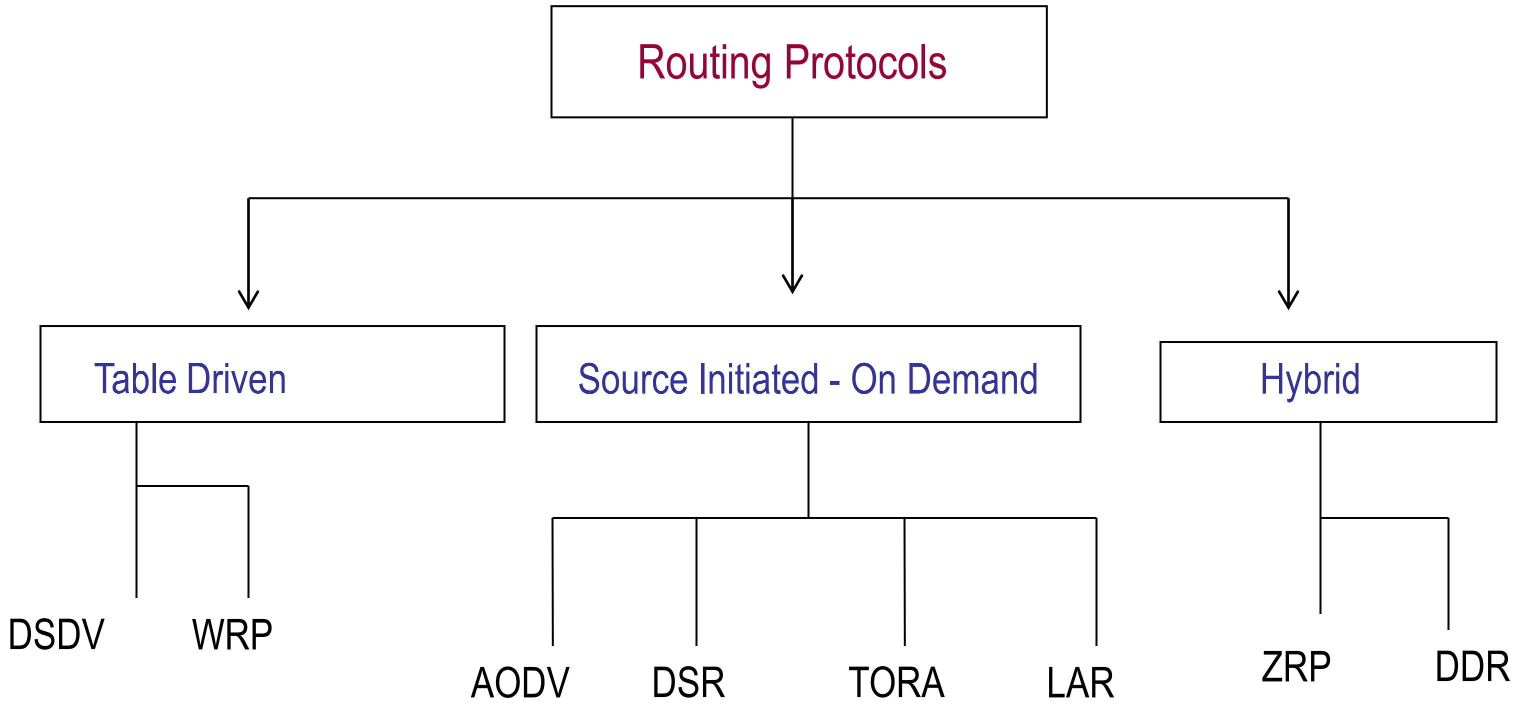
* Military Applications
  * » Battlefield, tanks, boats
* Personal Area network
  * » Communications between personal devices PDA's, Laptops, Watches, Play Stations
* SoHo
  * » Small office Home office
* Business Indoor Applications
  * » Exhibitions, Symposiums
  * » Demos, Meetings

Ad-Hoc Routing Requirements

* Distribution paths
  * » Multi hop paths
  * » Loop free
  * » Minimal transmission data overhead
* Self starting and adaptive to dynamic topology
* Low Consumption of Memory , BW, Power
  * » scalable with number of nodes
  * » localized effects of link failure

Unicast Routing

* Clasification



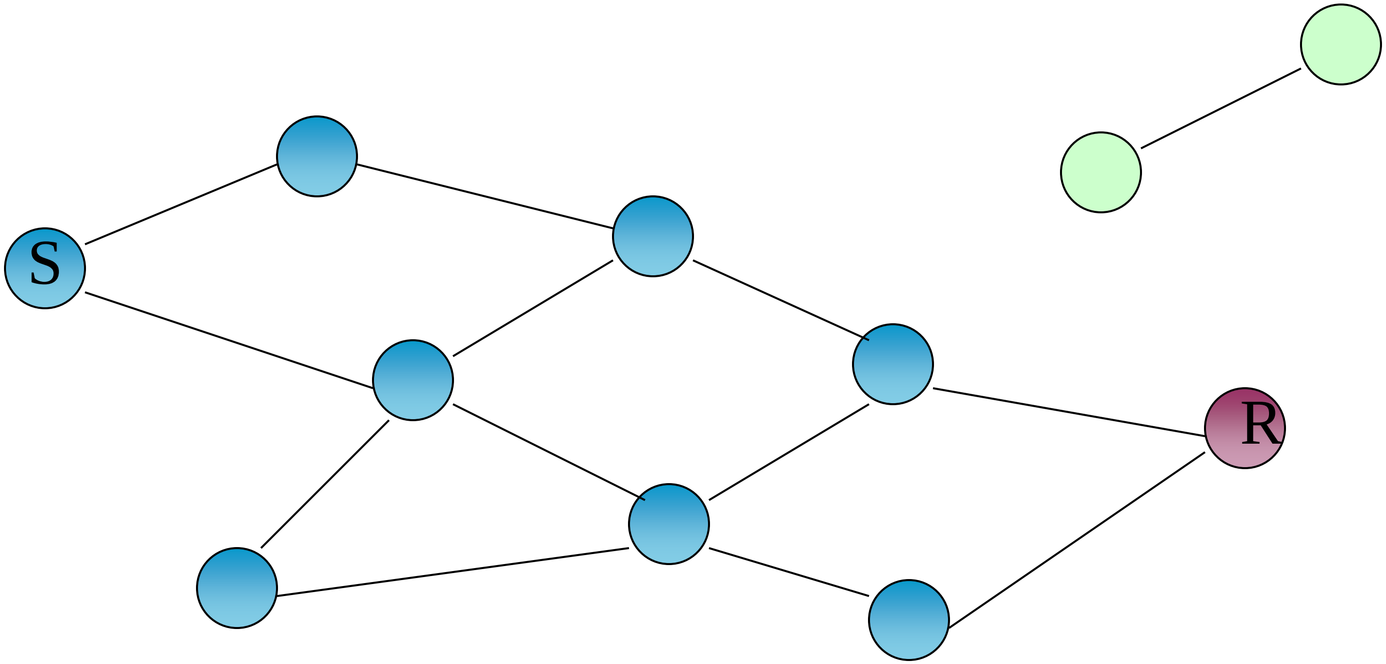
Problems using DV or LS

* DV protocols
  » may form loops wasteful in wireless environment: bandwidth and power
  » Loop avoidance may be complex

* LS protocols
  » Higher storage and communication overhead

Flooding ?

* Use Flooding for Data Delivery



* Flooding may deliver packets to too many nodes and in worst case all nodes reachable from sender may receive the packet

Alternatives To Flooding

* Flood only control packets
  - » Use flooding to set up the routes and use established routes for data
  - » Need to limit flooding as much as possible

Proactive Protocol

* Table Driven (Proactive Protocols)
  - » Maintain consistent, up-to-date routing information from each node to every other node in the network
  - » Each node has to maintain one or more tables to store the routing information
  - » Periodically propagate updates through out the network to account for link changes

DSDV

* Destination Sequenced Distance Vector
  - » Based on bellman-ford algorithm
  - » guarantees loop freedom
* Each node maintains a routing table
  - » Next Hop
  - » Cost metrics
  - » Destination Sequence number
  - » Each node periodically sends its local routing table with an incremented sequence number

On Demand Protocols

* AODV
  - » Primary Objectives
    - Provide, unicast, broadcast and multicast capability
    - Minimize broadcast of control packets
    - Disseminate information about link breakages to neighboring nodes using the link
  - » Characteristics
    - On Demand route creation
    - Two Dimensional routing metric

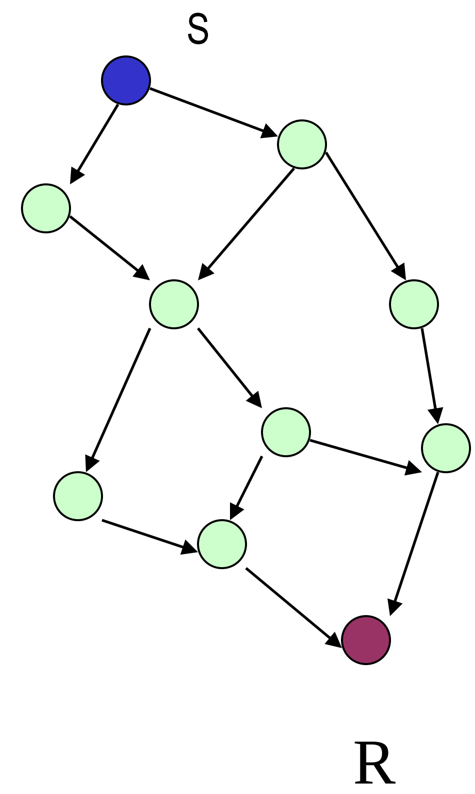
Unicast Route Discovery

* Source broadcasts *Route Request* (RREQ)

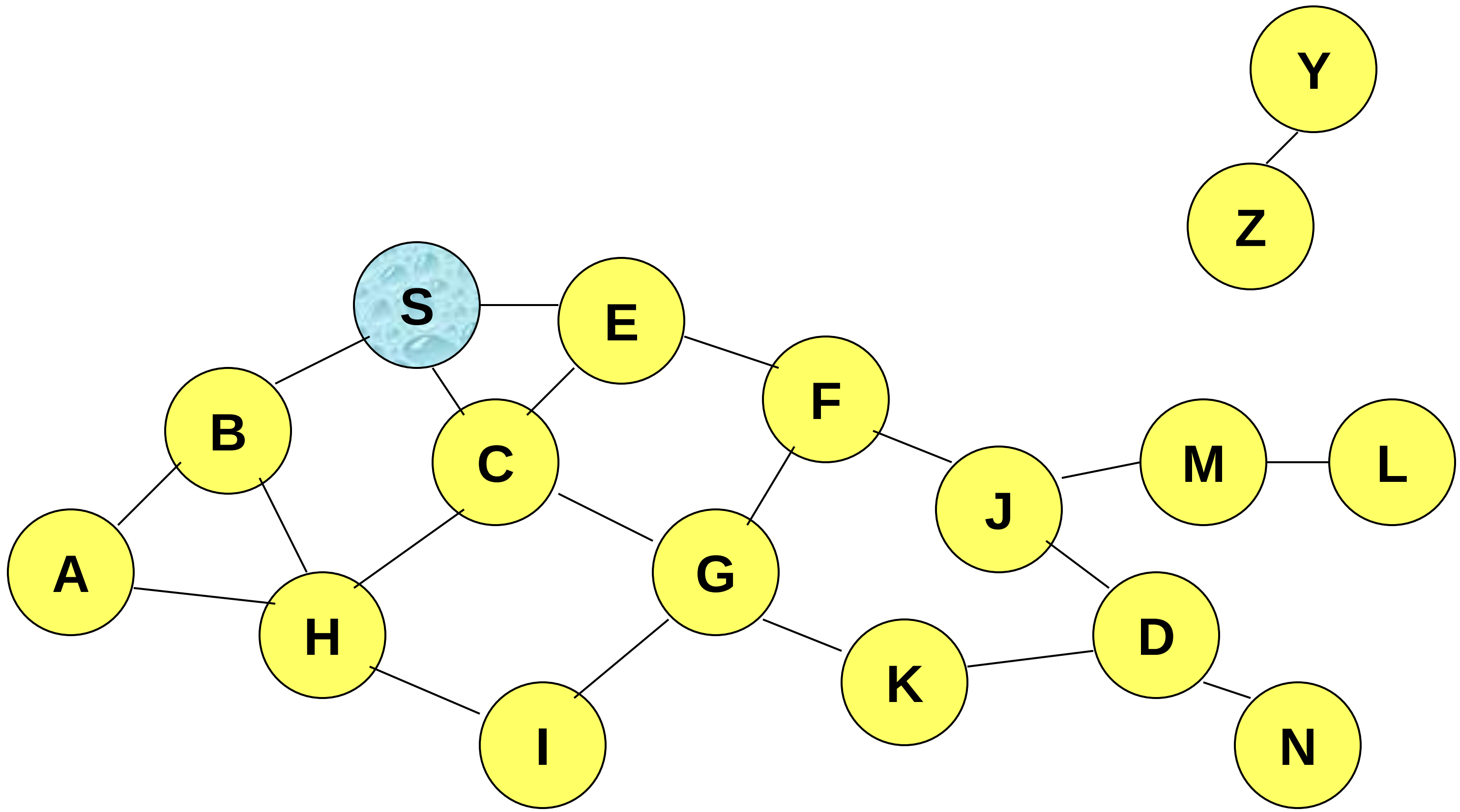
  <J_flag, Bcast_Id, Src_Addr, Src_Seq#, Dst_Addr, Dest_Seq#, Hop_Cnt>

* Node can reply to RREQ if
  - » It is the destination
  - » it has a fresh enough route to the destination

* Node create Reverse Route Entry
* Records Src_Addr, Bcast_Id to prevent multiple procesing.

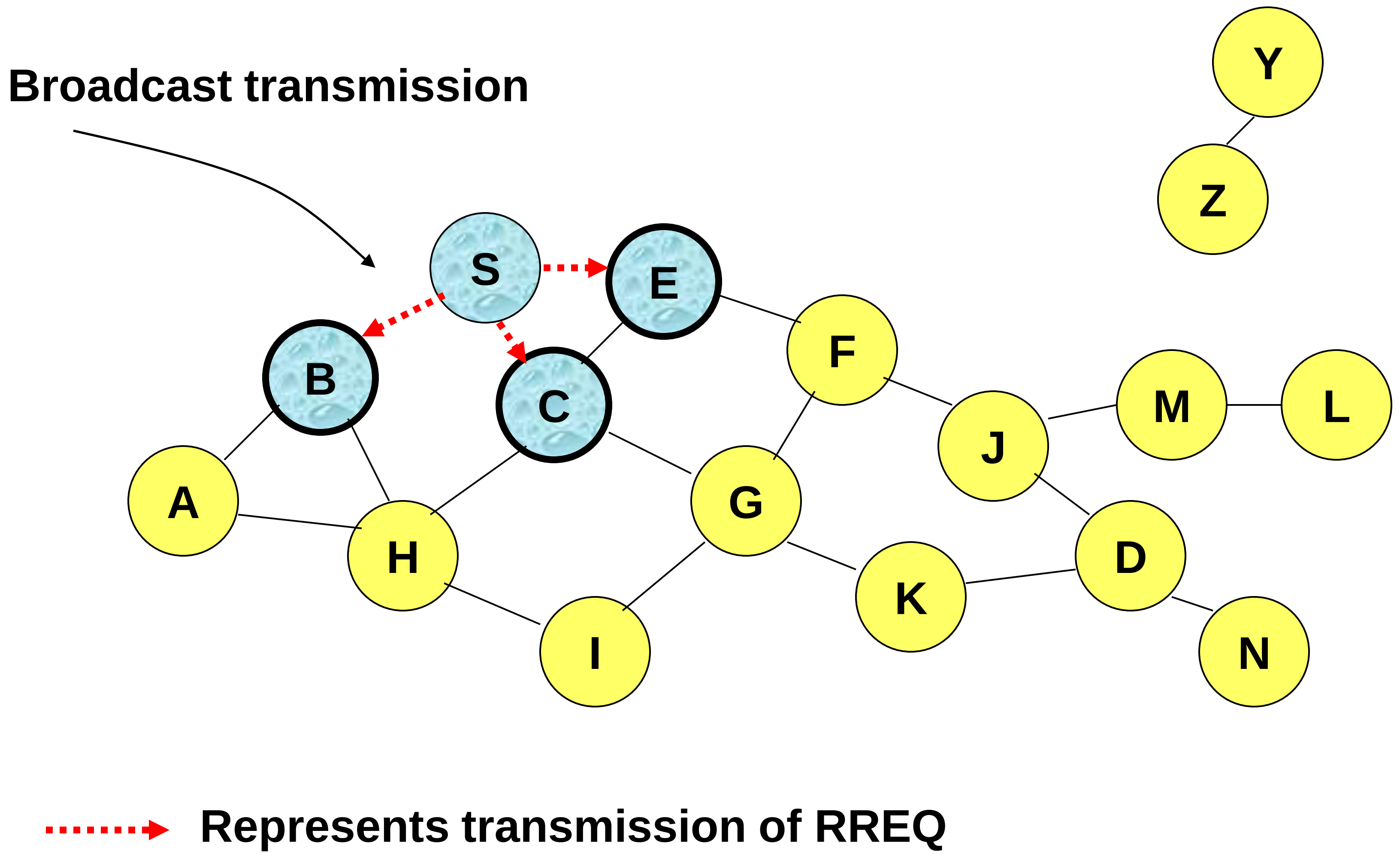


Route Requests in AODV

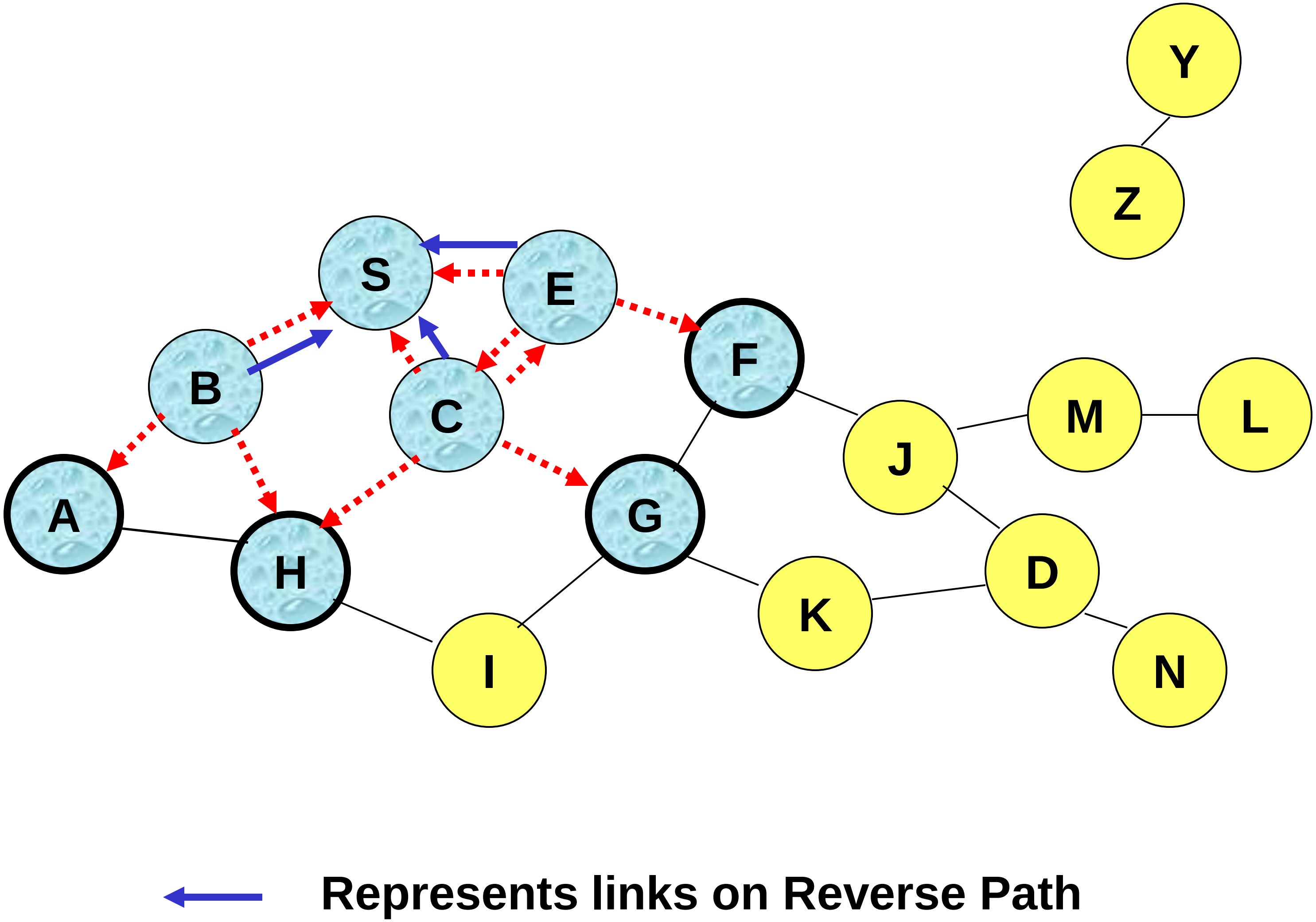


Represents a node that has received RREQ for D from S

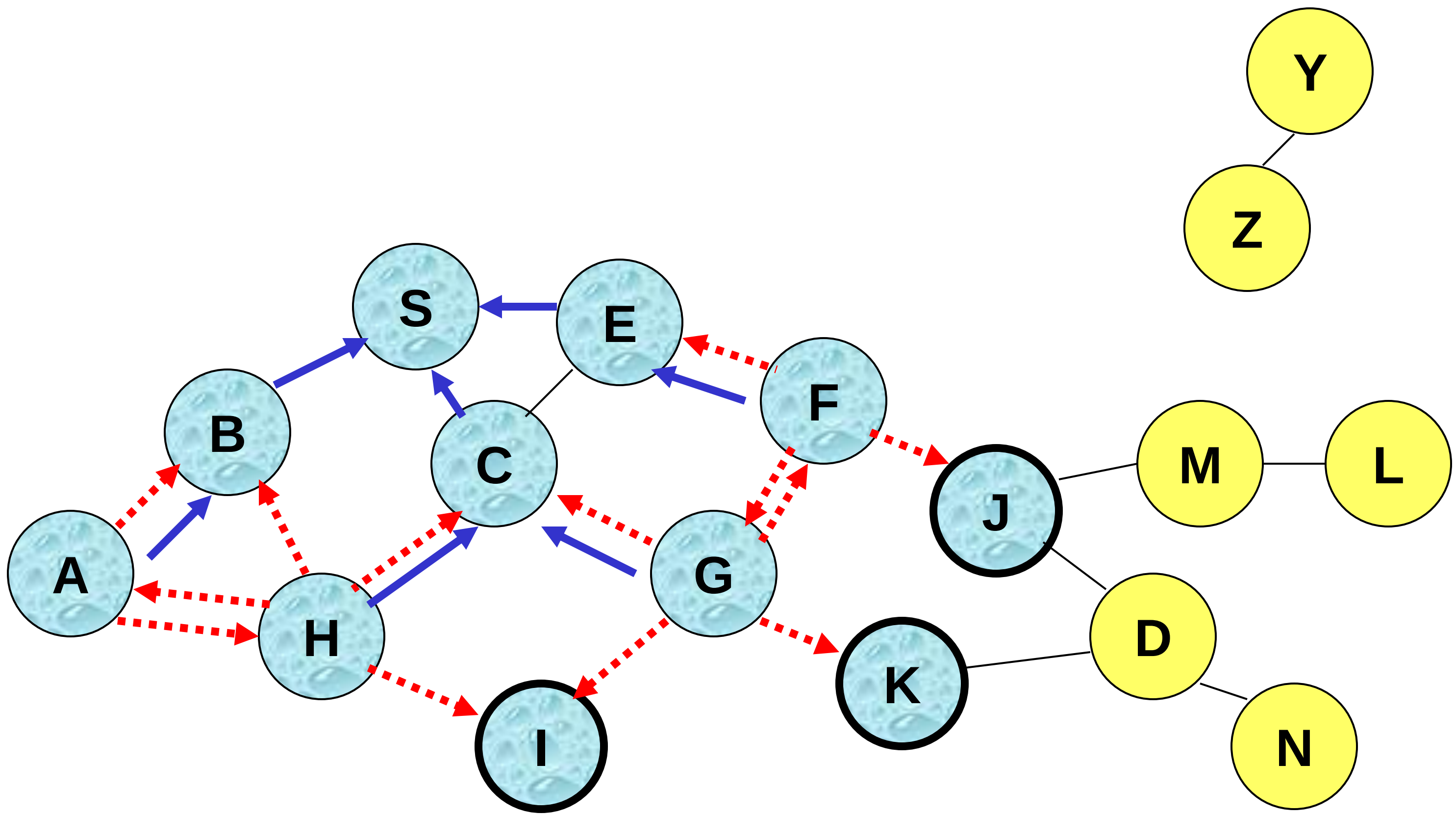
Route Requests in AODV



Route Requests in AODV

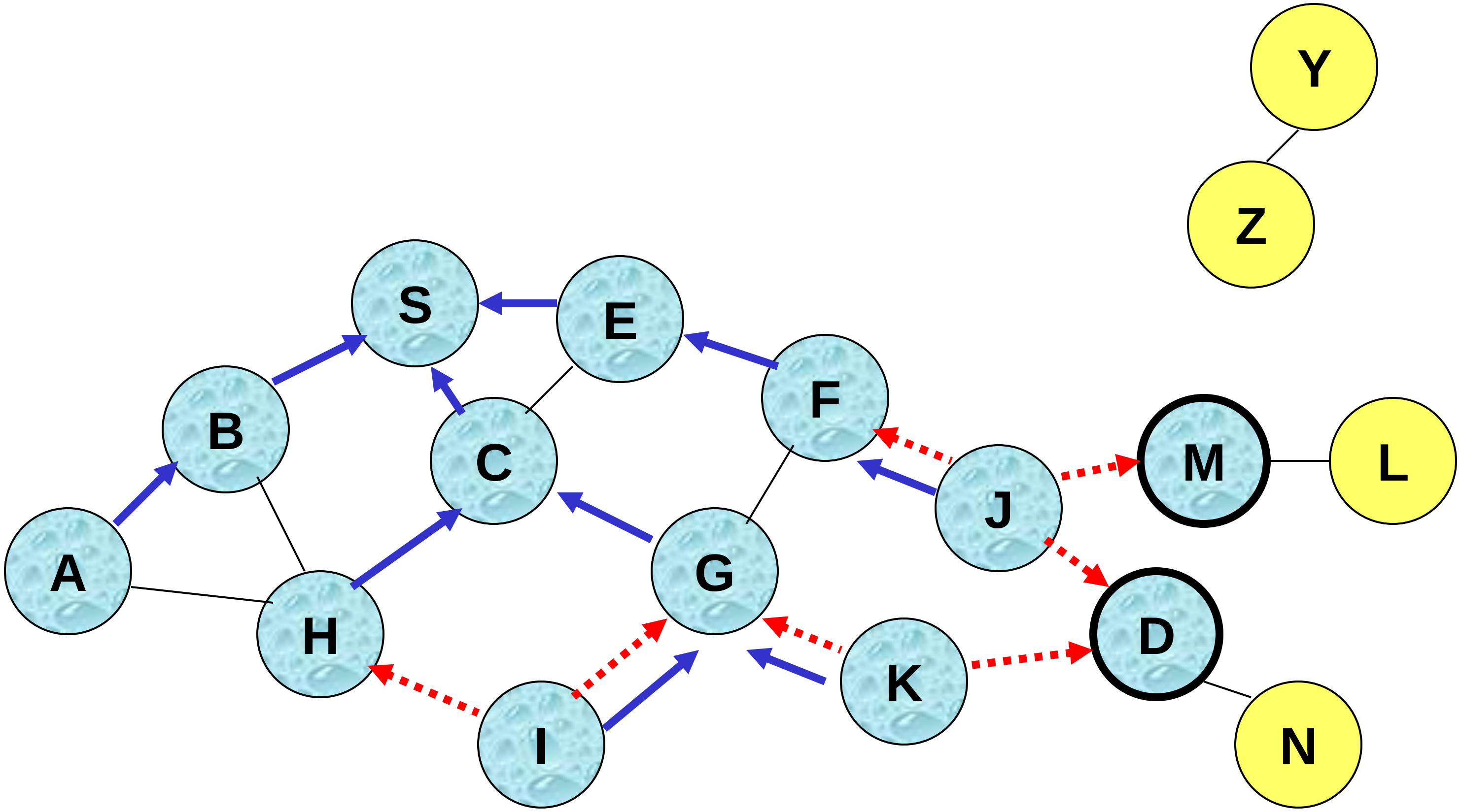


Reverse Path Setup in AODV

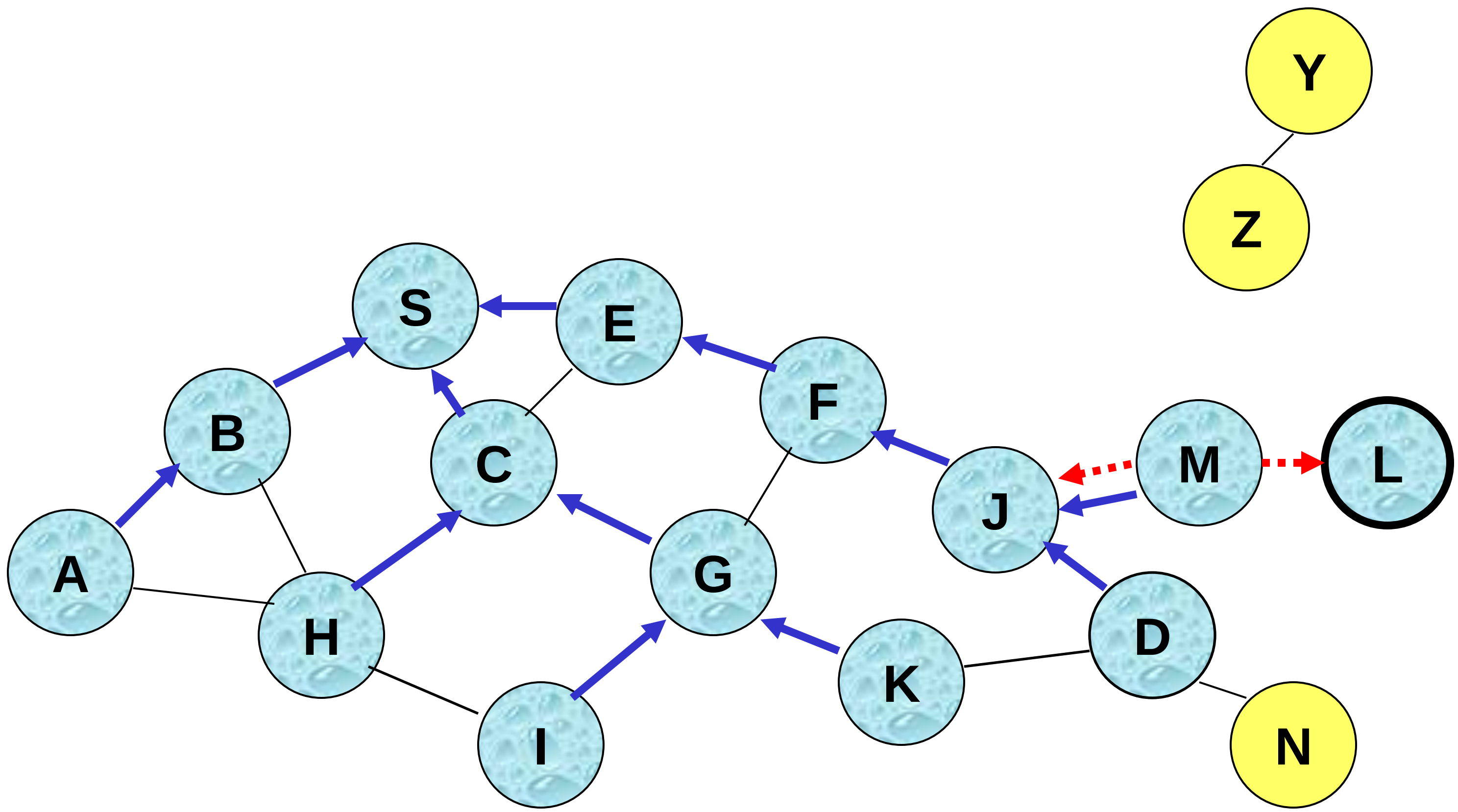


• Node C receives RREQ from G and H, but does not forward it again, because node C has already forwarded RREQ once

Reverse Path Setup in AODV

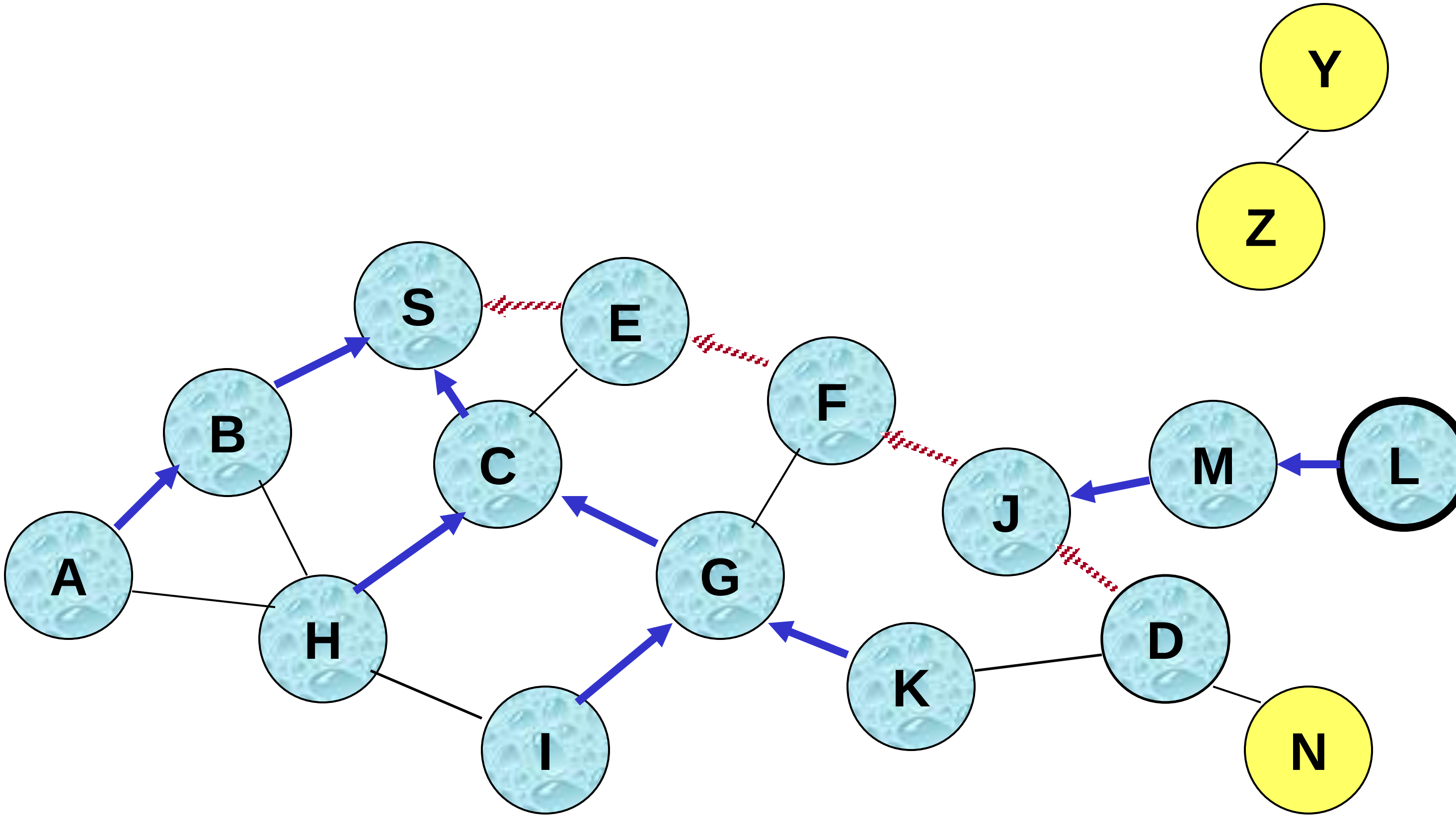


Reverse Path Setup in AODV



- 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

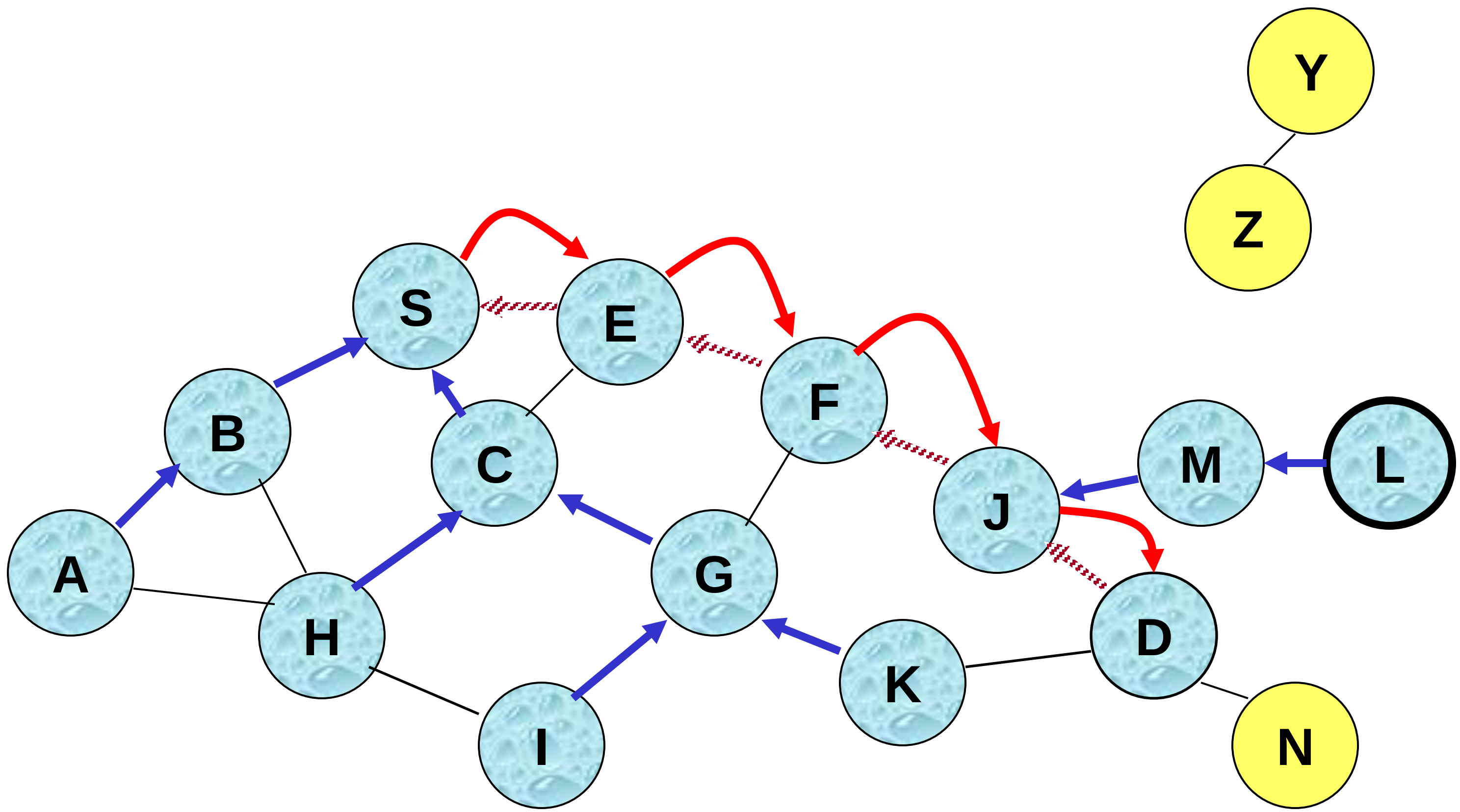
Route Reply in AODV

* 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

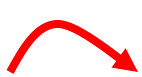
* To determine whether the path known to an intermediate node is more recent, *destination sequence numbers* are used

* 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

Forward Path Setup in AODV

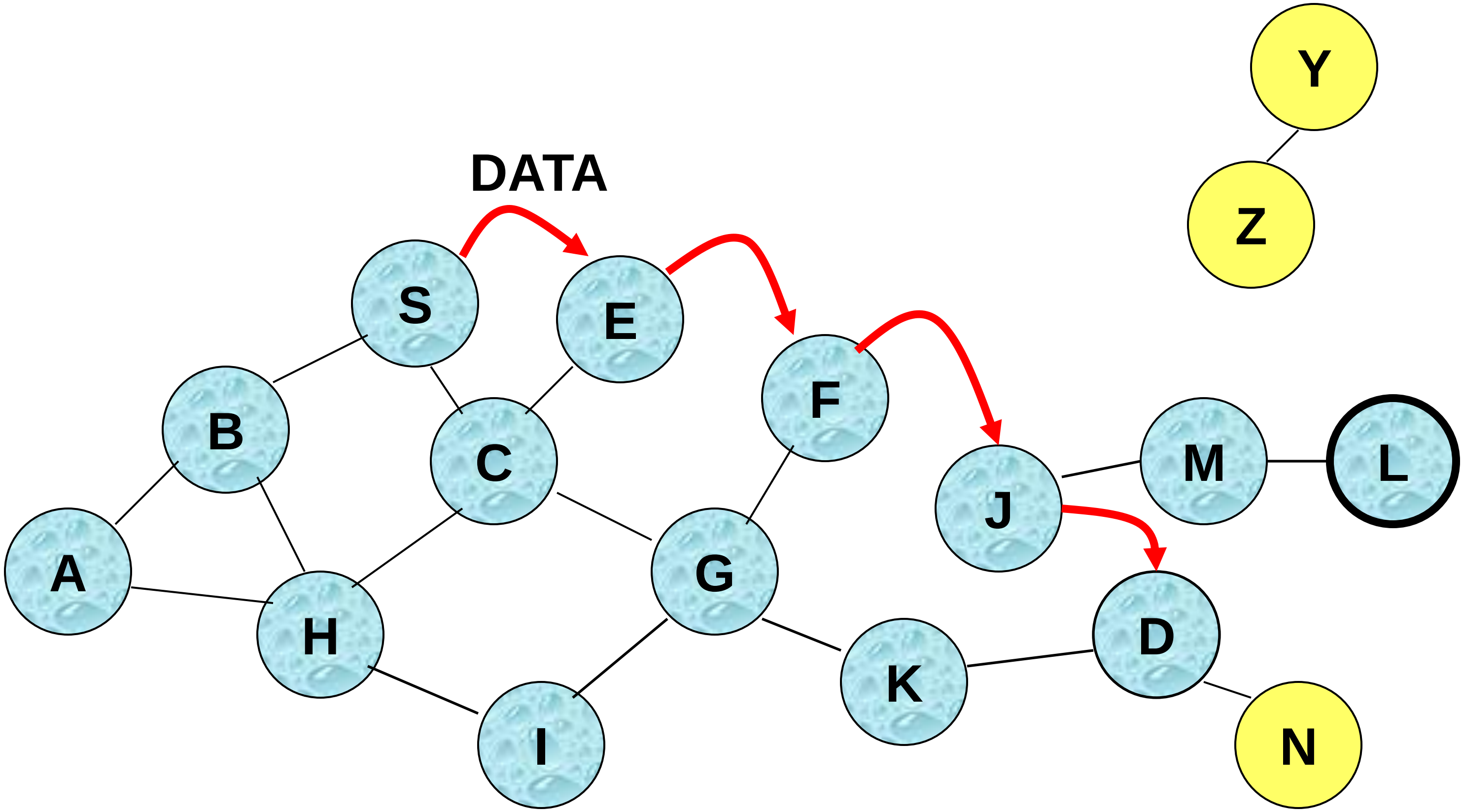


Forward links are setup when RREP travels along the reverse path



Represents a link on the forward path

Data Delivery in AODV



Routing table entries 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 a *active_route_timeout* interval
  » if no is data being sent using a particular routing table entry, that entry will be deleted from the routing table (even if the route may actually still be valid)

Link Failure 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

* 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*

Destination Sequence Number

* Continuing from the previous slide …
* 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

Optimization: Expanding Ring Search

* 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

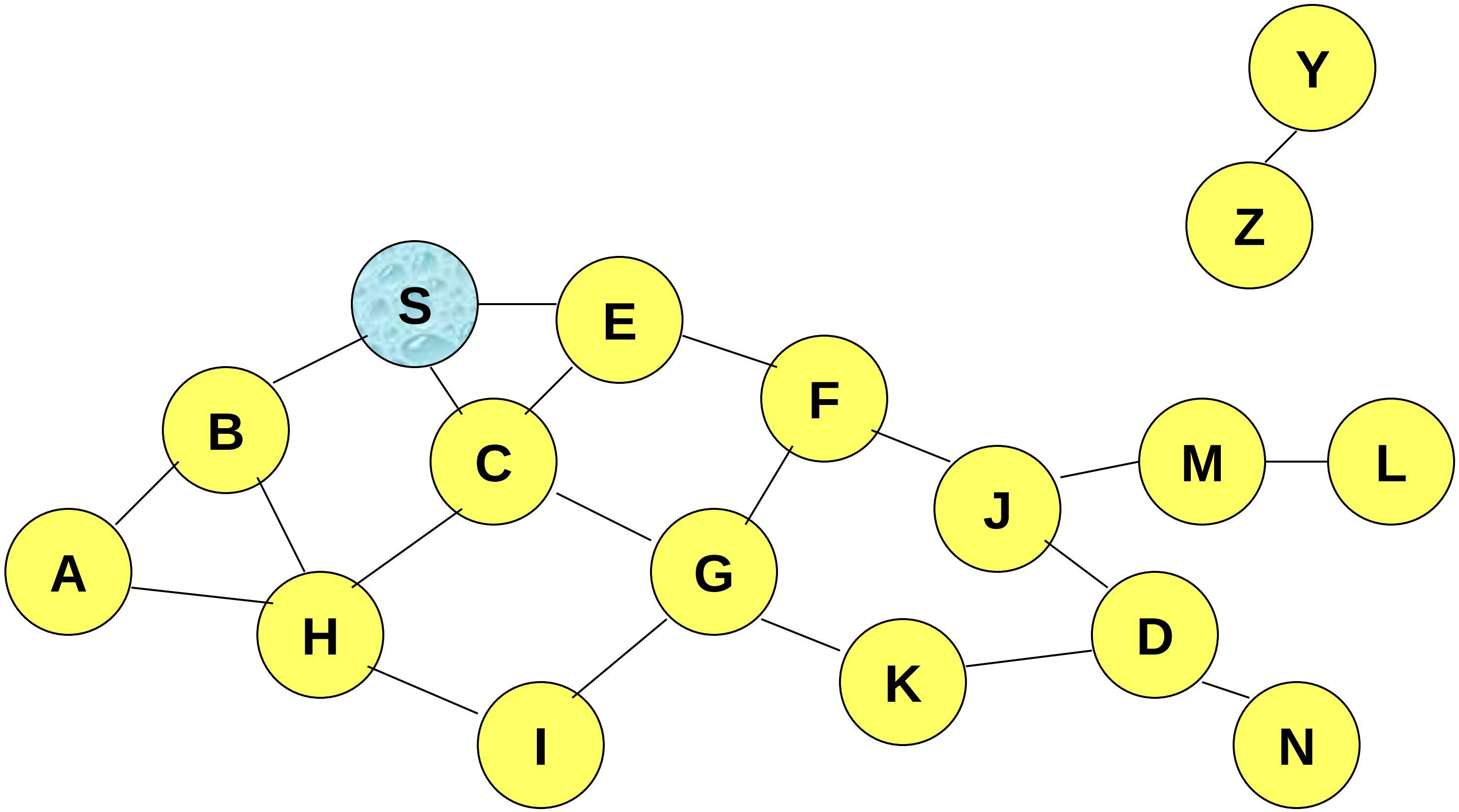
Dynamic Source Routing (DSR) [Johnson96]

* 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)

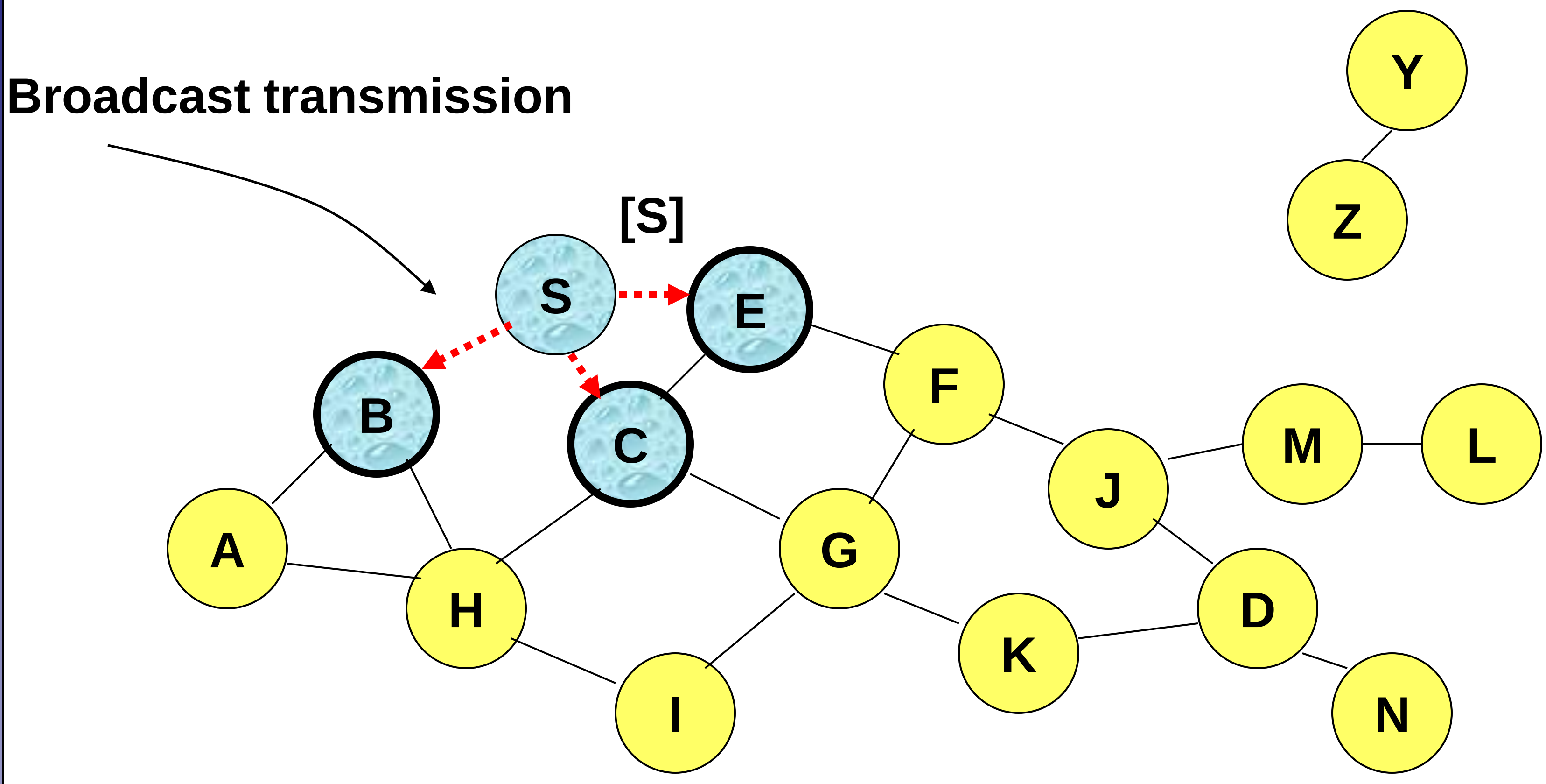
* Each node appends own identifier when forwarding RREQ

Route Discovery in DSR



Represents a node that has received RREQ for D from S

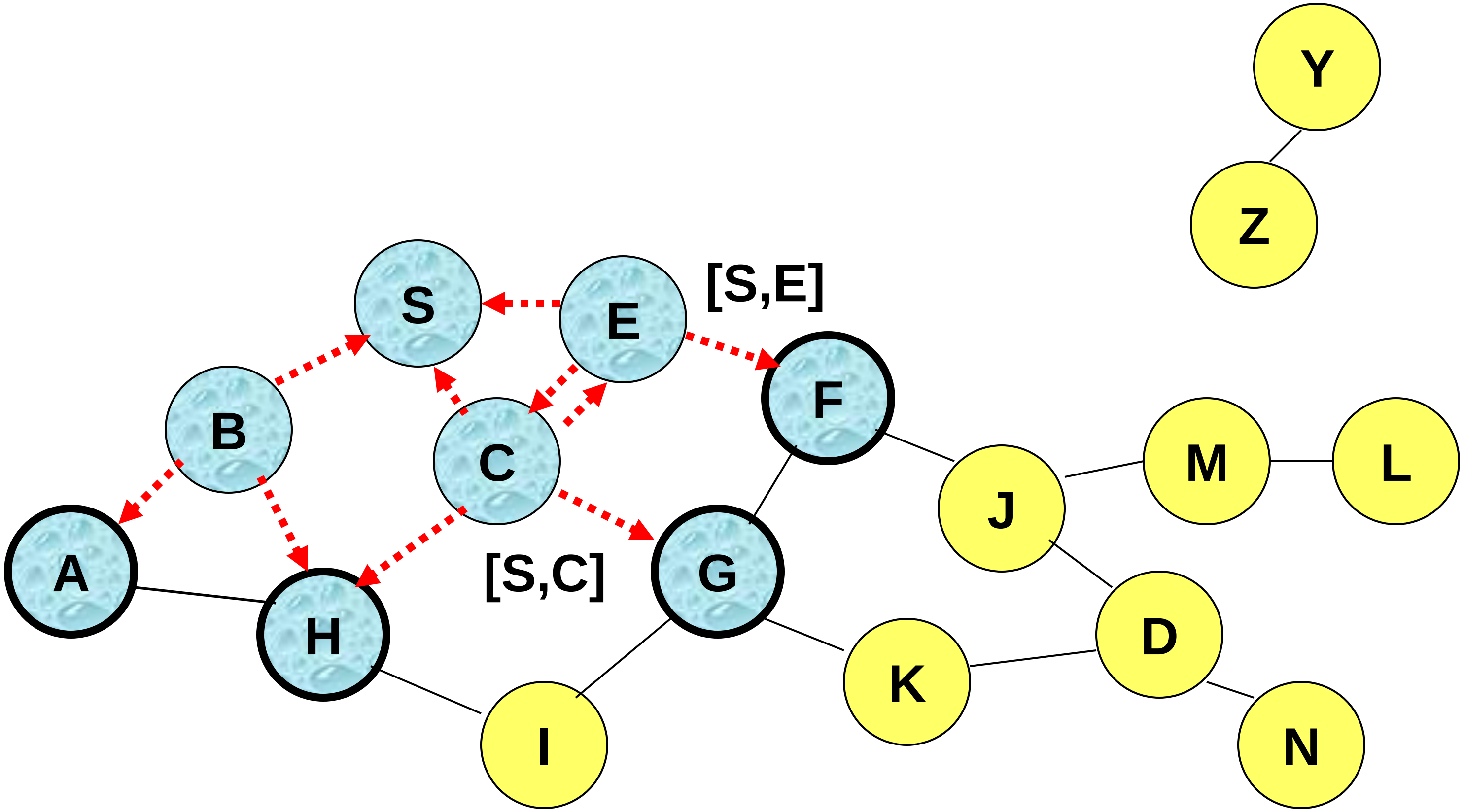
Route Discovery in DSR



Represents transmission of RREQ

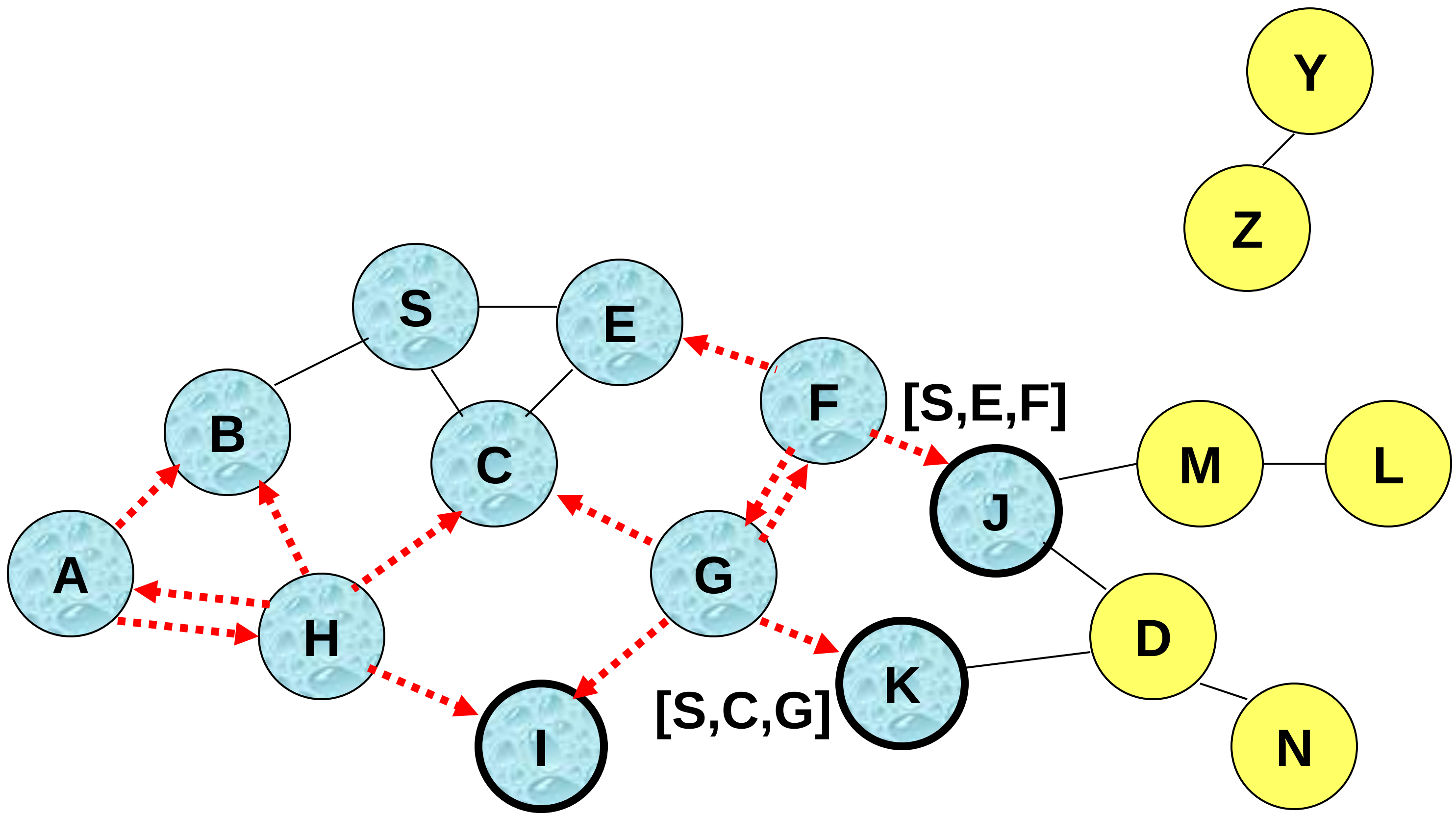
[X,Y] Represents list of identifiers appended to RREQ

Route Discovery in DSR



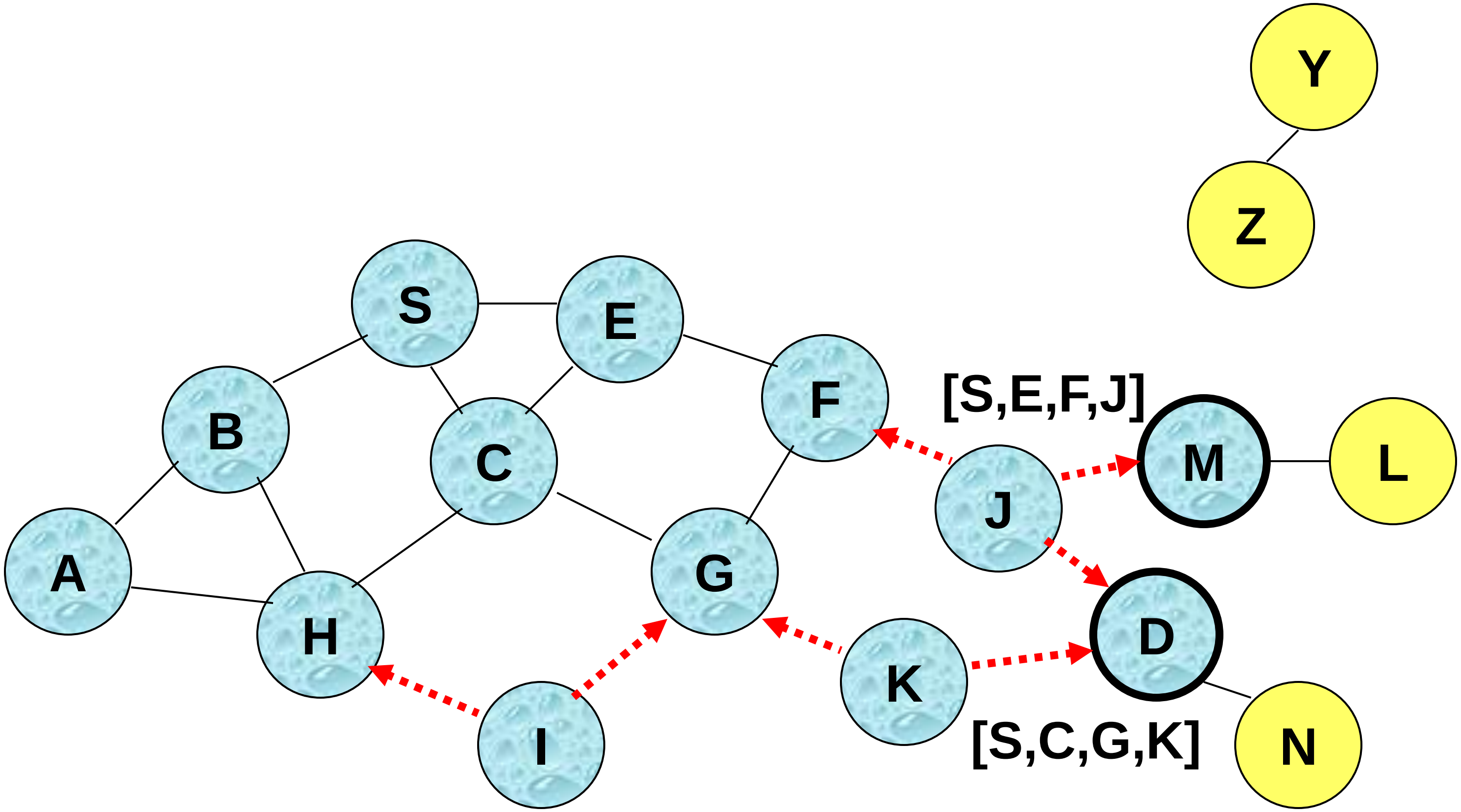
- Node H receives packet RREQ from two neighbors: potential for collision

Route Discovery in DSR



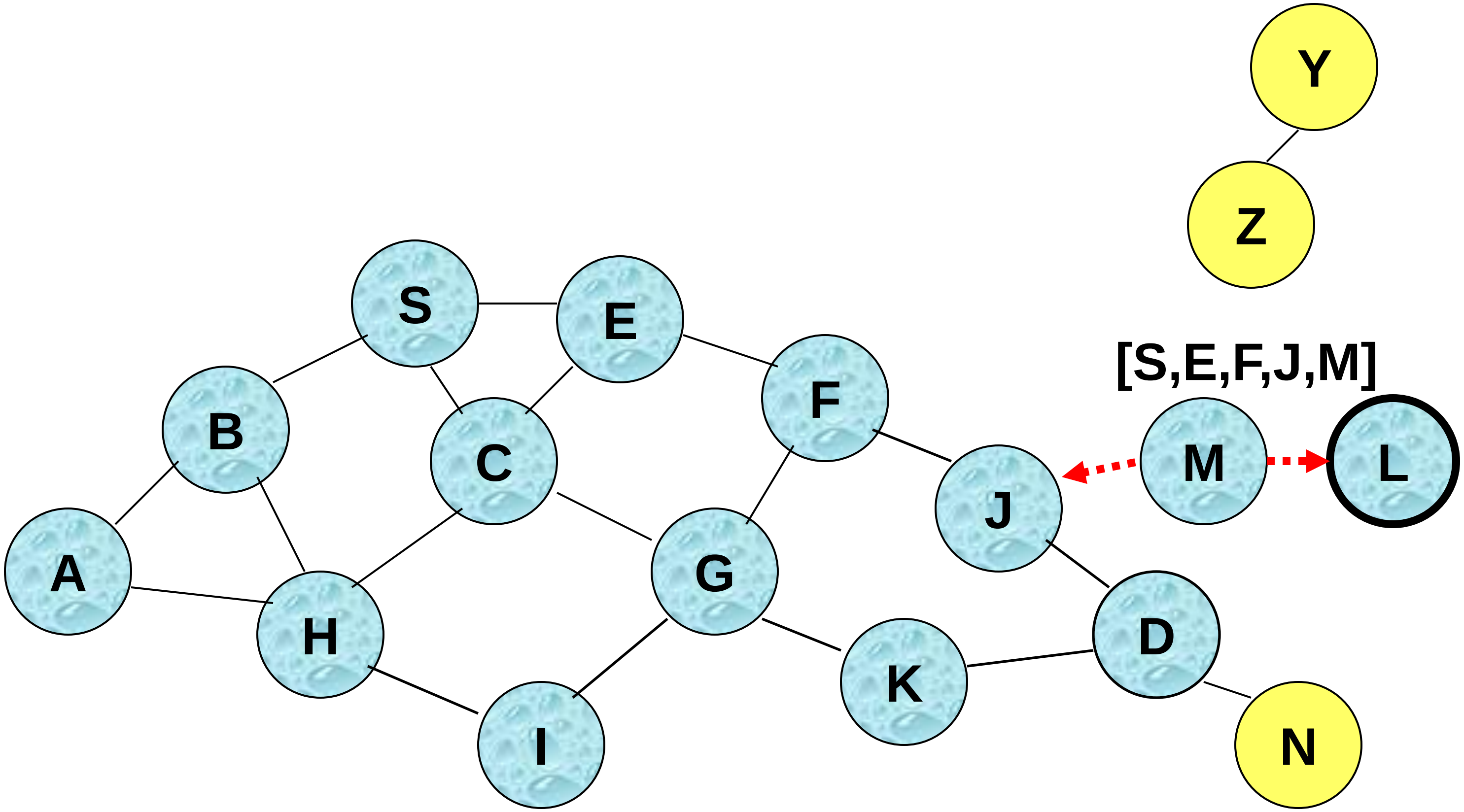
- Node C receives RREQ from G and H, but does not forward it again, because node C has already forwarded RREQ once

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

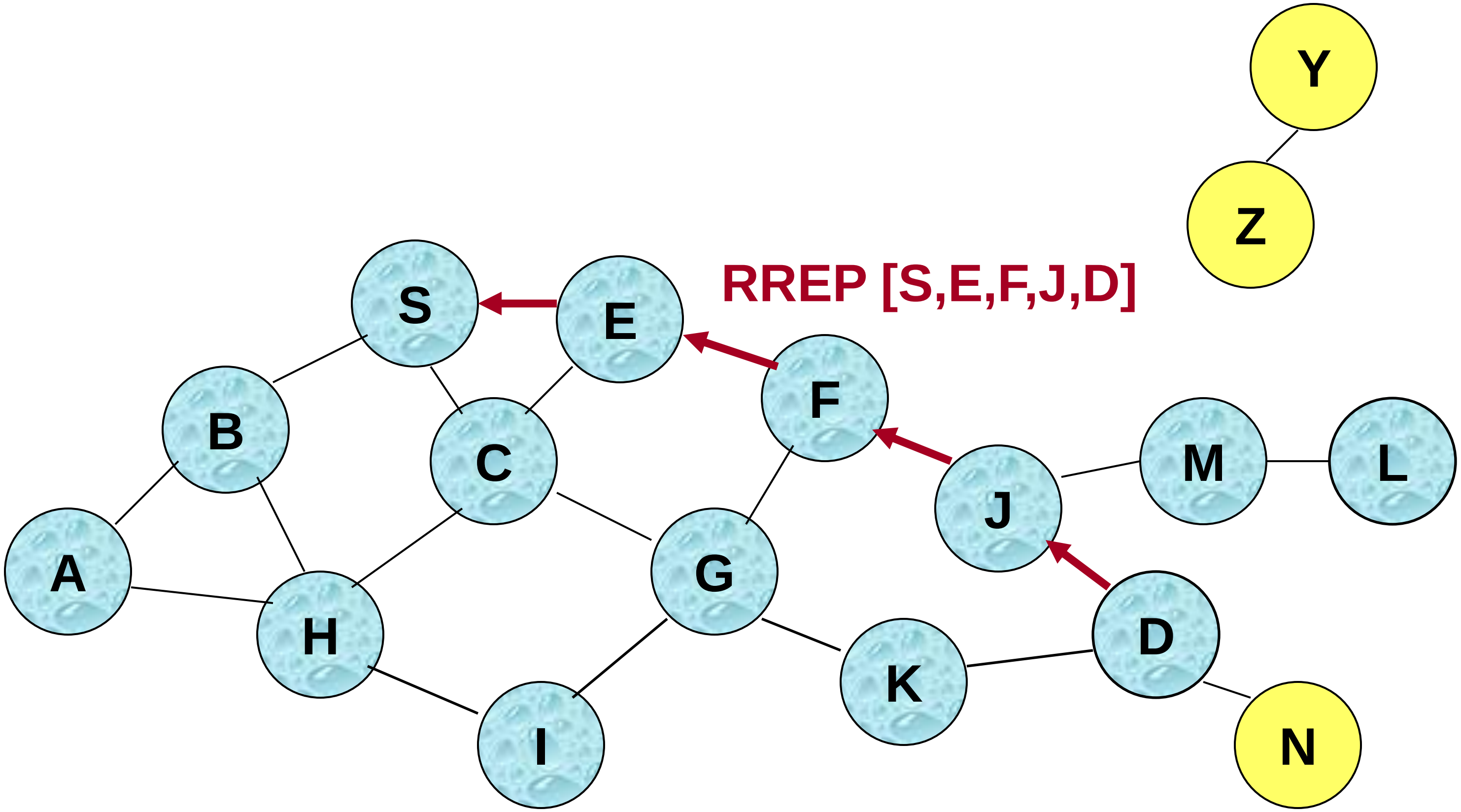
Route Discovery in DSR

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

Route Reply in DSR



Represents RREP control message

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
  » To ensure this, RREQ should be forwarded only if it received on a link that is known to be bi-directional

* If unidirectional (asymmetric) links are allowed, then RREP may need a route discovery for S from node D
  » Unless 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.

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

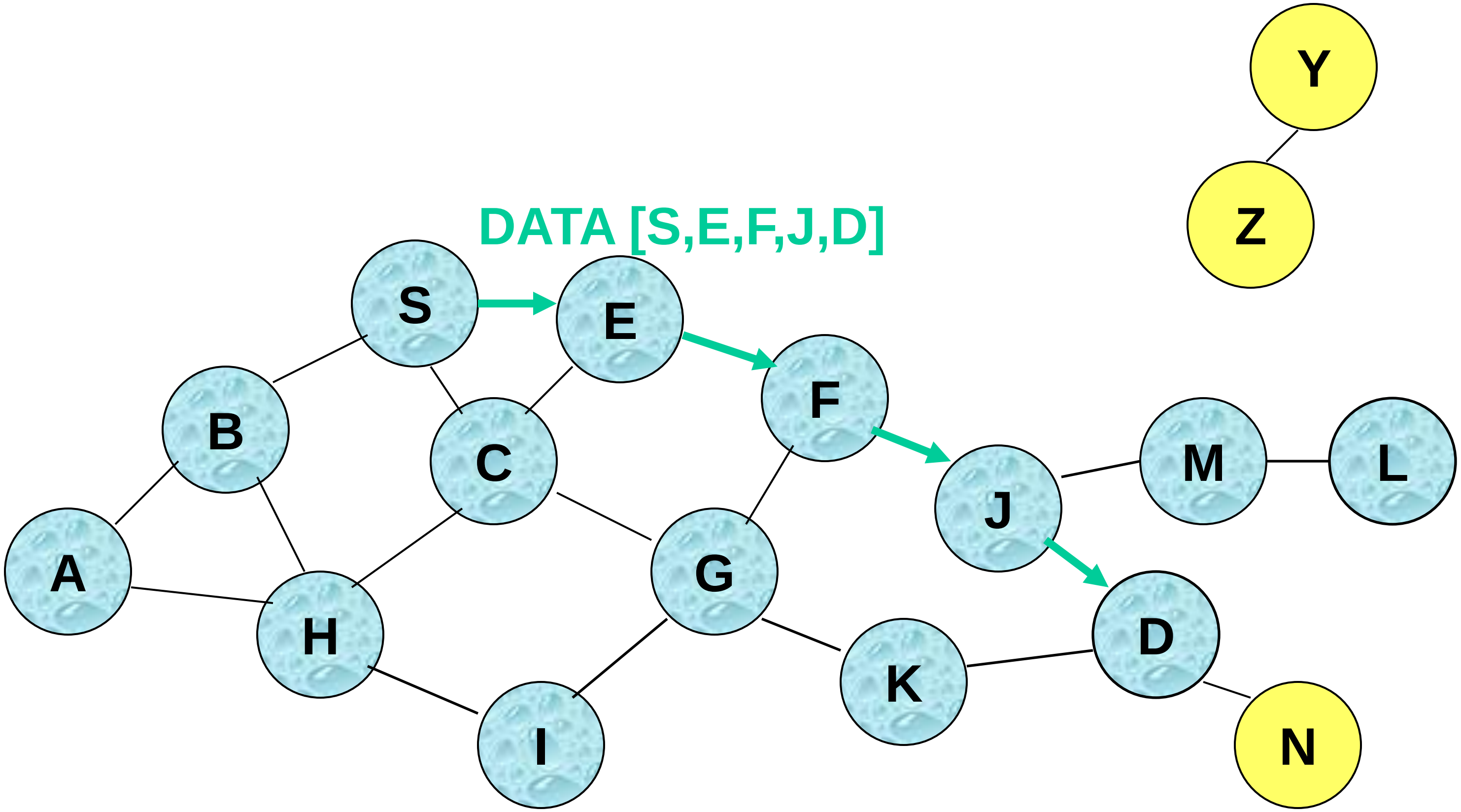
Dynamic Source Routing (DSR)

* 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

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 packets

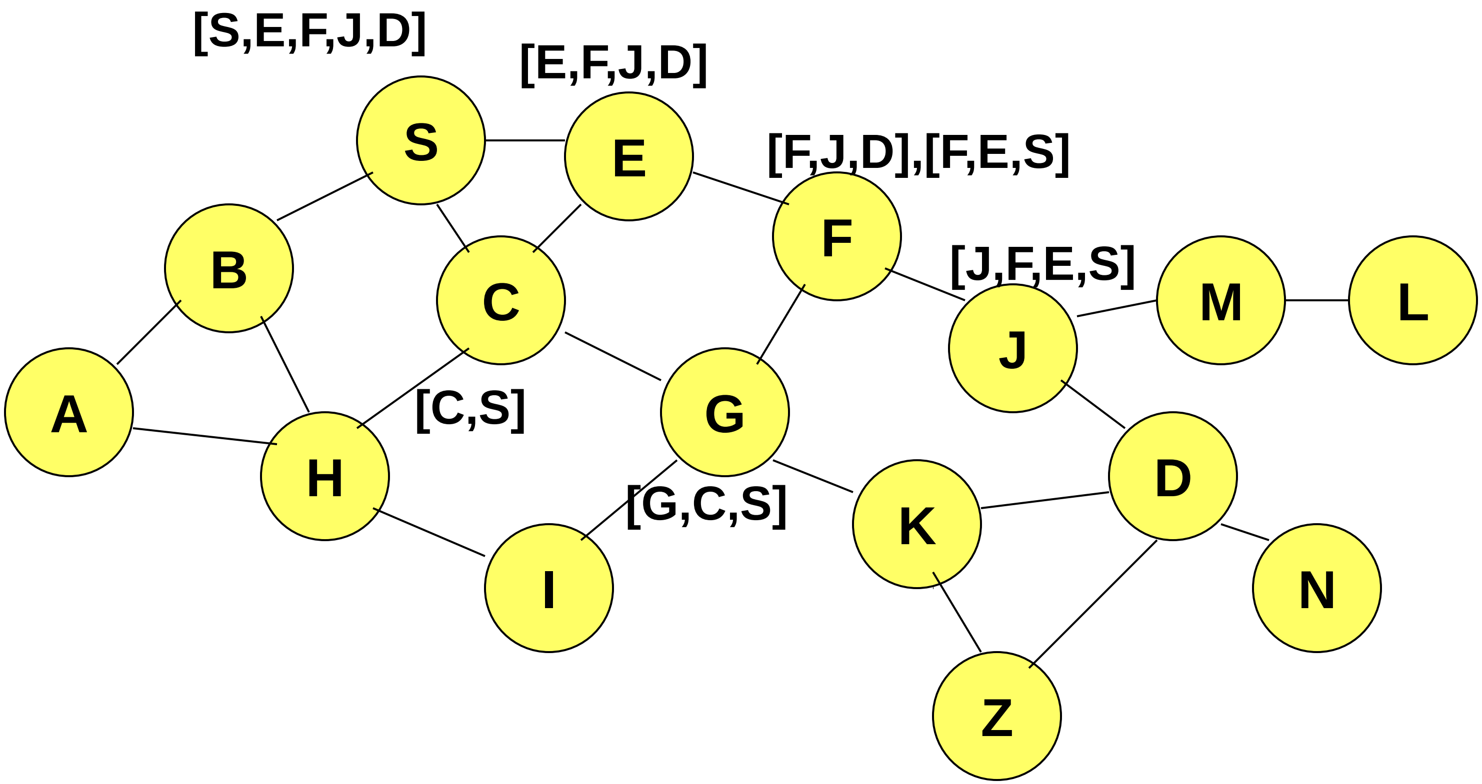
Use of Route Caching

* 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

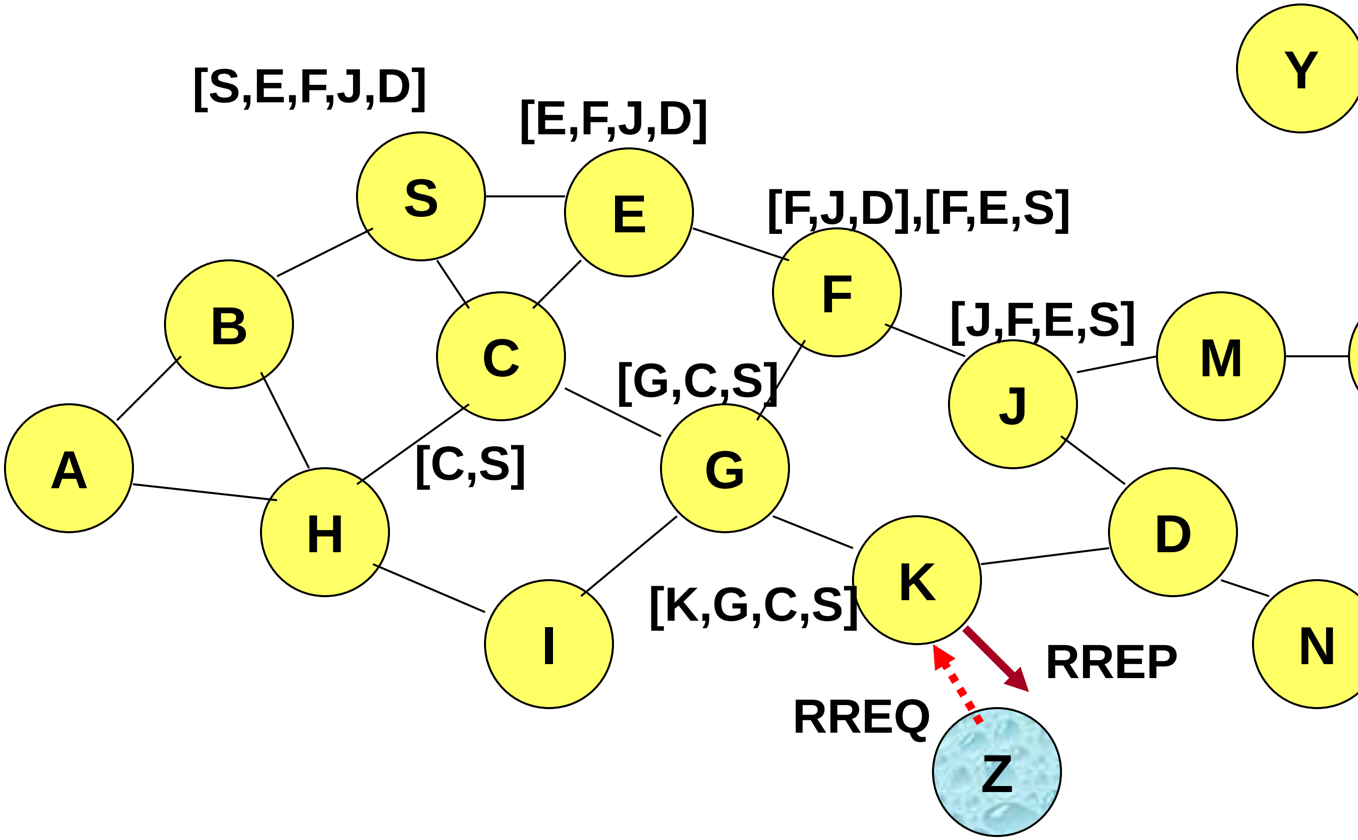
* Use of route cache
  - » can speed up route discovery
  - » can reduce propagation of route requests

Use of Route Caching



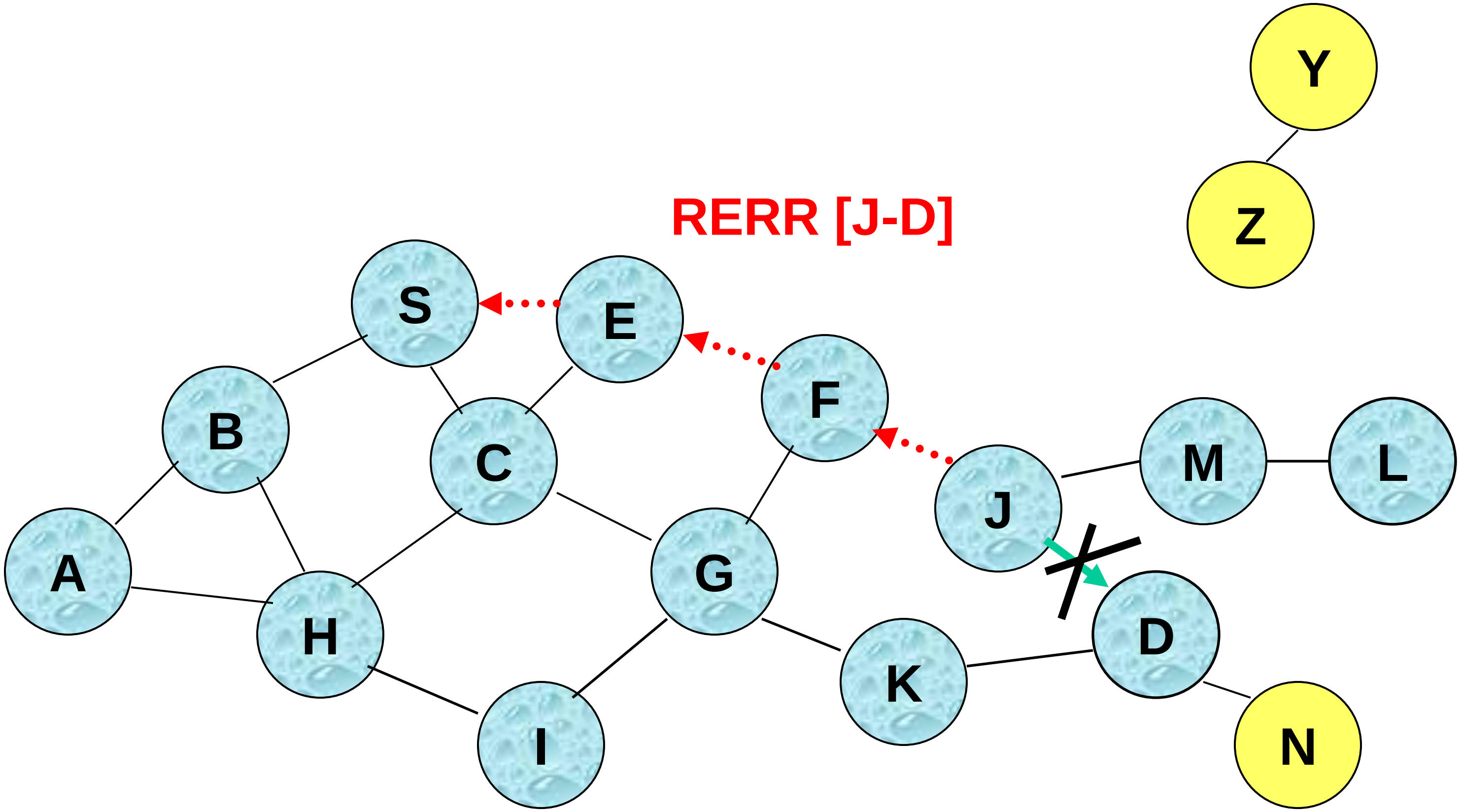
[P,Q,R] Represents cached route at a node
(DSR maintains the cached routes in a tree format)

Use of Route Caching



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

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]

Zone Routing Protocol (ZRP) [Haas98]

Zone routing protocol combines

* Proactive protocol: which pro-actively updates network state and maintains route regardless of whether any data traffic exists or not

* Reactive protocol: which only determines route to a destination if there is some data to be sent to the destination

ZRP

* 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

ZRP

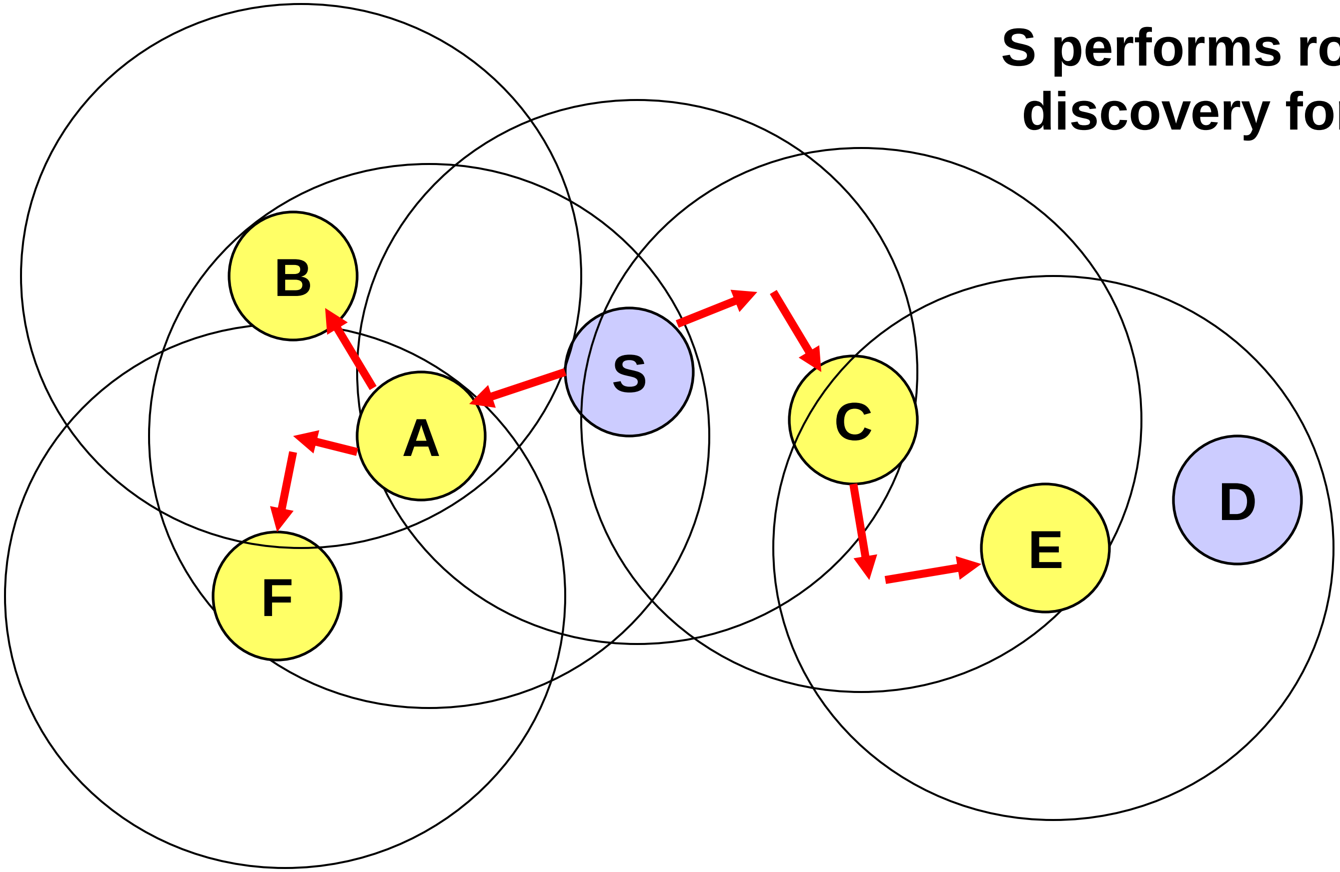
* Intra-zone routing: Pro-actively maintain state information for links within a short distance from any given node
  » Routes to nodes within short distance are thus maintained proactively (using, say, link state or distance vector protocol)

* Inter-zone routing: Use a route discovery protocol for determining routes to far away nodes. Route discovery is similar to DSR with the exception that route requests are propagated via peripheral nodes.

ZRP: Example with

Zone Radius = *d* = 2

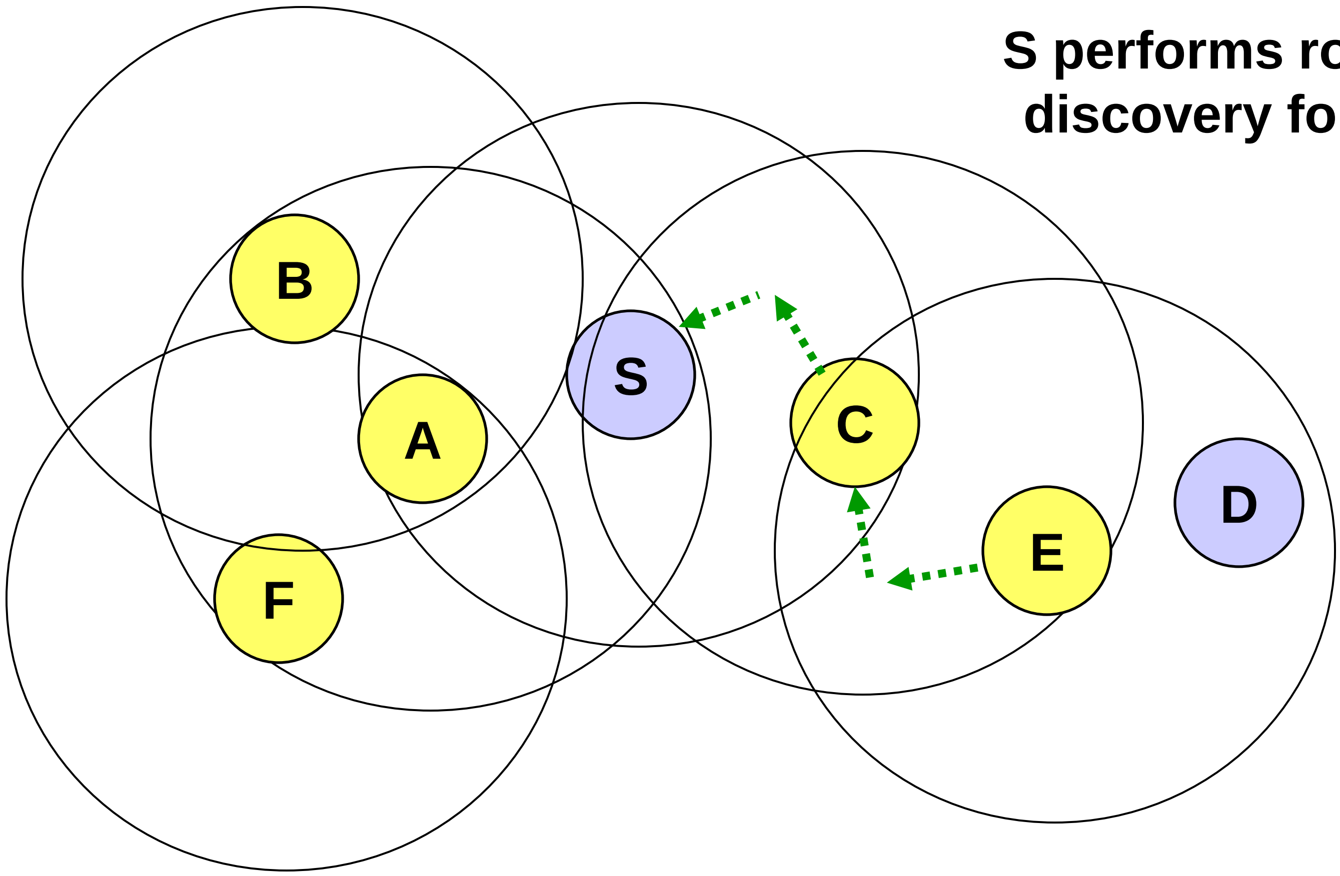
S performs route discovery for D



Denotes route request

ZRP: Example with *d = 2*

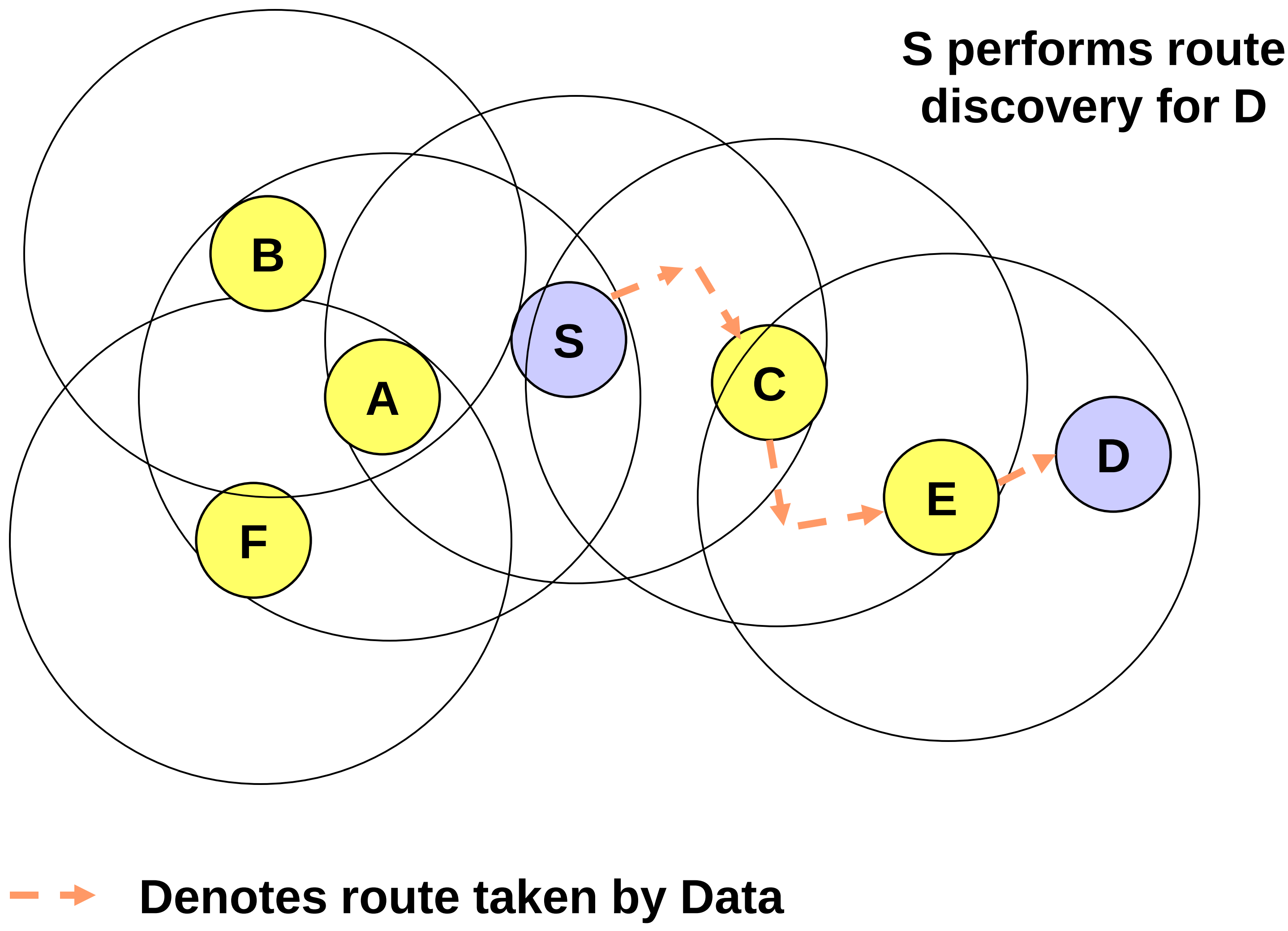
S performs route discovery for D



Denotes route reply

E knows route from E to D, so route request need not be forwarded to D from E

ZRP: Example with *d* = 2

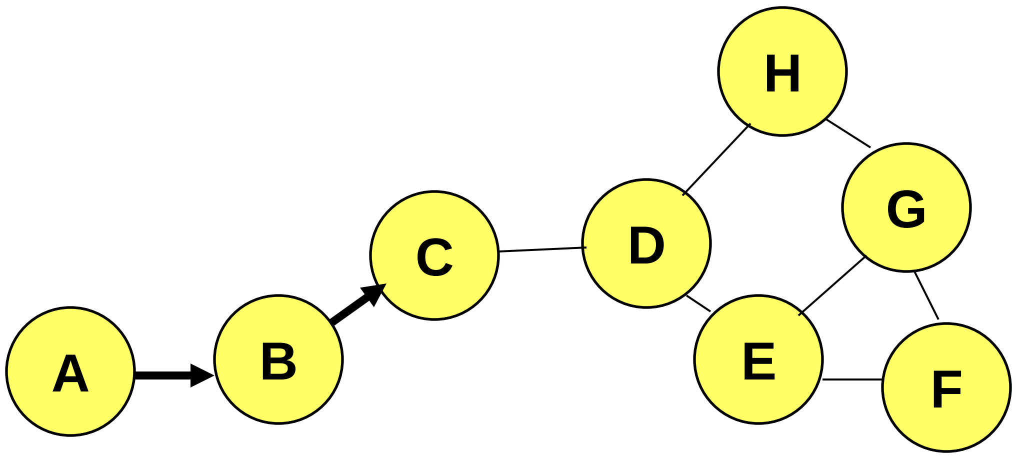


Landmark Routing (LANMAR) for MANET with Group Mobility [Pei00Mobihoc]

* A *landmark* node is elected for a group of nodes that are likely to move together
* A *scope* is defined such that each node would typically be within the scope of its landmark node
* Each node propagates *link state* information corresponding only to nodes within it *scope* and *distance-vector* information for all *landmark* nodes
  - » Combination of link-state and distance-vector
  - » Distance-vector used for landmark nodes outside the scope
  - » No state information for non-landmark nodes outside scope maintained

LANMAR Routing to Nodes Within Scope

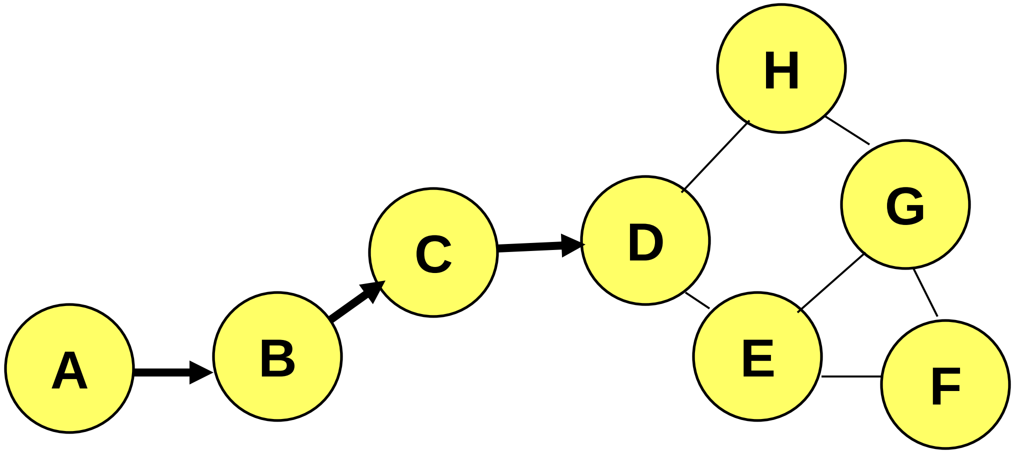
* Assume that node C is within scope of node A



* Routing from A to C: Node A can determine next hop to node C using the available link state information

LANMAR Routing to Nodes Outside Scope

* Routing from node A to F which is outside A's scope
* Let H be the landmark node for node F



* Node A somehow knows that H is the landmark for C
* Node A can determine next hop to node H using the available distance vector information