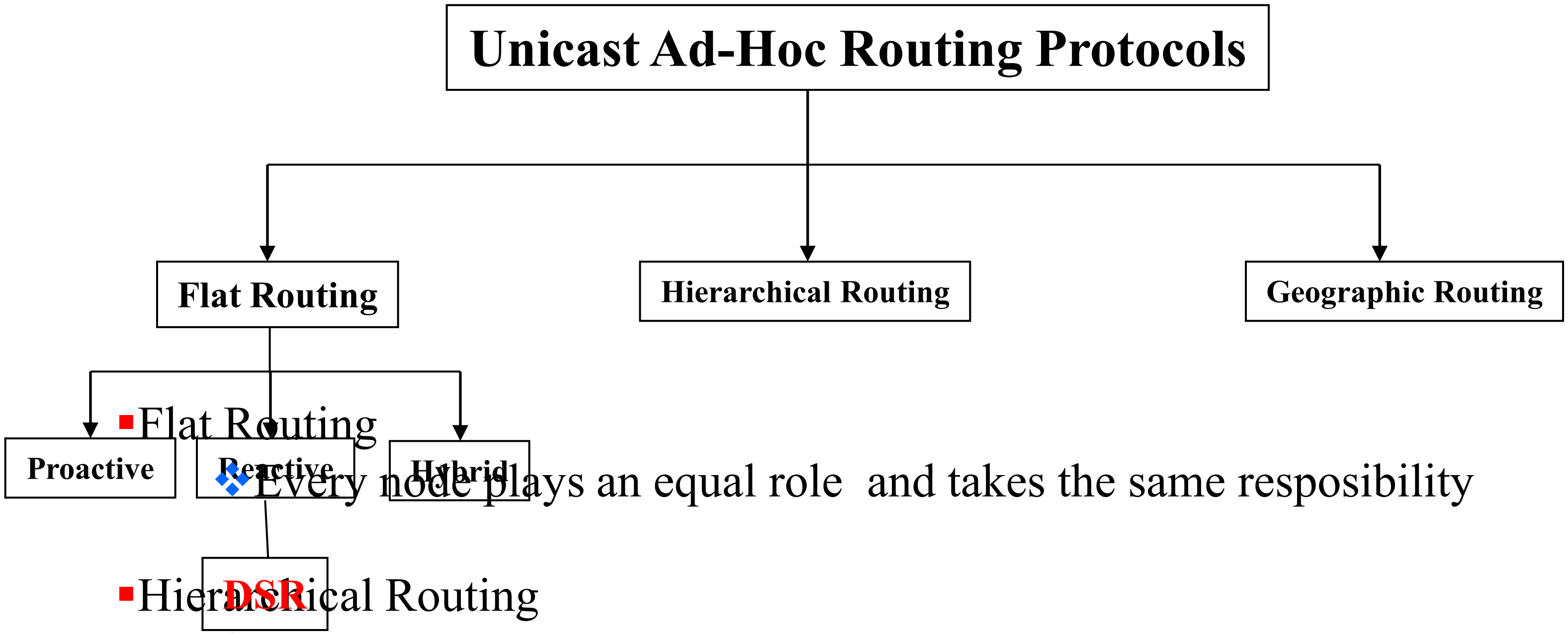
Dynamic Source Routing in Ad Hoc Wireless Networks

Background

- Difficult problem
  - Wireless communication
  - Dynamic topology
  - Frequent link broken and network partition
- No single protocol works well in all cases
- Many proposed protocols only address a sub-space of the problem domain
  - DSR only works for mobile ad hoc networks with middle or small size and nodes' speed is moderate and every node has enough energy

Classification of Unicast Ad-Hoc Routing Protocols



- Flat Routing
  - Every node plays an equal role and takes the same resposibility
- Hierarchical Routing
  - Nodes are grouped, some nodes are selected as group leaders
  - Leaders take more responsibilities than others
- Geographic Routing
  - Use geographic location information to make routing and forwarding decisions

- Proactive protocols
  - Maintain routes independent of traffic pattern
  - Traditional link-state and distance-vector routing protocols are proactive
- Reactive protocols
  - Determine routes only if needed

Trade-Off Between Proactive and Reactive

- Latency of route discovery
  - Proactive protocols have lower latency since routes are maintained all times
  - Reactive protocols have higher latency because a node needs to find a route when it has data to send

- Overhead of route maintenance
  - Reactive protocols have lower overhead since routes are maintained ed only if they are needed
  - Proactive protocols have higher overhead due to continuous route updating

- Which approach achieves a better trade-off depends on the traffic and mobility patterns
  - If most nodes always have data to send and nodes' mobility is high, proactive protocols may produce higher routing efficiency
  - If only a few nodes have data to send and nodes' mobility is low, reactive protocols may produce higher routing efficiency

Basic Assumptions in DSR

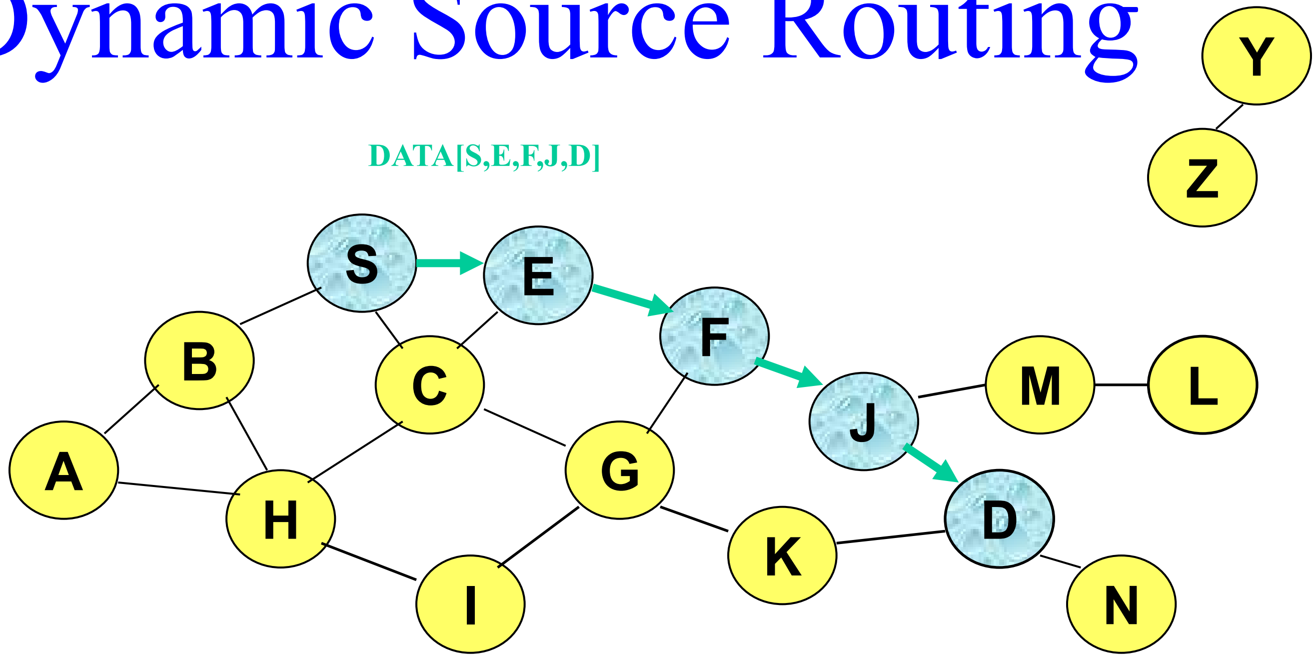
- All nodes are willing to forward packets for other nodes in the network

- The diameter of an ad-hoc network will not be too larger
  - Packet header will be bigger than payload if route is very longer

- The node's speed is moderate
  - Local route cache will become stale soon if node's speed is high

- All nodes are overhearing (promiscuous)
  - No energy saving

Dynamic Source Routing



- When S sends a data packet to D, the entire route is included in the packet header
- Intermediate nodes use the source route embedded in the packet's header to determine to whom the packet should be forwarded
- Different packets may have different routes, even they have the same source and destination

Hence called as dynamic source routing

Basic For DSR

- Two basic mechanisms
  - Route Discovery
    - Route Request (RREQ)
    - Route Reply (RREP)
  - Route Maintenance
    - Route Error (RERR)
- Key optimization
  - Each node maintains a route cache
    - Overhears data, RREQ, RREP, and RERR packets
    - Passively collects new routes as many as possible
    - Reduces the cost of Route Discovery and Route Maintenance

Route Discovery

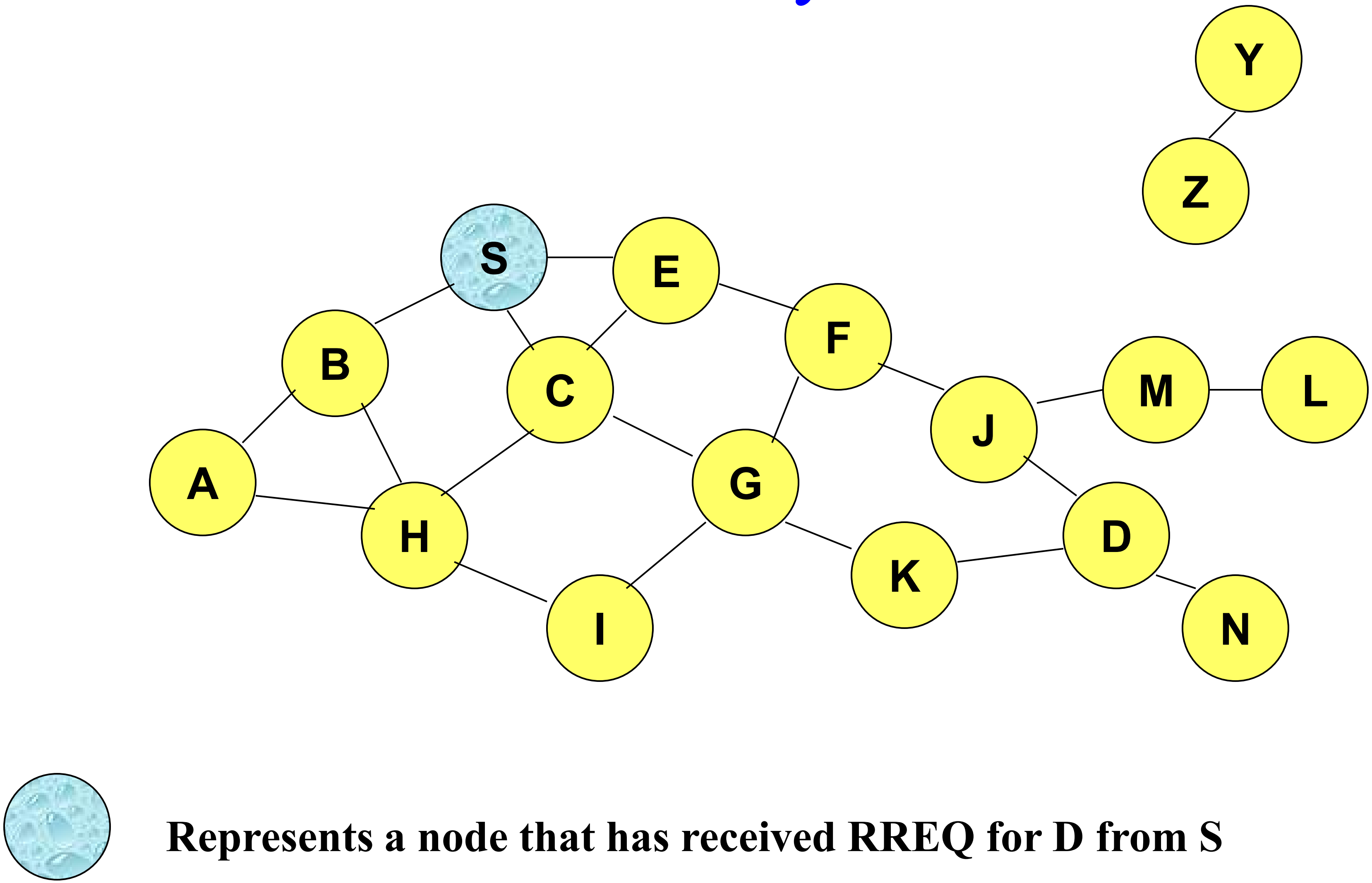
- When to perform a Route Discovery ?

- Every route request packet (RREQ) contains
  <target address, initiator address, route record, request ID>

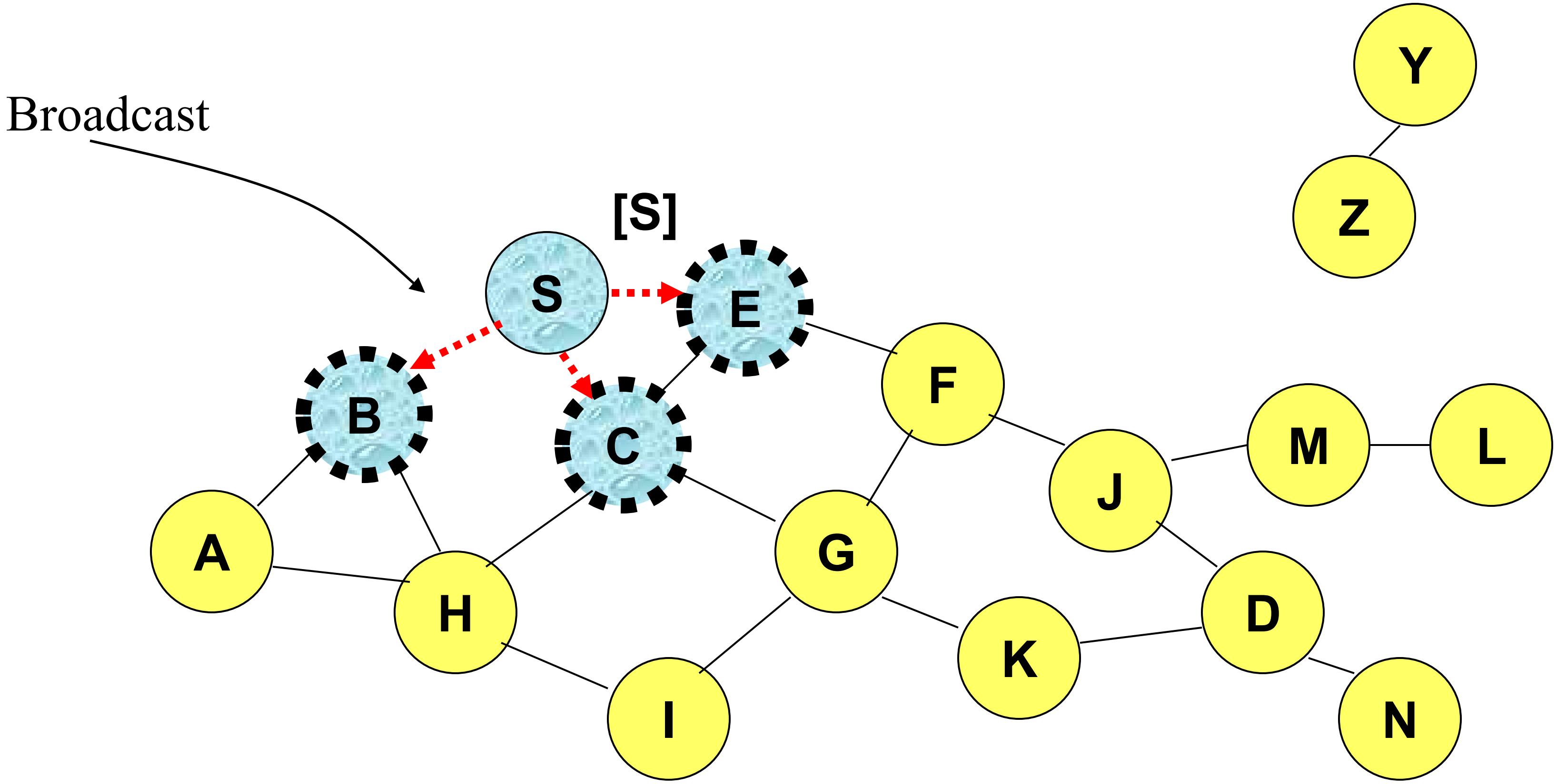
- Each node maintains a list of the < initiator address, request ID>

- When a node Y receives a RREQ
  - Discards the route request packet
    - if < initiator address, request ID> is in its list
  - Return a route reply packet which contains a route from initiator to target
    - If Y is target
    - If Y has an entry in its route cache for a route to target
  - Append itself address to the route record in RREQ and re-broadcast RREQ

Route Discovery in DSR



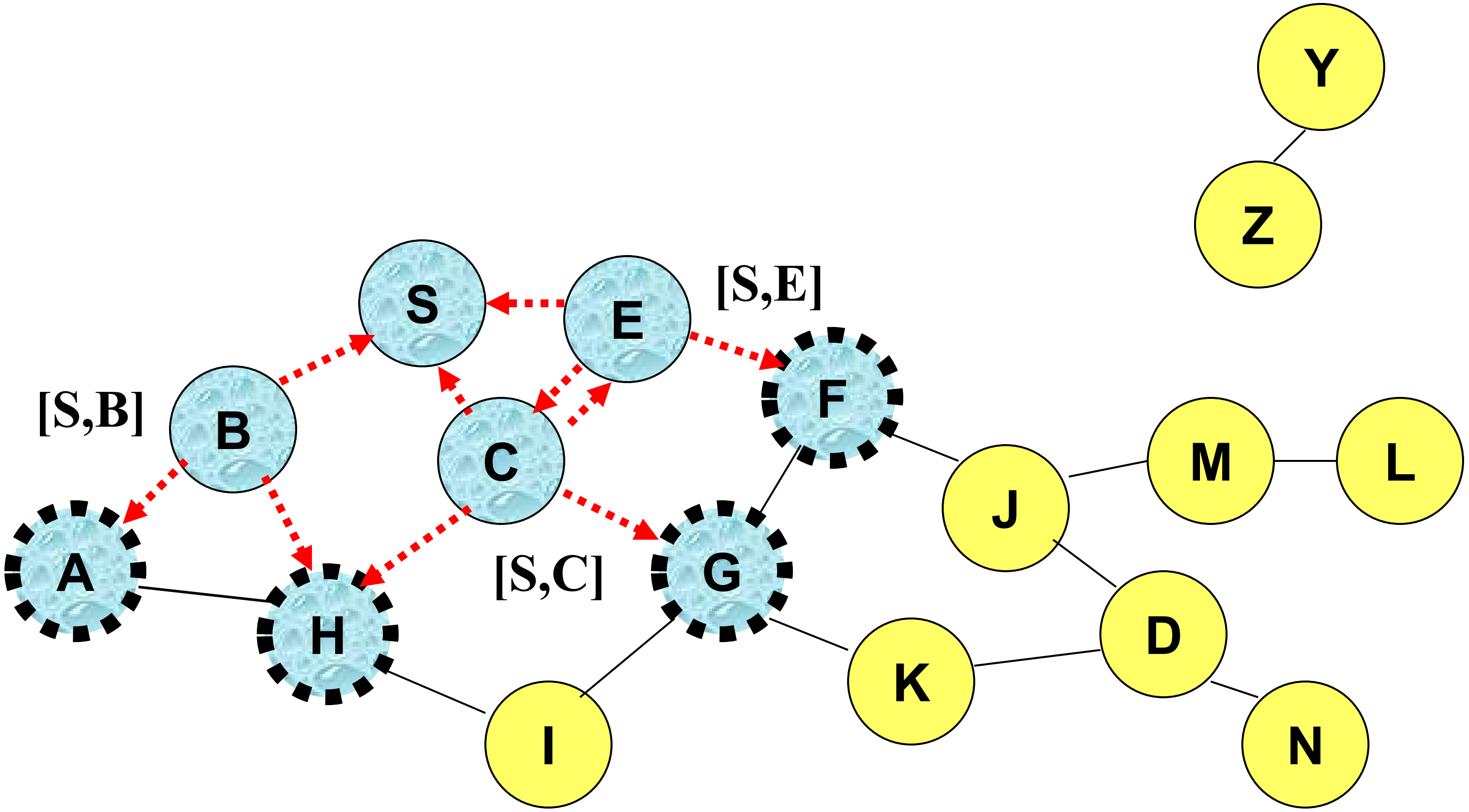
Route Discovery in DSR



Represents transmission of RREQ

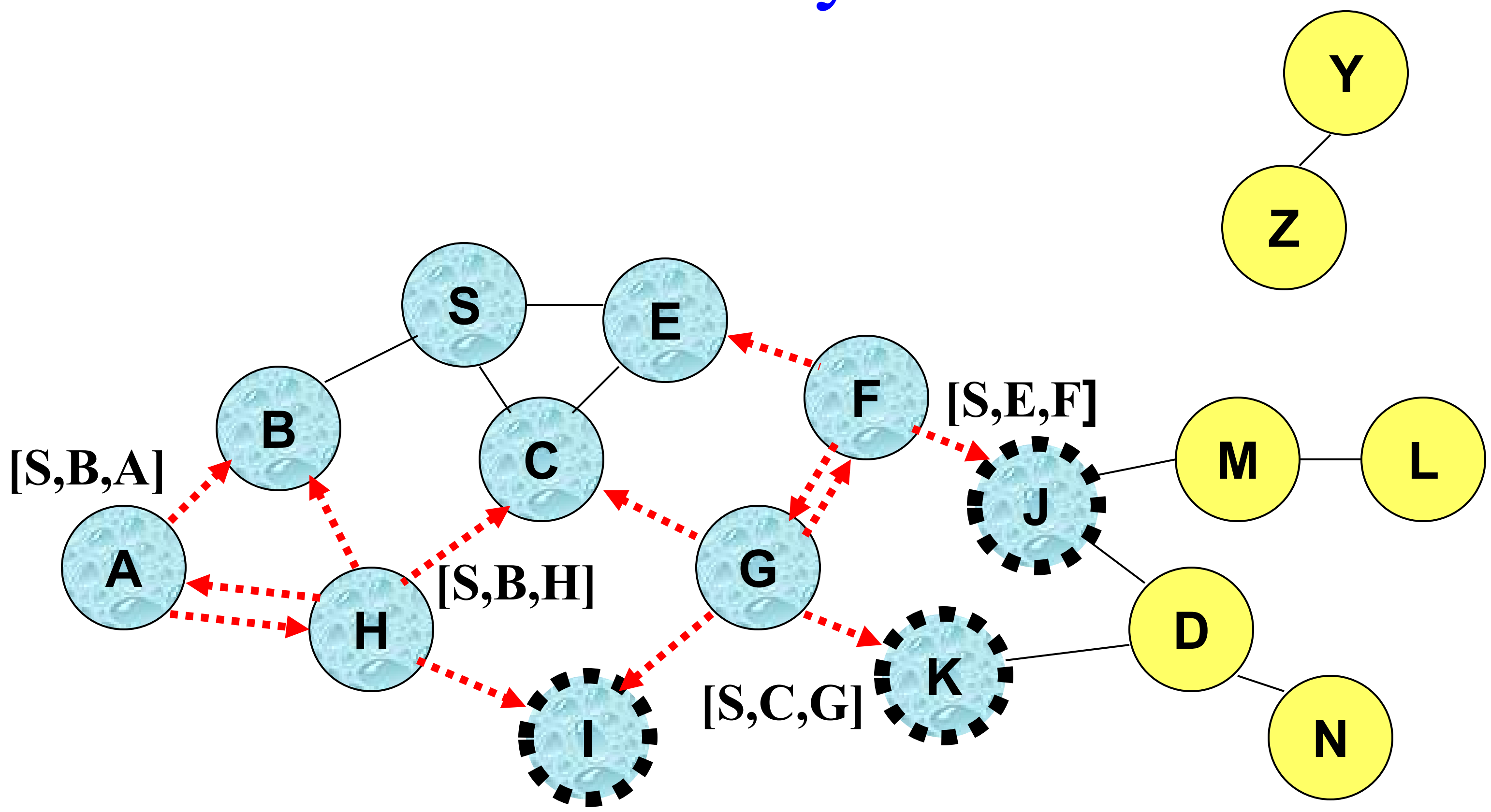
[X,Y] Represents route record stored in RREQ

Route Discovery in DSR



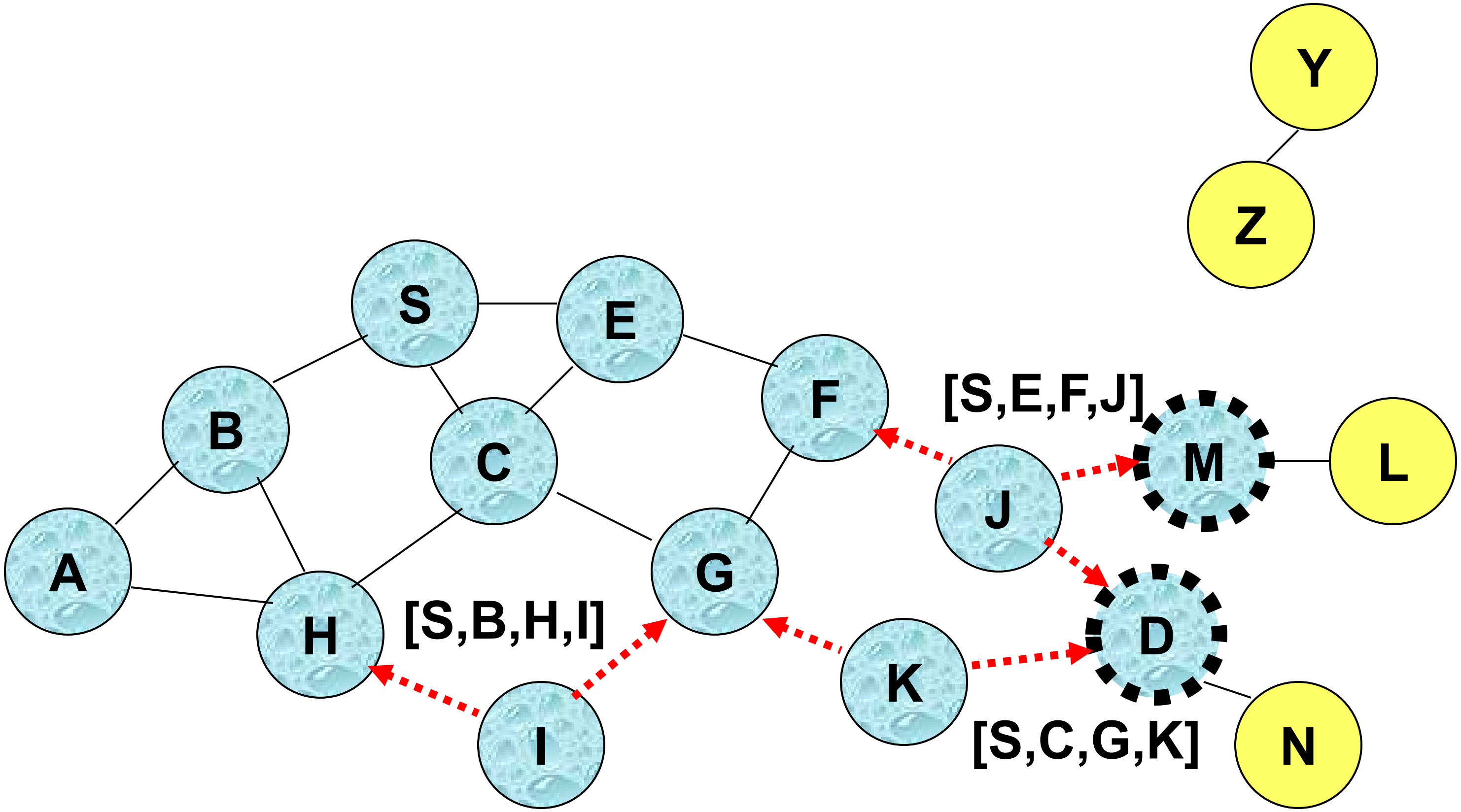
• **Node H receives packet RREQ from two neighbors:** **potential for collision**

Route Discovery in DSR



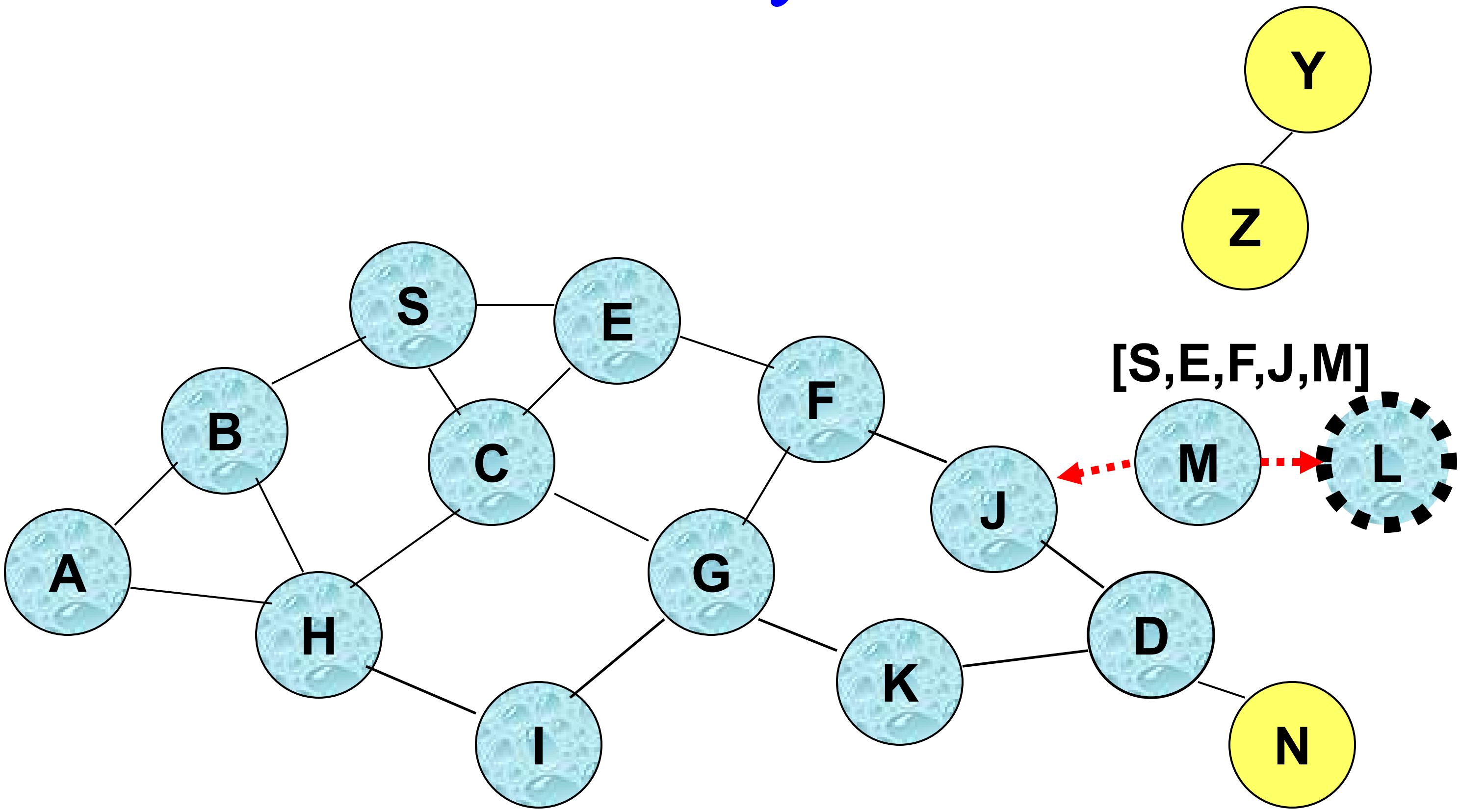
- C receives RREQ from G and H, but does not forward it again, because C has already forwarded RREQ once

Route Discovery in DSR



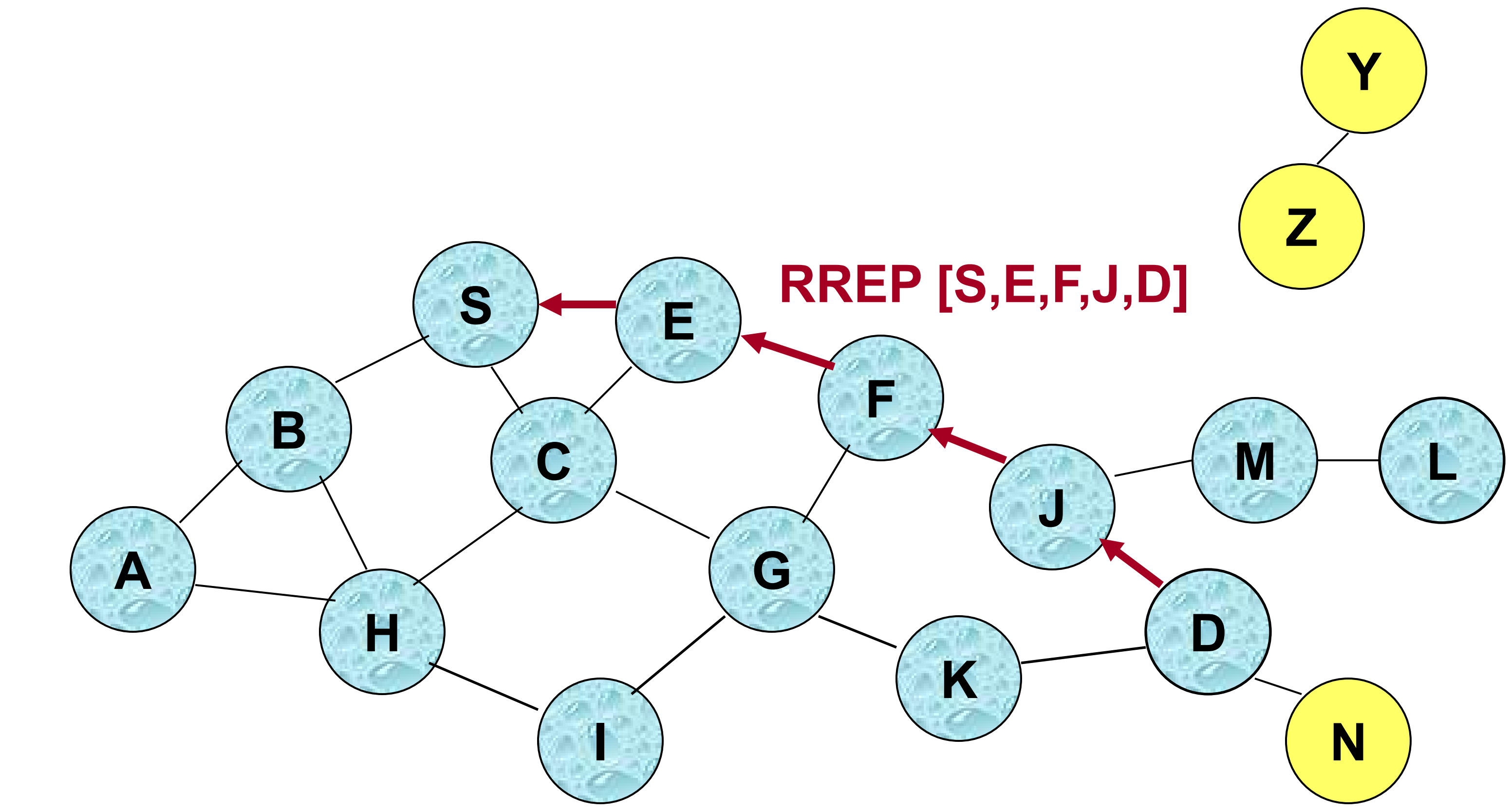
J and K both broadcast RREQ to D
Their transmissions may collide at D

Route Discovery in DSR



D does not forward RREQ, because D is the intended target

Route Reply in DSR



Represents RREP control message

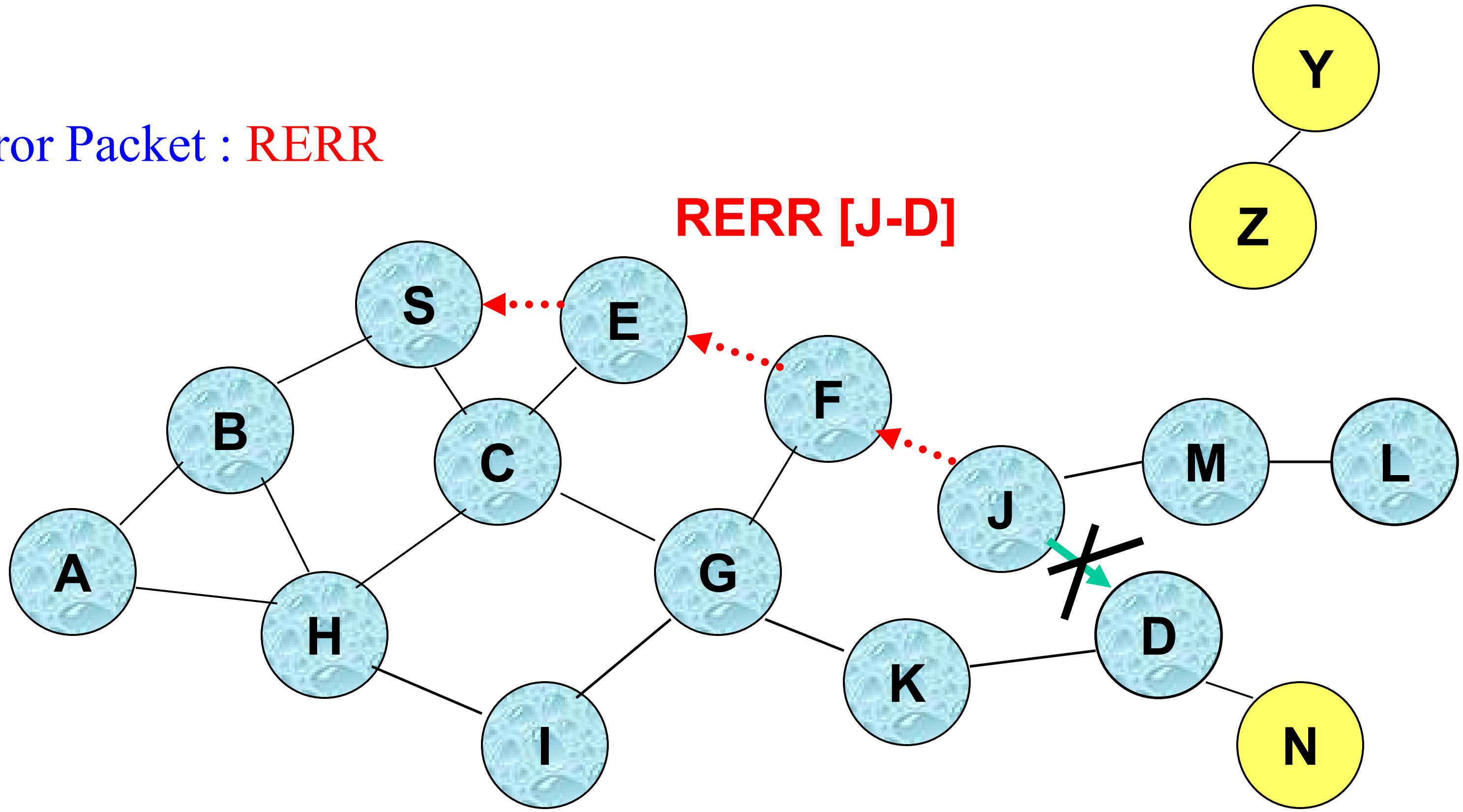
More Details For Route Reply in DSR

- Destination D on receiving the first RREQ, sends a Route Reply (RREP)
- RREP includes the route from S to D
- How Route Reply packet is sent to S?
  - Route Reply can be sent by reversing the route in Route Request (RREQ)
    - If links are bi-directional
  - If unidirectional (asymmetric) links are allowed, then a route to S is needed
    - Local route cache has a route to S
    - Piggybacking Route Reply in Route Request packet for S

NOTE: If IEEE 802.11 MAC is used, then links have to be bi-directional

An Example of Route Maintenance

Route Error Packet : RERR



J sends a route error to S along route J-F-E-S when it finds link [J-D] broken

Nodes hearing RERR update their route cache to remove all invalid routes related with link J-D

More Details about Route Maintenance

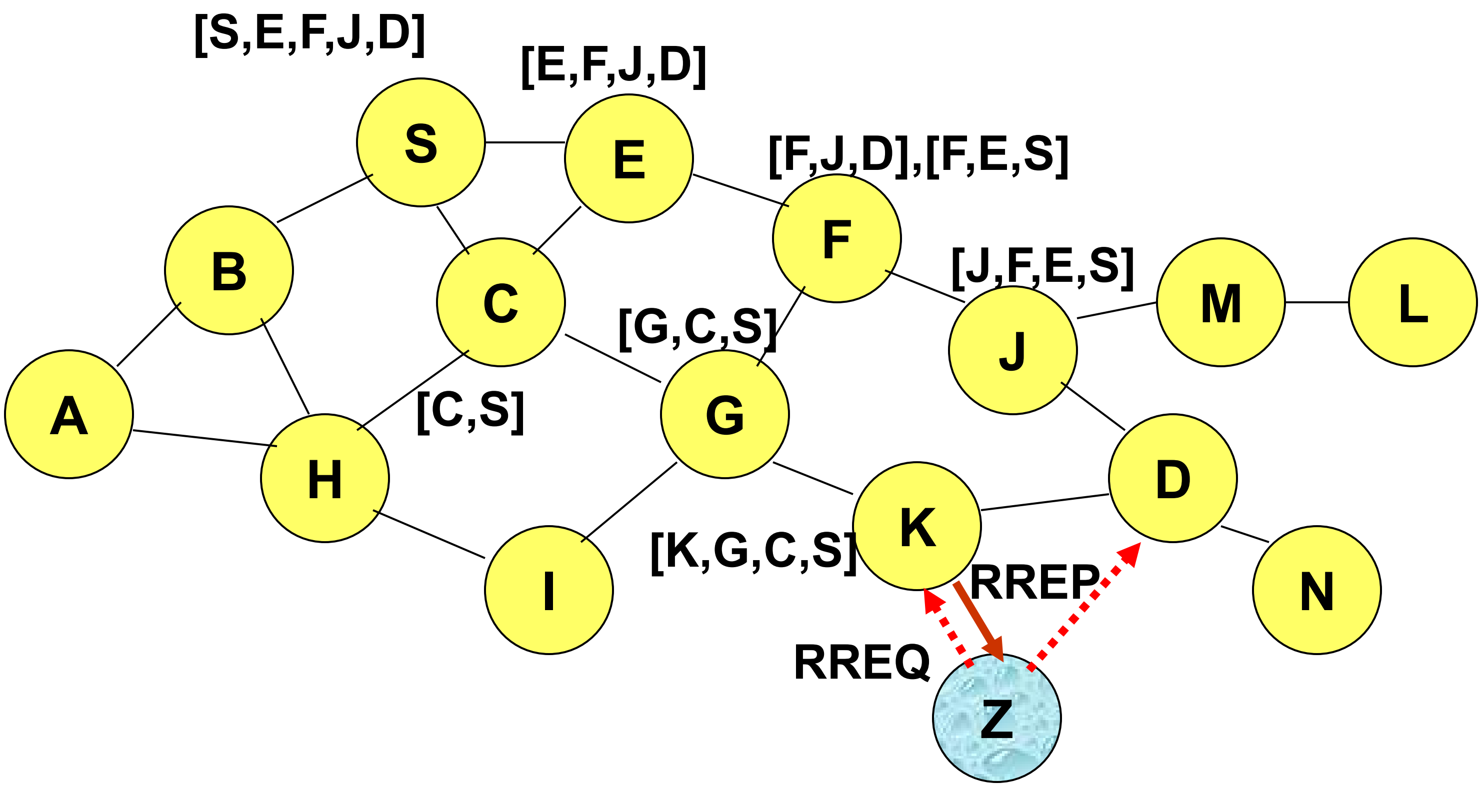
Route [S, node-1,node-2,……,node-k, D]

- Hop-by-hop maintenance (MAC or network layer)
  - How to find link [node-i,node(i+1)] is down ?
    - Utilize MAC level acknowledgement
    - Passive acknowledge (overhearing node(i+1) re-transmission)
    - Insert a bit in packet header to ask an explicit acknowledgement from node(i+1)
  - How to send route error packet to S?
    - Use the reverse route [node-i,node(i-1), ……,node-1, S]
    - Use node-i route cache to get a route to S
    - Piggybacking route error packet in route discovery packet S
- End-to-end maintenance (transport or application layer)
  - D sends ACK to S to indicate the route status
  - But S does not know which link is broken

DSR Optimization: Route Caching

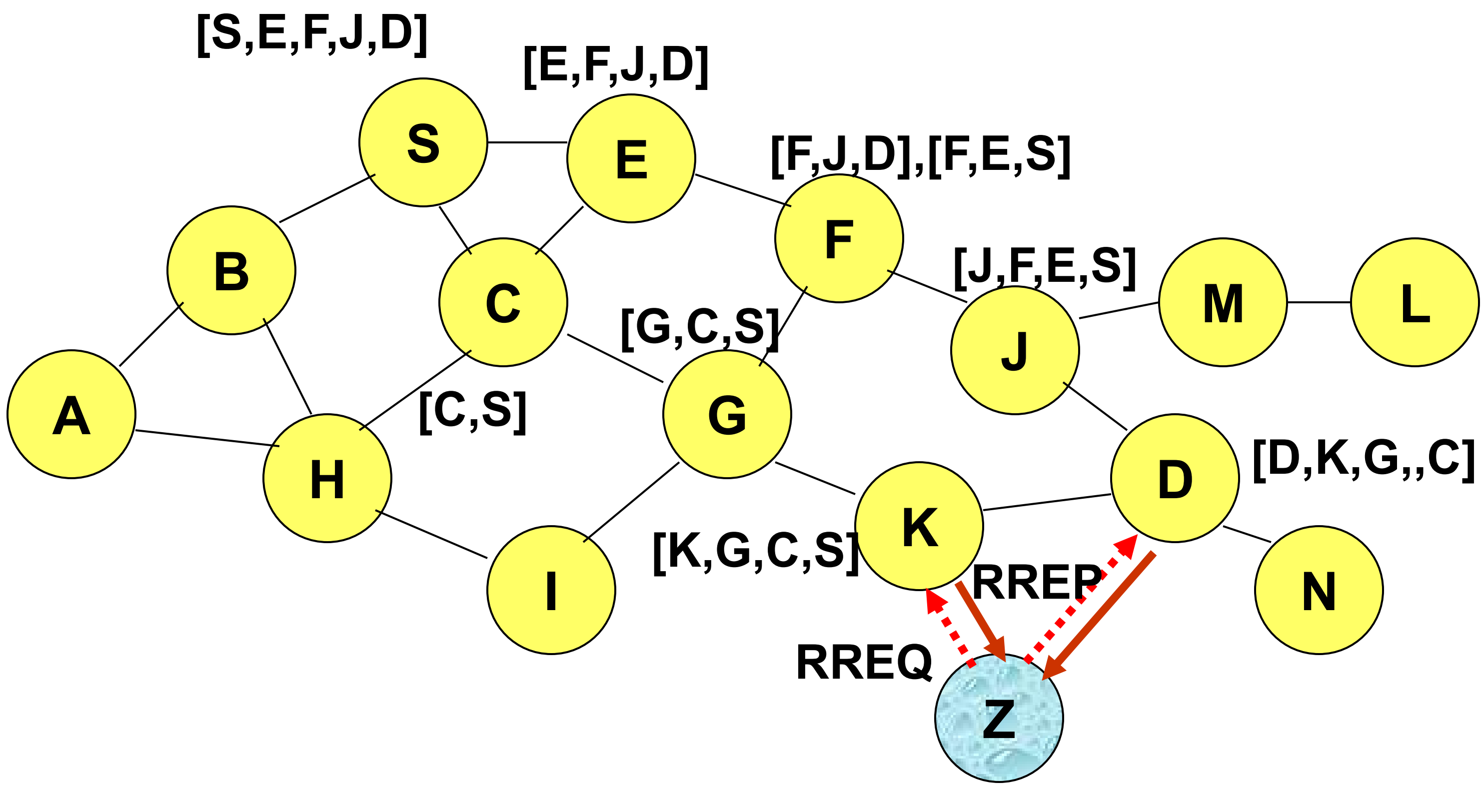
- Each node caches a new route it learns by *any means*
- When S finds route [S,E,F,J,D] to D, S also learns route [S,E,F] to F
- When K receives Route Request [S,C,G] destined for some node D, K learns route [K,G,C,S] to S if links are bi-directional
- F forwards Route Reply RREP [S,E,F,J,D], F learns route [F,J,D] to D
- When E forwards Data [S,E,F,J,D] it learns route [E,F,J,D] to D

Use of Route Caching
Can Speed up Route Discovery



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

Use of Route Caching
Can Reduce Propagation of Route Requests



Route Replies (RREP) from node K and D limit flooding of RREQ.

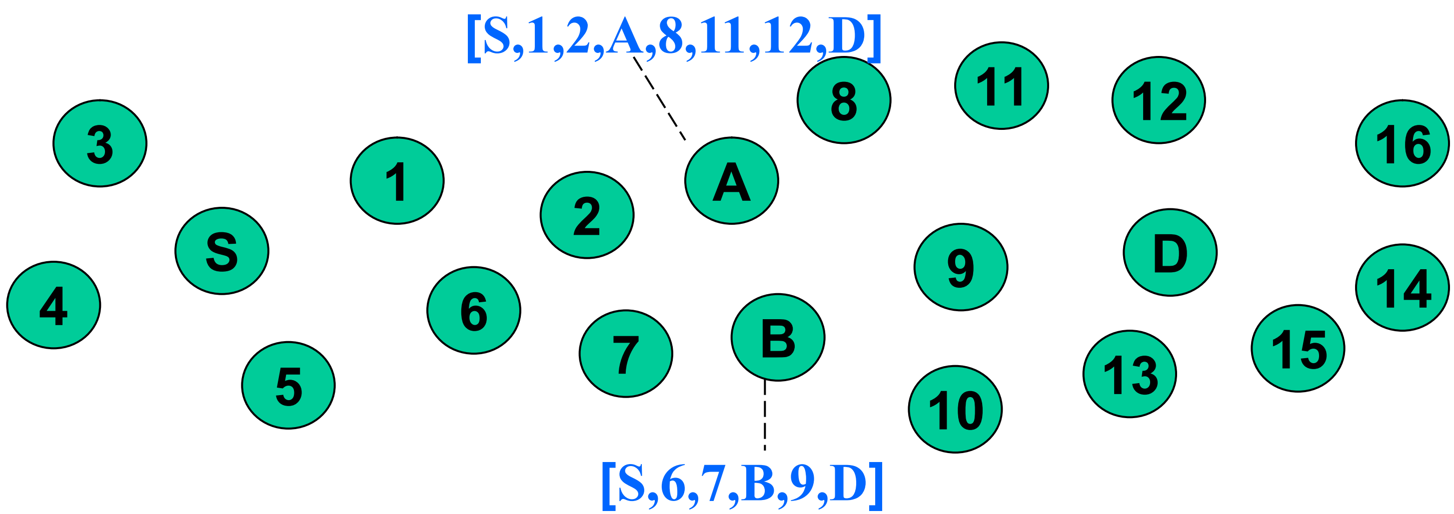
Route Caching: Beware!

- With time passing and node moving, cached routes may become invalid
- Stale caches can adversely affect on network performance
- Stale or invalid information may be propagated to whole network

Other DSR Optimizations (1)

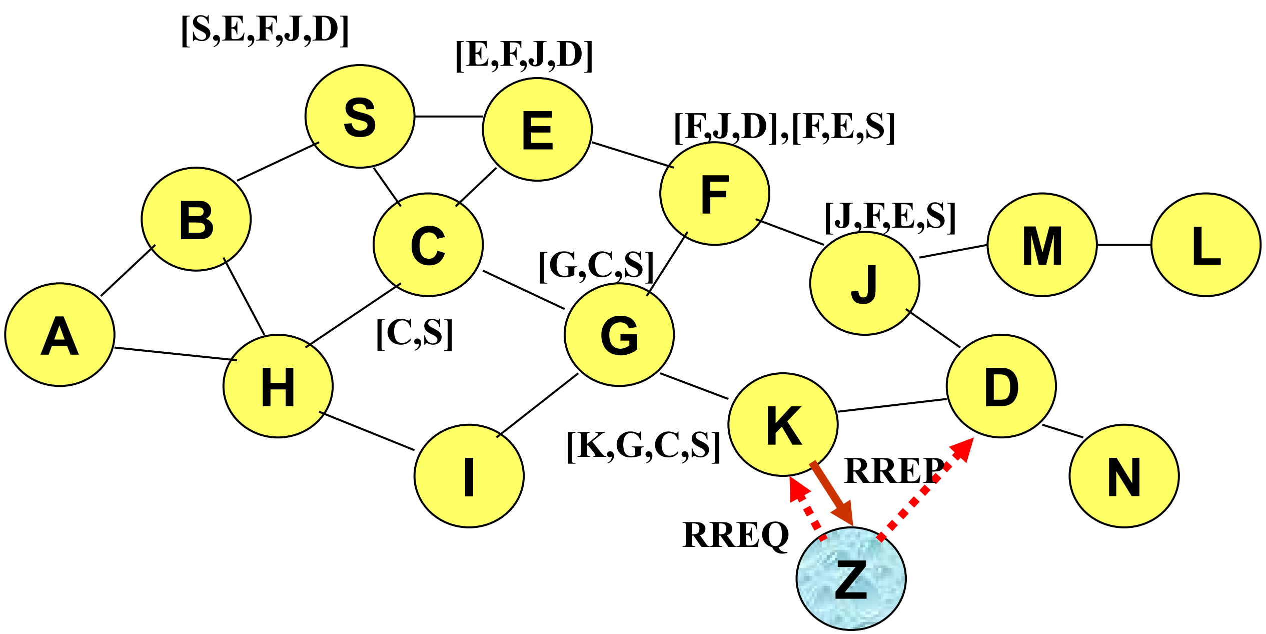
- Avoid *Route Reply Storm* problem
  - When intermediate nodes reply Route Request by using local route cache
    - Pick a random number d=H*(h-1+r)

    H= average per hop delay; h= length of the route; r = random num in [0,1]
    - Delay transmitting the route reply for a period of d
    - During this period, cancel the route reply if overhearing a packet contains a route from the same initiator to the same target with length no longer than h



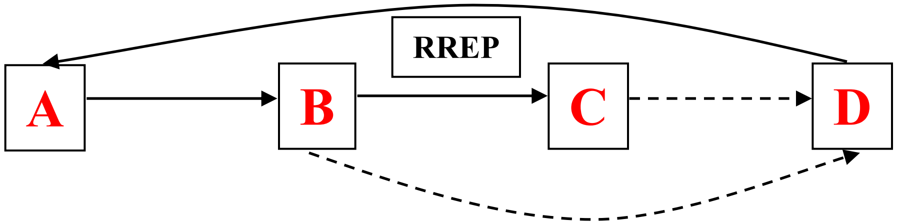
Other DSR Optimizations (2)

- Expanding Ring (limits the propagation of RREQ packets)
  - First, set TTL = 1 for first route request packet
  - If no route reply is received after some time period, set TTL = maximum for next route request packet



Other DSR Optimizations (3)

- Reflecting shorter route
  - D may find that route may be shortened by excluding the intermediate hop C, D then sends an unsolicited route reply packet to A to inform the shorter route



- Improved handling of errors due to network partitions
  - Limit the rate of route discovery packet transmission
    - Exponential backoff

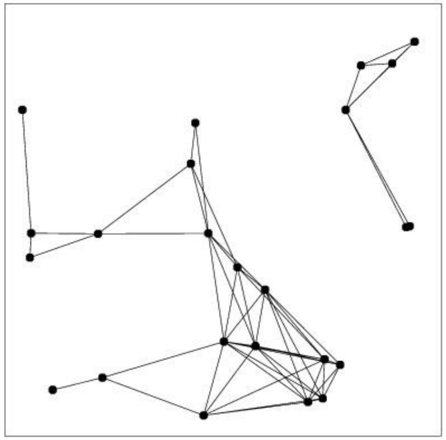
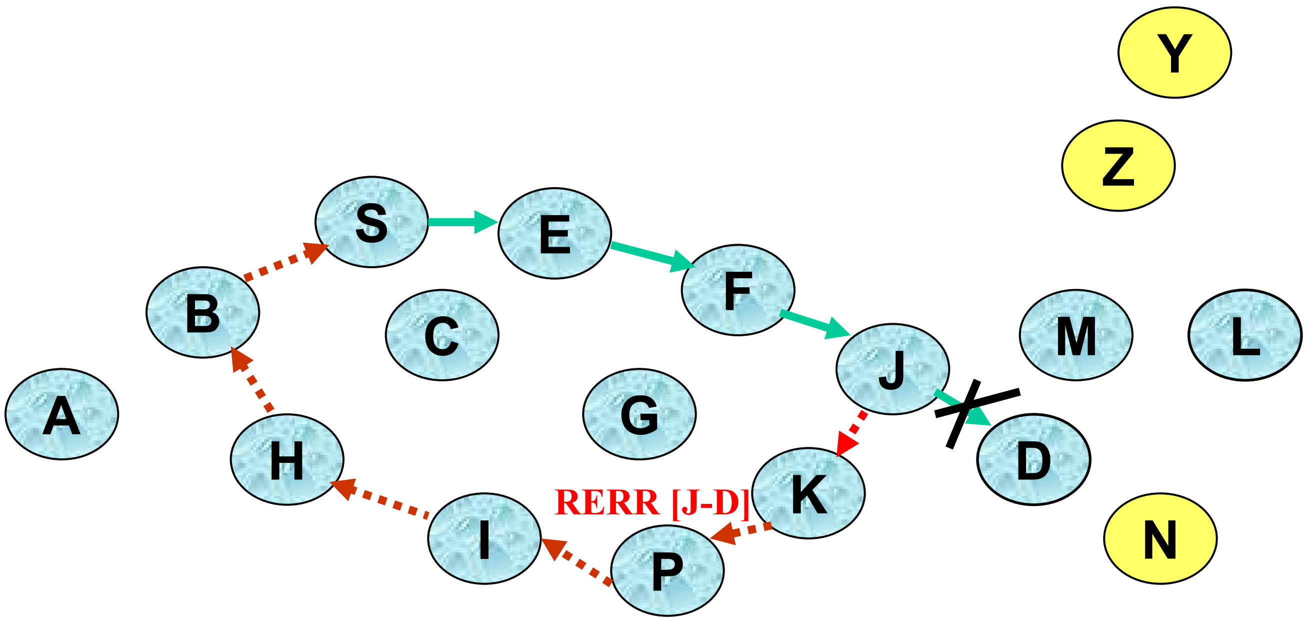


Figure 6 Example of disconnected clusters with 24 hosts

Other DSR Optimizations (4)

- Improved handling of route errors if links are unidirectional
  - S retransmits route error packet along the original route if links are unidirectional



Simulation Parameters

C= 100 kbps **Medium size room: 9 by 9 meters** **Velocity: 0.3~0.7 meter/second**

Transceiver range 3 meters **Error Rate = 5%** **Packet sizes: 1000 Bytes (75%) or 32 Bytes (25%)**

Packet inter arrival time in a conversation : exponential distribution with mean 2~5 packets/second

Conversation life time: exponential distribution with mean 15 second

Total packets of a conversation : geometric distribution with mean 1000 packets

Table 1 Parameter values used in the simulation

Parameter	Value
Period between nonpropagating route requests	5 sec.
Nonpropagating route request time out	100 msec.
Route request time out	500 msec.
Route request slot length	500 msec.
Maximum route request period	10 sec.
Route reply holdoff per-hop delay **(H)**	4 msec.

Low Overhead

Mobility pattern: Each node pauses at its current position for a period (pause time) and then randomly choose a new location to move to. This behavior is repeated by every node

Num transmits: sum of (packet)*(route length)
Optimal num transmits: sum of (data packet)* (optimal route length)

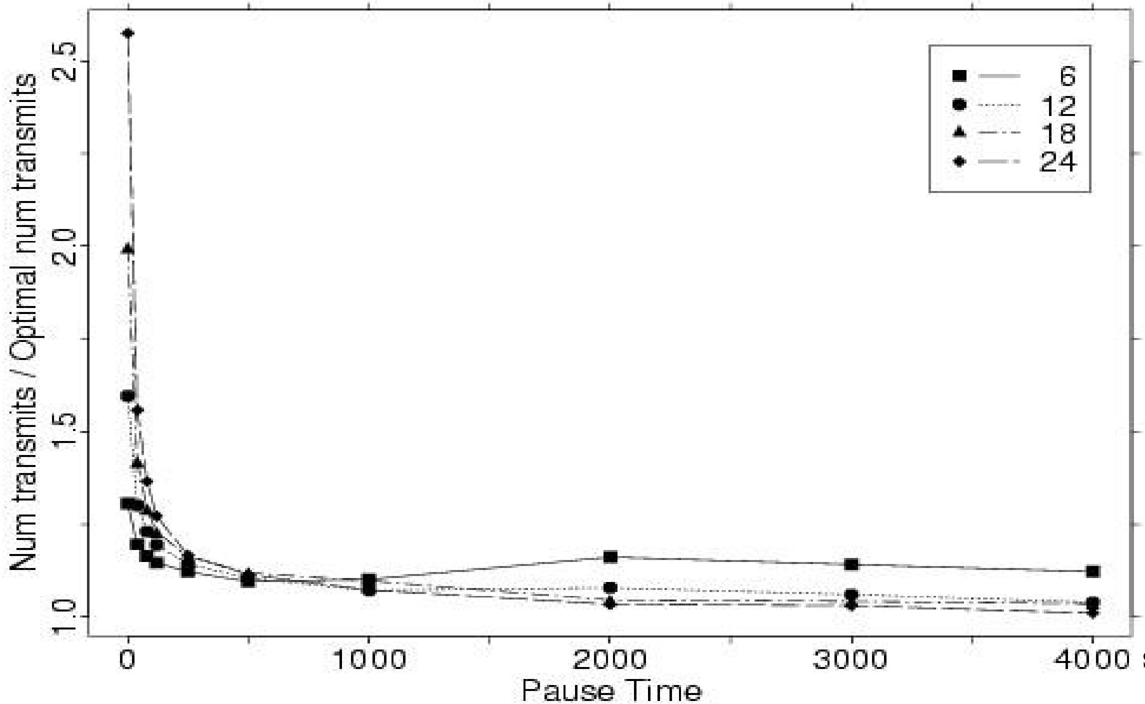


Figure 5 Average total number of transmissions performed relative to optimal (20 runs)

Suboptimal Route

Average route length: DSR routes
Optimal route length : ideal routes

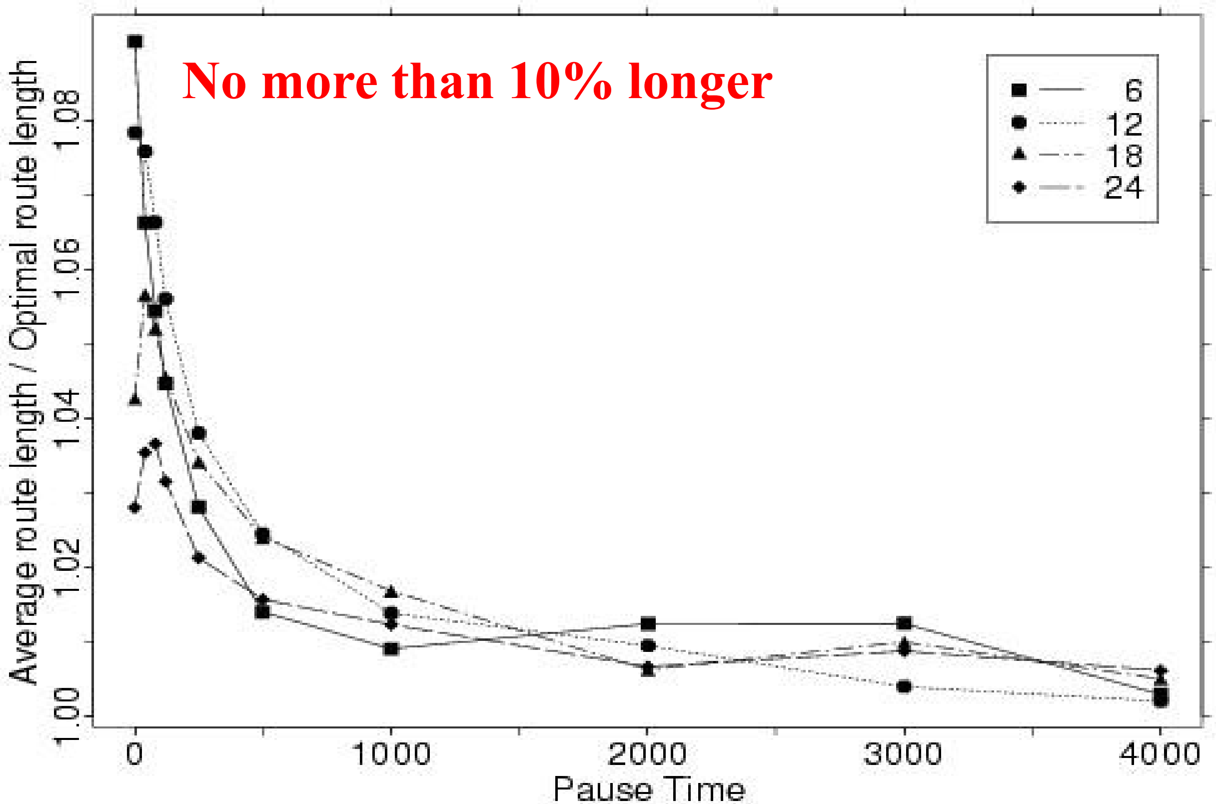


Figure 7 Average route length used relative to optimal (20 runs)

Dynamic Source Routing: Advantages

- Routes are discovered only they are needed
  - Reduces overhead of route maintenance
- Route caching reduce the cost of route discovery
- A single route discovery may yield many routes to the destination, due to intermediate nodes may reply route request from local caches
- Does not require symmetric links

Dynamic Source Routing: Disadvantages

- Packet header size grows with route length due to source routing
  - Inefficiency
- Route request packet may potentially reach all nodes in the network
  - RREQ flooding
- Route requests may collide at the targeted node
  - Pay so much but get nothing
- Every node needs to turn on its receiver all the time
  - No energy saving
- Increased contention if too many route replies come back
  - Route Reply *Storm*
- An intermediate node may send Route Reply using a stale cached route, thus polluting other nodes' caches
  - Mess up routing and forwarding