Announcements

Outline

- A Performance Comparison of Multi-Hop Wireless Ad Hoc Network Routing Protocols – CMU

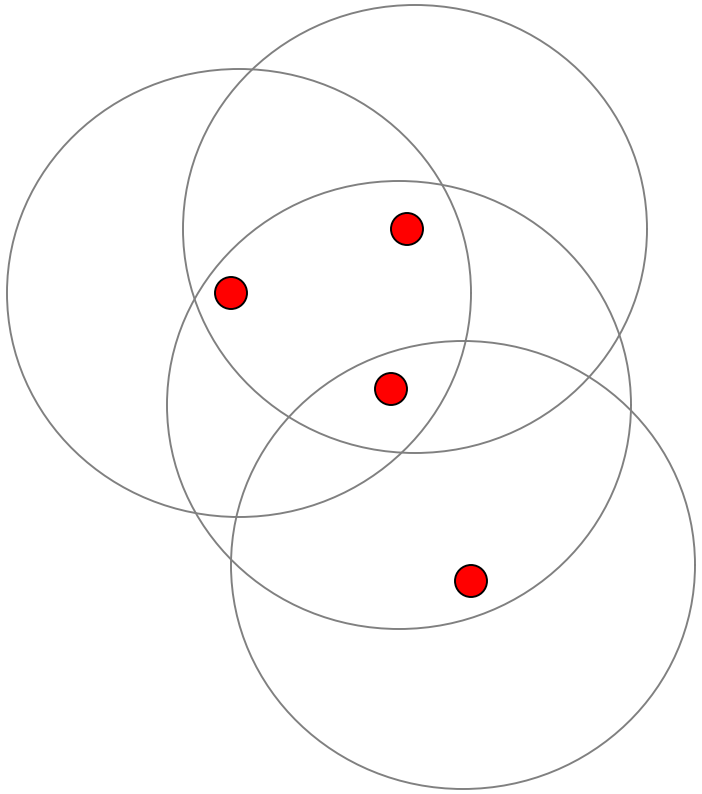
- Slides courtesy of Nitin Vaidya @ Texas A&M

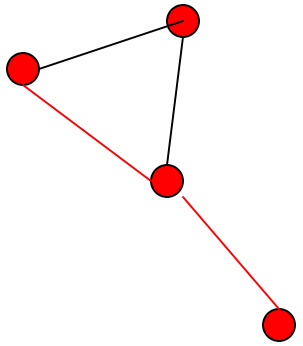
Mobile Ad Hoc Networks

- Formed by wireless hosts which may be mobile
- Without (necessarily) using a pre-existing infrastructure
- Routes between nodes may potentially contain multiple hops

Mobile Ad Hoc Networks

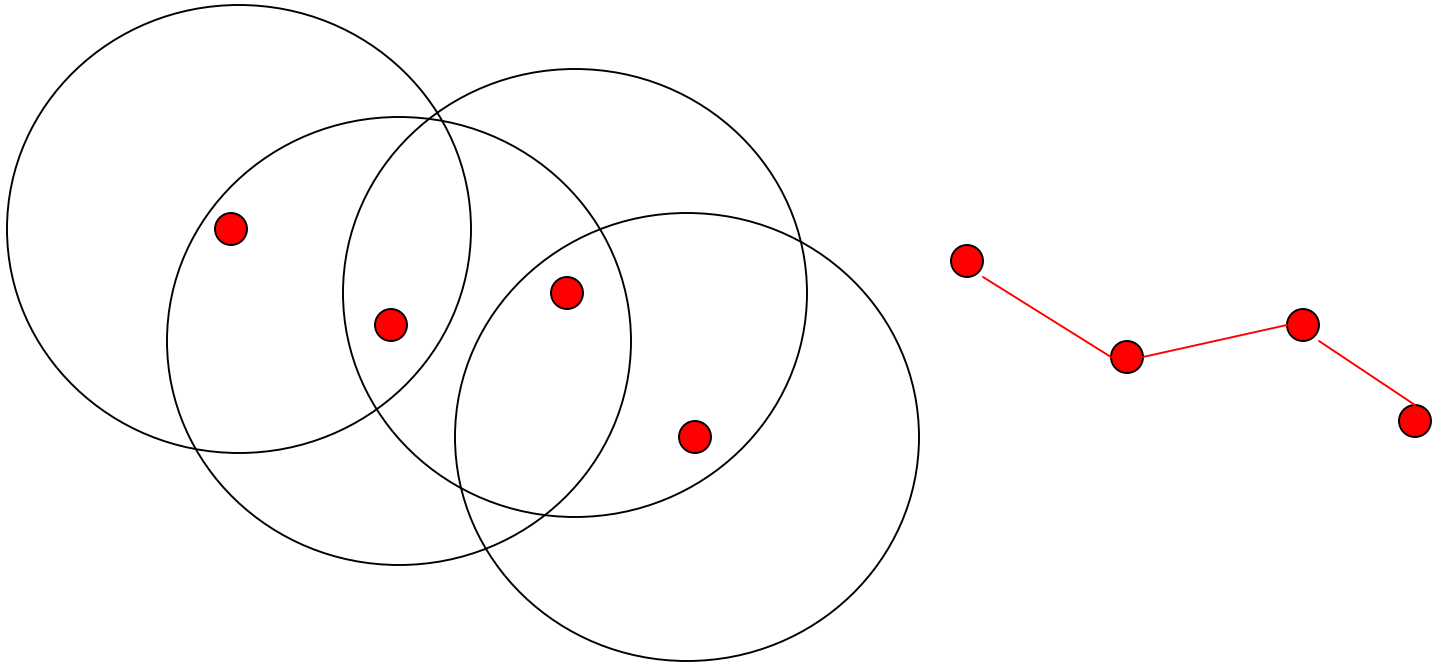
- May need to traverse multiple links to reach a destination





Mobile Ad Hoc Networks (MANET)

- Mobility causes route changes



Why Ad Hoc Networks ?

- Ease of deployment
- Speed of deployment
- Decreased dependence on infrastructure

Many Applications

- Personal area networking
  - cell phone, laptop, ear phone, wrist watch
- Military environments
  - soldiers, tanks, planes
- Civilian environments
  - taxi cab network
  - meeting rooms
  - sports stadiums
  - boats, small aircraft
- Emergency operations
  - search-and-rescue
  - policing and fire fighting

Many Variations

- Fully Symmetric Environment
  - all nodes have identical capabilities and responsibilities
- Asymmetric Capabilities
  - transmission ranges and radios may differ
  - battery life at different nodes may differ
  - processing capacity may be different at different nodes
  - speed of movement
- Asymmetric Responsibilities
  - only some nodes may route packets
  - some nodes may act as leaders of nearby nodes (e.g., cluster head)

Many Variations

- Traffic characteristics may differ in different ad hoc networks
  - bit rate
  - timeliness constraints
  - reliability requirements
  - unicast / multicast / geocast
  - host-based addressing / content-based addressing / capability-based addressing

- May co-exist (and co-operate) with an infrastructure-based network

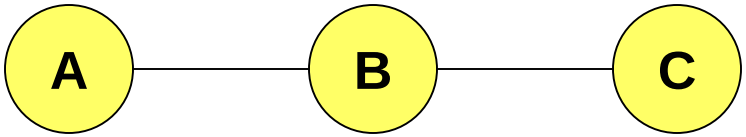
Many Variations

- Mobility patterns may be different
  - people sitting at an airport lounge
  - New York taxi cabs
  - kids playing
  - military movements
  - personal area network
- Mobility characteristics
  - speed
  - predictability
    - direction of movement
    - pattern of movement
  - uniformity (or lack thereof) of mobility characteristics among different nodes

Challenges

- Limited wireless transmission range
- Broadcast nature of the wireless medium
  - Hidden terminal problem (see next slide)
- Packet losses due to transmission errors
- Mobility-induced route changes
- Mobility-induced packet losses
- Battery constraints
- Potentially frequent network partitions
- Ease of snooping on wireless transmissions (security hazard)

Hidden Terminal Problem



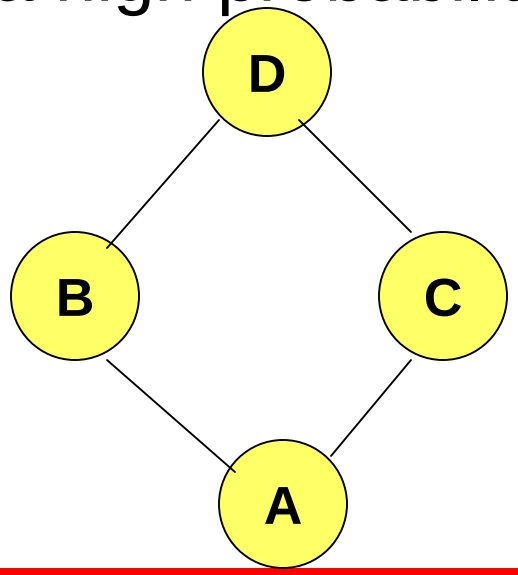
Nodes A and C cannot hear each other

Transmissions by nodes A and C can collide at node B

Nodes A and C are hidden from each other

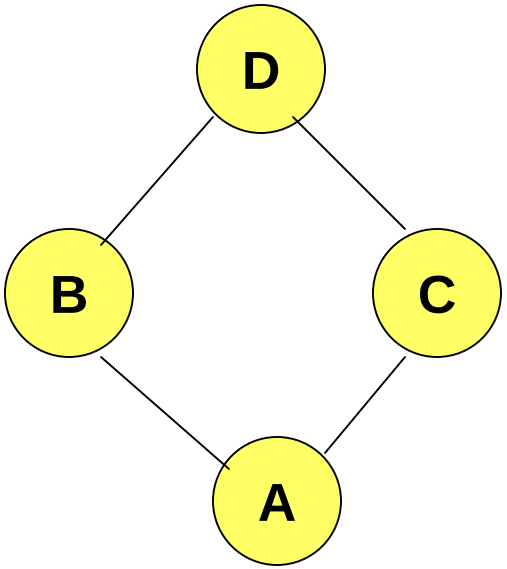
Broadcast Storm Problem

- When node A broadcasts a route query, nodes B and C both receive it
- B and C both forward to their neighbors
- B and C transmit at about the same time since they are reacting to receipt of the same message from A
- This results in a high probability of collisions



Broadcast Storm Problem

- Redundancy: A given node may receive the same route request from too many nodes, when one copy would have sufficed
- Node D may receive from nodes B and C both



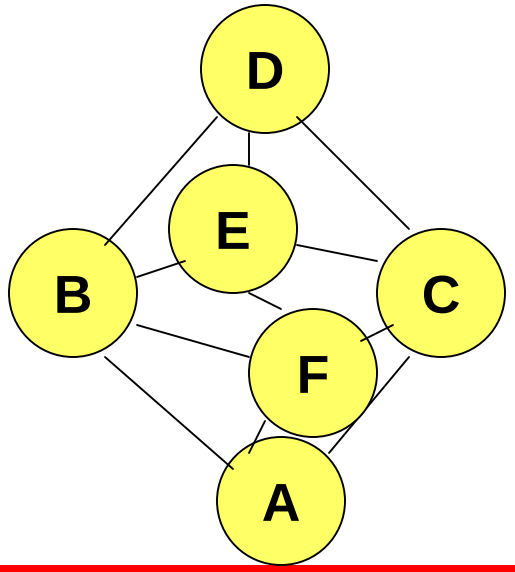
Solutions for Broadcast Storm

- Probabilistic scheme: On receiving a route request for the first time, a node will re-broadcast (forward) the request with probability p

- Also, re-broadcasts by different nodes should be staggered by using a collision avoidance technique (wait a random delay when channel is idle)
  - this would reduce the probability that nodes B and C would forward a packet simultaneously in the previous example

Solutions for Broadcast Storms

- Counter-Based Scheme: If node E hears more than *k* neighbors broadcasting a given route request, before it can itself forward it, then node E will not forward the request
- Intuition: *k* neighbors together have probably already forwarded the request to all of E's neighbors



Summary: Broadcast Storm Problem

- Flooding is used in many protocols, such as Dynamic Source Routing (DSR)
- Problems associated with flooding
  - collisions
  - redundancy
- Collisions may be reduced by “jittering” (waiting for a random interval before propagating the flood)
- Redundancy may be reduced by selectively re-broadcasting packets from only a subset of the nodes

Routing Protocols

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

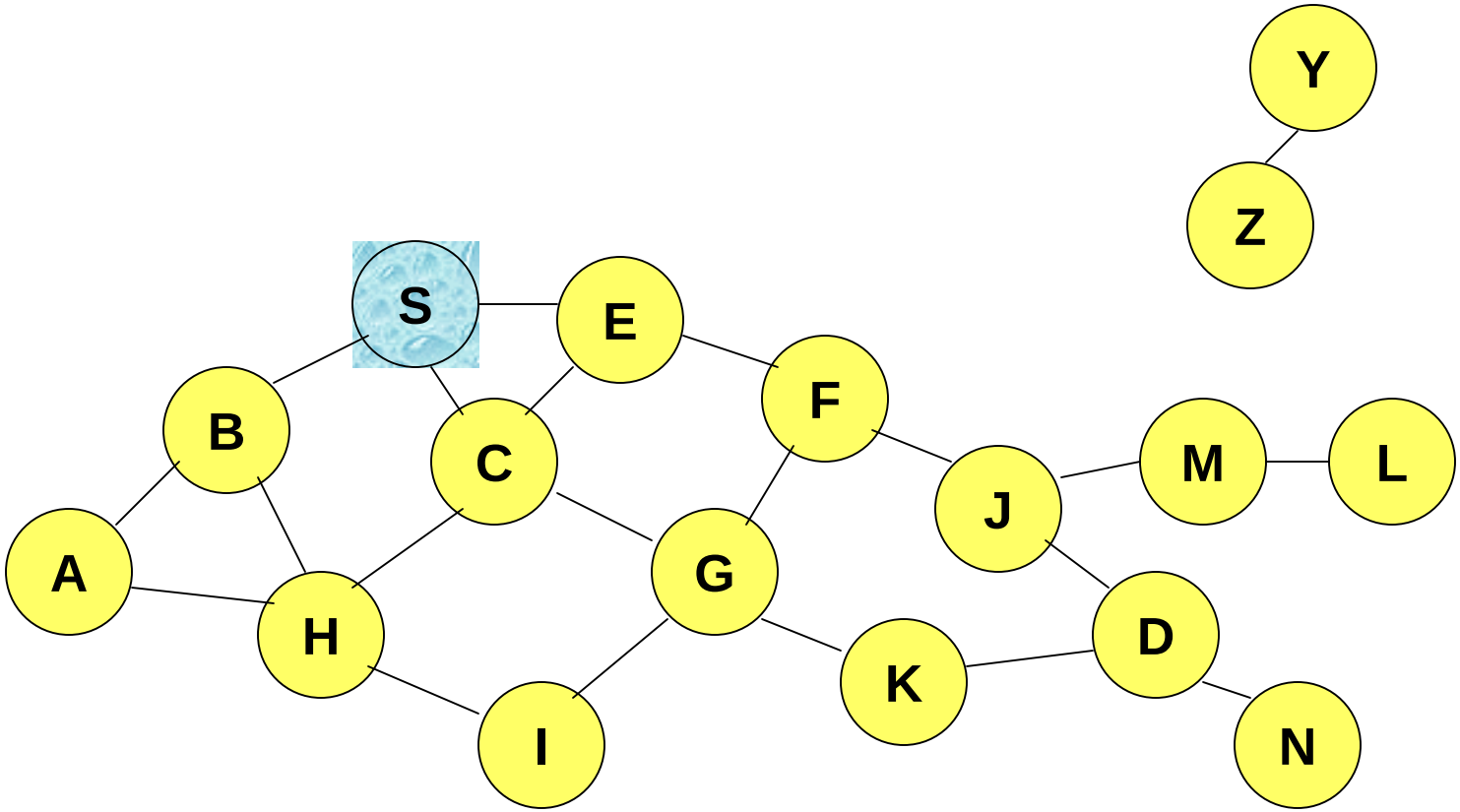
Trade-Off

- Latency of route discovery
  - Proactive protocols may have lower latency since routes are maintained at all times
  - Reactive protocols may 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
  - Reactive protocols may have lower overhead since routes are determined only if needed
  - Proactive protocols can (but not necessarily) result in higher overhead due to continuous route updating
- Which approach achieves a better trade-off depends on the traffic and mobility patterns

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

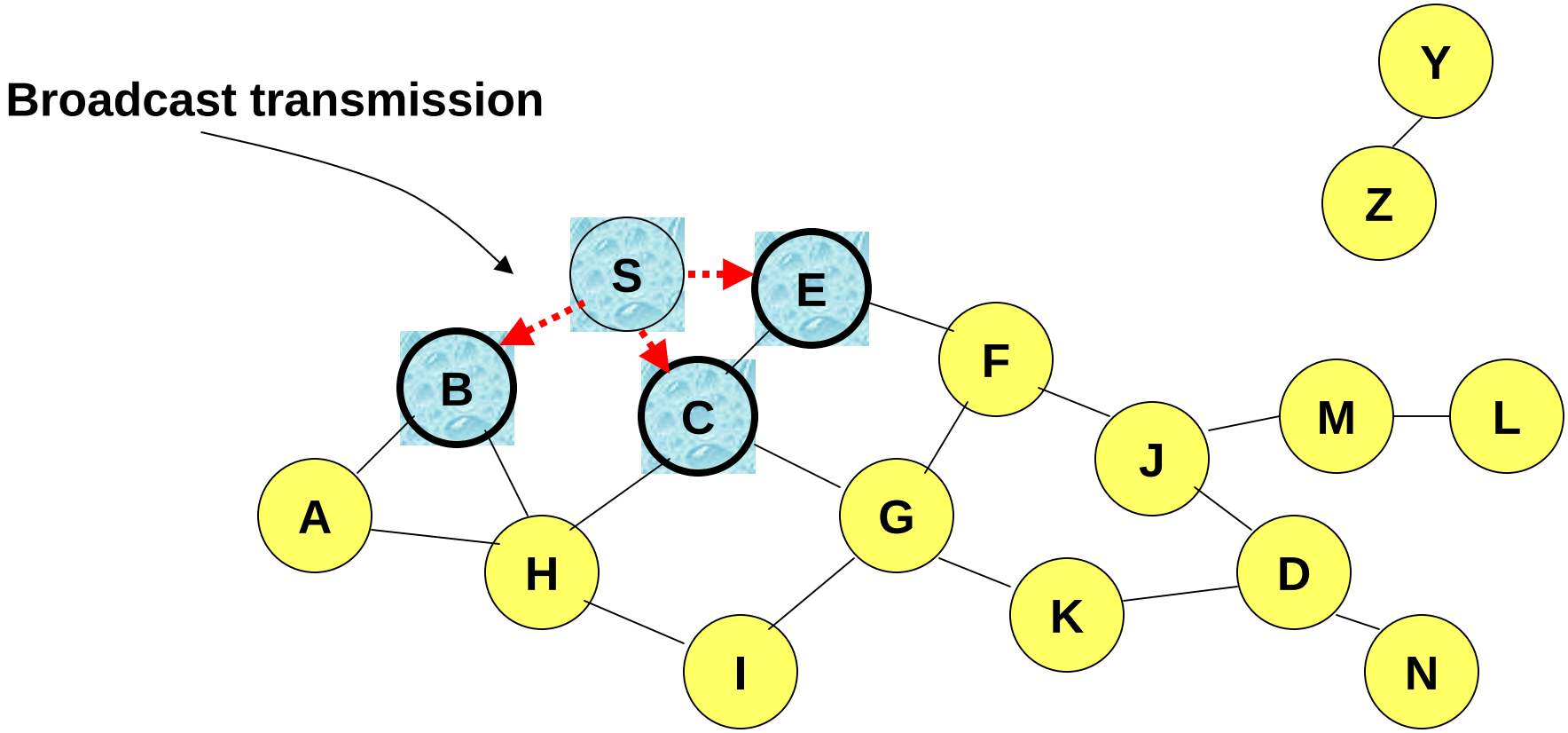
Flooding for Data Delivery



Represents a node that has received packet P

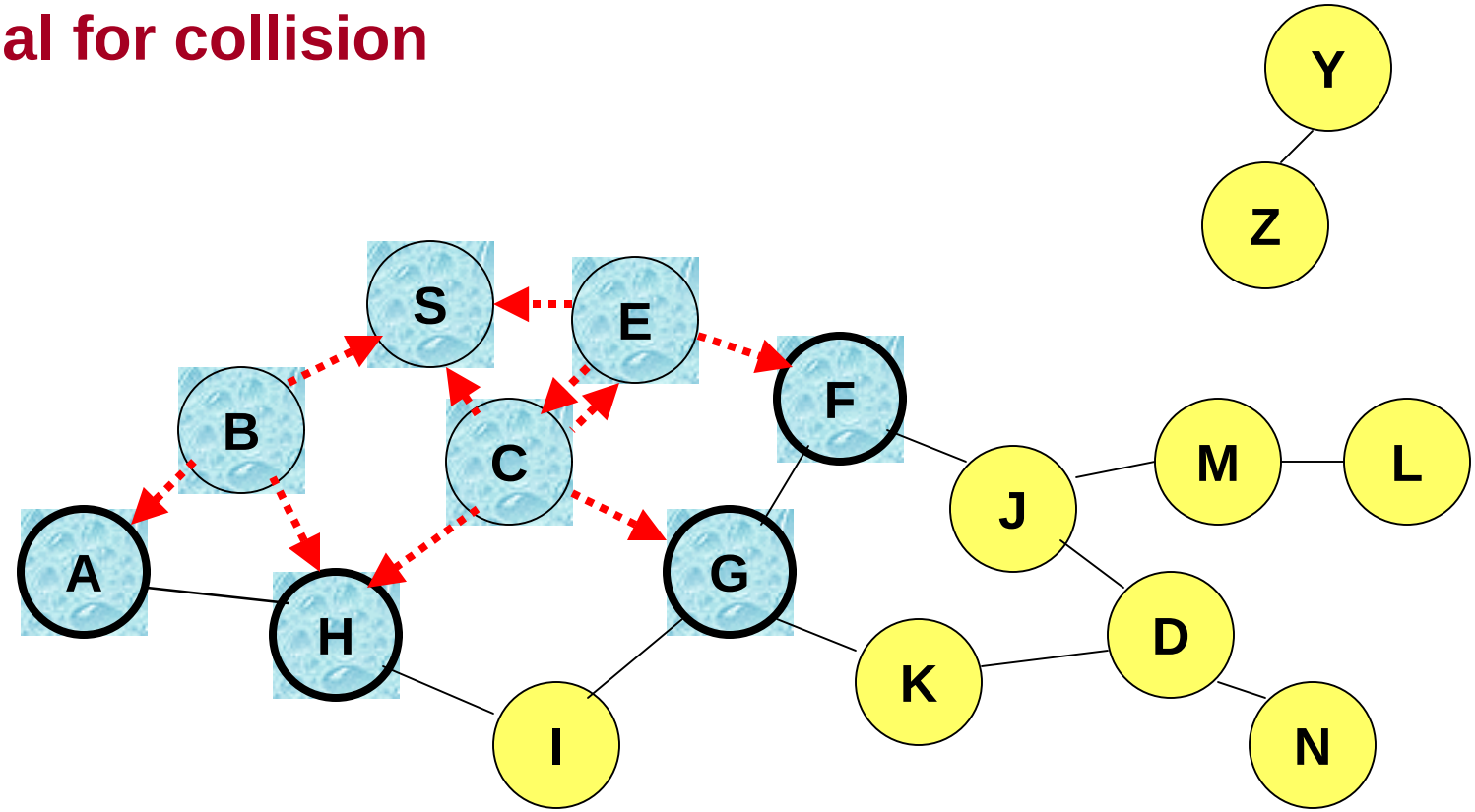
Represents that connected nodes are within each other's transmission range

Flooding for Data Delivery



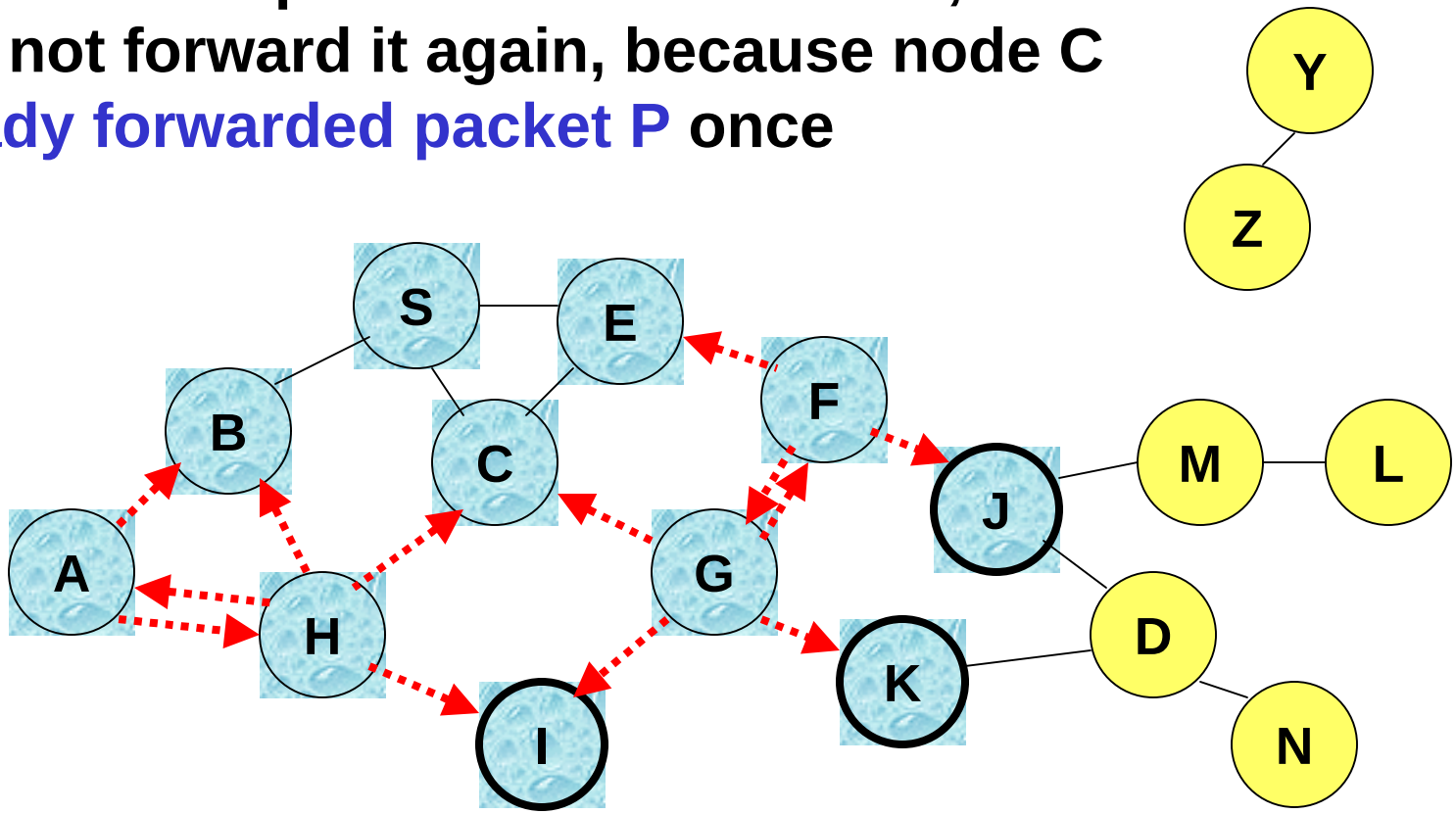
Flooding for Data Delivery

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



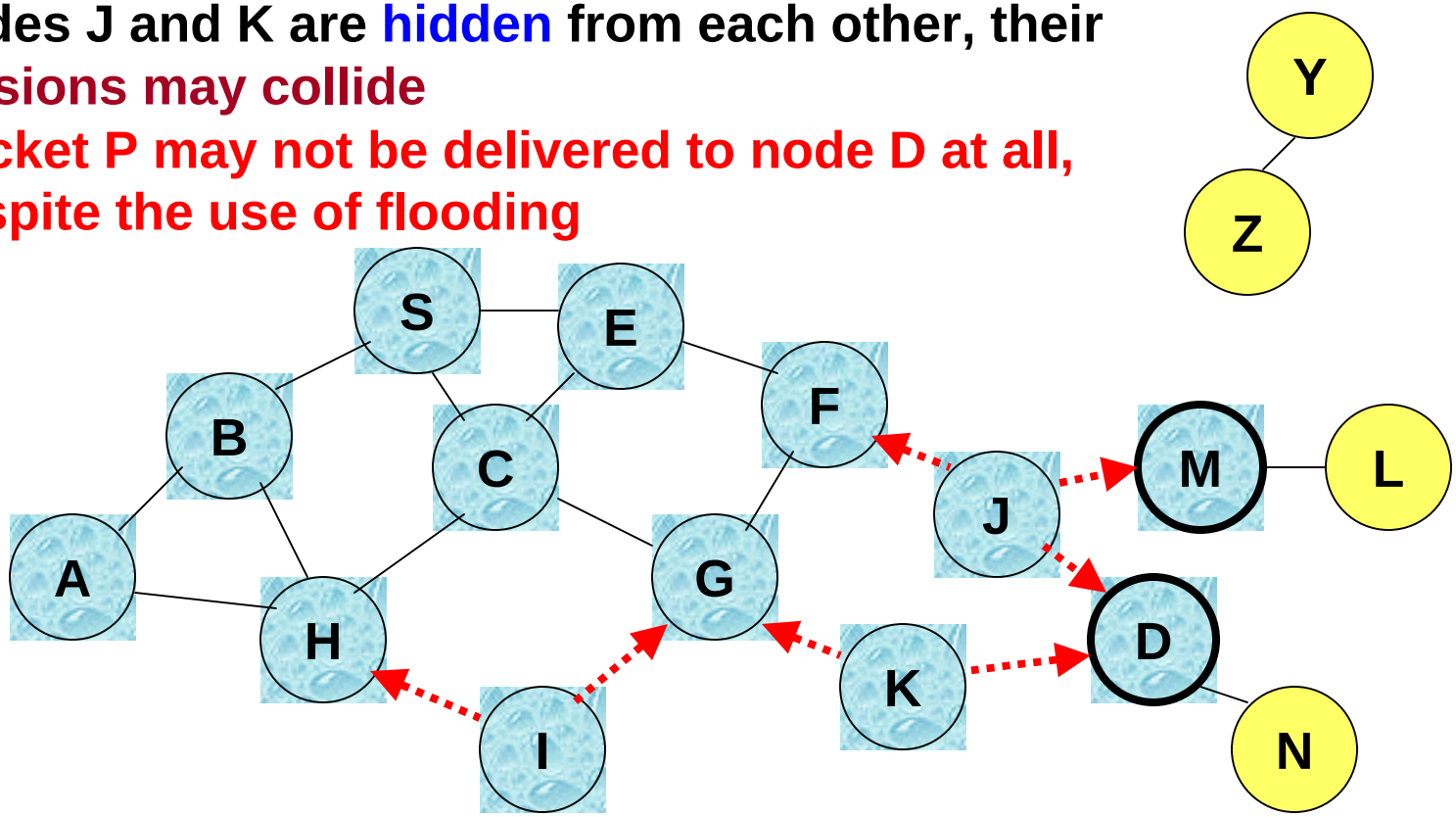
Flooding for Data Delivery

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



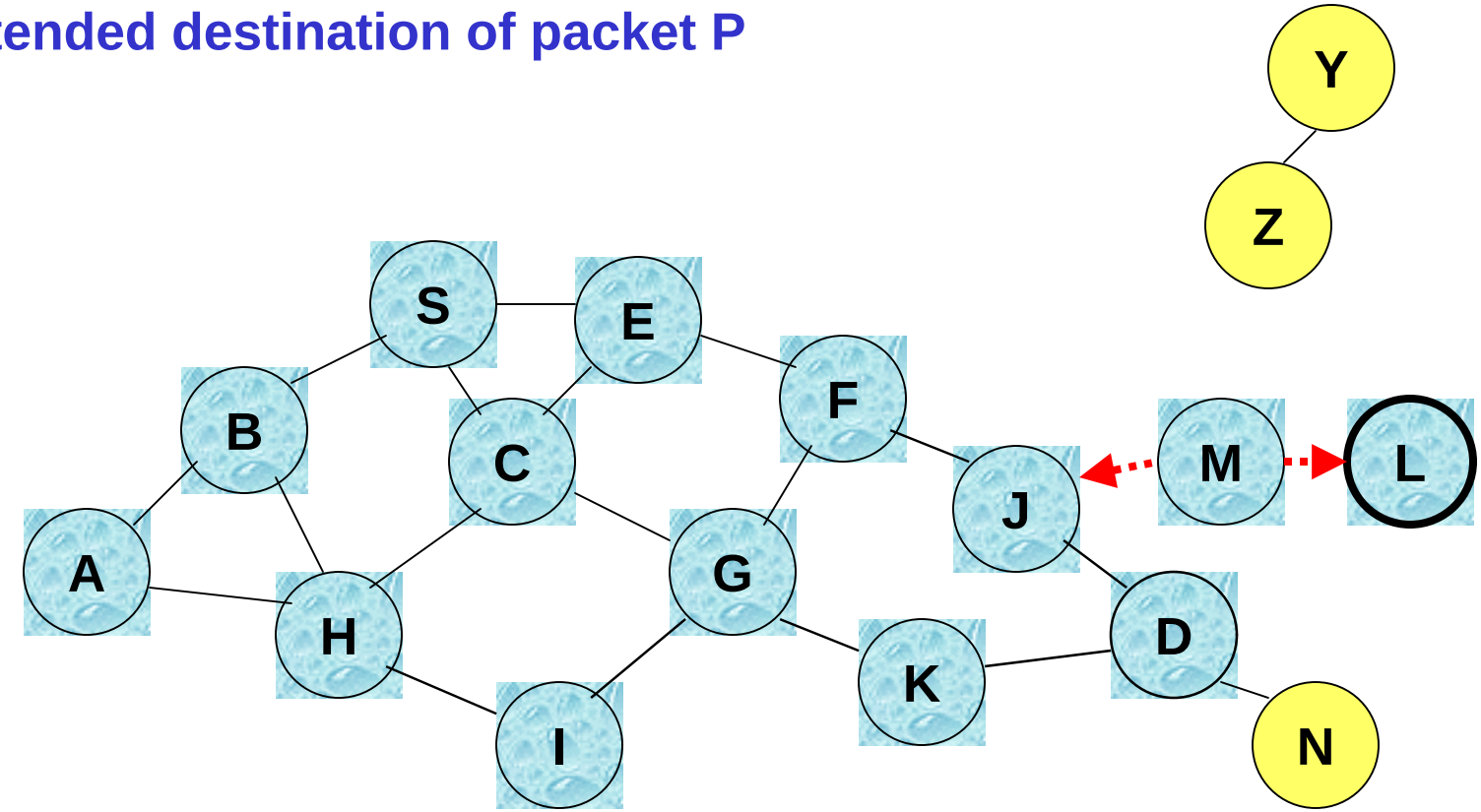
Flooding for Data Delivery

- 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



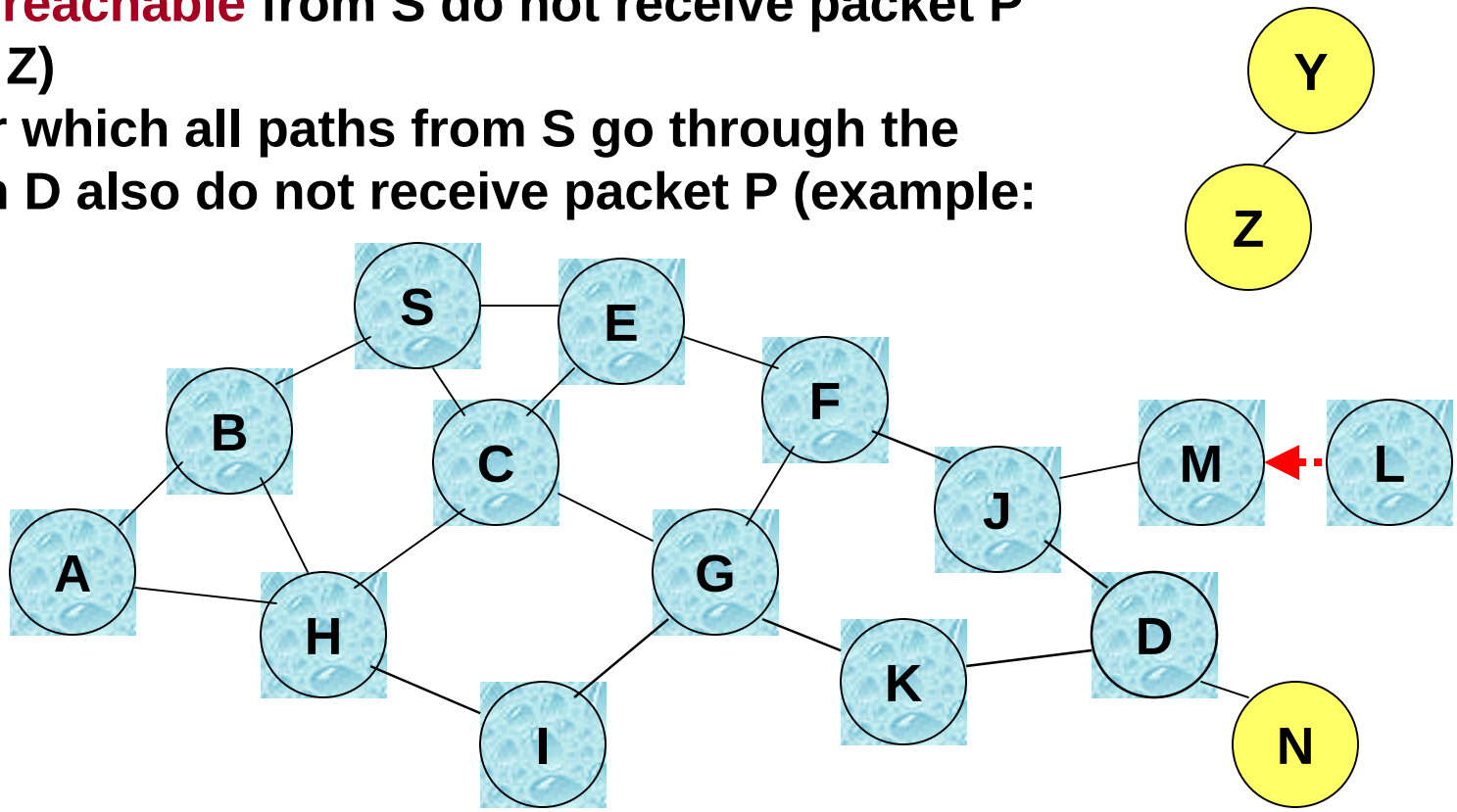
Flooding for Data Delivery

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



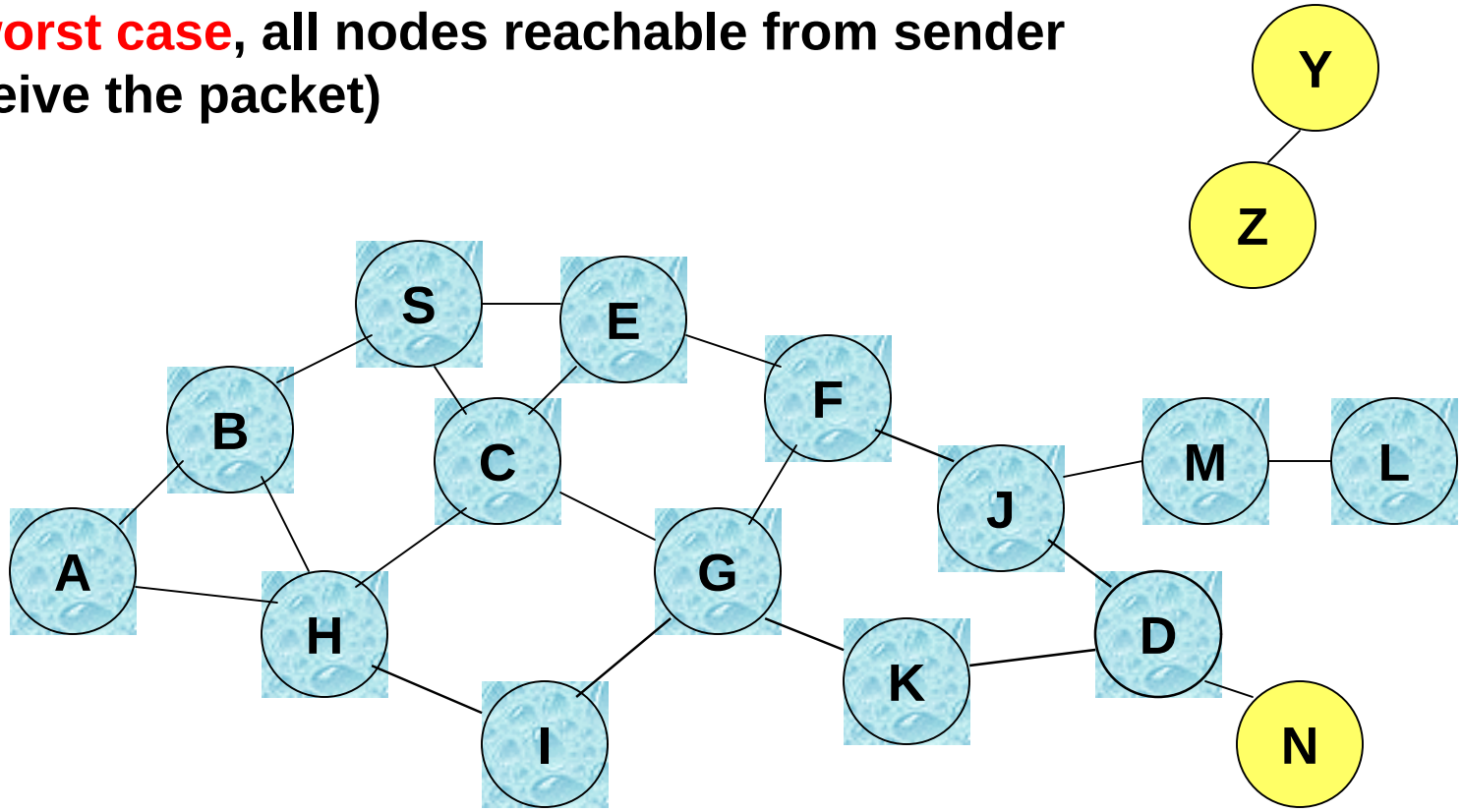
Flooding for Data Delivery

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



Flooding for Data Delivery

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



Flooding for Data Delivery: Advantages

- Simplicity
- May be more efficient than other protocols when rate of information transmission is low enough that the overhead of explicit route discovery/maintenance incurred by other protocols is relatively higher
  - this scenario may occur, for instance, when nodes transmit small data packets relatively infrequently, and many topology changes occur between consecutive packet transmissions
- Potentially higher reliability of data delivery
  - Because packets may be delivered to the destination on multiple paths

Flooding for Data Delivery: 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
  - Flooding uses broadcasting -- hard to implement reliable broadcast delivery without significantly increasing overhead
    - Broadcasting in IEEE 802.11 MAC is unreliable
  - In our example, nodes J and K may transmit to node D simultaneously, resulting in loss of the packet
    - in this case, destination would not receive the packet at all

Flooding of Control Packets

- Many protocols perform (potentially *limited*) flooding of control packets, instead of data packets
- The control packets are used to discover routes
- Discovered routes are subsequently used to send data packet(s)
- Overhead of control packet flooding is amortized over data packets transmitted between consecutive control packet floods

CMU Implementation: Lessons Learned

- "Wireless propagation is not what you would expect" [Maltz99]
  - Straight flat areas with line-of-sight connectivity had worst error rates

- "Bystanders will think you are nuts" [Maltz99]
  - If you are planning experimental studies in the streets, it may be useful to let police and security guards know in advance what you are up to

Implementation Issues:

- Where to Implement Ad Hoc Routing
  - Link layer
  - Network layer
  - Application layer

Implementation Issues:

- Address Assignment
  - Restrict all nodes within a given ad hoc network to belong to the same subnet
    - Routing within the subnet using ad hoc routing protocol
    - Routing to/from outside the subnet using standard internet routing
  - Nodes may be given random addresses
    - Routing to/from outside world becomes difficult unless Mobile IP is used

Implementation Issues:

- Address Assignment
  - How to assign the addresses ?
- Non-random address assignment:
  - DHCP for ad hoc network ?
- Random assignment
  - What happens if two nodes get the same address ?
  - Duplicate address detection needed
  - One procedure for detecting duplicates within a connected component: When a node picks address A, it first performs a few route discoveries for destination A. If no route reply is received, then address A is assumed to be unique.

Implementation Issues:

- Security
  - How can I trust you to forward my packets without tampering?
    - Need to be able to detect tampering
  - How do I know you are what you claim to be ?
    - Authentication issues
    - Hard to guarantee access to a certification authority

Implementation Issues

- Can we make any guarantees on performance?
  - When using a non-licensed band, difficult to provide hard guarantees, since others may be using the same band

- Must use an licensed channel to attempt to make any guarantees

Implementation Issues

- Only some issues have been addresses in existing implementations
- Security issues typically ignored
- Address assignment issue also has not received sufficient attention

Routing In Bluetooth

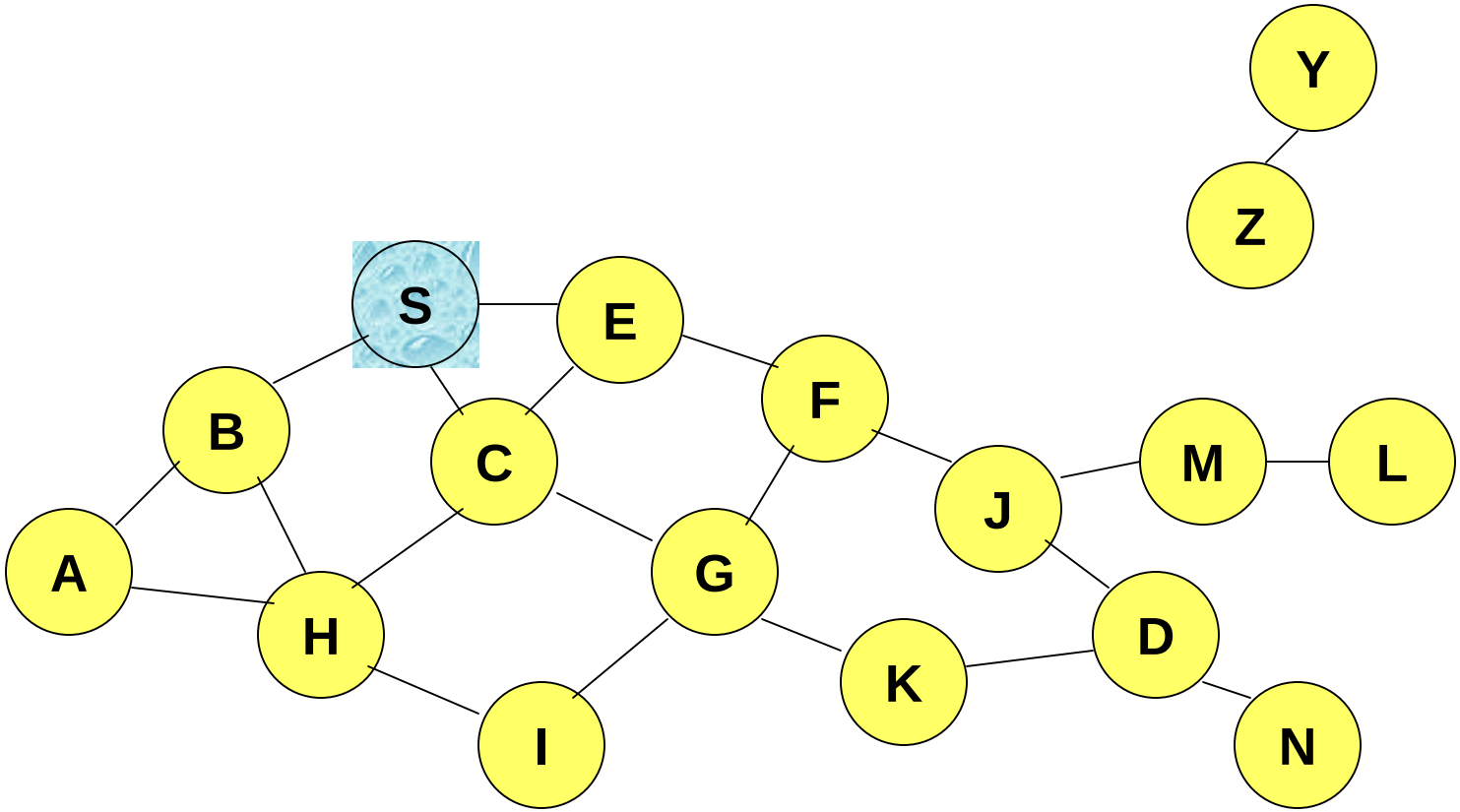
- Ad hoc routing protocols needed to route between multiple piconets

- Existing protocols may need to be adapted for Bluetooth
  - For instance, not all nodes within transmission range of node X will hear node X
    - Only nodes which belong to node X's current piconet can hear the transmission from X
  - Flooding-based schemes need to take this limitation into account

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 node appends own identifier when forwarding RREQ

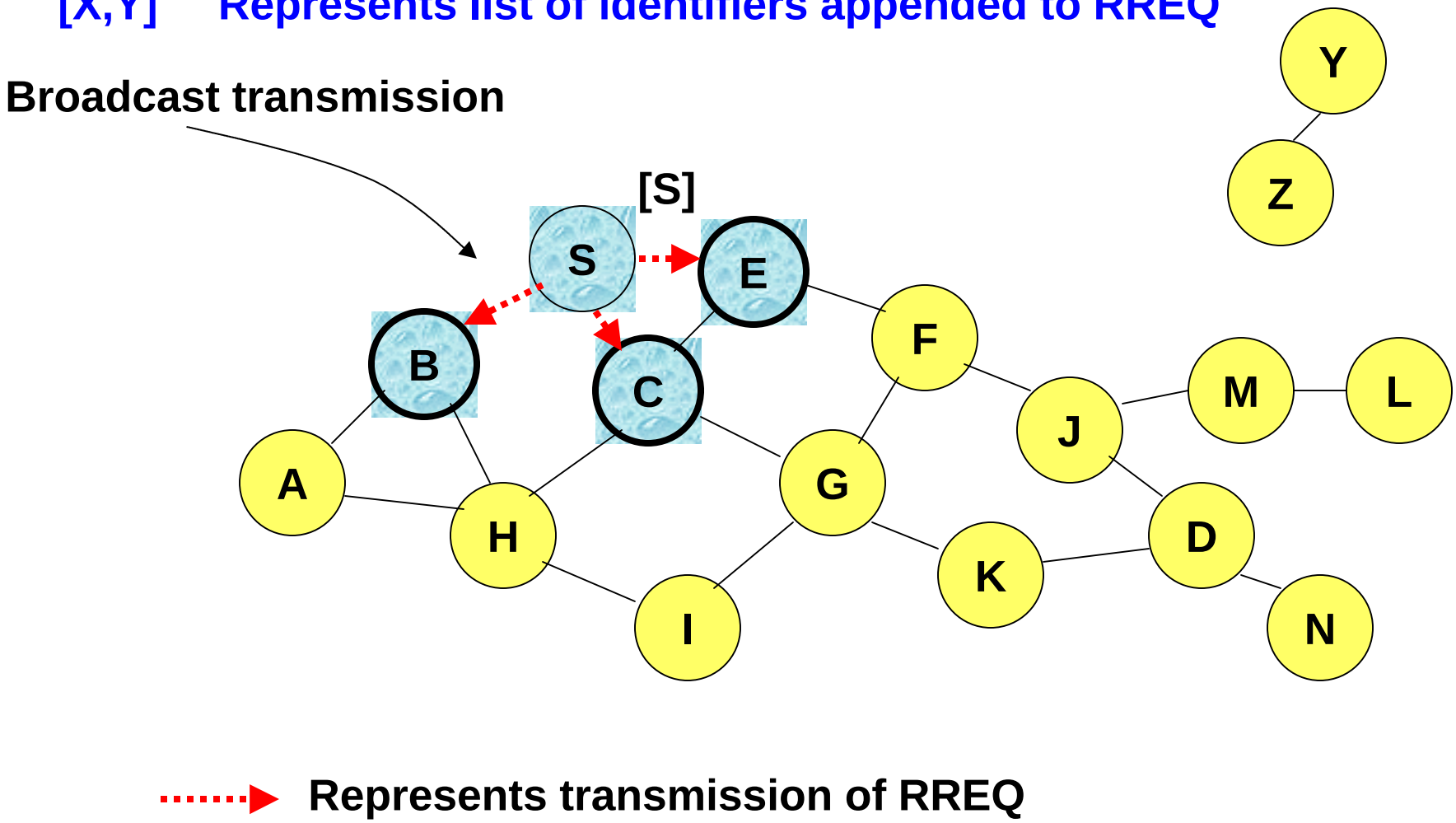
Route Discovery in DSR



Represents a node that has received RREQ for D from S

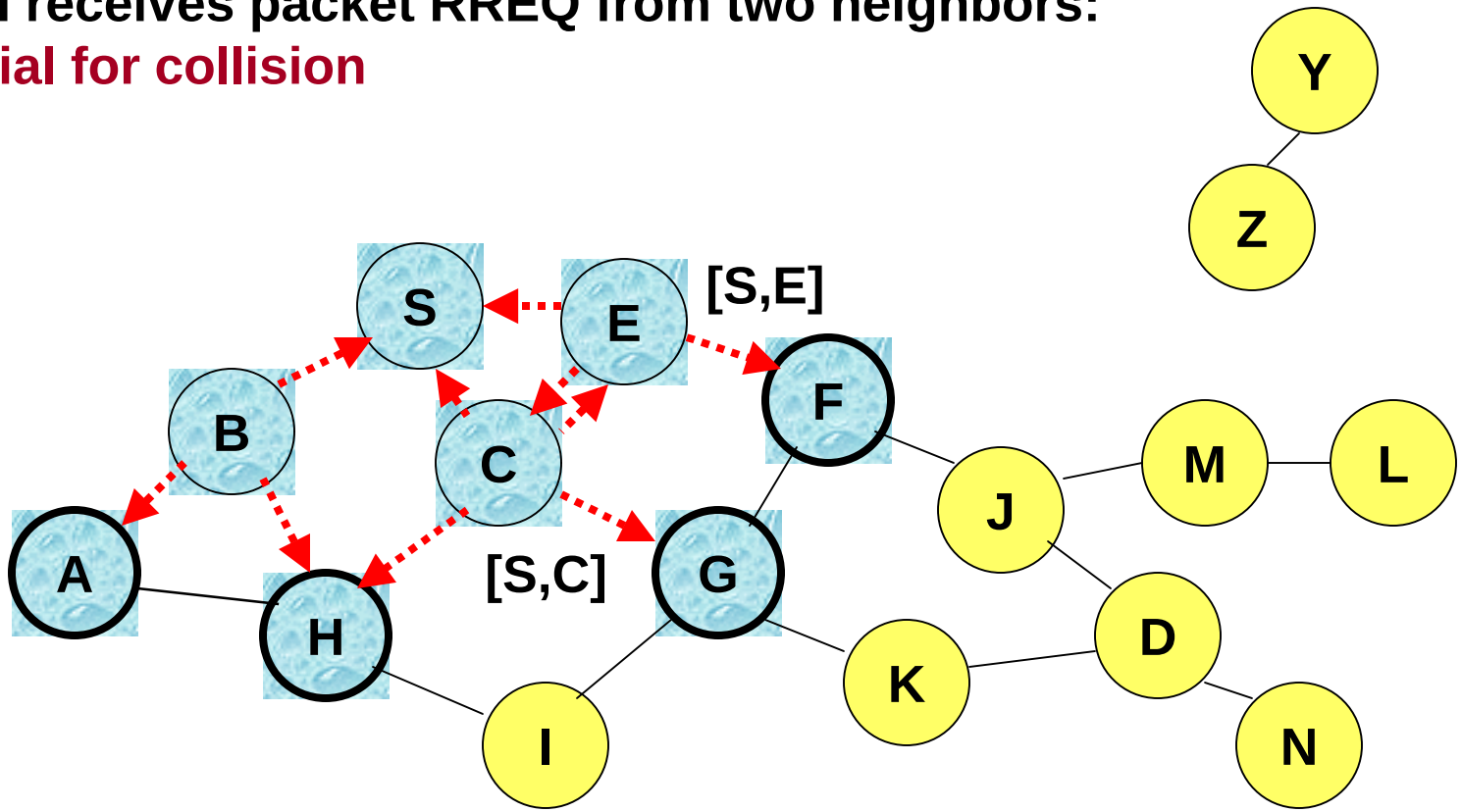
Route Discovery in DSR

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



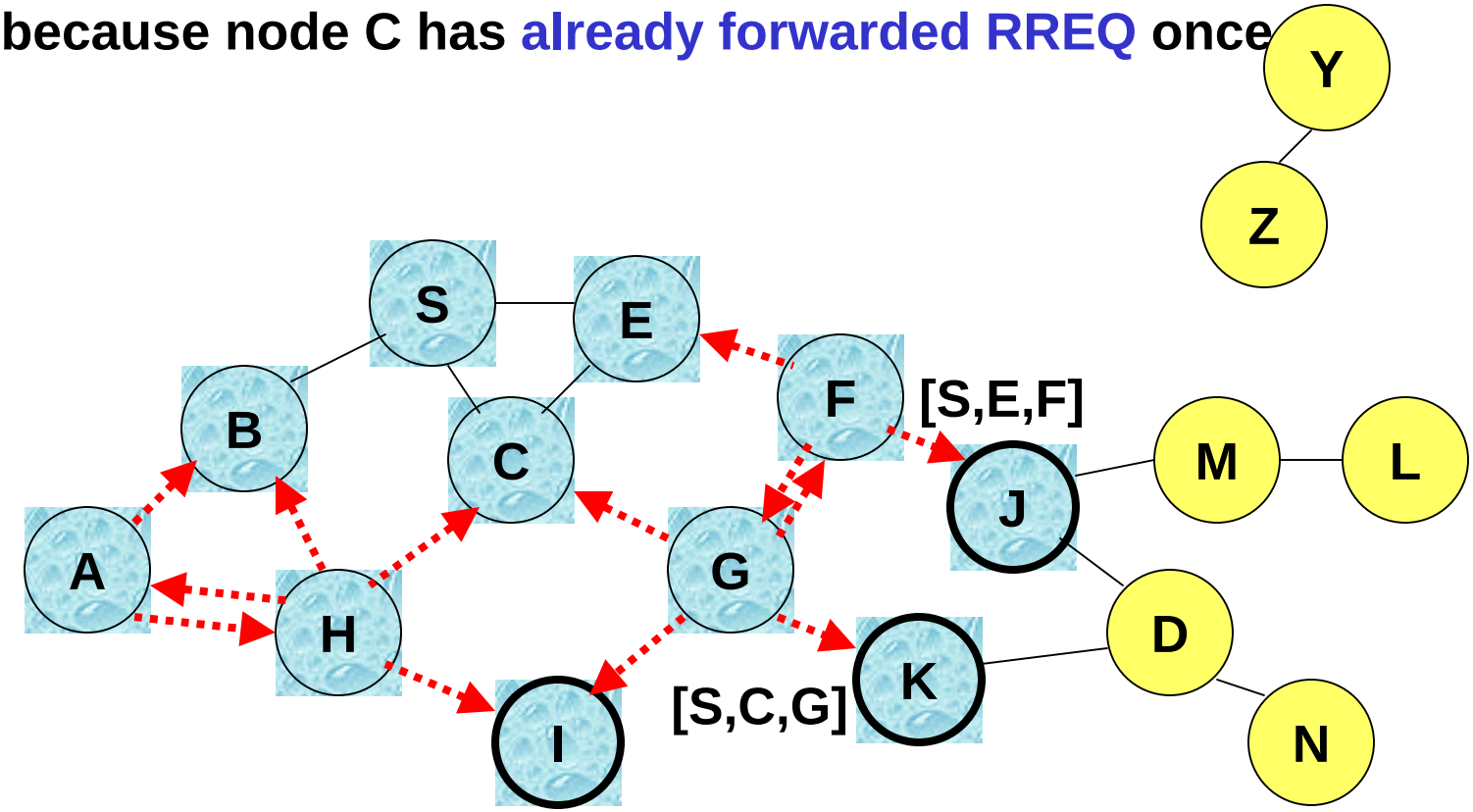
Route Discovery in DSR

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



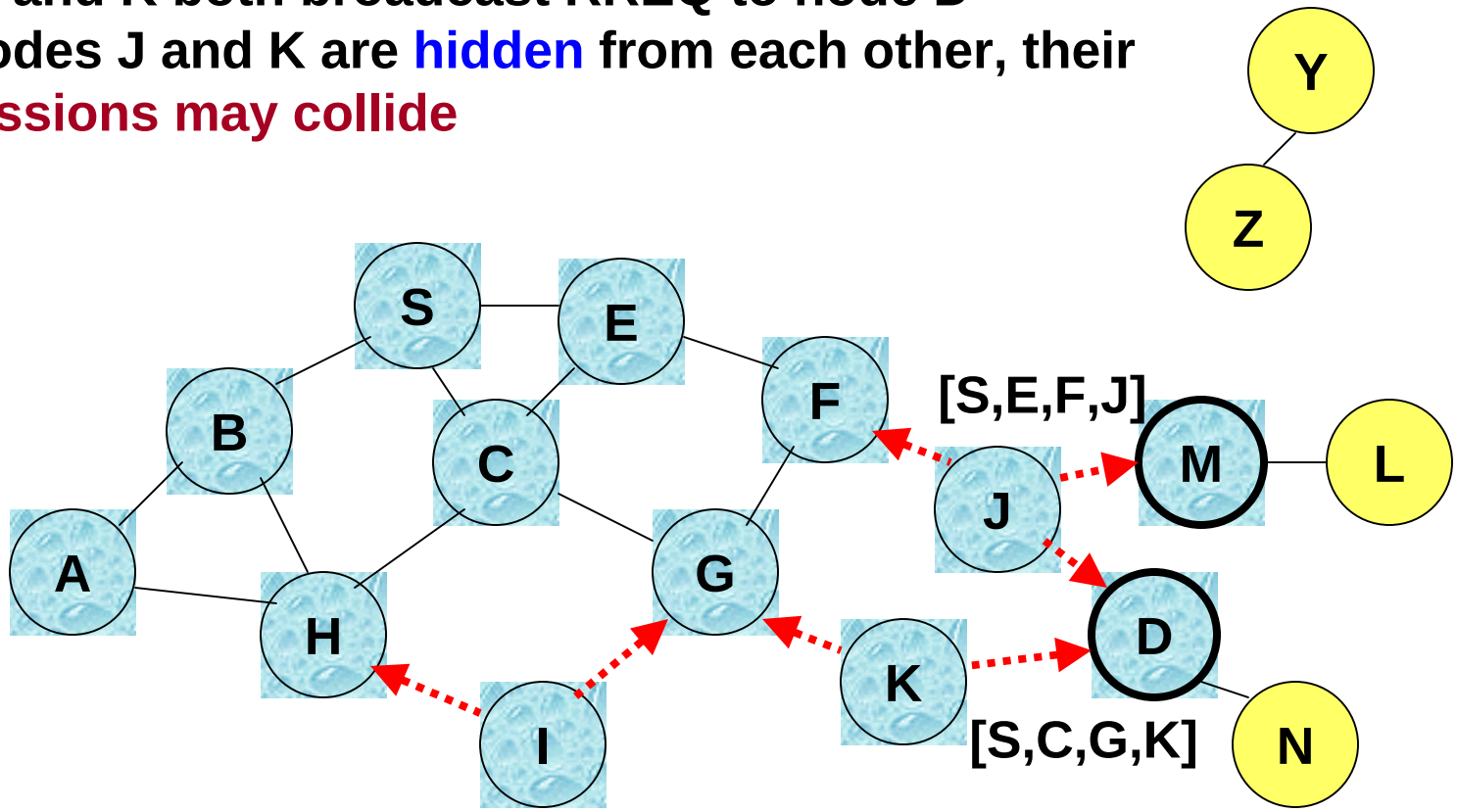
Route Discovery in DSR

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



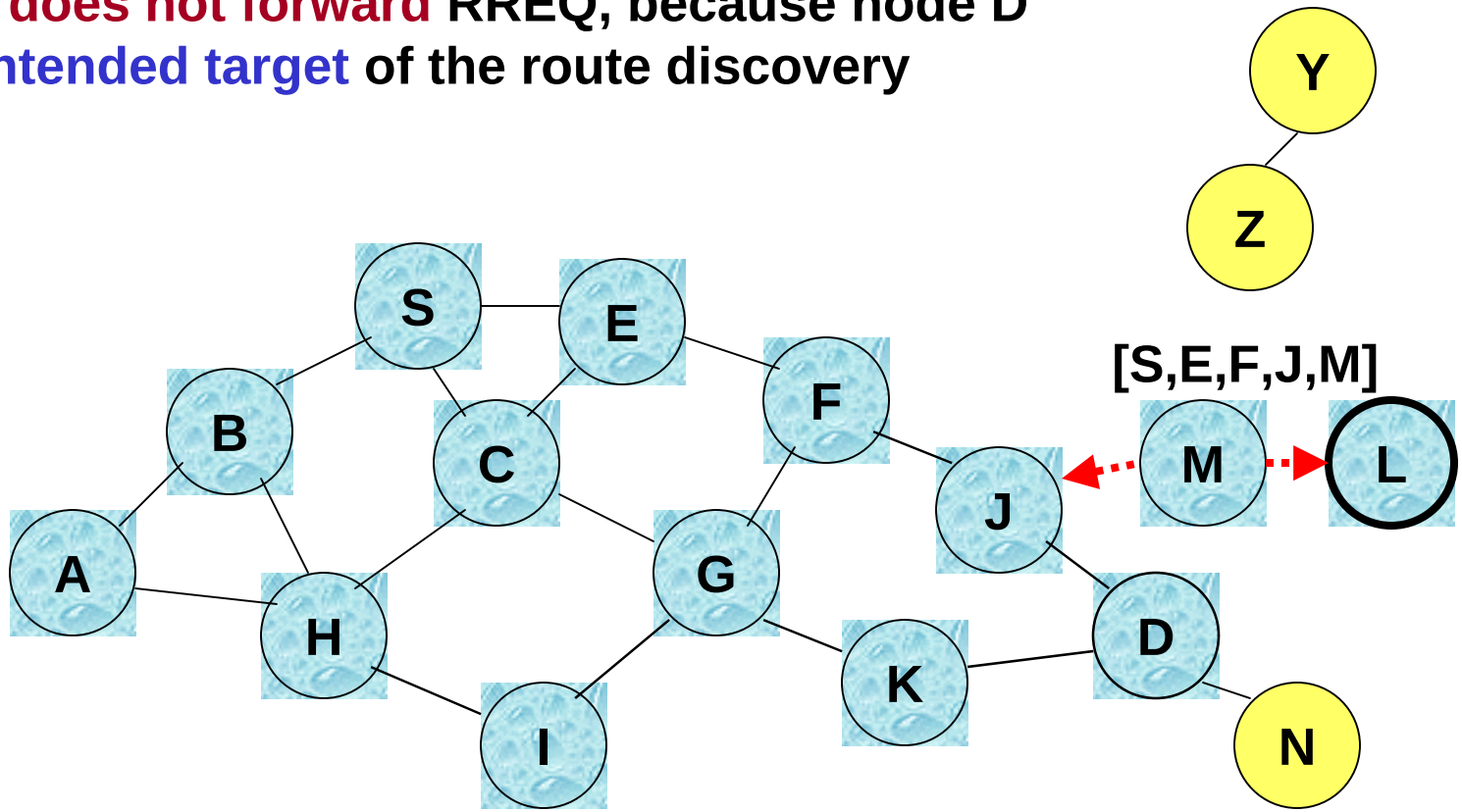
Route Discovery in DSR

- 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**

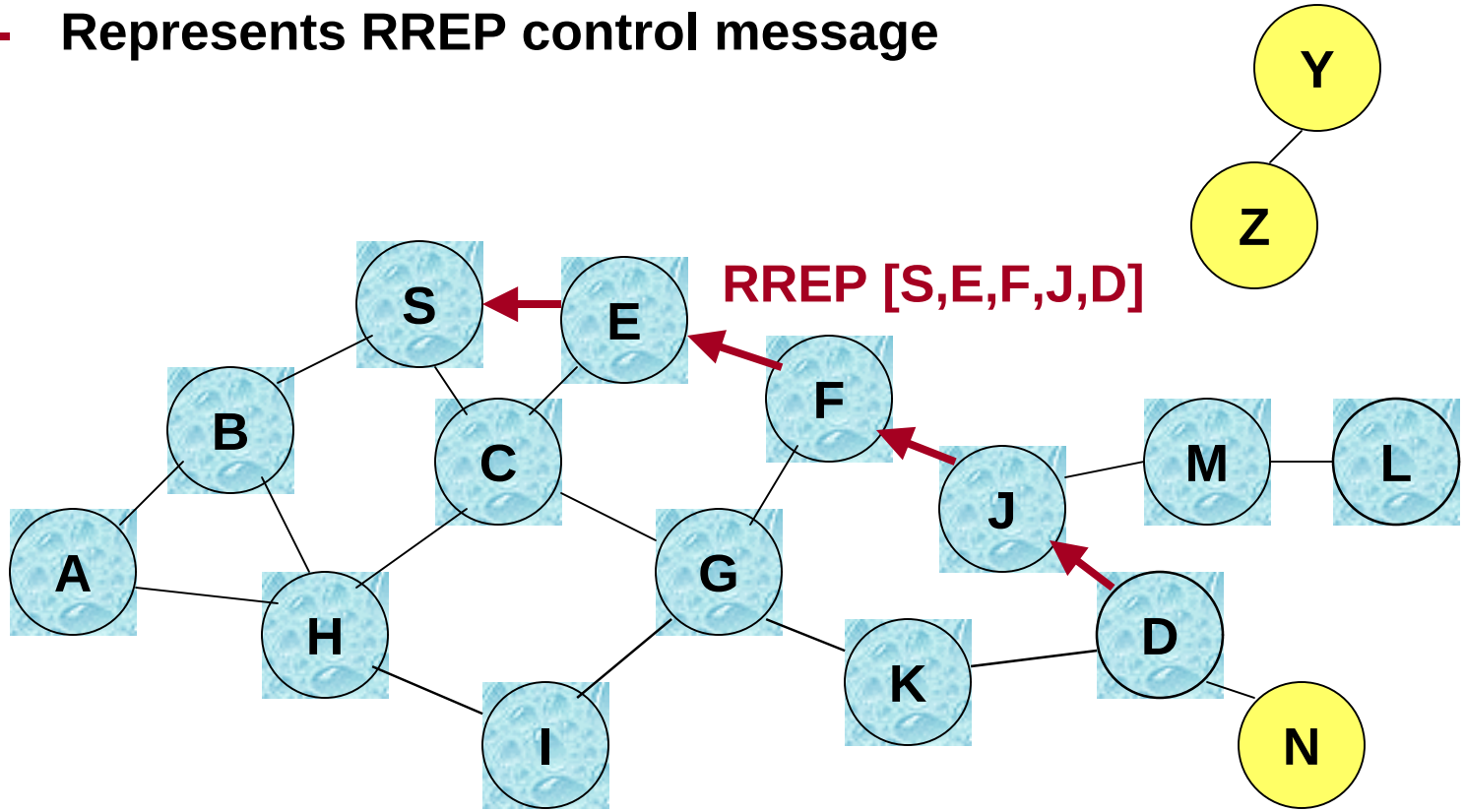


Route Discovery in DSR

- 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

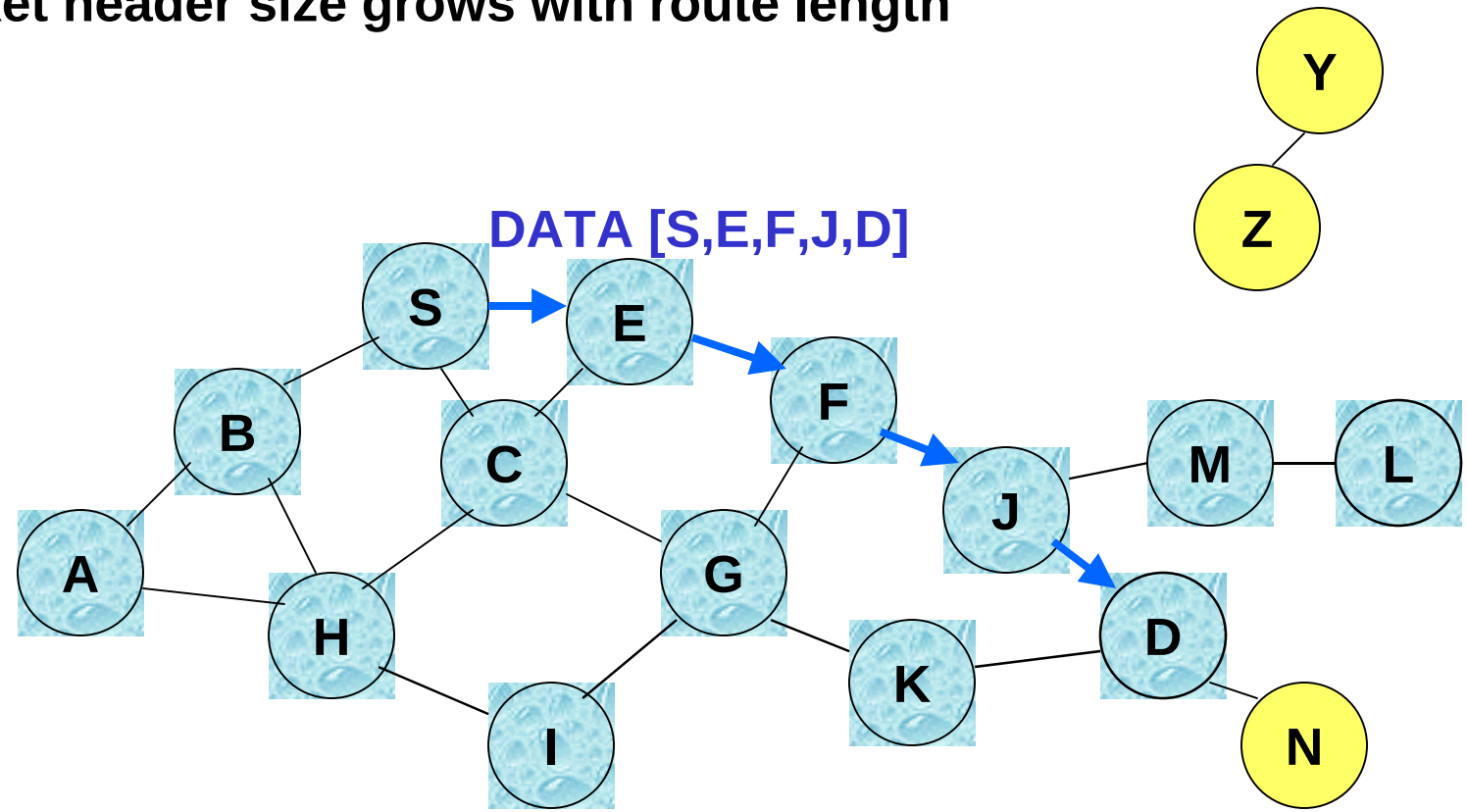
- Route Reply can be sent by reversing the route in Route Request (RREQ) only if links are guaranteed to be bi-directional
  - To ensure this, RREQ should be forwarded only if it received on a link that is known to be bi-directional
- If unidirectional (asymmetric) links are allowed, then RREP may need 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



When to Perform a Route Discovery

- When node S wants to send data to node D, but does not know a valid route node D

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

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
  - Reply storm may be eased by preventing a node from sending RREP if it hears another RREP with a shorter route

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.

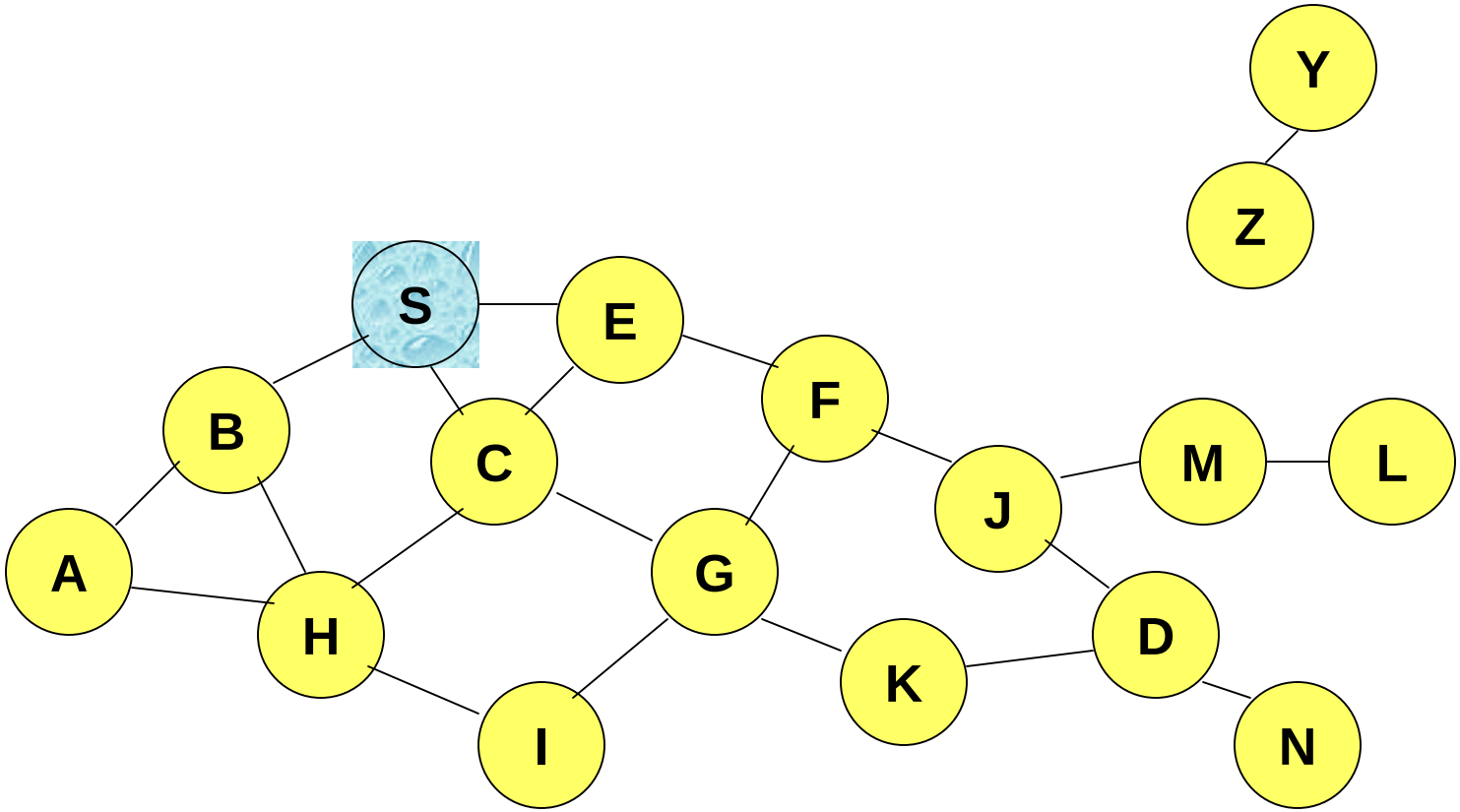
Ad Hoc On-Demand Distance Vector (AODV)

- 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

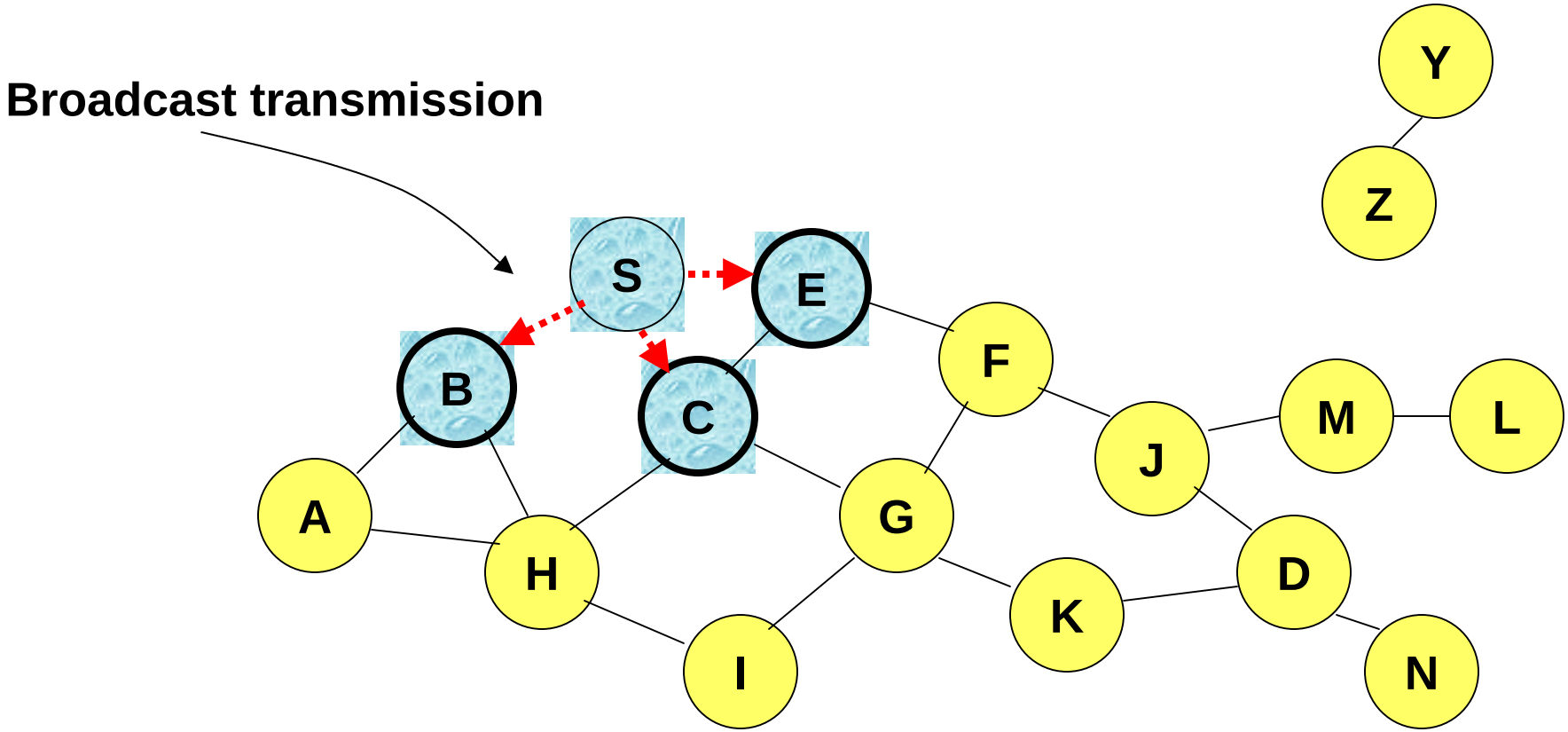
- Route Requests (RREQ) are forwarded in a manner similar to DSR
- When a node re-broadcasts a Route Request, it sets up a reverse path pointing towards the source
  - AODV assumes symmetric (bi-directional) links
- When the intended destination receives a Route Request, it replies by sending a Route Reply
- Route Reply travels along the reverse path set-up when Route Request is forwarded

Route Requests in AODV

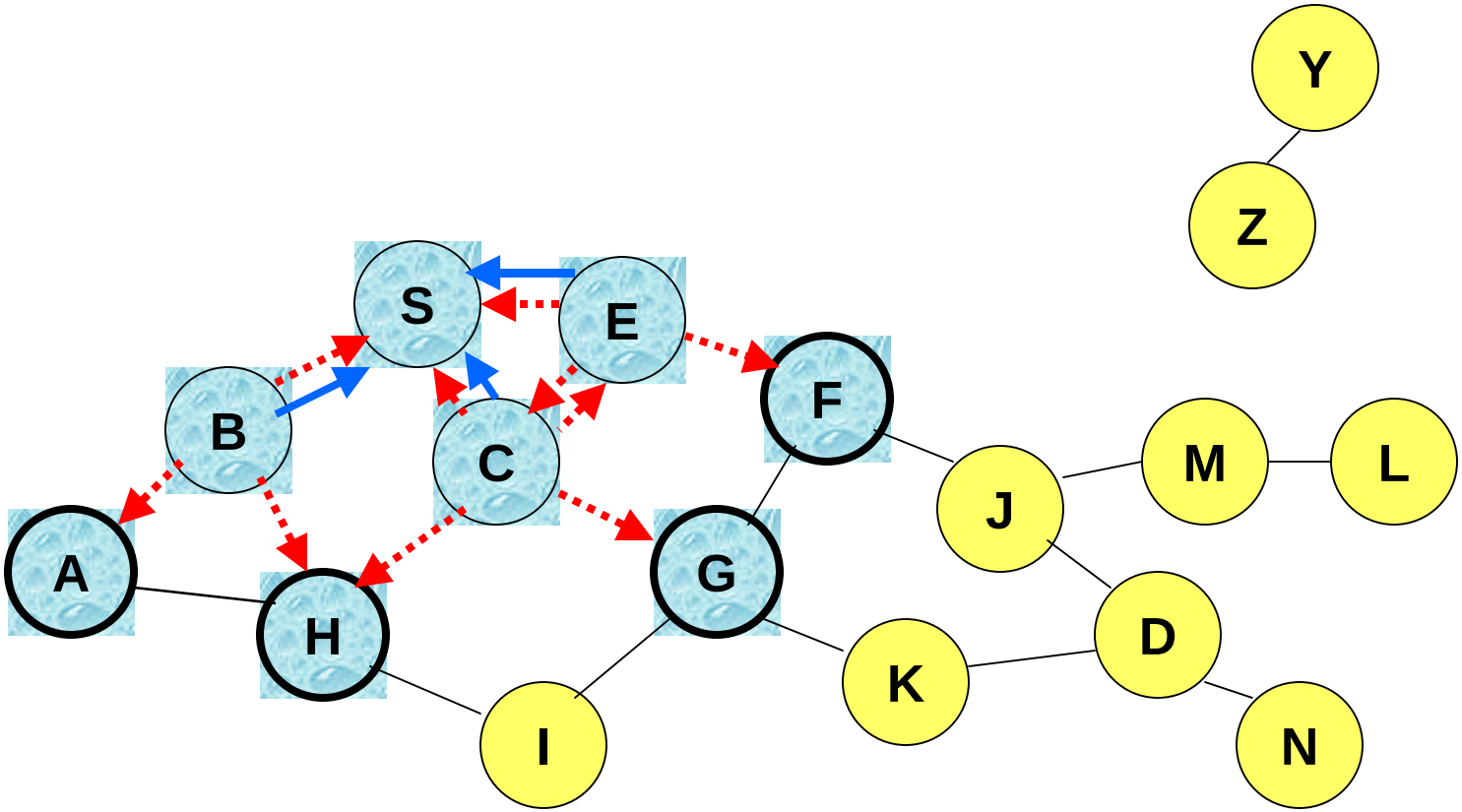


Represents a node that has received RREQ for D from S

Route Requests in AODV

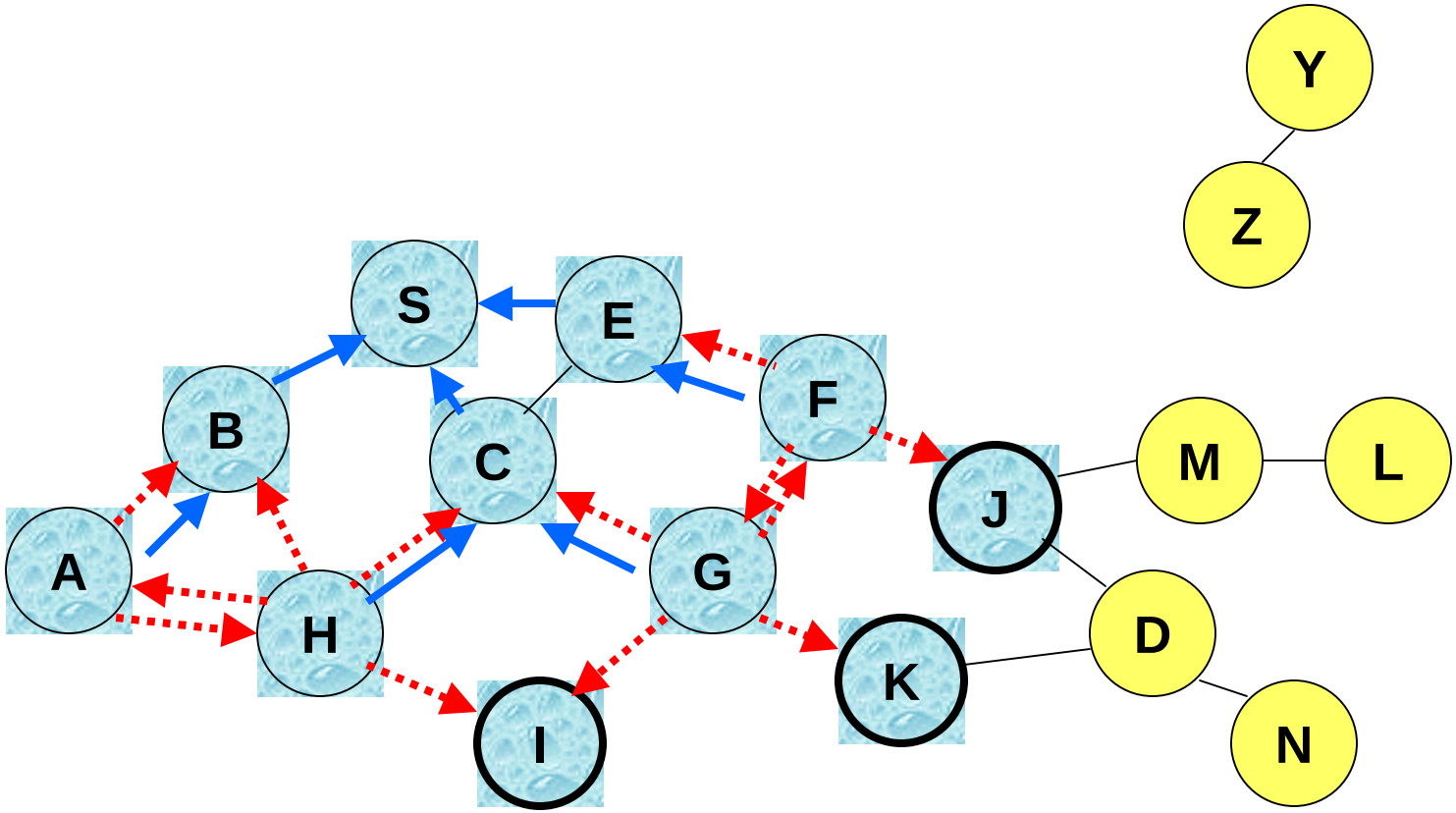


Route Requests in AODV



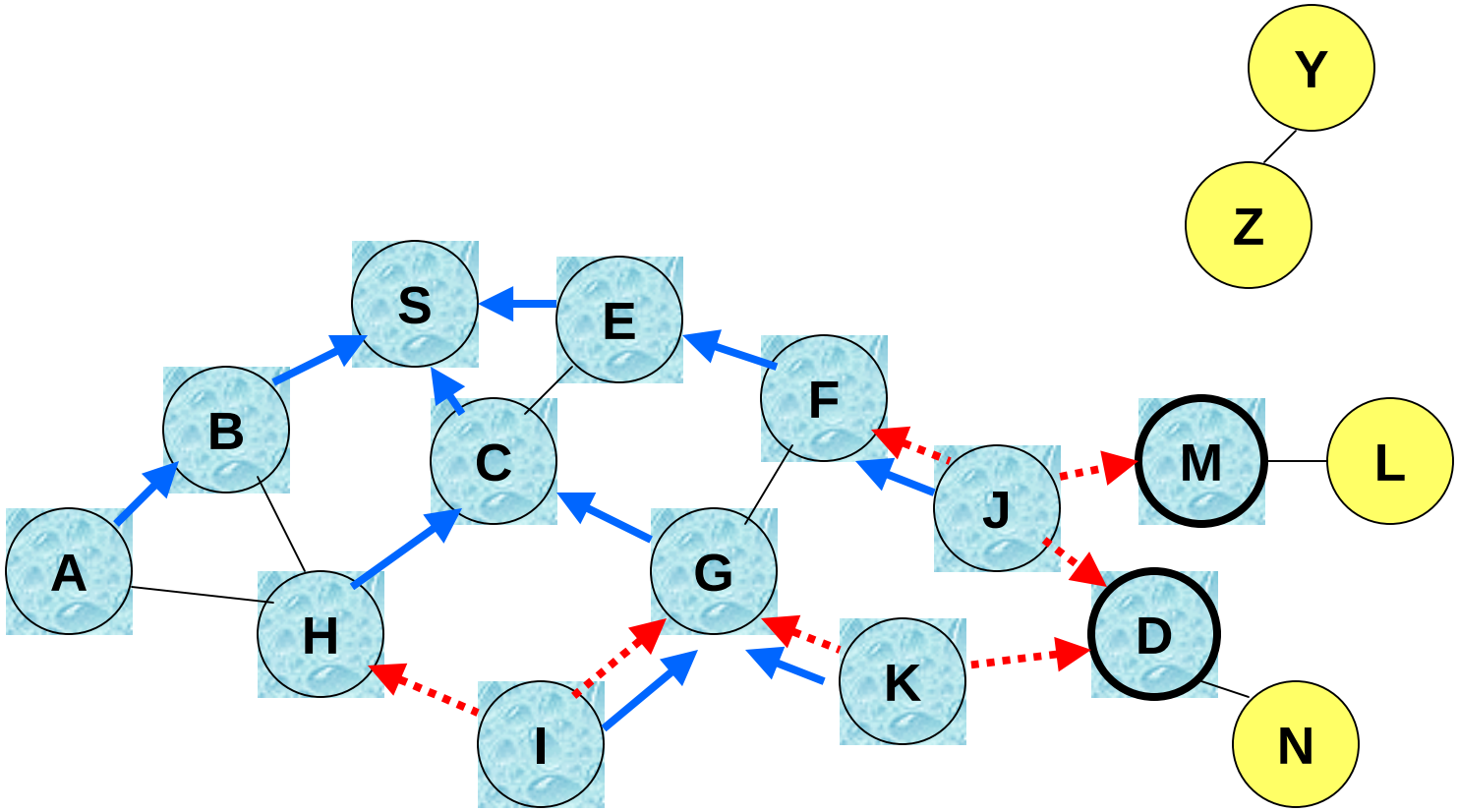
Represents links on Reverse Path

Reverse Path Setup in AODV



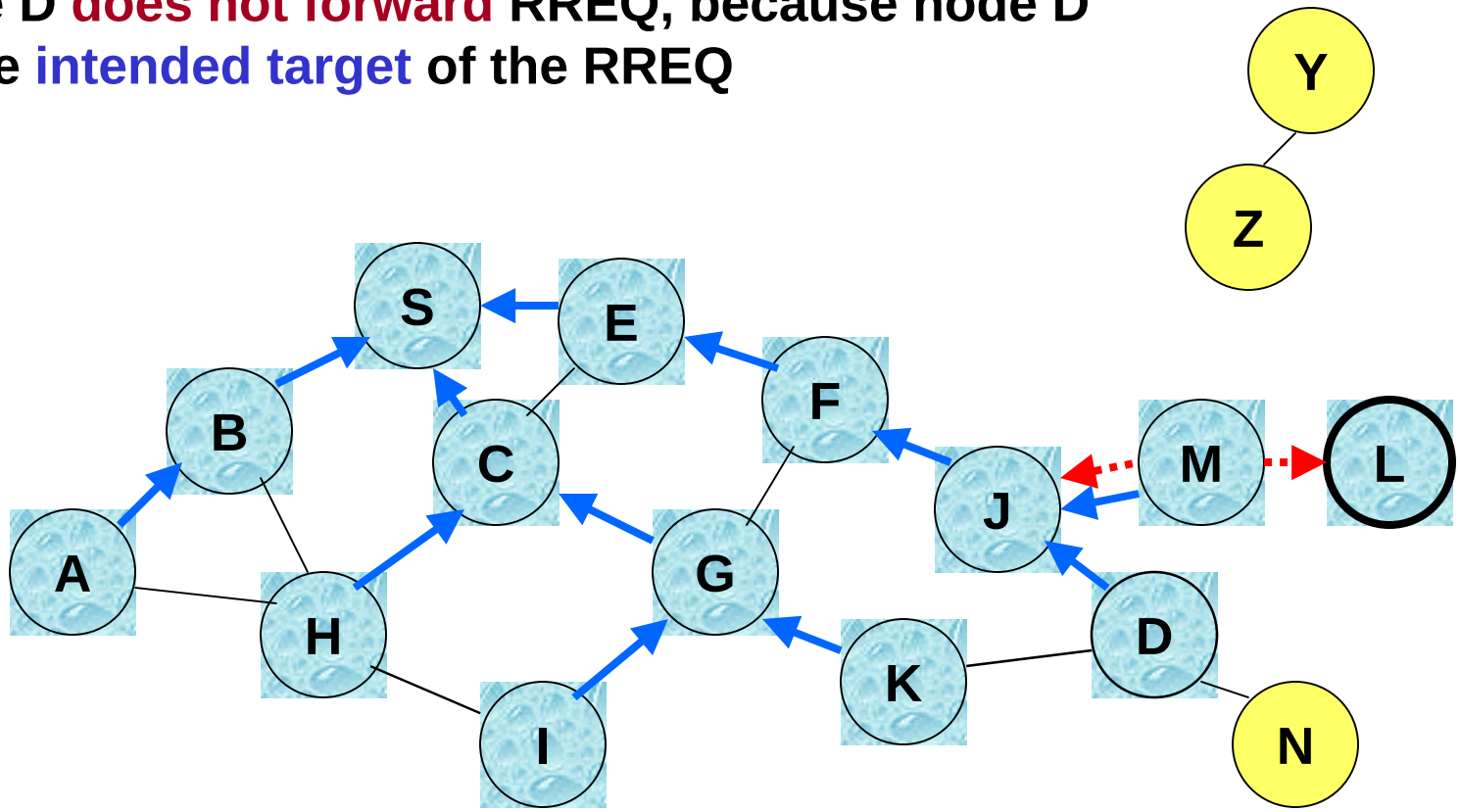
- Node C receives RREQ from G and H, but does not forward it again, because node C has already forwarded RREQ once

Reverse Path Setup in AODV

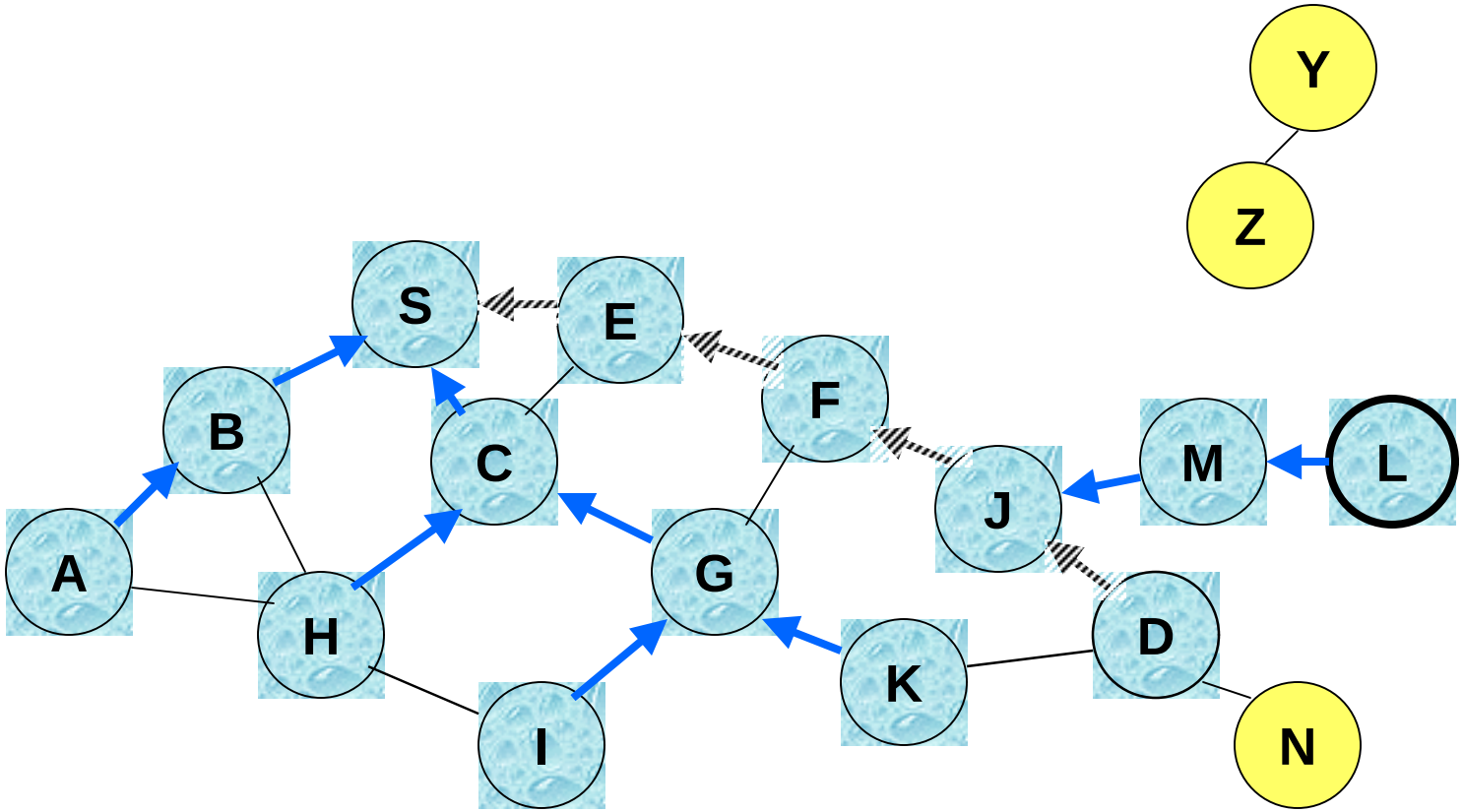


Reverse Path Setup in AODV

- **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

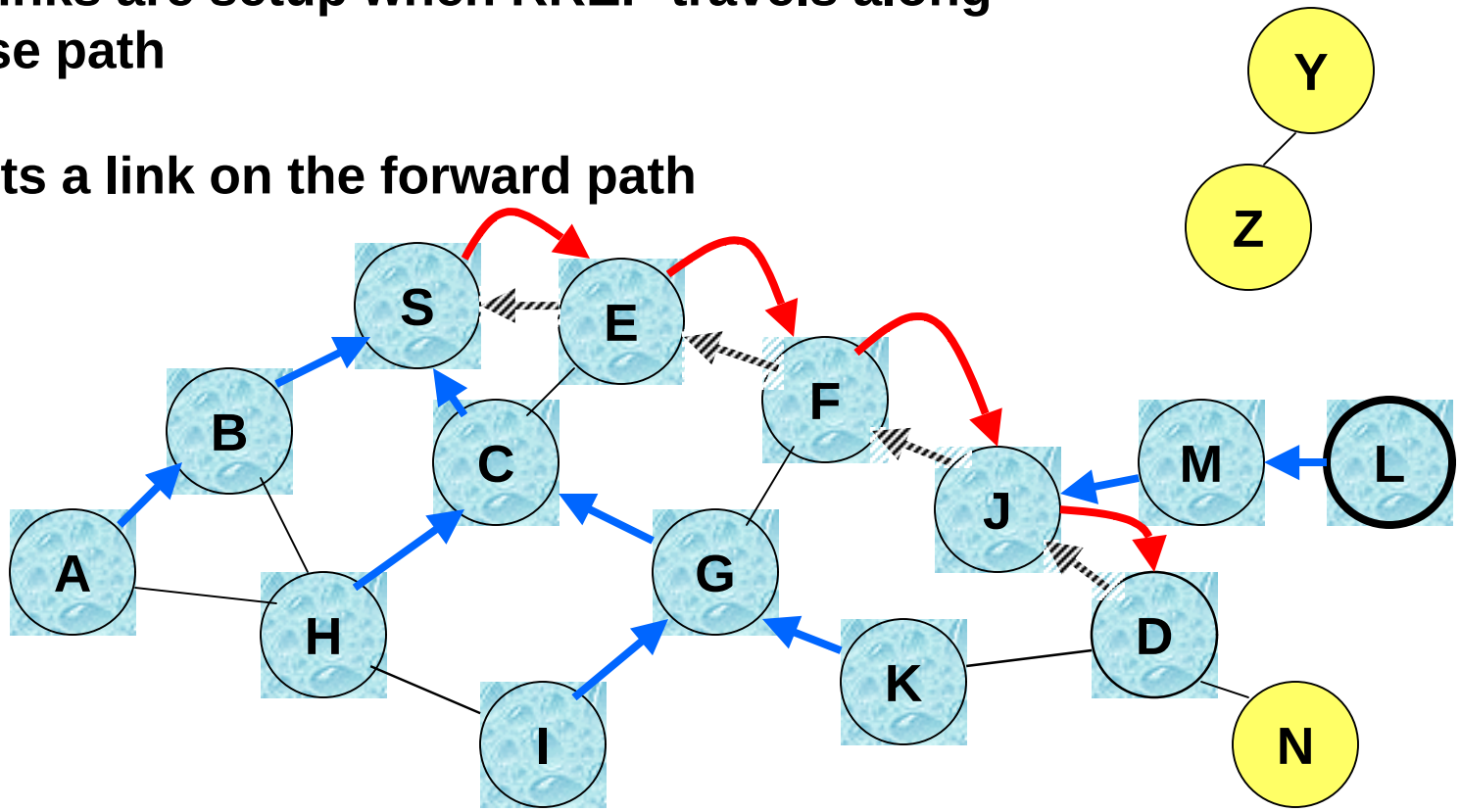
Route Reply in AODV

- 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

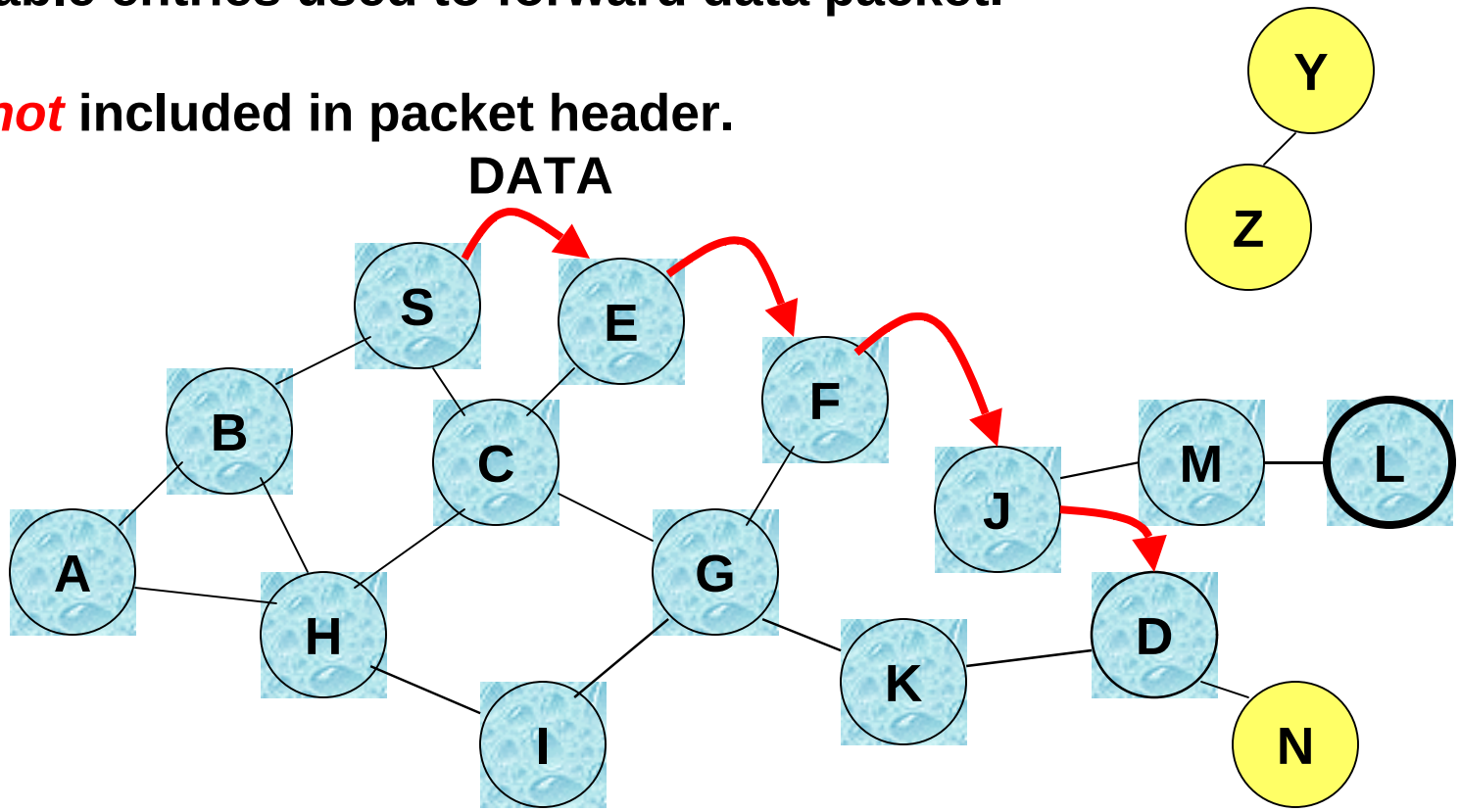
Represents a link on the forward path



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 which was forwarded using that entry

- 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

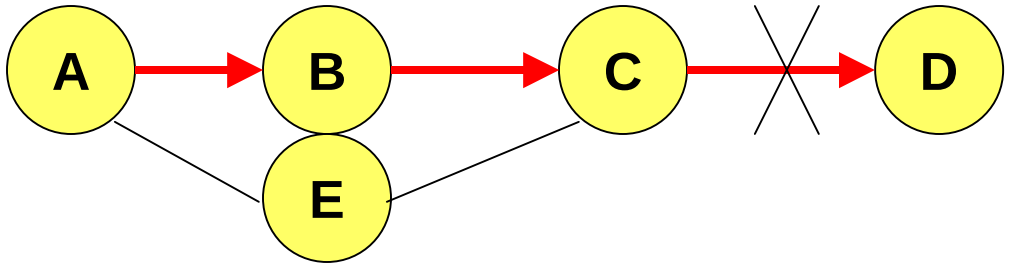
- Absence of hello message is used as an indication of link failure

- Alternatively, failure to receive several MAC-level acknowledgement may be used as an indication of link failure

Why Sequence Numbers in AODV

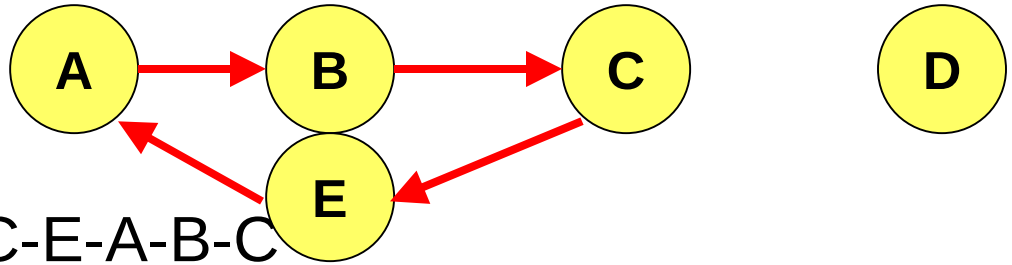
- To avoid using old/broken routes
  - To determine which route is newer

- 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)
  - 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



– Loop C-E-A-B-C

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

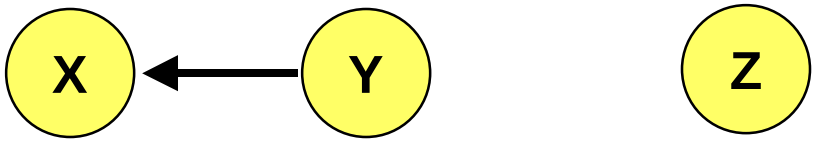
Destination-Sequenced Distance-Vector (DSDV)

- Each node maintains a routing table which stores
  - next hop towards each destination
  - a cost metric for the path to each destination
  - a destination sequence number that is created by the destination itself
  - Sequence numbers used to avoid formation of loops

- Each node periodically forwards the routing table to its neighbors
  - Each node increments and appends its sequence number when sending its local routing table
  - This sequence number will be attached to route entries created for this node

Destination-Sequenced Distance-Vector (DSDV)

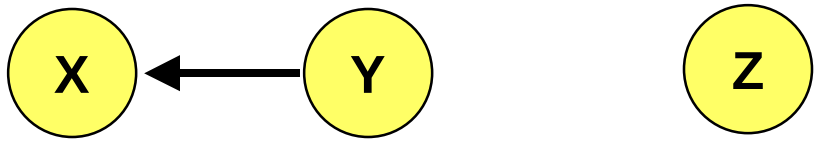
- Assume that node X receives routing information from Y about a route to node Z



- Let S(X) and S(Y) denote the destination sequence number for node Z as stored at node X, and as sent by node Y with its routing table to node X, respectively

Destination-Sequenced Distance-Vector (DSDV)

- Node X takes the following steps:

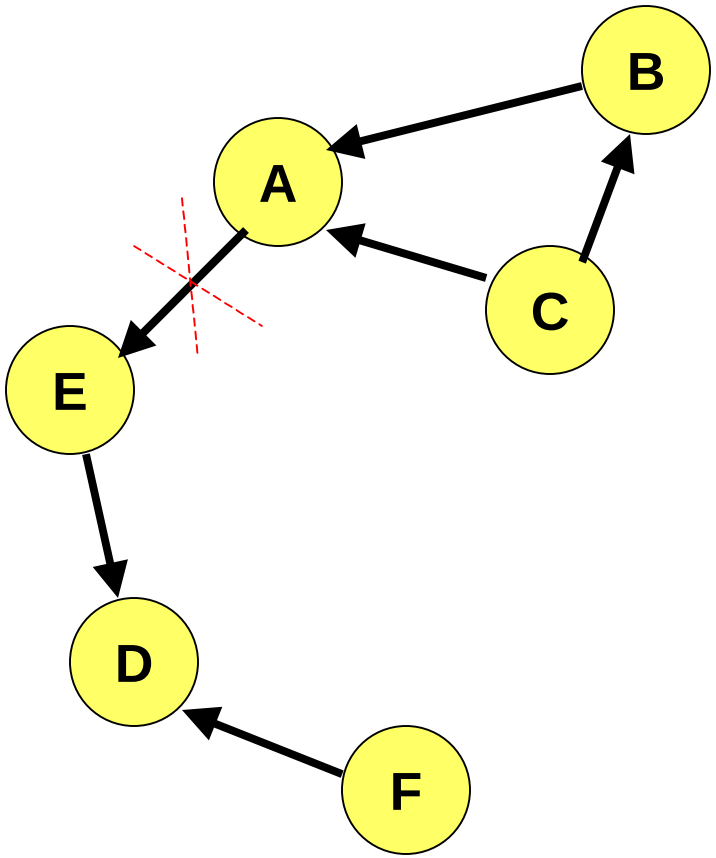


  - If S(X) > S(Y), then X ignores the routing information received from Y
  - If S(X) = S(Y), and cost of going through Y is smaller than the route known to X, then X sets Y as the next hop to Z
  - If S(X) < S(Y), then X sets Y as the next hop to Z, and S(X) is updated to equal S(Y)

Temporally-Ordered Routing Algorithm (TORA)

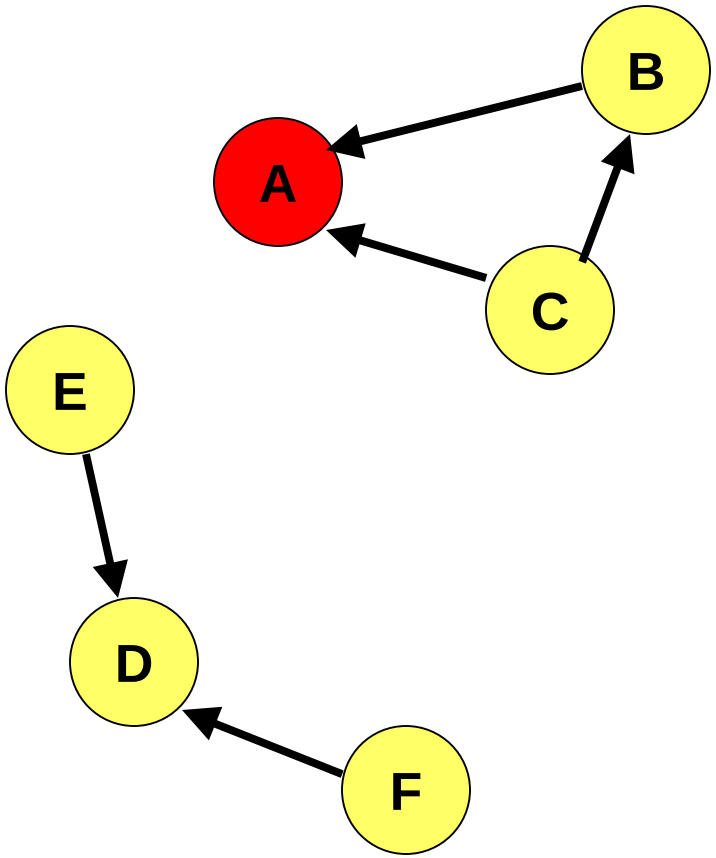
- TORA modifies the partial link reversal method to be able to detect partitions
- When a partition is detected, all nodes in the partition are informed, and link reversals in that partition cease

Partition Detection in TORA



DAG for destination D

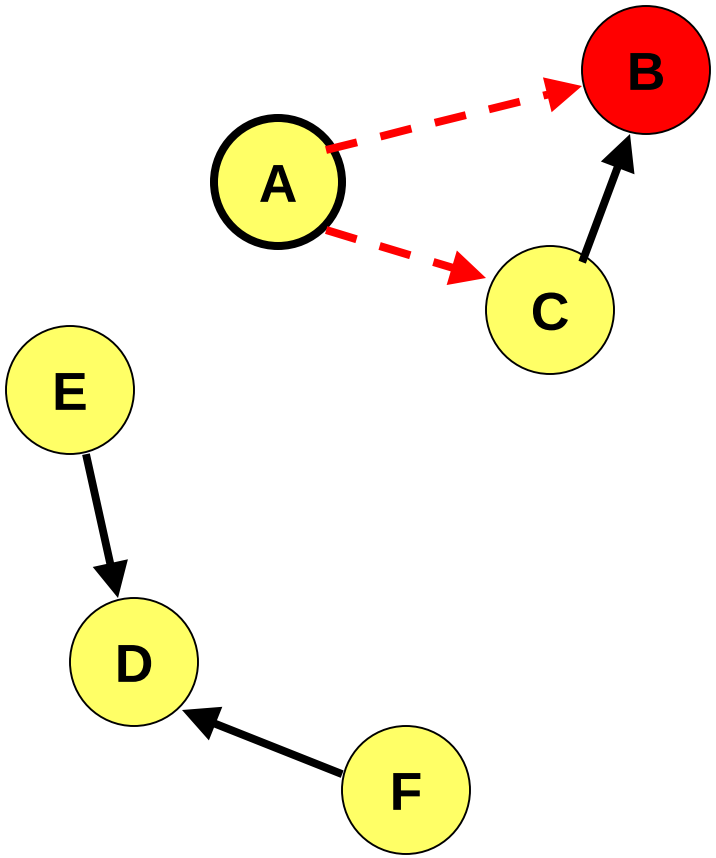
Partition Detection in TORA



TORA uses a modified partial reversal method

Node A has no outgoing links

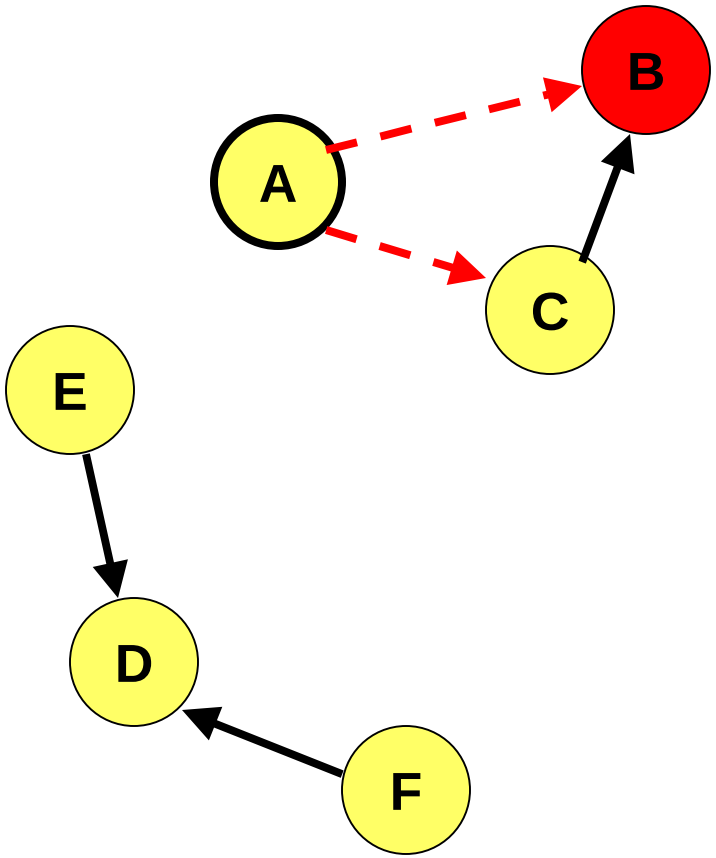
Partition Detection in TORA



TORA uses a modified partial reversal method

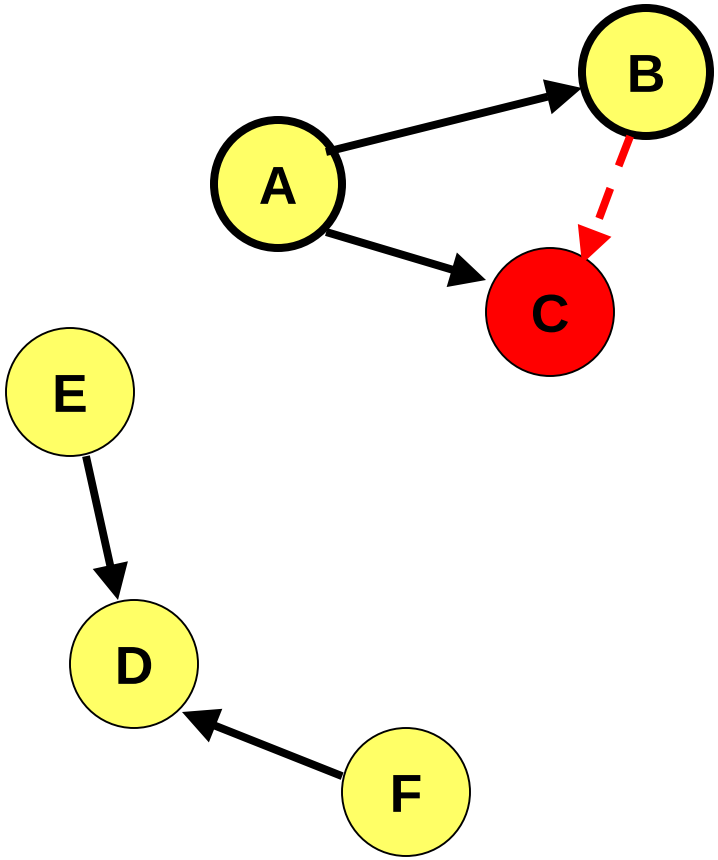
Node B has no outgoing links

Partition Detection in TORA



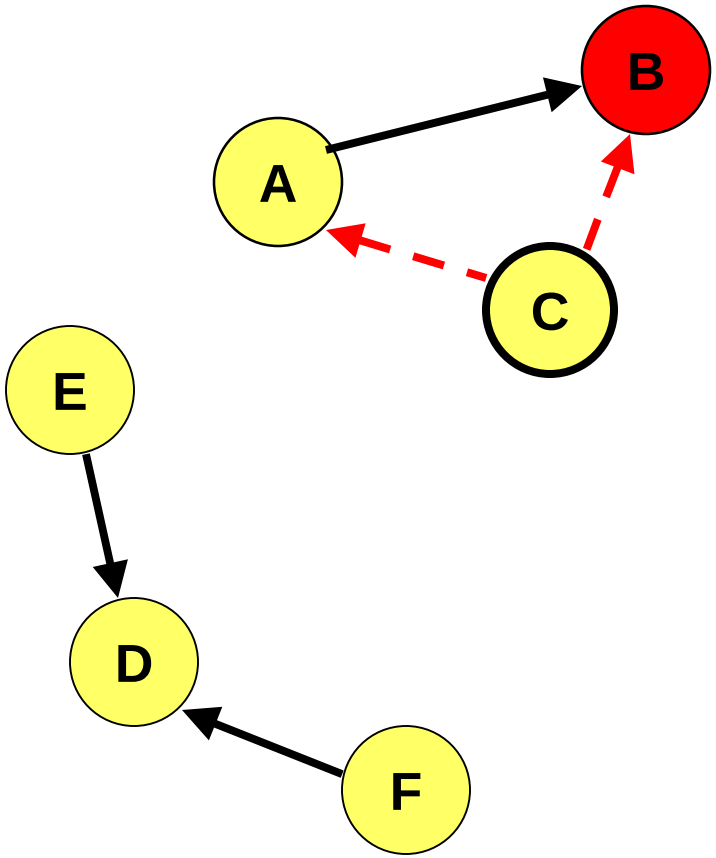
Node B has no outgoing links

Partition Detection in TORA



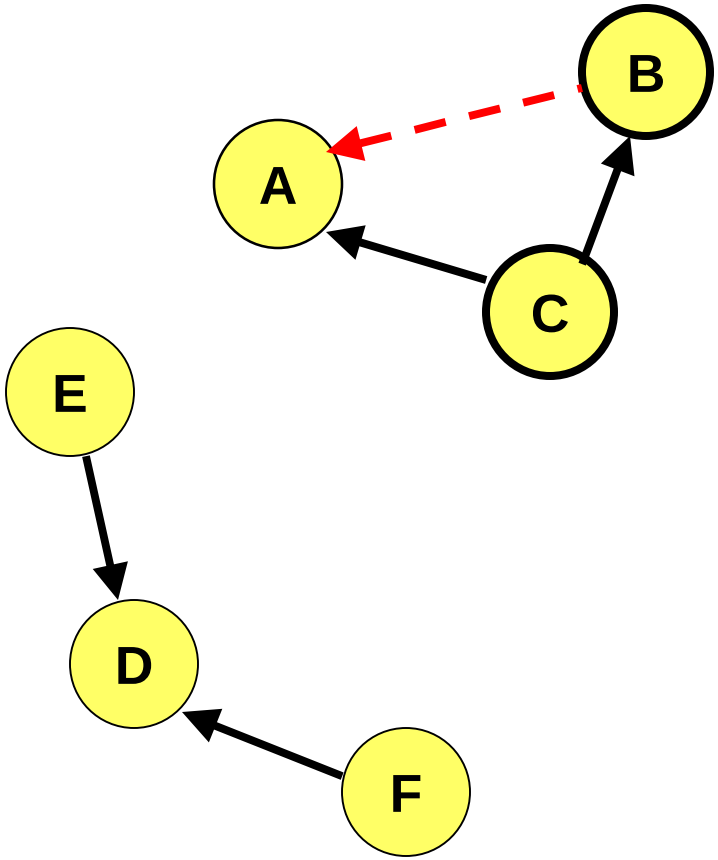
Node C has no outgoing links -- all its neighbor have reversed links previously.

Partition Detection in TORA



Nodes A and B receive the reflection from node C
Node B now has no outgoing link

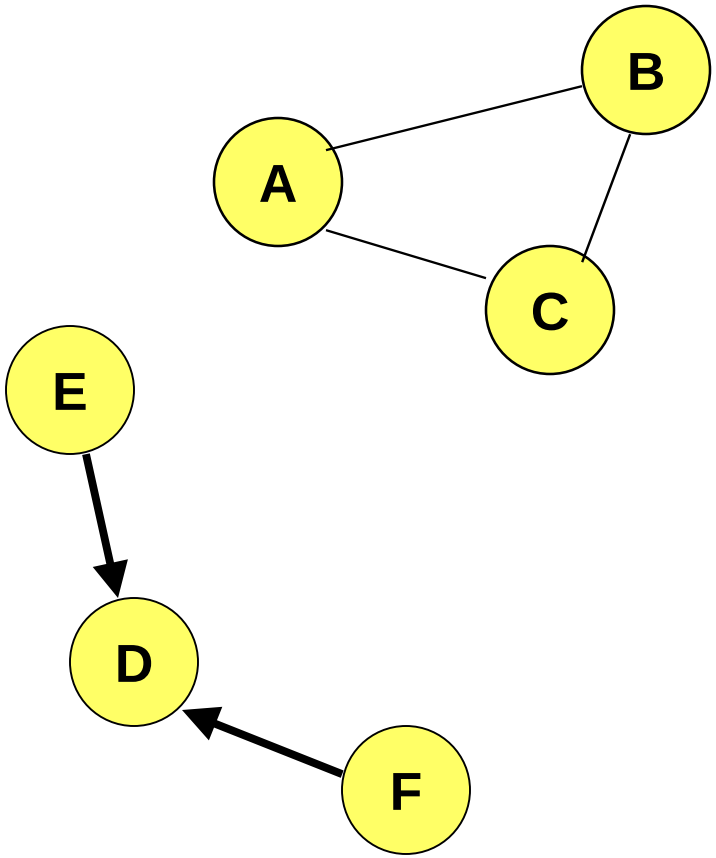
Partition Detection in TORA



Node B propagates the reflection to node A

Node A has received the reflection from all its neighbors.
Node A determines that it is partitioned from destination D.

Partition Detection in TORA



On detecting a partition, node A sends a clear (CLR) message that purges all directed links in that partition

TORA

- Improves on the partial link reversal method by detecting partitions and stopping non-productive link reversals
- Paths may not be shortest
- The DAG provides many hosts the ability to send packets to a given destination
  - Beneficial when many hosts want to communicate with a single destination

TORA Design Decision

- TORA performs link reversals as dictated by [Gafni81]
- However, when a link breaks, it looses its direction

- When a link is repaired, it may not be assigned a direction, unless some node has performed a route discovery after the link broke
  - if no one wants to send packets to D anymore, eventually, the DAG for destination D may disappear

- TORA makes effort to maintain the DAG for D only if someone needs route to D
  - Reactive behavior

TORA Design Decision

- One proposal for modifying TORA optionally allowed a more proactive behavior, such that a DAG would be maintained even if no node is attempting to transmit to the destination
- Moral of the story: The link reversal algorithm in [Gafni81] does not dictate a proactive or reactive response to link failure/repair

- Decision on reactive/proactive behavior should be made based on environment under consideration

Discussion