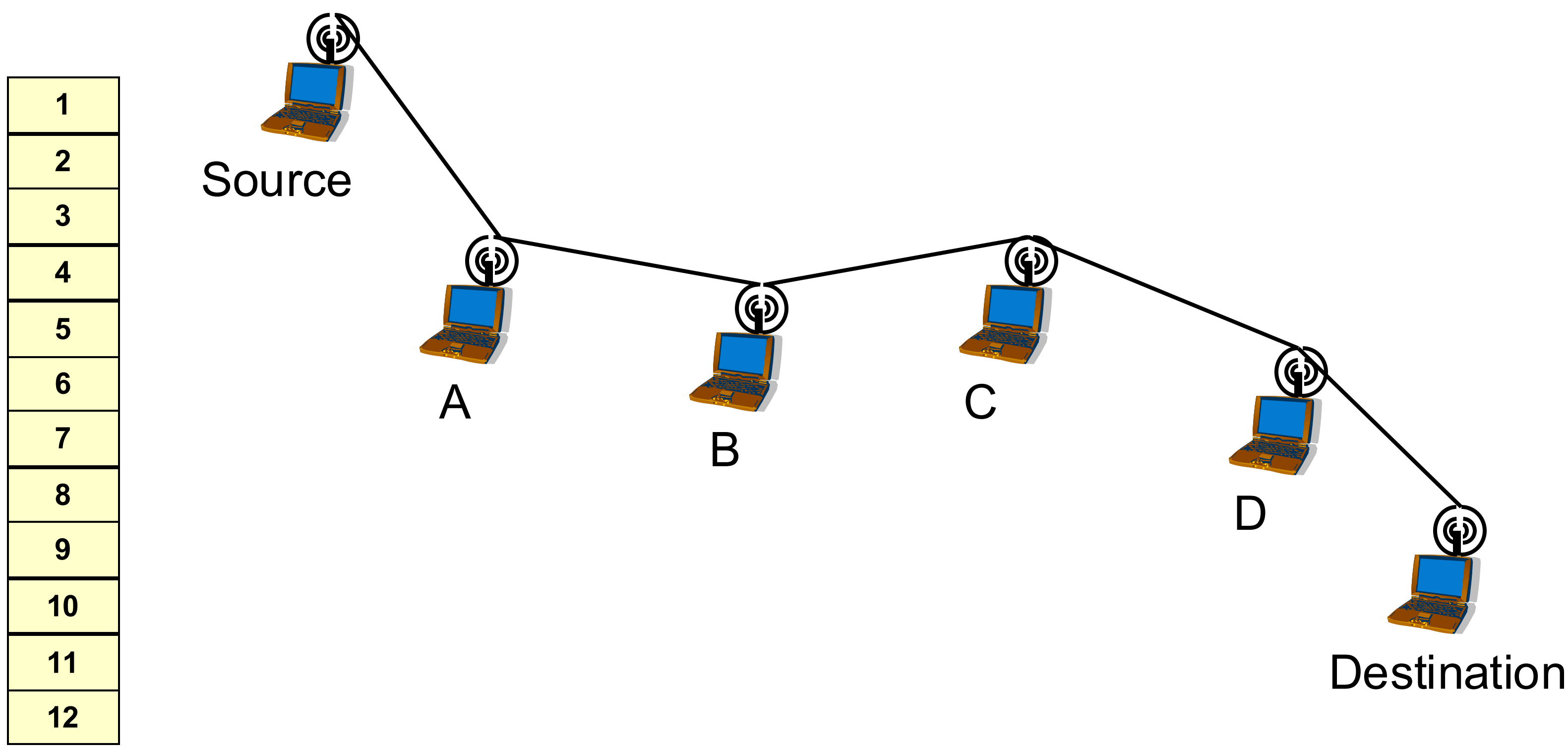
CS 439: Wireless Networking

Multihop and Infrastructureless Wireless Networks

 Fall 2025

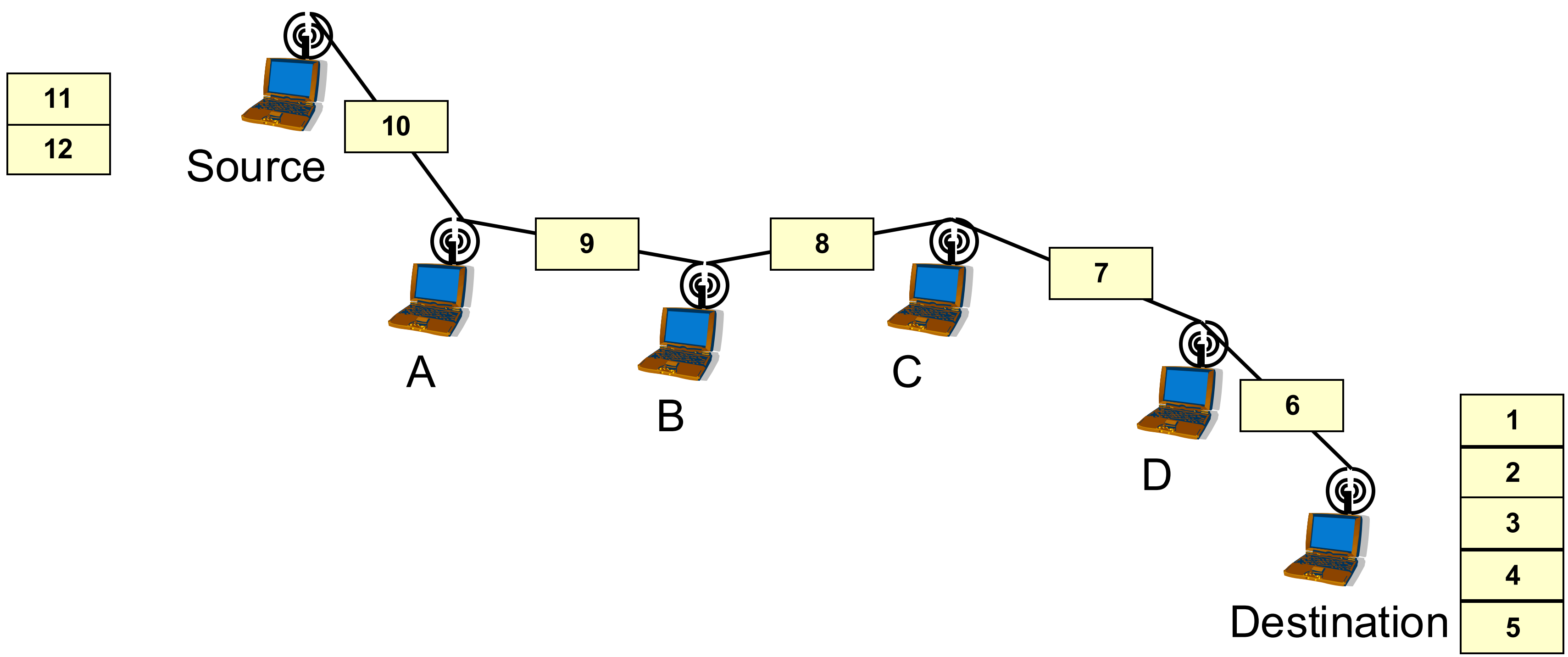
Multi-Hop Wireless Networks

- In an ideal world …



Multi-Hop Wireless Networks

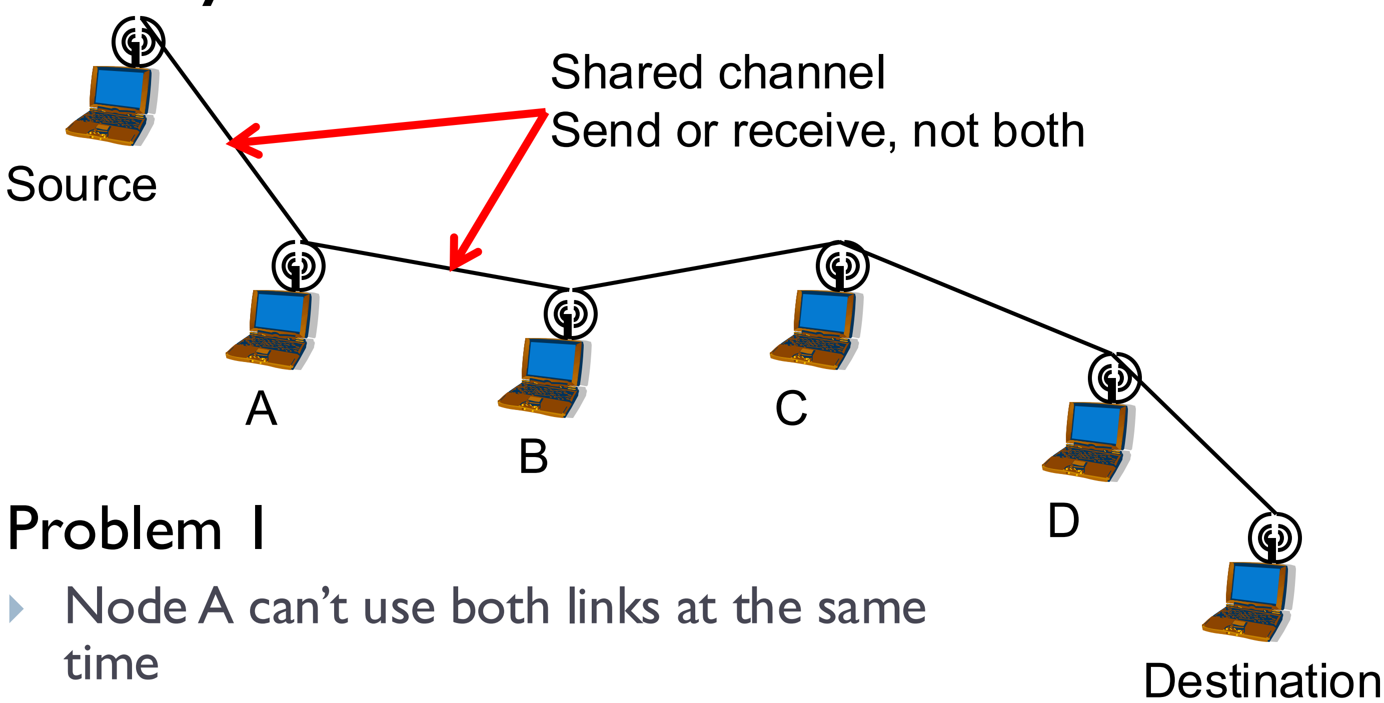
- In an ideal world …



Multi-Hop Wireless Networks

- Reality check …

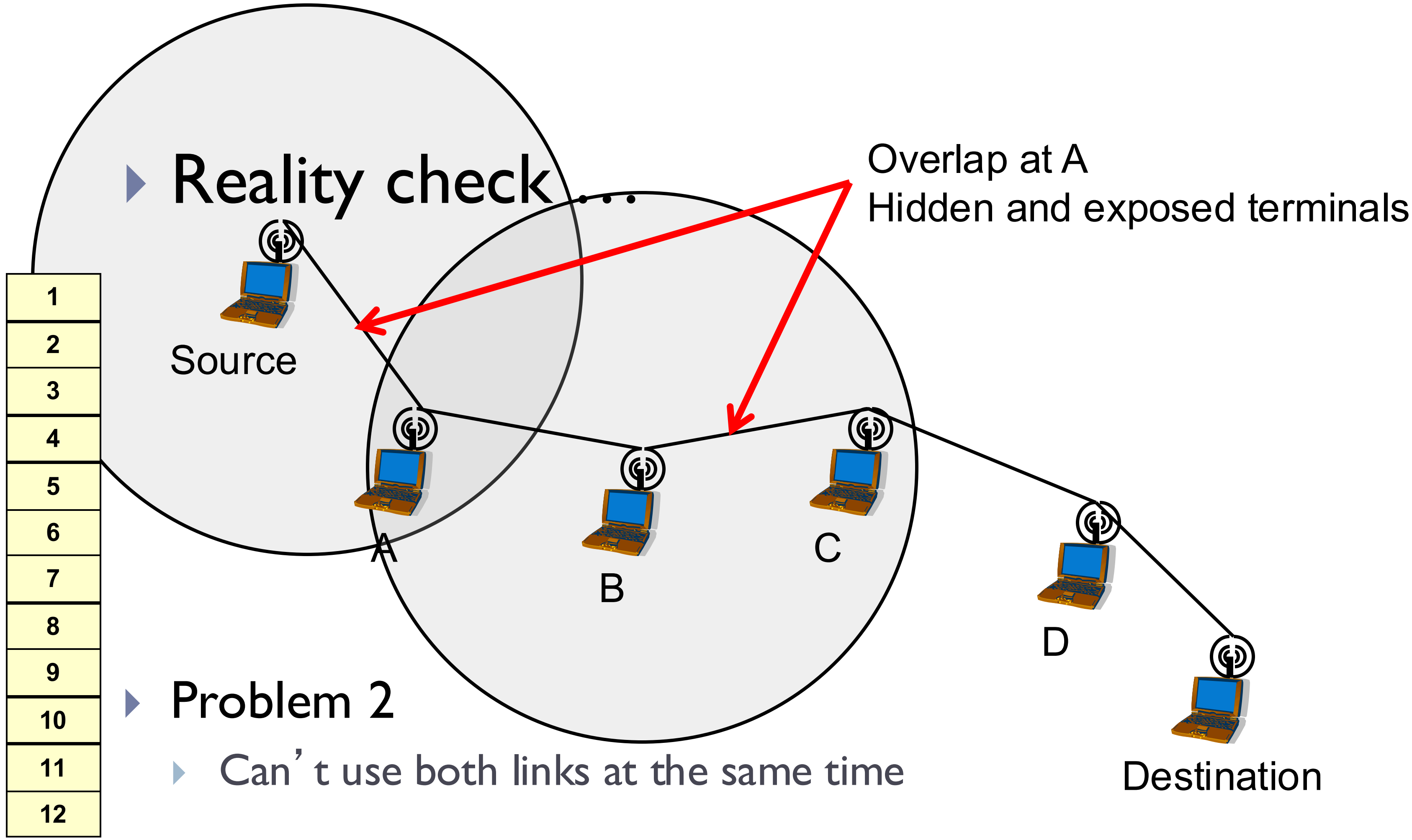
1
2
3
4
5
6
7
8
9
10
11
12



- Problem 1
  - Node A can't use both links at the same time

Multi-Hop Wireless Networks

- Reality check…



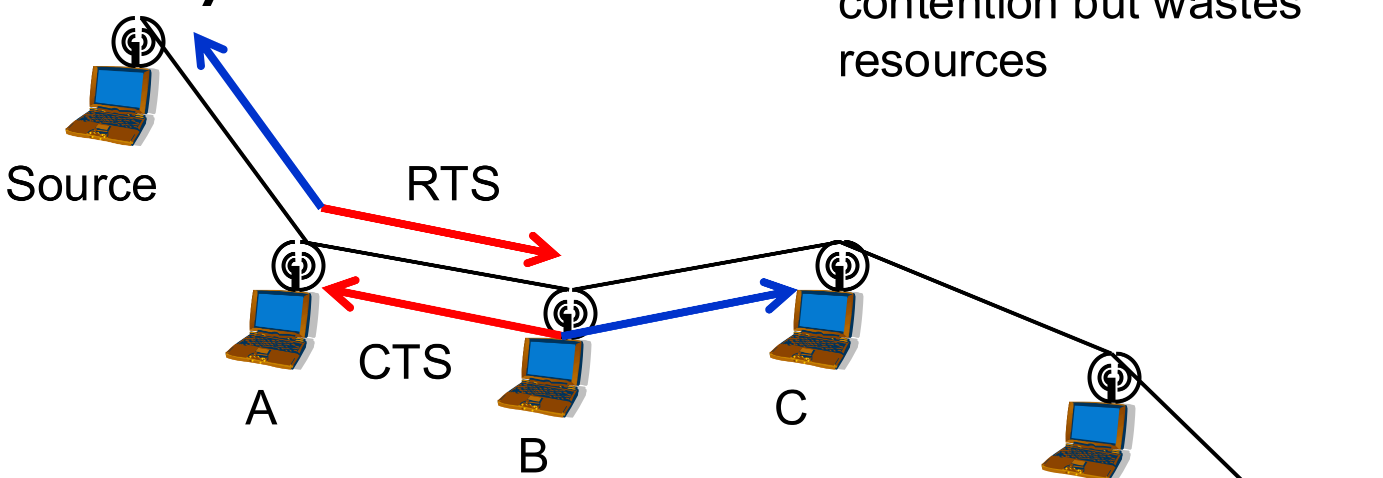
- Problem 2
  - Can't use both links at the same time

Multi-Hop Wireless Networks

1
2
3
4
5
6
7
8
9
10
11
12

- Reality check …

RTS/CTS helps with contention but wastes resources

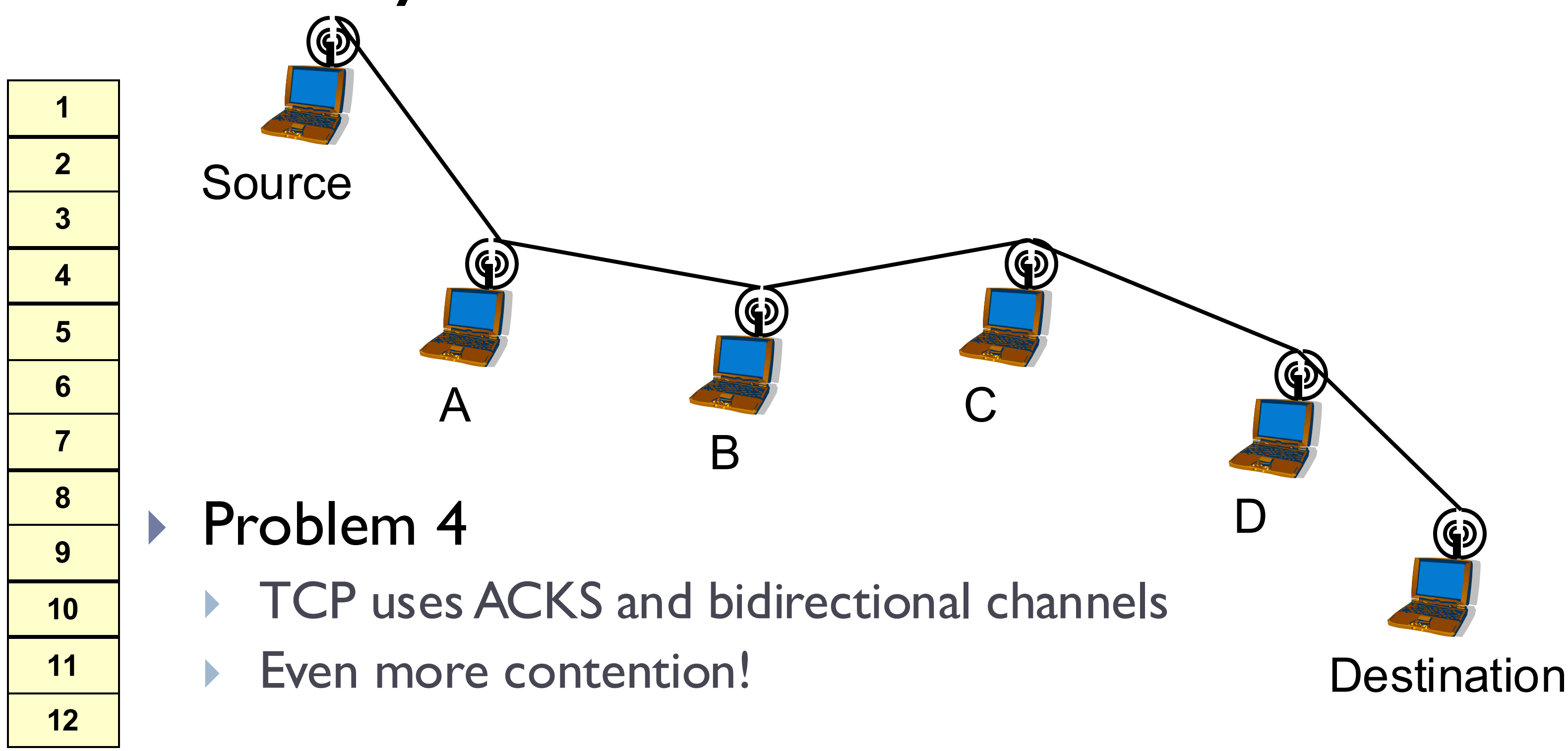


- Problem 3
  - Lots of contention for the channels
  - Everyone wants to send

Multi-Hop Wireless Networks

- Reality check …

Higher layer protocols

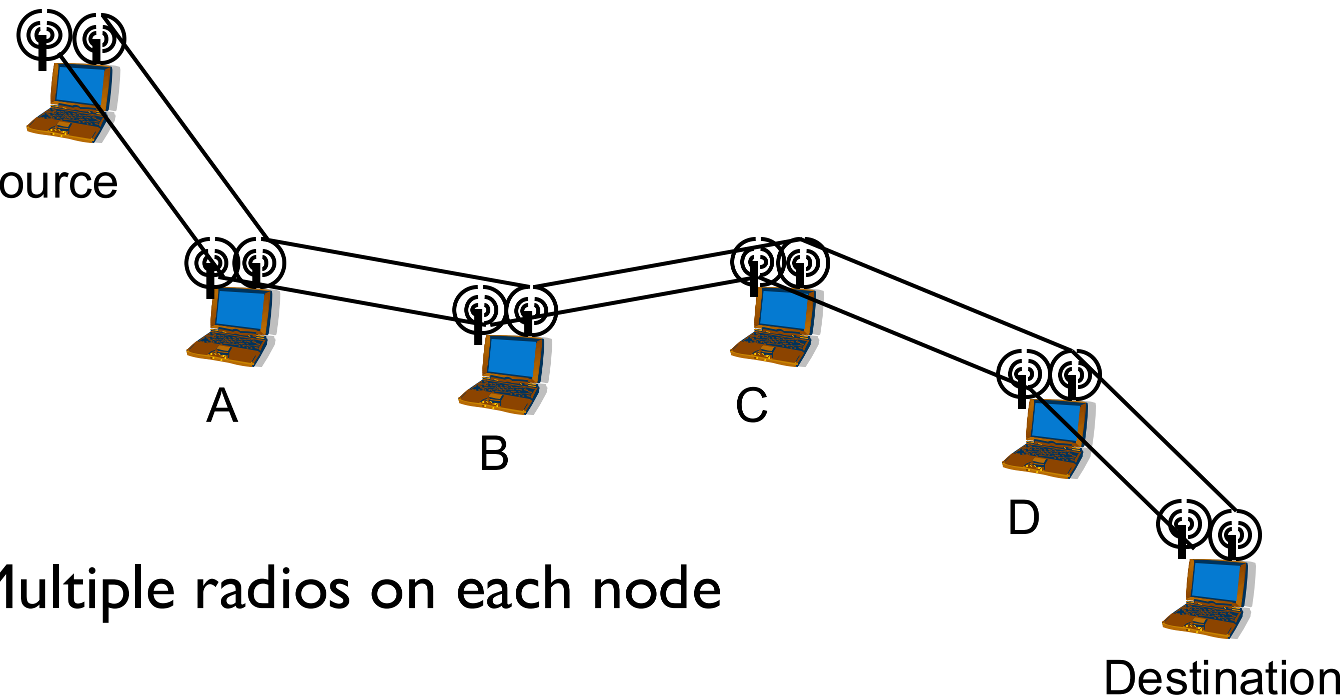


- Problem 4
  - TCP uses ACKS and bidirectional channels
  - Even more contention!

Multi-Hop Wireless Networks

- What if …

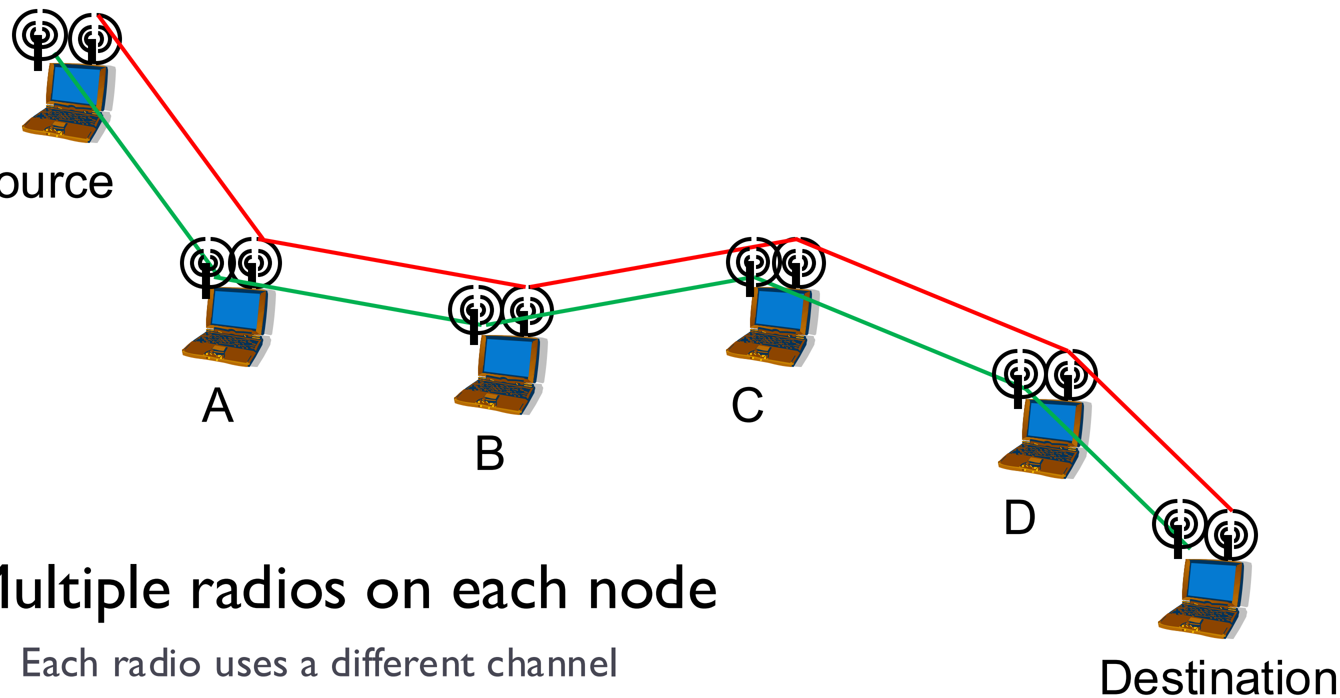
1
2
3
4
5
6
7
8
9
10
11
12



- Multiple radios on each node

Multi-Hop Wireless Networks

- What if …

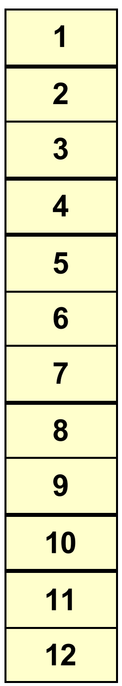


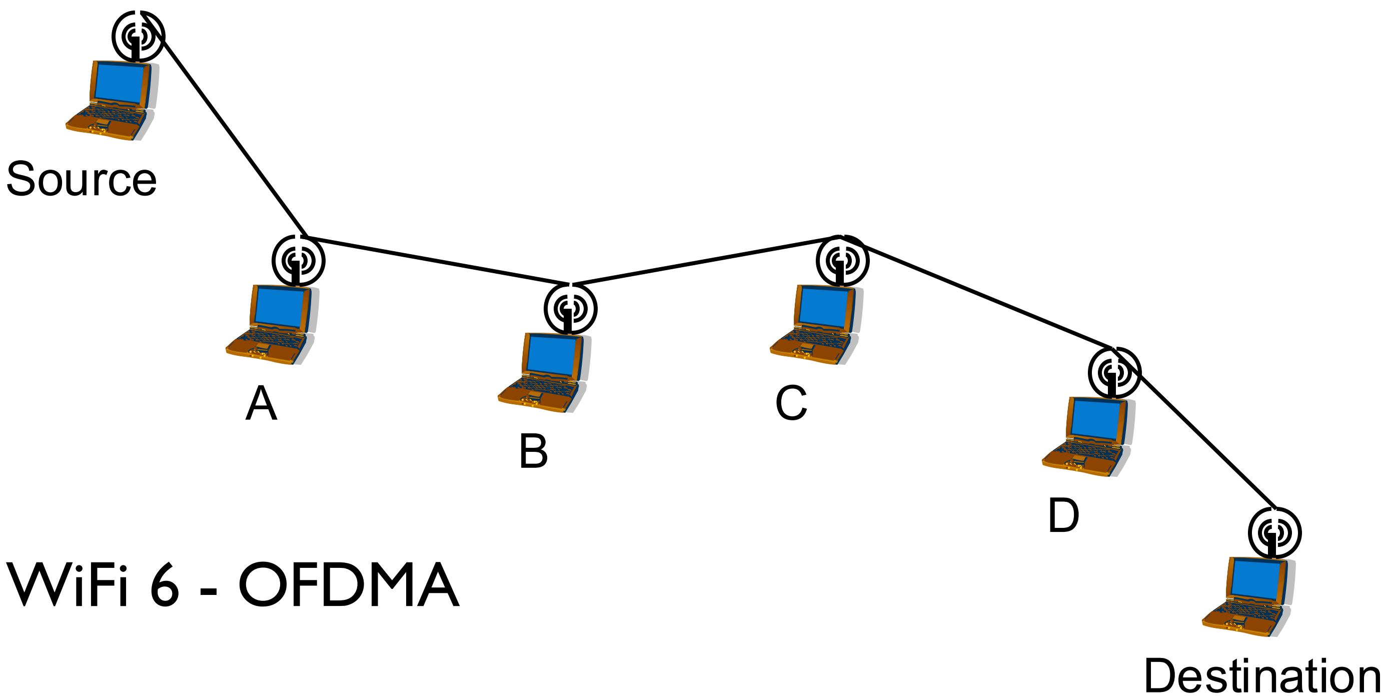
1
2
3
4
5
6
7
8
9
10
11
12

- Multiple radios on each node
  - Each radio uses a different channel

Multi-Hop Wireless Networks

- What if





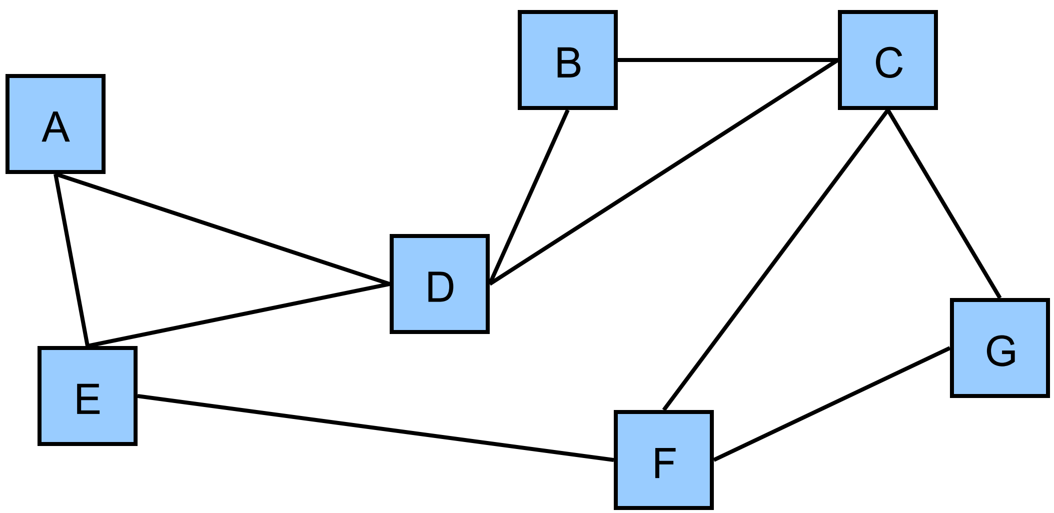
- WiFi 6 - OFDMA

Routing

- Goals
  - Capture the notion of "best" routes
  - Propagate changes effectively
  - Require limited information exchange
- Conceptually
  - A network can be represented as a graph where each host/router is a node and each physical connection is a link

Routing: Ideal Approach

- Maintain information about each link
- Calculate fastest path between each directed pair



For each direction, maintain:

- Bandwidth
- Latency
- Queueing delay

Routing: Ideal Approach

- Problems
  - Unbounded amount of information
  - Queueing delay can change rapidly
  - Graph connectivity can change rapidly
- Solution
  - Dynamic
    - Periodically recalculate routes
  - Distributed
    - No single point of failure
    - Reduced computation per node
  - Abstract Metric
    - "Distance" may combine many factors
    - Use heuristics

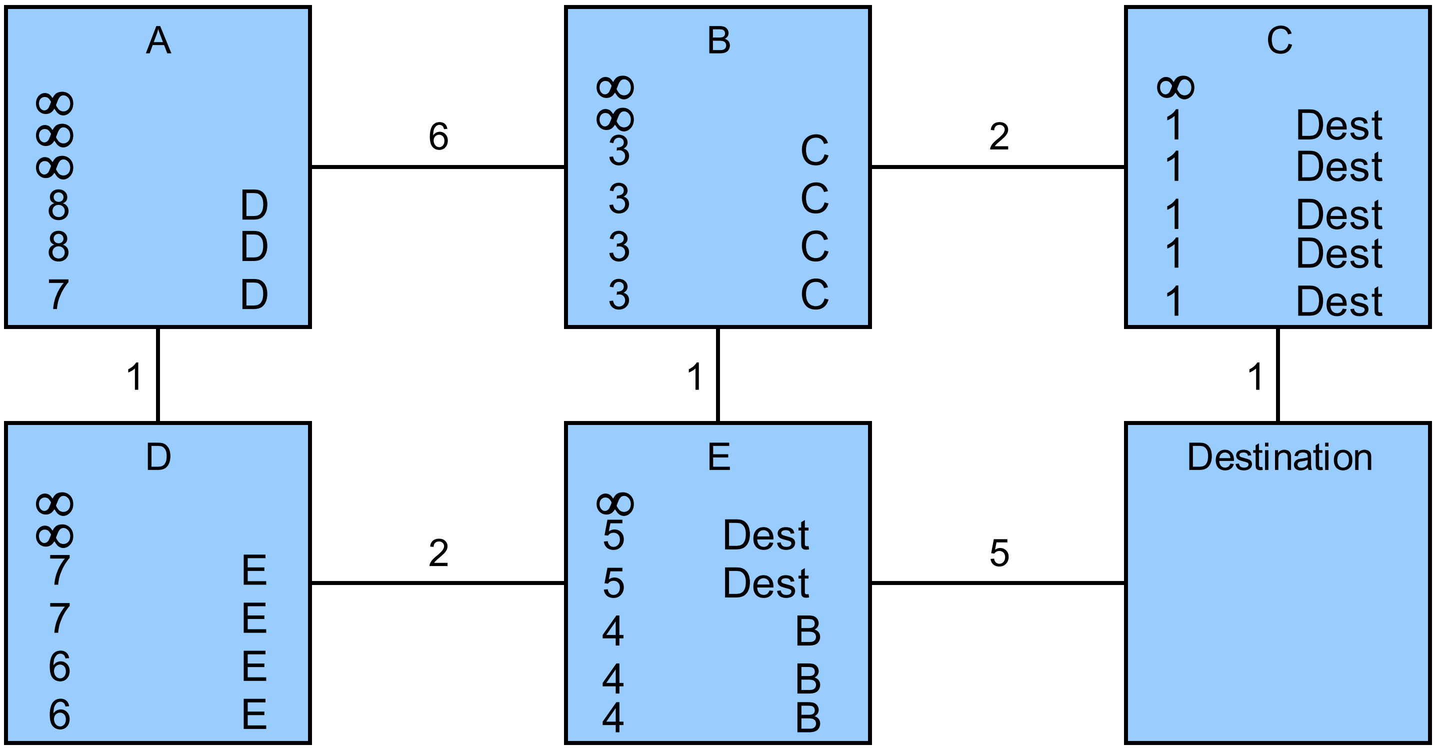
Routing Overview

- Algorithms
  - Static shortest path algorithms
    - Bellman-Ford
      - Based on local iterations
    - Dijkstra's algorithm
      - Build tree from source
  - Distributed, dynamic routing algorithms
    - Distance vector routing
      - Distributed Bellman-Ford
    - Link state routing
      - Implement Dijkstra's algorithm at each node

Bellman-Ford Algorithm

- Based on repetition of iterations
  - For every node A and every neighbor B of A
    - Is the cost of the path (A $\rightarrow$ B $\rightarrow$ $\rightarrow$ $\rightarrow$ destination) smaller than the currently known cost from A to destination?
    - If YES
      - Make B the successor node for A
      - Update cost from A to destination
  - Can run iterations synchronously or all at once

Bellman-Ford Algorithm



Distance Vector Routing

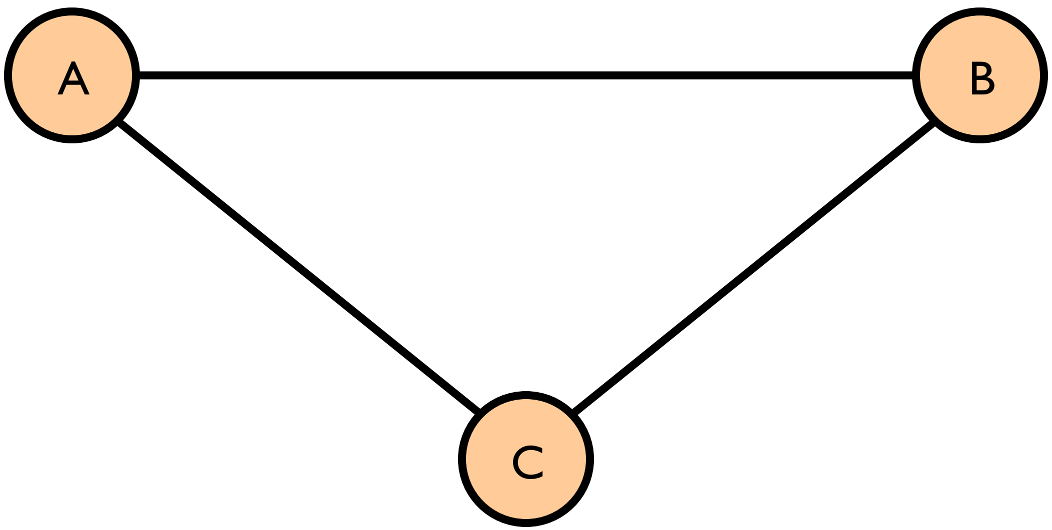
- Distributed dynamic version of Bellman-Ford
- Each node maintains a table of
  - <destination, distance, successor>
- Information acquisition
  - Assume nodes initially know cost to immediate neighbor
  - Nodes send <destination, distance> vectors to all immediate neighbors
    - Periodically – seconds, minutes
    - Whenever vector changes – triggered update

Distance Vector Routing

- When a route changes
  - Local failure detection
    - Control message not acknowledged
    - Timeout on periodic route update
  - Current route disappears
  - Newly advertised route is shorter than previous route

Distance Vector Routing

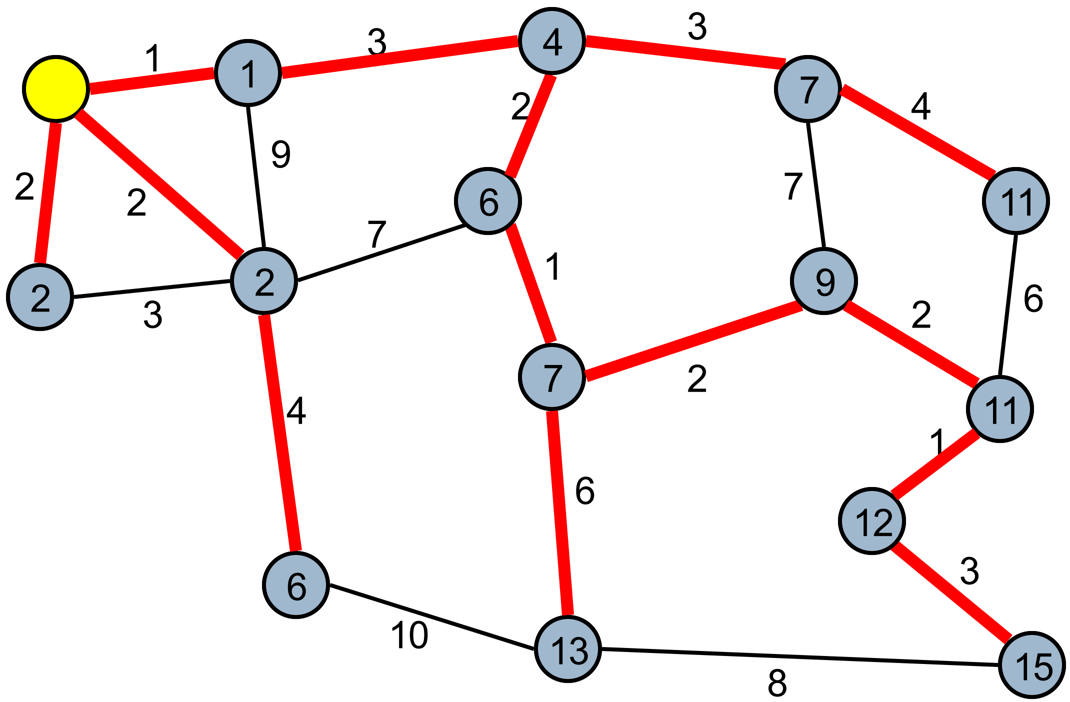
- Problem
  - Node X notices that its link to Y is broken
  - Other nodes believe that the route through X is still good
  - Mutual deception!



Dijkstra's Algorithm

- Greedily grow set C of confirmed least cost paths
- Initially C = {source}
- Loop N-1 times
  - Determine the node M outside C that is closest to the source
  - Add M to C and update costs for each node P outside C
    - Is the path (source → → … → M → P) better than the previously known path for (source → P)?
    - If YES
      - Update cost to reach P

Dijkstra’s Algorithm



Link State Routing

- Strategy
  - Send all nodes information about directly connected links
  - Status of links is flooded in link state packets (LSPs)
- Each LSP carries
  - ID of node that created the LSP
  - Vector of <neighbor, cost of link to neighbor> pairs for the node that created the LSP
  - Sequence number
  - Time-to-live (TTL)
- Each node maintains a list of (ideally all) LSP's and runs Dijkstra's algorithm on the list

Link State Routing

- LSP must be delivered to all nodes
- Information acquisition via reliable flooding
  - Create local LSP periodically with increasing sequence number
  - Send local LSP to all immediate neighbors
  - Forward new LSP out on all other links
- What does "new" mean?
  - New sequence number
  - TTL accounts for wrapped sequence numbers
    - Decrement TTL for stored nodes

Source Routing

- Variant of link state routing
  - Like link state, distribute network topology and compute shortest paths at source
  - …but only at source, not every hop!

Link State Routing

- Pros
  - Stabilizes quickly, does not generate much traffic, responds to topology changes or node failures
- Cons
  - Amount of information stored at each node is large

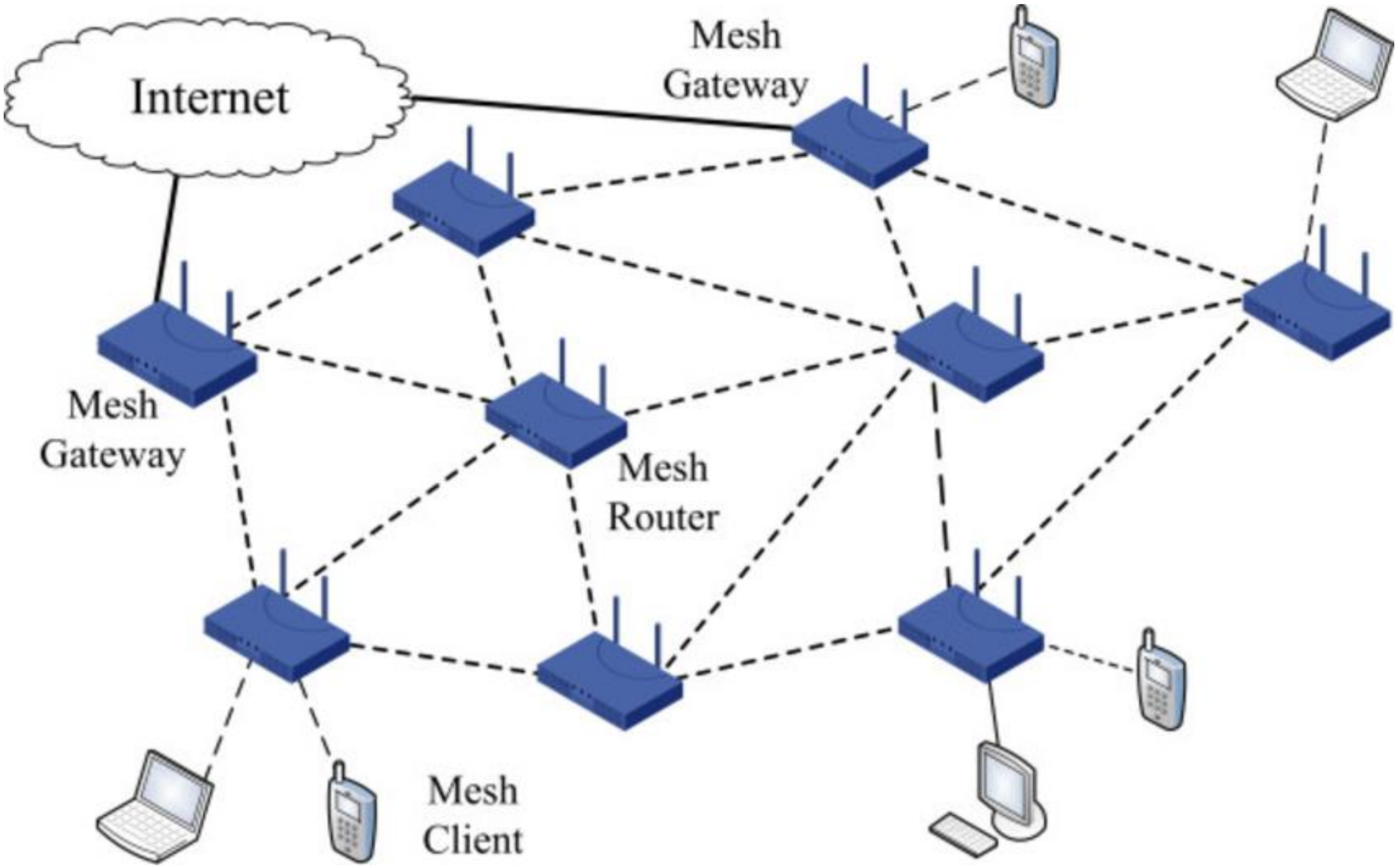
LS vs. DV

- DV
  - Send everything you know to your neighbors
- LS
  - Send info about your neighbors to everyone
- Message size
  - Small with LS
  - Potentially large with DV
- Message exchange
  - LS: O(nE)
  - DV: only to neighbors

When the network just isn't there …

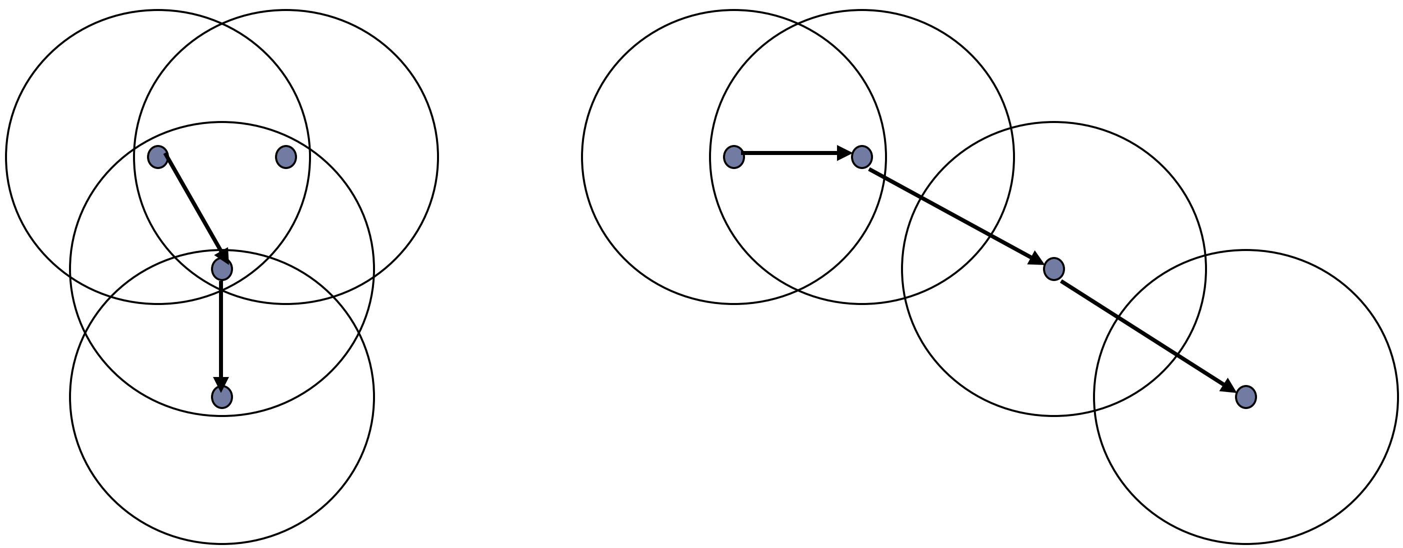
- Mesh networks
  - Core nodes are stable
  - Paths may exist between a src/dst pair
- Ad hoc networks
  - Group of cooperating nodes
  - Nodes are mobile
  - Paths eventual exist between a src/dst pair
  - All nodes are routers

Mesh Network



Ad Hoc Networks

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



Why Ad Hoc Networks ?

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

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 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
- 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

The Holy Grail

- A one-size-fits-all solution
  - Perhaps using an adaptive/hybrid approach that can adapt to situation at hand
- Difficult problem
- Many solutions proposed trying to address a sub-space of the problem domain

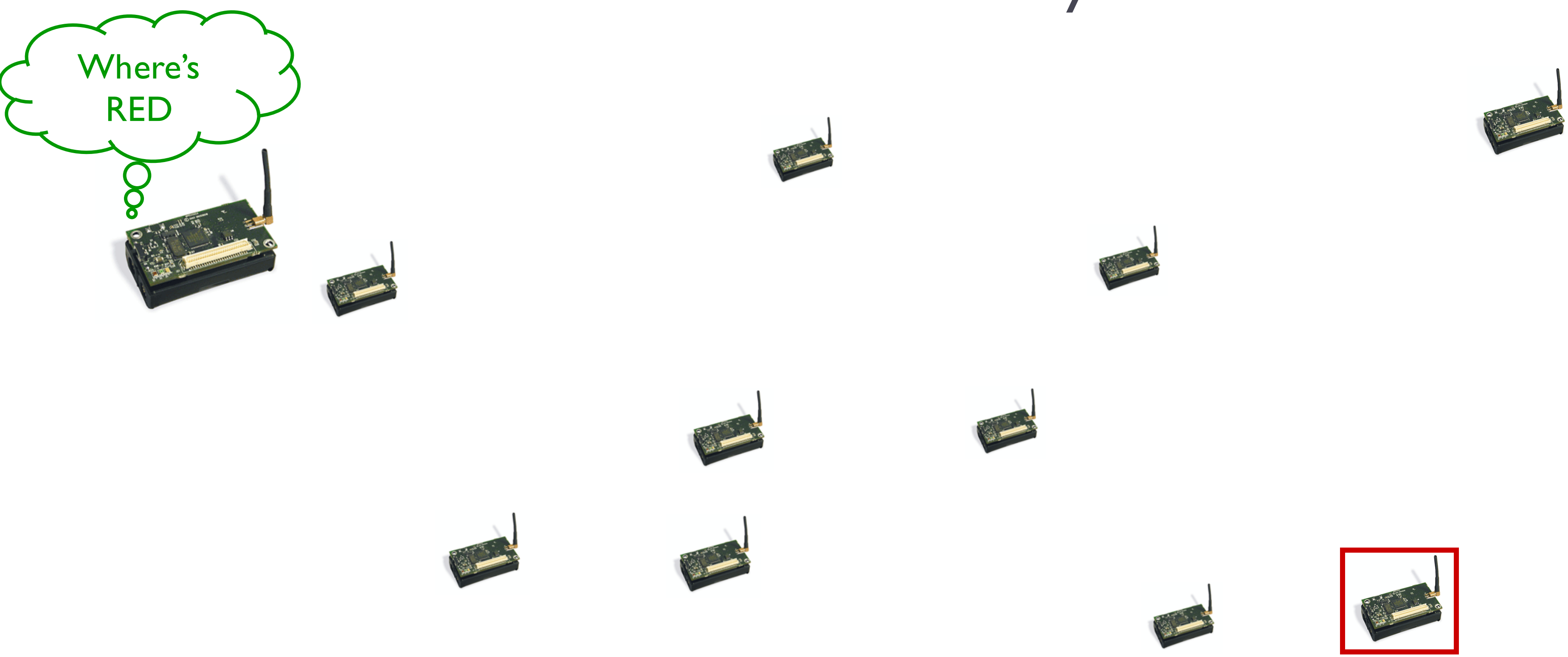
Unicast Routing in Ad Hoc Networks

Why is routing in wireless ad hoc networks different/difficult?

- Link instability causes many routing issues
  - Shortest hop routing often worst choice
  - Scarce bandwidth makes overhead conspicuous
  - Battery power a concern
  - Security and misbehavior …
- Host mobility
  - Link failure/repair due to mobility may have different characteristics than those due to other causes
  - Rate of link failure/repair may be high when nodes move fast
- New performance criteria may be used
  - Route stability despite mobility
  - Energy consumption

Routing in Mobile Networks

- Imagine hundreds of hosts moving
  - Routing algorithm needs to cope up with varying wireless channel and node mobility



Unicast Routing Protocols

- Many protocols have been proposed
  - Some have been invented specifically for ad hoc networks
  - Others are adapted from wired network routing

- No single protocol works well in all environments
  - Some attempts made to develop adaptive protocols

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
  - Maintain routes to nearby nodes
  - Discover routes for far away nodes

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

Trade-Off

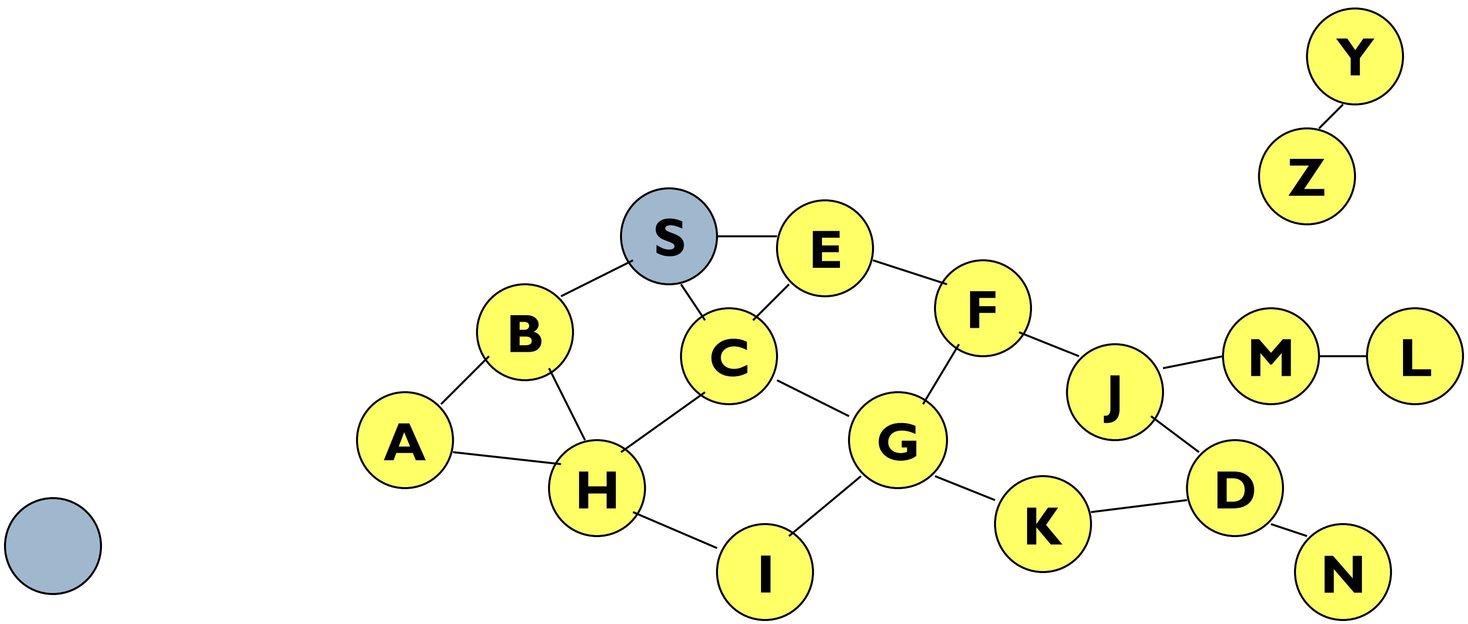
- Overhead of route discovery/maintenance
  - Reactive protocols
    - May have lower overhead since routes are determined only if needed
  - Proactive protocols
    - Can (but not necessarily) result in higher overhead due to continuous route updating

- Which approach achieves a better trade-off depends on the traffic and mobility patterns

Flooding for Data Delivery

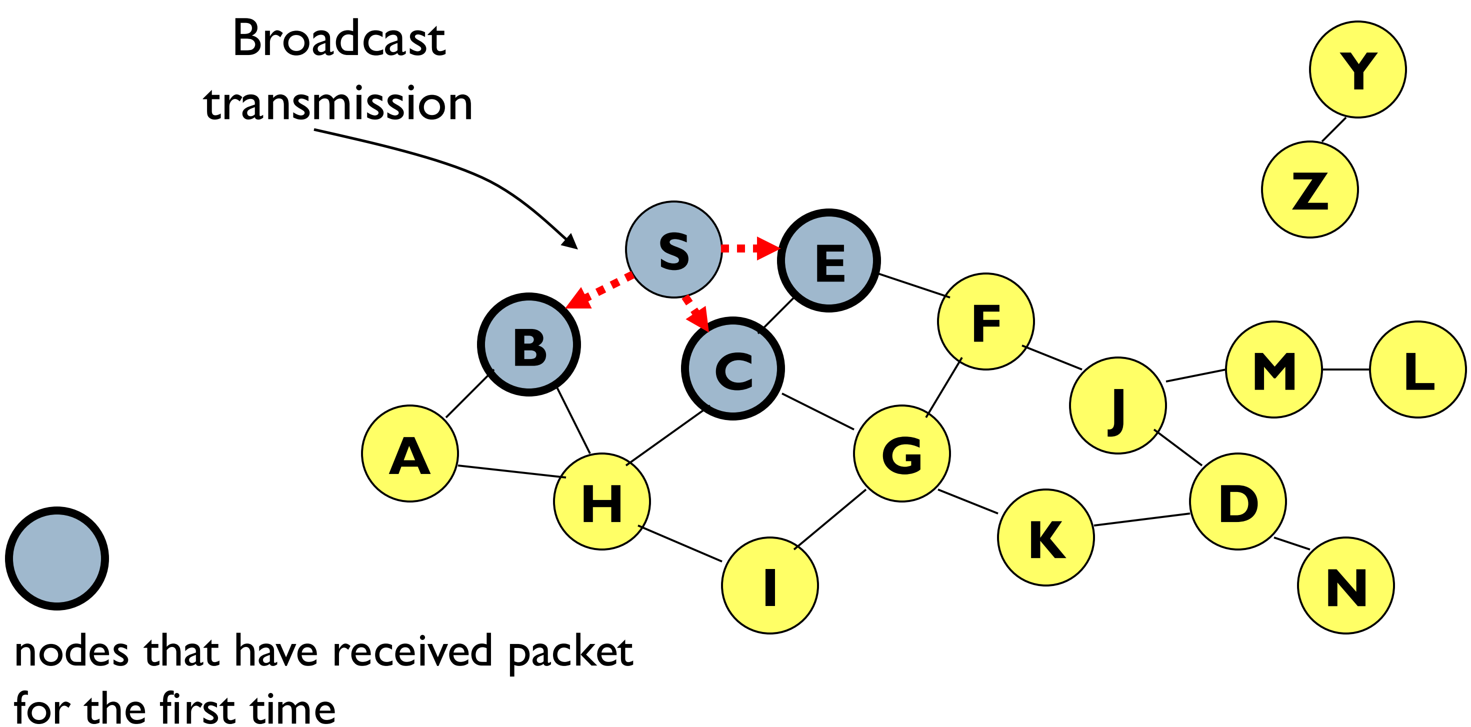
- Sender
  - Broadcasts data packet P to all its neighbors
- Intermediate nodes
  - Forward P to its neighbors
- Sequence numbers
  - Used to avoid the possibility of forwarding the same packet more than once
- Destination
  - Packet P reaches destination D provided that D is reachable from sender S
  - Node D does not forward the packet

Flooding for Data Delivery



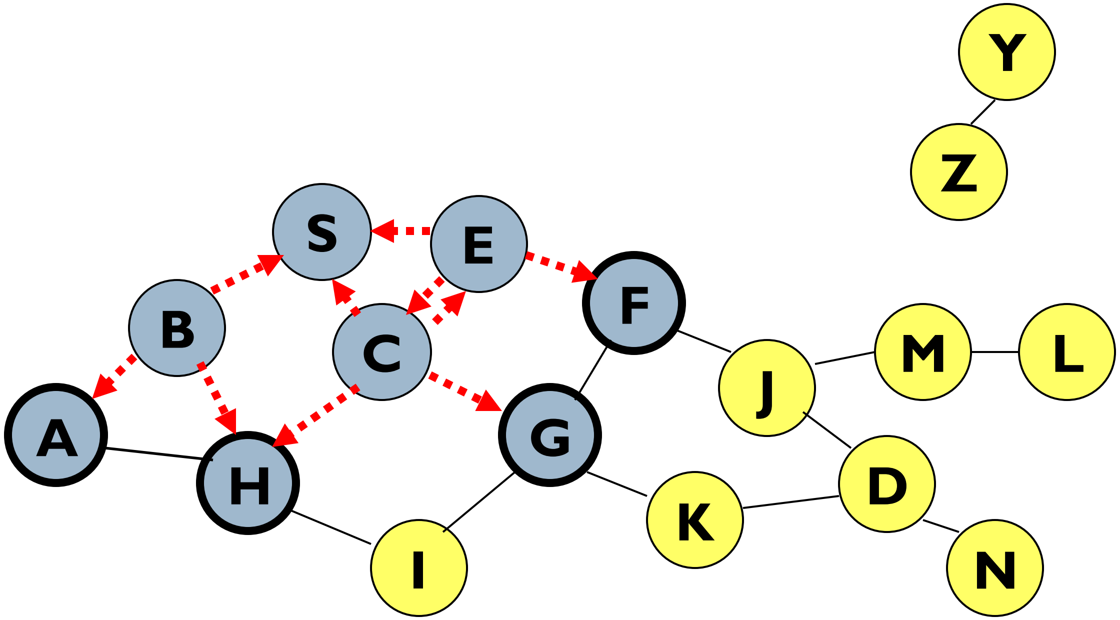
nodes that have received packet

Flooding for Data Delivery



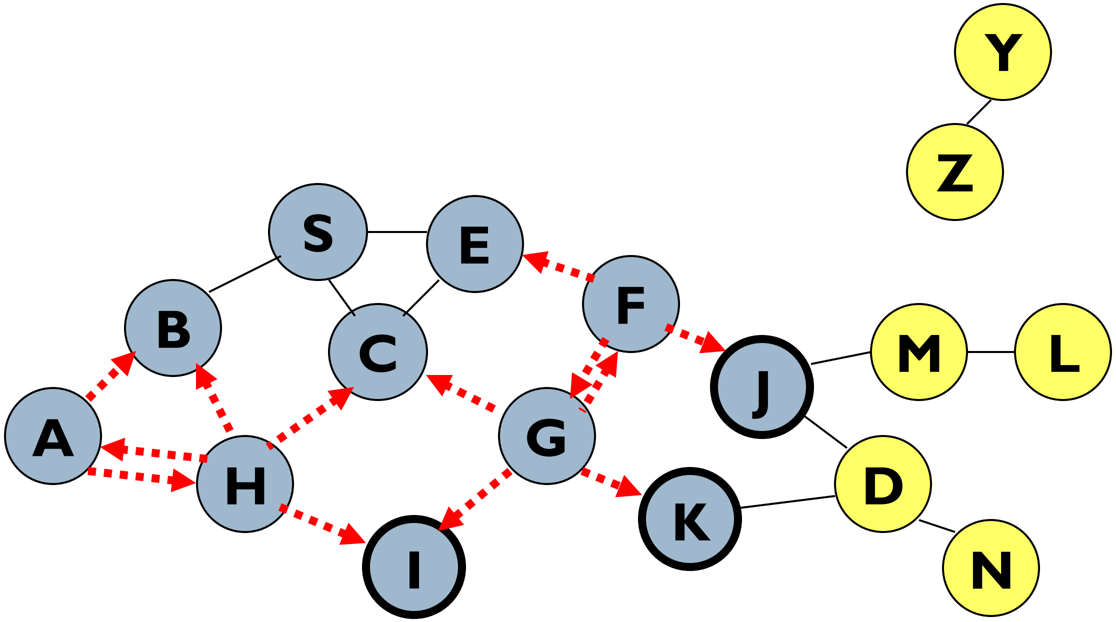
Flooding for Data Delivery

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



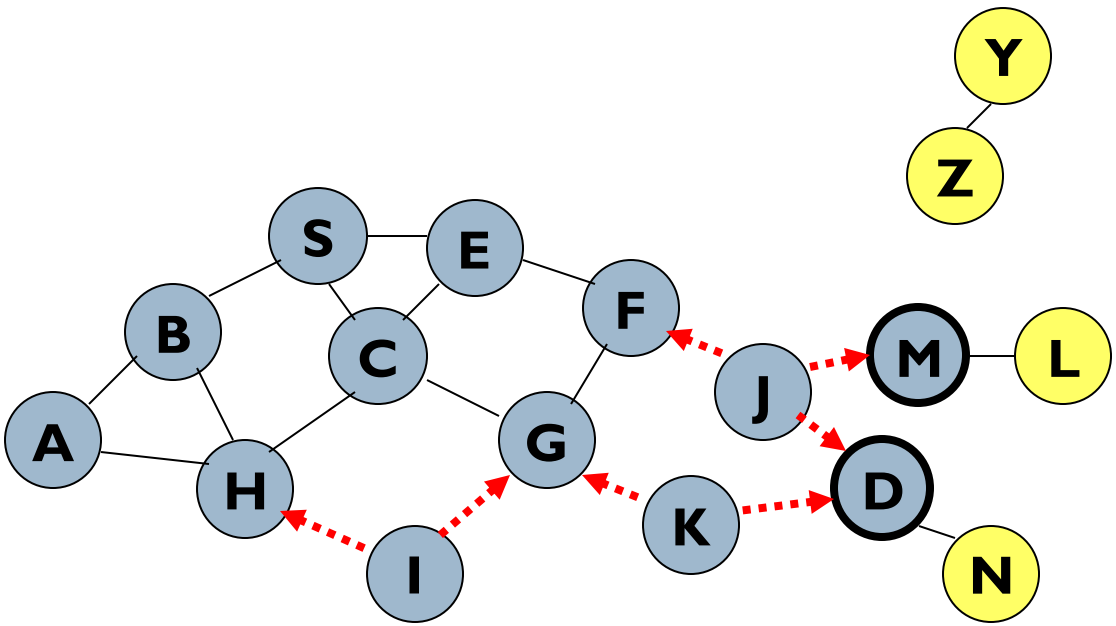
Flooding for Data Delivery

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



Flooding for Data Delivery

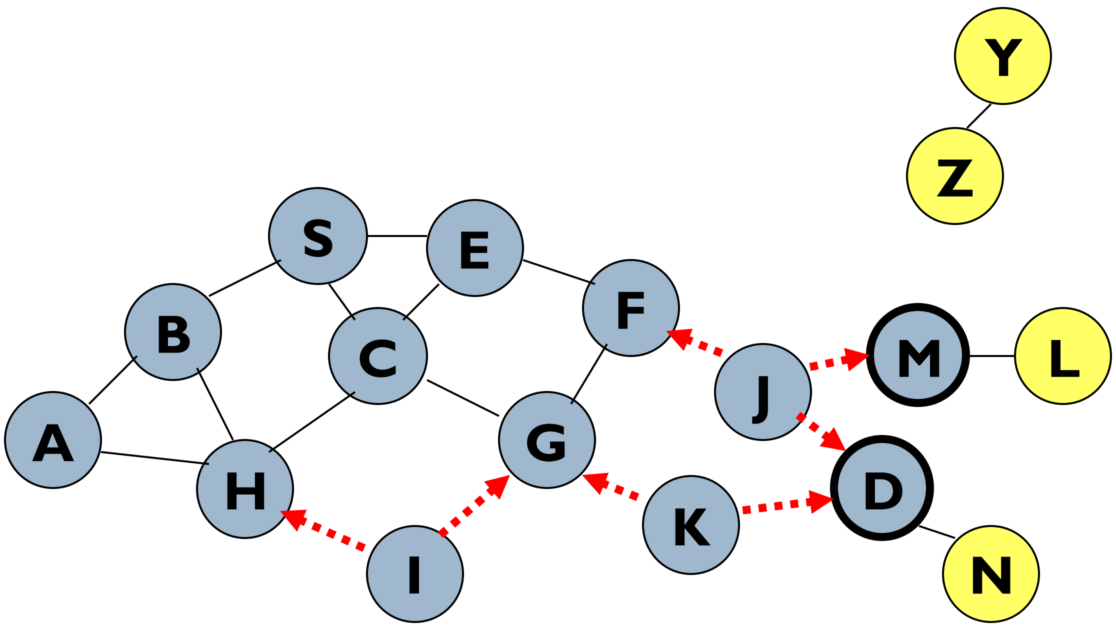
- Nodes J and K both broadcast packet to node D
  - Since nodes J and K are hidden from each other, their transmissions may collide



Flooding for Data Delivery

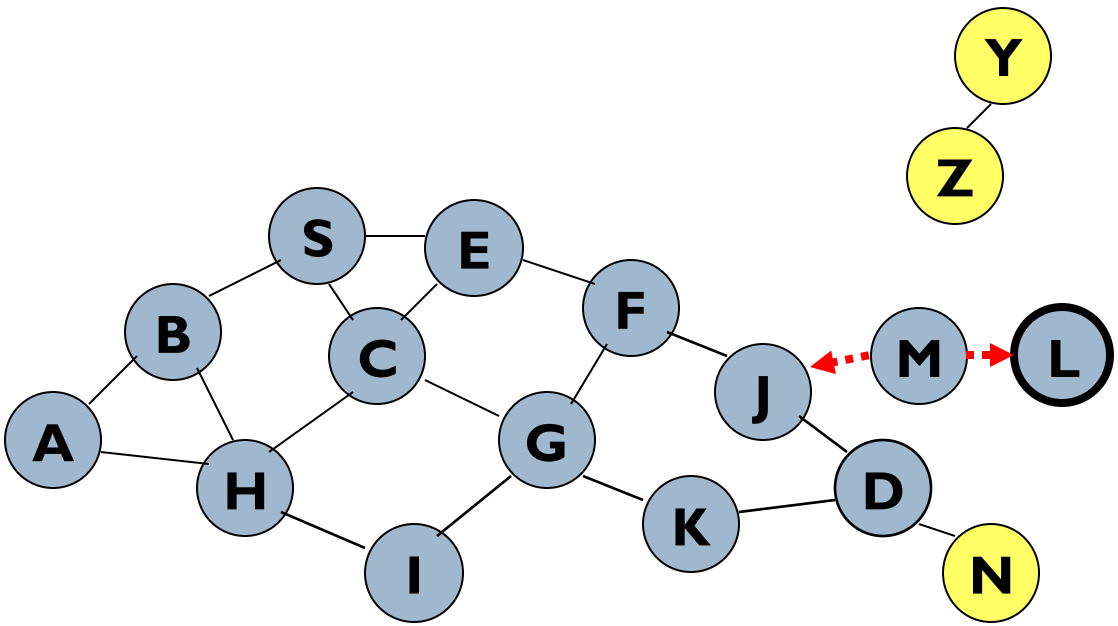
- Nodes J and K both broadcast packet to node D

  => Packet may not be delivered to node D at all, despite the use of flooding



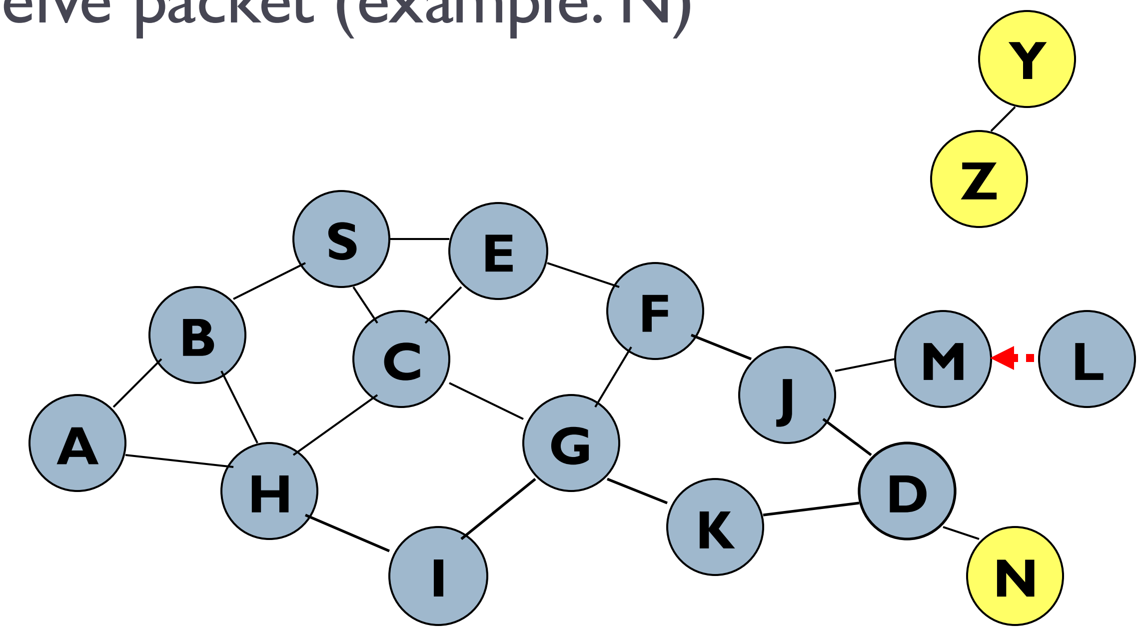
Flooding for Data Delivery

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



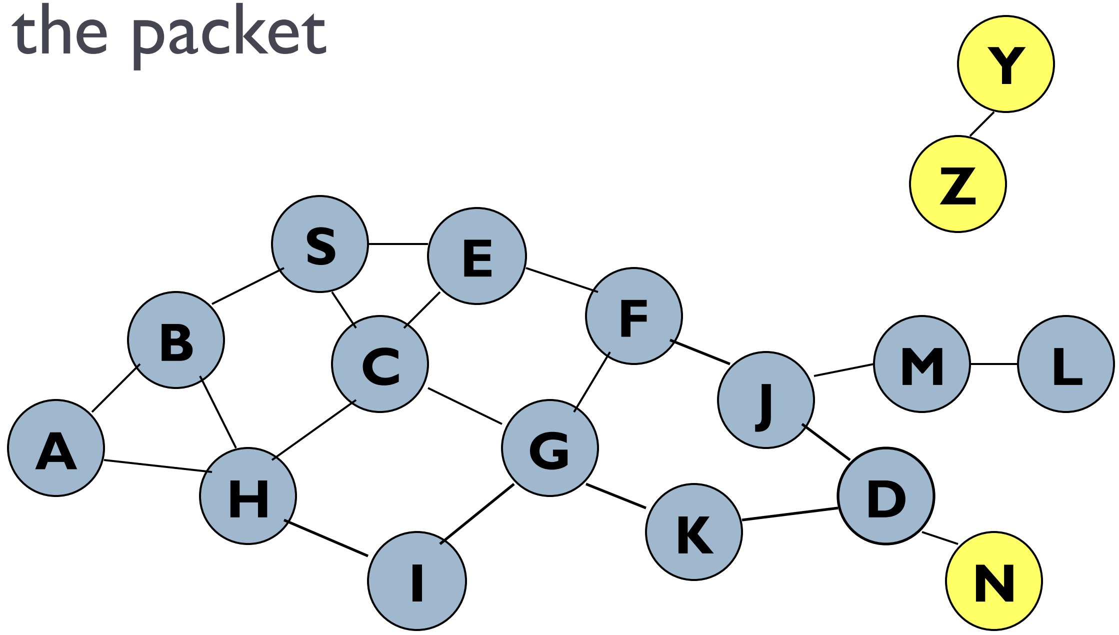
Flooding for Data Delivery

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



Flooding for Data Delivery

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



Flooding for Data Delivery: Advantages

- Simplicity
- Efficiency
  - Low rate of information transmission
    - Overhead of explicit route discovery/maintenance incurred by other protocols is relatively higher
    - For example, 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
      - Broadcast in IEEE 802.11 MAC is unreliable
    - e.g., 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

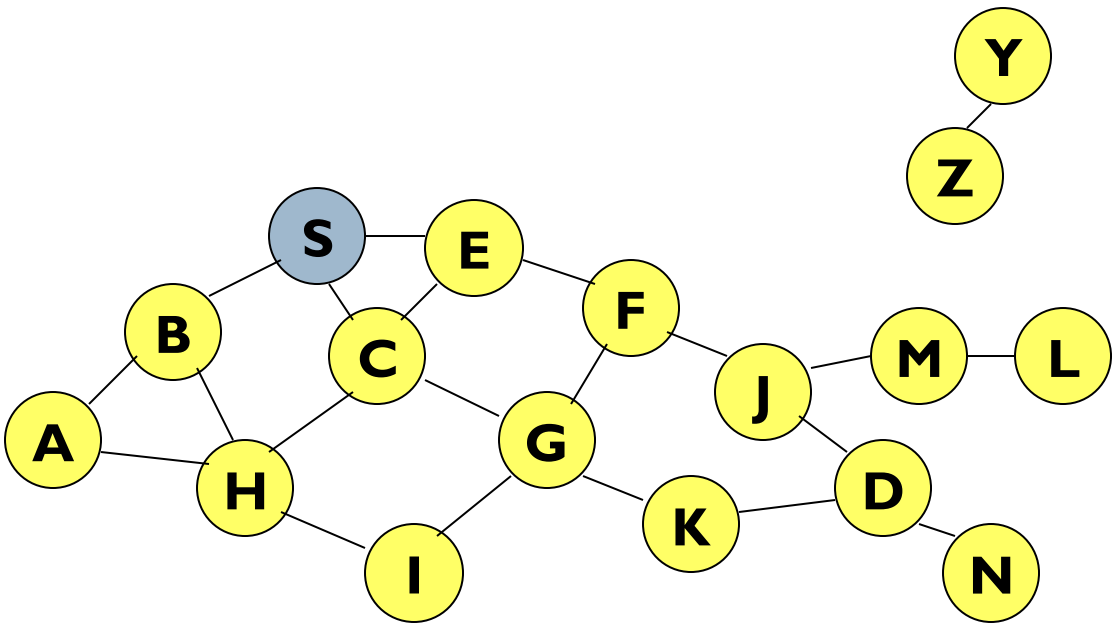
Flooding of Control Packets

- 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

Dynamic Source Routing (DSR)

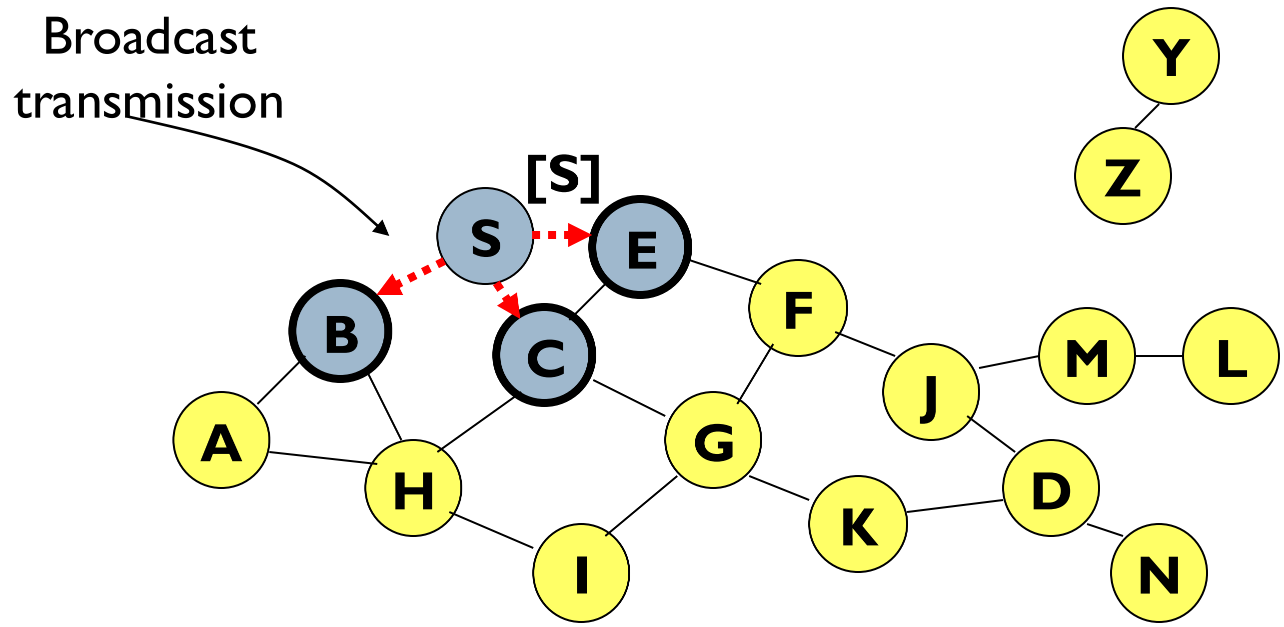
- Route Discovery
  - When node S wants to send a packet to node D, but does not know a route to D, node S initiates a route discovery
  - Source node S floods Route Request (RREQ)
  - Each node appends own identifier when forwarding RREQ

Route Discovery in DSR



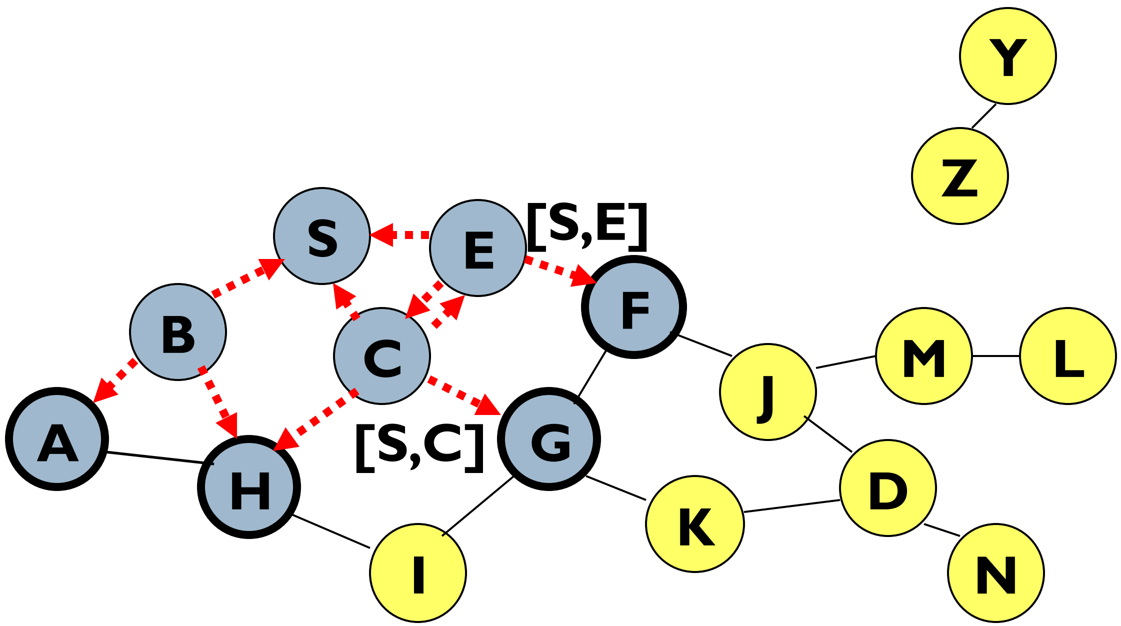
Route Discovery in DSR

- [X,Y]: list of identifiers appended to RREQ



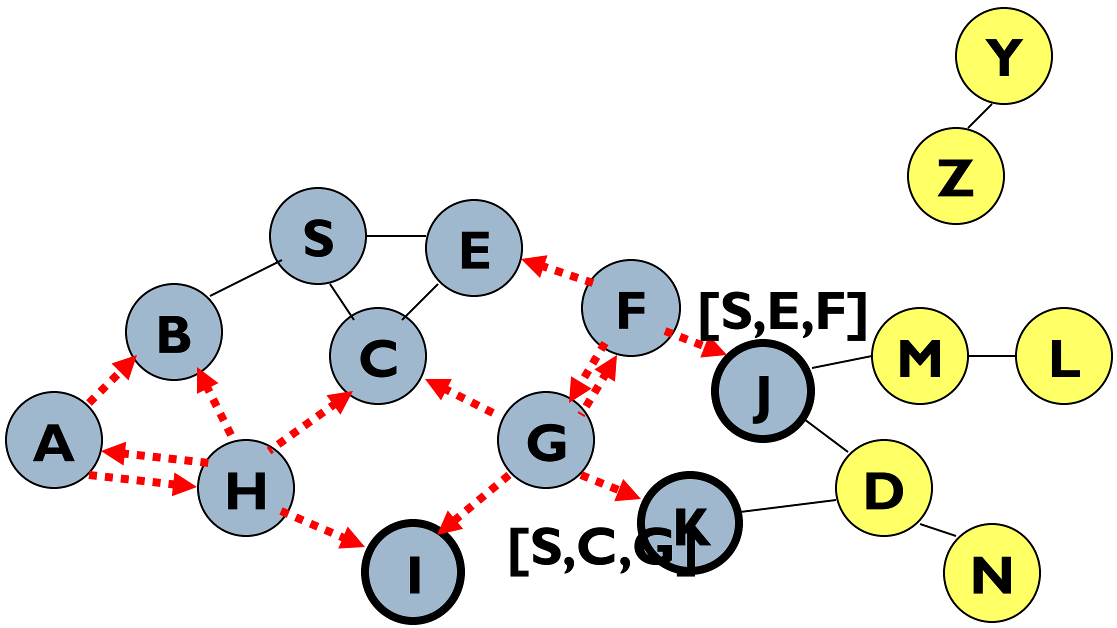
Route Discovery in DSR

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



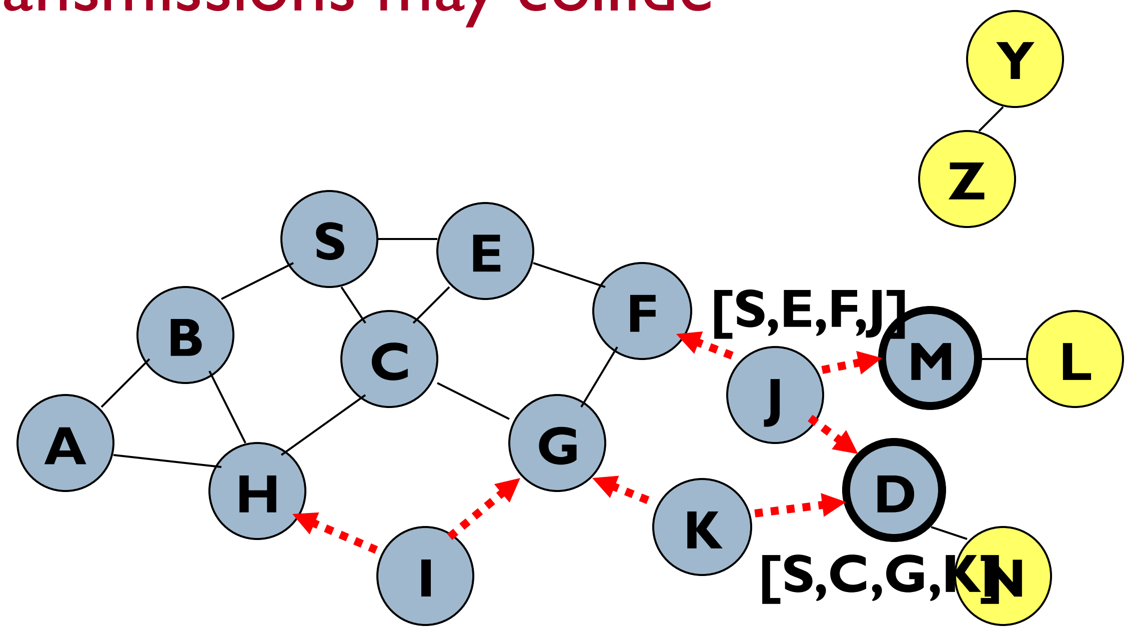
Route Discovery in DSR

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



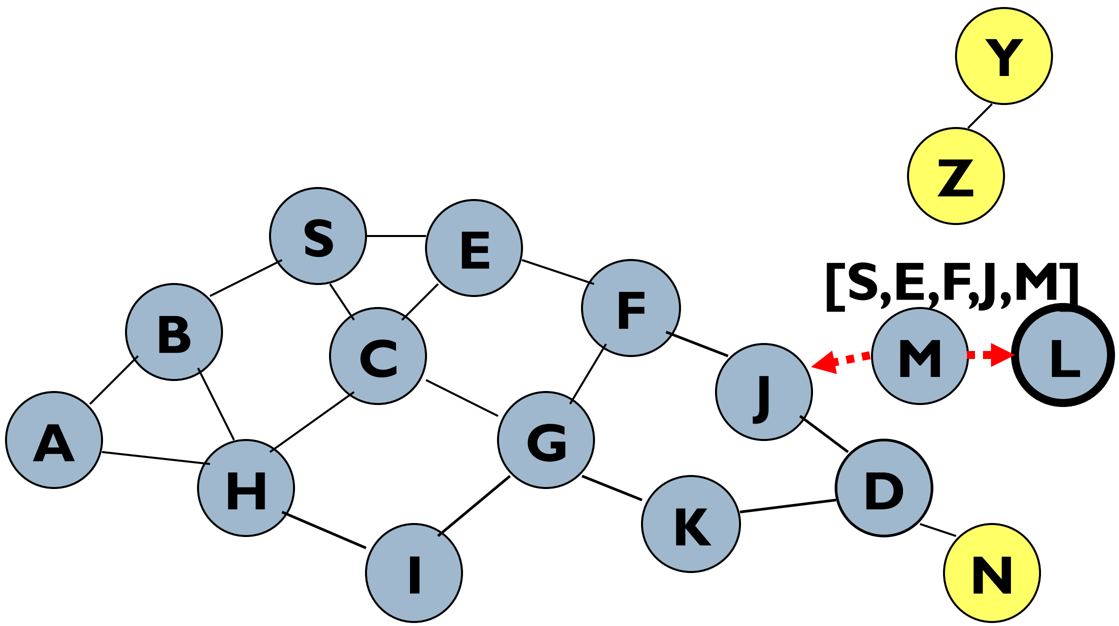
Route Discovery in DSR

- Nodes J and K both broadcast RREQ to node D
  - Since nodes J and K are hidden from each other, their transmissions may collide



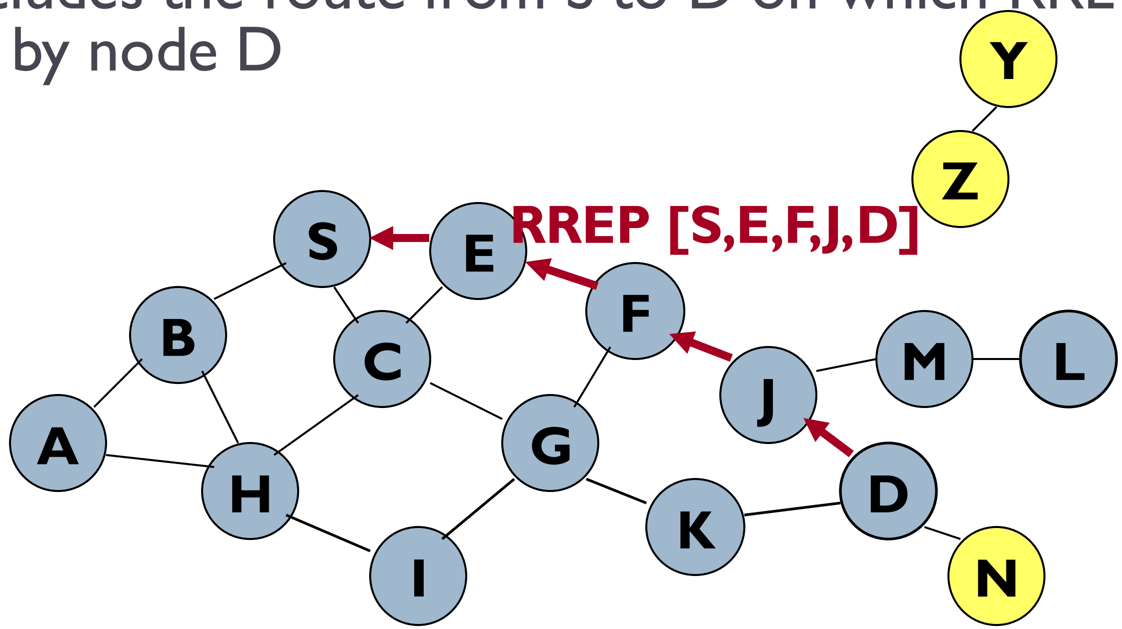
Route Discovery in DSR

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



Route Reply in DSR

- Destination D
  - On receiving the first RREQ, send 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

- Route Reply
  - Bi-directional links
    - Reverse route in Route Request (RREQ)
    - RREQ should be forwarded only if received on a link that is known to be bi-directional
  - Unidirectional (asymmetric) links
    - RREP may need a route discovery for S from node D
      - Route Reply is piggybacked on the Route Request from D
    - Unless node D already knows a route to node S

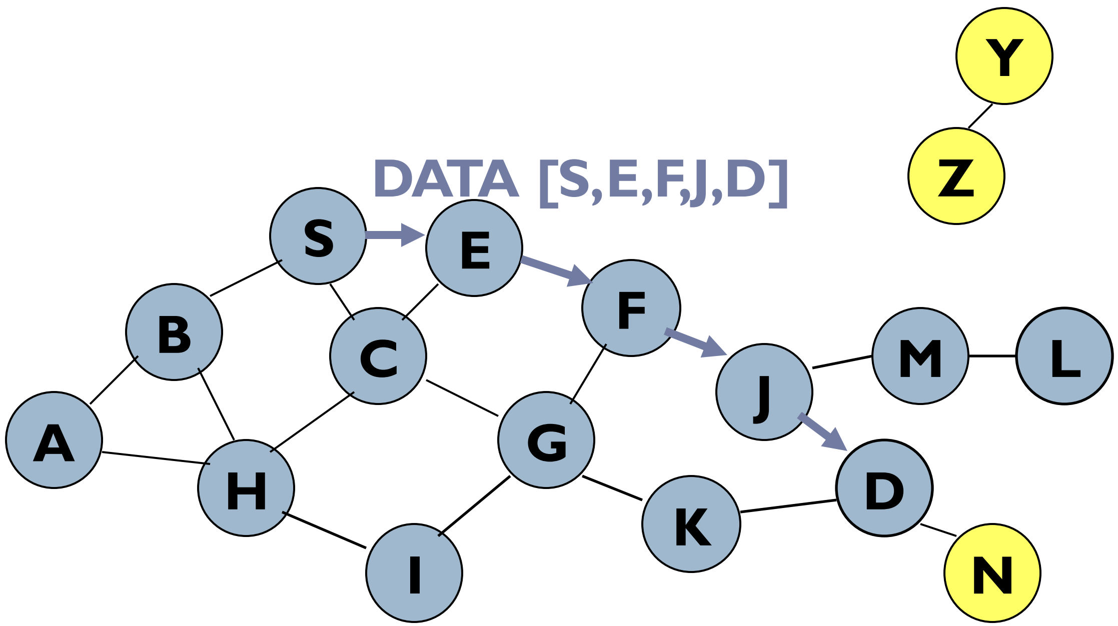
- IEEE 802.11 MAC
  - Links must be bi-directional (since ACK is used)

Dynamic Source Routing (DSR)

- On receiving RREP
  - Cache the route included in the RREP
- Sending
  - 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

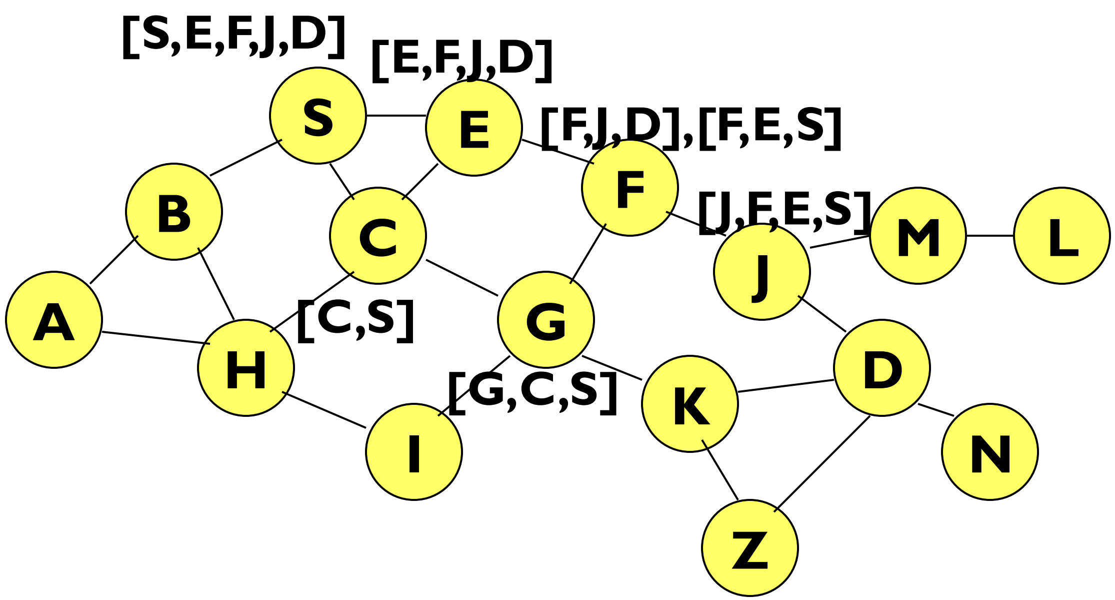
- Caching
  - Each node caches a new route it learns by any means
  - Snooping
    - A node may also learn a route when it overhears Data packets
- Use of Route Caching
  - Broken routes
    - Use another route from the local cache
    - Otherwise, initiate new route discovery
  - Intermediate response
    - On receiving a Route Request for some node D
      - Node X can send a Route Reply if node X knows a route to node D
  - Use of route cache
    - Speed up route discovery
    - Reduce propagation of route requests

Use of Route Caching

- Broken routes
  - Use another route from the local cache
  - Otherwise, initiate new route discovery
- Intermediate response
  - On receiving a Route Request for some node D
    - Node X can send a Route Reply if node X knows a route to node D
- Use of route cache
  - Speed up route discovery
  - Reduce propagation of route requests

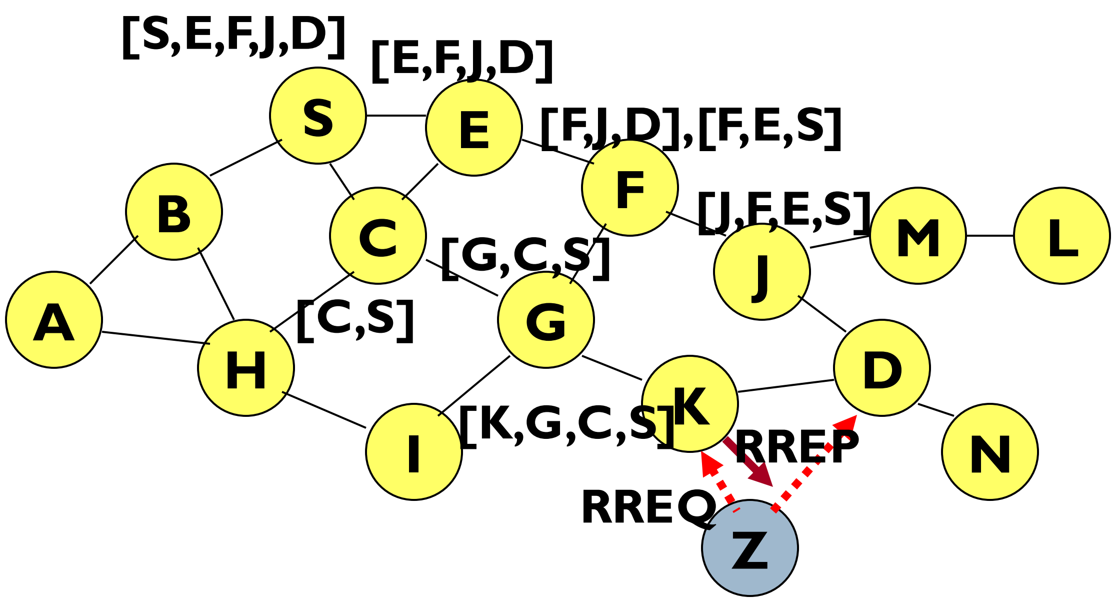
Use of Route Caching

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



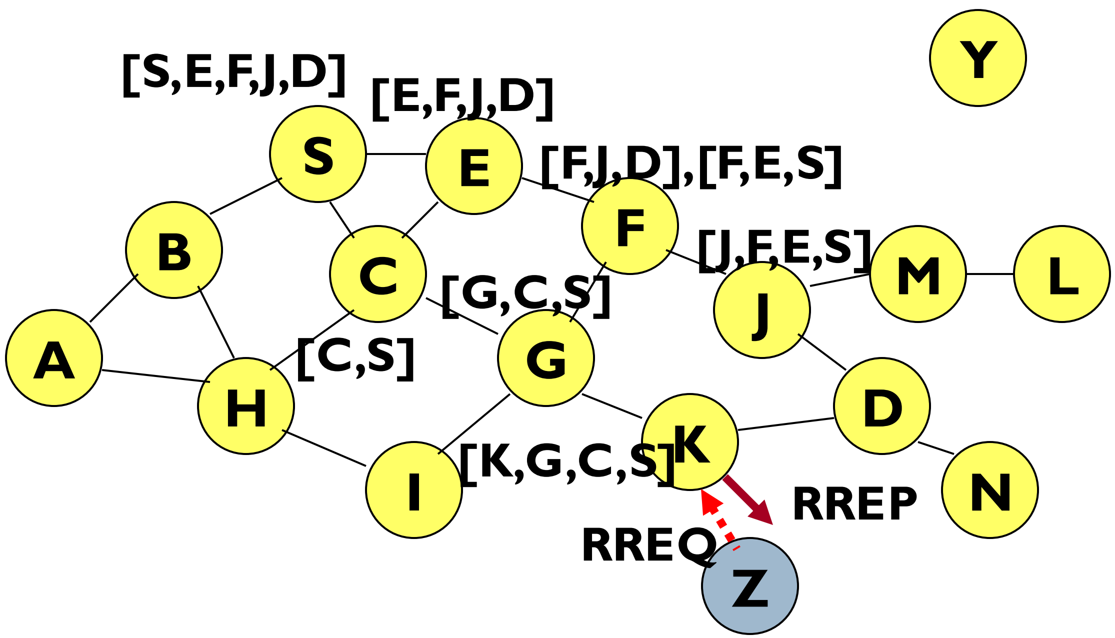
Use of Route Caching: Speed up Route Discovery

- 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



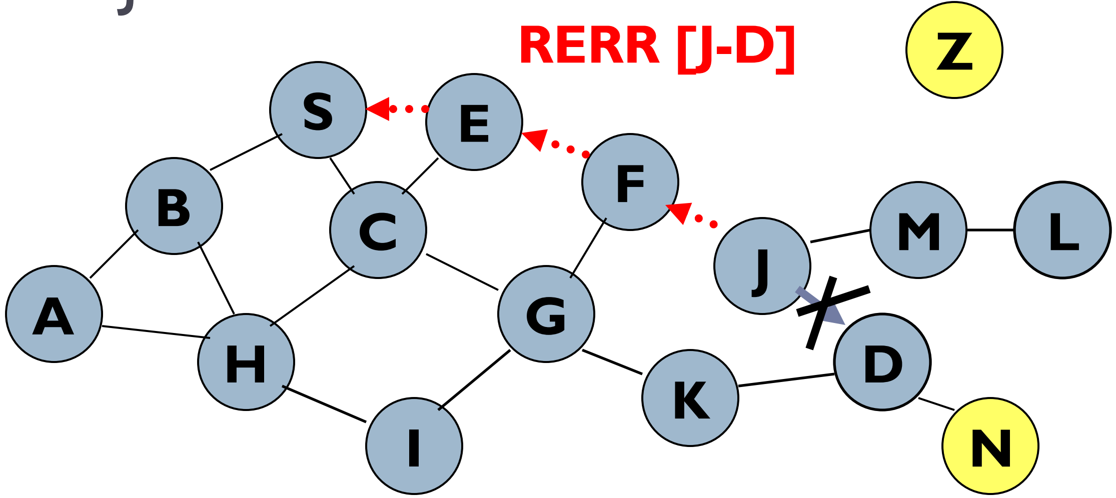
Use of Route Caching: Reduce of Route Requests

- 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)

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



Route Caching: Beware!

- Stale caches
  - Can adversely affect performance
- Timeliness
  - With passage of time and host mobility, cached routes may become invalid
- Know when to give up
  - A sender host may try several stale routes (obtained from local cache, or replied from cache by other nodes), before finding a good route

Dynamic Source Routing: Advantages

- On-demand
  - Routes maintained only between nodes that 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

- Size
  - Packet header size grows with route length
- Packets
  - Flood of route requests may reach all nodes
- Timing
  - Must avoid route requests collisions
    - Insertion of random delays before forwarding RREQ
  - Route Reply Storm problem
    - Too many nodes reply using local cache
    - Prevent a node from sending RREP if it hears another RREP with a shorter route

Dynamic Source Routing: Disadvantages

- Pollution
  - An intermediate node may send Route Reply using a stale cached route
  - Need some mechanism to purge (potentially) invalid cached routes
- For some proposals for cache invalidation
  - Static timeouts
  - Adaptive timeouts based on link stability

Flooding of Control Packets

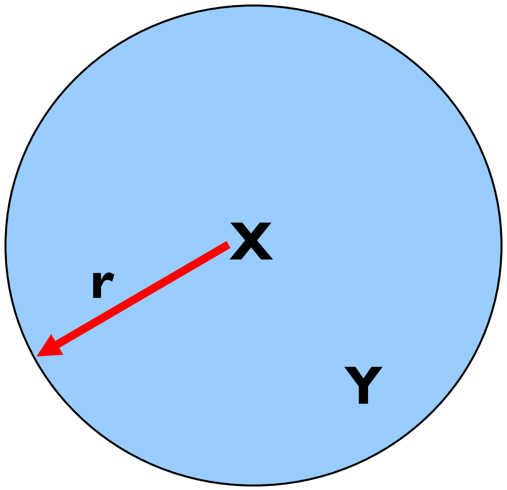
- How to reduce the scope of the route request flood ?
  - LAR
- How to reduce redundant broadcasts ?
  - The Broadcast Storm Problem

Location-Aided Routing (LAR)

- Exploit location information to limit scope of flood
  - Location information may be obtained using GPS
- Expected Zone
  - A region that is expected to hold the current location of the destination
  - 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

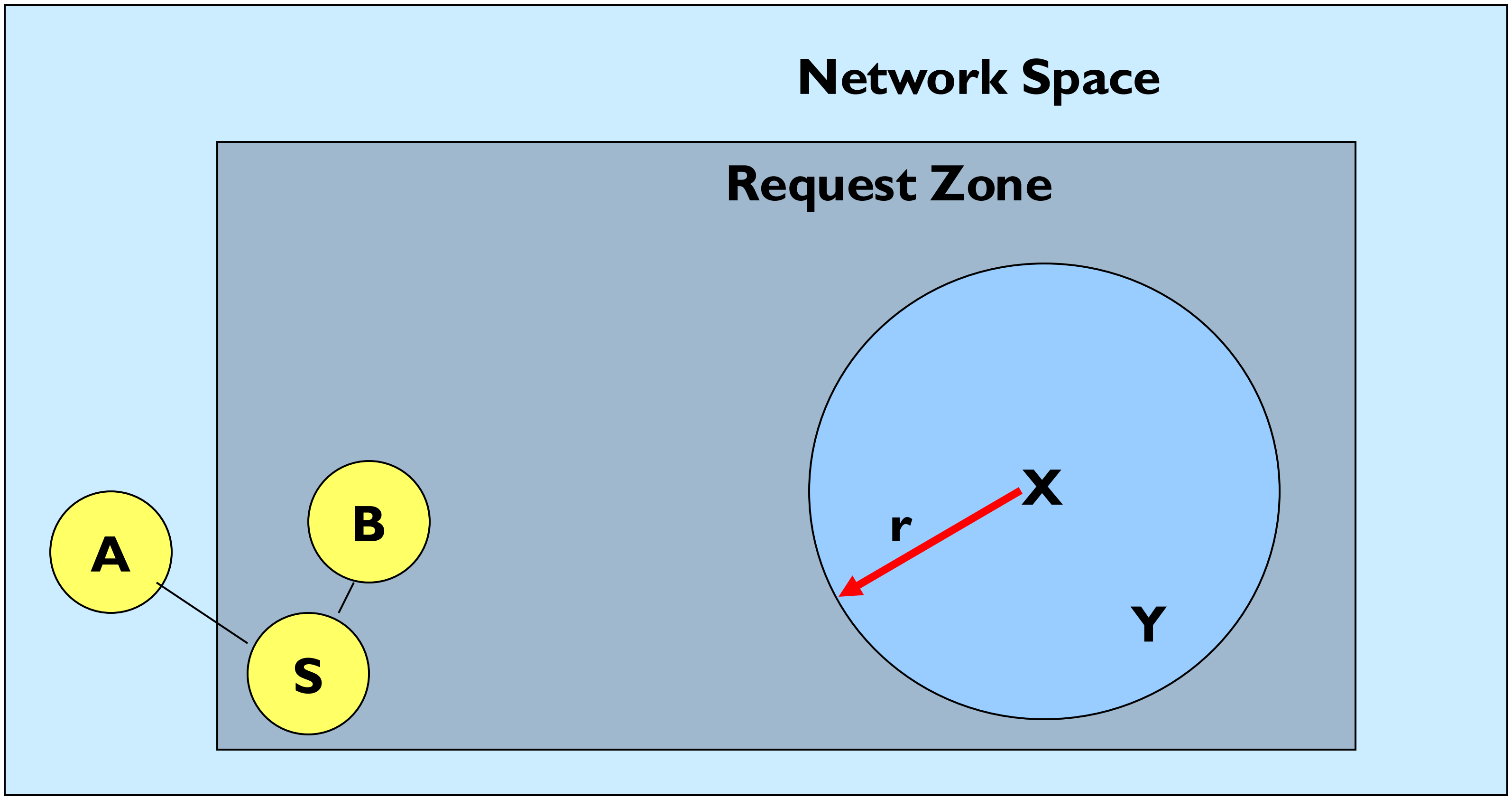
Expected Zone in LAR

- X = last known location of node D, at time t0
- Y = location of node D at current time t1, unknown to node S
- r = (t1 - t0) * estimate of D's speed



Expected Zone

Request Zone in LAR



LAR

- Zone
  - Explicitly specified in the route request
  - Each node must know its physical location to determine whether it is within the request zone
- Forwarding
  - Only nodes within the request zone forward route requests
- Failure
  - Initiate another route discovery (after a timeout) using a larger request zone
  - the larger request zone may be the entire network

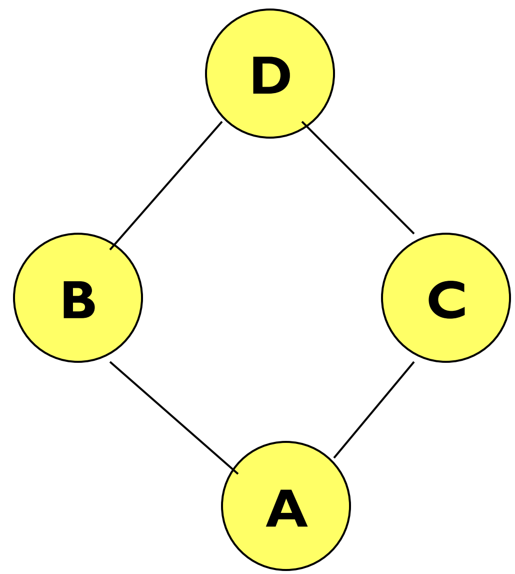
- Rest of route discovery protocol similar to DSR

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

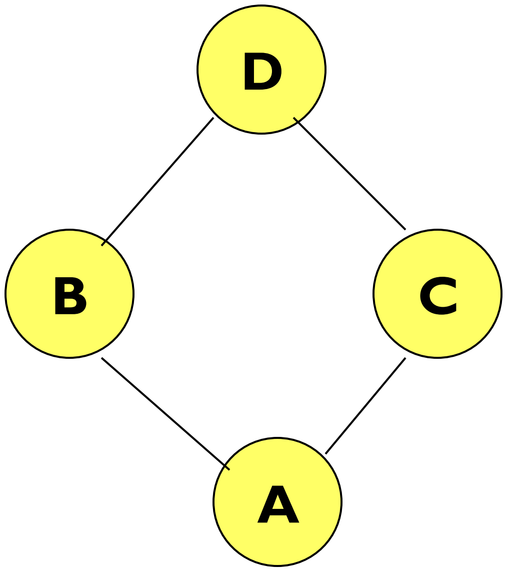
Broadcast Storm Problem

- 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



Solutions for Broadcast Storm

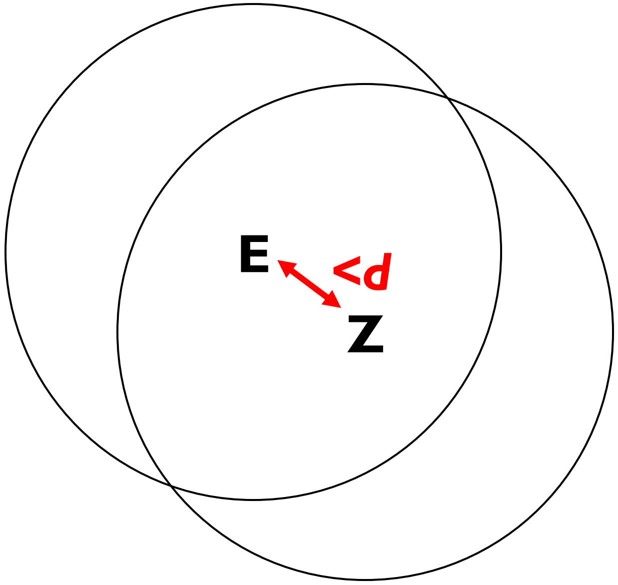
- Probabilistic scheme
  - Re-broadcast (forward) the request with probability p
  - Re-broadcasts by different nodes should be staggered by using a collision avoidance technique
  - Reduce the probability that nodes B and C would forward a packet simultaneously

Solutions for Broadcast Storm

- 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

Solutions for Broadcast Storm

- Distance-Based Scheme
  - If node E hears RREQ broadcasted by some node Z within physical distance d, then E will not re-broadcast the request
- Intuition
  - Z and E are close, so transmission areas covered by Z and E are not very different



Summary: Broadcast Storm Problem

- Flooding is used in many protocols, such as Dynamic Source Routing (DSR)
- Problems associated with flooding
  - 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

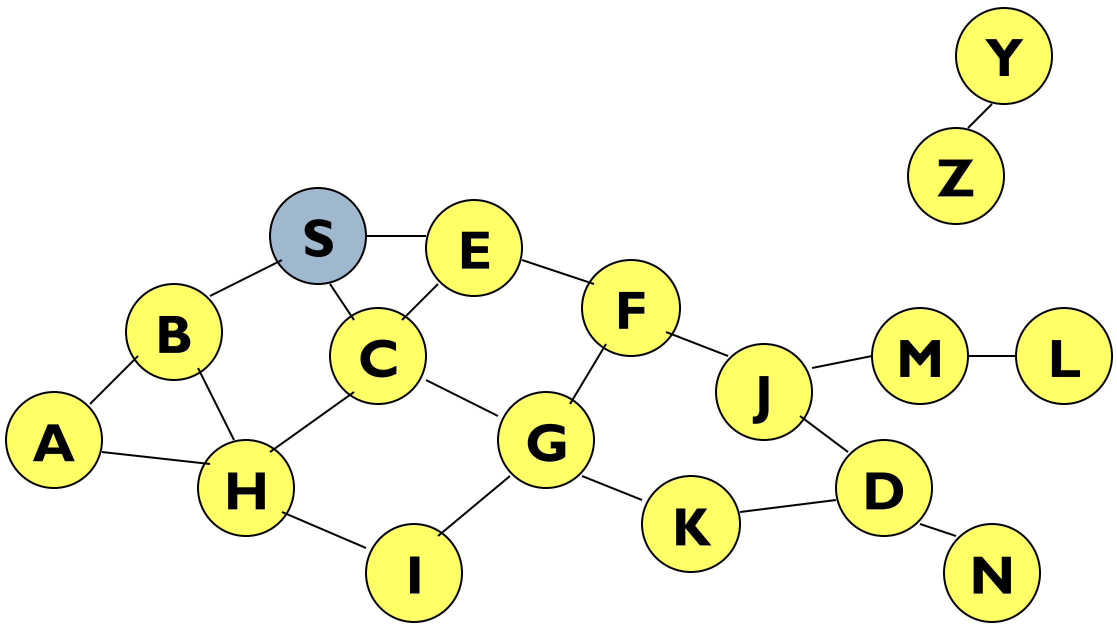
Ad Hoc On-Demand Distance Vector Routing (AODV)

- Source routing
  - Large headers
  - Particularly when data contents of a packet are small
- AODV
  - Maintaining routing tables at the nodes
  - Routes are maintained only between nodes which need to communicate

AODV

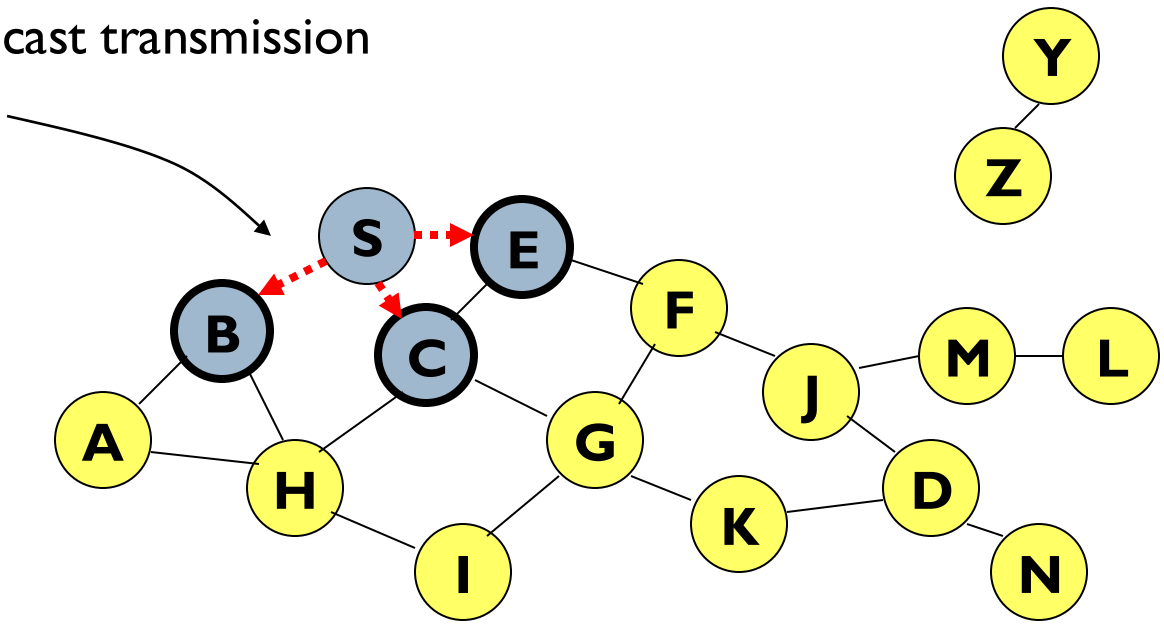
- Route Requests (RREQ)
  - Forwarded in a manner similar to DSR
- Routes
  - When a node re-broadcasts a Route Request, it sets up a reverse path pointing towards the source
  - AODV assumes symmetric (bi-directional) links
- Destination
  - Destination replies to Route Request with a Route Reply
- Route Reply
  - Follows reverse path set-up by Route Request

Route Requests in AODV

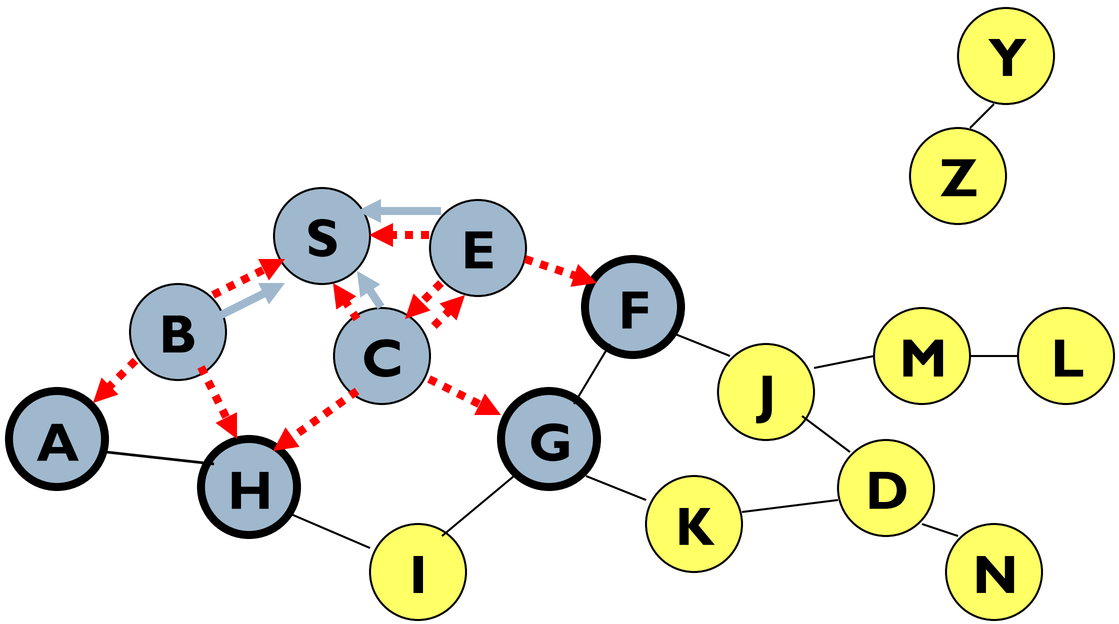


Route Requests in AODV

Broadcast transmission

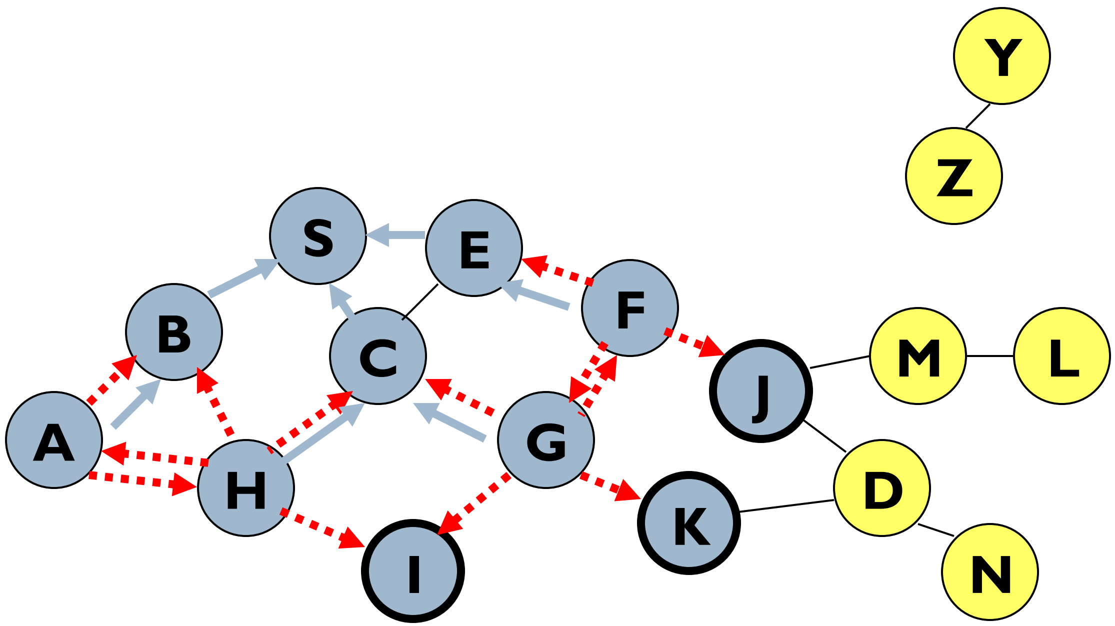


Route Requests in AODV

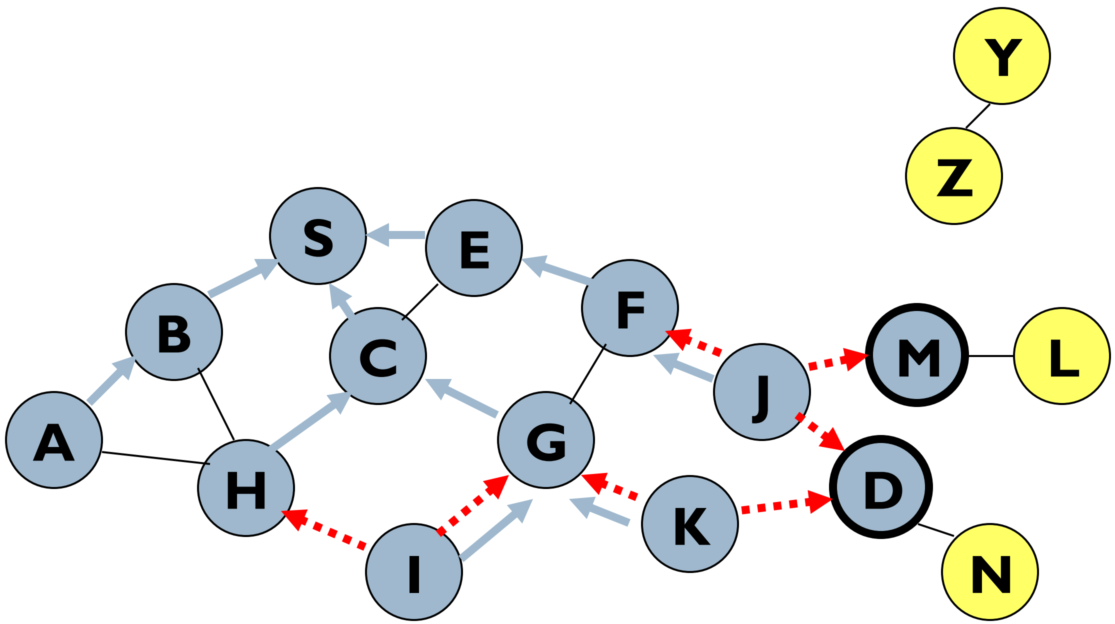


Reverse Path Setup in AODV

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

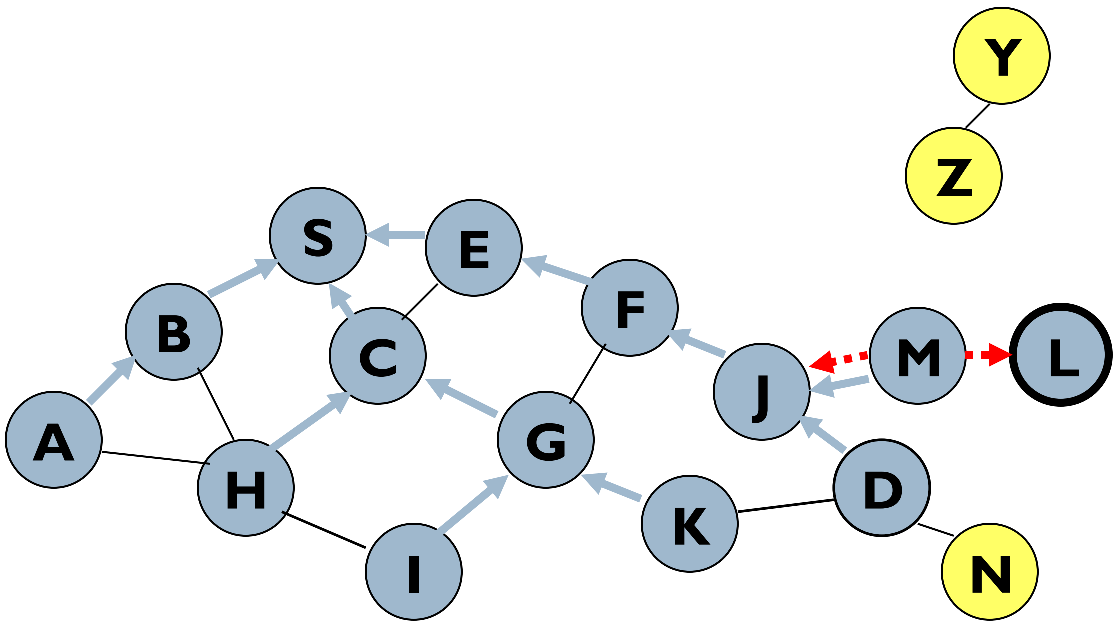


Reverse Path Setup in AODV

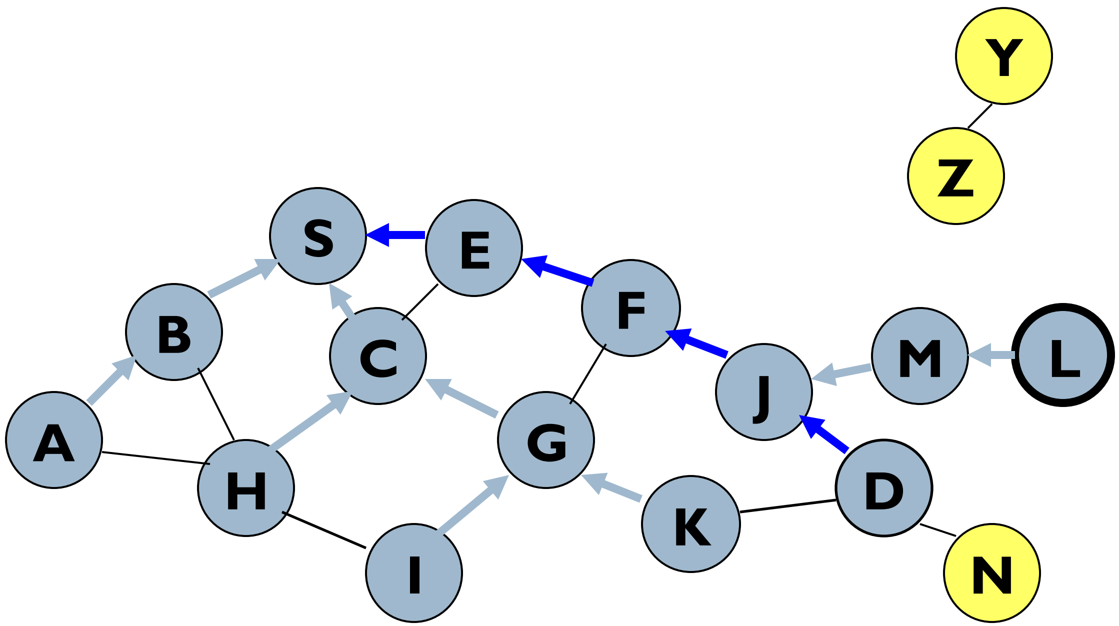


Reverse Path Setup in AODV

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



Route Reply in AODV

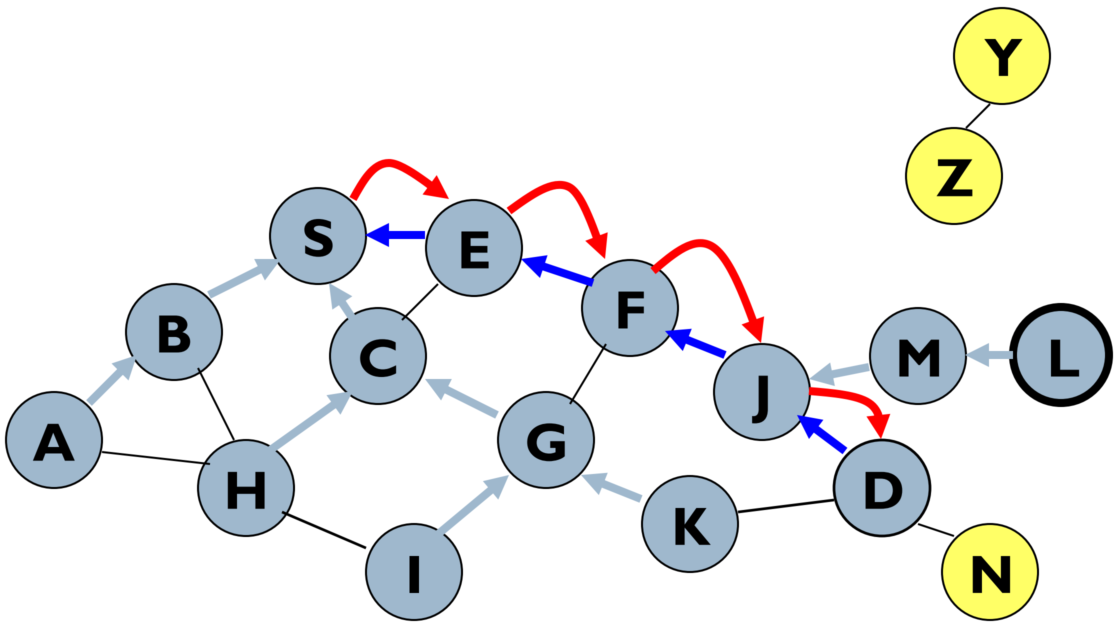


Route Reply in AODV

- Intermediate node reply
  - Send a Route Reply (RREP) if it knows a more recent path than the one previously known to sender
- Sequence Numbers
  - Destination sequence numbers are used to determine age
- Fewer intermediate replies than DSR
  - A new Route Request for a destination is assigned a higher destination sequence number
  - An intermediate node that knows a route with a smaller sequence number cannot send Route Reply

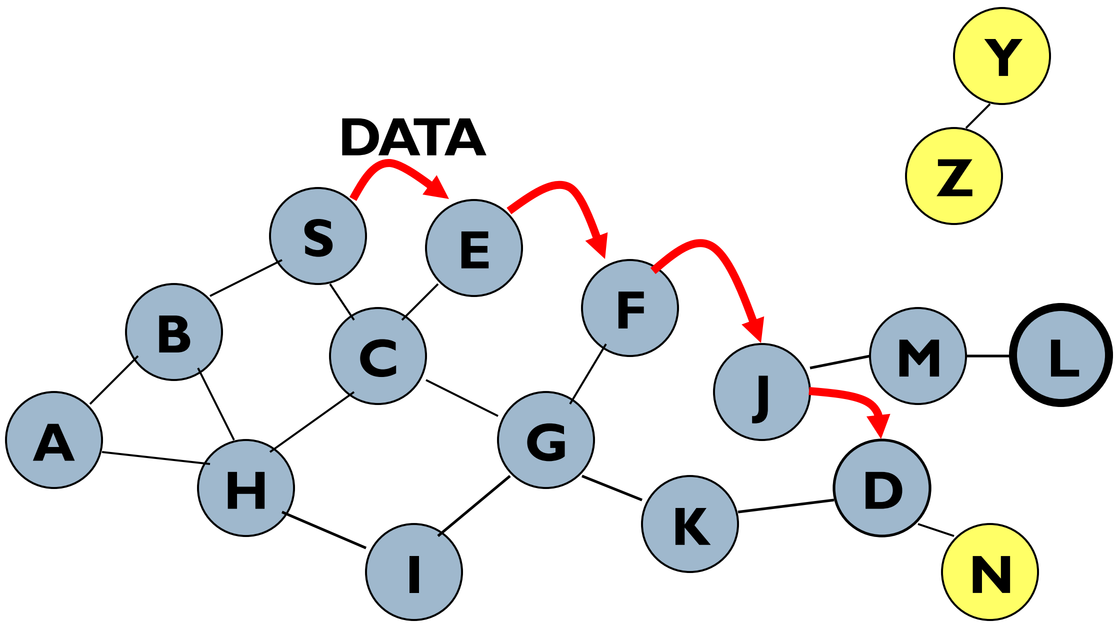
Forward Path Setup in AODV

- Forward links are setup when RREP travels along the reverse path



Data Delivery in AODV

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



Timeouts

- Routing table entries
  - Reverse Paths
    - Purged after a timeout interval
    - Timeout should be long enough to allow RREP to come back
  - Forward Paths
    - If no is data being sent using a particular routing table entry
      - Entry is deleted from the routing table (even if the route may actually still be valid)

Link Failure Reporting

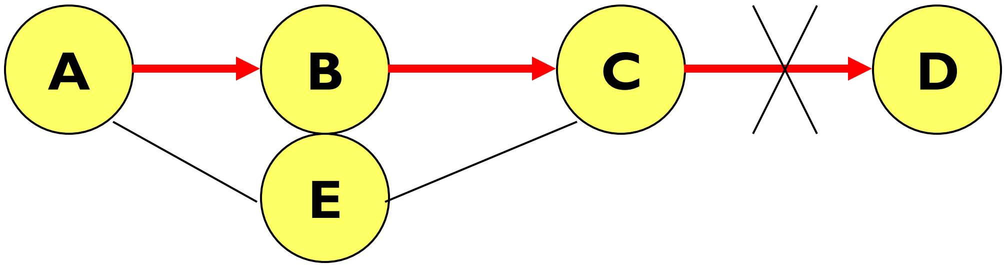
- Link Failure
  - When the next hop link in a routing table entry breaks, all active neighbors are informed
  - Active neighbors
    - Any neighbor that sent a packet within active_route_timeout interval which was forwarded using that entry
- Link failures
  - Propagated by means of Route Error messages
  - Also update destination sequence numbers

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

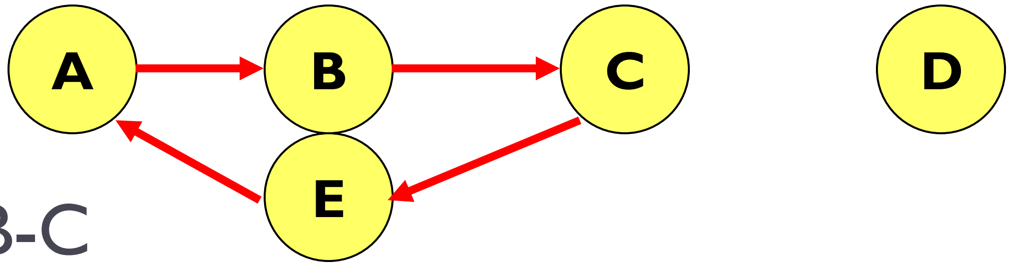
Why Sequence Numbers in AODV

- To avoid using old/broken routes
  - To determine which route is newer
- To prevent formation of loops



- RERR sent by C is lost
  - A does not know about failure of link C-D
- C performs a route discovery for D
  - Node A receives the RREQ (say, via path C-E-A)
- Node A replies 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
  - Initially sent with small Time-to-Live (TTL) field, to limit propagation
  - DSR also includes a similar optimization
- If no Route Reply is received
  - Larger TTL

Summary: AODV

- Routes need not be included in packet headers
- Nodes maintain routing tables
  - 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

Routing Metrics

Power-Aware Routing

- Define optimization criteria as a function of energy consumption
- Examples
  - Minimize energy consumed per packet
  - Minimize time to network partition due to energy depletion
  - Maximize duration before a node fails due to energy depletion

Power-Aware Routing

- Assign a weight to each link
- Weight of a link may be a function of
  - Energy consumed when transmitting a packet
  - Residual energy level
  - Low residual energy level may correspond to a high cost
- Prefer a route with the smallest aggregate weight

Link Stability-Based Routing

- Idea
  - A node X re-broadcasts a Route Request received from Y only if the (X,Y) link is deemed to have a strong signal stability
- Signal stability
  - Evaluated as a moving average of the signal strength of packets received on the link in recent past
- Alternative approach
  - Assign a cost as a function of signal stability

Connection Stability-Based Routing

- Only utilize links that have been stable for some minimum duration
  - If a link has been stable beyond some minimum threshold
    - It is likely to be stable for a longer interval
  - If it has not been stable longer than the threshold
    - It may soon break (could be a transient link)
- Prefer paths with high aggregate stability

Expected Transmission Count (ETX)

- Link Metric
  - Fewest expected retransmissions
  - Probe packets measure the packet delivery ratio in both forward and reverse directions
- Path Metric
  - Sum of the individual link ETX values

Expected Transmission Count (ETX)

- Forward delivery ratio = probability of a packet arriving at the destination
- Reverse delivery ratio = probability of a receiver's acknowledgment (ACK) arriving back at the sender.
- Link ETX = 1/(Forward delivery ratio x Reverse delivery ratio)

Expected Transmission Count (ETX)

- Strength
  - Significant improvement over simple hop-count metrics by preferring reliable, low-loss links.
- Limitation
  - Designed for single-radio networks
  - Does not account for differences in link bandwidth or the effects of inter-channel interference in multi-radio networks

IEEE 802.11s: WLAN Mesh

- Routing
  - Hybrid Wireless Mesh Protocol
    - Ad hoc On-Demand Distance Vector (AODV) Routing
    - Tree-based routing
  - Metrics
    - Airtime Link Metric (ALM)
      - Time spent transmitting data and the packet error rate on a link.

Weighted Cumulative Expected Transmission Time (WCETT)

- Goal
  - Minimize total expected transmission time.
- Metric
  - Link Metric
    - Expected Transmission Time (ETT), which incorporates both the link loss rate and bandwidth
    - Link ETT = (Link ETX) * (Packet Size / Bandwidth)
  - Intra-channel interference
    - The sum of the Expected Transmission Times (ETTs) of links that share the same wireless channel
  - Path Metric:
    - The cumulative ETT of all links on the path