Mobile Ad Hoc Networks

Outline

- Introduction
- Medium Access Control
- Routing (unicast)
  - Reactive Protocols
  - Proactive Protocols
  - Hybrid Protocols
- Transport Issues
- Summary and Conclusions

Wireless Networks

- Need: Access computing and communication services, on the move

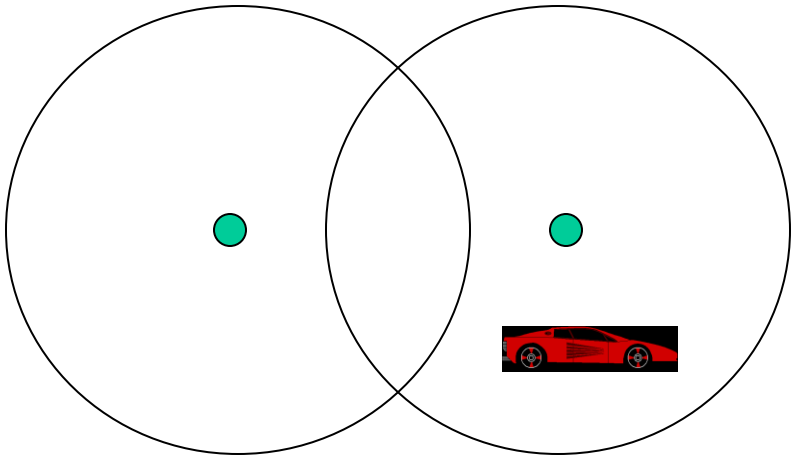
- Infrastructure-based Networks
  - traditional cellular systems (base station infrastructure)

- Wireless LANs
  - Infrared (IrDA) or radio links (Wavelan)
  - very flexible within the reception area; ad-hoc networks possible
  - low bandwidth compared to wired networks (1-10 Mbit/s)

- Ad hoc Networks
  - useful when infrastructure not available, impractical, or expensive
  - military applications, rescue, home networking

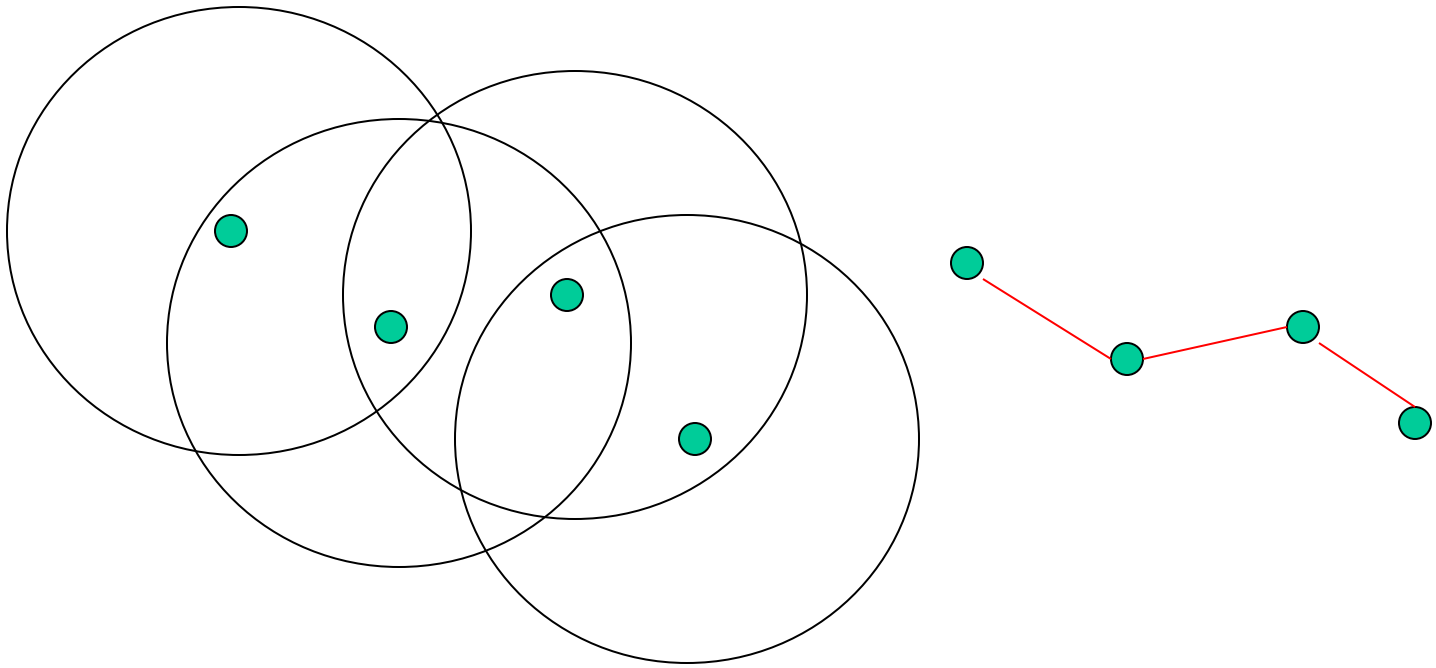
Cellular Wireless

- Single hop wireless connectivity to the wired world
  - Space divided into cells
  - A base station is responsible to communicate with hosts in its cell
  - Mobile hosts can change cells while communicating
  - Hand-off occurs when a mobile host starts communicating via a new base station



Multi-Hop Wireless

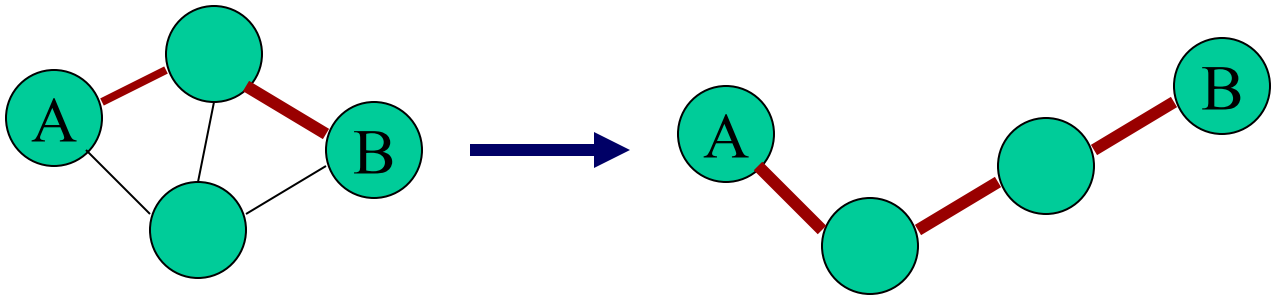
- May need to traverse multiple links to reach destination



- Mobility causes route changes

Mobile Ad Hoc Networks (MANET)

- Host movement frequent
- Topology change frequent



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

Why Ad Hoc Networks ?

- Setting up of fixed access points and backbone infrastructure is not always viable
  - Infrastructure may not be present in a disaster area or war zone
  - Infrastructure may not be practical for short-range radios; Bluetooth (range ~ 10m)

- Ad hoc networks:
  - Do not need backbone infrastructure support
  - Are easy to deploy
  - Useful when infrastructure is absent, destroyed or impractical

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

Challenges in Mobile Environments

- Limitations of the Wireless Network
  - packet loss due to transmission errors
  - variable capacity links
  - frequent disconnections/partitions
  - limited communication bandwidth
  - Broadcast nature of the communications
- Limitations Imposed by Mobility
  - dynamically changing topologies/routes
  - lack of mobility awareness by system/applications
- Limitations of the Mobile Computer
  - short battery lifetime
  - limited capacities

Effect of mobility on the protocol stack

- Application
  - new applications and adaptations
- Transport
  - congestion and flow control
- Network
  - addressing and routing
- Link
  - media access and handoff
- Physical
  - transmission errors and interference

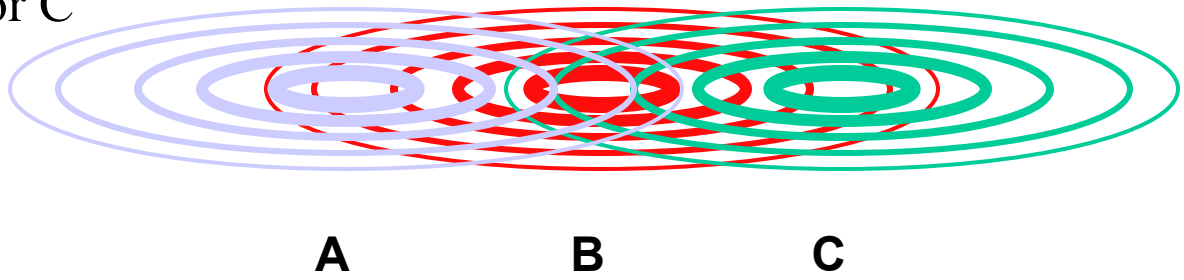
Medium Access Control in MANET

Motivation

- Can we apply media access methods from fixed networks?
- Example CSMA/CD
  - **C**arrier **S**ense **M**ultiple **A**ccess with **C**ollision **D**etection
  - send as soon as the medium is free, listen into the medium if a collision occurs (original method in IEEE 802.3)
- Medium access problems in wireless networks
  - signal strength decreases proportional to the square of the distance
  - sender would apply CS and CD, but the collisions happen at the receiver
  - sender may not “hear” the collision, i.e., CD does not work
  - CS might not work, e.g. if a terminal is “hidden”

Hidden and Exposed Terminals

- Hidden terminals
  - A sends to B, C cannot receive A
  - C wants to send to B, C senses a “free” medium (CS fails)
  - collision at B, A cannot receive the collision (CD fails)
  - A is “hidden” for C



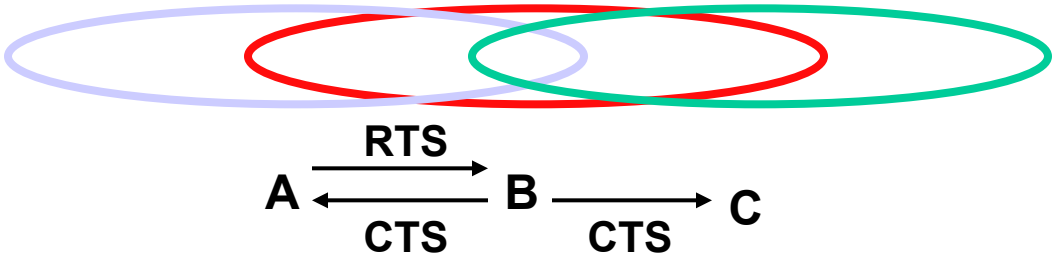
- Exposed terminals
  - B sends to A, C wants to send to another terminal (not A or B)
  - C senses carrier, finds medium in use and has to wait
  - A is outside the radio range of C, therefore waiting is not necessary
  - C is “exposed” to B

Multiple Access with Collision Avoidance (MACA) [Karn90]

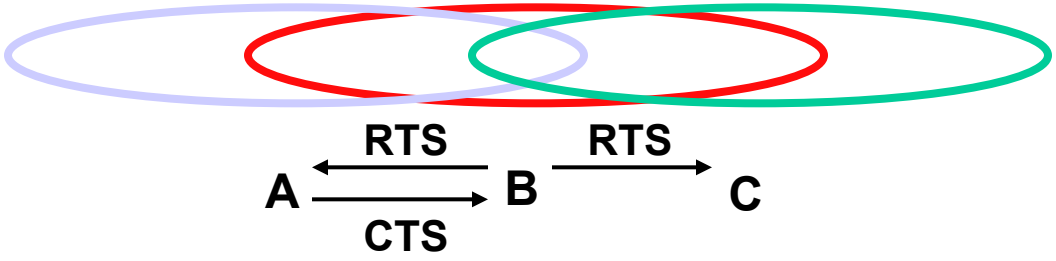
- MACA uses signaling packets for collision avoidance
  - RTS (request to send)
    - sender request the right to send from a receiver with a short RTS packet before it sends a data packet
  - CTS (clear to send)
    - receiver grants the right to send as soon as it is ready to receive
- Signaling packets contain
  - sender address
  - receiver address
  - packet size
- Variants of this method are used in IEEE 802.11

MACA Solutions [Karn90]

- MACA avoids the problem of hidden terminals
  - A and C want to send to B
  - A sends RTS first
  - C waits after receiving CTS from B



- MACA avoids the problem of exposed terminals
  - B wants to send to A, C to another terminal
  - now C does not have to wait, as it cannot receive CTS from A



MAC: Reliability

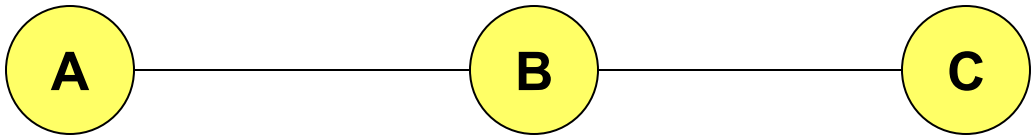
- Wireless links are prone to errors. High packet loss rate is detrimental to transport-layer performance.

- Solution: Use of acknowledgements
  - When node B receives a data packet from node A, node B sends an Acknowledgement (Ack).
  - If node A fails to receive an Ack, it will retransmit the packet
  - This approach adopted in many protocols [Bharghavan94, IEEE 802.11]

- IEEE 802.11 Wireless MAC
  - Distributed and centralized MAC components
    - Distributed Coordination Function (DCF)
    - Point Coordination Function (PCF)
  - DCF suitable for multi-hop ad hoc networking

IEEE 802.11 DCF

- Uses RTS-CTS exchange to avoid hidden terminal problem
  - Any node overhearing a CTS cannot transmit for the duration of the transfer
- Uses ACK to achieve reliability
- Any node receiving the RTS cannot transmit for the duration of the transfer
  - To prevent collision with ACK when it arrives at the sender
  - When B is sending data to C, node A will keep quiet



MAC: Collision Avoidance

- With half-duplex radios, collision detection is not possible
- Collision avoidance: Once channel becomes idle, the node waits for a randomly chosen duration before attempting to transmit

- IEEE 802.11 DCF
  - When transmitting a packet, choose a backoff interval in the range [0,cw]; cw is contention window
  - Count down the backoff interval when medium is idle
  - Count-down is suspended if medium becomes busy
  - When backoff interval reaches 0, transmit RTS

- Time spent counting down backoff intervals is a part of MAC overhead
- *large cw* leads to larger backoff intervals
- *small cw* leads to larger number of collisions

MAC: Congestion Control

- IEEE 802.11 DCF: Congestion control achieved by dynamically choosing the contention window *cw*

- Binary Exponential Backoff in DCF:
  - When a node fails to receive CTS in response to its RTS, it increases the contention window
    - *cw* is doubled (up to an upper bound)
  - When a node successfully completes a data transfer, it restores *cw* to *CWmin*

MAC: Energy Conservation

- Proposals typically suggest turning the radio off when not needed

- Power Saving Mode in IEEE 802.11 (Infrastructure Mode)
  - An Access Point periodically transmits a beacon indicating which nodes have packets waiting for them
  - Each power saving (PS) node wakes up periodically to receive the beacon
  - If a node has a packet waiting, then it sends a PS-Poll
    - After waiting for a backoff interval in [0,CWmin]
  - Access Point sends the data in response to PS-poll

MAC Protocols: Summary

- Wireless medium is prone to hidden and exposed terminal problems

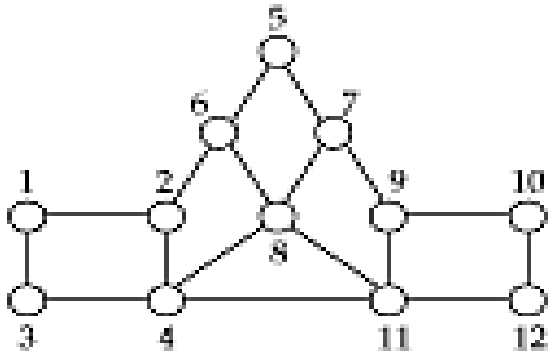
- Protocols are typically based on CSMA/CA
- RTS/CTS based signaling
- Acks for reliability

- Contention window is used for congestion control
- IEEE 802.11 wireless LAN standard
- Fairness issues are still unclear

Routing Protocols

Traditional Routing

- A *routing protocol* sets up a *routing table* in *routers*



ROUTING TABLE AT 1

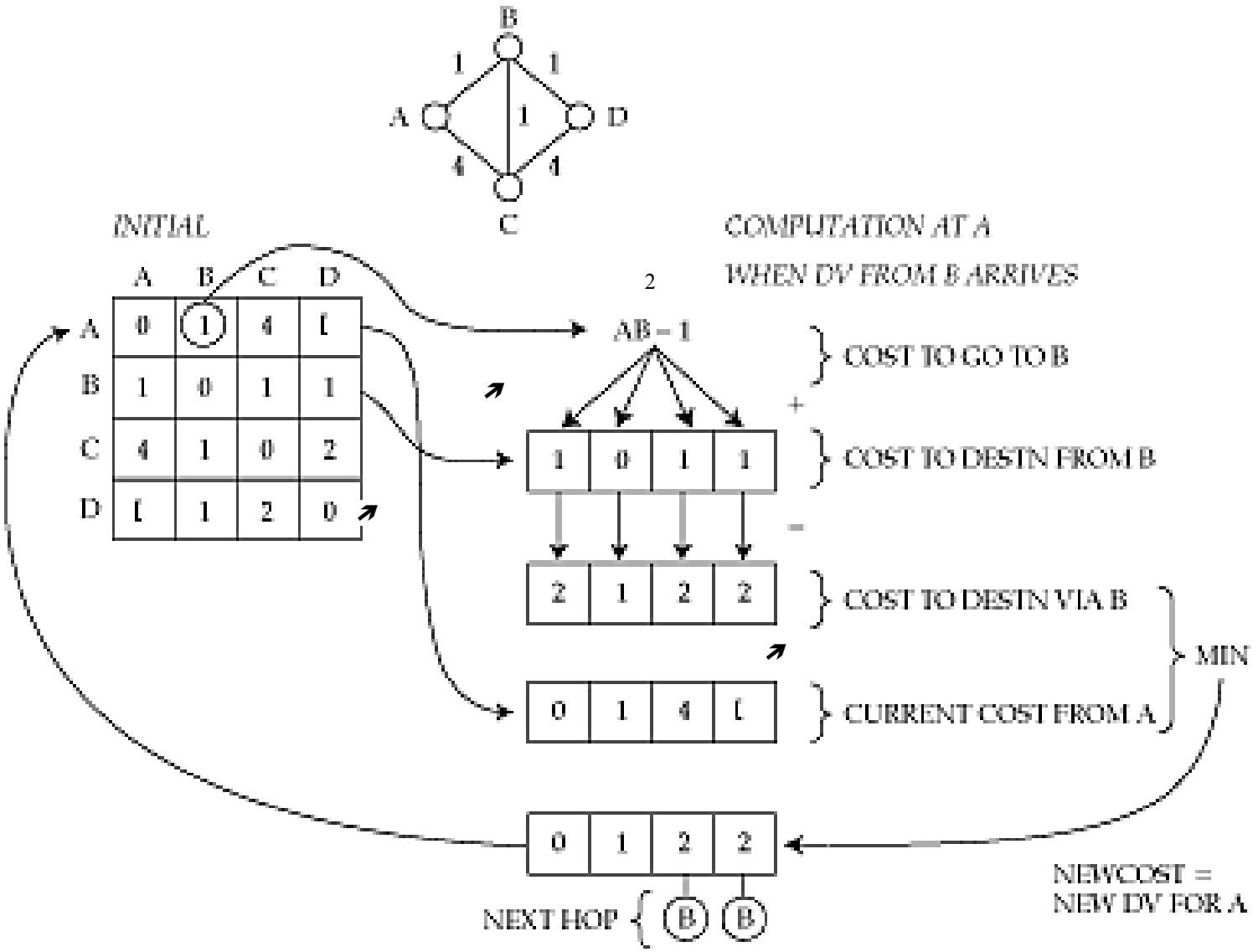
Destination	Next hop	Destination	Next hop
1	—	7	2
2	2□	8□	2□
3	3□	9□	2□
4	3□	10□	2□
5	2□	11□	3□
6	2	12	3

- A node makes a *local* choice depending on *global* topology

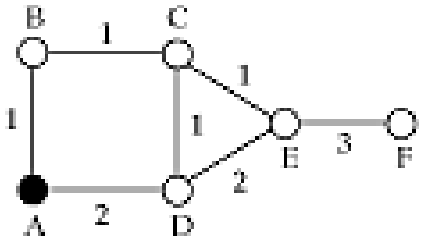
Distance-vector & Link-state Routing

- Both assume router knows
  - address of each neighbor
  - cost of reaching each neighbor
- Both allow a router to determine global routing information by talking to its neighbors
- Distance vector - router knows cost to each destination
- Link state - router knows entire network topology and computes shortest path

Distance Vector Routing: Example

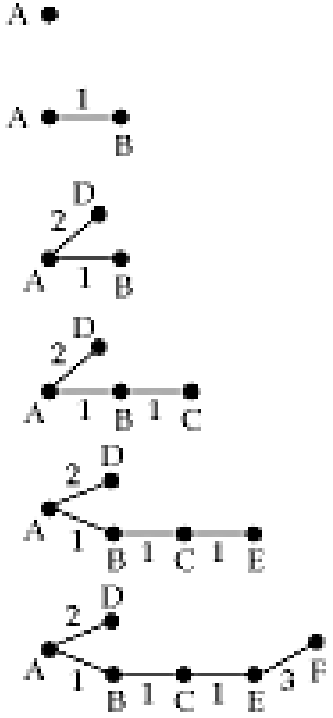


Link State Routing: Example



B(A,1) means B was reached by A, cost 1

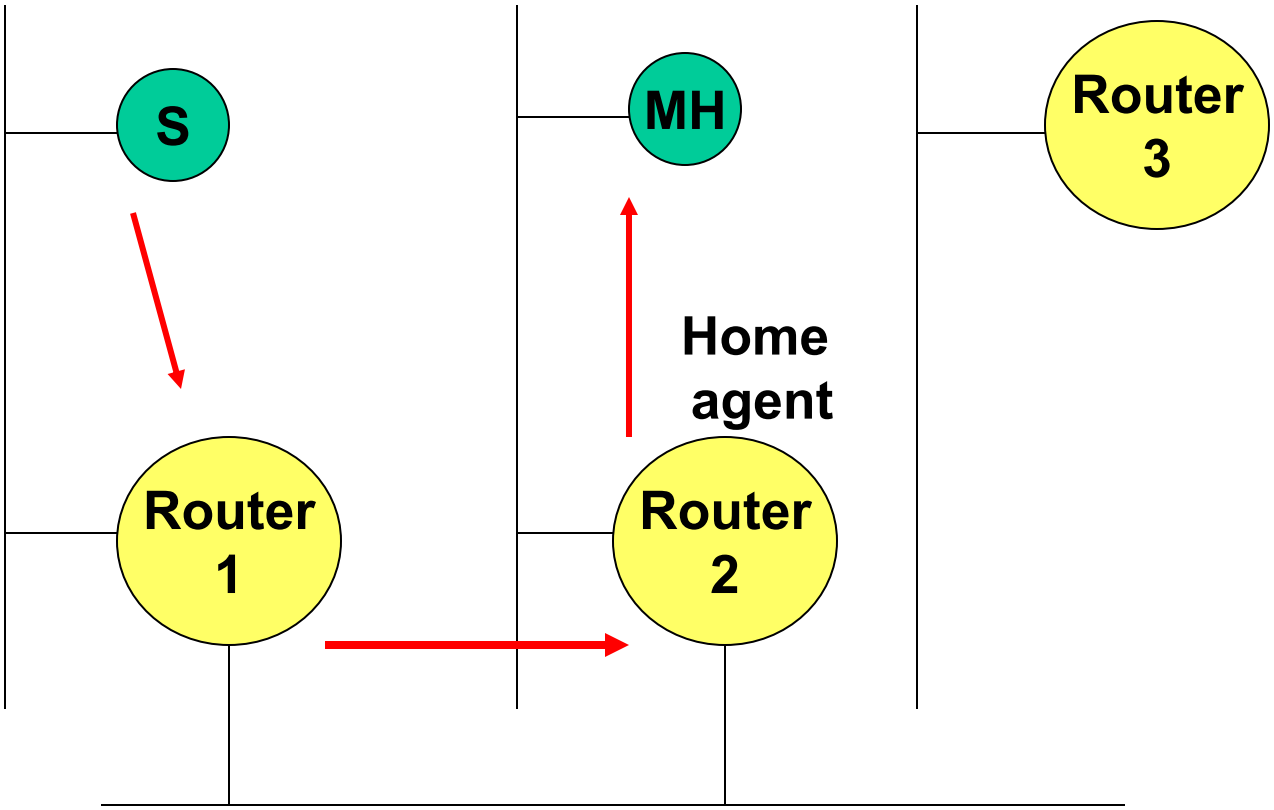
PERMANENT	TEMPORARY	COMMENTS
A	B(A,1), D(A,2)	ROOT AND ITS NEIGHBORS
A, B(A,1)	D(A,2), C(B,2)	ADD C(B,2)
A, B(A,1) D(A,2)	E(D,4), C(B,2)	C(D,3) DIDN'T MAKE IT
A, B(A,1) D(A,2), C(B,2)	E(C,3)	E(D,4) TOO LONG
A, B(A,1) D(A,2), C(B,2) E(C,3)	F(E,6)	
A, B(A,1) C(B,2), D(A,2) E(C,3), F(E,6)	NULL	STOP



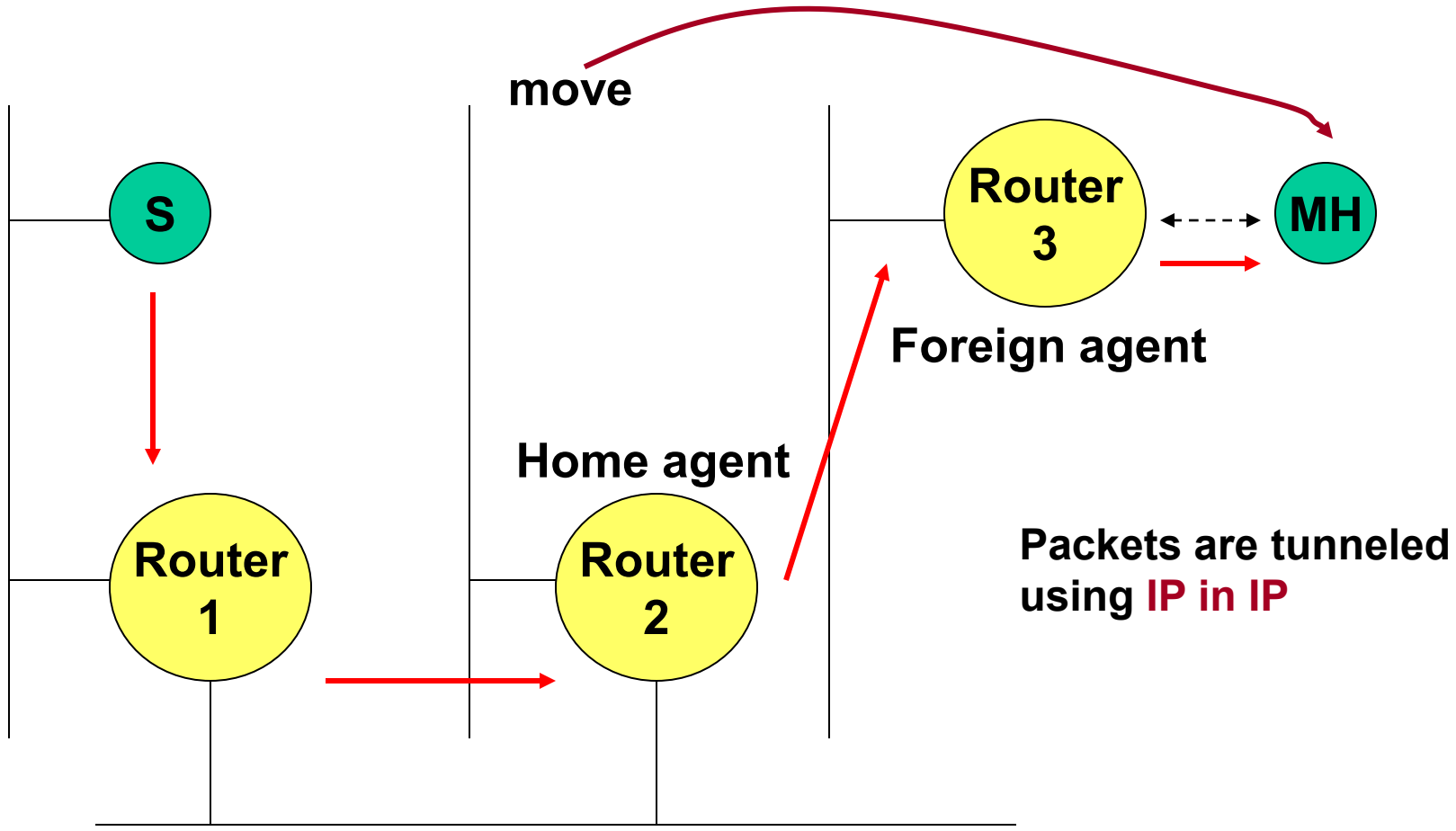
Routing and Mobility

- Finding a path from a source to a destination
- Issues
  - Frequent route changes
    - amount of data transferred between route changes may be much smaller than traditional networks
  - Route changes may be related to host movement
  - Low bandwidth links
- Goal of routing protocols
  - decrease routing-related overhead
  - find short routes
  - find “stable” routes (despite mobility)

Mobile IP



Mobile IP


move
S
Router 3
MH
Foreign agent
Home agent
Router 1
Router 2
Packets are tunneled using IP in IP

Routing in MANET

Unicast Routing Protocols

- Many protocols have been proposed

- Some specifically invented for MANET
- Others adapted from protocols for wired networks

- No single protocol works well in all environments
  - some attempts made to develop adaptive/hybrid protocols

- Standardization efforts in IETF
  - MANET, MobileIP working groups
  - **http://www.ietf.org**

Routing Protocols

- Proactive protocols
  - Traditional distributed shortest-path protocols
  - Maintain routes between every host pair at all times
  - Based on periodic updates; High routing overhead
  - Example: DSDV (destination sequenced distance vector)

- Reactive protocols
  - Determine route if and when needed
  - Source initiates route discovery
  - Example: DSR (dynamic source routing)

- Hybrid protocols
  - Adaptive; Combination of proactive and reactive
  - Example : ZRP (zone routing protocol)

Protocol Trade-offs

- Proactive protocols
  - Always maintain routes
  - Little or no delay for route determination
  - Consume bandwidth to keep routes up-to-date
  - Maintain routes which may never be used

- Reactive protocols
  - Lower overhead since routes are determined on demand
  - Significant delay in route determination
  - Employ flooding (global search)
  - Control traffic may be bursty

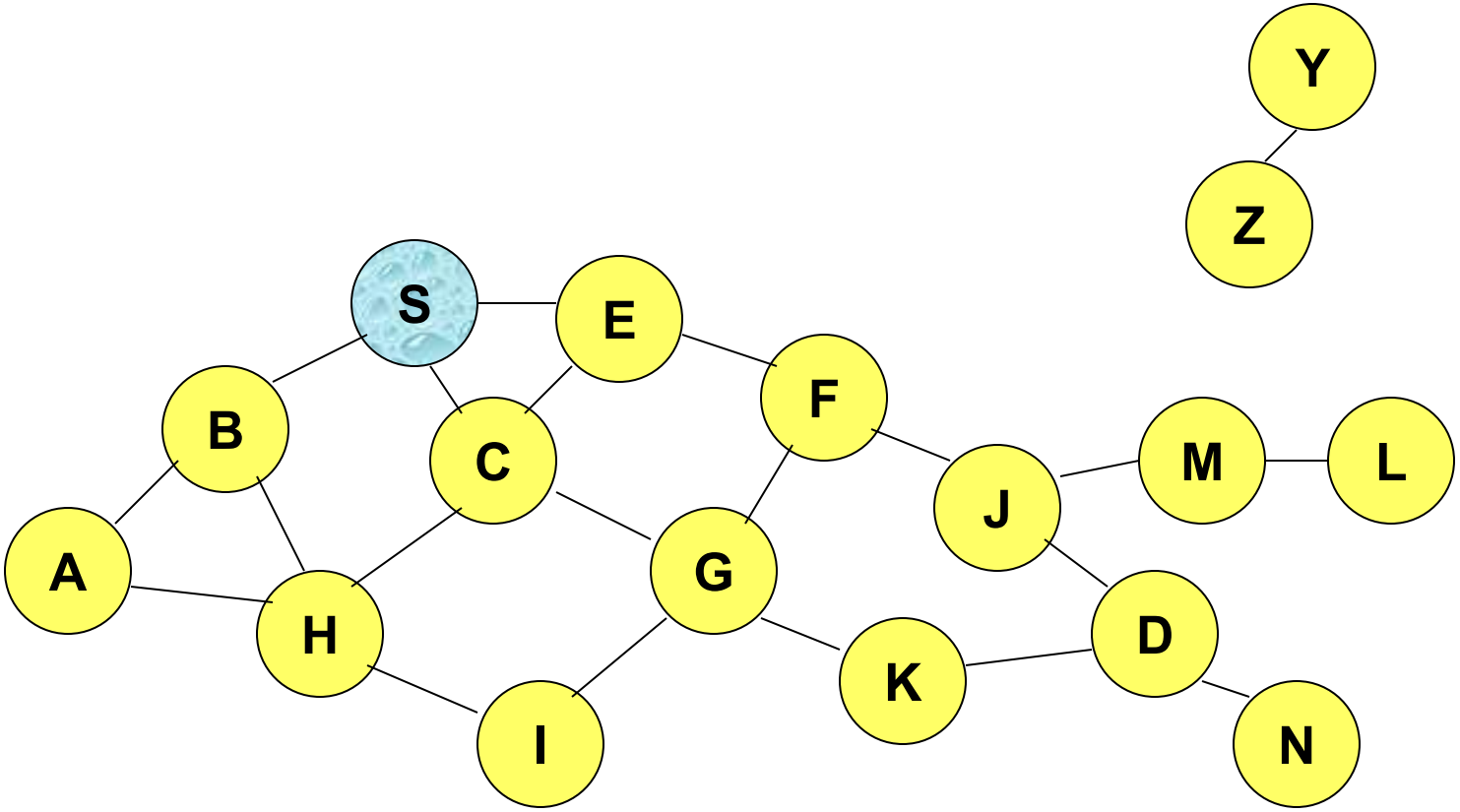
- Which approach achieves a better trade-off depends on the traffic and mobility patterns

Reactive Routing Protocols

Dynamic Source Routing (DSR) [Johnson96]

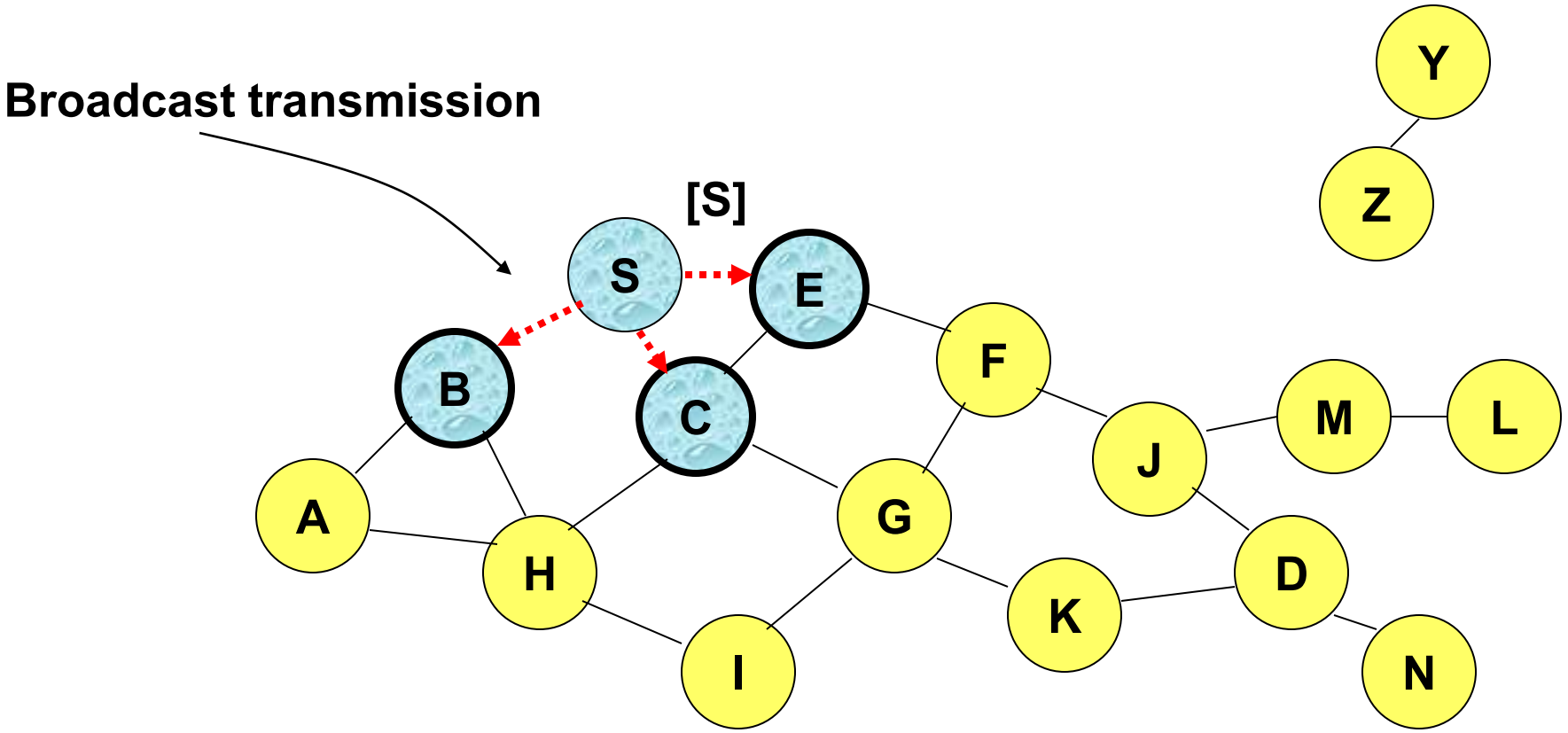
- When node S wants to send a packet to node D, but does not know a route to D, node S initiates a route discovery
- Source node S floods Route Request (RREQ)
- Each node *appends own identifier* when forwarding RREQ

Route Discovery in DSR



Represents a node that has received RREQ for D from S

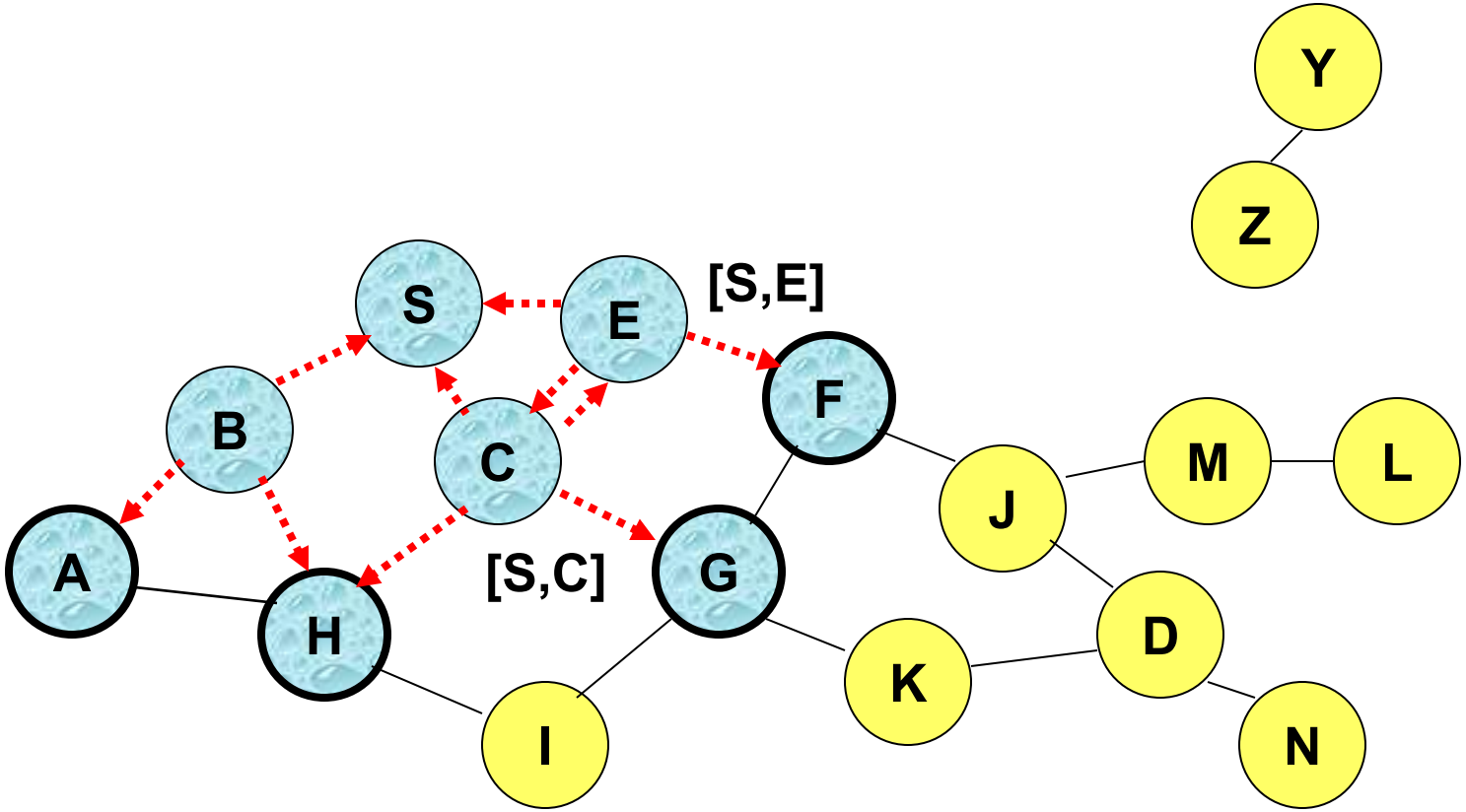
Route Discovery in DSR



Represents transmission of RREQ

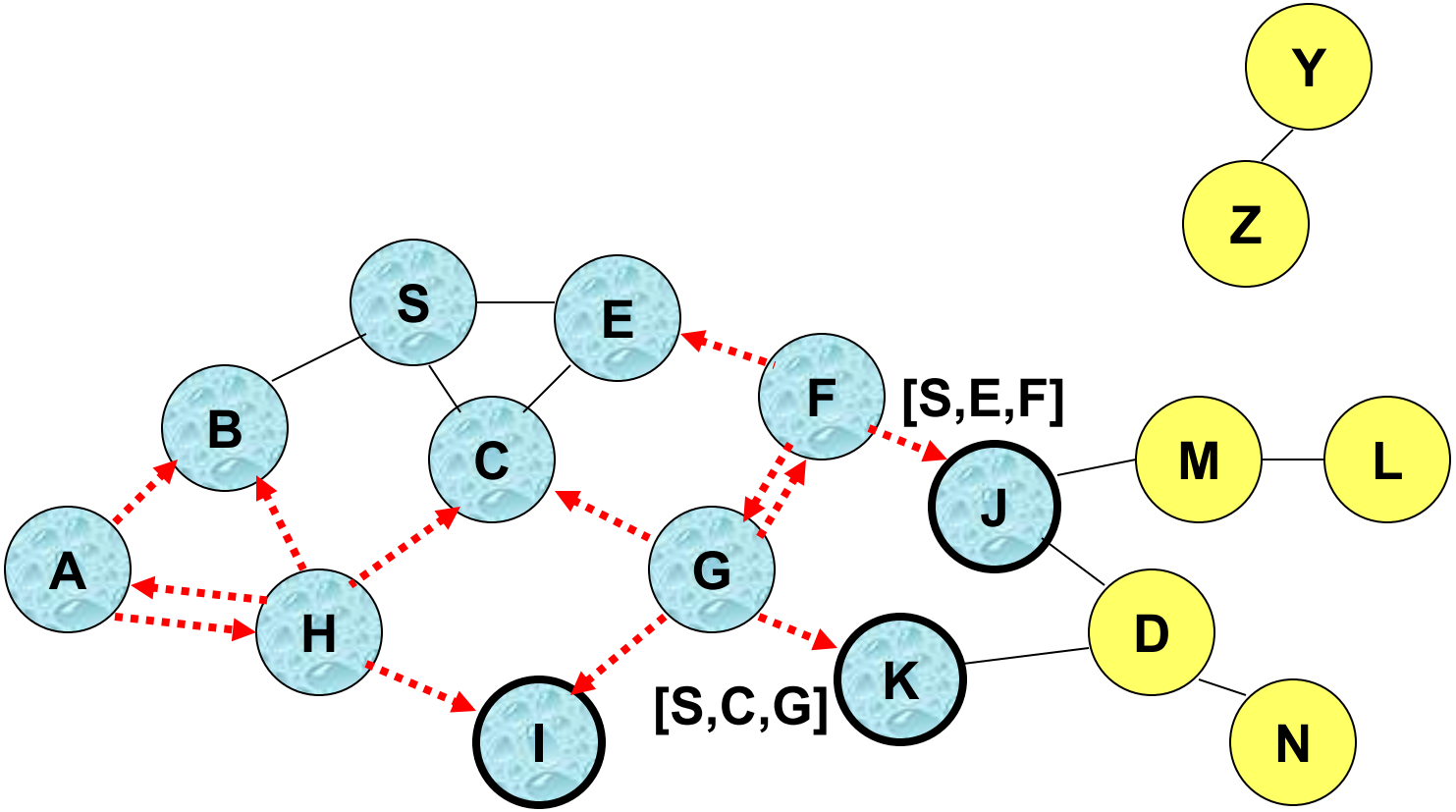
[X,Y] Represents list of identifiers appended to RREQ

Route Discovery in DSR



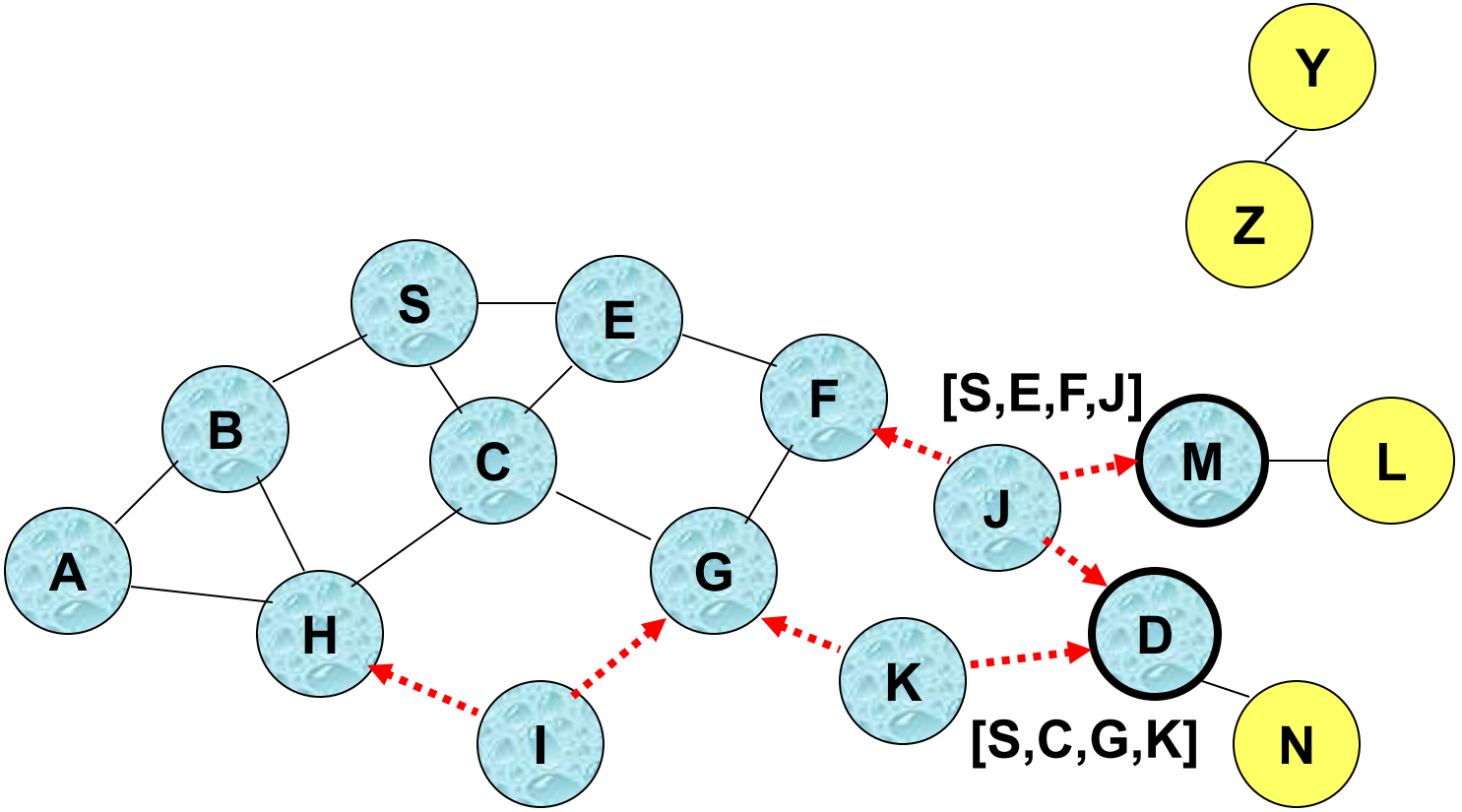
- **Node H receives packet RREQ from two neighbors:**
  potential for collision

Route Discovery in DSR



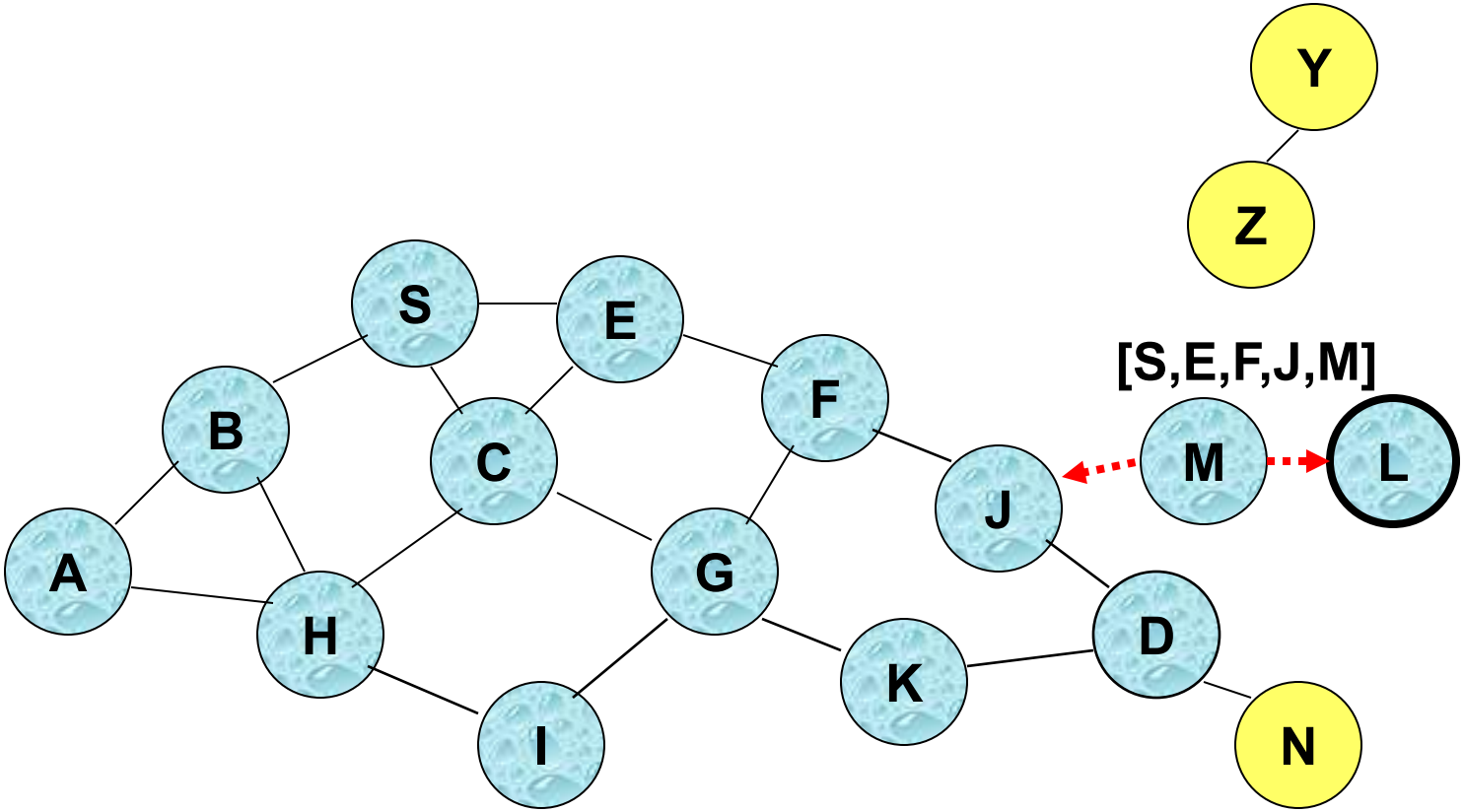
- Node C receives RREQ from G and H, but does not forward it again, because node C has already forwarded RREQ once

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

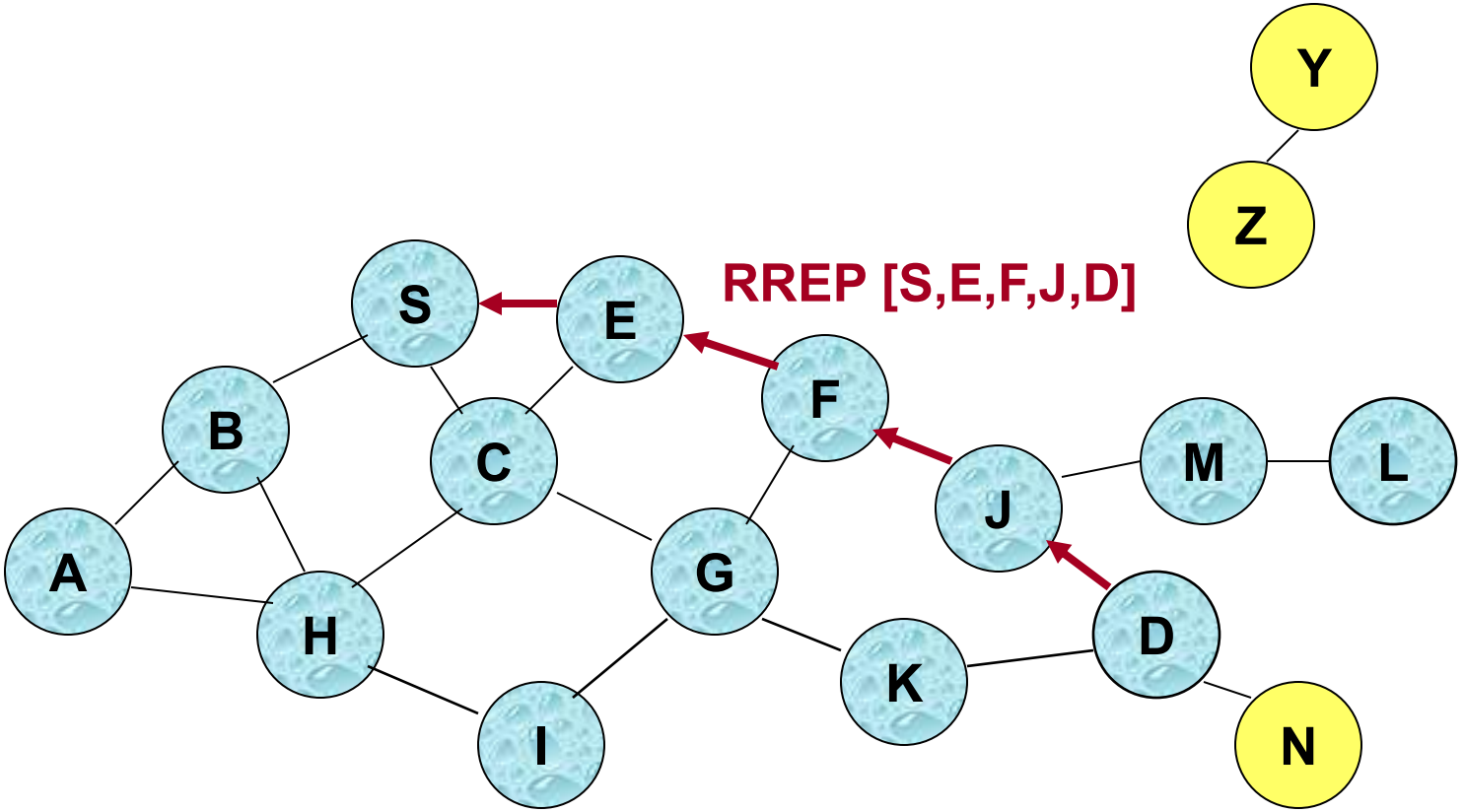


- Node D **does not forward** RREQ, because node D is the **intended target** of the route discovery

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

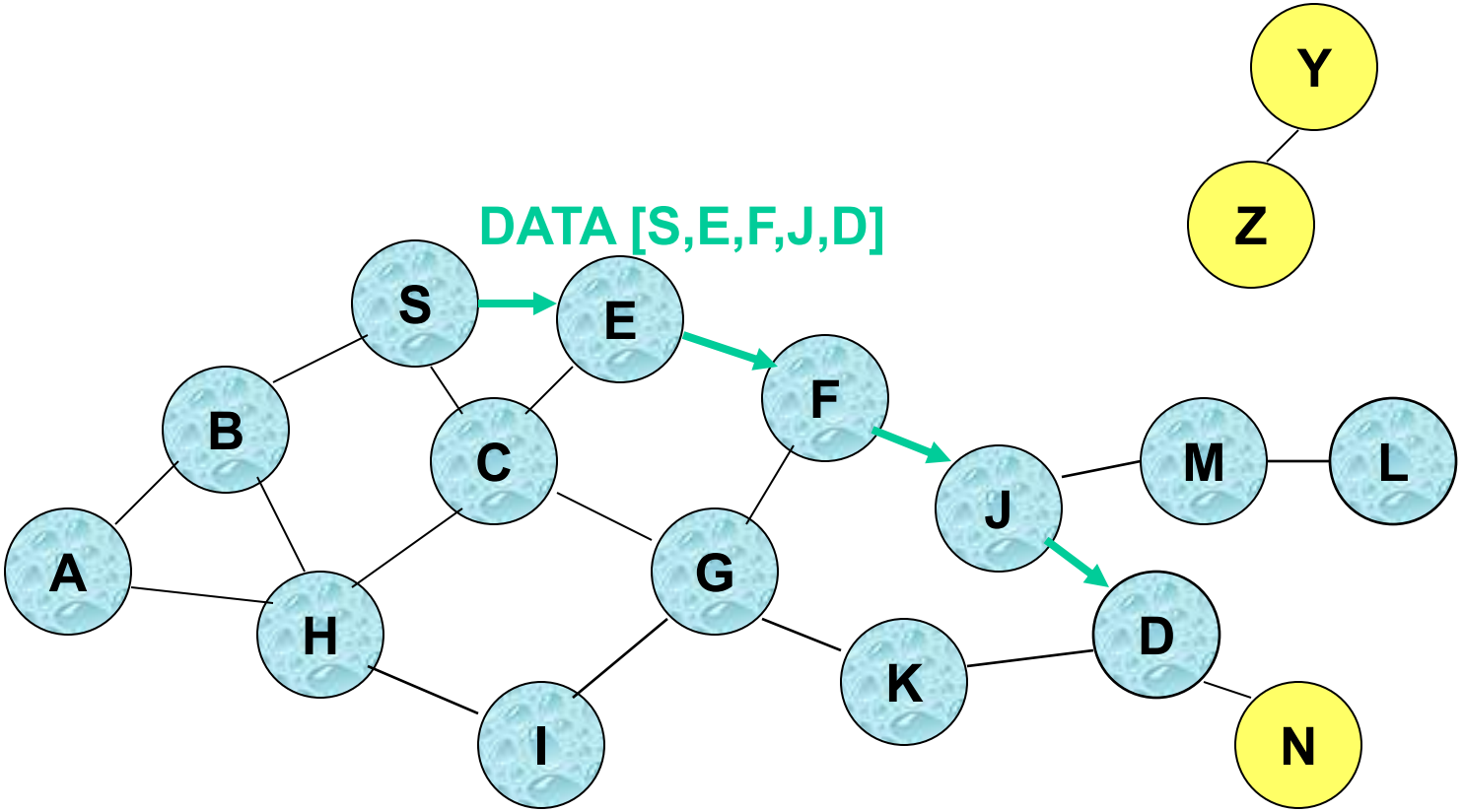


Represents RREP control message

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
- Problem: Stale caches may increase overheads

Dynamic Source Routing: Advantages

- Routes maintained only between nodes who need to communicate
  - reduces overhead of route maintenance
- Route caching can further reduce route discovery overhead
- A single route discovery may yield many routes to the destination, due to intermediate nodes replying from local caches

Dynamic Source Routing: Disadvantages

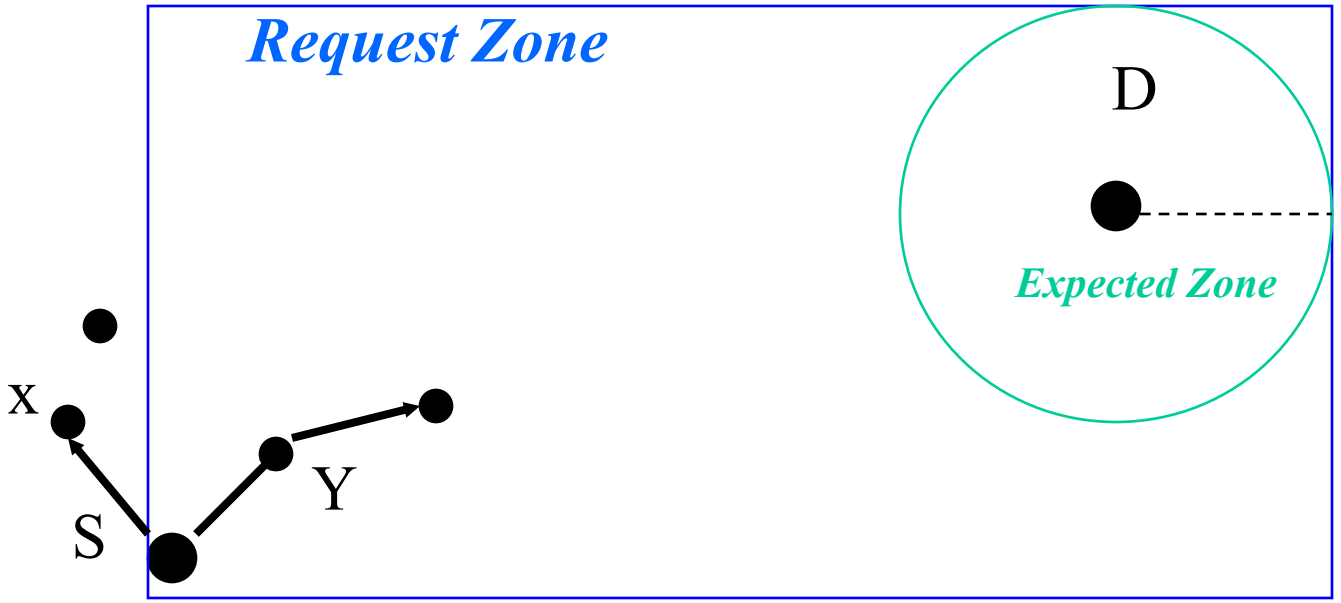
- Packet header size grows with route length due to source routing
- Flood of route requests may potentially reach all nodes in the network
- Potential collisions between route requests propagated by neighboring nodes
  - insertion of random delays before forwarding RREQ
- Increased contention if too many route replies come back due to nodes replying using their local cache
  - Route Reply *Storm* problem
- Stale caches will lead to increased overhead

Location-Aided Routing (LAR) [Ko98Mobicom]

- Exploits location information to limit scope of route request flood
  - Location information may be obtained using GPS
- *Expected Zone* is determined as a region that is expected to hold the current location of the destination
  - Expected region determined based on potentially old location information, and knowledge of the destination's speed
- Route requests limited to a *Request Zone* that contains the Expected Zone and location of the sender node

Request Zone

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



Location Aided Routing (LAR)

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

Ad Hoc On-Demand Distance Vector Routing (AODV) [Perkins99Wmcsa]

- DSR includes source routes in packet headers
- Resulting large headers can sometimes degrade performance
  - particularly when data contents of a packet are small

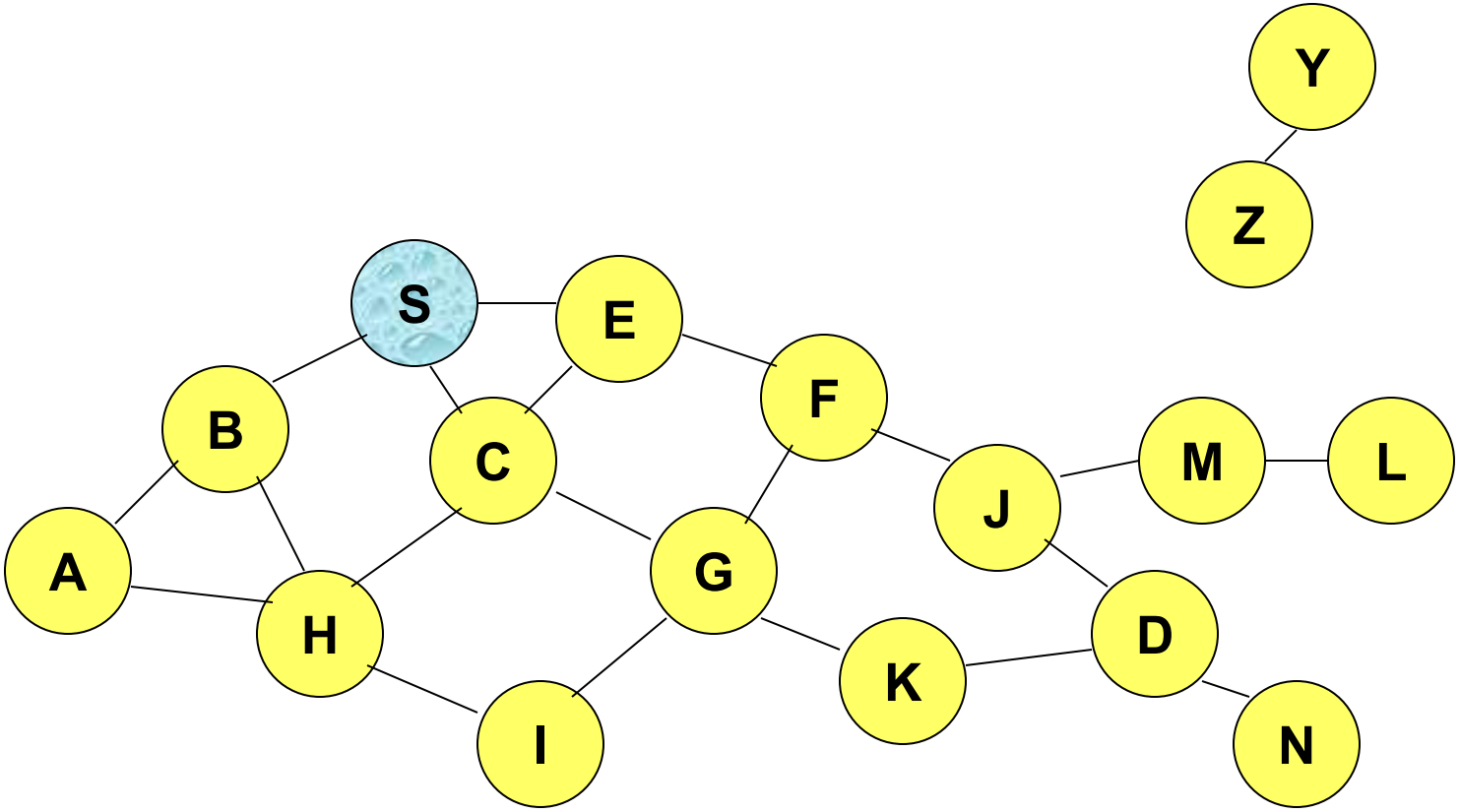
- AODV attempts to improve on DSR by maintaining routing tables at the nodes, so that data packets do not have to contain routes

- AODV retains the desirable feature of DSR that routes are maintained only between nodes which need to communicate

AODV

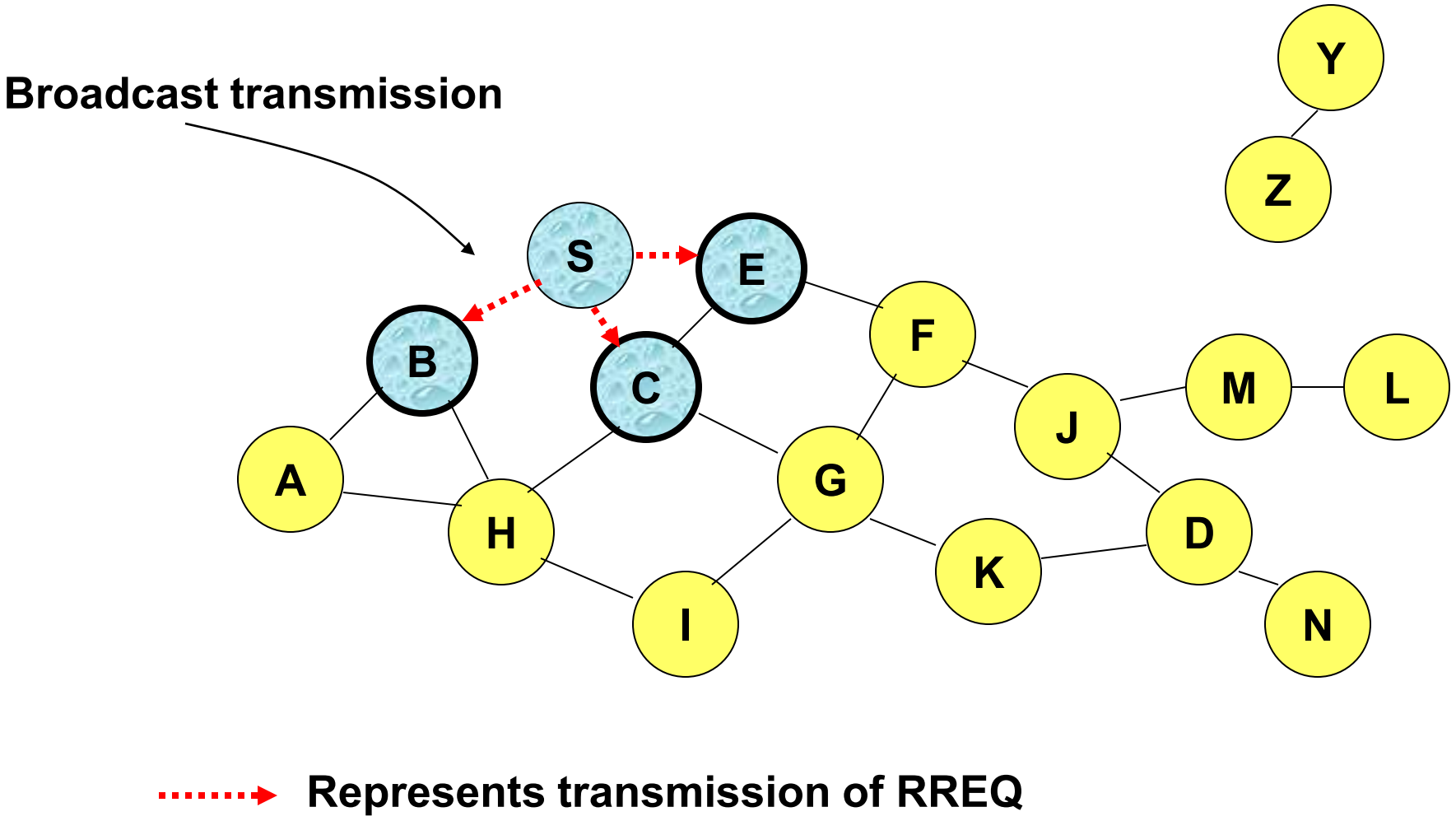
- Route Requests (RREQ) are forwarded in a manner similar to DSR
- When a node re-broadcasts a Route Request, it sets up a reverse path pointing towards the source
  - AODV assumes symmetric (bi-directional) links
- When the intended destination receives a Route Request, it replies by sending a Route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

Route Requests in AODV

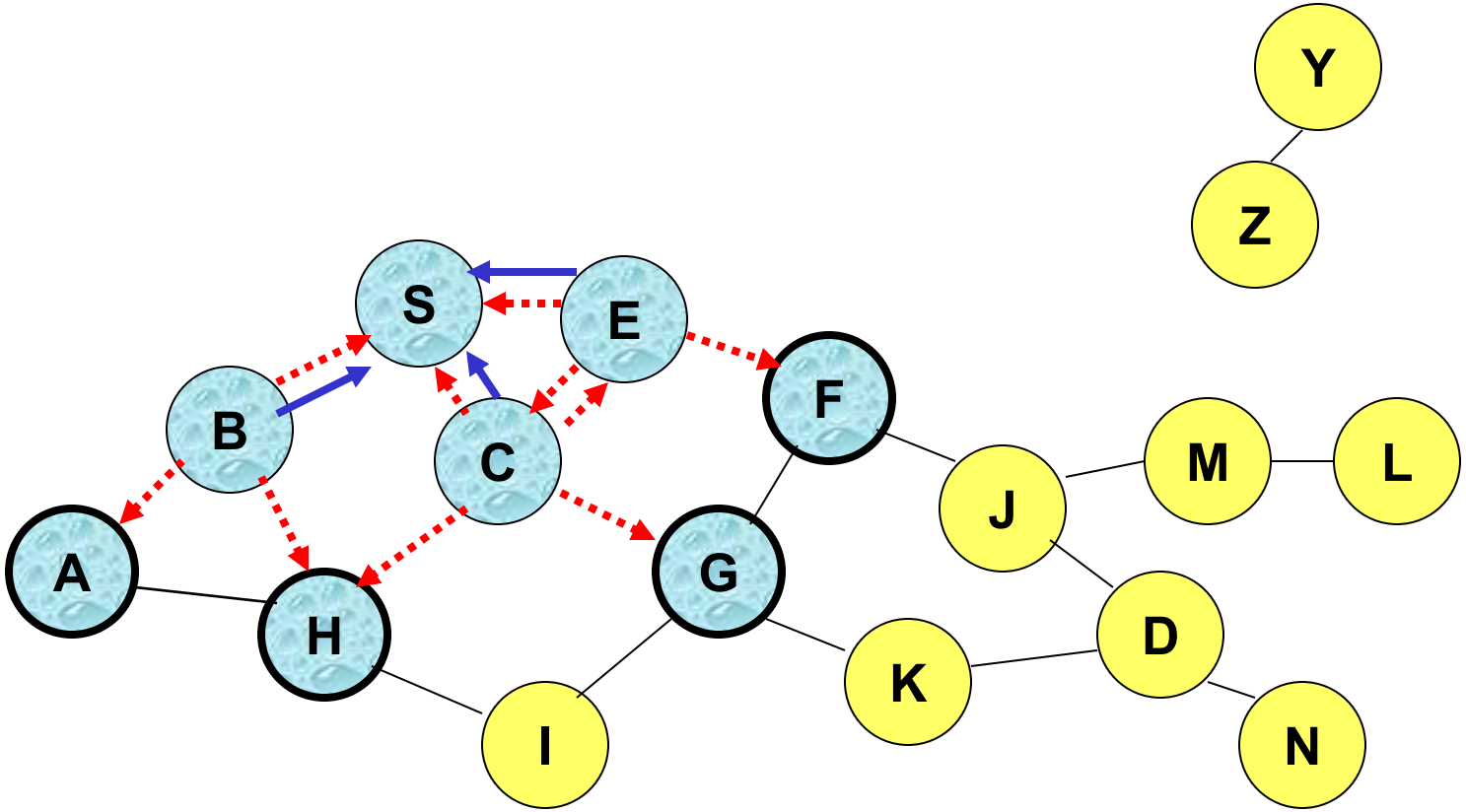


Represents a node that has received RREQ for D from S

Route Requests in AODV

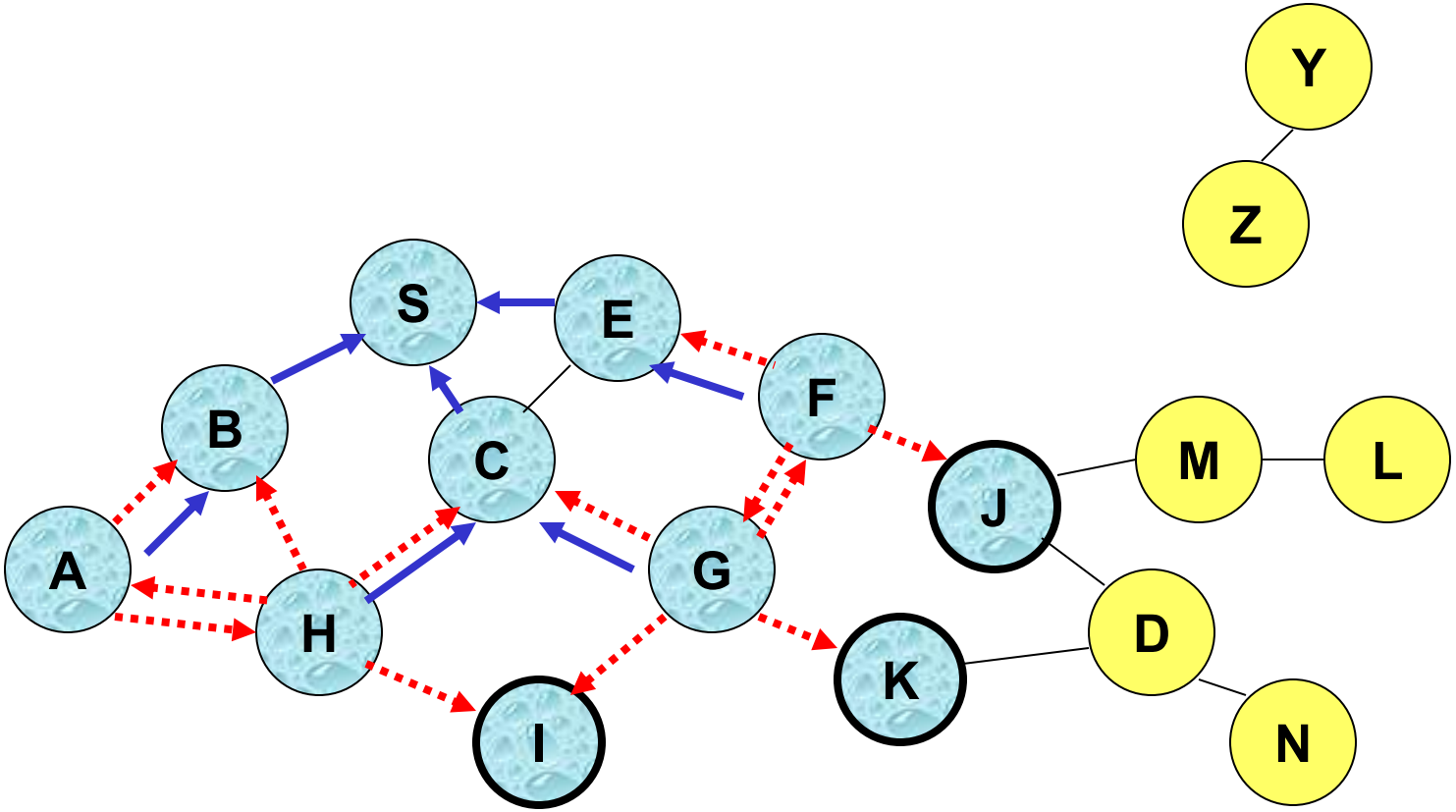


Route Requests in AODV



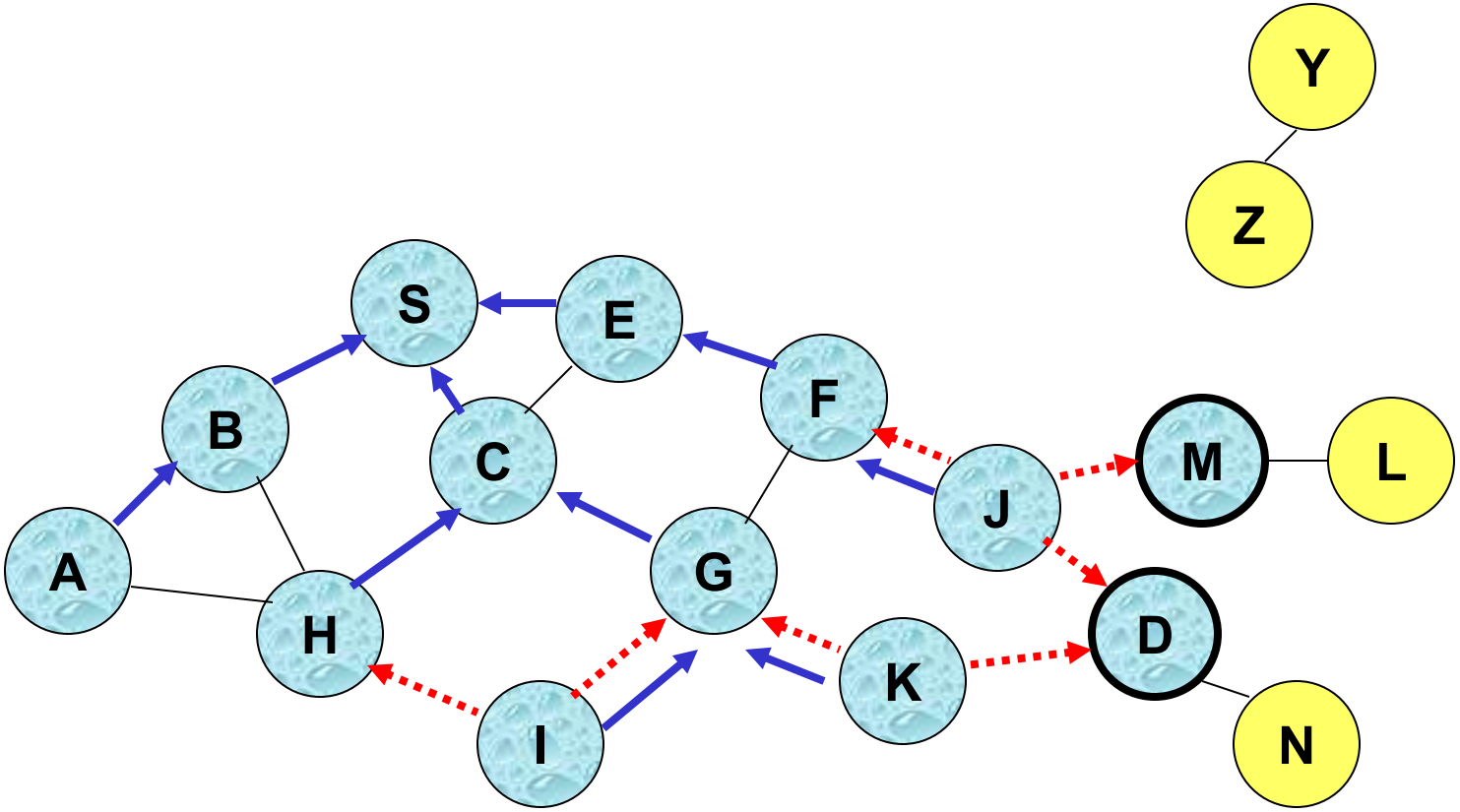
Represents links on Reverse Path

Reverse Path Setup in AODV

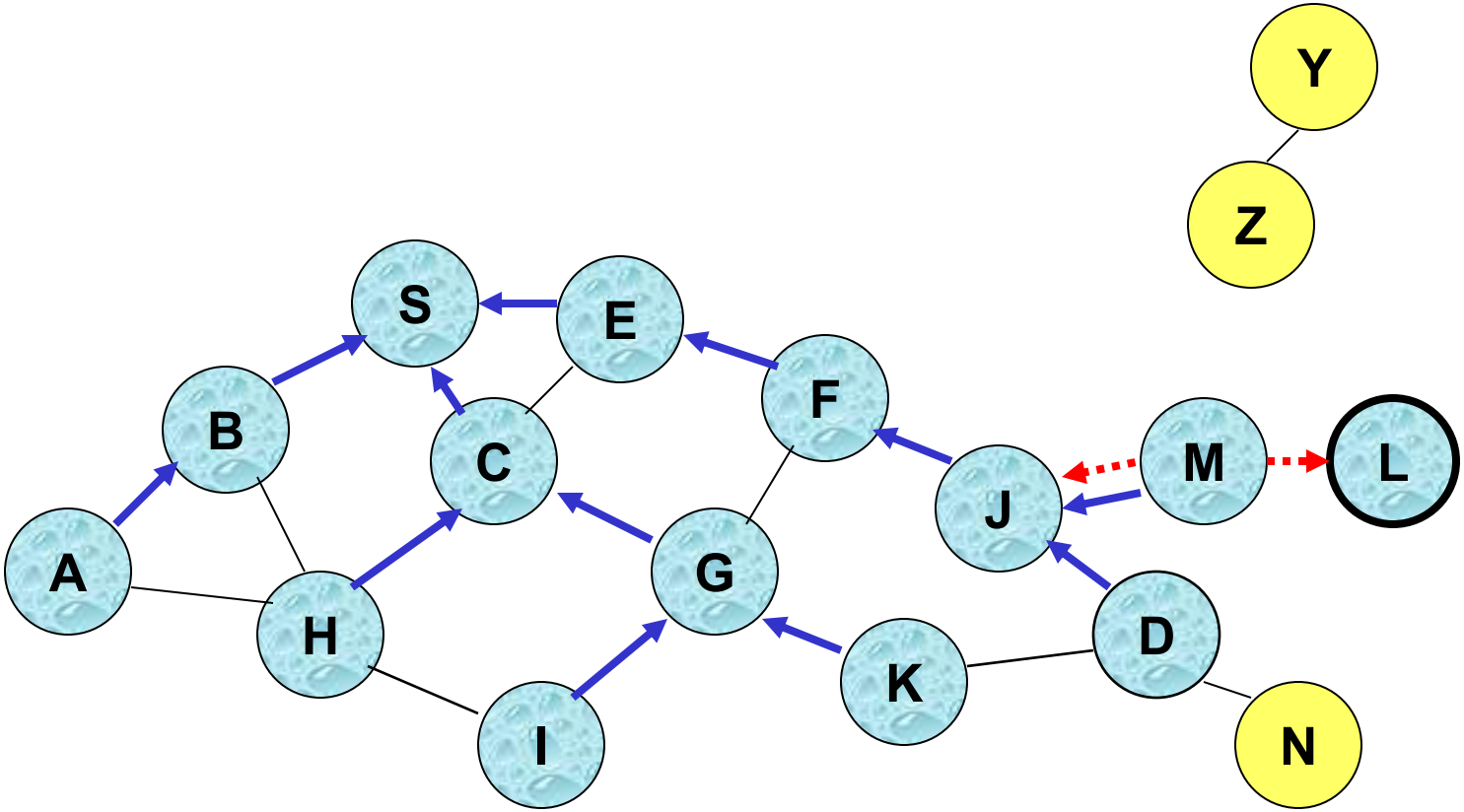


- Node C receives RREQ from G and H, but does not forward it again, because node C has already forwarded RREQ once

Reverse Path Setup in AODV

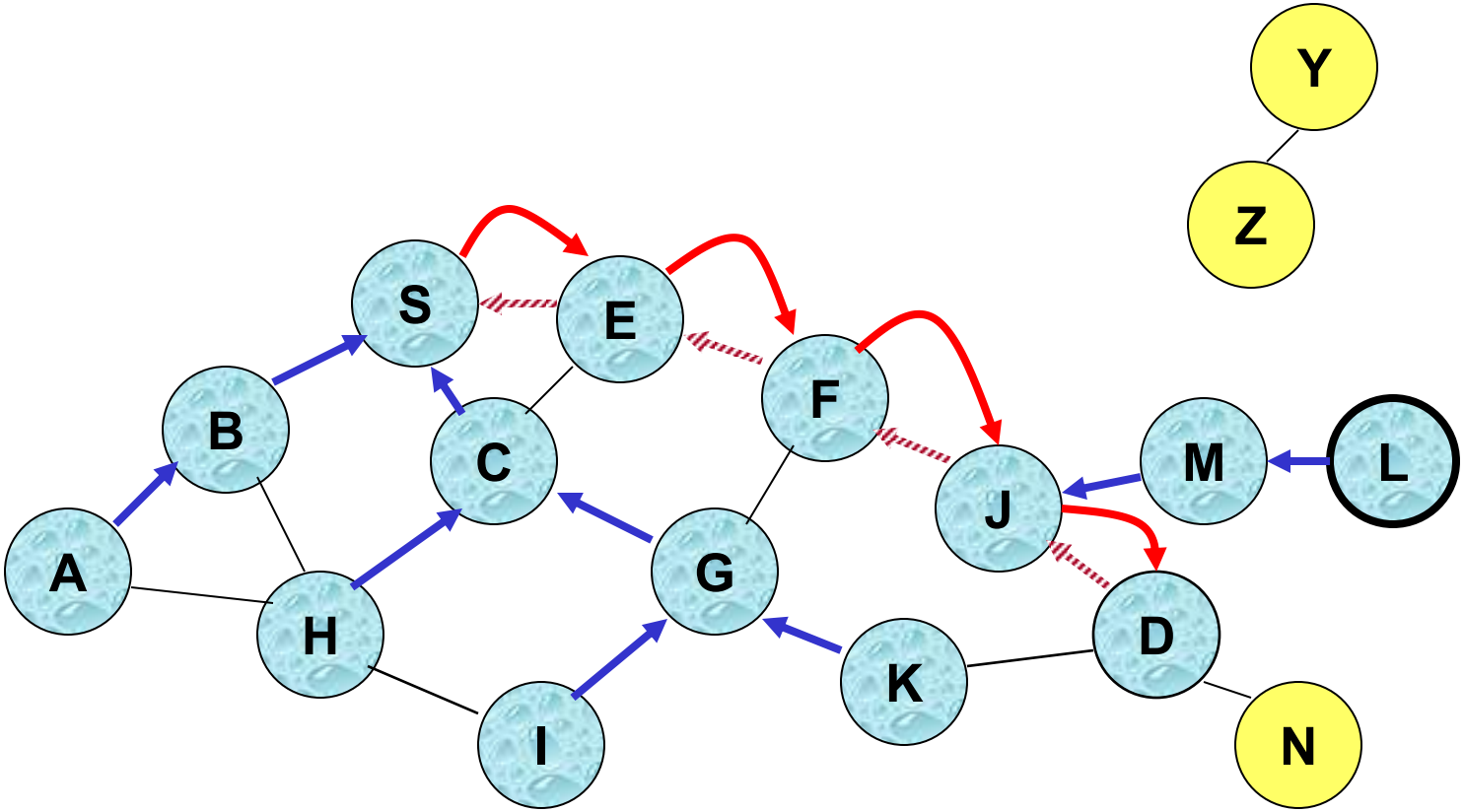


Reverse Path Setup in AODV



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

Forward Path Setup in AODV



Forward links are setup when RREP travels along the reverse path

Represents a link on the forward path

Route Request and Route Reply

- Route Request (RREQ) includes the last known sequence number for the destination

- An intermediate node may also send a Route Reply (RREP) provided that it knows a more recent path than the one previously known to sender
- Intermediate nodes that forward the RREP, also record the next hop to destination

- A routing table entry maintaining a reverse path is purged after a timeout interval
- A routing table entry maintaining a forward path is purged if *not used* for a *active_route_timeout* interval

Link Failure

- A neighbor of node X is considered active for a routing table entry if the neighbor sent a packet within *active_route_timeout* interval which was forwarded using that entry
- Neighboring nodes periodically exchange hello message
- 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 (RERR) 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*
- 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

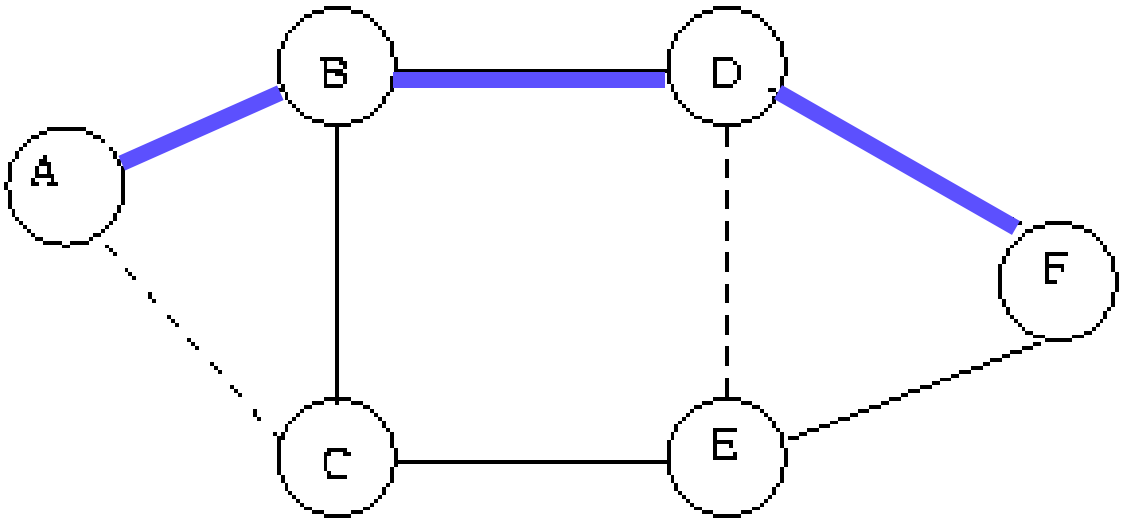
AODV: Summary

- Routes need not be included in packet headers
- Nodes maintain routing tables containing entries only for routes that are in active use
- At most one next-hop per destination maintained at each node
  - DSR may maintain several routes for a single destination
- Sequence numbers are used to avoid old/broken routes
- Sequence numbers prevent formation of routing loops
- Unused routes expire even if topology does not change

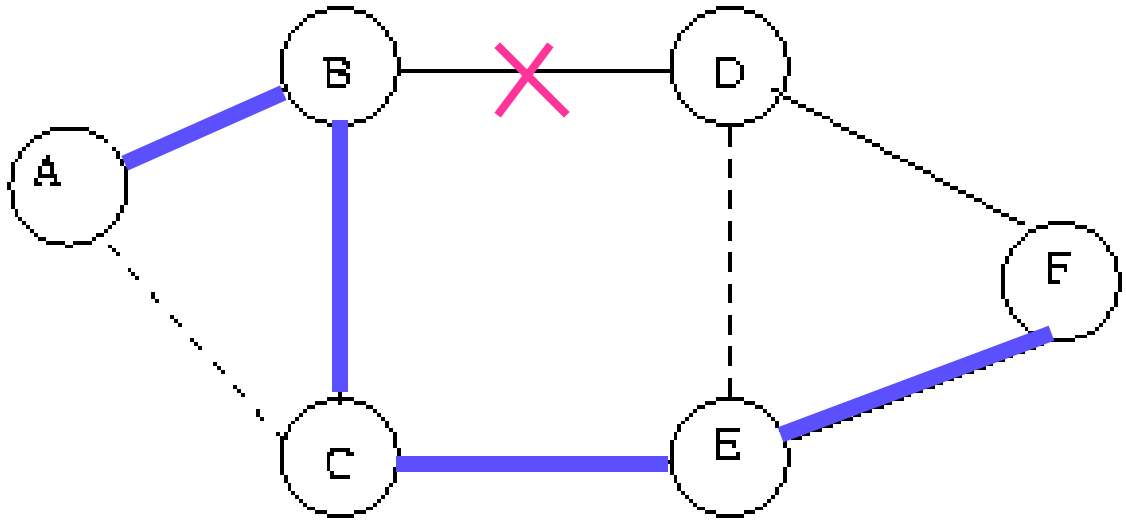
Other Protocols

- Many variations of using control packet flooding for route discovery
- Power-Aware Routing [Singh98Mobicom]
  - Assign a weight to each link: function of energy consumed when transmitting a packet on that link, as well as the residual energy level
  - Modify DSR to incorporate weights and prefer a route with the smallest aggregate weight
- Associativity-Based Routing (ABR) [Toh97]
  - Only links that have been stable for some minimum duration are utilized
  - Nodes increment the associativity ticks of neighbors by using periodic beacons
- Signal Stability Based Adaptive Routing (SSA) [Dube97]
  - A node X re-broadcasts a Route Request received from Y only if the (X,Y) link has a *strong signal stability*
  - Signal stability is evaluated as a moving average of the signal strength of packets received on the link in recent past

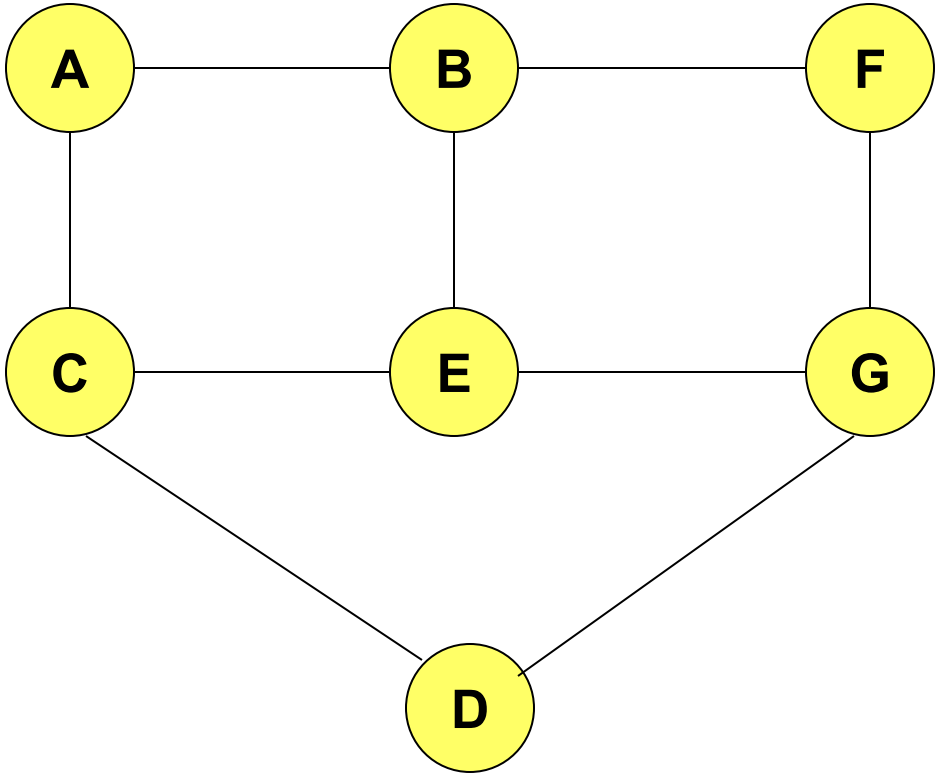
Signal Stability Routing (SSA)



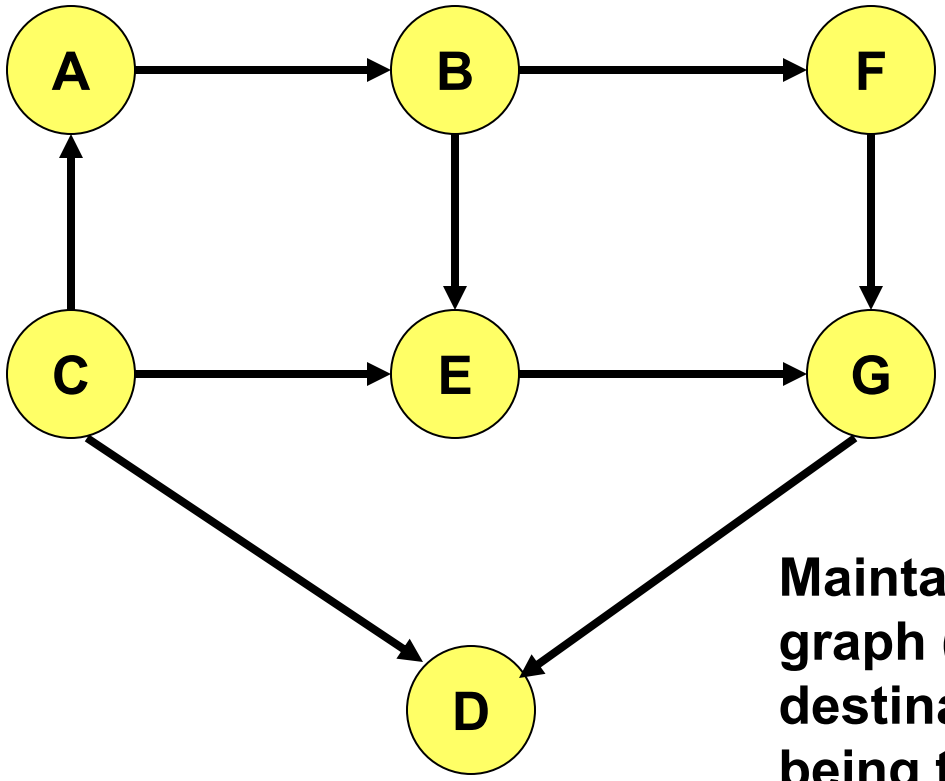
Signal Stability Routing (SSA)



Link Reversal Algorithm [Gafni81]



Link Reversal Algorithm



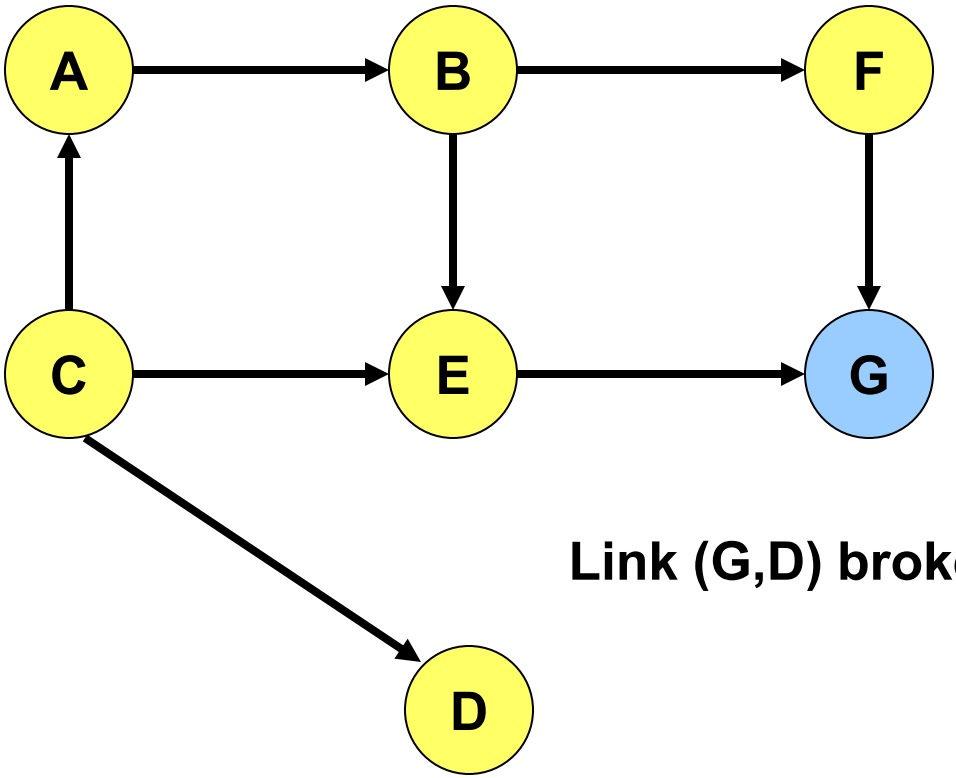
Links are bi-directional

But algorithm imposes logical directions on them

Maintain a directed acyclic graph (DAG) for each destination, with the destination being the *only sink*

This DAG is for *destination node D*

Link Reversal Algorithm

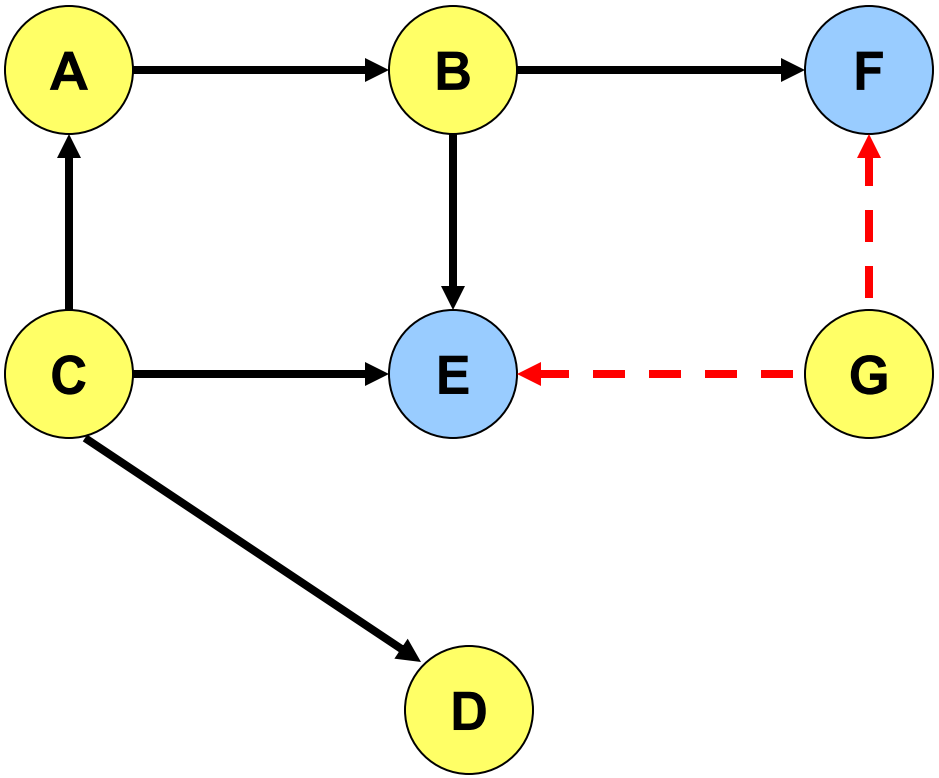


Link (G,D) broke

Any node, other than the destination, that has no outgoing links reverses all its incoming links.

Node G has no outgoing links

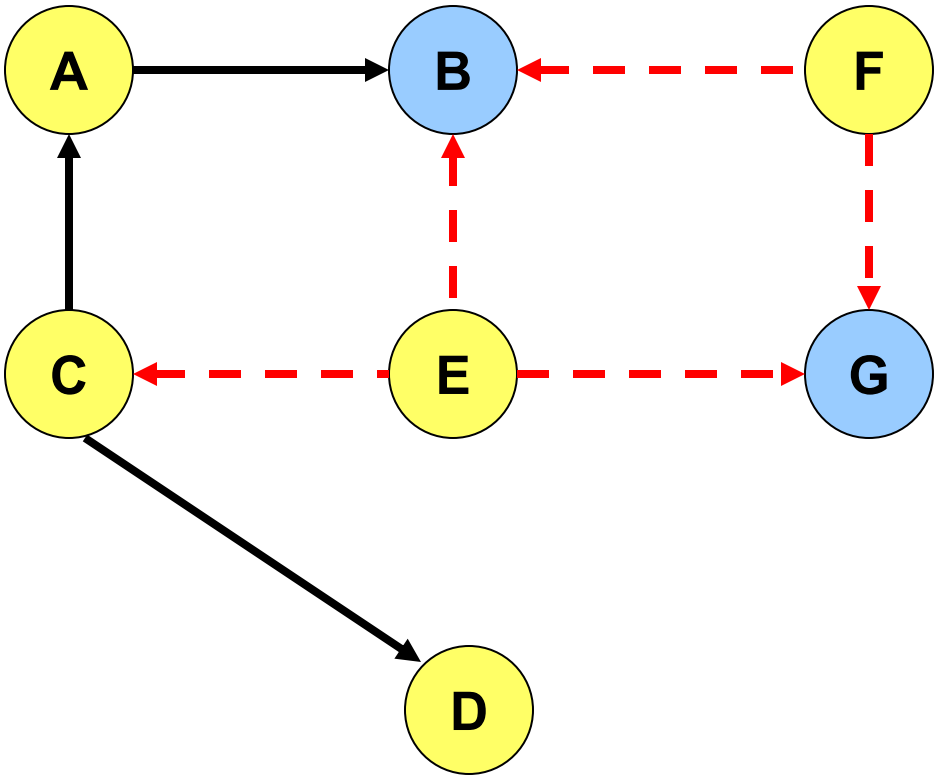
Link Reversal Algorithm



Represents a link that was reversed recently

Now nodes E and F have no outgoing links

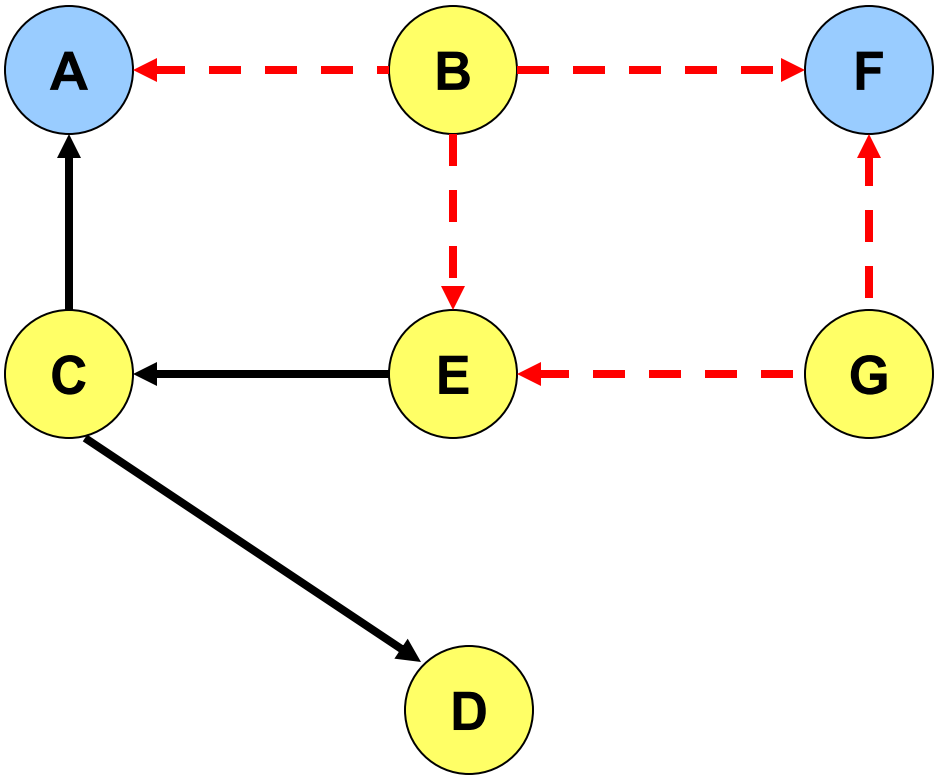
Link Reversal Algorithm



Represents a link that was reversed recently

Now nodes B and G have no outgoing links

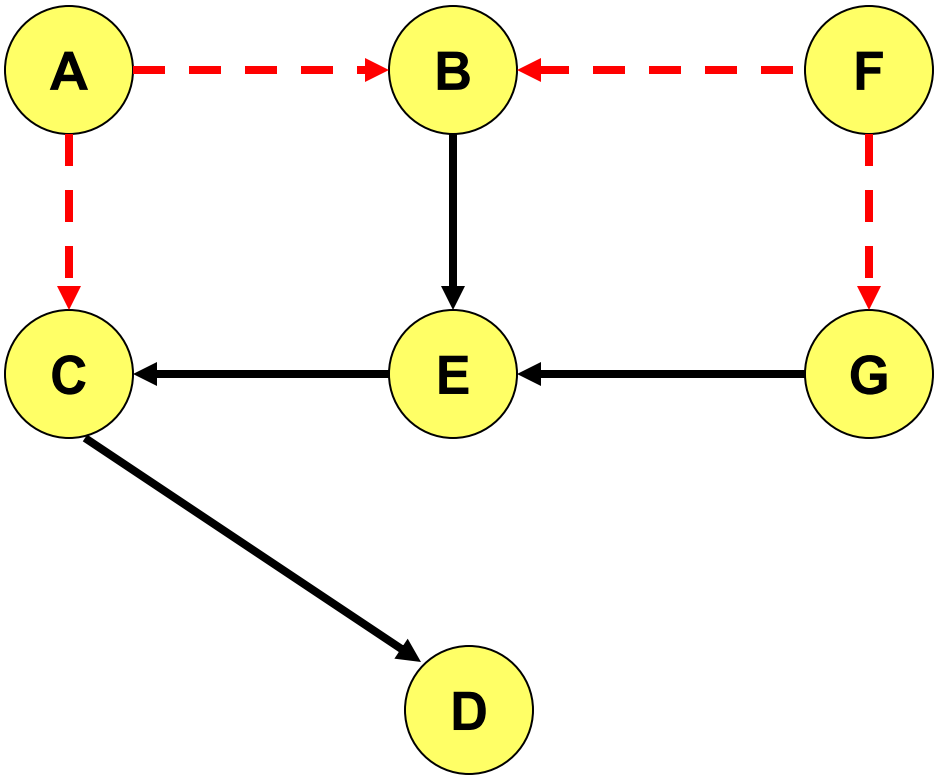
Link Reversal Algorithm



Represents a link that was reversed recently

Now nodes A and F have no outgoing links

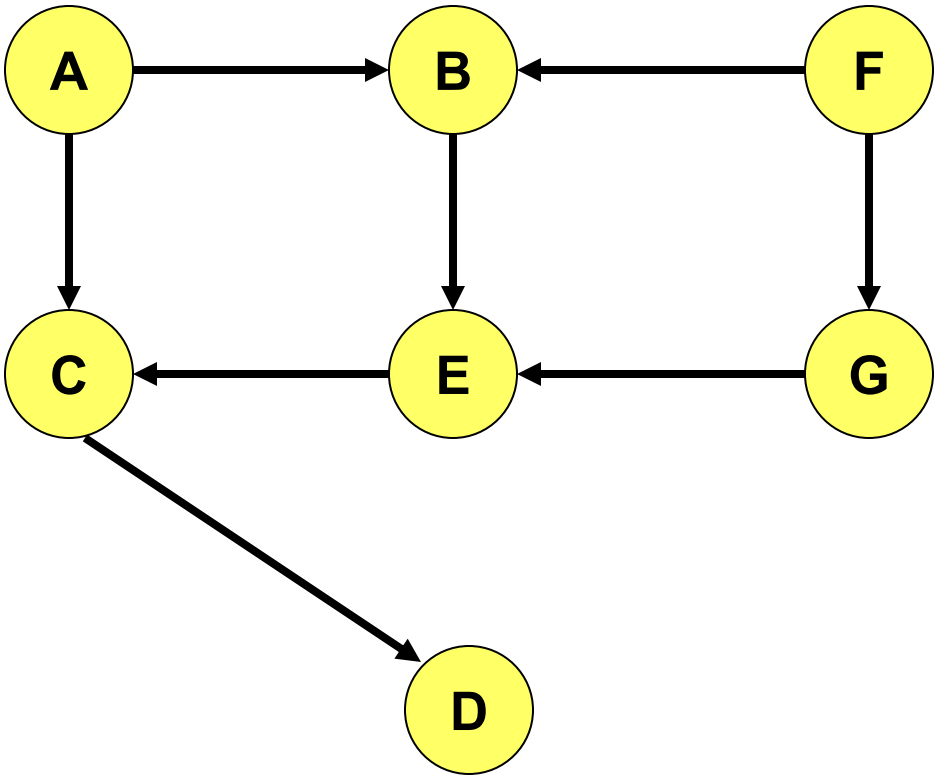
Link Reversal Algorithm



Represents a link that was reversed recently

Now all nodes (other than destination D) have an outgoing link

Link Reversal Algorithm



DAG has been restored with only the destination as a sink

Link Reversal Algorithm

- Attempts to keep link reversals local to where the failure occurred
  - But this is not guaranteed
- When the first packet is sent to a destination, the destination oriented DAG is constructed
- The initial construction does result in flooding of control packets

Link Reversal Algorithm

- The previous algorithm is called a full reversal method since when a node reverses links, it reverses *all* its incoming links

- Partial reversal method [Gafni81]: A node reverses incoming links from only those neighbors who have not themselves reversed links "previously"
  - If all neighbors have reversed links, then the node reverses all its incoming links
  - "Previously" at node X means *since the last link reversal done by node X*

Link Reversal Methods

- Advantages
  - Link reversal methods attempt to limit updates to routing tables at nodes in the vicinity of a broken link
    - Partial reversal method tends to be better than full reversal method
  - Each node may potentially have multiple routes to a destination

- Disadvantages
  - Need a mechanism to detect link failure
    - hello messages may be used
  - If network is partitioned, link reversals continue indefinitely

Temporally-Ordered Routing Algorithm (TORA) [Park97Infocom]

- Route optimality is considered of secondary importance; longer routes may be used

- At each node, a logically separate copy of TORA is run for each destination, that computes the height of the node with respect to the destination
- Height captures number of hops and next hop
- Route discovery is by using query and update packets

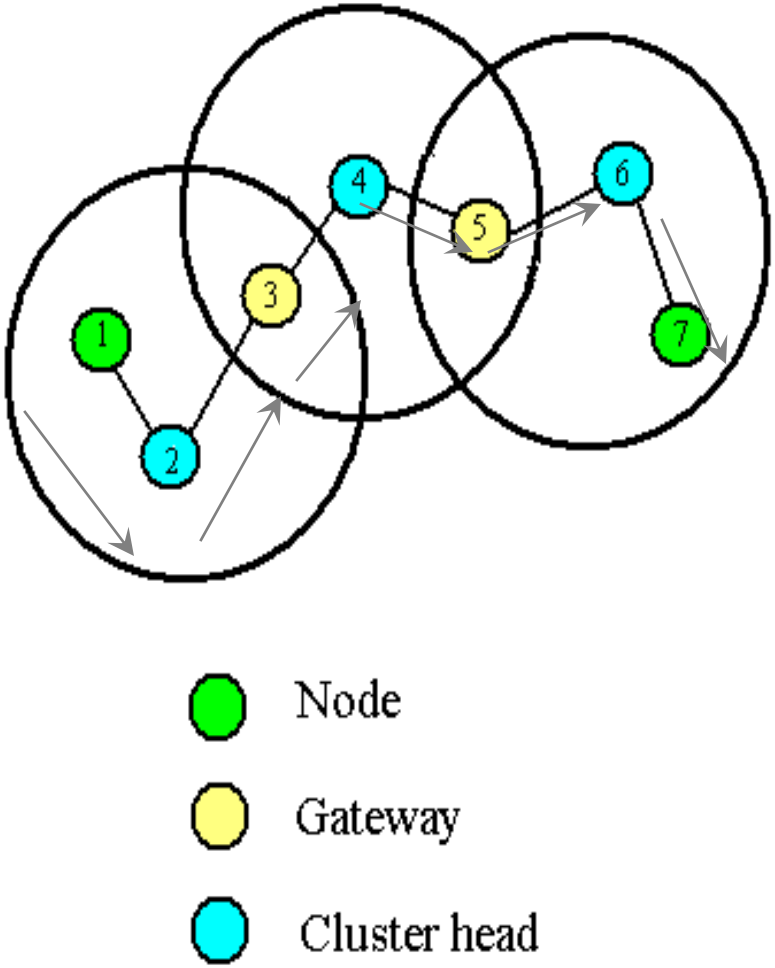
- TORA modifies the partial link reversal method to be able to detect partitions
- When a partition is detected, all nodes in the partition are informed, and link reversals in that partition cease

Asymmetric Algorithms

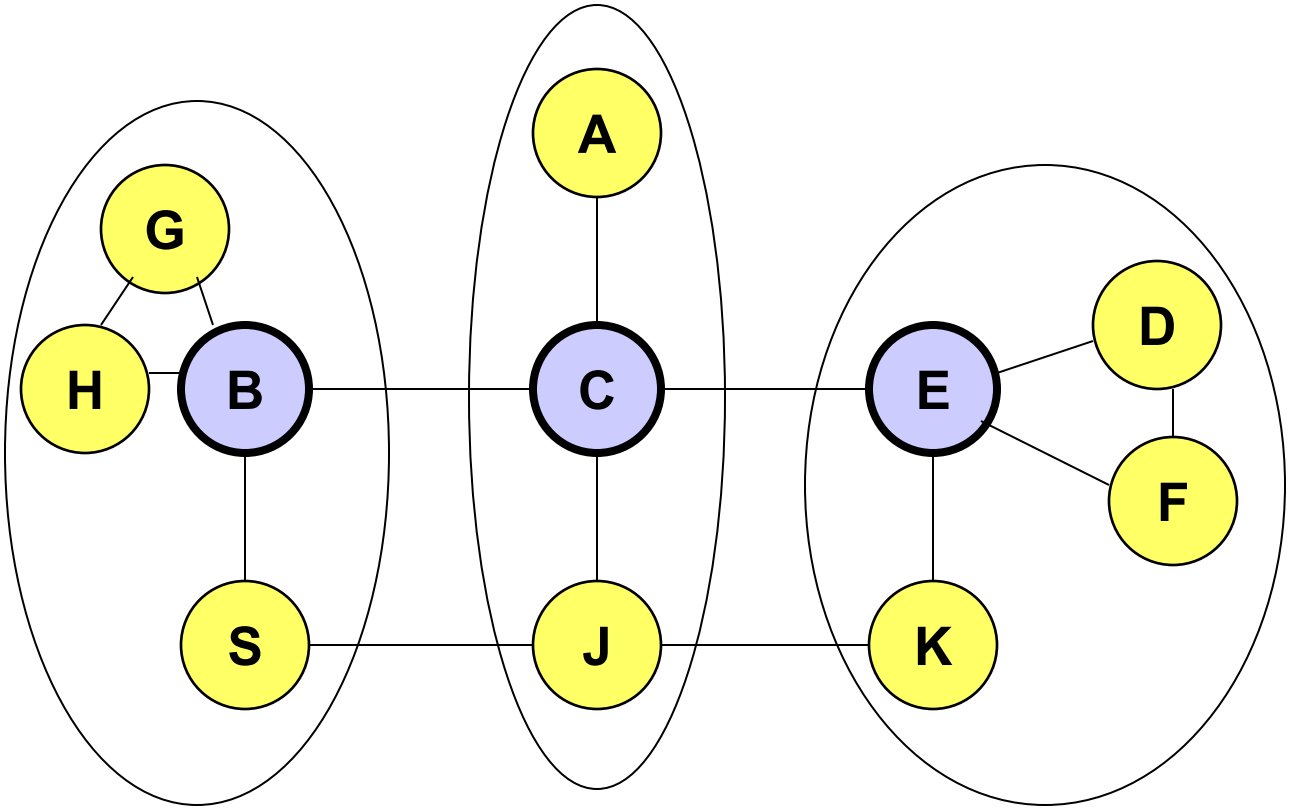
- Clusterhead Gateway Switch Routing (CGSR)
  - All nodes within a cluster communicate with a clusterhead
  - Routing uses a hierarchical clusterhead-to-gateway approach

- Core-Extraction Distributed Ad Hoc Routing (CEDAR) [Sivakumar99]
  - A subset of nodes in the network is identified as the *core*
  - Each node in the network must be adjacent to at least one node in the core
  - Each core node determines paths to nearby core nodes by means of a localized broadcast

CGSR



CEDAR



A core node

Node E is the dominator for nodes D, F and K

Proactive Routing Protocols

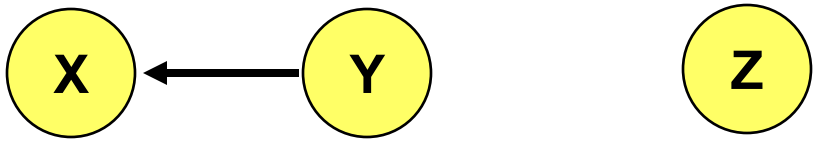
Destination-Sequenced Distance-Vector (DSDV) [Perkins94Sigcomm]

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

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

Destination-Sequenced Distance-Vector (DSDV)

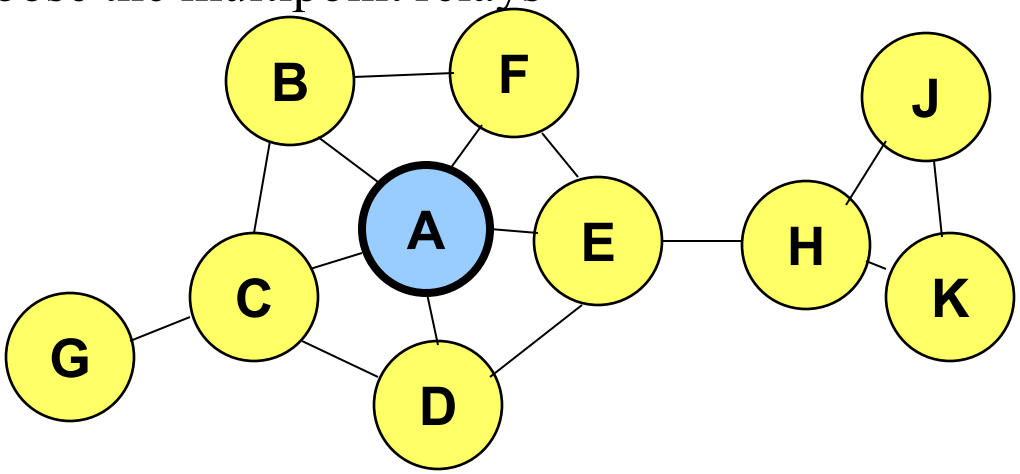
- When X receives information from Y about a route to Z
  - Let destination sequence number for Z at X be S(X), S(Y) is sent from Y



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

Optimized Link State Routing (OLSR)
[Jacquet00ietf]

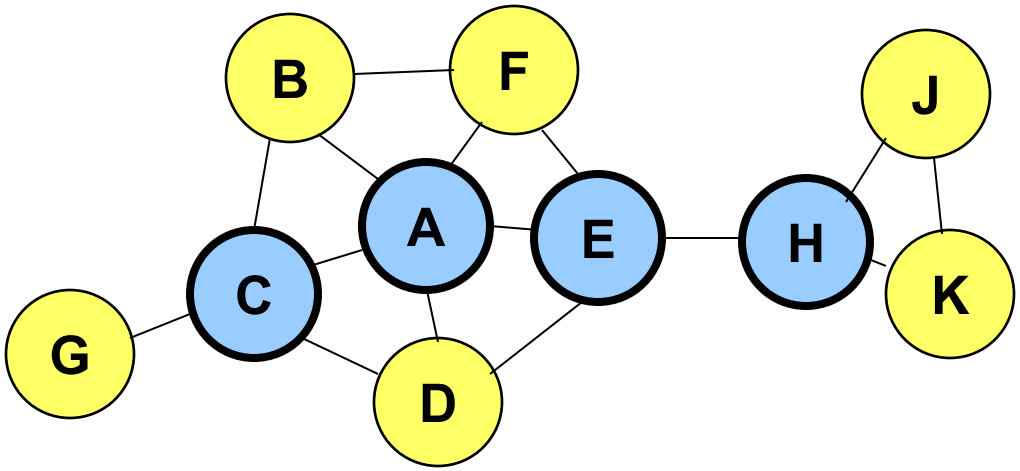
- Nodes C and E are multipoint relays of node A
  - Multipoint relays of A are its neighbors such that each two-hop neighbor of A is a one-hop neighbor of one multipoint relay of A
  - Nodes exchange neighbor lists to know their 2-hop neighbors and choose the multipoint relays



Node that has broadcast state information from A

Optimized Link State Routing (OLSR)

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



Node that has broadcast state information from A

Hybrid Routing Protocols

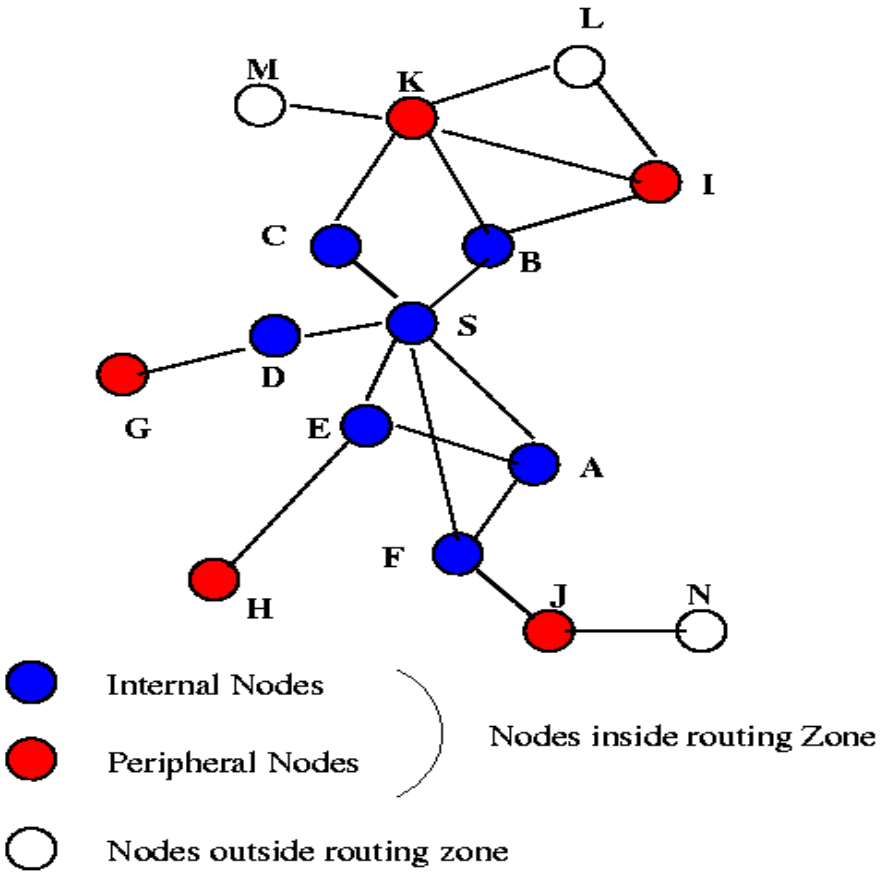
Zone Routing Protocol (ZRP) [Haas98]

- ZRP combines proactive and reactive approaches

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

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

Zone Routing Protocol (ZRP)



Radius of routing zone = 2

Routing Summary

- Protocols
  - Typically divided into proactive, reactive and hybrid
  - Plenty of routing protocols. Discussion here is far from exhaustive
- Performance Studies
  - Typically studied by simulations using ns, discrete event simulator
  - Nodes (10-30) remains stationary for pause time seconds (0-900s) and then move to a random destination (1500m X300m space) at a uniform speed (0-20m/s). CBR traffic sources (4-30 packets/sec, 64-1024 bytes/packet)
  - Attempt to estimate latency of route discovery, routing overhead …
- Actual trade-off depends a lot on traffic and mobility patterns
  - Higher traffic diversity (more source-destination pairs) increases overhead in on-demand protocols
  - Higher mobility will always increase overhead in all protocols

Transport in MANET

User Datagram Protocol (UDP)

- Studies comparing different routing protocols for MANET typically measure UDP performance

- Several performance metrics are used
  - routing overhead per data packet
  - packet delivery delay
  - throughput/loss
- Many variables affect performance
  - Traffic characteristics
  - Mobility characteristics
  - Node capabilities
- Difficult to identify a single scheme that will perform well in all environments

- Several relevant studies [Broch98Mobicom, Das9ic3n, Johansson99Mobicom, Das00Infocom, Jacquet00Inria]

Transmission Control Protocol (TCP)

- Reliable ordered delivery
  - Reliability achieved by means of retransmissions if necessary
- End-to-end semantics
  - Receiver sends cumulative acknowledgements for in-sequence packets
  - Receiver sends duplicate acknowledgements for out-of-sequence packets
- Implements congestion avoidance and control using sliding-window
  - Window size is minimum of
    - receiver's advertised window - determined by available buffer space at the receiver
    - congestion window - determined by the sender, based on feedback from the network
  - Congestion window size bounds the amount of data that can be sent per round-trip time

Detection of packet loss in TCP

- Retransmission timeout (RTO)
  - sender sets retransmission timer for only one packet
  - if Ack not received before timer expiry, the packet is assumed lost
  - RTO dynamically calculated, doubles on each timeout

- Duplicate acks
  - sender assumes packet loss if it receives three consecutive duplicate acknowledgements (dupacks)

- On detecting a packet loss, TCP sender assumes that network congestion has occurred and drastically reduces the congestion window

TCP in MANET

Several factors affect TCP performance in MANET:

- Wireless transmission errors
  - may cause fast retransmit, which results in
    - retransmission of lost packet
    - reduction in congestion window
  - reducing congestion window in response to errors is unnecessary
- Multi-hop routes on shared wireless medium
  - Longer connections are at a disadvantage compared to shorter connections, because they have to contend for wireless access at each hop
- Route failures due to mobility

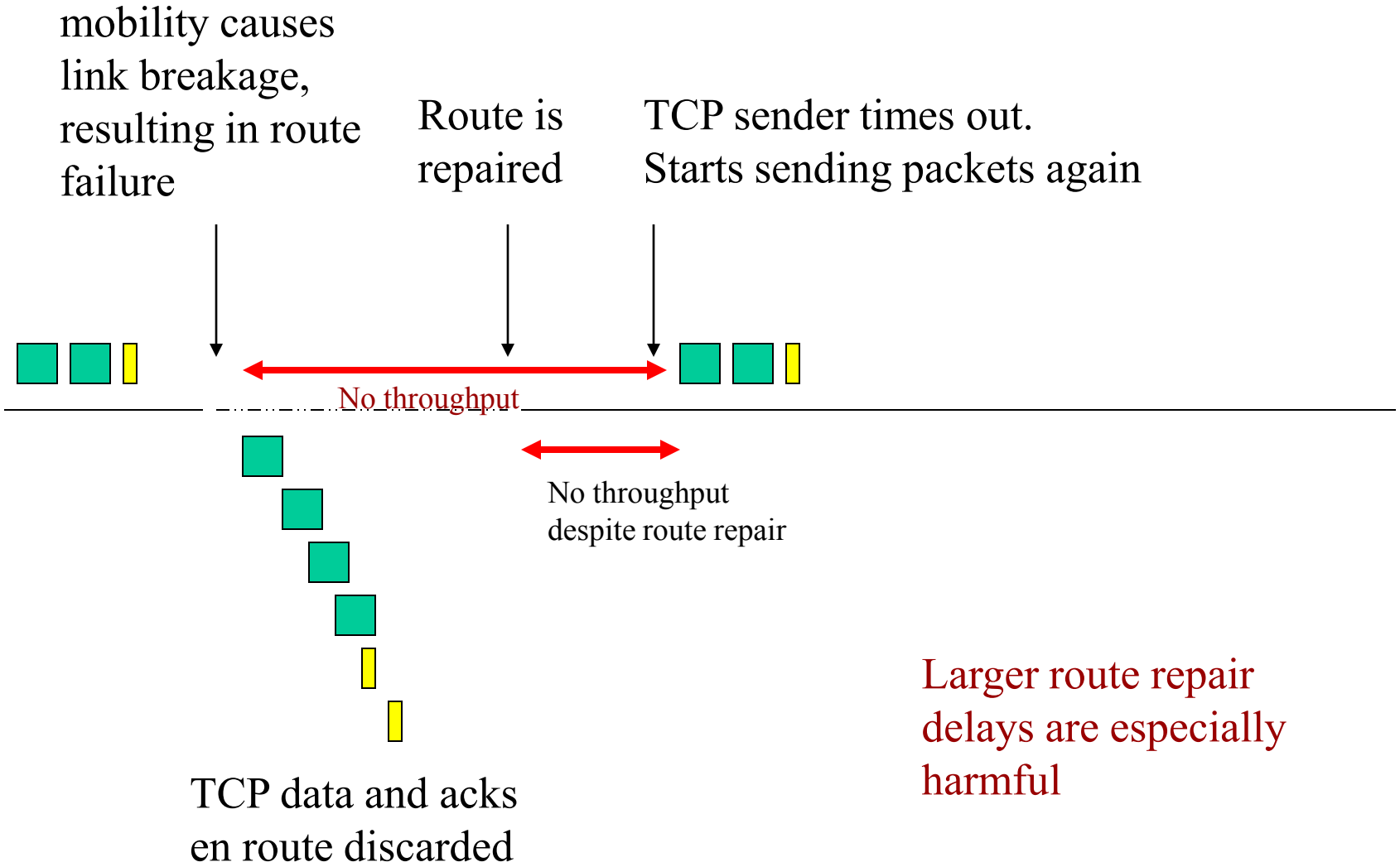
Impact of Multi-hop Wireless Paths

TCP throughput degrades with increase in number of hops

- Packet transmission can occur on at most one hop among three consecutive hops
  - Increasing the number of hops from 1 to 2, 3 results in increased delay, and decreased throughput
- Increasing number of hops beyond 3 allows simultaneous transmissions on more than one link, however, degradation continues due to contention between TCP Data and Acks traveling in opposite directions
- When number of hops is large enough (>6), throughput stabilizes [Holland99]

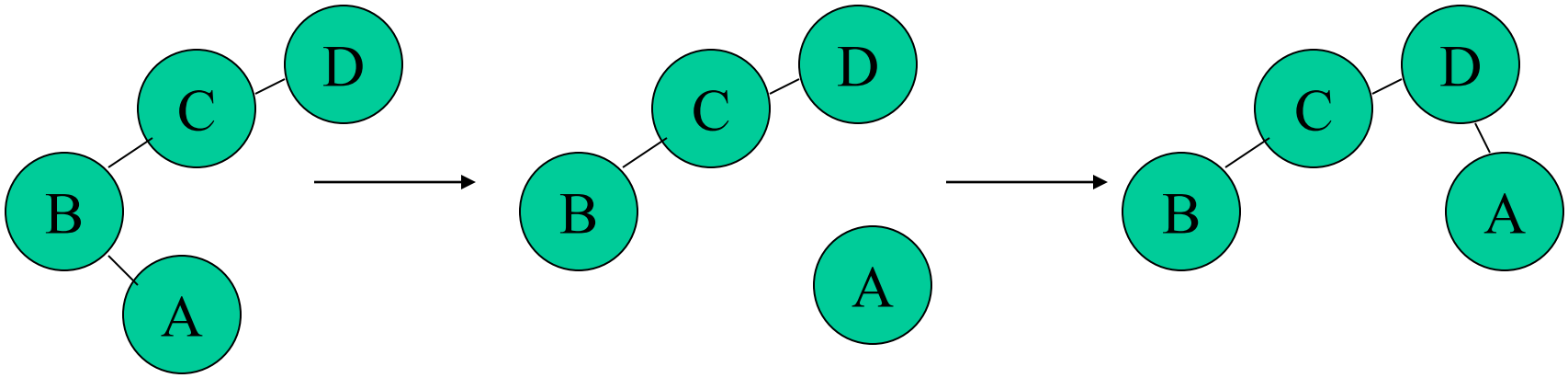
Impact of Node Mobility

TCP throughput degrades with increase in mobility but not always



Larger route repair delays are especially harmful

Improved Throughput with Increased Mobility



- Low speed: (Route from A to D is broken for ~1.5 seconds)
  - When TCP sender times after 1 second, route still broken.
  - TCP times out after another 2 seconds, and only then resumes

- High speed: (Route from A to D is broken for ~0.75 seconds)
  - When TCP sender times out after 1 second, route is repaired

- TCP timeout interval somewhat (not entirely) independent of speed
- Network state at higher speed may sometimes be more favorable than lower speed

Impact of Route Caching

TCP performance typically degrades when caches are used for route repair

- When a route is broken, route discovery returns a cached route from local cache or from a nearby node
- After a time-out, TCP sender transmits a packet on the new route.

  However, typically the cached route has also broken after it was cached

- Another route discovery, and TCP time-out interval
- Process repeats until a good route is found

Caching and TCP performance

- Caching can result in faster route repair
  - Faster does not necessarily mean correct
  - If incorrect repairs occur often enough, caching performs poorly
- If cache accuracy is not high enough, gains in routing overhead may be offset by loss of TCP performance due to multiple time-outs
- Need mechanisms for determining when cached routes are stale

Impact of Acknowledgements

- TCP Acks (and link layer acks) share the wireless bandwidth with TCP data packets
- Data and Acks travel in opposite directions
  - In addition to bandwidth usage, acks require additional receive-send turnarounds, which also incur time penalty
- Reduction of contention between data and acks, and frequency of send-receive turnaround
- Mitigation [Balakrishnan97]
  - Piggybacking link layer acks with data
  - Sending fewer TCP acks - ack every d-th packet (d may be chosen dynamically)
  - Ack filtering - Gateway may drop an older ack in the queue, if a new ack arrives

TCP Parameters after Route Repair

- Window Size after route repair
  - Same as before route break: may be too optimistic
  - Same as startup: may be too conservative
  - Better be conservative than overly optimistic
  - Reset window to small value; let TCP learn the window size

- Retransmission Timeout (RTO) after route repair
  - Same as before route break: may be too small for long routes
  - Same as TCP start-up: may be too large and respond slowly to packet loss
  - new RTO could be made a function of old RTO and route lengths

Improving TCP Throughput

- Network feedback
  - Network knows best (why packets are lost)
  - Need to modify transport & network layer to receive/send feedback
  - Need mechanisms for information exchange between layers
- Inform TCP of route failure by explicit message
- Let TCP know when route is repaired
  - Probing
  - Explicit notification
  - Better route caching mechanisms
- Reduces repeated TCP timeouts and backoff

Conclusion

Issues other than routing have received much less attention

Other interesting problems:

- Applications for MANET
- Address assignment
- QoS issues
- Improving interaction between protocol layers