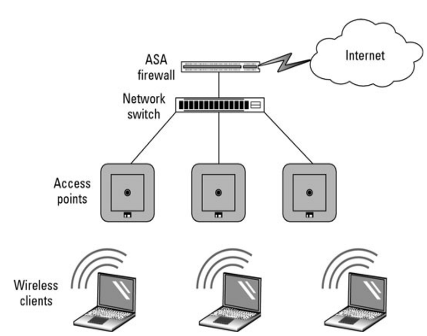
MOBILE COMPUTING

CSE 40814/60814

Spring 2021

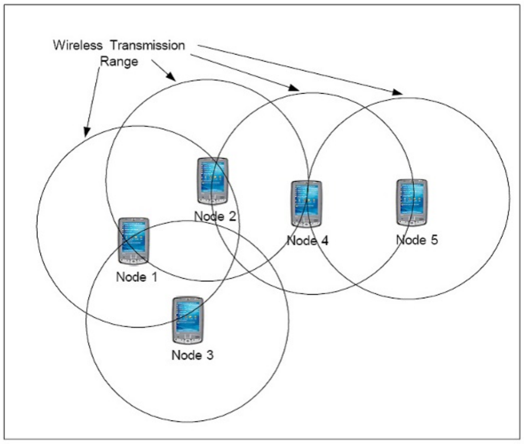
Infrastructure Networks

- Devices on the network all communicate through a single **access point:** a device that allows wireless devices to connect to a wired network using Wi-Fi
- **Problem:** the large overhead of maintaining the routing tables



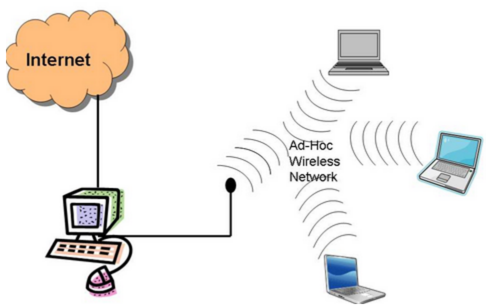
Infrastructure-Less (Ad-Hoc)

- Ad-hoc means '*for this purpose*'



Ad-Hoc Network

- **Decentralized** type of wireless network
- It is ad-hoc because it does not rely on
  - preexisting infrastructure such as routers in wired networks
  - access points in wireless networks



Mobile Ad-Hoc Network (MANET)

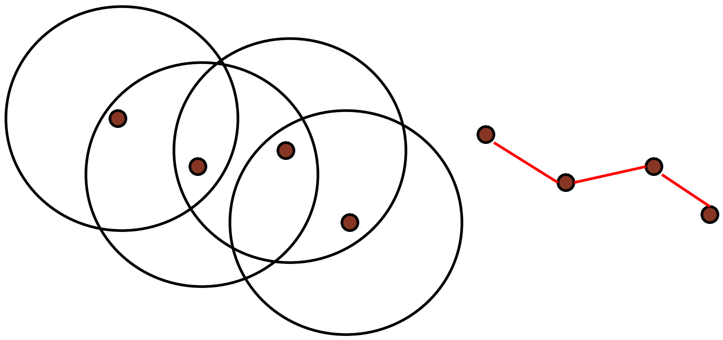
- It is a _continuously self-configuring_, _infrastructure-less_ network of **mobile devices** connected without wires
- Each device is free to move independently in any direction, and will therefore change its links to other devices frequently
- Hence, it has a **dynamic topology**
- The primary challenge in building a MANET is equipping each device to continuously maintain the information required to properly **route traffic**

Challenges

- **Infrastructure-less design** adds difficulty in fault detection and management
- **Dynamic topology** results in route changes and packet loss
- **Scalability** is still unsolved, challenges include addressing, routing, configuration management, interoperability, etc.
- **Varied link/node capabilities** cause variable processing capabilities
- **Energy constraints** limit processing power; ad-hoc networks rely on each node being a “router”

Routing

- Packets may need to traverse multiple links to reach destination
- Mobility causes route changes



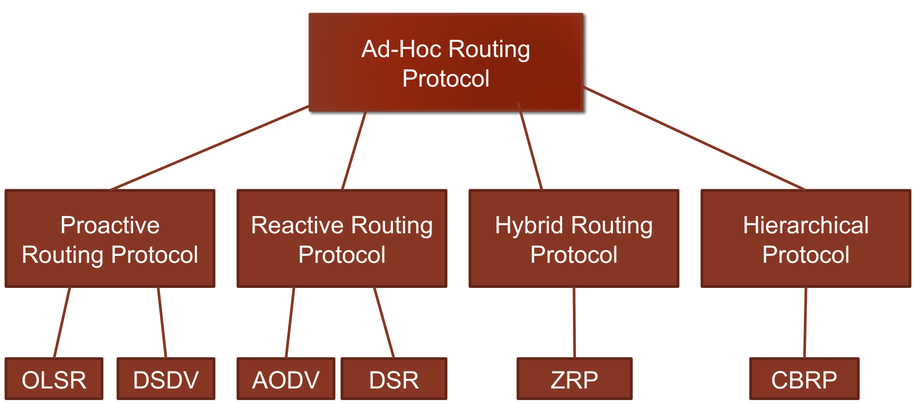
Ad-Hoc Routing Protocol

- An ad-hoc routing protocol is a convention that controls how nodes decide which way to route packets between computing devices in a mobile ad-hoc network
- Nodes are not familiar with the topology of their networks
- They have to discover it:
  - a new node announces its presence and listens for **announcements broadcast** (beacon or “alive” messages) by its neighbors
  - Each node learns about others nearby and how to reach them, and may announce that it too can reach them

Ad-Hoc Routing Protocol

- **Four Types:**
  - Table-driven (proactive) routing
    - Maintains fresh lists of destinations and their routes by periodically distributing routing tables throughout the network
  - On-demand (reactive) routing
    - Finds a route on demand by flooding the network with Route Request (RREQ) packets
  - Hybrid (both proactive and reactive) routing
    - Combines the advantages of proactive and reactive routing
  - Hierarchical routing protocols
    - The choice of proactive and of reactive routing depends on the hierarchic level in which a node resides (cluster-based routing)

Ad-Hoc Routing Protocols



Proactive Routing Protocol

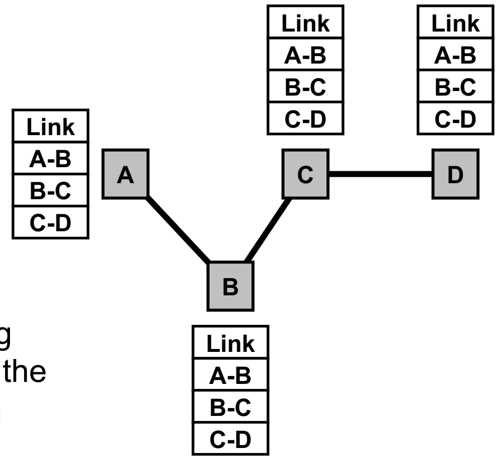
- Every node maintains **routing table** containing information about network topology
- Routing tables are updated periodically whenever the network topology changes
- These protocols maintain different numbers of routing tables varying from protocol to protocol
- Advantages
  - Route immediately available
  - Minimize **flooding**

OLSR – Optimized Link State Routing

- Proactive (table-driven) routing protocol
  - A route is available immediately when needed
- Based on the link-state algorithm
  - Traditionally, all nodes flood neighbor information in a link-state protocol, but not in OLSR

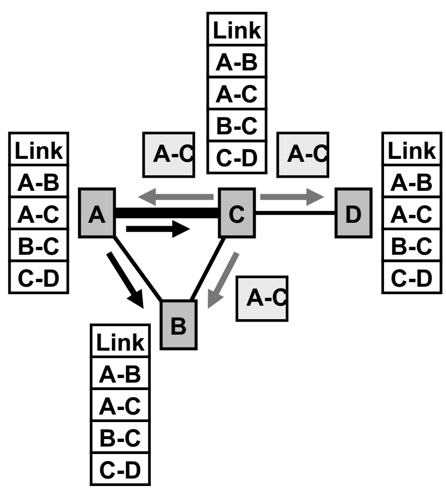
Link-State Algorithms

- Each node shares its link information so that all nodes can build a map of the full network topology
- Assuming the topology is stable for a sufficiently long period, all nodes will have the same topology information



Link-State Algorithms

- Link information is updated when a link changes state (goes up or down)
  - by sending small "hello" packets to neighbors
- Nodes A and C propagate the existence of link A-C to their neighbors and, eventually, to the entire network



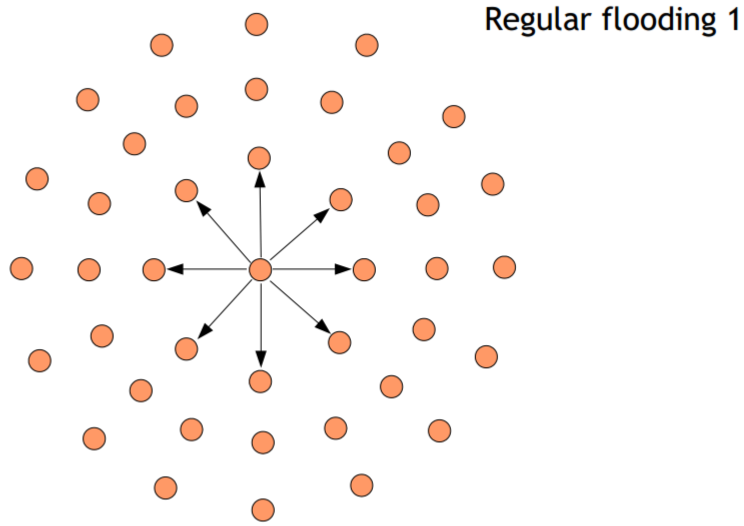
OLSR

- An optimization of Link State Protocol
  - **Reduces size of control packets :** Nodes advertise information only about links with neighbors who are in its *multipoint relay selector set*
  - **Reduces number of control packets by reducing duplicate transmissions :** Reduces flooding by using *only multipoint relay* nodes to send information in the network

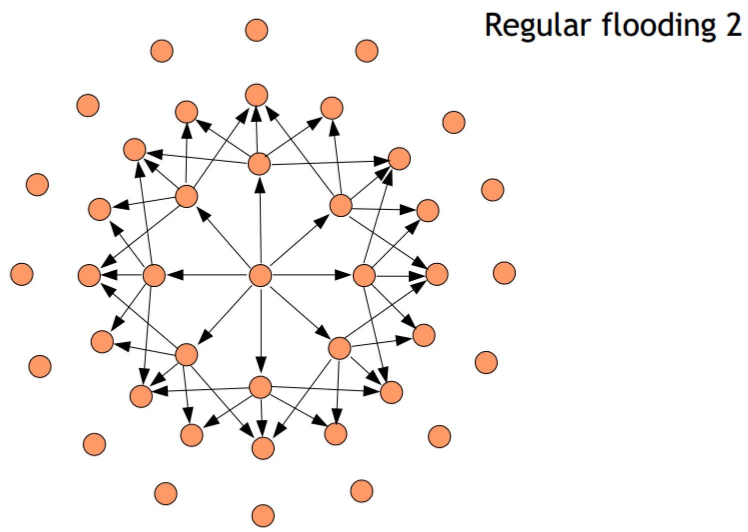
OLSR – Multipoint Relays

- MPRs = Set of selected neighbor nodes
- Minimize the flooding of broadcast packets

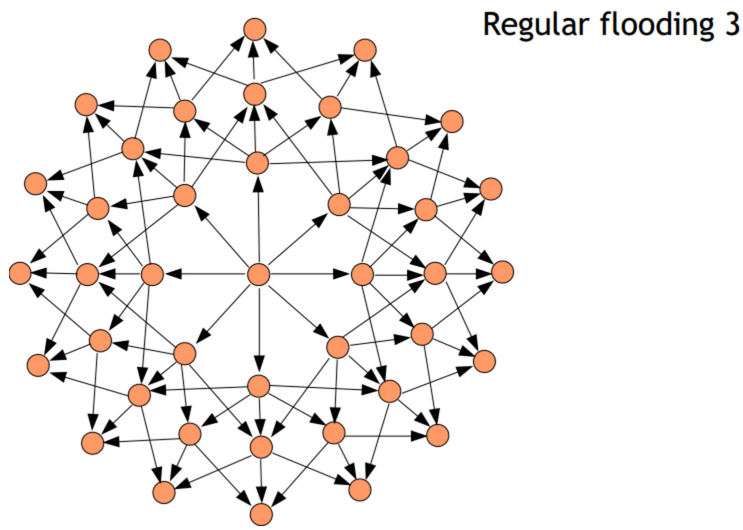
Regular Flooding



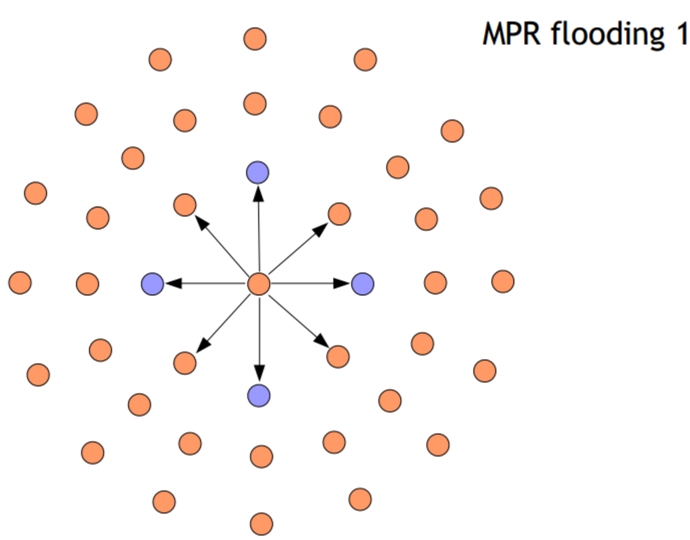
Regular Flooding



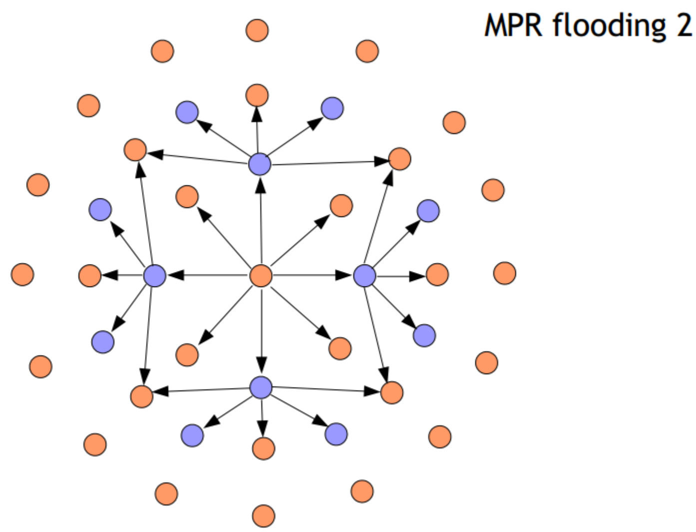
Regular Flooding



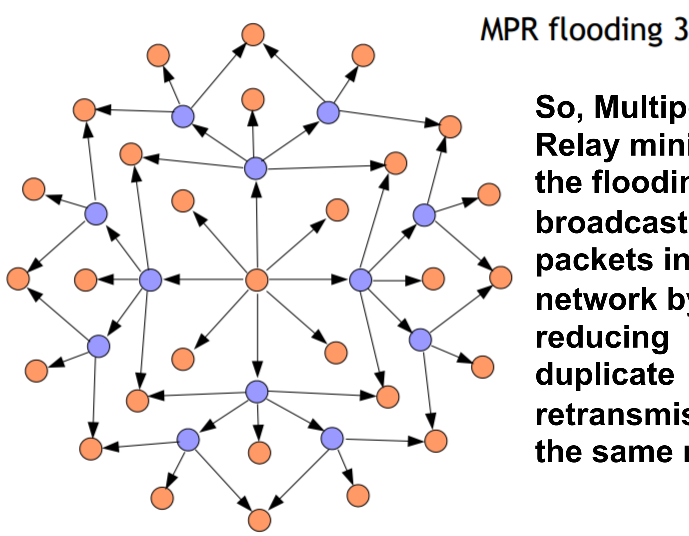
MRP Flooding



MRP Flooding

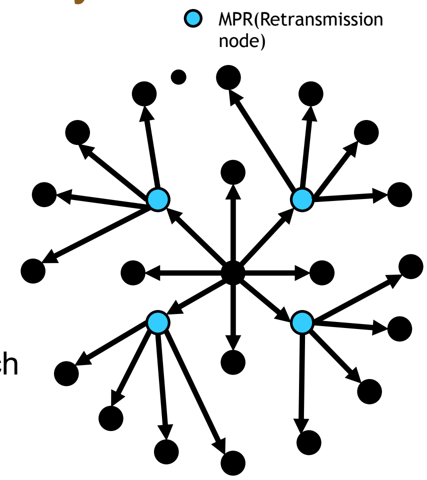


MRP Flooding



So, Multipoint Relay minimizes the flooding of broadcast packets in the network by reducing duplicate retransmission in the same region

OLSR – Multipoint Relays



- Each node selects its MPRs among its one hop neighbors
  - The set covers all the nodes that are two hops away
- These nodes retransmit the packets.
- The neighbors of any node, which are not in its MPR set, read and process the packet but do not retransmit the broadcast packet received from original node.

Neighbor Sensing

- Check for bi-directional links:
  - Each node periodically broadcasts its HELLO messages containing the information about its neighbors and their link status
  - Hello messages are received by all one-hop neighbors
- **HELLO message** contains:
  - List of addresses of the neighbors to which there exists a valid bi-directional link
  - List of addresses of the neighbors which are heard by node (a HELLO has been received)
    - But link is not yet validated as bi-directional

Neighbor Sensing

- HELLO messages :
  - Serves for link sensing
  - Permits each node to learn about its neighbors within up to two-hops (neighbor detection)
  - On the basis of this information, each node performs the selection of its multipoint relays in OLSR

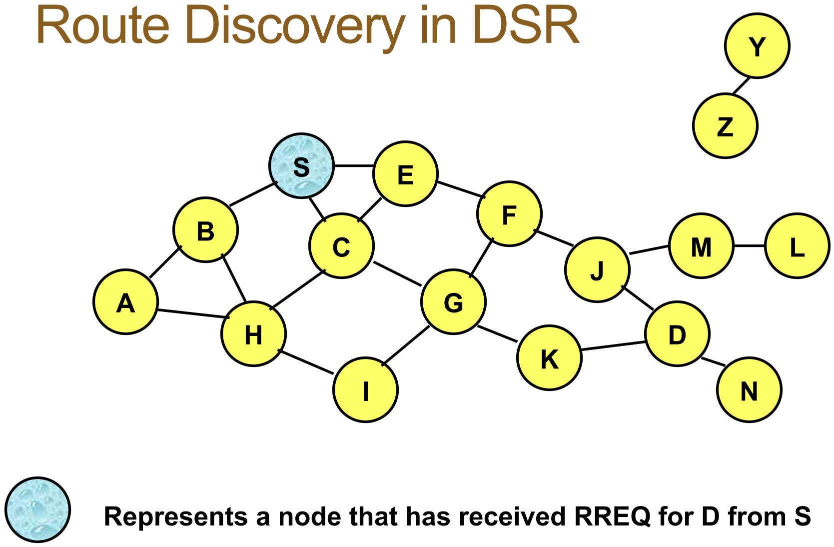
Dynamic Source Routing (DSR)

- Each packet header contains a route, which is represented as a complete sequence of nodes between a source-destination pair
- Protocol consists of two phases
  - **route discovery**
  - **route maintenance**
- Optimizations for efficiency
  - Route cache
  - Piggybacking
  - Error handling

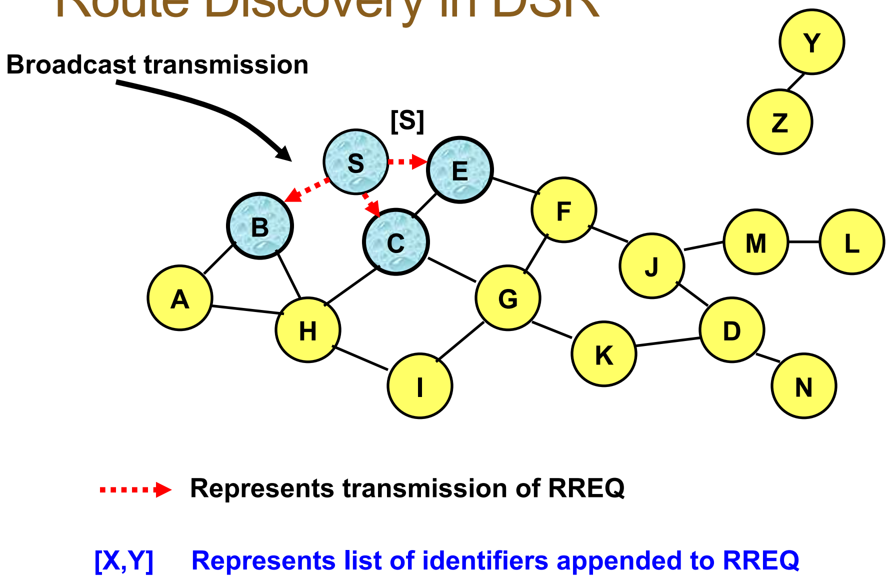
DSR Route Discovery

- Source broadcasts route request **RREQ** (contains sender & target)
- Intermediate node action:
  - Discard if node is source or node is in *route record*
  - If node is the target, *route record* contains the full route to the target; return a route reply **RREP**
  - Else append address in *route record*; rebroadcast
- Use existing routes to source to send route rep*ly*

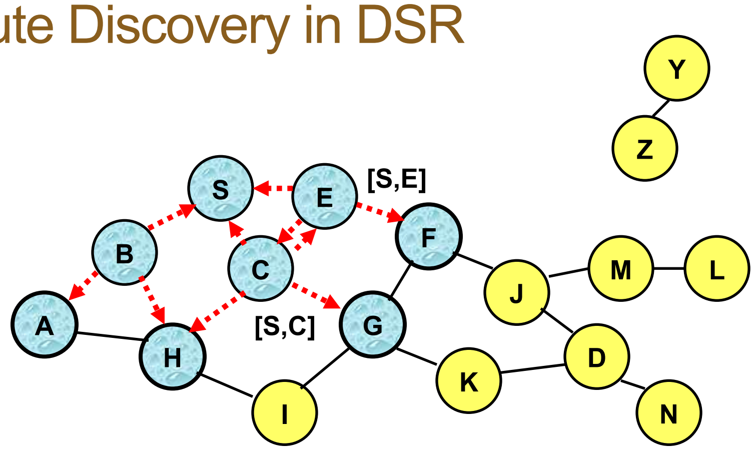
Route Discovery in DSR



Route Discovery in DSR

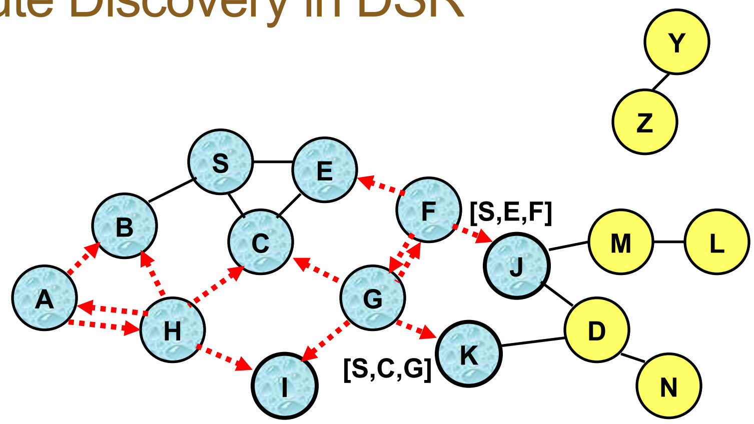


Route Discovery in DSR



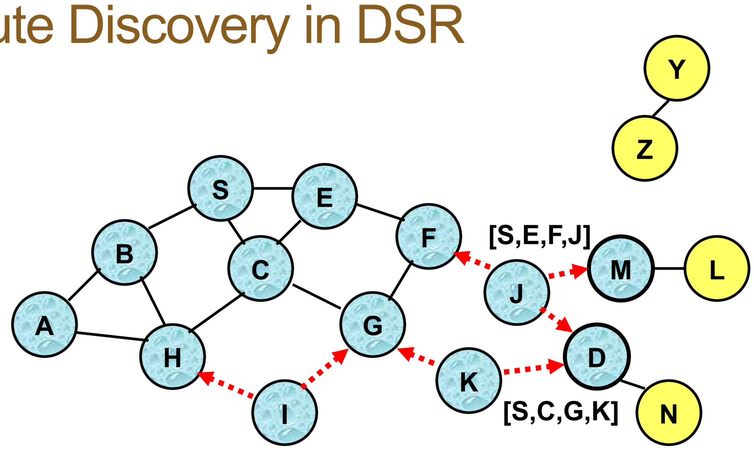
- Node H receives packet RREQ from two neighbors: potential for collision

Route Discovery in DSR



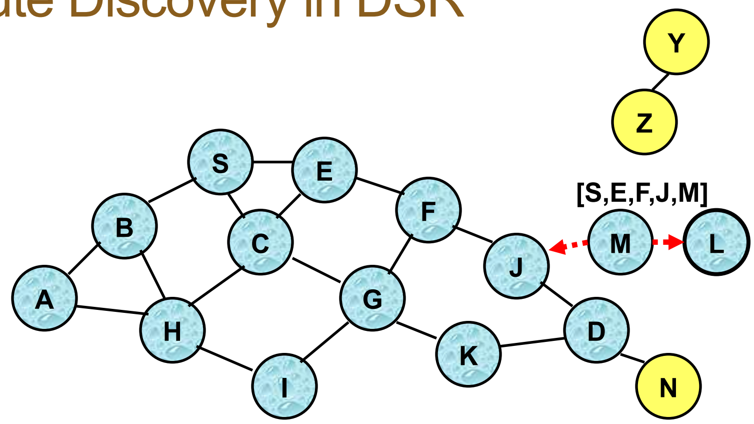
- Node C receives RREQ from G and H, but does not forward it again, because node C has already forwarded RREQ once

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

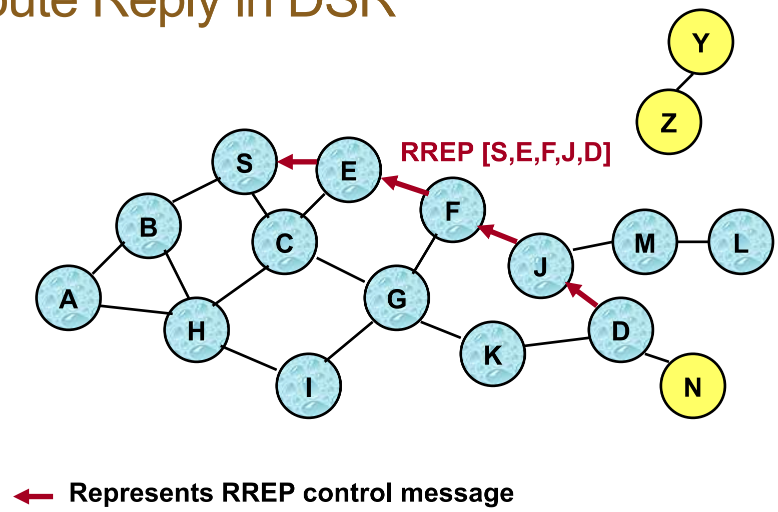


- Node D **does not forward** RREQ, because node D is the **intended target** of the route discovery

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

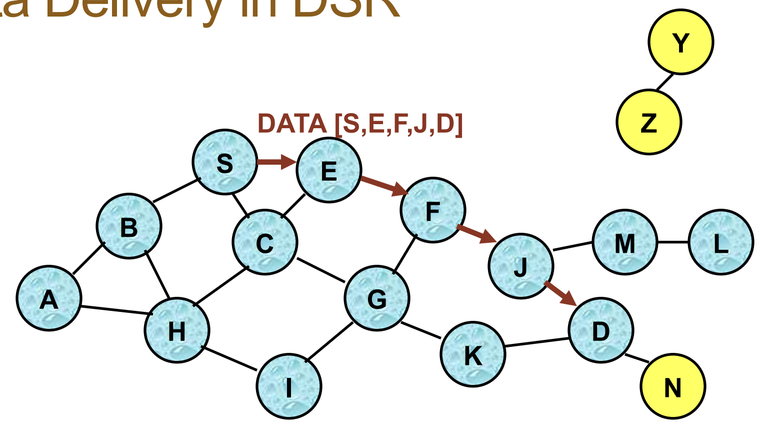
Route Reply in DSR



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 ("source routing")
- Intermediate nodes use the source route included in a packet to determine the neighbor to send the packet

Data Delivery in DSR



Packet header size grows with route length

Route Caching

- Source node S learns [S,E,F,J,D]:
  - What does S know?
- K gets route request [S,C,G]:
  - What does K know?
- F forwards route reply [S,E,F,J,D]:
  - What does F know?
- Neighbors overhear packets and can learn routes
- Cache this information and use when needed

- Problem: information ages! (“stale cache”)

DSR: Advantages

- Only establish/maintain routes between nodes needed them
  - Cheaper route management
  - In contrast: tables (LS, DV) store ALL routes
- Route caching further reduces management cost
- A single route discovery may yield many routes

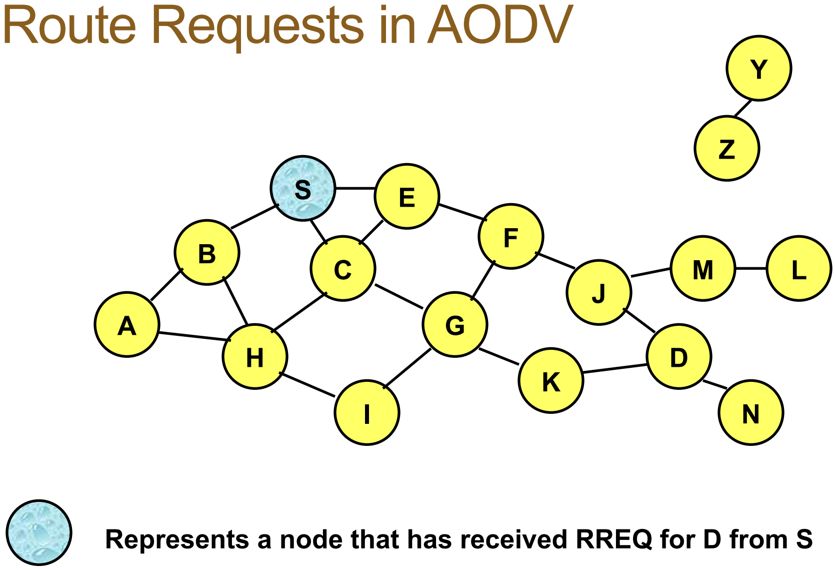
DSR: Disadvantages

- Packet header size grows with route length
- Route request requires flooding
- Rebroadcasting may lead to collisions
  - Use random delays (what does that remind you of?)
- Many route replies may come back (local caches)
  - More contention, “route reply storm” problem
- Stale caches contain outdated routes
- Initial delay before transmissions can begin
  - In contrast: table-based protocols are ready immediately

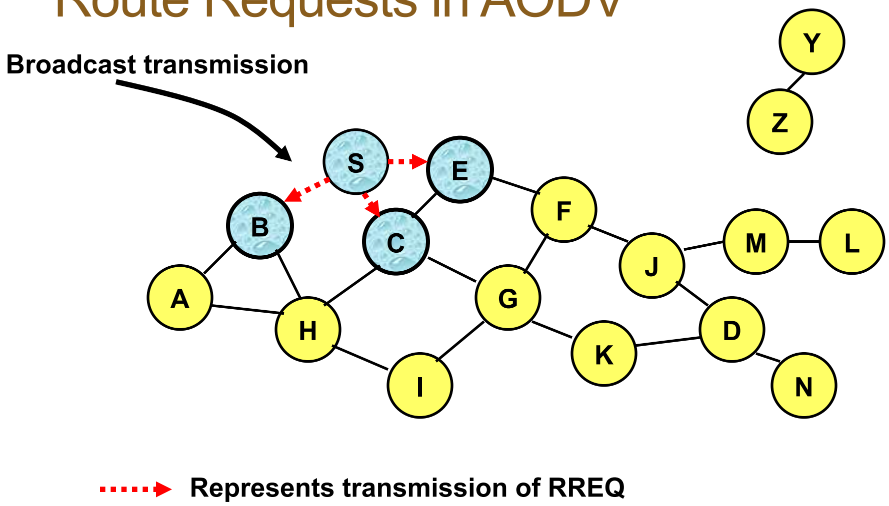
AODV

- RREQs for route discovery, similar to DSR
- Does NOT store route in packets
- Instead, each forwarder remembers reverse path to transmitter
- Target replies with RREP; travels along reverse path

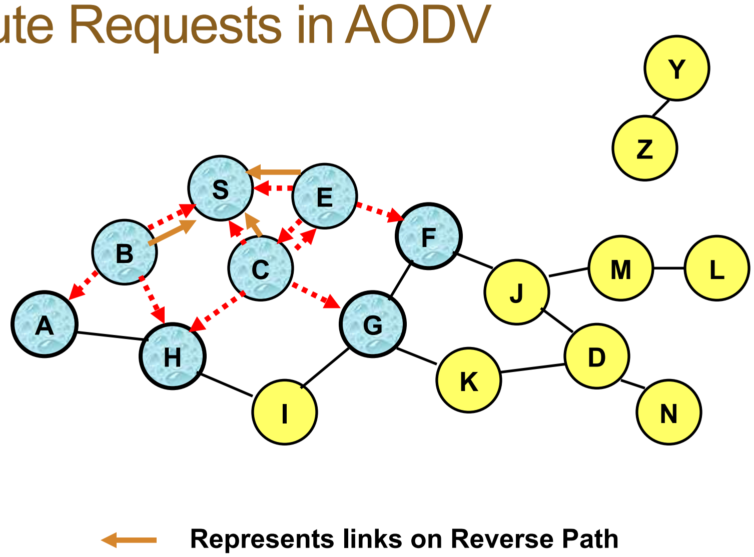
Route Requests in AODV



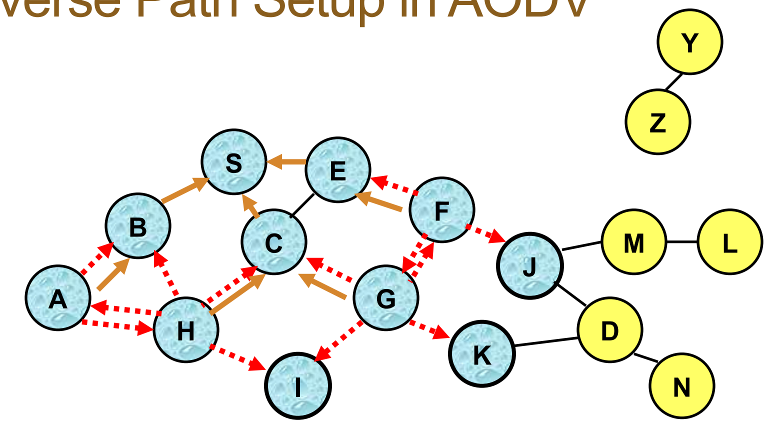
Route Requests in AODV



Route Requests in AODV

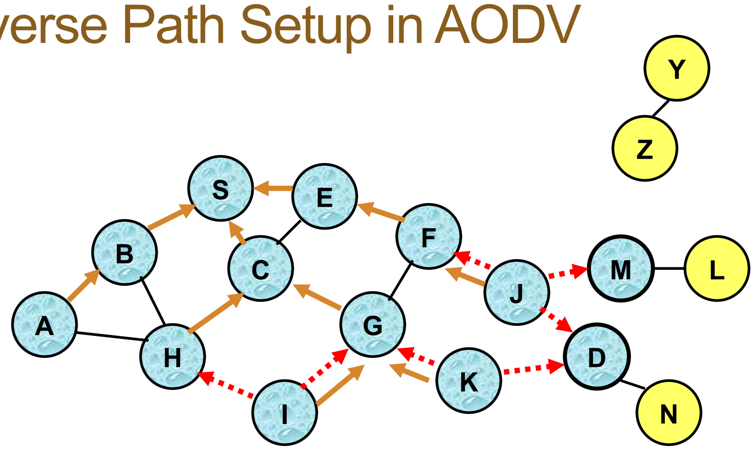


Reverse Path Setup in AODV

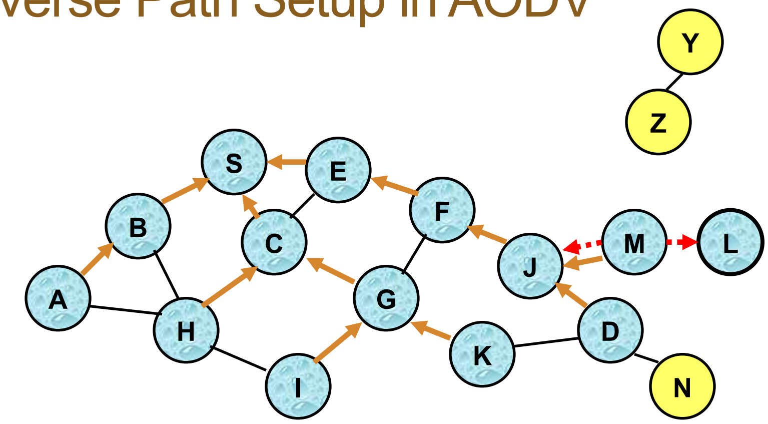


- Node C receives RREQ from G and H, but does not forward it again, because node C has **already forwarded RREQ** once

Reverse Path Setup in AODV

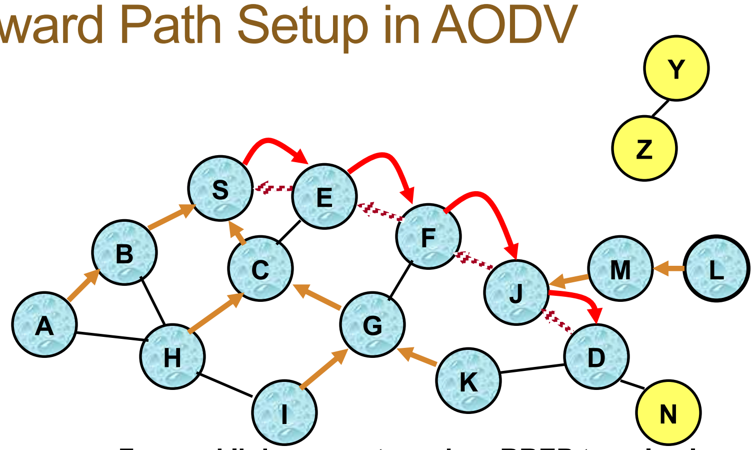


Reverse Path Setup in AODV



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

Forward Path Setup in AODV



Forward links are setup when RREP travels along the reverse path

 Represents a link on the forward path

AODV

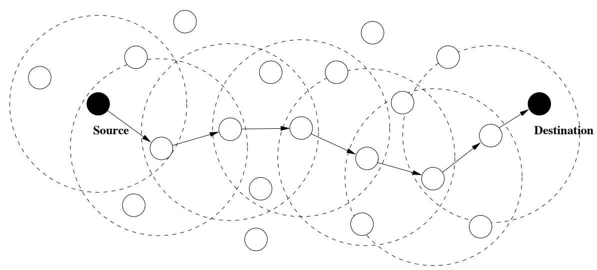
- Routes need not be included in packet headers
- Nodes maintain routing tables containing entries only for routes that are in active use
- At most one next-hop per destination maintained at each node
  - DSR may maintain several routes for a single destination
- Sequence numbers are used to avoid old/broken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

Location-Based Routing

- Also referred to as geographic routing
- Used when nodes are able to determine their (approximate) positions
- Nodes use location information to make routing decisions
  - sender must know the locations of itself, the destination, and its neighbors
  - location information can be queried or obtained from a location broker
- Types of geographic routing:
  - unicast: single destination
  - multicast: multiple destinations
  - geocast: data is propagated to nodes within certain geographic area

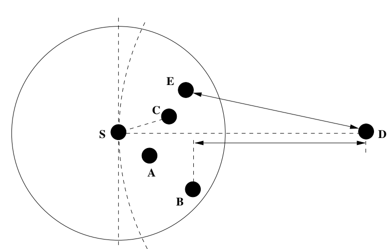
Unicast Location-Based Routing

- One single destination
- Each forwarding node makes localized decision based on the location of the destination and the node's neighbors (greedy forwarding)
- Challenge: packet may arrive at a node without neighbors that could bring packet closer to the destination (voids or holes)



Forwarding Strategies

- Greedy: minimize distance to destination in each hop
- Nearest with Forwarding Progress (NFP): nearest of all neighbors that make positive progress (in terms of geographic distance) toward destination
- Most Forwarding Progress within Radius (MFR): neighbor that makes greatest positive progress (progress is distance between source and its neighbor node projected onto a line drawn from source to destination)
- Compass Routing: neighbor with smallest angle between a line drawn from source to the neighbor and the line connecting source and destination



Geocasting

- Packet is sent to all or some nodes within specific geographic region
- Example: query sent to all sensors within geographic area of interest

- Routing challenge:
  - propagate a packet near the target region (similar to unicast routing)
  - distribute packet within the target region (similar to multicast routing)

Geographic-Forwarding-Perimeter-Geocast