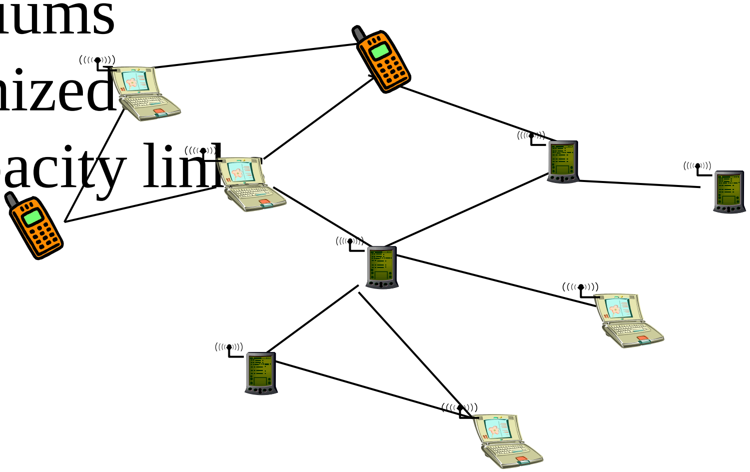
Ad-Hoc Networks

Mestrado em Engenharia de Computadores e Telemática

2023/2024

Mobile Ad-hoc networks

- Terminals may appear and disappear anywhere and anytime, and may freely move
- Nodes can act as routers or terminals
- Networks independently formed, can be merged and splitted anytime
- Dynamic topologies
- Coexistence of different access mediums
- Network is intelligent and self-organized
- Bandwidth constrained, variable capacity link
- Energy constrained operation
- Limited physical security



Challenges in Mobile Environments – Ad-hoc increases them

- Limitations of the wireless network
  - Lack of central entity for organization available
  - Limited range of wireless communication
  - 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

Application Scenarios

Ad-hoc applications

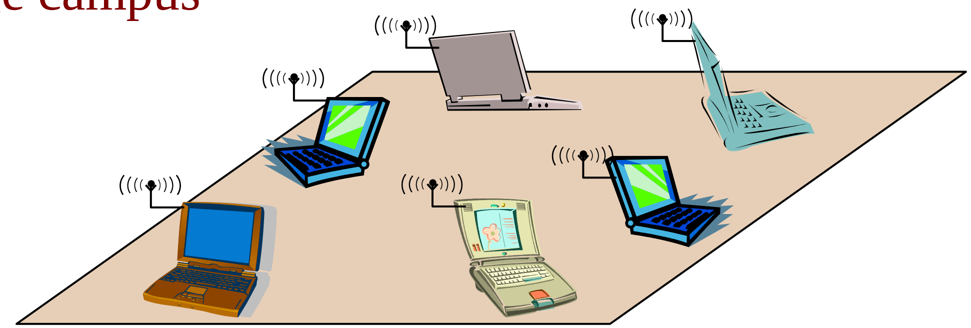
- Personal area networking
  - Cell phone, laptop, ear phone, wrist watch
- Military environments
  - Soldiers, tanks, planes
- Civilian environments
  - Taxi cab network
  - Meeting rooms
  - Sports stadiums
  - Boats, small aircraft
- Emergency operations
  - Search-and-rescue
  - Policing and fire fighting

Usage scenarios – in general

- 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
- Or when the objective is to have
  - Self-adapting and self-sufficient networks
  - Networks that require mobility
  - Moving networks
  - Requirement to absent any external configuration and management process

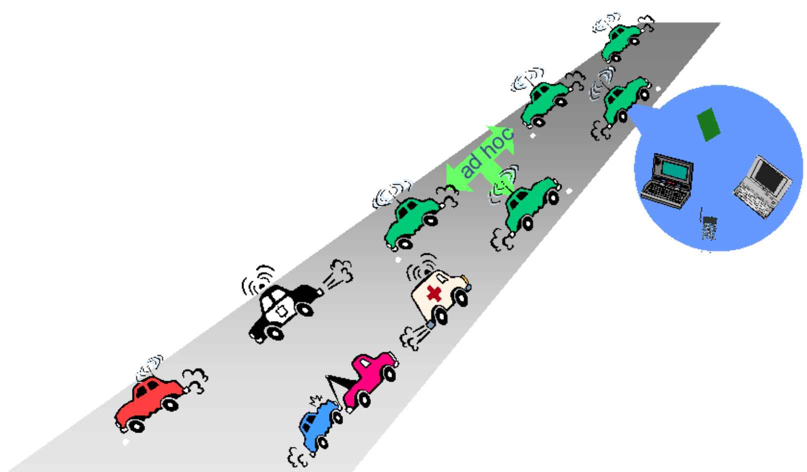
Civilian environments

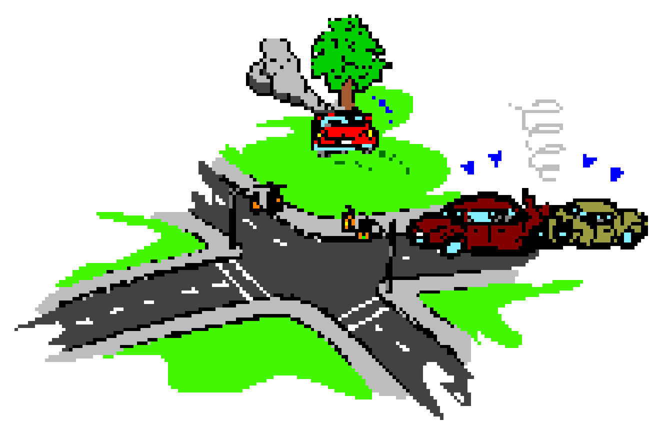
- Computer science classroom
  - Ad-hoc network between student laptops
- Conference
  - Users in different rooms accessing services through other users
- Shopping mall, restaurant, coffee shops
  - Customers spend part of the day in a networked mall of speciality shops, coffee shops, and restaurants
- Large campus
  - Employees of a company moving within a large campus with laptops, and cellphones



Civilian environments

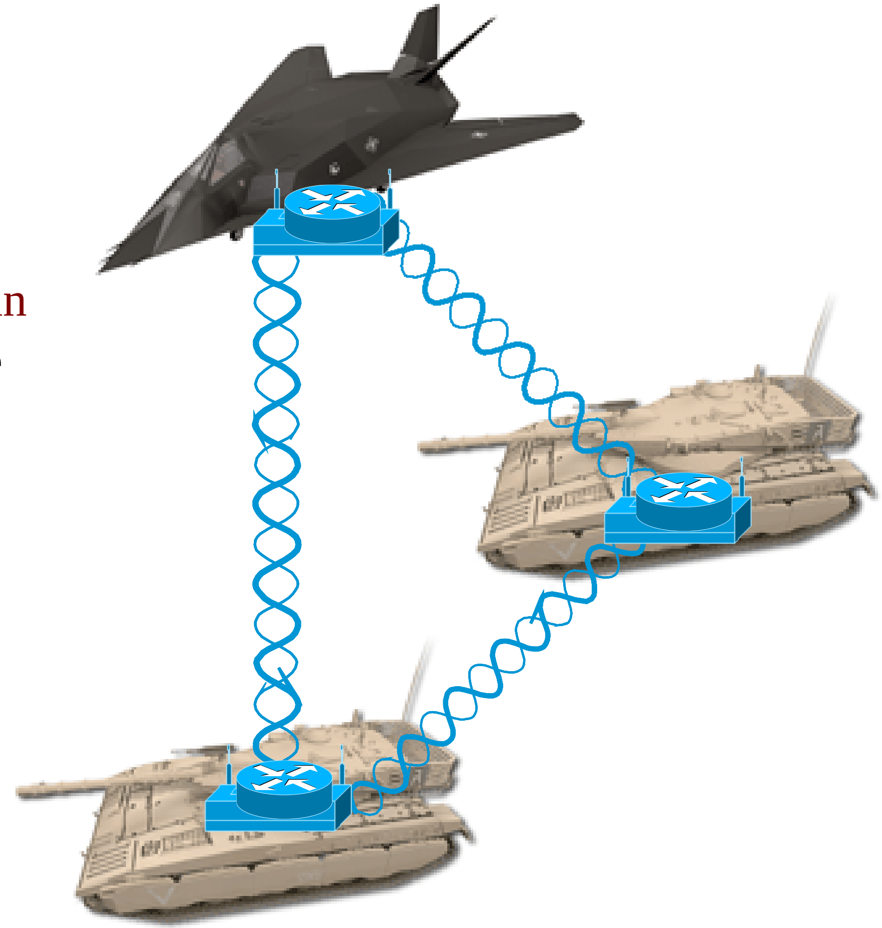
- Traffic networks (smart cars and
smart roads)
- Board systems talk with the road
  - Map delays and blocks
  - Obtain maps
  - Inform the road about its actions
- Finding out empty parking lots in a city, without asking a server
- Car-to-car communication





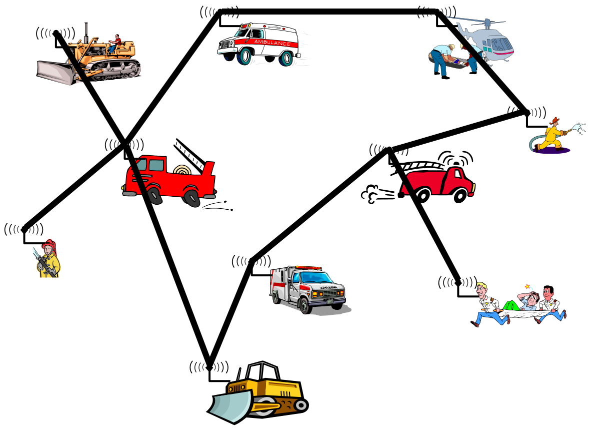
Military environments

- Combat regiment in the field
  - Around 4000-8000 objects in constant and umpredictable movement
- Force intercommunication
  - Proximity, function, battle plan
- Moving soldiers with wearable computers
  - Eavesdropping, denial-of-service and impersonation attacks can be launched
- Advantages
  - Low detection probability
  - Random topology and association between nodes

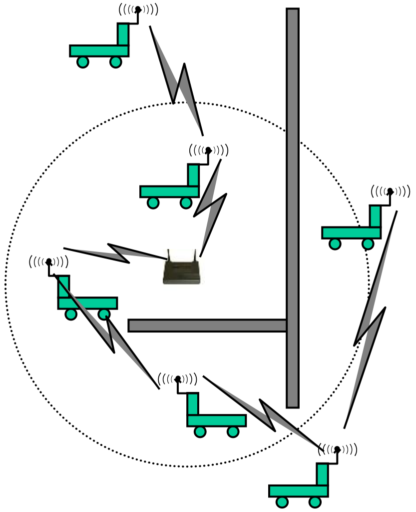


Others...

- Disaster recovery



- Factory floor automation



Routing

Routing: Challenges and Requirements

- Major challenges
  - Mobility – path breaks, packet collisions, transient loops
  - Bandwidth constraint – channel shared by all nodes in the broadcast region
  - Error-prone and shared channel – take into account the larger BERs in wireless ad-hoc
  - Location-dependent contention – high when the number of nodes increases
- Major requirements
  - Minimum route acquisition delay
  - Quick route reconfiguration (handle path breaks)
  - Loop-free routing (avoid waste of resources)
  - Distributed routing approach (reduce bandwitdth consumed)
  - Minimum control overhead (bandwidth, collisions)
  - Scalability (scale with large network – minimize control overhead)
  - Provisioning of QoS (provide QoS levels) - support for time-sensitive traffic
  - Security and privacy (resilient to threats and vulnerabilities)

Proactive and Reactive Protocols

- 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

AODV - Ad Hoc On-Demand Distance Vector Routing

Ad Hoc On-Demand Distance Vector Routing (AODV)

- AODV maintains routing tables at the nodes, so that data packets do not have to contain routes
- Routes are maintained only between nodes which need to communicate

AODV operation

- Route Requests (RREQ)
- 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 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

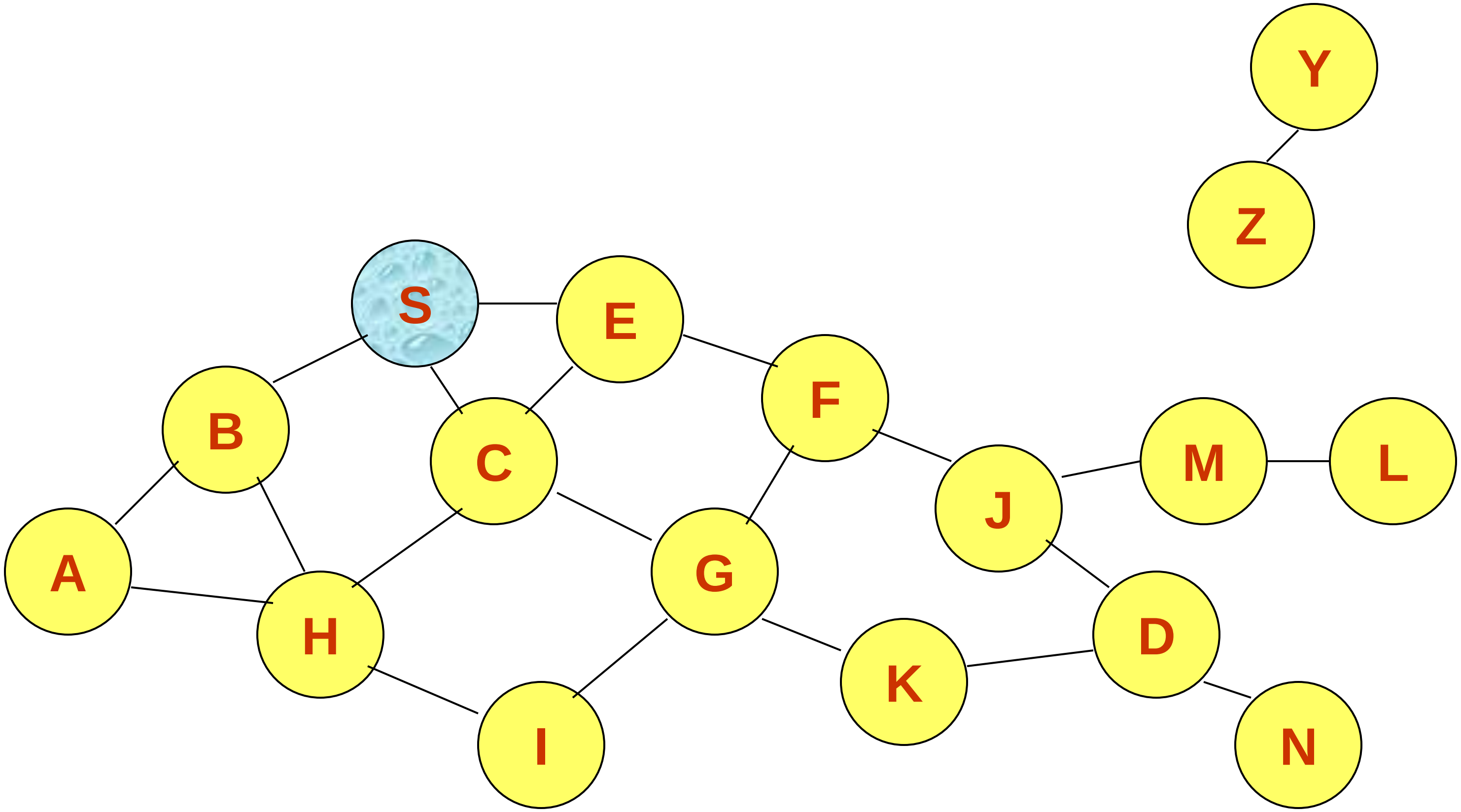
```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-
|     Type      |J|R|G|D|U|   Reserved          |   Hop Count   |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-
|                            RREQ ID                            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-
|                    Destination IP Address                     |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-
|                  Destination Sequence Number                  |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-
|                    Originator IP Address                      |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-
|                  Originator Sequence Number                   |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-
```

AODV operation

- Each node maintains non-decreasing sequence numbers
  - Sent in RREQ, RREP messages; incremented with each new message
  - Used to "timestamp" routing table entries for "freshness" comparison
- Intermediate node may return RREP if it has routing table entry for destination which is "fresher" than source's (or equal with lower hop count)

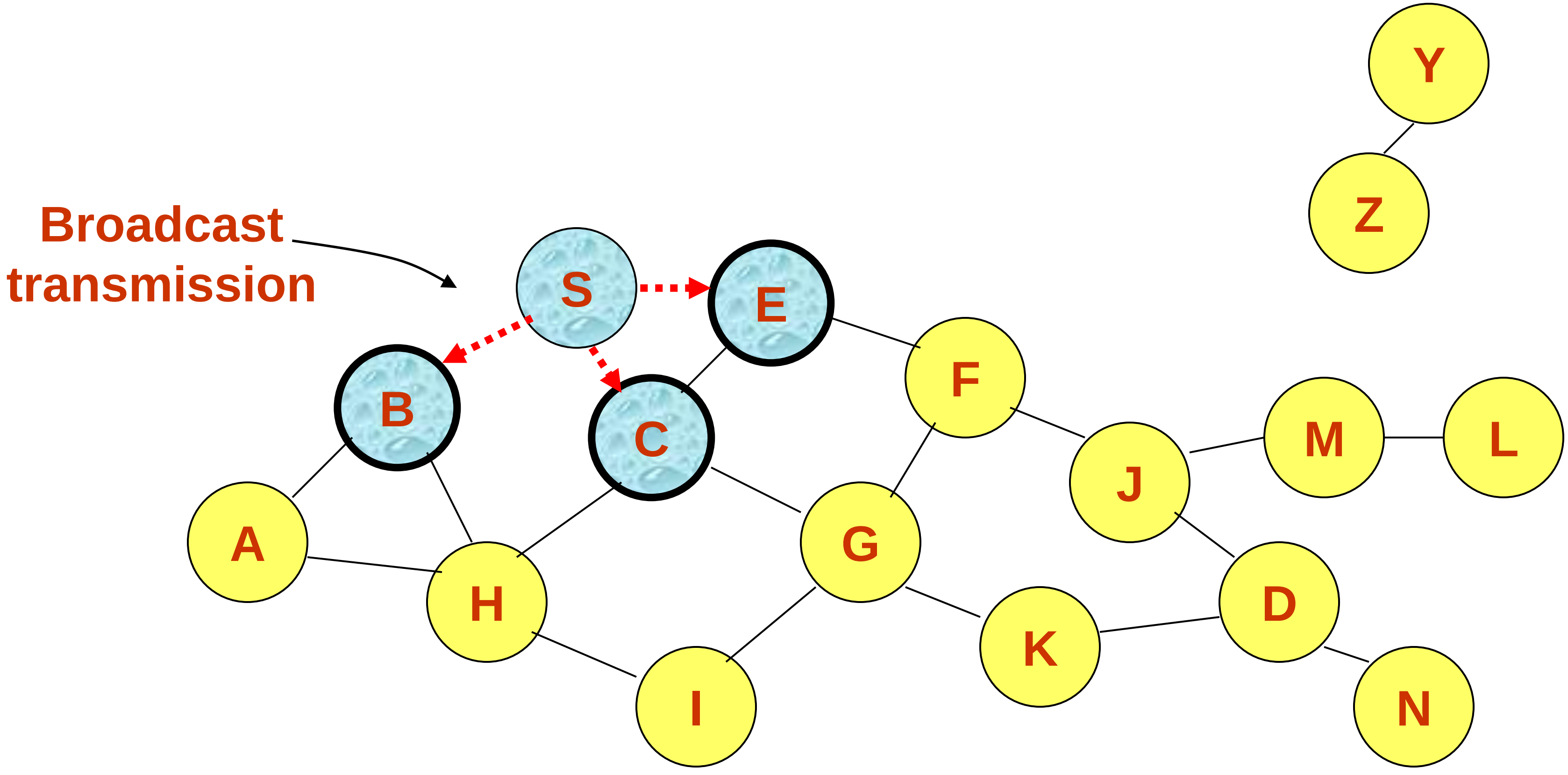
- Routing table entries assigned "lifetime", deleted on expiration
- Unique ID included in RREQ for duplicate rejection

Route Requests in AODV



Represents a node that has received RREQ for D from S

Route Requests in AODV



Represents transmission of RREQ

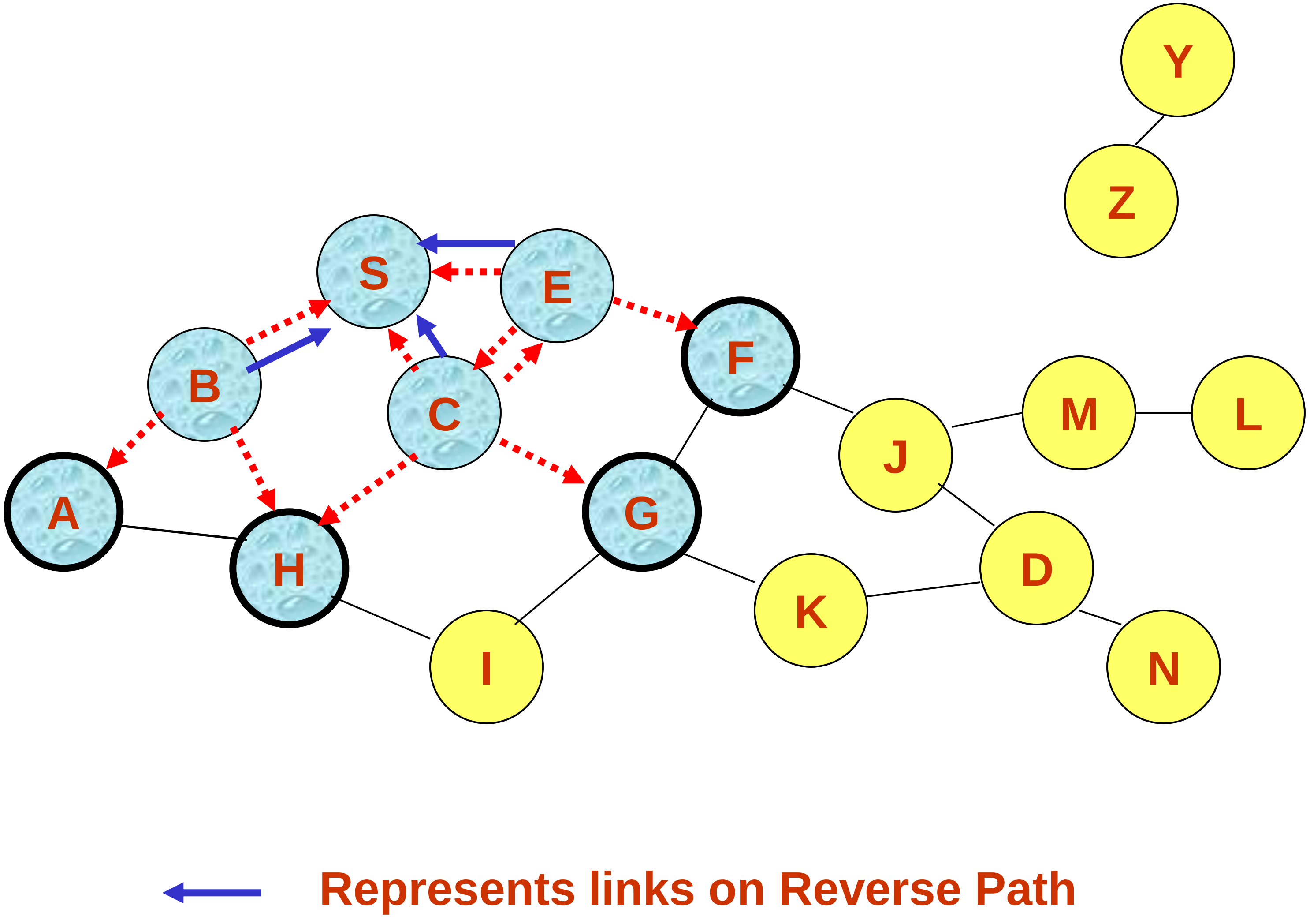
Node E receives RREQ

Makes reverse route entry for S

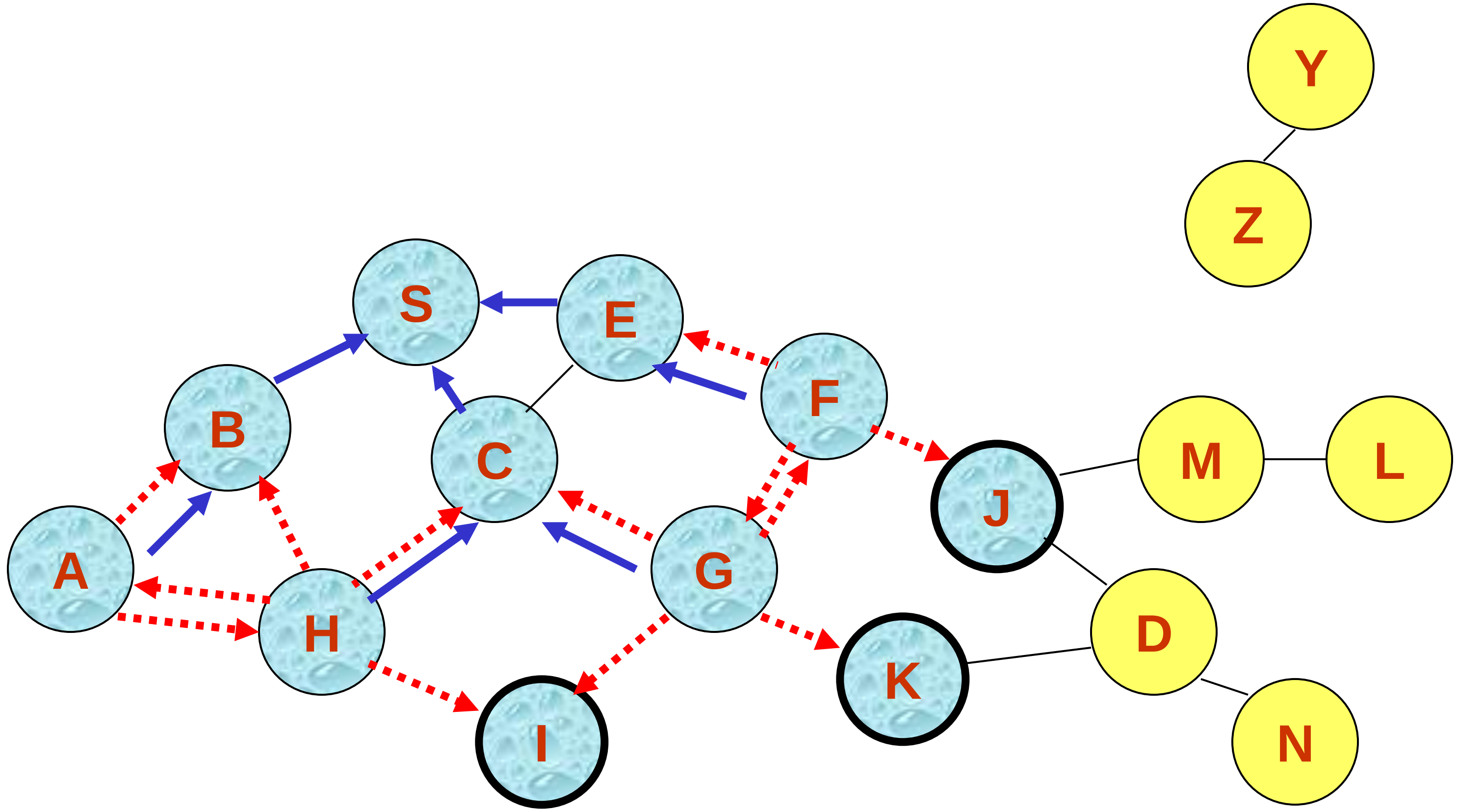
dest = S, next hop = S, hop cnt = 1

It has no route to D, so it rebroadcasts RREQ

Route Requests in AODV

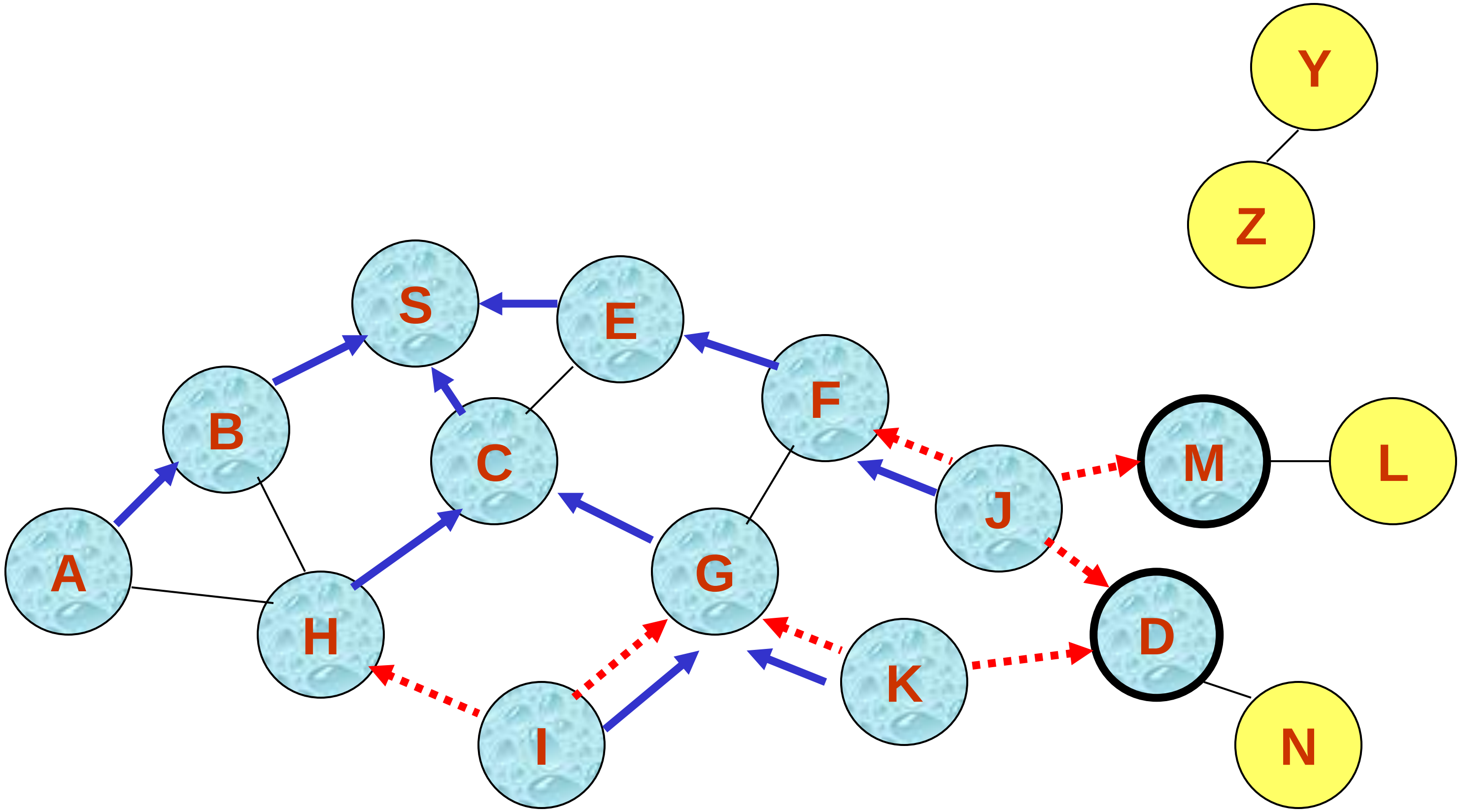


Reverse Path Setup in AODV



- 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



Node J receives RREQ

Makes reverse route entry for S, dest = S, next hop = F, hop cnt = 3

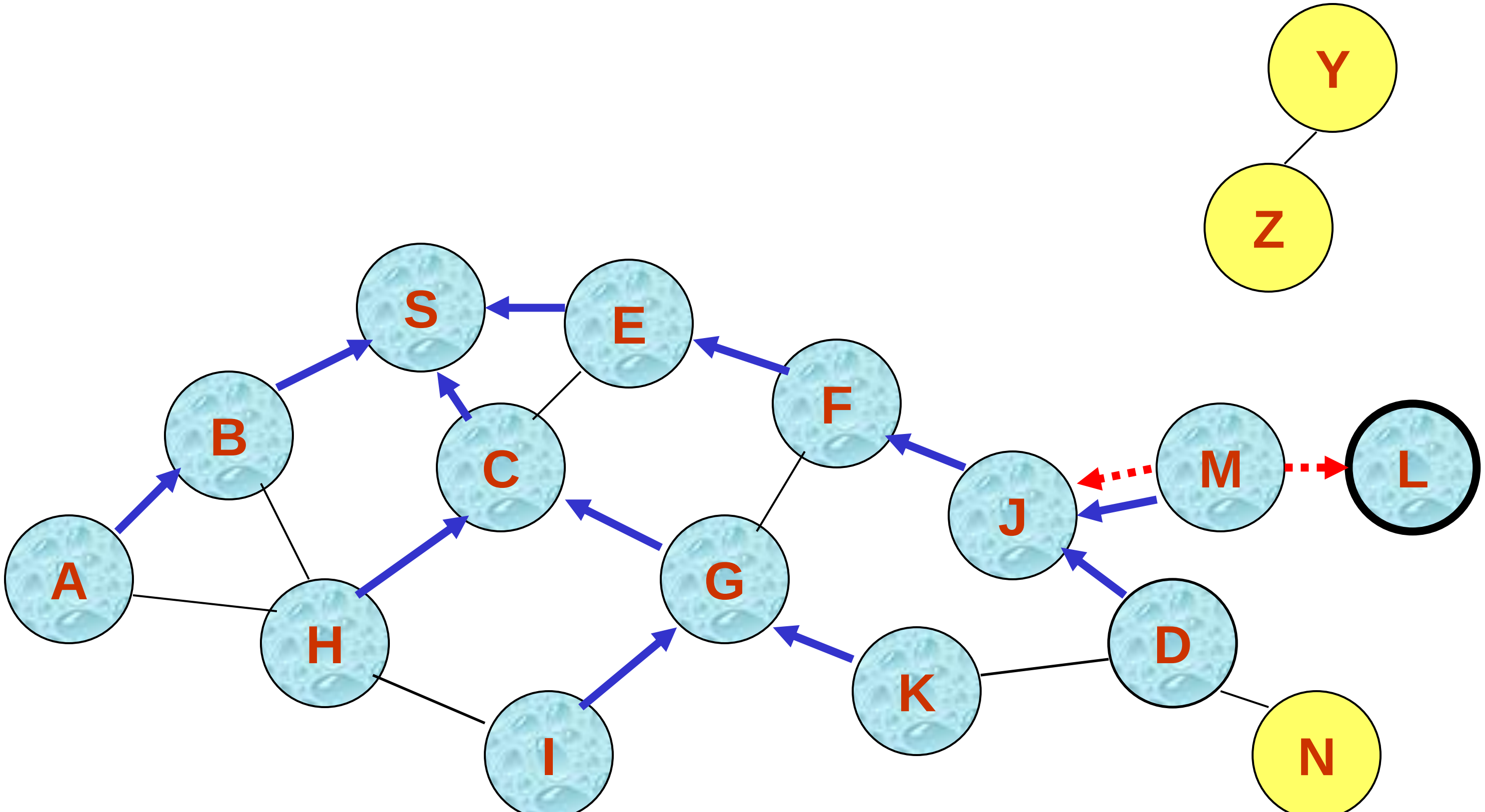
It has a route to D, and the seq# of the route to D is <D's seq# in RREQ (outdated route)

Or

Makes reverse route entry for S, dest = S, next hop = F, hop cnt = 3

It has a route to D, and the seq# of the route to D is > = D's seq# in RREQ (updated route)

Reverse Path Setup in AODV



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

Node D sends RREP

D creates a Route Reply (RREP), Enters D's IP addr, seq #S's IP addr, hop count to D (=0)

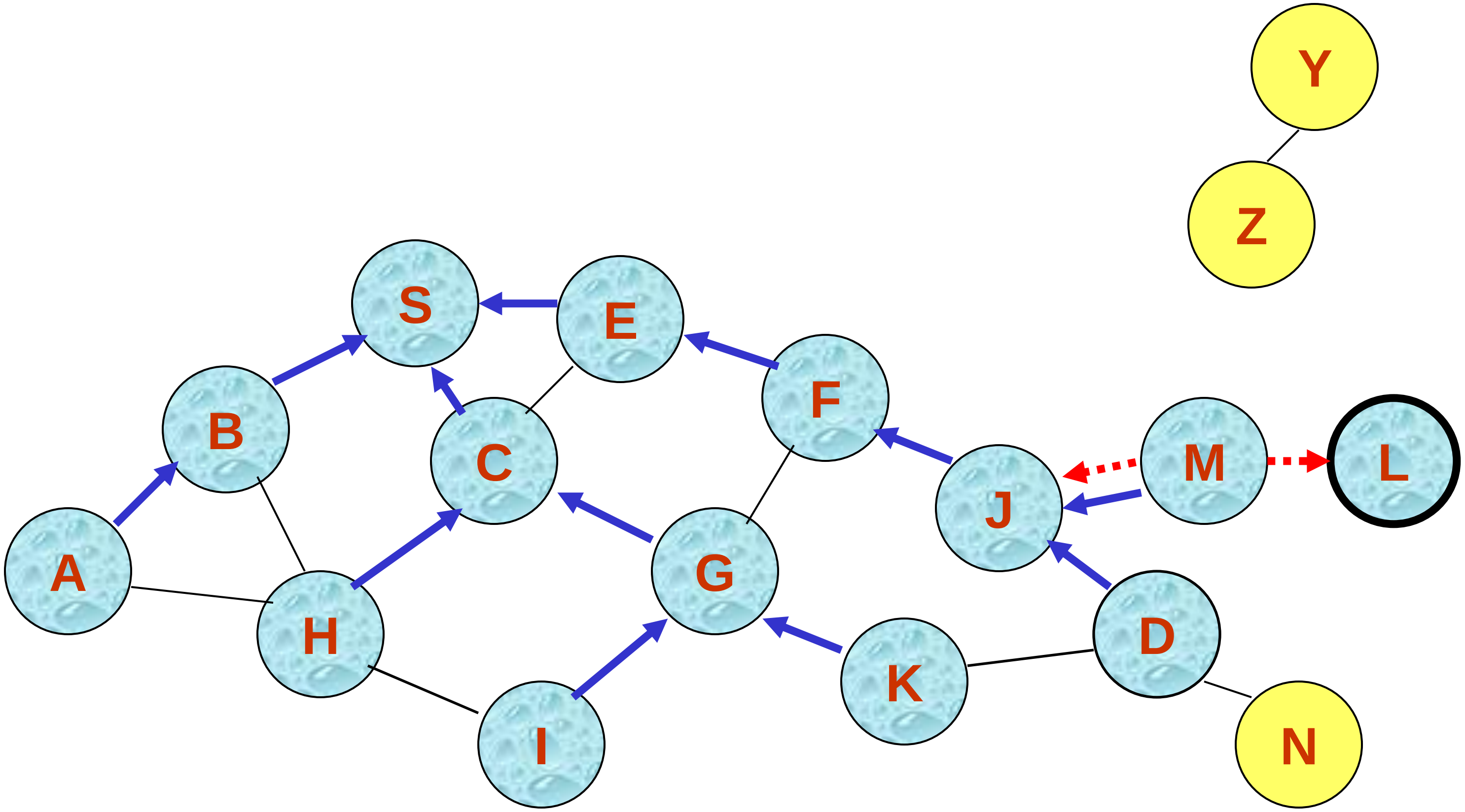
Unicasts RREP towards J

Or Node J sends RREP

J creates a Route Reply (RREP), Enters D's IP addr, seq #S's IP addr, hop count to D (=1)

Unicasts RREP towards F

Reverse Path Setup in AODV



Node E receives RREP

Makes forward route entry to D

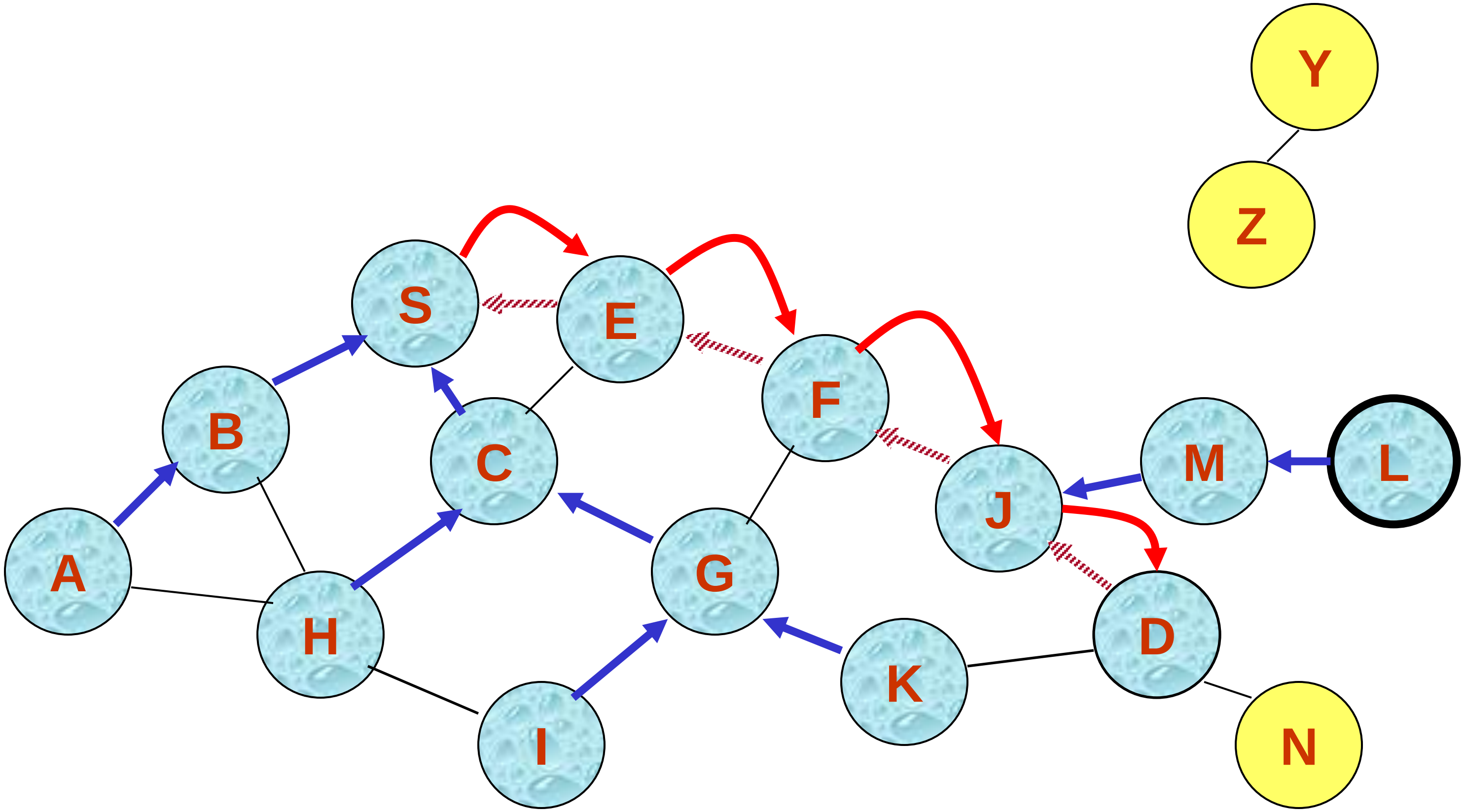
dest = D, next hop = F, <u>hop count = 3</u>, Lifetime, Unicasts RREP to S

Node S receives RREP

Makes forward route entry to D

dest = D, next hop = E, <u>hop count = 4</u>, Lifetime

Forward Path Setup in AODV



Forward links are setup when RREP travels along the reverse path

If multiple replies, uses one with lowest hop count

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 messages
- Periodic route response to neighbors acts as hello, installing and refreshing route
- 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*

Local RERR

- Used when link breakage occurs
  - Link breakage detected by link-layer ACK, "passive ACK", AODV "Hello" messages
- Detecting node may attempt "local repair"
  - Send RREQ for destination from intermediate node
- Route Error (RERR) message generated
  - Sent to "precursors": neighbors who recently sent packet which was forwarded over the broken link
    - Propagated recursively

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
- Sequence numbers are used to avoid old/broken routes
- Unused routes expire even if topology does not change

Proactive routing protocols

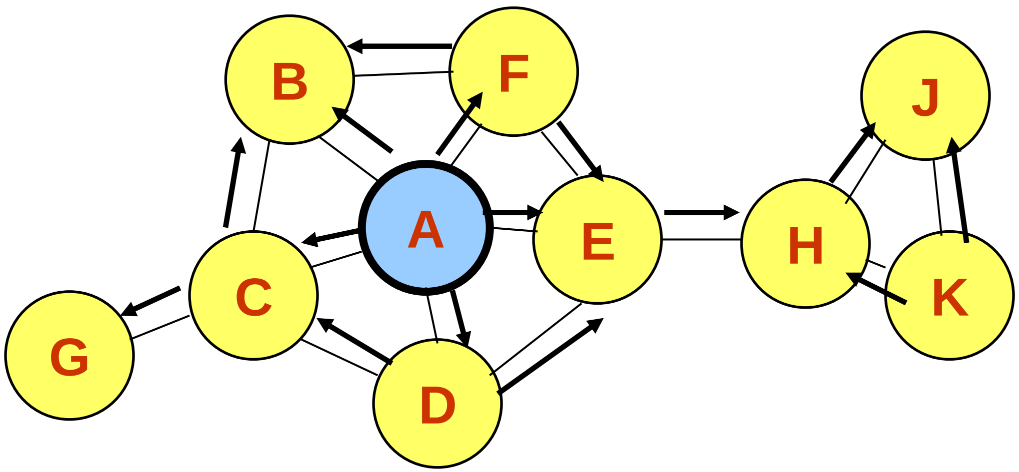
OLSR - Optimized Link State Routing Protocol

Optimized Link State Routing Protocol (OLSR)

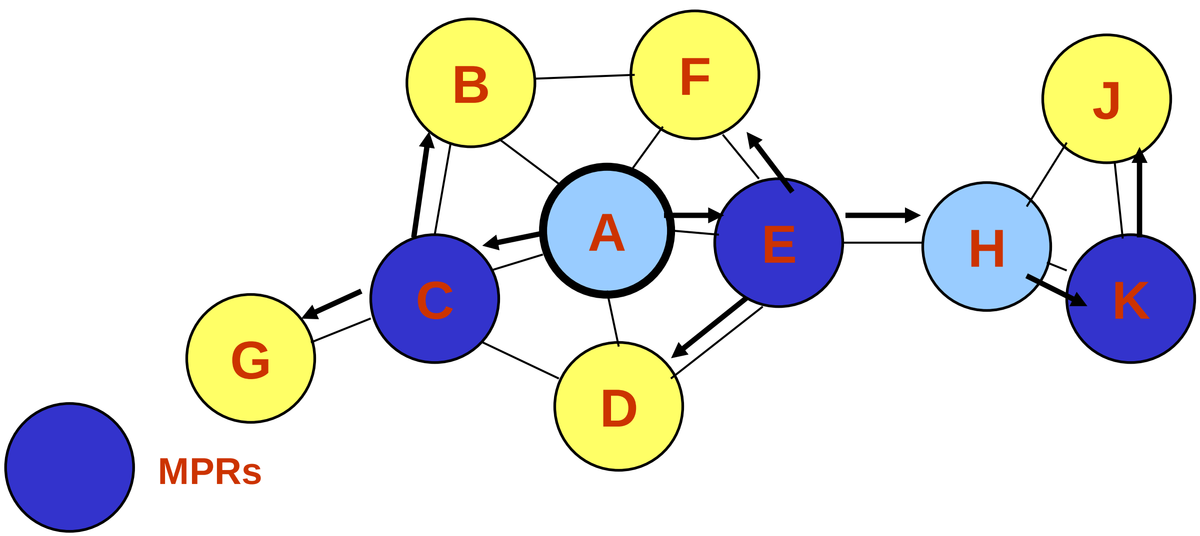
- **Proactive protocol**
- **Efficient link state packet forwarding mechanism**
  - **Multipoint relaying**
    - Reduced size of the control packets
      - Only a subset of the links in the link state updates
        - » Packet forwarding performed only by multipoint relays
    - Reduced number of links used for forwarding the link state packets
      - Multipoint relays

Example of MPR in OLSR

- **Simple flooding**

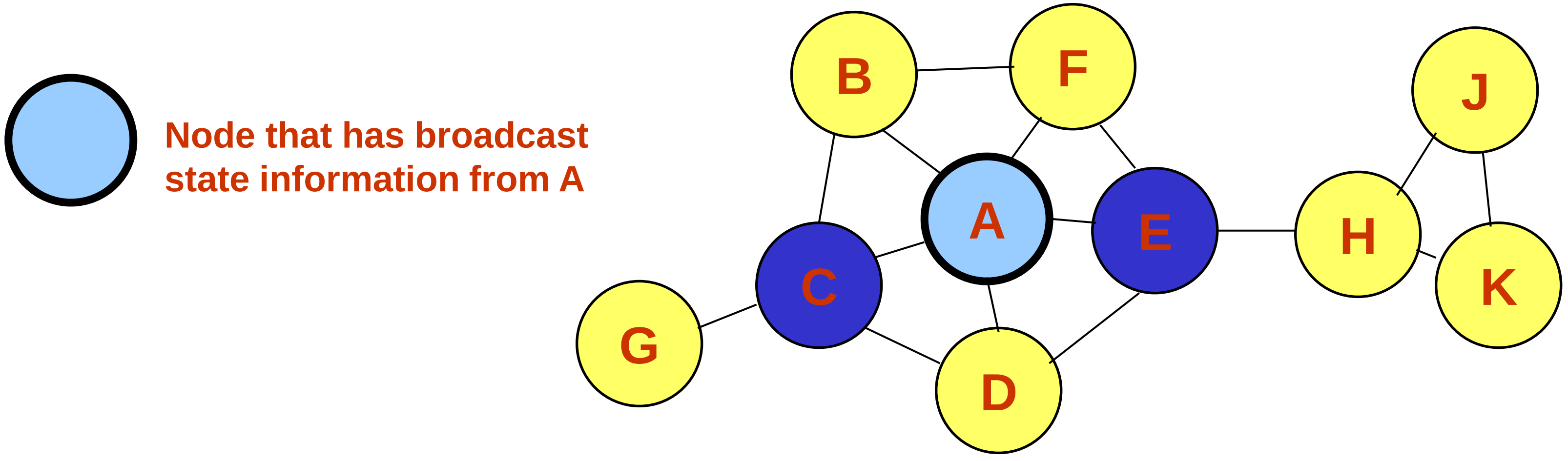


- **OLSR**



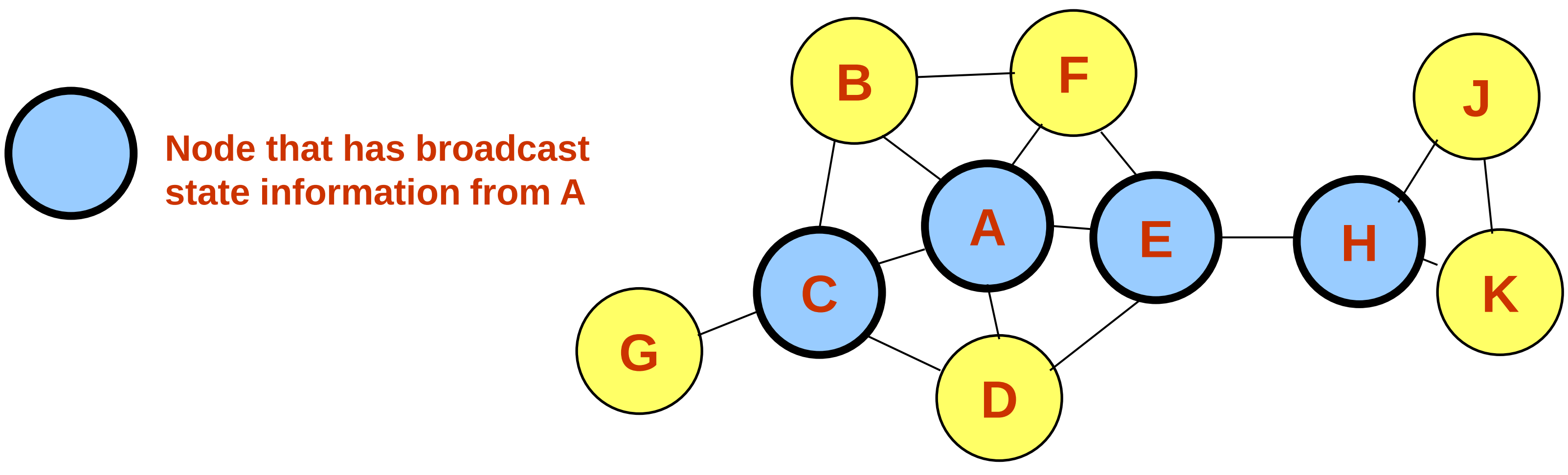
Link state forwarding

- 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

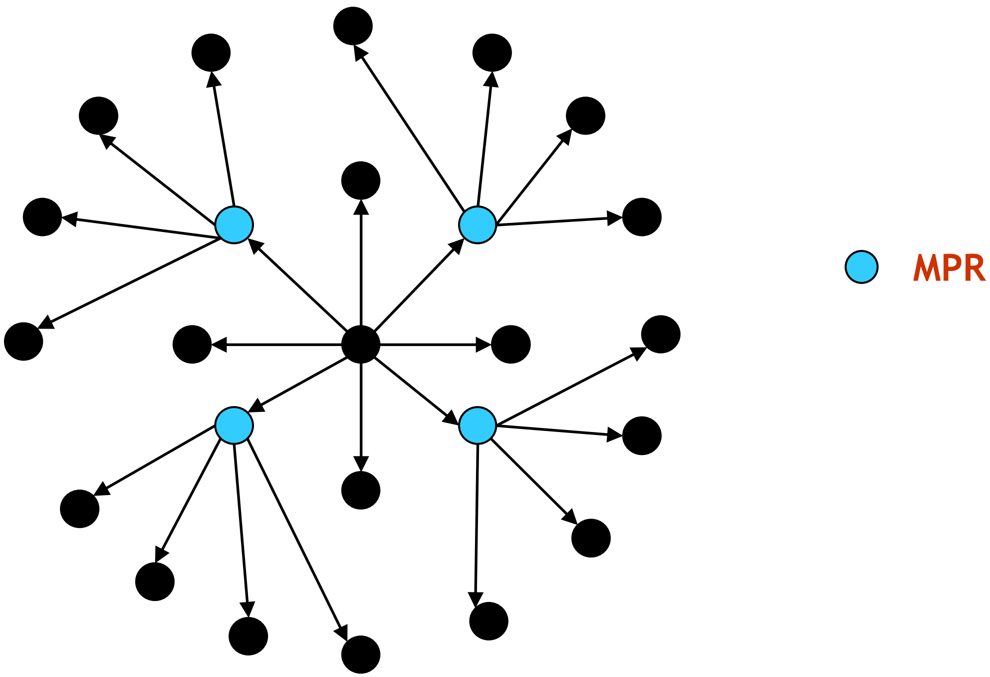


Link state forwarding

- 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



OLSR: Example



4 retransmission to diffuse a message up to 2 hops

MPR sets and MPR selectors

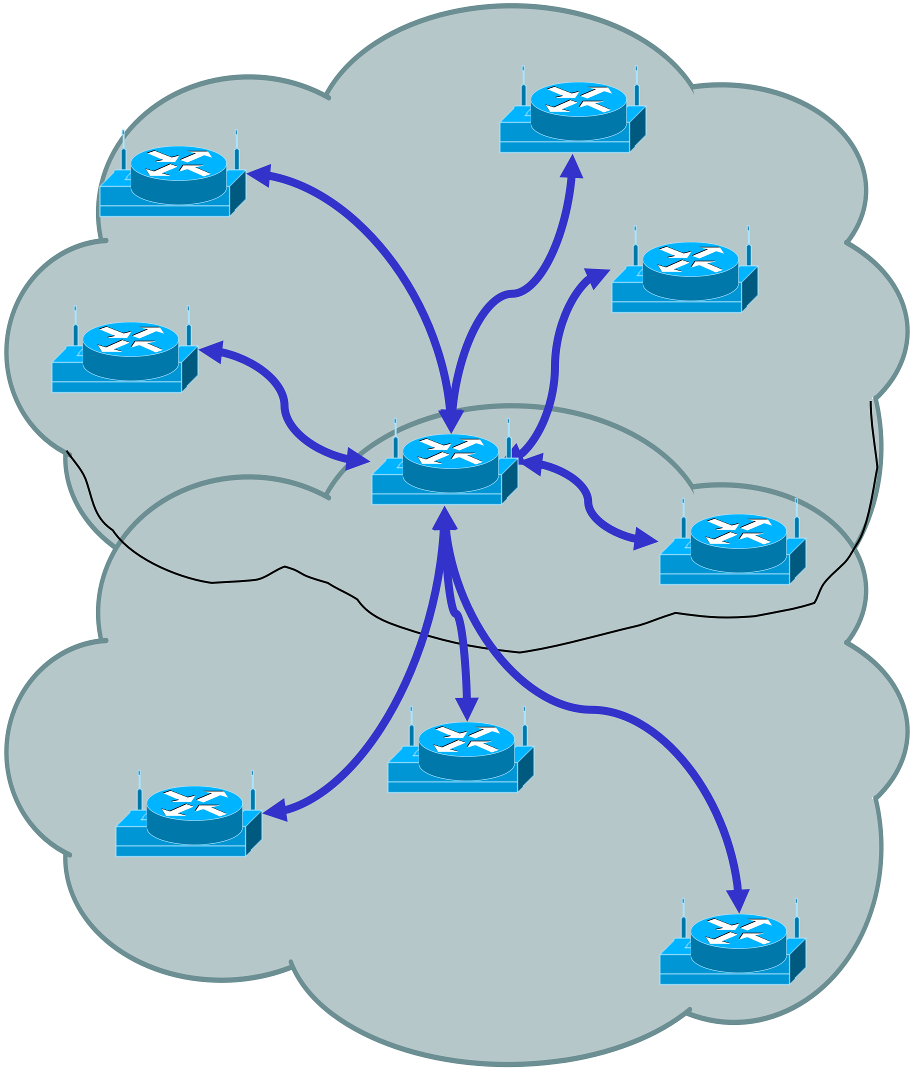
- MPR sets
  - Set of nodes that are multipoint relays
  - Each node selects an MPRset to process and forward every link state packet originated by it
  - Other nodes process the link state packets but do not forward them
- MPR selectors
  - Set of neighbors that have selected the node as multipoint relay
  - MPR forwards packets received from MPR selectors
- Members of MPR sets and MPR selectors change over time – efficient selection mechanisms

Selection of MPR

- Select as MPR every node in the node’s two-hop neighborhood that has a bidirectional link with the node
- Select as MPR, the nodes covering ”isolated” nodes, i.e. for which there is a neighbor which has another node as single parent
- Select as MPR the node which covers the maximal number of nodes

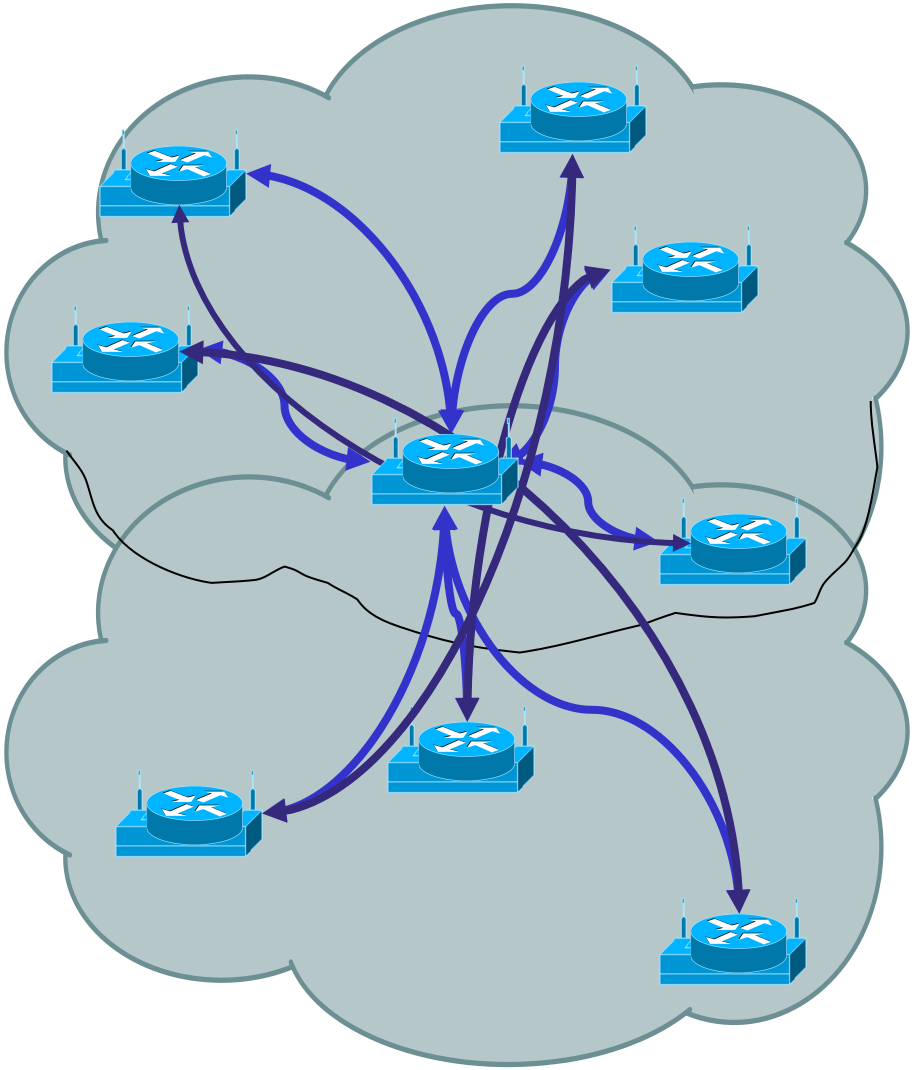
Neighbor relationships

- Each device emits a periodic “Hello”
  - –Advertise itself to its neighbors
  - –Determine who else is there
  - –Select some systems to act as MultiPoint Relays



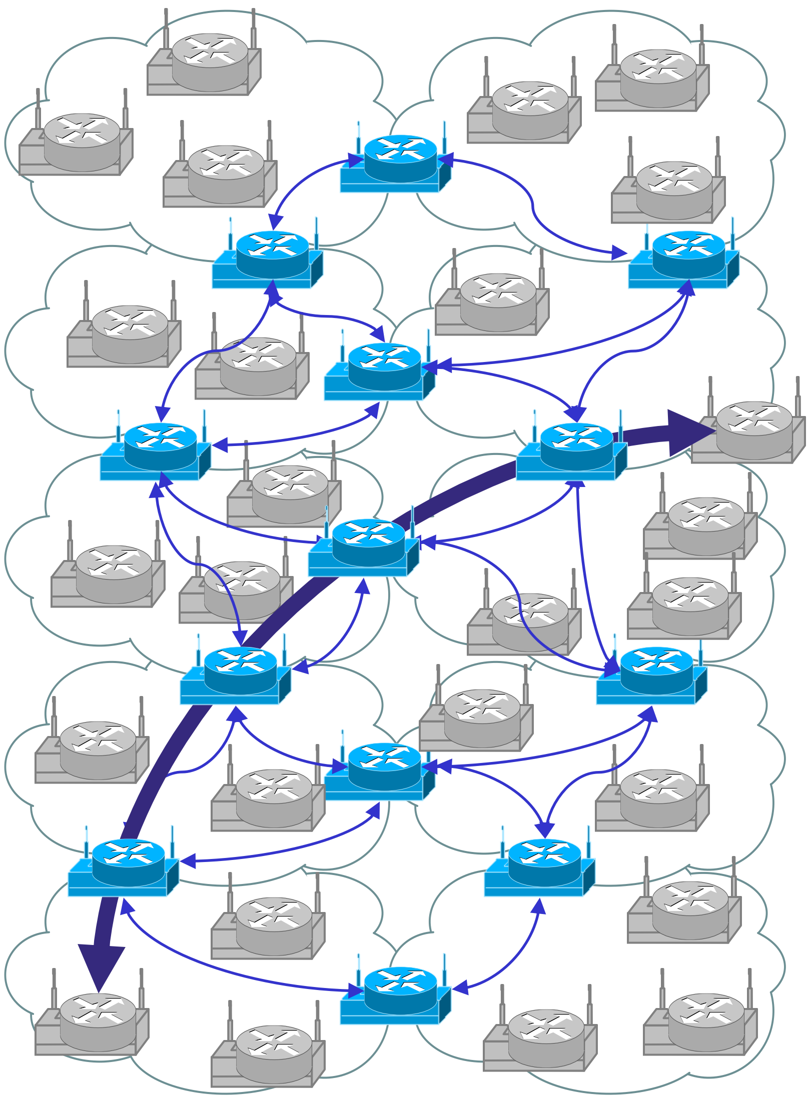
MultiPoint Relays

- Passes Topology Information (topology control messages)
  - –Acts as router between hosts
  - –Minimizes information retransmission
  - –Forms a routing backbone



Structure of an OLSR Network

- MPRs form routing backbone
  - –Other nodes act as “hosts”
- As devices move
  - –Topological relationships change
  - –Routes change
  - –Backbone shape and composition changes



Location-based routing protocols

LAR – Location Aided Routing

The main problems of previous mechanisms

- Nodes location changes rapidly
- No information regarding
  - Current location
  - Speed
  - Direction
- Knowing the location
  - Minimizes the search zone
  - No need to flood the network
- Knowing the speed and/or direction
  - More minimization of the search zone
  - Increases the probability to find the necessary node

Location-Aided Routing (LAR)

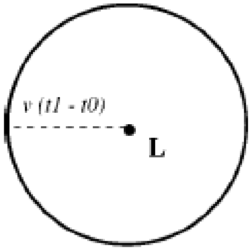
- Each node knows its location in every moment
- Using location information for route discovery
- Routing is done using the last known location + an assumption
- Route discovery is initiated when
  - S does not know a route to D
  - Previous route from S to D is broken
- Assumptions
  - Location knowledge
  - No error
  - 2D movement
  - Full cooperation

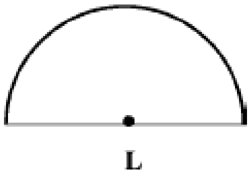
Location information

- Alignment of satellites and ground stations
- Global Positioning System (GPS) - USA
- Global Navigation Satellite System (GLONASS) - Russia
- Galileo – EU
- 3D positioning
- Accuracy 3-100 meters
- Can provide further information
  - Velocity
  - Time

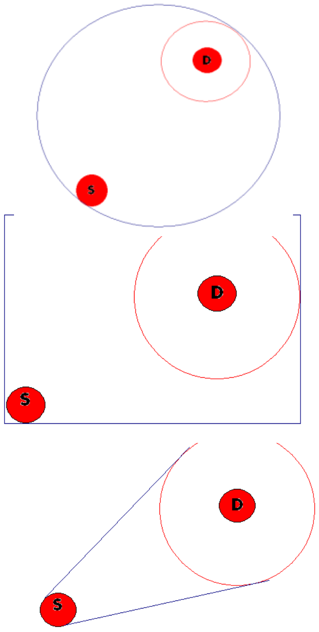
LAR - Definitions

- Expected Zone (EZ)
  - S knows the location L of D in t_0
  - Current time t_1
  - The location of D in t_1 is the expected zone
  - Assume Max/Avg speed v



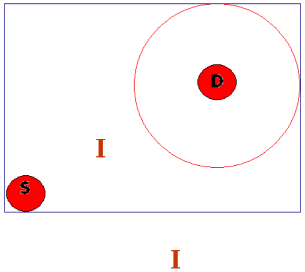


- Request Zone (RZ)
  - Flood with a modification
  - Node S defines a request zone for the route request
  - How to determine the size and shape of the request zone?
  - Several considerations
    - If the destination’s EZ does not include the source node, other regions must be included in the RZ
    - Not always a route will be found using a certain RZ



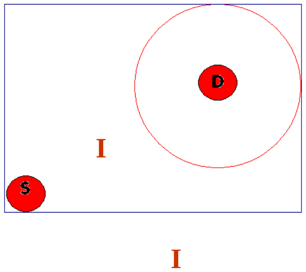
LAR – scheme 1 (Algorithm)

- Node I receives RREQ
  - – Location of I – (X_i, Y_i)
  - – If I is within the rectangular, I forwards the RREQ to its neighbors
  - – Else I discards the RREQ
- Node D receives the RREQ
  - – Replies RREP
  - – Adds its current location



LAR – scheme 1 (some issues)

- The rectangular size is proportional to
  - Average speed (v)
  - Time elapsed ($t_{1-}t_0$)



  Therefore
  - Low speed ⇒ small v in the same ($t_{1-}t_0$) ⇒ smaller RZ
  - High speed ⇒ large v in the same ($t_{1-}t_0$) ⇒ larger RZ
- Improvements
  - D can add its speed/avg. speed in the RREP, this can help other nodes in future route discoveries
  - D can piggyback its location in other packets

BATMAN

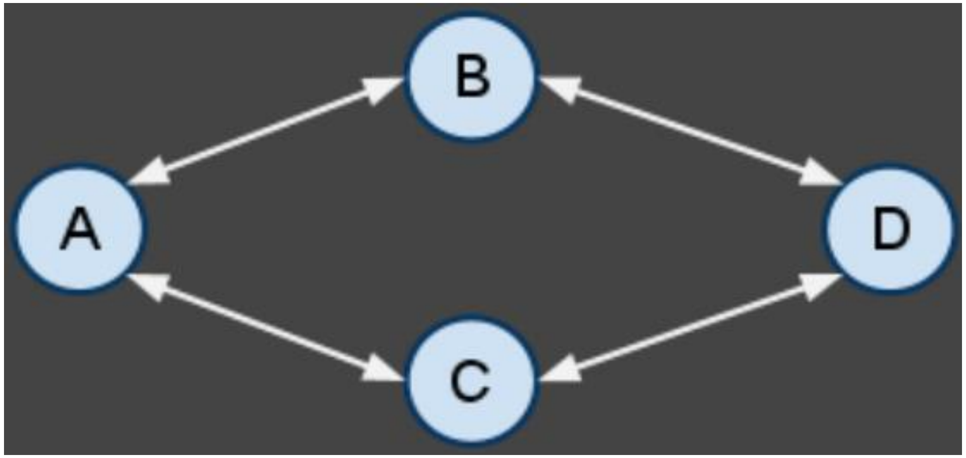
A Better Approach to Mobile Ad hoc Networking

https://www.researchgate.net/publication/320172464_Better_approach_to_mobile_ad-hoc_networking_BATMAN

https://www.open-mesh.org/projects/batman-adv/wiki/BATMAN_IV

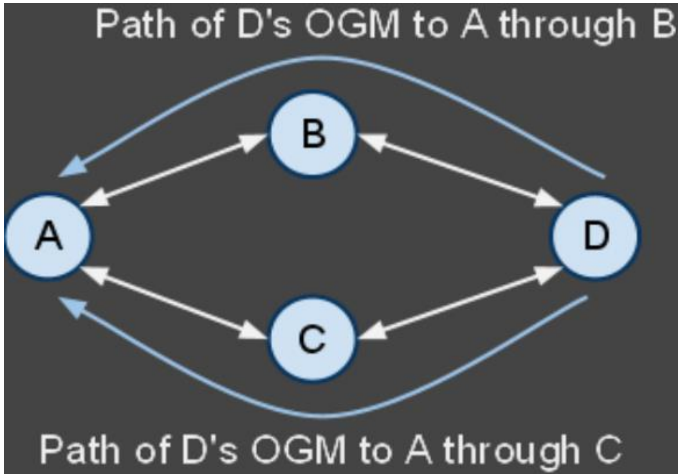
Batman

- Traditionally, nodes exchange control packets that contain information about link state (current link utilization, bandwidth, etc).
  - Nodes determine best paths based on control packets.
  - Every node must have near exhaustive information about the entire network
- BATMAN takes a different approach:
  - The presence or absence of control packets is used to indicate link (and path) quality.



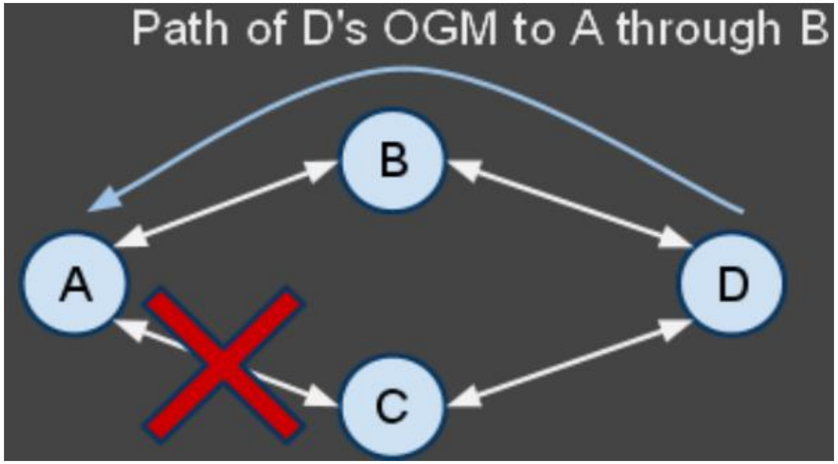
Batman Operation

- Each node has a set of direct-link neighbors
  - In the figure, node A has neighbors B and C. These are the nodes through which A sends and receives all its packets.
- Each node in the network sends an Originator Message (OGM) periodically, in order to inform all other nodes of its presence
  - OGMs include a sequence number
- If all shown links are perfect, Node A will receive node D's OGM through both of its neighbors B and C.
  - If all of D's OGMs arrive through both B and C, then when A needs to send something to D, it can use either B or C as the next hop towards the destination node D.



Batman Operation

- If the link between nodes A and C goes down
  - Node D's OGM will only arrive to A through node B.
  - Node A therefore considers node B as the best next hop neighbor for all packets destined for node D.
  - Further, Node C's OGMs will also only reach node A through node B. Node B is the best next hop for data destined for Node C.



Batman: sliding window

- If some but not all OGMs arrive through a link
  - Sliding window
- A sliding window indicates which of the last WINDOW_SIZE (in the example, 8) sequence numbers have been received
  - Uses the sequence numbers received through OGMs

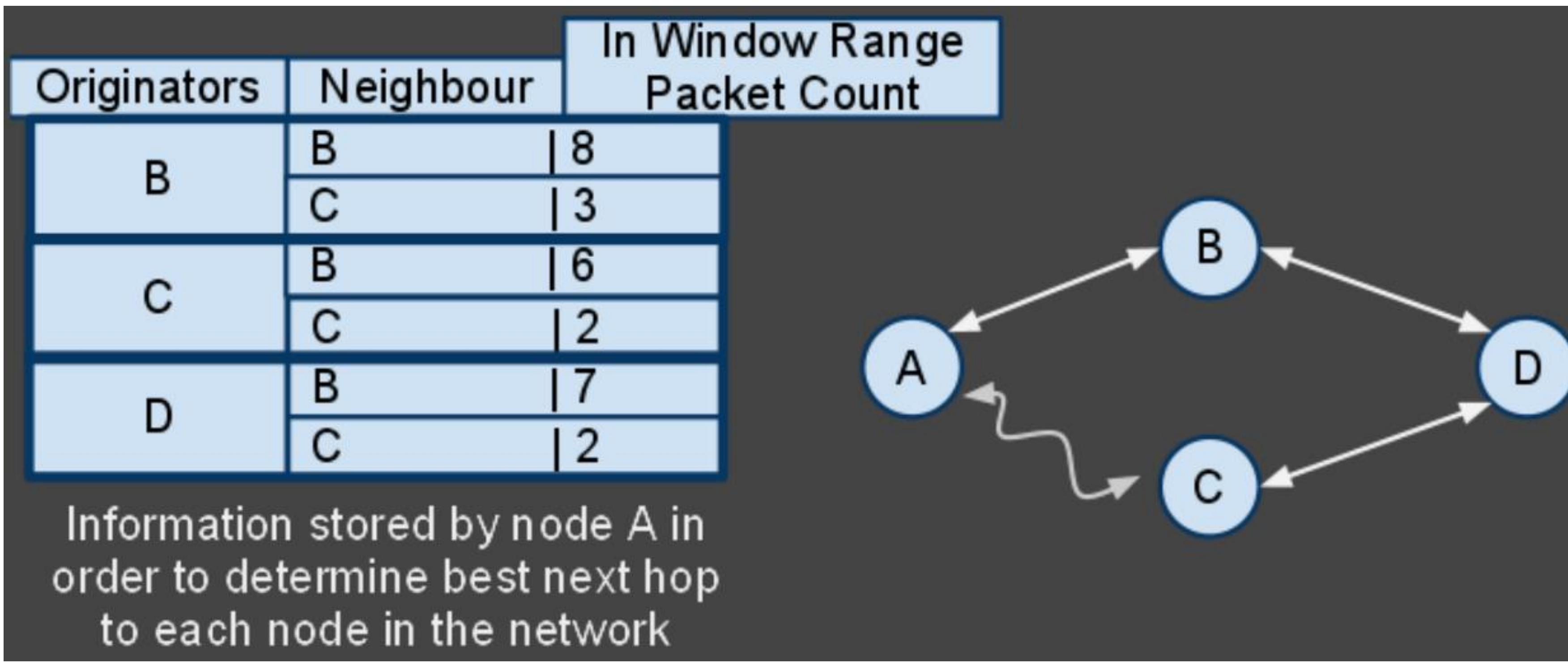
	Out of Range				In Window Range								Out of Range			
Seq. Numbers:	...	5	6	7	8	9	10	11	12	13	14	15	16	17	18	...
Arrived:	...	-	-	-	1	1	1	0	1	0	1	1	-	-	-	...

Sequence numbers and sliding window

- When an out of range sequence number is received, in this case seq# 17, the window shifts up.
  - From 6 sequence numbers in-range to only 5.

Routing table

- All nodes have a sliding window for each originator (other node) in the network for each neighbor

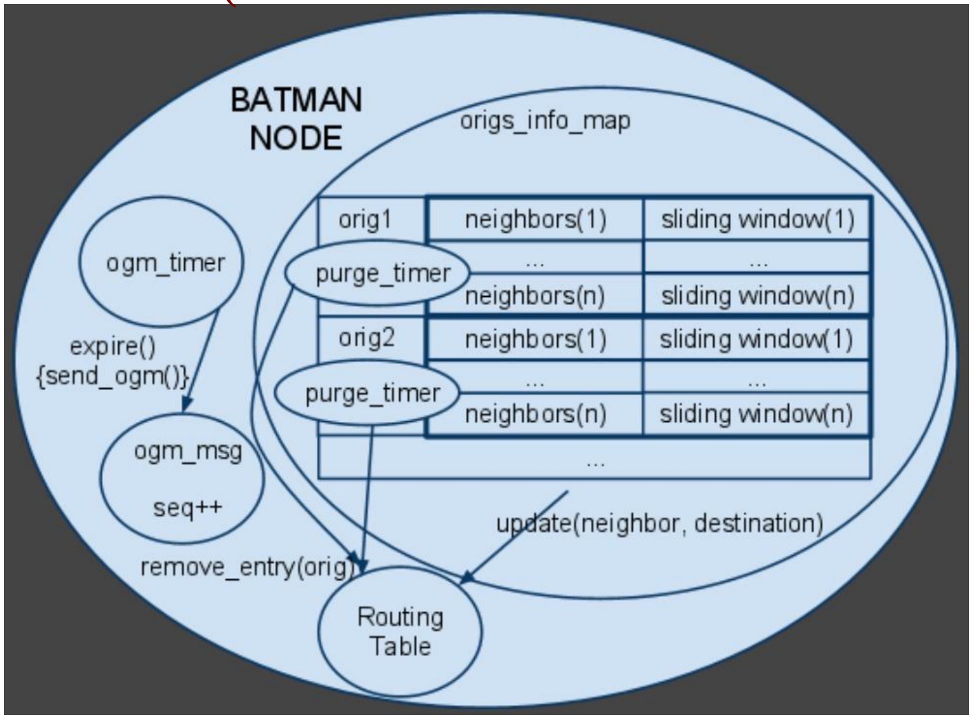


Originators	Neighbour	In Window Range Packet Count
B	B	8
	C	3
C	B	6
	C	2
D	B	7
	C	2

Information stored by node A in order to determine best next hop to each node in the network

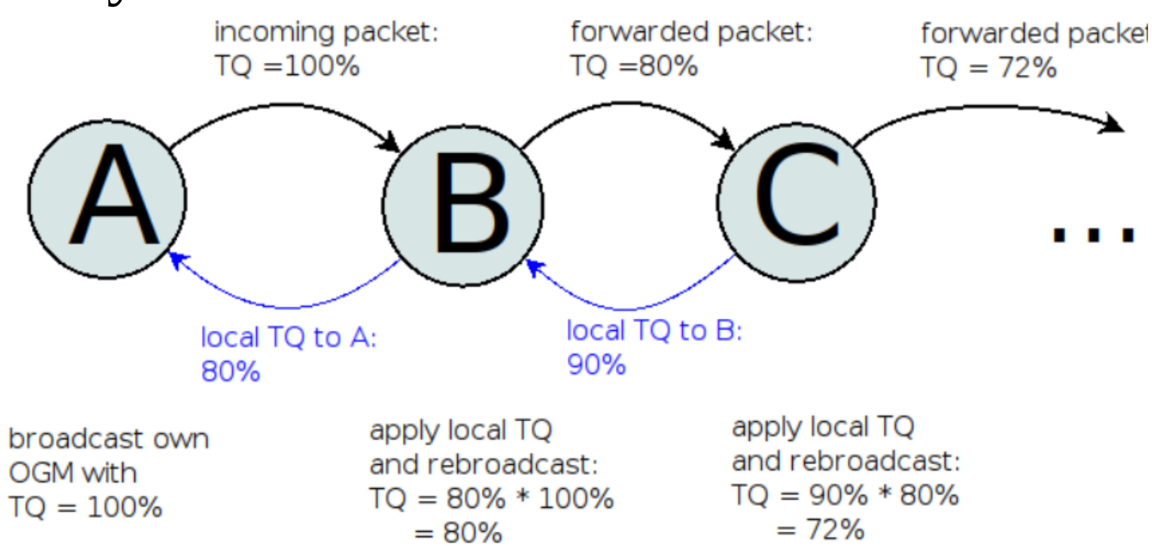
Batman Operation

- BATMAN receives information about link (and path) quality through the presence or absence of control packets.
  - Collective intelligence - retransmission of an OGM implies it arrived successfully through a best-link neighbour
  - No node needs to have exhaustive knowledge of the network, just the next hops to the destination (information about the destinations comes from OGMs)



Transmission Quality (Batman v.4)

- To add the local link quality in the TQ value, the following calculation is performed:
- TQ = TQ_{incoming} * TQ_{local}
- Example: Node A broadcasts the packet with TQ max. Node B receives it, applies the TQ calculation and rebroadcasts it. When node C gets the packet it knows about the transmit quality towards node A.



Transmission Quality (Batman v.5)

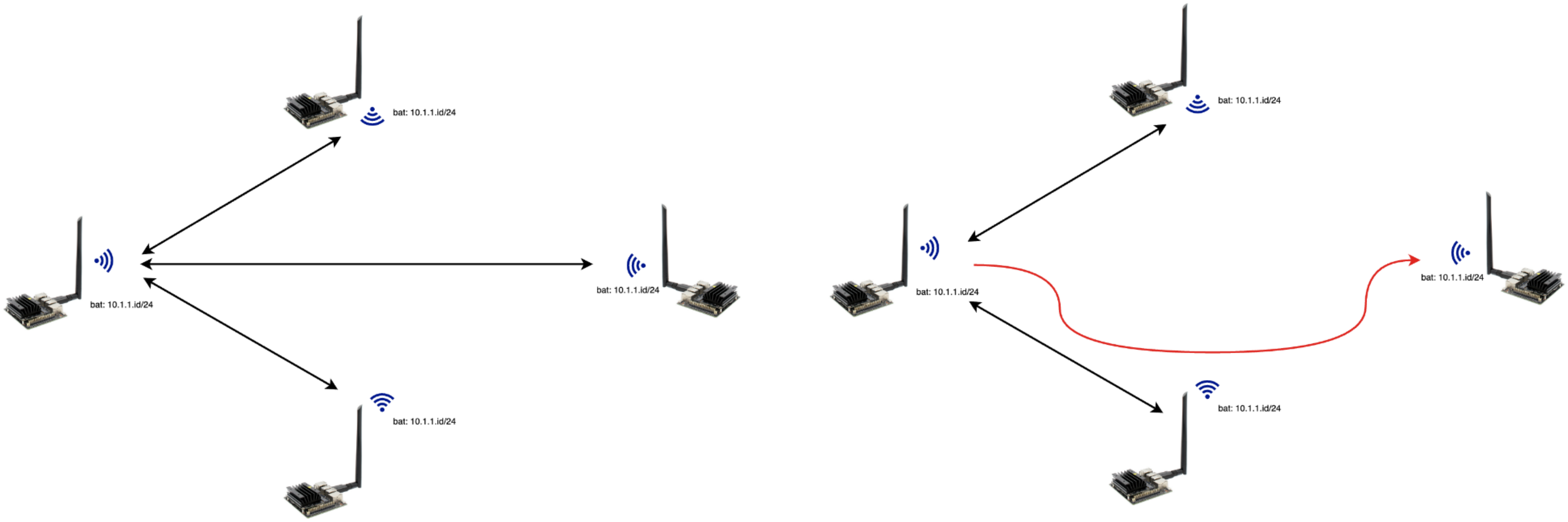
- Packet loss based metric is not adequate
  - Increasing number devices & link types with little to no packet loss

- Packet throughput as mesh-wide metric.
- Determine the throughput automatically:
  - wireless: Modern WiFi drivers export the estimated throughput per WiFi neighbor. This value is retrieved on a periodic basis and averaged before propagated in the mesh.
  - wired: Most Ethernet capable devices export their theoretical throughput and duplex capabilities via the ethtool API.
  - throughput meter (upcoming): If the throughput can not be queried via some API and is not manually configured, BATMAN V will run a periodic throughput test with its built-in throughput test protocol.

- Throughput estimation relies on the WiFi driver being able to estimate the throughput
  - With payload traffic to be sent to each neighbor for the estimation to be accurate
  - On idle links BATMAN V will initiate payload traffic from time to time to feed the WiFi driver's estimation logic.

- The path throughput between node A and node B is computed as the minimum between the throughput value of all given links on the path between node A and node B

Example


bat: 10.1.1.id/24
bat: 10.1.1.id/24
bat: 10.1.1.id/24
bat: 10.1.1.id/24
bat: 10.1.1.id/24
bat: 10.1.1.id/24
bat: 10.1.1.id/24
bat: 10.1.1.id/24

Comparison

- AODV pros and cons
  - Low overhead
  - Slow discovery and recovery
- OLSR pros and cons
  - Medium overhead
  - Fast discovery and recovery
  - MPRs automation
- LAR pros and cons
  - Medium overhead
  - How to discover the location of destination?
- Batman pros and cons
  - Medium-high overhead
  - Implicit quality information
  - Fast discovery and recovery