COMP9332
Network Routing & Switching

MANET Routing I (AODV)

www.cse.unsw.edu.au/~cs9332

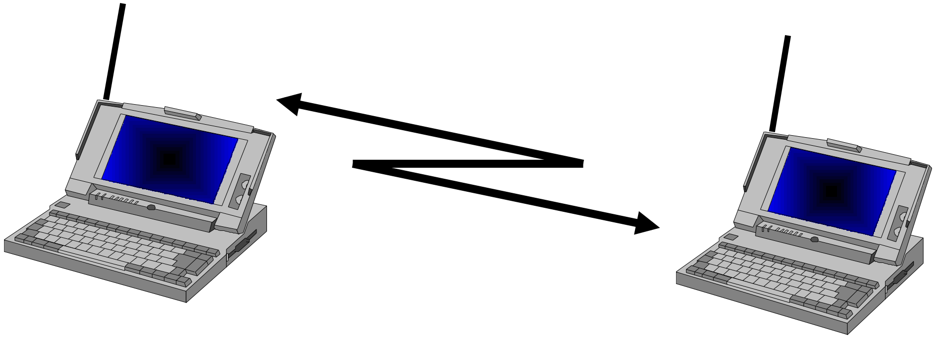
This lecture

- Consider a different kind of networks called MANET (mobile ad hoc networks)

- Ad hoc networks
  - How are they different from traditional networks?
  - Why and what routing philosophy is needed?
  - Example routing algorithms and protocols

What are ad hoc networks?

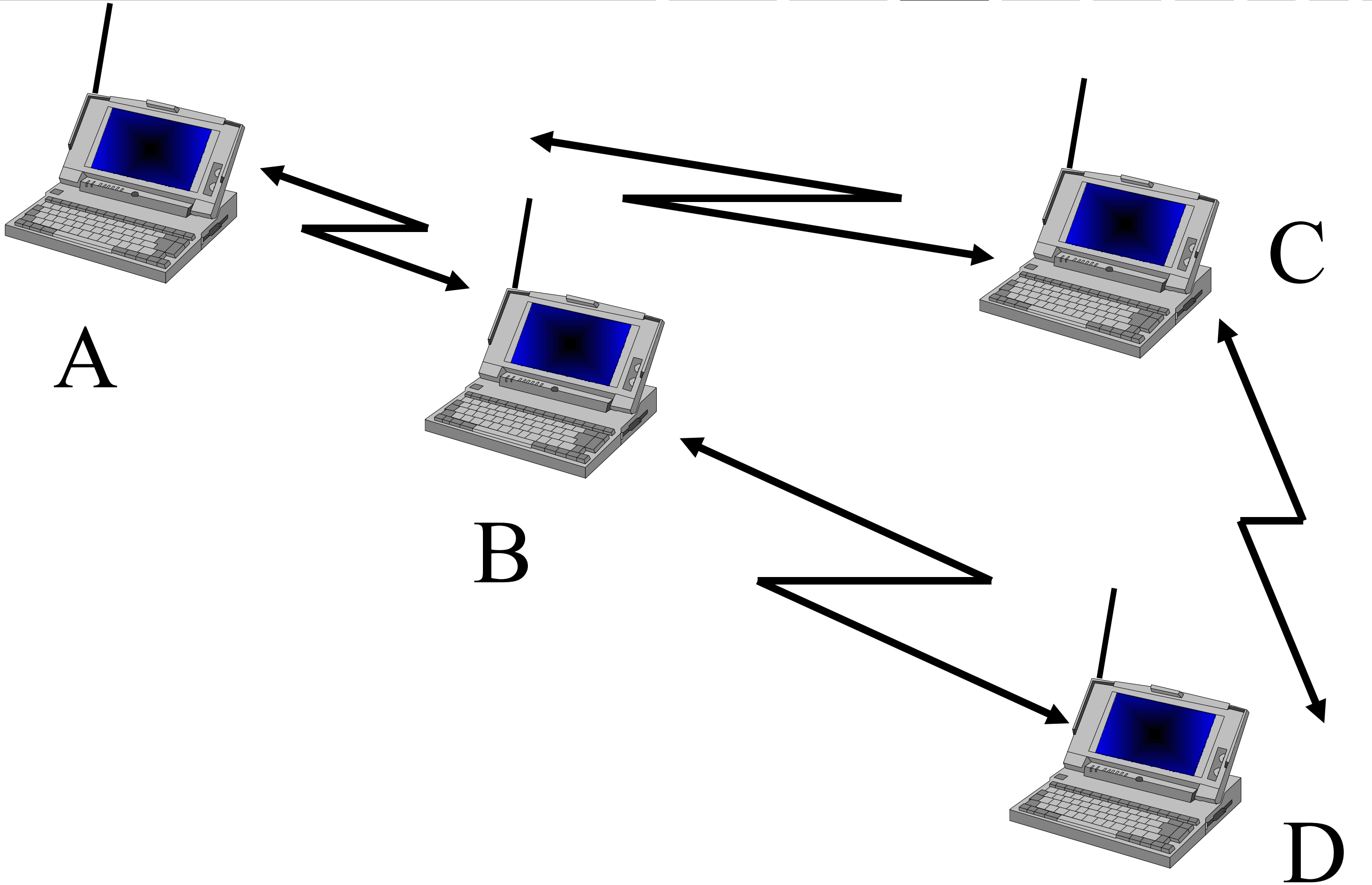
- Networks without fixed infrastructure and nodes communicate over wireless links
  - Example of fixed infrastructure: access points, routers, switches, Ethernet
  - The simplest ad hoc network consist of 2 devices communicating over a wireless link
    - » Devices can be laptops, bluetooth phone, PDA etc



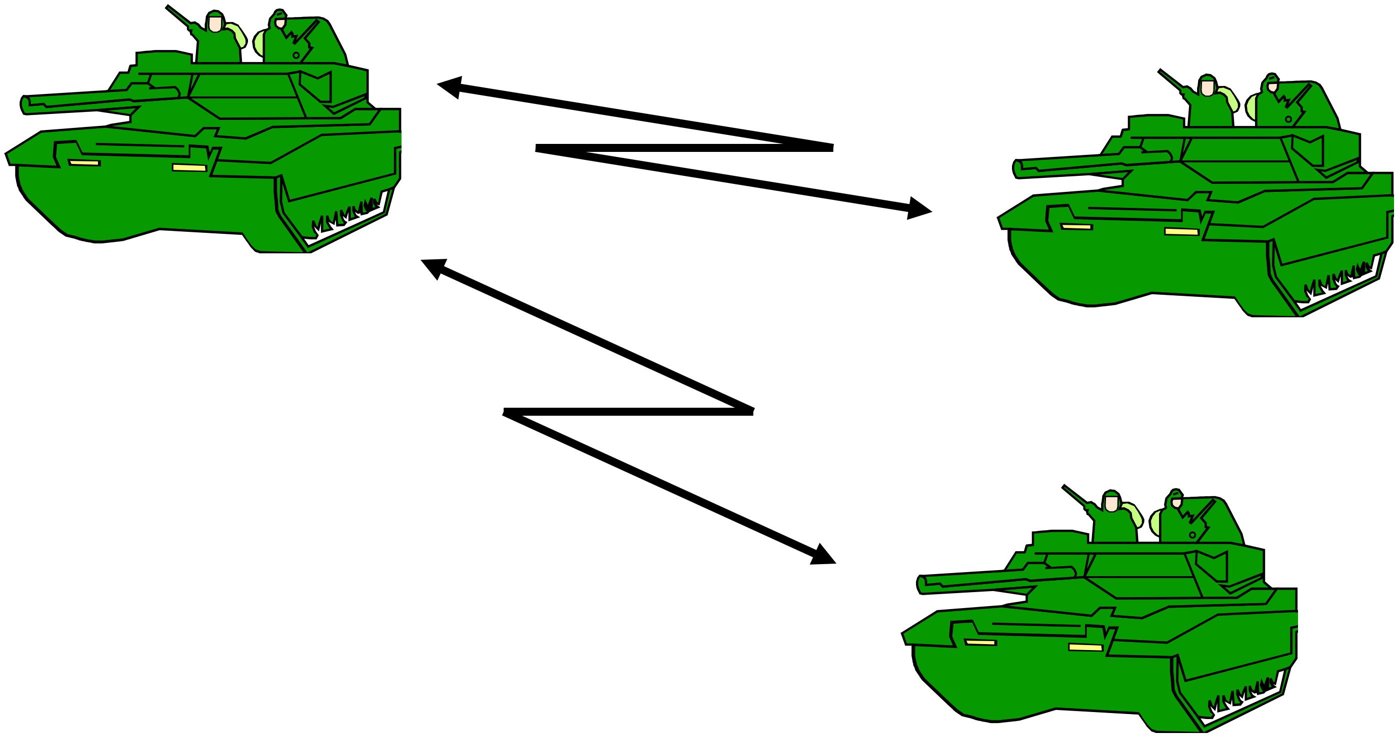
More complex ad hoc networks

- Many hosts/devices communicating with each other wirelessly [illustration - next slide]
- The key feature is: no infrastructure
- Hosts can be stationary or mobile
- If hosts are mobile, it is known as Mobile ad hoc network (MANET)

Ad hoc network - an illustration

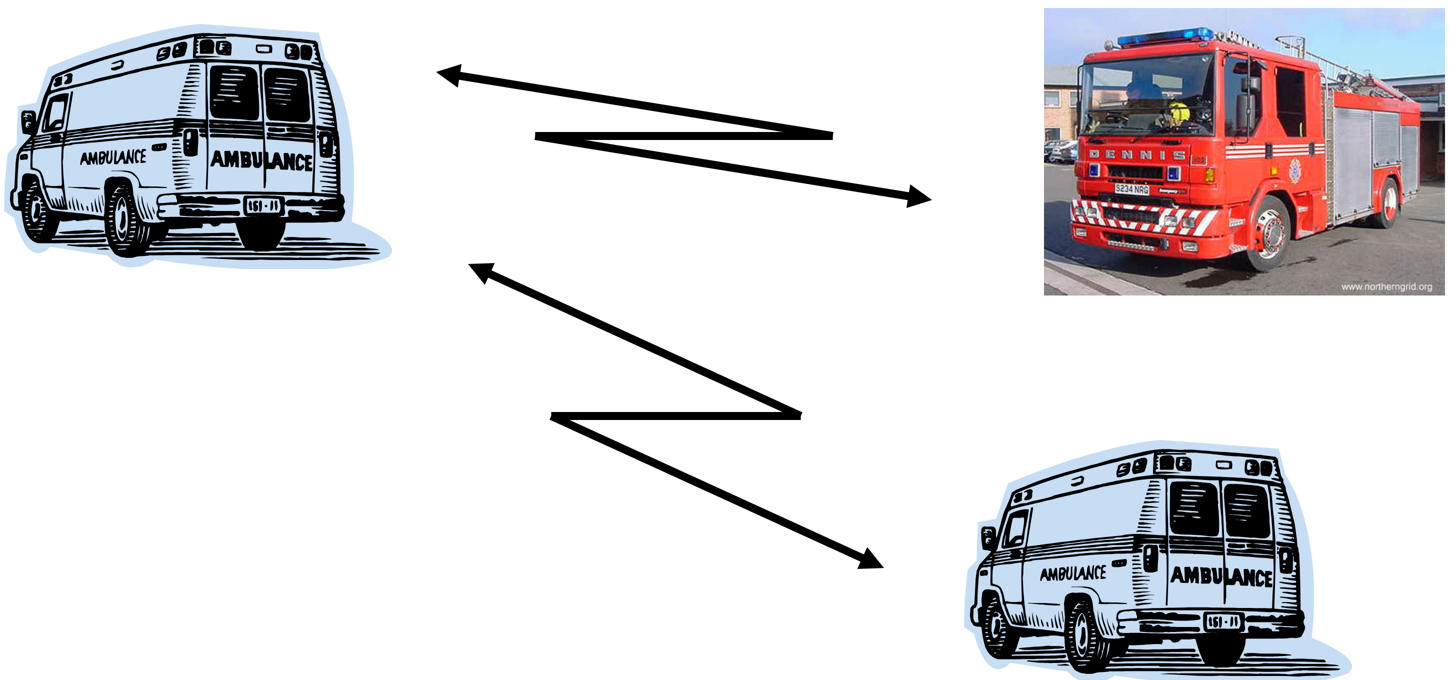


MANET application (1)



Battle field communications e.g. among tanks and soldiers
Military origin of MANET

MANET application (2)

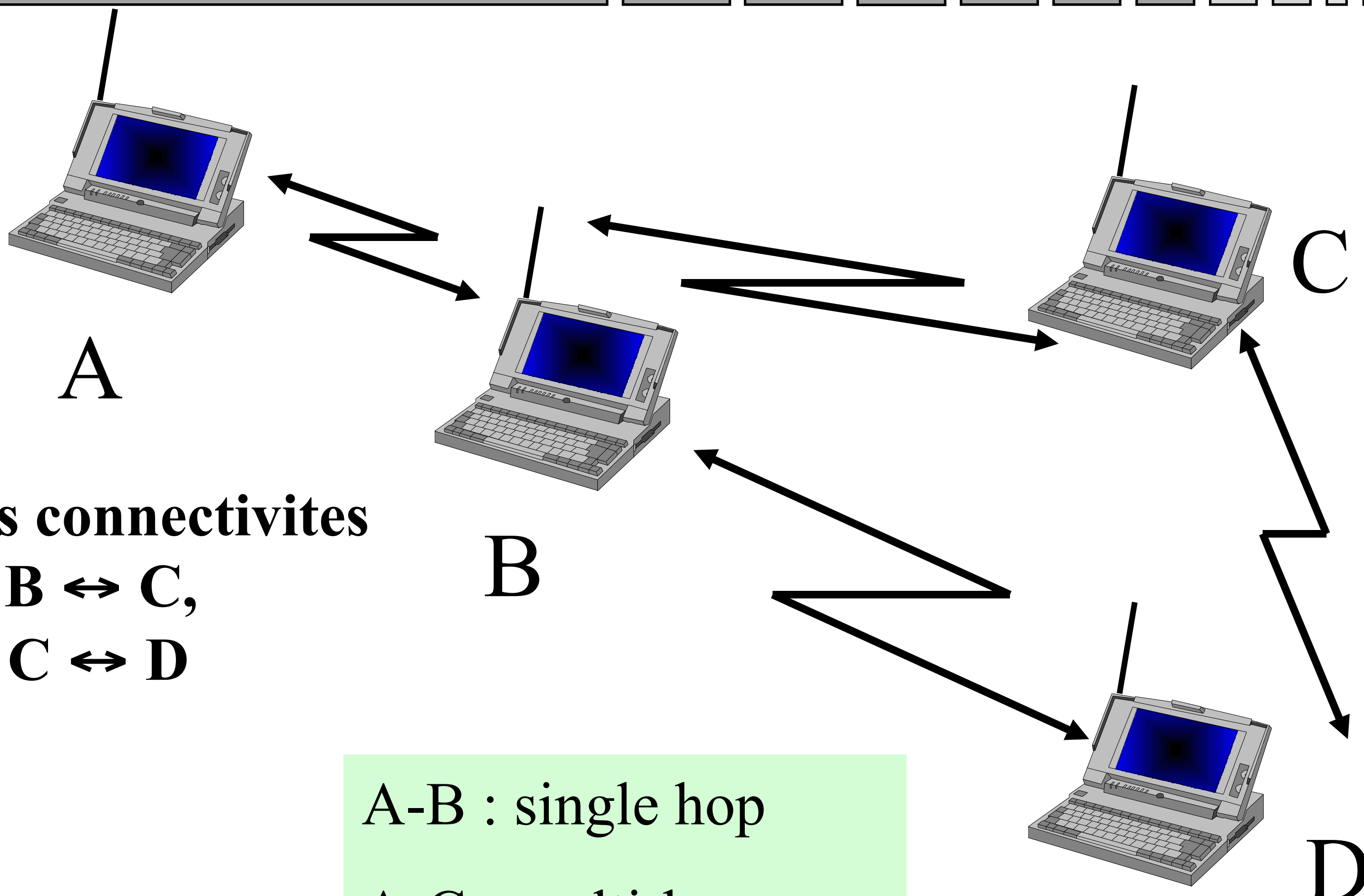


Search and rescue
Mobile devices on personnel and vehicles
Good for remote areas

Features of MANET

- Nodes use wireless communications with limited range
  - Nodes can only directly communicate with nodes within their radio range (multi-hop radio)
- Hosts are also routers (no dedicated routers)
  - Nodes forward the traffic for each other
  - In contrast: Hosts in the Internet do not do routing
- Nodes may change location
- Nodes could be small mobile devices
  - Battery powered (limited energy supply)
  - Limited processing capacity

Multi-hop radio communications

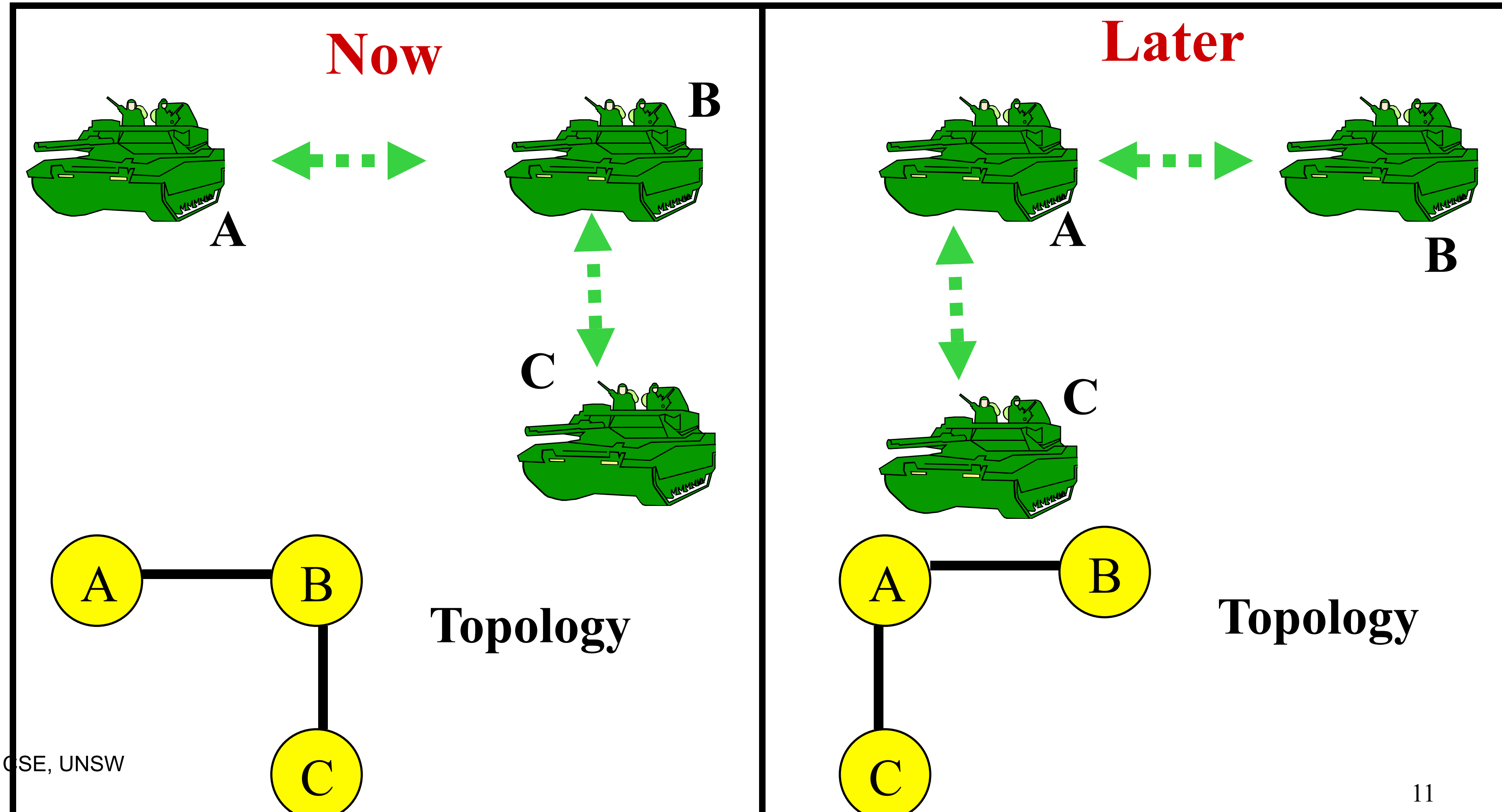


Wireless connectivites
A ↔ B, B ↔ C,
D ↔ B, C ↔ D

A-B : single hop

A-C : multi-hop

Movement of nodes



MANET Routing Philosophy

- Must contain routing protocol overhead (why?)
- Ways to minimize overhead
  - On-demand routing (no maintenance of up-to-date tables)
  - CDS-based routing (cover entire network with a few 'backbone' nodes)
  - Position-based routing (use geographical location for next-hop selection)

Features of on-demand routing

- On-Demand
  - up-to-date routes to all destinations are not maintained
  - no periodic broadcast of tables
  - route discovery process is invoked when a host needs to send a datagram to a destination
  - route remains valid 'till destination remains reachable

RFC3561 - AODV

ad hoc on-demand distance vector routing

Ad-hoc On Demand Distance Vector Routing (AODV)

- The on-demand variant of traditional distance vector routing protocol
- Routes are found on-demand
- When source needs to know the route, it initiates a *route discovery process*
- The source learns the route as the outcome of the route discovery process

AODV - key components

- Messages
  - Route request (RREQ)
  - Route reply (RREP)
  - Route repair
- Components
  - Route discovery
  - Route repair
  - Maintaining neighbour information

AODV - terminology

- Originating node
  - The node which initiates the route discovery process
- Two types of paths
  - Forward path: source to destination
  - Reverse path: destination to source

Route discovery process (1)

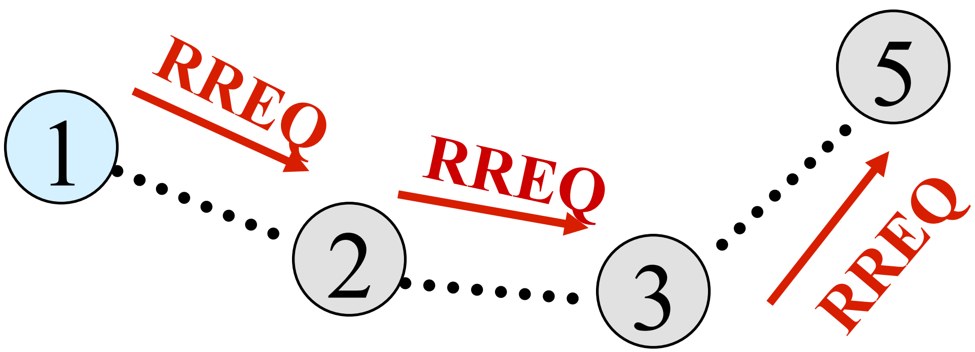
- Originating node S has packets to send to destination node D
  - S doesn't know about a route to D
  - S performs route discovery by broadcasting an RREQ message to its neighbours
    - » The RREQ message is identified by a sequence number
  - Intermediate nodes re-broadcast this message if
    - » It hasn't seen this message before, and
    - » It doesn't know about a route to D

Route discovery process (2)

- If none of the intermediate nodes in the network knows a route to D
  - This is the same as flooding the RREQ message in the network

Maintaining reverse path information (1)

Node 1 = Source node



Node 2 adds the following information to its routing table:
Destination: Node 1
Next Hop: Node 1

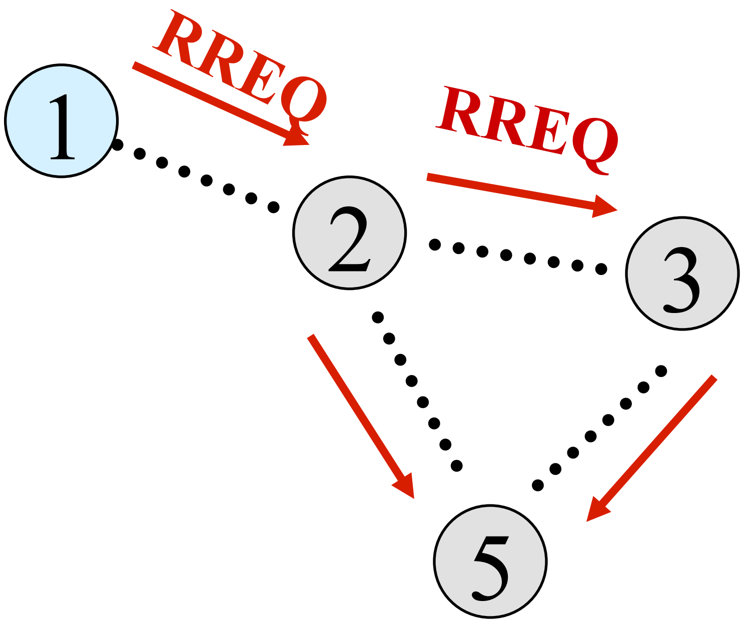
Node 3 adds the following information to its routing table:
Destination: Node 1
Next Hop: Node 2

Node 5 adds the following information to its routing table:
Destination: Node 1
Next Hop: Node 3

When a node receives an RREQ message, it enters reverse path information in its routing table

Maintaining reverse path information (2)

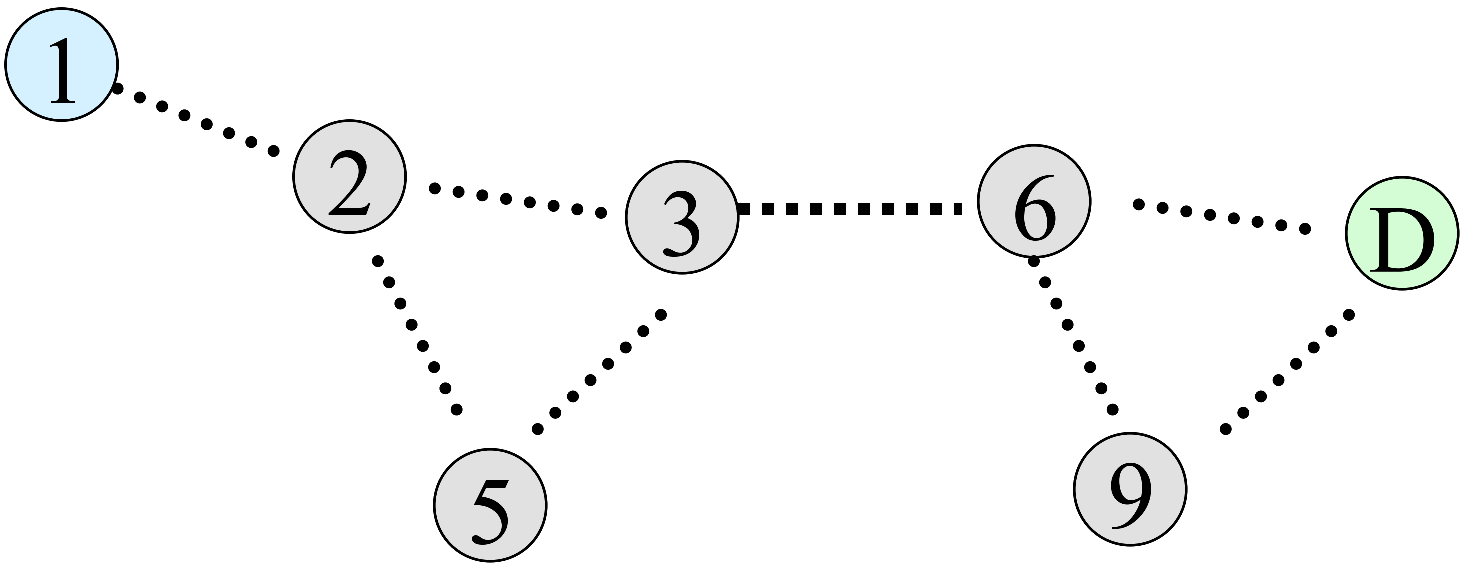
Node 1 = Source node



Node 5 receives an RREQ from node 2 and adds the following information to its routing table:
Destination: Node 1
Next Hop: Node 2

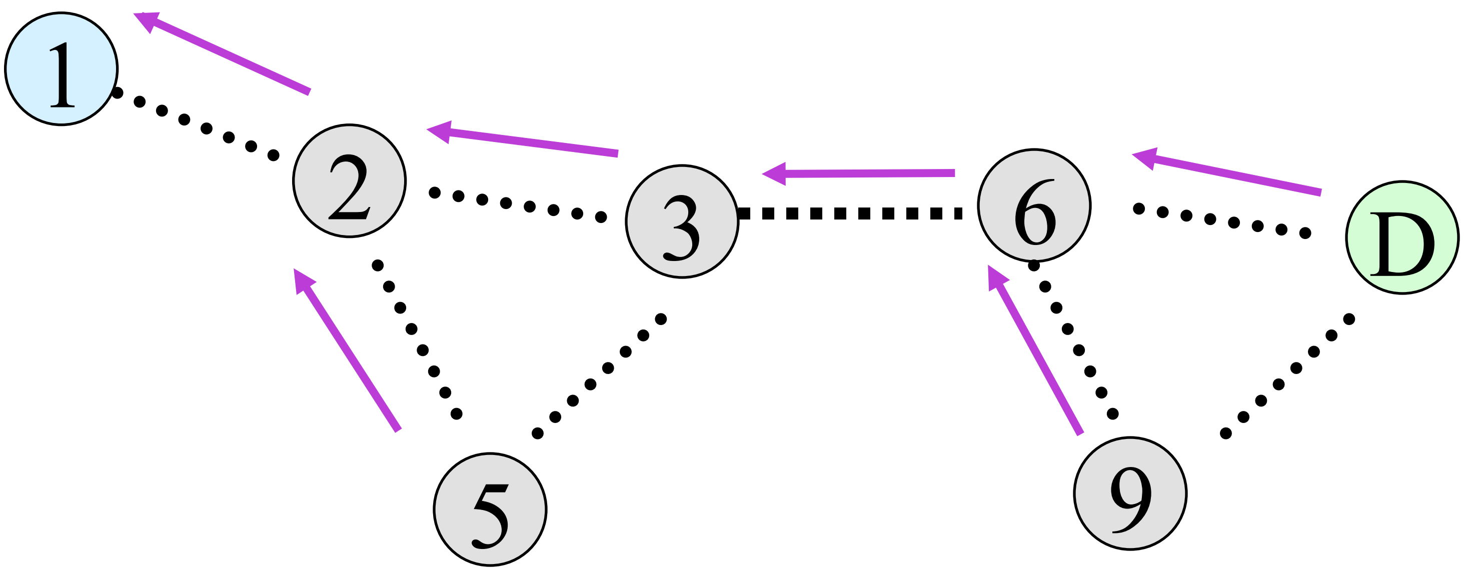
Later, Node 5 receives the same RREQ from node 3, should it modify its routing table?

Exercise



Assuming Node 1 = Source, Node D = Destination.
If no intermediate nodes knows a route to D.
What are the reverse paths that have been established when the RREQ message reaches the destination?

Solution



Assuming node-3 receives RREQ from node-2 before it receives the request from node 5, and node-D receives it from node-6 before from node-9.

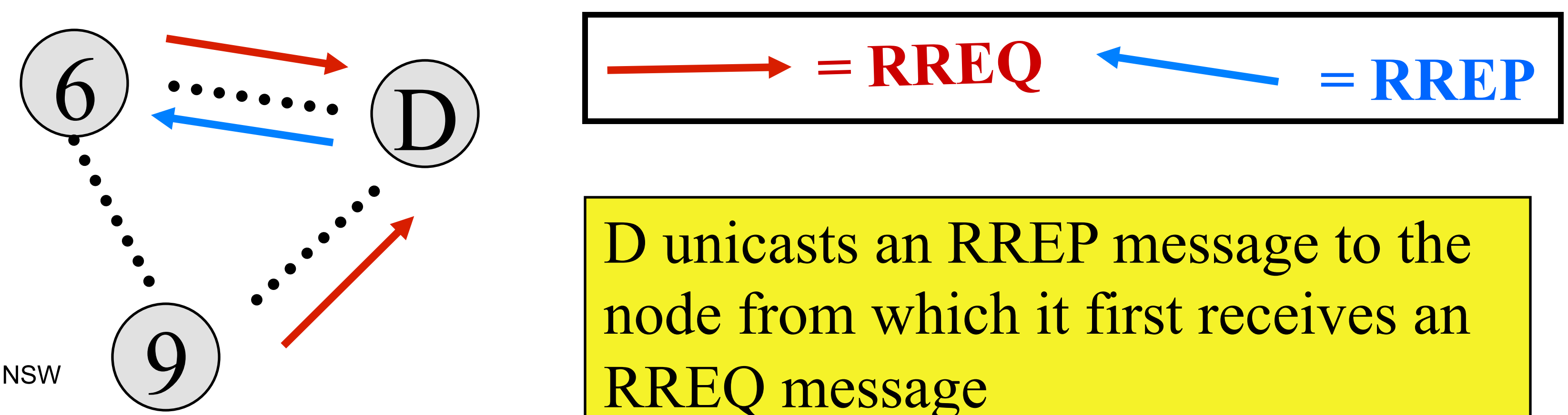
RRREQ packet format

Source address	Reqst ID	Dest address	Hop count

- An RREQ message is uniquely identified by source address and request id
- The message is broadcast by using IP limited broadcast address 255.255.255.255
- Each intermediate node increments the hop count field by 1

Creating the forward path (1)

- If the destination is in the network, the RREQ message will eventually
  - Reaches the destination node, or,
  - A node in the network which knows a route to the destination
- These nodes will reply with an RREP message



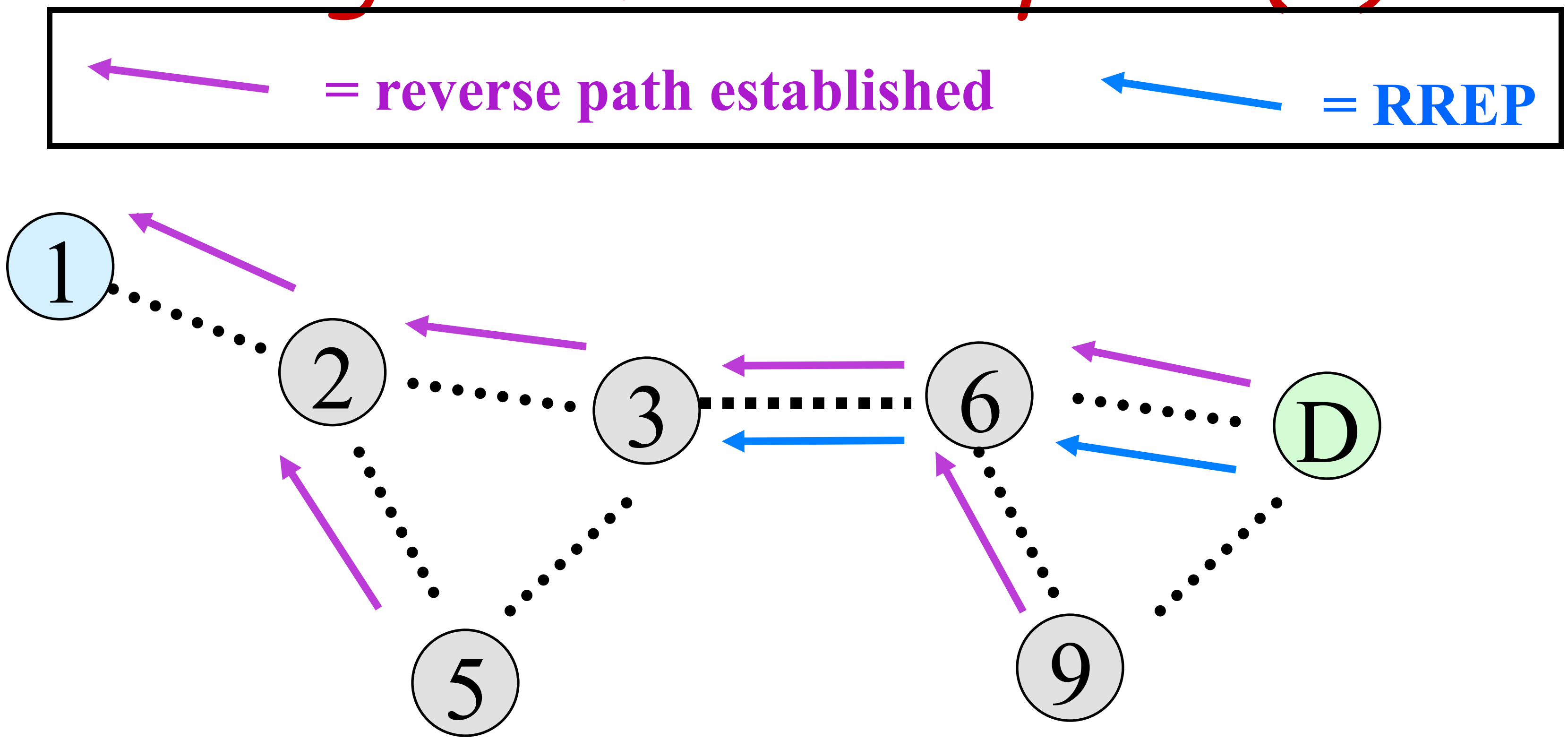
D unicasts an RREP message to the node from which it first receives an RREQ message

RREP message format

Source address	Dest address	Hop count	Life time

- RREP message identifies the
  - Originating node
  - Destination node
- Lifetime - validity time

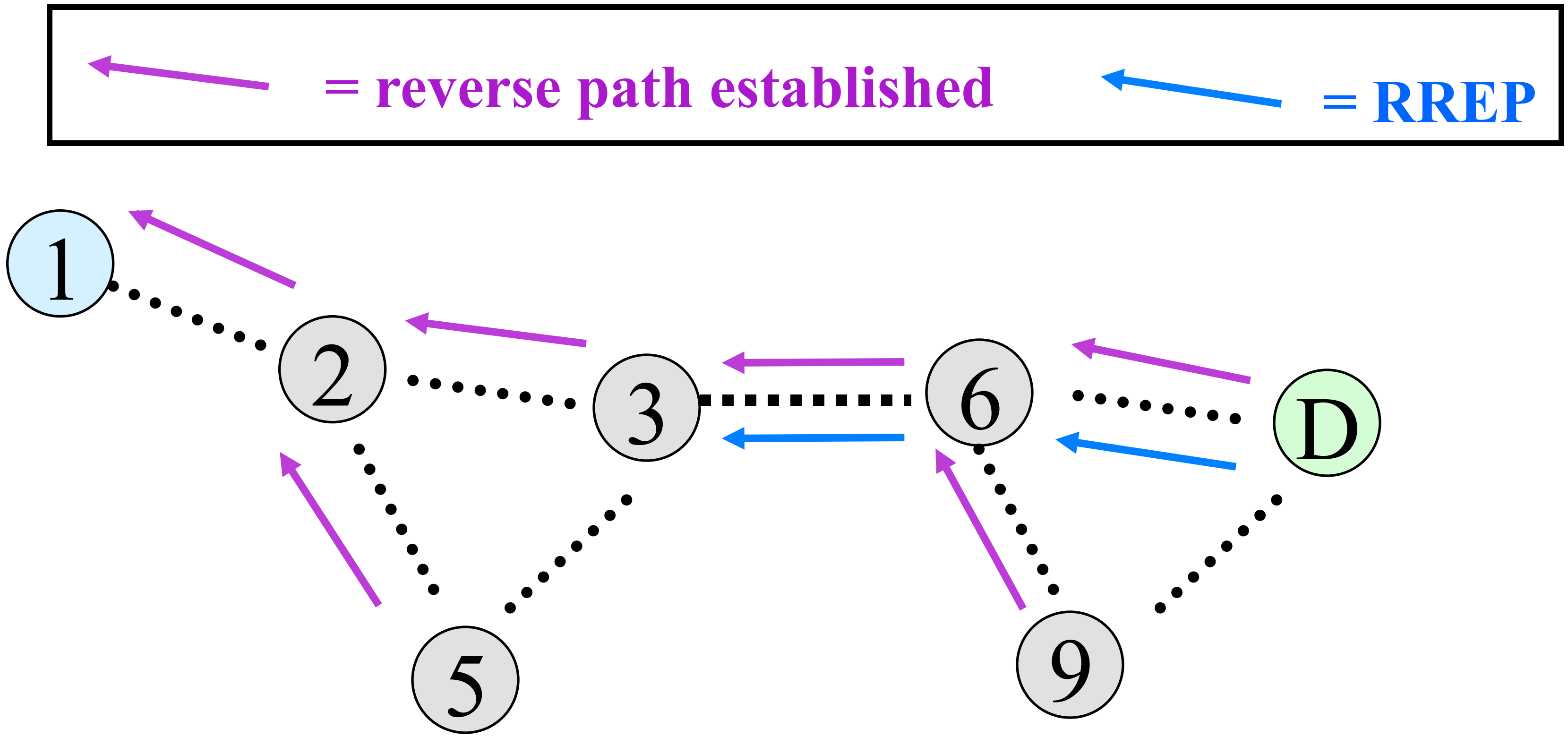
Creating the forward path (2)



When Node 6 receives the RREP message

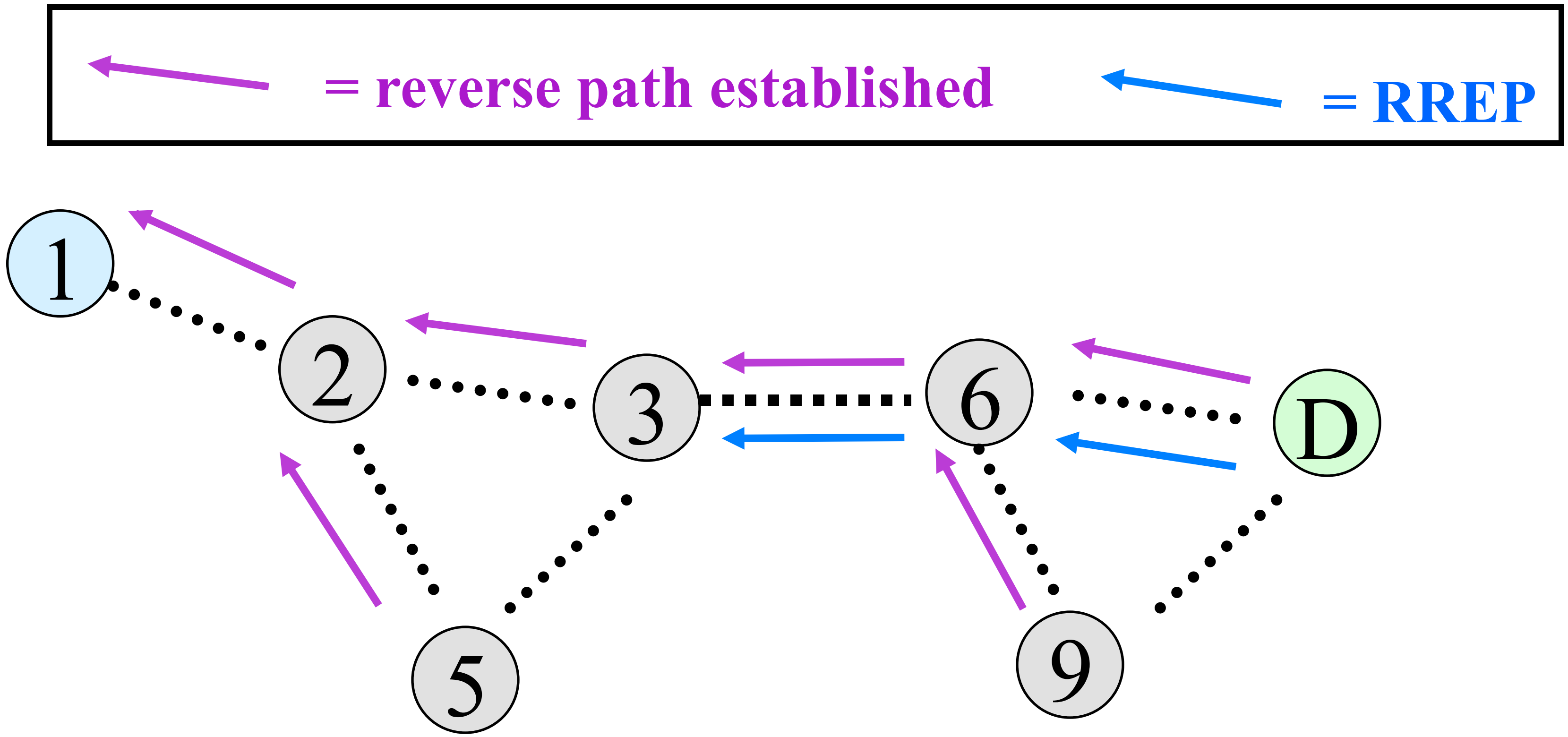
1. It enters the following information in its routing table Destination = D, NextHop = D.
2. It unicasts the RREP message along the reverse path that has already been established.

Exercise:



When Node 3 receives the RREP message from node 6, what information will it enter in its routing table? What actions will it take after that?

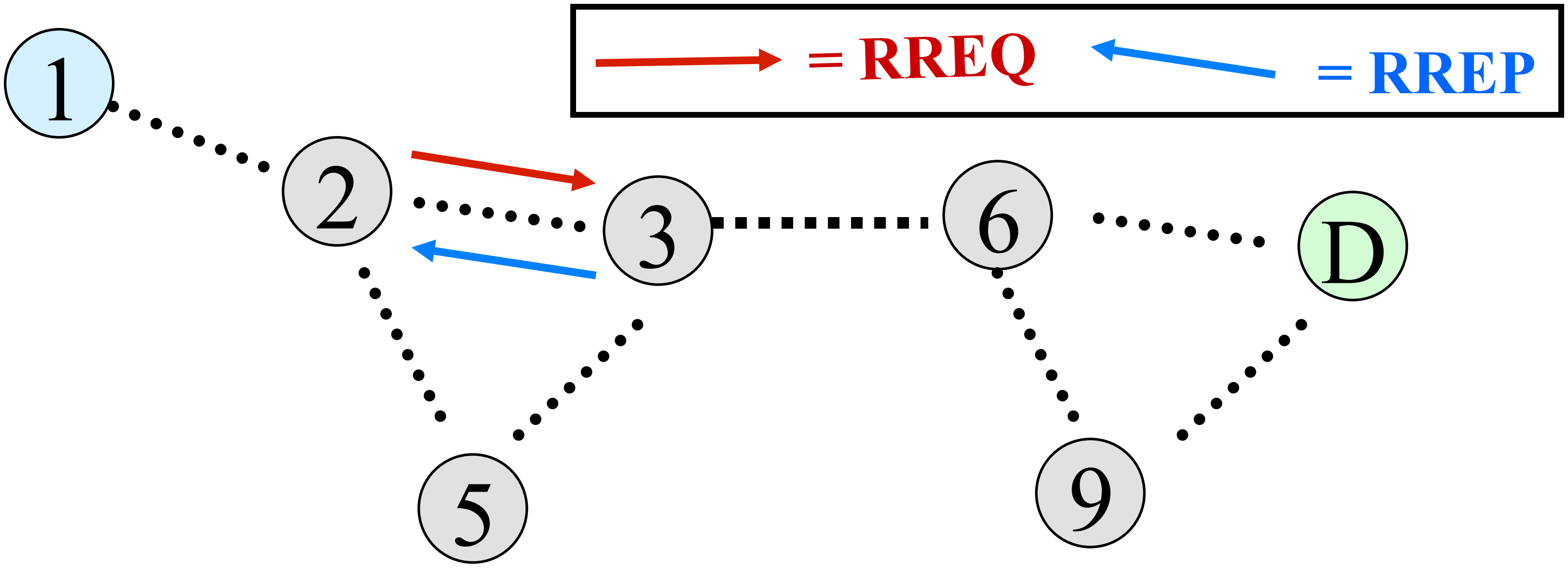
Solution:



Since it receives the RREP message from node 6, it enters Destination = Node D, Next Hop = Node 6.
It then consults its routing table and find that the next hop to Node 1 (the originating node) is Node 2 (this is the reverse path) so it unicasts an RREP to Node 2

Creating the forward path (3)

- Note that, if an intermediate node knows of a path to the destination, it can send a RREP message



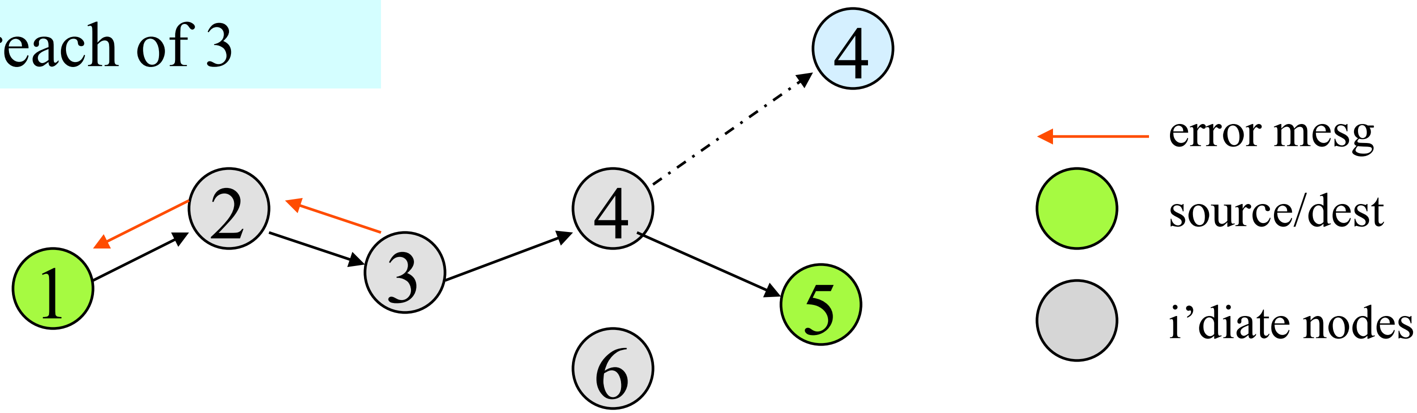
Node 3 knows a valid route to D, it can send a RREP.

Route Maintenance

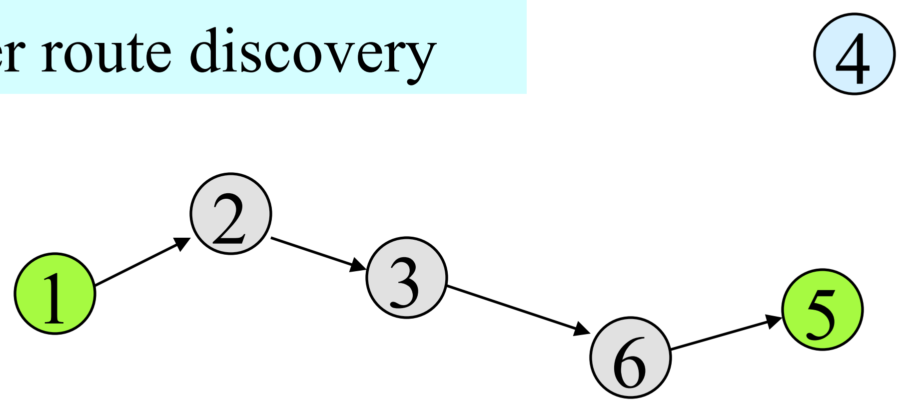
- Due to movements of nodes, routes may be broken during a session
- Such broken routes must be repaired and maintained until the end of the session
- The node which detects a link-break in an active route, back propagates an error message to the source
- The source reinitiates the route discovery process to find a new route

Route Repair (Detour Creation)

4 goes out of reach of 3



After another route discovery



Detecting Route Failure (1)

- Each node broadcasts a *hello message* periodically
- If a node does not receive a hello message from a neighbour in the last period, it means the neighbour has moved out of range
- It purges all entries in its routing table for which the next hop is this unreachable neighbour

Detecting Route Failure (2)

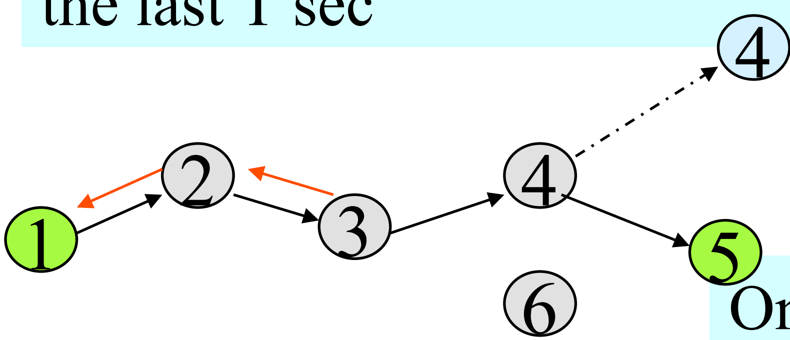
- It also sends error message to all *active neighbours* (for which the next hop is this unreachable neighbour) so they can purge these routes as well from their routing tables
- These *active neighbours* relay the error message to their *active neighbours* and so on

What is an Active Neighbour?

For each destination in a node's routing table, a neighbour is an active neighbour if it fed a packet to this node for this destination in the last T sec

Destination	Next Hop	Active Nghbrs
1	2	4
2	2	4,6
5	4	2
6	6	2

Routing Table at Node 3



Only shaded entries are affected when 4 becomes unreachable to 3. Active neighbour 2 has to be notified.

Route discovery overhead

- Should RREQ be disseminated network wide?
- If RREQ fails to find a path to the destination the first time
  - Should the source node re-initiate another route discovery?
  - How many times should it re-try?

Controlling RREQ message dissemination

- Rationale
  - The destination may be located near the node initiating the route discovery
  - A node nearby may know a route to the destination
- Disseminating RREQ message throughout the network may be unnecessary

Expanding ring search (ERS)

- Searches incrementally bigger area if the previous search fails to find a route
- Implemented by manipulating the TTL field in the IP header
- Similar method is also used in some peer-to-peer networks (e.g. Gnutella)

ERS implementation (1)

- Originating node
  - First route discovery round
    - » Recall RREQ is encapsulated in an IP packet
    - » Set TTL in IP header to TTL_Start
    - » Wait for an RREP message
      - Timeout set to RING_TRAVERSAL_TIME
    - » Case 1: Receives an RREP $\Rightarrow$ Okay
    - » Case 2: No RREP receives and timeout occurs $\Rightarrow$ Next round of route discovery

ERS implementation (2)

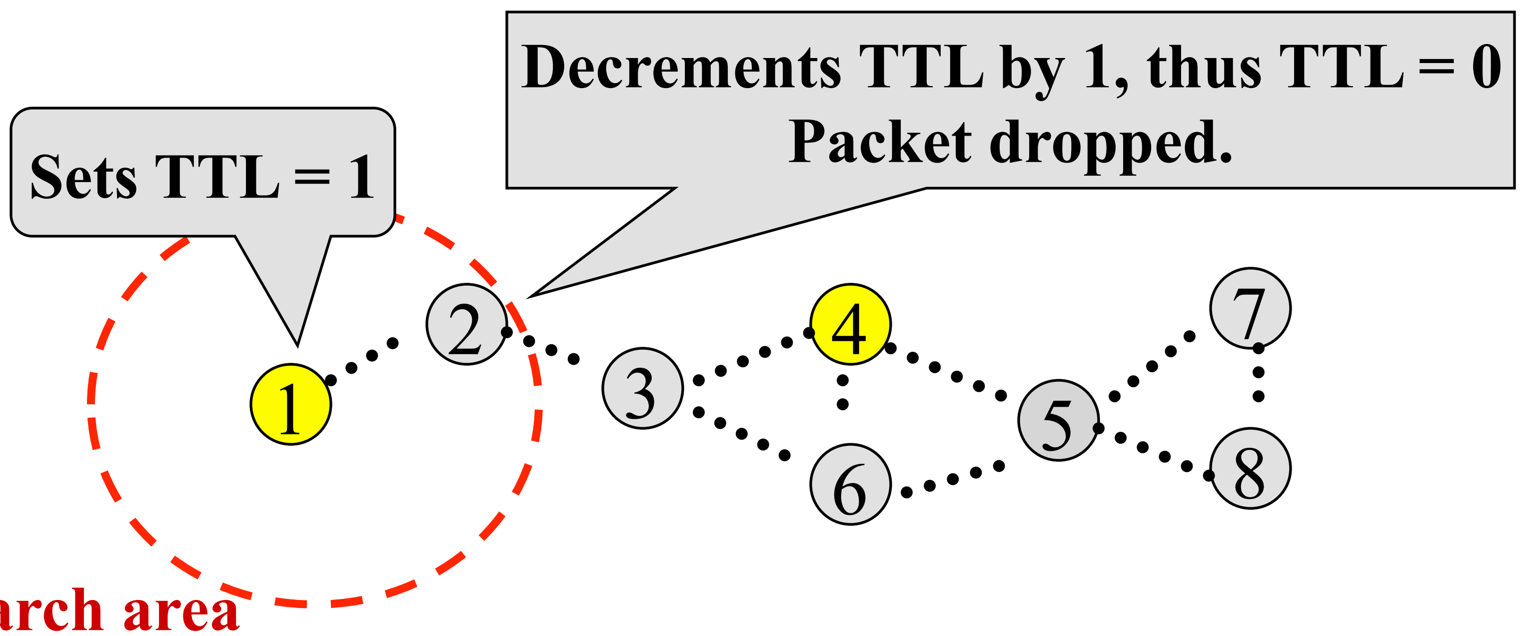
- Second route discovery round
  - » Set TTL in IP header to TTL_Start + TTL_Increment
  - » Wait for an RREP message
- More route discovery rounds are required if the originating node still fails to find a route
  - » For each new round, the TTL is incremented by TTL_Increment
  - » This is continued until TTL reaches TTL_Threshold
    - Initiate network-wide broadcast by setting TTL = Net_Diameter

ERS parameters

- The RFC recommends the following setting
  - TTL_START = 1
  - TTL_Increment = 2
  - TTL_Threshold = 7
  - Net_Diameter = 35
  - Node_Traversal_Time = 40ms
  - Ring_Traversal_Time = 2 x Node_Traversal_Time x (TTL_value + Timeout_Buffer)
  - Timeout_Buffer = 2
    - Configurable parameter to account for possible congestion

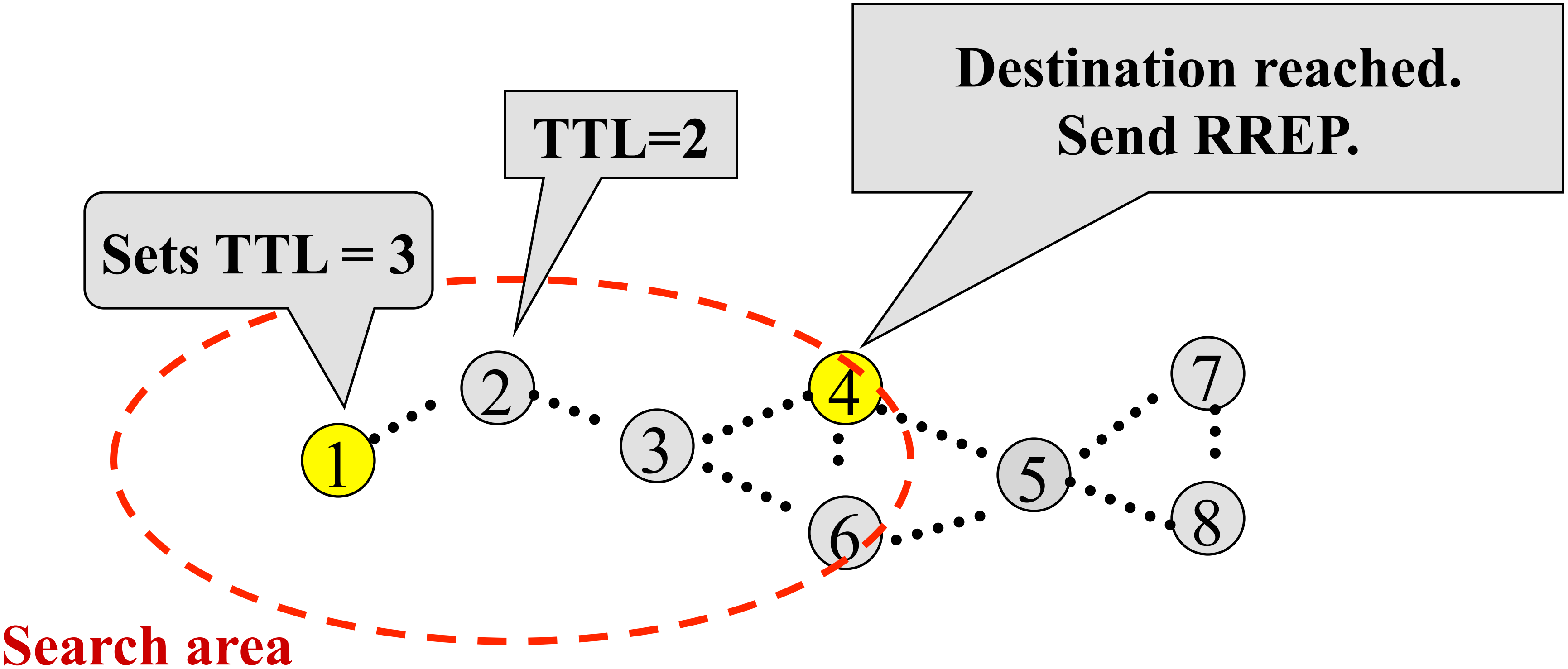
ERS example (1)

Assumptions: Node 1: Source. Node 4: Destination.
No nodes know about a route to node 4.



Originating node timeout as no RREP is received.
It sets TTL = 3 in the next round of search.

ERS example (2)



Note that RREQ does not reach nodes 5,7 and 8.
Thus saving them from processing RREQ messages.

Controlling repeated route discovery attempts (1)

- When TTL is set to Net_Diameter, how many times should the originating nodes be attempting to discover route in a network wide manner?
- RFC says this should be done using binary exponential backoff

Controlling repeated route discovery attempts (2)

- First time when a network-wide route discovery is done
  - Timeout = Net_Traversal_Time
  - Wait this long for RREP before the 2nd attempt for the same destination is made
- Binary exponential backoff means
  - Timeout for attempt (k+1) = 2 x timeout for attempt k
  - Example:
    - » Timeout for the 2nd attempt = 2 x Net_Traversal_Time
    - » Timeout for the 3rd attempt = 4 x Net_Traversal_Time
- Maximum number of retries = RREQ_RETRIES
  - RFC recommends RREQ_RETRIES be set to 2

References

- RFC3561 Ad hoc on-demand distance vector (AODV) routing