Wireless networks

Routing: DSR, AODV

Routing in Ad Hoc Networks

- Goals
  - Adapt quickly to topology changes
  - No centralization
  - Loop free routing
  - Load balancing among different routes in case of congestion
  - Supporting asymmetic communications
  - Low overhead and memory requirements
  - Security

Routing in Ad Hoc Networks (2)

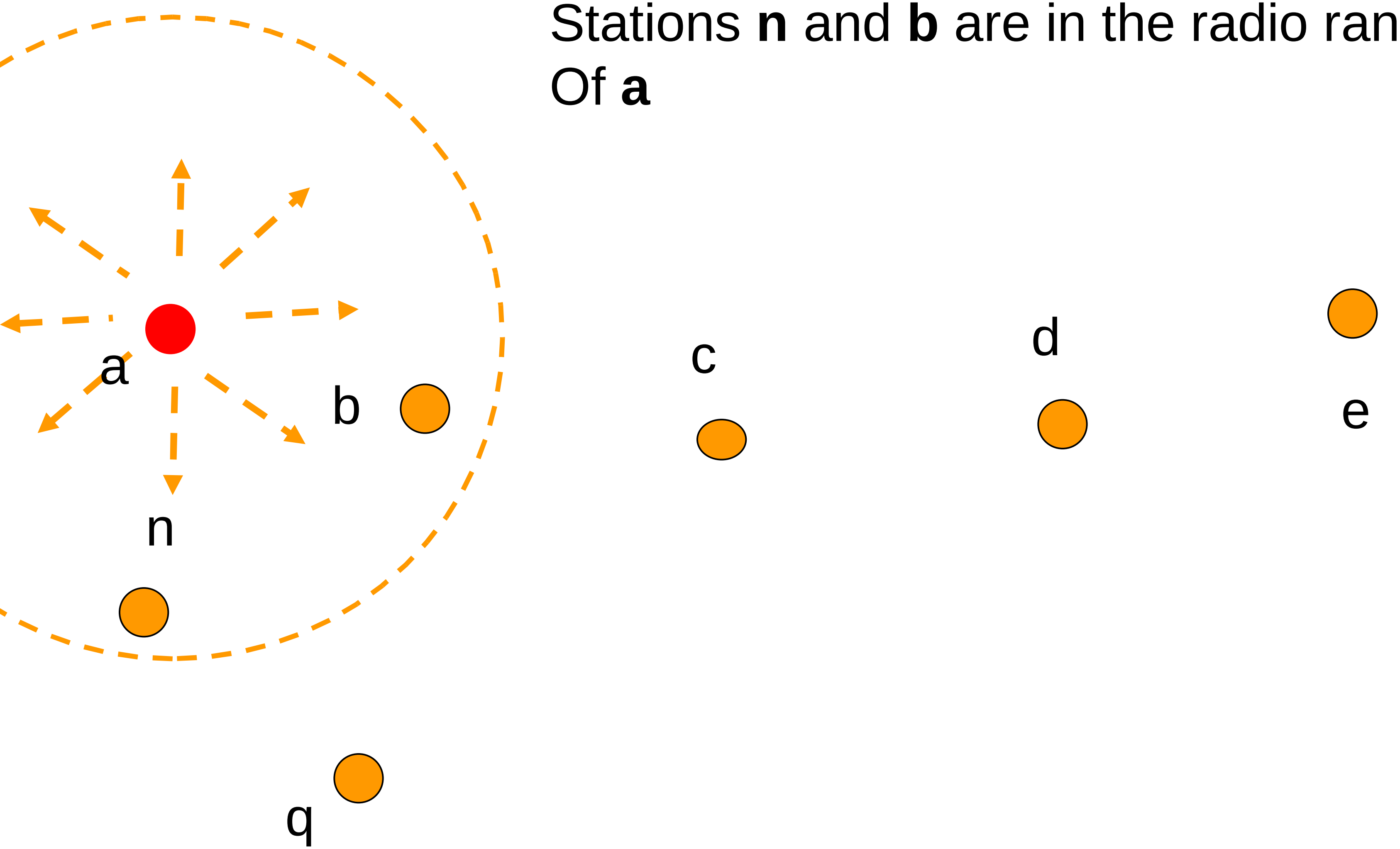
- Many proposal
- Proactive routing protocols
  - attempt to maintain consistent, up to date routing information from each node to every other node in the network
- Reactive
  - Discover a route when desired by the source node
  - DSR AODV
- Hybrid

Routing Model: HN as graphs

- In the discussion of routing algorithm we use a more convenient model for ad Hoc Networks
  - Each terminal/station is represented as a node in a graph
  - Each directed arc *(i,j)* states that station j is within the radio range of *i*, and can receive packets from *i*

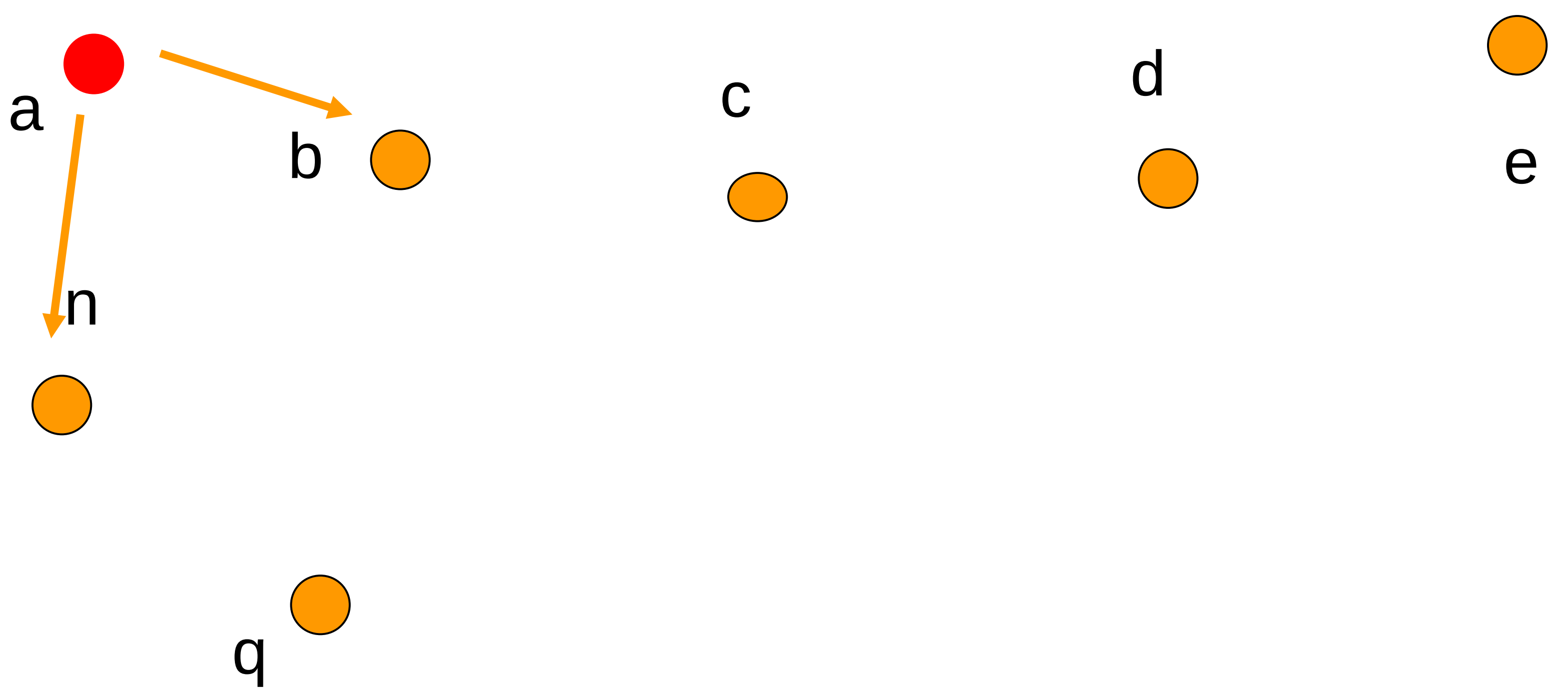
Example: HN graph

Stations **n** and **b** are in the radio range Of **a**



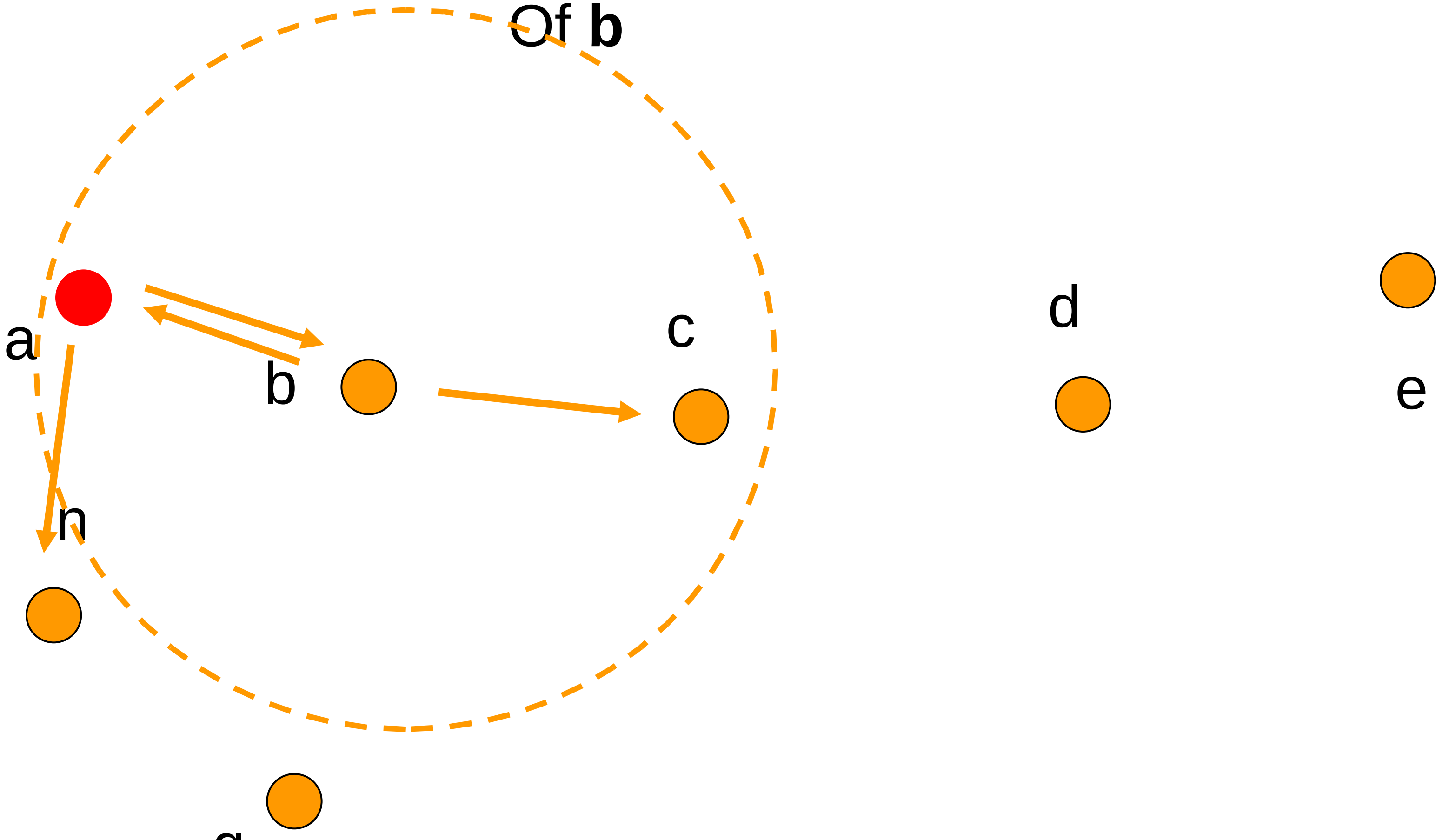
Example: HN graph (2)

Stations **n** and **b** are in the radio range
Of **a**



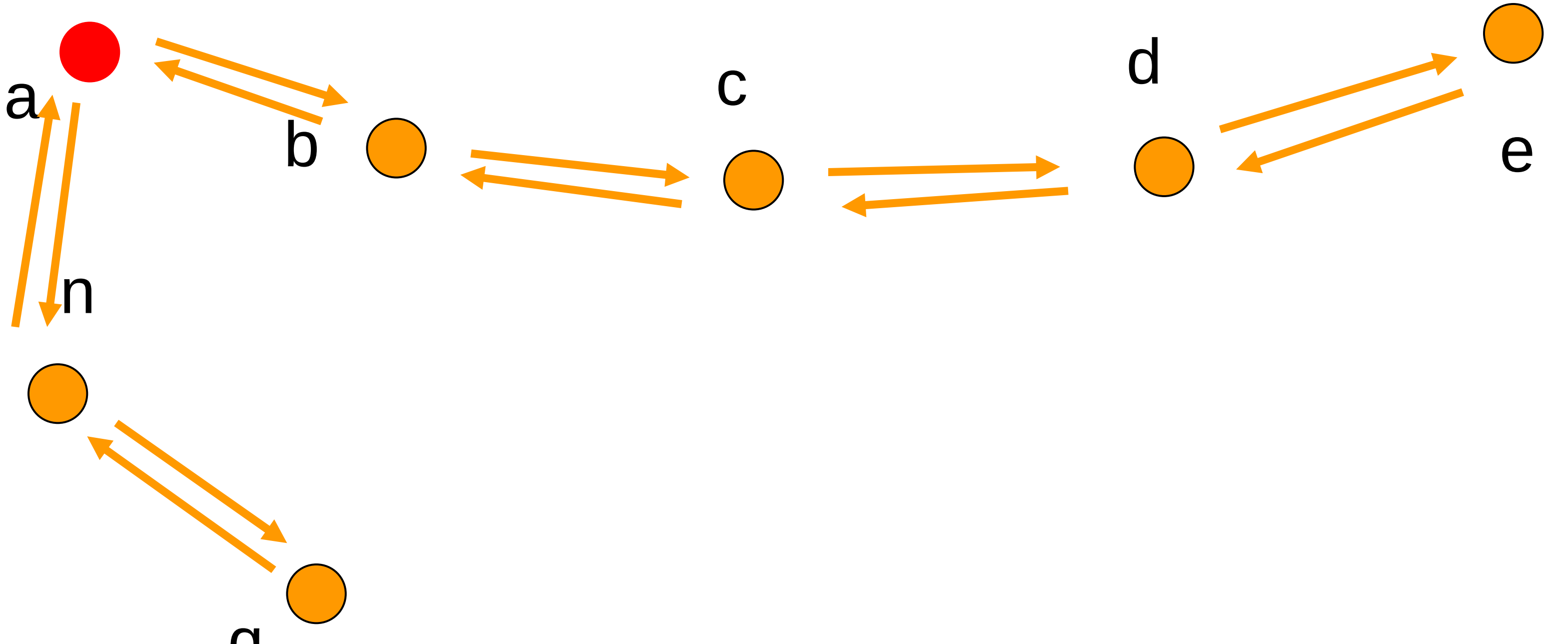
Example: HN graph (3)

Stations **a** and **c** are in the radio range Of **b**



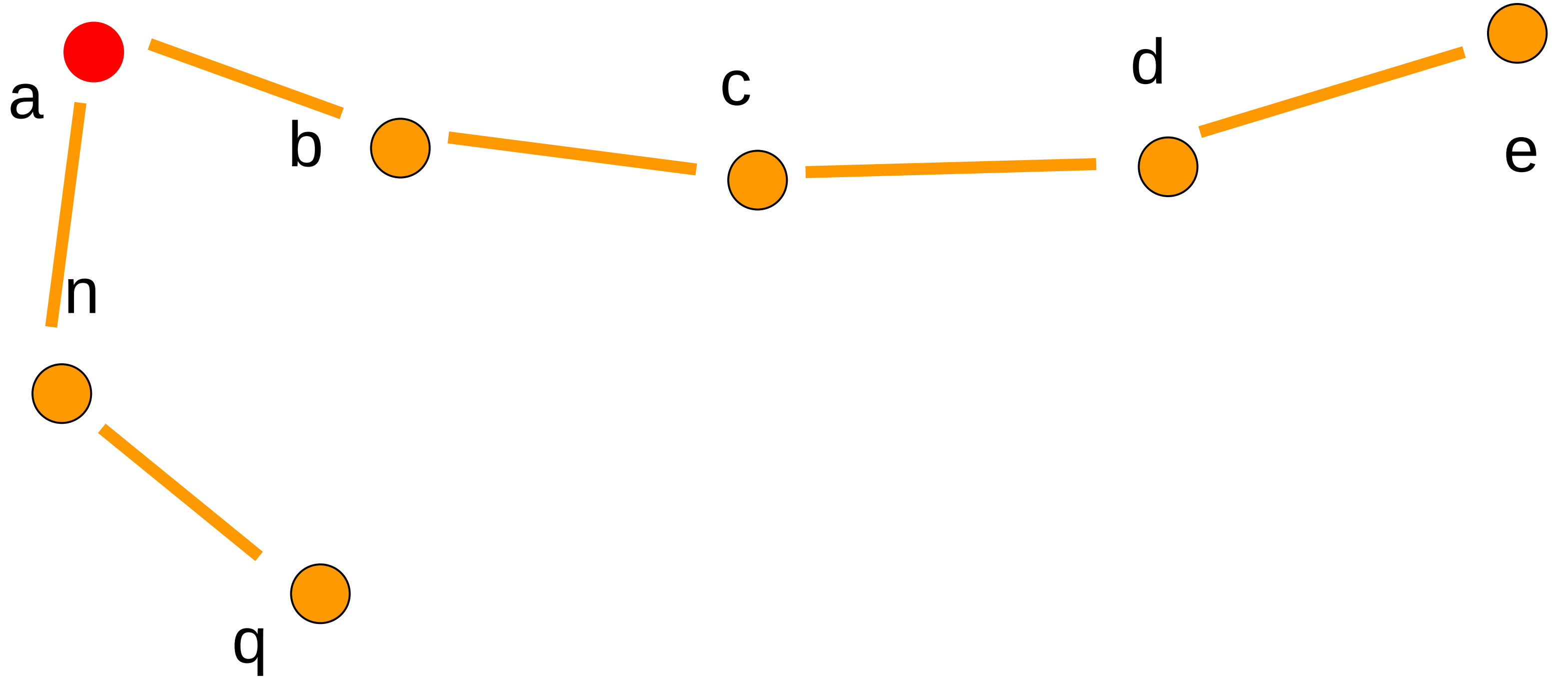
Example: HN graph (3)

Following ....



Example: HN graph (4)

If we assume same radio range for
All nodes and circular transmission area
Links are *symmetric* and we can use an
undirected graph



Dynamic Source Routing

- RFC 4728 IETF-MANET working group
- Proposed in 1994 by Johnson
  - Monarch Project - CMU
- Source routing
- Goals:
  - Low overhead
  - React quickly to changes in the network
  - No centralization point

DSR: assumptions

- Cooperative nodes:
  - All nodes want to participate fully in the network protocol and *will forward packets for other nodes*
- Small network diameter
  - The number of hops needed to travel from any node at the extreme edge of the network (*diameter*) is small (around 5-10) but greater than 1

DSR: assumptions (2)

- Corrupted packets
  - A corrupted packet can be recognized and discarded by its destination
- Mobile nodes
  - Nodes in the network may move at any time without notice. Speed is moderate wrt packet transmission latency
- Bidirectional Symmetric links
  - If node *i* can reach node *j* in its radio range, then then communication from *j* to *i* can be established as well

DSR: basic mechanisms

- Route Discovery (RD)
  - By which a node **S** wishing to send a packet to **D** obtains a source route to D. It is used only if no route is already known.
- Route Maintenance (RM)
  - By which **S**, which already knows a route to **D**, is able to detect that topology has changed and that route is no longer available. In this case S can use any other route it happens to know or invoke RD again

DSR: Route Discovery

- S originated a packet for D
  - **S** searches for a source route **r** to **D** in its *Route Cache*
  - If it finds it, **S** places **r** in the header of the new packet and sends it
  - Otherwise it starts a Route Discovery protocol sending a *route request*
    - **S** is called the *initiator* of RD
    - **D** is called the *target* of RD

DSR: Route Discovery Protocol

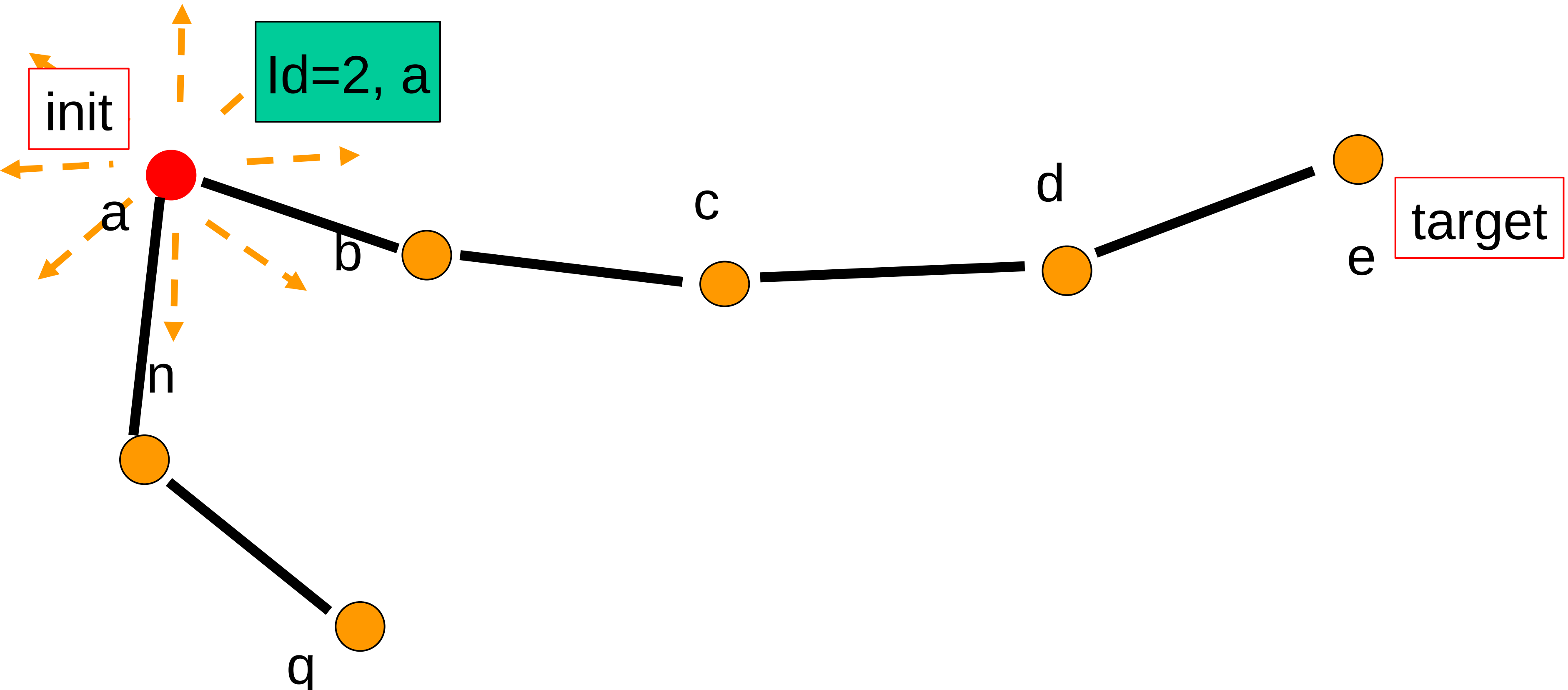
- Sending a Route Request
  - **S** sends a *route request* message to all the nodes it can reach directly (local broadcast)
  - Each copy of *route request* contains
    - initiator, target and (unique) request ID
    - The list of nodes through which this particular copy has been forwarded (initially empty)

DSR: Route Discovery Protocol (2)

- Replying to a Route Request
  - When the target **D** gests a *route request*
    - Returns a route reply message to the route discovery initiator S including a copy of the accumulated route record in the request. When the initiator gest this reply it caches the route in its cache for subsequent use.
  - A node **N** (not the target) that gests a *route request*
    - If it is already present in the list, or it has received a request recently with the same ID: discards the message
    - Otherwise it appends its own addres to the route record and forwards it with a local broadcast

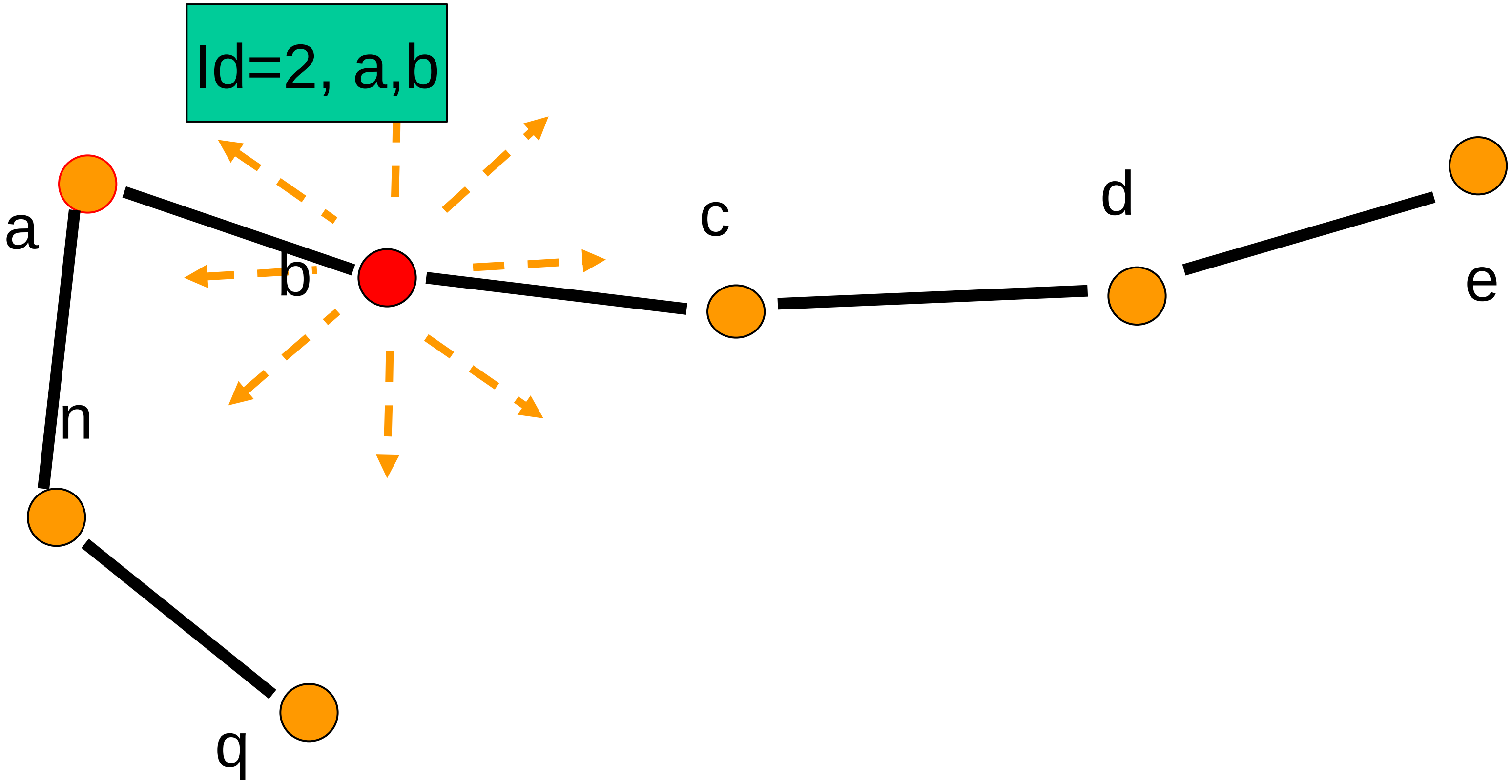
Example: Route Discovery Protocol

a (the intiator) sends a route request, with request ID **2** and route record **a**



Example: RD Protocol (2)

b forwards the route request (ID **2)** with route record **a,b**

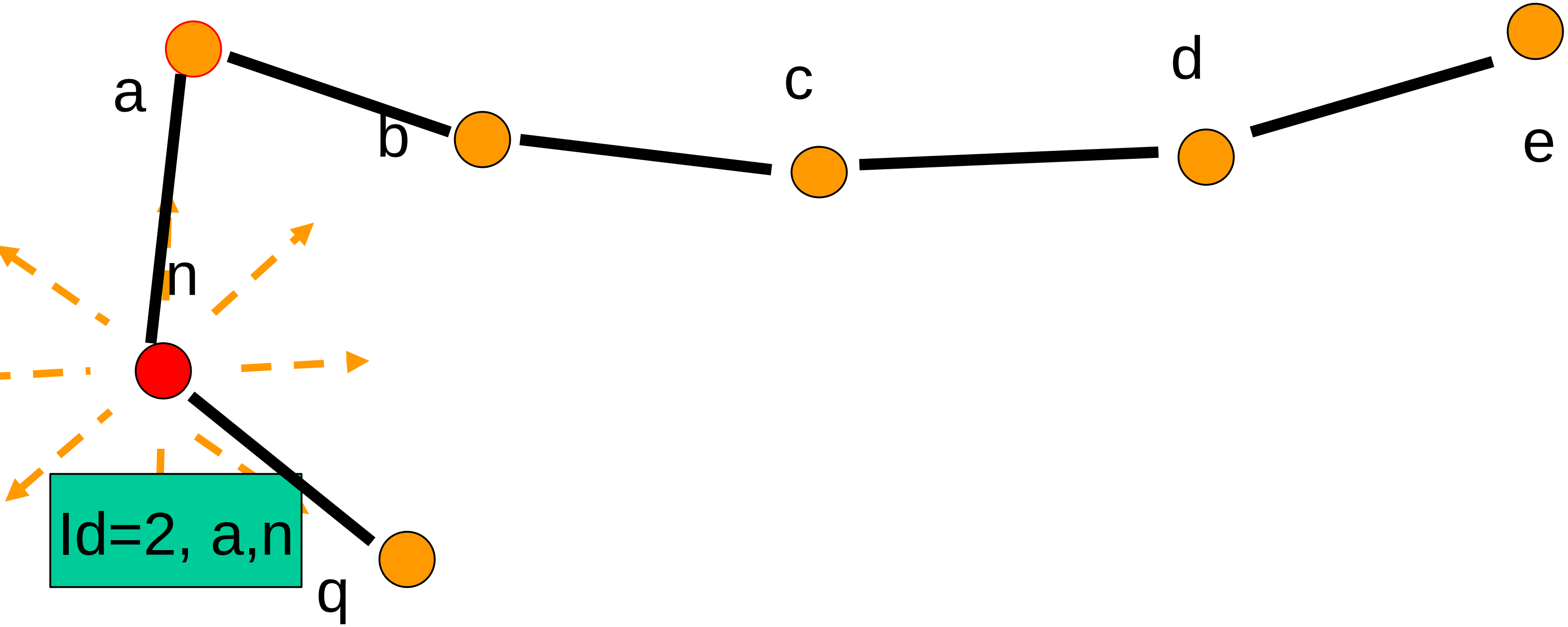


Example: RD Protocol (3)

n forwards the route request (ID **2)** with route record **a,n**

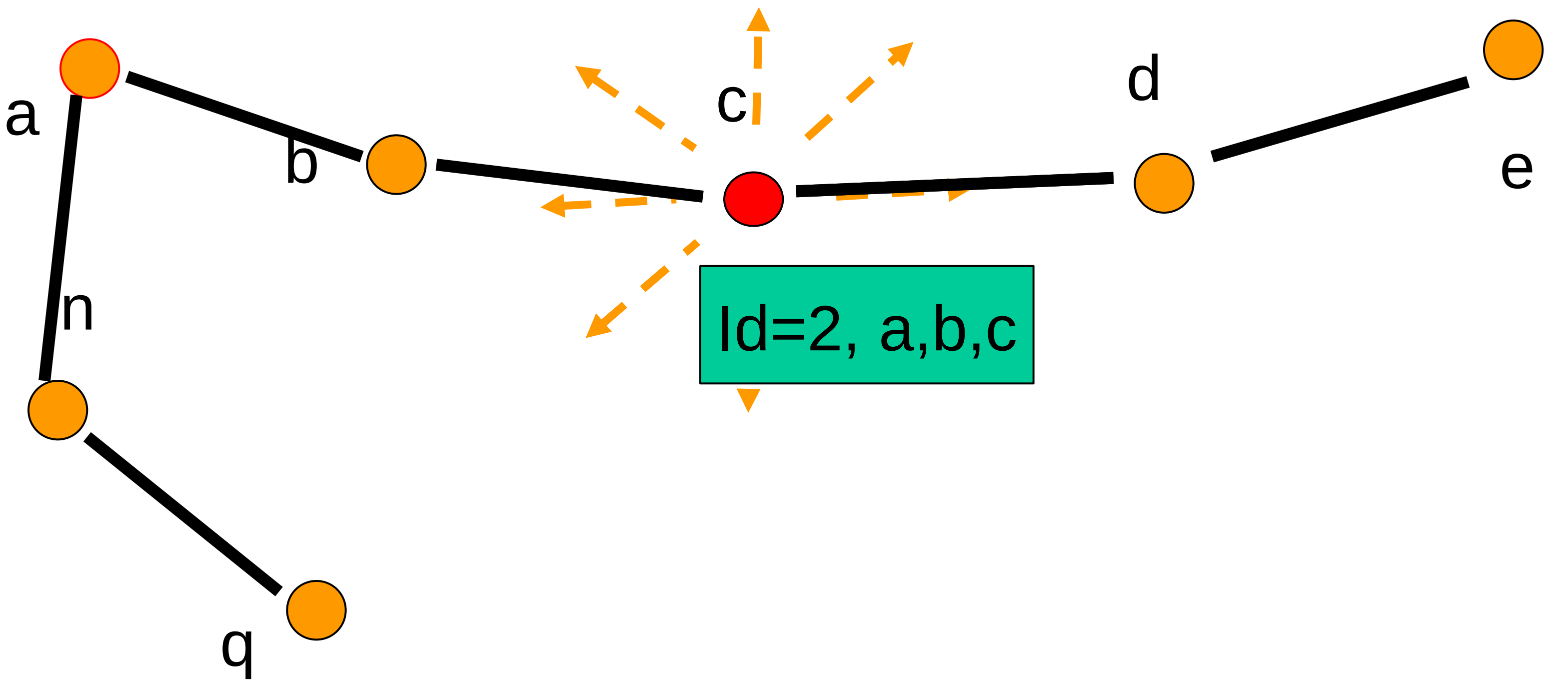
a will discard the msg (already in list)

And b will discard the msg (just processed same id)



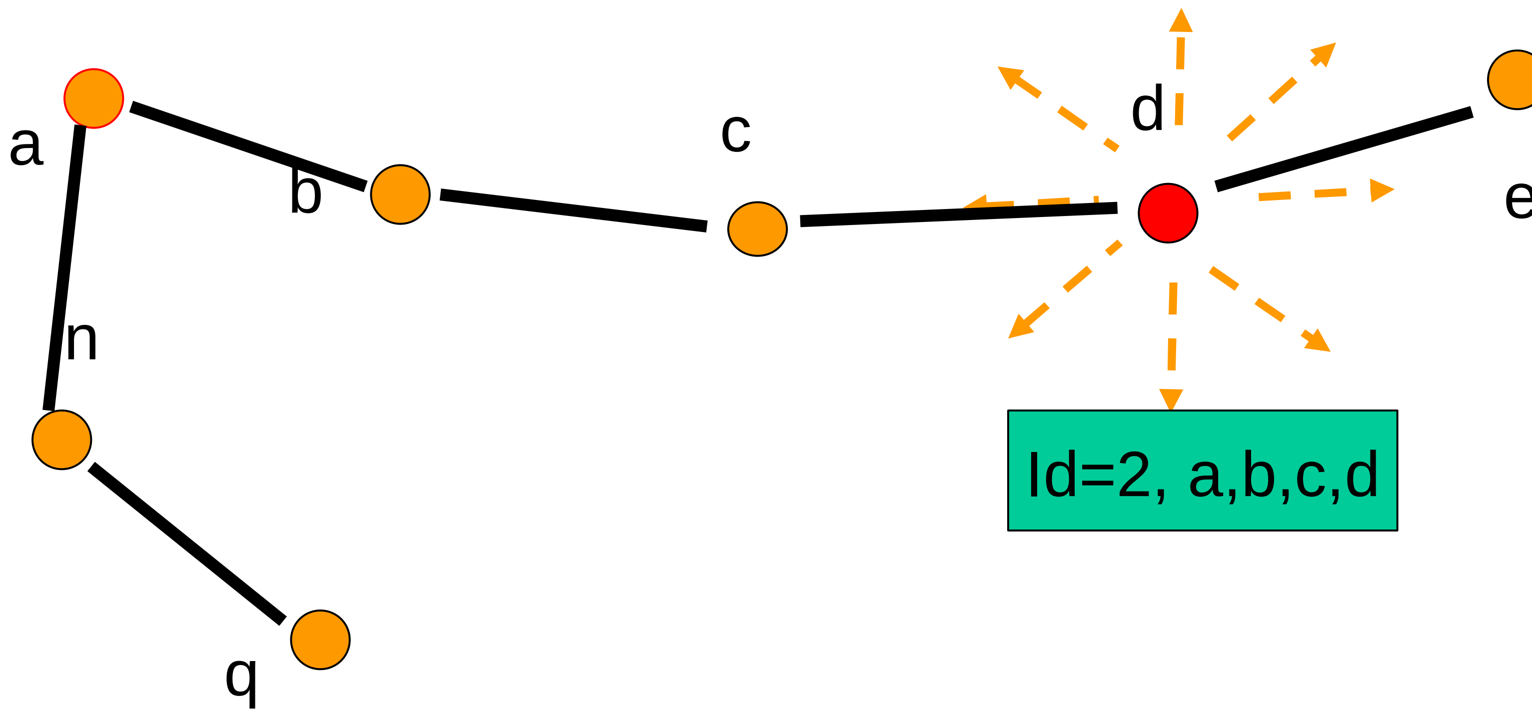
Example: RD Protocol (4)

c forwards the route request (ID **2)** with route record **a,b,c**



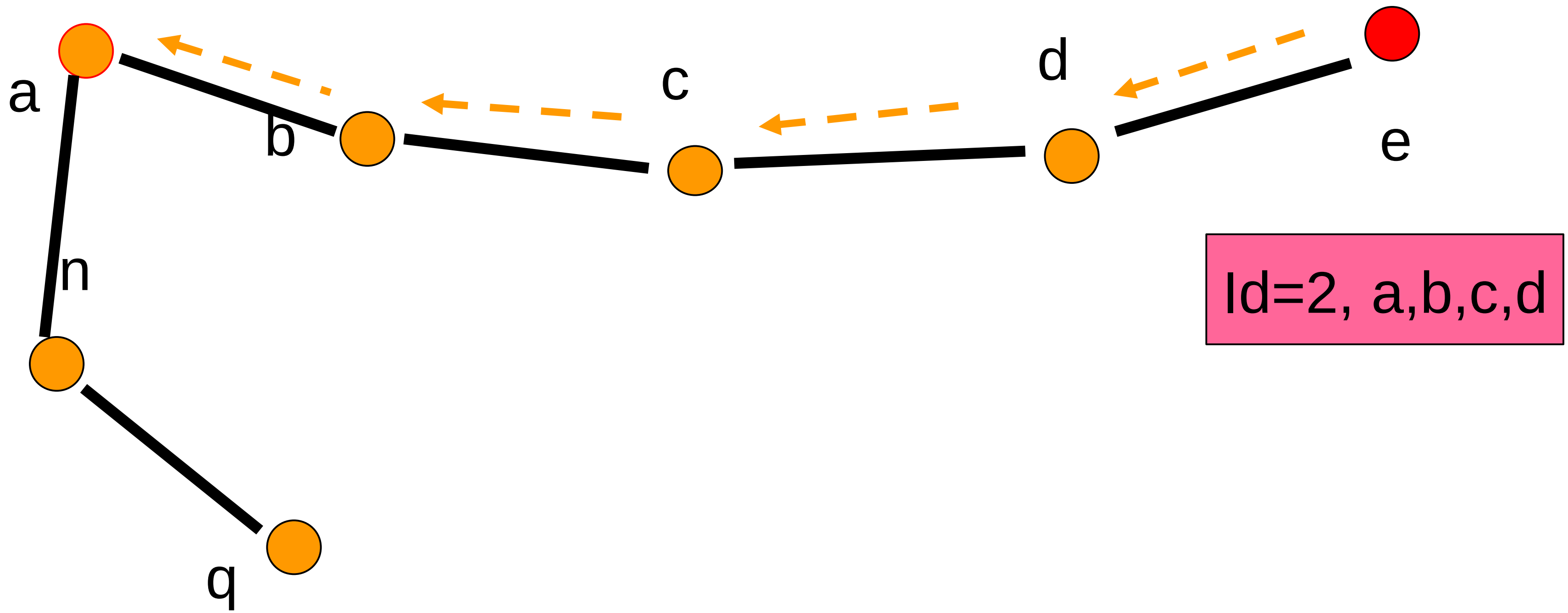
Example: RD Protocol (5)

d forwards the route request (ID **2)** with route record **a,b,c,d**



Example: RD Protocol (6)

e (the target) is reached and sends
a **<u>route reply</u>** with route record **a,b,c,d**



DSR: Returning a route reply

- Examines Route Cache
  - Target looks for a route back to initiator in its own cache and, if found, uses it for delivering the packet containing the Route Reply
- If a route is not known
  - Starts a route discovery, possibly combined with Route Reply packet
  - Reverses the route found from target (works only if we have all bidirectional links)

DSR: Packets waiting for a RD

- Packets that cannot be send because no route has been discovered yet are kept in a *Send Buffer*
  - Expire and are deleted after a timeout
  - Can be evicted with some policy (eg, FIFO) to prevent Send Buffer from overflowing
- While a packet is in the Send Buffer
  - The node occasionally starts a route discovery
  - The rate for new discoveries of the same address is limited to not overflow the network (target may be unreachable). Use *exponential* back-off.

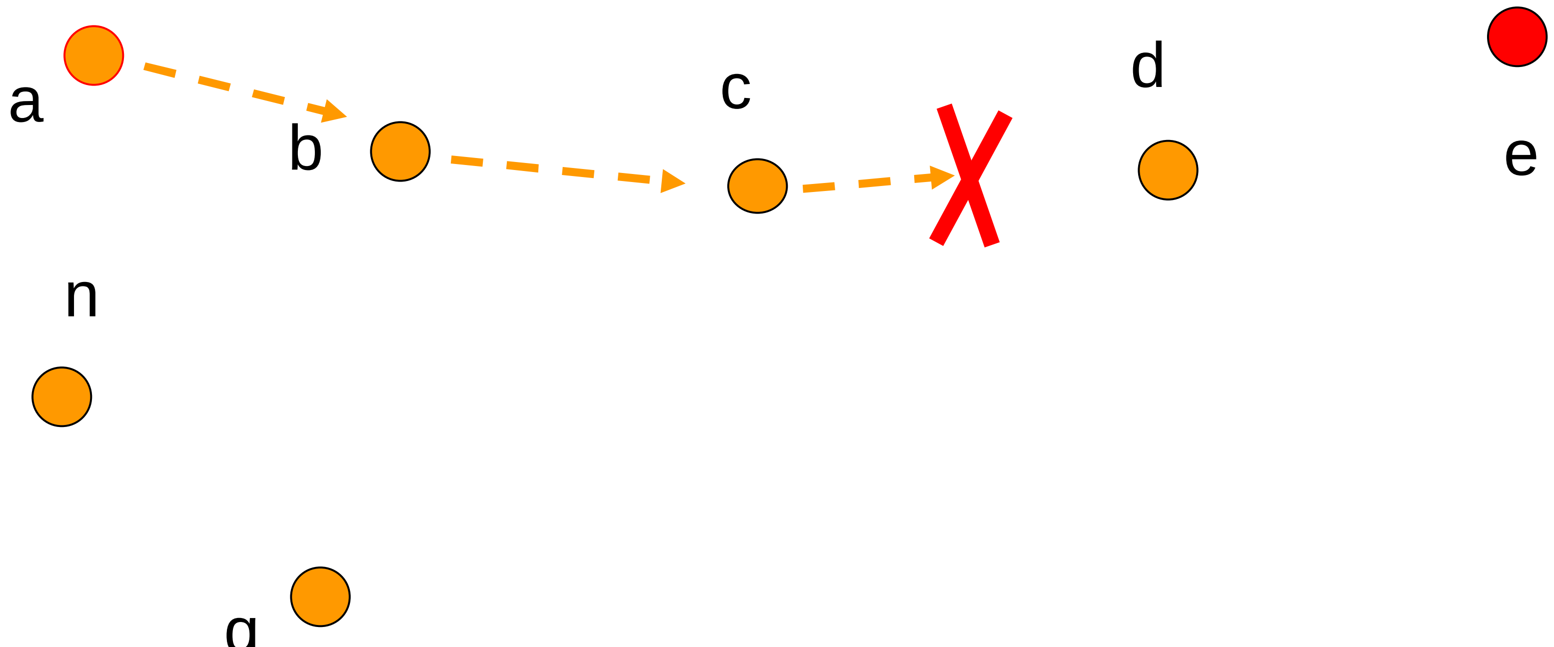
DSR: Route Maintenance

- When sending a packet along a route
  - Each node in the route is responsible of the receipt of the packet at the following hop in the route
  - For instance: Sending from **a** to **e** on route **a-b-c-d**,
    - **a** is responsible for packet receipt at **b**,
    - **b** for packet receipt at **c** etc
  - Packet is retransmitted on a hop up to a max number of times until ack is received
  - Ack can be provided by MAC layer or explicitely sent by DSR level

Example: Route maintenance

Sending to e with route **a-b-c-d**

Link **c** to **d** is down



DSR: Route Maintenance (2)

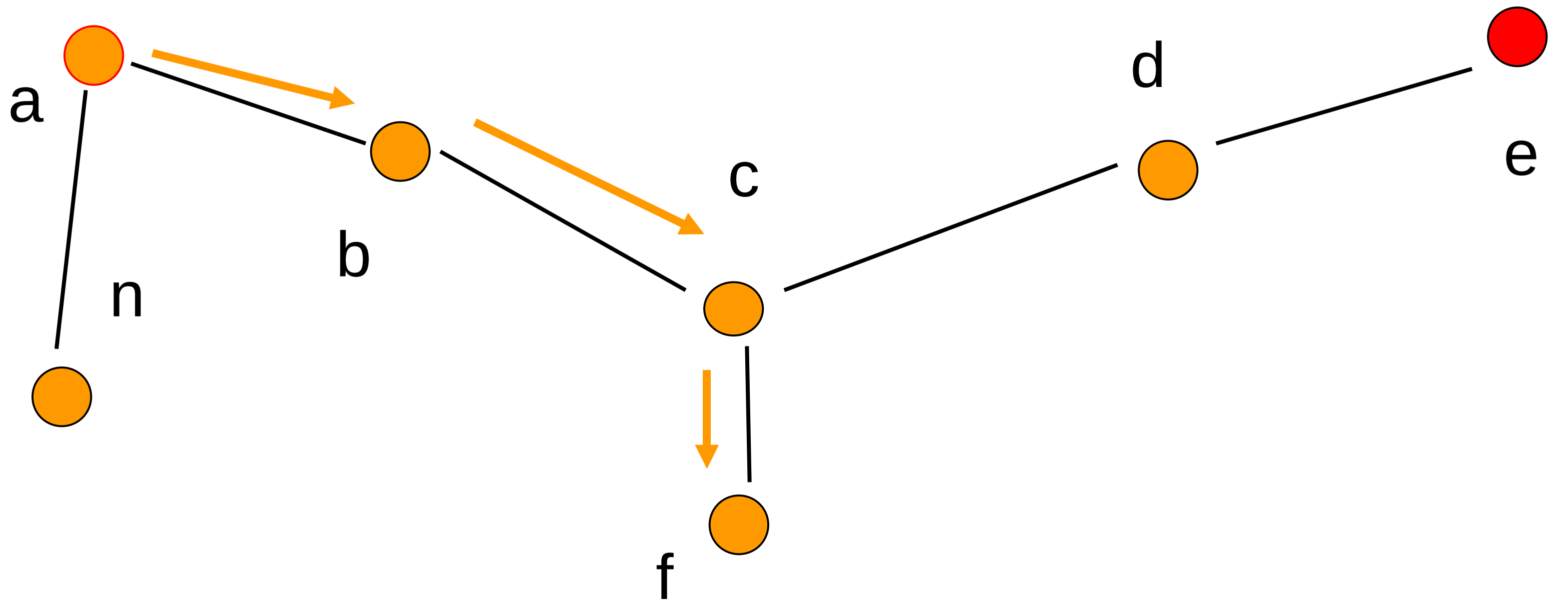
- When a route link is down
  - A *Route Error* is returned to the sender stating “link broken”
  - The sender removes this route from the cache (and others contating the same link)
  - Error is reported to the upper layers that can decide for retransmission
  - When retransmission is asked a new route can be extracted from cache or an RD protocol started (if none is present)

DSR: Additional RD features

- Overheard routig information
  - A node not only caches the results of a RD procedure
  - It can also cache the accumulated route in a Route Request, the route in a Route Reply, or the source route used in a data packet
- Replying to route requests using cached routes
  - If an intermediate node has already a source route to the target in its cache it reply directly with a Route Reply to the initiator

Example: Using cached routes

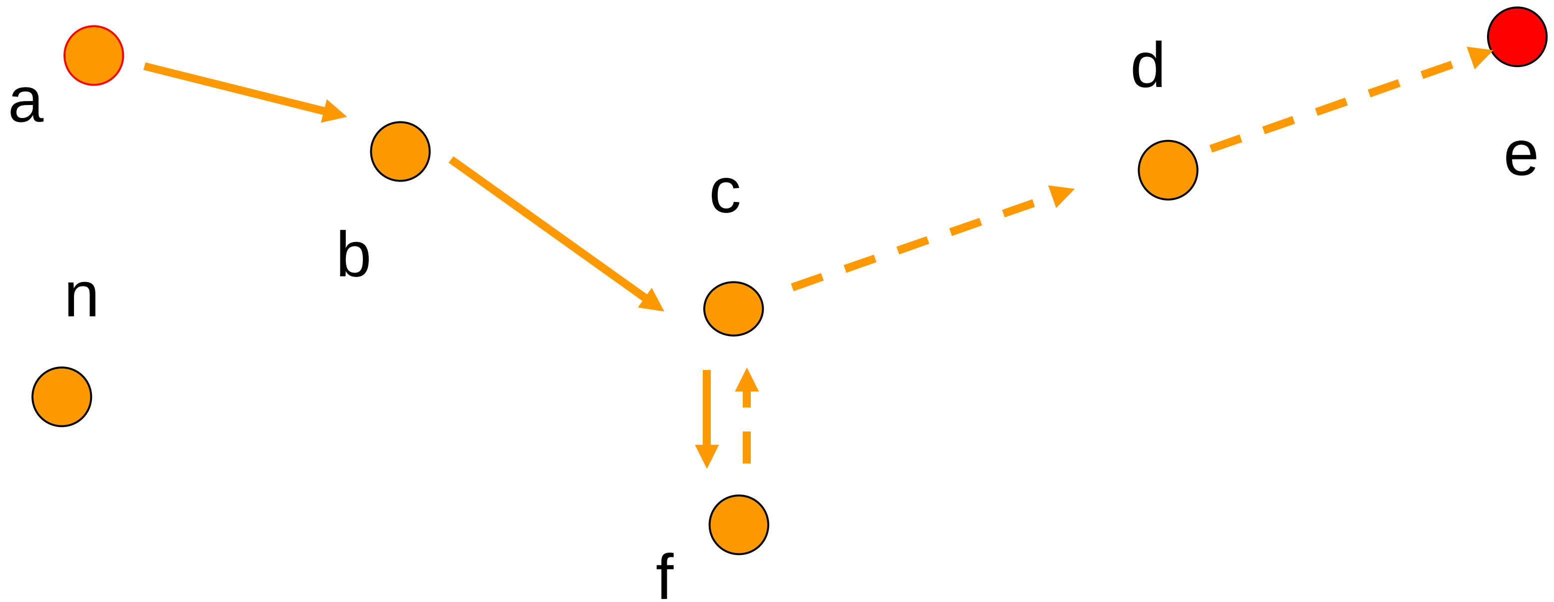
Route discovery from **a** to **e** reaches **f**
Node f has already a route f-c-d to e in its cache



Example: Using cached routes (2)

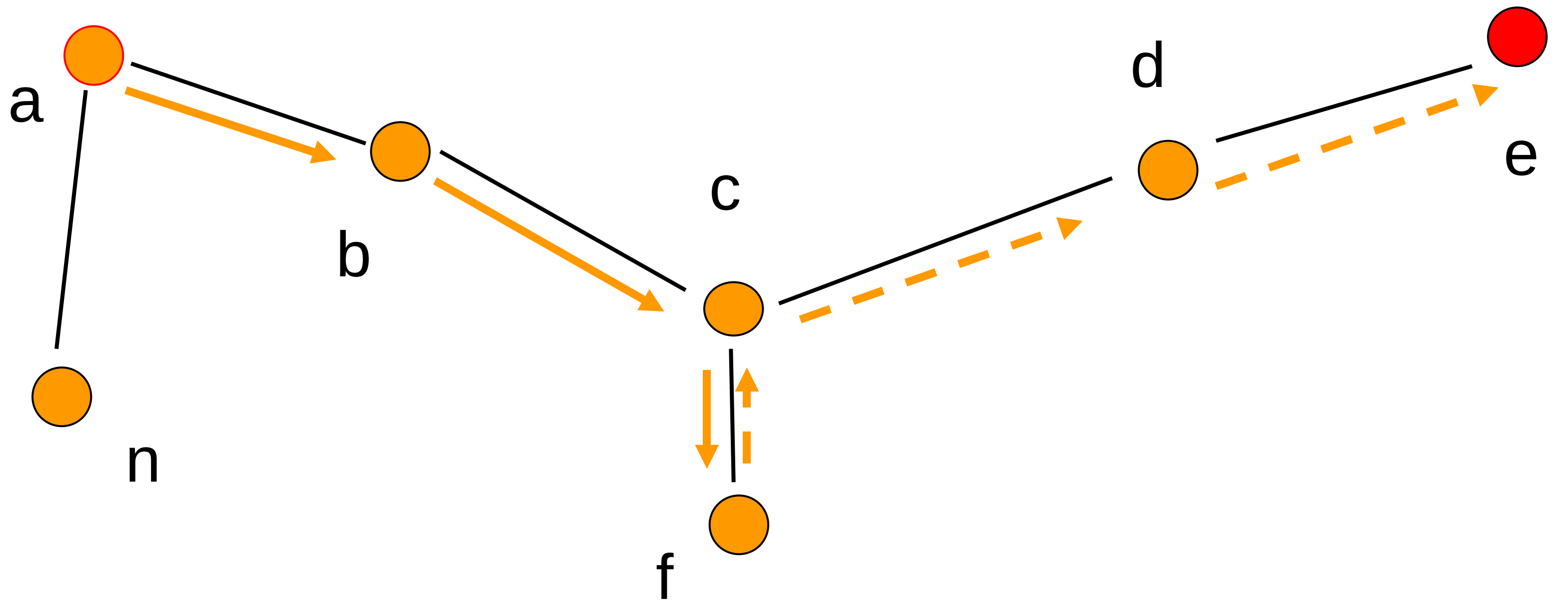
Route discovery from **a** reaches **f**

Node **f** has already a route **f-c-d-e** in its cache



Example: Using cached routes (3)

Route discovery from **a** reaches **f**
Node **f** has already a route **f-c-d**-**e** in its cache
f cannot reply directly because **a-b-c-f-c-d-e** has duplicated nodes



DSR: Additional RD features (2)

- Avoiding route reply storms
  - When a node starts an RD many neighbors may have cached route and respond directly
  - To avoid collisions and to favour shortest routes a node must wait for a random period

    $$\boldsymbol{d = H * (h - 1 + r)}$$

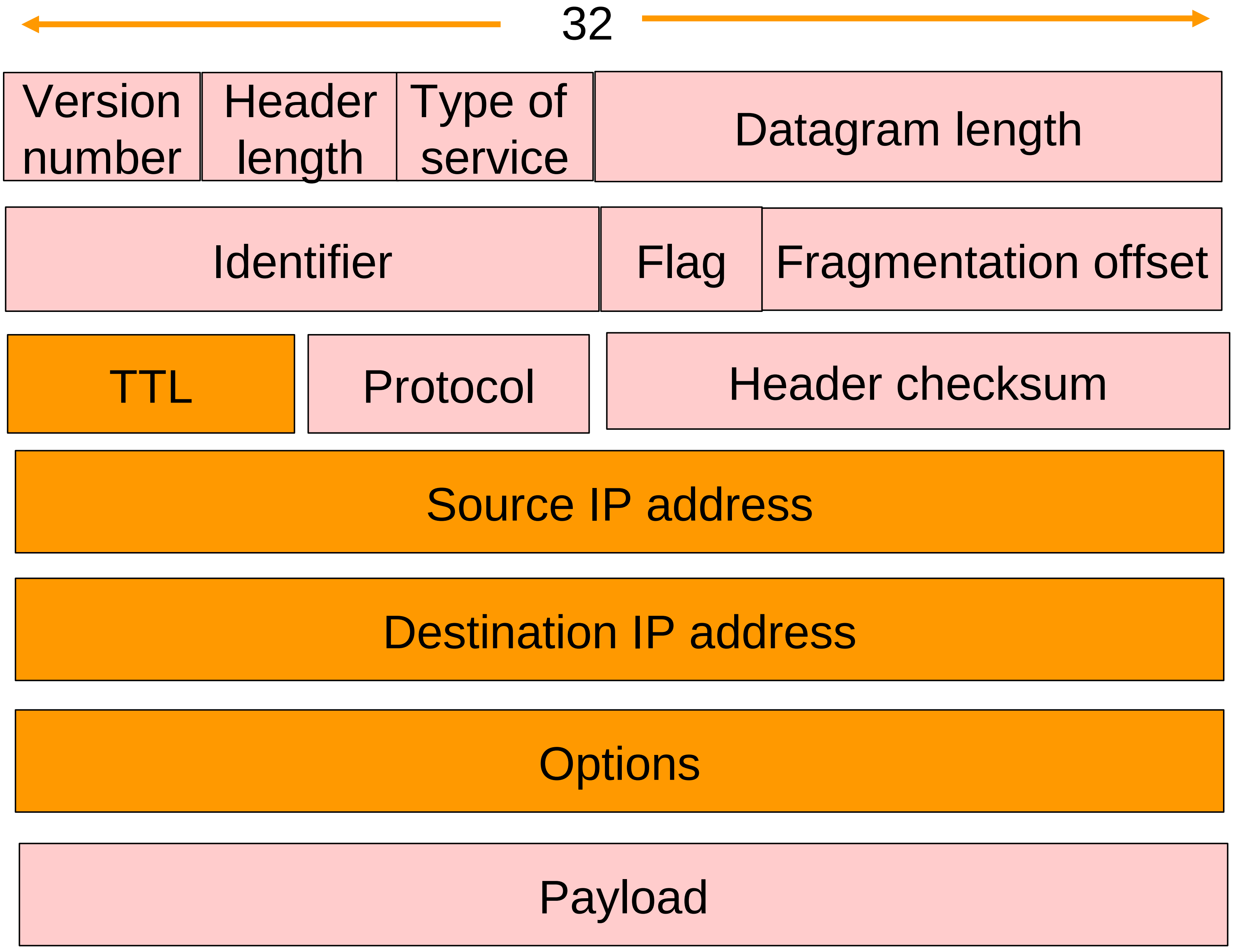
    - ***h*** is the length in number of the network hops for the route to be returned
    - ***r*** is a random number between 0 and 1
    - ***H*** a constant delay (at least twice the maximum wireless proagation delay)

DSR: Additional features

- In RD we can ask for routes with a limited number of hops
- In RM we can:
  - Have intermediate nodes automatically shortening routes where intermediate hopes are no longer needed
    - They send back a new route reply with the new route
  - Cache broken links to prevent their use in other routes

DSR: Packets

- DSR packets are standard IP packets
  - Use a special header in options (next slide for IPv4)
  - Uses standard IP fields such as source and destination address, TTL for hop counting
- DSR option header
  - Fixed portion (4 bytes), including total header length
  - Variable portion : zero or more DSR options, including source route when sending packets
  - Variable formats for different type of packets


32
Version number
Header length
Type of service
Datagram length
Identifier
Flag
Fragmentation offset
TTL
Protocol
Header checksum
Source IP address
Destination IP address
Options
Payload

DSR: References

[Johnson 1994]

D.B. Johnson. *Routing in Ad Hoc networks of Mobile Hosts*. Proceedings of IEEE Mobile Computing Systems and Applications.Dec 1994. 158-163

[Johnson et al. 2001]

D.B. Johnson, D.A. Maltz, J. Broch. DSR the dynamic Source routing protocol for multihop wireless ad hoc networks. Cap 5 of *Ad hoc networking* (C.E. Perkins Ed.), Addison-Wesley, 2001.

[RFC4728]

http://www.ietf.org/rfc/rfc4728.txt

AODV

- RFC 3561 IETF-MANET working group
- Proposed in 1994 by Perkins
  - Monarch Project - CMU
- NO Source routing
- Goals:
  - Low overhead
  - React quickly to changes in the network
  - No centralization point
  - Integrating unicast, multicast, broadcast

AODV: assumptions

- Cooperative nodes:
  - All nodes want to participate fully in the network protocol and *<u>will forward packets for other nodes</u>*
- Bidirectional symmetric links
  - A node which has received a packet from a neighbor is able to route it back to the sender using the same link

AODV: basic mechanisms

Unicast route establishment

- *Route discovery*:
  - Broadcasting a RREQ packet
  - Answering a RREP packet
- *Route maintenance*
  - Handling RERR packets
  - Aging routes

AODV: basic data structures

- Route table
  - Uses a route table for unicast and one for mulkticast
  - It contains at most one route to each destination
    - For each destination it maintains the next hop to destination and a precursor in the route
    - Each table entry is tagged with a lifetime, if not used within lifetime it expires
- Sequence numbers
  - Each node maintains a monotonic sequence number.
    - The sequence number is increased each time a node learns a change in its neighborhood
  - Each multicast group maintains a separate seq. number

AODV: Route Discovery

- S originated a packet for D
  - **S** checks the route table for a current route to **D**. If it finds it, **S** sends the packet along it
  - Otherwise **S** starts a *Route Discovery process*. **S** broadcasts an RREQ packet including:
    - IP address of source and its current sequence number
    - IP address of destination and last known sequence number
    - Broadcast ID, which is incremented each time node S initiates a RREQ
    - Hop count initially set to 0
  - **S** sets a timer to wait for a reply

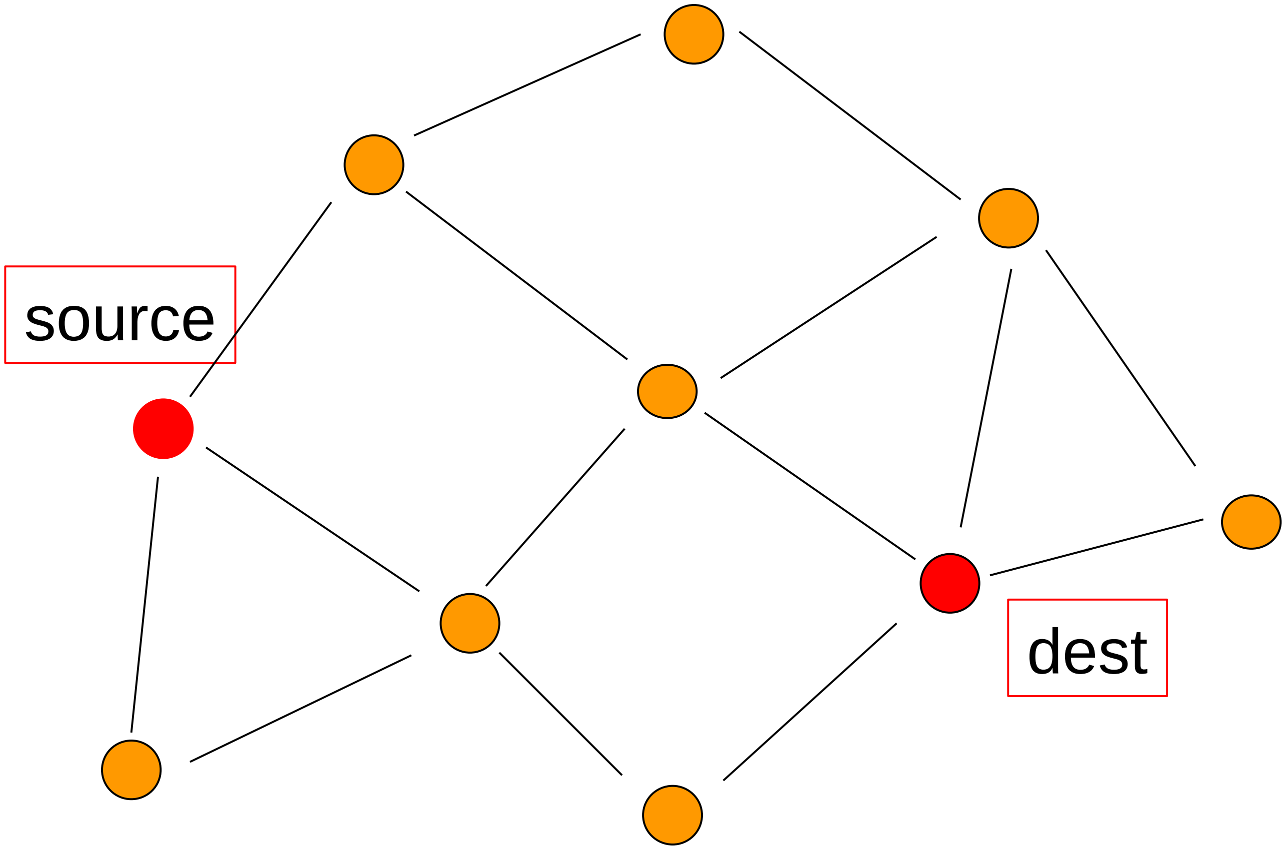
AODV: Route Discovery (2)

- When a node receives a RREQ
  - it first checks if it has seen it before (IP source and broadcast ID)
    - Each node maintains a note of all RREQ seen for a specified length of time
    - If already seen it silently discards it
    - Olterwise it records it and processes it

AODV: Route Discovery (3)

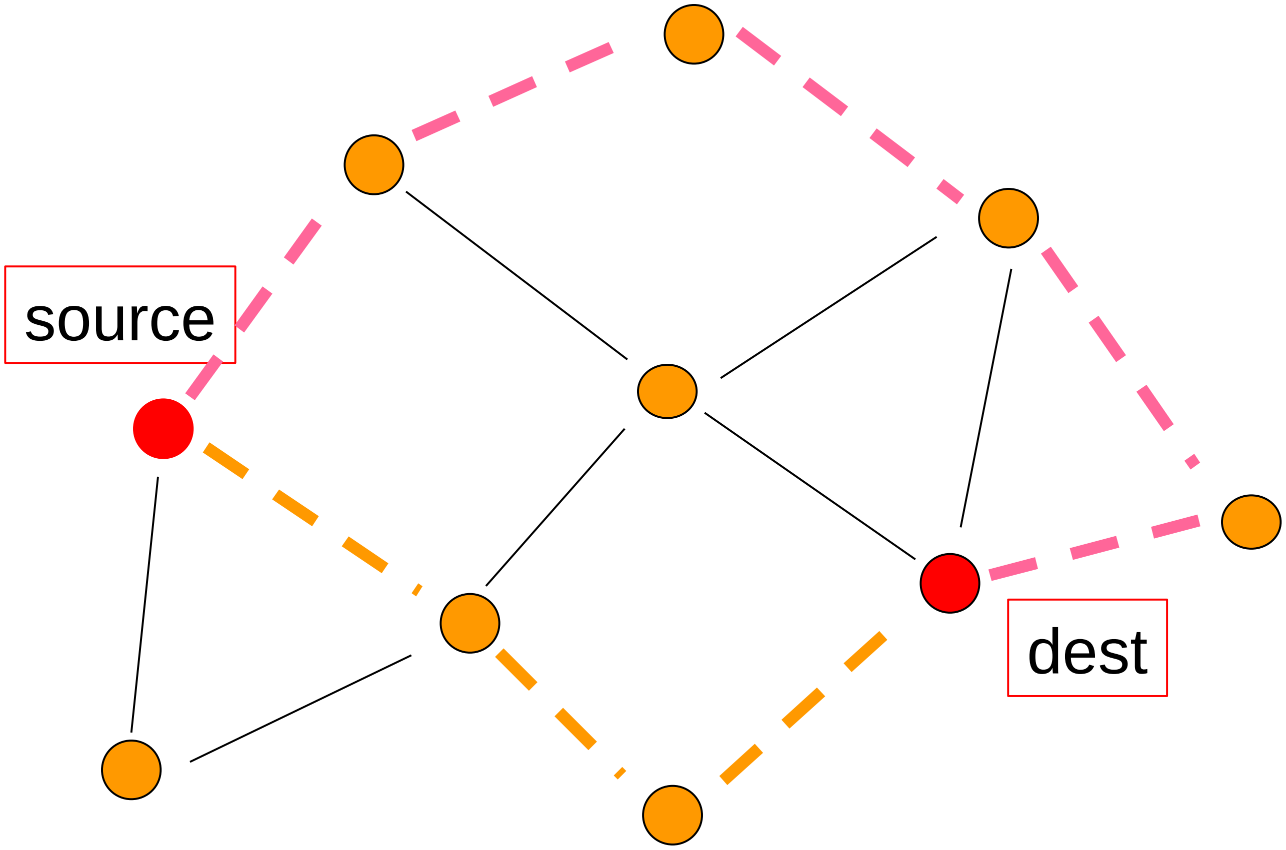
- Processing a RREQ
  - The node sets up a *reverse route* entry for the source node in its routing table
    - IP source and sequence number, number of hops to the source, IP of the neighbor from which request has been received
    - In this way, the node can forward back a RREP if it receives one later on
    - Reverse route entry has a lifetime, if not used for lifetime is deleted to prevent stale info hanging around

Example: Route Discovery



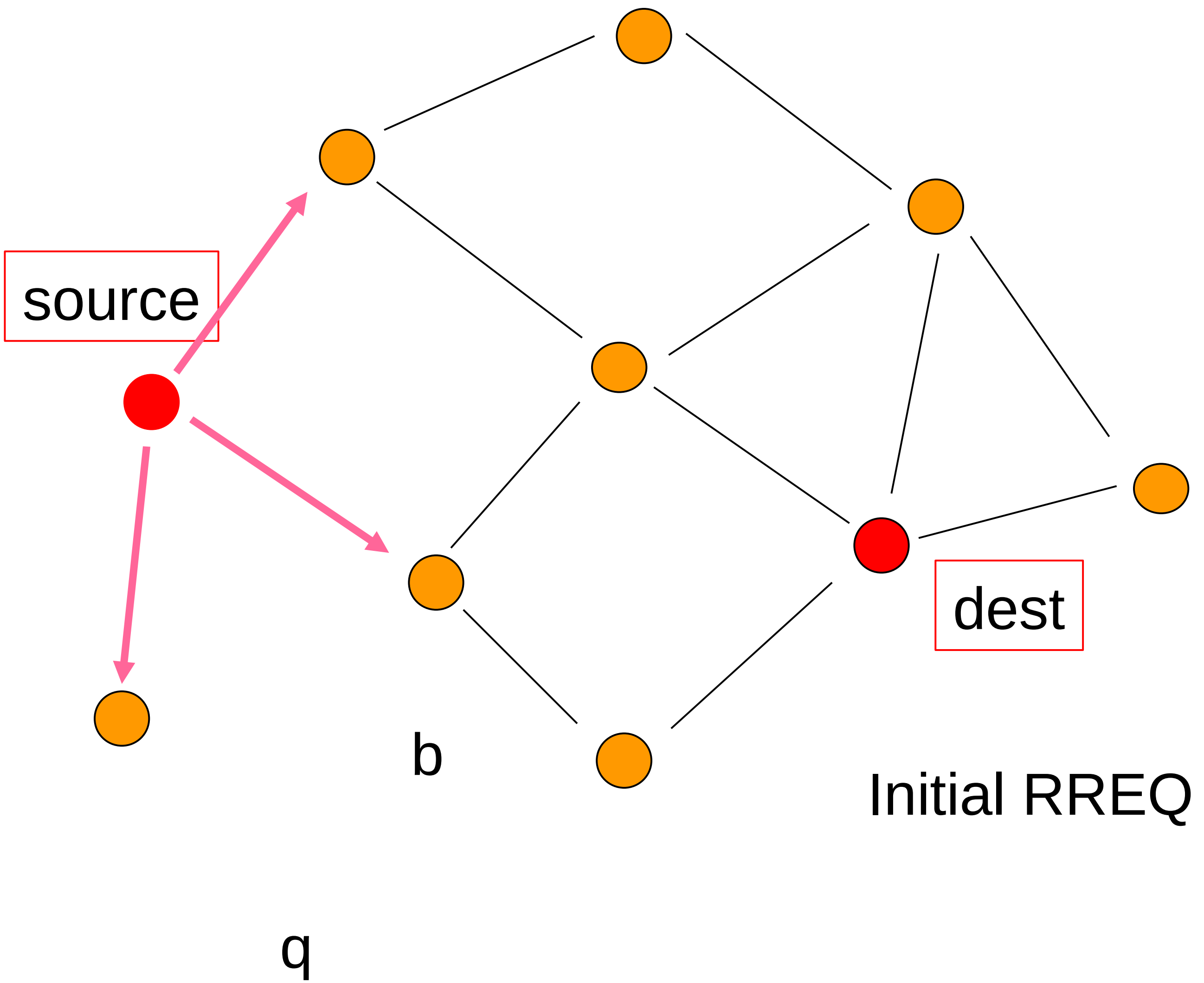
Actual network connectivity

Example: Route Discovery (2)



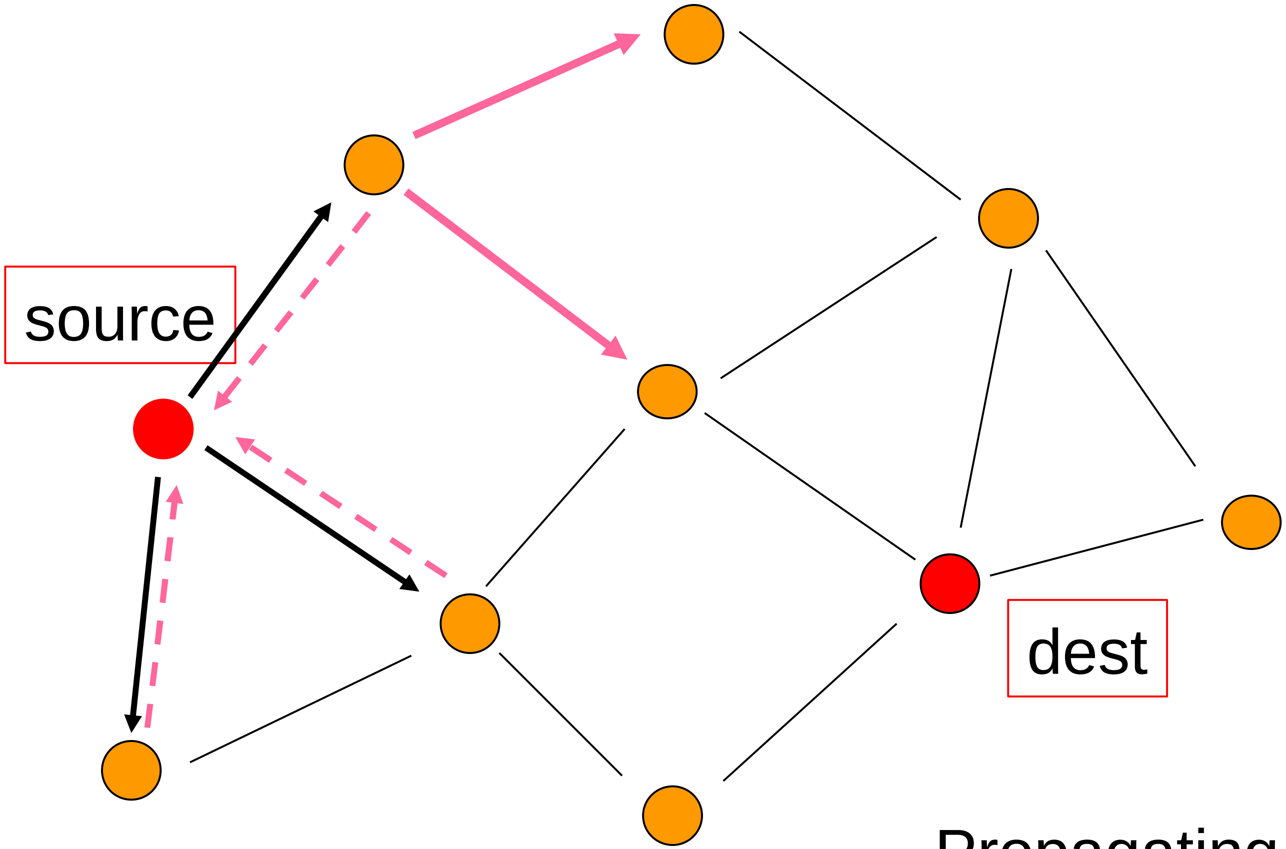
Many possible routes

Example: Route Discovery (3)



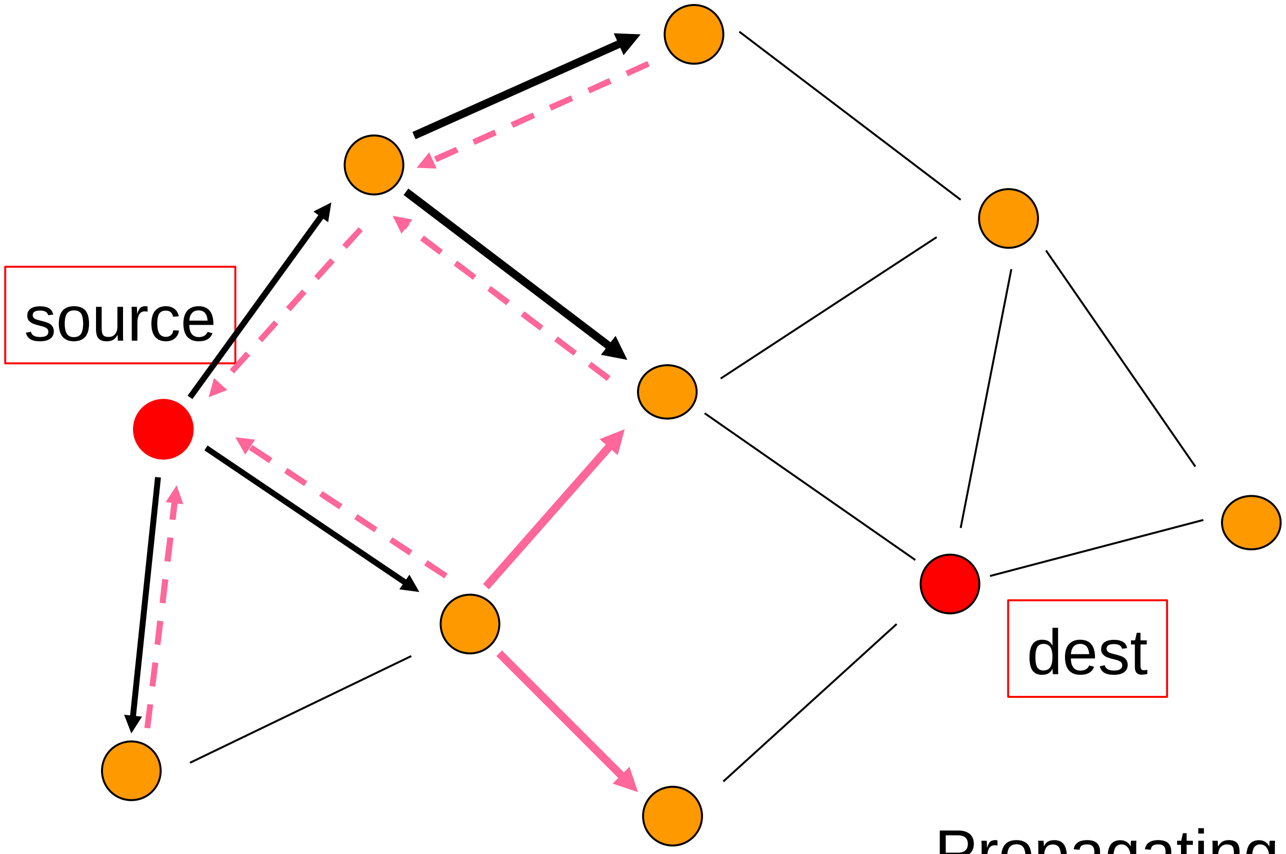
Initial RREQ broadcast

Example: Route Discovery (4)



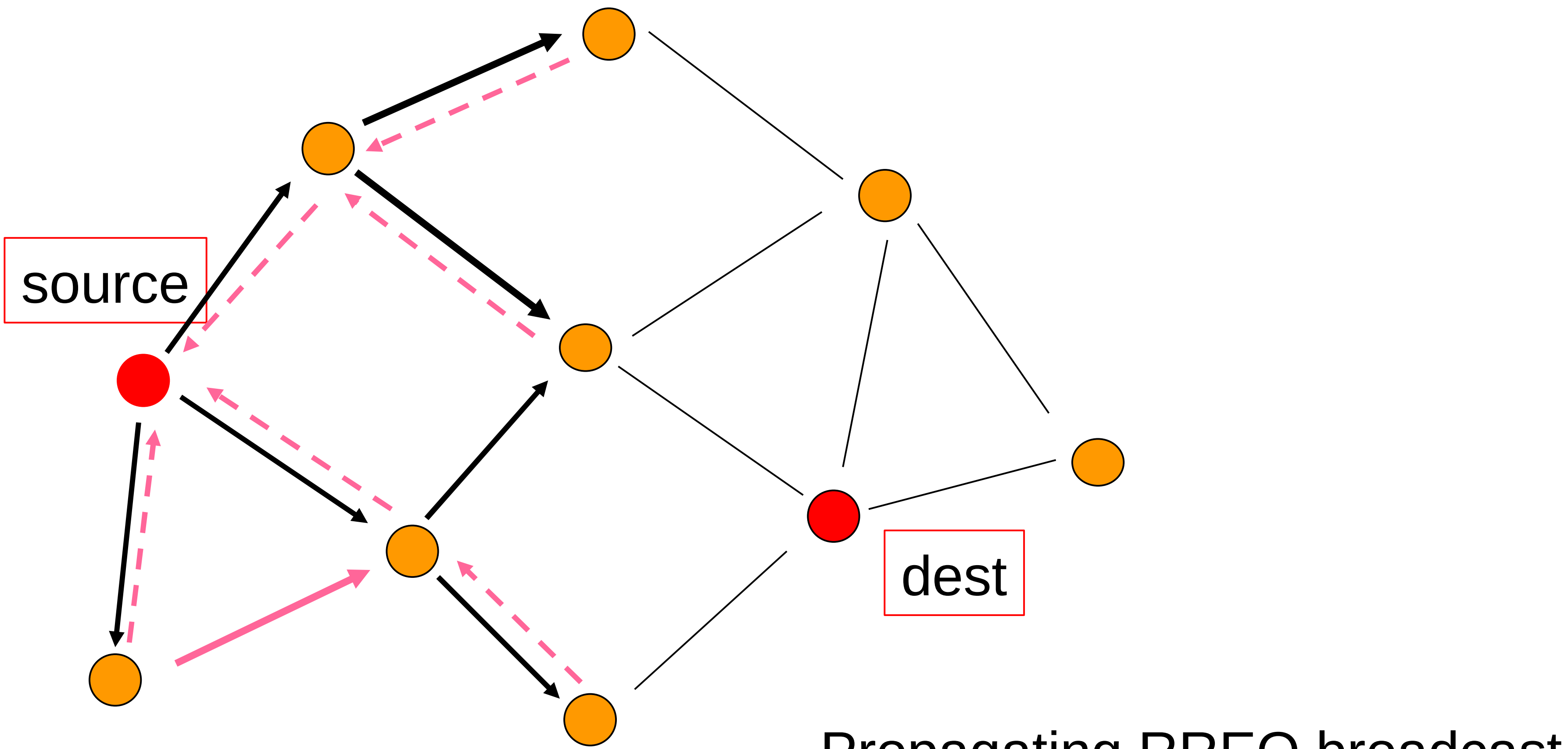
Propagating RREQ broadcast

Example: Route Discovery (5)

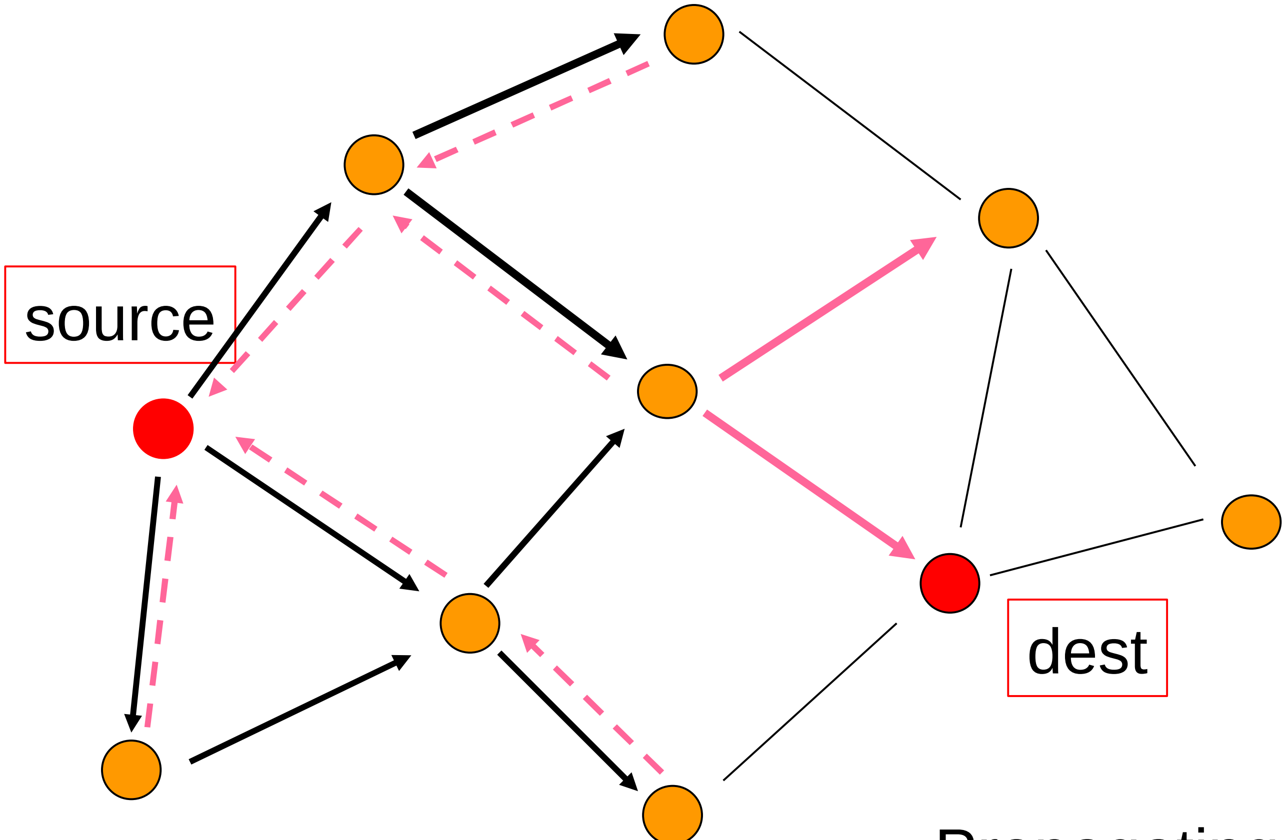


Propagating RREQ broadcast

Example: Route Discovery (6)

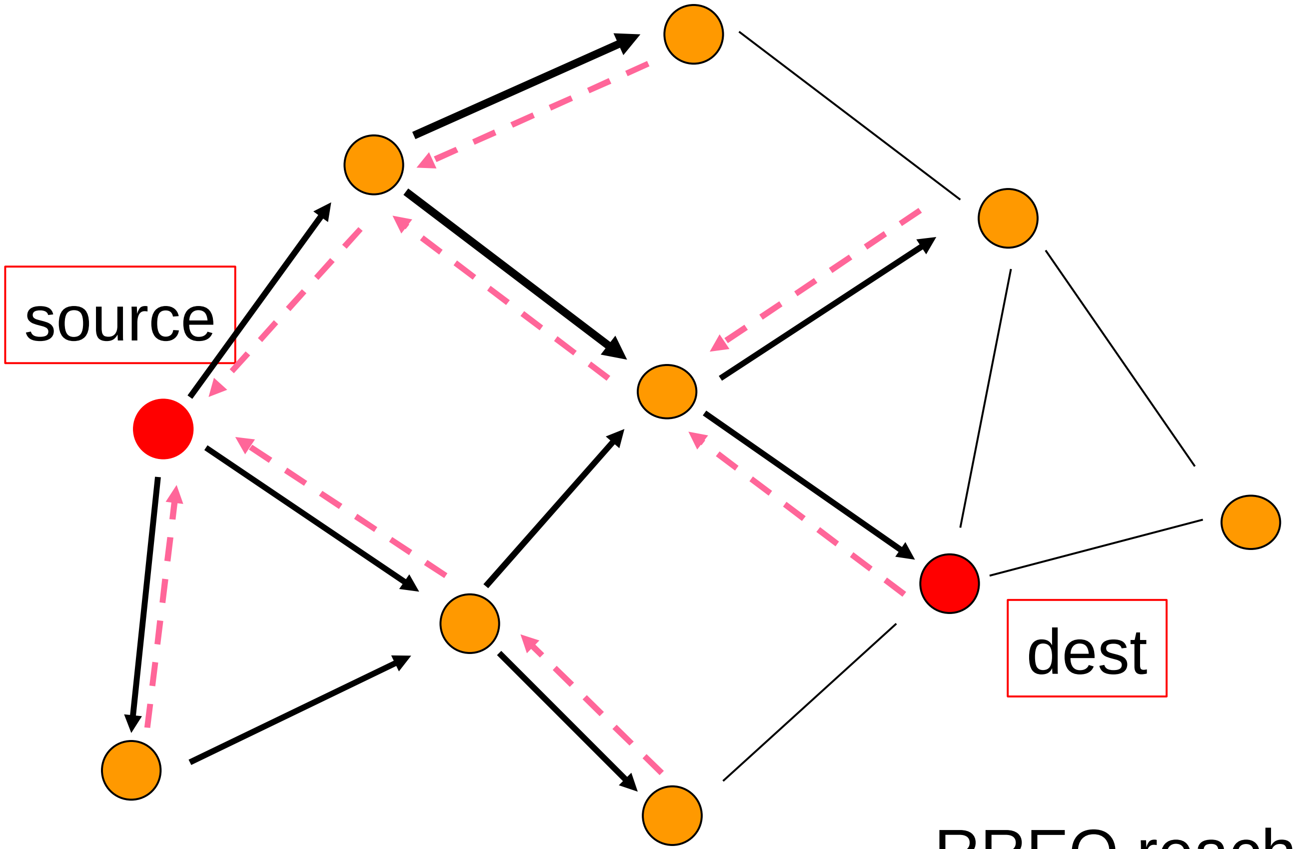


Example: Route Discovery (7)



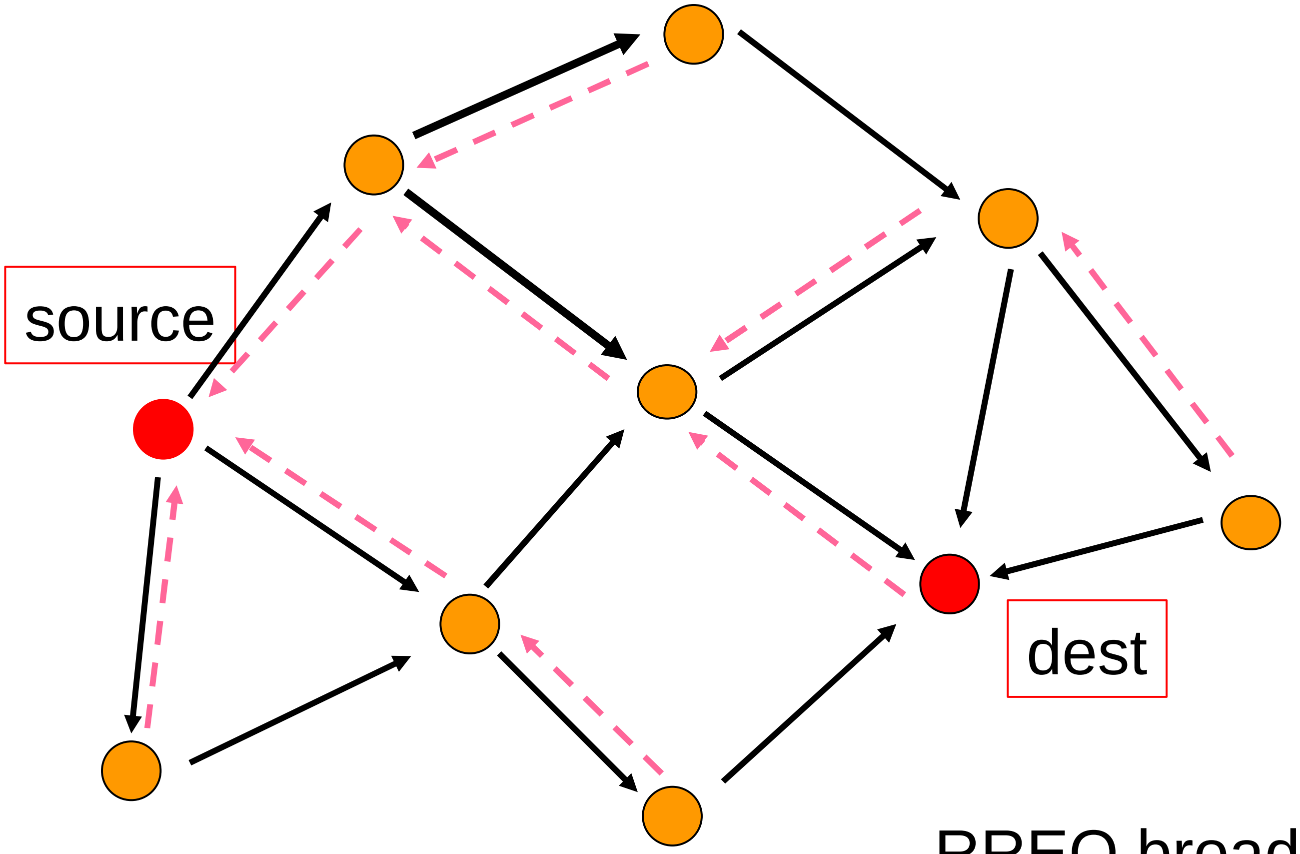
Propagating RREQ broadcast

Example: Route Discovery (8)



RREQ reaches deatination

Example: Route Discovery (9)



RREQ broadcast :
Possible completion

AODV: Route Discovery (4)

- Responding to a RREQ
  - To respond, a node must have an unexpired entry for the destination in its route table
    - The sequence number associated with that detsination must be at least as great as the *destination sequence number* included in the RREQ
      - Loop prevention: the route returned is never old enough to point to a previous intermediate node (otherwise the previous node would have responded to the RREQ before)
  - If the node is able to respond it unicasts a RREP back to the source using reverse route entries

AODV: Route Discovery (5)

- Responding to a RREQ (contd.)
  - If a node is not able to respond to RREQ it increments hop count in RREQ and broadcasts the packet to neighbors
  - Destination is always able to respond!
  - If a RREQ is lost, the source node can try **`rreq_retries`** additional attempts

AODV: Expanding ring search

- Flooding all the network may be expensive
  - Set TTL (Time To Live) to initial value **`ttl_start`** to simulate expanding rings of research across the network
  - If no answer is seen next time increments TTL
  - After a number of trials RREQ broadcasted across all the network for **`rreq_retries`** attempts
  - When a route is established the distance to the destination is recorded to set the initial TTL in the next RD for the same destination

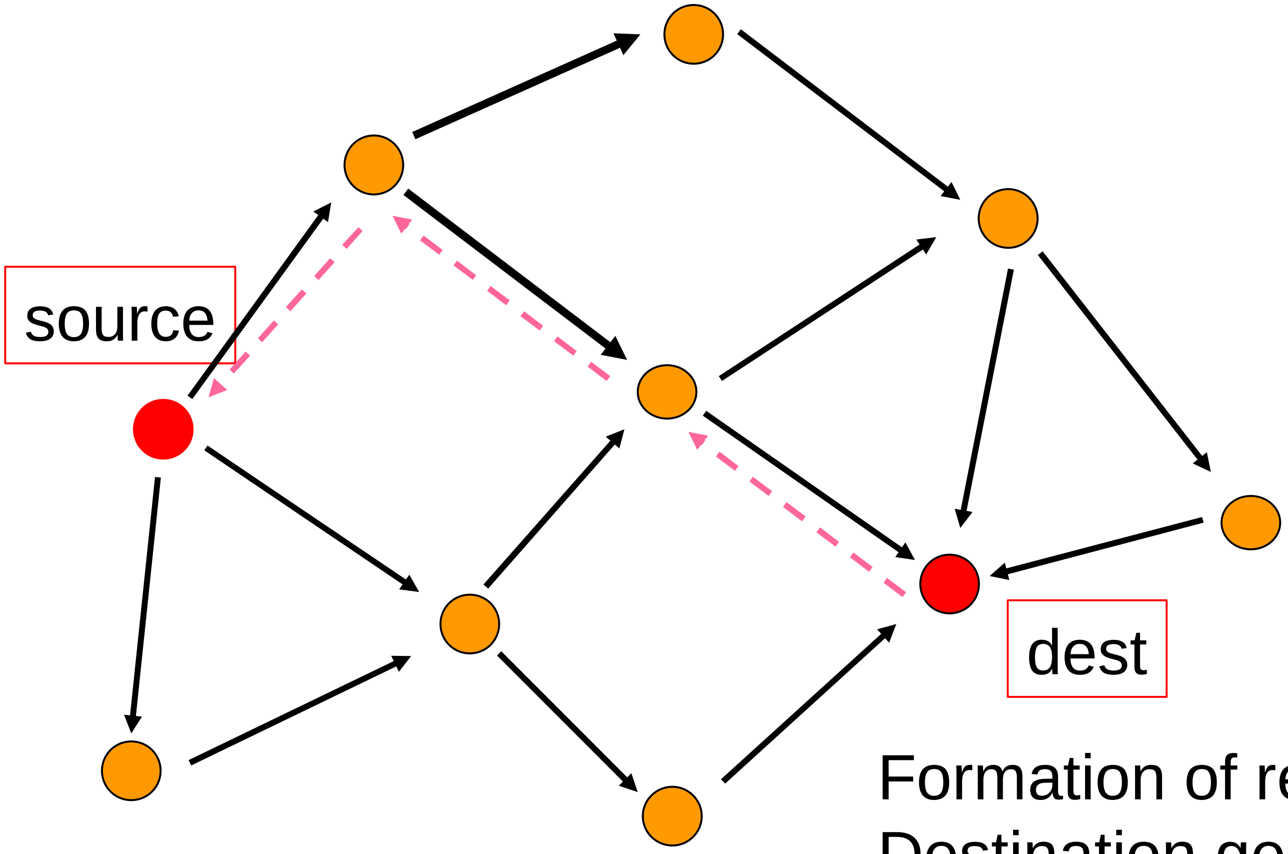
AODV: Forward Path Setup

- Generation of a RREP message
  - RREP contains the IP address of source and destination, destination sequence number, route lifetime and hop count
  - If destination is responding:
    - hop count = 0
    - Sequence number is its current sequence
  - If an intermediate node is responding
    - Hops count is its distance from destination
    - Sequnce number is last known sequence number for dest

AODV: Forward Path Setup (2)

- When an intermediate node receives a RREP
  - Sets up a *forward path* entry in the RT for the destination
    - IP addrees of destination, IP address of the node from which RREP arrived, hop count (distance) to the destination, lifetime
    - Distance is computed adding 1 to the hop count in RREP
    - Lifetime is taken from RREP
  - Forwards RREP to the source

Example: Route Discovery (10)



Formation of reverse route:
Destination generates RREP
And forwards using reverse route
Entries in RT

AODV: Forward Path Setup (3)

- When more RREP are received
  - A new RREP is forwarded only if
    - Destination seq number is greater
    - Hop count is smaller
  - Otherwise packet is discarded
- The source node can use the first RREP to start transmission
  - Subsequent RREP are used to update RT for subsequent transmission to the same destination

AODV: Route Maintenance

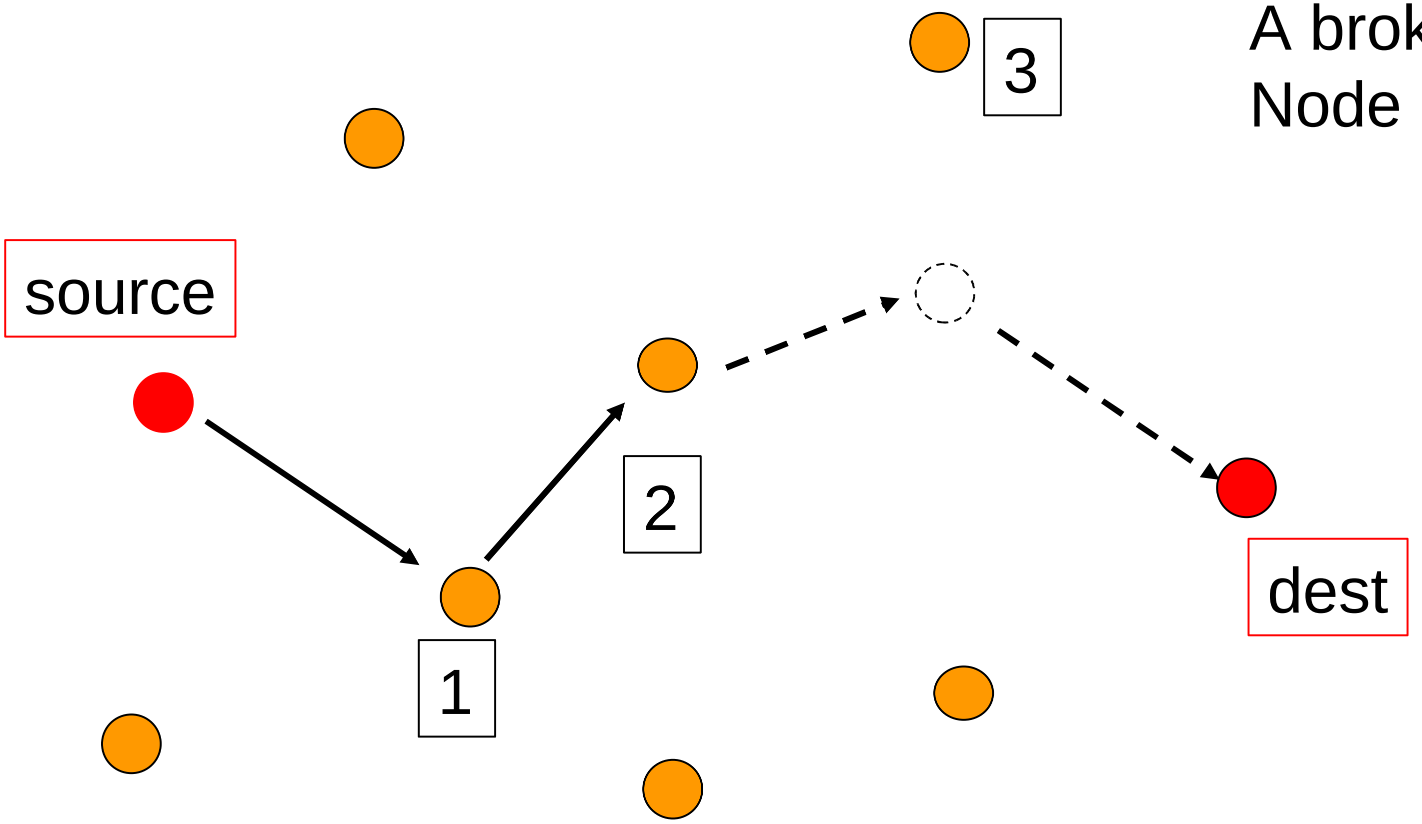
- Each node maintains only *active paths*
  - Active paths corrspond to the routes in use
  - *active paths* that include a node which has moved need maintenance
- How we know a node has moved
  - If source moves it is trivial, it knows it must start route discovery again
  - Otherwise we receive a RERR packet while we try to use a route
    - Packets for broken routes are stopped and a RERR generated back

AODV: Route Maintenance (2)

- Dealing wit ha RERR packet
  - A RERR packets includes a list of now unreacheable destination due to a broken link
  - Forwards RERR to all precursor in routes from source
  - Nodes receiving a RERR for a route, mark it as invalid setting distance to the destination equal to infinity and in turn propagate to precursors
  - When source node receives a RERR packet it starts RD again

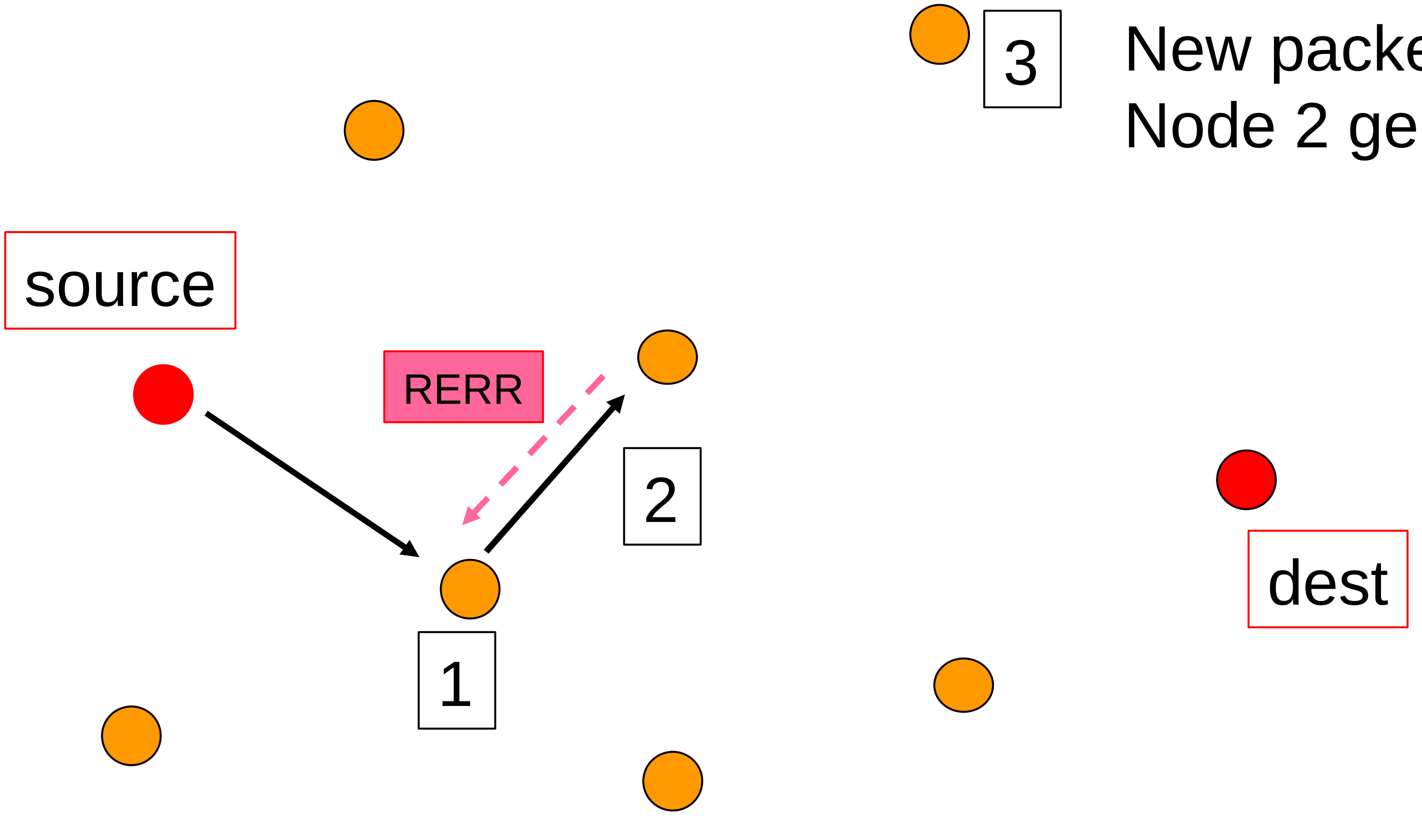
Example: Route Maintenance

A broken route:
Node 3 has moved

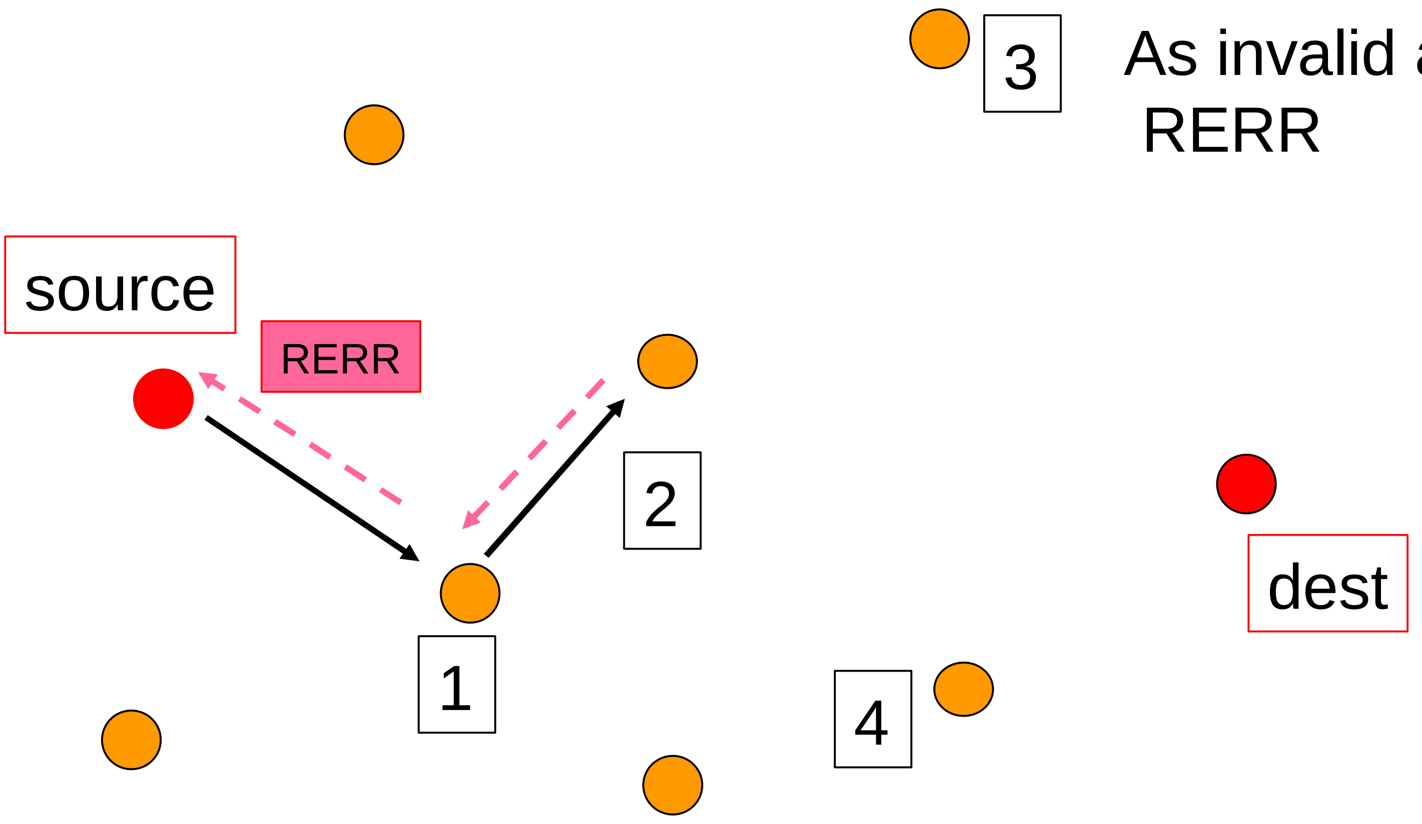


Example: Route Maintenance (2)

Sources sends a
New packet to dest
Node 2 generates RERR



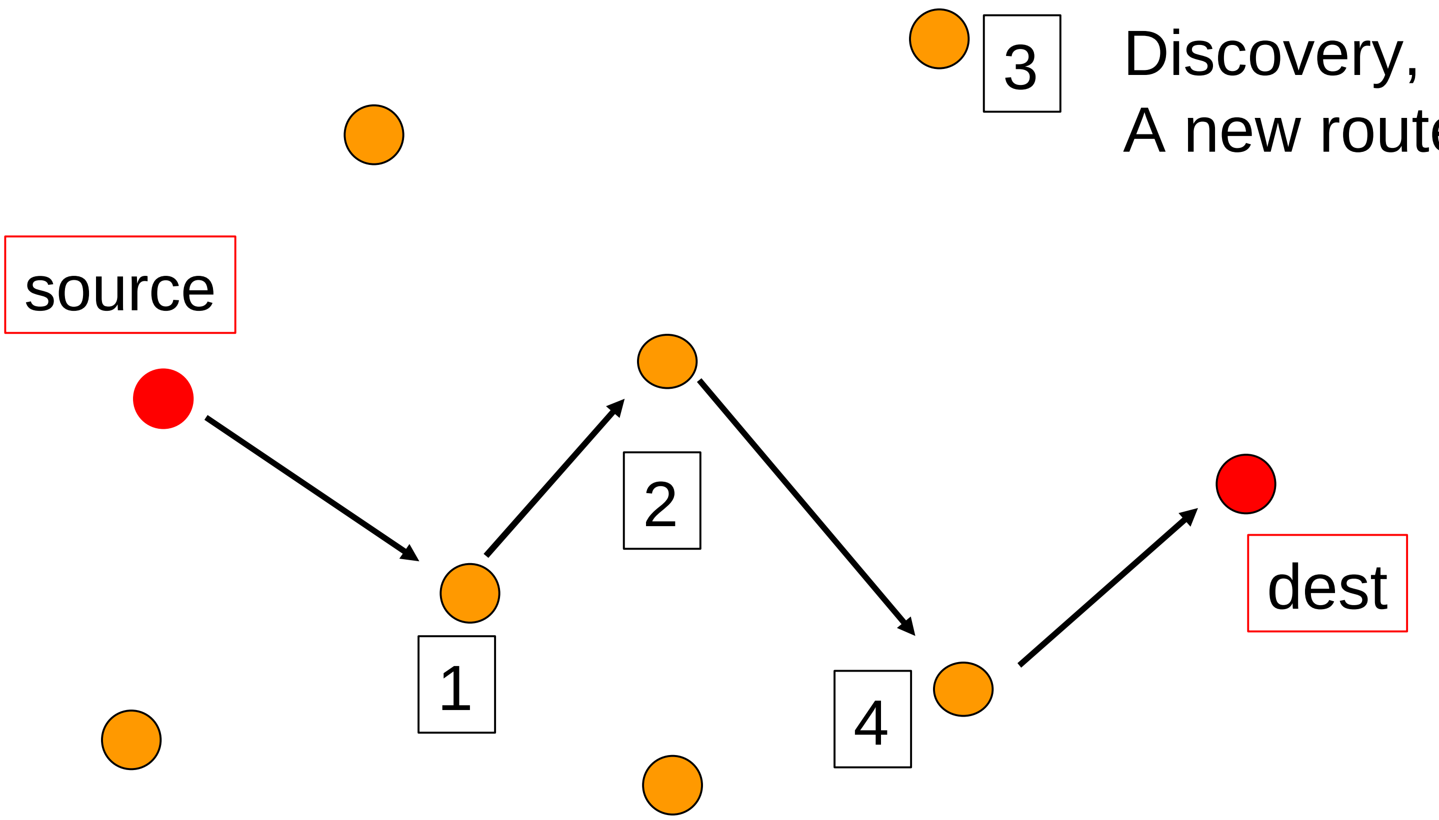
Example: Route Maintenance (3)



Node 1 marks route
As invalid and forwerds
 RERR

Example: Route Maintenance (4)

Source starts route
Discovery, resulting in
A new route



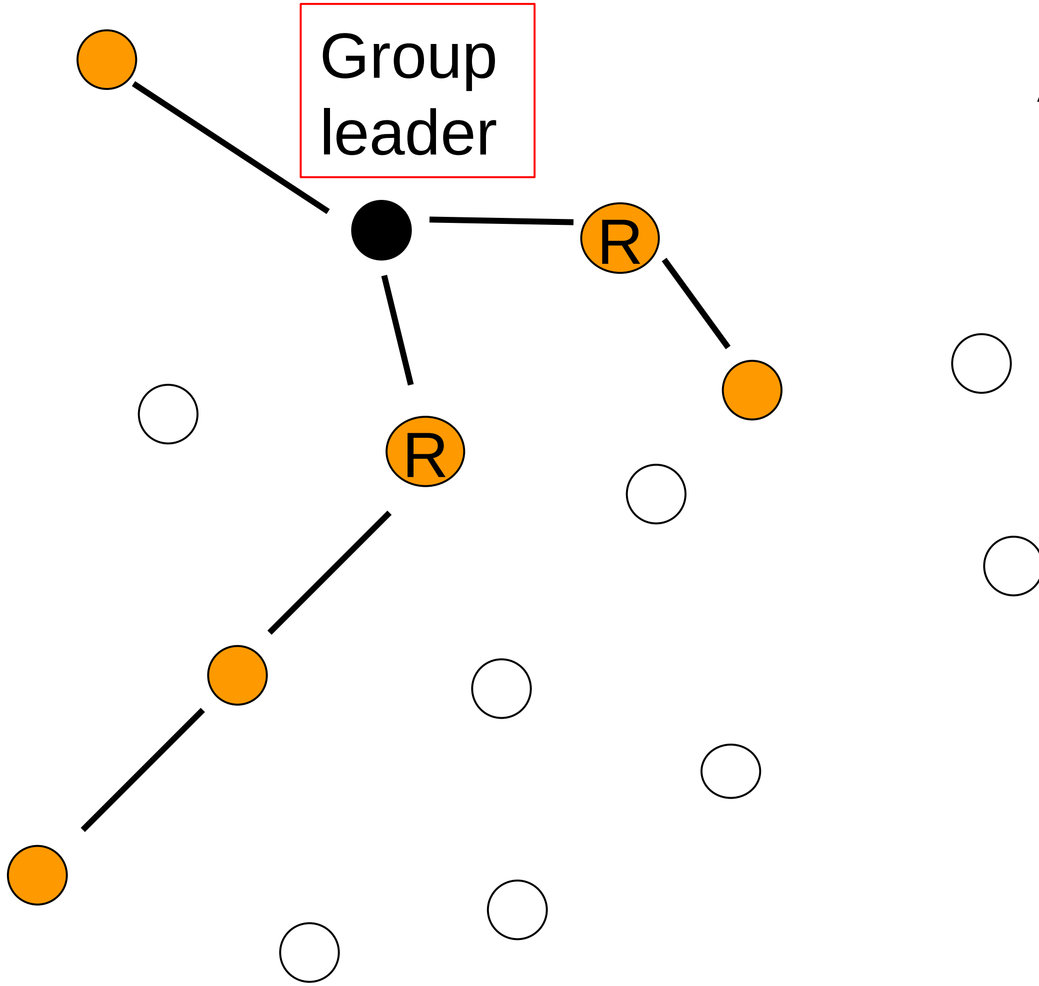
AODV: Route Maintenance (3)

- Connectivity management
  - Each time a broadcast from a neighbor is received lifetime of the route to neighbor is updated
  - If neighbor is not in table entry is added
  - Periodically a Hello packet is sent to inform neighbors we are still alive and to trigger RT update
    - Special RREP with IP address and sequence number of sender. TTL is 1 to prevent resend.
  - When no Hallo messages are received for a period of time node is considered gone and connectivity updated

AODV: Multicast

- Multicast groups
  - Each MG has a leader and a bidirectional multicast tree
  - Each MG has a sequnce number
    - maintained by the leader
  - Nodes can join and leave a group any time
  - Groups use RREQ and RREP
- Multicast Route table
  - Records routes for multicast groups

Example: Multicast group



A multicast group

AODV: Multicast RD

- Multicast route discovery
  - Started when a node wishes to join a group or to send data to a group
  - Node creates a RREQ packet wit join flag set if it wants to join
    - RREQ includes known sequence number for group
  - For join, only members of the multicast tree (router nodes) can respond
  - Otherwise any node with knowledge of a route can respond

AODV: Multicast join RD

- For a join RREQ
  - Nodes not belonging to the group receiving a join create a reverse route entry in the MRT and broadcasts the request to its neighbors
  - Non routing nodes add an unactivate entry for the source node in the MRT,
    - until link is enabled (activated) no message is forwarded for the group

AODV: Multicast join RD (2)

- Join -- Forward Path Setup
  - A routing node may answer a join RREQ only if its recorded sequence number is greater than the one in the RREQ
  - The group leader can always reply
  - The responding node updates MRT placing requesting node next hop information and sending back a RREP to the source.
  - Nodes along the path to the source set up a forward path entry for the multicast group in the MRT, incrementing hop count as usual

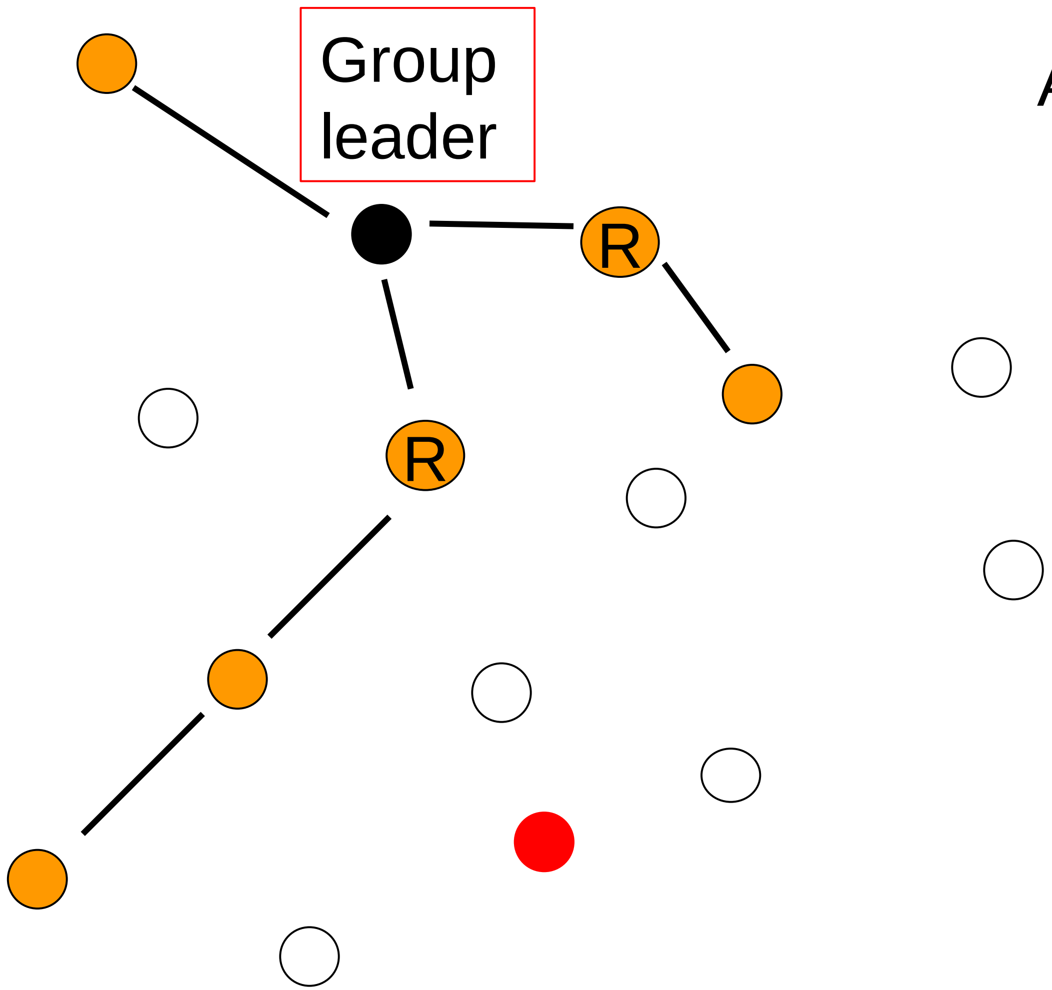
AODV: Multicast join RD (3)

- Route Activation
  - Different RREP may reach the source, only one path need to be selected to connect the join node to the tree
    - The route with highest sequence number and mimumum hop count is selected
    - This route is selected by unicasting a multicast activation message (MACT) to activate the corresponding entry in MRTs
- Leaving the tree
  - A non leaf leaving the tree, must still work as router for the others

AODV: Multicast join RD (4)

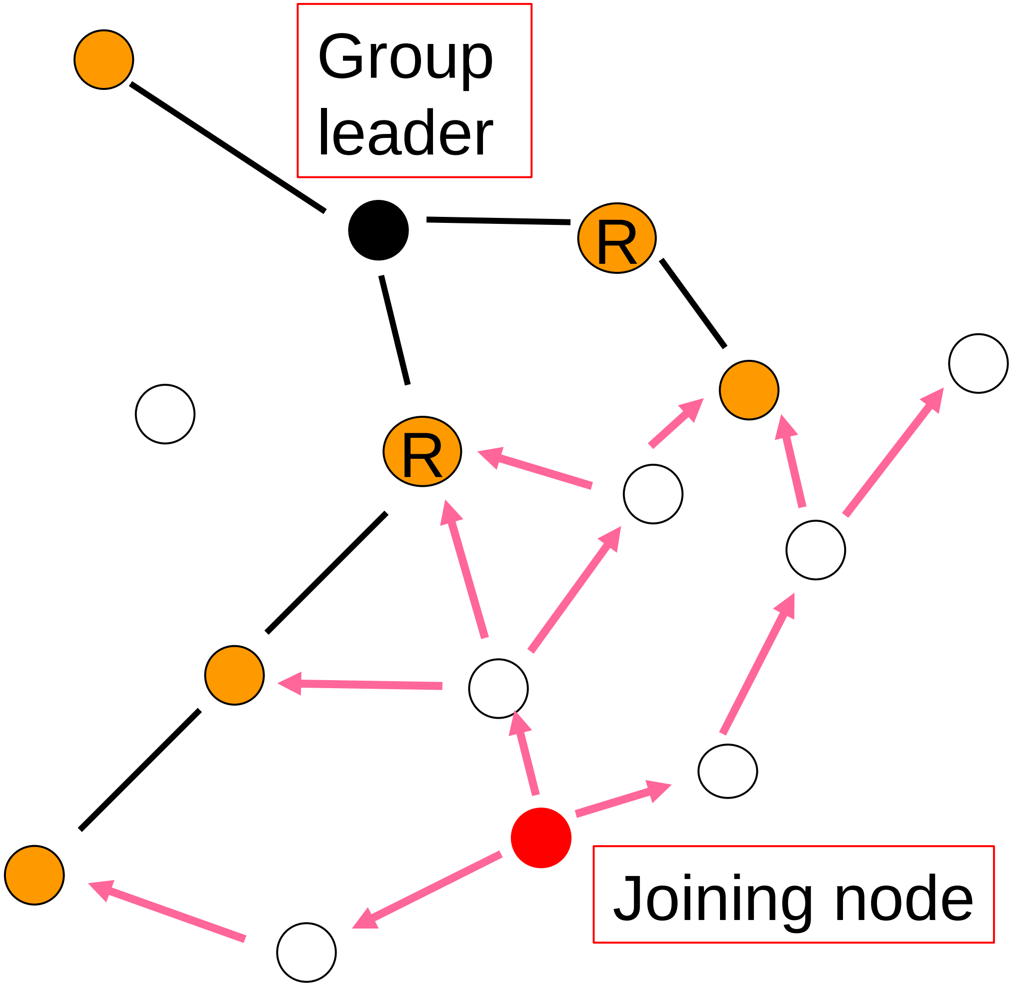
- Leaving the tree
  - A leaf node may leave the tree simply sending a message to its parent, if this is not part of the tree and it is now a leaf it can propagate pruning at the upper levels
  - A non leaf node leaving the tree, must still work as router for the others

Example: Join a multicast group



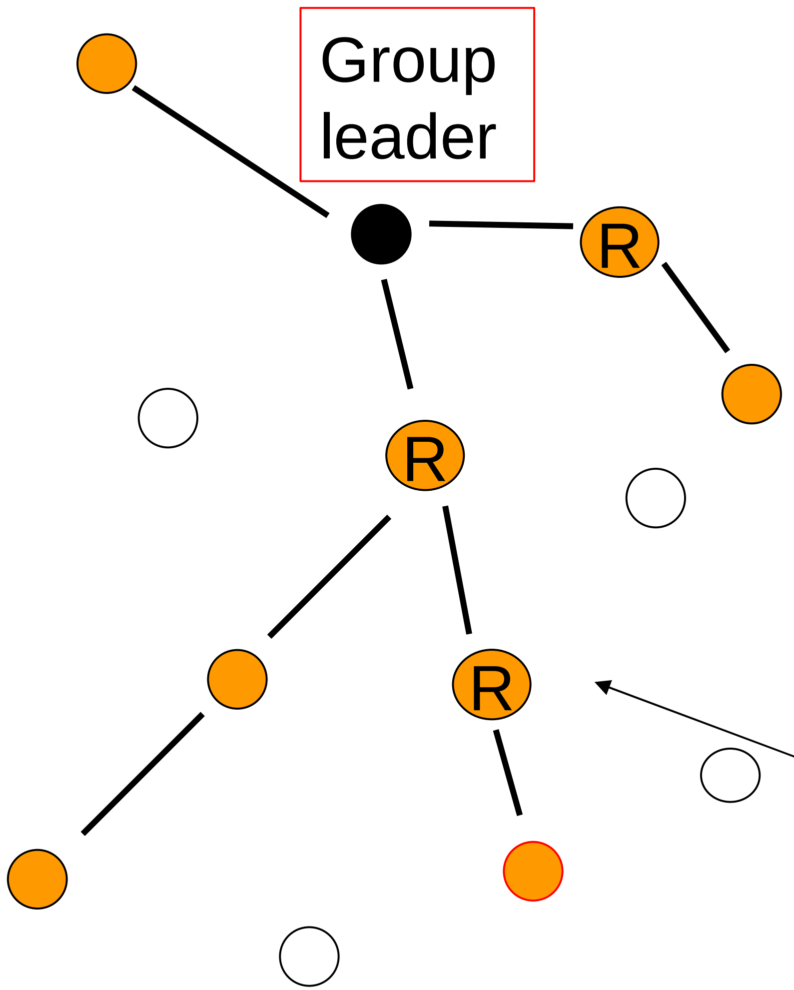
A multicast group

Example: Join a multicast group (2)



RREQ prpagation
Of a joining node

Example: Join a multicast group (3)



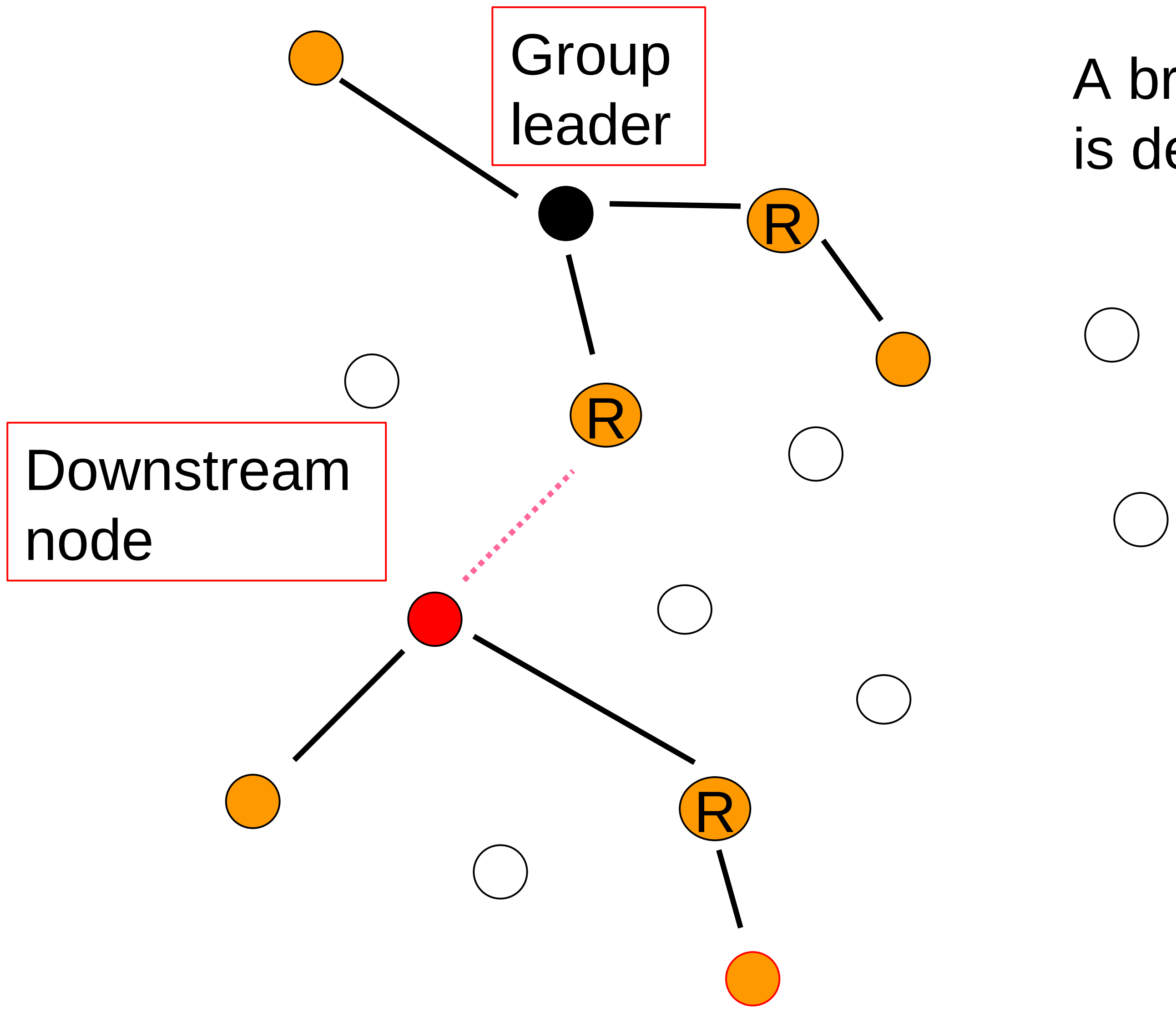
Multicast tree branch addition

This node is not part Of the MT, but it acts as router

AODV: Multicast Route Maintenance

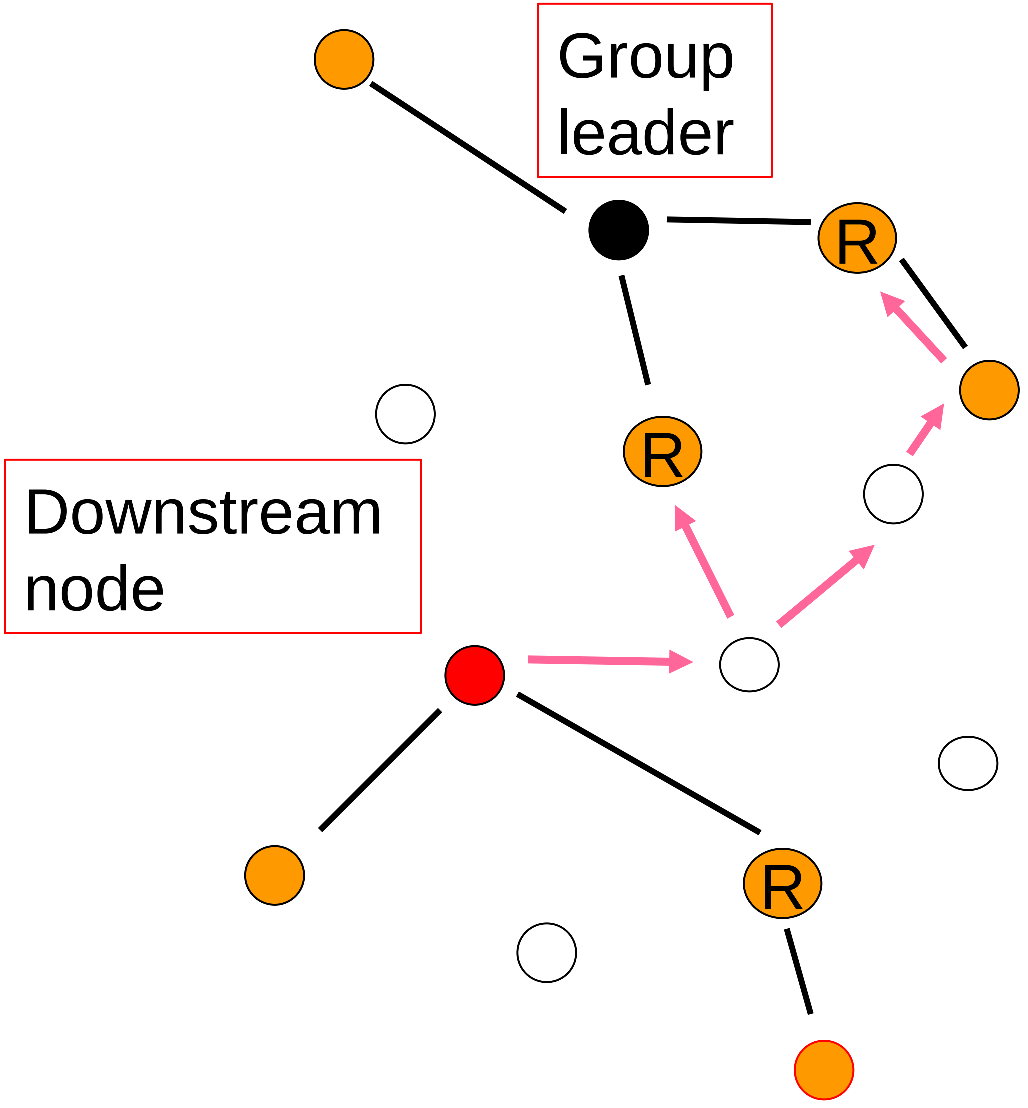
- It must be done as soon as the fault is discovered
  - Nodes in the group must stay connected aven if no messeges are sent to them
  - The downstream node (fartest from the leader) is responsible to repair the link, using a RREQ to join the group again
  - RREQ includes node distance from the group leader and only nodes that far or more can reply to repair
    - This is to involve the “right” nodes, the ones on the group leader side

Example: Broken link



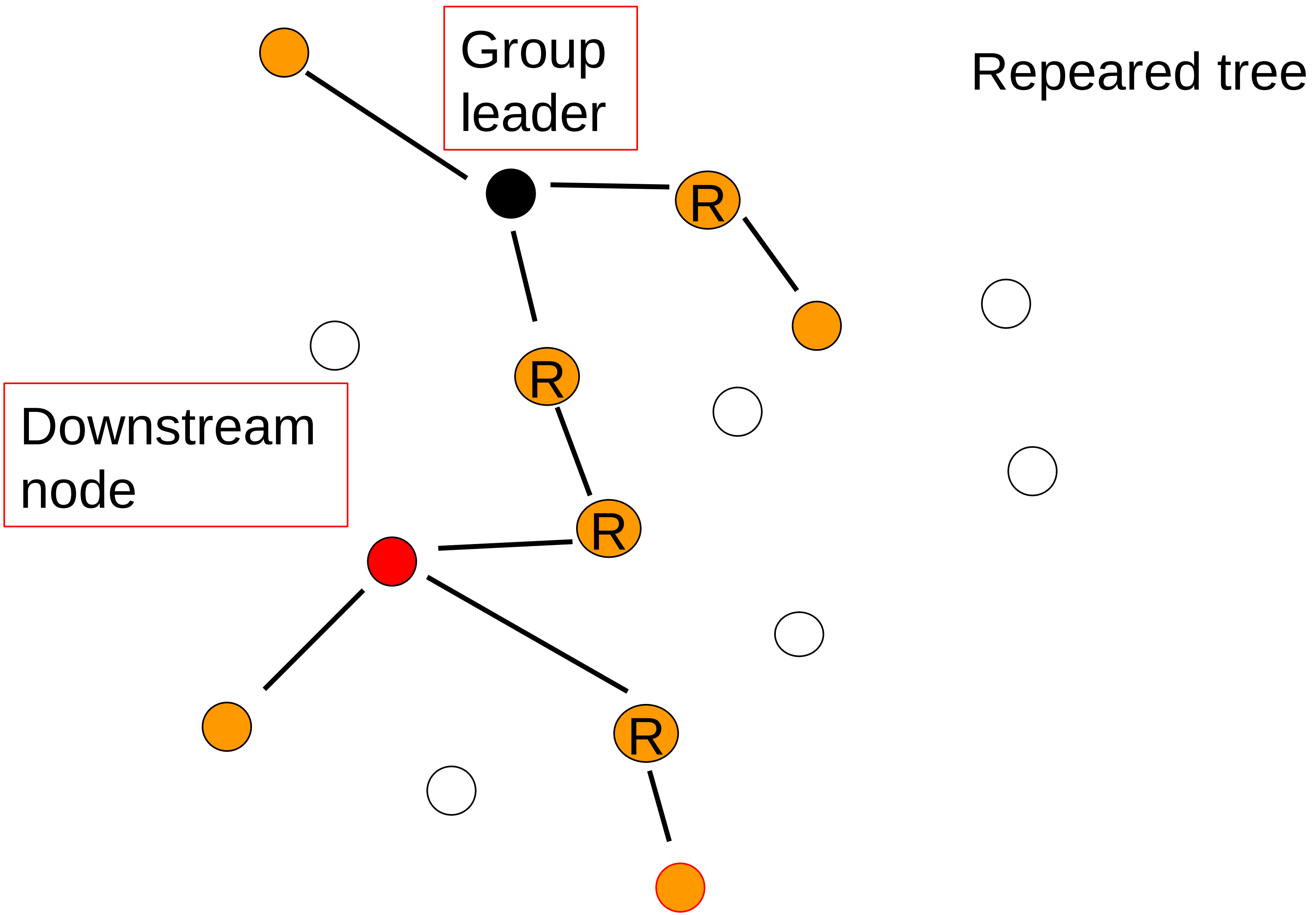
A broken link is detected

Example: Broken link (2)



Join RREQ
Hop count set to 2

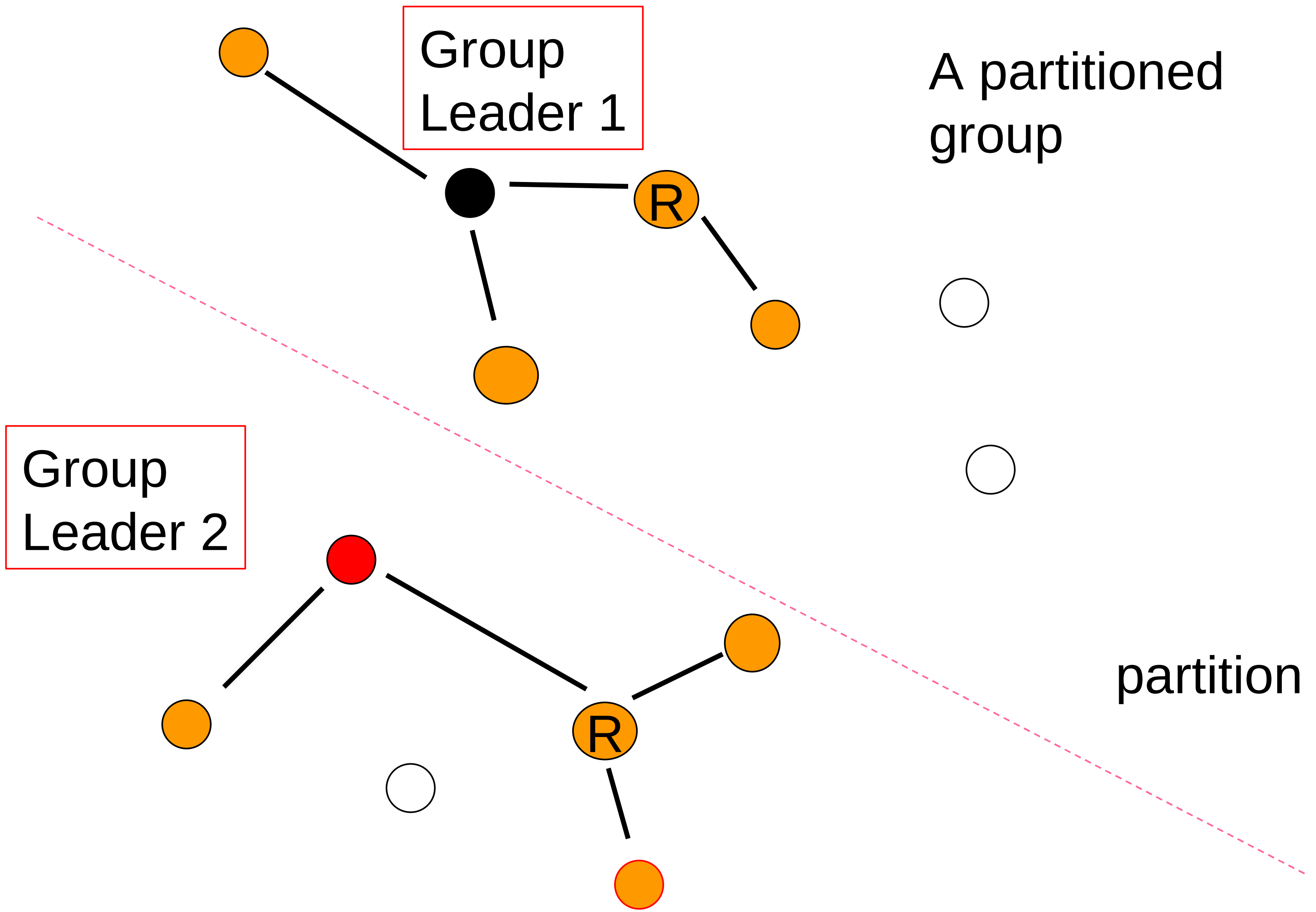
Example: Broken link (3)



AODV: Multicast RM (2)

- If the broken link cannot be repaired
  - The two parts of group remain partitioned
  - A new leader must be elected
  - If down stream is part of the tree it becomes the leader
  - If its is simply a router non part of MT, it sends a message to the other members to select the new leader

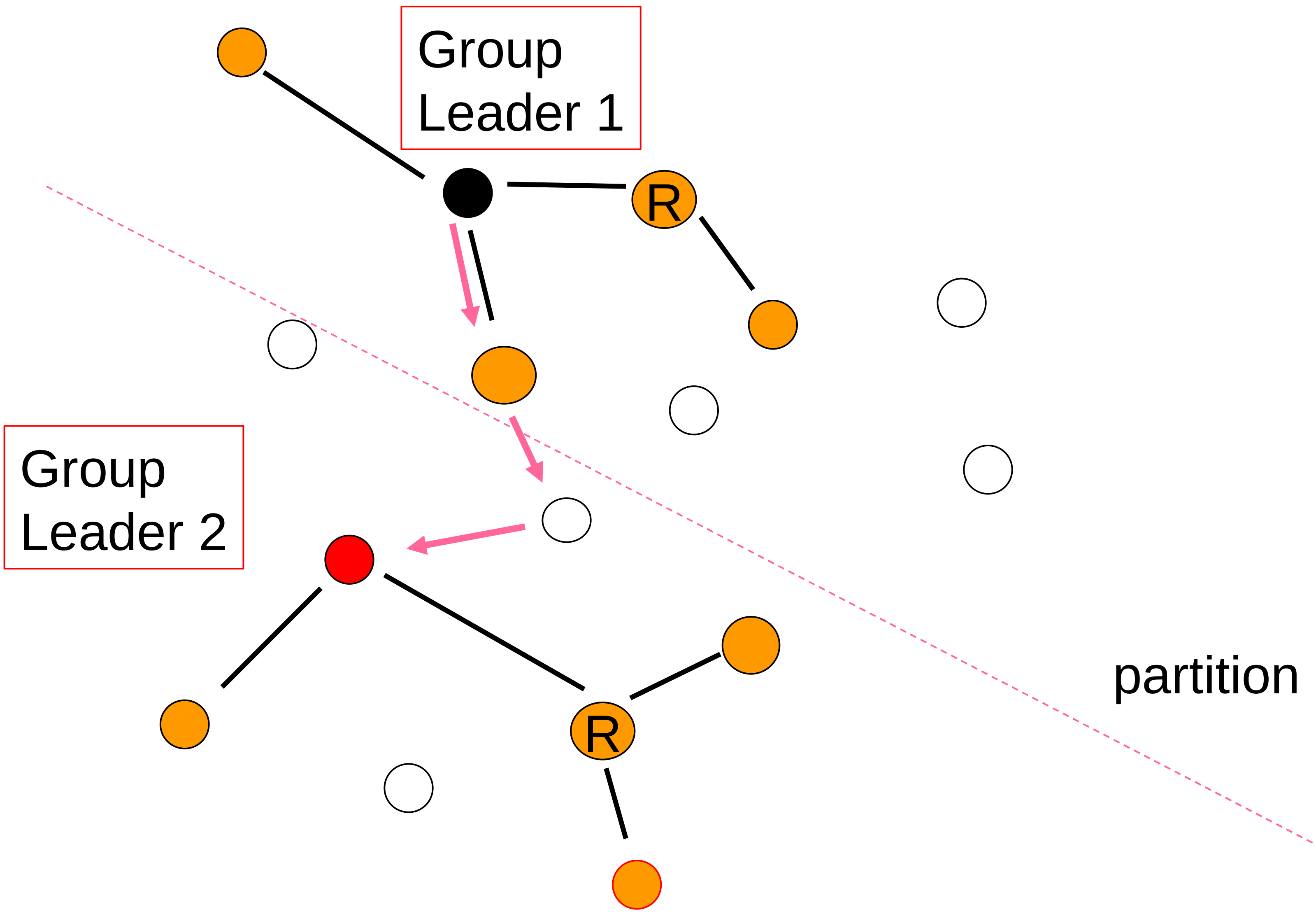
Example: Partitioned Group



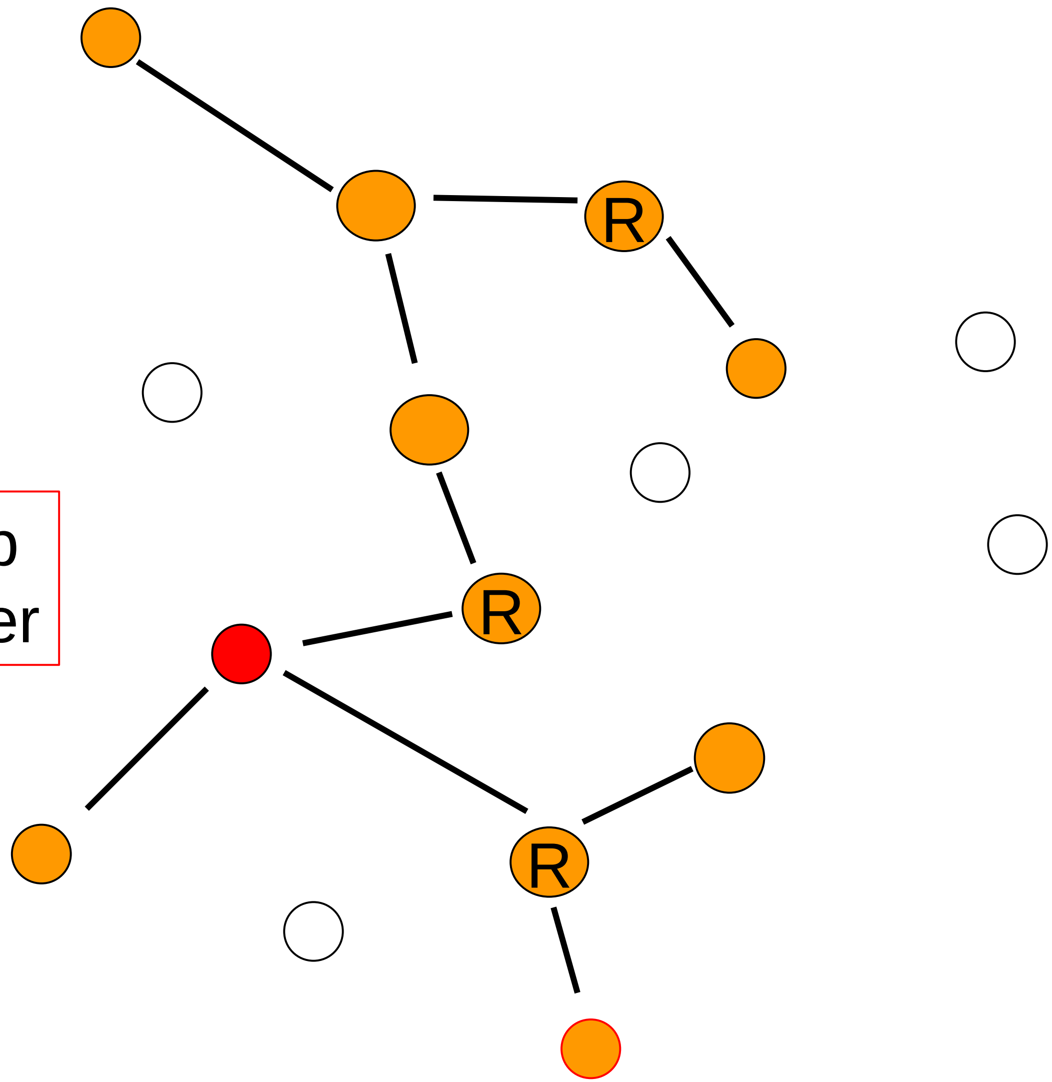
AODV: Multicast RM (3)

- Reconnecting a partitioned group
  - Group leaders broadcast a periodic GRPH (Group Hallo) to the network
    - GRPH includes IP of group leader and sequence number
  - If a leader heards a GPRH with another leader the two groups are within each other RR
    - They must e reconnected
    - GL with lower IP address starts procedure
    - Uses RREQ with repair flag set on

Example: Partitioned Group (2)



Example: Partitioned Group (3)



Group Leader

AODV: Packets

- AODV packets are standard IP packets
  - Uses standard IP fields such as source and destination address, TTL for hop counting
  - Details on RFC

AODV-DSR: Comparison

- Many studies in the literature
- DSR
  - Allows multiple routes
  - Supports unidirectional links
- AODV
  - Supports multicast

AODV-DSR: Comparison (2)

- With low traffic and low mobility
  - Both have an acceptable end-to-end delay, and small routing overhead (control packets)
- With high mobility, high traffic
  - AODV has an higher routing overhead due to control packets:
    - routes become congested and need to be rediscovered
  - DSR pays for multiple routes
    - With high mobility it is difficult to make sensible choices

AODV: References

[Perkins Royer 1999]

C.E. Perkins and E.M. Royer. *Ad-Hoc On-Demand Distance Vector Routing*. Proceedings of IEEE Mobile Computing Systems and Applications. Feb 1999. 90-100.

[Perkins Royer 2001]

C.E. Perkins and E.M. Royer. *Ad-Hoc On-Demand Distance Vector Routing*. Cap 6 of *Ad hoc networking* (C.E. Perkins Ed.), Addison-Wesley, 2001.

[RFC 3561]

http://www.ietf.org/rfc/rfc3561.txt