IP Routing and MANET Routing Algorithms

By Quan Le-Trung, ***Dr.techn.***
http://sites.google.com/site/quanletrung/

Lecture Objectives

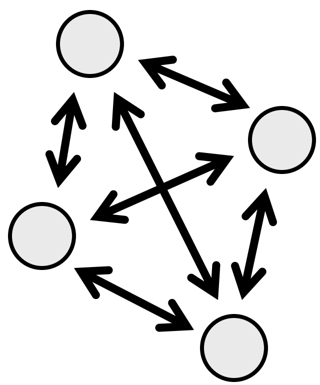
- **Present the basic principles of routing in general packet-switched networks**
- **Describe the basic principles of mobile ad hoc networks (MANETs) and MANET routing protocols**
- **Describe AODV and OLSR as example MANET routing protocols**
- **Discuss issues related to mobile ad hoc networks and MANET routing protocols**

Agenda

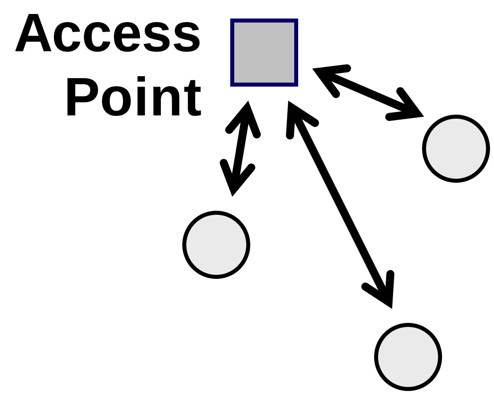
- Layer 2 routing
- Routing basics
  - Distance vector algorithms
  - Link-state algorithms
- Mobile ad hoc networks
- Example MANET routing protocols
  - OLSR
  - AODV

So far…

- **Nodes in a 802.11 basic service set or Bluetooth piconet are directly connected to each other**
- **There is no need for routing and IP (layer 3) provides essentially no functionality**



Ad Hoc Mode

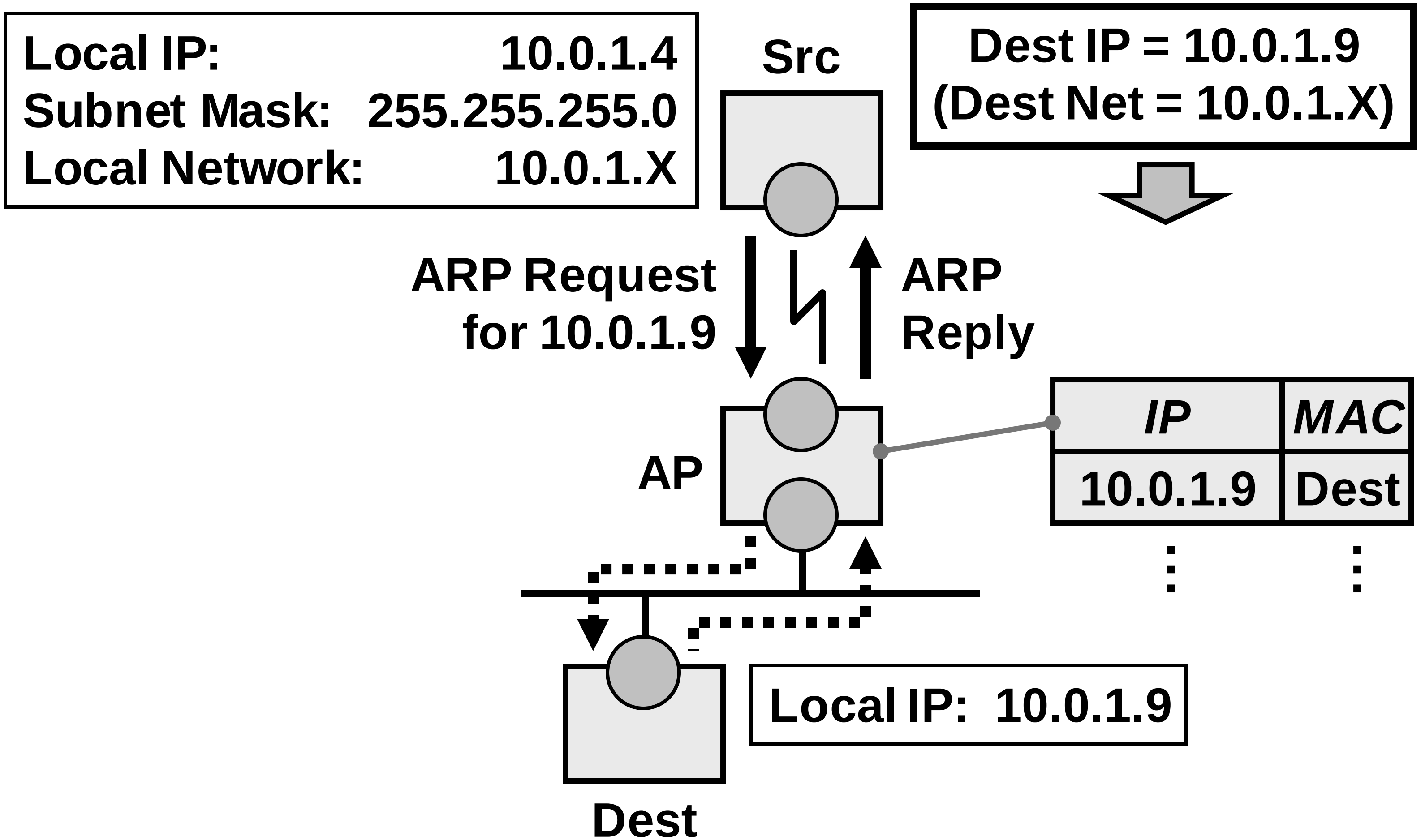


Infrastructure Mode

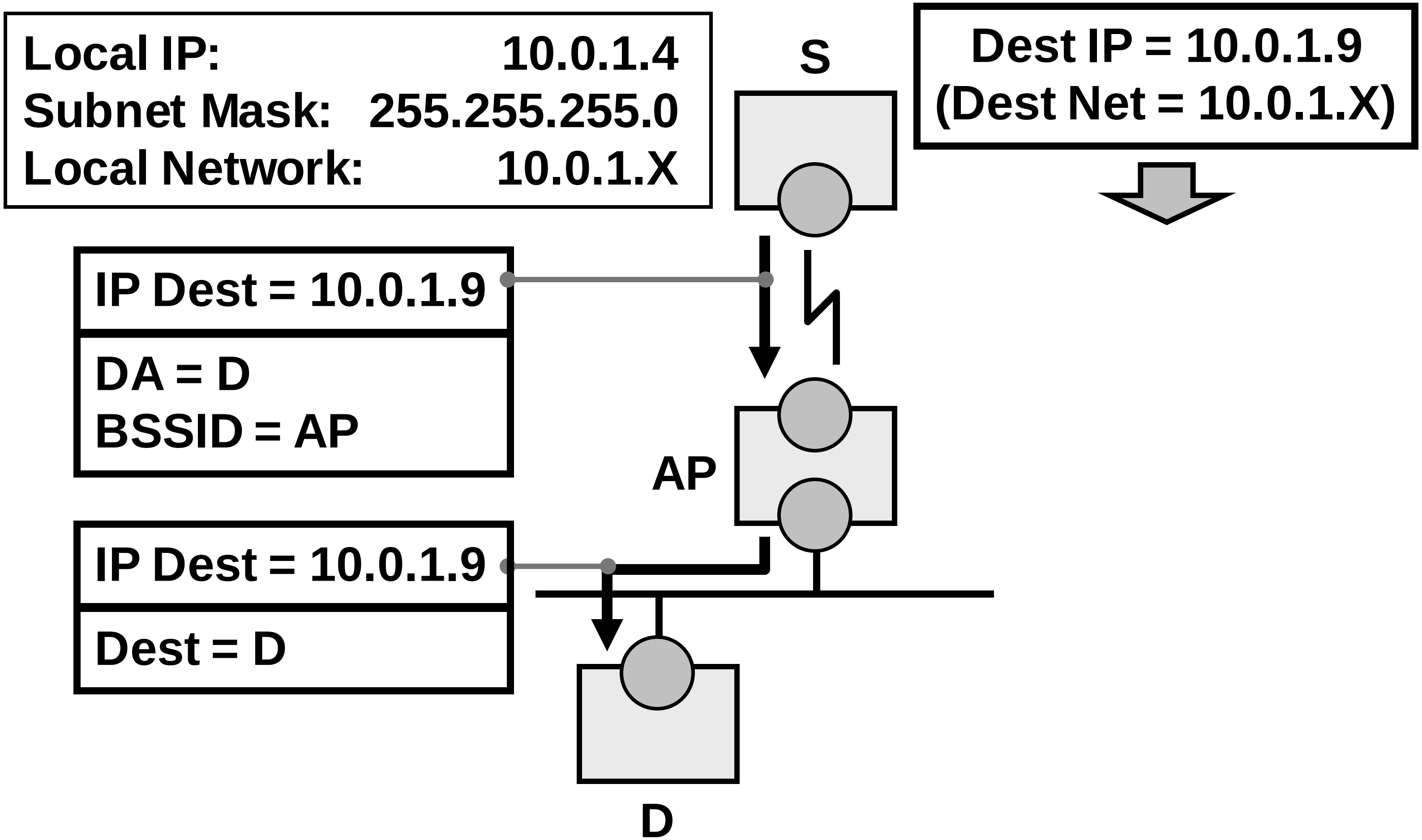
Layer 2 Routing (1)

- The source determines that the destination interface is in the same IP subnet
  - This necessarily implies that the source and destination are directly connected by a Layer 2 network
- ARP allows the source to determine the Layer 2 (MAC) address of the destination
- The source encapsulates the IP datagram in a Layer 2 frame, addresses the frame appropriately, and transmits the frame
- Layer 2 “interworking units” (e.g., Ethernet bridges, 802.11 APs) may need to perform some forwarding or routing functions

Layer 2 Routing (2)

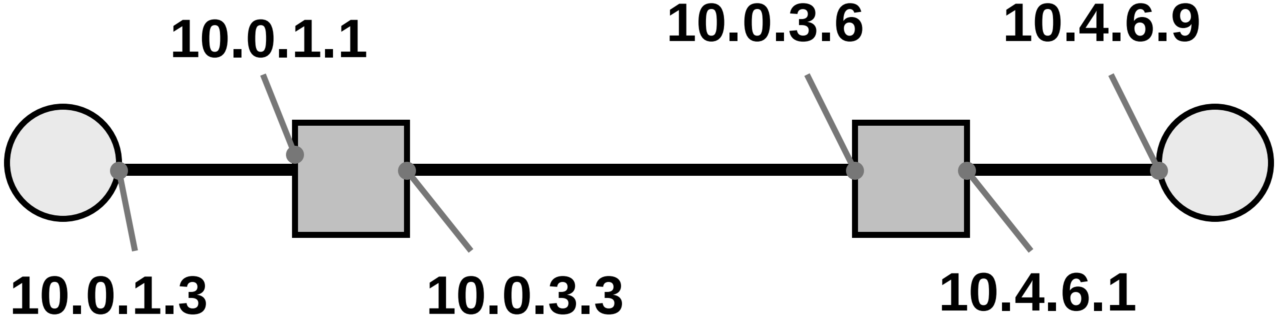


Layer 2 Routing (3)



Need for Layer 3 Routing

- Of course, nodes may not be connected via Layer 2
  - Nodes that are in a different IP subnet, i.e., the destination IP network is different than the local IP network
  - Nodes that are out of radio range in an ad hoc wireless network
- Layer 3, or IP, routing is needed in this case



Routing

- Routing consists of two fundamental steps
  - Forwarding packets to the next hop (from an input interface to an output interface in a traditional wired network)
  - Determining how to forward packets (building a routing table or specifying a route)
- Forwarding packets is easy, but knowing where to forward packets (especially efficiently) is hard
  - Reach the destination
  - Minimize the number of hops (path length)
  - Minimize delay
  - Minimize packet loss
  - Minimize cost

Routing Decision Point

- **Source routing**
  - Sender determines a route and specifies it in the packet header
  - Supported in IP, although not the typical routing scheme
- **Hop-by-hop (datagram) routing**
  - A routing decision is made at each forwarding point (at each router)
  - Standard routing scheme for IP
- **Virtual circuit routing**
  - Determine and configure a path prior to sending first packet
  - Used in ATM (and analogous to voice telephone system)

Routing Table

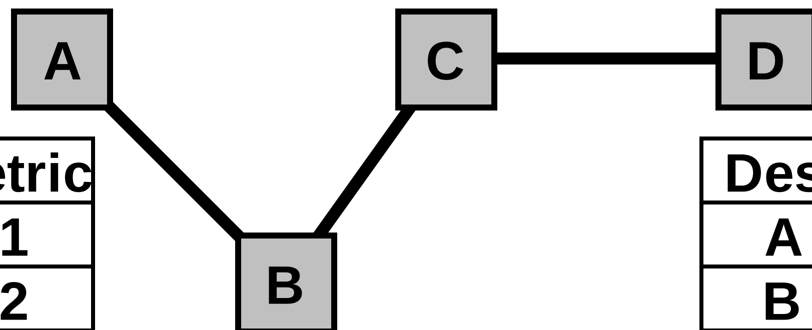
- **A routing table contains information to determine how to forward packets**
  - **Source routing: Routing table is used to determine route to the destination to be specified in the packet**
  - **Hop-by-hop routing: Routing table is used to determine the next hop for a given destination**
  - **Virtual circuit routing: Routing table used to determine path to configure through the network**
- **A distributed algorithm is required to build the routing table**
  - **Distance vector algorithms**
  - **Link state algorithms**

Distance Vector Algorithms (1)

- "Distance" of each link in the network is a metric that is to be minimized
  - Each link may have "distance" 1 to minimize hop count
  - Algorithm attempts to minimize distance
- The routing table at each node…
  - Specifies the next hop for each destination
  - Specifies the distance to that destination
- Neighbors can exchange routing table information to find a route (or a better route) to a destination

Distance Vector Algorithms (2)

Dest	Next	Metric
A	B	2
B	B	1
D	D	1



Dest	Next	Metric
B	B	1
C	B	2
D	B	3

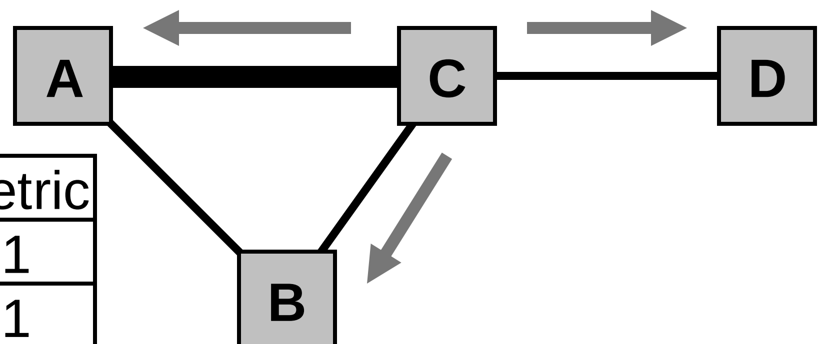
Dest	Next	Metric
A	A	1
C	C	1
D	C	2

Dest	Next	Metric
A	C	3
B	C	2
C	C	1

Distance Vector Algorithms (3)

- **Node A will learn of Node C's shorter path to Node D and update its routing table**

Dest	Next	Metric
A	A	1
B	B	1
D	D	1

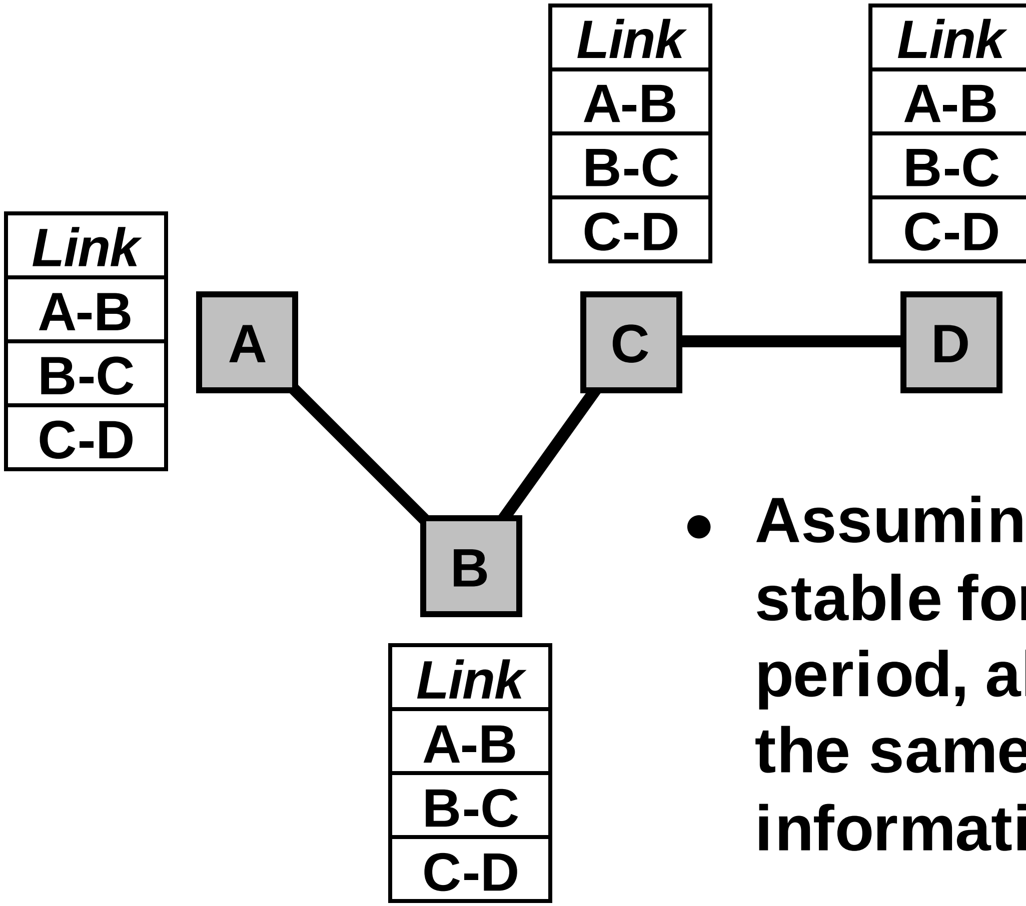


Dest	Next	Metric
B	B	1
C	C	1
D	C	2

Link-State Algorithms (1)

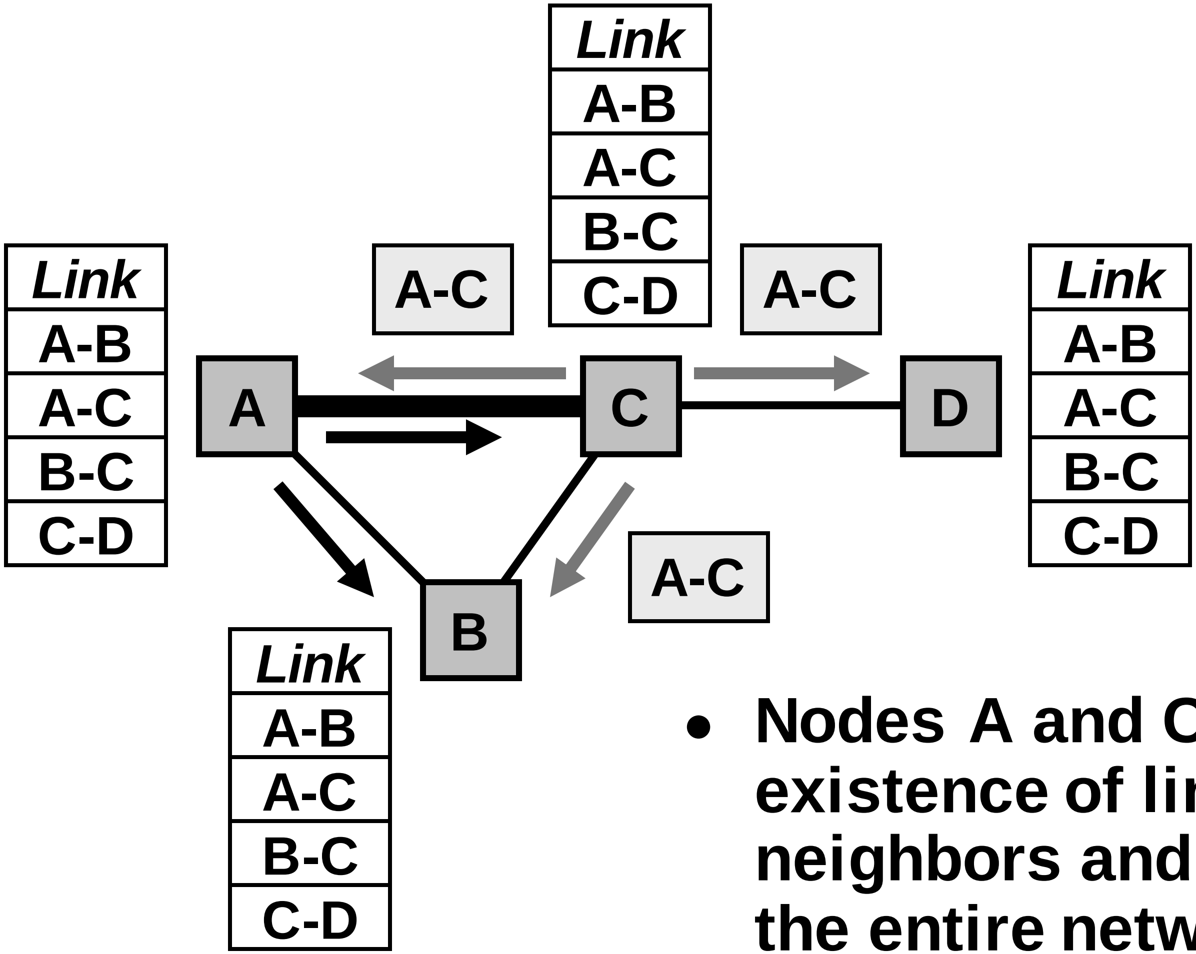
- Each node shares its link information so that all nodes can build a map of the full network topology
- Link information is updated when a link changes state (goes up or down)
  - Link state determined by sending small “hello” packets to neighbors
- Given full topology information, a node can determine the next best hop or a route from the source

Link-State Algorithms (2)



- **Assuming the topology is stable for a sufficiently long period, all nodes will have the same topology information**

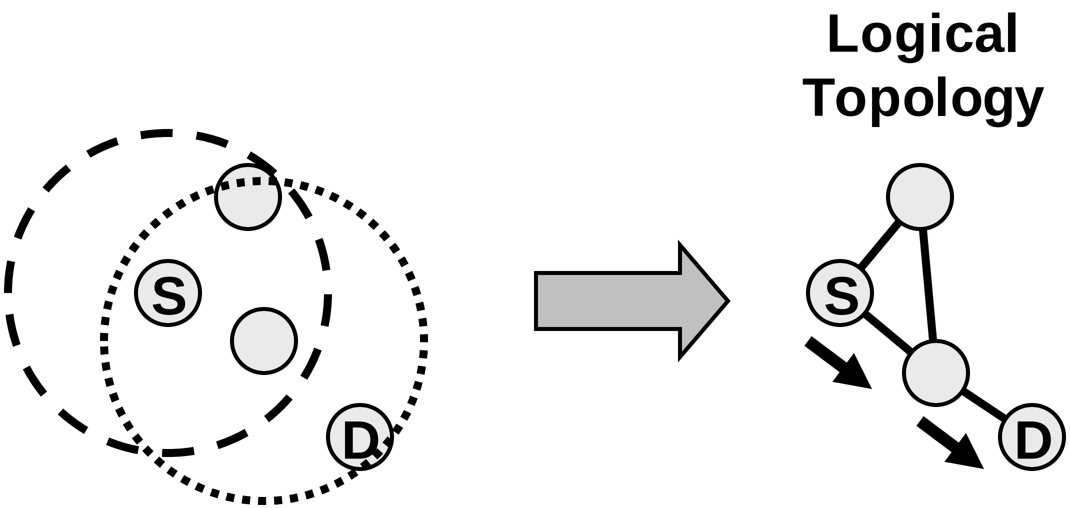
Link-State Algorithms (3)



- **Nodes A and C propagate the existence of link A-C to their neighbors and, eventually, to the entire network**

MANETs

- **A mobile ad hoc network (MANET) is characterized by…**
  - **Multi-hop routing so that nodes not directly connected at Layer 2 can communicate through Layer 3 routing**
  - **Wireless links**
  - **Mobile nodes**



MANET vs. Traditional Routing (1)

- Every node is potentially a router in a MANET, while most nodes in traditional wired networks do not route packets
  - Nodes transmit and receive their own packets and, also, forward packets for other nodes
- Topologies are dynamic in MANETs due to mobile nodes, but are relatively static in traditional networks
- Routing in MANETs must consider both Layer 3 and Layer 2 information, while traditional protocols rely on Layer 3 information only
  - Link layer information can indicate connectivity and interference

MANET vs. Traditional Routing (2)

- MANET topologies tend to have many more redundant links than traditional networks
- A MANET "router" typically has a single interface, while a traditional router has an interface for each network to which it connects
  - Routed packet sent forward when transmitted, but also sent to previous transmitter
- Channel properties, including capacity and error rates, are relatively static in traditional networks, but may vary in MANETs

MANET vs. Traditional Routing (3)

- Interference is an issue in MANETs, but not in traditional networks
  - For example, a forwarded packet from B-to-C competes with new packets sent from A-to-B
- Channels can be asymmetric with some Layer 2 technologies
  - Note that the IEEE 802.11 MAC assumes symmetric channels
- Power efficiency is an issue in MANETs, while it is normally not an issue in traditional networks
- MANETs may have gateways to fixed network, but are typically "stub networks," while traditional networks can be stub networks or transit networks

MANET vs. Traditional Routing (4)

- **There is limited physical security in a MANET compared to a traditional network**
  - **Increased possibility of eavesdropping, spoofing, and denial-of-security attacks**
- **Traditional routing protocols for wired networks do not work well in most MANETs**
  - **MANETs are too dynamic**
  - **Wireless links present problems of interference, limited capacity, etc.**

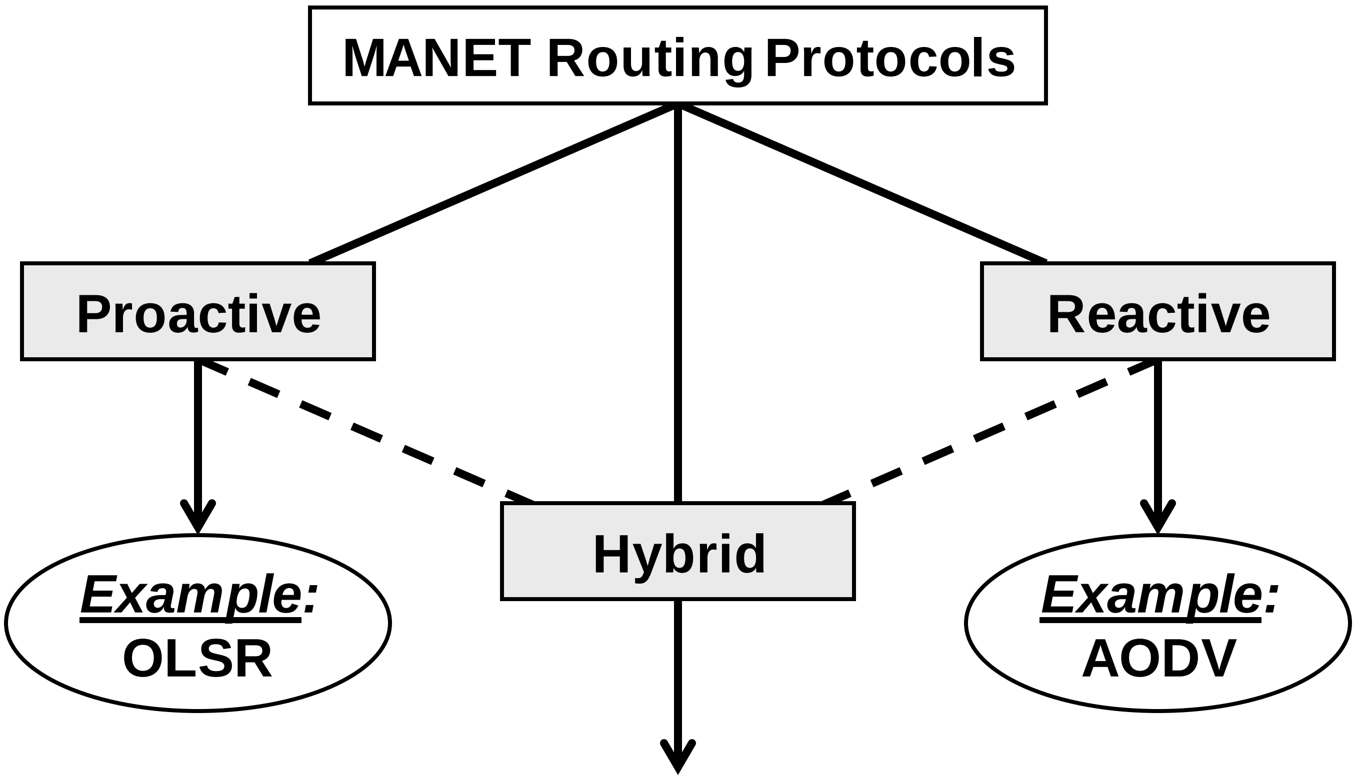
MANET Routing

- Nodes must determine how to forward packets
  - Source routing: Routing decision is made at the sender
  - Hop-by-hop routing: Routing decision is made at each intermediate node
- Difficult to achieve good performance
  - Routes change over time due to node mobility
  - Best to avoid long delays when first sending packets
  - Best to reduce overhead of route discovery and maintenance
  - Want to involve as many nodes as possible – to find better paths and reduce likelihood of partitions

MANET Routing Approaches

- **Decision time**
  - Proactive or table-driven – maintain routing tables
  - Reactive or on-demand – determine routing on an as-needed basis
- **Network structure**
  - Hierarchical – impose a hierarchy on a collection of nodes and reflect this hierarchy in the routing algorithm
    - May use a proactive protocol for routing within a cluster or zone
    - May use a reactive protocol for routing between distinguished "cluster heads"
  - Non-hierarchical – make decisions among all nodes

Types of MANET Routing



Common Features

- **MANET routing protocols must…**
  - Discover a path from source to destination
  - Maintain that path (e.g., if an intermediate node moves and breaks the path)
  - Define mechanisms to exchange routing information
- **Reactive protocols**
  - Discover a path when a packet needs to be transmitted and no known path exists
  - Attempt to alter the path when a routing failure occurs
- **Proactive protocols**
  - Find paths, in advance, for all source-pair destinations
  - Periodically exchange routing information to maintain paths

IETF MANET Working Group (1)

http://www.ietf.org/html.charters/manet-charter.html

"The purpose of this working group is to standardize IP routing protocol functionality suitable for wireless routing application within both static and dynamic topologies. The fundamental design issues are that the wireless link interfaces have some unique routing interface characteristics and that node topologies within a wireless routing region may experience increased dynamics, due to motion or other factors."

IETF MANET Working Group (2)

- Currently trying to move four proposed MANET routing protocols to Experimental RFC status
  - Ad Hoc On Demand Distance Vector (AODV) protocol
  - Dynamic Source Routing (DSR) protocol
  - Optimized Link State Routing (OLSR) protocol
  - Topology Broadcast based on Reverse-Path Forwarding (TBRPF) protocol
- URLs
  - http://www.ietf.org/html.charters/manet-charter.html
  - http://protean.itd.nrl.navy.mil/manet/manet_home.html

OLSR

- **Optimized Link State Routing (OLSR) protocol**
  - **On track to become an IETF Experimental RFC**
- **References**
  - **C. Adjih, *et al*., "Optimized Link State Routing Protocol," IETF Internet Draft, draft-ietf-manet-olsr-08.txt, March 3, 2003.**
  - **P. Jacquet, P. Muhlethaler, T. Clausen, A. Laouiti, A. Qayyum, and L. Viennot, "Optimized Link State Routing Protocol for Ad Hoc Networks," *Proceedings IEEE INMIC*, 2001, pp. 62-68.**

OLSR Concepts (1)

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

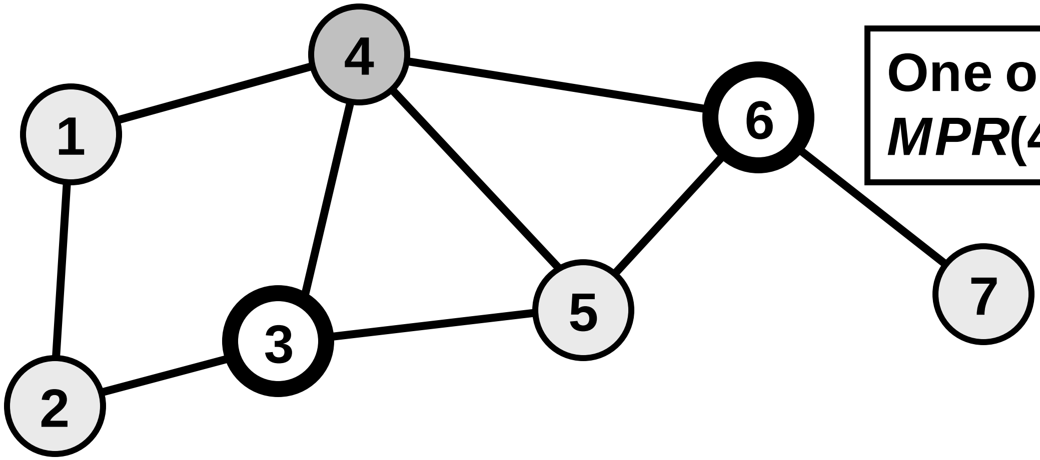
OLSR Concepts (2)

- **Does not require reliable transfer, since updates are sent periodically**
- **Does not need in-order delivery, since sequence numbers are used to prevent out-of-date information from being misinterpreted**
- **Uses hop-by-hop routing**
  - **Routes are based on dynamic table entries maintained at intermediate nodes**

MPR(N): set of 1-hop neighbor nodes selected by N to forward N's broadcasted packets, to cover all 2-hop neighbor nodes

Multipoint Relays

- **Each node *N* in the network selects a set of neighbor nodes as multipoint relays, *MPR*(*N*), that retransmit control packets from *N***
  - **Neighbors not in *MPR*(*N*) process control packets from *N*, but they do not forward the packets**
- ***MPR*(*N*) is selected such that all two-hop neighbors of *N* are covered by (one-hop neighbors) of *MPR*(*N*)**



One optimal set for Node 4:
***MPR*(4) = { 3, 6 }**

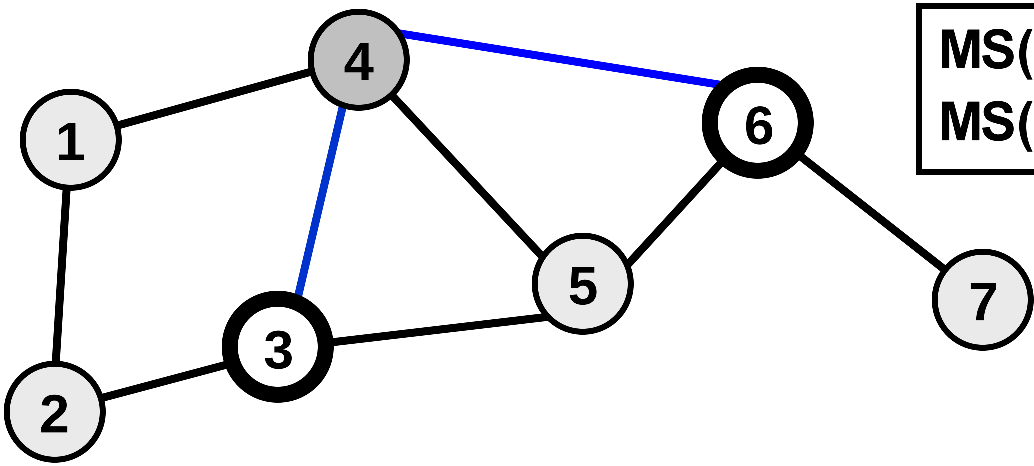
Is there another optimal *MPR*(4)?

MS(N): set of SOURCE NODES (in 1-hop neighbor) selected N to forward THEIR broadcasted packets, to cover all 2-hop neighbor nodes, i.e.,

$$\forall V_i \in \{SOURCE\_NODES\} : N \in MPR(V_i)$$

- **The multipoint relay selector set for Node *N*, *MS*(*N*), is the set of nodes that choose Node *N* in their multipoint relay set**
  - **Only links *N-M*, for all *M* such that *N*∈*MS*(*M*) will be advertised in control messages**

Only link 4-3 advertised by 3 in control message: 4 ∈MS(3)
Only link 4-6 advertised by 6 in control message: 4 ∈MS(6)



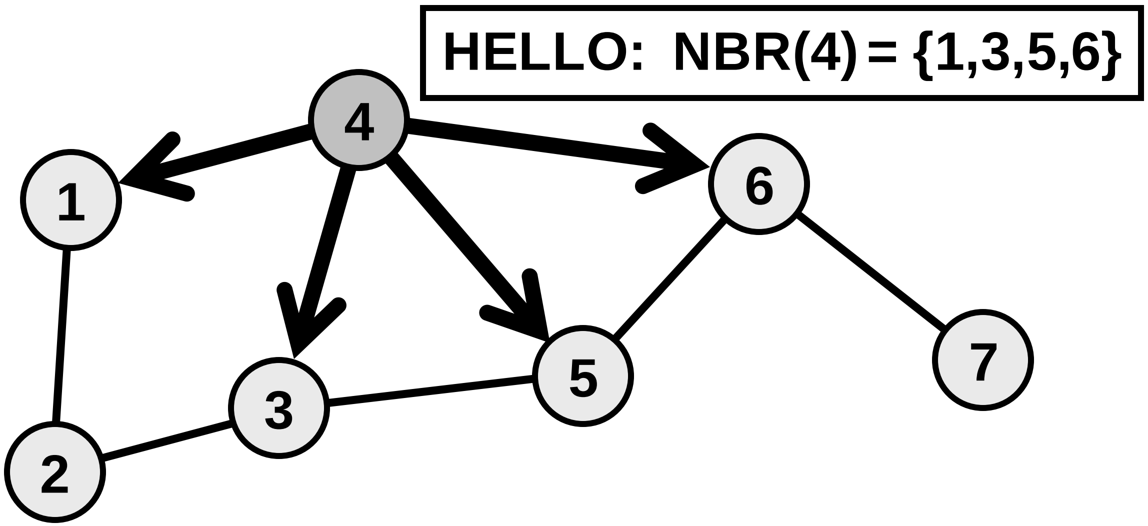
MS(3) = {…, 4, …}
MS(6) = {…, 4, …}

(Assuming bidirectional links)

Multipoint Relay Selector Set

HELLO Messages (1)

- Each node uses HELLO messages to determine its MPR set
- All nodes periodically broadcast HELLO messages to their one-hop neighbors (bidirectional links)
- HELLO messages are not forwarded

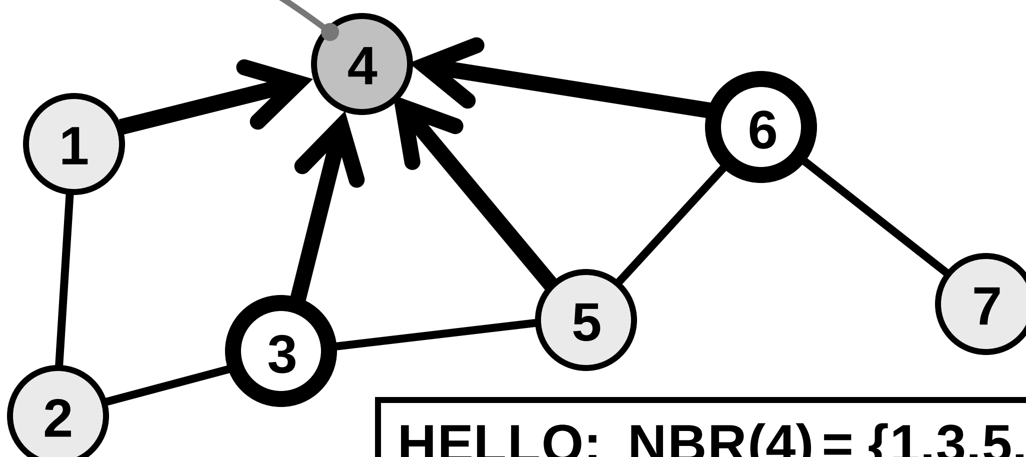


HELLO Messages (2)

- **Using the neighbor list in received HELLO messages, nodes can determine their two-hop neighborhood and an optimal (or near-optimal) MPR set**
- **A sequence number is associated with this MPR set**
  - **Sequence number is incremented each time a new set is calculated**

At Node 4:
NBR(1) = {2}
NBR(3) = {2,5}
NBR(5) = {3,6}
NBR(6) = {5,7}

MPR(4) = {3,6}

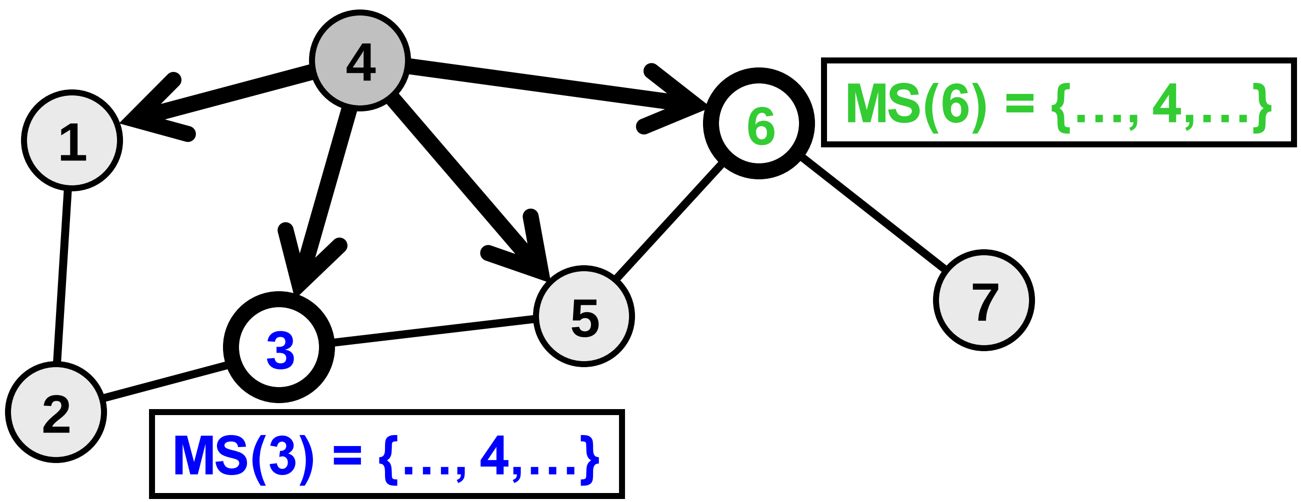


HELLO: NBR(4) = {1,3,5,6}

HELLO Messages (3)

- Subsequent HELLO messages also indicate neighbors that are in the node’s MPR set
- MPR set is recalculated when a change in the one-hop or two-hop neighborhood is detected

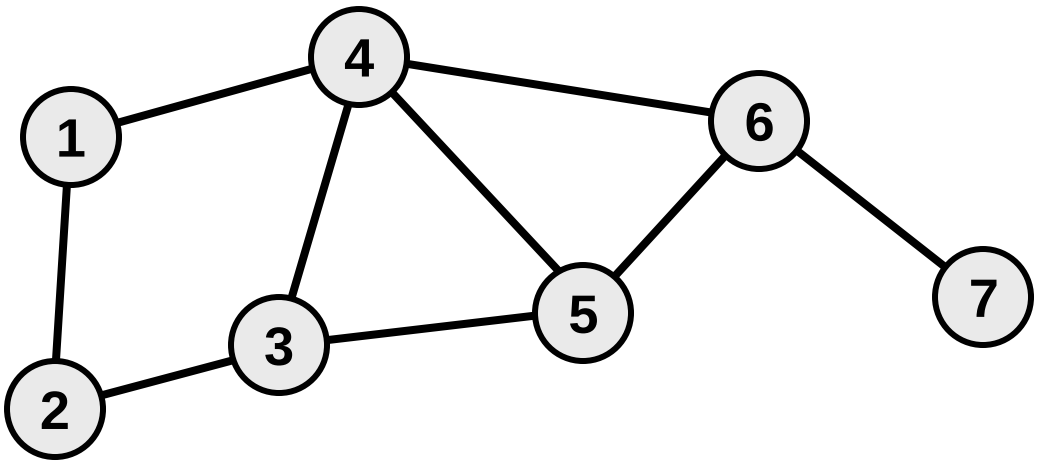
HELLO: NBR(4) = {1,3,5,6}, MPR(4) = {3,6}



TC Messages

- Nodes send topology information in Topology Control (TC) messages
  - List of advertised neighbors (link information)
  - Sequence number (to prevent use of stale information)
- You needed me, i.e., in my MS set
  - Any links changed from me to you [CASE-I]
    - Generate TC messages for you
      - A node generates TC messages only for those neighbors in its MS set
        - Only MPR nodes generate TC messages
        - Not all links are advertised
  - Received TC from you [CASE-II]
    - Continue forwarding for you
      - A nodes processes all received TC messages, but only forwards TC messages if the sender is in its MS set
        - Only MPR nodes propagate TC messages

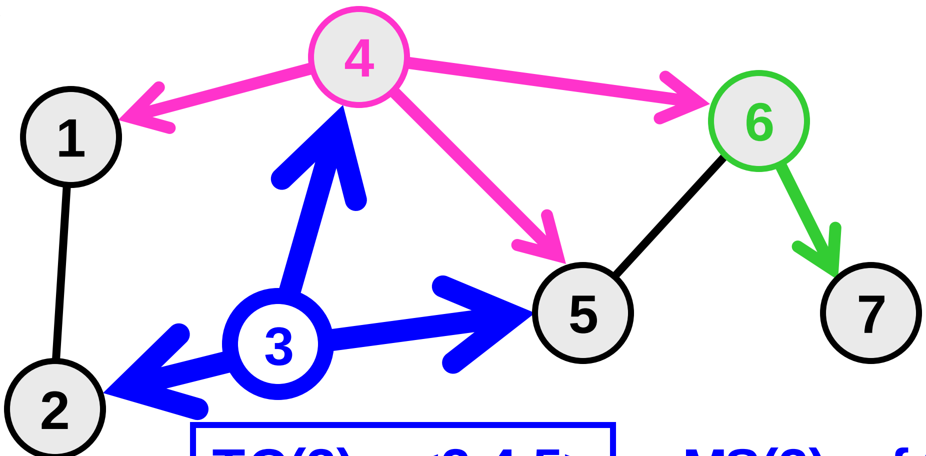
OLSR Example (1)



MPR	MS
MPR(1) = { 4 }	MS(1) = { }
MPR(2) = { 3 }	MS(2) = { }
MPR(3) = { 4 }	MS(3) = { 2, 4, 5 }
MPR(4) = { 3, 6 }	MS(4) = { 1, 3, 5, 6 }
MPR(5) = { 3, 4, 6 }	MS(5) = { }
MPR(6) = { 4 }	MS(6) = { 4, 5, 7 }
MPR(7) = { 6 }	MS(7) = { }

OLSR Example (2)

A node generates TC messages only for those neighbors in its MS set, i.e., only MPR nodes generate TC messages



TC(3) = <2,4,5>

MS(3) = { 2, 4, 5 }

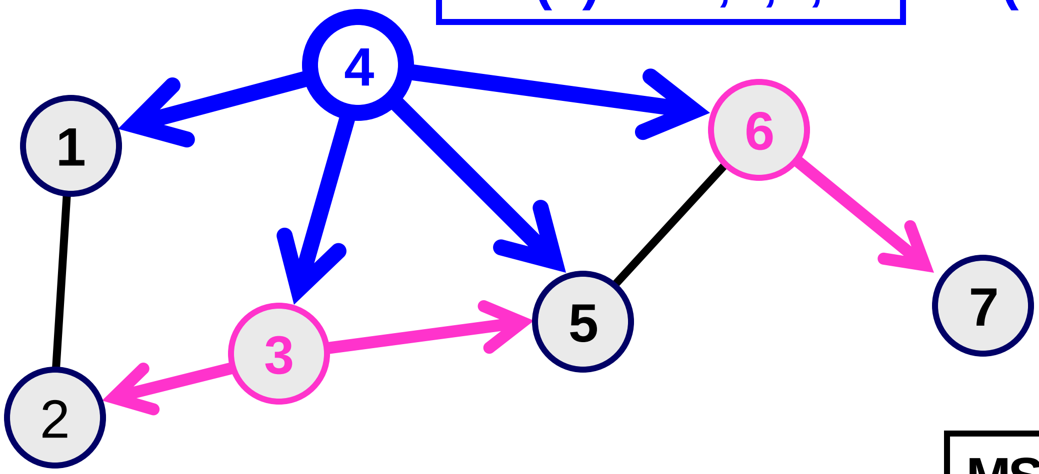
A nodes processes all received TC messages, but only forwards TC messages if the sender is in its MS set, only MPR nodes propagate TC messages

- **Node 3 generates a TC message advertising nodes in MS(3) = {2, 4, 5}**
- **Node 4 forwards Node 3's TC message since Node 3 $\in$ MS(4) = {1, 3, 5, 6}**
- **Node 6 forwards TC(3) since Node 4 $\in$ MS(6)**

MS(1) = { }
MS(2) = { }
MS(3) = { 2, 4, 5 }
MS(4) = { 1, 3, 5, 6 }
MS(5) = { }
MS(6) = { 4, 5, 7 }
MS(7) = { }

OLSR Example (3)

TC(4) = <1,3,5,6> MS(4) = { 1, 3, 5, 6 }

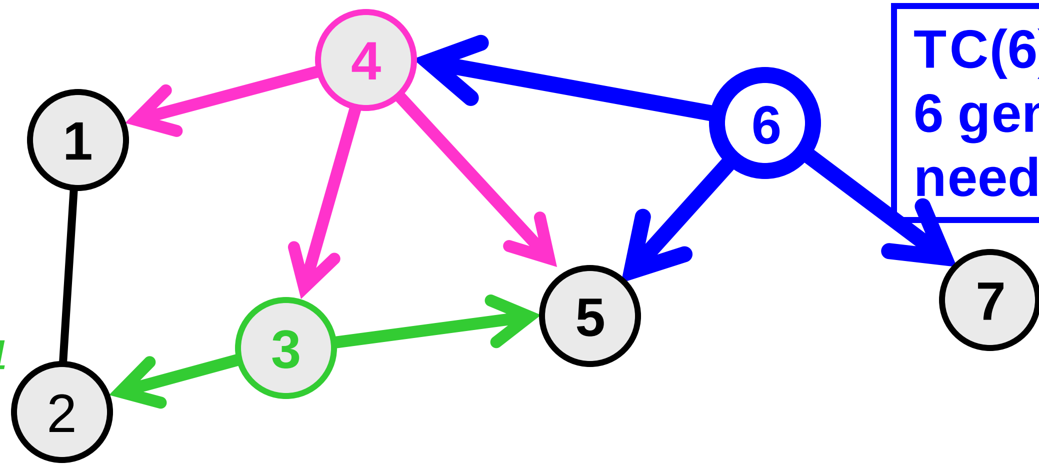


- **Node 4 generates a TC message advertising nodes in MS(4) = {1, 3, 5, 6}**
- **Nodes 3 and 6 forward TC(4) since Node 4 ∈ MS(3) and Node 4 ∈ MS(6)**

MS(1) = { }
MS(2) = { }
MS(3) = { 2, 4, 5 }
MS(4) = { 1, 3, 5, 6 }
MS(5) = { }
MS(6) = { 4, 5, 7 }
MS(7) = { }

OLSR Example (4)

6 needs 4, then 4 forwards for 6



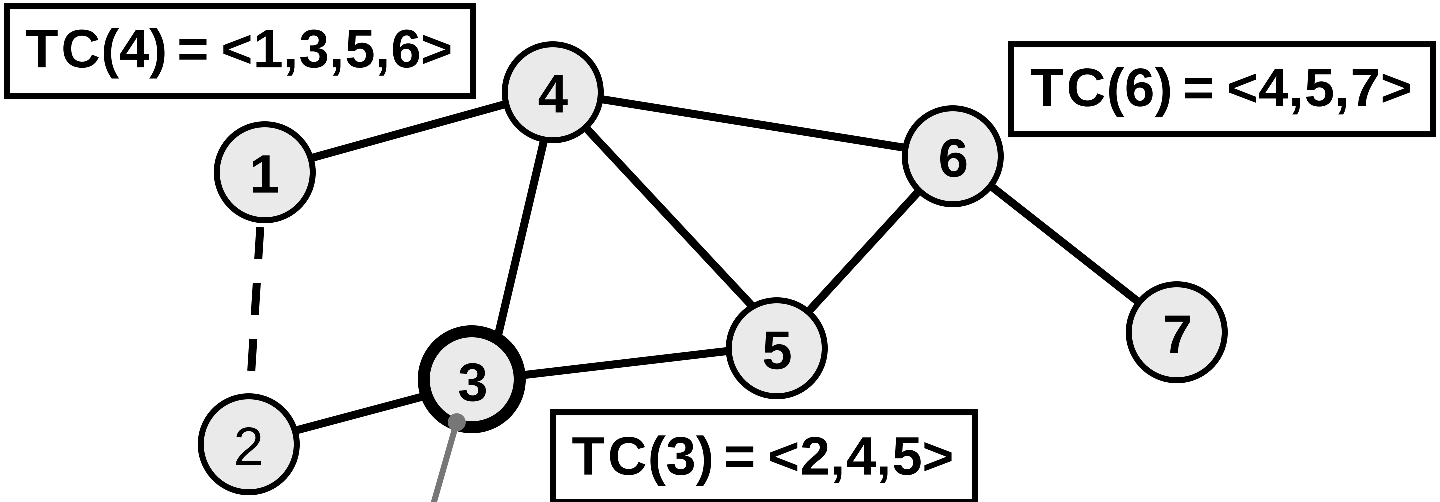
TC(6) = <4,5,7>
6 generates TC to nodes needing it as forwarders

4 needs 3, then 3 forwards for 4

- **Node 6 generates a TC message advertising nodes in MS(6) = {4, 5, 7}**
- **Node 4 forwards TC(6) from Node 6 and Node 3 forwards TC(6) from Node 4**
- **After Nodes 3, 4, and 6 have generated TC messages, all nodes have link-state information to route to any node**

MS(1) = { }
MS(2) = { }
MS(3) = { 2, 4, 5 }
MS(4) = { 1, 3, 5, 6 }
MS(5) = { }
MS(6) = { 4, 5, 7 }
MS(7) = { }

OLSR Example (5)



Dest	*Next*	*Hops*
1	4	2
2	2	1
4	4	1
5	5	1
6	4 (5)	2
7	4 (5)	3

- **Given TC information, each node forms a topology table**
- **A routing table is calculated from the topology table**
- **Note that Link 1-2 is not visible except to Nodes 2 and 3**

AODV

- **AODV: Ad hoc On-demand Distance Vector routing protocol**
  - **On track to become an IETF Experimental RFC**
- **References**
  - **C. E. Perkins, E. M. Belding-Royer, and S. R. Das, "Ad hoc On-Demand Distance Vector (AODV) Routing," IETF Internet Draft, draft-ietf-manet-aodv-13.txt, Feb. 17, 2003 (work in progress).**
  - **C. E. Perkins and E. M. Royer, "Ad hoc On-Demand Distance Vector Routing,"** ***Proceedings 2nd IEEE Workshop on Mobile Computing Systems and Applications*****, February 1999, pp. 90-100.**

AODV Concepts (1)

- **Pure on-demand routing protocol**
  - A node does not perform route discovery or maintenance until it needs a route to another node or it offers its services as an intermediate node
  - Nodes that are not on active paths do not maintain routing information and do not participate in routing table exchanges
- **Uses a broadcast route discovery mechanism**
- **Uses hop-by-hop routing**
  - Routes are based on dynamic table entries maintained at intermediate nodes
  - Similar to Dynamic Source Routing (DSR), but DSR uses source routing

AODV Concepts (2)

- **Local HELLO messages are used to determine local connectivity**
  - Can reduce response time to routing requests
  - Can trigger updates when necessary
- **Sequence numbers are assigned to routes and routing table entries**
  - Used to supersede stale cached routing entries
- **Every node maintains two counters**
  - Node sequence number
  - Broadcast ID

AODV Route Request (1)

- Initiated when a node wants to communicate with another node, but does not have a route to that node
- Source node broadcasts a route request (RREQ) packet to its neighbors

<table>
<tr><td>type</td><td>flags</td><td>resvd</td><td>hopcnt</td></tr>
<tr><td colspan="4">broadcast_id</td></tr>
<tr><td colspan="4">dest_addr</td></tr>
<tr><td colspan="4">dest_sequence_#</td></tr>
<tr><td colspan="4">source_addr</td></tr>
<tr><td colspan="4">source_sequence_#</td></tr>
</table>

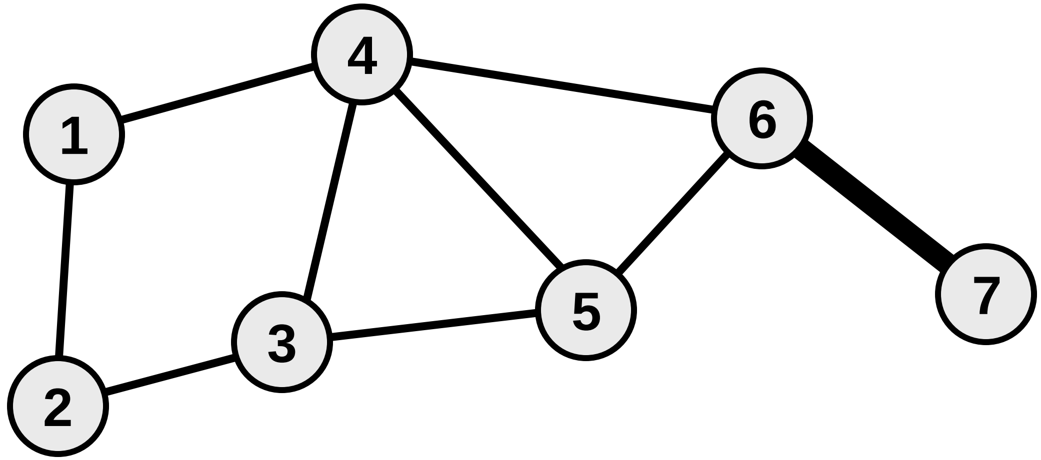
AODV Route Request (2)

- **Sequence numbers**
  - Source sequence indicates "freshness" of reverse route to the source
  - Destination sequence number indicates freshness of route to the destination
- **Every neighbor receives the RREQ and either …**
  - Returns a route reply (RREP) packet, or
  - Forwards the RREQ to its neighbors
- **(source_addr, broadcast_id) uniquely identifies the RREQ**
  - broadcast_id is incremented for every RREQ packet sent
  - Receivers can identify and discard duplicate RREQ packets

AODV Route Request (3)

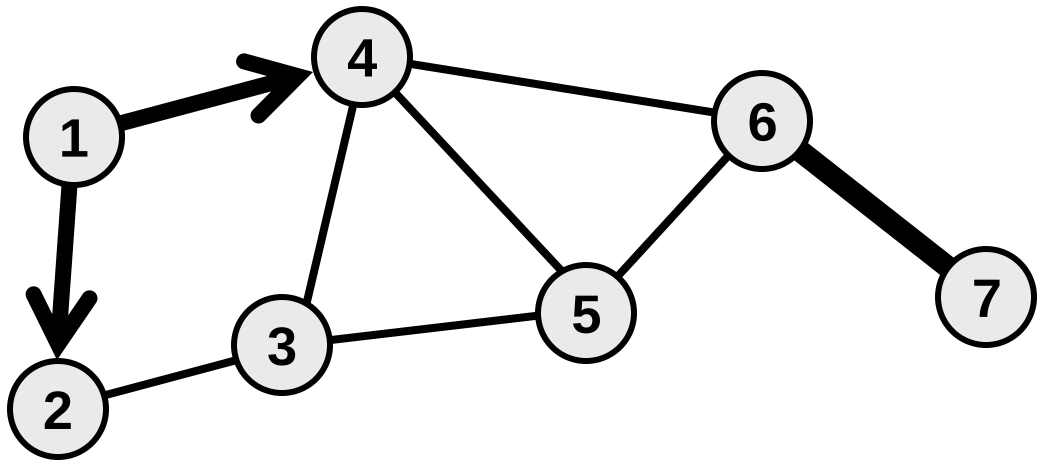
- **If a node *cannot* respond to the RREQ**
  - **The node increments the hop count**
  - **The node saves information to implement a reverse path set up (AODV assumes symmetrical links)**
    - **Neighbor that sent this RREQ packet**
    - **Destination IP address**
    - **Source IP address**
    - **Broadcast ID**
    - **Source node's sequence number**
    - **Expiration time for reverse path entry (to enable garbage collection)**

AODV Example (1)



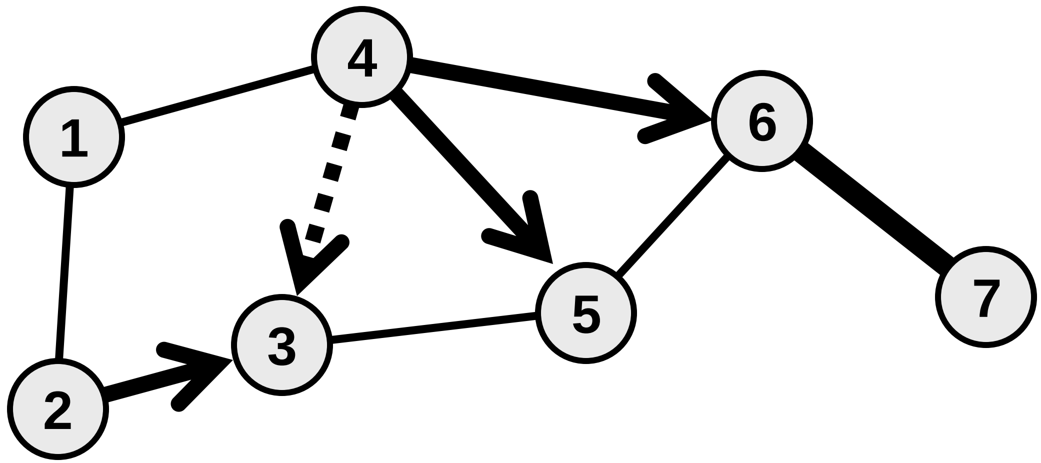
- Node 1 needs to send a data packet to Node 7
- Assume Node 6 knows a current route to Node 7
- Assume that no other route information exists in the network (related to Node 7)

AODV Example (2)



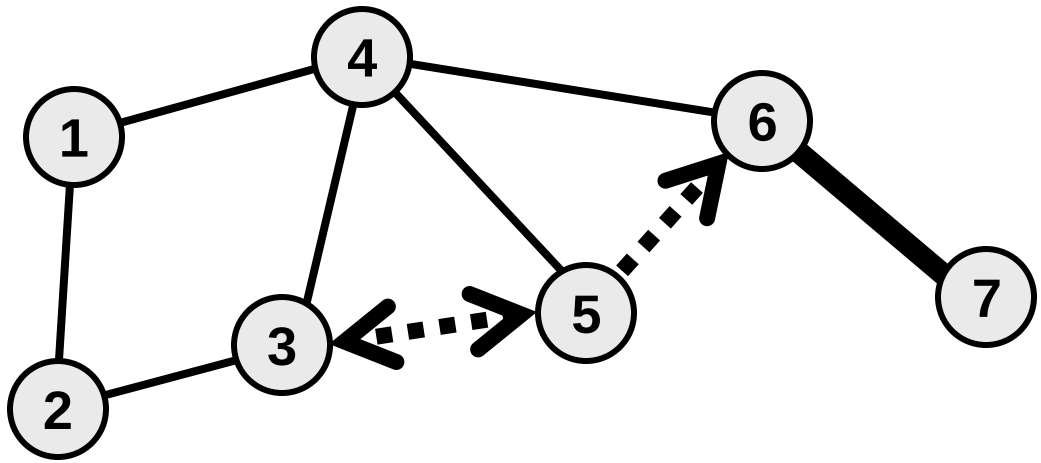
- Node 1 sends a RREQ packet to its neighbors
  - source_addr = 1
  - dest_addr = 7
  - broadcast_id = broadcast_id + 1
  - source_sequence_# = source_sequence_# + 1
  - dest_sequence_# = last dest_sequence_# for Node 7

AODV Example (3)



- Nodes 2 and 4 verify that this is a new RREQ and that the source_sequence_# is not stale with respect to the reverse route to Node 1
- Nodes 2 and 4 forward the RREQ
  - Update source_sequence_# for Node 1
  - Increment hop_cnt in the RREQ packet

AODV Example (4)



- RREQ reaches Node 6, which knows a route to 7
  - Node 6 must verify that the destination sequence number is less than or equal to the destination sequence number it has recorded for Node 7
- Nodes 3 and 5 will forward the RREQ packet, but the receivers recognize the packets as duplicates

AODV Route Reply (1)

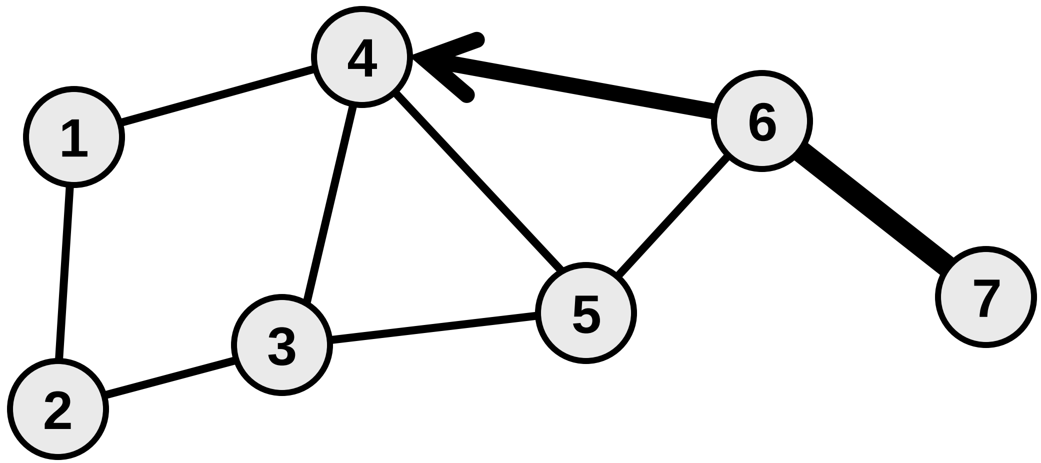
- **If a node receives an RREQ packet and it has a current route to the target destination, then it unicasts a route reply packet (RREP) to the neighbor that sent the RREQ packet**

type	flags	rsvd	prsz	hopcnt
dest_addr				
dest_sequence_#				
source_addr				
lifetime				

AODV Route Reply (2)

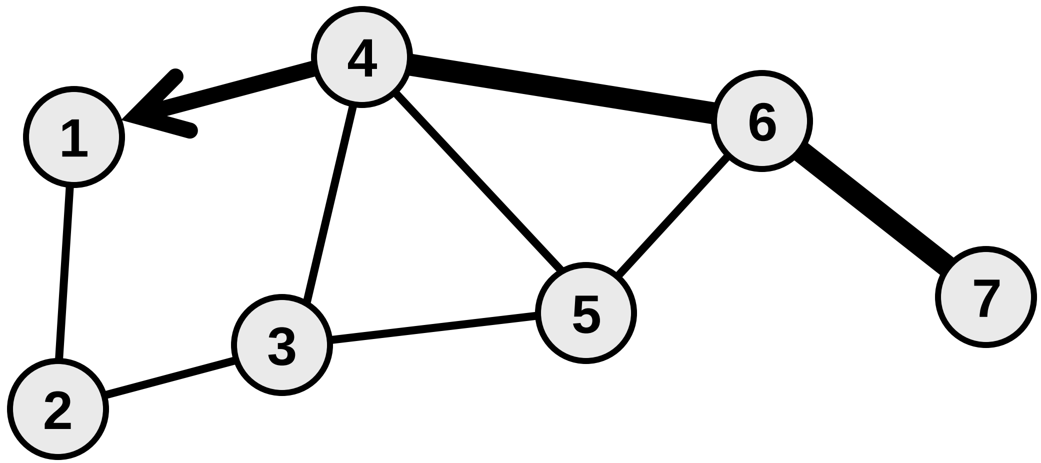
- Intermediate nodes propagate the first RREP for the source towards the source using cached reverse route entries
- Other RREP packets are discarded unless…
  - dest_sequence_# number is higher than the previous, or
  - destination_sequence_# is the same, but hop_cnt is smaller (i.e., there's a better path)
- RREP eventually makes it to the source, which can use the neighbor sending the RREP as its next hop for sending to the destination
- Cached reverse routes will timeout in nodes not seeing a RREP packet

AODV Example (5)



- **Node 6 knows a route to Node 7 and sends an RREP to Node 4**
  - **source_addr = 1**
  - **dest_addr = 7**
  - **dest_sequence_# = maximum(own sequence number, dest_sequence_# in RREQ)**
  - **hop_cnt = 1**

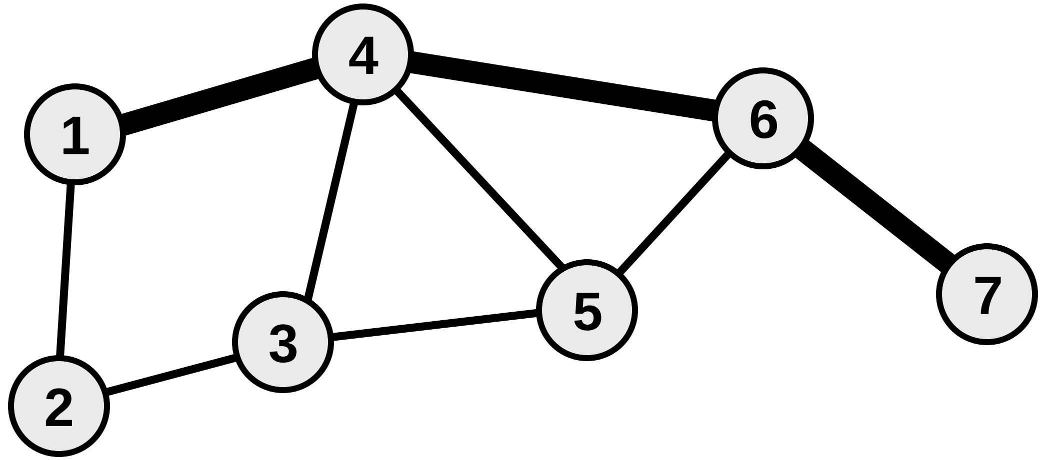
AODV Example (6)



- Node 4 verifies that this is a new route reply (the case here) or one that has a lower hop count and, if so, propagates the RREP packet to Node 1
  - Increments hop_cnt in the RREP packet

AODV Example (7)

Dest	Next	Hops
7	4	3

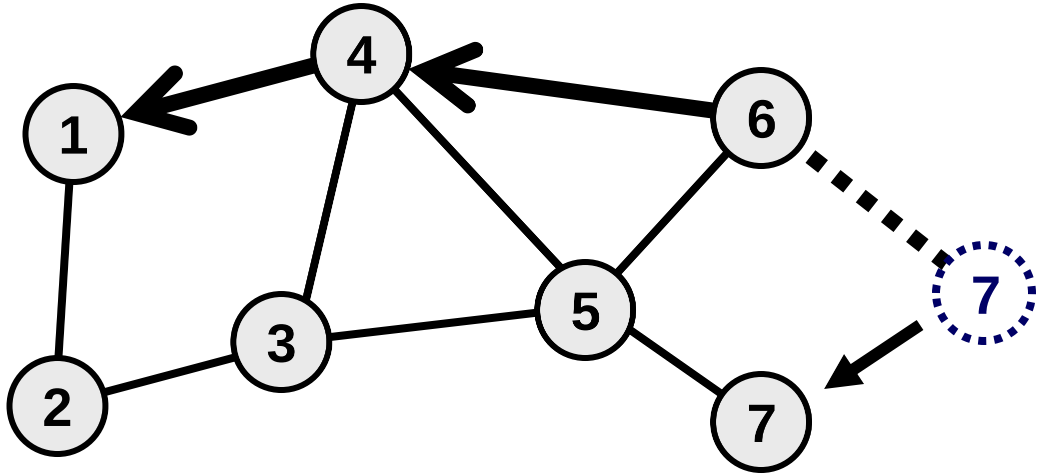


- Node 1 now has a route to Node 7 in three hops and can use it immediately to send data packets
- Note that the first data packet that prompted path discovery has been delayed until the first RREP was returned

AODV Route Maintenance

- Route changes can be detected by…
  - Failure of periodic HELLO packets
  - Failure or disconnect indication from the link level
  - Failure of transmission of a packet to the next hop (can detect by listening for the retransmission if it is not the final destination)
- The upstream (toward the source) node detecting a failure propagates an route error (RERR) packet with a new destination sequence number and a hop count of infinity (unreachable)
- The source (or another node on the path) can rebuild a path by sending a RREQ packet

AODV Example (8)



- Assume that Node 7 moves and link 6-7 breaks
- Node 6 issues an RERR packet indicating the broken path
- The RERR propagates back to Node 1
- Node 1 can discover a new route

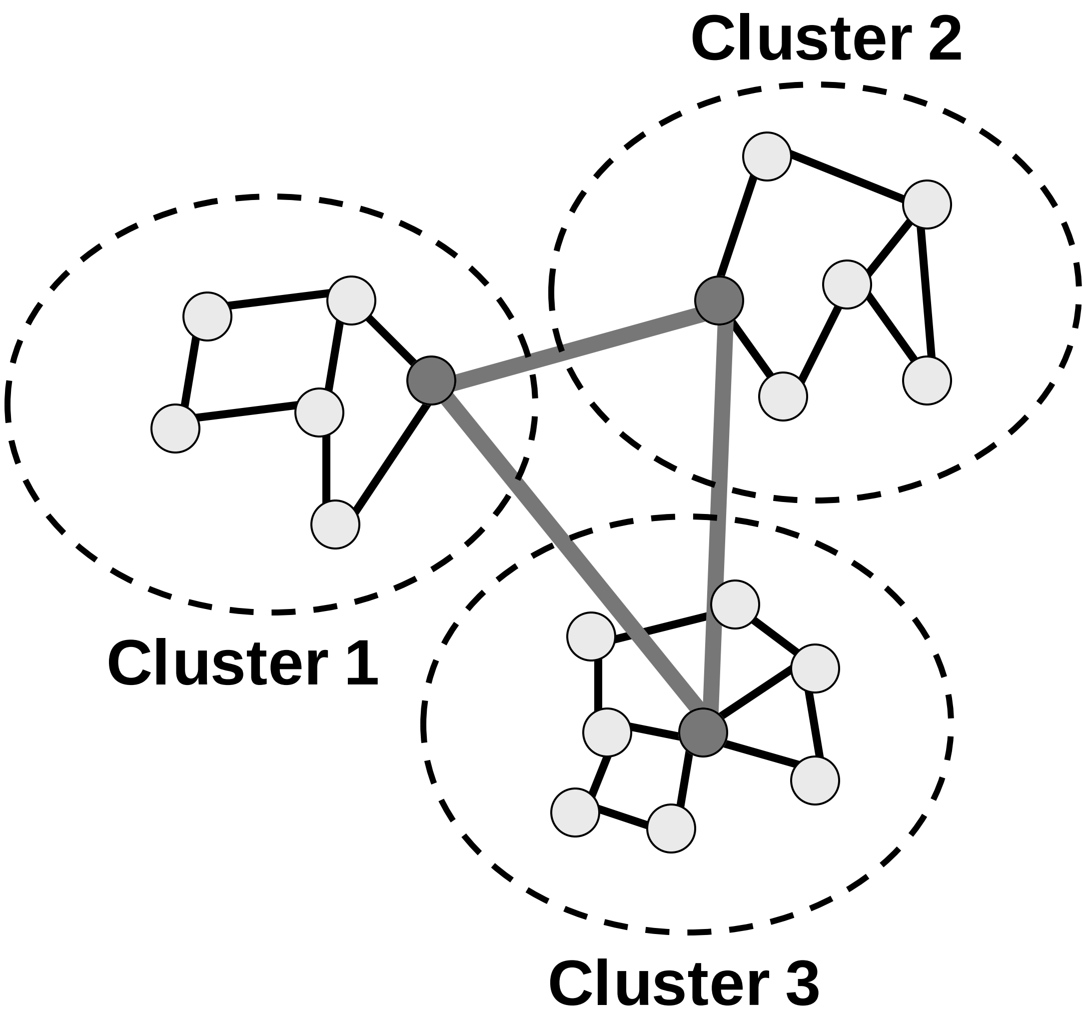
Hierarchical Algorithms (1)

- **Scalability – MANET protocols often do not perform well for large networks (especially if not dense)**
  - Global topology is based on the connectivity of each mobile node
- **Clusters can be used to provide scalability**
  - Clusters are formed (dynamically, of course) to provide hierarchy
  - Global routing is done to clusters
  - Local routing is done to nodes within a cluster
  - Clusters of clusters (super-clusters) can be formed to extend hierarchy
  - Similar in principle to IP subnets

Hierarchical Algorithms (2)

- **A special node, called the *cluster-head*, is designated in each cluster**
  - **Responsible for routing data to or from other clusters**
  - **May be a special node, or may be designated through a clustering algorithm**
- **Algorithms**
  - **Clustering -- form clusters**
  - **Cluster-head identification -- may be an integral part of the clustering algorithm**
  - **Routing -- some routing algorithm is still needed**
    - **Applied at each level of the hierarchy**

Hierarchical Algorithm Example



Summary

- Layer 3 routing is needed to extend wireless mobile networks beyond local area networks of directly connected nodes
- Mobile ad hoc networks use multi-hop routing to enable communications in dynamic topologies
- MANET routing is hard to do well – it experiences the problems of both wireless <u>and</u> mobility
- A number of reactive and proactive MANET routing protocols have been proposed
- MANETs are still a niche application and they are relatively immature