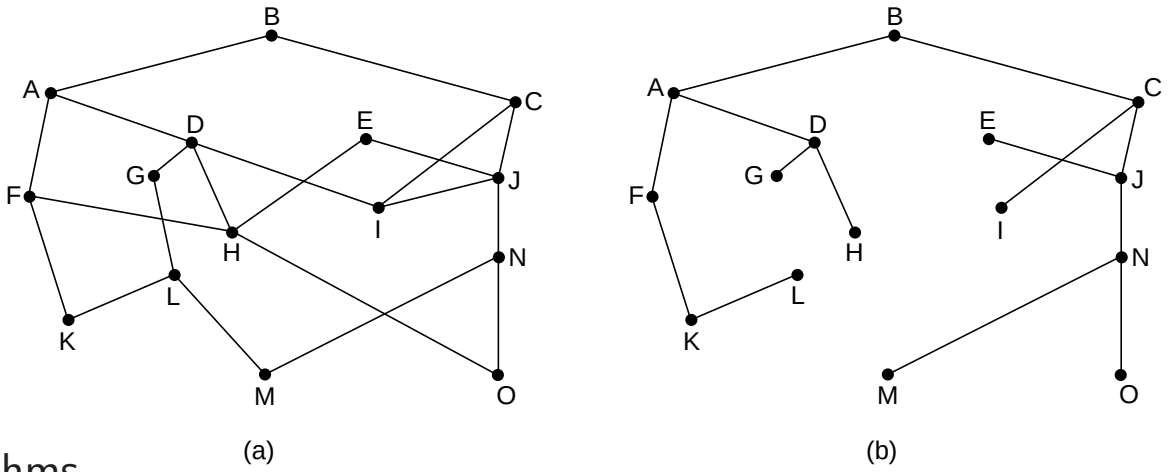
Routing Overview

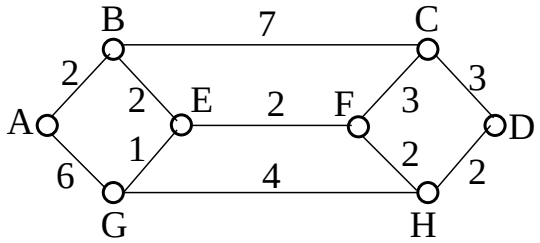
- Main issue is how the routers that constitute the network layer of a network cooperate to find the best routes between all pairs of stations
- Routing algorithm at a router decides which output line an incoming packet should go, i.e. making a routing decision. It should process properties, like correctness, simplicity, robustness, stability, fairness, and optimality. Note optimality is always with respect to chosen criterion
- **Optimality principle**: if router J is on optimal path from router I to router K, then optimal path from J to K also falls along same route
- **Sink tree**: set of optimal routes from all sources to given destination form a tree rooted at the destination



- There are two classes of routing algorithms
  - Non adaptive (static): routing decisions are computed in advance, off line, downloaded to routers at booting time and fixed, e.g. shortest path, flooding, and flow-based
  - Adaptive: routing decisions are adapted to reflect changes in topology and traffic, e.g. distance vector, link state, hierarchical, broadcast, multicast

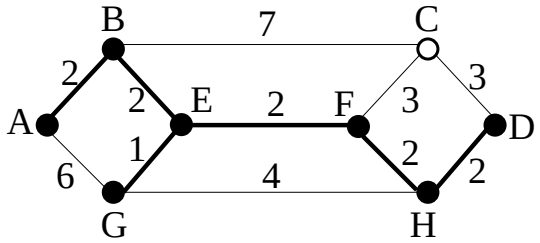
Shortest Path Routing

- It is really the least-cost path routing, based on dynamic programming for optimisation
- Basic idea: at each step, **select a newly reachable node at the lowest cost, and add an edge to that node**, so to connect it to the tree built up so far
- Consider network configuration, where nodes labeled as A to H are routers, and each link has a cost associated with it



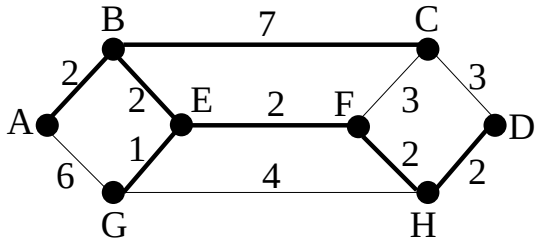
- Least-cost path for $A \longrightarrow D$:

  1. B is lowest-cost node reachable from A, adds edge $A-B$; 2. E is lowest-cost node reachable from the tree with A and B, adds lowest-cost edge $B-E$; 3. G is lowest-cost node reachable from the tree with A, B and E, adds lowest-cost edge $E-G$; 4. F is the one with lowest-cost edge $E-F$; 5. H is the one with lowest-cost edge $F-H$; 6. Adding lowest-cost edge $H-D$ gives "shortest" path from A to D: $A-B-E-F-H-D$



- Sink tree with A at root:

  Incidentally, we also find shortest path from A to G and, according to the optimality principle, those to B, E, F and H. To construct the sink tree with A at its root, we only need to find shortest path from A to C, which is either $A-B-C$ or $A-B-E-F-C$



- What is the sink tree with D at root?

Flooding

- Basic idea: router forwards an incoming packet to all outgoing links except the one that it came in
- Problem: number of packets in circulation just for a single source packet quickly grows without bound, and some measure must be taken to prevent this from happening, called damming flood
  - **Hop count**: each time a router passes a packet, it decrements the hop count in the packet by one. When the hop count reaches zero, the packet is discarded

    This method needs to appropriately set hop count to an initial maximum value
  - **Remember identity**: each router remembers the identity of those packets it has already sent out. When duplicate copies of the packet arrives back, they are discarded

    This is achieved by source router putting a sequence number in the packet. Each router needs a list per source router telling which sequence numbers originating at that source have been seen
  - **Selective**: routers only send an incoming packet out on those lines that make sense

    For this to work, routers must have some ideas of network configuration
- Flooding are impractical for most applications but has two remarkable properties
  - **Robustness**: all possible routes between source and destination are tried. A packet will always get through as long as at least one path between source and destination exists
  - **Optimality**: because all routes are tried, at least on copy of the packet to arrive at destination will have used a minimum-hop route

Distance Vector Routing

- The basic idea: look at costs that your neighbours are advertising to get a packet to a destination; select the neighbour whose advertised cost, added with the cost to get to that neighbour, is lowest

  You also need to advertise your costs to your neighbours too. The cost is usually delay time

- Formally, each router maintains two vectors called **delay** and **successor node**, so router i has

$$D_i = [d_{i1} \cdots d_{iN}]^T \quad \text{and} \quad S_i = [s_{i1} \cdots s_{iN}]^T$$

  where d_{ij} is current estimate of minimum delay from router i to j ($d_{ii} = 0$), N is number of routers in network, and s_{ij} is next node in current minimum-delay route from i to j

- Periodically, each router exchanges its delay vector with all its neighbours, and on the basis of all incoming delay vectors, router k updates both its vectors:

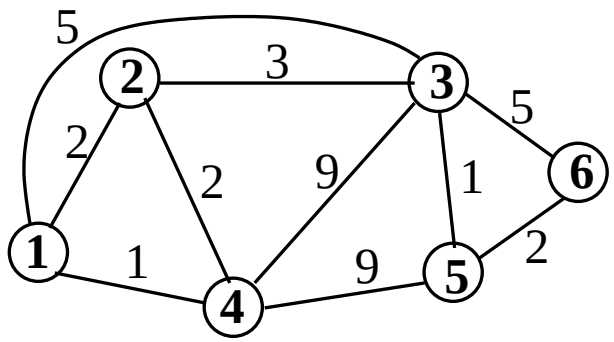
$$d_{kj} = \min_{i \in A} \{l_{ki} + d_{ij}\} \quad \text{and} \quad s_{kj} = i^* \quad \text{with} \quad i^* = \arg\min_{i \in A} \{l_{ki} + d_{ij}\}$$

  where A denotes set of neighbour routers for k, and l_{ki} is current estimate of delay from k to i

- The distance vector routing has some problems: responds slowly to congestion and delay increases, and has been replaced by link state routing

Distance Vector Routing (continue)

- Consider the simple network topology, where the number on each link is the initial link delay



destination	delay	next node
1	0	-
2	2	2
3	5	3
4	1	4
5	6	3
6	8	3

D_1 S_1

router 1 before update

3	7	5
0	4	2
3	0	2
2	2	0
3	1	1
5	3	3

D_2 D_3 D_4

delay vectors sent to router 1 from neighbours

1	0	-
2	2	2
3	3	4
4	1	4
5	2	4
6	4	4

router 1 after update

- After delay vector exchange, router 1 received the delay vectors D_2, D_3 and D_4 from its three neighbours, and this enables it to update its routing table D_1 and S_1:
- Since $l_{12} = 2$, $l_{13} = 5$ and $l_{14} = 1$,

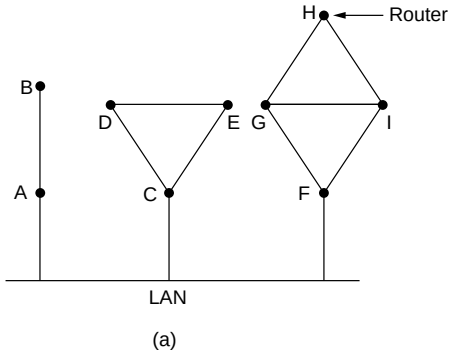
$$d_{13} = \min\{l_{12} + d_{23}, l_{13} + d_{33}, l_{14} + d_{43}\} = \min\{2 + 3, 5 + 0, 1 + 2\} = 3 = l_{14} + d_{43}$$

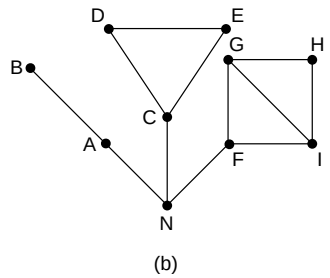
and $S_{13} = 4$, etc.

Link State Routing

- The idea behind the link state routing is simple: each router must do the followings
  - Find out its neighbours and get their network addresses
  - Measure the delay or cost to each of its neighbours
  - Construct a link state packet to tell all it has just learnt
  - Send this packet to all the other routers
  - Find the shortest path to every other router, i.e. a sink tree
- **Who are my neighbours**: A router knows its network interfaces $\longrightarrow$ just sends a HELLO packet on each point-to-point link, and the router at the other end must reply telling who it is with its unique network address

  Situation is complicated with a "broadcast" LAN, i.e. two or more routers are connected by a LAN: solution is to consider such a LAN as a node itself





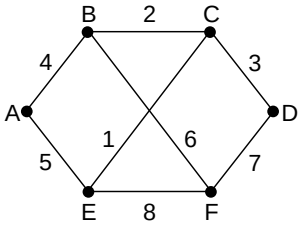
- **What is link cost to neighbour**: Simple, send an ECHO packet to the neighbour, measure the round-trip delay and divide it by two, and this will give a reasonable estimate of the actual delay

  Do we take the local load into account in measuring the delay? Arguments can be made both ways

Link State Routing (continue)

- **Build link state packets**: After collecting information needed, each router builds link state packet with its identity, sequence number and age (used in distributing), followed by list of neighbours (identity and link cost), e.g.

  When to build link state packets: once an hour is often enough



(a)

Link	A	B	C	D	E	F
Seq.						
Age						
B	4	A	4	B	2	C	3	A	5	B	6
E	5	C	2	D	3	F	7	C	1	D	7
		F	6	E	1			F	8	E	8

State Packets

(b)

- **Distribute link state packets**: flooding but keep the flood in check with sequence number

  Each router maintains list of (source, seq.number) pairs they saw. When a LSP arrives, it is checked against the list. If it is new, it is forwarded on all lines except the one it arrived on; if it is a duplicate, it is discarded. To safeguard against old data, link down etc., age is decremented once a second and every time it is forwarded by a router. When the age reaches zero, the LSP is discarded

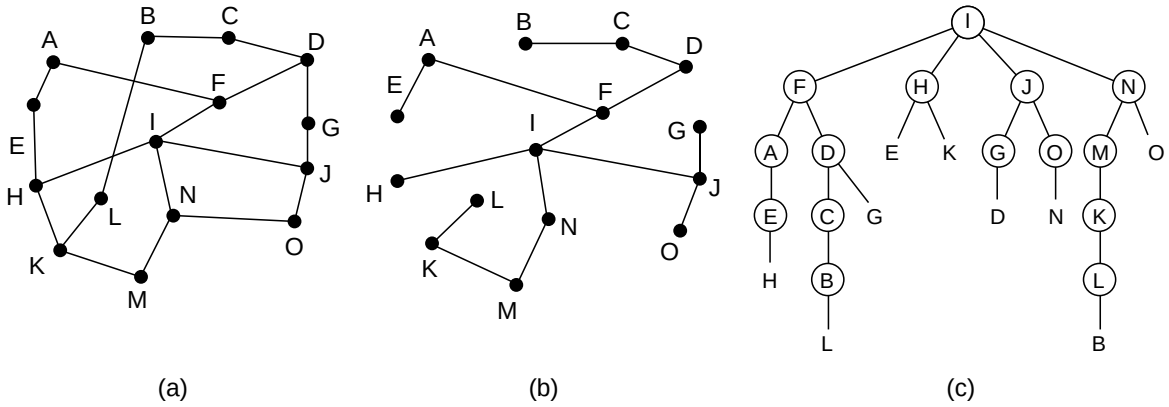
			Send flags			ACK flags			
Source	Seq.	Age	A	C	F	A	C	F	Data
A	21	60	0	1	1	1	0	0	
F	21	60	1	1	0	0	0	1	
E	21	59	0	1	0	1	0	1	
C	20	60	1	0	1	0	1	0	
D	21	59	1	0	0	0	1	1	

  e.g. B: packet source; its sequence number and age; send and ACK flags for each B's three links, A, C and F

- **Compute sink tree**: When a router has all the LSPs, it constructs shortest paths to all possible destinations, i.e. a sink tree, and this is then used in routing decision

Broadcast Routing

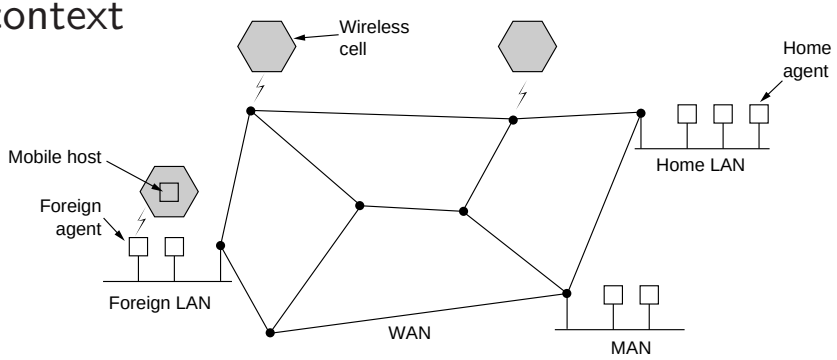
- Sending a packet to every destinations simultaneously is called broadcasting, and various ways are:
  - Send an individual packet addressed to each destination, not really a good idea
  - Use flooding, provided that the flood can effectively be kept in check
  - Use multidestination routing (see Tanenbaum book for details)
  - Build a sink tree rooted at source, and use it in routing (This sink tree is also a spanning tree, as every nodes are on it) $\longrightarrow$ If this can be done, it is great but if not, how to do it approximately?
- **Reverse path forwarding**: Assume normally when router A forwards packet to router B, it uses outgoing link that lies on sink tree rooted at B. This algorithm is remarkably simple:



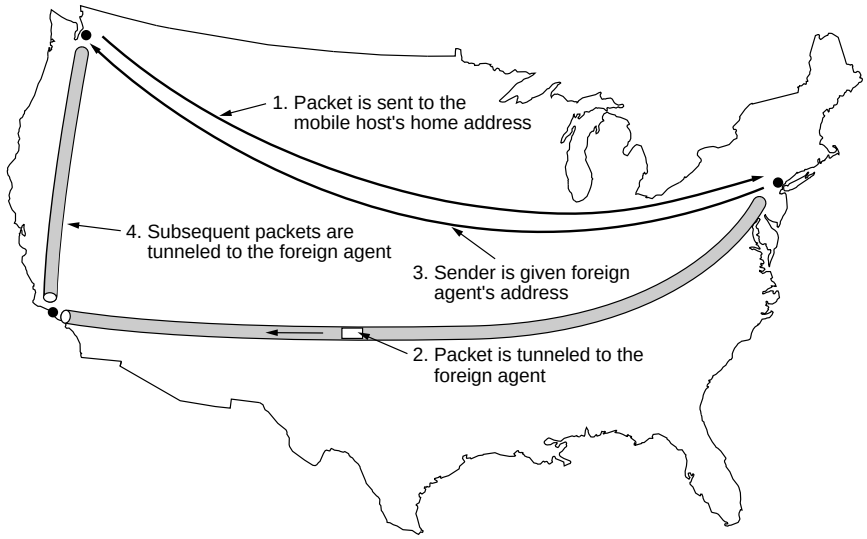
When a broadcast packet arrives, router checks to see if the packet arrived on the line that is normally used for sending packets to the source of broadcast. If so, there is an excellent chance that the best route was used and this is the first copy to arrive at the router. The router then forwards the packet onto all lines except the one it arrived on. If, however, the broadcast packet arrived on a line other than the preferred one for reaching the source, the packet is discarded as a likely duplicate

Routing for Mobile

- How to forward packets to mobile hosts in wide-area context
  - Mobile host has a fixed **home** location with a permanent home address
  - When mobile host enters an area, it must register with **foreign** agent in charge of the area
  - The foreign agent can then inform the mobile's home agent at the mobile's home location that the mobile is under its jurisdiction

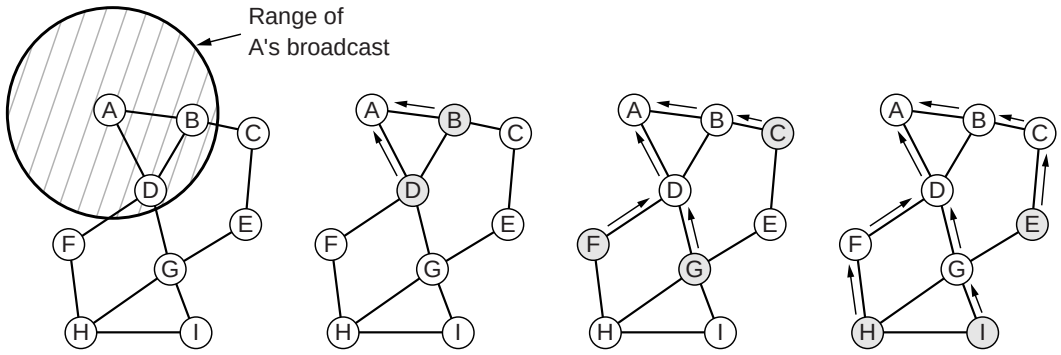


- When a packet is sent to a mobile host, it is routed to the mobile's home address
  - Mobile's home agent is then **tunnelling** it to foreign agent where the mobile is currently in
  - Home agent also informs source where the mobile is
  - New address sent back by home agent enables source to adapt its routing table
  - Subsequent packets can be sent directly to the foreign agent where the mobile is with



Routing in Ad Hoc Networks

- Now, like hosts, routers can come and go. We will discuss a routing algorithm for ad hoc networks, called **ad hoc one-dimensional distance vector**, a distant relative of distance vector routing
- Each node maintains a table, keyed by destination, giving information about that destination, including which neighbour to send packets in order to reach the destination
- **Route discovery**: Consider node A wants to send a packet to node I but it does not know how. A broadcasts a ROUTE REQUEST (RReq) packet
  - When RReq packet arrives a node (B and D in this case as they can receive from A), it is checked to see if it is a duplicate or not. If a duplicate, it is discarded; otherwise do the next
  - If the receiver knows a fresh route to the destination, it sends a ROUTE REPLY (RRep) packet back to sender, basically telling source to "use me" to reach destination; otherwise it rebroadcasts RReq packet



Source address	Request ID	Destination address	Source sequence #	Dest. sequence #	Hop count

ROUTE REQUEST

Source address	Destination address	Destination sequence #	Hop count	Lifetime

ROUTE REPLY

Routing in Ad Hoc Networks (continue)

- **Route discovery**: Eventually, I receives the RReq packet, and it replies with a RRep packet, which is sent back using the route that RReq packet came in and this provides A routing information
- In route discovery, flooding is used, so many measures are employed to keep flood in check, and to make sure the route discovered is a fresh (live) one
- **Route maintenance**: nodes can move or be switched off $\longrightarrow$ network topology can change
  - Periodically, each node broadcasts a Hello message, and each of its neighbours is expected to respond to it
  - If no response is forthcoming, broadcaster knows that the specific neighbour either has moved out of its range or no longer exists
  - Similarly, if a node sends a packet to a neighbour that does not respond, it learns that that neighbour is no longer available
- This information is used to purge routes that no longer work, and also
  - When any of its neighbours becomes unreachable, the node checks its routing table to see which destinations have routes using this now-gone neighbour
  - For each of these routes, the active neighbours are informed that they must do purging too
  - The active neighbours then tell their active neighbours, and so on

Summary

- Routing is a key function at network layer, optimality principle, sink tree
- Non-adaptive routing algorithms:

  shortest path or least cost routing, and (dammed) flooding routing
- Adaptive routing algorithms:

  distance vector routing, and link state routing

  broadcasting routing (reverse path forwarding)

  routing for mobiles in WANs (fixed home address, home agent, foreign agent, tunnelling)

  routing in ad hoc networks (route discovery and maintenance)