Network Routing Algorithms

Network Performance Measures

- Two Performance Measures
  - Quantity of Service (Throughput)
    - How much data travels across the net?
    - How long does it take to transfer long files?
  - Quality of Service (Average packet delay)
    - How long does it take for a packet to arrive at its destination?
    - How responsive is the system to user commands?
    - Can the network support real-time delivery such as audio and video?

Fairness versus Optimality

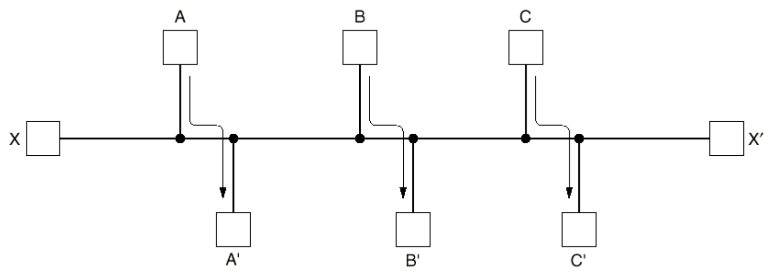


Fig. 5-4. Conflict between fairness and optimality.

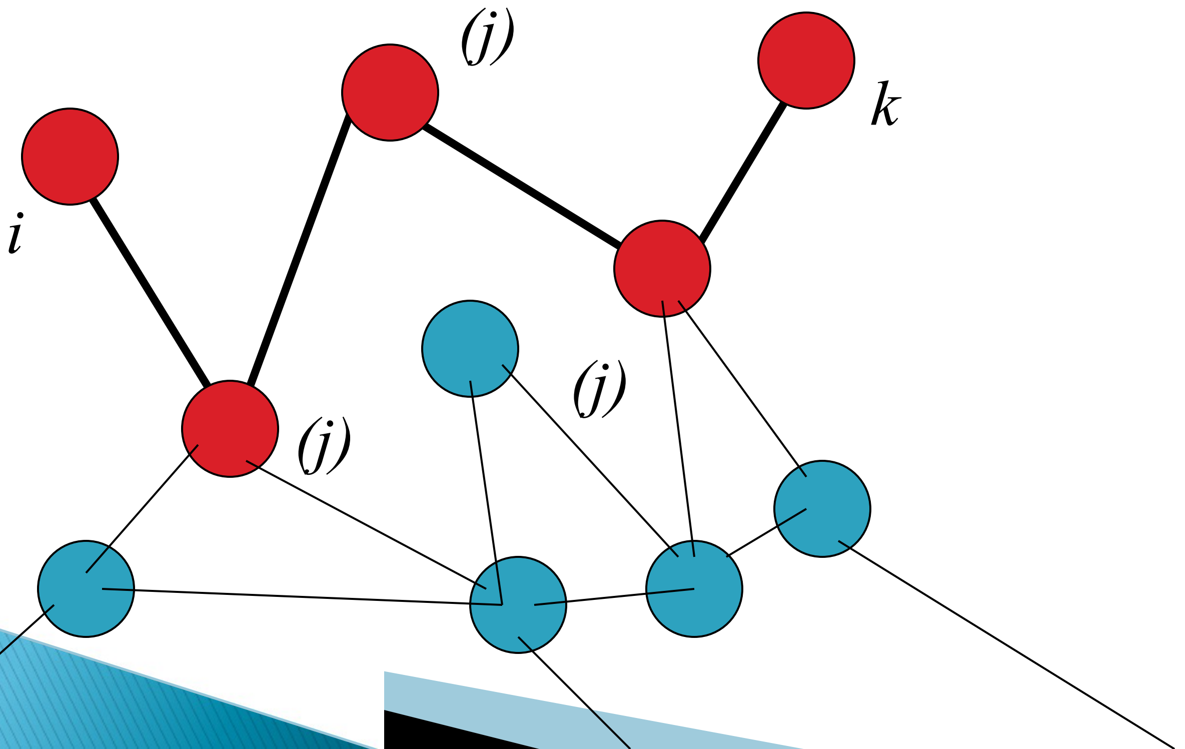
- Quantity of service versus quality of service.
- To optimize throughput, saturate paths between A and A’, B and B’, and C and C’, but what happens to the response time from X to X’?

Types of Routing Algorithms

- Nonadaptive (static)
  - Do not use measurements of current conditions
  - Static routes are downloaded at boot time
- Adaptive Algorithms
  - Change routes dynamically
    - Gather information at runtime
      - locally
      - from adjacent routers
      - from all other routers
    - Change routes
      - Every delta T seconds
      - When load changes
      - When topology changes

Optimality principle

- If router *j* is on the optimal path from *i* to *k*, then the optimal path from *j* to *k* also falls along the same route.



Sink Trees

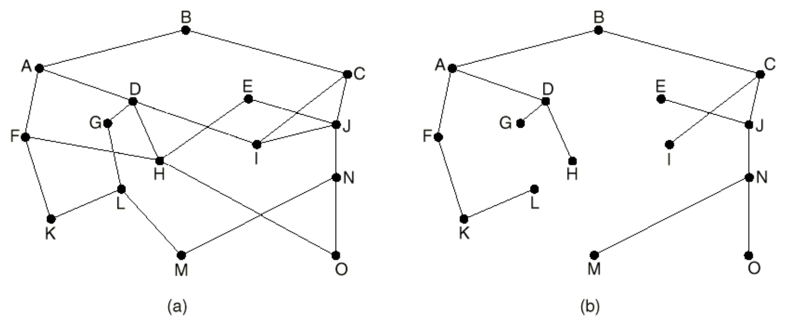


Fig. 5-5. (a) A subnet. (b) A sink tree for router *B*.

- The set of optimal routes to a particular node forms a sink tree.
- Sink trees are not necessarily unique
- Goal of all routing algorithms
  - Discover sink trees for all destinations

Shortest Path Routing
(a nonadaptive routing algorithm)

- Given a network topology and a set of weights describing the cost to send data across each link in the network
- Find the shortest path from a specified source to all other destinations in the network.
- Shortest path algorithm first developed by E. W. Dijkstra

Shortest Path Routing
(a nonadaptive routing algorithm)

Mark the source node as permanent.

Designate the source node as the working node.

Set the tentative distance to all other nodes to infinity.

While some nodes are not marked permanent

- Compute the tentative distance from the source to all nodes adjacent to the working node. If this is shorter than the current tentative distance replace the tentative distance of the destination and record the label of the working node there.

- Examine ALL tentatively labeled nodes in the graph. Select the node with the smallest value and make it the new working node. Designate the node permanent.

Example of Shortest Path Routing

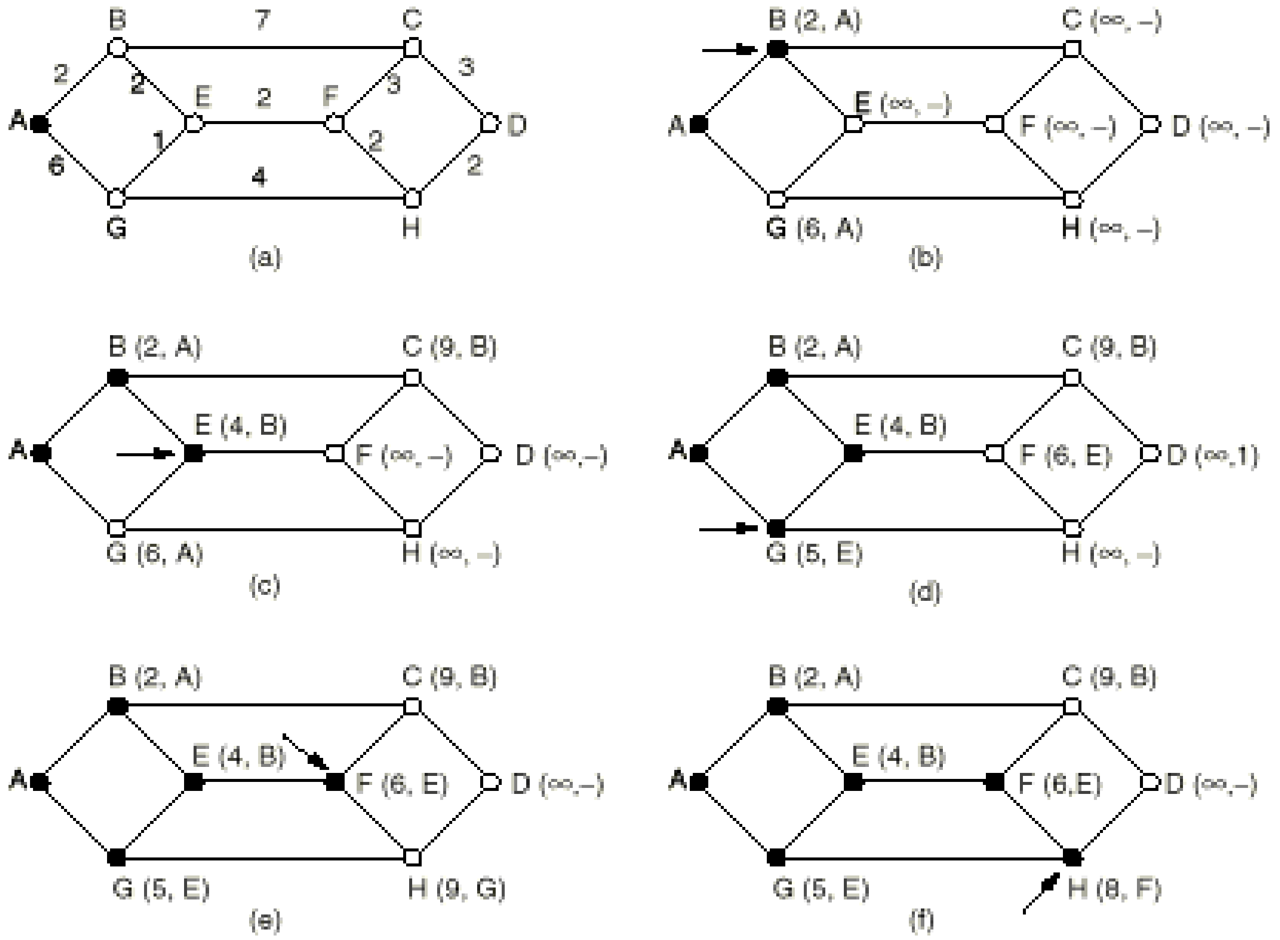
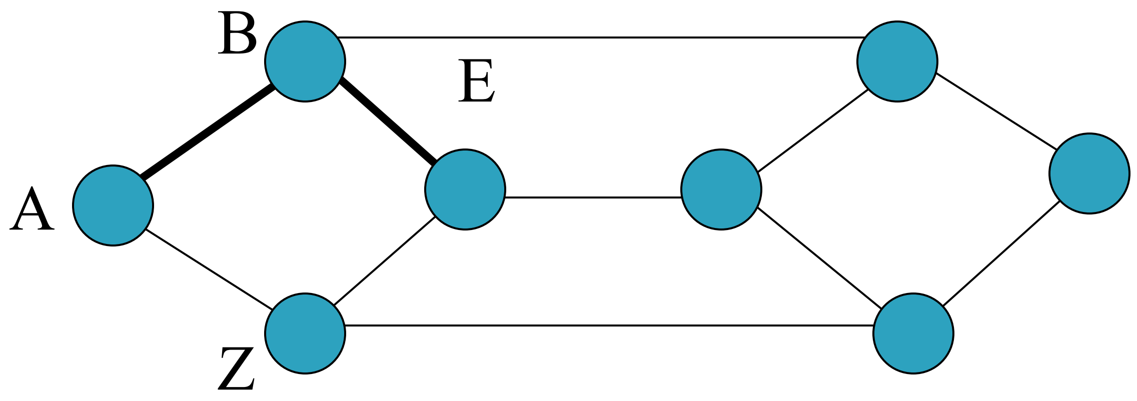


Fig. 5-6. The first five steps used in computing the shortest path from *A* to *D*. The arrows indicate the working node.

Why the Shortest Path Algorithm Works



- Perhaps A*ZE is a better path to E than ABE
- Two cases
  1.) If Z is permanent, then we have already checked A*ZE
  2.) If Z is tentatively labeled, paths to Z must be longer than paths to E, otherwise Z would have been made permanent

Flooding
(a nonadaptive routing algorithm)

- Brute force routing
  - Every incoming packet is sent on every outgoing line
  - Always finds the shortest path quickly
  - Also finds many long paths
  - Time to live is set to size of subnet
- Selective Flooding
  - Flood only in the direction of the destination
- Practical in a few settings
  - Military Applications
  - Distributed Databases
  - Metric for comparison

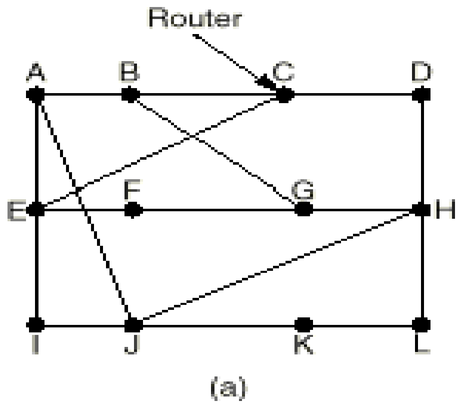
Distance Vector Routing (an adaptive routing algorithm)

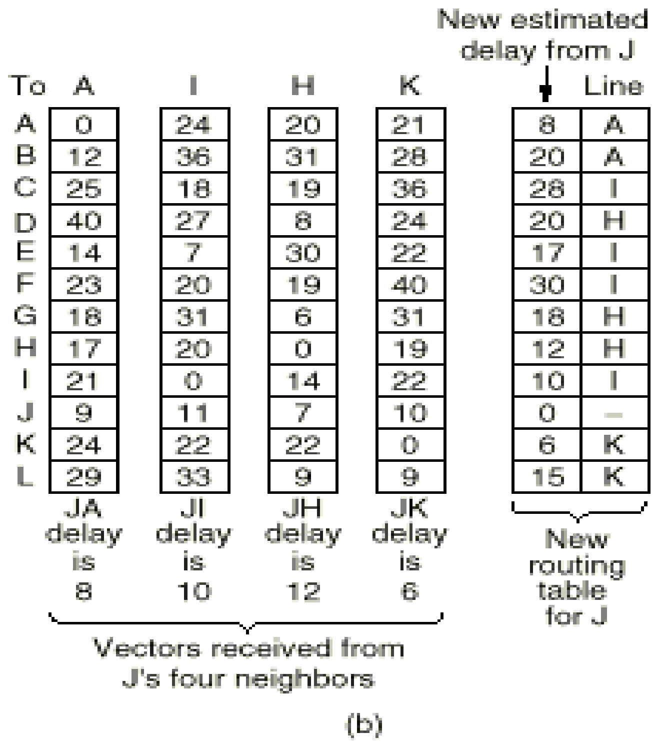
- Bellman–Ford Routing
- Ford Fulkerson Algorithm
- Original ARPANET routing algorithm
- Previously used on Internet (RIP)
- Early version of DecNet and Novell's IPX
- AppleTalk and Cisco routers use improved versions of this algorithm

Distance Vector Routing
(an adaptive routing algorithm)

- Neighboring routers periodically exchange information from their routing tables.
- Routers replace routes in their own routing tables anytime that neighbors have found better routes.
- Information provided from neighbors
  - Outgoing line used for destination
  - Estimate of time or distance
    - can be number of hops, time delay, packet queue length, etc.

Distance Vector Routing (an adaptive routing algorithm)





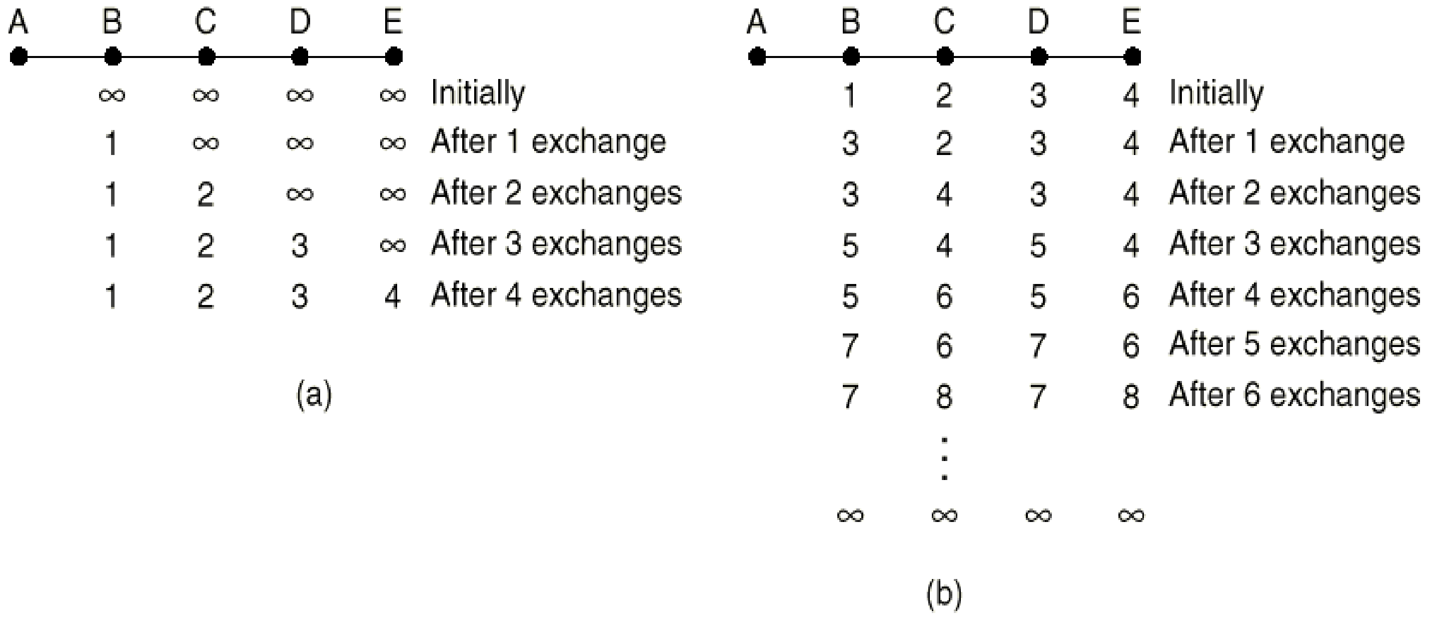
To	A	I	H	K	New estimated delay from J	Line
A	0	24	20	21	8	A
B	12	36	31	28	20	A
C	25	18	19	36	28	I
D	40	27	8	24	20	H
E	14	7	30	22	17	I
F	23	20	19	40	30	I
G	18	31	6	31	18	H
H	17	20	0	19	12	H
I	21	0	14	22	10	I
J	9	11	7	10	0	–
K	24	22	22	0	6	K
L	29	33	9	9	15	K
	JA delay is 8	JI delay is 10	JH delay is 12	JK delay is 6	New routing table for J	

Vectors received from J's four neighbors

(b)

Fig. 5-10. (a) A subnet. (b) Input from *A*, *I*, *H*, *K*, and the new routing table for *J*.

The Count to Infinity Problem



(a)

A	B	C	D	E	
	∞	∞	∞	∞	Initially
	1	∞	∞	∞	After 1 exchange
	1	2	∞	∞	After 2 exchanges
	1	2	3	∞	After 3 exchanges
	1	2	3	4	After 4 exchanges

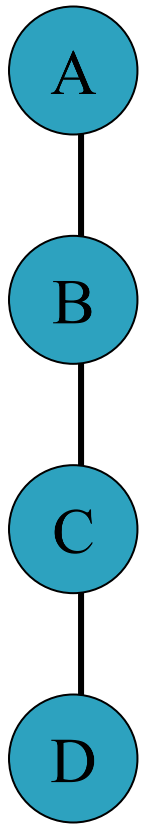
(b)

A	B	C	D	E	
	1	2	3	4	Initially
	3	2	3	4	After 1 exchange
	3	4	3	4	After 2 exchanges
	5	4	5	4	After 3 exchanges
	5	6	5	6	After 4 exchanges
	7	6	7	6	After 5 exchanges
	7	8	7	8	After 6 exchanges
		⋮			
	∞	∞	∞	∞	

Fig. 5-11. The count-to-infinity problem.

The Split Horizon Hack

- Actual distance to a destination is not reported on the line on which packets to that destination are sent.
- Instead these distances are reported as "infinity."



C tells D the truth about its distance to A, but lies to B and says the distance is infinity.

A topology where split horizon fails

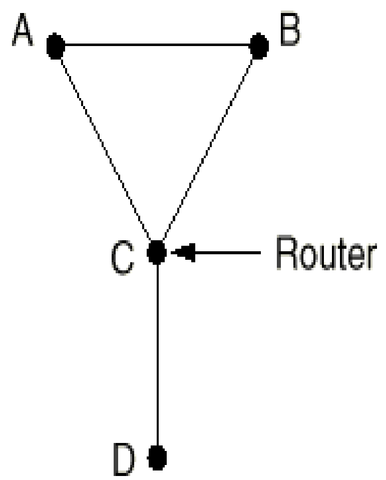


Fig. 5-12. An example where split horizon fails.

Suppose that D becomes unreachable from C.

A and B are reporting infinite distances to C, but they are reporting distances of length 2 to each other.

A and B will count to infinity.

Link State Routing (an adaptive routing algorithm)

- Five Steps

1.) Discover your neighbors and learn their addresses.

2.) Measure the cost (delay) to each neighbor.

3.) Construct a packet containing all this information

4.) Send this packet to all other routers.

5.) Compute the shortest path to every other router.

1.) Discovering Your Neighbors

- Send “Hello” packet on each point-to-point line. Destination node replies with its address.

2.) Measuring Line Cost

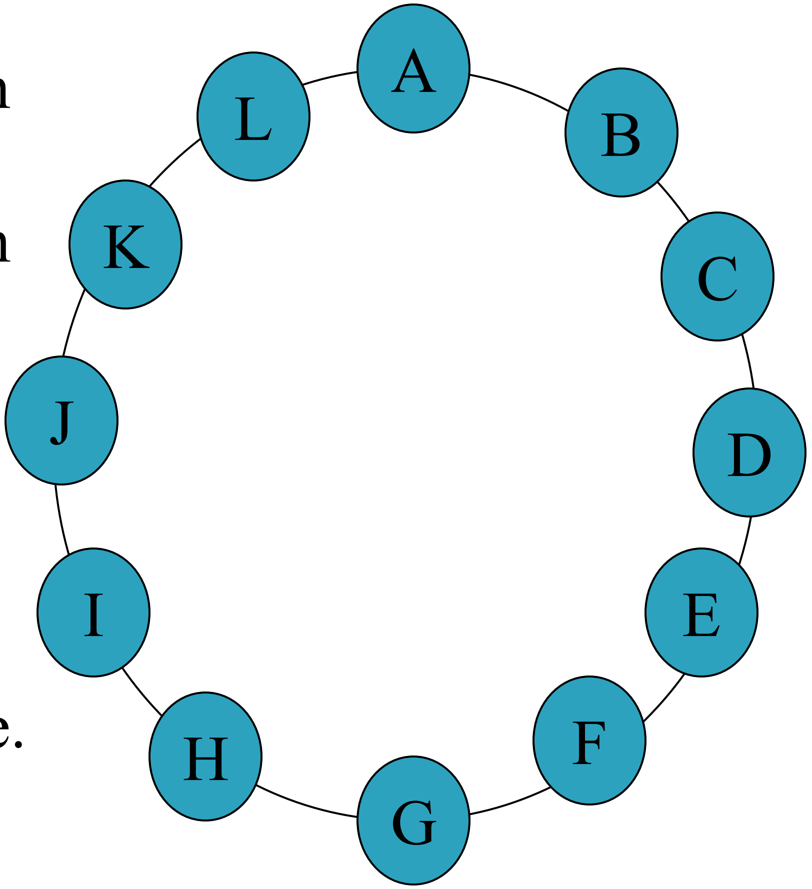
- Send an “ECHO” packet over the line.
- Destination is required to respond to “ECHO” packet immediately.
- Measure the time required for this operation.
- Question: Should we measure just the time it takes to transmit the packet, or should we include the time that the packet waits in the queue?

Argument 2:

- We should include the time that the packet spends in the queue, as this provides a more accurate picture of the real delays.
- We should only include the transmission times, otherwise the network is likely to oscillate between preferred paths.

Oscillating Paths

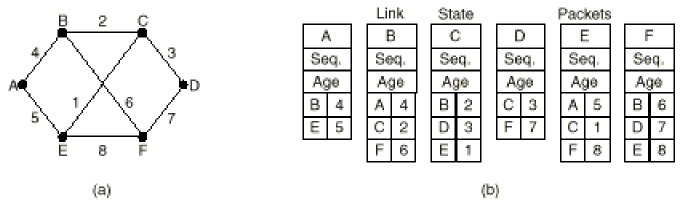
Consider the situation where all nodes are sending to destination A.



The cost of routing clockwise is the number of other nodes routing clockwise.

Each node must determine to either route clockwise or counter clockwise.

Build Link State Packets



A	B	C	D	E	F
Seq.	Seq.	Seq.	Seq.	Seq.	Seq.
Age	Age	Age	Age	Age	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

Fig. 5-15. (a) A subnet. (b) The link state packets for this subnet.

Distributing the Link State Packets

- Use selective flooding
- Sequence numbers prevent duplicate packets from being propagated
- Lower sequence numbers are rejected as obsolete

Computing the New Routes

- Dijkstra's Shortest Path algorithm is used to determine the shortest path to each destination.

Hierarchical Routing

- Addresses the growth of routing tables
- Routers are divided into regions
- Routers know the routes for their own regions only
- Works like telephone routing
- Possible hierarchy
  - city, state, country, continent
- Optimal number of levels for an N router subnet is lnN

Routing Mobile Hosts

- Networking portable computers
- Tanenbaum’s proposed solution
  - All mobile agents are assumed to have a permanent home location
  - When a portable computer is attached to a remote network it contacts a process that acts as the local foreign agent.
  - Each home location has a process that acts as the home agent

The Agents on the Network

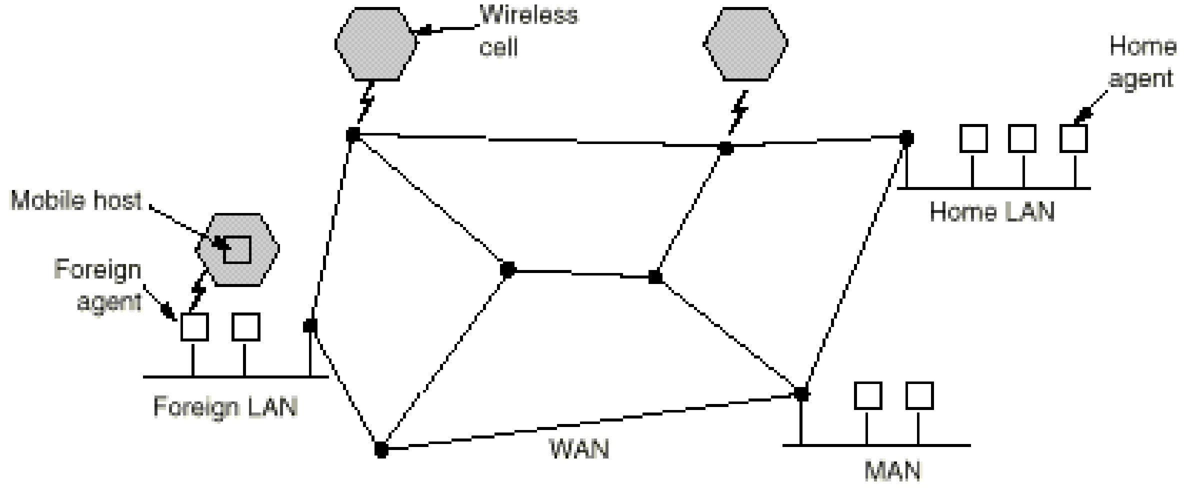


Fig. 5-18. A WAN to which LANs, MANs, and wireless cells are attached.

Registering a Mobile Agent

- Periodically the foreign agent broadcasts its address
- The mobile agent registers with the foreign agent and supplies its home address
- The foreign agent contacts the mobile agent’s home agent reporting the mobile agent’s location.
  - Security must be used to verify the identity of the mobile agent.
- The foreign agent registers the mobile agent

Routing Packets to a Mobile Agent

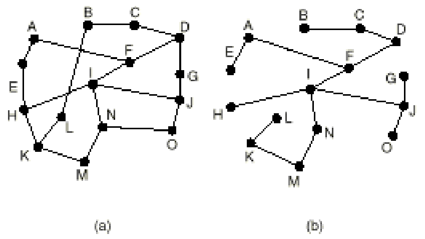
- Packets sent to the mobile agent are routed to the users home network
- The home agent routes the packets to the foreign agent
- The home agent provides the source of incoming packets with the remote address of the mobile agent

Broadcast Routing

- Send a separate packet to each destination
- Use flooding
- Use multidestination routing
  - Each packet contains a list of destinations
  - Routers duplicate packet for all matching outgoing lines
- Use spanning tree routing
  - a subset of the subnet that includes all routers but contains no loops.

Spanning Tree Broadcasting

- Uses the minimum number of packets necessary
- Routers must be able to compute spanning tree
  - Available with link state routing
  - Not available with distance vector routing



Broadcast Routing (continued)

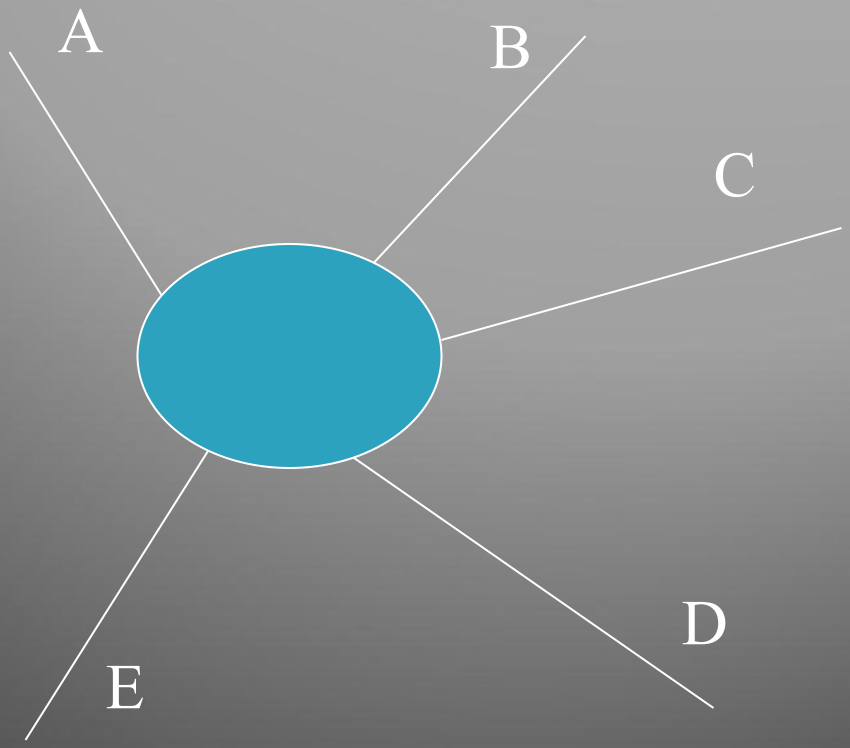
- Reverse Path Forwarding
  - Use When knowledge of a spanning tree is not available
  - Provides an approximation of spanning tree routing
  - Routers check to see if incoming packet arrives from the same line that the router uses to route outgoing packets to the broadcast source
    - If so, the router duplicates the packet on all other outgoing lines
    - Otherwise, the router discards the packet

Reverse Path Forwarding Example

This router routes packets bound for 128.173.41.41 to via line A.

Any broadcast from 128.173.41.41 that arrives from line A is broadcast on lines B, C, D, and E

Any broadcast from 128.173.41.41 that ar
C, D, or E



Multicast Routing

- A method to broadcast packets to well-defined groups
- Hosts can join multicast groups.
  - They inform their routers
  - Routers send group information throughout the subnet
- Each router computes a spanning tree for each group. The spanning tree includes all the routers needed to broadcast data to the group

Spanning Trees for Multicast Routing

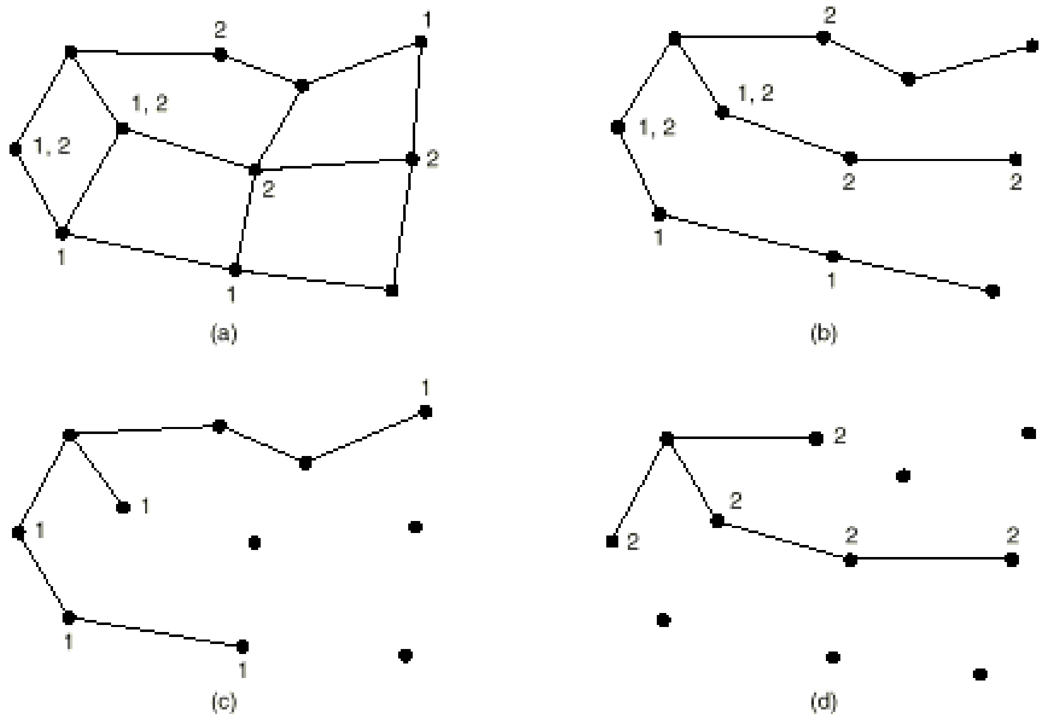


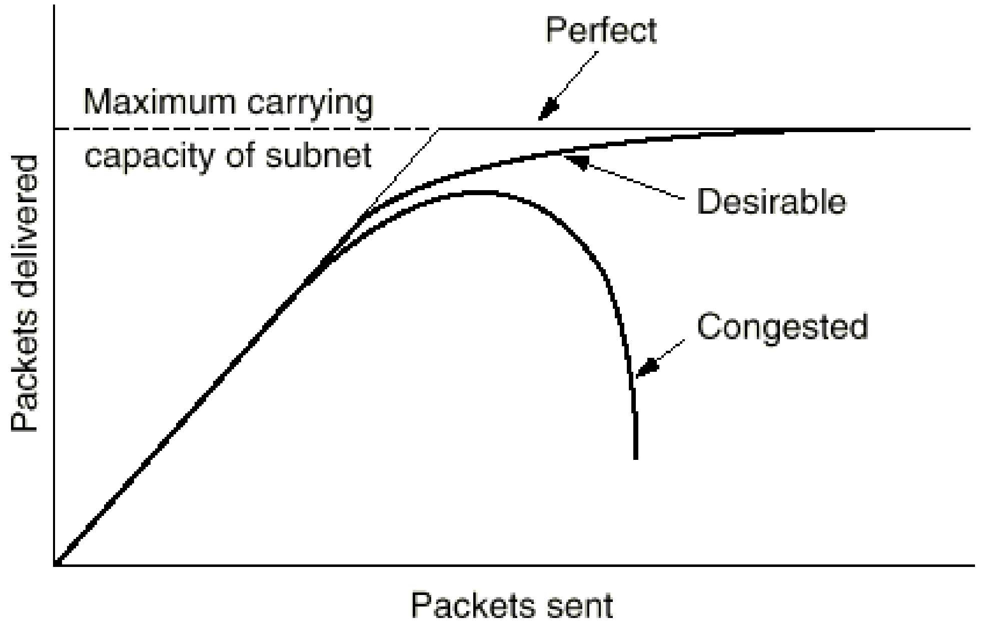
Fig. 5-21. (a) A subnet. (b) A spanning tree for the leftmost router. (c) A multicast tree for group 1. (d) A multicast tree for group 2.

Multicast Routing (continued)

- With Link State Routing the routers are aware of network topology and the spanning tree can be computed
- With Distance Vector Routing reverse path forwarding is used.
  - When a router receives a packet for a multicast group for which it has no subscribers (hosts or other routers), the router sends a PRUNE message to the source router.

Congestion Control Algorithms

- Congestion – the situation in which too many packets are present in the subnet.



Causes of Congestion

- Congestion occurs when a router receives data faster than it can send it
  - Insufficient bandwidth
  - Slow hosts
  - Data simultaneously arriving from multiple lines destined for the same outgoing line.
- The system is not balanced
  - Correcting the problem at one router will probably just move the bottleneck to another router.

Congestion Causes More Congestion

- Incoming messages must be placed in queues
  - The queues have a finite size
    - Overflowing queues will cause packets to be dropped
    - Long queue delays will cause packets to be resent
    - Dropped packets will cause packets to be resent
  - Senders that are trying to transmit to a congested destination also become congested
    - They must continually resend packets that have been dropped or that have timed-out
    - They must continue to hold outgoing/unacknowledged messages in memory.

Congestion Control versus Flow Control

- Flow control
  - controls point-to-point traffic between sender and receiver
  - e.g., a fast host sending to a slow host
- Congestion Control
  - controls the traffic throughout the network

Two Categories of Congestion Control

- Open loop solutions
  - Attempt to prevent problems rather than correct them
  - Does not utilize runtime feedback from the system
- Closed loop solutions
  - Uses feedback (measurements of system performance) to make corrections at runtime.

General Principles of Closed Loop Congestion Control

- Monitor the system to detect when and where congestion occurs.
- Pass this information to places where action can be taken.
- Adjust the system operation to correct the problem.

Metrics Used in Closed Loop Congestion Control

- Percentage of packets discarded due to buffer overflow
- Average queue length
- Percentage of packets that time-out
- Average packet delay
- Standard deviation of packet delay

Reducing Congestion

- Two Methods
  - Increase resources
    - Get additional bandwidth
      - Use faster lines
      - Obtain additional lines
      - Utilize alternate pathways
      - Utilize “spare” routers
  - Decrease Traffic
    - Send messages to senders telling them to slow down
    - Deny service to some users
    - Degrade service to some or all users
    - Schedule usage to achieve better load balance

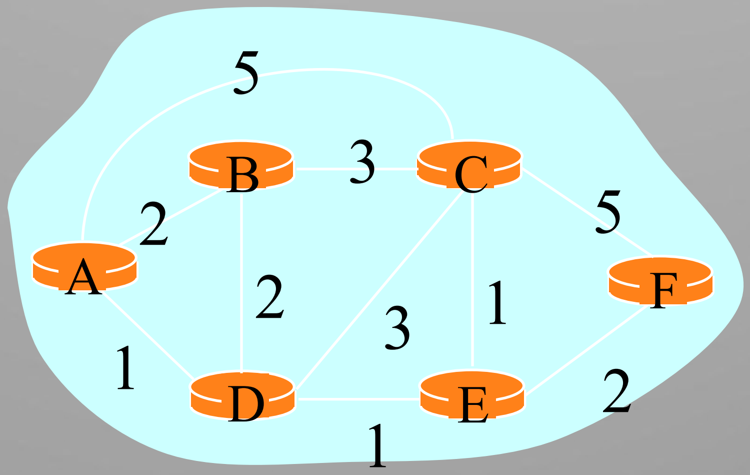
Network Routing: algorithms & protocols

Goal: find "good" path to each destination

- Graph abstraction of a network
  - Nodes: routers
  - Edges: physical links (with assigned cost)

route computation algorithms

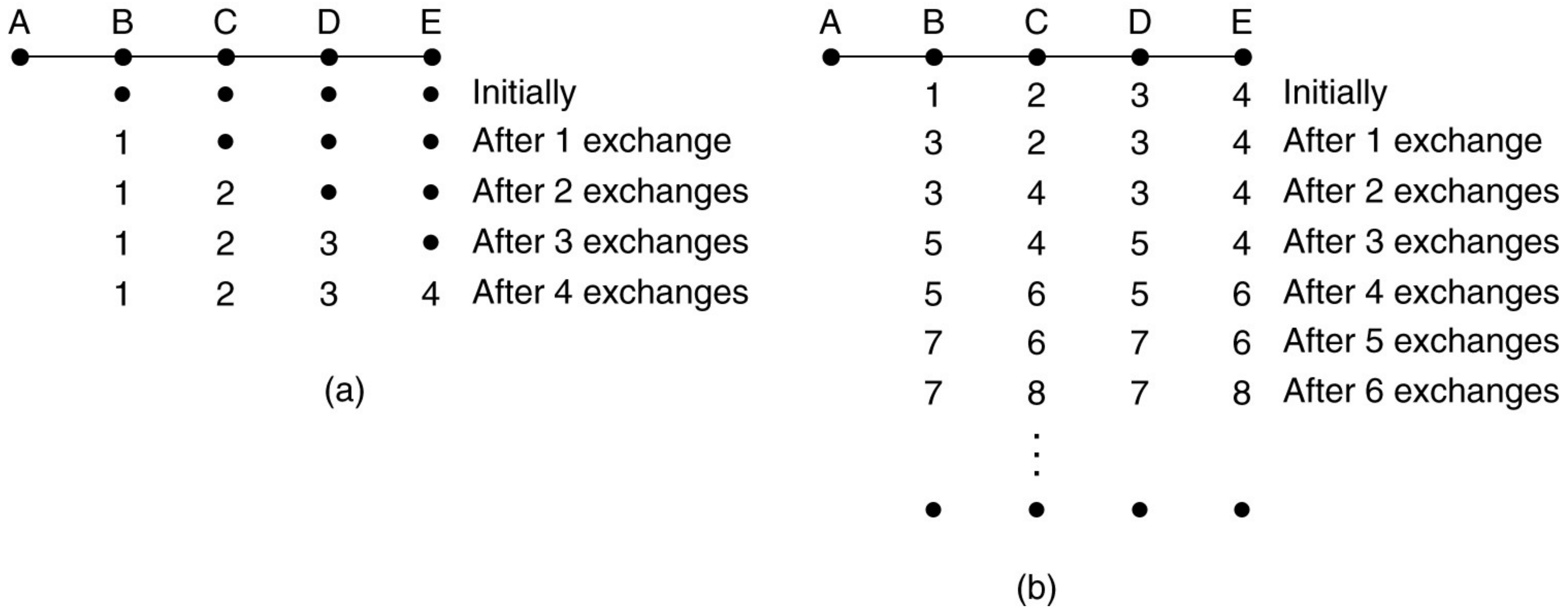
- link-state (Dijkstra)
  - each router knows complete topology & link cost information
  - Run routing algorithm to calculate shortest path to each destination
- distance-vector (Bellman-Ford)
  - Each router knows direct … ts to neig…
  - Calculate th… st path to each destination th… ough an



Routing protocols

- define the format of routing information exchanges
- define the computation upon receiving routing updates
- network topology changes over time, routing protocol must continuously update the routers with latest changes

Distance Vector Routing (2)


A B C D E
• • • • Initially
1 • • • After 1 exchange
1 2 • • After 2 exchanges
1 2 3 • After 3 exchanges
1 2 3 4 After 4 exchanges
(a)
A B C D E
1 2 3 4 Initially
3 2 3 4 After 1 exchange
3 4 3 4 After 2 exchanges
5 4 5 4 After 3 exchanges
5 6 5 6 After 4 exchanges
7 6 7 6 After 5 exchanges
7 8 7 8 After 6 exchanges
• • • •
(b)

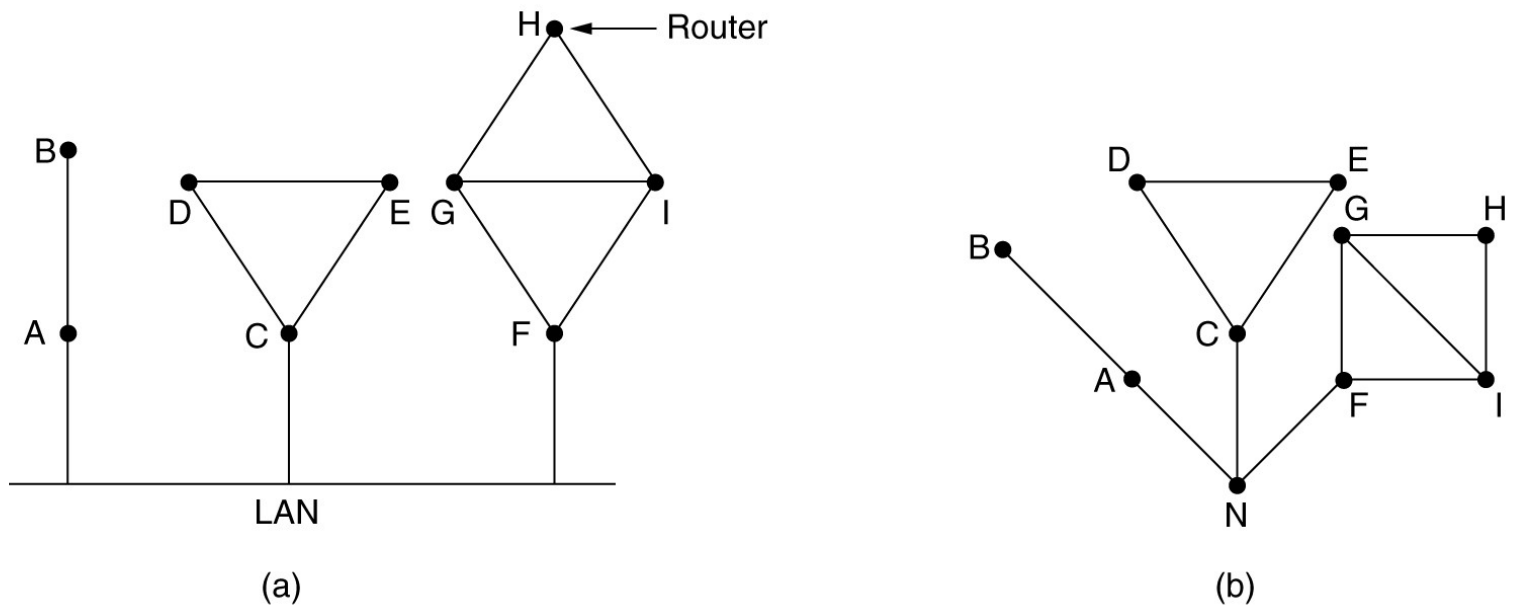
The count-to-infinity problem.

Link State Routing

Each router must do the following:

1. Discover its neighbors, learn their network address.
2. Measure the delay or cost to each of its neighbors.
3. Construct a packet telling all it has just learned.
4. Send this packet to all other routers.
5. Compute the shortest path to every other router.

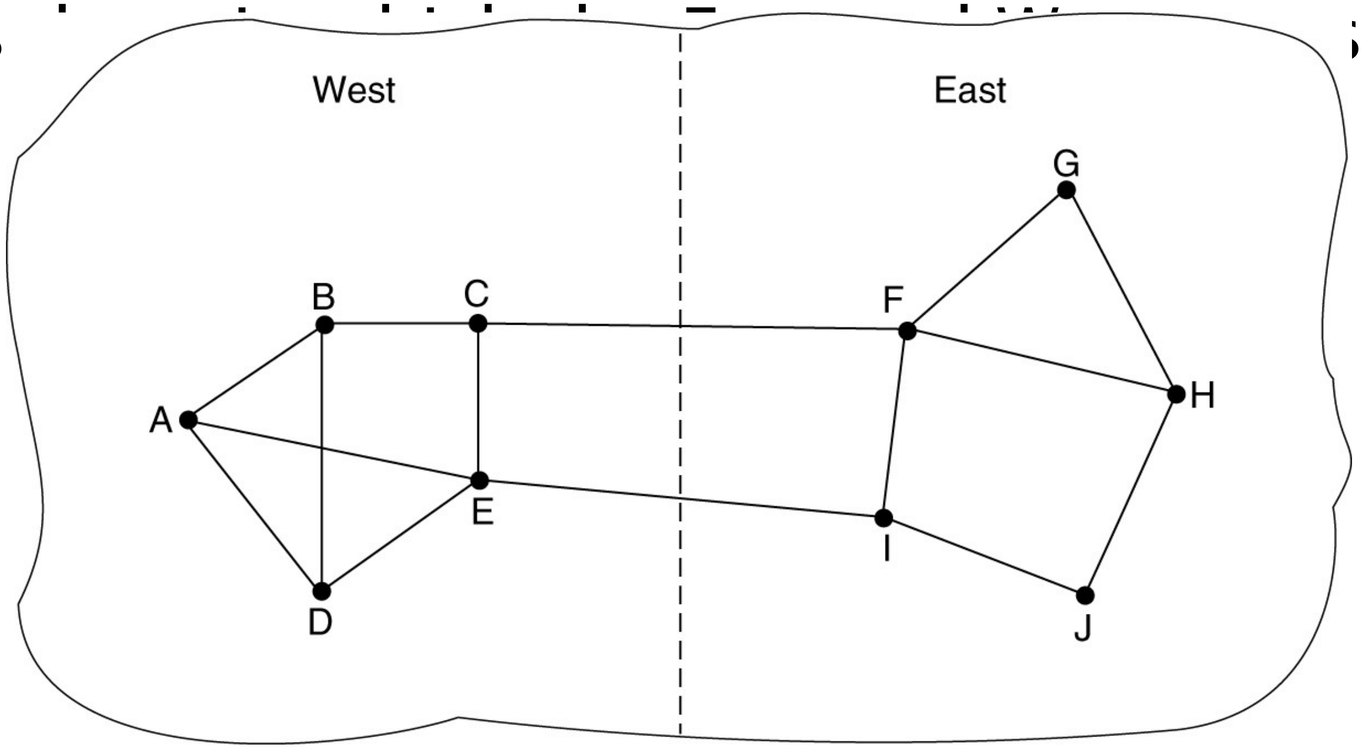
Learning about the Neighbors



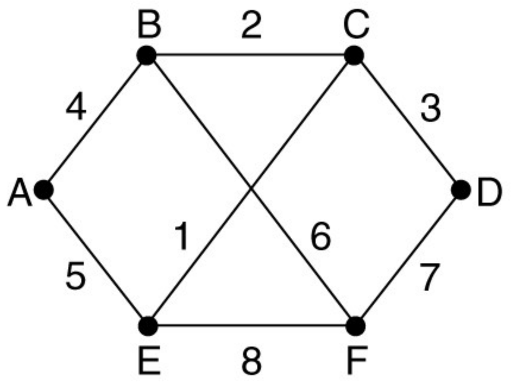
(a) Nine routers and a LAN. (b) A graph model of (a).

Measuring Line Cost

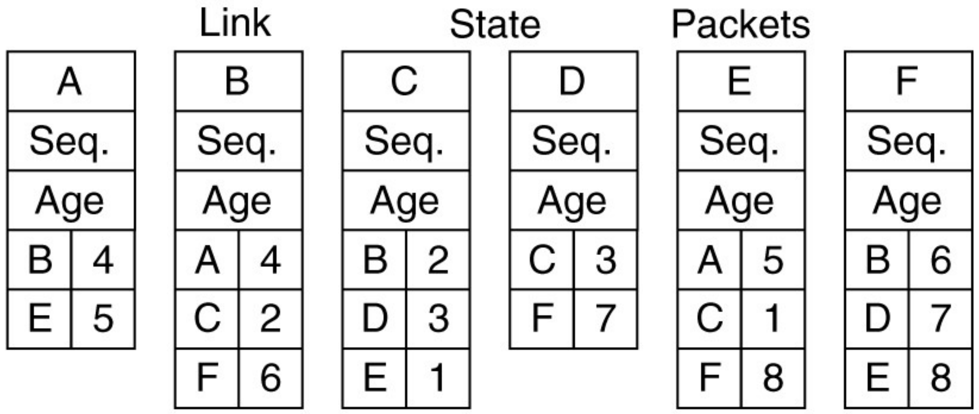
A s ; are



Building Link State Packets



(a)



Link State Packets

A		B		C		D		E		F	
Seq.		Seq.		Seq.		Seq.		Seq.		Seq.	
Age		Age		Age		Age		Age		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

(b)

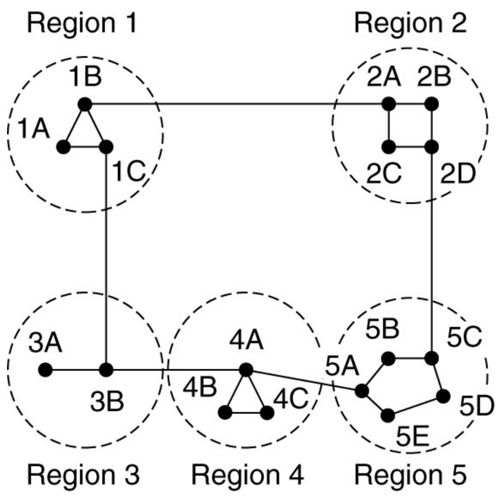
(a) A subnet. (b) The link state packets for this subnet.

Distributing the Link State Packets

The packet buffer for router B in the previous

			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	

Hierarchical Routing



(a)

Full table for 1A

Dest.	Line	Hops
1A	–	–
1B	1B	1
1C	1C	1
2A	1B	2
2B	1B	3
2C	1B	3
2D	1B	4
3A	1C	3
3B	1C	2
4A	1C	3
4B	1C	4
4C	1C	4
5A	1C	4
5B	1C	5
5C	1B	5
5D	1C	6
5E	1C	5

(b)

Hierarchical table for 1A

Dest.	Line	Hops
1A	–	–
1B	1B	1
1C	1C	1
2	1B	2
3	1C	2
4	1C	3
5	1C	4

(c)