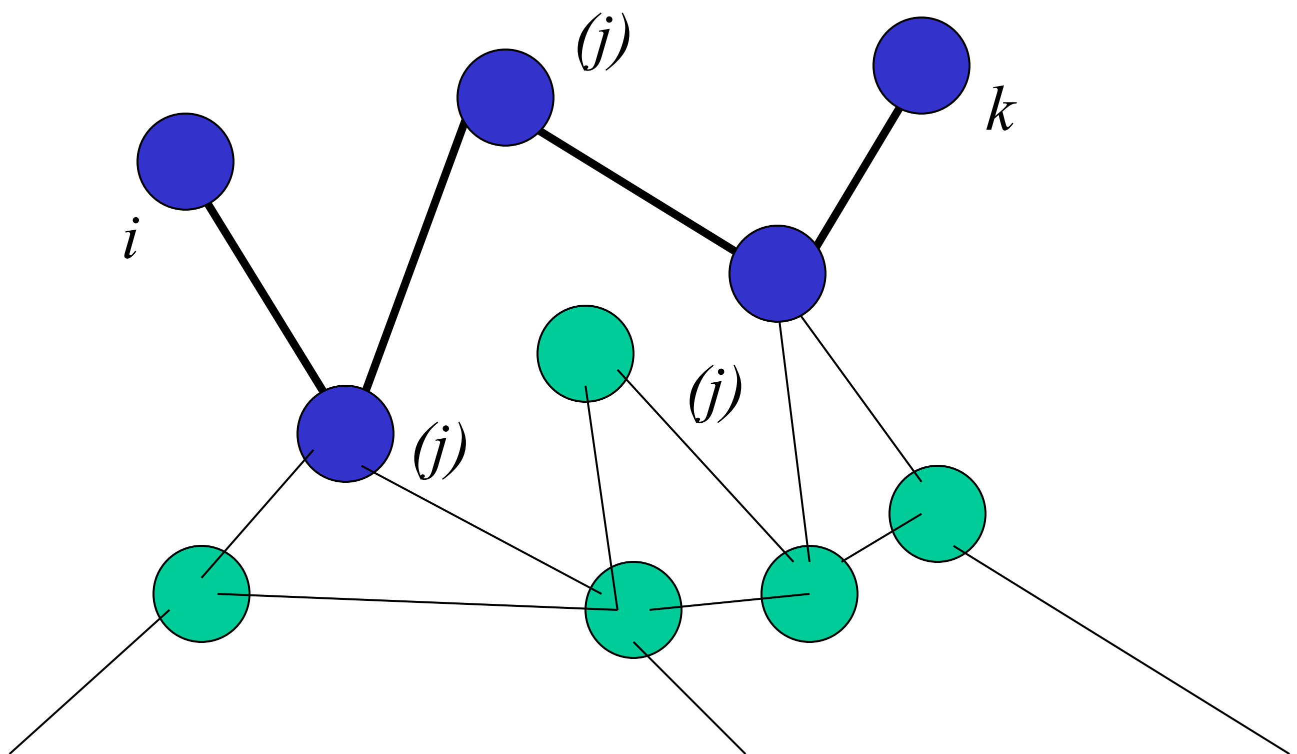
# Chapter 5
# Network Layer

# 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

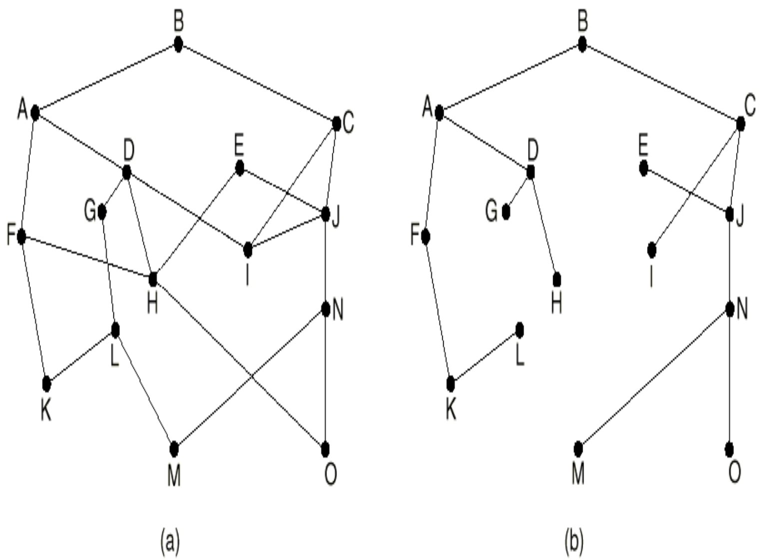



Fig. 5-5. (a) A subnet. (b) A sink tree for router *B*.

- The set of optimal routes to a particular node forms a sink tree.
- Calles as Directed Acyclic Graph (DAG).
- Sink trees are not necessarily unique.
- Goal of all routing algorithms.
  - ✓ Discover sink trees for all destinations.

# Shortest Path Routing

- 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(1)

- Optimization criterion:
  - Distance,
  - Bandwidth,
  - Average Traffic,
  - Communication cost,
  - Mean Queue Length,
  - Measured Delay.
- Algorithms:
  - Dijkstra
  - Flooding
    - Selective Flooding

# Shortest Path Routing(2)

- 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.

# Shortest Path Routing(3) [Example]

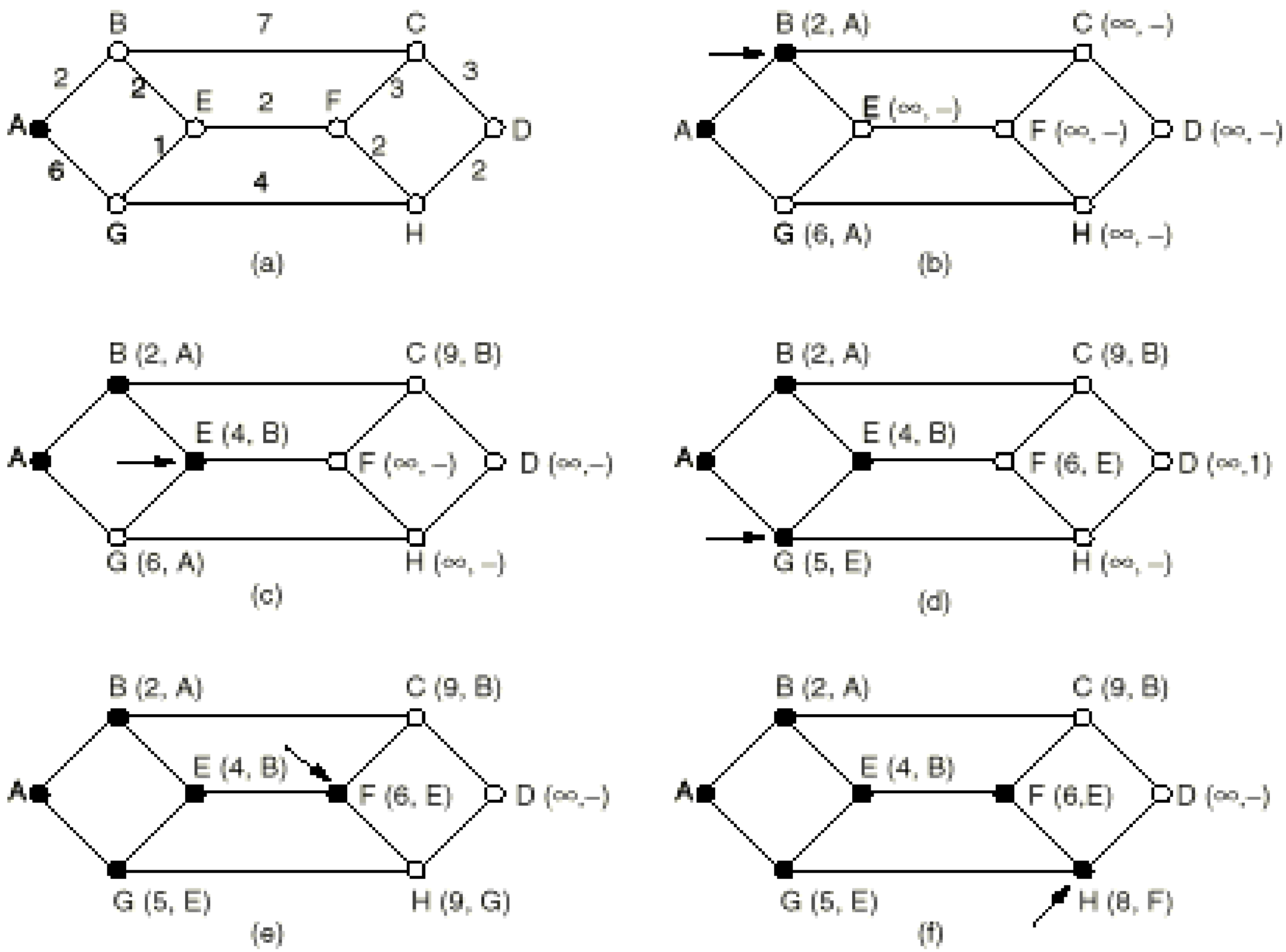



**Fig. 5-6.** The first five steps used in computing the shortest path from *A* to *D*. The arrows indicate the working node.

# Flooding

- Brute force routing
  - Every incoming packet is sent on every outgoing line.
  - Flooding obviously generates vast numbers of **duplicate packets**, in fact, an infinite number unless some measures are taken to damp the process.
  - One such measure is to have a hop counter that is decremented at each hop, with the packet being discarded when the counter reaches zero.

# Flooding

- 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
  - Broadcast
  - Military Applications.
  - Distributed Databases.
  - Metric for comparison.

a) Difference between Dijkstra and flooding algorithm
b) Which one is better?

# Dynamic/Adaptive Routing

- More complex
- More efficient (because they find shortest paths for the current topology)
- Computer networks generally use dynamic routing algorithms.
- Two most popular Dynamic Routing Algorithms are:
  - **Distance Vector Routing**
  - **Link State Routing.**

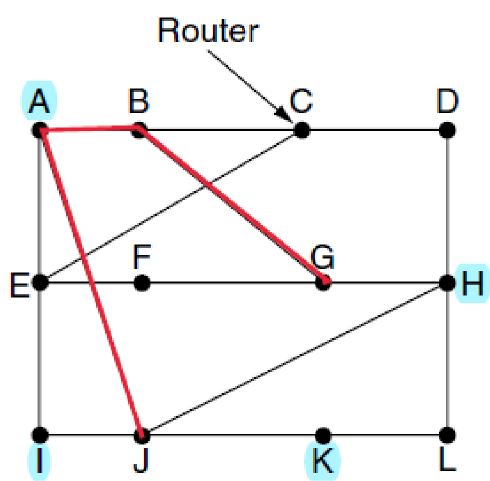
# Dynamic/Adaptive Routing

## ❑ Distance Vector Routing

It operates by having each router maintain a table (i.e., a vector) giving the best known distance to each destination and which link to use to get there.

- ❑ 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




(a)

| 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 | |

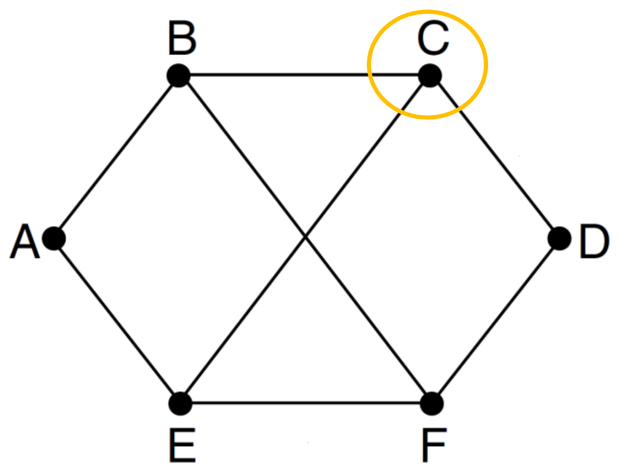
Vectors received from J's four neighbors

(b)

**Figure 5-9.** (a) A network. (b) Input from *A*, *I*, *H*, *K*, and the new routing table for *J*.

# Distance Vector Routing

Consider the network below. Distance vector routing is used, and the following vectors have just come in to router C:

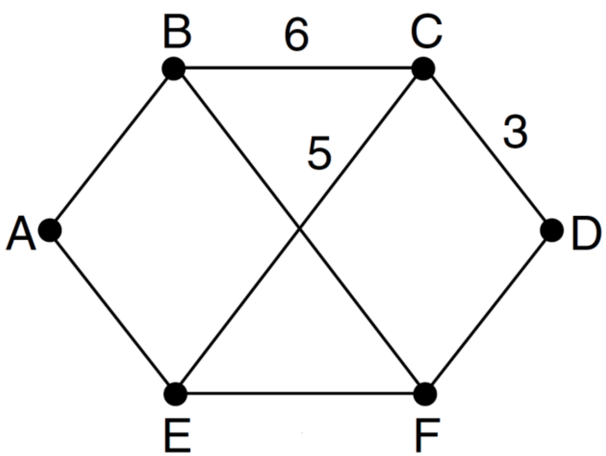



| To | B | D | E |
|---|---|---|---|
| A | 5 | 16 | 7 |
| B | 0 | 12 | 6 |
| C | 8 | 6 | 3 |
| D | 12 | 0 | 9 |
| E | 6 | 9 | 0 |
| F | 2 | 10 | 4 |

Vectors received from C's 3 neighbors

# Distance Vector Routing

Consider the network below. Distance vector routing is used, and the following vectors have just come in to router C:




The cost of the links from C to B, D, and E, are 6, 3, and 5, respectively. What is C's new routing table?

| To | B | D | E |
|---|---|---|---|
| A | 5 | 16 | 7 |
| B | 0 | 12 | 6 |
| C | 8 | 6 | 3 |
| D | 12 | 0 | 9 |
| E | 6 | 9 | 0 |
| F | 2 | 10 | 4 |

# Count–to-Infinity Problem

The settling of routes to best paths across the network is called convergence.

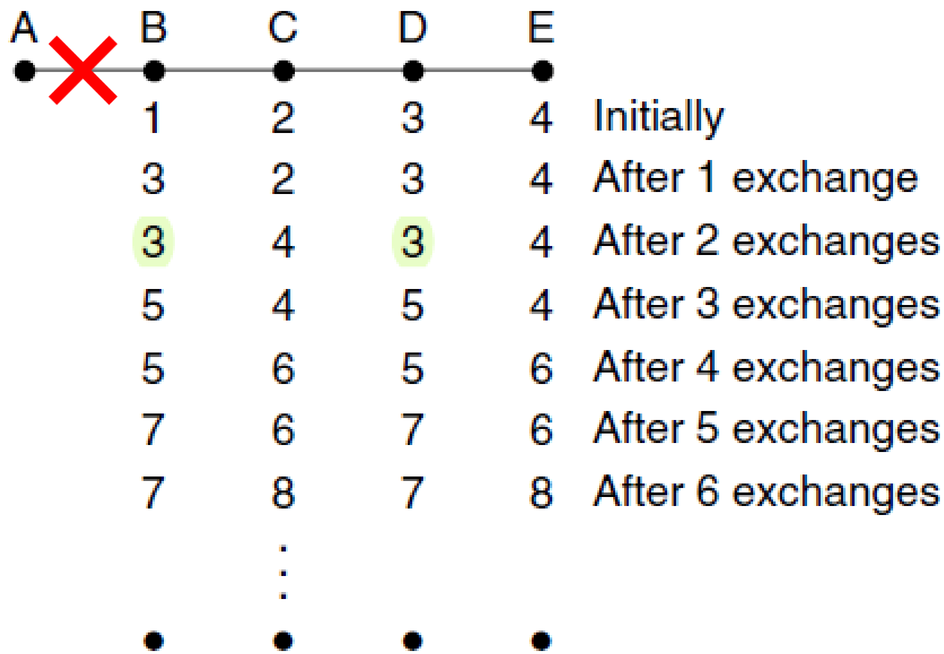
- Bad news" Propagate slowly.
- "Good news" Propagate fast:
  - "Good news" typically refers to an improvement in the network's connectivity or the discovery of a new, shorter path to a destination.

| 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 |

- Only 4 exchange path converges.

# Count–to-Infinity Problem

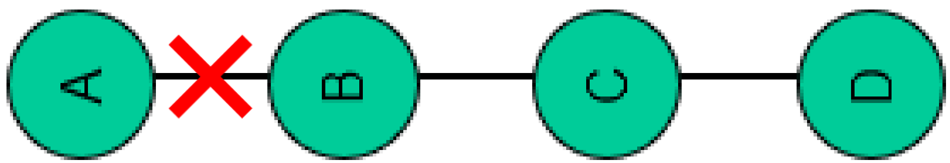
- “Bad news” Propagate slowly.
  - “Bad news" typically refers to a degradation in the network's connectivity or the discovery of a longer path to a destination.




- Even after 6 exchange path doesn’t converges (Slow Converge)

# Count–to-Infinity Problem

- "Bad news" Propagate slowly:
  - ✓ The core of the problem is that when X tells Y that I has a path somewhere, Y has no way of knowing whether it itself is on the path.

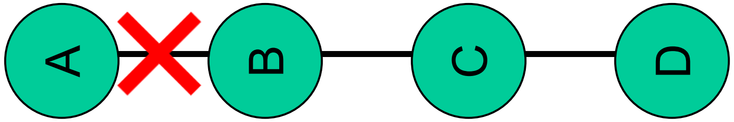



  - ✓ Suppose Link between A and B is down.
    - ✓ C tells B that "I have path from A".
    - ✓ B has no way of knowing whether it itself is on the path.

  - ✓ Solution: Split Horizon & Link State

# 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

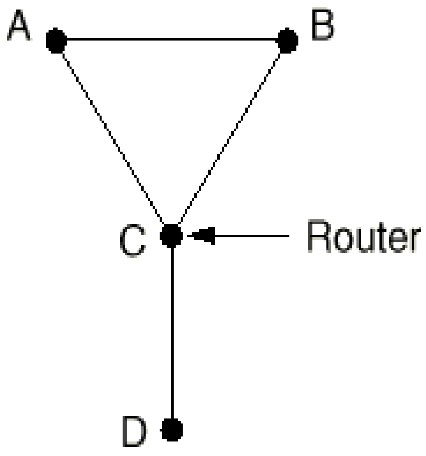



Fig. 5-12. An example where split horizon fails.

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.

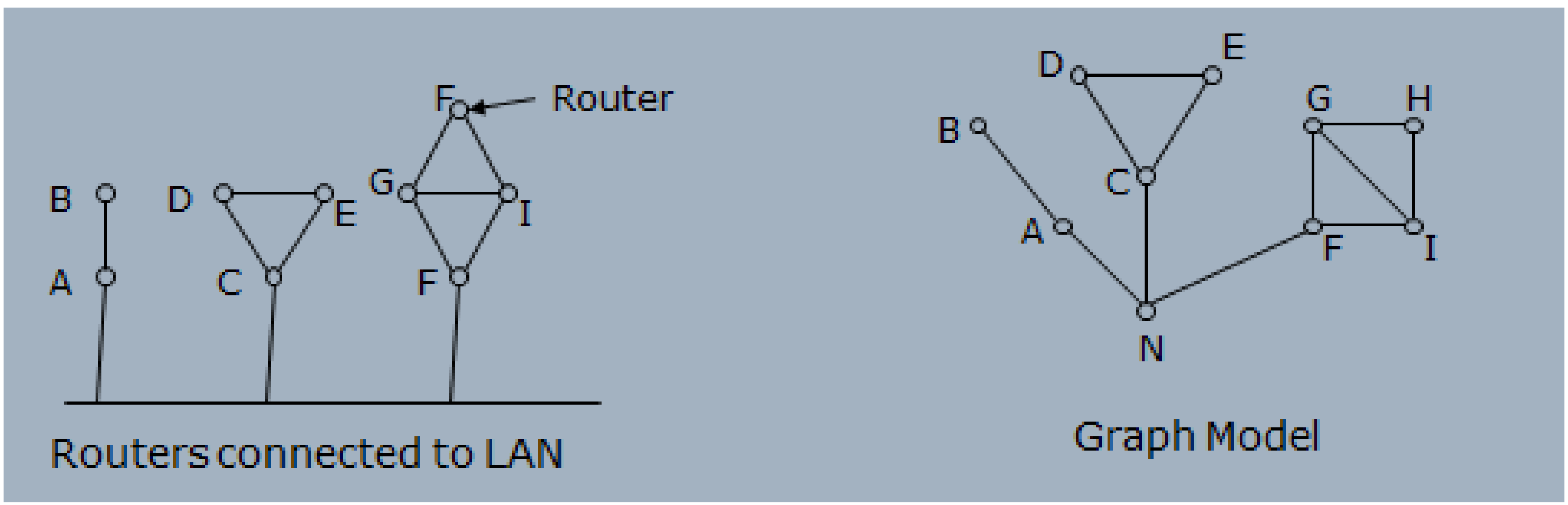
# Link State Routing

Each router must do the following: **Five Steps**

1. Discover its neighbors and learn their network addresses.
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.

# 1. Discovering Your Neighbors

- "HELLO" packed send on each point-to-point line from a booted router.
- Router on the other end must reply by sending its globally unique "name".
- Example of routers connected by a LAN.



# 2. Measuring Line Cost

- It is required by the Link State Routing algorithm that each router to have a reasonable estimate of the delay/cost to each of its neighbors.
  - ✓ Send "ECHO" packet (ping) that the other side is required to send back immediately.
    - Measure Round Trip time; Divide by 2 to get an estimate.
    - More accurate estimate by repeating the process several times and by averaging estimates.
    - Assumes symmetric delay.
- Channel Load Issue when Measuring Delay
  - ✓ To factor the load in: round trip timer must be started when the ECHO packed is queued.
  - ✓ To ignore the load: round trip timer must be started when ECHO packed reaches front of the queue.

# 3. Building Link State Packet

- Packet Format:
  - ✓ Identity of Sender
  - ✓ Sequence Number
  - ✓ Age
  - ✓ List of Neighbors
  - ✓ Corresponding Delay
- Packets easily built – problem with knowing when to built them.

| A | |
|---|---|
| Seq. | |
| Age | |
| B | 4 |
| E | 5 |

| B | |
|---|---|
| Seq. | |
| Age | |
| A | 4 |
| C | 2 |
| F | 6 |

| C | |
|---|---|
| Seq. | |
| Age | |
| B | 2 |
| D | 3 |
| E | 1 |

| D | |
|---|---|
| Seq. | |
| Age | |
| C | 3 |
| F | 4 |

| E | |
|---|---|
| Seq. | |
| Age | |
| A | 5 |
| C | 1 |
| F | 8 |

| F | |
|---|---|
| Seq. | |
| Age | |
| B | 6 |
| D | 4 |
| E | 8 |

# 4. Distributing the Link State Packets

- Distributing Link State Packets ***Reliably*** is tricky:
  - ✓ As the packets are distributed and installed, the routers getting the first ones will change their routes before other routers in the network update their routing tables.
  - ✓ Different Routers may be using different versions of the topology (inconsistencies, loops, etc.)
- Basic Algorithm: ***Flooding***
  - ✓ Sequence Number (incremented for each new packet sent) is used to keep the flood in check.
  - ✓ Routers keep track of all the source router packets they have been sent to.
  - ✓ New link state packets is checked against the track list:
    - If new/unseen (based on the sequence number) then it is broadcasted to all neighboring routers with exception of the sender.
    - If duplicate, it is disregarded.
    - If sequence number is lower than the highest one in the track list, it is rejected.

# 4. Distributing the Link State Packets (2)

- Problems with basic algorithm:
  1. Sequence Number wrap around.
     - Make a long precision number (e.g., 32-bit)
  2. Crash of a router: losing track of sequence number.
  3. Corruption of sequence number.
- Solution: Include Age of each packet.
  - ✓ Decrement this value once per second.
  - ✓ When zero, this state information is disregarded.
  - ✓ Normally a new packed is send every 10 sec.
    - Router information times out when:
      - Router is down, or
      - A Number of (e.g., 6) consecutive packets have been lost.

# Hierarchical Routing

- Large Networks:
  - ✓ Proportionally large routing tables are required for each router
  - ✓ More CPU time is needed to scan them
  - ✓ More bandwidth is needed to send status reports.
  - ✓ At certain point network may grow so large where it is no longer feasible for every router to have an entry for every other router.
  - ✓ Solution: Routing has to be done hierarchically.

# Hierarchical Routing(2)

- Routers divided in Regions (as in telephone network):
  - ✓ Each router knows how to route packets to destinations within its own region.
  - ✓ However, router does not have any information regarding the topology of the network of other regions.
- When different networks are interconnected they are regarded as a separate region in order to free the routers in one network form having to know the topological structure of the other ones.

# Hierarchical Routing(3)

- Huge networks will require more than two-level hierarchy.
- How many hierarchical levels are optimal.
  - ✓ Kamoun and Kleinrock (1979): optimal number for an N router subnet is ln(N), requiring total of e*ln(N) entries per router.
- Example:

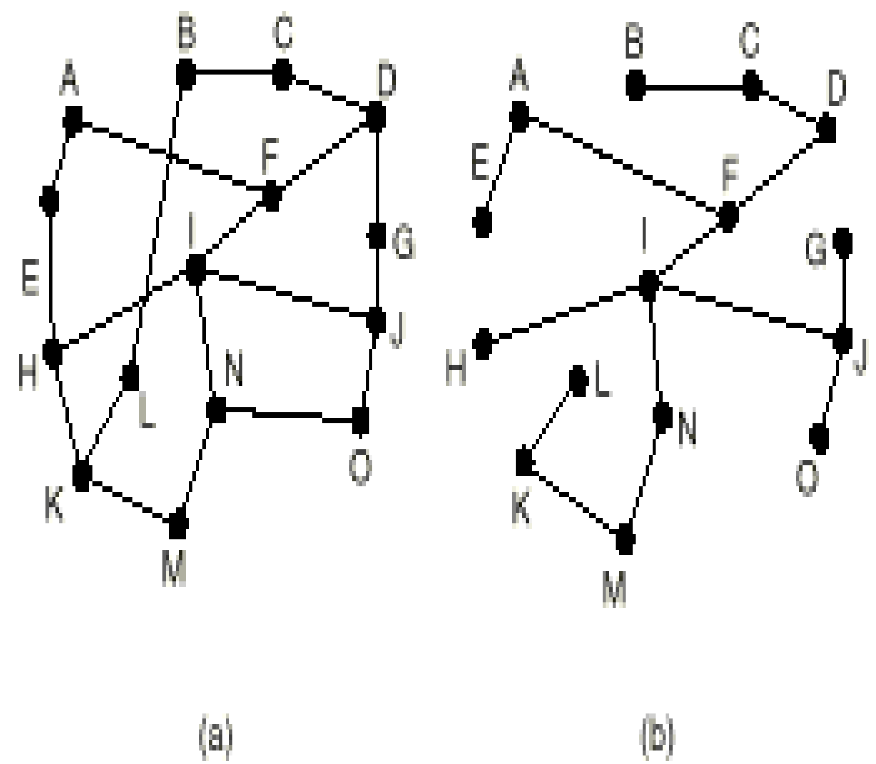
  Berkley, California Router to Malindi, Kenya.
- Berkley to Los Angeles router (out-of-state traffic)
- Los Angeles to New York router (international traffic)
- New York to Nairobi …

# Broadcast Routing

- Sending a packet to all destinations simultaneously is called ***Broadcasting***.
  - Direct Method: Source sends a distinct packet to each destination routers in the subnet:
    1. Wasteful of the bandwidth.
    2. It requires source to have a list of all destinations.
  - Use spanning tree routing
    - a subset of the subnet that includes all routers but contains no loops.
  - Flooding:
    - Ordinarily ill suited for point-to-point communication:
      - Generates to many packets, and
      - Consumes to much bandwidth.

# Spanning Tree Broadcasting(2)

- 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



# 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.
- Use spanning tree protocol.