# Announcements

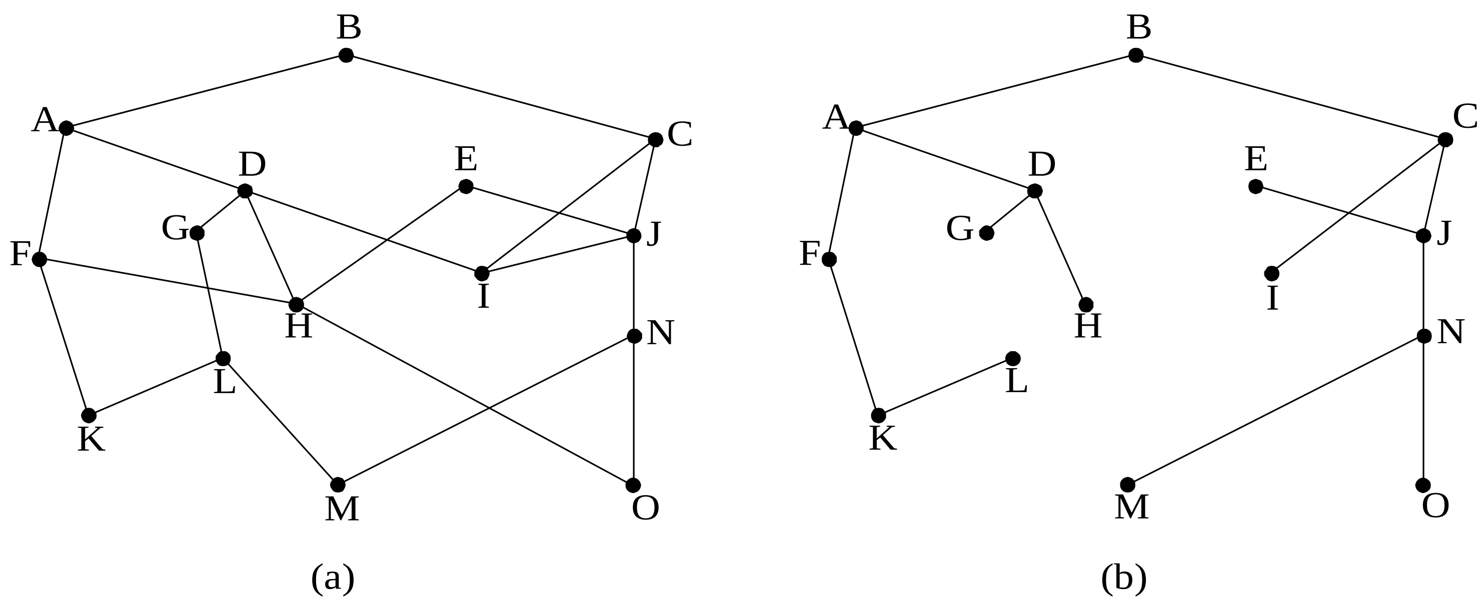
- Reading
  - Today: Chapter 5 (5.2), Pthreads book - Chapter 1
  - Thursday: Pthreads book - Chapters 2 & 3
- Program #2 was distributed

# Do Routes Change During Network Operation?

- nonadaptive routing (static routing)
  - information loaded a boot time
  - never changes during network operation
- adaptive routing
  - changes in network operation alter routes
  - issue: where to get this data to make choices
    - locally from neighbors
    - globally from all routers (or a NIC - network information center)
  - issue: when to change routes
    - only on topology changes (links or routers change)
    - in response to changes in load
  - issue: metric to optimize
    - distance, number of hops, estimated latency

# Optimality Principal

- If J is on the optimal route from I to K
  - then the optimal route from I to K shares the optimal route from J to K
- transitive result of this is a sink tree
  - can construct a tree from all nodes to a specific node



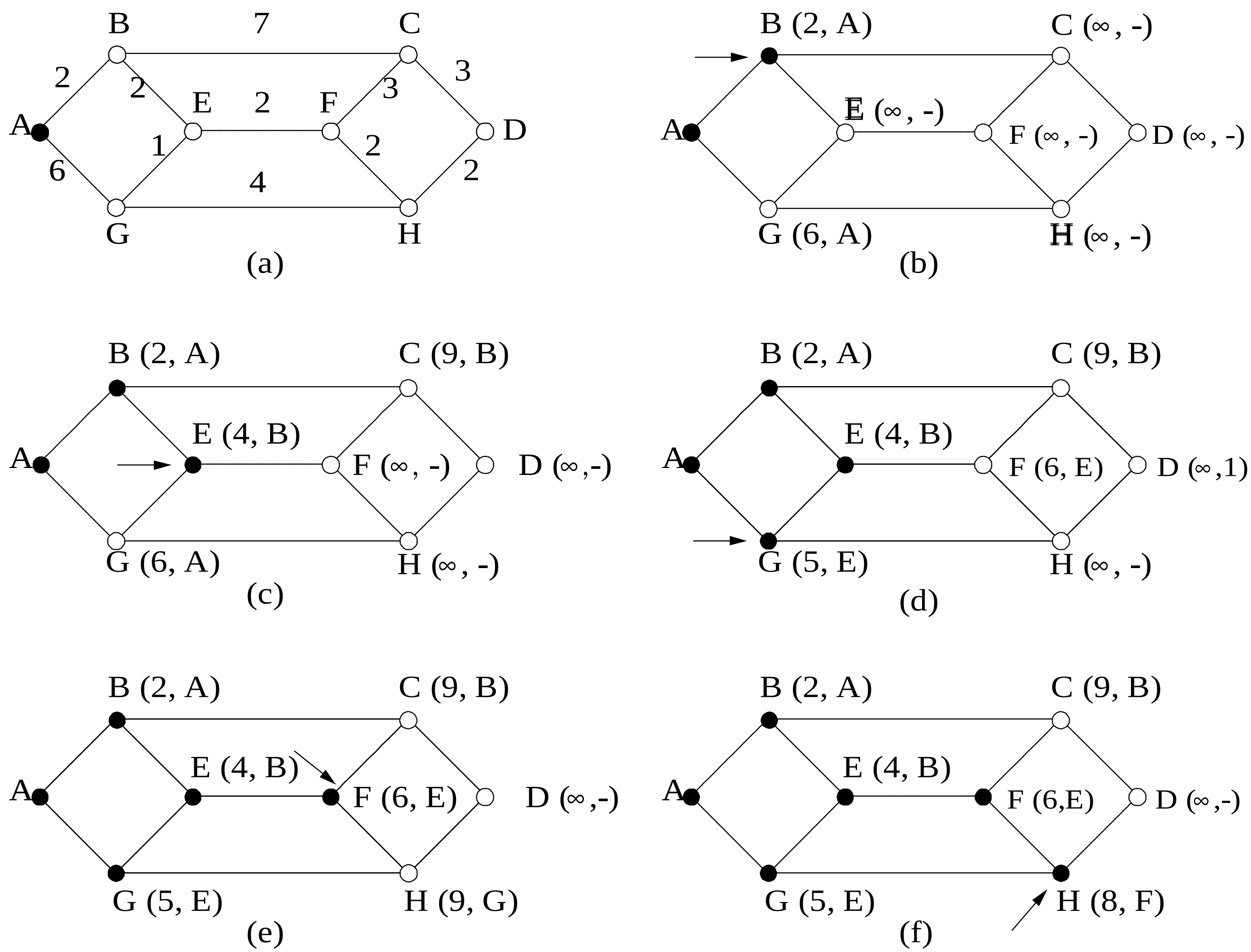

From: *Computer Networks*, 3rd Ed. by Andrew S. Tanenbaum, (c)1996 Prentice Hall.

# Shortest Path Routing

- Graph Representation
  - nodes are routers
  - arcs are links
  - to get between two routes, select a the shortest path
  - need to decide metric to use for minimization
- Dijkstra's Algorithm

```
select source as current node
while current node is not destination
    foreach neighbor of current
        if route via current is better update its tentative route
        label node with <distance, current Node>
    find tentative node with shortest route
        mark a permanent
        make it current
```

# Shortest Path Example




From: *Computer Networks*, 3rd Ed. by Andrew S. Tanenbaum, (c)1996 Prentice Hall.

# Flood Routing

- Every Incoming packet is resent on every outbound link
- generates many duplicate packets
- potentially infinite packets unless they are damped
  - multiple paths to the same destination result in loops
  - can use a lifetime (max hops) to damp traffic
  - can also keep track in routers if the packet has been seen
- good metric to compare algorithms
  - flooding always chooses the shortest path
  - must ignore overhead and congestion due to flooding

# Flow-Based Routing

- Compute optimal routes off-line if we know *in advance*:
  - link capacity
  - topology
  - traffic for foreach <src,dest> pair
- Testing a routing table:
  - given a tentative routing table
  - for each link we can compute mean delay

$$T = \frac{1}{mC - l}$$

  - C is link capacity bps, 1/μ is mean packet size, λ is actual traffic in packets/sec
  - then compute overall utilization (as mean or max of delays)
  - possible to exhaustively try all routing tables this way

# Distance Vector Routing

- Also known as Bellman-Ford or Ford-Fulkerson
  - original ARPANET routing algorithm
  - early versions of IPX and DECnet used it too
- Each router keeps a table of tuples about all other routers
  - outbound link to use to that router
  - metric (hops, etc.) to that router
  - routers also must know "distance" to each neighbor
- Every T sec., each router sends it table to its neighbors
  - each router then updates its table based on the new info
- Problems:
  - fast response to good news
  - slow response to bad news
    - takes max hops rounds to learn of a downed host
    - known as count-to-infinity problem