# Announcements

- Reading
  - Today: Chapter 5 (5.3)
  - Thursday: Chapter 5 (5.4)

# Hierarchical Routing

- Routing grows more complex with more routers
  - takes more space to store routing tables
  - requires more time to compute routes
  - uses more link bandwidth to update routes
- Solution:
  - divide the world into several hierarchies
    - Do I really care that router z at foo U just went down?
  - only store info about
    - your local area
    - how to get to higher up routers
  - optimal number of levels for an N router network is ln N
    - requires a total of $e$ ln N entries per router

# Routing for Mobility

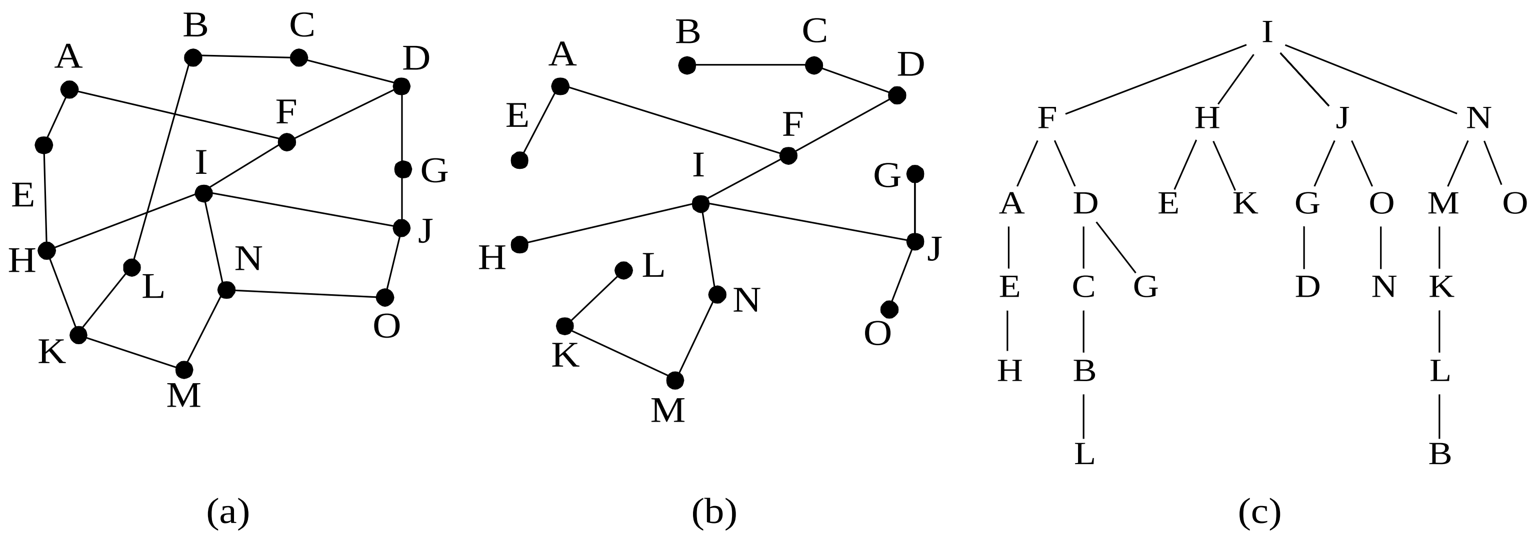
- Or What happens when computers move?
- Two types of mobility:
  - migratory: on the net in many locations but not while in motion
  - roaming: on the net while in motion
- Basic idea:
  - everyone has a home
    - you spend much of your time near home
    - when not at home, they know where to find you
  - home agents: know where you are (or that you are missing)
  - foreign agents: inform home agents of your location
    - informs users that future communication should be sent via them (this is a huge potential security hole)

# Broadcast Routing

- Sometimes information needs to go to everyone
  - routing updates in link-state
  - stock data, weather data, etc.
- sender iterates over all destinations
  - wastes bandwidth
  - sender must know who is interested
- flooding
  - see routing updates for issues
- multi-destination routing
  - routers support having multiple destinations
  - routers copy output packets to correct link(s)
- spanning tree
  - contains subset of graph with no loops
  - efficient use of bandwidth
  - requires info to be present in routers (but it is for link state)

# Routing Broadcast Traffic (cont.)

- Reverse path forwarding
  - check link a packet arrives on
  - if the inbound link is the one the router would use to the source, then
    - forward it out all other links
  - else
    - discard the packet
  - requires no special data sorted in each router




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

# Multicast Routing

- Specify a (relatively) small list of hosts to receive traffic
  - may need to exchange traffic as a group
  - must create/destroy group
- Using spanning trees
  - prune links that are have no members of mulicast group
  - for distance-vector use a variation on reverse path forwarding
    - when a router gets a message it doesn’t need it send a prune message back
    - recursively prunes back un-needed subnets
- core-based trees
  - one tree for group not one per group member
  - hosts send to “core” and it multicasts it out

# Congestion

- Too much traffic can destroy performance
  - goal is to permit the network to operate near link capacity
  - can reach a knee in the packets sent vs. delivered curve
- Sources
  - all traffic is destined for a single out link
    - backup in traffic consumes buffers
    - other (cross traffic) will not get through due to lack of buffers
  - slow router CPU
    - can't service all requests at link speed
      - links still backup
- Often feeds on itself
  - queuing delays can cause packets to timeout
    - introduces more traffic due to re-transmissions

# Congestion Control

- Two possible approaches
  - open loop: prevent congestion from every happening
    - tends to be conservative and result in under utilizaion
  - closed loop: detect and correct
    - some congestion will still occur until it is corrected
- Open loop
  - request resources before using them
  - global (or regional) resource allocation
    - responds yes or no to each request for service
- Closed loop
  - monitor network to detect congestion
  - pass information back to location where action can be taken
  - adjust system operation to correct the problem

# Responding to Congestion

- Add more resources
  - dialup network: start making additional connections
  - SMDS: request additional bandwidth from provider
  - split traffic: use all routes not just optimal
- Decrease load
  - deny service to some users: based on priorities
  - degrade service to some or all users
  - require users to schedule their traffic