CprE 489
Lecture 17

Fall Semester 1997

Routing

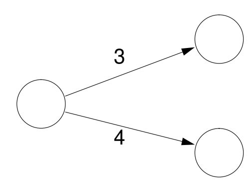
- The network layer provides routing of packets. Routing decisions must be made at each router according to some routing algorithm.
  - packet-by-packet routing as in a datagram service
  - session routing as in a virtural circuit service
- Goals of a routing algorithm
  - correctness
  - simplicity
  - robustness - must run for a very long time without ever crashing network
  - stability - adapts well to changing in the network
  - fairness - customers get their fair share of the network bandwidth and packets are delivered with low delay.
  - high utilization - in high loads the network bandwidth is used to capacity

Routing Algorithms

- Routing algorithms are used for each packet in a datagram service and for set up in a virtual circuit service.
- Routing tables of one form or another are typically used, but not always.
- Adaptive routing - tables vary over time depending on the state of the network.
- Static routing - table are fixed and rarely change. Static routing is typically used when the load does not change much over time.
- Deciding the routes
  - Centralized - a Routing Control Center (RCC) decide based on information from the routers
  - Distributed - Each router builds its routing table based on information from other routers.
    + Total information
    + Partial information

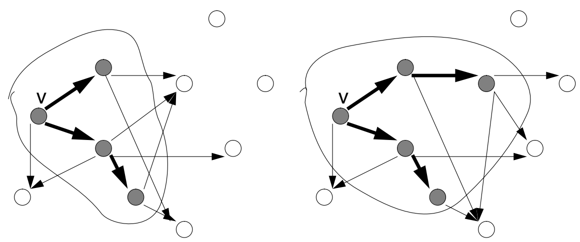
Shortest Path Routing

- Each router has local information about transmission on its outgoing links:
  - delay - queue length and/or physical distance to next router
  - utilization
  - history of past
- With this information a number can be assigned to each outgoing link. Thus, we have a labeled directed graph which indicates the preference of one edge over another on an absolute scale.



Dijkstra’s Algorithm

- Given a node v, it finds the shortest path from v to every other node.
- The general idea: Build a shortest path tree from v.



Dijkstra’s Algorithm Continued

- Each node u has the label length(u) which is the length of the shortest path from v to u which only passes through tree nodes.
- Each tree node u has its label length(u) which is the length of the shortest path from v to u. Following tree edges from v to u gives the shortest path from v to u.
- The algorithm

  - Initialization:

    length(v) :=0; tree(v) = true: new := v;
    length(u) = infinity; tree(u)=false for all other u.

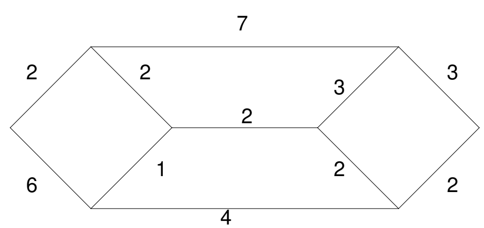
  repeat

  - Update the labels:
    if tree(u) = false, length(u) := min[length(u), length(new) + dist(new,u)]

  - Find new tree node:
    Find the u such that tree(u)=false and length(u) is minimal. new := u

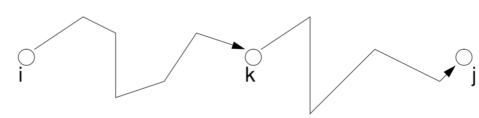
  until all nodes are tree nodes

Example



Consistency Property

- If k is on the route of a shortest path from i to j then the segment of the route from k to j is a shortest path from k to j. This property makes the routing tables consistent with no cycles.



- Thus routing tables can be computed in a distributed fashion without worry about consistency as long as each node uses the same information as all others.