CSE and EE 461
Lecture 19

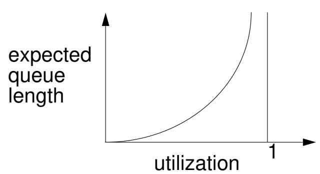
Autumn Quarter 1997

Routing

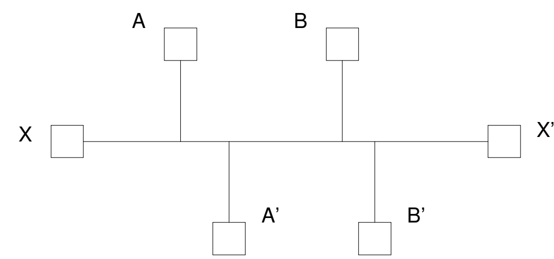
- The network layer provides routing of packets. Routing decisions must be made at each router according to some routing algorithm.
  - packet-by-packet routing as in a datagram service
  - session routing as in a virtural circuit service
- Goals of a routing algorithm
  - correctness
  - simplicity
  - 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

Delay vs. Utilization

- In any statistical queuing system such as a computer network, the higher the utilization the longer the delay.



Fairness vs. High Utilization



- Suppose ε ups need to flow from X to X' and 1 ups each from A to A' and B to B'. Suppose the capacity of the X to X' channel is 1 ups.

  By blocking X to X' traffic we get 2 ups through the system.

  Allowing X to transmit gives us 2 - ε ups through the system.

Congestion

- Congestion can build up for two reasons

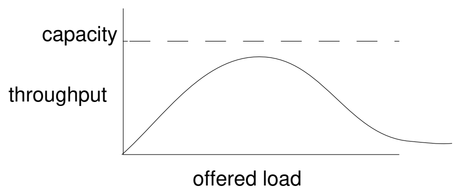
  - routers process packets slowly allowing queues to build up

  - Outgoing lines at routers at capacity causing queues to build up

- Under high loads a network can have a decrease in overall throughput

  - Packets are discarded and resent packets fill the network.

  - If queues are to be built up they should be at the hosts not within the network.



Congestion Control vs. Flow Control

- Congestion is a network wide problem that needs a network wide solution.
- A flow problem is an end-to-end problem which the hosts can manage themselves using sliding windows or other mechanisms.

  Flow control can be used to avoid congestion, but there are also other mechanisms.

Routing Algorithms

- Routing algorithms are used for creating the routing tables in a datagram network service and for facilitating 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 - Link State Routing

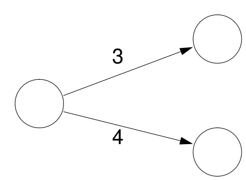
- Each router has local information about transmission on its outgoing links:

  - delay - queue length and/or physical distance to next router

  - utilization

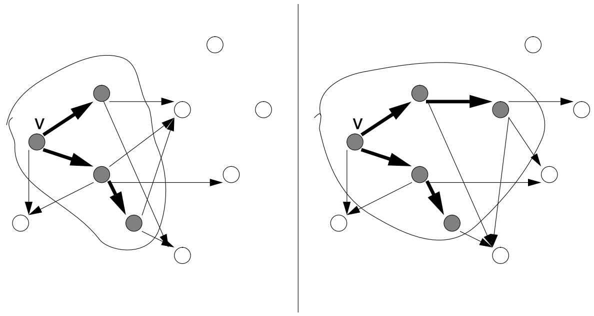
  - history of past

- 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:

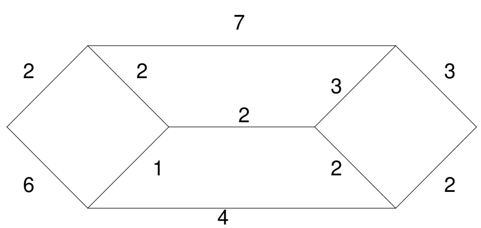
    length(v) :=0; tree(v) = true: new := v;
    length(u) = infinity; tree(u)=false for all other u.

  repeat

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

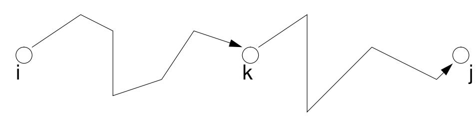
  - Find new tree node:
    Find the u such that tree(u)=false and length(u) is minimal. new := u, tree(new) := true

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.

Distance Vector Routing - RIP

- Old Arpanet standard, still used in the Internet under the name RIP
- Each router, using an ECHO packet determines the distance to its immediate neighbors. Let d_{XA}, d_{XB}, and d_{XC} be the distance from router X to its immediate neighbors A, B, and C.
- Each router keeps an estimate of the current distance to every other router and shares that information with its immediate neighbors. Let D_{AY}, D_{BY} and D_{CY} be the distance from each of X's neighbors A, B, and C to a destination Y.
- Router X updates its routing table for destination Y as follows.

  $D_{XY} = \min\{d_{XA} + D_{AY}, d_{XB} + D_{BY}, d_{CY} + D_{CY}\}$ and the next hop is the router that gives this minimum value.
- Although distance vector is still in use, it has some poor behavior that makes it undesirable. See the count-to-infinity problem.

Alternative Routing Algorithms

- Multipath routing- routing tables contain alternative next hops to the same destination
- hot potato routing - route a packet on the shortest outgoing queue
- delta routing - route a packet on any outgoing link that is within a certain bound delta of the best outgoing link
- random routing - route a packet randomly
- flooding - route duplicates of a packet out all outgoing links

  - to avoid a massive number of duplicates, a copy can be kept so that if a duplicate arrives it is discarded. Must be careful not to discard the copy too soon.

  - flooding is used to send local information to all other nodes to construct routing tables.
- Optimal routing - given all the end-to-end demands compute the best possible routes to meet all the demands.