# More Routing Algorithms

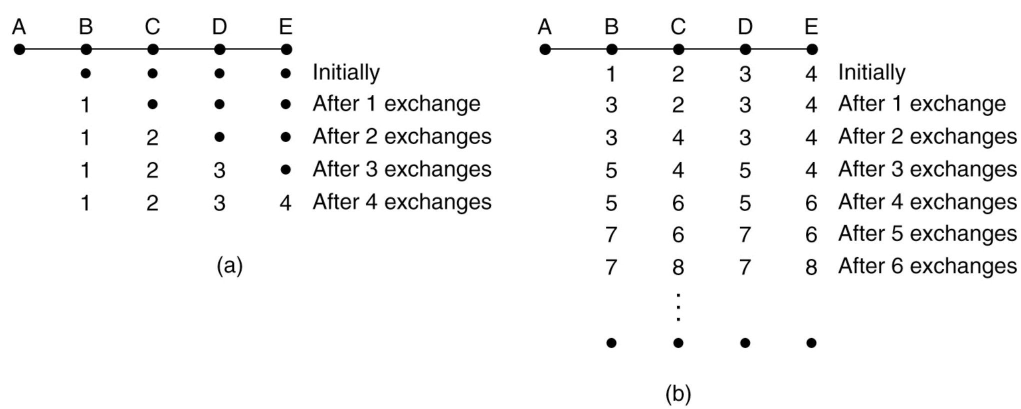
CS158a
Chris Pollett
Apr 9, 2007.

# Outline

- Problems with Distance Vector Routing
- Link State Routing
- Other Kinds of Routing
- Congestion Control Algorithms
- Quality of Service

# Problems with Distance Vector Routing

- Although  distance vector routing does converge to the correct answer, it tends to do it slowly.
- As an example, consider the situations below in a linear network where on the left A comes alive at some time t and on right where it fails at some later time. On the left it takes a significant amount of time for all machines to get the correct distance to A. On the right the machines keep incrementing the distance to A forever (**count to infinity problem)**



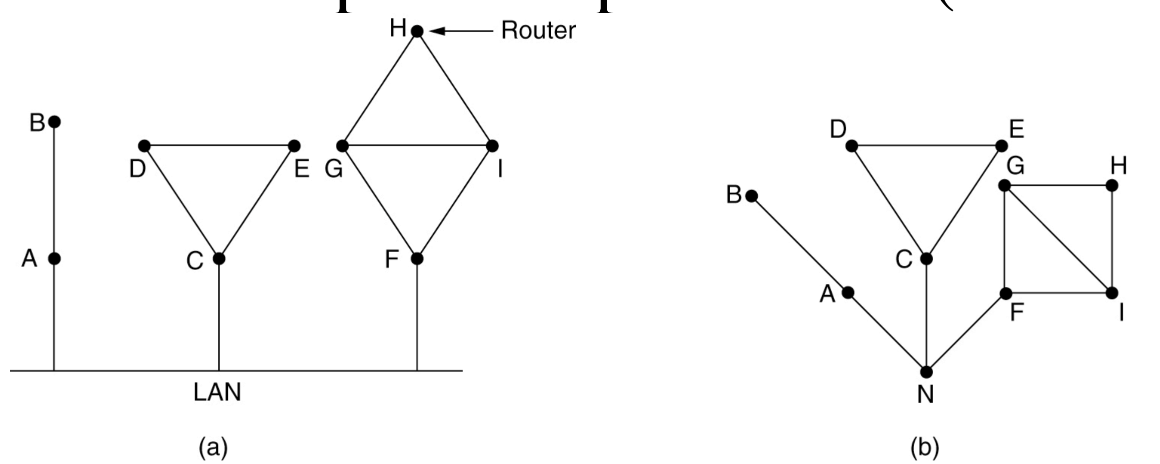

- Another problem with Distance Vector routing is that it does not take line bandwidth into account. (This is okay in the original set-up where are all lines were 56kbps but is inefficient in modern settings.)

# Link State Routing

- Distance Vector Routing was used on ARPANET until 1979.
- Currently, **link state routing** is used on the internet.
- It consists of five parts which each router must do:
  1. Discover one’s 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 paths to very other router. (using Dijkstra’s algorithm).

# Learning Neighbors

- A router does this by sending a special Hello packet on each of its point to point lines.
- The router on the other end is expected to reply with who it is (this number must be unique).
- Connections to LANs are modeled by viewing the LAN itself as a point-to-point node (N below):

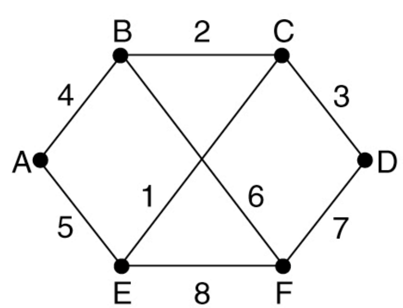


# Measuring Line Cost

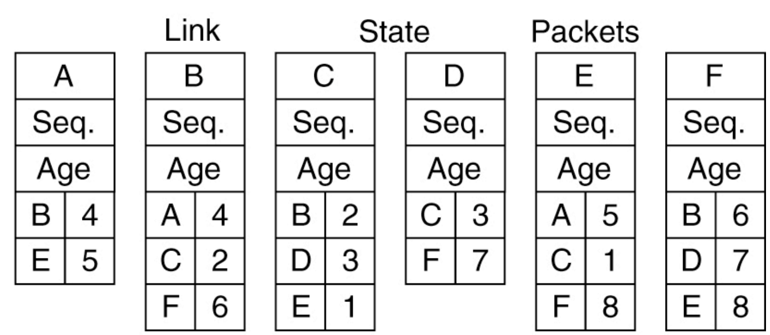
- To determine the delay to neighbors, a router sends a special ECHO packet to its neighbors.
- The neighbors are supposed to reply immediately. The round-trip time (RTT) can be estimated by the time until the reply.

# Telling ones Neighbors

- Once the information of the previous two slides has been collected the router needs to build a packet containing all this data.
- This packet, called a **link state packet,** consists of: the identity of the sender, followed by a sequence number and age, followed by a list of neighbors.
- The link state packets for each router in the graph (a) below are listed in (b).




(a)




(b)

# Telling ones Neighbors (II)

- Link State Packets could be calculated periodically or could be calculated if a significant change to the network occurs.
- Link State Packets are distributed basically by using flooding.
- The sequence number is used to keep the flood in check. It is incremented for each new packet sent.
- Routers keep track of (source router, sequence) pairs and duplicates are discarded. Non-duplicates are forwarded using flooding.
- The sequence number is 32-bits so is unlikely to wrap before being discarded.
- If a router goes down, it starts at sequence number 0 again. To avoid its packets being immediately discarded, a second field called age. This is decremented once per second by the routers who are remembering a particular router, sequence number pair, if it hits zero, the router discards the entry.
- When a link state packet arrives it is put into a holding area before being queued. If another packet arrives from the same source, they are compared and the older or duplicate is discarded.
- All link state packets are acknowledged

# Computing New Routes

- After a router has been accumulating a set of link state packets for a while, it can construct the entire subnet as each link is represented in some link state packet.
- Dijkstra's algorithm can then be used to construct the shortest paths to all possible destinations, this can be installed in the routing tables, and normal operations can be resumed.
- A specific variant of the Link State Algorithm, called OSPF is used in the internet. We will discuss it later.
- Another  link state protocol which was very influential was IS-IS (Intermediate System - Intermediate System) which was developed for DECnet and later used by ISO in its connectionless network layer protocol, CLNP.

# Other Kinds of Routing

- **Hierarchal Routing** -- to keep routing tables from going too large, routers are divided into **regions**, each router knowing how to route within its regions but not knowing details of other regions. Each has some top router which can be used to route within different regions. Such networks can have more than two layers, the optimal being O(ln n) layers.
- **Broadcast Routing** -- here we want to send the same packet to several targets. It is wasteful to resend the packet. You could use flooding but this tends to be wasteful of bandwidth. Instead, typically use **multidestinational routing.** Each packet contains a list or bitmap of destinations. When it arrives at a router. The router determines the set of output lines corresponding to these destinations. The router generates a new packet appropriate for each output line corresponding to the subset of destinations on that line. Other techniques are to calculate a spanning tree and send along it, or to use reverse path forwarding.
- **Multicast Routing** -- here we want to send to a group rather than everyone as in broadcast routing. Typically, this involves each router computing a spanning tree and then pruning it to the group in question.
- **Mobile, Ad-hoc Routing, Peer-to-peer** -- these are routing algorithms for mobile devices, for situations where the routers are mobile,etc . We won’t discuss these besides mentioning they exist.

# Intro to Congestion Control Algorithms

- If there are too many packets in the subnet, the performance degrades.
- This is called **congestion.**
- It can be caused by:
  - Several packets all needing the same output line on a router
  - Slow Processors
- Congestion is not the same as flow control in that it involves global properties of the network rather than transmission issues on point to point connections.

# More on Congestion Control

- Congestion control algorithms are split into two groups:
  - Open loop -- try to make sure the problem doesn't occur in the first place
  - Closed loop -- uses a feedback loop to prevent congestion:
    - Monitor the system to detect when and where congestion occurs
    - Pass this info to where action can be taken
    - Adjust the system

# Quality of Service