CS3200: Computer Networks

Lecture 18

IIT Palakkad

17 Sep, 2019

Flooding

- When a routing algorithm is implemented, each router must make decisions based on local knowledge, not the complete picture of the network.
- A simple local technique is **flooding**, in which every incoming packet is sent out on every outgoing line except the one it arrived on.
- Hop counter contained in the header of each packet that is decremented at each hop, with the packet being discarded when the counter reaches zero.
- Not practical for sending most packets, but flooding is tremendously robust.

Distance Vector Routing

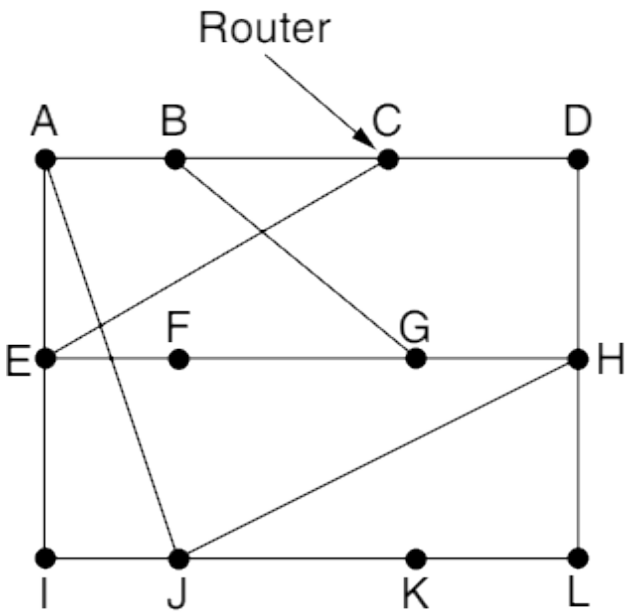
- A **distance vector routing** algorithm operates by having each router maintain a table (i.e., a vector) giving the best known distance to each destination and which link to use to get there. These tables are updated by exchanging information with the neighbors. Eventually, every router knows the best link to reach each destination.

- The distance vector routing algorithm is sometimes called by other names, most commonly the distributed **Bellman-Ford** routing algorithm, after the researchers who developed it (Bellman, 1957; and Ford and Fulkerson, 1962).

Distance Vector Routing

- As an example, assume that delay is used as a metric and that the router knows the delay to each of its neighbors.
- Once every T msec, each router sends to each neighbor a list of its estimated delays to each destination. It also receives a similar list from each neighbor.
- Imagine that one of these tables has just come in from neighbor X, with X_i being X's estimate of how long it takes to get to router i.
- If the router knows that the delay to X is m *msec*, it also knows that it can reach router i via X in $X_i + m$ *msec*. By performing this calculation for each neighbor, a router can find out which estimate seems the best and use that estimate and the corresponding link in its new routing table.

Distance Vector Routing



New estimated delay from J

To	A	I	H	K	Line
A	0	24	20	21	8 A
B	12	36	31	28	20 A
C	25	18	19	36	28 I
D	40	27	8	24	20 H
E	14	7	30	22	17 I
F	23	20	19	40	30 I
G	18	31	6	31	18 H
H	17	20	0	19	12 H
I	21	0	14	22	10 I
J	9	11	7	10	0 –
K	24	22	22	0	6 K
L	29	33	9	9	15 K

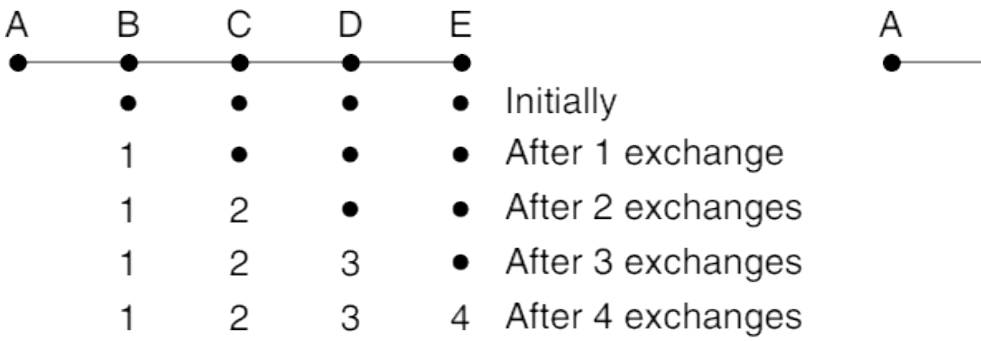
JA delay is 8 | JI delay is 10 | JH delay is 12 | JK delay is 6

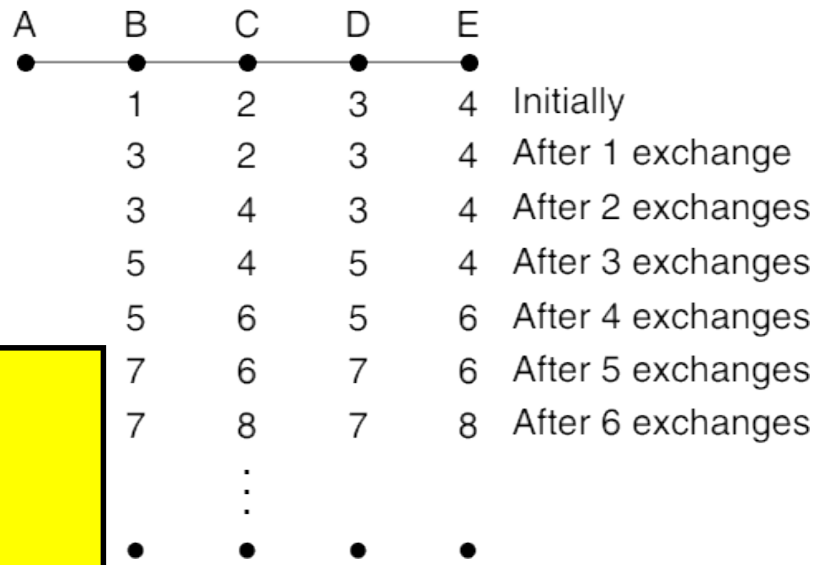
Vectors received from J's four neighbors

New routing table for J

The Count-to-Infinity Problem

The settling of routes to best paths across the network is called **convergence**. Distance vector routing is useful as a simple technique by which routers can collectively compute shortest paths, but it has a serious drawback in practice: although it converges to the correct answer, it may do so slowly. In particular, it reacts rapidly to good news, but leisurely to bad news.





bad news = links going down here link b/w A and B is down

Link State Routing

- Distance vector routing was used in the ARPANET until 1979, when it was replaced by link state routing.
- The primary problem that caused its demise was that the algorithm often took too long to converge after the network topology.
- Consequently, it was replaced by an entirely new algorithm, now called link state routing. Variants of link state routing called **IS-IS** and **OSPF** are the routing algorithms that are most widely used inside large networks and the Internet today.

Link State Routing

- Discover its neighbors and learn their network addresses.
- Set the distance or cost metric to each of its neighbors.
- Construct a packet telling all it has just learned.
- Send this packet to and receive packets from all other routers.
- Compute the shortest path to every other router.

Link State Routing

When a router is booted, its first task is to learn who its neighbors are. It accomplishes this goal by sending a special **HELLO** packet on each point-to-point line. The router on the other end is expected to send back a reply giving its name. These names must be globally unique.

The link state routing algorithm requires each link to have a distance or cost metric for finding shortest paths. The cost to reach neighbors can be set automatically, or configured by the network operator. A common choice is to make the cost inversely proportional to the bandwidth of the link.