Lecture 17
Plan: flood routing, distant vector routing, and link state routing

Review:

- Network layer: routing packets in a store and forward manner
- Virtual circuit and datagram
- Routing algorithm:
    - optimization criteria: hops, delay, bandwidth
    - Static Global shortest path (Dijkstra's algorithm) to find the shortest path between any two nodes.
    - Con: static, have to the global structure and parameters.

Topics: Other routing algorithms

Another routing algorithm; flooding
Idea: router forwards every packet to all outgoing links except the one that the packet comes from. In the end, the packet will reach the destination router.
Problem: there might be infinite routing if the packets are unchecked (loops)

Each source when generating a packet, it inserts a sequence number in it. When it generates a new one, the sequence number is increased by 1.

Each router keeps a list of source as well the highest sequence number it has seen from that source. If a packet arrives, then that number is compared with the list. If it is smaller what it has seen, then it is discarded. Or else the packet is forwarded.

Pro: Very robust → leads to military use; always find the minimum hop route
Con about flooding: → very slow

The routing algorithm used in WAN and Internet are

- distance vector routing (local, adaptive)
- link state routing (global, adaptive)
- path vector routing (global, adaptive)

Distance Vector Routing (or RIP: routing information protocol)
Suppose we have n routers and each router has a set of other routers to which it has a direct link.
Idea: Each router keeps a table of n entries (one for each other router). **Neighboring** routers exchange their distance tables periodically and based on the information they received, they update their table.

Details:
each entry contains

- Estimated distance to that destination (distance may be hops, queue length, or delay)

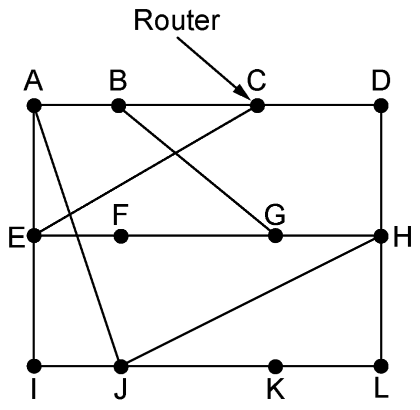
Q:/ How to get delays?

A:/ send a special echo message and the destination will respond as soon as it can with a timestamp.

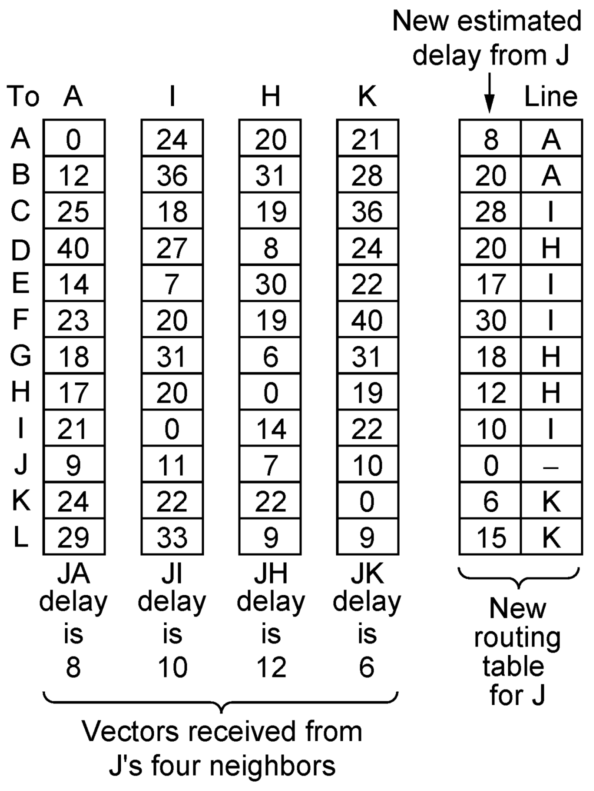
- Which adjacent node should be used to forward packets to that destination
- Before each update, each entry is assigned with the cost of direct link to the destination. If there is no direct link, then it is set to be infinitiy

Periodically, the table is exchanged among routers

Q:/ How to update my table



(a)



To	A	I	H	K	New estimated delay from J	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	New routing table for J	

Vectors received from J's four neighbors

(b)

A:/ Suppose we have a subnet with 12 routers and links among them
For router J, it received four tables from A,I,H,K (its neighbors). We will take delays as our distance (the smaller, the better)

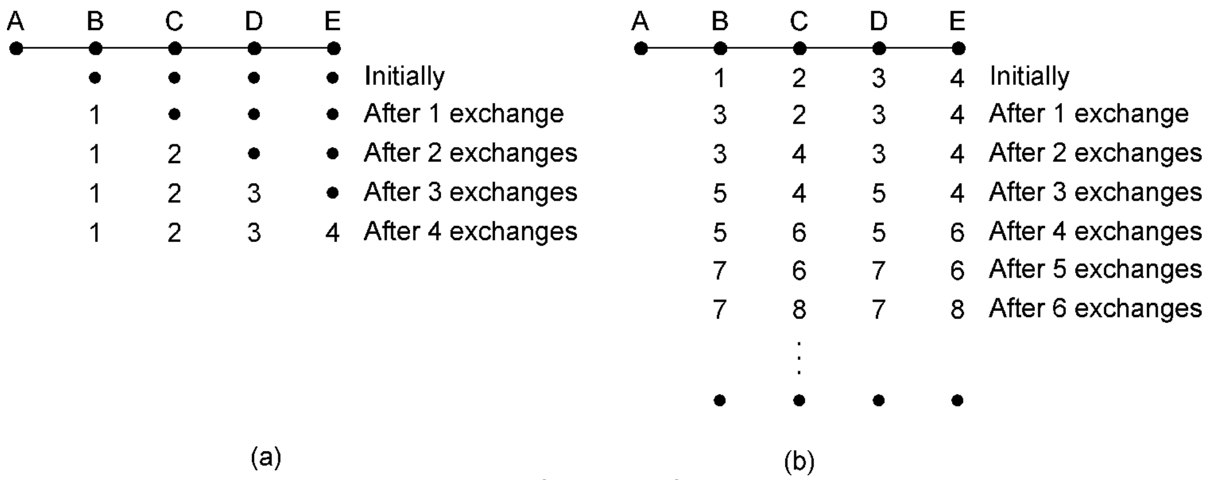
For router J, it knows the delay to its neighbors and it also knows from its neighbor’s table how long it takes to go to a specific router via its neighbor. Then it can update its table entry.

E.g. If J wants to know if it has to update the entry to go to G
For from the table from A, AG takes 18 ms, JG takes 8ms. So JG via A takes 26 ms.
Similarly, JG via I is 41ms, JG via H is 18ms, and JG via K is 37ms.

So JG via H is the smallest, so the table in J is updated as 18, H
Note: The delays of J to its neighbors are measured before the update, not from the old routing table.

Convergence -- If something changes, how long does it take for everyone to know about it?

Good news scenario: If A is down initially and then it comes online, How long does it take for everyone to know about it?



	A	B	C	D	E	
(a)		•	•	•	•	Initially
		1	•	•	•	After 1 exchange
		1	2	•	•	After 2 exchanges
		1	2	3	•	After 3 exchanges
		1	2	3	4	After 4 exchanges
(b)		1	2	3	4	Initially
		3	2	3	4	After 1 exchange
		3	4	3	4	After 2 exchanges
		5	4	5	4	After 3 exchanges
		5	6	5	6	After 4 exchanges
		7	6	7	6	After 5 exchanges
		7	8	7	8	After 6 exchanges
		•	•	•	•	

A:/ initially, B,C,D,E all have their entry for A as infinity
Suppose all routers exchange their tables at the same time and use hops as our distance metric
Then B first got the table from A with delay as 0. So B updates its entry to A as (1,A)
Next time, C learns that B has a path to A with 1 hop, so C updates its entry to A as (2,B)
Next time, D learns from C and updates its entry as (3,C). Similarly for E as (4,D)

So the worst case scenario in the good news case is it take the max length of the subnet to update all routers

Bad news Scenario: Suppose A is initially online then it suddenly went off line
Initially, B (1,A), C(2,B), D(3,C), E(4,D)
step 1: B didn't hear anything from A, but so it sets its delay to A as infinity. But from the table from C, it found out that C has a link to A with 2 hops. So B updates its table entry for A as (3, C).
step 2: C got two entries from B and D respectively claiming that they have routes to A with 3 hops. C picks one of them and update its entry to A as (4, B)
…

Counter to infinity problem
Q:/ How to solve it?
A:/ setup a maximum number of hops. It is reached, then we believe that the router is offline

Problem with Distance Vector Routing: Slow convergence

To improve the convergence speed, Link state routing was proposed.
Idea: we want to build up a global picture and use the Dijstrak's algorithm to find the minimum path between any two nodes
We will use delay in here for illustration.

Q:/ What do we need?
A:/ basically the delay between any two neighbors.

Steps:

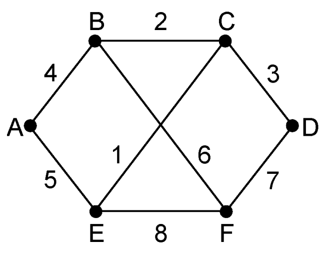
- Use ECHO packets to determine delay to neighbors (each router must have a global name)
- Collect that information in a "link state" packet (LSP); send it to all other routers (by flooding)
- Construct distance graph for the entire set of routers; that is, a global graph that all the routers agree on.
- Apply Dijkstra's shortest path algorithm.

Q:/ How to construct the LSP?
1. sequence number is used to avoid forwarding the same packet again

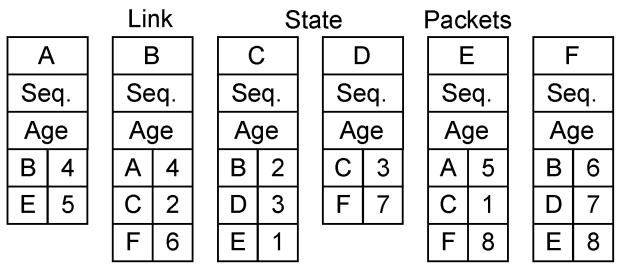
A:/ give each packet from each source a sequence number. → only route the new ones
Basically, each router keeps a counter k for each router in the subnet. If a packet from a router is received, the sequence number of that packet is compared with the counter for that router. If it is higher, then forward. if it is smaller or equal, discard it.

2. to live is used to avoid infinite forwarding
A:/ set a maximum number of hops. After each routing, the hop number of the packet is incremented by 1. If the max is reached, discard the packet.
Q:/ How to set the maximum number of hops?
A:/ either an estimate of the hop or the max length of the subnet

Q:/ How often should LSP be exchanged?
A:/ periodically or when something happens to the subnet
e.g.



(a)



Link State Packets

A	B	C	D	E	F
Seq.	Seq.	Seq.	Seq.	Seq.	Seq.
Age	Age	Age	Age	Age	Age
B 4	A 4	B 2	C 3	A 5	B 6
E 5	C 2	D 3	F 7	C 1	D 7
	F 6	E 1		F 8	E 8

(b)

Pro: adaptive and shortest path is guaranteed.
con: overhead to build the topology. All routers have to use the same metric → path vector routing takes care of the metric problem.