

CS423: Lecture 22, Route Computation

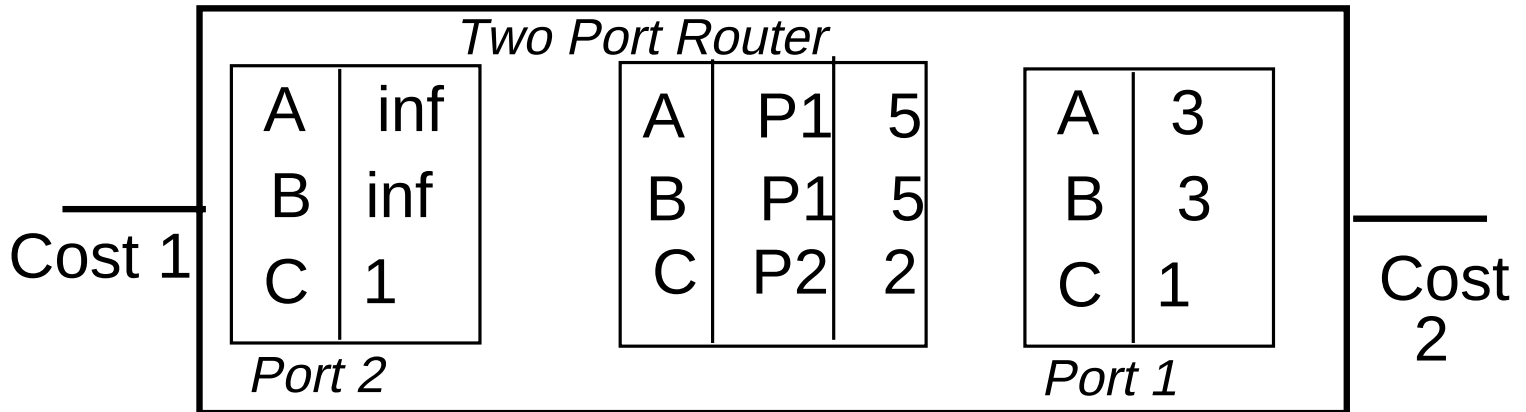
George Varghese

November 11, 1996

Distance Vector

- Two principal methods for Route Computation: distance vector (IP) and link state (OSI). We will study both.
- Distance vector: how can we use spanning tree protocol idea. We found distances to min ID node by “gossip”. Use same idea for updating distance to all nodes.
- Previously we kept (Root, distance, parent). Now we keep a vector of (ID, distance). Hence called distance vector.

Distance Vector Databases



- As in Spanning Tree, we have port and central databases.
- Central is computed based on best port database.

Link Failures and Distance Vector

- On link failure, delete stored distance vector for that port. Link failure is reported by neighbor discovery (because we haven't received a hello from that neighbor for a while).
- Different from spanning tree in that there is no aging of information except the local aging used to detect link failures. Instead must rely on something call count-to-infinity.
- Also unlike spanning tree, you send whenever information changes. Periodic sending is a good idea for robustness but not strictly necessary.

Data Packet Looping

- After A crashes, B and C keep thinking the best way to get to A is through each other.
- Thus a data packet destined to A will keep looping until either the hop-count in the packet reaches its maximum value or B and C finally decide that A is down.

Link State History

- The ARPANET is a large national network that is part of the global Internet. (ARPANET has a net number). Classic network in a historic sense.
- Originally, ARPANET used distance vector. However, failure recovery times were very slow after node failures because of count-to-infinity problem. Also data packets kept looping during this period.
- New ARPANET moved to link state routing which has quicker response to failures and no count-up problem. Similar design used by OSI Routing. packets

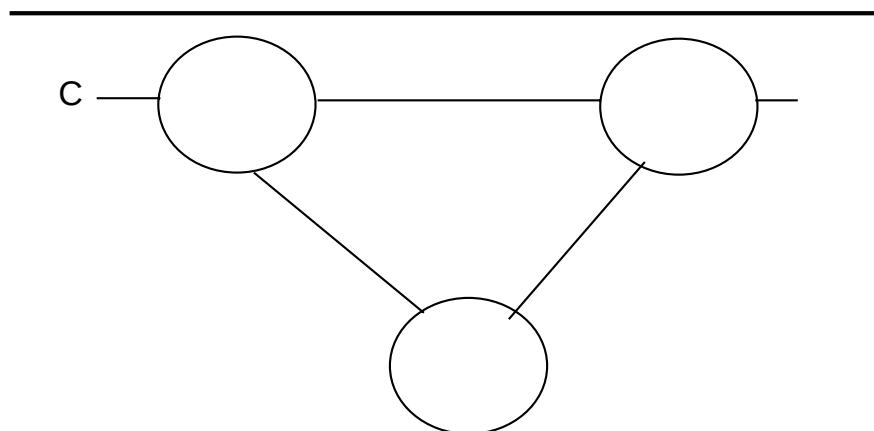
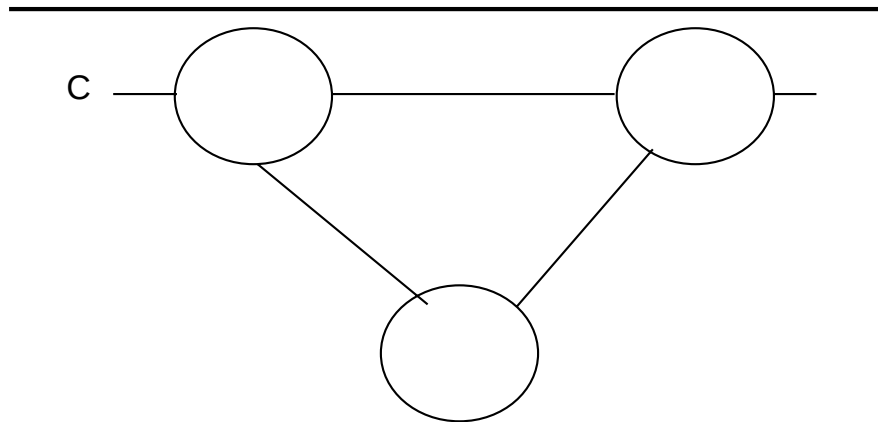
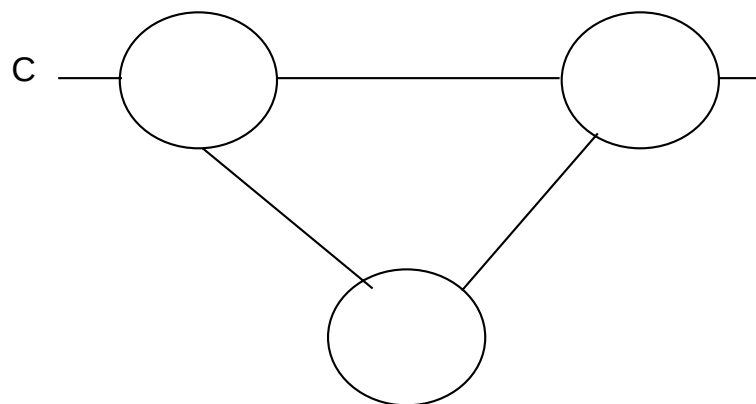
Link State: the basic idea

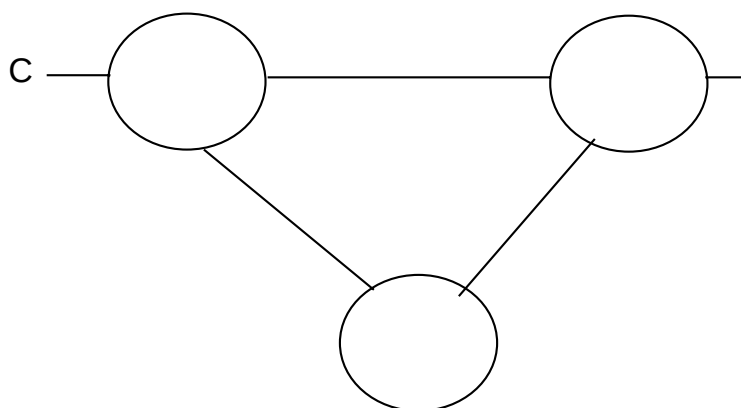
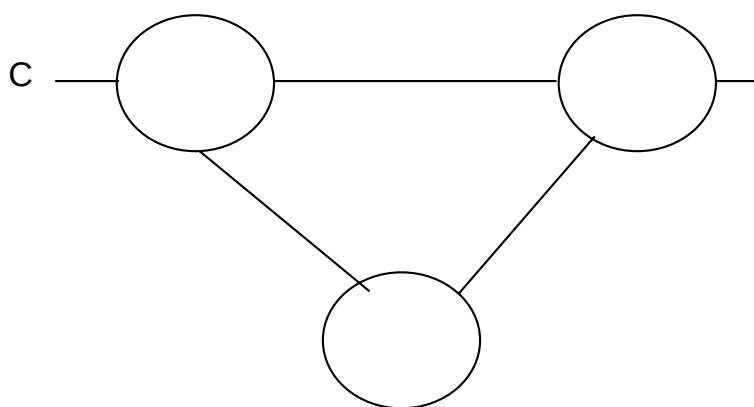
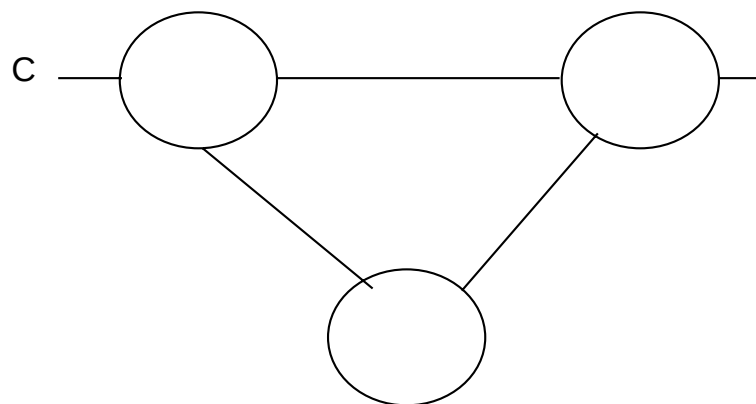
- Each node knows the default (or manager settable) cost of its outgoing links. Neighbor discovery is used to compile a list of neighbors that are UP. This information, along with link costs, is placed in a Link State Packet (LSP).
- Each source broadcasts its LSP to *all* other nodes using a primitive flooding mechanism called intelligent flooding.
- After the LSP propagation process stabilizes. each node has a complete and identical picture of the network graph. Then each node S uses any shortest path algorithm (i.e., Dijkstra's) to compute the next node on the shortest path from S to every other node D .

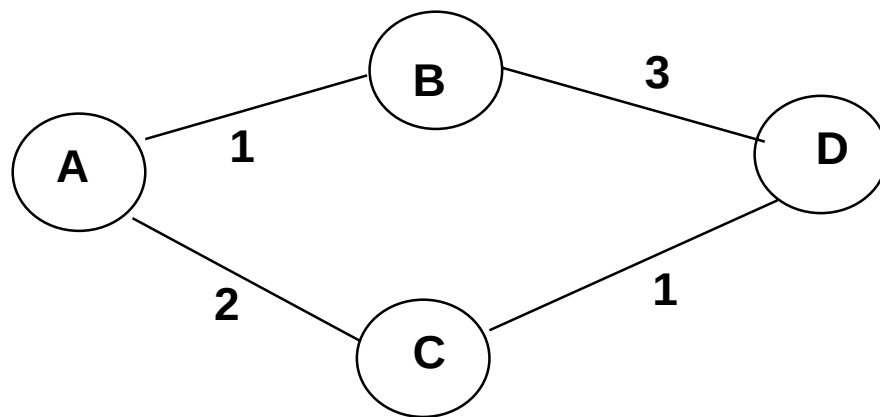
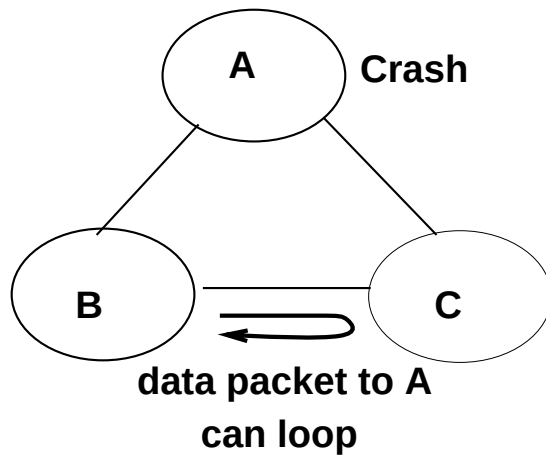
LSP Generation

LSP Generation on Failure

- If link AC fails, neighbor discovery in A and C will eventually detect failure.
- Only A and C recompute their LSP values and broadcast their LSPs again to all other nodes. Other nodes do not recompute or rebroadcast their LSPs.

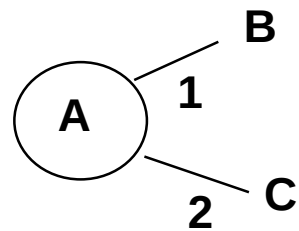


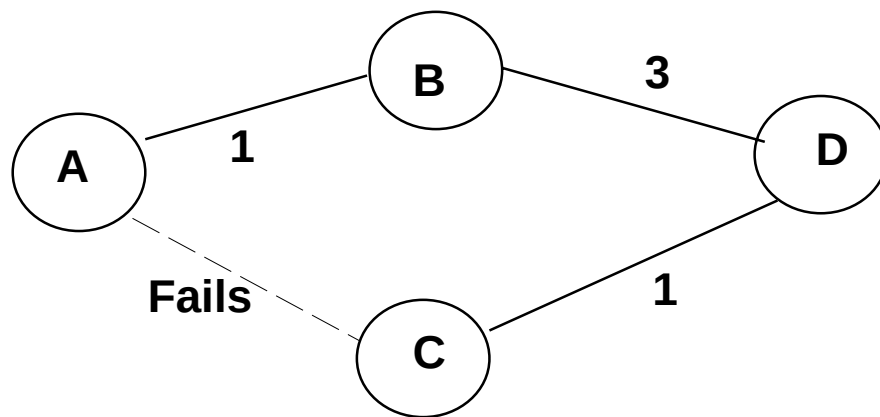




LSP
A
B
1
C
2

which means:





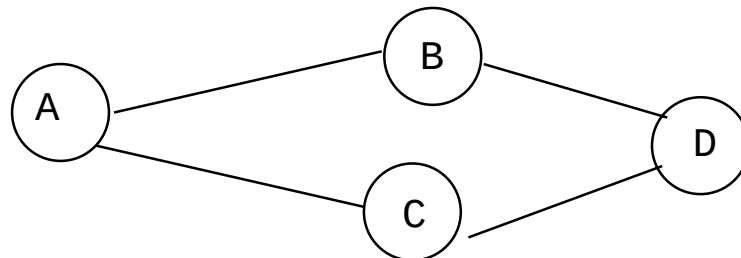
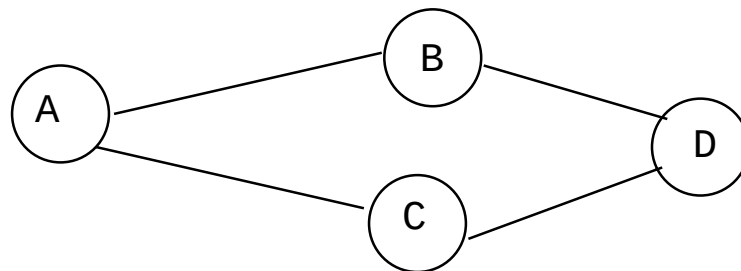
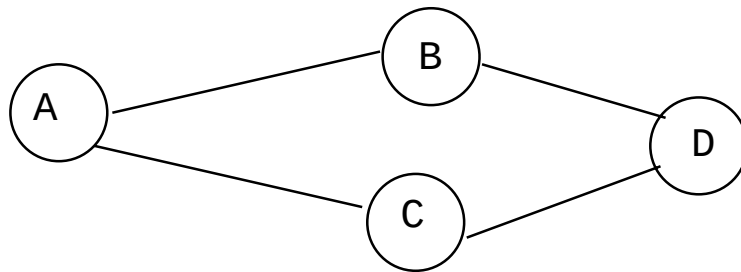
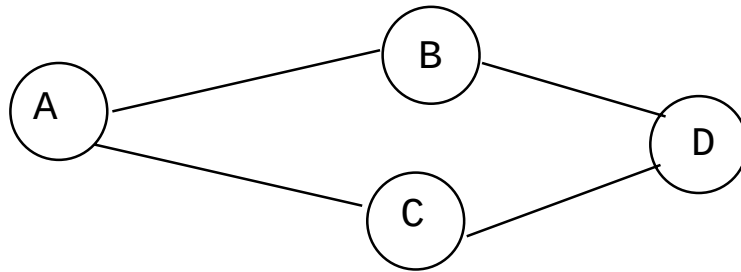
LSP
A
B
1

which means:

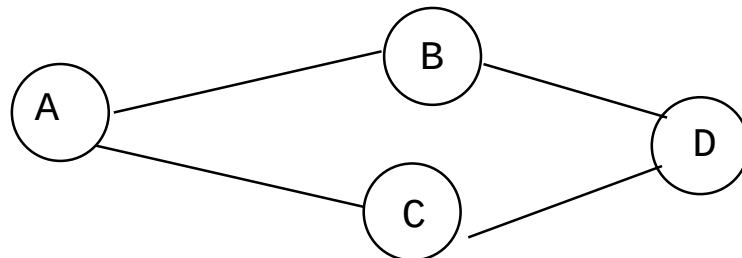
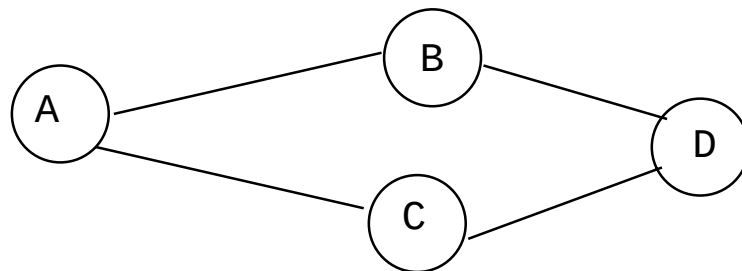
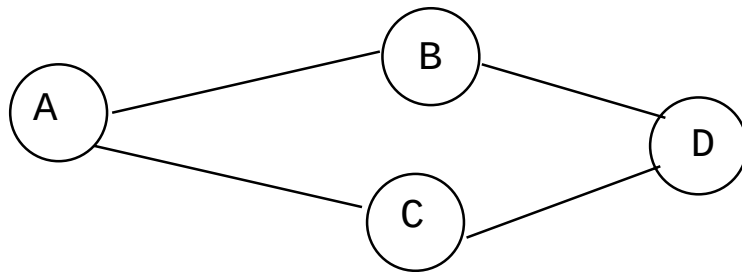
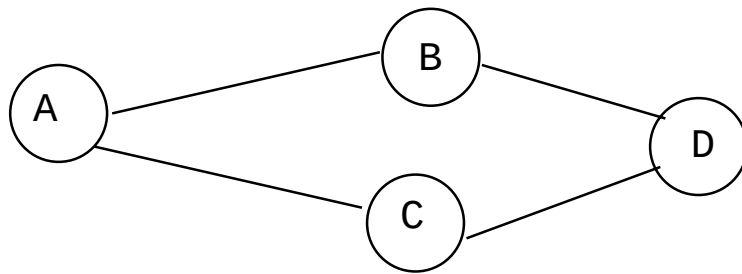
```

graph LR
    A((A)) ---|1| B
  
```

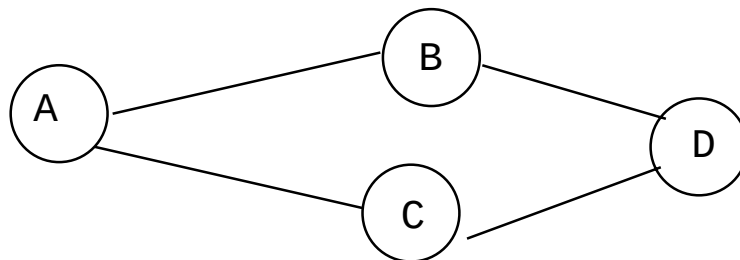
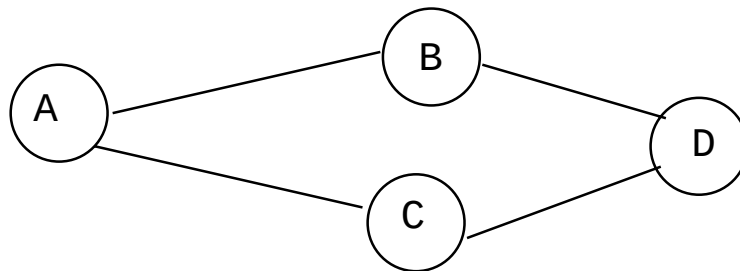
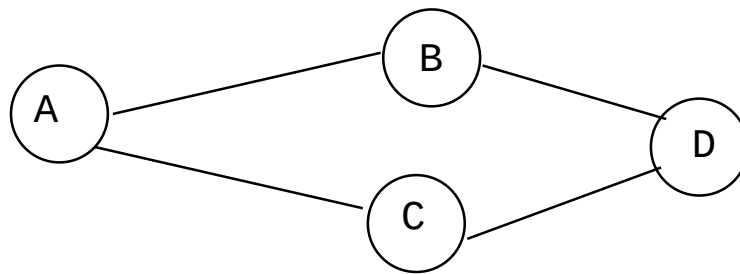
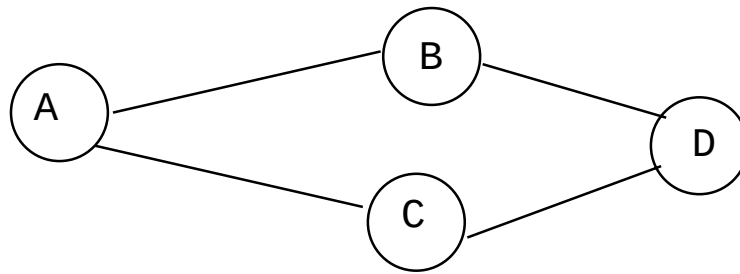
LSP PROPAGATION: FLOODING?



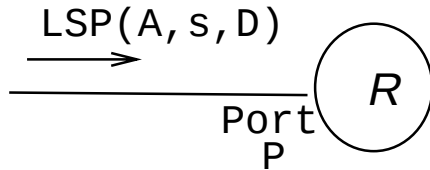
INTELLIGENT FLOODING



INTELLIGENT FLOODING: CRASHES



LINK STATE CODE



RECEIVE LSP(A, s, D) on PORT P

```
IF s < SEQ(A) THEN
    SEND STORED-LSP(A) ON PORT P
    ACK([A, P] = TRUE
ELSEIF s > SEQ(A) THEN
    IF A = ME THEN (*source must jump*)
        SEQ(ME) = s + 1;
        SEND STORED-LSP(A) ON ALL PORTS
        FOR ALL PORTS Q, ACK[ME, Q] = TRUE
    ELSE
        STORED-LSP(A) = LSP(A, s, D)
        SEND STORED-LSP(A) ON ALL PORTS P<>Q
        FOR ALL PORTS Q<>P, ACK[A, Q] = TRUE
ELSE
    SEND ACK(A, s) ON PORT P
    ACK([A, P] = FALSE
```

PERIODICALLY

```
FOR ALL PORTS P and A with ACK[A, P] = TRUE
    SEND STORED-LSP(A) ON PORT P
```

RECEIVE ACK (A, s)

```
IF s = SEQ(A) THEN
    ACK([A, P] = FALSE
```

LINK ON PORT P COMES UP

```
FOR ALL SOURCES A DO
    SET ACK[A, P] = TRUE (*send all LSPs on P*)
```