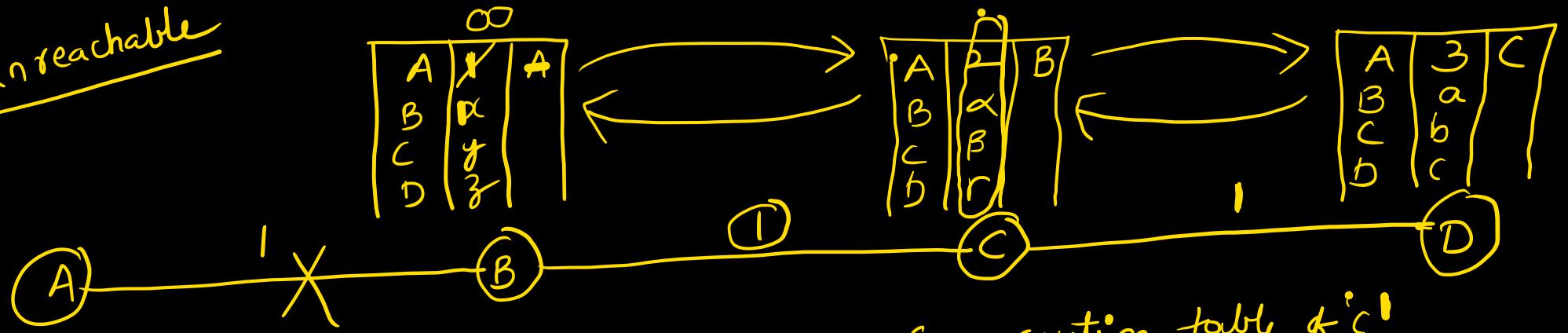
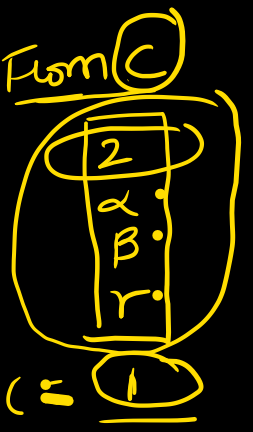
Bad news
→ A is unreachable

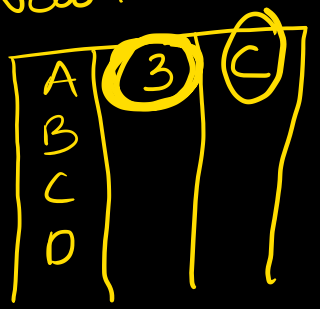


At B: From C



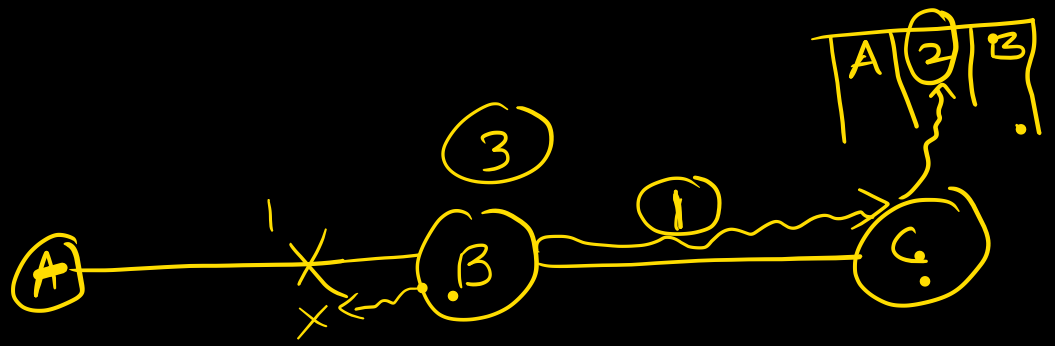
BC = 1

new RT at B



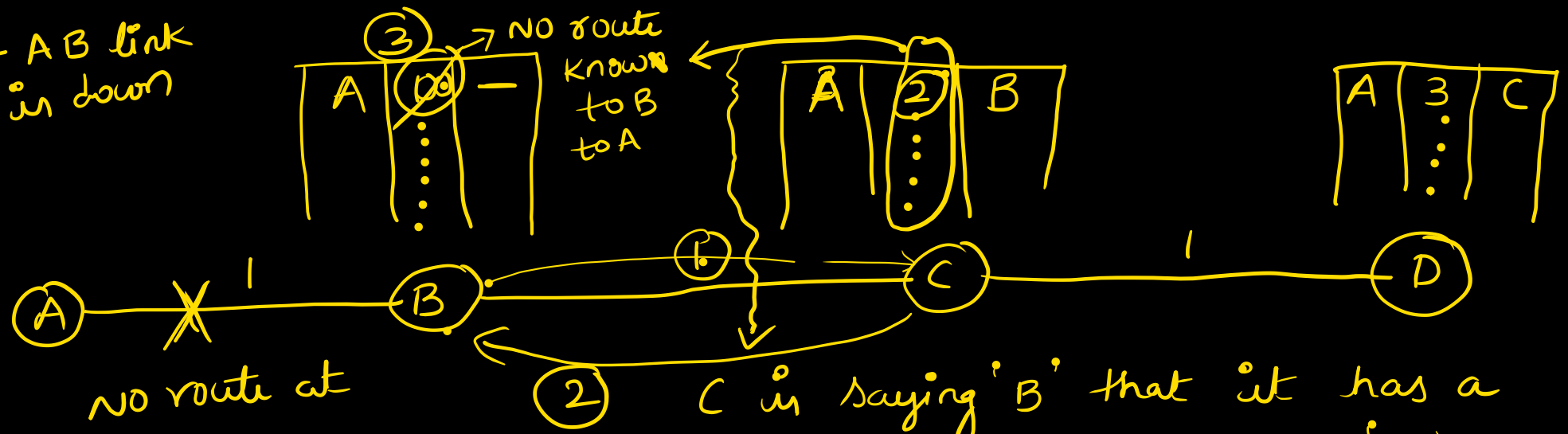
which is wrong.

See routing table of 'C'
For A, distace is 2 'it in through 'B'! At B, see dv of C. It is saying it can take B to A in 2 hops.



For C to go to A, it has to go through 'B'.
Now 'AB' is down, but C is saying B that
it can take to A in 2 hops

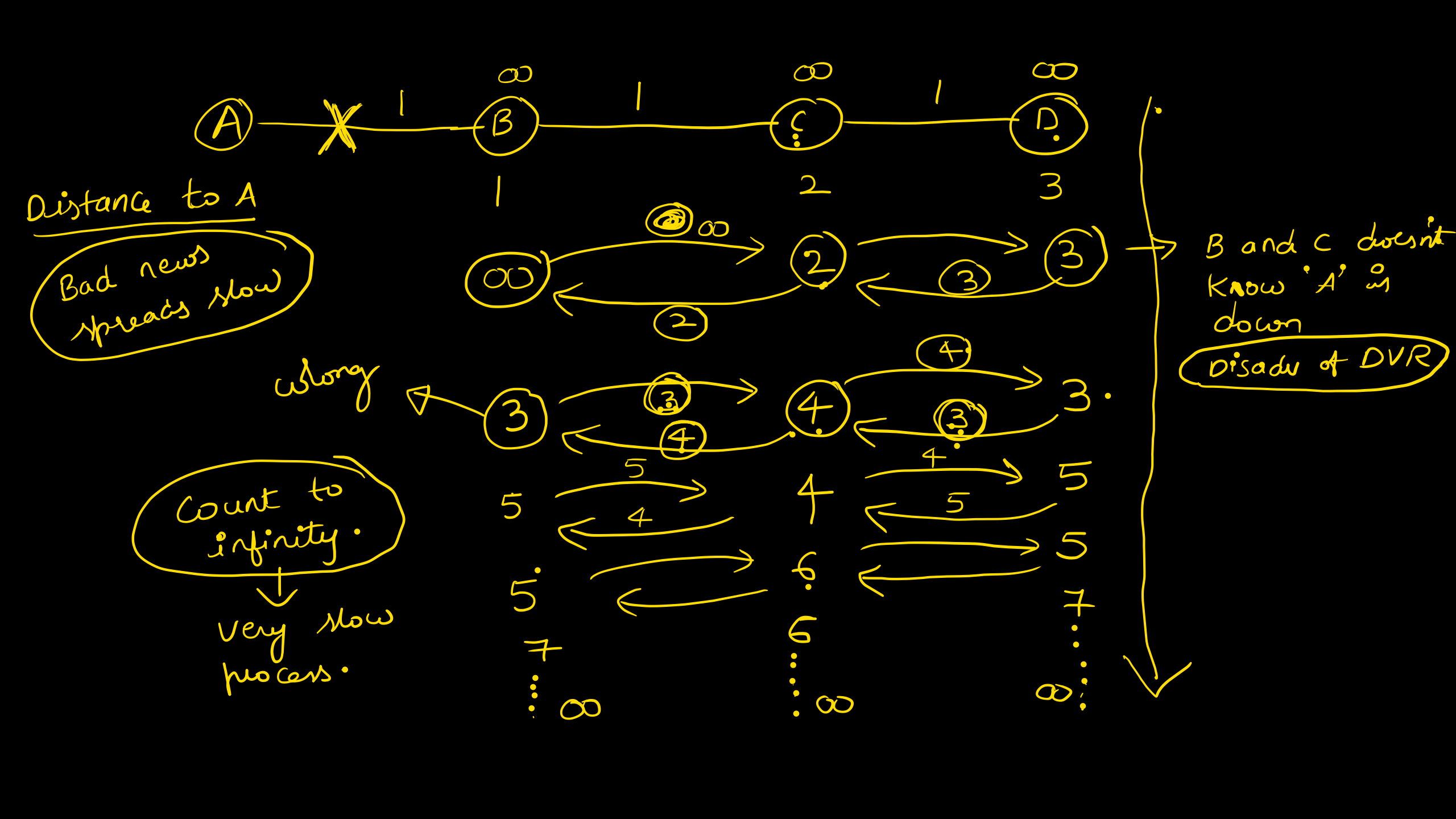
Bad news – AB link is down



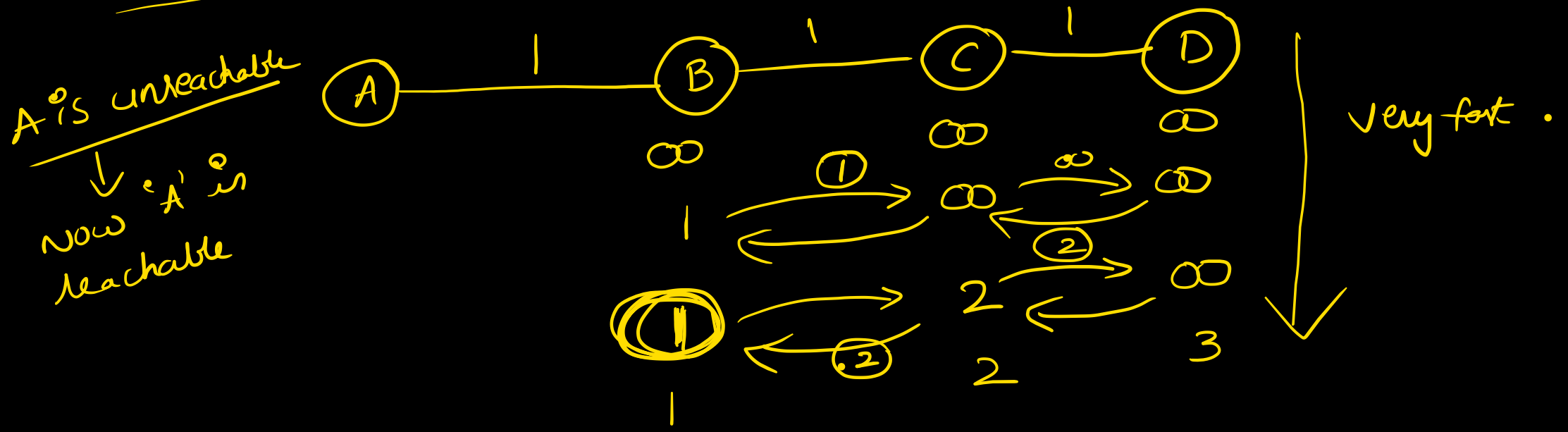
No route at all.

But B is getting ① + ② '3' as distace. which is wrong.

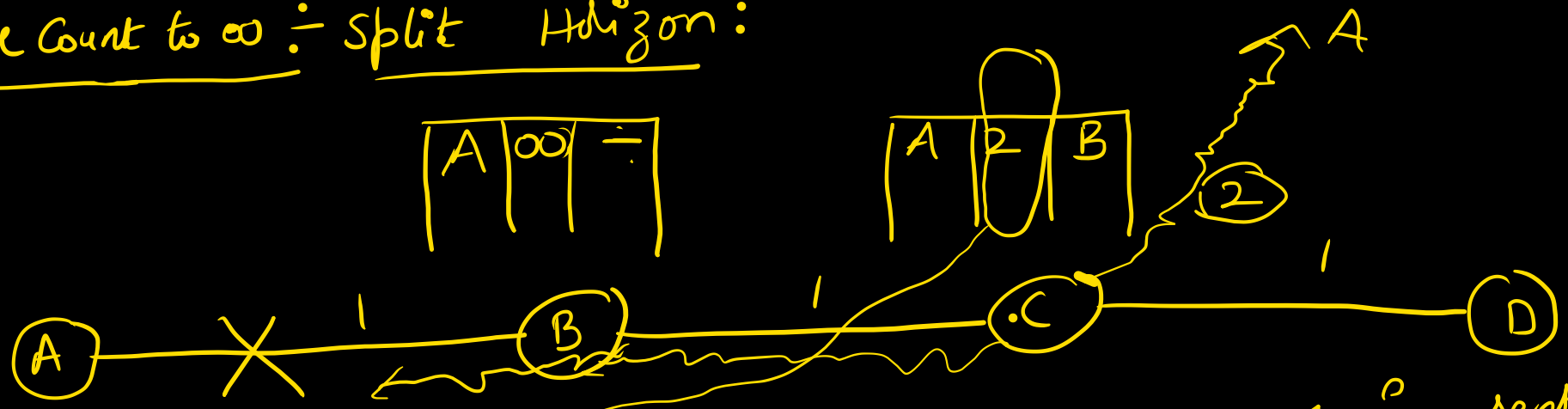
C is saying 'B' that it has a route to A with distance '2'


A
B
C
D
1
1
1
∞
∞
∞
1
2
3
Distance to A
Bad news spreads slow
B and C doesn't know 'A' is down
Disadv of DVR
along
Count to infinity.
very slow process.

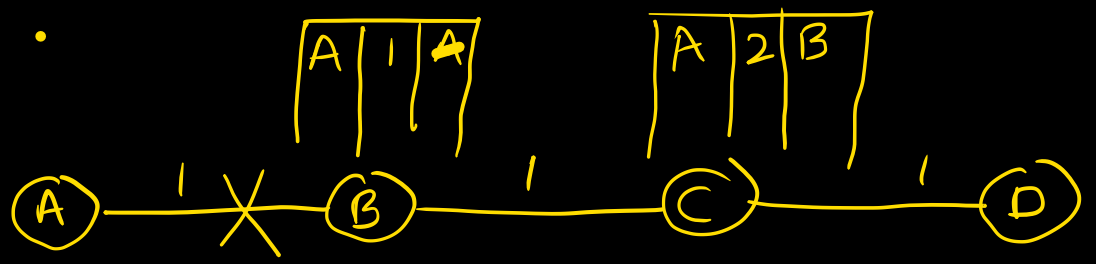
Good news spreads fast:

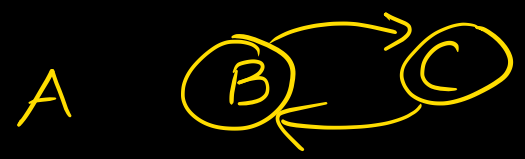


How to solve Count to ∞ :- Split Horizon :



Problem with DVR is only distance vector is sent, not next hop. when 'C' sends '2', it is saying 'B' that it has a path to 'A' but it is not saying that path is through 'B' only. So 'B' is thinking that 'C' has a different path.

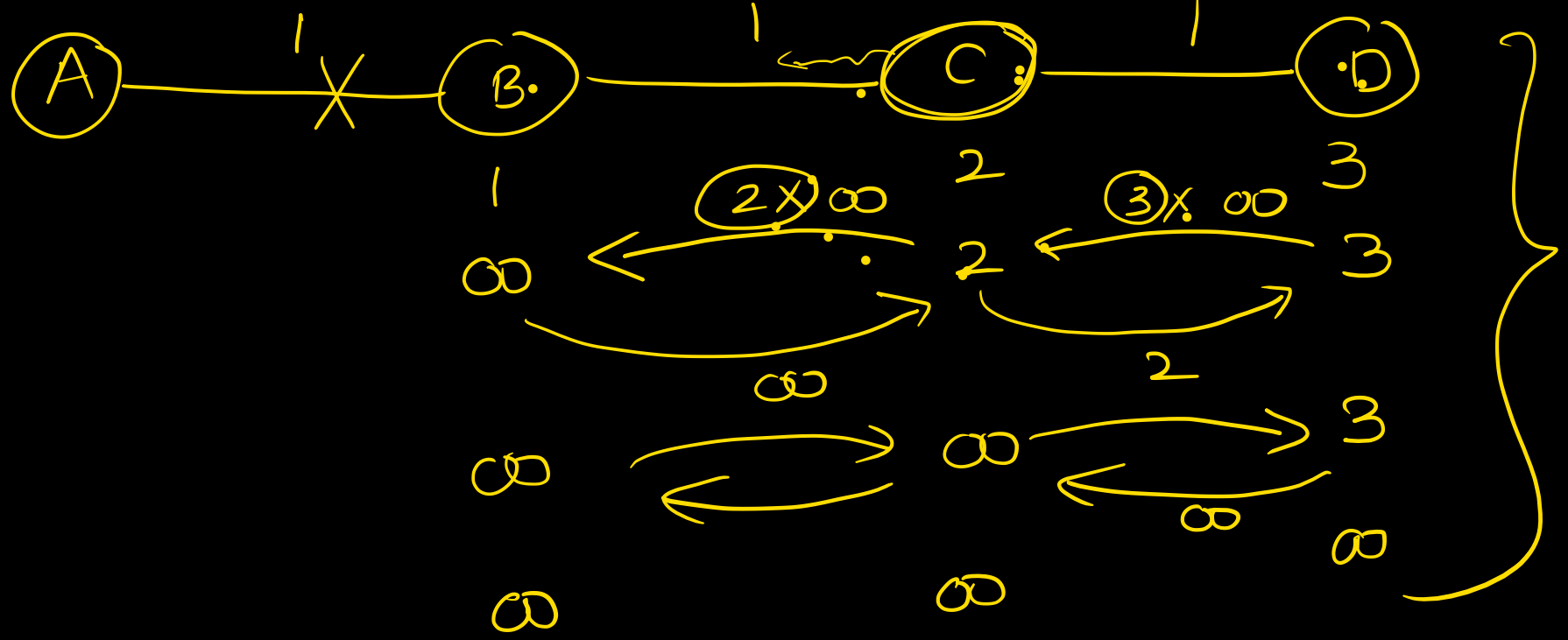




'B' is helping 'C' to reach 'A'.

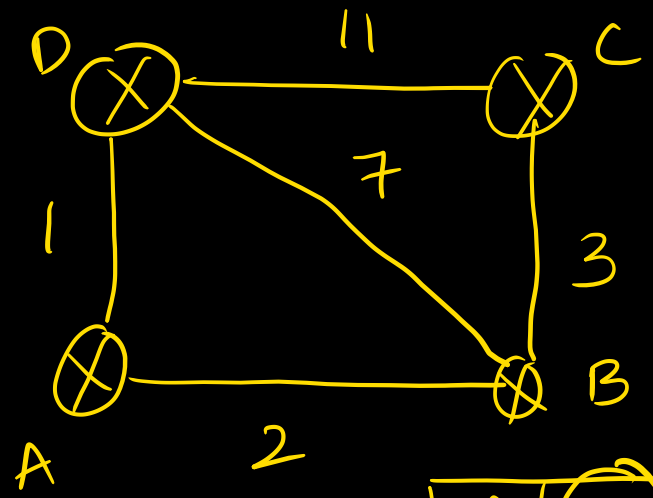
'C' is helping 'B' to reach A when AB is down.

So the rule is, dont help 'B' if 'B' is helping you


A
B
C
D
1
1
1
1
2
3
2X ∞
3X ∞
∞
2
3
∞
2
3
∞
∞
∞
∞
∞

If a node is dependent on other node (NH), then send infinity.

DVR with split horizon:



what should 'B' send to 'A'

TO A
∞
0
3
∞

RT of B

A	2	A
B	0	B
C	3	C
D	3	A

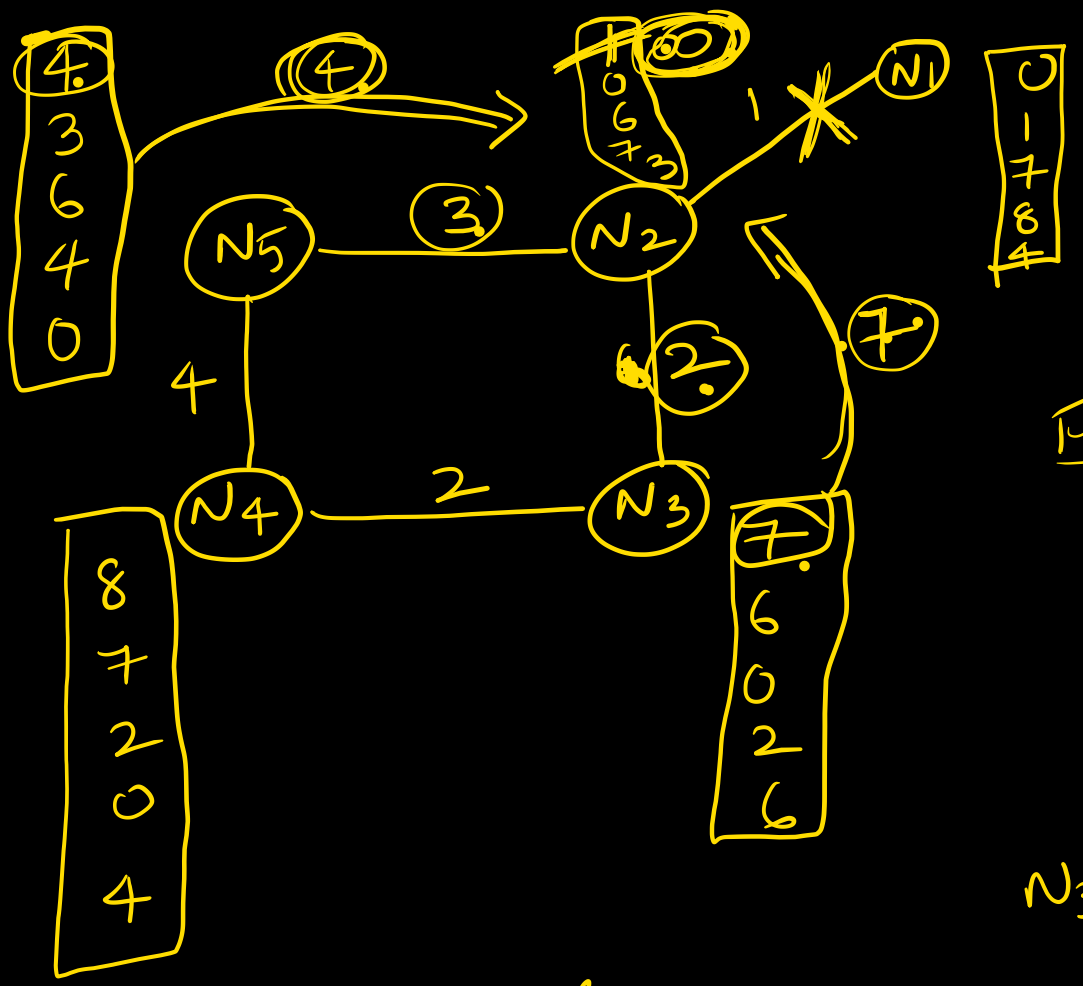
TO C:

2
0
∞
3

At B: RT:

A	1	A
B	0	B
C	3	C
D	4	D

To A:	To C:	To D:
∞	1	1
0	0	0
3	∞	3
4	4	∞

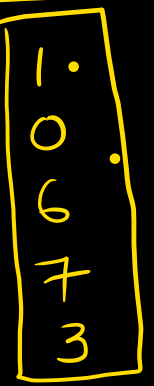


Gate question

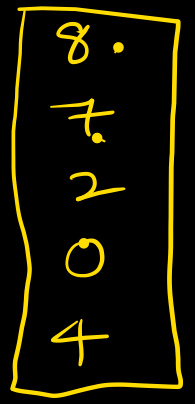
new RT at N_3, after getting distance vectors from N_2 and N_4

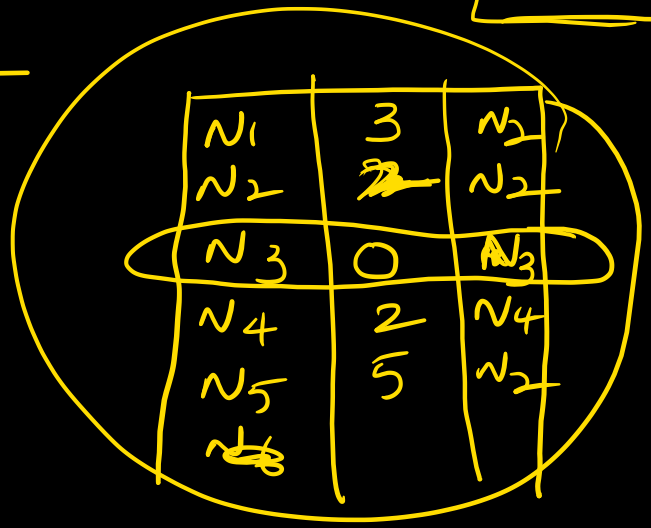
OVR without Split horizon

From N_2:



From N_4





N_3N_2: 2 N_3N_4: 2

What is new value at 'ω'? '7' → gate

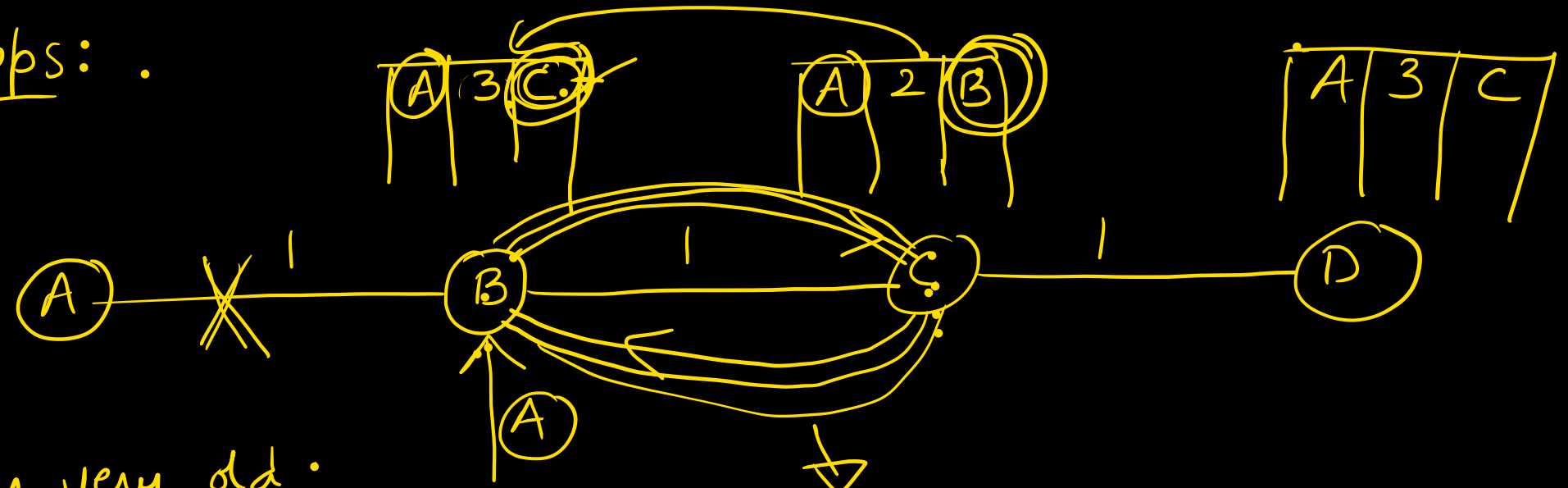
Disadv of DVR:

(i) count to infinity.

(ii) slow Convergence → Final Routing table will come slow.

(iii) loops are possible.

Loops:



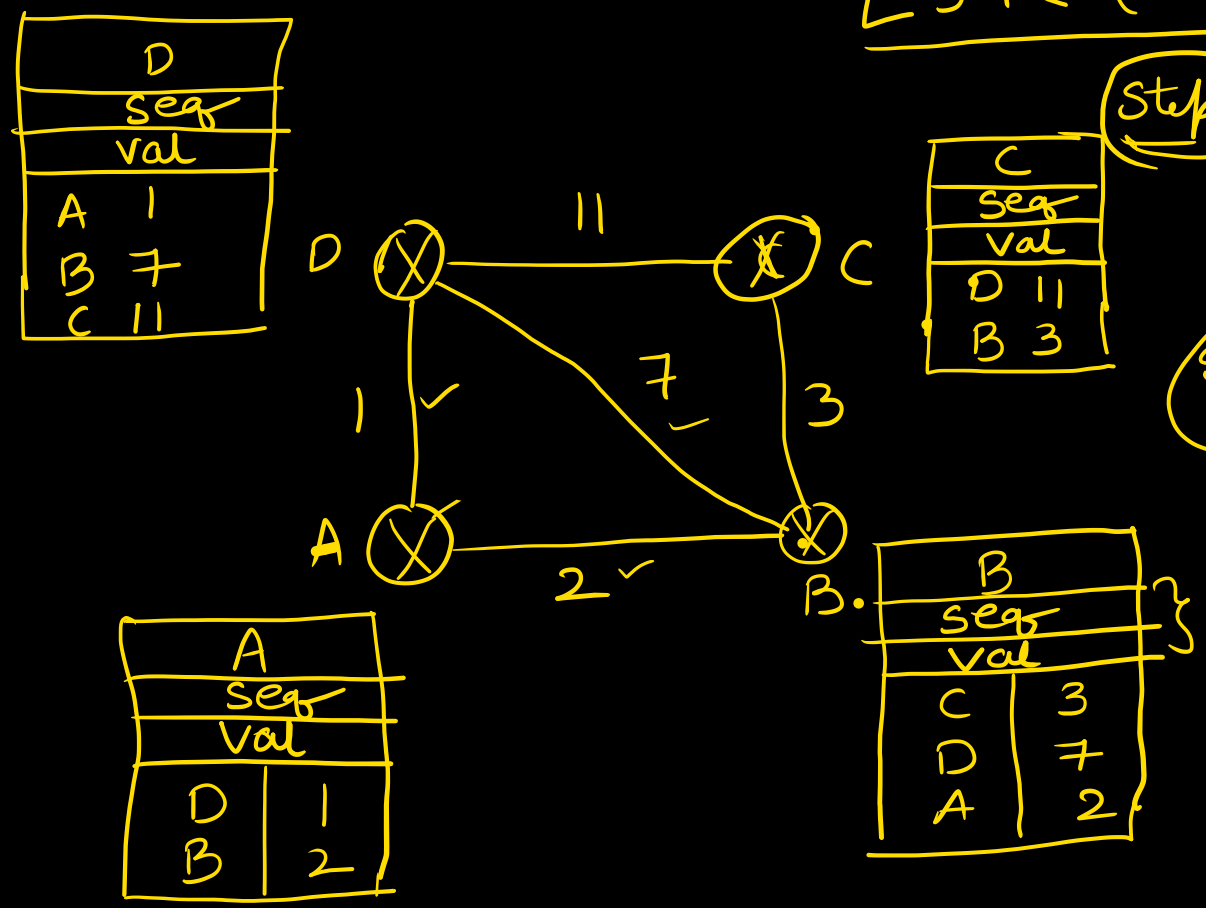
DVR is very old.

1980's ✓.

LSR → 1990's

Loops may be formed.

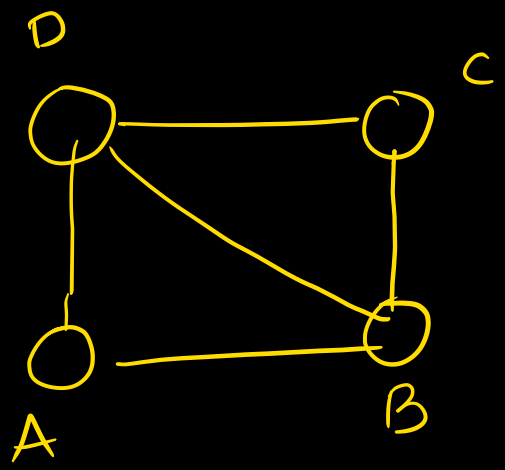
LSR (link state routing):



Step 1: Every Router will create a (link state packet) about links it is connected to.

Step 2: All (LSPs are flooded) to all the routers (not just neighbours) (This is the key diff b/w DVR and LSR) In DVR only neighbours get data. Here every router gets data.

∴ DVR is based on local knowledge.
LSR is based on global knowledge.



In DVR:

A → D, B
C → D, B
B → A, C, D
D → A, B, C.

only from neighbours.

In LSR

A → B, C, D
B → A, C, D
C → A, B, D
D → A, B, C.

→ from all.

Step 3: All routers will construct the graph in memory using the LSPs.

At A:

From B

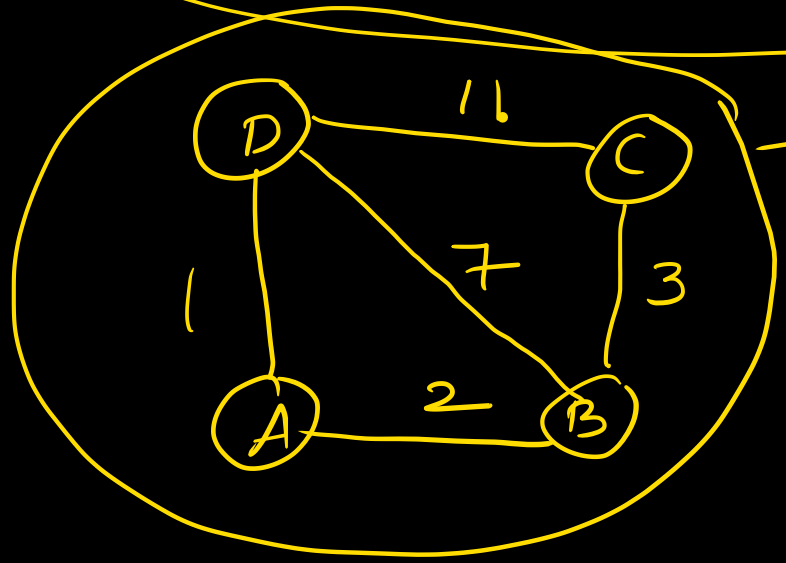
A	2
D	7
C	3

From D:

A	4
B	7
C	11

From C:

D	11
B	3



→ Entire graph is construction.

Now A will apply Dijkstra algo for shortest path to find Routing table at 'A'.

→ Step 4

DVR grows step by step.

1 edge → 2 edges → 3 edges

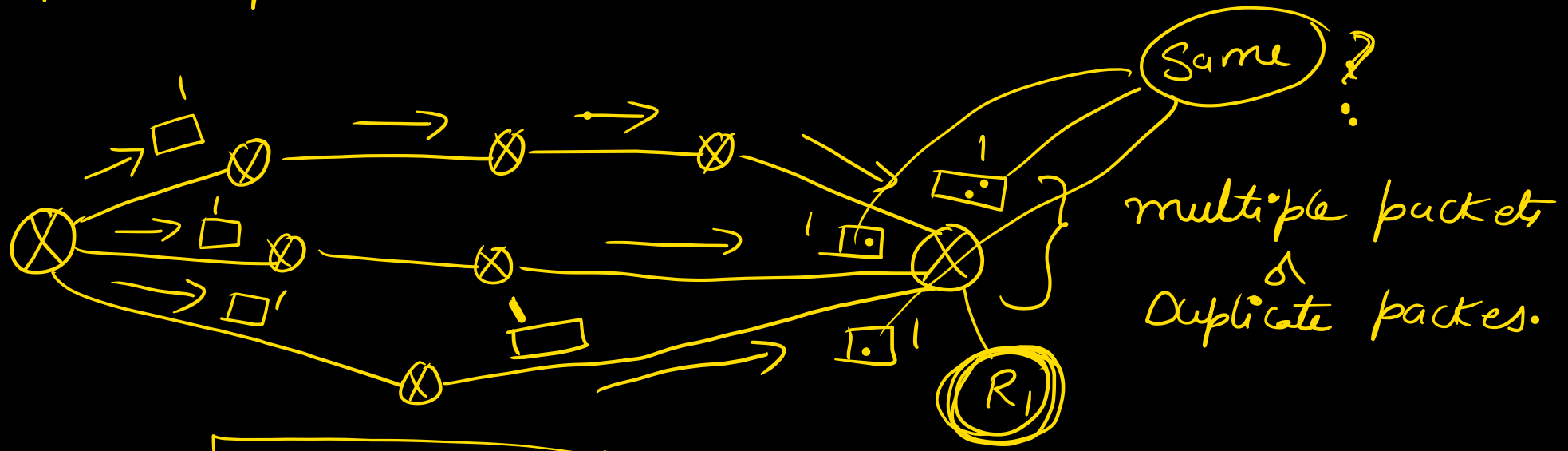
LSR is fast not step by step.

∴ LSR Converges faster compared to DVR.

But LSR → required more BW, because LSPs are bigger and flooding is used.

There is a problem with LSR. → flooding

→ Duplicate packets.



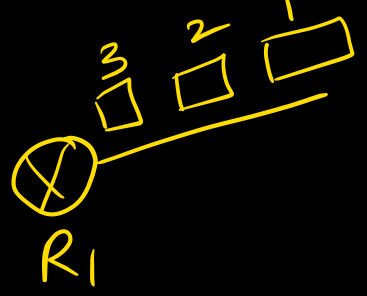
we use ← Seq numbers.

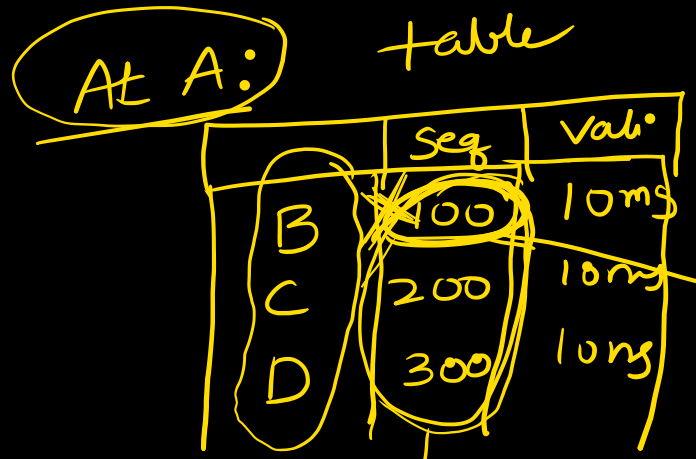
meaning is same.

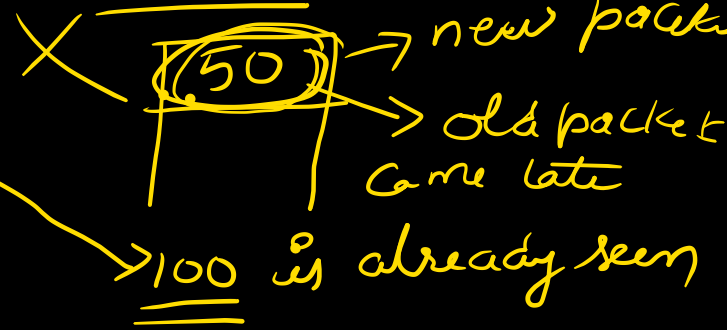
Identification

↓

IP datagrams.

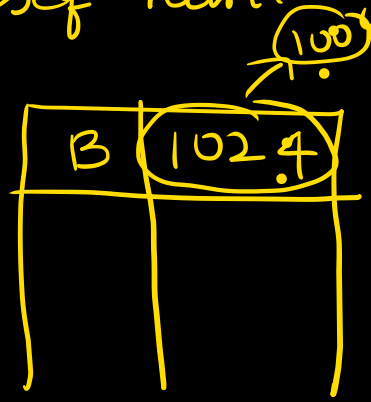






Problem:

Due to some error, if seq num gets corrupted; 100 → 1024



Latest seq number seen

101 ✗ 102 ✗ ... ... 1024 ✗

seq → wrap around.