Distance Vector - Part 2

Dr. Kiran Manjappa
Dept. of IT, NITK

Previous Session

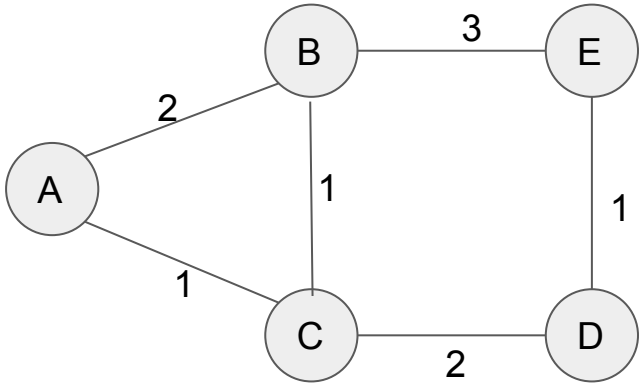
- Sharding
  - New clustering
  - Why sharding ?- To provide the dedicated service (w.r.t our problem statement)
  - Different ways of Sharding (Clustering) - Partitional Clustering.
- SINR
  - Unwanted Noise in the actual signal
  - Interference - caused by other signals
  - Noise - Caused by other reasons (signal processing, receiving …)

Distance Vector Routing Algorithm

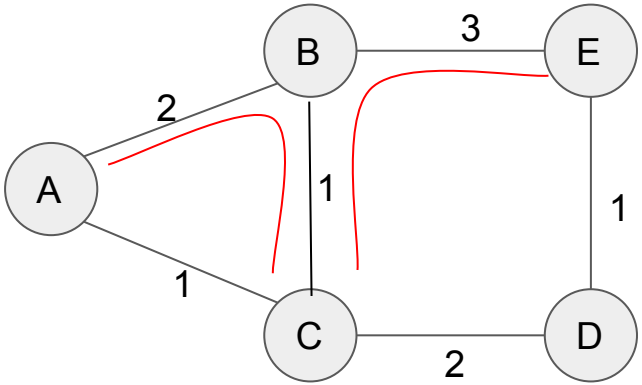
- Dynamic
- Router contains distance (cost) to all other routers as a vector.
- Routing Table Exchange
- Cost will be estimated first, and then will be finalised over the time.

Neighbor	Cost

Routing Table Structure



C Rout. Tab	
A	1
B	1
D	2

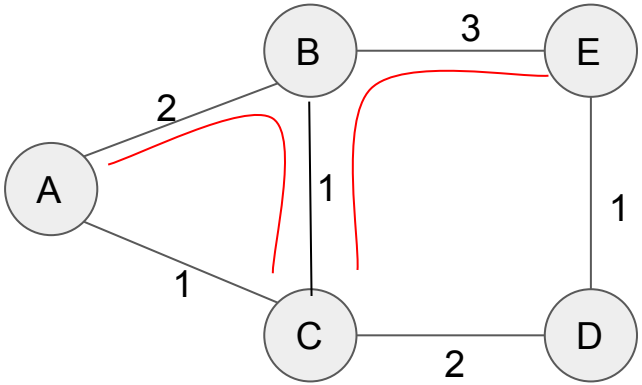


C Rout. Tab	
A	1
B	1
D	2

B Rout. Tab	
A	2
E	3
C	1

C - > B - > A

C - > B - > E

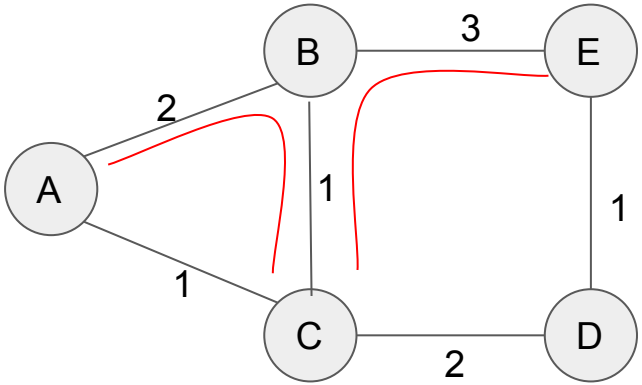


C Rout. Tab	
A	1
B	1
D	2

B Rout. Tab	
A	2
E	3
C	1

C - > B - > A = 1 + 2 = 3

C - > B - > E = 1 + 3 = 4

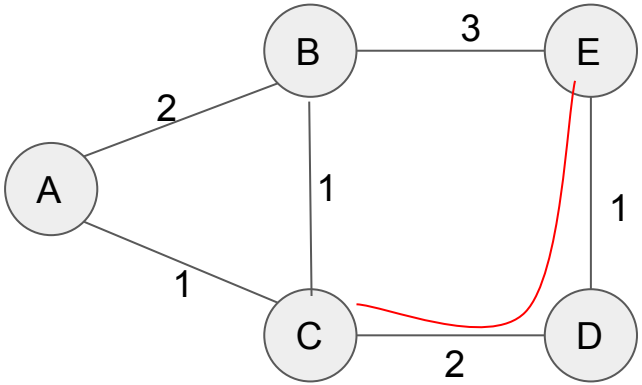


C Rout. Tab	
A	1
B	1
D	2
E	4

B Rout. Tab	
A	2
E	3
C	1

C - > B - > A = 1 + 2 = 3 No Changes will be done for C - > A as 3 > 1

C - > B - > E = 1 + 3 = 4 A New Entry will be made for E

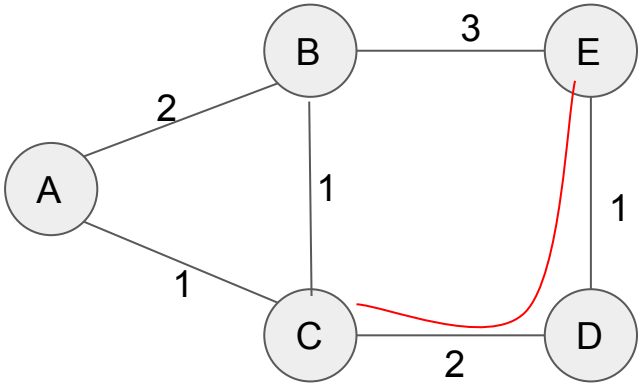


C Rout. Tab	
A	1
B	1
D	2
E	4

B Rout. Tab	
A	2
E	3
C	1

D Rout. Tab	
E	1
C	2

C - > D - > E

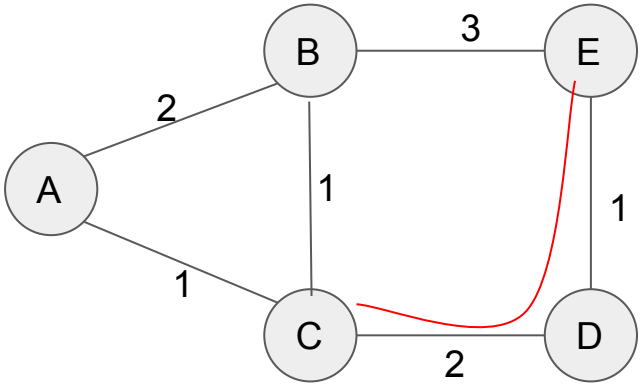


C Rout. Tab	
A	1
B	1
D	2
E	4

B Rout. Tab	
A	2
E	3
C	1

D Rout. Tab	
E	1
C	2

C - > D - > E = 2 + 1 = 3



C Rout. Tab	
A	1
B	1
D	2
E	~~4~~ 3

B Rout. Tab	
A	2
E	3
C	1

D Rout. Tab	
E	1
C	2

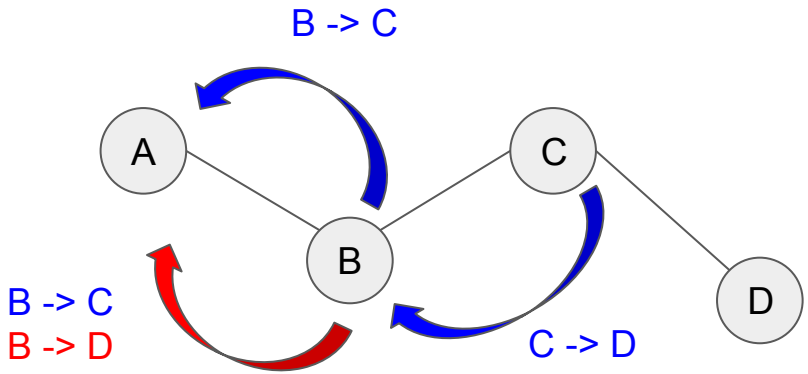
C - > D - > E = 2 + 1 = 3

Entry for E is changed to 3;

3 < 4 , smaller cost to reach E

DV Algorithm

- To begin with, each router knows the distance of its immediate neighbors.
- At every *n* seconds, each router exchanges its routing table to its immediate neighbors.
  - At *'t'* seconds first exchange, each router knows the best distance to two hops.
  - At *'t + n'* seconds, second exchange, then each router knows the best distance to three hops.
  - At *'t + n^2'* seconds, third ……. four hops…. '



First routing table exchange @ *t* seconds

@ A : B gives its best distance to C, A will come to know about C (two hops)

@ B : C gives its best distance to D , B will come to know about D (two hops)

Second routing table exchange @ *t* + *n* seconds

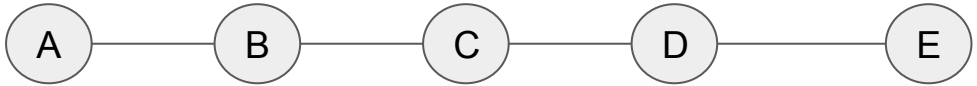
@ A : B gives its best distance to D, A will come to know about D (three hops)

More about DV

- Complete Local Knowledge
  - Believes Neighbors.
- Periodic updates - @ every *t* seconds routing tables exchanges.
- Triggered Updates - when link fails, or node fails.
  - Before *t* seconds routing tables will be exchanged.
- “I’m Alive” messages , No need to send the entire routing table, just a HELLO is enough.
- “Probe - Ack” messages - When a node has some doubt about the neighbor node.

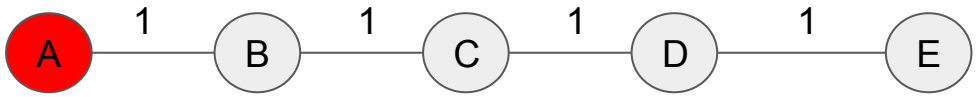
On the other hand

- Reacts rapidly to good news, but leisurely to bad news.
  - Router *P* has a considerably large cost to router *Q*
  - On single routing table exchange, if it finds a much lesser cost to Q, it immediately updates.
- The blind belief on neighbors !!
- It can lead to problems.



1 2 3 4

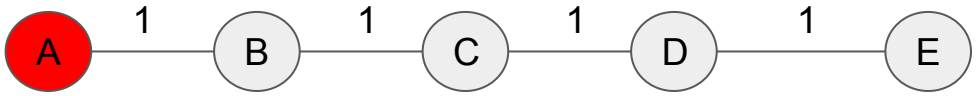
Initially



1 2 3 4

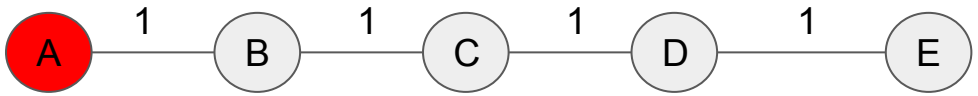
Initially

- A fails or link A->B fails. It's a triggered update.
- Before B sends its updated routing table with distance to A as ∞,
- C does the periodic routing table exchange.
- C says *" I can reach A with a cost of 2 "*
- *B believes C and updates its routing table*



3 2 3 4

After 1 Exchange



1	2	3	4	Initially
3	2	3	4	After 1 Exchange
3	2	3	4	After 2 Exchange

@C : No neighbor has 2 to reach A !!

Might be a wrong entry in the routing table. It updates its routing table..

3	4	3	4	After 2 Exchange
5	4	5	4	After 3 Exchange
5	6	5	6	After 4 Exchange
..	..	..	..	After *n* Exchange

Counting to Infinity Problem.

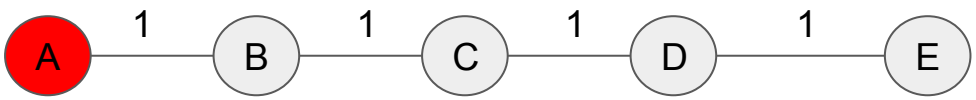
Optimization

- Restricting the path hop to *N* - *make infinity small*
  - If the path cost is 25, once all the routing table entry reaches 25, Something is wrong...
- Send complete path information to neighbors.
  - Neighbors might come to know about the failed node/link
- Split horizon - Carefully forward the routing information.
- Change the routing table structure

Routing Table Structure

Neighbor	Cost

Destination	Cost	Next Hop



1 2 3 4 Initially

- C says *“ I can reach A with a cost of 2 through B“*

Dest.	Cost	Next Hop
A	2	B

B observes that the path goes through it self !! , it will not update its routing table.