SRM Institute of Science and Technology

21CSC302J-COMPUTER NETWORKS

Unit- III

Distance Vector Routing

- ***Dynamic Routing protocol***

- Also called as Bellman-Ford algorithm or Ford Fulkerson algorithm

- ***Used to calculate the shortest path***

- ***Calculates the distance and direction of the next hop*** from the information obtained by the neighboring router.

- ***Necessary to keep track of the topology and inform neighboring devices if any changes occur*** in the topology.

- It is mainly used in ARPANET, and RIP.

- **Each router maintains a distance table *known as Vector***

key points

- Network Information
  - *Every node in the network should have information about its neighboring node.*
  - *Each node in the network is designed to share information with all the nodes in the network.*
- Routing Pattern
  - *In DVR the data shared by the nodes are transmitted only to that node that is linked directly to one or more nodes in the network.*
- Data sharing
  - *The nodes share the information with the neighboring node from time to time as there is a change in network topology.*

key points

- The Distance vector algorithm is ***iterative, asynchronous and distributed.***
  - ***Distributed***
    - ***each node receives information from one or more of its directly attached neighbors,***
    - ***performs calculation and then distributes the result back to its neighbors.***
  - ***Iterative***
    - ***its process continues until no more information is available to be exchanged between neighbors.***
  - ***Asynchronous***
    - ***It does not require that all of its nodes operate in the lock step with each other.***

key points

- Knowledge about the whole network
  - *Each router shares its knowledge through the entire network.*
  - *The Router sends its collected knowledge about the network to its neighbors.*

- Routing only to neighbors
  - *The router sends its knowledge about the network to only those routers which have direct links.*
  - *The router sends whatever it has about the network through the ports.*
  - *The information is received by the router and uses the information to update its own routing table.*

key points

- **Information sharing at regular intervals**
  - ***Within 30 seconds, the router sends the information to the neighboring routers.***

How the DVR Protocol Works

Each router maintains a routing table. It contains only one entry for each router. It contains two parts – a preferred outgoing line to use for that destination and an estimate of time (delay). Tables are updated by exchanging the information with the neighbor's nodes

Each router knows the delay in reaching its neighbors

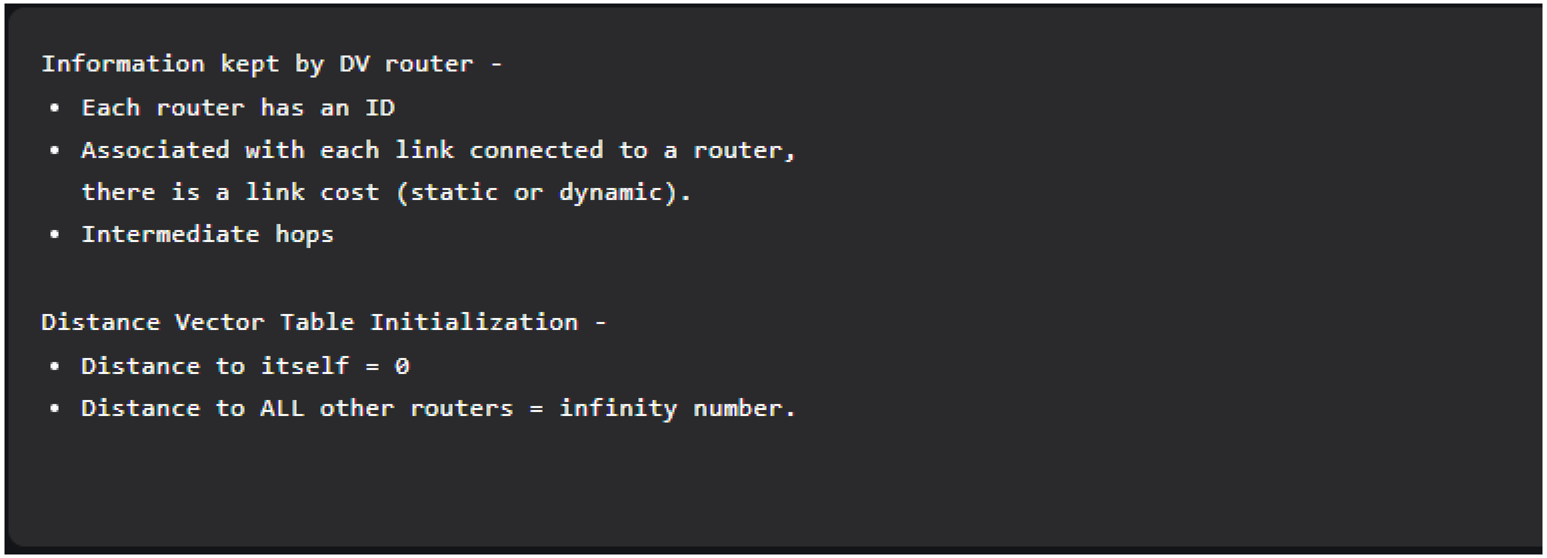
Routers periodically exchange routing tables with each of their neighbors.

How the DVR Protocol Works

It compares the delay in its local table with the delay in the neighbor's table and the cost of reaching that neighbor.

If the path via the neighbor has a lower cost, then the router updates its local table to forward packets to the neighbor.

How the DVR Protocol Works

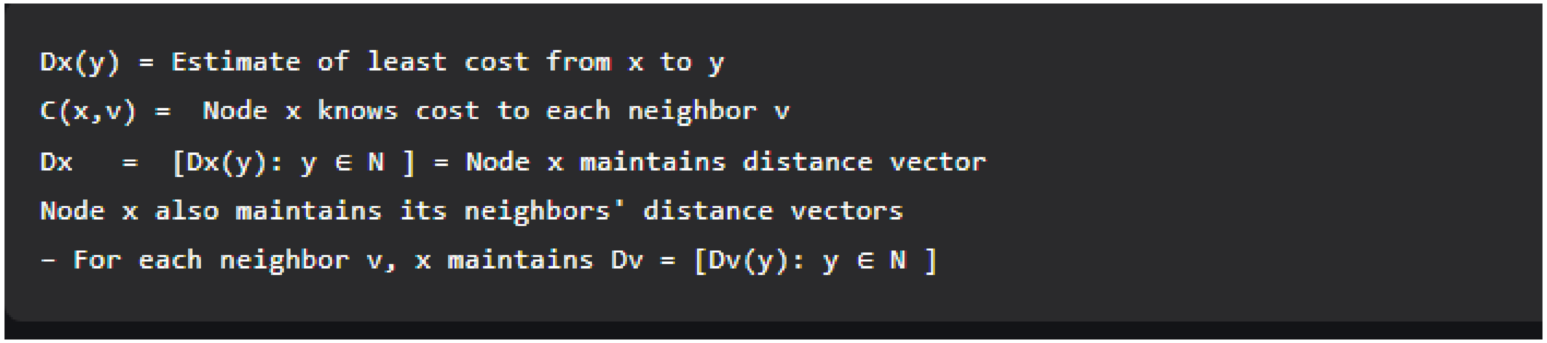


Information kept by DV router -

- Each router has an ID
- Associated with each link connected to a router, there is a link cost (static or dynamic).
- Intermediate hops

Distance Vector Table Initialization -

- Distance to itself = 0
- Distance to ALL other routers = infinity number.



Dx(y) = Estimate of least cost from x to y

C(x,v) = Node x knows cost to each neighbor v

Dx   = [Dx(y): y ∈ N] = Node x maintains distance vector

Node x also maintains its neighbors' distance vectors

– For each neighbor v, x maintains Dv = [Dv(y): y ∈ N]

How the DVR Protocol Works

When a node x receives new DV estimate from any neighbor v, it saves v's distance vector and it updates its own DV using

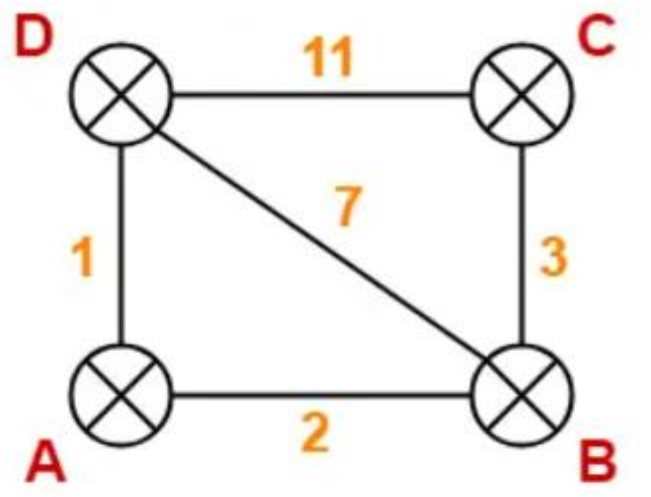
$$d_x(y) = \min_v\{c(x,v) + d_v(y)\}$$

```
At each node x,

Initialization

for all destinations y in N:
$D_x(y) = c(x,y)$     // If y is not a neighbor then $c(x,y) = \infty$
for each neighbor w
$D_w(y) = ?$     for all destination y in N.
for each neighbor w
send distance vector $D_x = [ D_x(y) : y \in N ]$ to w
loop
    wait(until I receive any distance vector from some neighbor w)
    for each y in N:
    $D_x(y) = \min_v\{c(x,v)+D_v(y)\}$
If $D_x(y)$ is changed for any destination y
Send distance vector $D_x = [ D_x(y) : y \in N ]$ to all neighbors
forever
```

How the DVR Protocol Works – STEP 01



At Router A-

Destination	Distance	Next Hop
A	0	A
B	2	B
C	∞	–
D	1	D

At Router B-

Destination	Distance	Next Hop
A	2	A
B	0	B
C	3	C
D	7	D

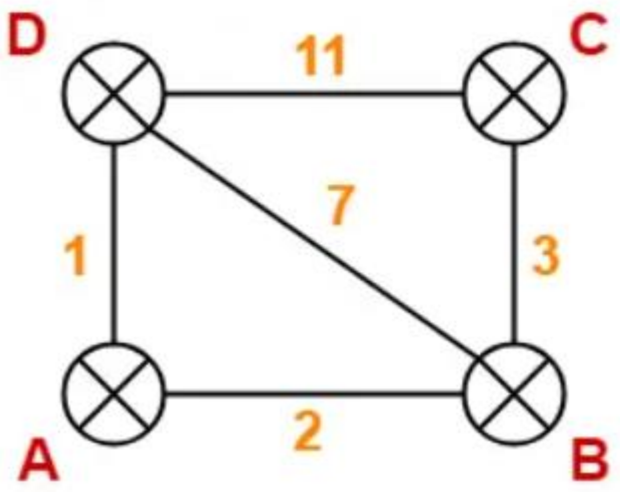
How the DVR Protocol Works - STEP 01

At Router C-

Destination	Distance	Next Hop
A	∞	–
B	3	B
C	0	C
D	11	D

At Router D -

Destination	Distance	Next Hop
A	1	A
B	7	B
C	11	C
D	0	D



STEP 02

At Router A- *Router A receives distance vectors from its neighbors B and D.*

From B
2
0
3
7

Cost(A→B) = 2

From D
1
7
11
0

Cost(A→D) = 1

Destination	Distance	Next hop
A	0	A
B		
C		
D		

New Routing Table at Router A

Cost of reaching destination B from router A = min { 2+0 , 1+7 } = 2 via B.

Cost of reaching destination C from router A = min { 2+3 , 1+11 } = 5 via B.

Cost of reaching destination D from router A = min { 2+7 , 1+0 } = 1 via D

STEP 02

At Router A-

- Router A can reach the destination router B via its neighbor B or neighbor D.
- It chooses the path which gives the minimum cost.
- Cost of reaching router B from router A via neighbor B = Cost (A→B) + Cost (B→B)= **2 + 0** = 2
- Cost of reaching router B from router A via neighbor D = Cost (A→D) + Cost (D→B) = **1 + 7** = 8
- Since the cost is minimum via neighbor B, so router A chooses the path via B.
- It creates an entry (2, B) for destination B in its new routing table.
- Similarly, we calculate the shortest path distance to each destination router at every router.

Destination	Distance	Next Hop
A	0	A
B	2	B
C	5	B
D	1	D

STEP 02

At Router B- *Router B receives distance vectors from its neighbors A, C and D*

From A	From C	From D
0	∞	1
2	3	7
∞	0	11
1	11	0
Cost (B→A) = 2	Cost (B→C) = 3	Cost (B→D) = 7

Destination	Distance	Next hop
A		
B	0	B
C		
D		

New Routing Table at Router B

Cost of reaching destination A from router B = min { 2+0 , 3+∞ , 7+1 } = 2 via A

Cost of reaching destination C from router B = min { 2+∞ , 3+0 , 7+11 } = 3 via C

Cost of reaching destination D from router B = min { 2+1 , 3+11 , 7+0 } = 3 via A

STEP 02

At Router B-

Destination	Distance	Next Hop
A	2	A
B	0	B
C	3	C
D	3	A

STEP 02

At Router C- *Router C receives distance vectors from its neighbors B and D*

From B
2
0
3
7

Cost (C→B) = 3

From D
1
7
11
0

Cost (C→D) = 11

Destination	Distance	Next hop
A		
B		
C	0	C
D		

New Routing Table at Router C

Cost of reaching destination A from router C = min { 3+2 , 11+1 } = 5 via B

Cost of reaching destination B from router C = min { 3+0 , 11+7 } = 3 via B

Cost of reaching destination D from router C = min { 3+7 , 11+0 } = 10 via B

STEP 02

At Router C-

Destination	Distance	Next Hop
A	5	B
B	3	B
C	0	C
D	10	B

STEP 02

At Router D- *Router D receives distance vectors from its neighbors A, B and C*

From A	From B	From C
0	2	∞
2	0	3
∞	3	0
1	7	11
Cost (D→A) = 1	Cost (D→B) = 7	Cost (D→C) = 11

Destination	Distance	Next hop
A		
B		
C		
D	0	D

New Routing Table at Router D

Cost of reaching destination A from router D = min { 1+0 , 7+2 , 11+∞ } = 1 via A

Cost of reaching destination B from router D = min { 1+2 , 7+0 , 11+3 } = 3 via A

Cost of reaching destination C from router D = min { 1+∞ , 7+3 , 11+0 } = 10 via B

STEP 02

At Router D-

Destination	Distance	Next Hop
A	1	A
B	3	A
C	10	B
D	0	D

STEP 03

Each router exchanges its distance vector obtained in Step-02 with its neighboring routers

After exchanging the distance vectors, each router prepares a new routing table

STEP 03

At Router A- *Router A receives distance vectors from its neighbors B and D.*

From B

2
0
3
3

Cost(A→B) = 2

From D

1
3
10
0

Cost(A→D) = 1

Destination	Distance	Next hop
A	0	A
B		
C		
D		

New Routing Table at Router A

Cost of reaching destination B from router A = min { 2+0 , 1+3 } = 2 via B

Cost of reaching destination C from router A = min { 2+3 , 1+10 } = 5 via B

Cost of reaching destination D from router A = min { 2+3 , 1+0 } = 1 via D

SRM INSTITUTE OF SCIENCE & TECHNOLOGY
Deemed to be University u/s 3 of UGC Act, 1956

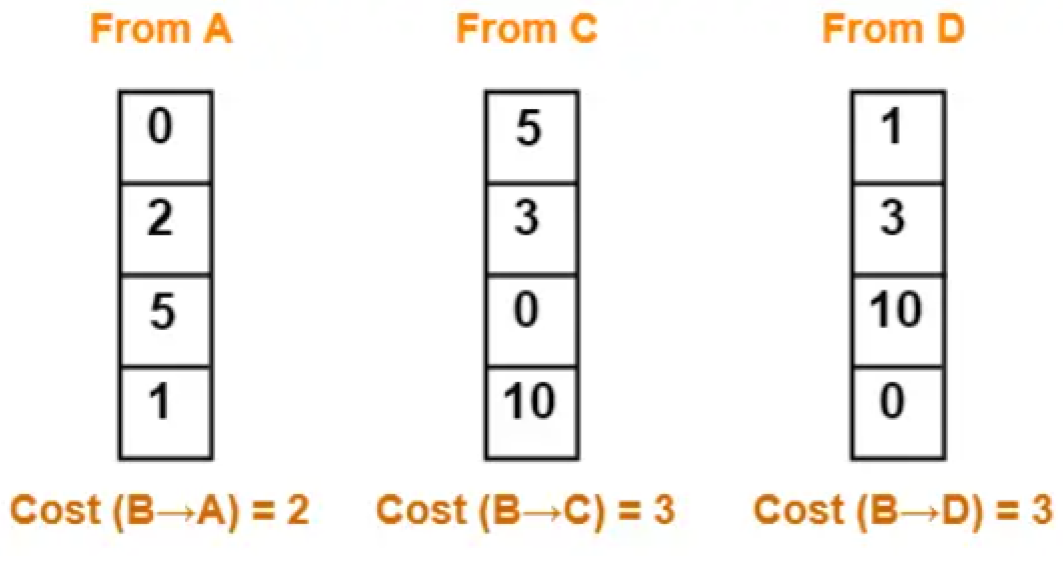
STEP 03

At Router A-

Destination	Distance	Next Hop
A	0	A
B	2	B
C	5	B
D	1	D

STEP 03

At Router B- *Router B receives distance vectors from its neighbors A, C and D*



Destination	Distance	Next hop
A		
B	0	B
C		
D		

New Routing Table at Router B

Cost of reaching destination A from router B = min { 2+0 , 3+5 , 3+1 } = 2 via A

Cost of reaching destination C from router B = min { 2+5 , 3+0 , 3+10 } = 3 via C

Cost of reaching destination D from router B = min { 2+1 , 3+10 , 3+0 } = 3 via A

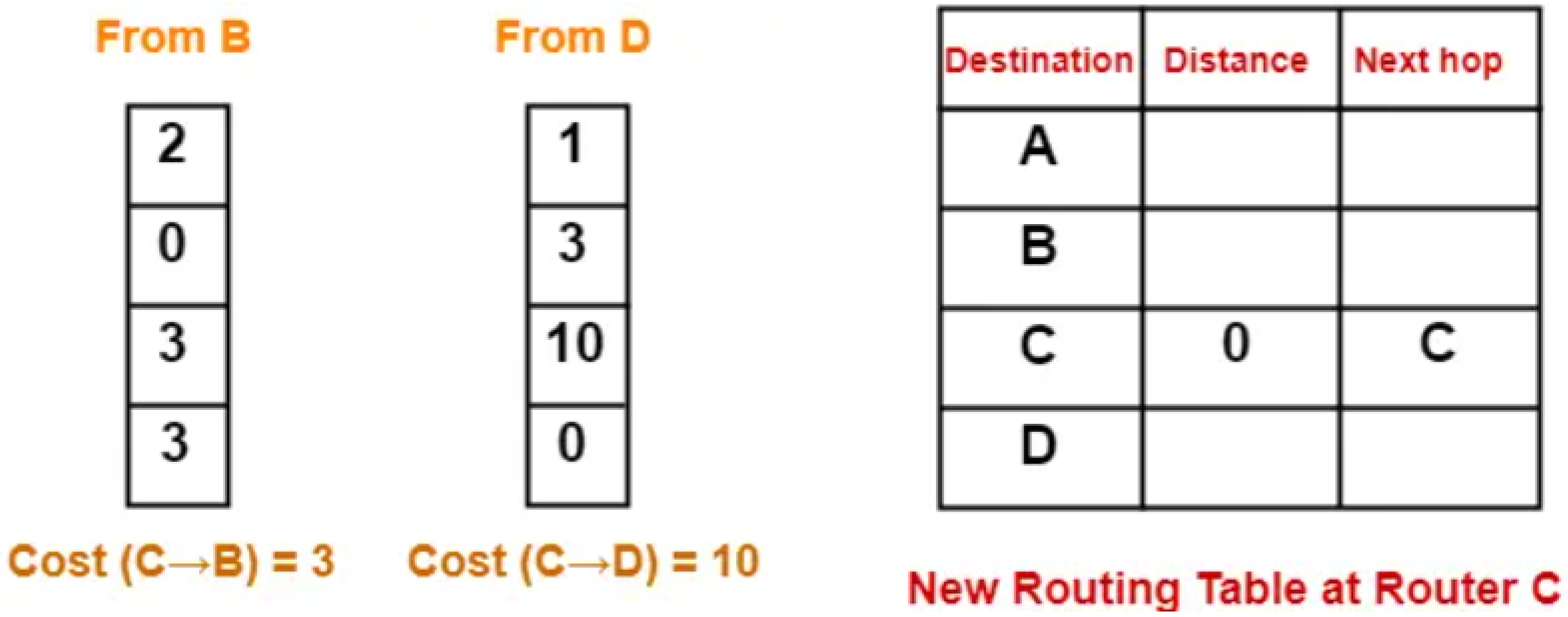
STEP 03

At Router B-

Destination	Distance	Next Hop
A	2	A
B	0	B
C	3	C
D	3	A

STEP 03

At Router C- *Router C receives distance vectors from its neighbors B and D*



From B

2
0
3
3

Cost (C→B) = 3

From D

1
3
10
0

Cost (C→D) = 10

Destination	Distance	Next hop
A		
B		
C	0	C
D		

New Routing Table at Router C

Cost of reaching destination A from router C = min { 3+2 , 10+1 } = 5 via B

Cost of reaching destination B from router C = min { 3+0 , 10+3 } = 3 via B

Cost of reaching destination D from router C = min { 3+3 , 10+0 } = 6 via B

SRM INSTITUTE OF SCIENCE & TECHNOLOGY
Deemed to be University u/s 3 of UGC Act, 1956

STEP 03

At Router C-

Destination	Distance	Next Hop
A	5	B
B	3	B
C	0	C
D	6	B

SRM INSTITUTE OF SCIENCE & TECHNOLOGY
Deemed to be University u/s 3 of UGC Act, 1956

STEP 03

At Router D- *Router D receives distance vectors from its neighbors A, B and C*

From A

0
2
5
1

Cost (D→A) = 1

From B

2
0
3
3

Cost (D→B) = 3

From C

5
3
0
10

Cost (D→C) = 10

Destination	Distance	Next hop
A		
B		
C		
D	0	D

New Routing Table at Router D

Cost of reaching destination A from router D = min { 1+0 , 3+2 , 10+5 } = 1 via A

Cost of reaching destination B from router D = min { 1+2 , 3+0 , 10+3 } = 3 via A

Cost of reaching destination C from router D = min { 1+5 , 3+3 , 10+0 } = 6 via A

STEP 02

At Router D-

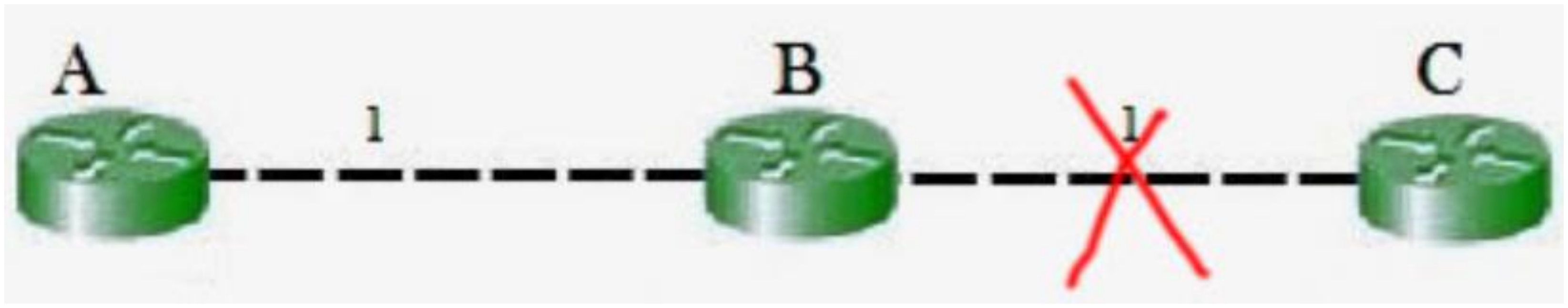
Destination	Distance	Next Hop
A	1	A
B	3	A
C	6	A
D	0	D

Count to Infinity problem

- The main issue with Distance Vector Routing (DVR) protocols is Routing Loops since Bellman-Ford Algorithm cannot prevent loops.

- This routing loop in the DVR network causes the Count to Infinity Problem.

- Routing loops usually occur when an interface goes down or two routers send updates at the same time.



Count to Infinity problem

- If the link between B and C is disconnected, then B will know that it can no longer get to C via that link and will remove it from its table.
- Before it can send any updates it’s possible that it will receive an update from A which will be advertising that it can get to C at a cost of 2.
- B can get to A at a cost of 1, so it will update a route to C via A at a cost of 3.
- A will then receive updates from B later and update its cost to 4.

Count to Infinity problem

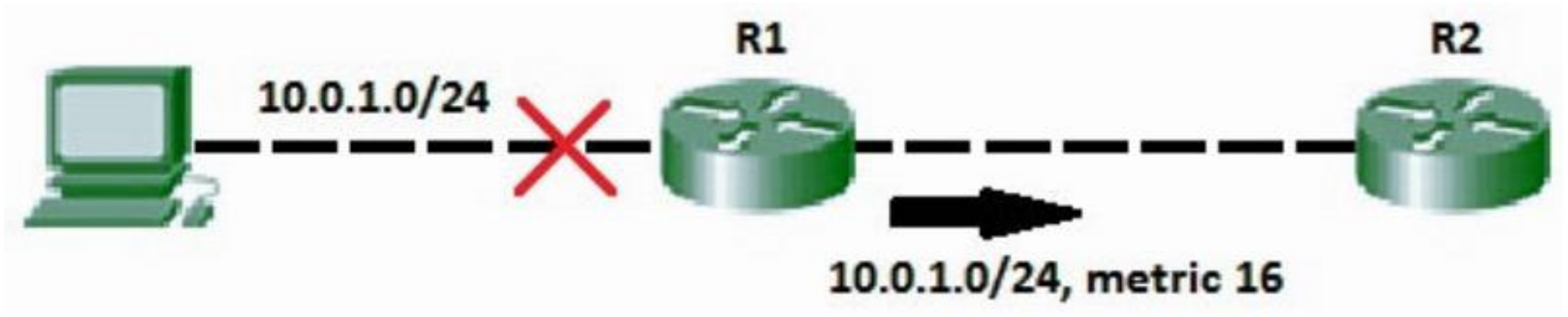
- They will then go on feeding each other bad information toward infinity which is called as Count to Infinity problem.

Solution

- Route Poisoning
  - ***When a route fails, distance vector protocols spread the bad news about a route failure by poisoning the route.***
  - ***Route poisoning refers to the practice of advertising a route, but with a special metric value called Infinity.***
  - ***Routers consider routes advertised with an infinite metric to have failed.***
  - ***Each distance vector routing protocol uses the concept of an actual metric value that represents infinity.***
  - ***RIP defines infinity as 16.***
  - ***The main disadvantage of poison reverse is that it can significantly increase the size of routing announcements in certain fairly common network topologies.***

Solution

- Route Poisoning

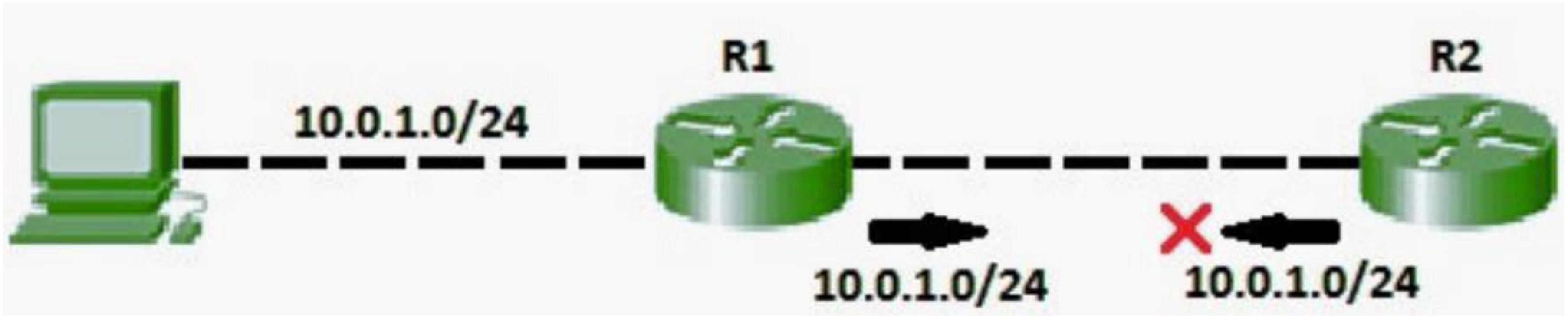


Solution

- Split Horizon
  - *If the link between B and C goes down, and B had received a route from A, B could end up using that route via A.*
  - *A would send the packet right back to B, creating a loop. But according to the Split horizon Rule, Node A does not advertise its route for C (namely A to B to C) back to B.*
  - *On the surface, this seems redundant since B will never route via node A because the route costs more than the direct route from B to C.*

Solution

- Consider the following network topology showing Split horizon



- In addition to these, we can also use split horizon with route poisoning were above both techniques will be used combined to achieve efficiency and less increase the size of routing announcements.

- Split horizon with Poison reverse technique is used by Routing Information Protocol (RIP) to reduce routing loops.

Solution

- Additionally, Holddown timers can be used to avoid the formation of loops.
- The hold-down timer immediately starts when the router is informed that the attached link is down.
- Till this time, the router ignores all updates of the down route unless it receives an update from the router of that downed link.
- During the timer, If the downlink is reachable again, the routing table can be updated.

Link State Routing

- Each router shares the knowledge of its neighborhood with every other router in the internetwork.

keys to understand

- **Knowledge about the neighborhood**
  - ***Instead of sending its routing table, a router sends the information about its neighborhood only.***
  - ***A router broadcast its identities and cost of the directly attached links to other routers.***

- **Flooding**
  - ***Each router sends the information to every other router on the internetwork except its neighbors.***
  - ***This process is known as Flooding. Every router that receives the packet sends the copies to all its neighbors.***
  - ***Finally, each and every router receives a copy of the same information.***

keys to understand the Link State Routing algorithm

- Information sharing
  - ***A router sends the information to every other router only when the change occurs in the information.***

Link state protocols

- Also called shortest-path-first protocols.
- Have a complete picture of the network topology.
- Know more about the whole network than any distance vector protocol.
- Three separate tables are created on each link state routing enabled router.
  - ***One table is used to hold details about directly connected neighbors,***
  - ***one is used to hold the topology of the entire internetwork and***
  - ***the last one is used to hold the actual routing table.***

Link state protocols

- send information about directly connected links to all the routers in the network.

- Examples of Link state routing protocols include
  - ***OSPF - Open Shortest Path First and***
  - ***IS-IS - Intermediate System to Intermediate System.***

- Link State routing protocols hold 3 distinctive tables:
  - ***a neighbor table,***
  - ***a topology table, and***
  - ***an actual routing table.***

Link state protocols

- Follows four simple steps; each link state enabled router must perform the following:
  - ***Discover its neighbors and build its neighbor table***
  - ***Measure the cost (delay, bandwidth, etc.,) to each of its neighbors***
  - ***Construct and send a routing update telling all it has learned to all routers in the network***
  - ***Apply the Dijkstra algorithm to construct the shortest path to all possible destinations***

Four Step Process

Neighbor discovery

Each Link State enabled router *periodically sends a HELLO message* on each of its links. Neighbor routers respond to these HELLO messages identifying themselves. Within the replies, network addresses of the routers are attached and are used by the HELLO initiator to build up its neighbor table.

Four Step Process

Measuring Link Cost

A set of tests is performed on each router to measure the cost to each of its neighbors. The cost could be a measure of the end-to-end delay, throughput, or a combination of these metrics. The important thing to know is that each link state enabled router has to somehow possess an estimate of the cost for each of its links.

Four Step Process

Building and Distributing Link State Packets

Each router builds up a packet containing its neighbors and the corresponding link costs to these neighbors. At the beginning of the packet, each router adds its identity along with a sequence number and an age parameter, the latter being used to ensure no packet will wander around for an indefinite period of time. After the construction process, the packet is flooded in the network.

Four Step Process

Evaluating Shortest Paths

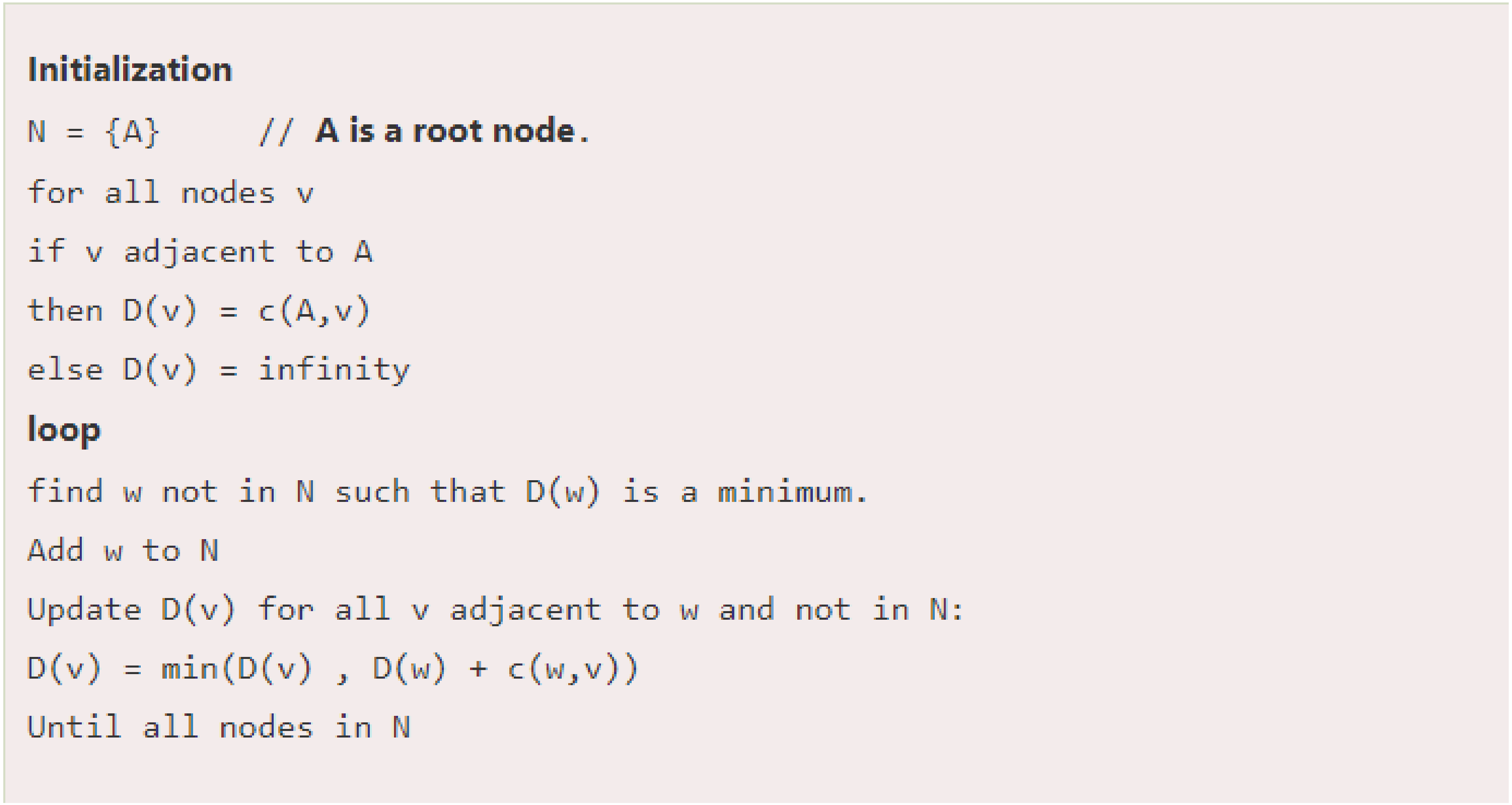
Using all the details from its link state table, a router is able to compute, using the Dijkstra algorithm, the shortest path to any given destination.

- c(i , j): Link cost from node i to node j. If i and j nodes are not directly linked, then c(i , j) = ∞.

- D(v): Defines the cost of the path from source to destination v that has the least cost currently.

- P(v): Defines the previous node (neighbor of v) along with current least cost path from source to v.

- N: The total number of nodes available in the network.

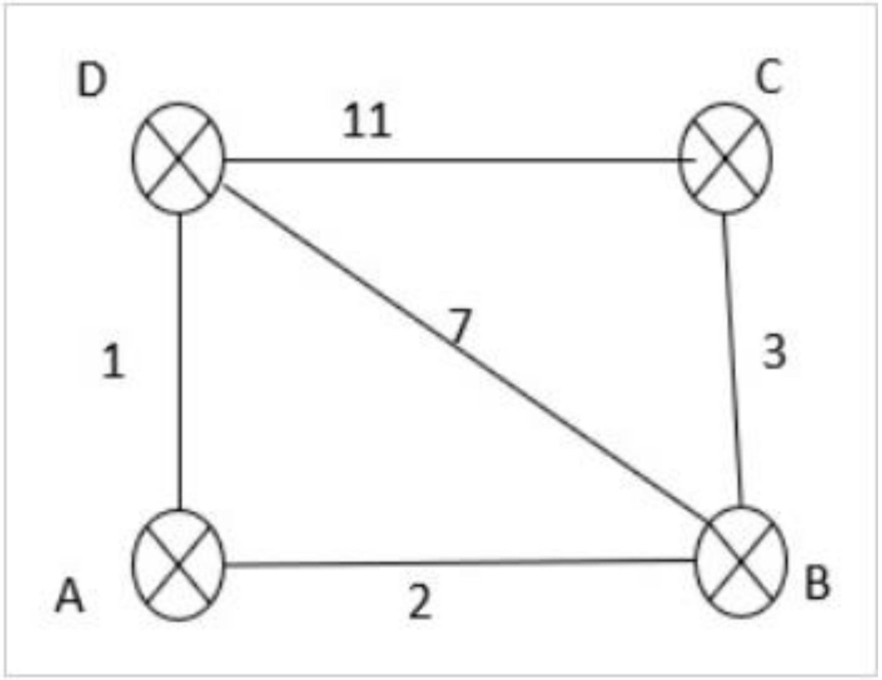


Initialization

```
N = {A}      // A is a root node.
for all nodes v
if v adjacent to A
then D(v) = c(A,v)
else D(v) = infinity
```

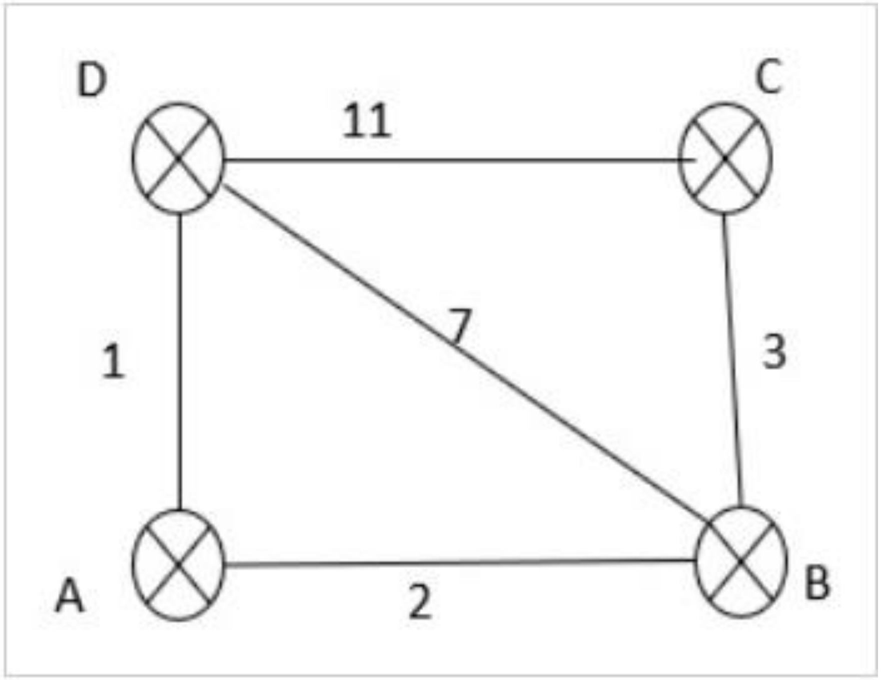
loop

```
find w not in N such that D(w) is a minimum.
Add w to N
Update D(v) for all v adjacent to w and not in N:
D(v) = min(D(v) , D(w) + c(w,v))
Until all nodes in N
```



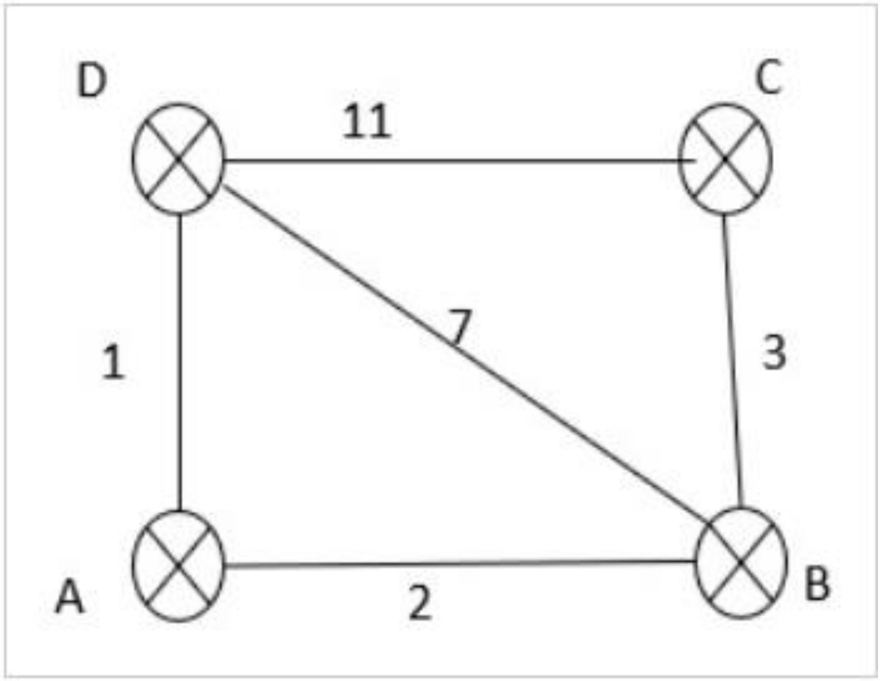
Router D

C	11
B	7
A	1



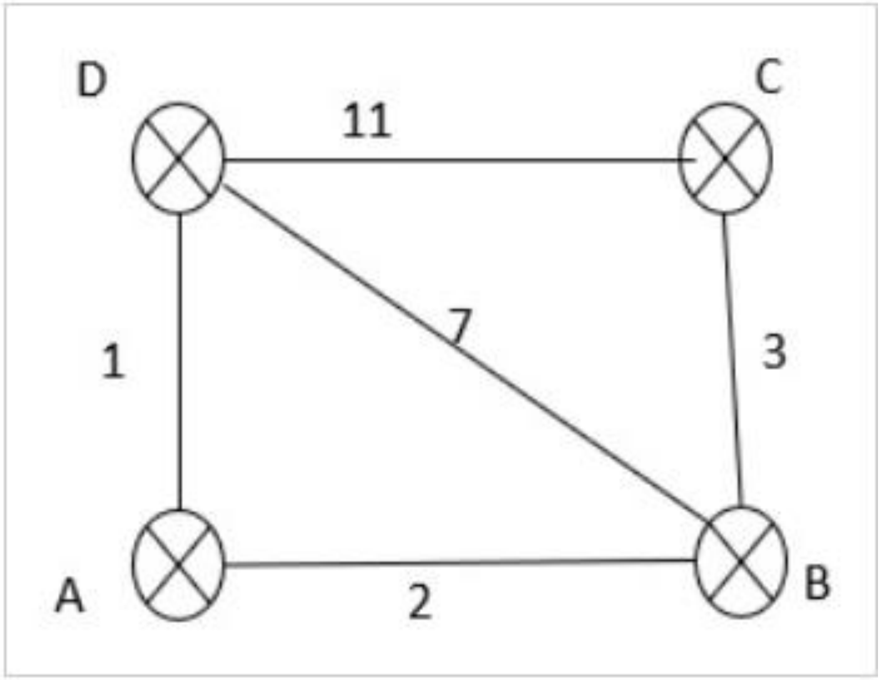
Router C

D	11
B	3



Router A

B	2
D	1



Router B

A	2
D	7
C	3

Advantages

- Fast Network Convergence
  - ***It is the main advantage of the link-state routing protocol.***
  - ***Because of receiving an LSP, link-state routing protocols immediately flood the LSP out of all interfaces without any changes except for the interface from which the LSP was received.***

- Topological Map
  - ***Link-state routing uses a topological map or SPF tree for creating the network topology.***
  - ***Using the SPF tree, each router can separately determine the shortest path to every network.***

Advantages

- Hierarchical Design
  - *Link-state routing protocols use multiple areas and create a hierarchical design to network areas.*
  - *The multiple areas allow better route summarization.*
- Event-driven Updates
  - *After initial flooding of LSPs, the LSPs are sent only when there is a change in the topology and contain only the information regarding that change.*
  - *The LSP contains only the information about the affected link. The link-state never sends periodic updates.*

Disadvantages

- Memory Requirements
  - ***The link-state routing protocol creates and maintains a database and SPF tree.***
  - ***The database and SPF tree required more memory than a distance vector protocol.***

- Processing Requirements
  - ***Link-state routing protocols also require more CPU processing because the SPF algorithm requires more CPU time than distance-vector algorithms just like Bellman-Ford because link-state protocols build a complete map of the topology.***

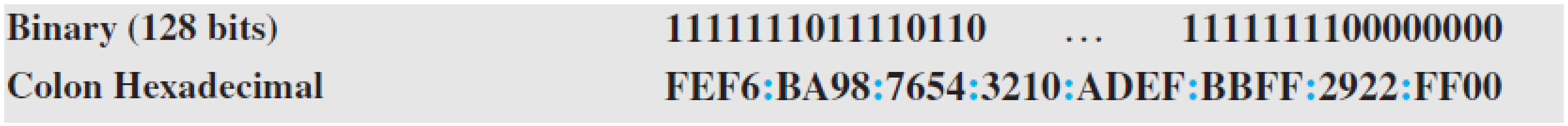
Disadvantages

- Bandwidth Requirements
  - ***The link-state routing protocol floods link-state packet during initial start-up and also at the event like network breakdown, and network topology changes, which affect the available bandwidth on a network.***
  - ***If the network is not stable it also creates issues on the bandwidth of the network.***

IPV6 Addressing

- The main reason for migration from IPv4 to IPv6 is the small size of the address space in IPv4.

- An IPv6 address is 128 bits or 16 bytes (octets) long, four times the address length in IPv4.



Binary (128 bits)	1111111011110110	...	1111111100000000
Colon Hexadecimal	FEF6:BA98:7654:3210:ADEF:BBFF:2922:FF00		

- Binary notation is used when the addresses are stored in a computer.

- The colon hexadecimal notation (or colon hex for short) divides the address into eight sections, each made of four hexadecimal digits separated by colons.

- Although an IPv6 address, even in hexadecimal format, is very long, many of the digits are zeros.

- The leading zeros of a section can be omitted. Using this form of abbreviation, 0074 can be written as 74, 000F as F, and 0000 as 0.

- often called zero compression,

- can be applied to colon hex notation if there are consecutive sections consisting of zeros only.

- We can remove all the zeros and replace them with a double semicolon.

Thank You