AMRITA
VISHWA VIDYAPEETHAM
DEEMED TO BE UNIVERSITY

15CSE312
COMPUTER NETWORKS
3-0-0 3

Chapter 4: Network Layer

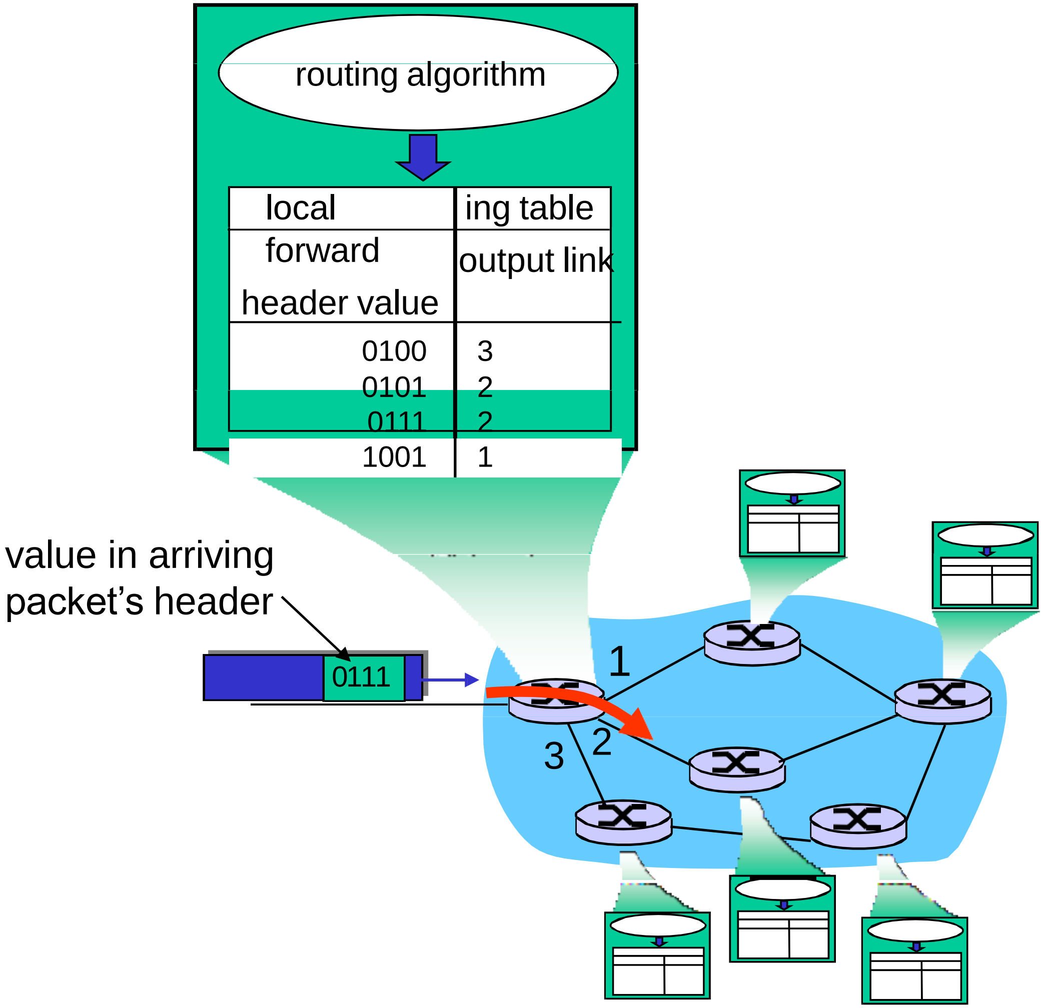
Routing Protocols

- Link state
- Distance Vector
- Hierarchical routing

Chapter 4: Network Layer

- 4. 1 Introduction
- 4.2 Virtual circuit and datagram networks
- 4.3 What's inside a router
- 4.4 IP: Internet Protocol
  - Datagram format
  - IPv4 addressing
  - ICMP
  - IPv6
- 4.5 Routing algorithms
  - Link state
  - Distance Vector
  - Hierarchical routing
- 4.6 Routing in the Internet
  - RIP
  - OSPF
  - BGP
- 4.7 Broadcast and multicast routing

Interplay between routing, forwarding

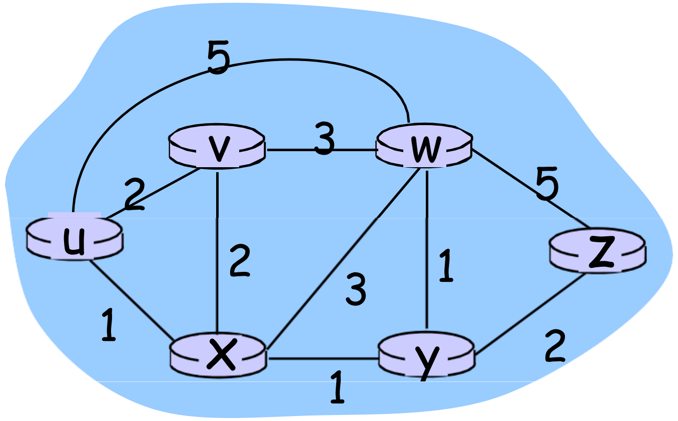


local ing table
forward

header value	output link
0100	3
0101	2
0111	2
1001	1

value in arriving packet's header

Graph abstraction



Graph: G = (N,E)

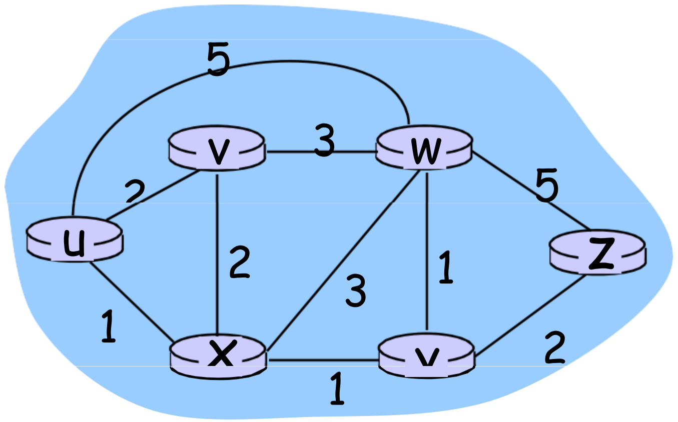
N = set of routers = { u, v, w, x, y, z }

E = set of links ={ (u,v), (u,x), (v,x), (v,w), (x,w), (x,y), (w,y), (w,z), (y,z) }

Remark: Graph abstraction is useful in other network contexts

Example: P2P, where N is set of peers and E is set of TCP connections

Graph abstraction: costs



- c(x,x') = cost of link (x,x')

  - e.g., c(w,z) = 5

- cost could always be 1, or inversely related to bandwidth, or inversely related to congestion

Cost of path $(x_1, x_2, x_3,\ldots, x_p) = c(x_1,x_2) + c(x_2,x_3) + \ldots + c(x_{p-1},x_p)$

Question: What's the least-cost path between u and z ?

Routing algorithm: algorithm that finds least-cost path

Routing Algorithm classification

Global or decentralized information?

Global:

- all routers have complete topology, link cost info
- "link state" algorithms

Decentralized:

- router knows physically- connected neighbors, link costs to neighbors
- iterative process of computation, exchange of info with neighbors
- "distance vector" algorithms

Static or dynamic?

Static:

- routes change slowly over time

Dynamic:

- routes change more quickly
    - periodic update
    - in response to lk cost changes

Chapter 4: Network Layer

- 4. 1 Introduction
- 4.2 Virtual circuit and datagram networks
- 4.3 What's inside a router
- 4.4 IP: Internet Protocol
  - Datagram format
  - IPv4 addressing
  - ICMP
  - IPv6
- 4.5 Routing algorithms
  - Link state
  - Distance Vector
  - Hierarchical routing
- 4.6 Routing in the Internet
  - RIP
  - OSPF
  - BGP
- 4.7 Broadcast and multicast routing

A Link-State Routing Algorithm

Dijkstra's algorithm

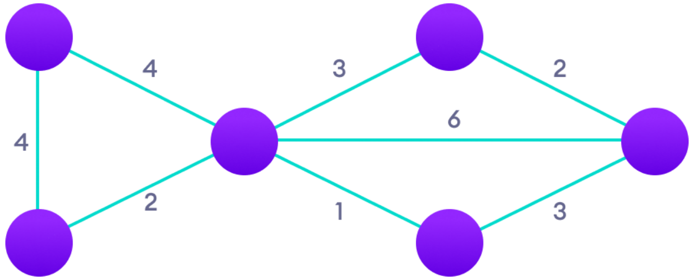
- net topology, link costs known to all nodes
  - accomplished via "link state broadcast"
  - all nodes have same info
- computes least cost paths from one node ('source") to all other nodes
  - gives forwarding table for that node
- iterative: after k iterations, know least cost path to k dest.'s

Notation:

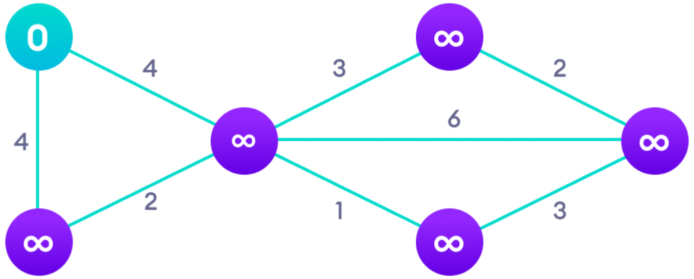
- c(x,y): link cost from node x to y; = ∞ if not direct neighbors
- D(v): current value of cost of path from source to dest. v
- p(v): predecessor node along path from source to v
- N': set of nodes whose least cost path definitively known

Dijsktra's Algorithm

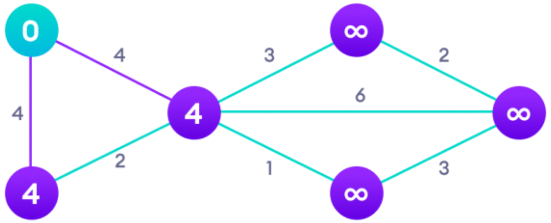
```
Initialization:
  $N' = \{u\}$
  for all nodes v
    if v adjacent to u
        then $D(v) = c(u,v)$
    else $D(v) = \infty$

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' :
     $D(v) = \min( D(v), D(w) + c(w,v) )$
     /* new cost to v is either old cost to v or known
      shortest path cost to w plus cost from w to v */
until all nodes in N'
```



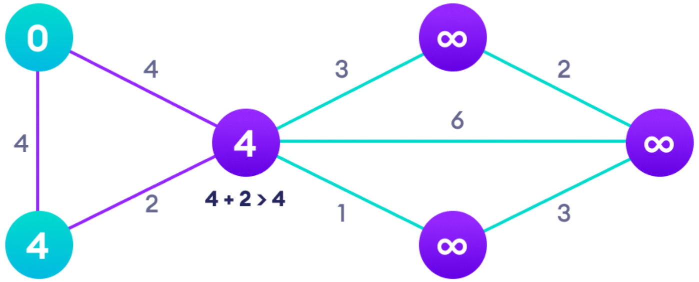
Step: 1



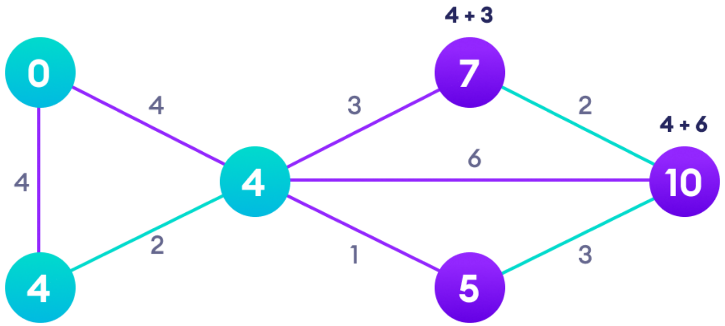
Step: 2



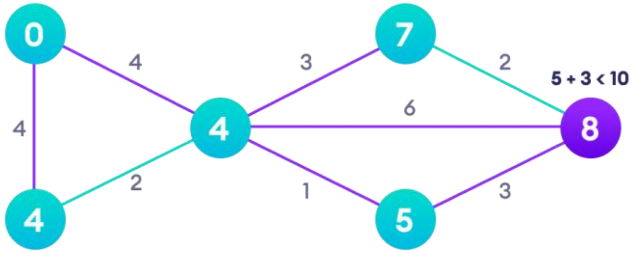
Step: 3



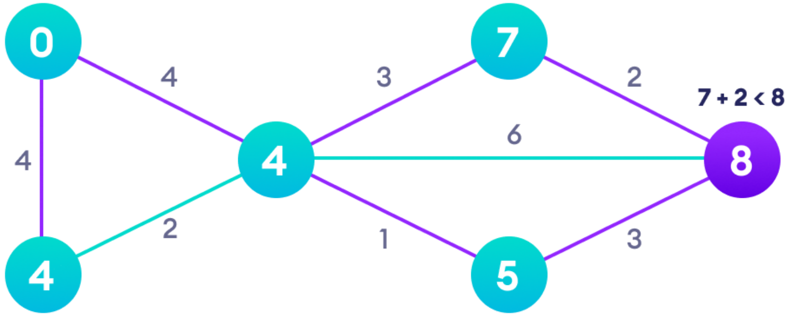
Step: 4



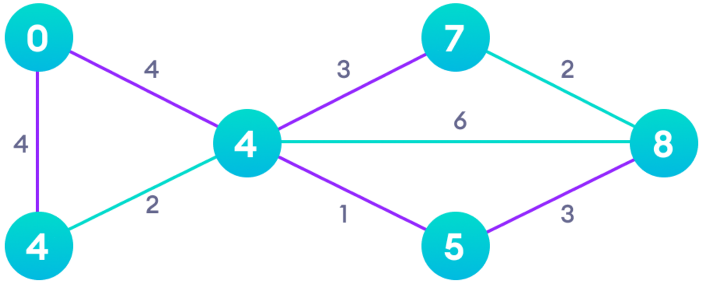
Step: 5



Step: 6



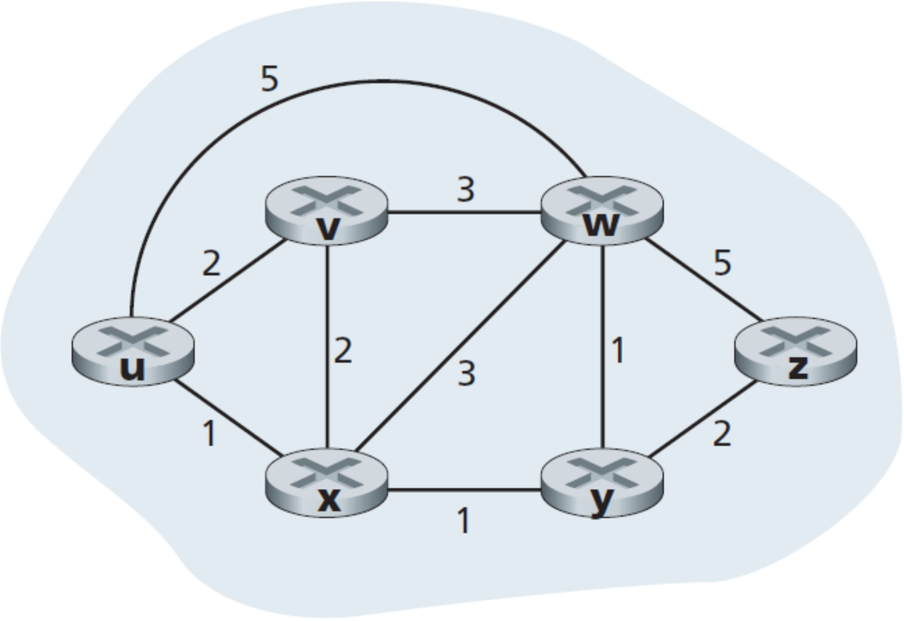
Step: 7



Step: 8

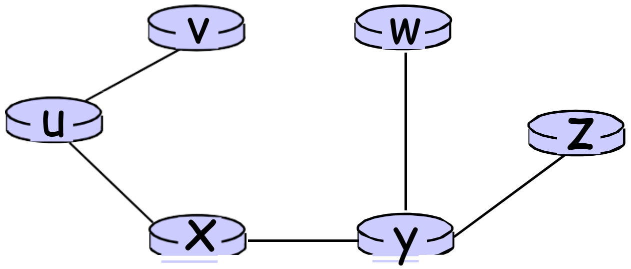
Dijkstra'

Step	N'	D(v),p(v)	D(w),p(w)	D(x),p(x)	D(y),p(y)	D(z),p(z)
0	u	2,u	5,u	1,u	∞	∞
1	ux	2,u	4,x		2,x	∞
2	uxy	2,u	3,y			4,y
3	uxyv		3,y			4,y
4	uxyvw					4,y
5	uxyvwz					



Dijkstra’s algorithm: example (2)

Resulting shortest-path tree from u:



Resulting forwarding table in u:

destination	link
v	(u,v)
x.	(u,x)
y.	(u,x)
w	(u,x)
z	(u,x)

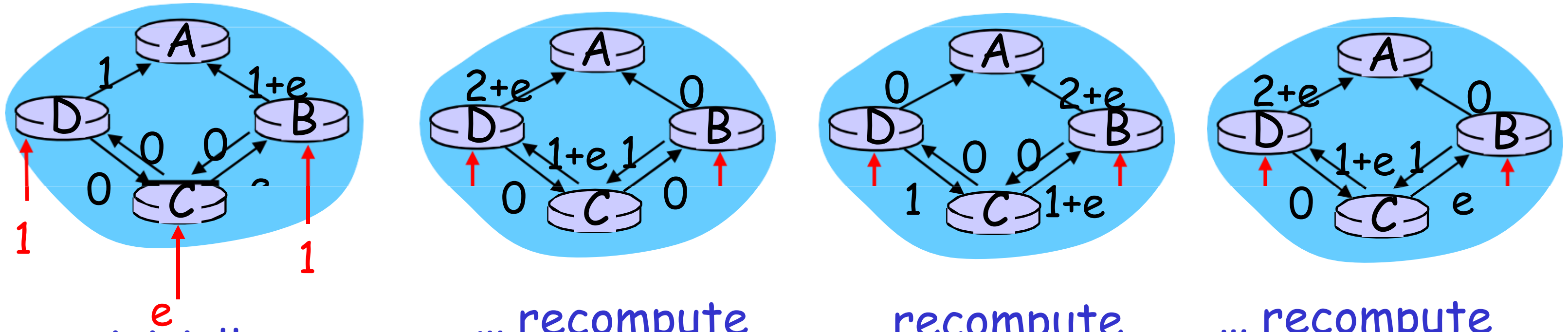
Dijkstra's algorithm, discussion

Algorithm complexity: n nodes

- each iteration: need to check all nodes, w, not in N
- n(n+1)/2 comparisons: $O(n^2)$
- more efficient implementations possible: O(nlogn)

Oscillations possible:

- e.g., link cost = amount of carried traffic



initially

… recompute routing

… recompute

… recompute

Chapter 4: Network Layer

- 4. 1 Introduction
- 4.2 Virtual circuit and datagram networks
- 4.3 What's inside a router
- 4.4 IP: Internet Protocol
  - Datagram format
  - IPv4 addressing
  - ICMP
  - IPv6
- 4.5 Routing algorithms
  - Link state
  - Distance Vector
  - Hierarchical routing
- 4.6 Routing in the Internet
  - RIP
  - OSPF
  - BGP
- 4.7 Broadcast and multicast routing

Distance Vector Algorithm

Bellman-Ford Equation (dynamic programming)

Define

$d_x(y)$:= cost of least-cost path from x to y

Then

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

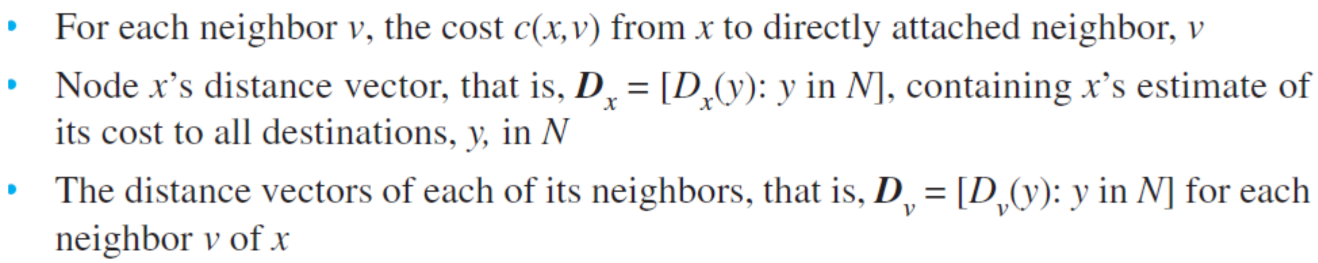
where min is taken over all neighbors v of x

Distance Vector Algorithm

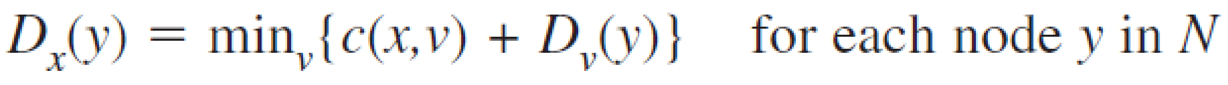
- ❐ $D_x(y)$ = estimate of least cost from x to y
- ❐ Node x knows cost to each neighbor v: $c(x,v)$
- ❐ Node x maintains distance vector $\mathbf{D}_x = [D_x(y): y \in N]$
- ❐ Node x also maintains its neighbors' distance vectors
  - ❍ For each neighbor v, x maintains $\mathbf{D}_v = [D_v(y): y \in N]$

Distance vector algorithm (4)

With the DV algorithm, each node *x* maintains the following routing information:



- For each neighbor *v*, the cost *c*(*x,v*) from *x* to directly attached neighbor, *v*
- Node *x*'s distance vector, that is, $\boldsymbol{D}_x = [D_x(y)$: *y* in *N*], containing *x*'s estimate of its cost to all destinations, *y*, in *N*
- The distance vectors of each of its neighbors, that is, $\boldsymbol{D}_v = [D_v(y)$: *y* in *N*] for each neighbor *v* of *x*



$$D_x(y) = \min_v\{c(x,v) + D_v(y)\} \quad \text{for each node } y \text{ in } N$$

The Bellman-Ford
After traveling from *x* to *v*, if we then take the
least-cost path from *v* to *y*, the path cost will be $c(x,v) + d_v(y)$.
Since we must begin
by traveling to some neighbor *v*, the least cost from *x* to *y* is the
minimum of $c(x,v) + d_v(y)$ taken over all neighbors *v*.

Basic idea:

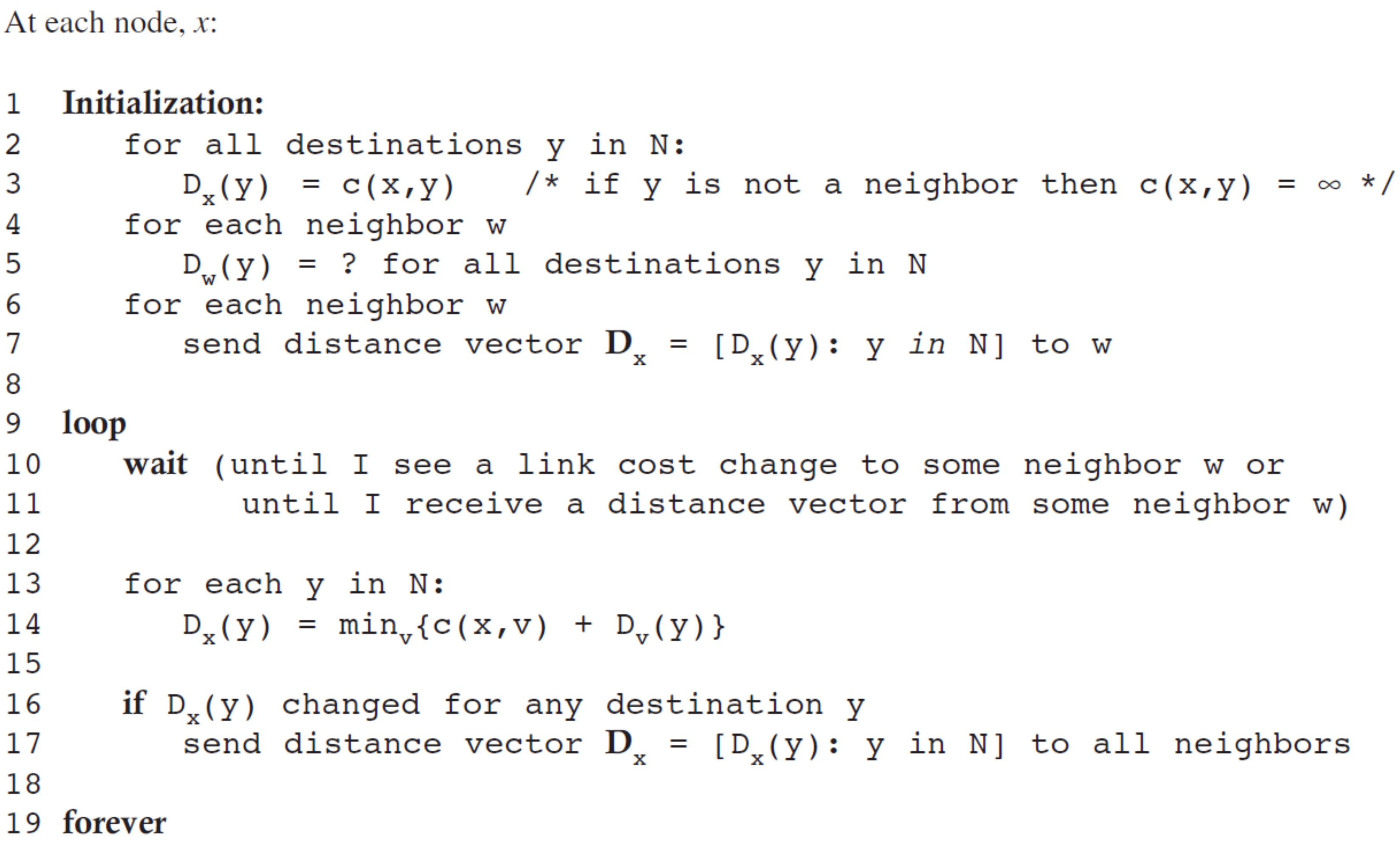
- ❒ From time-to-time, each node sends its own distance vector estimate to neighbors
- ❒ Asynchronous
- ❒ When a node x receives new DV estimate from neighbor, it updates its own DV using B-F equation:

$D\ \ (y) \leftarrow \min\{c(x,v) + D\ \ (y)\}$ for each node $y \in N$

- ❒ Under minor, natural conditions, the estimate

  x y) converge to the actual least cos

  $d_x(y)$



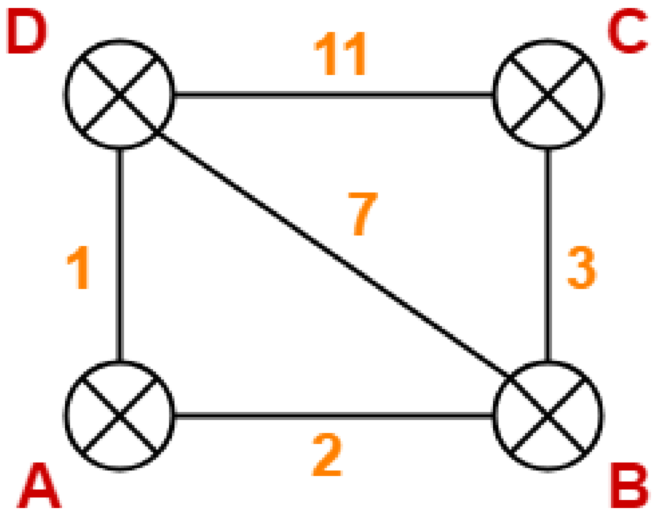
At each node, *x*:

```
Initialization:
   for all destinations y in N:
      $D_x(y)$ = c(x,y)    /* if y is not a neighbor then c(x,y) = ∞ */
   for each neighbor w
      $D_w(y)$ = ? for all destinations y in N
   for each neighbor w
      send distance vector $\mathbf{D}_x$ = [$D_x(y)$: y in N] to w

loop
   wait (until I see a link cost change to some neighbor w or
         until I receive a distance vector from some neighbor w)

   for each y in N:
      $D_x(y) = \min_v\{c(x,v) + D_v(y)\}$

   if $D_x(y)$ changed for any destination y
      send distance vector $\mathbf{D}_x$ = [$D_x(y)$: y in N] to all neighbors

forever
```

Distance Vector Routing Example-

Consider-

- There is a network consisting of 4 routers.
- The weights are mentioned on the edges.
- Weights could be distances or costs or delays.



Step-01:

Each router prepares its routing table using its local knowledge. Routing table prepared by each router is shown below-

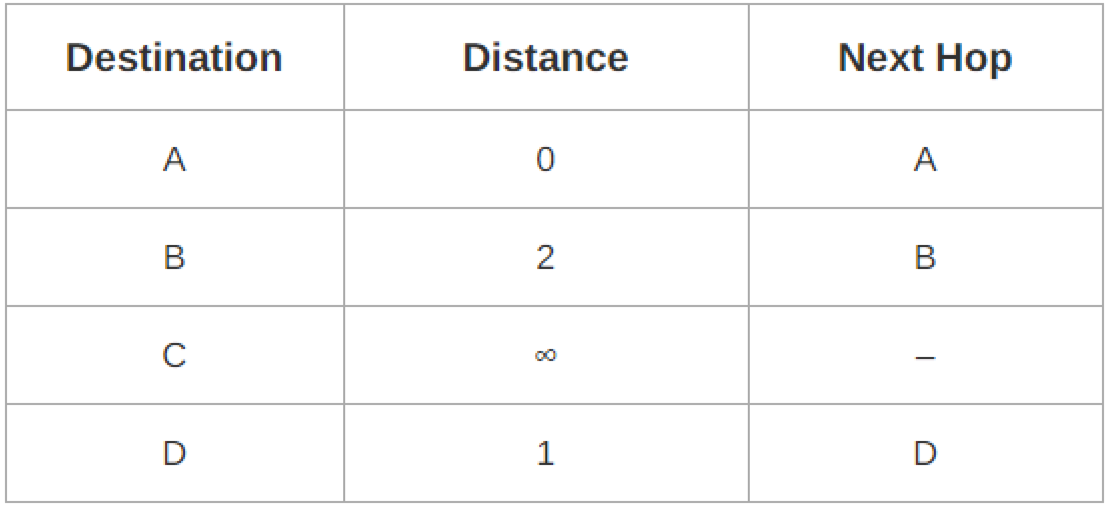
Step-02:

- Each router exchanges its distance vector obtained in Step-01 with its neighbors.
- After exchanging the distance vectors, each router prepares a new routing table.

Step-03:

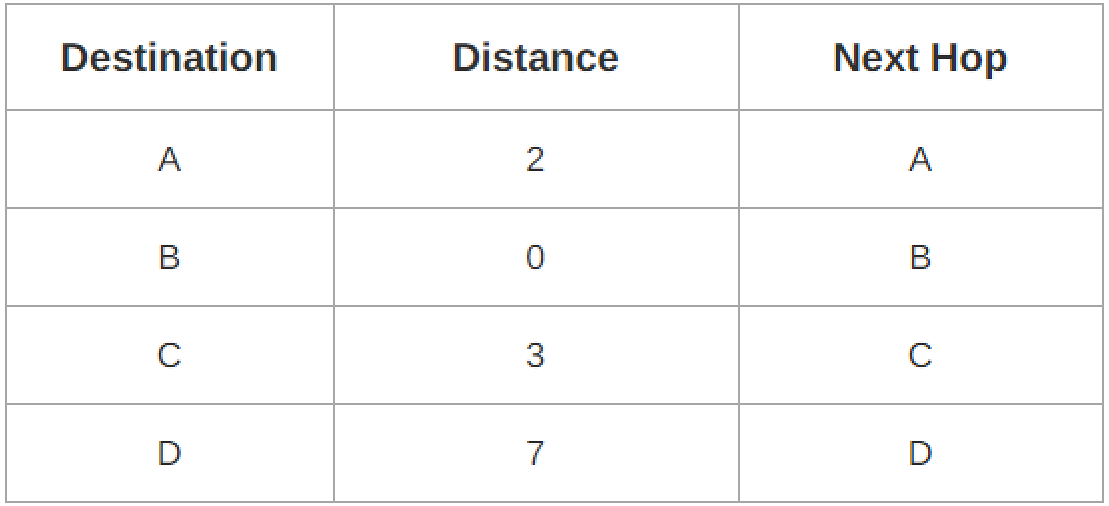
- Each router exchanges its distance vector obtained in Step-02 with its neighboring routers.
- After exchanging the distance vectors, each router prepares a new routing table.

Routing table and Distance Vector- Step 1



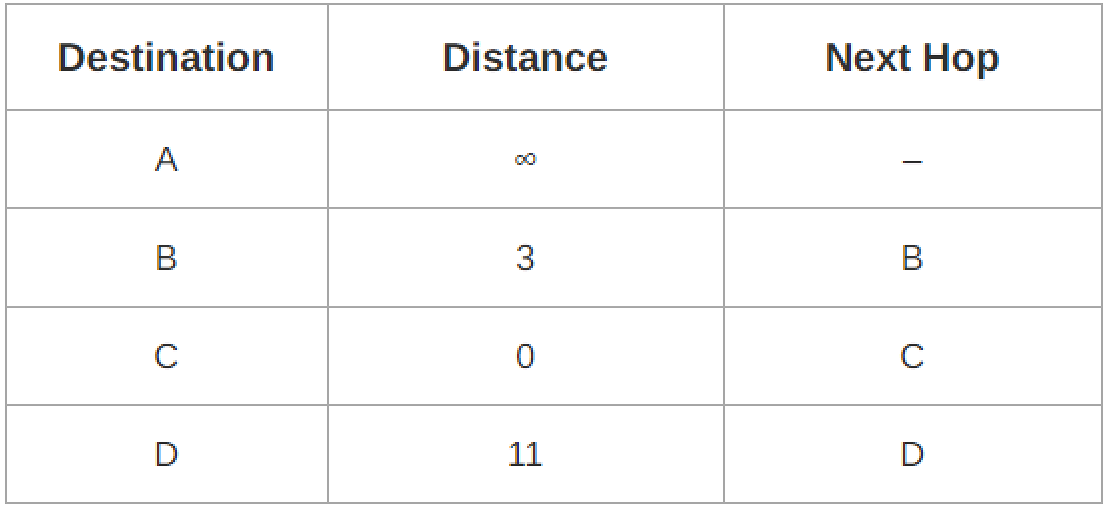
Destination	Distance	Next Hop
A	0	A
B	2	B
C	∞	–
D	1	D

Router A



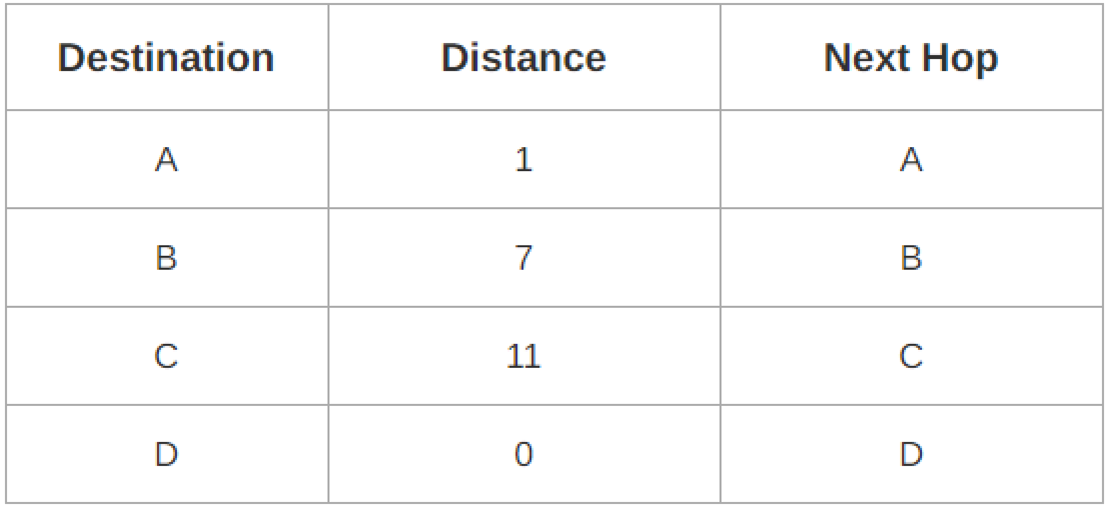
Destination	Distance	Next Hop
A	2	A
B	0	B
C	3	C
D	7	D

Router B



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

Router C



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

Router D

Routing table and Distance Vector- Step 2 (at Router A)

At Router A-

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

•Router A prepares a new routing table as-

Explanation For Destination B

Router A can reach the destination router B via its neighbor B or neighbor D.

•It chooses the path which gives the minimum cost.

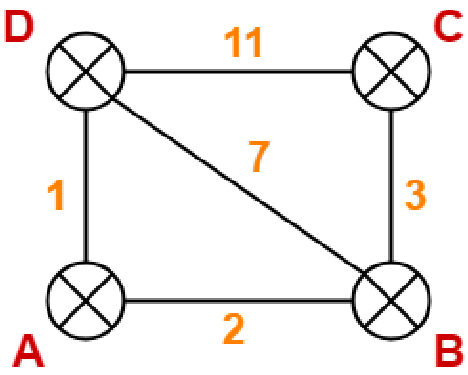
•Cost of reaching router B from router A via neighbor B = Cost (A→B) + Cost (B→B)= **2 + 0** = 2

•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.



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.

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

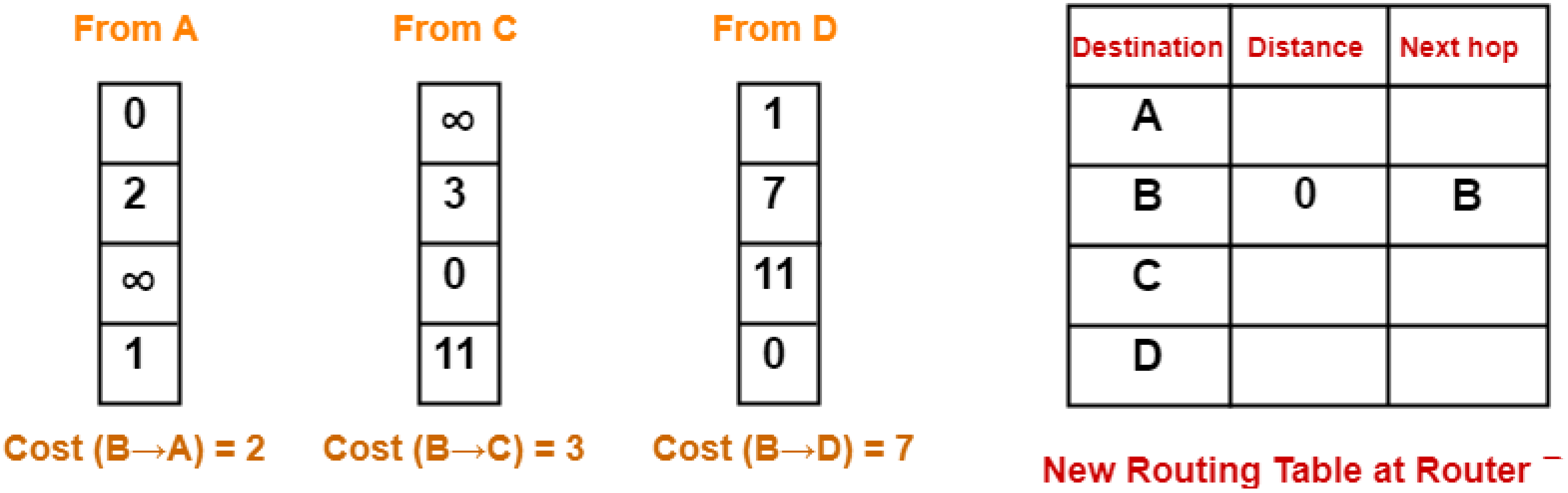
New Routing Table A

Routing table and Distance Vector- Step 2 (at Router B)

At Router B-

Router B receives distance vectors from its neighbors A, C and D.

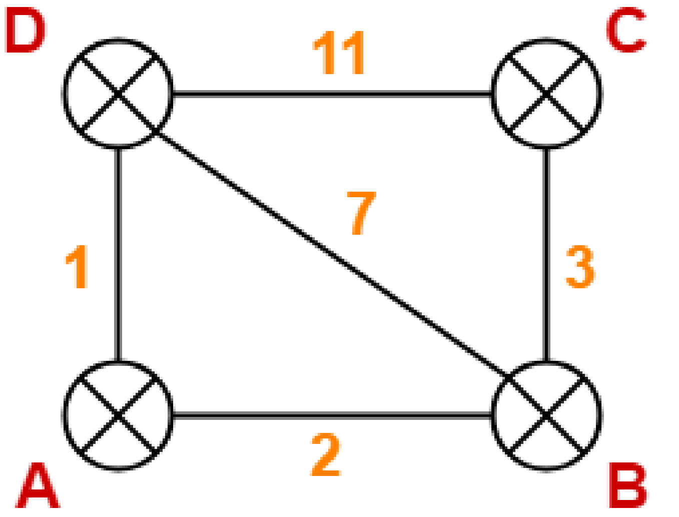
- Router B prepares a new routing table as-



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 –



New Routing Table at B

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

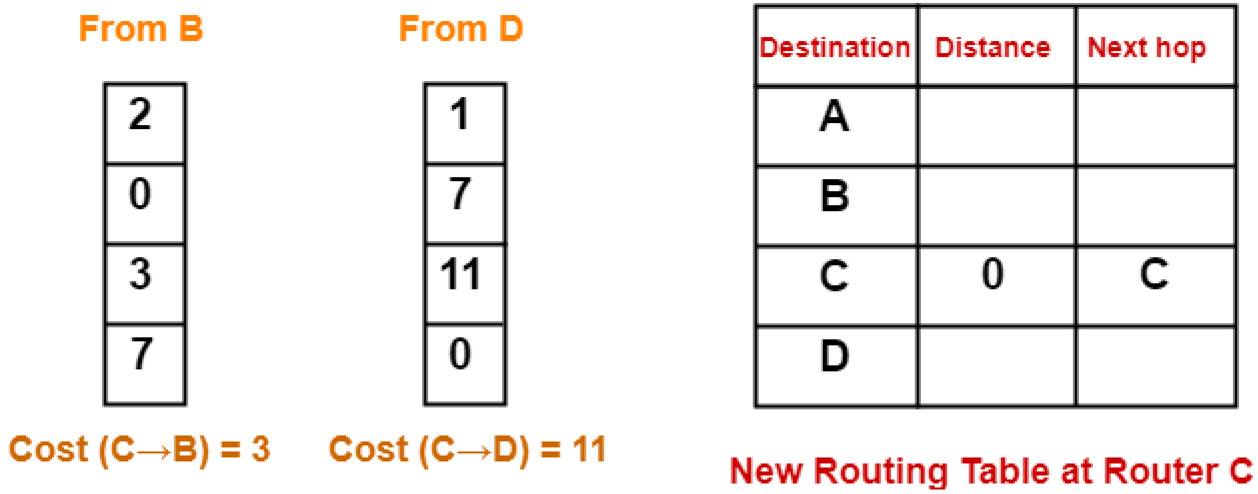
- Cost of reaching destination A from router B = min { 2+0 , 3+∞ , 7+1 } = 2 via A.
- Cost of reaching destination C from router B = min { 2+∞ , 3+0 , 7+11 } = 3 via C.
- Cost of reaching destination D from router B = min { 2+1 , 3+11 , 7+0 } = 3 via A.

Routing table and Distance Vector- Step 2 (at Router C)

At Router C-

Router C receives distance vectors from its neighbors B and D.

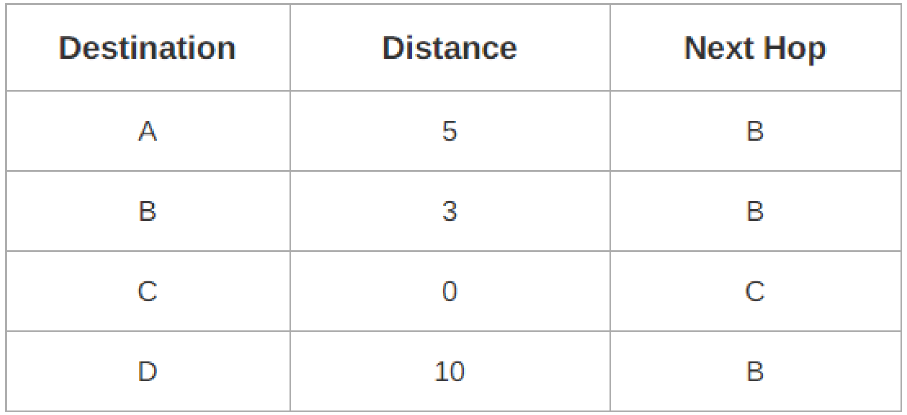
- Router C prepares a new routing table as-



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

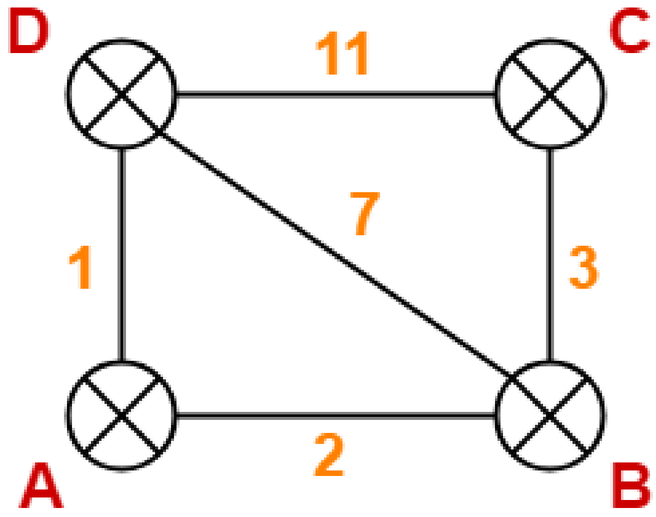
Destination	Distance	Next hop
A		
B		
C	0	C
D		

New Routing Table at Router C



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

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.

Routing table and Distance Vector- Step 2 (at Router D)

At Router D-

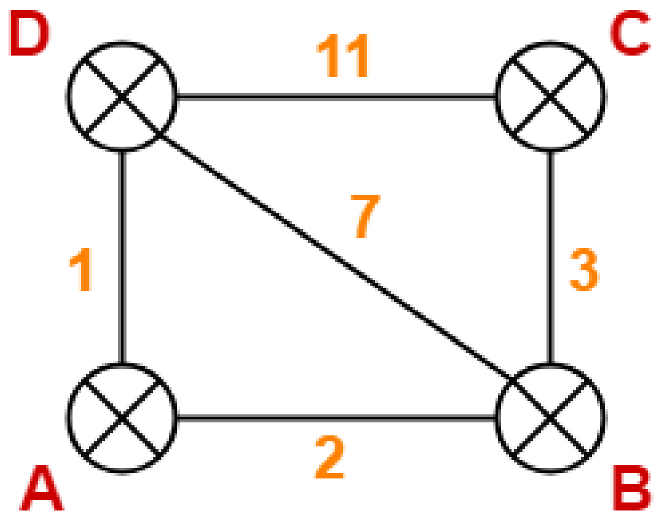
Router D receives distance vectors from its neighbors A, B and C.

- Router D prepares a new routing table as-

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

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

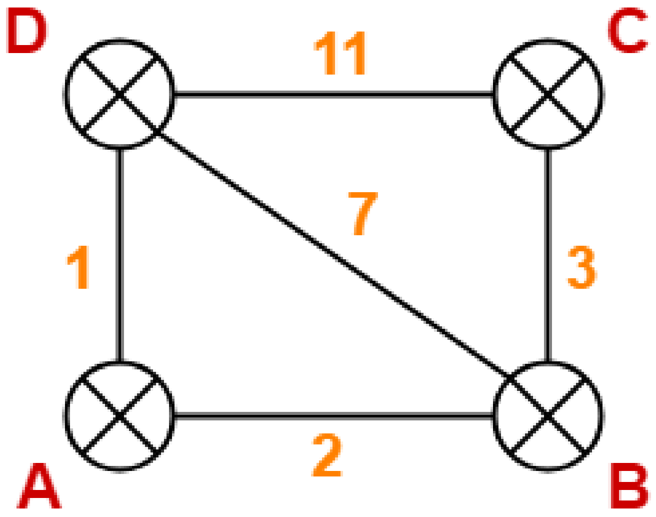
New Routing Table at Router D

Router A- Step3

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.



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

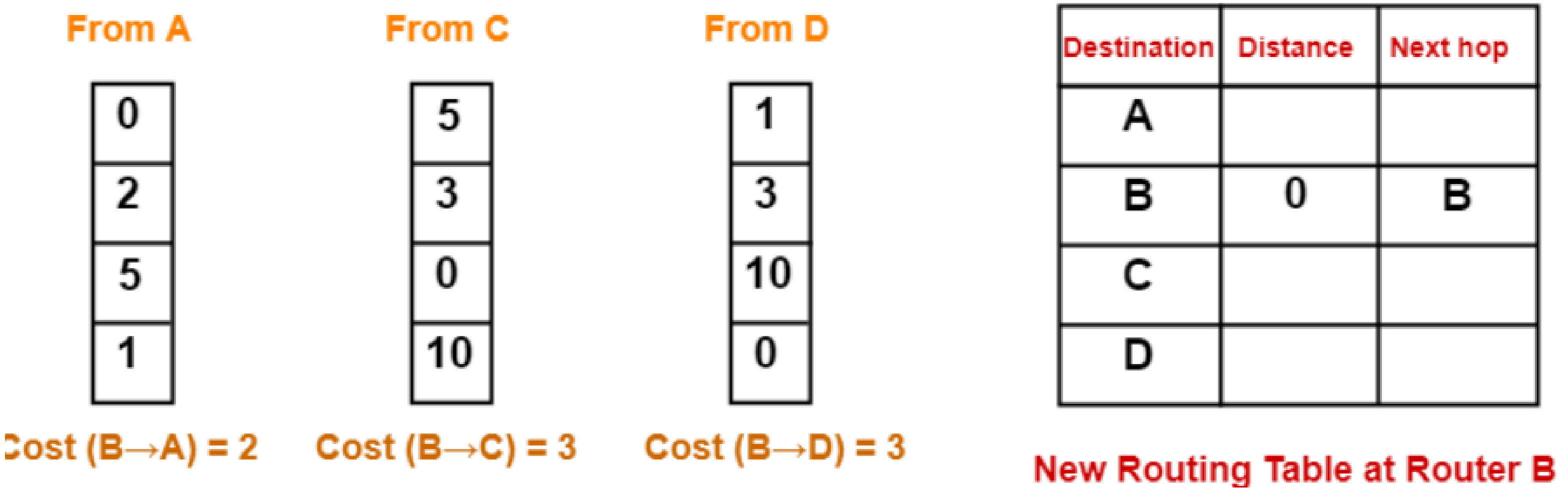
- Cost of reaching destination B from router A = min { 2+0 , 1+3 } = 2 via B.
- Cost of reaching destination C from router A = min { 2+3 , 1+10 } = 5 via B.
- Cost of reaching destination D from router A = min { 2+3 , 1+0 } = 1 via D.

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

New Routing Table A

Router B- Step3

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

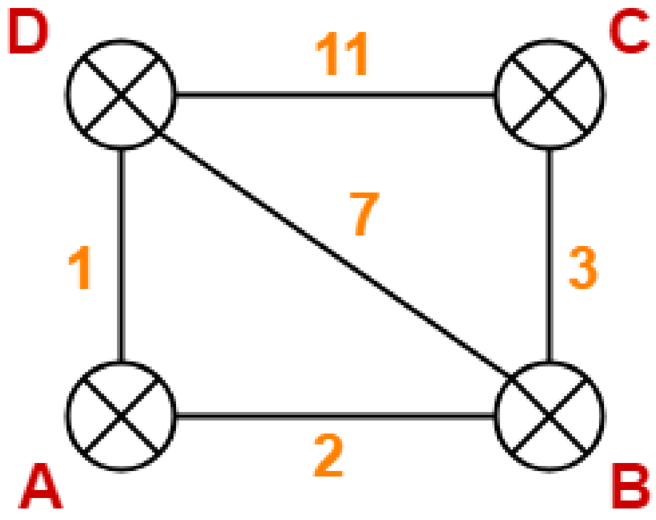


From A	From C	From D
0	5	1
2	3	3
5	0	10
1	10	0
Cost (B→A) = 2	Cost (B→C) = 3	Cost (B→D) = 3

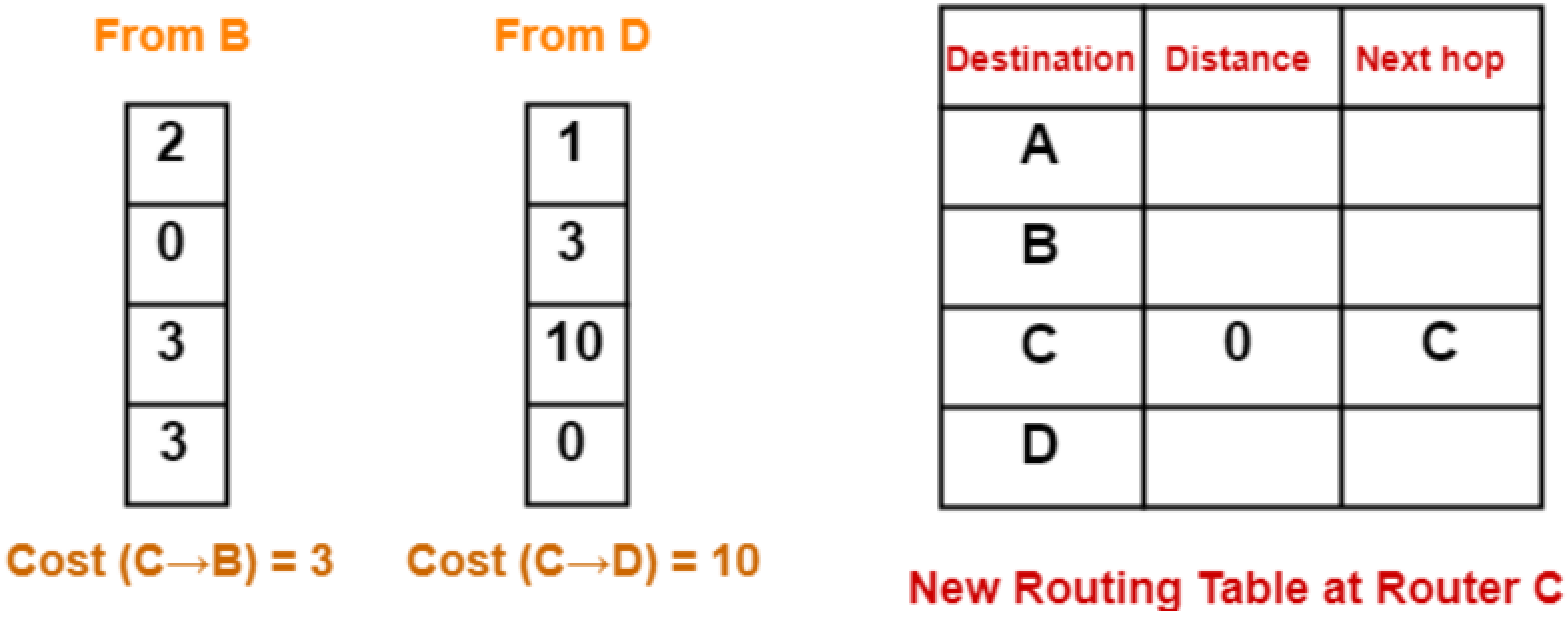
Destination	Distance	Next hop
A		
B	0	B
C		
D		

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.



Router C- Step3



From B

2
0
3
3

Cost (C→B) = 3

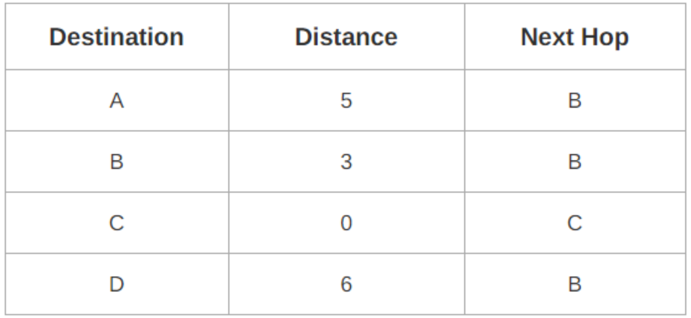
From D

1
3
10
0

Cost (C→D) = 10

Destination	Distance	Next hop
A		
B		
C	0	C
D		

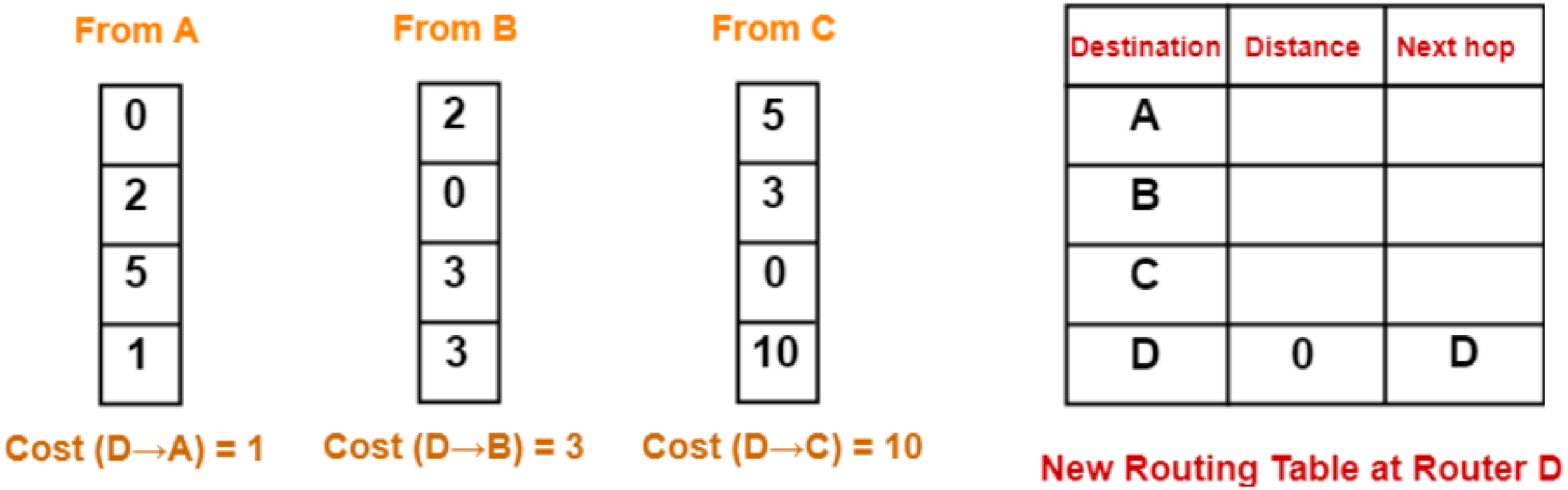
New Routing Table at Router C



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

- 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.

Router D- Step3



From A	From B	From C
0	2	5
2	0	3
5	3	0
1	3	10

Cost (D→A) = 1

Cost (D→B) = 3

Cost (D→C) = 10

Destination	Distance	Next hop
A		
B		
C		
D	0	D

New Routing Table at Router D

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

Identifying Unused Links-

After routing tables converge (becomes stable),

- Some of the links connecting the routers may never be used.
- In the above example, we can identify the unused links as-

We have-

- The value of next hop in the final routing table of router A suggests that only edges AB and AD are used.
- The value of next hop in the final routing table of router B suggests that only edges BA and BC are used.
- The value of next hop in the final routing table of router C suggests that only edge CB is used.
- The value of next hop in the final routing table of router D suggests that only edge DA is used.

Thus, edges BD and CD are never used.

$$D_x(y) = \min\{c(x,y) + D_y(y), c(x,z) + D_z(y)\}$$
$$= \min\{2+0 , 7+1\} = 2$$

$$D_x(z) = \min\{c(x,y) + D_y(z), c(x,z) + D_z(z)\}$$
$$= \min\{2+1 , 7+0\} = 3$$

node x table

cost to

from	x	y	z
x	0	2	7
y	∞	∞	∞
z	∞	∞	∞

cost to

from	x	y	z
x	0	2	3
y	2	0	1
z	7	1	0

node y table

cost to

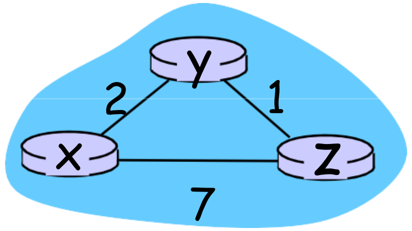
from	x	y	z
x	∞	∞	∞
y	2	0	1
z	∞	∞	∞

node z table

cost to

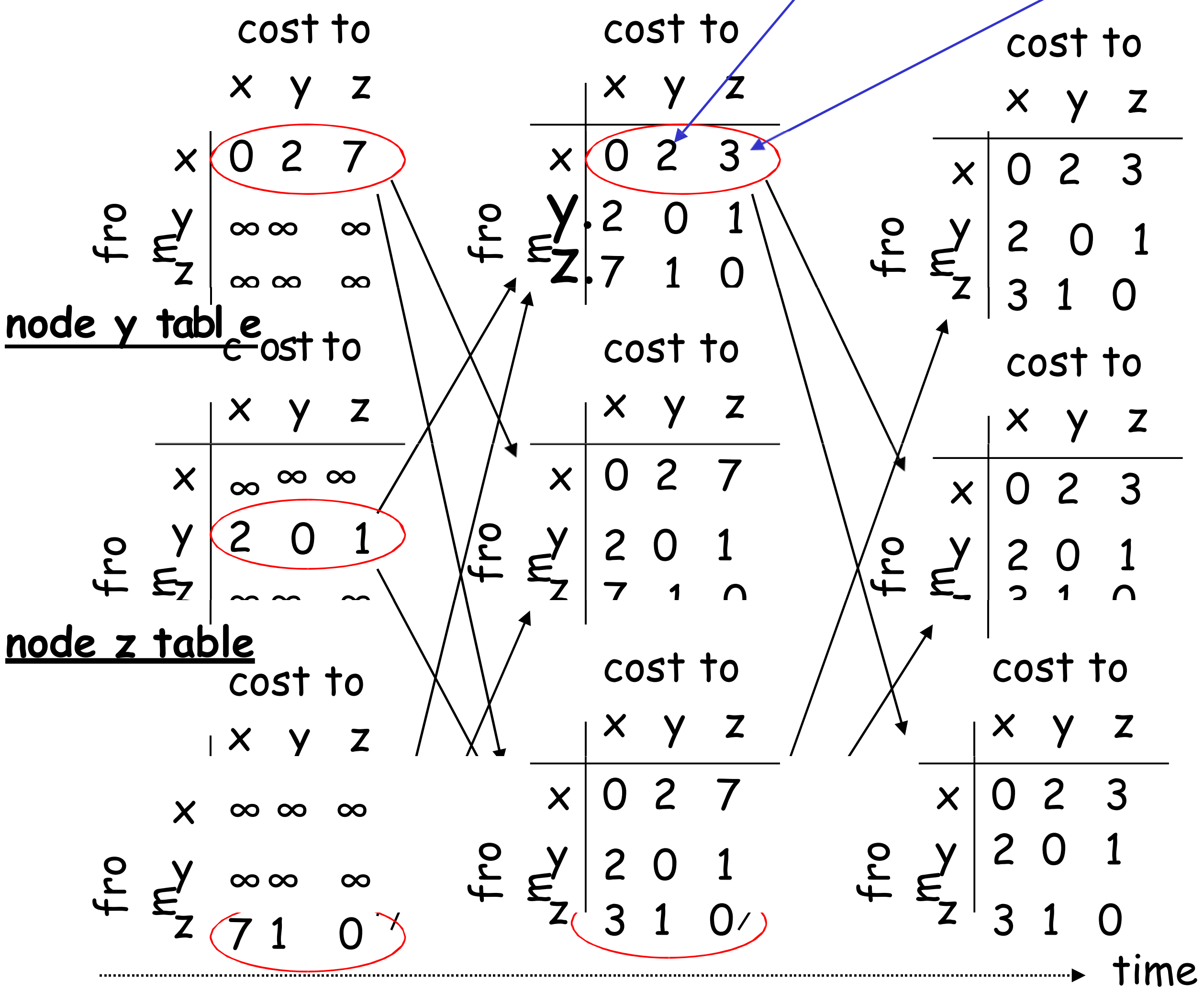
from	x	y	z
x	∞	∞	∞
y	∞	∞	∞
z	7	1	0

time



$$D_x(y) = \min\{c(x,y) + D_y(y), c(x,z) + D_z(y)\}$$
$$= \min\{2+0 , 7+1\} = 2$$

$$D_x(z) = \min\{c(x,y) + D_y(z), c(x,z) + D_z(z)\}$$
$$= \min\{2+1 , 7+0\} = 3$$



node x table

cost to

from	x	y	z
x	0	2	7
y	∞	∞	∞
z	∞	∞	∞

cost to

from	x	y	z
x	0	2	3
y	2	0	1
z	7	1	0

cost to

from	x	y	z
x	0	2	3
y	2	0	1
z	3	1	0

node y table

cost to

from	x	y	z
x	∞	∞	∞
y	2	0	1
z	∞	∞	∞

cost to

from	x	y	z
x	0	2	7
y	2	0	1
z	7	1	0

cost to

from	x	y	z
x	0	2	3
y	2	0	1
z	3	1	0

node z table

cost to

from	x	y	z
x	∞	∞	∞
y	∞	∞	∞
z	7	1	0

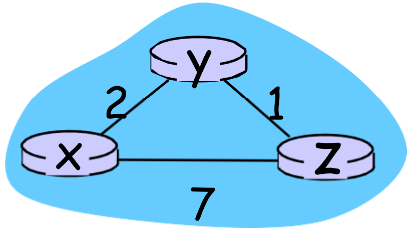
cost to

from	x	y	z
x	0	2	7
y	2	0	1
z	3	1	0

cost to

from	x	y	z
x	0	2	3
y	2	0	1
z	3	1	0

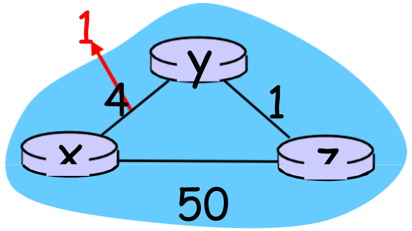
time



Distance Vector: link cost changes

Link cost changes:

- node detects local link cost change
- updates routing info, recalculates distance vector
- if DV changes, notify neighbors



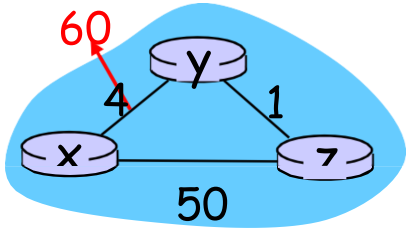
“good news travels fast”

At time t_0, *y* detects the link-cost change, updates its DV, and informs its neighbors.

At time $_{1}$, receives the update from and updates its table. It computes a new least cost to *x* and sends its neighbors its DV

At time , receives 's update and updates its distance table. *y*'s least costs do not change and hence *y* does *not* send any message to *z*.

Distance Vector: link cost changes



Link cost changes:

- good news travels fast
- bad news travels slow - "count to infinity" problem!
- 44 iterations before algorithm stabilizes: see text

Poisoned reverse:

- If Z routes through Y to get to X :
  - Z tells Y its (Z's) distance to X is infinite (so Y won't route to X via Z)
- will this completely solve count to infinity problem?

Chapter 4: Network Layer

- 4. 1 Introduction
- 4.2 Virtual circuit and datagram networks
- 4.3 What's inside a router
- 4.4 IP: Internet Protocol
  - Datagram format
  - IPv4 addressing
  - ICMP
  - IPv6
- 4.5 Routing algorithms
  - Link state
  - Distance Vector
  - Hierarchical routing
- 4.6 Routing in the Internet
  - RIP
  - OSPF
  - BGP
- 4.7 Broadcast and multicast routing

Hierarchical Routing

Scale. As the number of routers becomes large, the overhead involved in computing, storing, and communicating routing information LS updates or least-cost path changes) becomes prohibitive. Public Internet consists of hundreds of millions of hosts. Storing routing information at each of these hosts would clearly require enormous amounts of memory. The overhead required to broadcast LS updates among all of the routers in the public Internet would leave no bandwidth left for sending data packets!

A distance-vector algorithm that iterated among such a large number of routers would surely never converge. Clearly, something must be done to reduce the complexity of route computation in networks as large as the public Internet.

Administrative autonomy. Although researchers tend to ignore issues such as a company's desire to run its routers as it pleases (for example, to run whatever routing algorithm it chooses) or to hide aspects of its network's internal organization from the outside, these are important considerations. Ideally, an organization should be able to run and administer its network as it wishes, while still being able to connect its network to other outside networks

Hierarchical Routing

Our routing study thus far - idealization

- ❒ all routers identical
- ❒ network "flat"

… *not* true in practice

scale: with 200 million destinations:

- ❒ can't store all dest's in routing tables!
- ❒ routing table exchange would swamp links!

administrative autonomy

- ❒ internet = network of networks
- ❒ each network admin may want to control routing in its own network

Hierarchical Routing

- aggregate routers into regions, “autonomous systems” (AS)
- Each AS consist of a group of routers that are typically under the same administrative control (e.g., operated by the same ISP or belonging to the same company network)
- Routers within the same AS all run the same routing algorithm (for example, an LS or DV algorithm) and have information about each other—exactly as was the case in our idealized model in the preceding section
  - "

The routing algorithm running within an autonomous system is called an intraautonomous system routing protocol.

It will be necessary, of course, to connect ASs to each other,

one or more of the routers in an AS will have the added task of being responsible for forwarding packets to destinations outside the AS; these routers are called gateway routers.

Hierarchical Routing

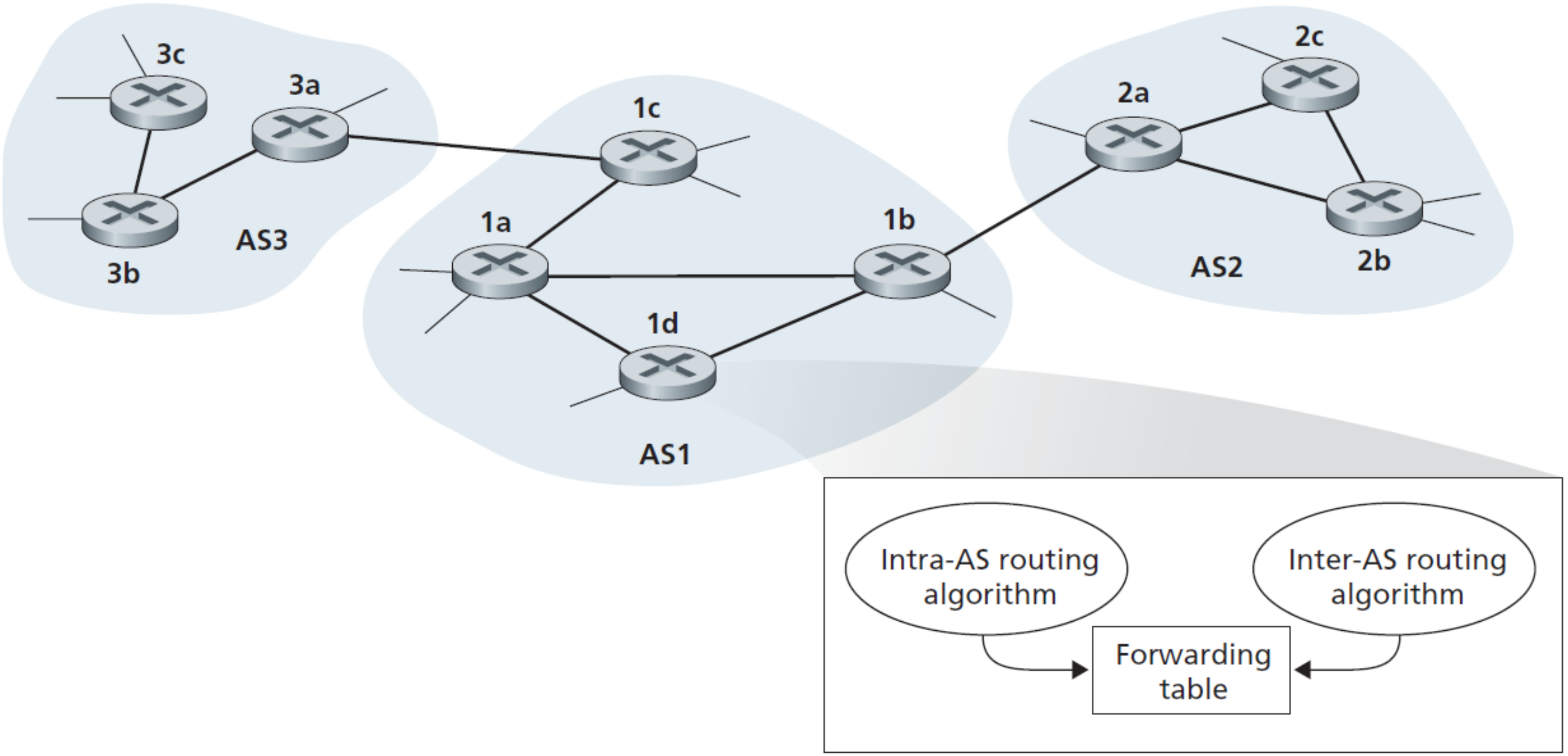


Figure 4.32 ♦ An example of interconnected autonomous systems

- Simple example with three ASs: AS1, AS2, and AS3.
- In this figure, the heavy lines represent direct link connections between pairs of routers. The thinner lines hanging from the routers represent subnets that are directly connected to the routers.
- AS1 has four routers—1a, 1b, 1c, and 1d—which run the intra-AS routing protocol used within AS1.

Intra-AS routing protocols running in AS1, AS2, and AS3 need not be the same. Also note that the routers 1b, 1c, 2a, and 3a are all gateway routers.

Why different Intra- and Inter-AS routing ?

Policy:

- ❐ Inter-AS: admin wants control over how its traffic routed, who routes through its net.
- ❐ Intra-AS: single admin, so no policy decisions needed

Scale:

- ❐ hierarchical routing saves table size, reduced update traffic

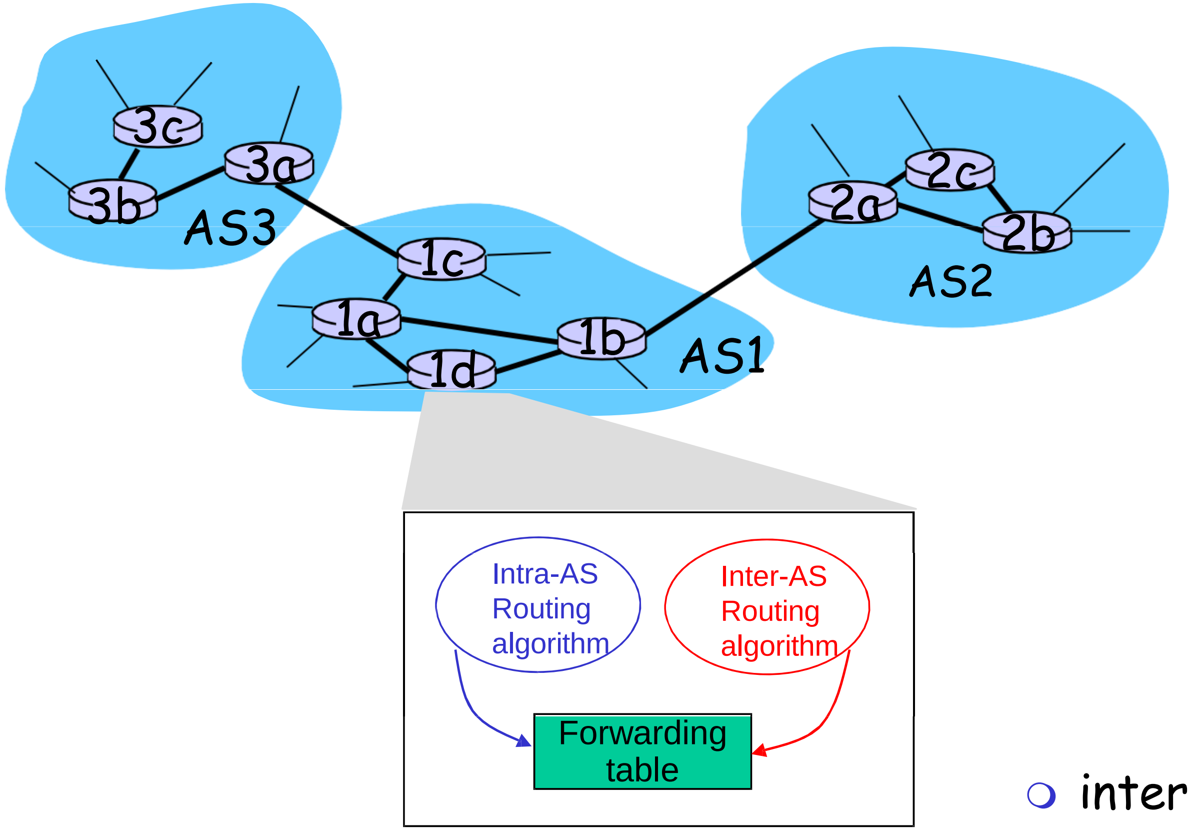
Performance:

- ❐ Intra-AS: can focus on performance
- ❐ Inter-AS: policy may dominate over performance

How does a router, within some AS, know how to route a packet to a destination that is outside the AS?

If the AS has only one gateway router that connects to only one other AS. In this case, because the AS's intra-AS routing algorithm has determined the least-cost path from each internal router to the gateway router, each internal router knows how it should forward the packet. The gateway router, upon receiving the packet, forwards the packet on the one link that leads outside the AS. The AS on the other side of the link then takes over the responsibility of routing the packet to its ultimate destination

Interconnected ASes



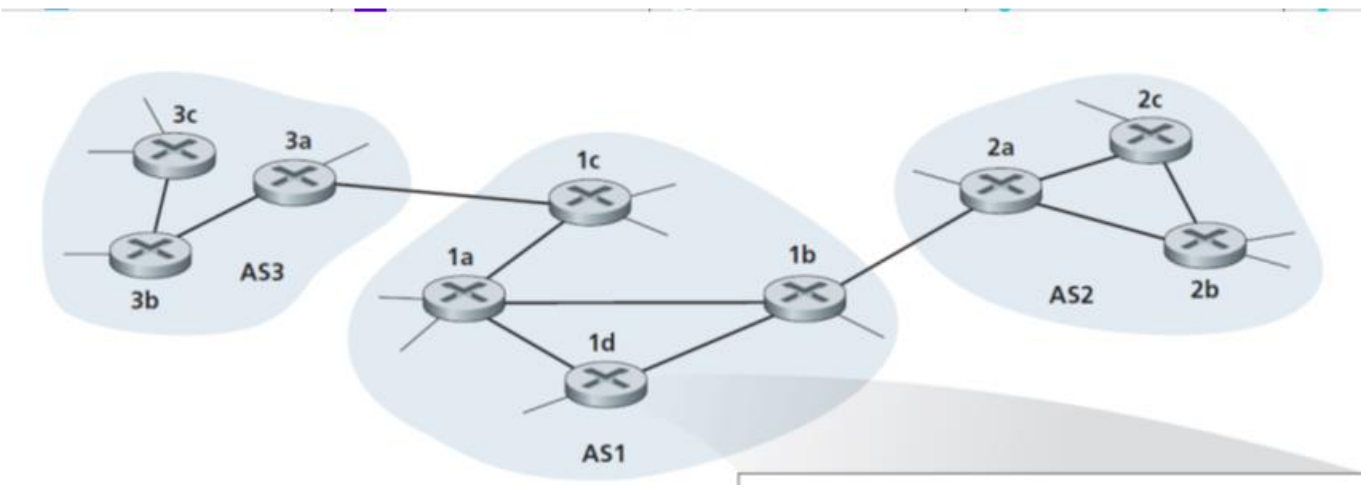
As an example, suppose router 2b in Figure receives a packet whose destination is outside of AS2. Router 2b will then forward the packet to either router 2a or 2c, as specified by router 2b's forwarding table, which was configured by AS2's intra-AS routing protocol.

The packet will eventually arrive to the gateway router 2a, which will forward the packet to 1b. Once the packet has left 2a, AS2's job is done with this one packet.

- inter-AS & intra-As sets entries for external dests

What if the source AS has two or more links (through two or more gateway routers) that lead outside the AS?

problem of knowing where to forward the packet becomes significantly more challenging.



This task of obtaining reachability information from neighboring ASs and propagating the reachability information to all routers internal to the AS—are handled by the
inter-AS routing protocol

The two communicating ASs must run the same inter-AS routing protocol.

For example, consider a router in AS1 and suppose it receives a packet whose destination is outside the AS. The router should clearly forward the packet to one of its two gateway routers, 1b or 1c, but which one?

To solve this problem, AS1 needs

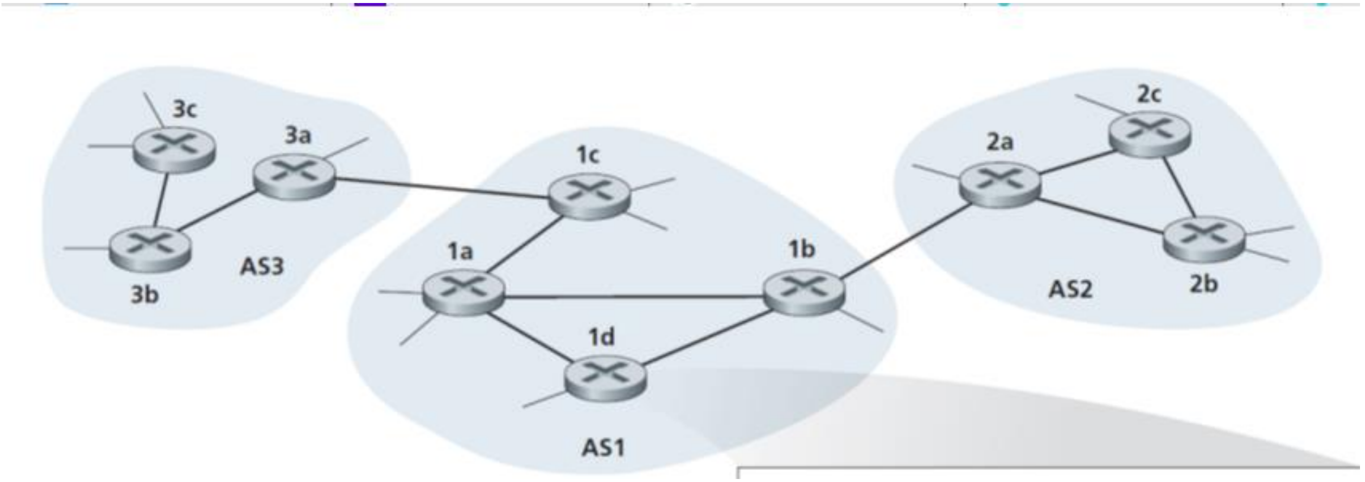
(1) to learn which destinations are reachable via AS2 and which destinations are reachable via AS3, and

(2) To propagate this reachability information to all the routers within AS1, so that each router can configure its forwarding table to handle external-AS destinations

inter-AS routing protocol and Forwarding table

Subnet x reachable only from AS3

In the Internet all ASs run the same inter-AS routing protocol, called BGP4, which is discussed in the next section. As shown in Figure 4.32, each router receives information from an intra-AS routing protocol and an inter-AS routing protocol, and uses the information from both protocols to configure its forwarding table.



As an example, consider a subnet *x* and suppose that AS1 learns from the inter-AS routing protocol that subnet *x* is reachable from AS3 but is *not* reachable from AS2.

AS1 then propagates this information to all of its routers. When router 1d learns that subnet *x* is reachable from AS3, and hence from gateway 1c,

it then determines, from the information provided by the intra-AS routing protocol, the router interface that is on the least-cost path from router 1d to gateway router 1c. Say this is interface *I*. The router 1d can then put the entry (*x, I*) into its forwarding table

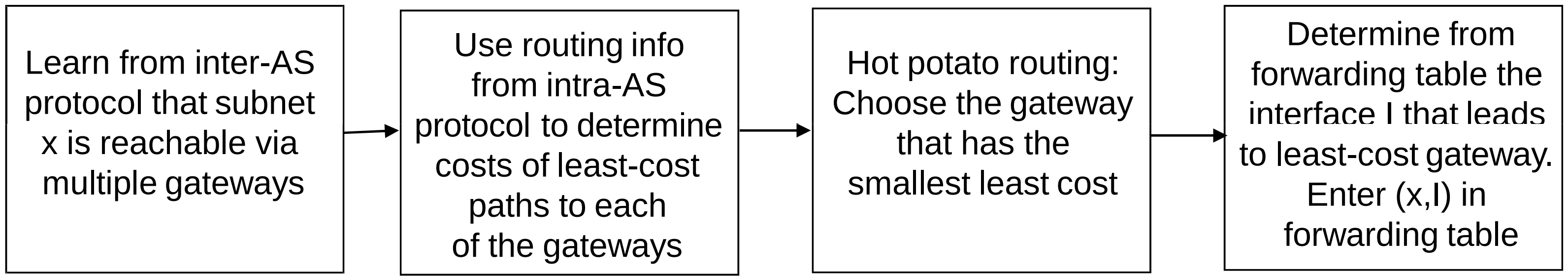
Example: Choosing among multiple ASes

Subnet x reachable from AS3 and AS2

now suppose AS1 learns from inter-AS protocol that subnet x is reachable from AS3 *and* from AS2.

- to configure forwarding table, router 1d must determine towards which gateway it should forward packets for dest x.
  - this is also job of inter-AS routing protocol!
- hot potato routing: send packet towards closest of two routers.

When an AS learns about a destination from a neighboring AS, the AS can advertise this routing information to some of its other neighboring ASs. For example, suppose AS1 learns from AS2 that subnet *x* is reachable via AS2. AS1 could then tell AS3 that *x* is reachable via AS1. In this manner, if AS3 needs to route a packet destined to *x*, AS3 would forward the packet to AS1, which would in turn forward the packet to AS2.



Learn from inter-AS protocol that subnet x is reachable via multiple gateways

Use routing info from intra-AS protocol to determine costs of least-cost paths to each of the gateways

Hot potato routing: Choose the gateway that has the smallest least cost

Determine from forwarding table the interface I that leads to least-cost gateway. Enter (x,I) in forwarding table

Internet consists of a hierarchy of interconnected ISPs. So what is the relationship between ISPs and ASs?

The routers in an ISP, and the links that interconnect them, constitute a single AS. Although this is often the case, many ISPs partition their network into multiple ASs. For example, some tier-1 ISPs use one AS for their entire network; others break up their ISP into tens of interconnected ASs

Which are the
Inter AS protocols?
Intra AS protocols?

Chapter 4: Network Layer

- ❒ 4. 1 Introduction
- ❒ 4.2 Virtual circuit and datagram networks
- ❒ 4.3 What's inside a router
- ❒ 4.4 IP: Internet Protocol
  - ❍ Datagram format
  - ❍ IPv4 addressing
  - ❍ ICMP
  - ❍ IPv6
- ❒ 4.5 Routing algorithms
  - ❍ Link state
  - ❍ Distance Vector
  - ❍ Hierarchical routing
- ❒ 4.6 Routing in the Internet
  - ❍ RIP
  - ❍ OSPF
  - ❍ BGP
- ❒ 4.7 Broadcast and multicast routing

Intra-AS Routing

- also known as Interior Gateway Protocols (IGP)
- most common Intra-AS routing protocols:
  - RIP: Routing Information Protocol
  - OSPF: Open Shortest Path First
  - IGRP: Interior Gateway Routing Protocol (Cisco proprietary)

Internet inter-AS routing:

- BGP (Border Gateway Protocol):

Thank You