Unicast Routing

INTRODUCTION

Unicast routing in the Internet, with a large number of routers and a huge number of hosts, can be done only by using hierarchical routing: routing in several steps using different routing algorithms.

General Idea

Routing a packet from its source to its destination means routing the packet from a source router to a destination router.

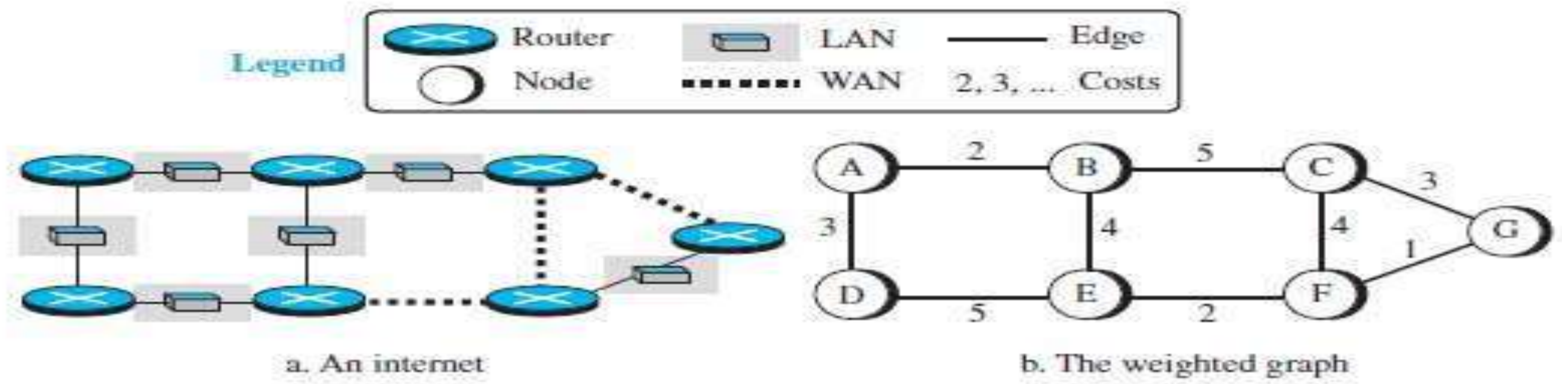
An Internet as a Graph

To find the best route, an internet can be modeled as a graph. A graph in computer science is a set of nodes and edges (lines) that connect the nodes. An internet is, in fact, modeled as a weighted graph, in which each edge is associated with a cost. In routing, however, the cost of an edge has a different interpretation in different routing protocols

Least-Cost Routing

The source router chooses a route to the destination router in such a way that the total cost for the route is the least cost among all possible routes.

Figure 20.1 *An internet and its graphical representation*

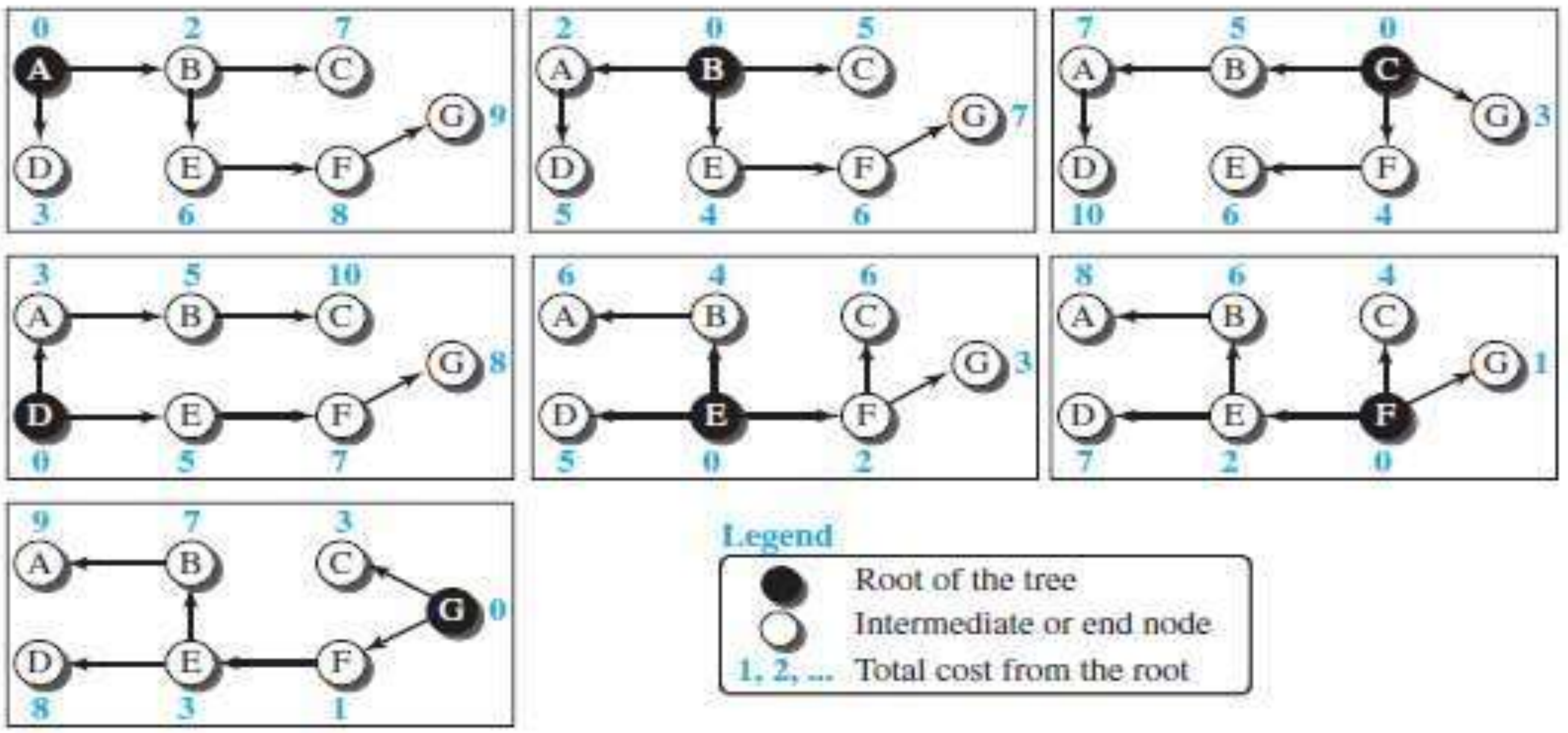


Least-Cost Trees

If there are N routers in an internet, there are (N − 1) least-cost paths from each router to any other router. This means we need N × (N − 1) least-cost paths for the whole internet.
A better way to see all of these paths is to combine them in a least-cost tree. A least-cost tree is a tree with the source router as the root that spans the whole graph (visits all other nodes) and in which the path between the root and any other node is the shortest. In this way, we can have only one shortest-path tree for each node

Continue..

Figure 20.2 *Least-cost trees for nodes in the internet of Figure 20.1*


A
B
C
D
E
F
G
Legend
● Root of the tree
○ Intermediate or end node
1, 2, ... Total cost from the root

ROUTING ALGORITHMS

Several routing algorithms have been designed in the past. The differences between these methods are in the way they interpret the least cost and the way they create the least-cost tree for each node.

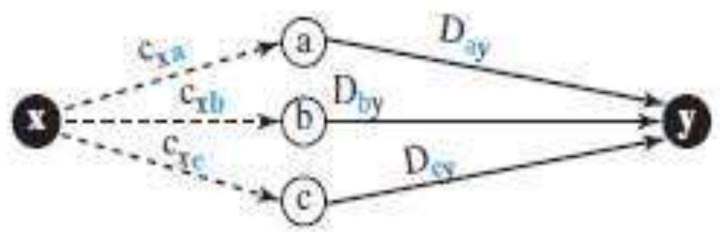
Distance-Vector Routing

In distance-vector (DV) routing, the first thing each node creates is its own least-cost tree with the rudimentary information it has about its immediate neighbors.

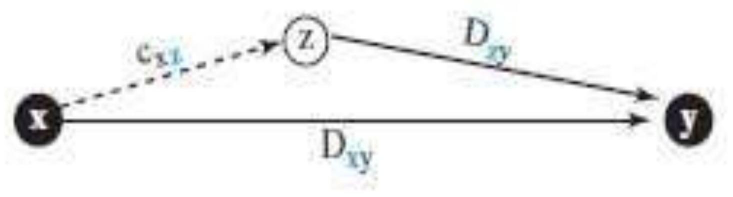
The incomplete trees are exchanged between immediate neighbors to make the trees more and more complete and to represent the whole internet.

Bellman-Ford Equation

Figure 20.3 *Graphical idea behind Bellman-Ford equation*



a. General case with three intermediate nodes

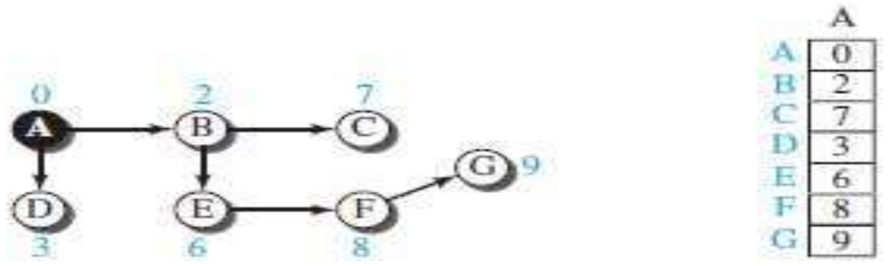


b. Updating a path with a new route

Distance Vectors

A least-cost tree is a combination of least-cost paths from the root of the tree to all destinations. These paths are graphically glued together to form the tree. Distance-vector routing unglues these paths and creates a distance vector, a one-dimensional array to represent the tree.

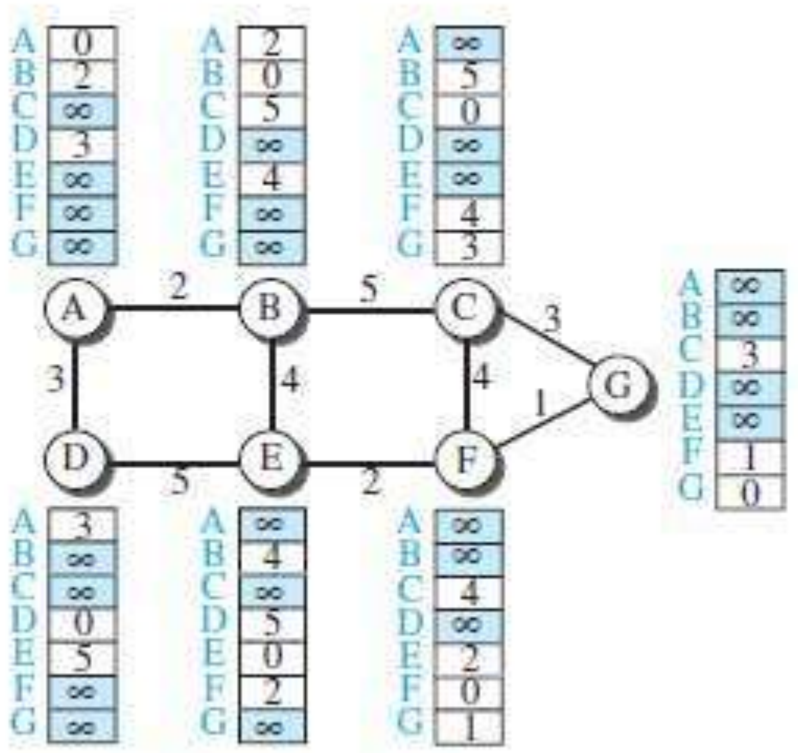
Figure 20.4 *The distance vector corresponding to a tree*



a. Tree for node A   b. Distance vector for node A

Continue..

Figure 20.5 *The first distance vector for an internet*


A B C D E F G
A 0, B 2, C ∞, D 3, E ∞, F ∞, G ∞
A 2, B 0, C 5, D ∞, E 4, F ∞, G ∞
A ∞, B 5, C 0, D ∞, E ∞, F 4, G 3
A ∞, B ∞, C 3, D ∞, E ∞, F 1, G 0
A 3, B ∞, C ∞, D 0, E 5, F ∞, G ∞
A ∞, B 4, C ∞, D 5, E 0, F 2, G ∞
A ∞, B ∞, C 4, D ∞, E 2, F 0, G 1
2
5
3
3
4
4
1
5
2

Department of Computer science and Information Technology

Continue..

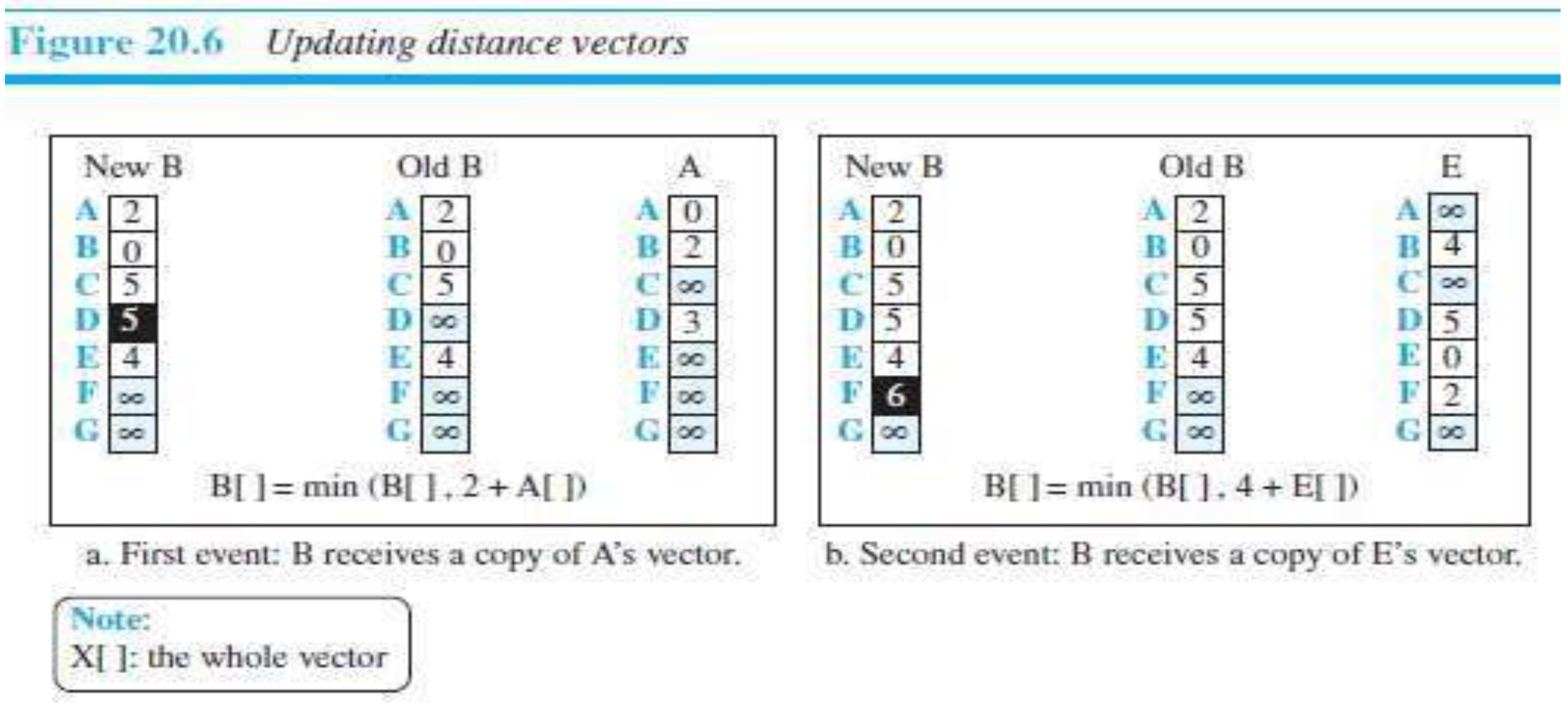


Figure 20.6 *Updating distance vectors*

	New B	Old B	A
A	2	2	0
B	0	0	2
C	5	5	∞
D	5	∞	3
E	4	4	∞
F	∞	∞	∞
G	∞	∞	∞

B[ ] = min (B[ ], 2 + A[ ])

a. First event: B receives a copy of A's vector.

	New B	Old B	E
A	2	2	∞
B	0	0	4
C	5	5	∞
D	5	5	5
E	4	4	0
F	6	∞	2
G	∞	∞	∞

B[ ] = min (B[ ], 4 + E[ ])

b. Second event: B receives a copy of E's vector.

Note:
X[ ]: the whole vector

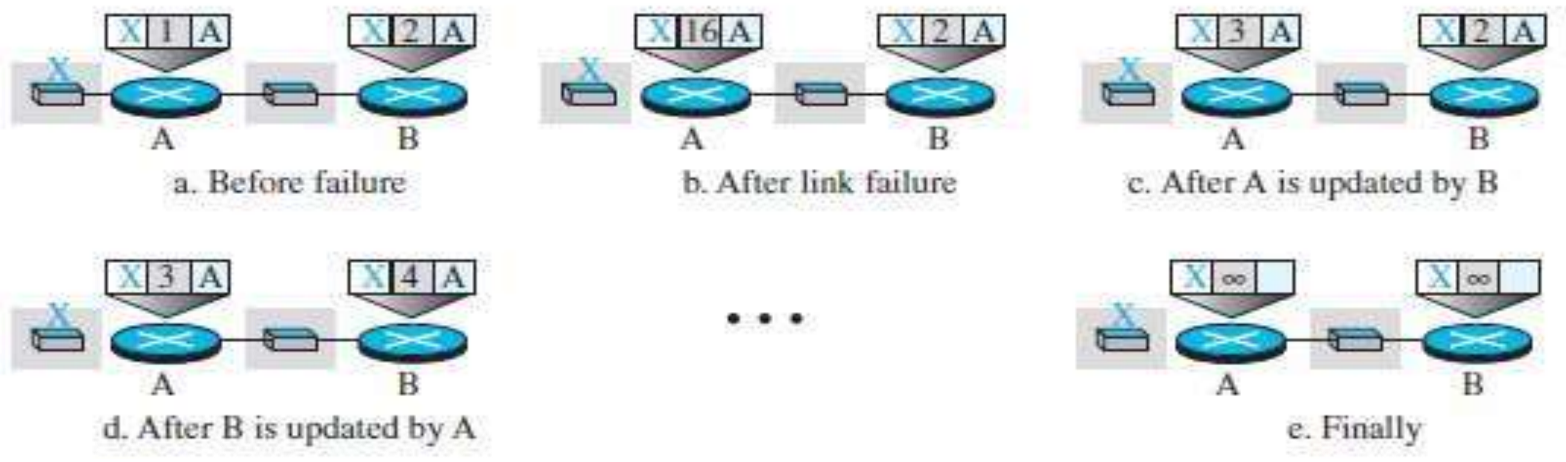
Count to Infinity

For a routing protocol to work properly, if a link is broken (cost becomes infinity), every other router should be aware of it immediately, but in distance-vector routing, this takes some time. The problem is referred to as count to infinity.

Two-Node Loop

One example of count to infinity is the two-node loop problem.

Figure 20.7 *Two-node instability*



Split Horizon

One solution to instability is called split horizon. In this strategy, instead of flooding the table through each interface, each node sends only part of its table through each interface. E.g. node B thinks that the optimum route to reach X is via A, it does not need to advertise this piece of information to A; the information has come from A (A already knows).

Link-State Routing

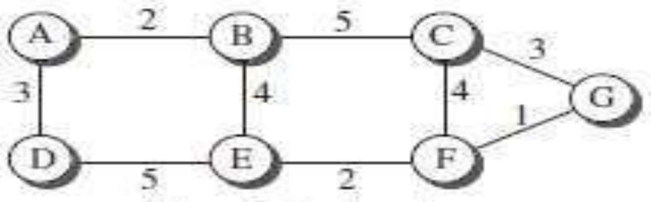
link-state (LS) routing method uses the term link-state to define the characteristic of a link (an edge) that represents a network in the internet.

Links with lower costs are preferred to links with higher costs; if the cost of a link is infinity, it means that the link does not exist or has been broken.

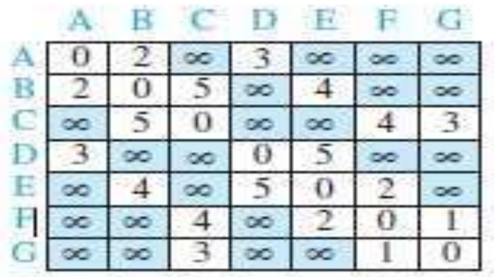
Link-State Database (LSDB)

To create a least-cost tree with this method, each node needs to have a complete map of the network, which means it needs to know the state of each link. The collection of states for all links is called the link-state database (LSDB). There is only one LSDB for the whole internet.

Figure 20.8 *Example of a link-state database*



a. The weighted graph



	A	B	C	D	E	F	G
A	0	2	∞	3	∞	∞	∞
B	2	0	5	∞	4	∞	∞
C	∞	5	0	∞	∞	4	3
D	3	∞	∞	0	5	∞	∞
E	∞	4	∞	5	0	2	∞
F	∞	∞	4	∞	2	0	1
G	∞	∞	3	∞	∞	1	0

b. Link state database

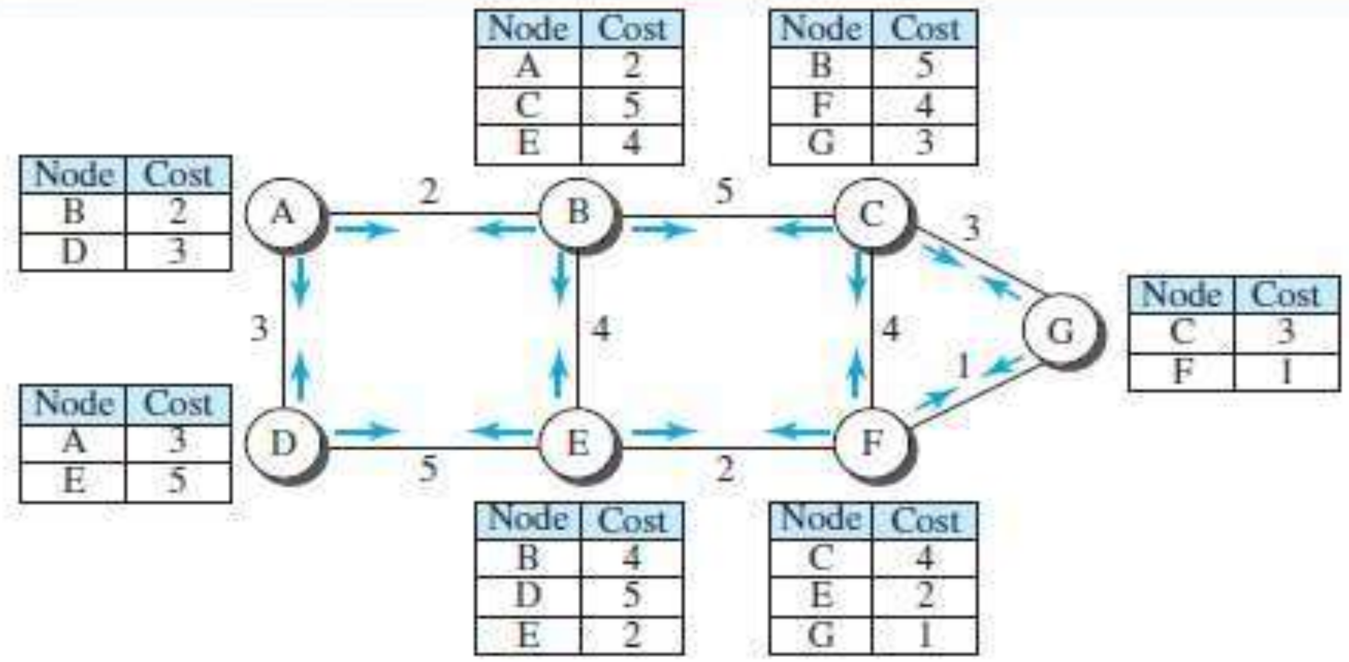
Continue...

How each node can create this LSDB that contains information about the whole internet can be done by a process called flooding.

Each node can send some greeting messages to all its immediate neighbors, to collect two pieces of information: the identity of the node and the cost of the link. The combination of these two pieces of information is called the LS packet (LSP). After receiving all new LSPs, each node creates the comprehensive LSDB.

Continue…

Figure 20.9 *LSPs created and sent out by each node to build LSDB*



A:

Node	Cost
B	2
D	3

B:

Node	Cost
A	2
C	5
E	4

C:

Node	Cost
B	5
F	4
G	3

D:

Node	Cost
A	3
E	5

E:

Node	Cost
B	4
D	5
E	2

F:

Node	Cost
C	4
E	2
G	1

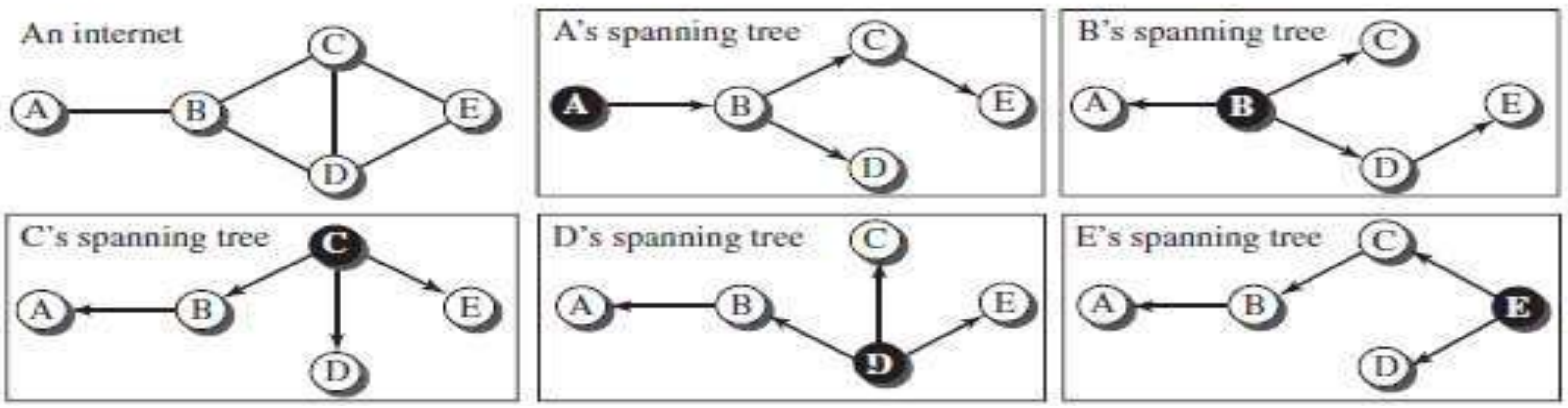
G:

Node	Cost
C	3
F	1

Path-Vector Routing

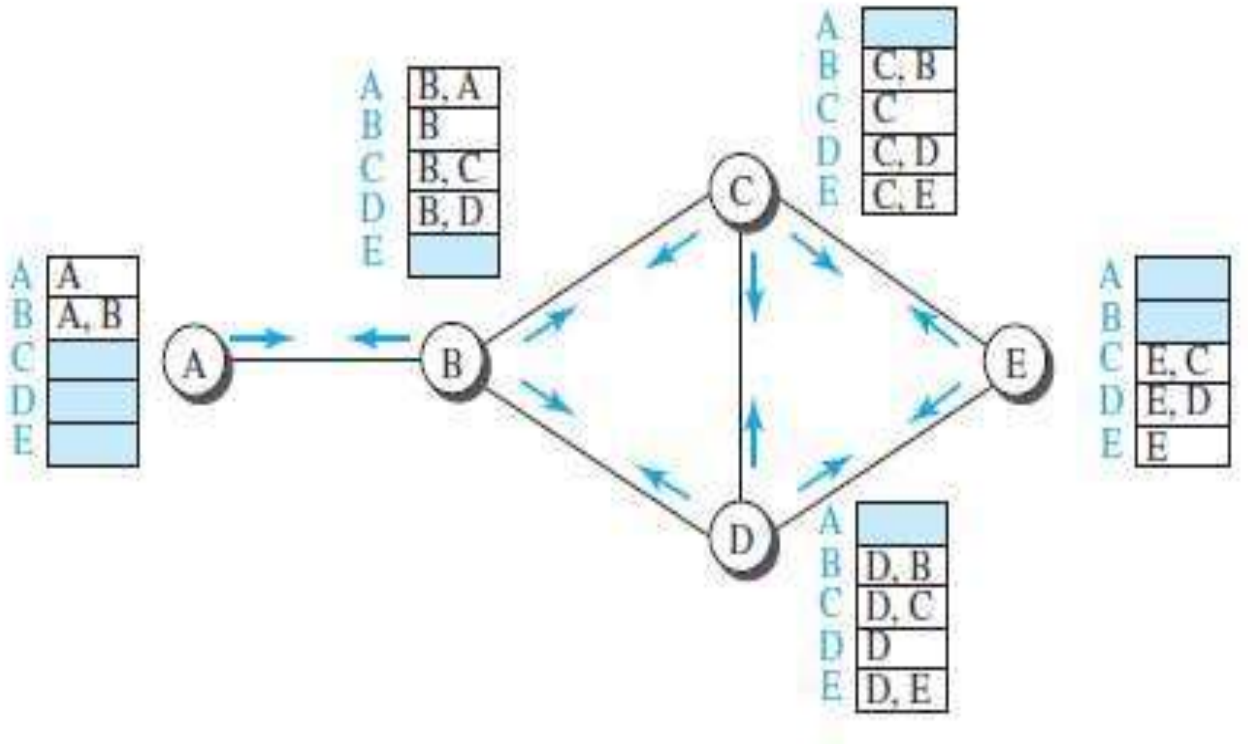
Figure 20.11 shows a small internet with only five nodes. Each source has created its own spanning tree that meets its policy. The policy imposed by all sources is to use the minimum number of nodes to reach a destination. The spanning tree selected by A and E is such that the communication does not pass through D as a middle node. Similarly, the spanning tree selected by B is such that the communication does not pass through C as a middle node.

Figure 20.11 *Spanning trees in path-vector routing*



Creation of Spanning Trees

Figure 20.12 *Path vectors made at booting time*



	A
A	A
B	A, B
C	
D	
E	

	B
A	B, A
B	B
C	B, C
D	B, D
E	

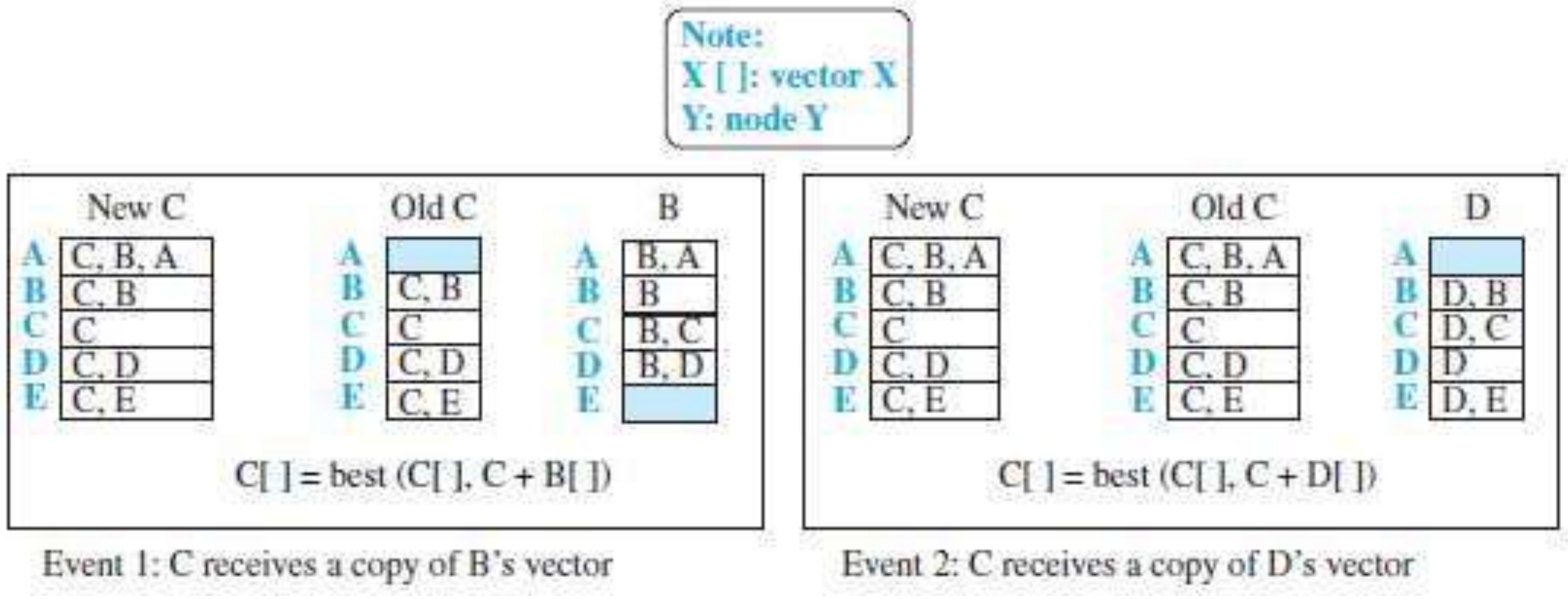
	C
A	
B	C, B
C	C
D	C, D
E	C, E

	D
A	
B	D, B
C	D, C
D	D
E	D, E

	E
A	
B	
C	E, C
D	E, D
E	E

Continue…

Figure 20.13 *Updating path vectors*


Note:
X [ ]: vector X
Y: node Y
C[ ] = best (C[ ], C + B[ ])
Event 1: C receives a copy of B's vector
C[ ] = best (C[ ], C + D[ ])
Event 2: C receives a copy of D's vector

Event 1: C receives a copy of B's vector

	New C	Old C	B
A	C, B, A		B, A
B	C, B	C, B	B
C	C	C	B, C
D	C, D	C, D	B, D
E	C, E	C, E	

Event 2: C receives a copy of D's vector

	New C	Old C	D
A	C, B, A	C, B, A	
B	C, B	C, B	D, B
C	C	C	D, C
D	C, D	C, D	D
E	C, E	C, E	D, E

References

- Data Communications and Networking, Behrouz A. Forouzan, Fifth Edition, TMH, 2013.