Data Communication & Networks

Chapter 5: Network Layer (Textbook)

Network Layer

- The network layer is concerned with getting packets from the source all the way to the destination.
- Getting to the destination may require making many hops at intermediate routers along the way.

- To achieve its goals, the network layer must learn about the topology of the network (i.e., the set of all routers and links) and compute appropriate paths through it, even for large networks.

- **Issues for autonomous systems:**

1. Coordinating traffic flows.
2. Managing network utilization

Network Layer Design Issues

- **Store-and-Forward Packet Switching**

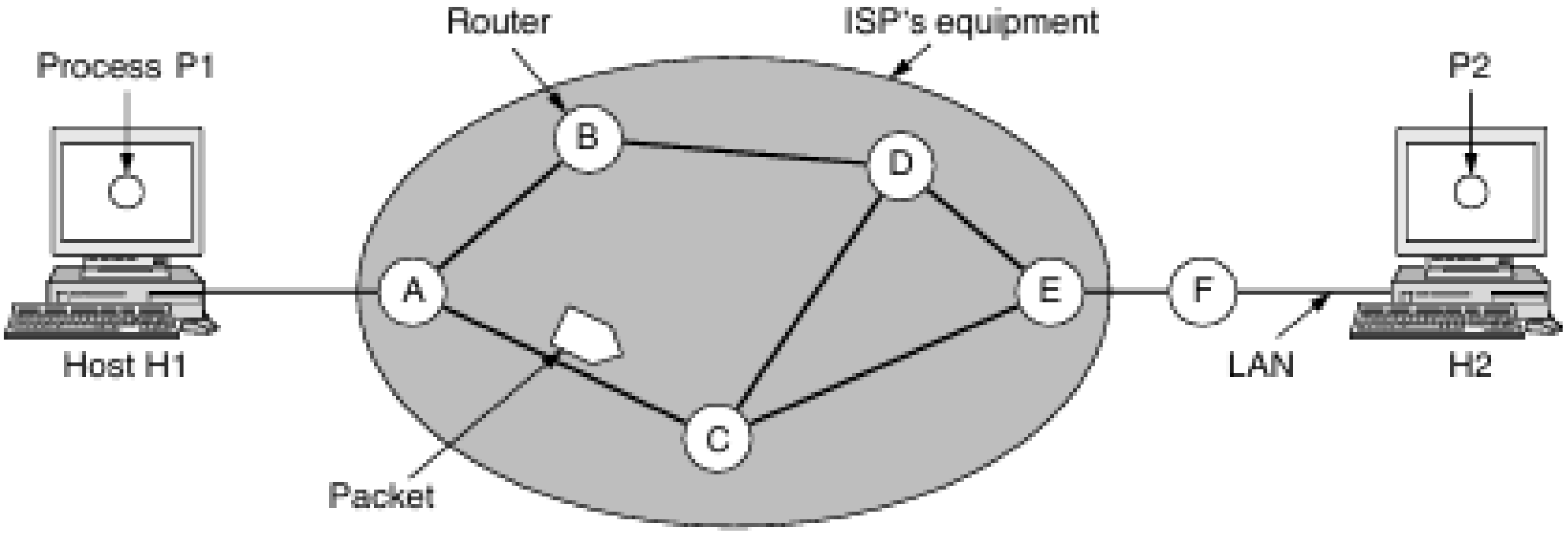


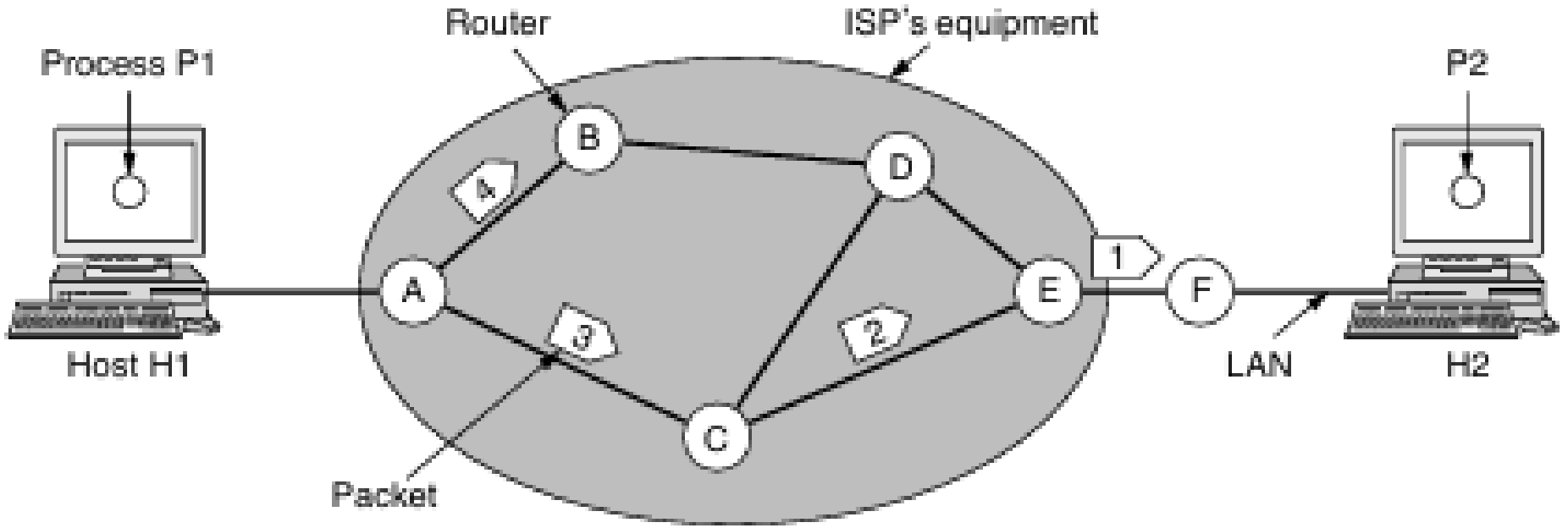
Figure 5-1. The environment of the network layer protocols.

Services Provided to the Transport Layer

- The network layer provides services to the transport layer at the network layer/transport layer interface.

1. The services should be independent of the router technology.
2. The transport layer should be shielded from the number, type, and topology of the routers present.
3. The network addresses made available to the transport layer should use a uniform numbering plan, even across LANs and WANs.

Implementation of Connectionless Service



A's table (initially)

Dest.	Line
A	–
B	B
C	C
D	B
E	C
F	C

A's table (later)

A	–
B	B
C	C
D	B
E	B
F	B

C's table

A	A
B	A
C	–
D	E
E	E
F	E

E's table

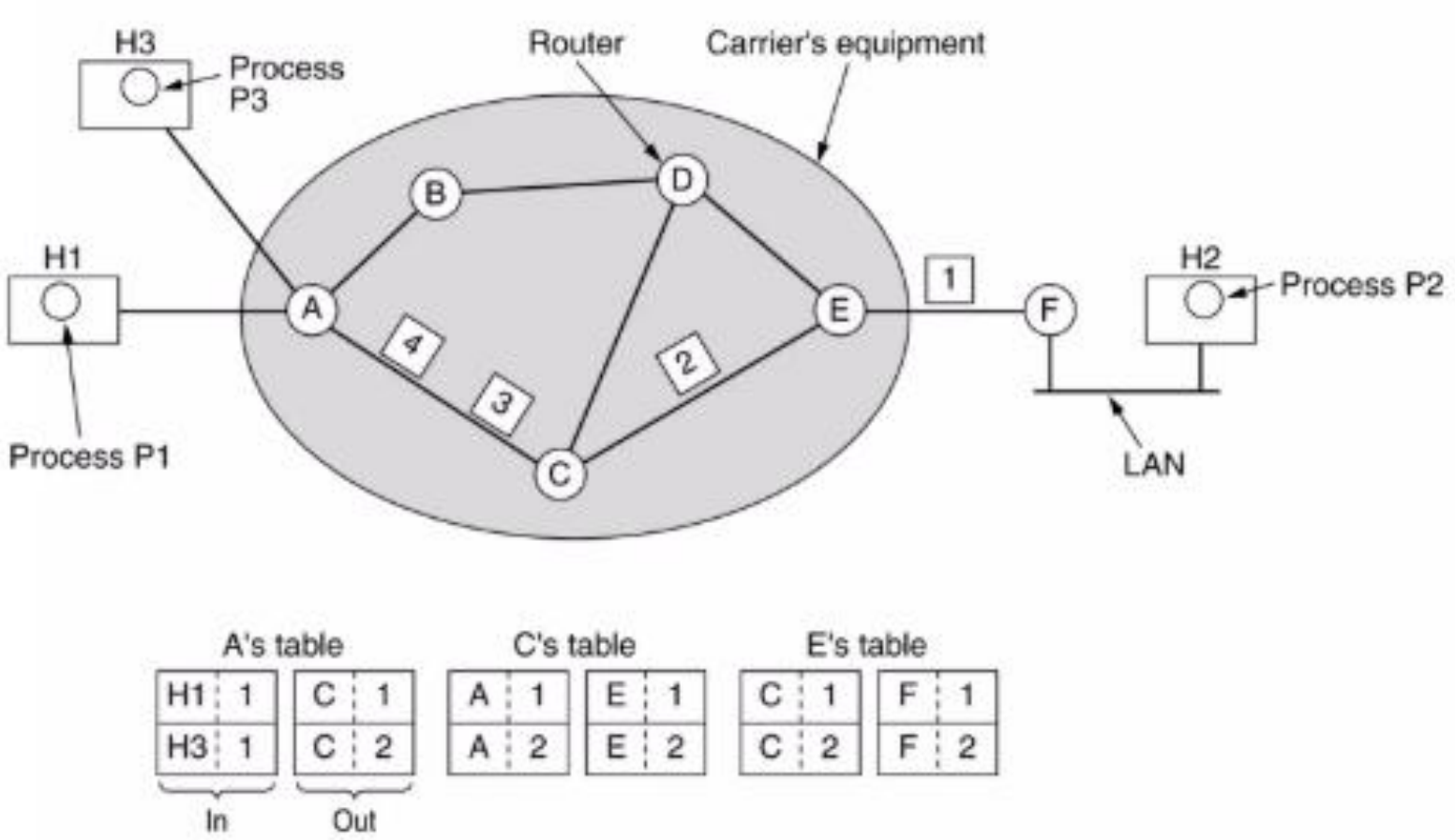
A	C
B	D
C	C
D	D
E	–
F	F

Figure 5-2. Routing within a datagram network.

Implementation of Connectionless Service

- The packets are frequently called datagrams (in analogy with telegrams) and the network is called a datagram network.
- The algorithm that manages the tables and makes the routing decisions is called the routing algorithm.

Implementation of Connection-oriented Service



Routing within a virtual-circuit subnet.

Comparison Between Datagram Network and Virtual-Circuit Network

Issue	Datagram network	Virtual-circuit network
Circuit setup	Not needed	Required
Addressing	Each packet contains the full source and destination address	Each packet contains a short VC number
State information	Routers do not hold state information about connections	Each VC requires router table space per connection
Routing	Each packet is routed independently	Route chosen when VC is set up; all packets follow it
Effect of router failures	None, except for packets lost during the crash	All VCs that passed through the failed router are terminated
Quality of service	Difficult	Easy if enough resources can be allocated in advance for each VC
Congestion control	Difficult	Easy if enough resources can be allocated in advance for each VC

Figure 5-4. Comparison of datagram and virtual-circuit networks.

Routing Algorithms

- The routing algorithm is that part of the network layer software responsible for deciding which output line an incoming packet should be transmitted on.

- For datagram, the routing decision must be made anew for every arriving data packet since the best route may have changed since last time.

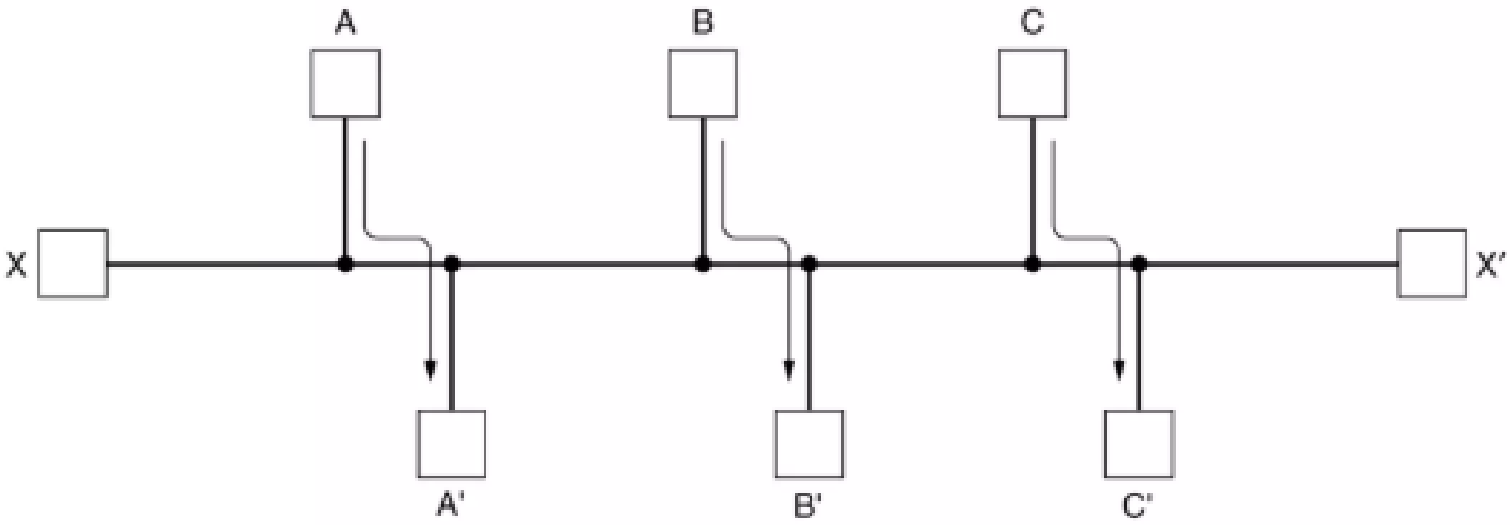
- For Virtual-Circuit Network, data packets just follow the already established route i.e. reservation. It is called session routing because a route remains in force for an entire session (e.g., while logged in over a VPN).

Routing Algorithms

- The Optimality Principle
- Shortest Path Routing
- Flooding
- Distance Vector Routing
- Link State Routing
- Hierarchical Routing
- Broadcast Routing
- Multicast Routing
- Routing for Mobile Hosts
- Routing in Ad Hoc Networks

Routing Algorithms

Routing Algorithms (2)



Conflict between fairness and optimality.

Routing Algorithms

- **Fairness** in a data communication network means that **each user or data flow gets a fair share of the available network resources**, ensuring that no single user monopolizes the bandwidth or starves others of service.

- **Optimality** means that the **network operates at its best performance level**, where resources (like bandwidth, buffer space, and link capacity) are used efficiently and no better solution exists under the given constraints.

Routing Algorithms: Open Shortest Path First (Dijkstra's)

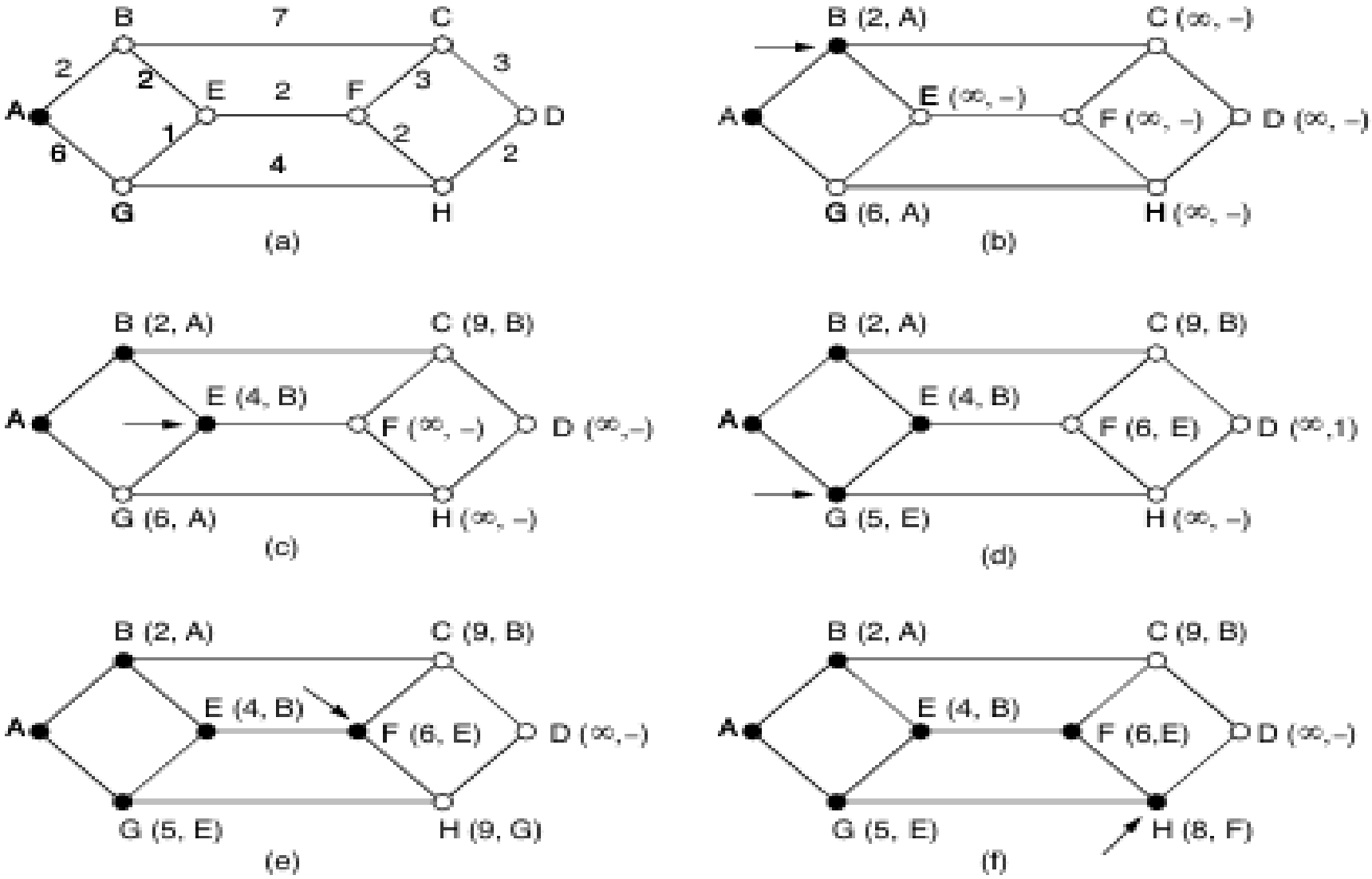
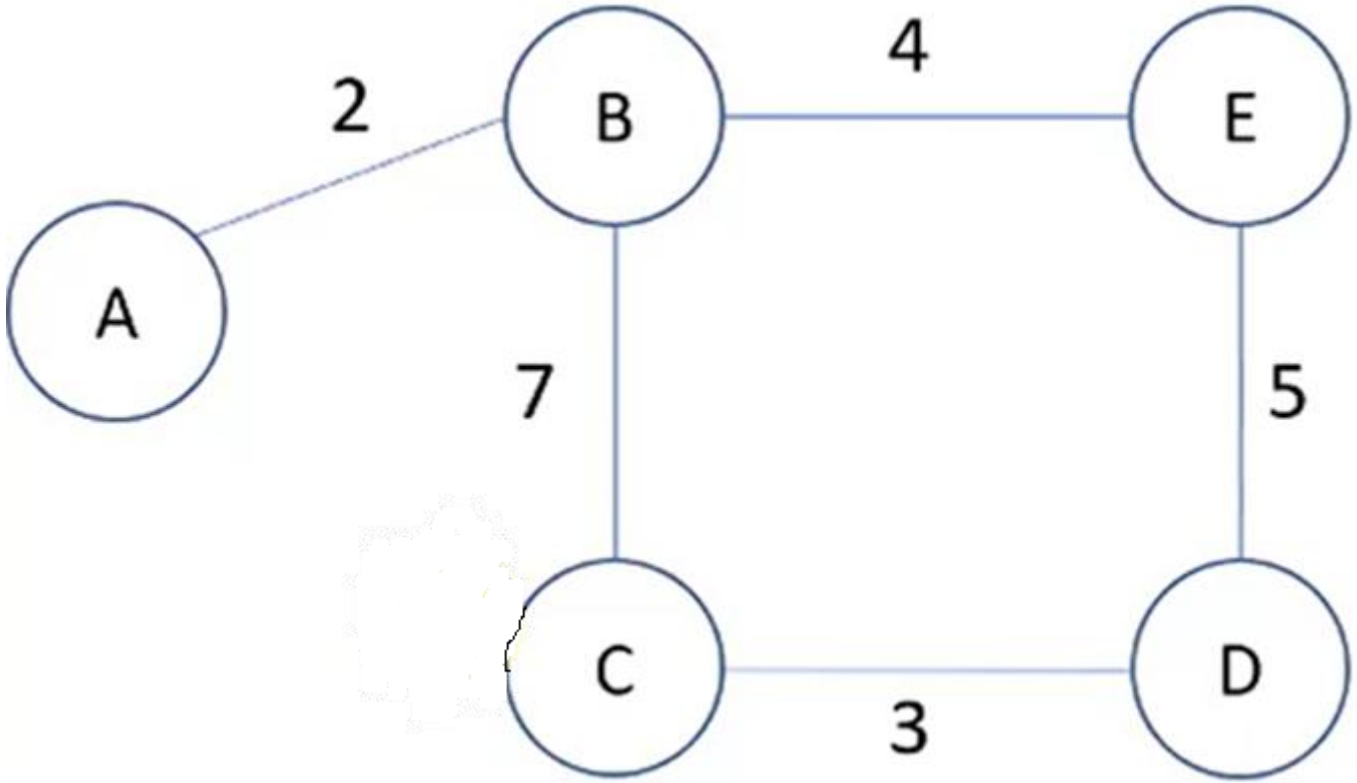


Figure 5-7. The first six steps used in computing the shortest path from *A* to *D*. The arrows indicate the working node.

Routing Algorithms: Distance Vector Routing Protocol/ Algorithm

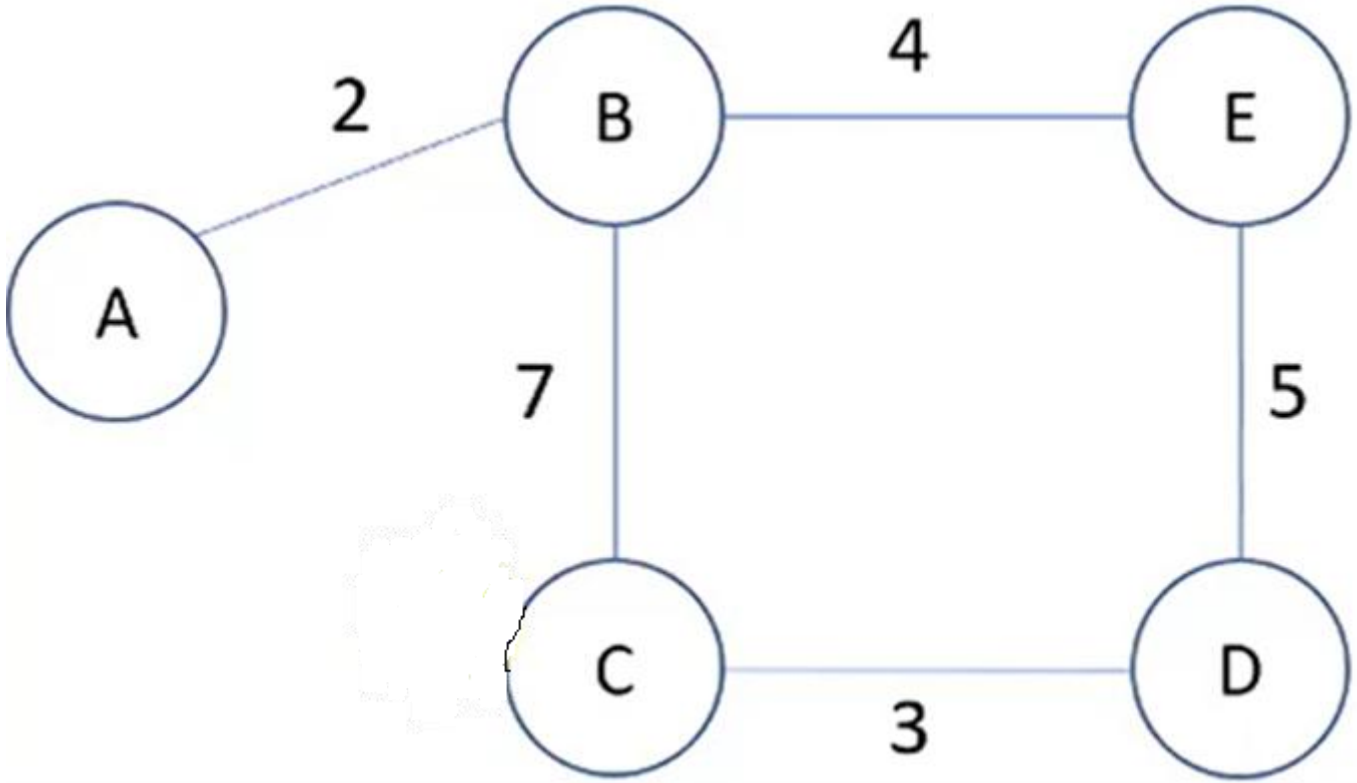
1. Local routing table
2. Share with neighbor (only distance vector)
3. Update routing tables

Routing Algorithms: Distance Vector Routing Protocol/ Algorithm



1. Local routing table
2. Share with neighbor (only distance vector)
3. Update routing tables

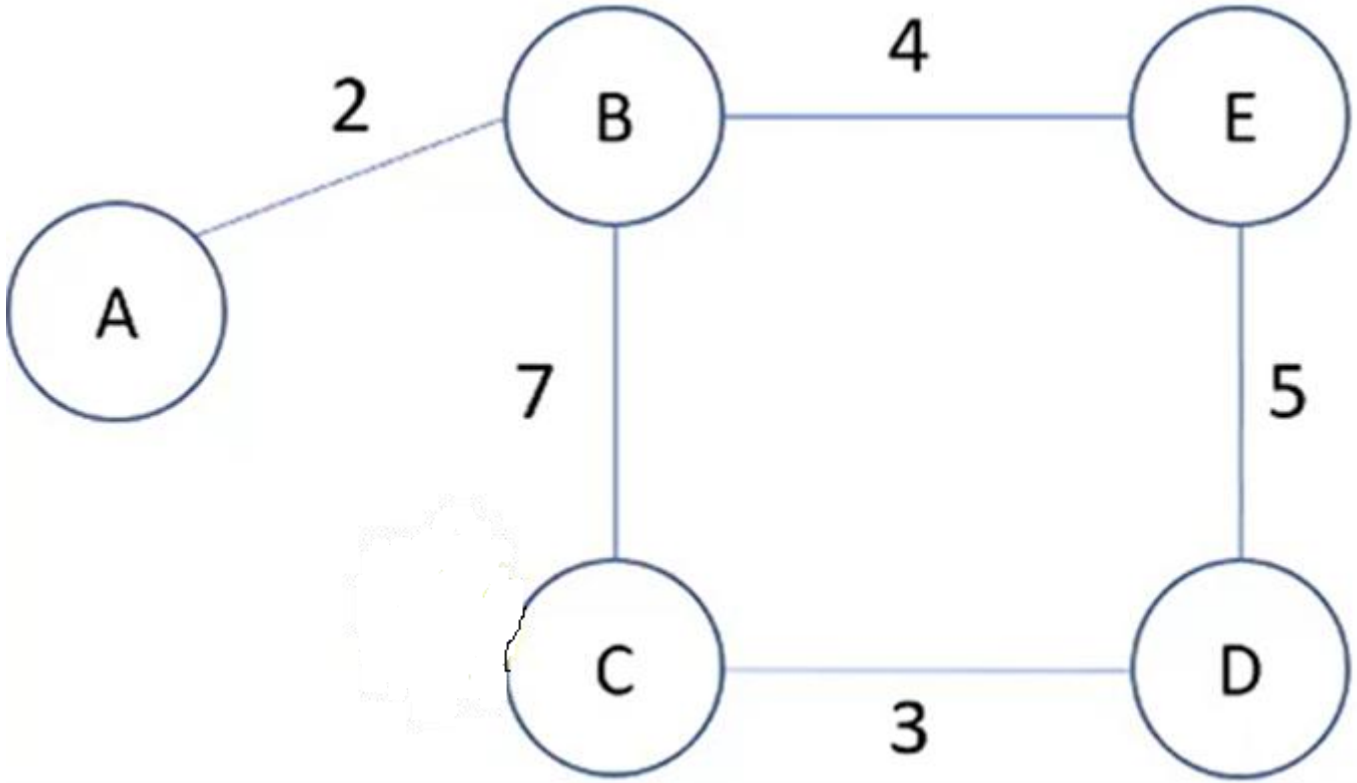
Routing Algorithms: Distance Vector Routing Protocol/ Algorithm



Local routing table A

Dest.	Dist.	Next Node
A	0	A
B	2	B
C	∞	-
D	∞	-
E	∞	-

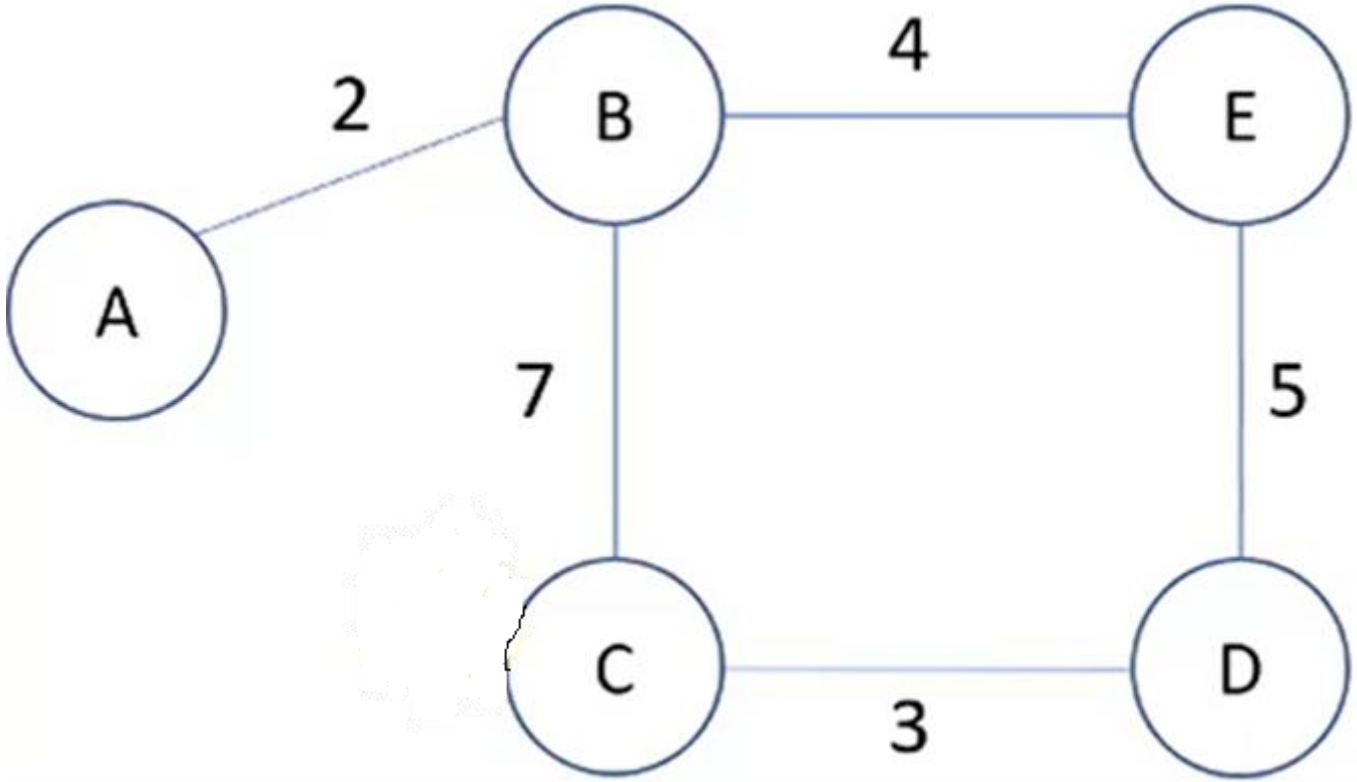
Routing Algorithms: Distance Vector Routing Protocol/ Algorithm



Local routing table B

Dest.	Dist.	Next Node
A	2	A
B	0	B
C	7	C
D	∞	-
E	4	E

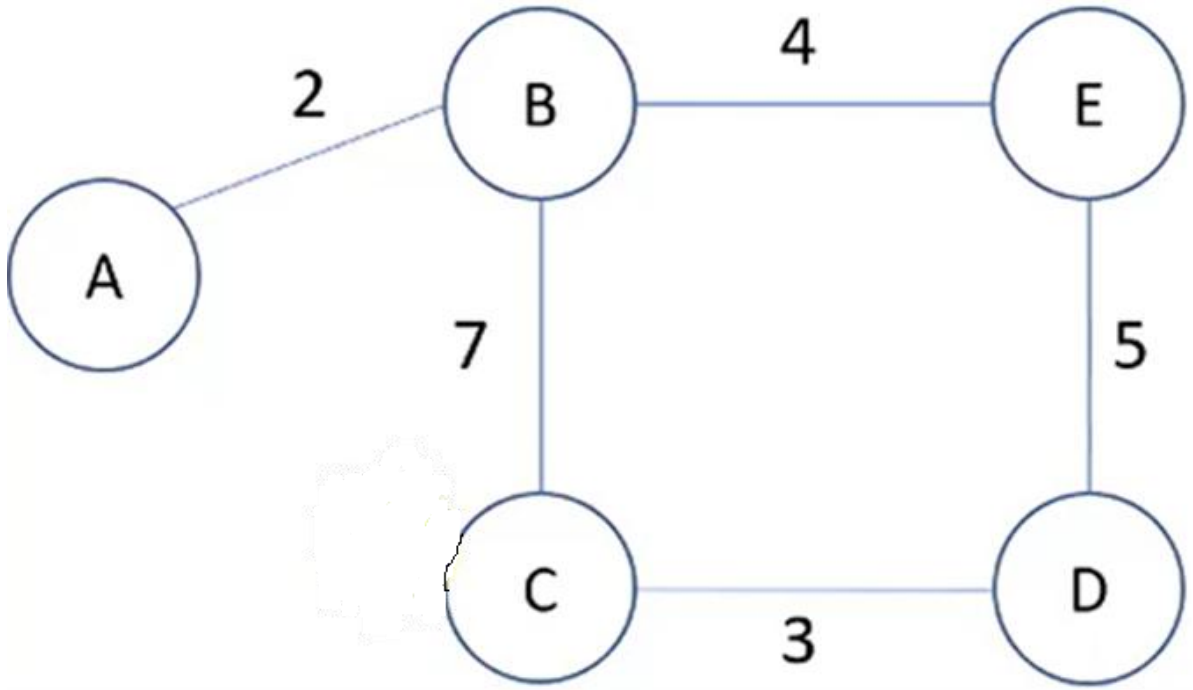
Routing Algorithms: Distance Vector Routing Protocol/ Algorithm



Local routing table C

Dest.	Dist.	Next Node
A	∞	-
B	7	B
C	0	C
D	3	D
E	∞	-

Routing Algorithms: Distance Vector Routing Protocol/ Algorithm



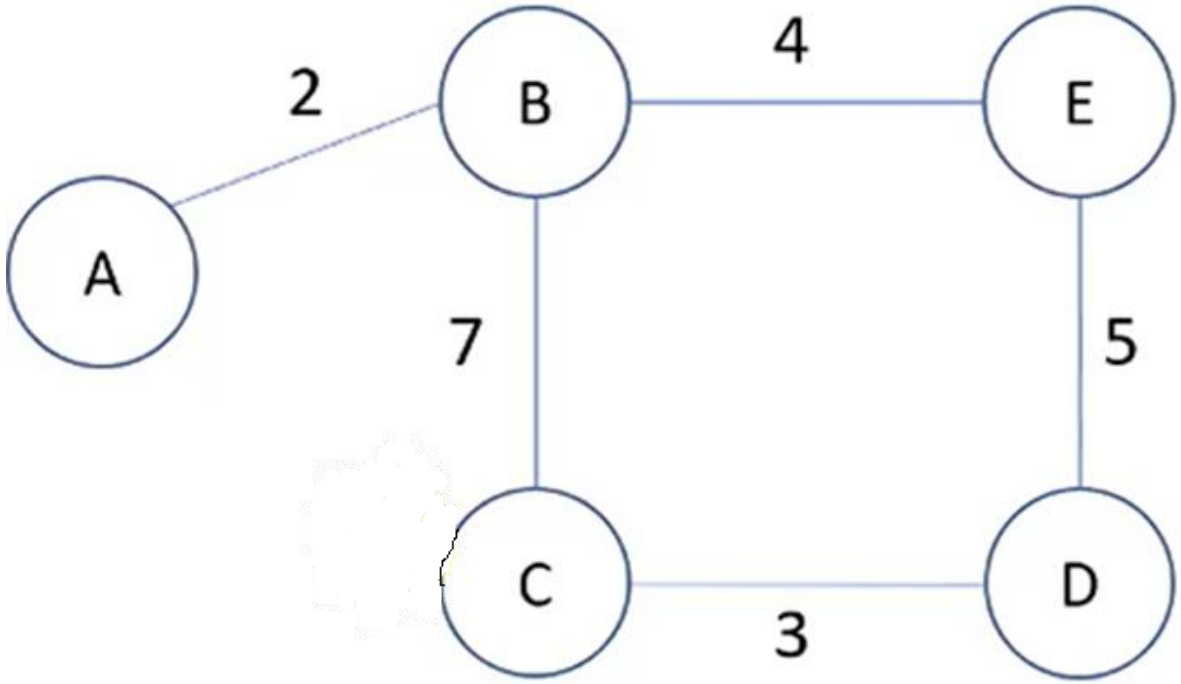
Local routing table D

Dest.	Dist.	Next Node
A	∞	-
B	∞	-
C	3	C
D	0	D
E	5	E

Local routing table E

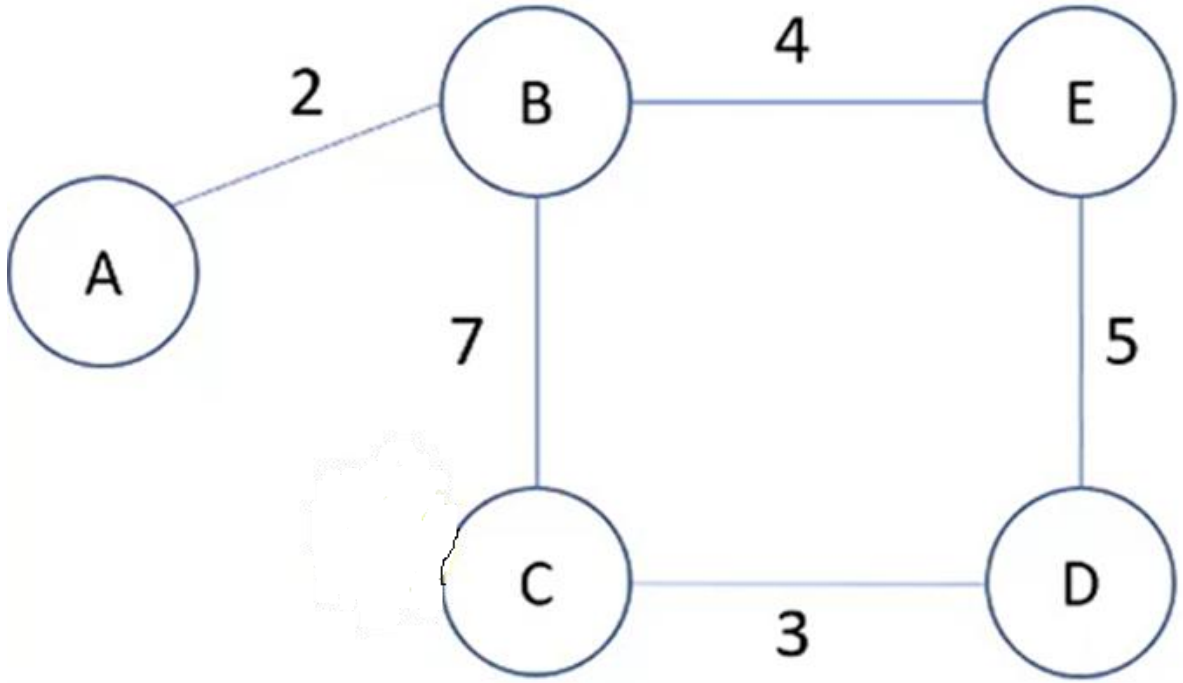
Dest.	Dist.	Next Node
A	∞	-
B	4	B
C	∞	-
D	5	D
E	0	E

Routing Algorithms: Distance Vector Routing Protocol/ Algorithm



A	B	C	D	E
DV	DV	DV	DV	DV
0	2	∞	∞	∞
2	0	7	∞	4
∞	7	0	3	∞
∞	∞	3	0	5
∞	4	∞	5	0

Routing Algorithms: Distance Vector Routing Protocol/ Algorithm



A

DV
0
2
∞
∞
∞

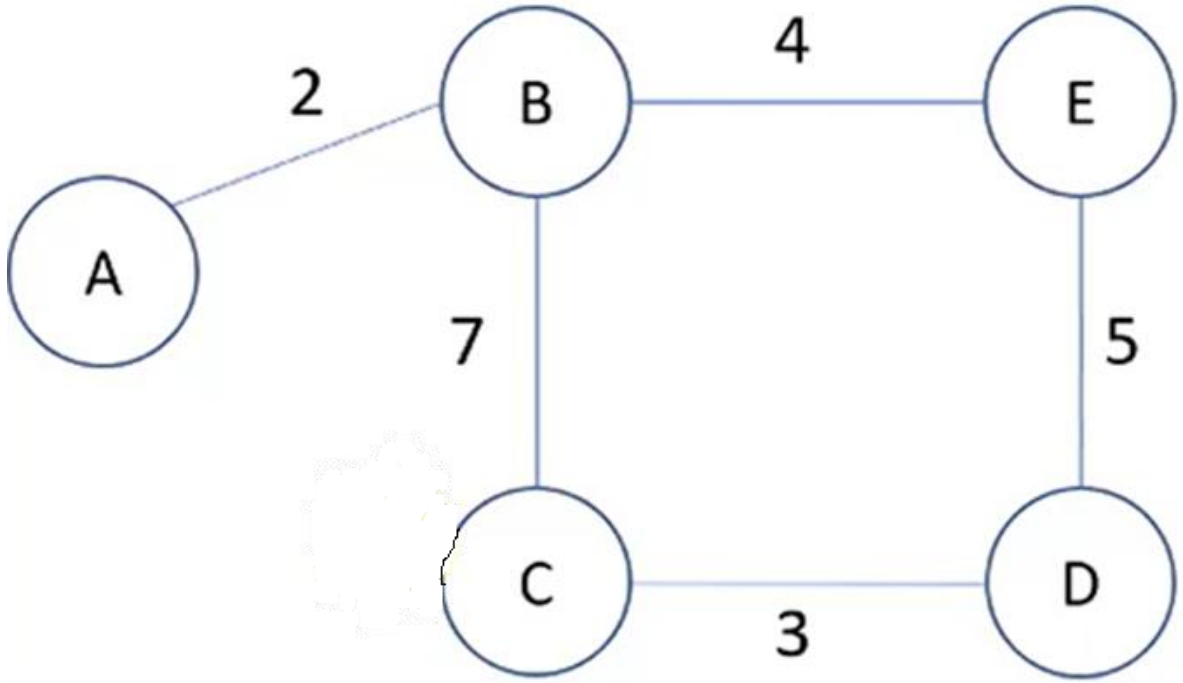
B

	DV
A	2
B	0
C	7
D	∞
E	4

Updated routing table A

Dest.	Dist.	Next Node
A	0	A
B	2	B
C	9	B,C
D	∞	-
E	6	B,E

Routing Algorithms: Distance Vector Routing Protocol/ Algorithm



B

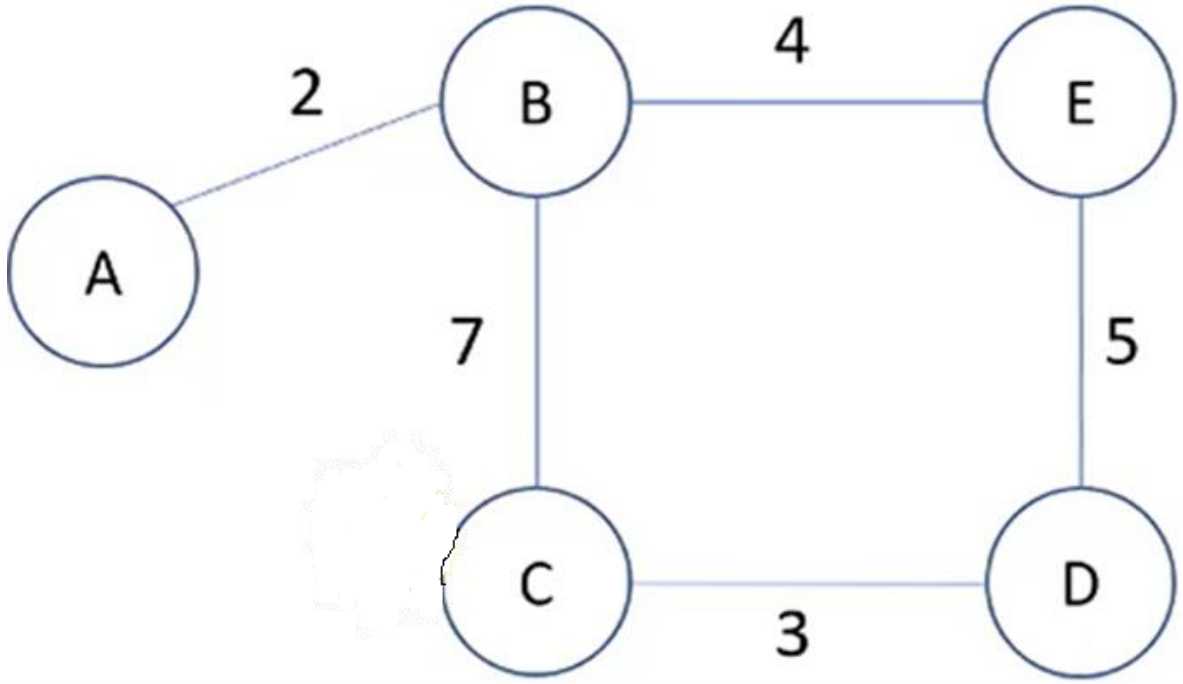
DV
2
0
7
∞
4

	A	C	E
	DV	DV	DV
A	0	∞	∞
B	2	7	4
C	∞	0	∞
D	∞	3	5
E	∞	∞	0

Updated routing table B

Dest.	Dist.	Next Node
A	2	A
B	0	B
C	7	C
D	9	E,D
E	4	E

Routing Algorithms: Distance Vector Routing Protocol/ Algorithm



C

DV
∞
7
0
3
∞

B

	DV
A	2
B	0
C	7
D	∞
E	4

D

DV
∞
∞
3
0
5

Updated routing table C

Dest.	Dist.	Next Node
A	9	B, A
B	7	B
C	0	D,C
D	3	D
E	8	D, E

Routing Algorithms: Distance Vector Routing Protocol/ Algorithm

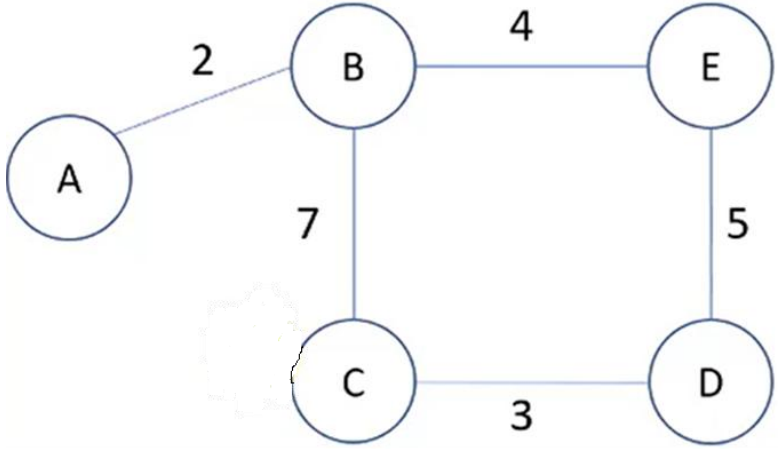
A	B	C	D	E
DV	DV	DV	DV	DV
0	2	∞	∞	∞
2	0	7	∞	4
∞	7	0	3	∞
∞	∞	3	0	5
∞	4	∞	5	0

Local distance vector

A	B	C	D	E
DV	DV	DV	DV	DV
0	2	9	∞	6
2	0	7	9	4
9	7	0	3	8
∞	9	3	0	5
6	4	8	5	0

distance vector after 1st iteration

Routing Algorithms: Distance Vector Routing Protocol/ Algorithm



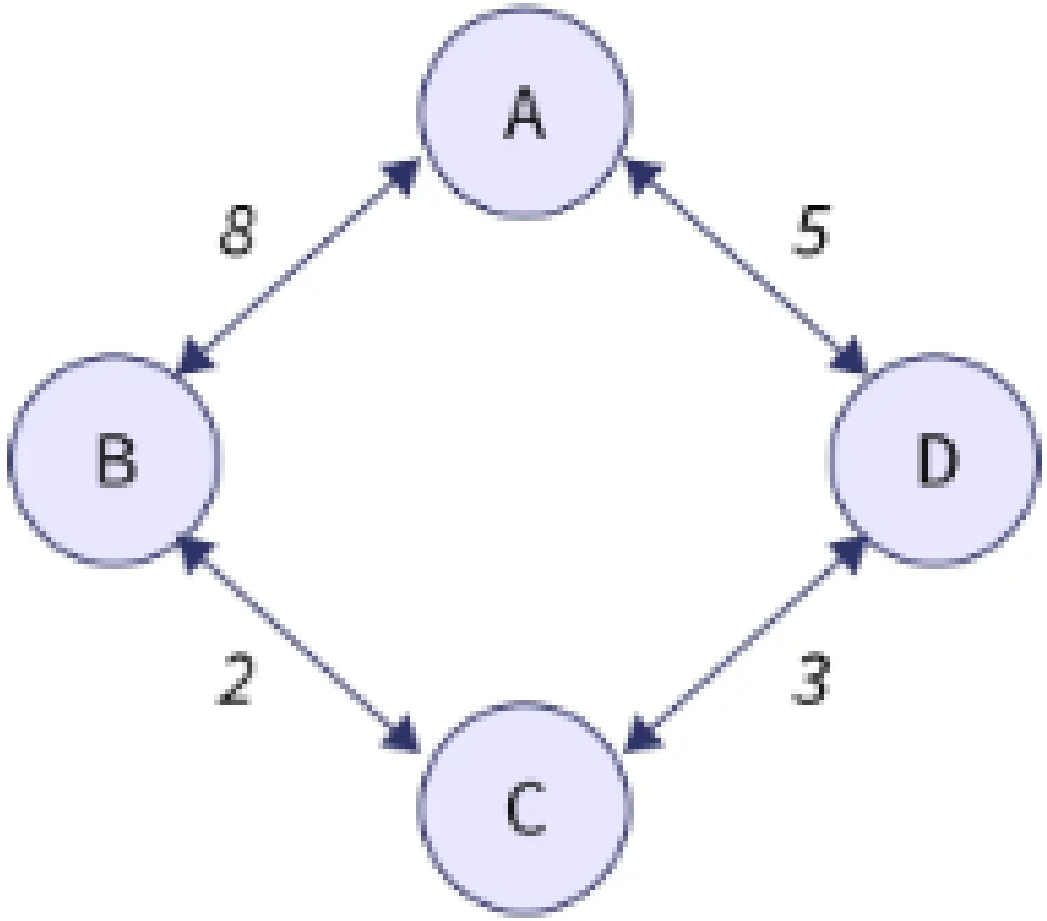
B Updated routing table A

	DV	Dest.	Dist.	Next Node
A	2	A	0	A
B	0	B	2	B
C	7	C	9	B,C
D	9	D	11	B,E,D
E	4	E	6	B,E

C E Updated routing table D

	DV	DV	Dest.	Dist.	Next Node
A	9	6	A	11	E,B,A
B	7	4	B	9	E, B
C	0	8	C	3	C
D	3	5	D	0	D
E	8	0	E	5	E

Distance Vector Routing Protocol/ Algorithm: Do Yourself



Example 1: Distance Vector Routing Protocol/ Algorithm

Routing table of A:

Destination	distant	Hop
A	0	A
B	8	B
C	infinity	-
D	5	D

Example 1: Distance Vector Routing Protocol/ Algorithm

Routing table of B:

Destination	distant	Hop
A	8	A
B	0	B
C	2	C
D	infinity	D

Example 1: Distance Vector Routing Protocol/ Algorithm

Routing table of C:

Destination	distant	Hop
A	infinity	-
B	2	B
C	0	C
D	3	D

Example 1: Distance Vector Routing Protocol/ Algorithm

Routing table of D :

Destination	distant	Hop
A	5	A
B	infinity	B
C	3	C
D	0	D

Example 1: Distance Vector Routing Protocol/ Algorithm

Consequently, A's new routing table is:

Destination	distant	Hop
A	0	A
B	8	B
C	8	D
D	5	D

Example 1: Distance Vector Routing Protocol/ Algorithm

Destination	distant	Hop
A	8	A
B	0	B
C	2	C
D	5	C

Example 1: Distance Vector Routing Protocol/ Algorithm

Consequently, C's new routing table is:

Destination	distant	Hop
A	8	D
B	2	B
C	0	C
D	3	D

Example 1: Distance Vector Routing Protocol/ Algorithm

Consequently, D's new routing table is:

Destination	distant	Hop
A	5	A
B	5	C
C	3	C
D	0	D

Logical Addressing : Chapter 19 (Behrouz Ferozan)

- IPv4 : 32 bit address.
- IPv6 : 128 bit address.

Logical Addressing: IPv4

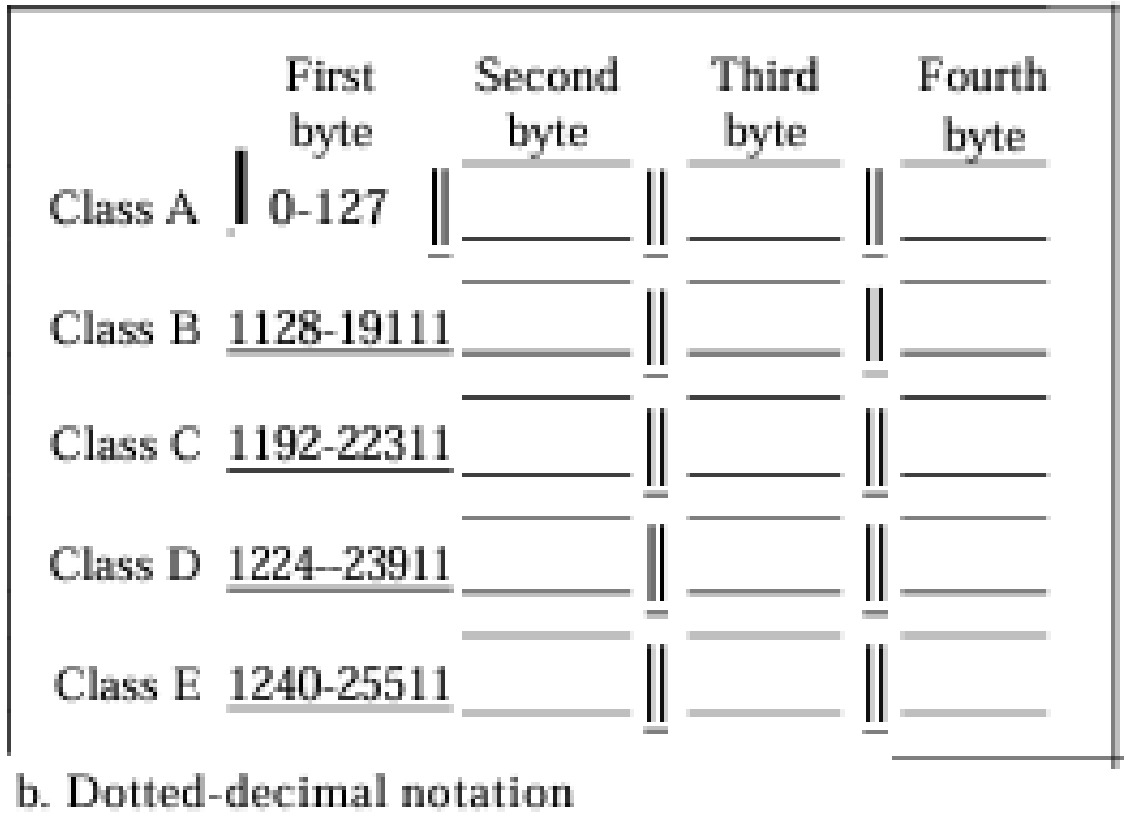
- IPv4 uses 32-bit addresses, which means that the address space is 232 or 4,294,967,296 (more than 4 billion).
- **Notations**
  - Binary Notation
    01110101 10010101 00011101 00000010
  - Dotted-Decimal Notation
    117.149.29.2
  - Solve Example 19.1, 19.2,19.3

Logical Addressing: IPv4

- IPv4 uses 32-bit addresses, which means that the address space is 232 or 4,294,967,296 (more than 4 billion).
- **Notations**
  - Binary Notation
    01110101 10010101 00011101 00000010
  - Dotted-Decimal Notation
    117.149.29.2
  - Solve Example 19.1, 19.2,19.3

Logical Addressing: IPv4

- Classful Addressing Scheme



b. Dotted-decimal notation

- Solve Example 19.4

Logical Addressing: IPv4

- Classful Addressing Scheme

Table 19.1 *Number of blocks and block size in classful IPv4 addressing*

Class	*Number of Blocks*	*Block Size*	*Application*
A	128	16,777,216	Unicast
B	16,384	65,536	Unicast
C	2,097,152	256	Unicast
D	1	268,435,456	Multicast
E	1	268,435,456	Reserved

- Class A addresses were designed for large organizations with a large number of attached hosts or routers. Class B addresses were designed for midsize organizations with tens of thousands of attached hosts or routers. Class C addresses were designed for small organizations with a small number of attached hosts or routers.

Logical Addressing: IPv4

- Classful Addressing Scheme

- **Netid and Hostid**
- In classful addressing, an IP address in class A, B, or C is divided into netid and hosted.

- **Mask**
- The mask can help us to find the netid and the hostid. For example, the mask for a class A address has eight 1s, which means the first 8 bits of any address in class A define the netid; the next 24 bits define the hostid.

Logical Addressing: IPv4

Table 19.2 *Default masks for classful addressing*

Class	*Binary*	*Dotted-Decimal*	*CIDR*
A	11111111 00000000 00000000 00000000	255.0.0.0	18
B	11111111 11111111 00000000 00000000	255.255.0.0	116
C	11111111 11111111 11111111 00000000	255.255.255.0	124

- The mask can help us to find the netid and the hostid. For example, the mask for a class A address has eight 1s, which means the first 8 bits of any address in class A define the netid; the next 24 bits define the hostid.

Logical Addressing: IPv4

Table 19.2 *Default masks for classful addressing*

Class	*Binary*	*Dotted-Decimal*	*CIDR*
A	11111111 00000000 00000000 00000000	255.0.0.0	*18*
B	11111111 11111111 00000000 00000000	255.255.0.0	*116*
C	11111111 11111111 11111111 00000000	255.255.255.0	*124*

- **Classful addressing, which is almost obsolete, is replaced with classless addressing.**

Logical Addressing: IPv4

- **Classless Addressing:**

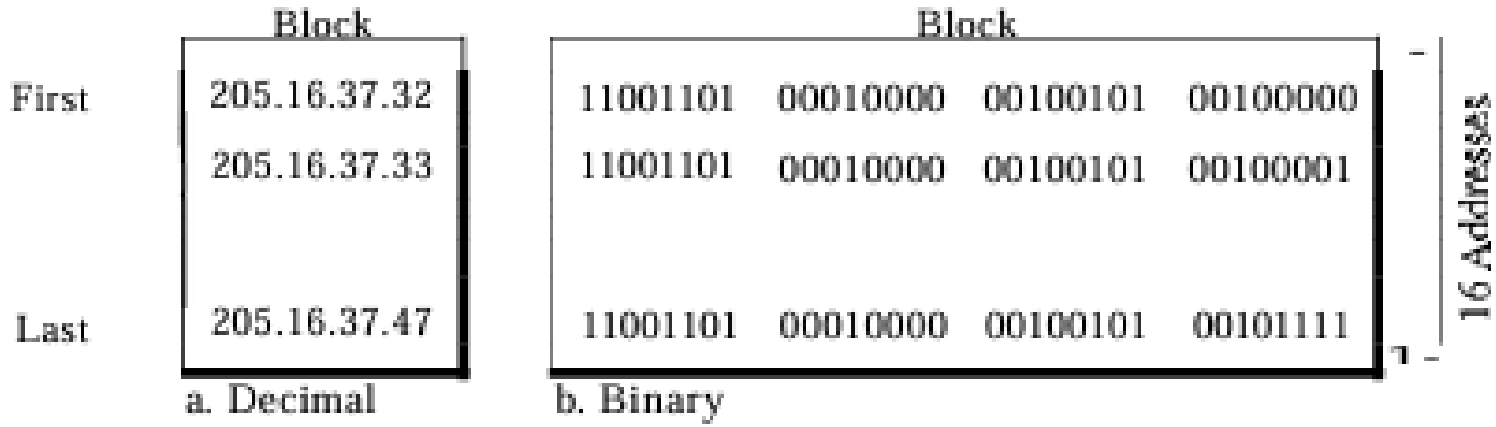
Address Blocks: when an entity, small or large, needs to be connected to the Internet, it is granted a block (range) of addresses. The size of the block (the number of addresses) varies based on the nature and size of the entity. For example, a household may be given only two addresses; a large organization may be given thousands of addresses. An ISP, as the Internet service provider, may be given thousands or hundreds of thousands based on the number of customers it may serve.

Logical Addressing: IPv4

- **Classless Addressing:**
- Restrictions

1. The addresses in a block must be contiguous, one after another.
2. The number of addresses in a block must be a power of 2 (I, 2, 4, 8, . . .).
3. The first address must be evenly divisible by the number of addresses.

Figure 19.3 *A block of* 16 *addresses granted to a small organization*



Logical Addressing: IPv4

- **Classless Addressing:**
- Mask

- In classless addressing, the mask for a block can take any value from 0 to 32. It is very convenient to give just the value of n preceded by a slash (Classless Inter domain routing: CIDR notation).

In 1Pv4 addressing, a block of addresses can be defined as

x.y.z.tln

in which x.y.z.t defines one of the addresses and the *In* defines the mask.

The first address in the block can be found by setting the rightmost 32 - *n* bits to Os.

Logical Addressing: IPv4

- Mask

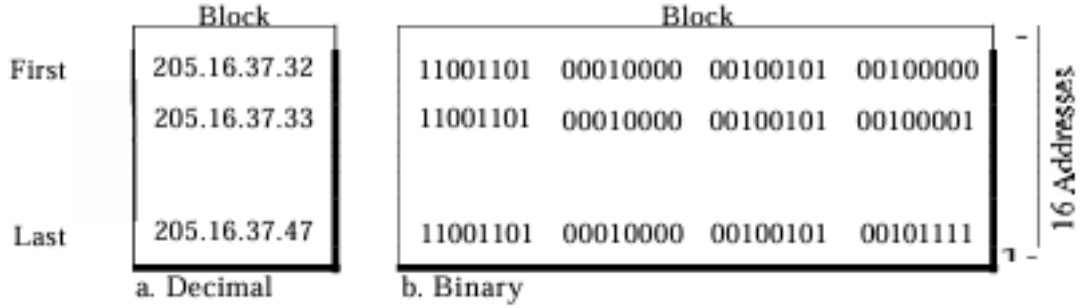
Example 19.6

A block of addresses is granted to a small organization. We know that one of the addresses is 205.16.37.39/28. What is the first address in the block?

Solution

The binary representation of the given address is 11001101 00010000 00100101 00100111. If we set 32 - 28 rightmost bits to 0, we get 11001101 00010000 00100101 0010000 or 205.16.37.32. This is actually the block shown in Figure 19.3.

Figure 19.3 *A block of 16 addresses granted to a small organization*



Logical Addressing: IPv4

- Mask

The last address in the block can be found by setting the rightmost 32 - *n* bits to Is.

Example 19.7

Find the last address for the block in Example 19.6.

Solution

The binary representation of the given address is 11001101 00010000000100101001001 11. If we set 32 - 28 rightmost bits to 1, we get 11001101 00010000 00100101001011 11 or 205.16.37.47. This is actually the block shown in Figure 19.3.