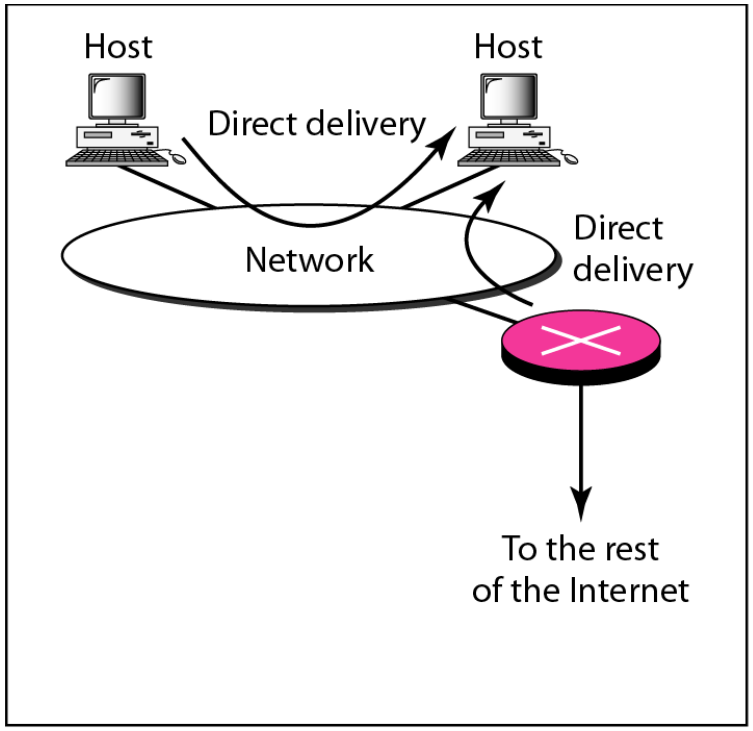
Network Layer: Delivery, Forwarding, and Routing

Delivery

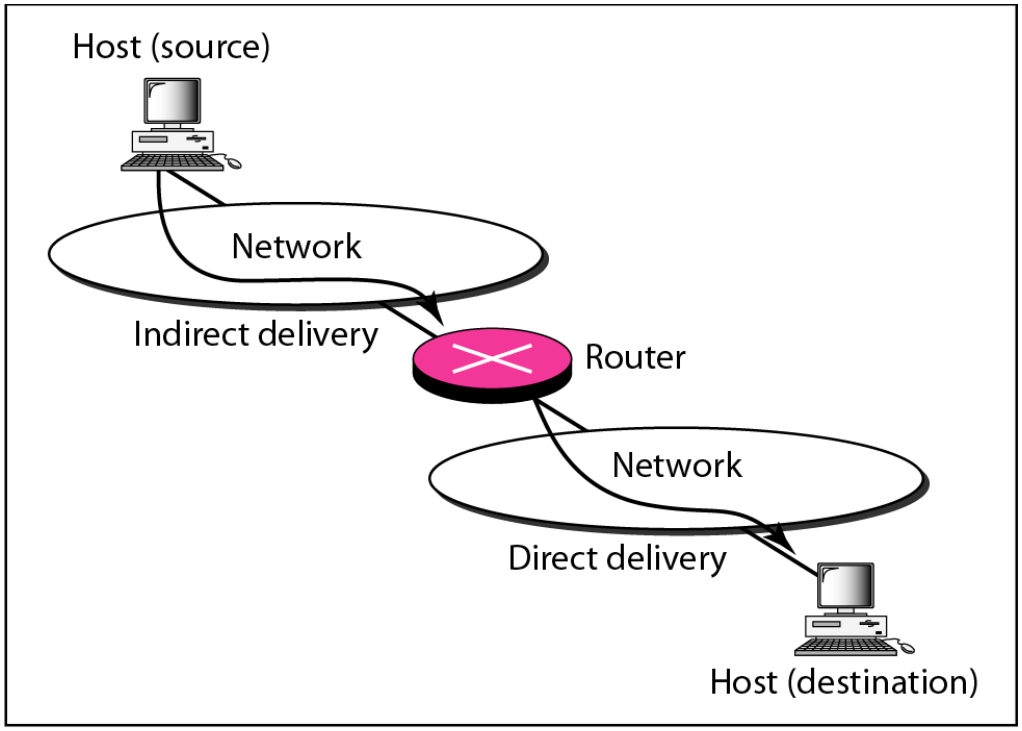
- ***The network layer supervises the handling of the packets by the underlying physical networks.***

- ***We define this handling as the delivery of a packet.***

Direct and indirect delivery



a. Direct delivery



b. Indirect and direct delivery

- **When the source and destination of the packet are located on the same physical network**
- **When the delivery is between the last router and the destination host**

- **When the destination host is not in the same network**
- **Packet goes from router to router**
- **But final delivery is the direct delivery**

Forwarding

- ***Forwarding means to place the packet in its route to its destination.***
- ***Forwarding requires a host or a router to have a routing table.***
- ***When a host has a packet to send or when a router has received a packet to be forwarded, it looks at this table to find the route to the final destination.***
- ***We will look into***
    - **Forwarding Techniques**
    - **Forwarding Process**
    - **Routing Table**

Forwarding Techniques: Route method versus next-hop method

a. Routing tables based on route

Destination	Route
Host B	R1, R2, host B

Routing table for host A

Destination	Route
Host B	R2, host B

Routing table for R1

Destination	Route
Host B	Host B

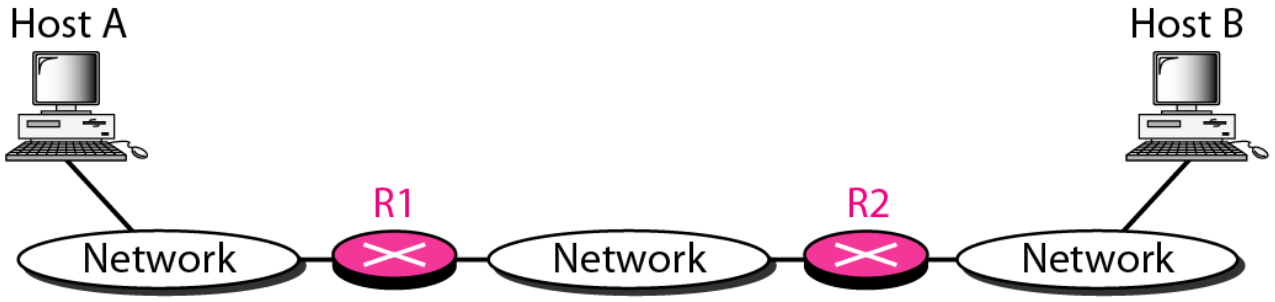
Routing table for R2

b. Routing tables based on next hop

Destination	Next hop
Host B	R1

Destination	Next hop
Host B	R2

Destination	Next hop
Host B	---



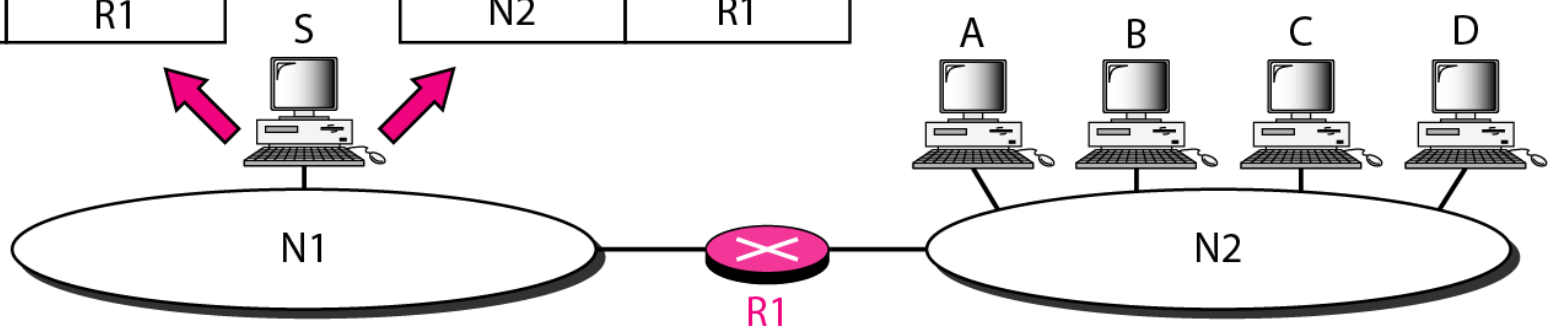
Forwarding Techniques: Host-specific versus network-specific method

Routing table for host S based on host-specific method

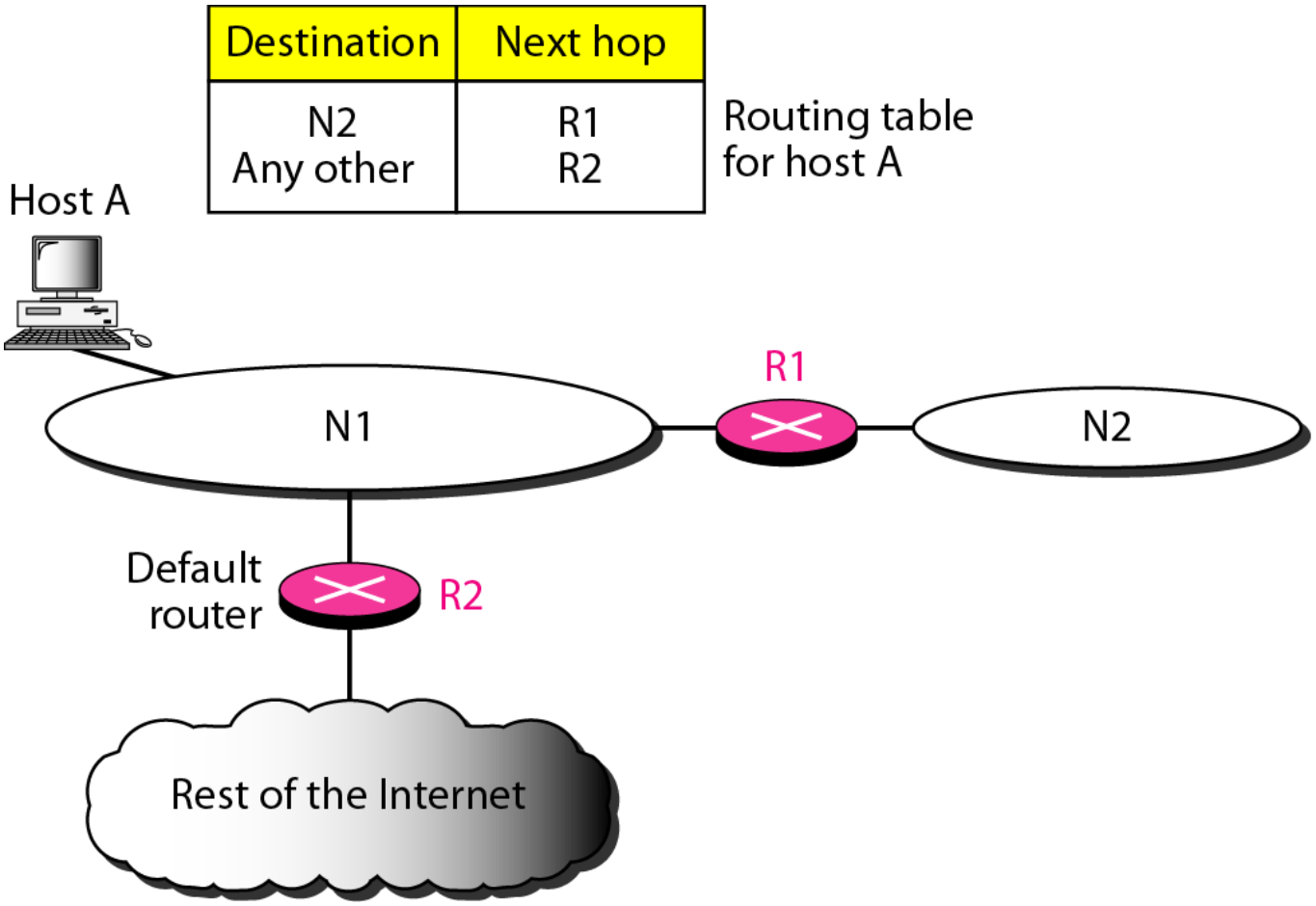
Destination	Next hop
A	R1
B	R1
C	R1
D	R1

Routing table for host S based on network-specific method

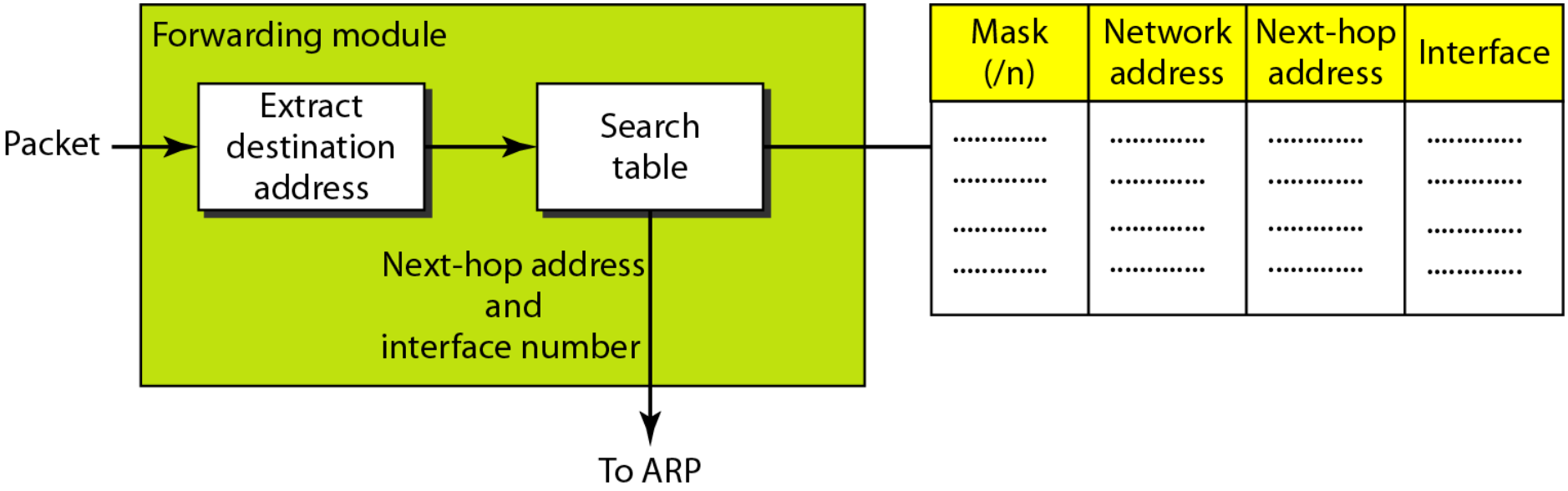
Destination	Next hop
N2	R1



Forwarding Techniques: Default method



Simplified forwarding module in classless address

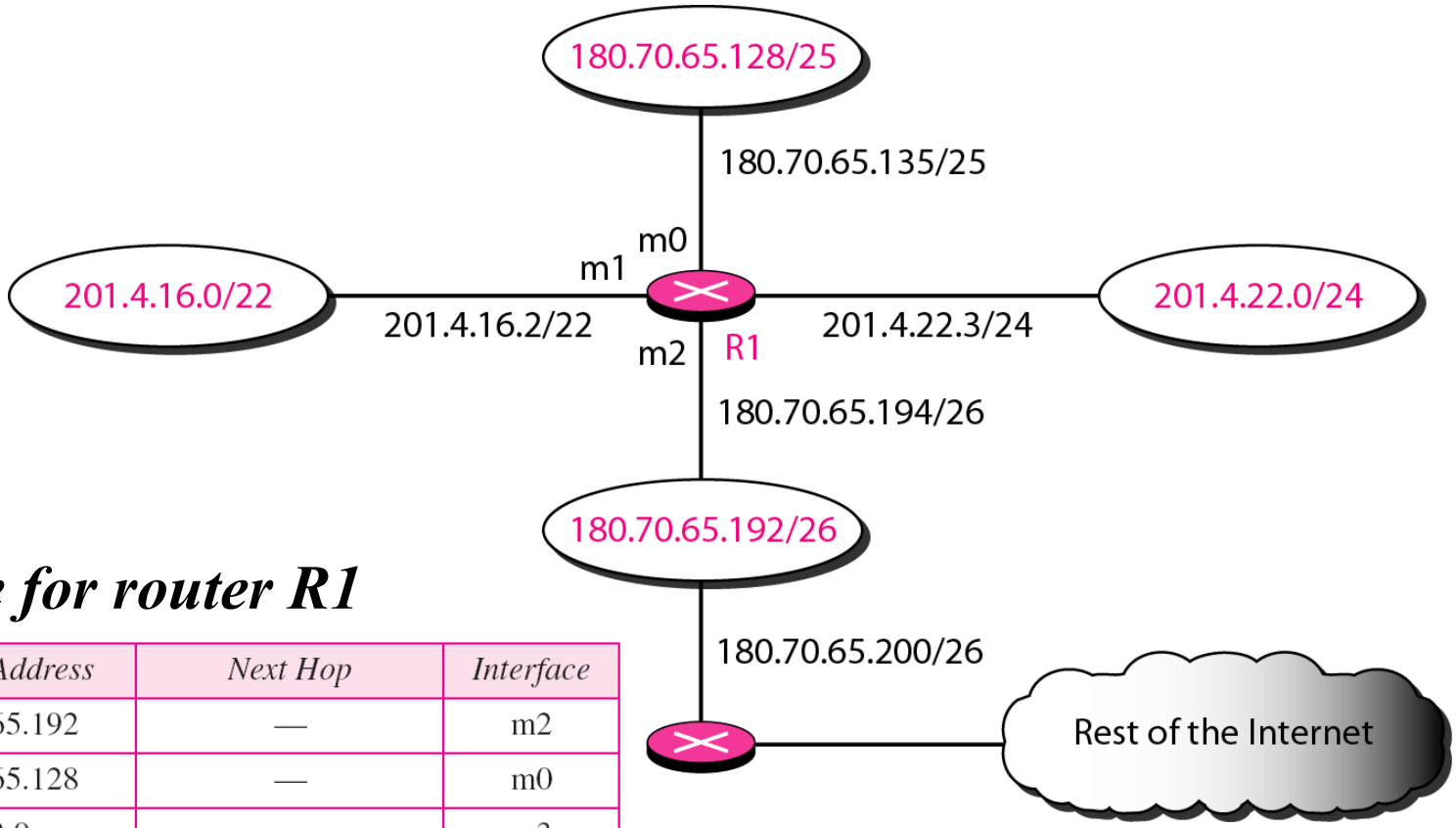


Note

In classless addressing, we need at least four columns in a routing table.

Example

Make a routing table for router R1, using the configuration in.



Routing table for router R1

Mask	*Network Address*	*Next Hop*	*Interface*
/26	180.70.65.192	—	m2
/25	180.70.65.128	—	m0
/24	201.4.22.0	—	m3
/22	201.4.16.0	....	m1
Any	Any	180.70.65.200	m2

Example

Show the forwarding process if a packet arrives at R1 in with the destination address 180.70.65.140.

Solution

The router performs the following steps:

1. ***The first mask (/26) is applied to the destination address. The result is 180.70.65.128, which does not match the corresponding network address.***
2. ***The second mask (/25) is applied to the destination address. The result is 180.70.65.128, which matches the corresponding network address. The next-hop address and the interface number m0 are passed to ARP for further processing.***

Example

Show the forwarding process if a packet arrives at R1 in with the destination address 201.4.22.35.

Solution

The router performs the following steps:

1. ***The first mask (/26) is applied to the destination address. The result is 201.4.22.0, which does not match the corresponding network address.***
2. ***The second mask (/25) is applied to the destination address. The result is 201.4.22.0, which does not match the corresponding network address (row 2).***

Example (continued)

3. ***The third mask (/24) is applied to the destination address. The result is 201.4.22.0, which matches the corresponding network address. The destination address of the packet and the interface number m3 are passed to ARP.***

Example

Show the forwarding process if a packet arrives at R1 in with the destination address 18.24.32.78.

Solution

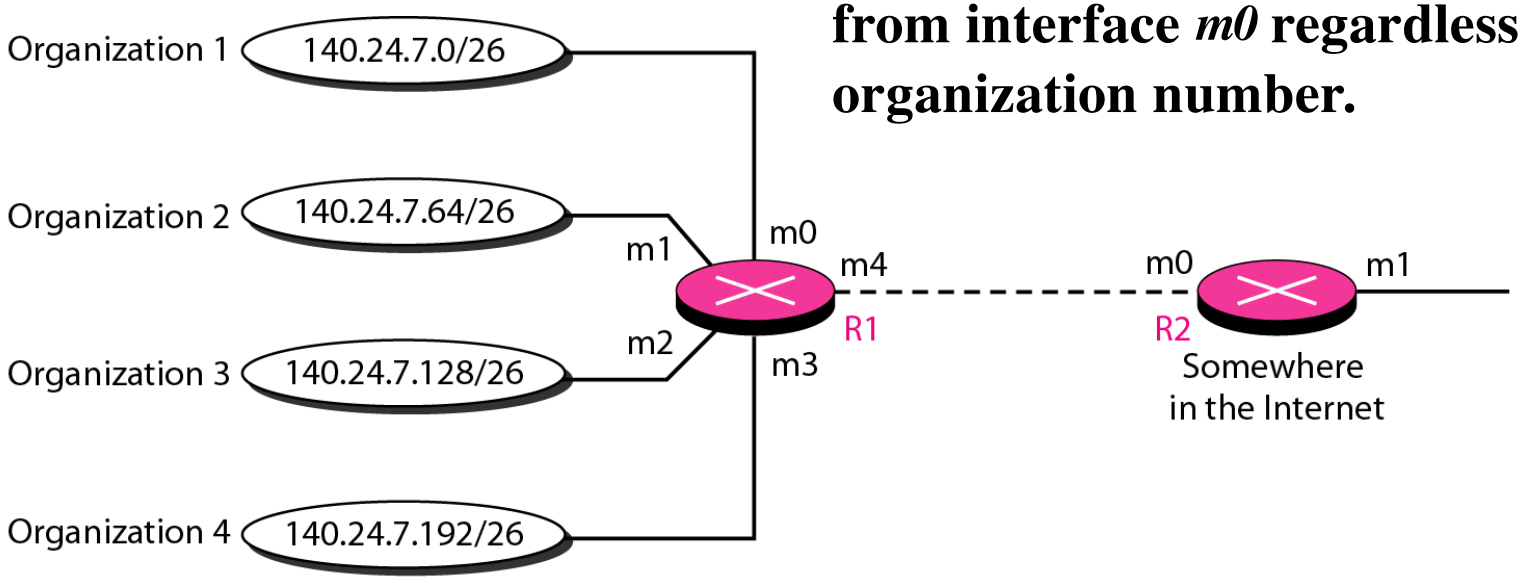
- ***This time all masks are applied, one by one, to the destination address, but no matching network address is found.***
- ***When it reaches the end of the table, the module gives the next-hop address 180.70.65.200 and interface number m2 to ARP.***
- ***This is probably an outgoing packet that needs to be sent, via the default router, to someplace else in the Internet.***

Address Aggregation

- Classless addressing increases the number of routing table entries.
- This is so because the classless addressing is to divide up the whole address space into manageable blocks.
- The increased size of the table results in an increase in the amount of time needed to search the table.
- To alleviate the problem, the idea of address aggregation was designed.

Address aggregation

For R2, any packet with destination 140.24.7.0 to 140.24.7.255 is sent out from interface *m0* regardless of the organization number.



Mask	Network address	Next-hop address	Interface
/26	140.24.7.0	----------	m0
/26	140.24.7.64	----------	m1
/26	140.24.7.128	----------	m2
/26	140.24.7.192	----------	m3
/0	0.0.0.0	Default	m4

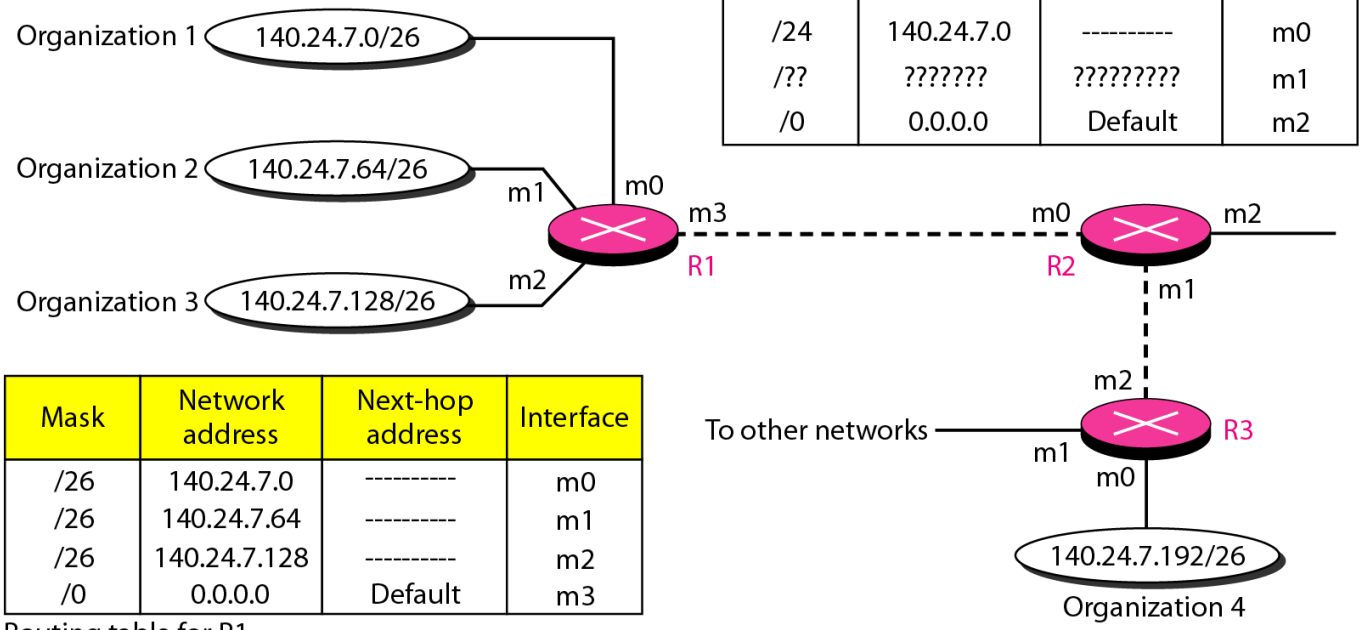
Routing table for R1

Mask	Network address	Next-hop address	Interface
/24	140.24.7.0	----------	m0
/0	0.0.0.0	Default	m1

Routing table for R2

Longest mask matching

Suppose a packet arrives for organization 4 with destination address 140.24.7.200. The first mask at router R2 is applied, which gives the network address 140.24.7.192.



Routing table for R2

Mask	Network address	Next-hop address	Interface
/26	140.24.7.192	----------	m1
/24	140.24.7.0	----------	m0
/??	???????	?????????	m1
/0	0.0.0.0	Default	m2

Mask	Network address	Next-hop address	Interface
/26	140.24.7.0	----------	m0
/26	140.24.7.64	----------	m1
/26	140.24.7.128	----------	m2
/0	0.0.0.0	Default	m3

Routing table for R1

What happens if one of the organizations in Figure is not geographically close to the other three?

Mask	Network address	Next-hop address	Interface
/26	140.24.7.192	----------	m0
/??	???????	?????????	m1
/0	0.0.0.0	Default	m2

Routing table for R3

Common fields in a routing table

Mask	Network address	Next-hop address	Interface	Flags	Reference count	Use
.....	.....	.....	.....	.....	.....	.....

- **Flags:**
  - This field defines up to five flags.
  - Flags are on/off switches that signify either presence or absence.
  - The five flags:
    - **U (up):** The router is up and running
    - **G (gateway):** The destination is in another network
    - **H (host-specific):** The entry in the network address field is a host specific address.
    - **D (added by redirection) and M (modified by redirection):**Routing information for this destination has been added/modified to host routing table by a redirection messages from ICMP

Common fields in a routing table

- **Reference count:**
  - This field gives the number of users of this route at the moment.
    - For example, if five people at the same time are connecting to the same host from this router, the value of this column is 5.
- **Use:**
  - This field shows the number of packets transmitted through this router for the corresponding destination.

Example

- ***One utility that can be used to find the contents of a routing table for a host or router is netstat in UNIX or LINUX.***
- ***We use two options: r and n.***
  - ***The option r indicates that we are interested in the routing table,***
  - ***the option n indicates that we are looking for numeric addresses.***
- ***Note that this is a routing table for a host, not a router.***

Example (continued)

$ netstat -rn

Kernel IP routing table

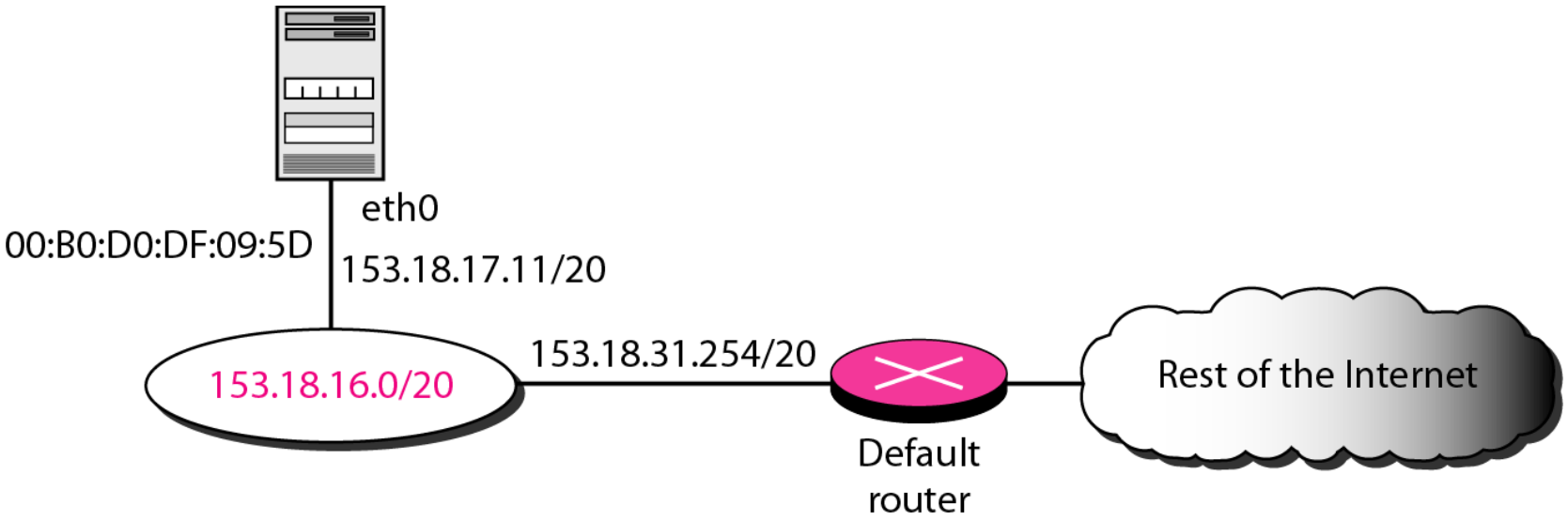
Destination	Gateway	Mask	Flags	Iface
153.18.16.0	0.0.0.0	255.255.240.0	U	eth0
127.0.0.0	0.0.0.0	255.0.0.0	U	lo
0.0.0.0	153.18.31.254	0.0.0.0	UG	eth0

- ***The destination column here defines the network address.***
- ***The term gateway used by UNIX is synonymous with router.***
- ***This column actually defines the address of the next hop. The value 0.0.0.0 shows that the delivery is direct.***
- ***The last entry has a flag of G, which means that the destination can be reached through a router (default router). The Iface defines the interface.***

Example (continued)

More information about the IP address and physical address of the server can be found by using the ifconfig command on the given interface (eth0).

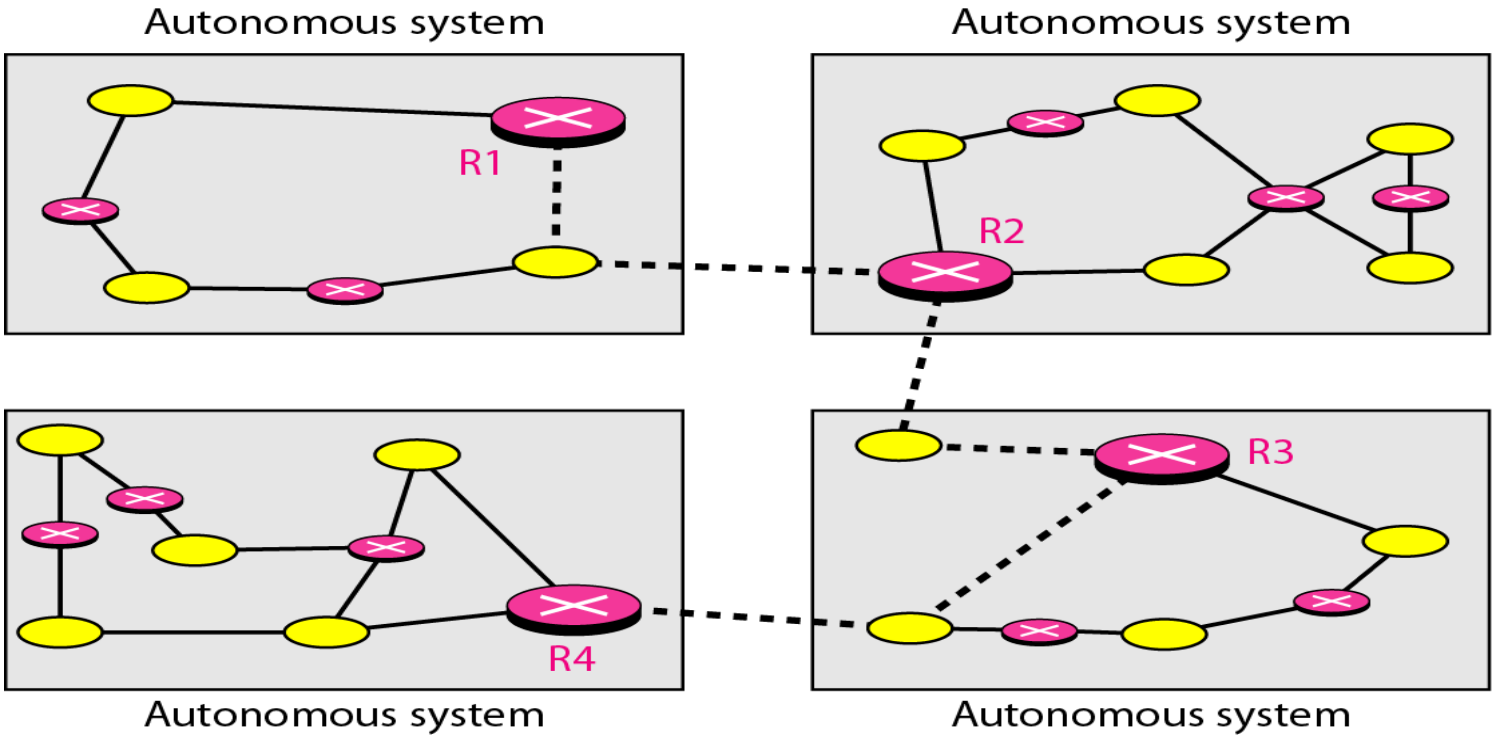
```
$ ifconfig eth0
eth0   Link encap:Ethernet  HWaddr 00:B0:D0:DF:09:5D
inet addr:153.18.17.11  Bcast:153.18.31.255  Mask:255.255.240.0
. . .
```



Unicast Routing Protocol

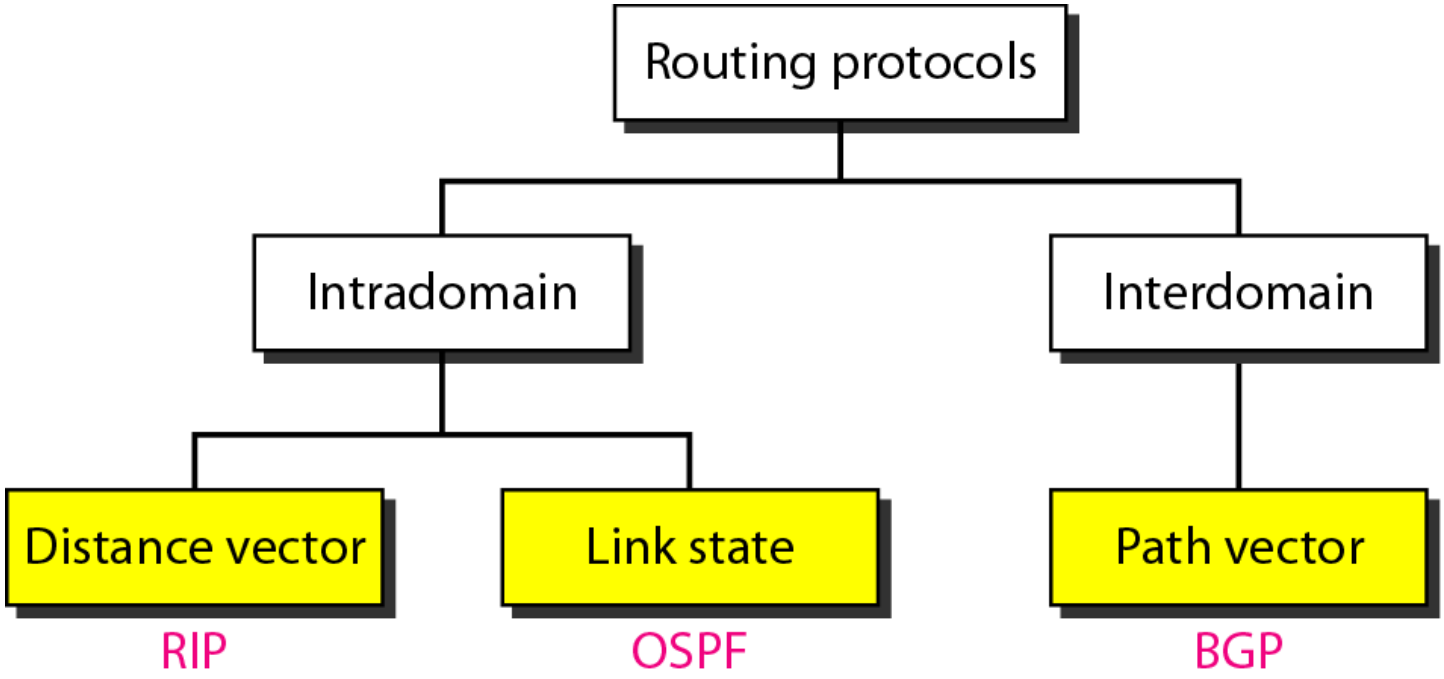
- *A routing table can be either static or dynamic.*
- *A static table is one with manual entries.*
- *A dynamic table is one that is updated automatically when there is a change somewhere in the Internet.*
- *A routing protocol is a combination of rules and procedures that lets routers in the Internet inform each other of changes.*

Autonomous systems



- **An autonomous system (AS) is a group of networks and routers under the authority of a single administration.**
- **Routing inside an autonomous system is referred to as *intradomain routing.***
- **Routing between autonomous systems is referred to as *interdomain routing.***

Popular routing protocols

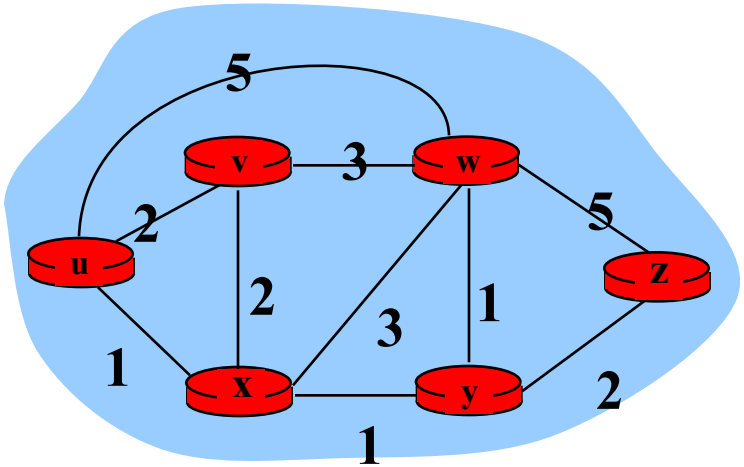


RIP: Routing Information Protocol
OSPF: Open Shortest Path Path
BGP: Border Gateway Protocol

An Internet as a graph

- Each router is considered as a node
- Each network between a pair of router is an edge
- We can consider the graph as a weighted graph
  - Each edge is associated with a cost
  - The cost of an edges has different interpretation in different routing protocols

Graph abstraction

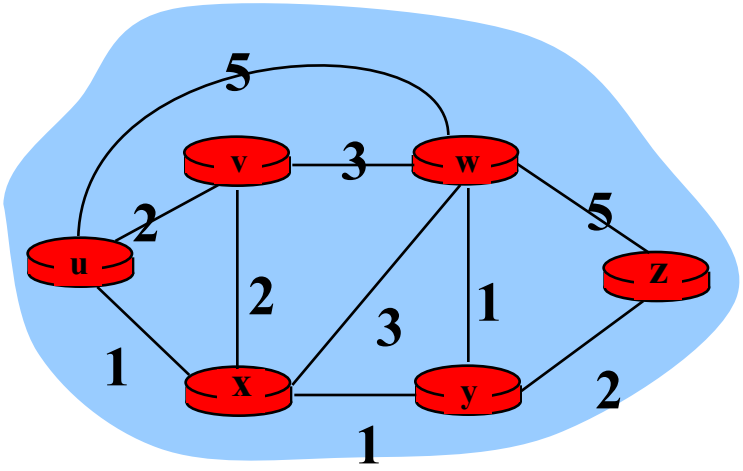


Graph: G = (N,E)

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)

Graph abstraction: costs



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

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

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

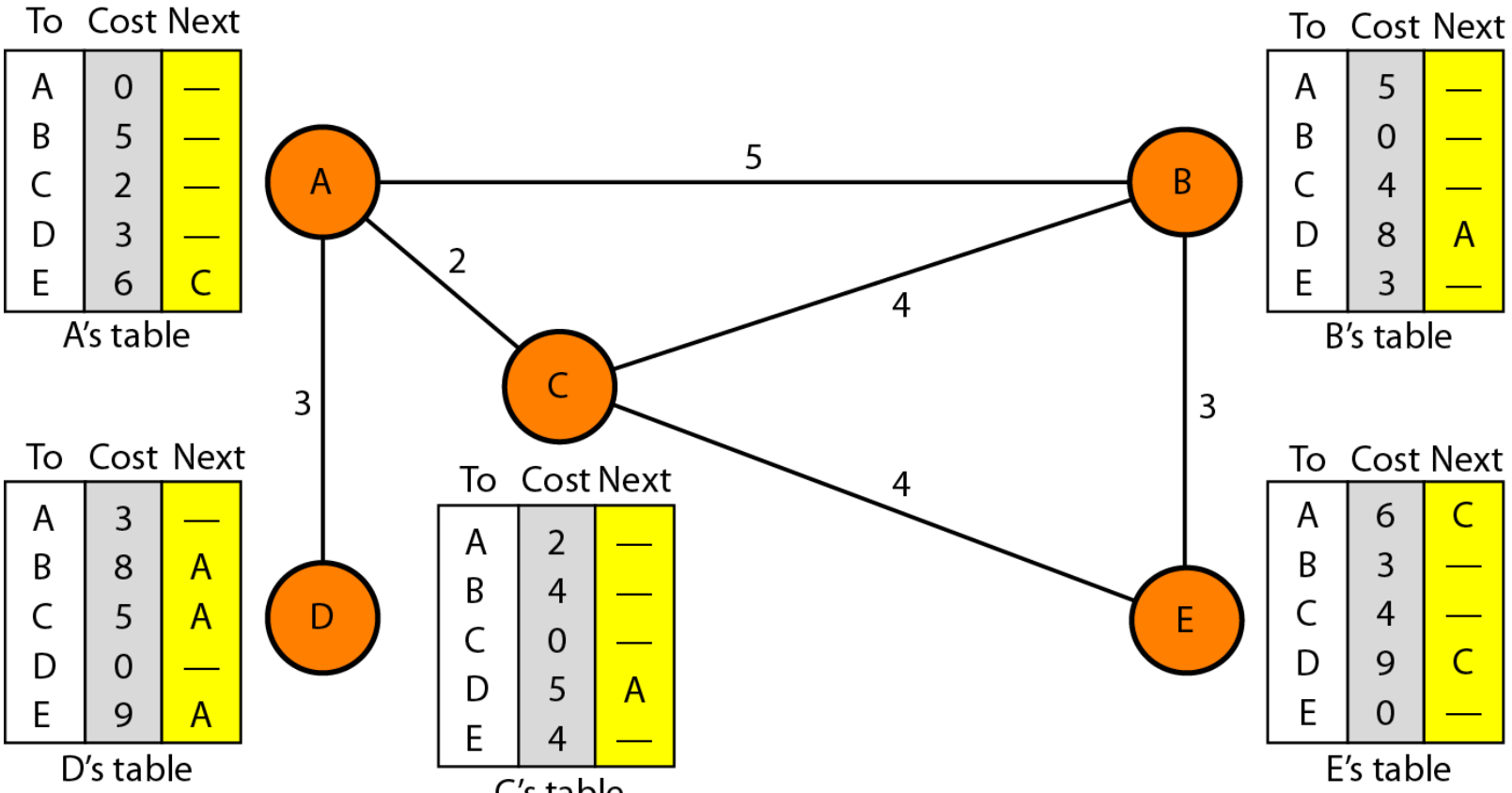
Cost of path (x1, x2, x3,…, xp) = c(x1,x2) + c(x2,x3) + … + c(xp-1,xp)

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

Routing algorithm: Algorithm that finds least-cost path

Distance vector routing tables

- In distance vector routing, the least-cost route between any two nodes is the route with minimum distance.
- In this protocol, as the name implies, *each node maintains a vector (table) of minimum distances to every node*.

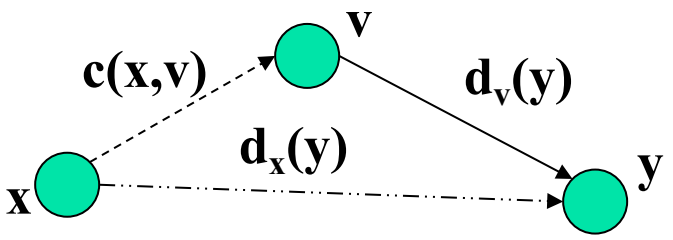


Distance Vector Algorithm (1)

Bellman-Ford Equation (dynamic programming)

Define

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

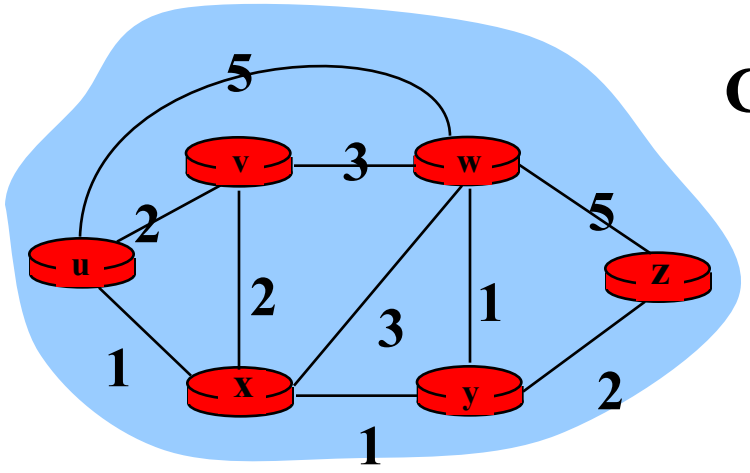


Then

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

where *min* is taken over *all neighbors of x*

Bellman-Ford example (2)



Clearly, $d_v(z) = 5$, $d_x(z) = 3$, $d_w(z) = 3$

B-F equation says:

$$d_u(z) = \min \{ c(u,v) + d_v(z), \; c(u,x) + d_x(z), \; c(u,w) + d_w(z) \}$$
$$= \min \{2 + 5, \; 1 + 3, \; 5 + 3\} = 4$$

Node that achieves minimum is next hop in shortest path ➜ forwarding table

Distance Vector Algorithm (3)

- $D_x(y)$ = Estimate of least cost from x to y
- Distance vector: $\mathbf{D}_x = [D_x(y): y \in N]$
- Node x knows cost to each neighbor v: $c(x,v)$
- Node x maintains $\mathbf{D}_x = [D_x(y): y \in N]$
- 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)

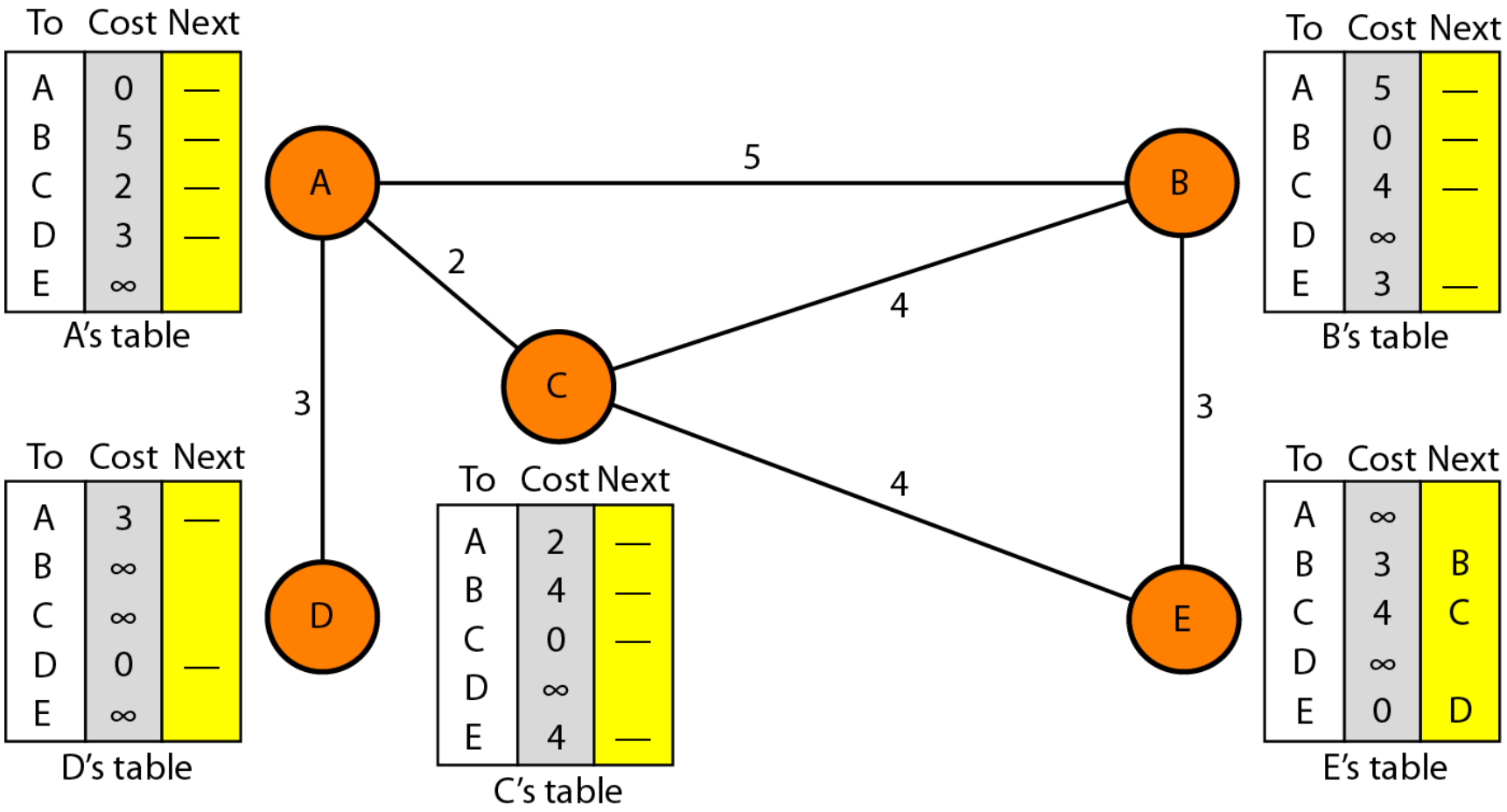
Basic idea:

- Each node periodically sends its own distance vector estimate to neighbors
- When node a node x receives new DV estimate from neighbor, it updates its own DV using B-F equation:

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

- **Under minor, natural conditions, the estimate $D_x(y)$ *converge the actual least cost* $d_x(y)$**

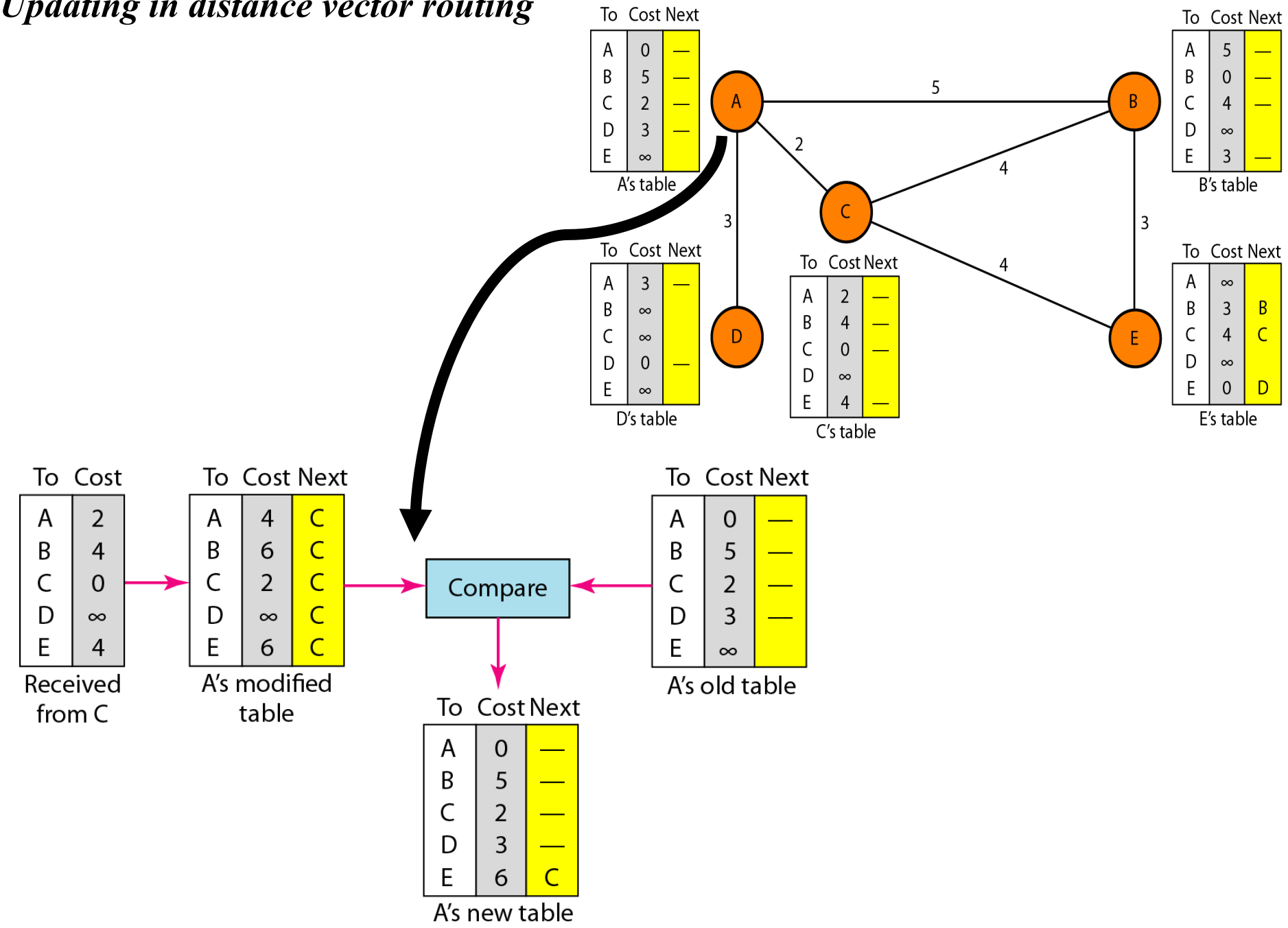
Initialization of tables in distance vector routing



Note

In distance vector routing, each node shares its routing table with its immediate neighbors periodically and when there is a change.

Updating in distance vector routing



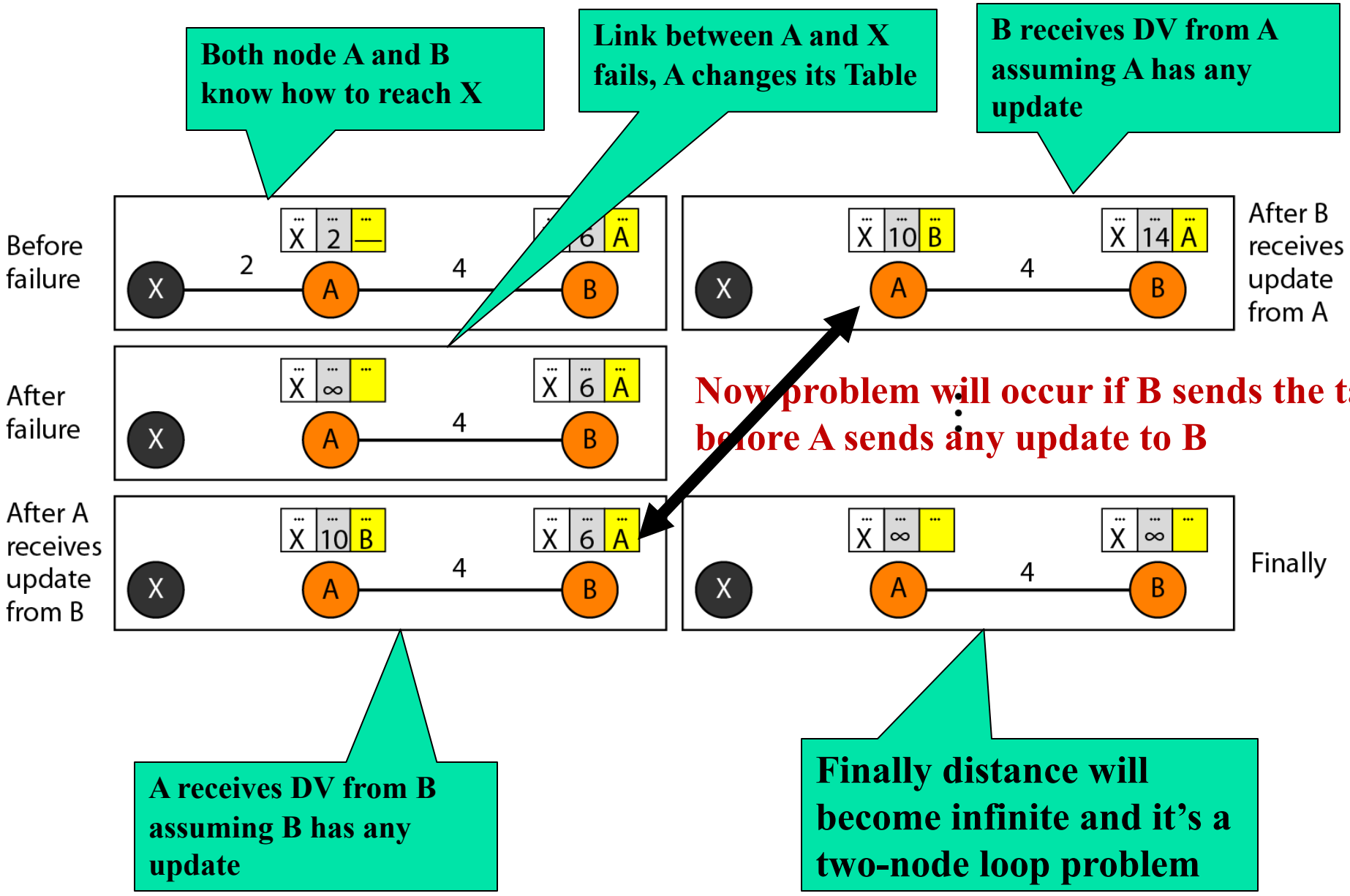
Distance Vector: link cost changes

Link cost changes:

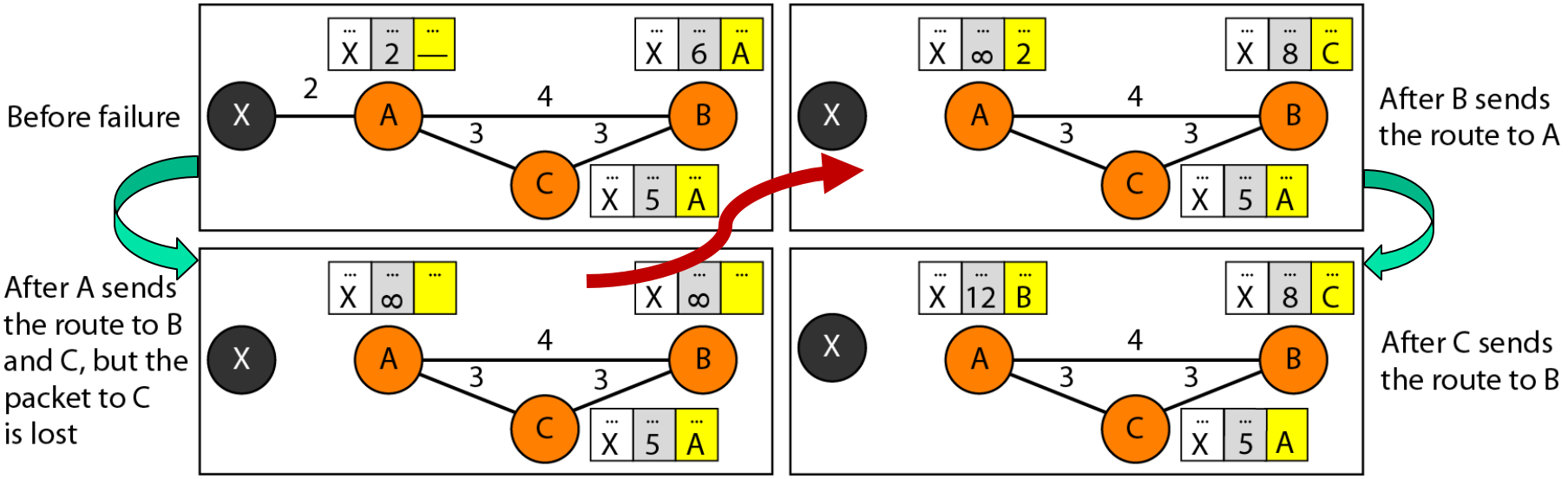
- **Node detects local link cost change**
- **Updates routing info, recalculates distance vector**
- **If DV changes, notify neighbors**

- **Good news travels fast**
- **Bad news travels slow - *"count to infinity"* problem!**

Two-node instability



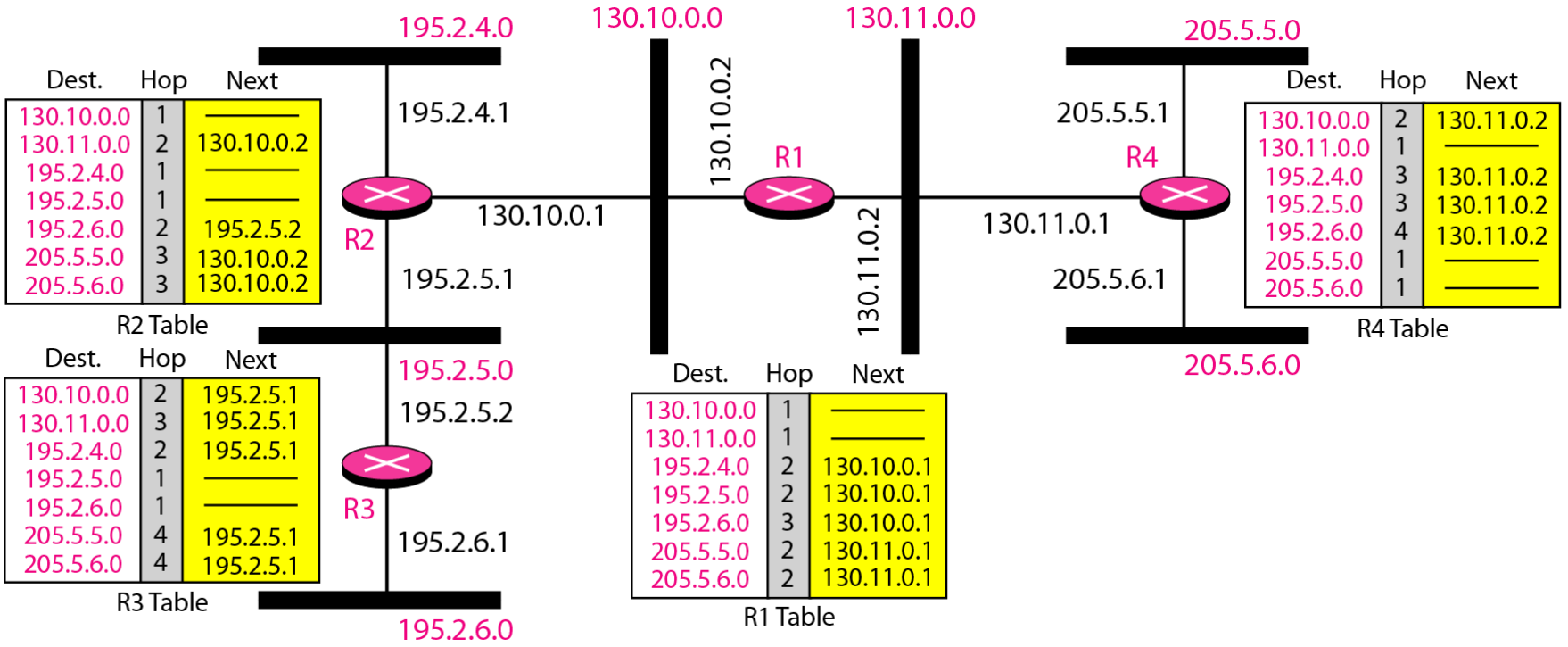
Three-node instability



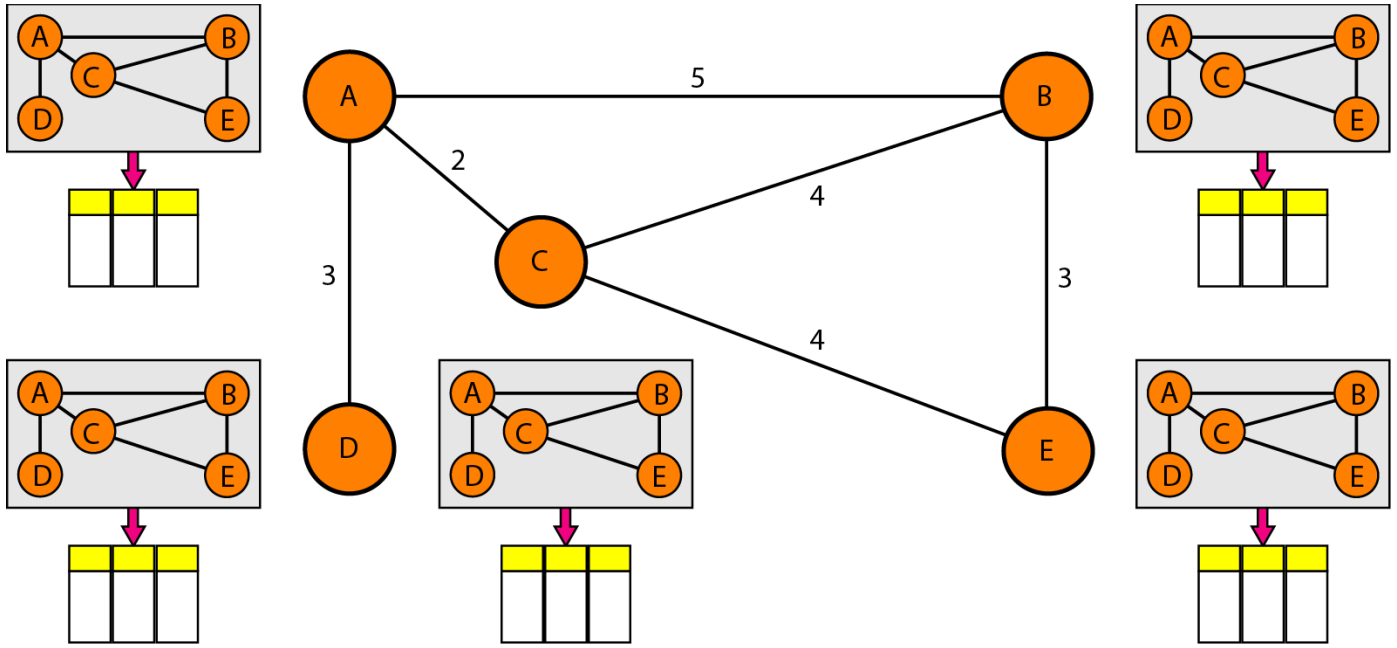
Routing Information Protocol (RIP)

- Most widely used intradomain routing protocol based on DVR algorithm
- This is used for an autonomous system
- RIP implements distance vector routing directly with some considerations:
  - In an autonomous system, we are dealing with routers and networks (links). The routers have routing tables; networks do not.
  - The destination in a routing table is a network, which means the first column defines a network address.
  - The metric in RIP is called a hop count.
  - Infinity is defined as 16, which means that any route in an autonomous system using RIP cannot have more than 15 hops.
  - The next-node column defines the address of the router to which the packet is to be sent to reach its destination.

Example of a domain using RIP

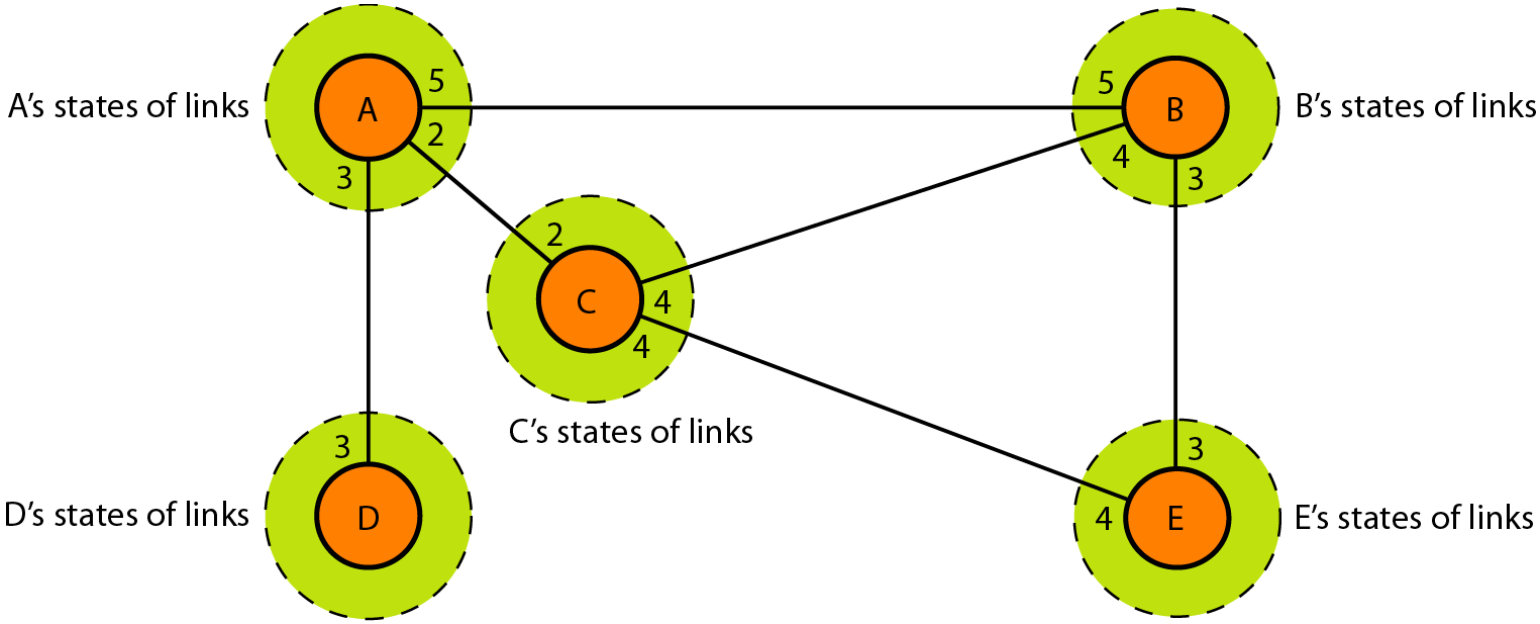


Concept of link state routing



- Each node needs to have a complete map of the network
  - Knows state of each link
  - Collection of states of each link is called the link state database (LSDB)
    - There is only one LSDB for whole internet
  - Each node copy this LSDB to create the least-cost path in between a pair of nodes.

Link state knowledge



Building Routing Table:

- **Four sets of actions are required to ensure that each node has the routing table showing the least-cost node to every other node.**
  - Creation of the states of the links by each node, called the link state packet (LSP).
  - Dissemination of LSPs to every other router, called **flooding,** in an efficient and reliable way.
  - Formation of a shortest path tree for each node.
  - Calculation of a routing table based on the shortest path tree.

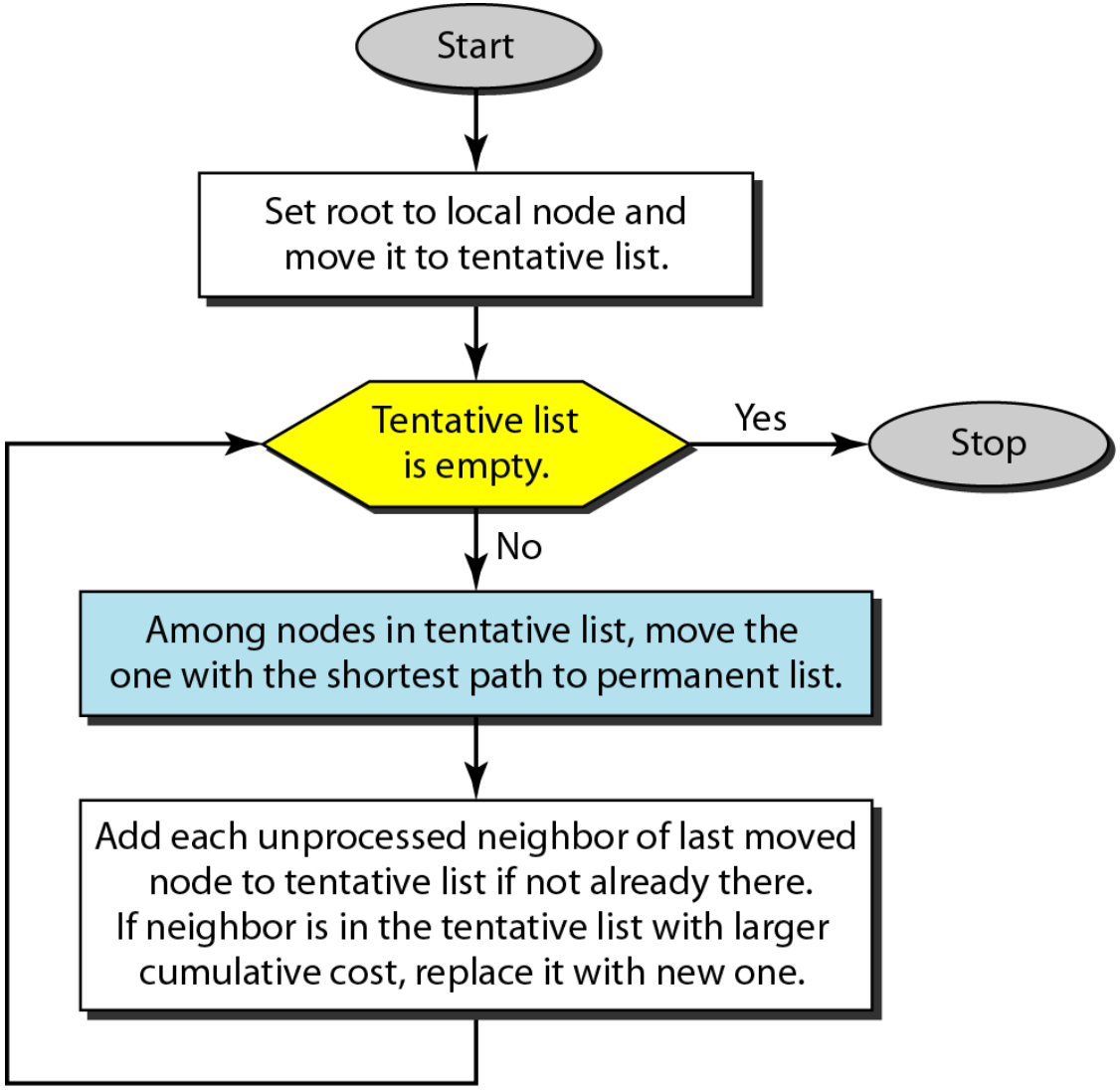
Flooding of LSPs

- The creating node sends a copy of the LSP out of each interface.
- A node that receives an LSP compares it with the copy it may already have.
- If the newly arrived LSP is older than the one it has (found by checking the sequence number), it discards the LSP.
- If it is newer, the node does the following:
  - It discards the old LSP and keeps the new one.
  - It sends a copy of it out of each interface except the one from which the packet arrived.
  - This guarantees that flooding stops somewhere in the domain (where a node has only one interface).

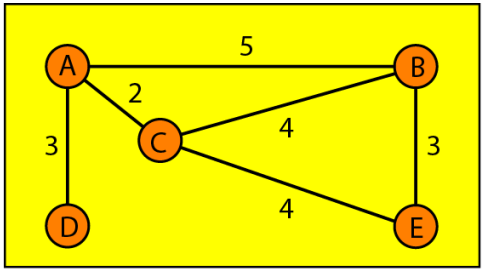
Formation of Shortest Path Tree

- After receiving all LSPs, each node will have a copy of the whole topology.
- Each node runs Dijkstra's algorithm to determine the shortest path between every other nodes.
- The Dijkstra algorithm also creates a shortest path tree from a graph.

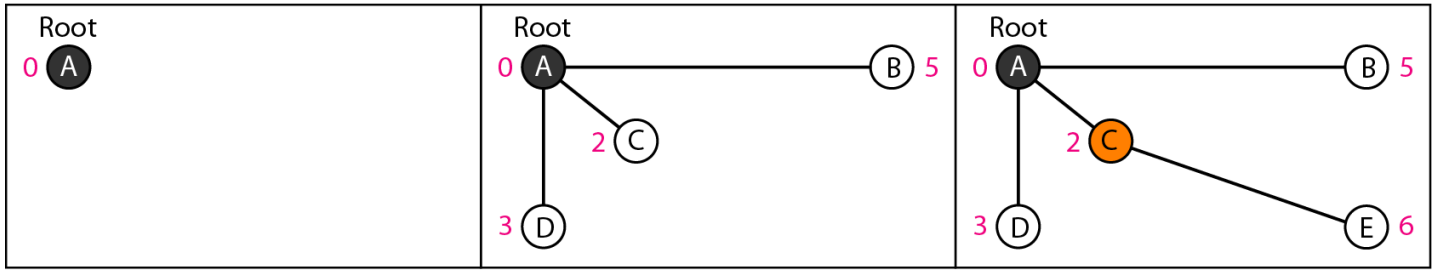
Dijkstra's Algorithm



Example of formation of shortest path tree



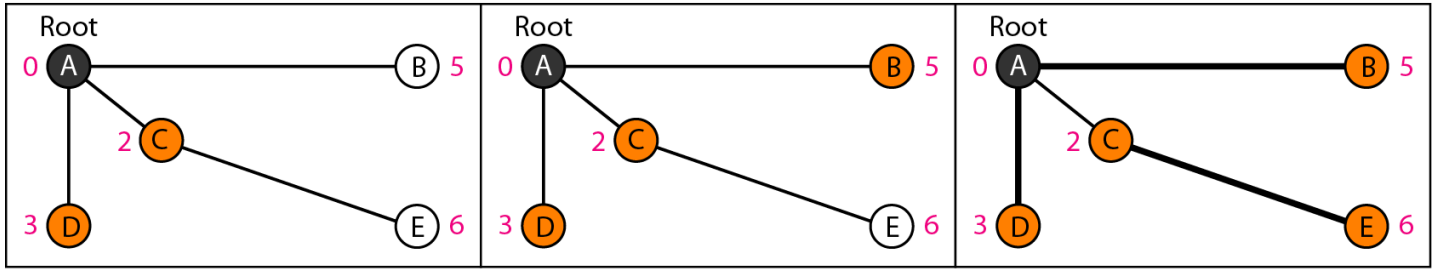
Topology



1. Set root to A and move A to tentative list.

2. Move A to permanent list and add B, C, and D to tentative list.

3. Move C to permanent and add E to tentative list.



4. Move D to permanent list.

5. Move B to permanent list.

6. Move E to permanent list (tentative list is empty).

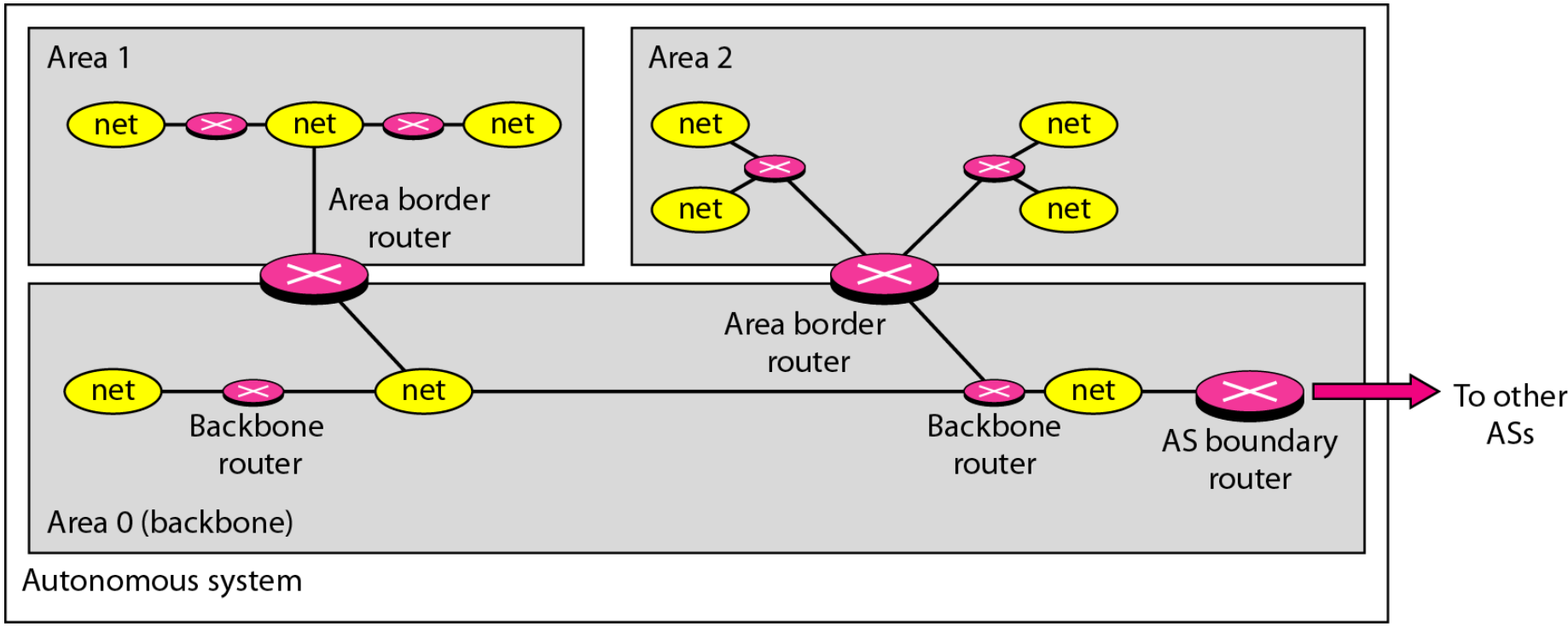
Routing table for node A

Node	*Cost*	*Next Router*
A	0	—
B	5	—
C	2	—
D	3	—
E	6	C

Open Shortest Path First (OSPF)

- The Open Shortest Path First or OSPF protocol is an intradomain routing protocol based on link state routing.
- Its domain is also an autonomous system.
- The OSPF protocol allows the administrator to assign a cost, called the metric, to each route.
- The metric can be based on a type of service (minimum delay, maximum throughput, and so on).
  - As a matter of fact, a router can have multiple routing tables, each based on a different type of service.

Areas in an autonomous system



- **OSPF divides an autonomous system into areas.**
- **An area is a collection of networks, hosts, and routers all contained within an autonomous system.**
- **All networks inside an area must be connected.**

Comparison of LS and DV algorithms

Message complexity

- LS: with n nodes, E links, O(nE) msgs sent
- DV: exchange between neighbors only
  - convergence time varies

Speed of Convergence

- LS: $O(n^2)$ algorithm requires O(nE) msgs
  - may have oscillations
- DV: convergence time varies
  - may be routing loops
  - count-to-infinity problem

Robustness: what happens if router malfunctions?

LS:

- node can advertise incorrect *link* cost
- each node computes only its *own* table

DV:

- DV node can advertise incorrect *path* cost
- each node's table used by others
  - error propagate thru network

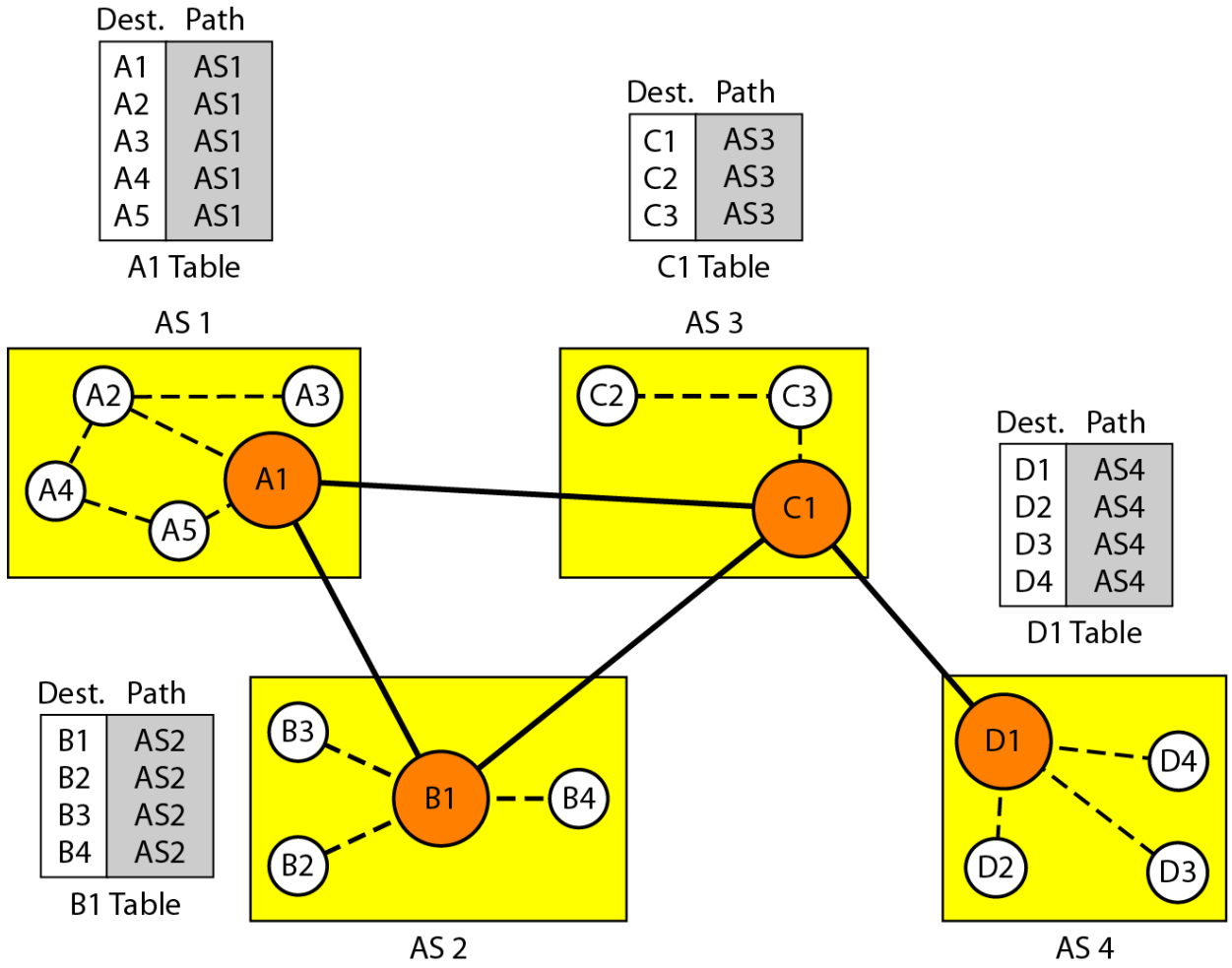
Path Vector Routing

- Distance vector and link state routing are both intradomain routing protocols.
- They can be used inside an autonomous system, but not between autonomous systems.
- These two protocols are not suitable for interdomain routing mostly because of scalability.
- Both of these routing protocols become intractable when the domain of operation becomes large.

Path Vector Routing

- In path vector routing, we assume that there is one node (there can be more) in each autonomous system that acts on behalf of the entire autonomous system.
  - Call it the **speaker node.**
- The speaker node in an AS creates a routing table and advertises it to speaker nodes in the neighbouring ASs.
- The idea is the same as for distance vector routing except that only speaker nodes in each AS can communicate with each other.
- A speaker node advertises the path, not the metric of the nodes, in its autonomous system or other autonomous systems.

Initial routing tables in path vector routing



Stabilized tables for three autonomous systems

- A speaker in an autonomous system shares its table with immediate neighbours.
- When a speaker node receives a two-column table from a neighbour,
  - it updates its own table by adding the nodes that are not in its routing table and adding its own autonomous system and the autonomous system that sent the table.

Dest.	Path
A1 … A5	AS1
B1 … B4	AS1-AS2
C1 … C3	AS1-AS3
D1 … D4	AS1-AS2-AS4

A1 Table

Dest.	Path
A1 … A5	AS2-AS1
B1 … B4	AS2
C1 … C3	AS2-AS3
D1 … D4	AS2-AS3-AS4

B1 Table

Dest.	Path
A1 … A5	AS3-AS1
B1 … B4	AS3-AS2
C1 … C3	AS3
D1 … D4	AS3-AS4

C1 Table

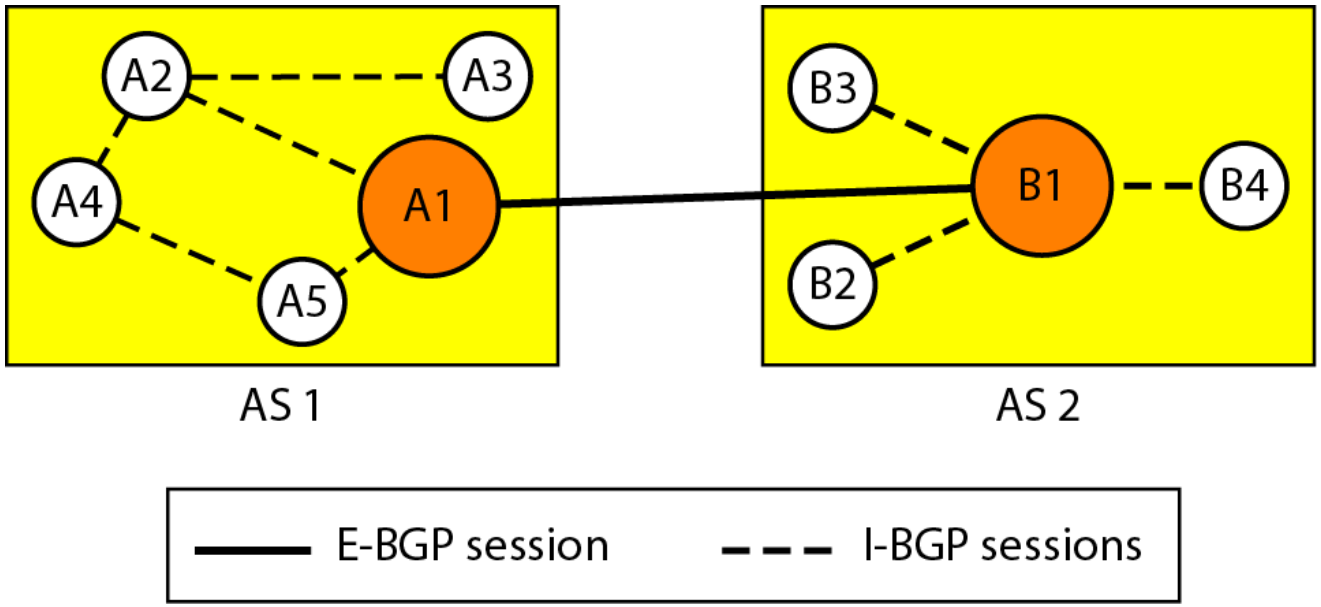
Dest.	Path
A1 … A5	AS4-AS3-AS1
B1 … B4	AS4-AS3-AS2
C1 … C3	AS4-AS3
D1 … D4	AS4

D1 Table

Border Gateway Protocol (BGP)

- Border Gateway Protocol (BGP) is an interdomain routing protocol using path vector routing.
- It first appeared in 1989 and has gone through four versions.
- The exchange of routing information between two routers using BGP takes place in a session.
- A session is a connection that is established between two BGP routers only for the sake of exchanging routing information.
- To create a reliable environment, BGP uses the services of TCP.
  - BGP sessions are sometimes referred to as *semi permanent connections.*

Internal and external BGP sessions



- The session established between AS 1 and AS2 is an E-BGP session.
- The two speaker routers exchange information they know about networks in the Internet.
- However, these two routers need to collect information from other routers in the autonomous systems.
- This is done using I-BGP sessions.