Chapter 14

Unicast Routing Protocols: RIP, OSPF, and BGP

Objectives

Upon completion you will be able to:

- *Distinguish between intra and interdomain routing*
- *Understand distance vector routing and RIP*
- *Understand link state routing and OSPF*
- *Understand path vector routing and BGP*

Introduction

- An internet is a combination of networks connected by routers
- How to pass a packet from source to destination ?
  - Which of the available pathways is the optimum pathway ?
- Depends on the metric
  - **Metric**: a cost assigned for passing through a network
  - A router should choose the route with the smallest metric

Introduction (Cont.)

- The metric assigned to each network depends on the type of protocol
  - RIP (Routing Information Protocol)
    - Treat each network as equal
    - The cost of passing through each network is the same: one hop count
  - Open Shortest Path First (OSPF)
    - Allow administrator to assign a cost for passing through a network based on the *type of serviced* required
      - For example, maximum throughput or minimum delay
  - Border Gateway Protocol (BGP)
    - The criterion is the policy, which can be set by the administrator

Introduction (Cont.)

- Routing table can be *static* or *dynamic*
  - An internet needs dynamic routing tables

- Dynamic routing table is achieved by the routing protocols

14.1 INTRA- AND INTERDOMAIN ROUTING

Routing inside an autonomous system is referred to as intradomain routing. Routing between autonomous systems is referred to as interdomain routing.

Interior and Exterior Routing

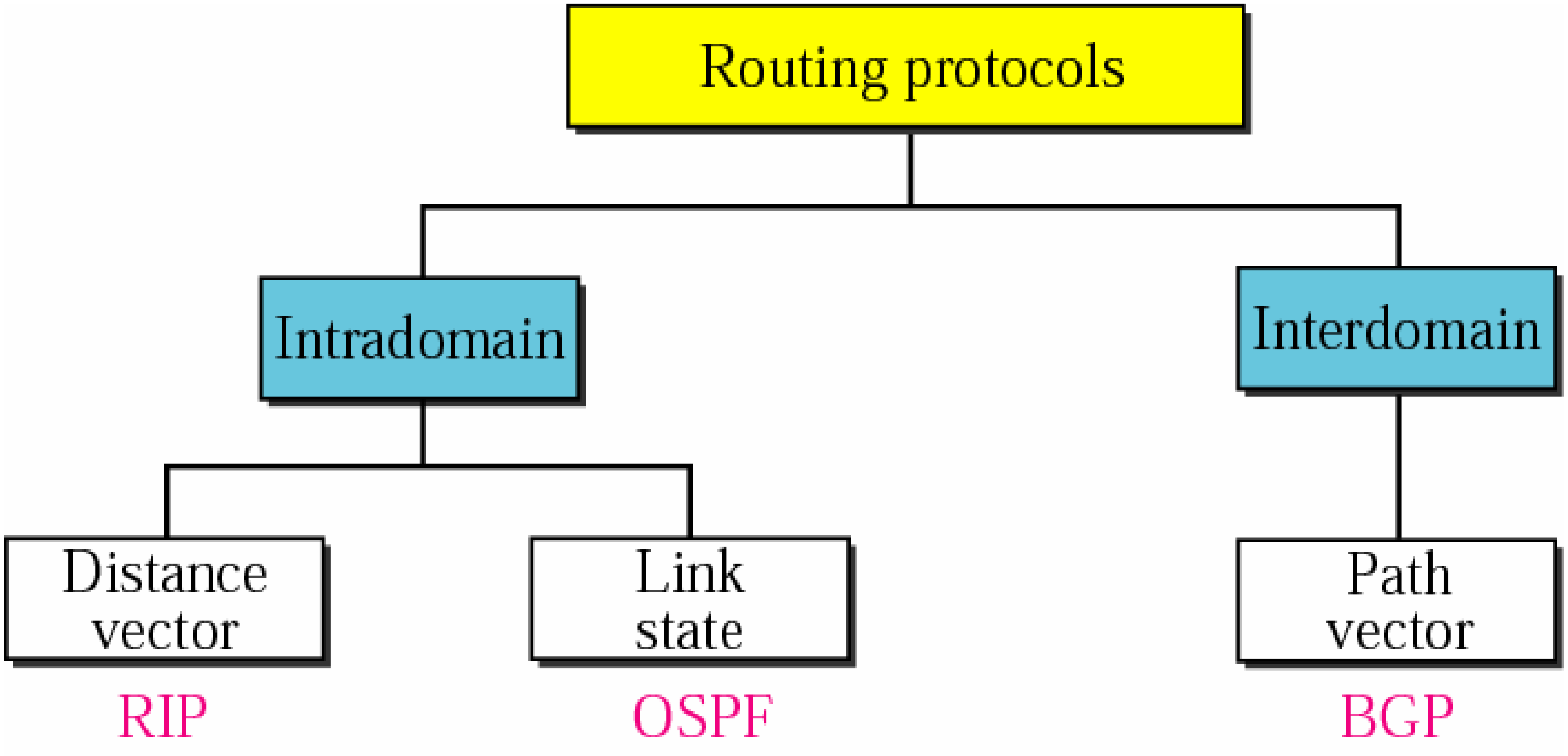
- An internet can be so large
  - One routing protocol cannot handle the task of updating routing table of all routers

- Thus, an internet is divided into ***autonomous systems (AS)***
  - AS is a group of networks and routers under the authority of a single administration

Intra- And Interdomain Routing

- ***intradomain routing***
  - Routing inside an autonomous system
  - Each AS can chose its own intradomain routing protocol
  - Examples: ***distance vector*** and ***link state***
- ***interdomain routing***
  - Routing between autonomous systems
  - Only one interdomain routing protocol is usually used between ASs
  - Examples: ***path vector***

Popular Routing Protocols



Intradomain Routing Algorithms

- Distance-vector routing algorithm
  - Classical *Distributed Bellman-Ford* algorithm
  - RIP (Routing Information Protocol)

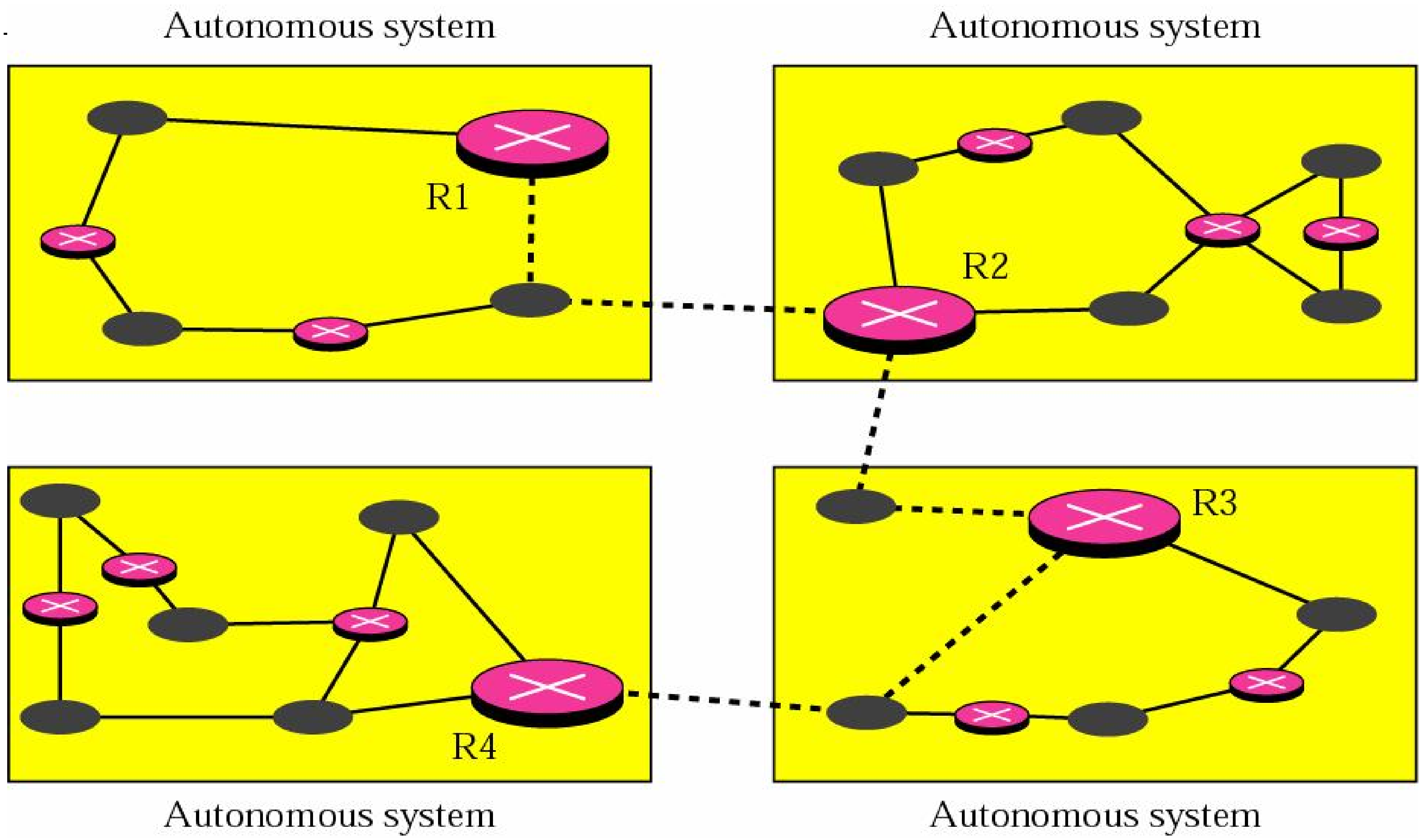
- Link-state routing algorithm
  - *Centralized* version of the shortest path computation
    - Every router has the *whole "picture"* of the internet
  - OSPF (Open Shortest Path First)

Example

- R1, R2, R3 and R4 use an intradomain and an interdomain routing protocol
- Solid thin lines
  - intradomain routing protocol
- Broken thick lines
  - interdomain routing protocol

Figure 13-2

Autonomous Systems



14.2 DISTANCE VECTOR ROUTING

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

The topics discussed in this section include:

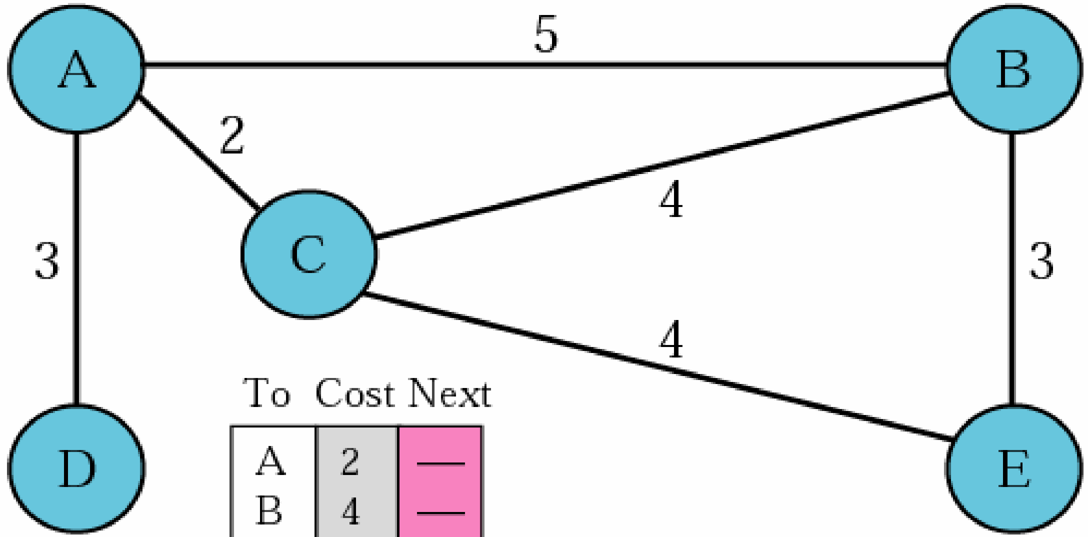
Initialization
Sharing
Updating
When to Share
Two-Node Loop Instability
Three-Node Instability

Distance Vector Routing

- The least cost route between any two nodes is the route with minimum distance.

- Each node maintains a vector (table) of minimum distances to every node

Distance Vector Routing Tables



To	Cost	Next
A	0	—
B	5	—
C	2	—
D	3	—
E	6	C

A's table

To	Cost	Next
A	5	—
B	0	—
C	4	—
D	8	A
E	3	—

B's table

To	Cost	Next
A	2	—
B	4	—
C	0	—
D	5	A
E	4	—

C's table

To	Cost	Next
A	3	—
B	8	A
C	5	A
D	0	—
E	9	A

D's table

To	Cost	Next
A	6	C
B	3	—
C	4	—
D	9	C
E	0	—

E's table

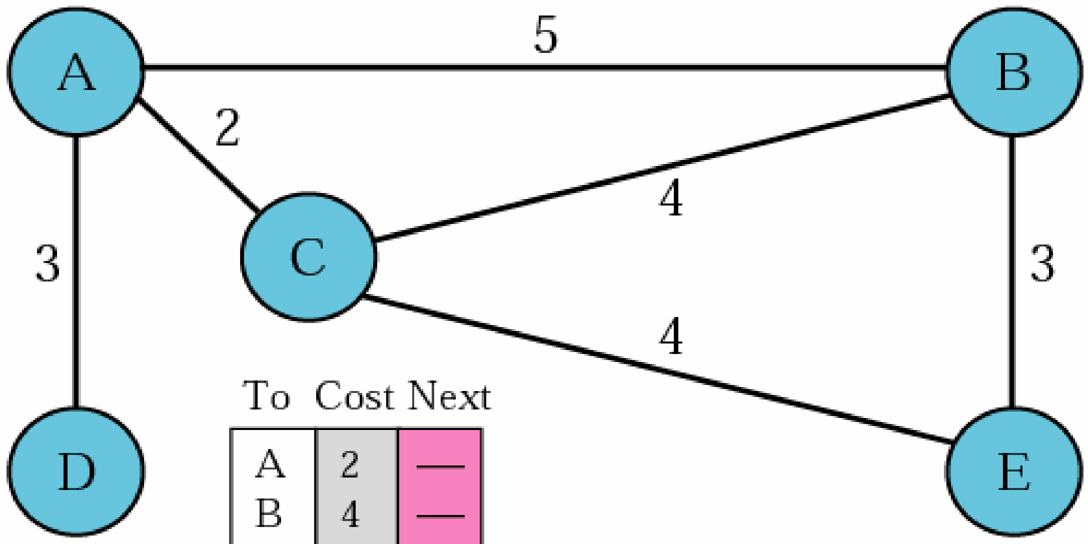
Initialization

- At the beginning
  - Each node can know only the distance between ***itself*** and its ***immediate neighbors***
  - We assume each node can send a message to the immediate neighbors and find the distance

Initialization of Tables in Distance Vector Routing

To	Cost	Next
A	0	—
B	5	—
C	2	—
D	3	—
E	∞	

A's Table



To	Cost	Next
A	5	—
B	0	—
C	4	—
D	∞	
E	3	—

B's Table

To	Cost	Next
A	3	—
B	∞	
C	∞	
D	0	—
E	∞	

D's Table

To	Cost	Next
A	2	—
B	4	—
C	0	—
D	∞	
E	4	—

C's Table

To	Cost	Next
A	∞	
B	3	B
C	4	C
D	∞	
E	0	D

E's Table

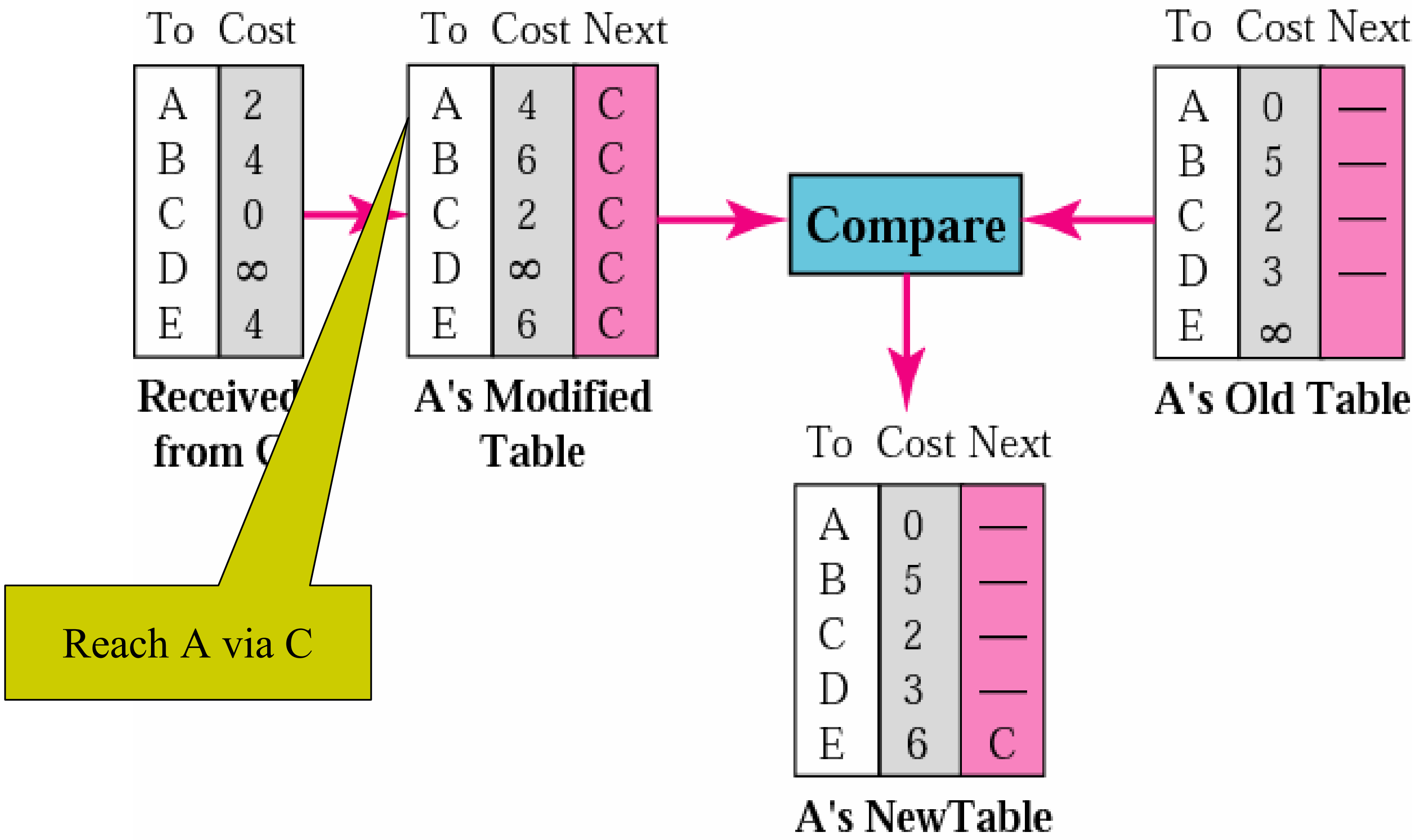
Sharing

- Idea of distance vector routing
  - Sharing of information between neighbors
  - ***In distance vector routing, each node shares its routing table with its immediate neighbors periodically and when there is a change***
- How much of the table must be shared ?
  - Send the entire table but contains only the first two columns
    - The third column must be changed

Updating

- *Receipt: a two-column table from a neighbor*
- *Add the cost between itself and the sending node to each value in the second column*
- *Repeat the following steps for each advertised destination*
  - *If (destination not in the routing table)*
    - *Add the advertised information to the table*
  - *Else*
    - *If (next-hop field is the same)*
      - *Replace retry in the table with the new advertised one*
    - *Else*
      - *If (advertised hop count smaller than one in the table)*
        - *Replace entry in the routing table*

Updating in Distance Vector Routing



When to Share

- The table is sent both ***periodically*** and when there is a ***change*** in the table
- ***Periodic update***
  - A node sends its routing table in a periodic update
  - Normally every 30 seconds
- ***Triggered update***
  - A node receives a table from a neighbor resulting in changes in its own table
  - A node detects some failure in the neighboring links which results in a distance change to infinity

Two-Node Loop Instability

- A problem with distance vector routing is ***instability***
  - A network using this protocol can become ***unstable***
- See the following table
  - 1. both node A and B know how to reach node X
  - 2. the link between A and X fails
    - Node A change its table
  - 3a. If node A can send its routing table to B immediately
    - Everything is fine
  - 3b. However, if node B sends its routing table to A first
    - Node A assumes that B has found a way to reach X
  - 4. A sends its new update to B and B also update its routing table
  - 5. B sends its new update to A and so on…***until the cost reach infinity***
  - 6. ***Then both A and B knows that the link is broken***

Two-Node Instability

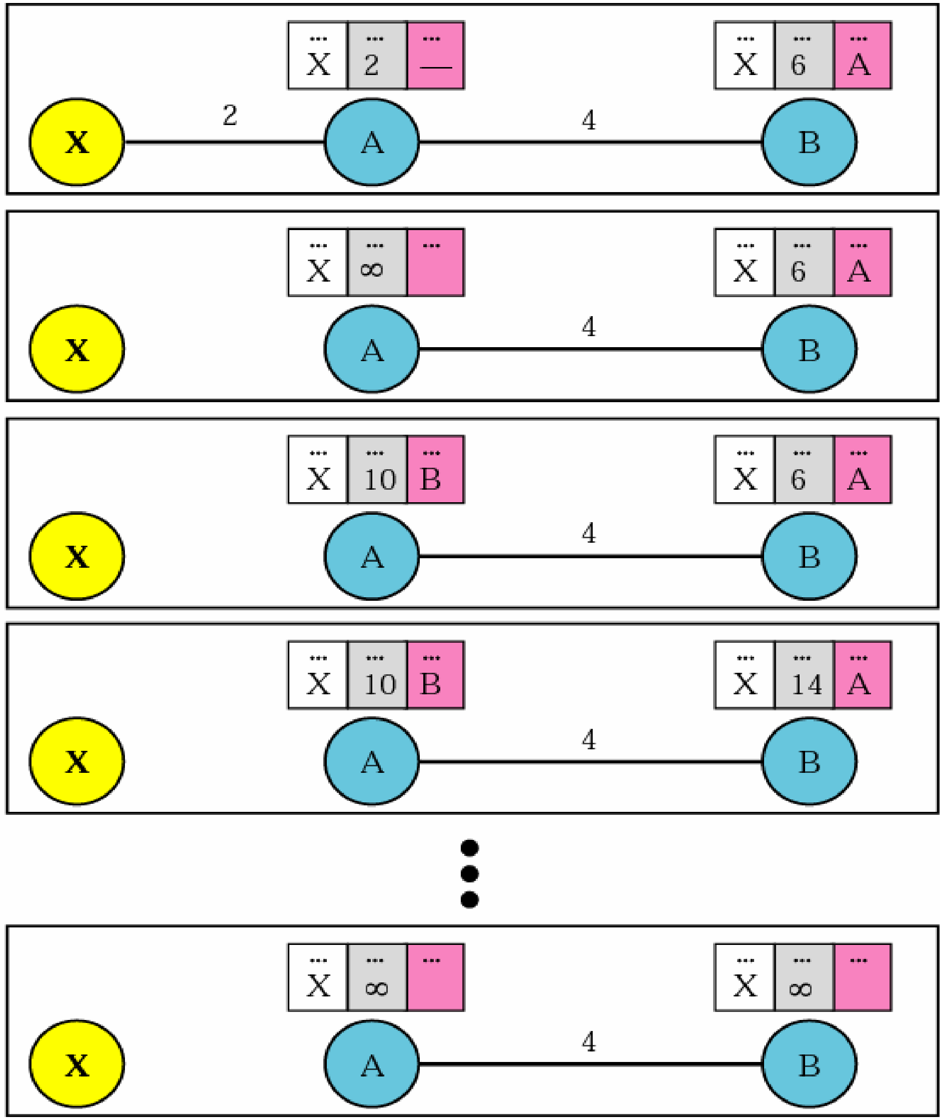
Before failure

After failure

After A receives update from B

After B receives update from A

Finally



Two-Node Loop Instability (Cont.)

- As a result, during the time before cost reaches infinity
  - A packet destined for X bounces between A and B
  - Create a ***two-node loop problem***
- Solutions
  - ***Defining infinity***
  - ***Split horizon***
  - ***Split horizon and poison reverse***

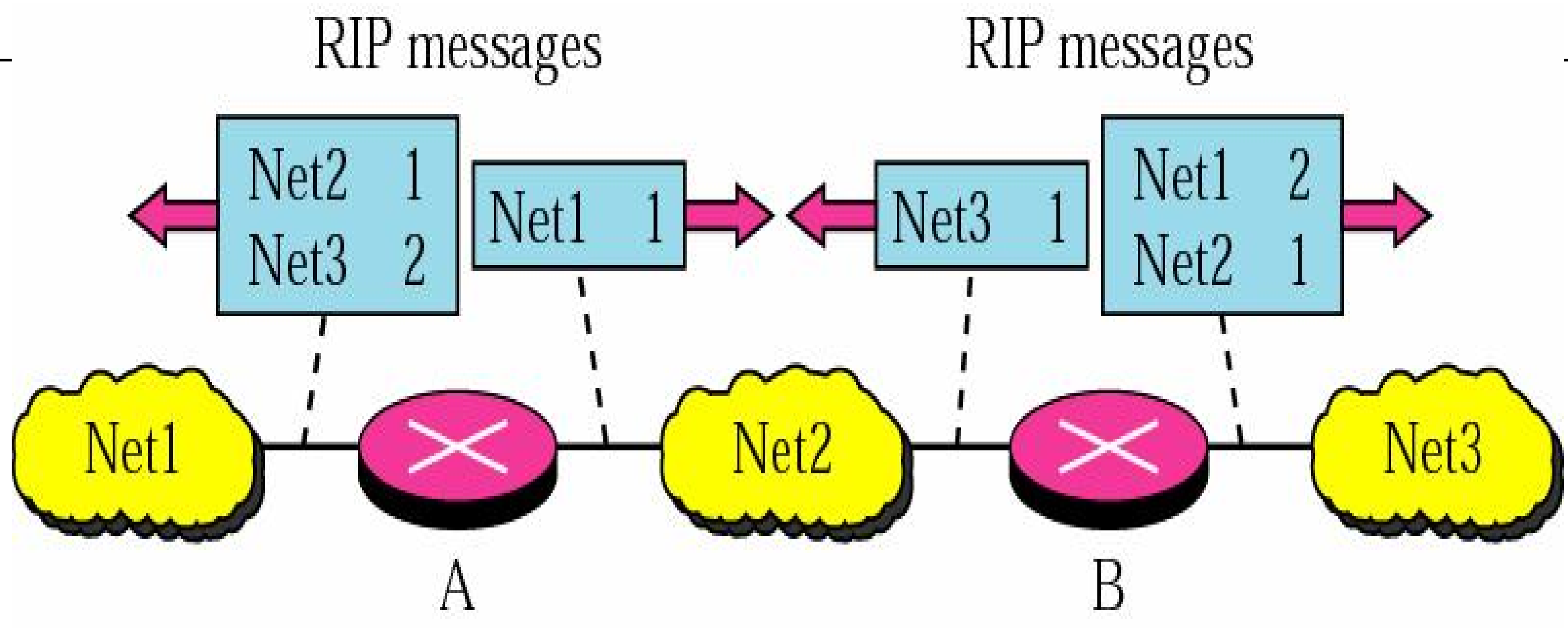
Defining Infinity

- Redefine infinity to a smaller number
  - Shorten the time of instability
- Most implementation define the distance between each node to be 1
  - Define 16 as infinity
- As a result
  - The distance vector scheme cannot be used in large system
  - The size of network, in each direction, can not exceed 15 hops

Split Horizon

- Do not flood the table through each interface and a router must distinguish between different interface
- If a router received route updating message from an interface
  - This same updated information must not be sent back through this interface
  - ***Since the information has come from the sending one***

Split Horizon



- B receives information about Net1 and Net2 through its left interface
- This information is updated and passed on through the right interface but not to the left

Split Horizon (Cont.)

- Thus, in the figure of two-node instability
  - Node B eliminates the last line of its routing table before it sends to A
    - Node A then keeps the value of infinity as the distance to X
  - Later when A sends its routing table to B
    - B then correct its routing table

- The system becomes stable ***after the first update***
  - Both node A and B know that X is not reachable

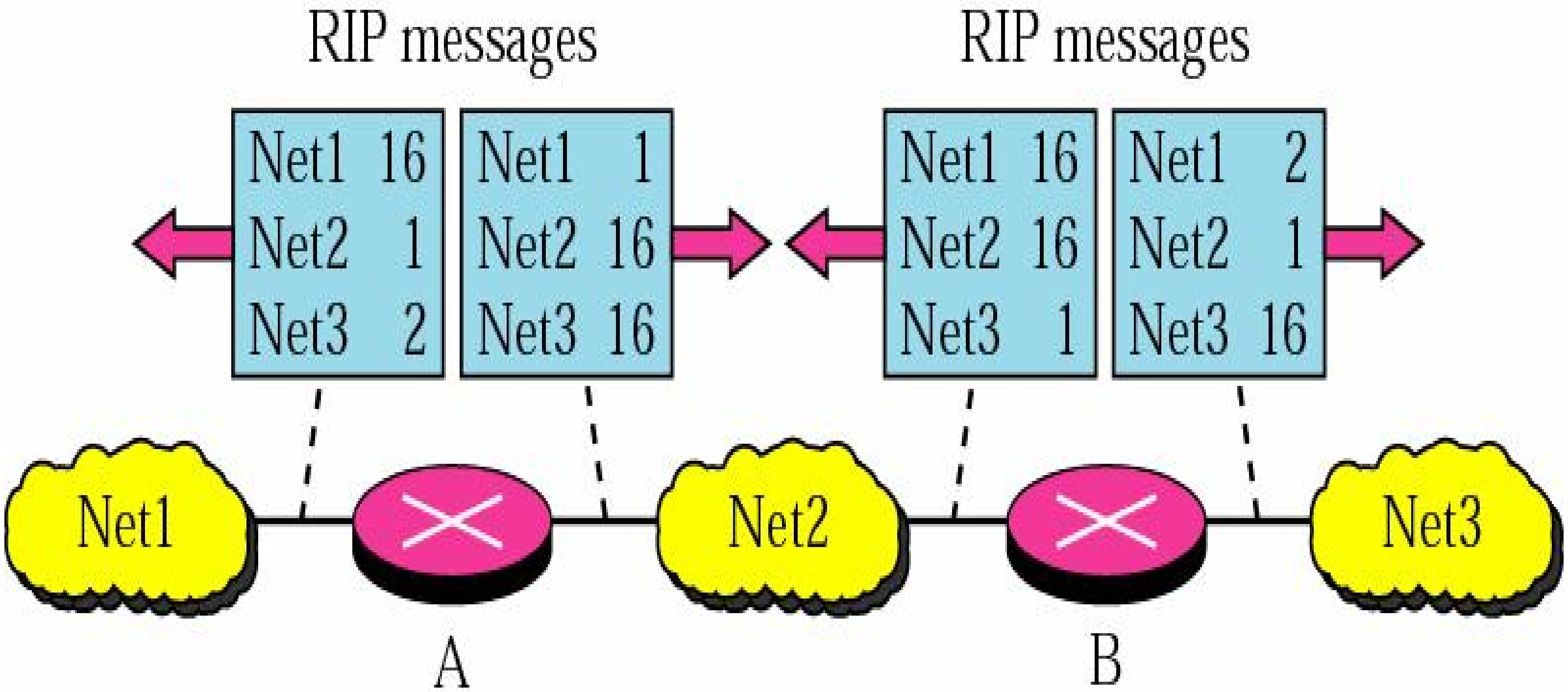
Split Horizon and Poison Reverse

- Drawback of split horizon
  - Distance vector uses a timer
    - If there is no news about a route within the time duration
    - Delete the route
  - Since Node B eliminates the route to X
    - Node A cannot decide it is due to split horizon or because B has not received any news about X recently
- Solution: Split Horizon and Poison Reverse

Split Horizon and Poison Reverse

- A variation of split horizons
- Information received is used to update routing table and then passed out to ***all*** interface
- However, a table entry is set to a metric of ***infinity*** *as it's come through and goes out interface are the same*
- For example
  - Router B has received information about Net1 and Net2 through its left interface
  - Thus, it sends information out about Net1 and Net2 with a metric of 16 to its left interface (assume 16 is infinity)

Split Horizon and Poison Reverse



Three-Node Instability

- Split Horizon and Poison Reverse cannot solve three-node instability
- 1. A detects X is not reachable
  - Sends a packet to B and C
- 2. B updates its table but the packet to C is lost
- 3. After a while, C sends to B its routing table
  - B is fooled and updates its routing table
- 5. B sends its routing table to A
  - A is fooled and updates its routing table
- 6. A then sends its routing table to B and C
- 7. The loop continues until the cost reach infinity

Three-Node Instability

Before failure

After A sends the route to B and C, but the packet to C is lost

After C sends the route to B

After B sends the route to A