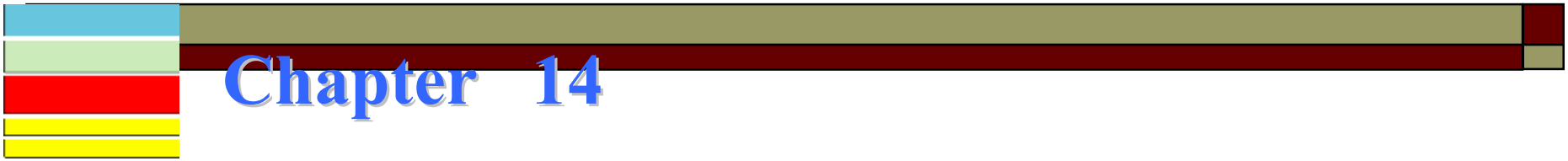


Chapter 14

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

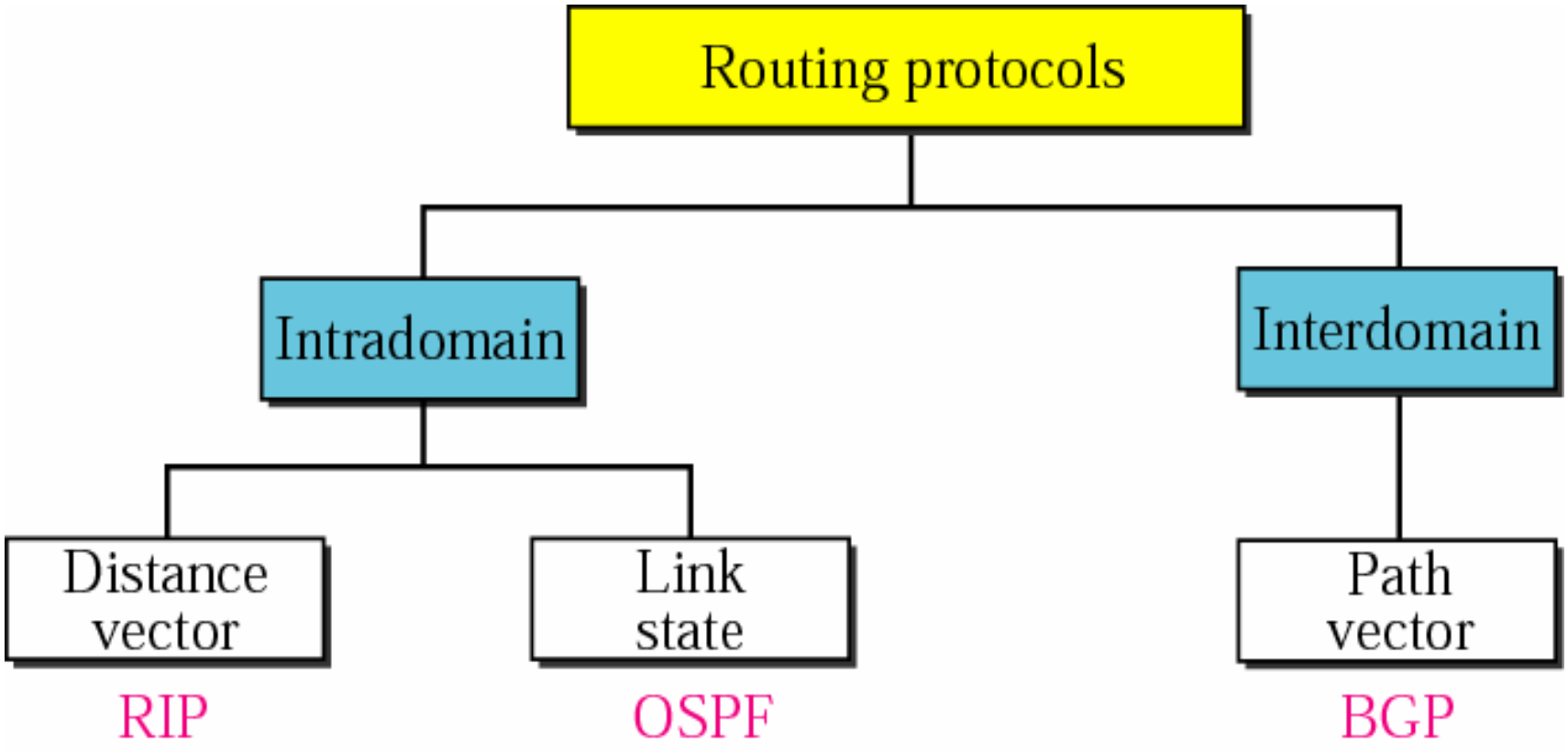
- An internet can be so large
  - One routing protocol cannot handle the task of updating routing table of all routers

- 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)

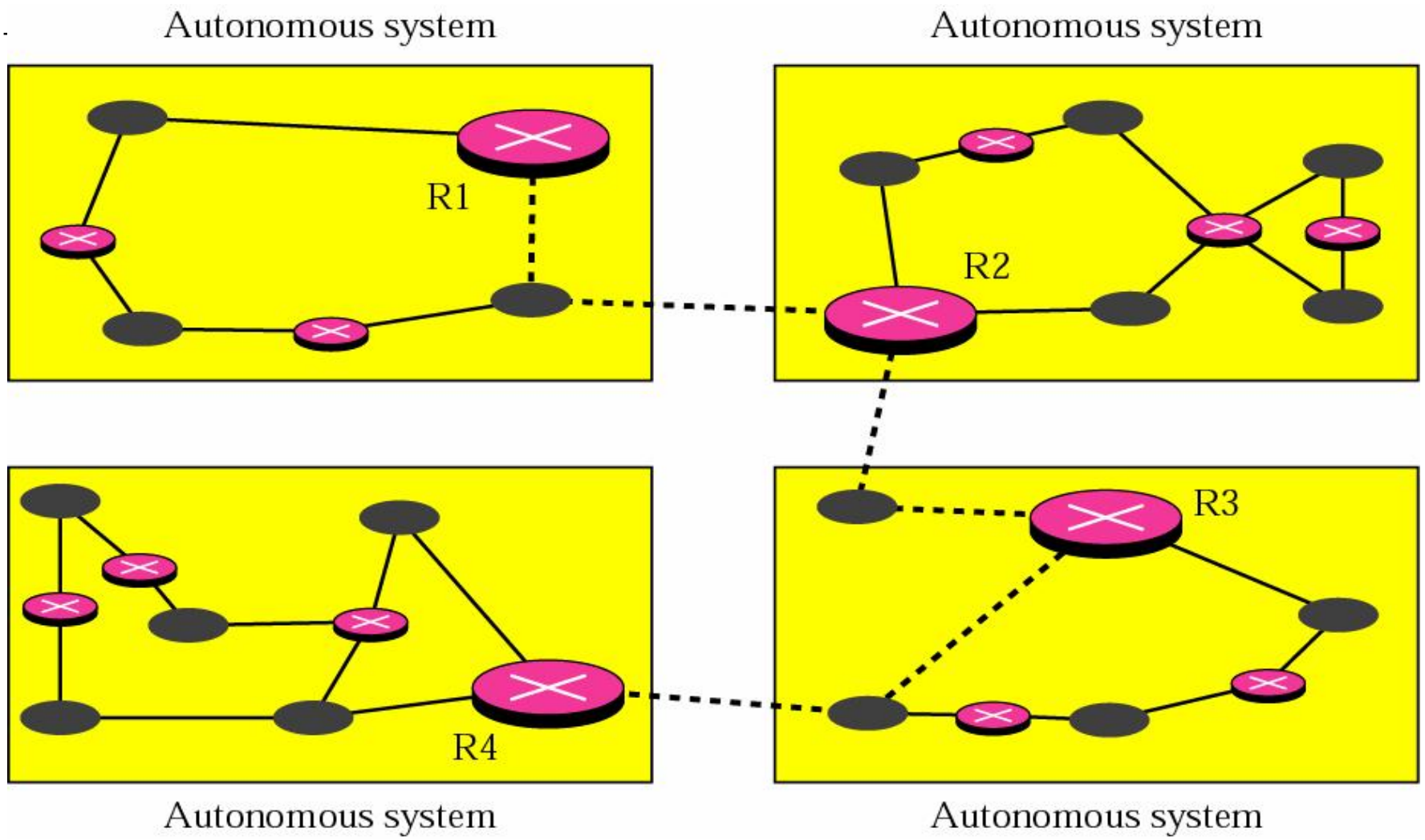
- Link-state routing algorithm
  - *Centralized* version of the shortest path computation
    - Every router has the *whole "picture"* of the internet
  - OSPF (Open Shortest Path First)

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:

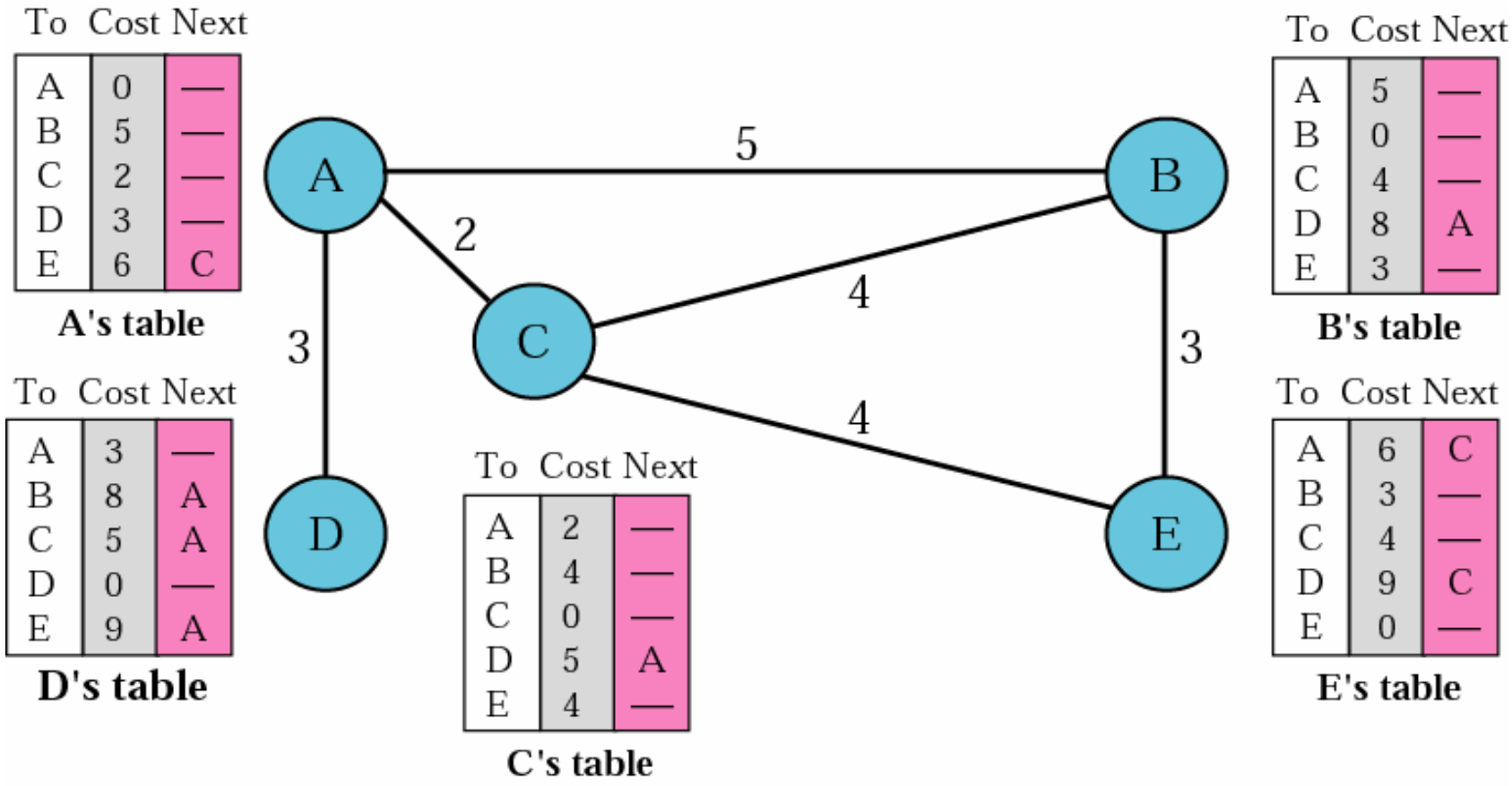
Initialization
Sharing
Updating
When to Share
Two-Node Loop Instability
Three-Node Instability

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

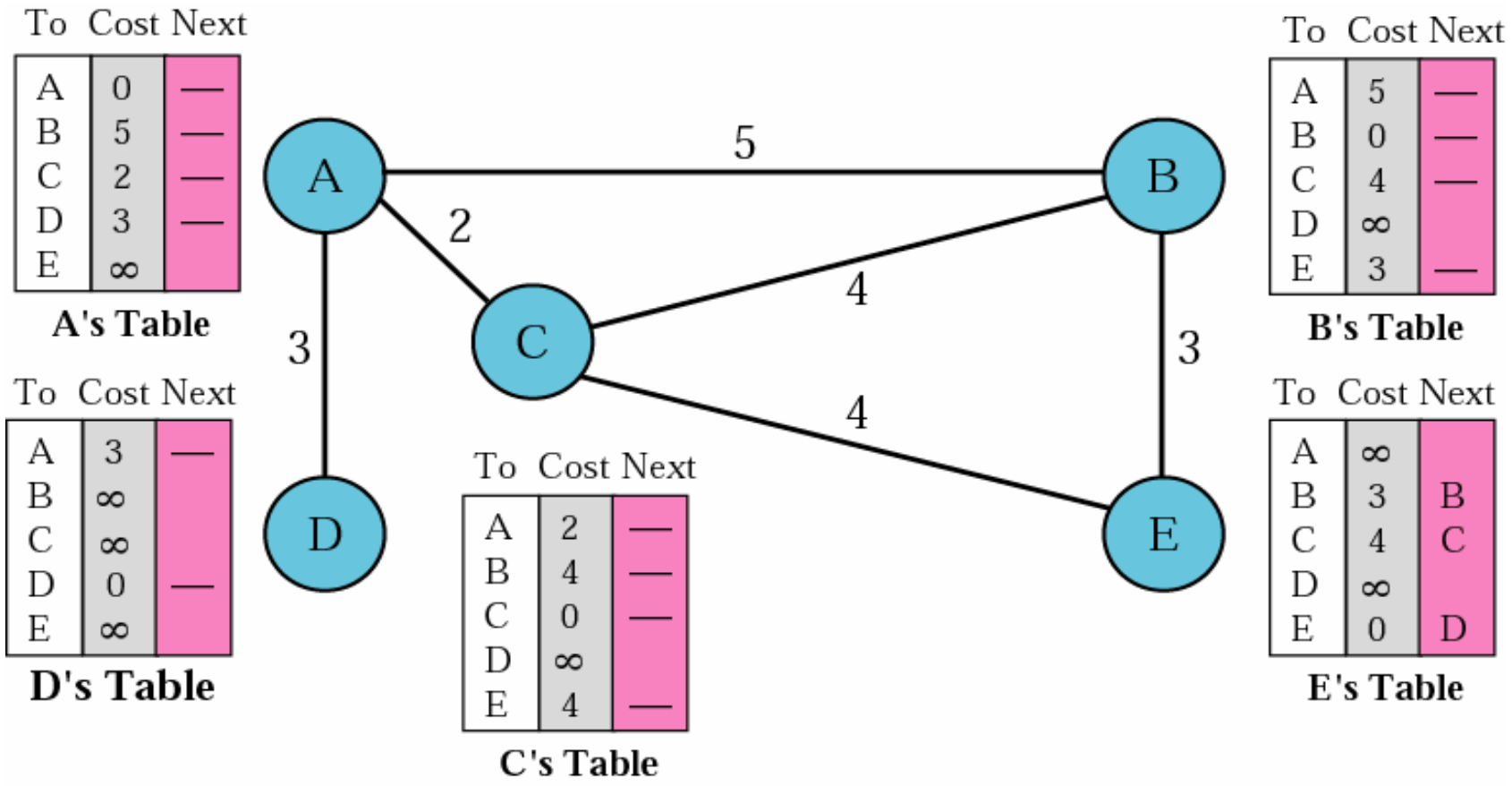
Distance Vector Routing Tables



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



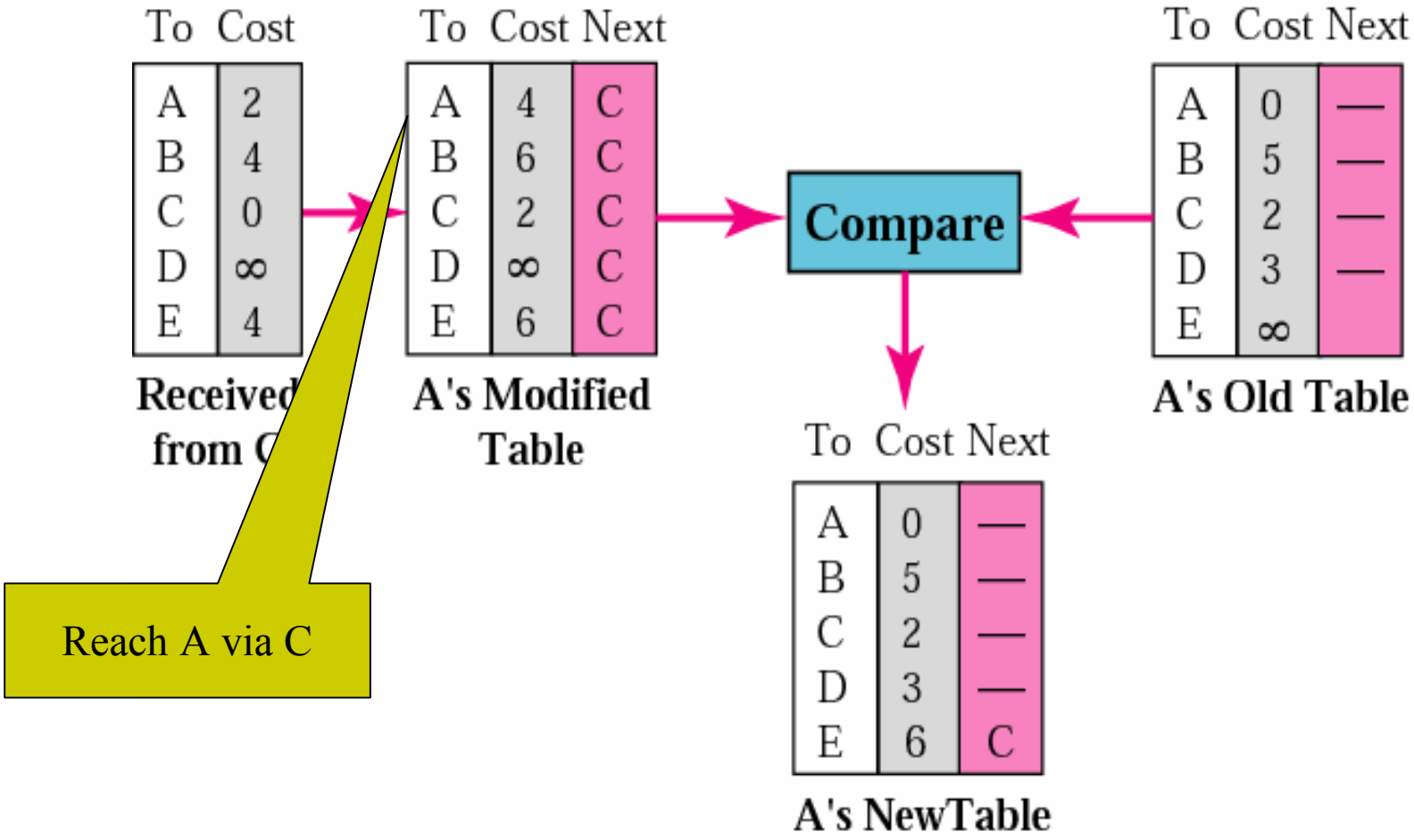
Sharing

- 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



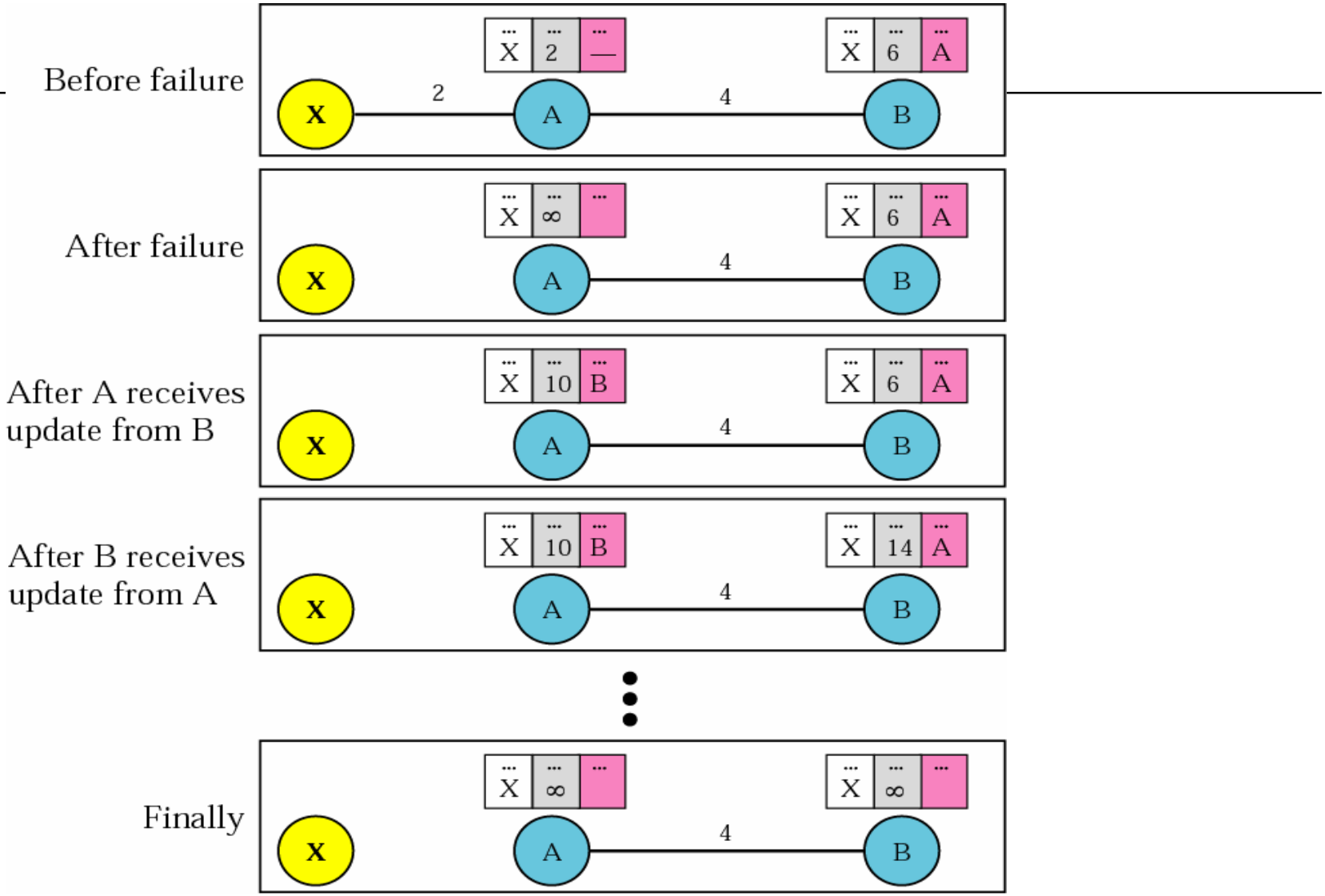
When to Share

- 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



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

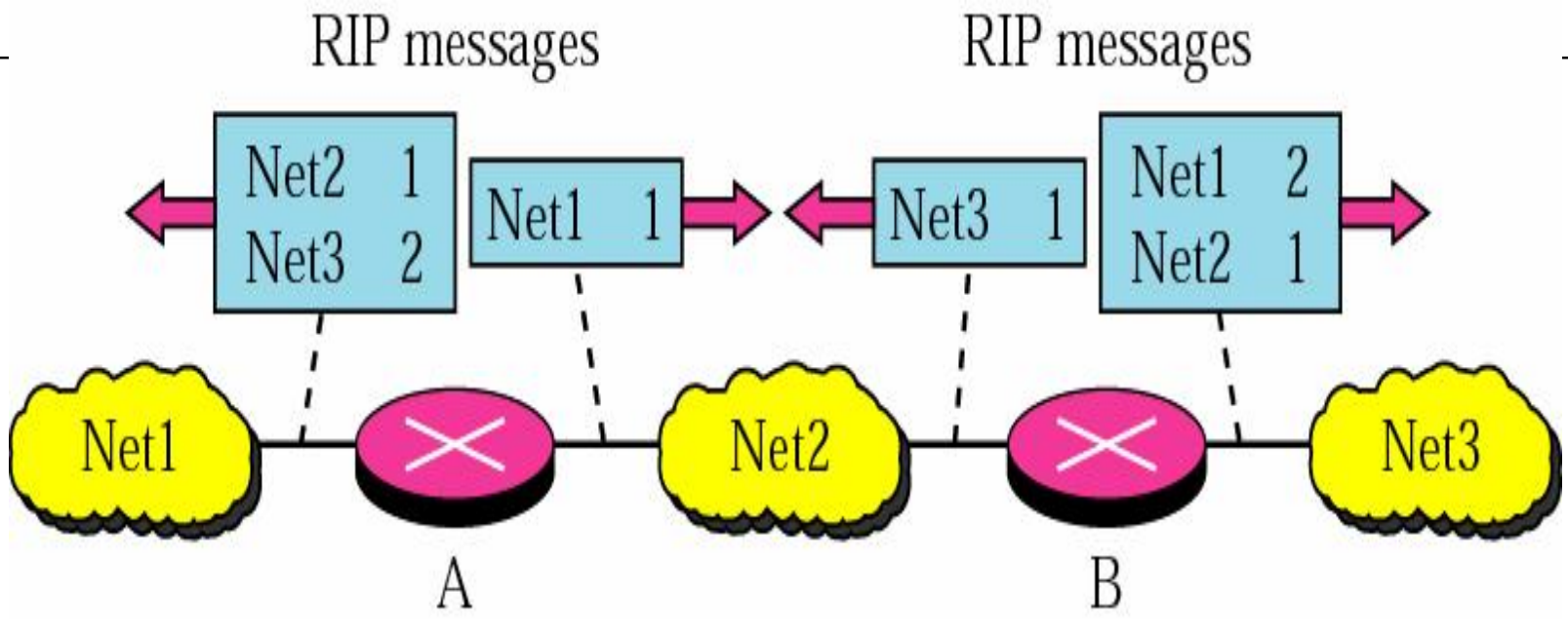
- Redefine infinity to a smaller number
  - Shorten the time of instability
- Most implementation define the distance between each node to be 1
  - Define 16 as infinity
- As a result
  - The distance vector scheme cannot be used in large system
  - The size of network, in each direction, can not exceed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

Figure 13.14

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

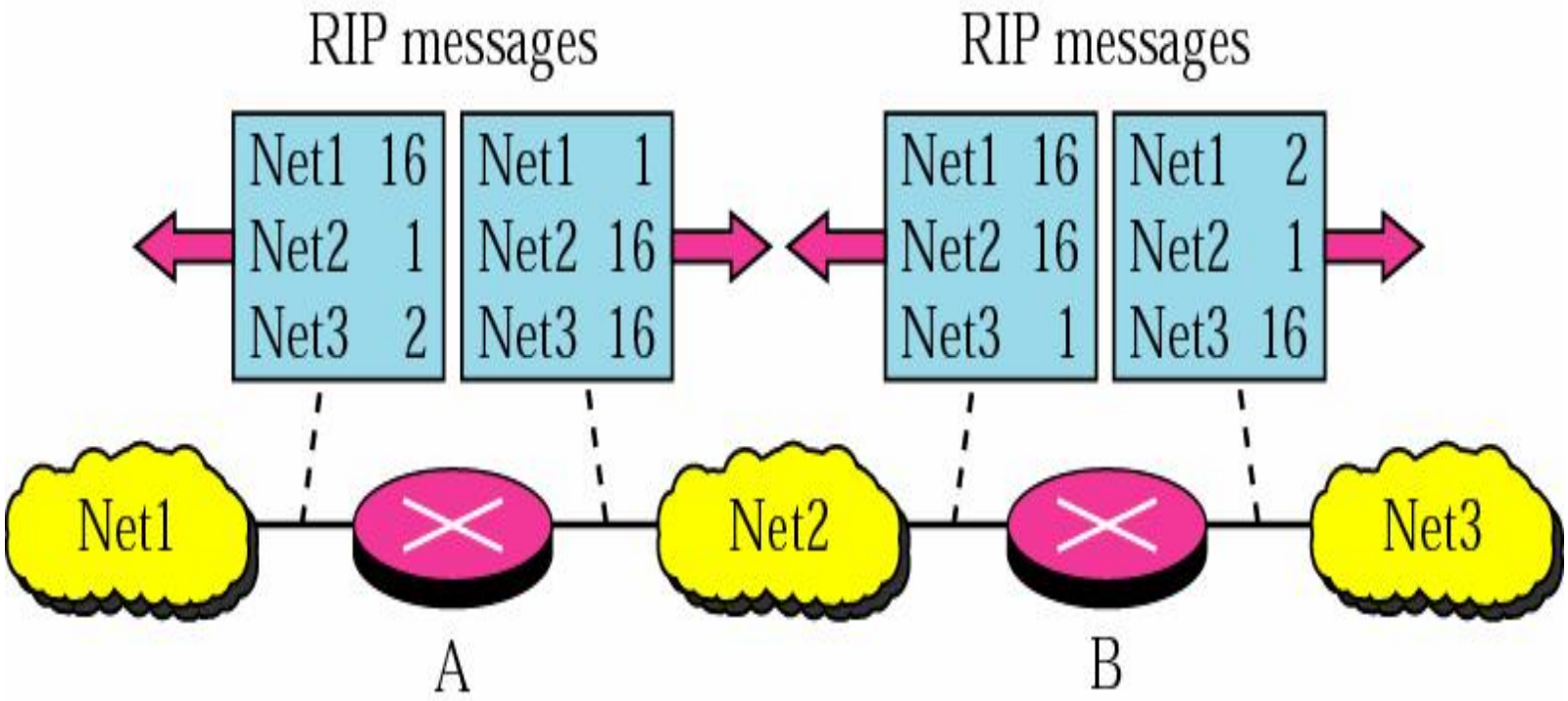
- Drawback of split horizon
  - Distance vector uses a timer
    - If there is no news about a route within the time duration
    - Delete the route
  - Since Node B eliminates the route to X
    - Node A cannot decide it is due to split horizon or because B has not received any news about X recently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

Figure 12.15

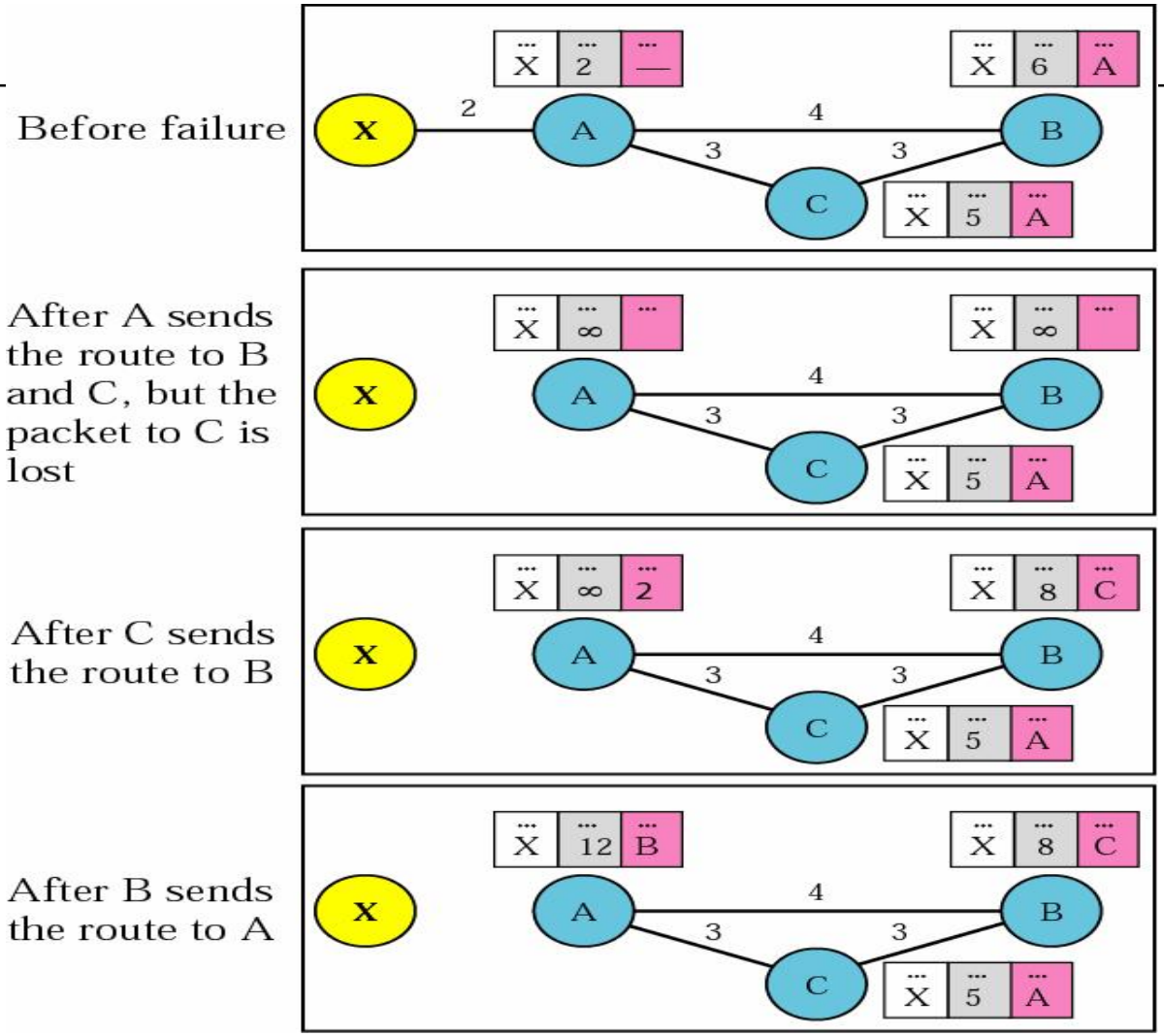
Split Horizon and Poison Reverse



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



14.3 RIP

The Routing Information Protocol (RIP) is an intradomain routing protocol used inside an autonomous system. It is a very simple protocol based on distance vector routing.

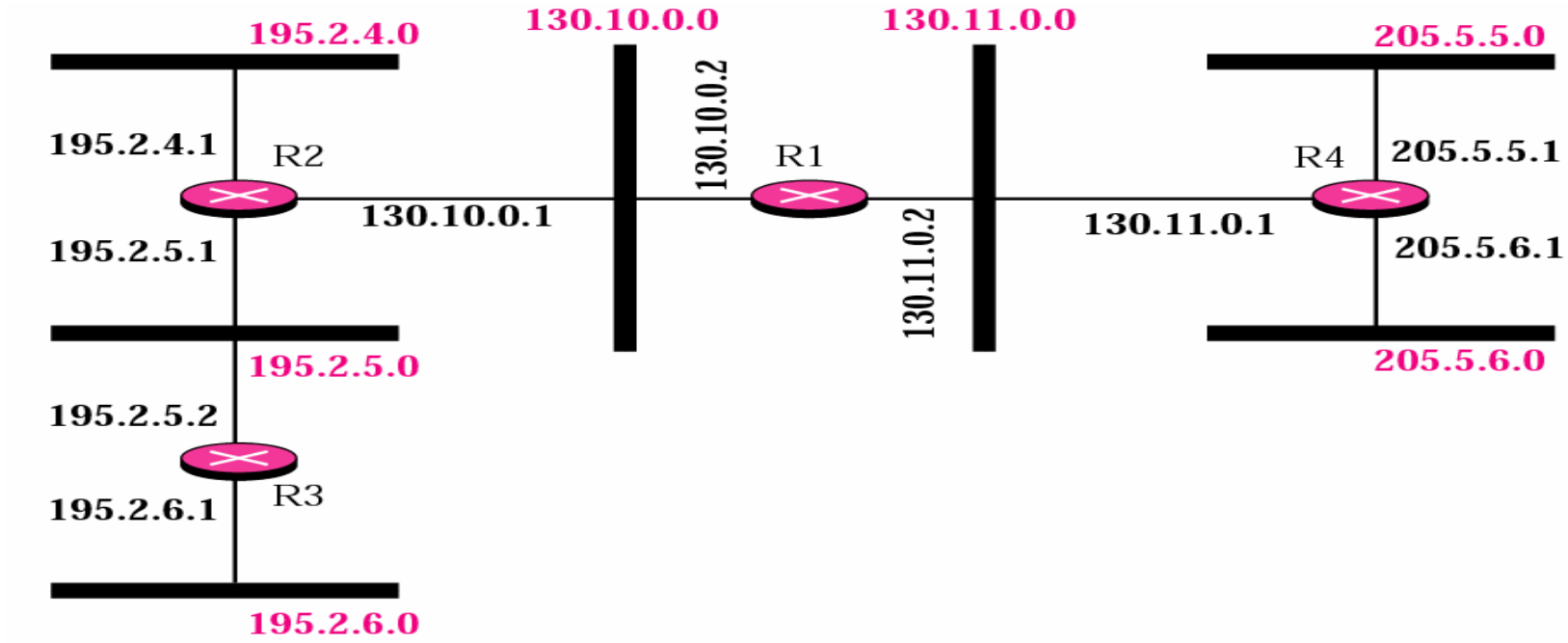
The topics discussed in this section include:

RIP Message Format
Requests and Responses
Timers in RIP
RIP Version 2
Encapsulation

RIP

- RIP: Routing Information Protocol
  - Based on *distance vector routing*
- Design considerations
  - In a AS, RIP deals with routers and networks (links)
  - The destination in a routing table is a network
    - The first column defines a ***network address***
  - The metric used in RIP is ***hop count***
  - Infinity is defined as 16
    - Any route in an AS cannot have more than 15 hops

Figure 14.8: Example of a Domain Using RIP



R1 Table

Dest.	Hop	Next
130.10.0.0	1	————
130.11.0.0	1	————
195.2.4.0	2	130.10.0.1
195.2.5.0	2	130.10.0.1
195.2.6.0	3	130.10.0.1
205.5.5.0	2	130.11.0.1
205.5.6.0	2	130.11.0.1

R2 Table

Dest.	Hop	Next
130.10.0.0	1	————
130.11.0.0	2	130.10.0.2
195.2.4.0	1	————
195.2.5.0	1	————
195.2.6.0	2	195.2.5.2
205.5.5.0	3	130.10.0.2
205.5.6.0	3	130.10.0.2

R3 Table

Dest.	Hop	Next
130.10.0.0	2	195.2.5.1
130.11.0.0	3	195.2.5.1
195.2.4.0	2	195.2.5.1
195.2.5.0	1	————
195.2.6.0	1	————
205.5.5.0	4	195.2.5.1
205.5.6.0	4	195.2.5.1

R4 Table

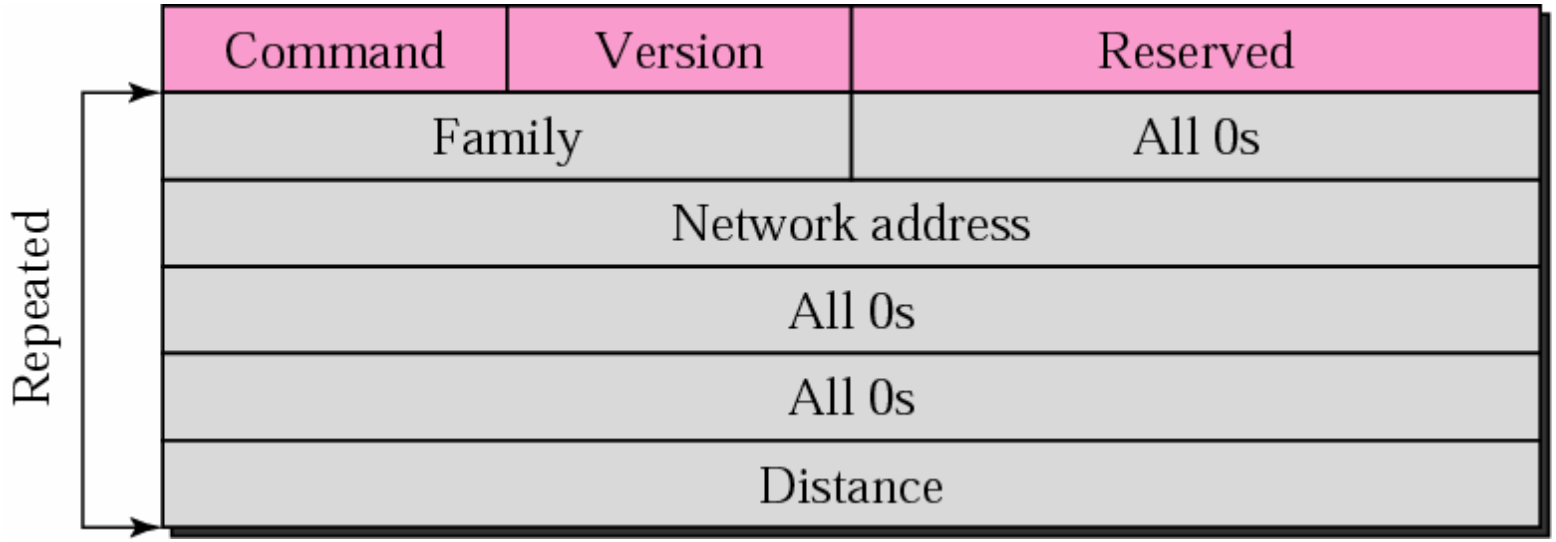
Dest.	Hop	Next
130.10.0.0	2	130.11.0.2
130.11.0.0	1	————
195.2.4.0	3	130.11.0.2
195.2.5.0	3	130.11.0.2
195.2.6.0	4	130.11.0.2
205.5.5.0	1	————
205.5.6.0	1	————

RIP Message Format

- Command: 8-bit
  - The type of message: request (1) or response (2)
- Version: 8-bit
  - Define the RIP version
- Family: 16-bit
  - Define the family of the protocol used
  - TCP/IP: value is 2
- Network Address: 14 bytes
  - Defines the address of the destination network
  - 14 bytes for this field to be applicable to any protocol
  - However, IP currently uses only 4 bytes, the rest are all 0s
- Distance: 32-bit
  - The hop count from the advertising router to the destination network

Figure 13-6

RIP Message Format



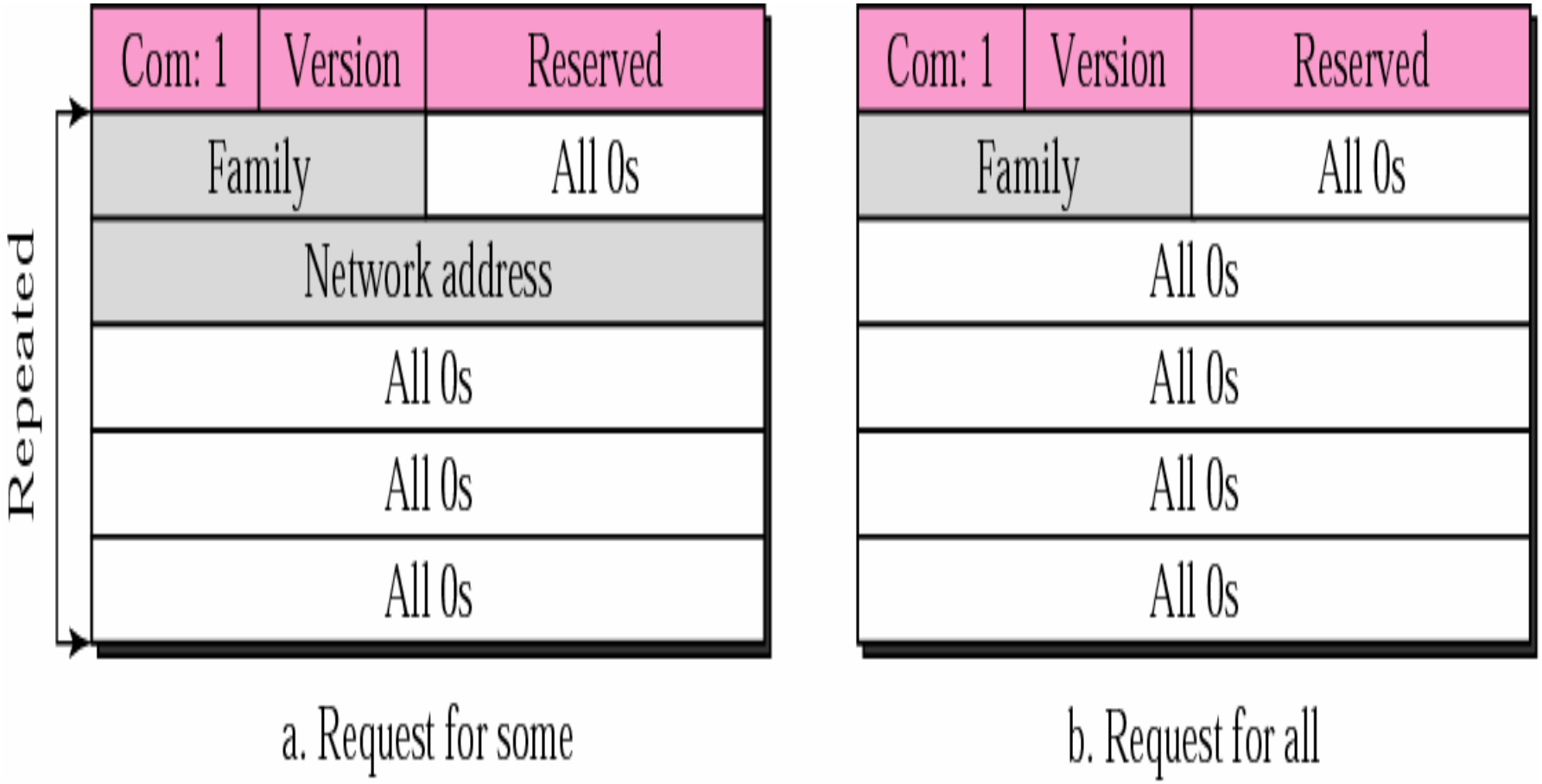
Requests and Response

- RIP uses two type of messages
  - *Request* and *response*

- Request
  - Sent by a router that *has just come up* or *has some time-out entries*
  - Can ask *specific entries* or *all entries*

Figure 13-7

Request Messages



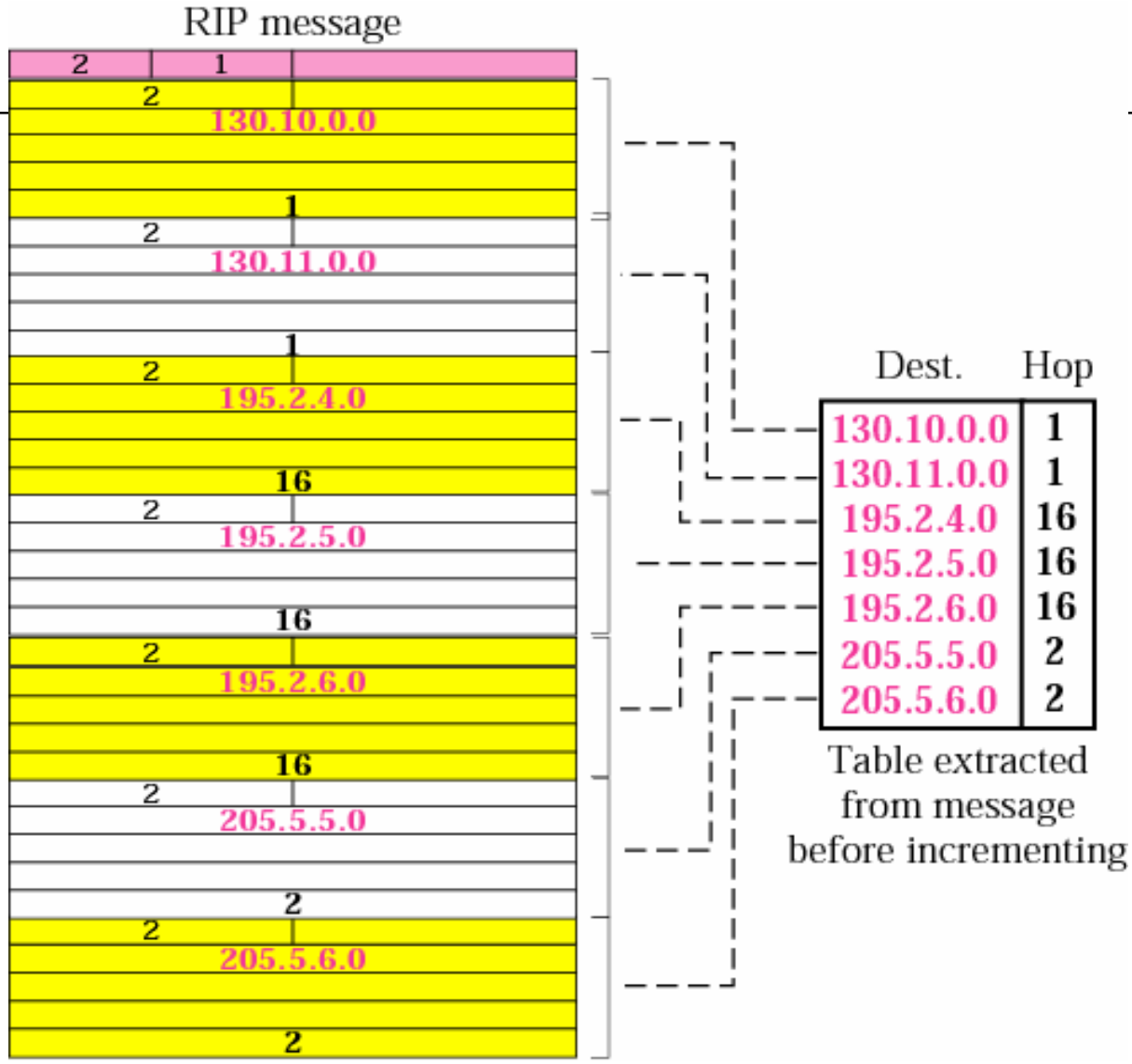
Requests and Response (Cont.)

- Response: solicited or unsolicited
  - A solicited response: sent only in answer to a request
    - Contain information about the destination specified in the corresponding request
  - An unsolicited response: sent *periodically*
    - Every 30s
    - Contains information about the entire routing table
    - Also called *update packet*

Example 1

- Following Figure shows the update message sent from router R1 to router R2 in Figure 14.8.
  - The message is sent out of interface 130.10.0.2
- The message is prepared with the combination of split horizon and poison reverse strategy in mind.
  - Router R1 has obtained information about networks 195.2.4.0, 195.2.5.0, and 195.2.6.0 from router R2.
  - When R1 sends an update message to R2,
    - Replace the actual value of the hop counts for these three networks with 16 (infinity) to prevent any confusion for R2.
- The figure also shows the table extracted from the message.
  - Router R2 uses the source address of the IP datagram carrying the RIP message from R1 (130.10.02) as the next hop address.

Figure 14.11 *Solution to Example 1*

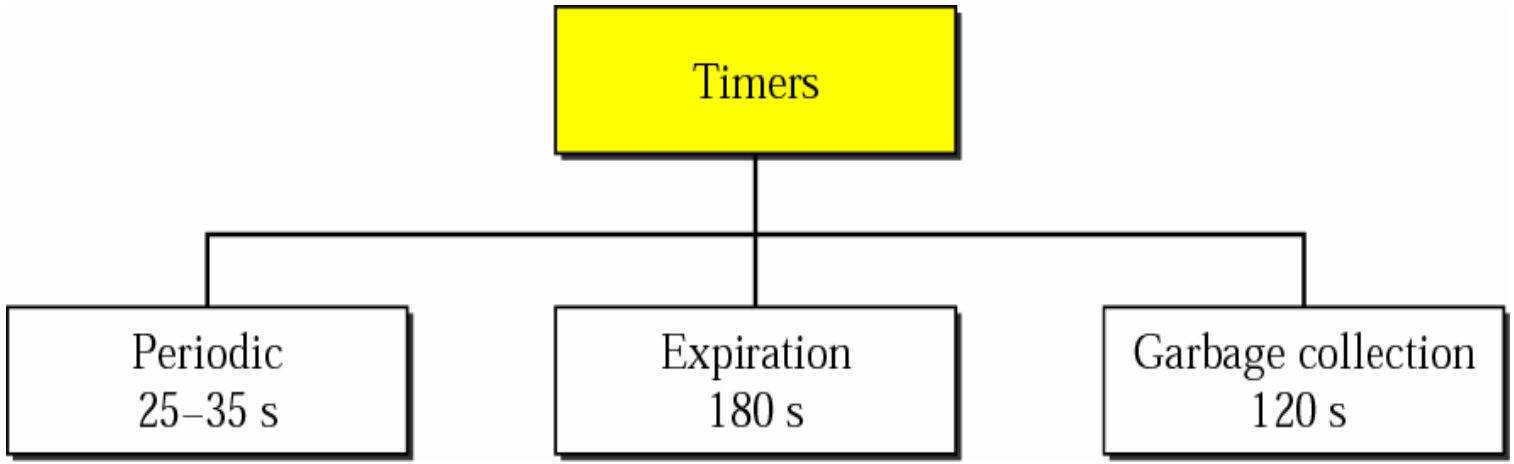


Timers in RIP

- RIP uses three timers
  - Periodic timer
  - Expiration timer
  - Garbage collection timer

Figure 12-19

RIP Timers



Periodic Timer

- Periodic timer
  - Control the advertising of regular update message
  - Although protocol specifies 30 s, the working model uses a random number between 25 and 35 s
    - Prevent routers update simultaneously

Expiration Timer

- Govern the validity of a route
- Set to 180 s for a route when a router receives update information for a route
  - If a new update for the route is received, the timer is reset
  - In normal operation, this occurs every 30 s
- If timer goes off, the route is considered expired
  - The hop count of the route is set to *16,* which means *destination is unreachable*

Garbage Collection Timer

- When a route becomes invalid, the router does not immediately purge that route from its table
- It continues advertise the route with a metric value of 16
- A garbage collection timer is set to 120 s for that route
- When the count reaches zero, the route is purged from the table
- *Allow neighbors to become aware of the invalidity of a route prior to purging*

Example 2

- A routing table has 20 entries.
- It does not receive information about five routes for 200 seconds.
- How many timers are running at this time?

Solution

- The timers are listed below:
  - Periodic timer: 1
  - Expiration timer: 20 - 5 = 15
  - Garbage collection timer: 5

RIP Version 2

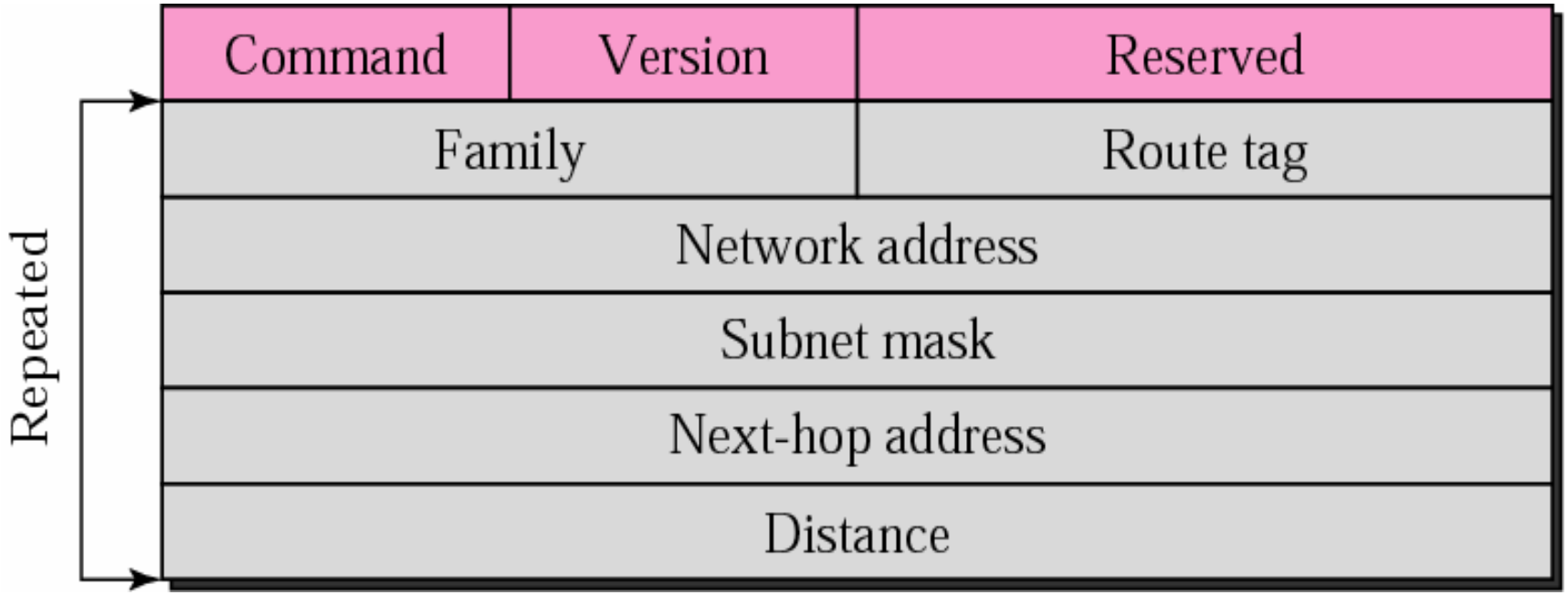
- Does not augment the length of the message of each entry
- Only replace those fields in version 1 that were filled with *0s* with some new fields

RIP Version 2

- New fields
  - **Routing Tag**: carries information such as the autonomous system number
    - Enable RIP to receive information from an exterior routing table
  - **Subnet mask**: carries the subnet mask (or prefix)
    - RIP2 support classless addressing and CIDR
  - **Next-hop address**: show the address of the next hop

Figure 13.16

RIP-v2 Format



Classless Addressing

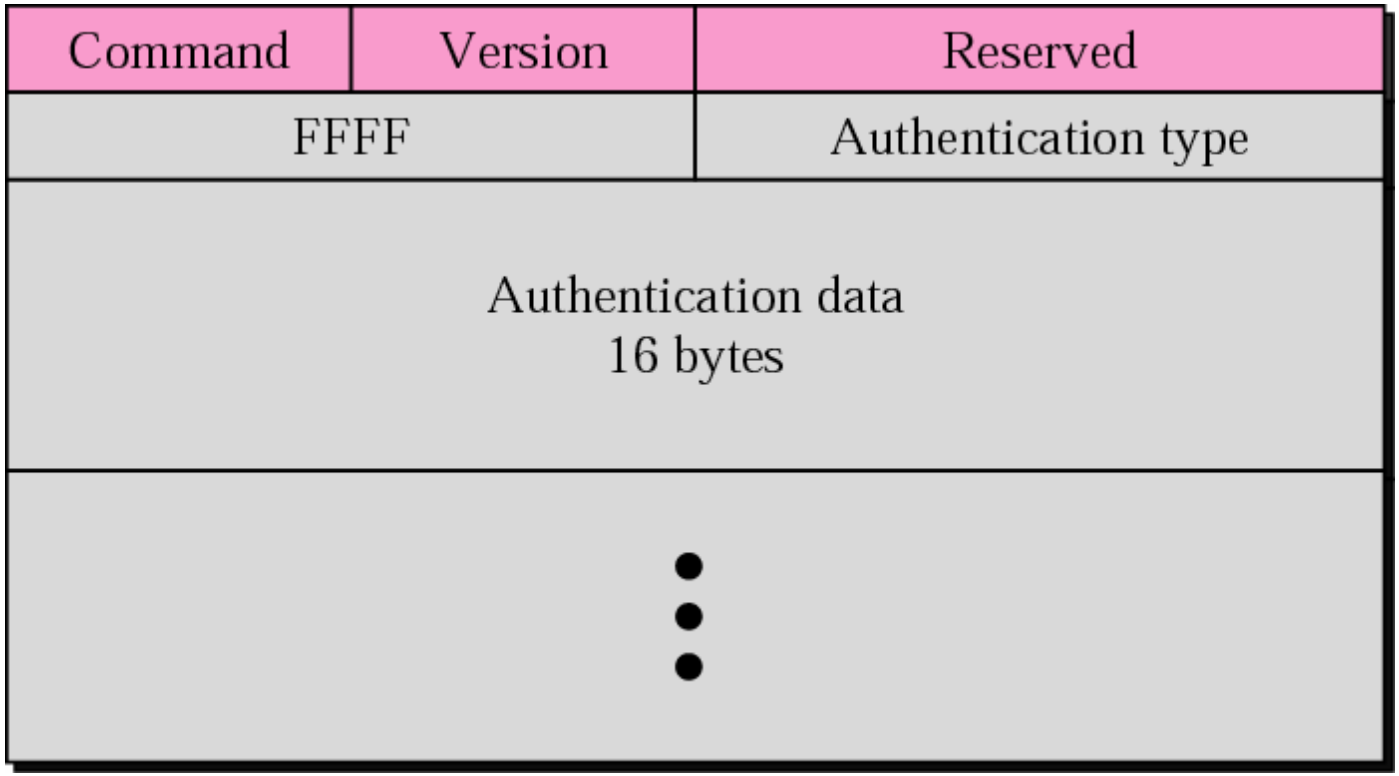
- The most important difference between the two versions
  - ***classful v.s. classless addressing***
- RIPv1 uses classful addressing
  - The only entry in the message format is the ***network address*** (with a default mask)
- RIPv2 support classless addressing
  - Adds one filed for the ***subnet mask***

Authentication

- Protect the message against unauthorized advertisement
- The first entry of the message is set aside for authentication information
  - Family field = $FFFF_{16}$
    - Not used for routing information but for authentication
  - Authentication type
    - Define the method used for authentication
  - Authentication data
    - Contain the actual authentication data

Figure 12-17

Authentication



Multicasting

- Version 1 of RIP uses broadcasting to send RIP message to every neighbor
  - All the routers and the hosts receive the packets
- RIP version 2
  - Uses the multicast address 224.0.0.9 to multicast RIP message only to RIP routers in the network

Encapsulation

- RIP message are encapsulated in UDP user datagram
- The well-known port assigned to RIP in UDP is port 520

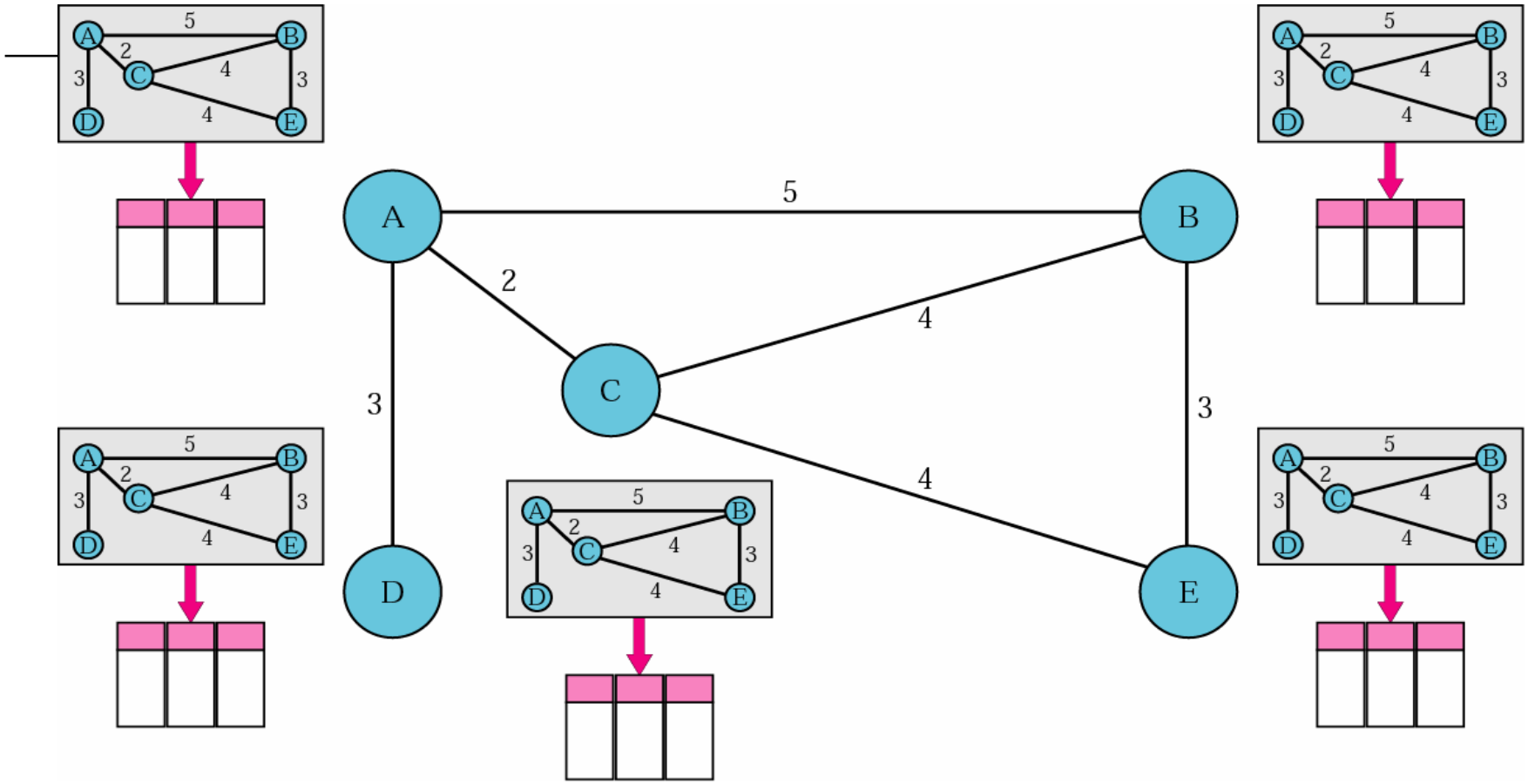
14.4 LINK STATE ROUTING

In link state routing, if each node in the domain has the entire topology of the domain, the node can use Dijkstra's algorithm to build a routing table.

The topics discussed in this section include:

Building Routing Tables

Figure 14.15 *Concept of link state routing*



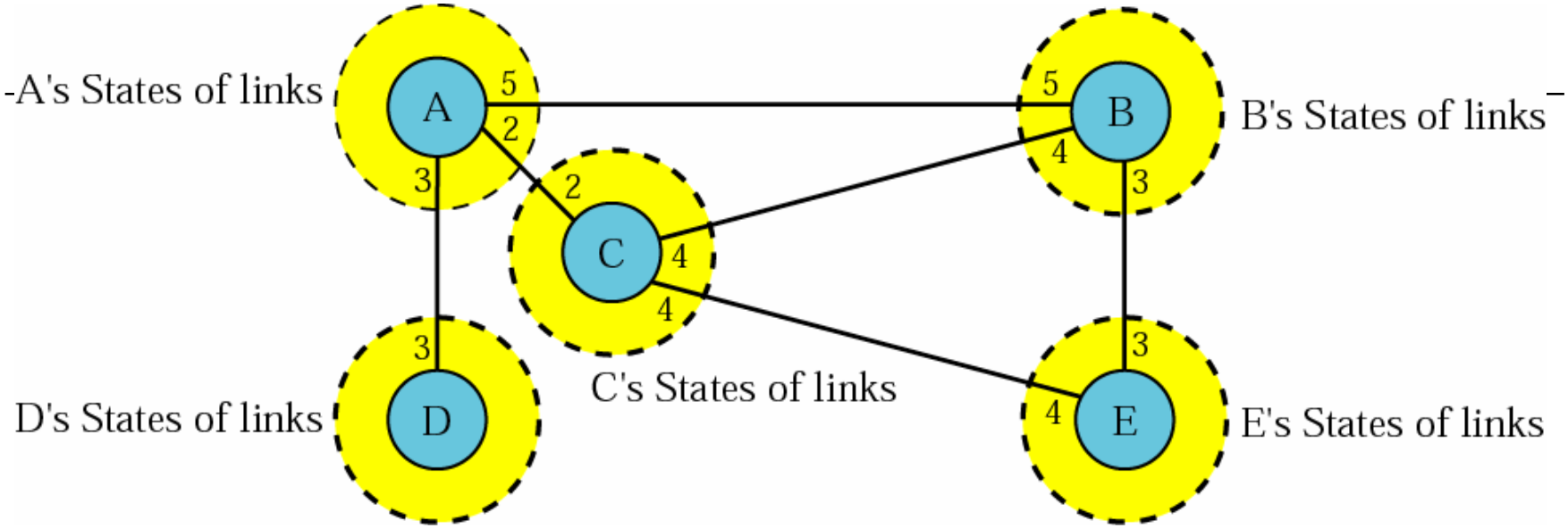
Link State Routing

- From Figure 14.15
  - Each node uses the same topology to create a routing table
  - But the routing table for each node is unique
    - Like a city map

Link State Routing

- Assumption of link state routing
  - Although the global topology knowledge is not clear and each node has partial knowledge
    - It knows the state (type, condition, cost) of its link
  - However, the while topology can be compiled from the partial knowledge of each node
    - See the Figure 14.16

Figure 14.16 *Link state knowledge*



- Each node has a partial knowledge of the network
- There is an overlap in the knowledge
- The overlap guarantees the creation of a common topology
  - A picture of the whole domain for each node

Building Routing Tables

- For sets of actions in link state routing
  - ***Creation*** of the states of the links by each node
    - Called the ***link state packet*** or ***LSP***
  - ***Dissemination*** of LSPs to every other router, called ***flooding***, in an efficient and reliable way
  - ***Formation*** of a shorten path tree for each node
  - ***Calculation*** of a routing table based on the shortest path tree

Creation of Link State Packet (LSP)

- Assume a LSP carries
  - ***The node identity***
  - ***The list of links***
    - Both are needed to make the topology
  - ***A sequence number***
    - Distinguishes new LSPs from old ones
  - ***Age***
    - Prevent old LSPs from remaining in the domain for a long time

Creation of Link State Packet (LSP)

- LSP are generated on two occasions
  - When there is a ***change*** in the topology of the domain
    - Quickly inform any node to update its topology
  - On a ***periodic*** basis
    - The period is much longer compared to the distance vector routing
    - 60 minutes or 2 hours

Flooding of LSPs

- Flooding: the LSP must be disseminated to all other nodes in the domain
  - Not only to its neighbors
- Rules
  - The creating node sends a copy of the LSP out of each interface
  - All receiving nodes compare the incoming one with the copy it may already have
    - If the newly LSP is older than the one it has by checking sequence number
      - Discard the LSP
    - Else
      - Discard the old LSP
      - Sends a copy of it out of each interface except the incoming one

Formation of Shortest Path Tree: Dijkstra Algorithm

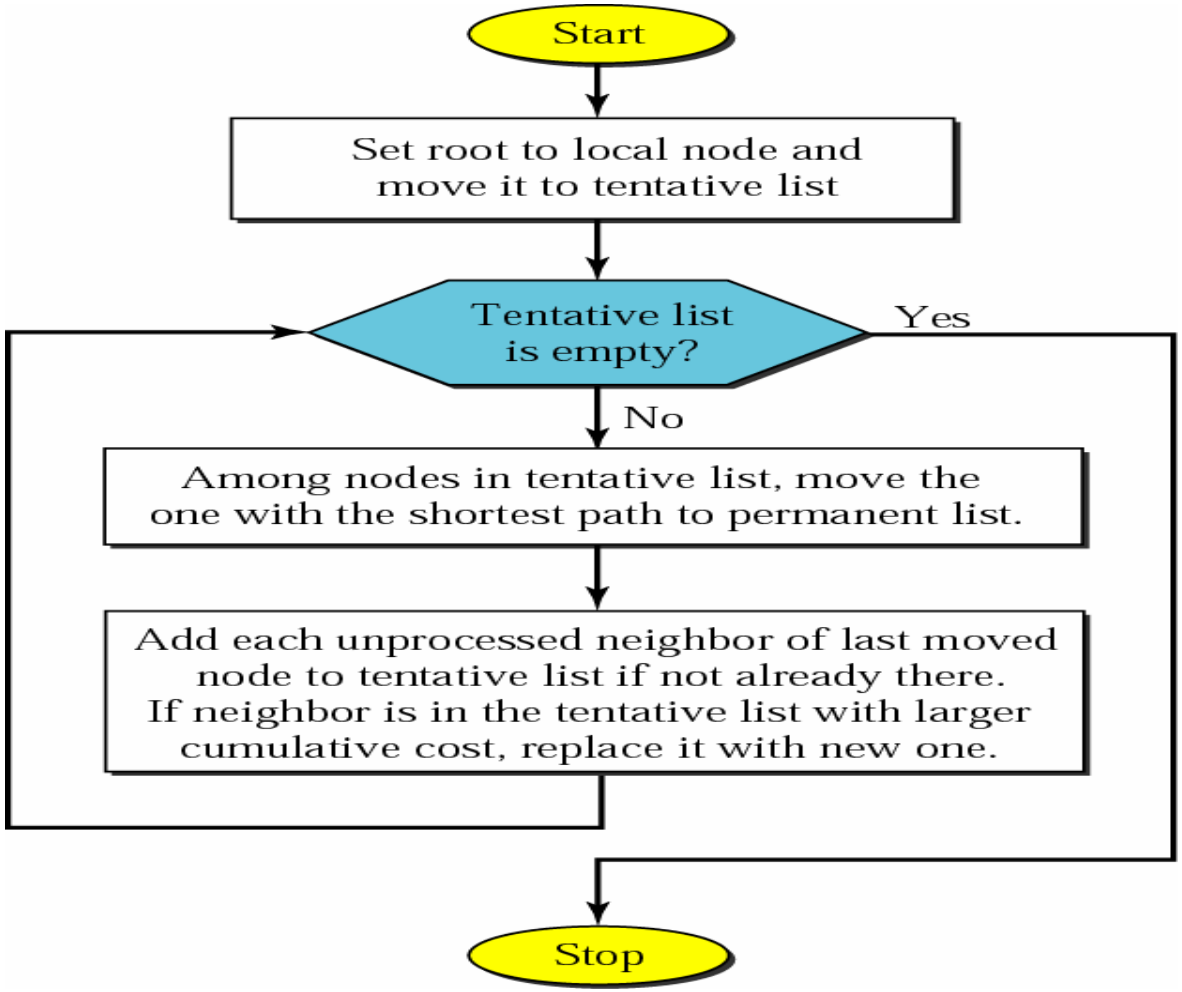
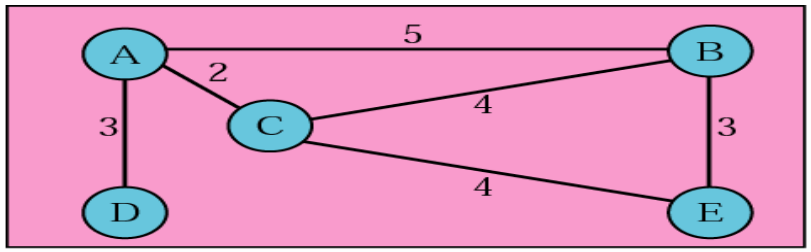
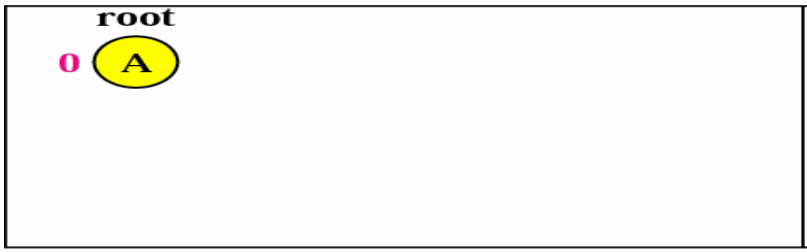


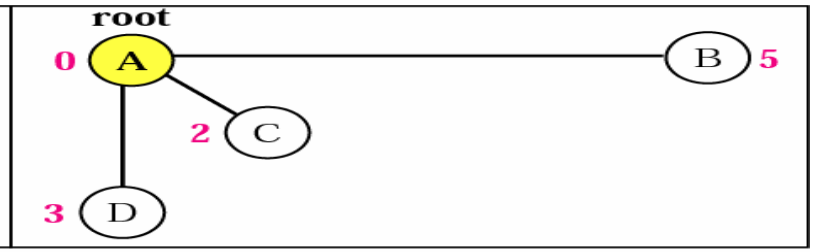
Figure 14.18 *Example of formation of shortest path tree*



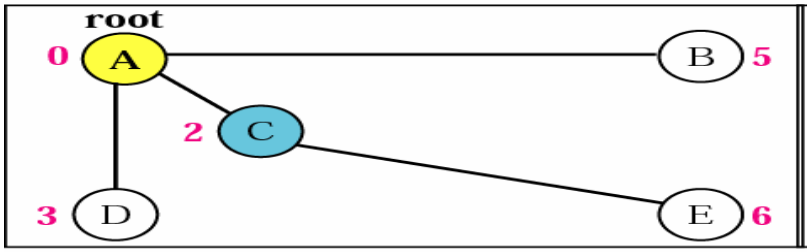
Topology



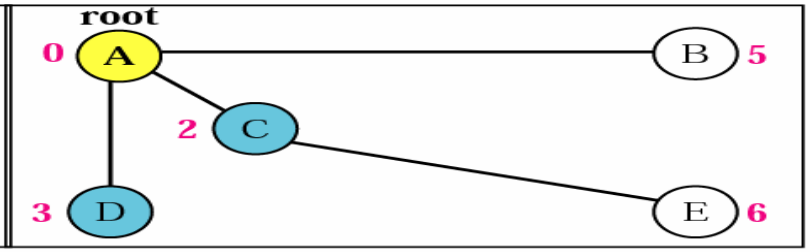
1. Set root to A and move A to tentative list



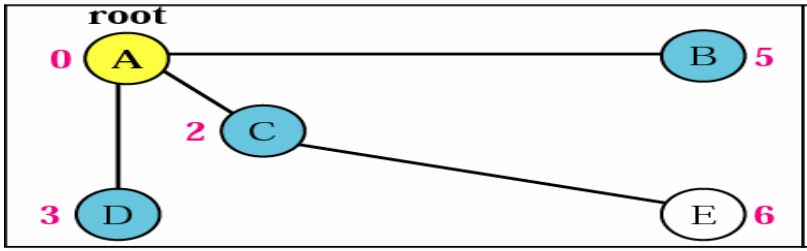
2. Move A to permanent list and add B, C, and D to tentative list



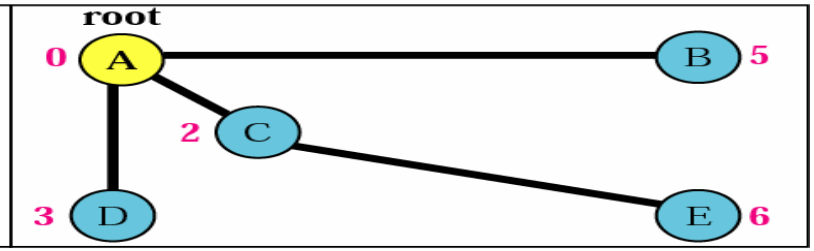
3. Move C to permanent and add E to tentative list



4. Move D to permanent list.



5. Move B to permanent list



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

Calculation of Routing Table from Shortest Path Tree

- Example:

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

14.5 OSPF

The Open Shortest Path First (OSPF) protocol is an intradomain routing protocol based on link state routing. Its domain is also an autonomous system.

The topics discussed in this section include:

Areas
Metric
Types of Links
Graphical Representation
OSPF Packets
Link State Update Packet
Other Packets
Encapsulation

Areas

- OSPF divides an autonomous system into *areas*
  - To handle routing efficiently and in a timely manner
- A collection of networks, hosts, and routers all contained within an autonomous system
- Thus, an autonomous system can be divided into many different areas
- All networks inside an area must be connected

Areas (Cont.)

- Routers inside an area *flood the area* with *routing information*

- At the border of an area, special routers called *area border routers*
  - Summarize the information about the area and sent it to other areas

Areas (Cont.)

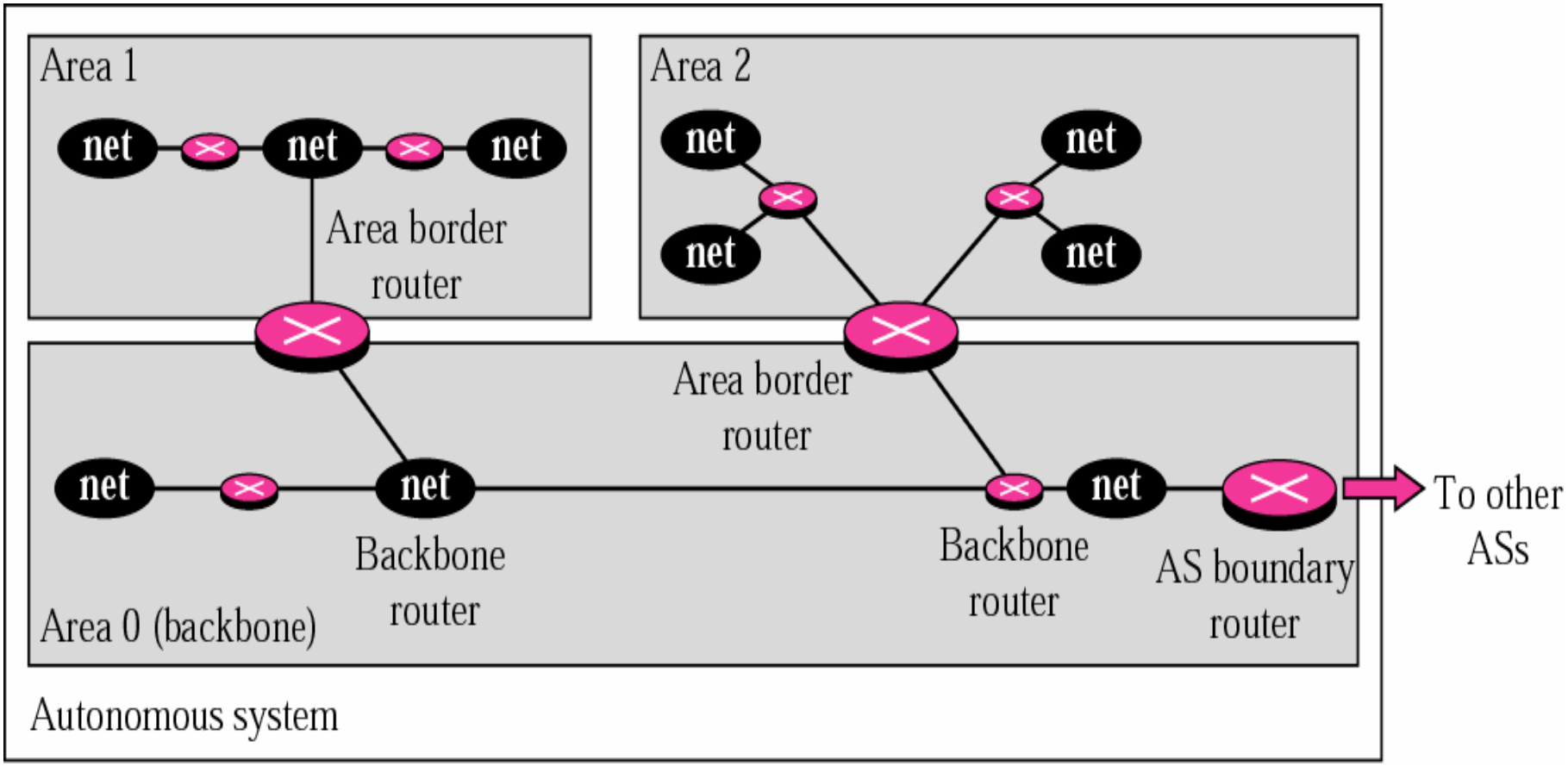
- Among the area inside an autonomous system is a *special area* called *backbone*
  - All of the areas inside an AS must be connected to the backbone
- The routers inside the backbone are called the *backbone routers*
  - A *backbone router* can also be an *area border router*

Areas (Cont.)

- If the connectivity between a backbone and an area is broken
  - A *virtual link* must be created by the administration
- Each area has an *area identification*
  - The area identification of the backbone is zero

Figure 12-18

Areas in an Autonomous System



Metrics

- OSPF allows the administrator to assign a cost, called the *metric*, to each route
- Metric can be based on a type of service
  - Minimum delay
  - Maximum throughput
- A router can have multiple routing tables
  - Each based on a different type of service

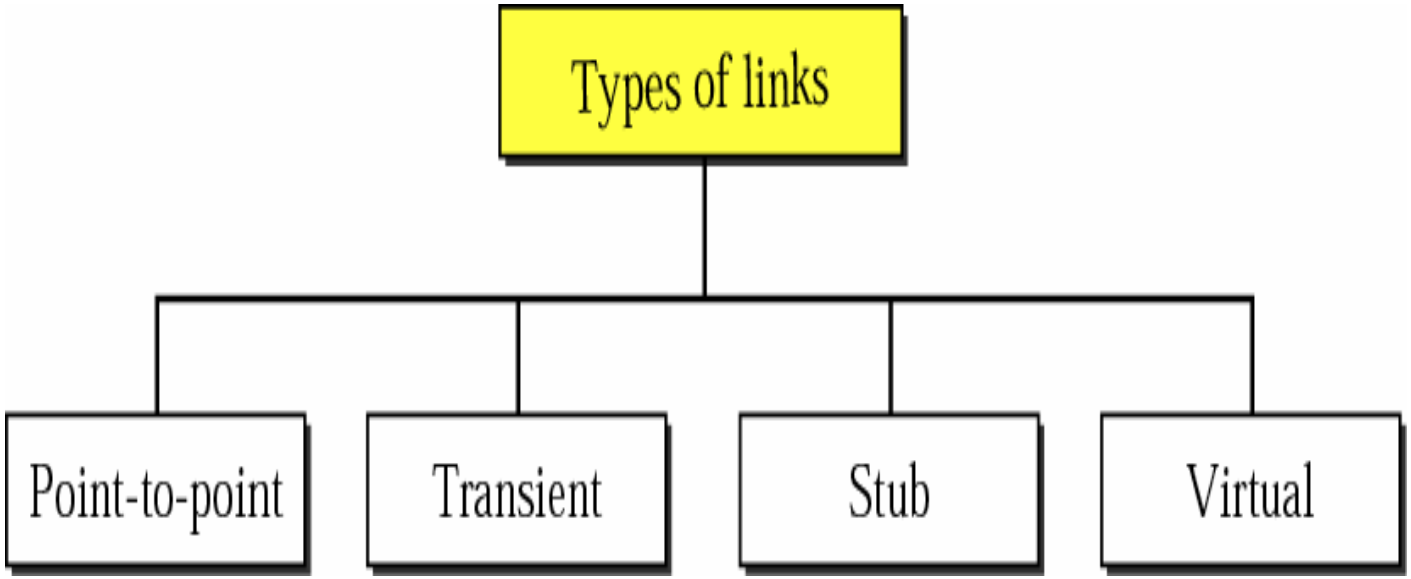
Types of Links

- In OSPF, a connection is called a *link*

- Four types of links
  - Point-to-point
  - Transient
  - Stub
  - Virtual

Figure 12-19

Types of Links

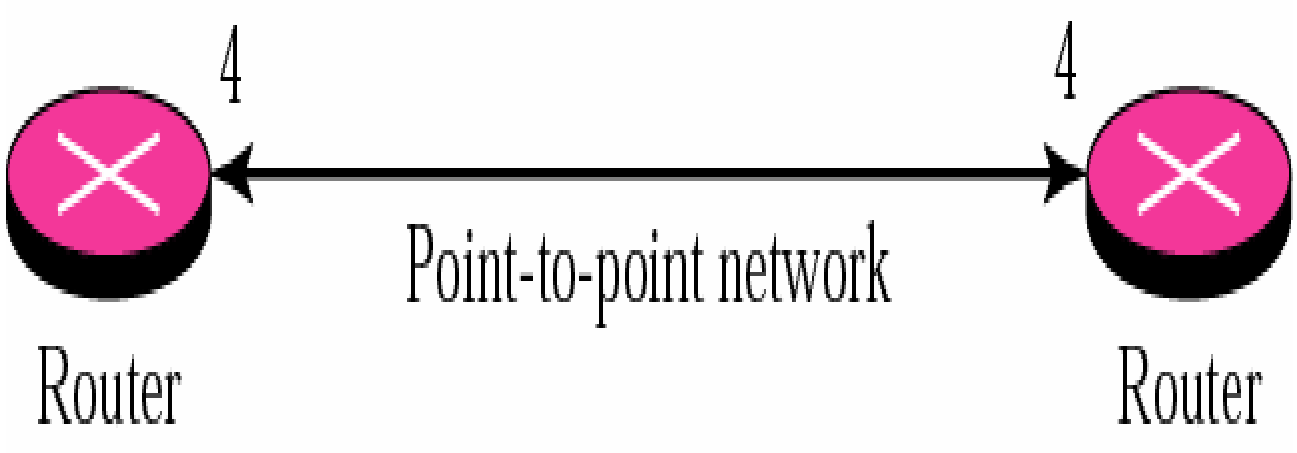


Point-to-Point Link

- Connect two routers without any other host or router in these two routers
- Example
  - Telephone line
  - T-line
- Graphically representation
  - The routers are represented by *nodes*
  - The link is represented by a *bidirectional edge*
- The *metric*
  - Usually the same at the two ends

Figure 13-20

Point-to-Point Link



Transient Link

- A network with several routers attached to it
  - Data can enter through any of the routers and leave through any router
- Example
  - All LANs and some WANs with two or more routers

Transient Link (Cont.)

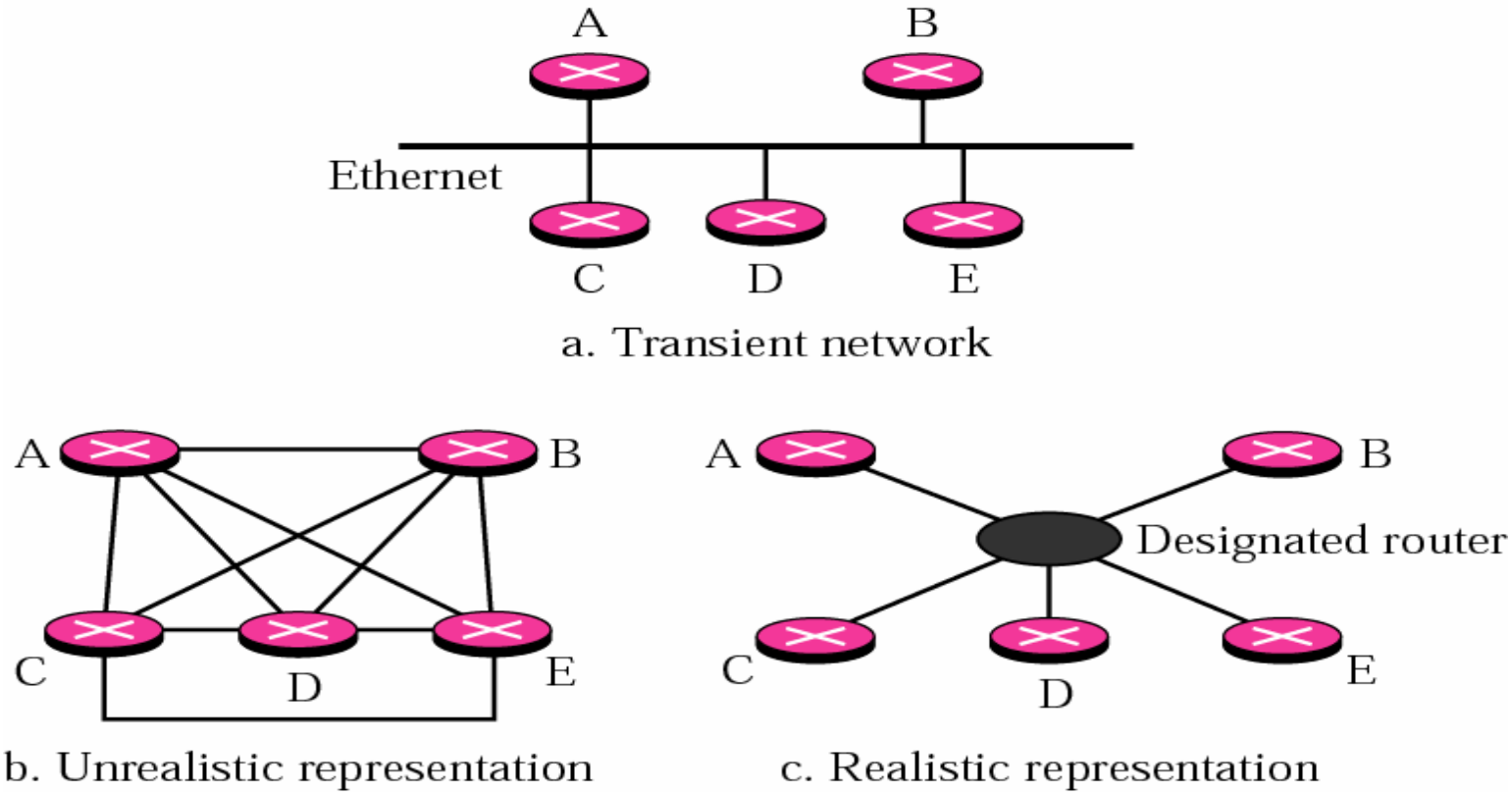
- Graphically representation
  - Figure b in the next slide. However, it is
    - **Not efficient**: each router need to advertise the neighborhood of four other routers
      - For a total of 20 advertisement
    - **Not realistic:** there is no single network (link) between each pair of routers
      - There should be only one network that serves as a crossroad between all five routers

Transient Link (Cont.)

- Reality: each router should be connected to every router *through one single network*
  - The network is represented by a node
  - However, network is not a machine
    - Cannot function as a router
- Solution: one of the routers acts as a single network
  - This router has a dual purpose: a *true router* and a *designated router*
    - The link is represented as a *bidirectional edge*
    - Figure c in the next slide

Figure 12.21

Transient Link

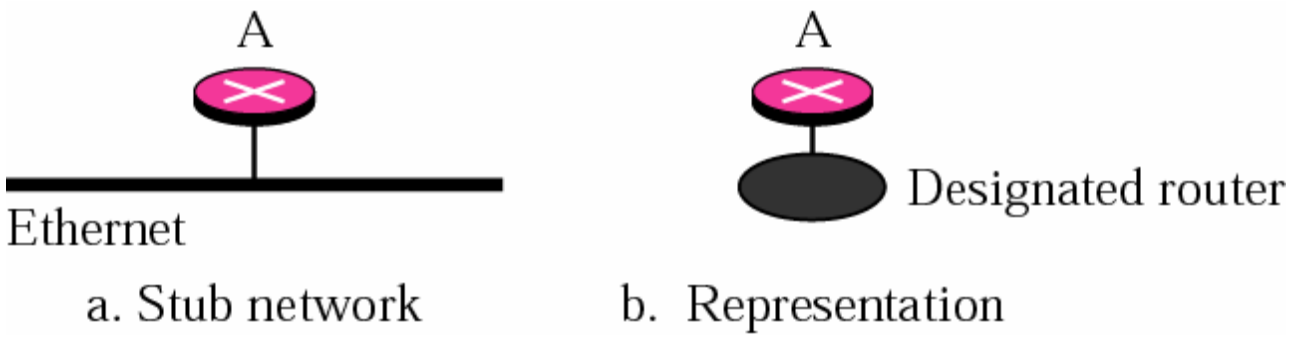


Stub Link

- A network that is connected to only one router
  - Data packet enter and leave through this only one router
- A special case of transient network
- Graphically representation
  - The router as a node
  - The designated router as the network
  - Note, the link is only one-directional
    - From the router to the network
    - Because the network is the end point in the graph representation
      - See the following third slides

Figure 12.22

Stub Link

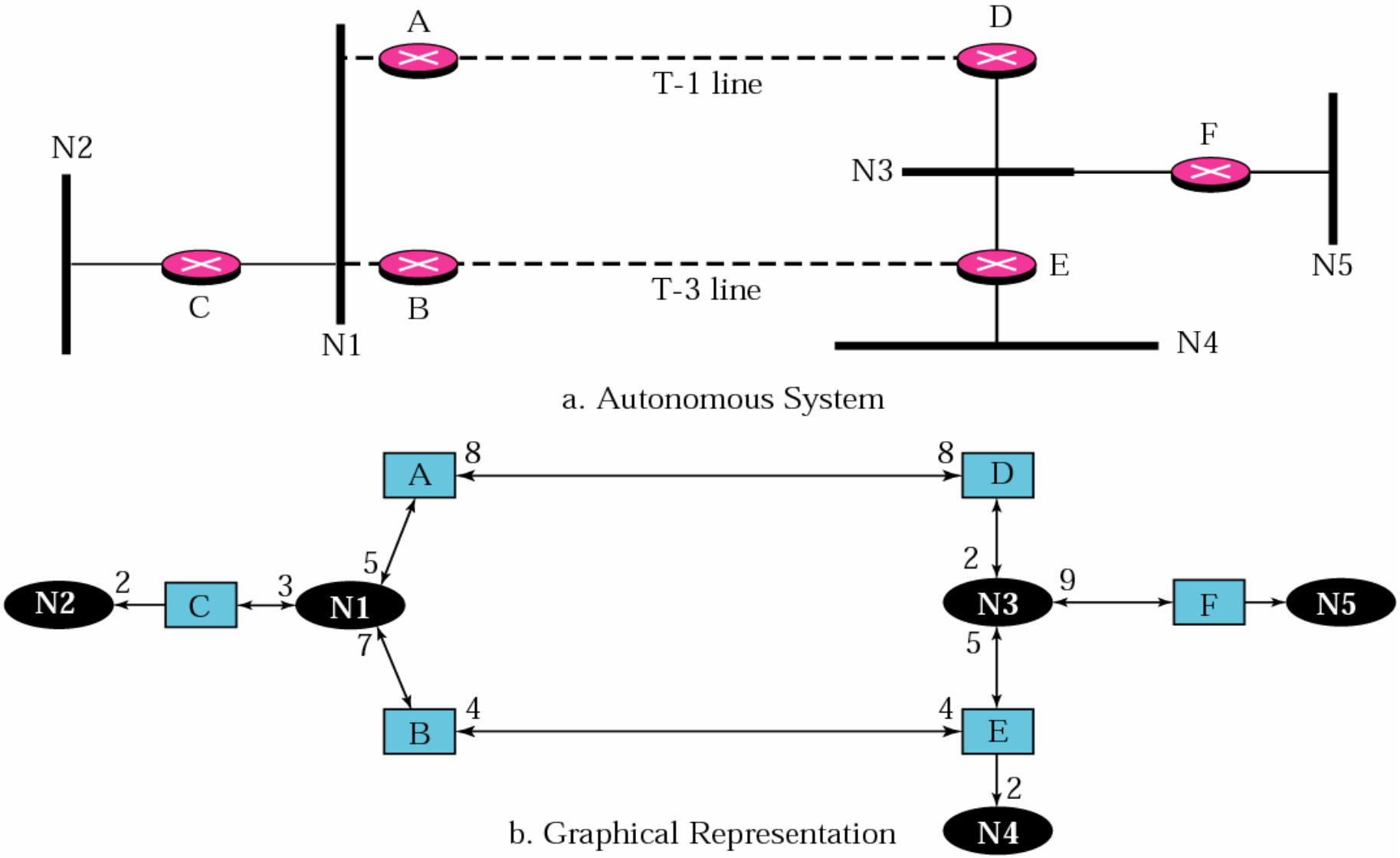


a. Stub network b. Representation

Virtual Link

- When the link between two routers is broken
  - The administrator may create a virtual path between them using a longer path and may go through several routers

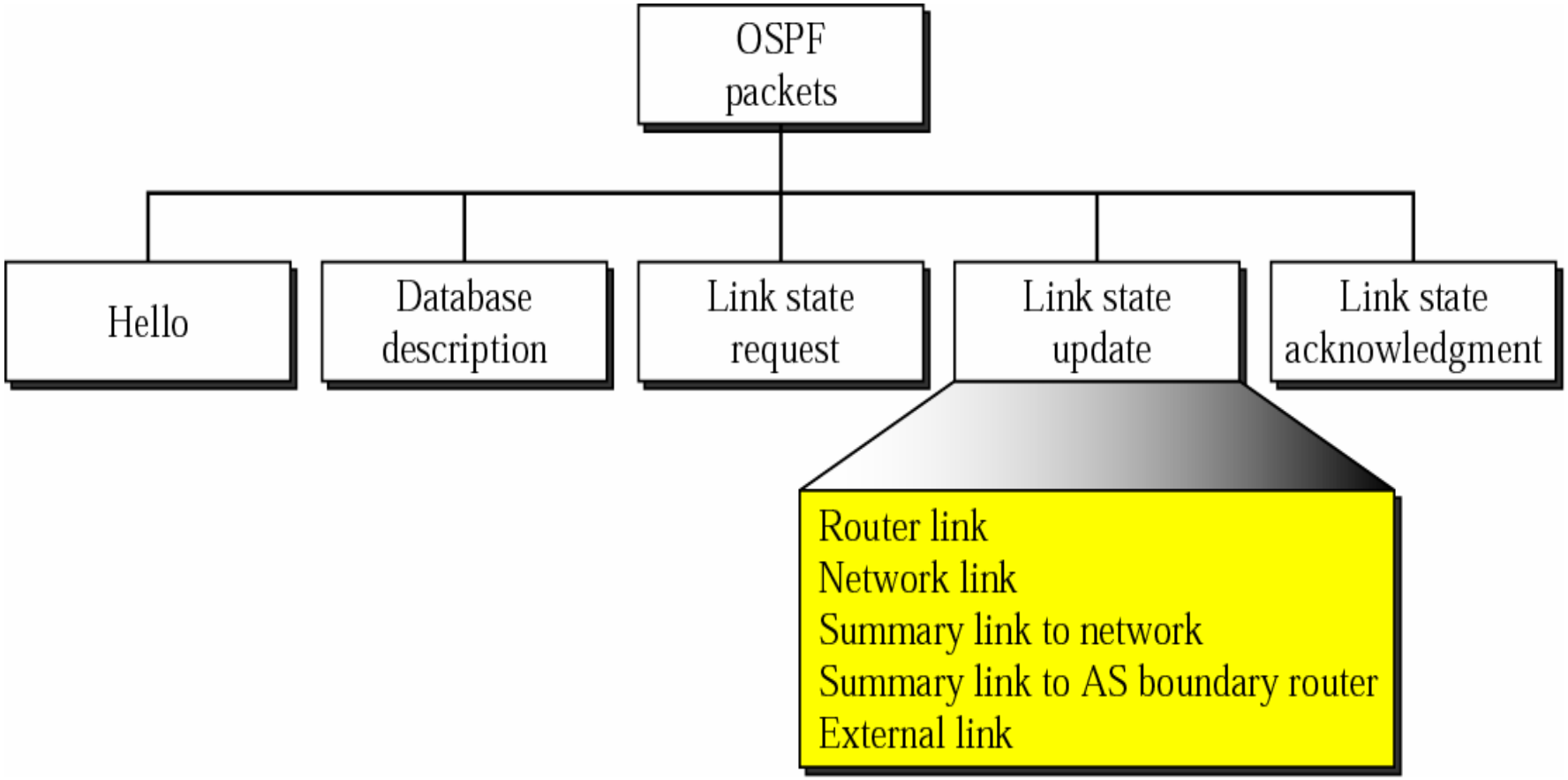
Figure 14.24 *Example of an AS and its graphical representation in OSPF*



Types of Packets

- OSPF uses five different packets
  - Hello packet
  - Database description packet
  - Link state request packet
  - Link state update packet
    - Router link
    - Network link
    - Summary link to network
    - Summary link to AS boundary router
    - External link
  - Link state acknowledgment packet

Types of OSPF Packets

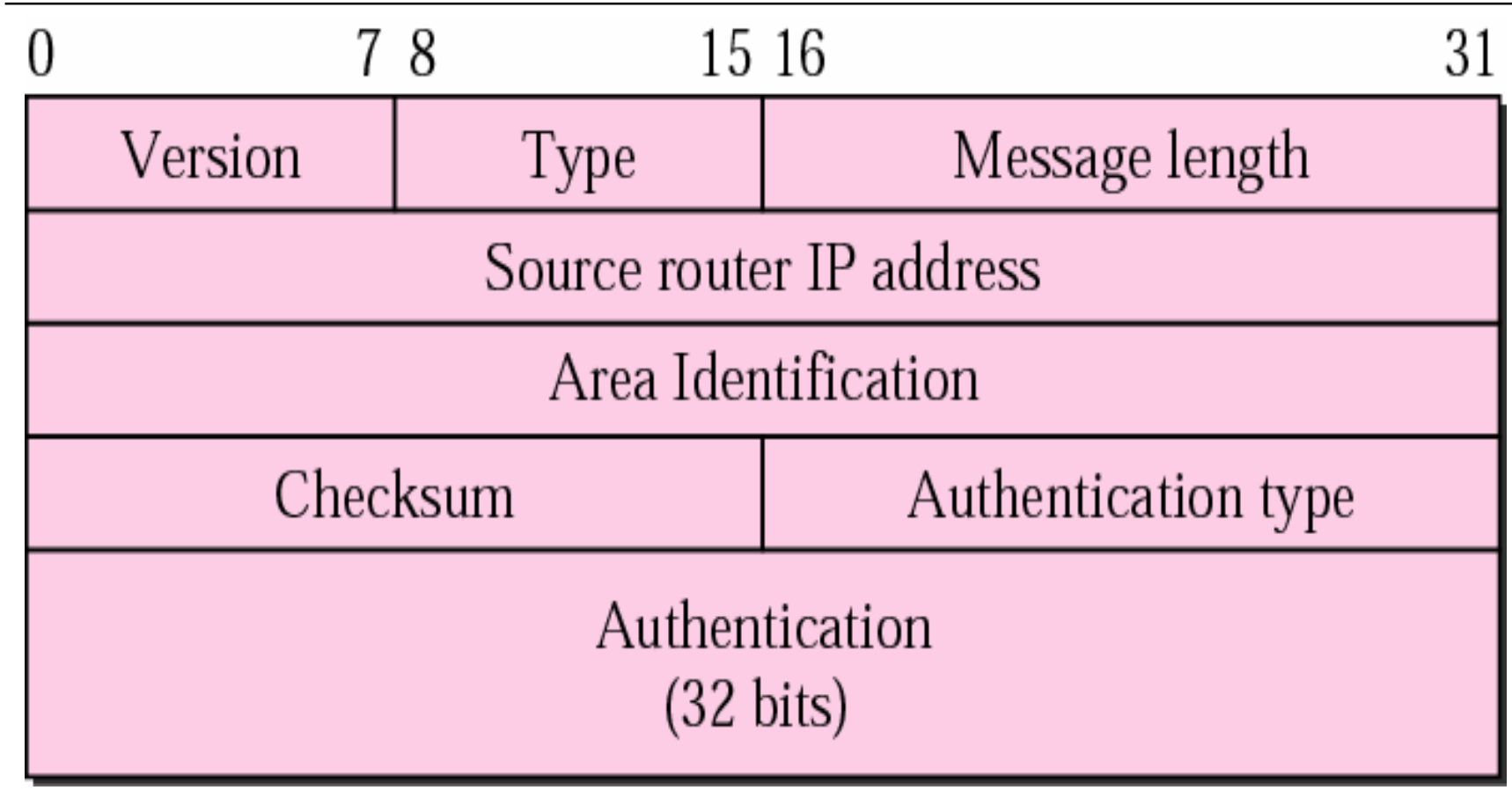


Common Header

- All OSPF packets share the same header
  - Version: 8-bit
    - The version of the OSPF protocol. Currently, it is 2
  - Type: 8-bit
    - The type of the packet
  - Message length: 16-bit
    - The length of the total message including the header
  - Source router IP address: 32-bit
    - The IP address of the router that sends the packet

Figure 12.24

OSPF Common Header



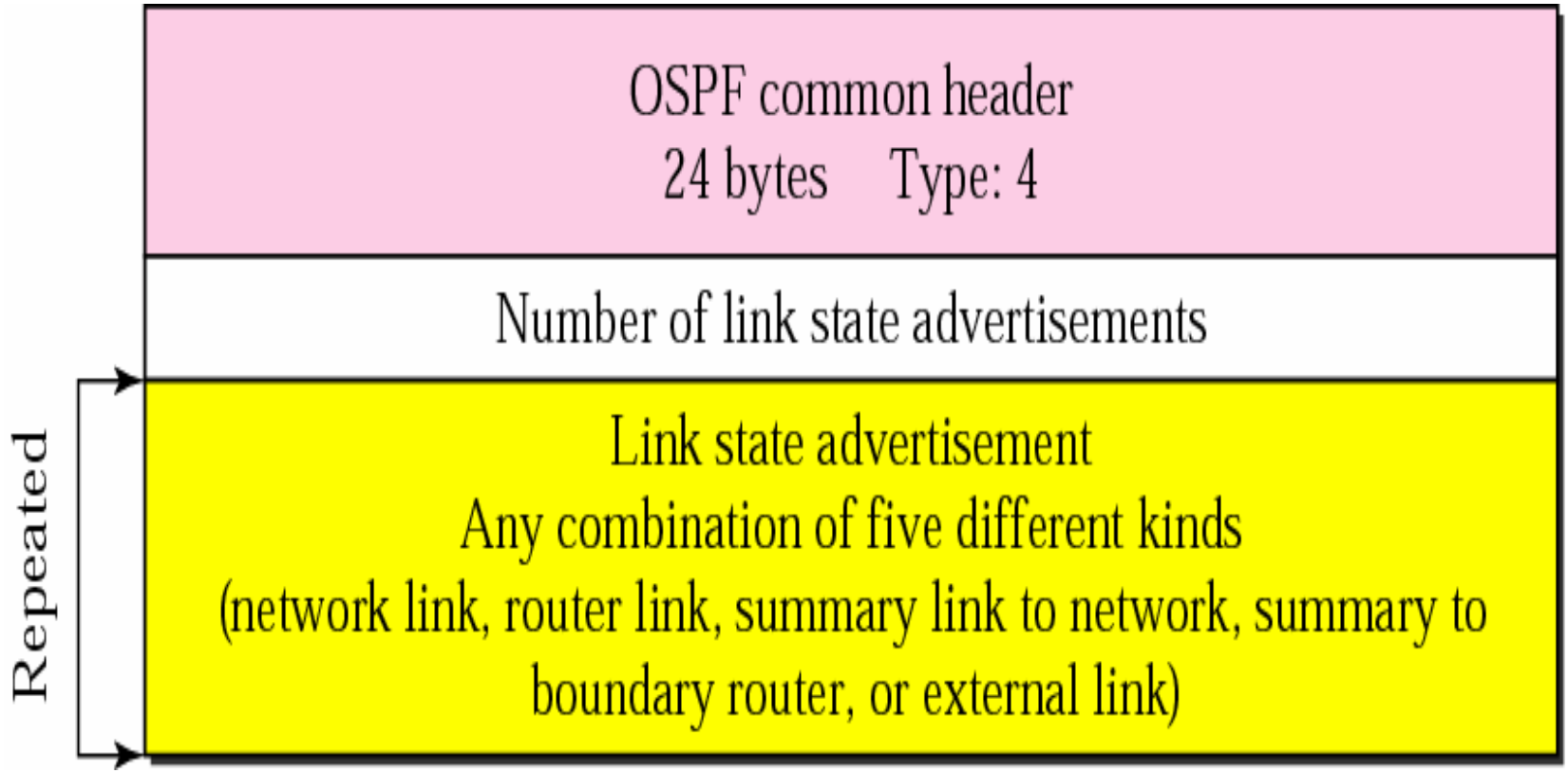
Common Header (Cont.)

- Area identification: 32-bit
  - The area within which the routing take place
- Checksum: 16-bit
  - Error detection on the entire packet excluding the authentication type and authentication data field
- Authentication type: 16-bit
  - Define the authentication method used in this area
  - 0: none, 1: password
- Authentication: 64-bit
  - The actual value of the authentication data
  - Filled with 0 if type = 0; eight-character password if type = 1

Link State Update Packet

- Used by a router to advertise the state of its links
- Each update packet may contain several different LSAs (List State Advertisement)
- Packet format
  - Number of link state advertisements: 32-bit
  - Link state advertisement
    - There are five different LSAs, as discussed before
    - All have the same general header, but different bodies

Figure 14.27 *Link state update packet*



LSA General Header

- Link state age: the number of seconds elapsed since this message was first generated
  - LSA goes from router to router, i.e., flooding
  - When a router create a message, age = 0
  - When each successive router forwards this message
    - Estimate the transmit time and add it to the age field
- E flag: if 1, the area is a stub area
  - i.e., an area that is connected to the backbone area by only one path

LSA General Header (Cont.)

- T flag: if 1, the router can handle multiple types of service
- Link state type
  - 1: router link
  - 2: network link
  - 3: summary link to network
  - 4: summary link to AS boundary router
  - 5: external link

LSA General Header (Cont.)

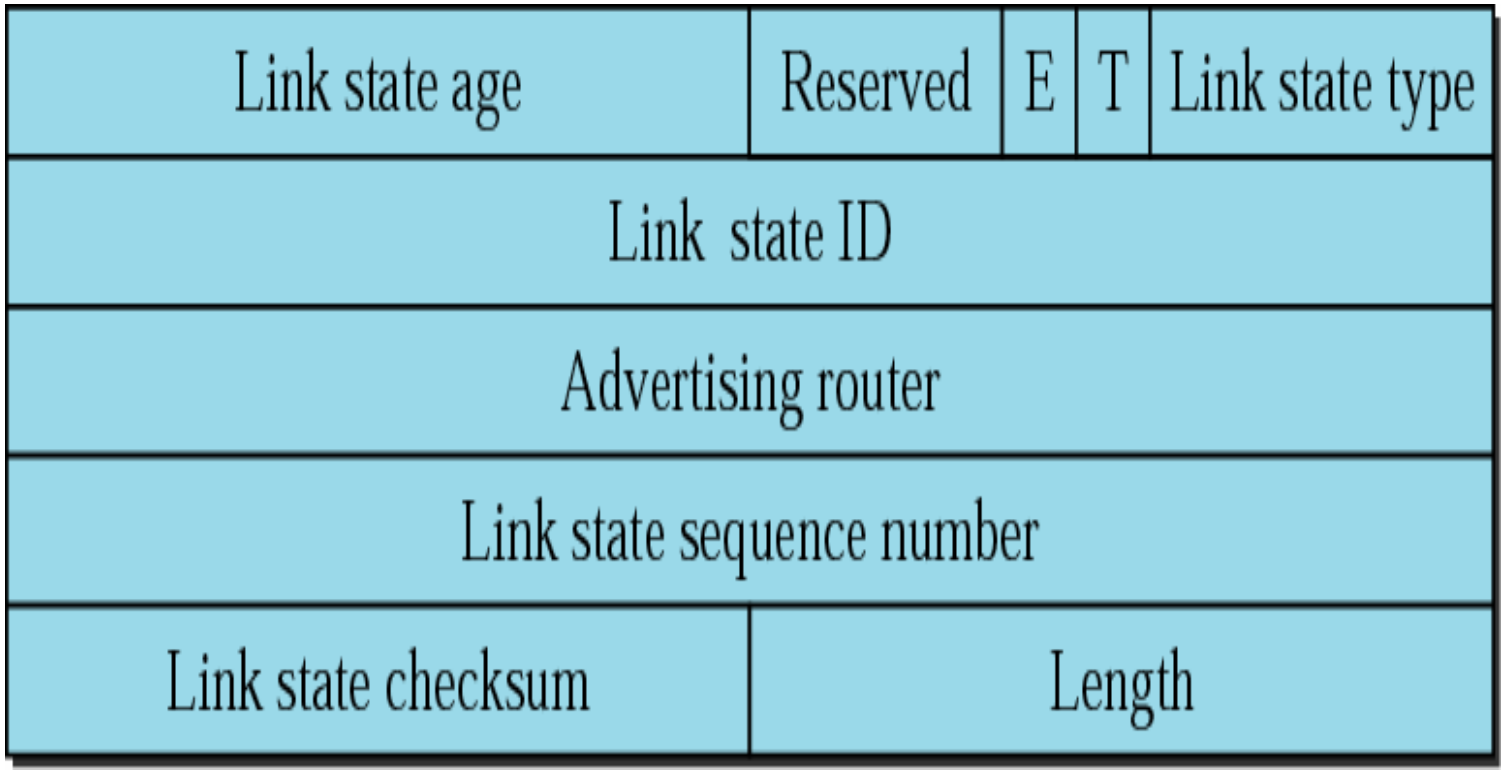
- Link state ID: depend on the type of link
  - Router link: IP address of the router
  - Network link: IP address of the designated router
  - Summary link to network: address of the network
  - Summary link to AS boundary router: IP address of the AS boundary router
  - External link: address of the external network

LSA General Header (Cont.)

- Advertisement router:
  - IP address of the router advertising this message
- Link state sequence number:
  - Sequence number assigned to each link state update message
- Link state checksum:
  - A special checksum algorithm: Fletcher's checksum
- Length:
  - Total packet length

Figure 13-29

LSA General Header

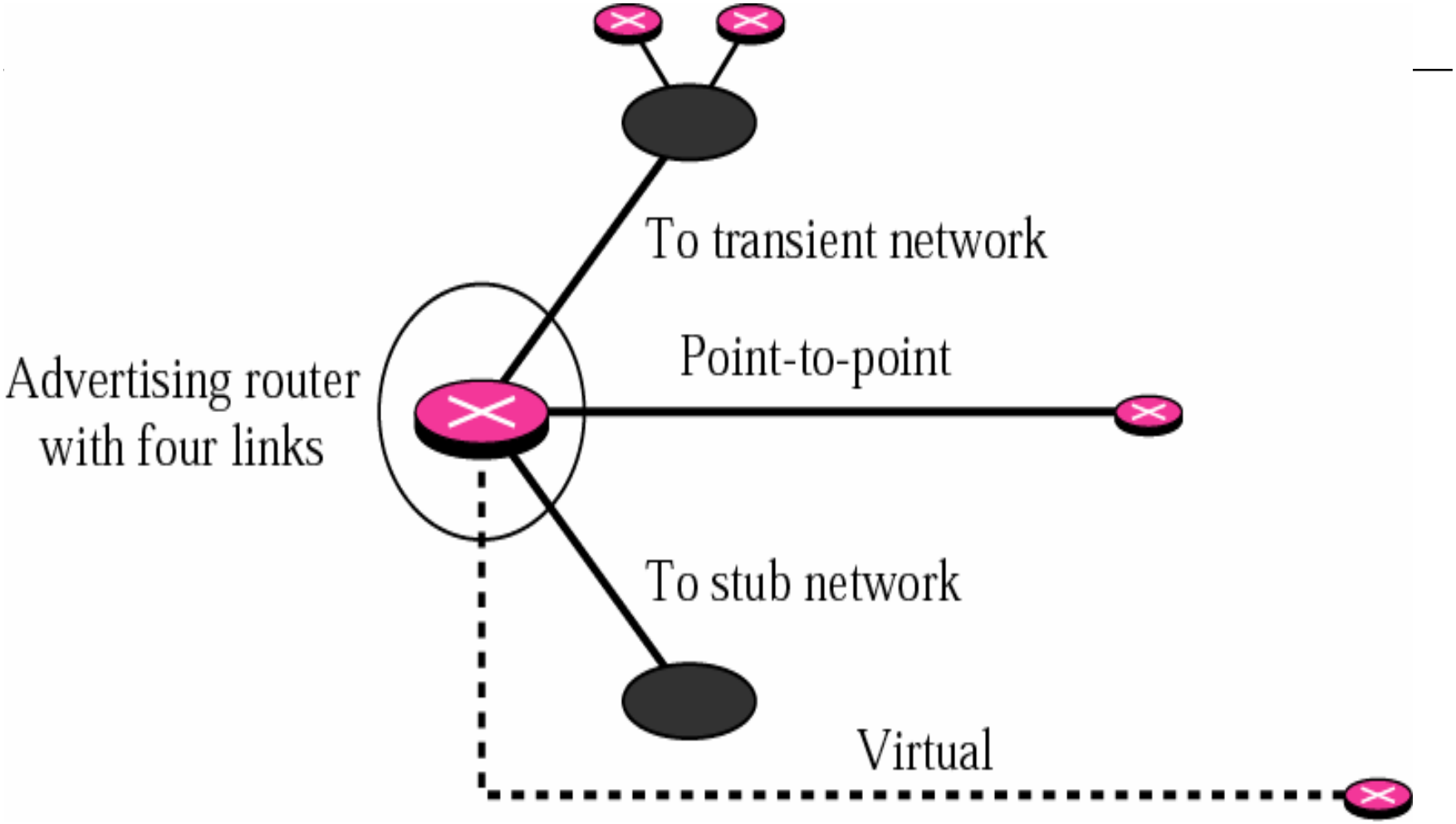


Router Link LSA

- Define the link of a true router

- A true router uses this advertisement to announce information about
  - All of its links
  - What is at the other side of the links (neighbors)

Figure 14.29 *Router link*



Router Link LSA (Cont.)

- **Format**
  - Link ID:
    - Depend on the type of link, see Table 14.2
  - Link data:
    - Give additional information about the link, also depend on the type of link, see Table 14.2
  - Link type:
    - Four different types of links are defined based on the type of network, see Table 14.2
  - Number of types of services (TOS)
    - The number of type of services announced for each link
  - Metric for TOS 0:
    - Define the metrics for the default type of service (TOS 0)
  - TOS:
    - Define the type of service
  - Metric:
    - Define the metric for the corresponding TOS

Figure 14.30 *Router link LSA*

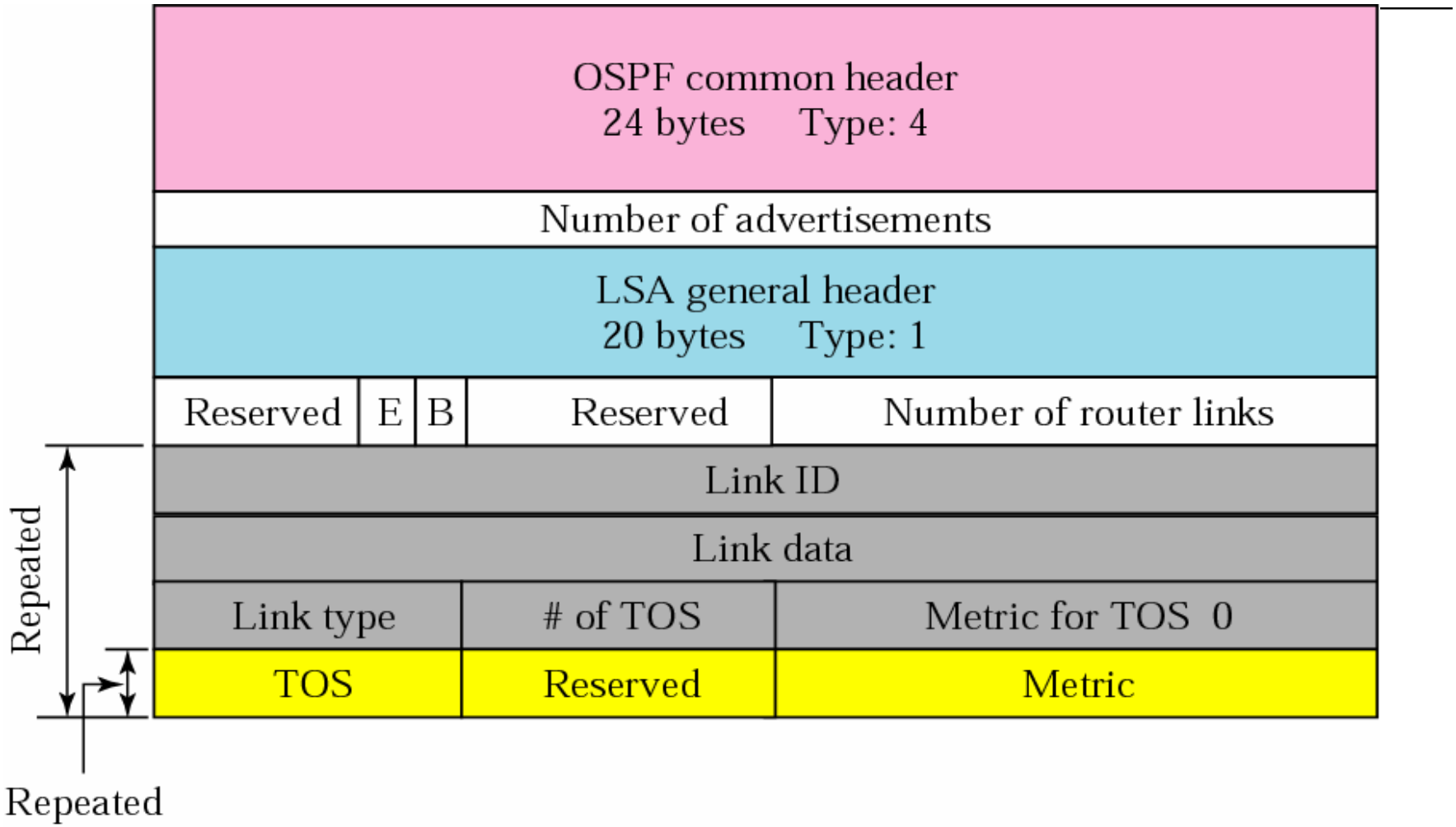


Table 14.2 *Link types, link identification, and link data*

Link Type	*Link Identification*	*Link Data*
Type 1: Point-to-point	Address of neighbor router	Interface number
Type 2: Transient	Address of designated router	Router address
Type 3: Stub	Network address	Network mask
Type 4: Virtual	Address of neighbor router	Router address

Example 3

Give the router link LSA sent by router 10.24.7.9 in Figure 14.31.

- This router has three links
  - Two of type 1 (point-to-point)
  - One of type 3 (stub network)
- Figure 14.32 shows the router link LSA

Figure 14.31 *Example 3*

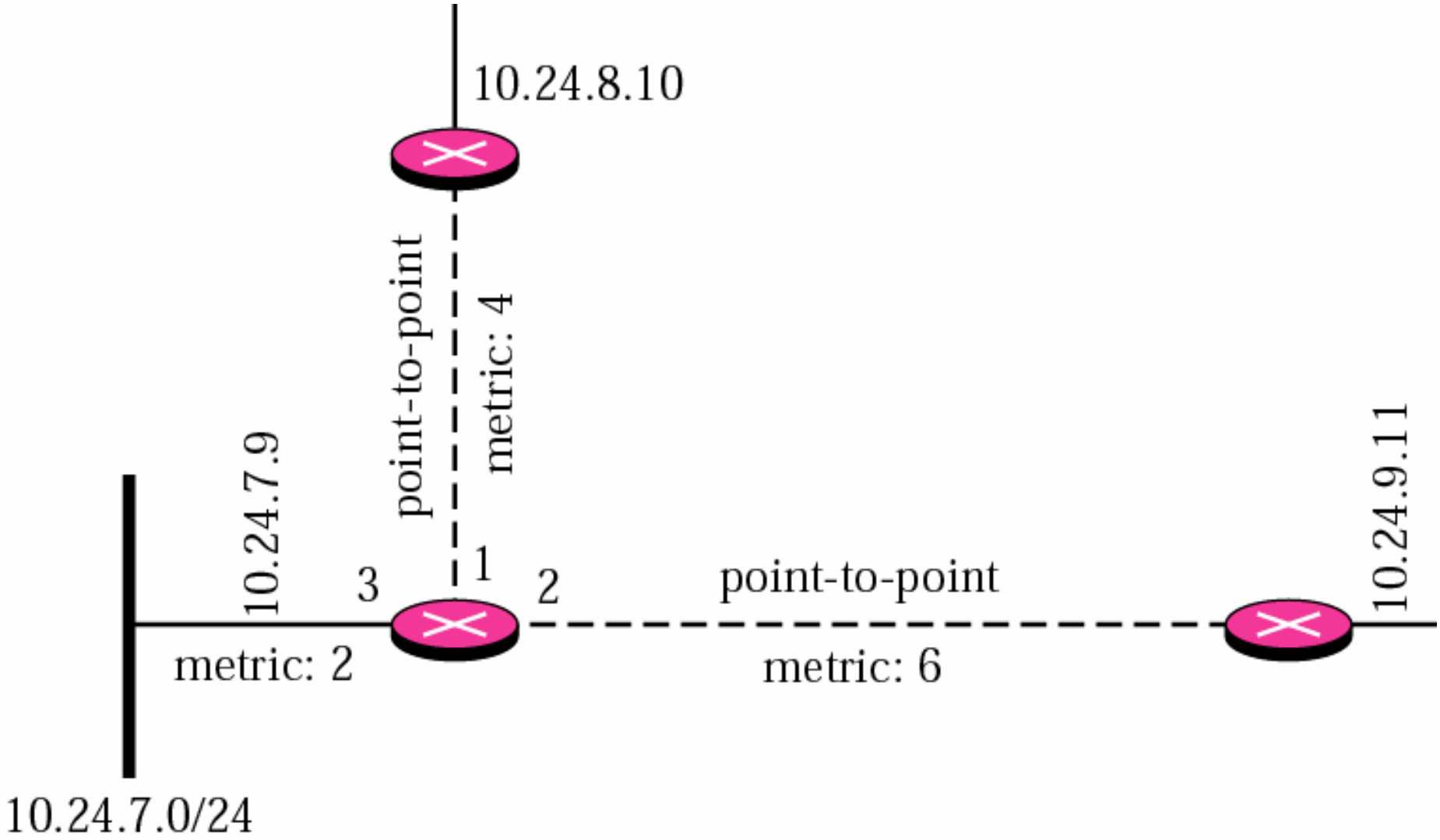
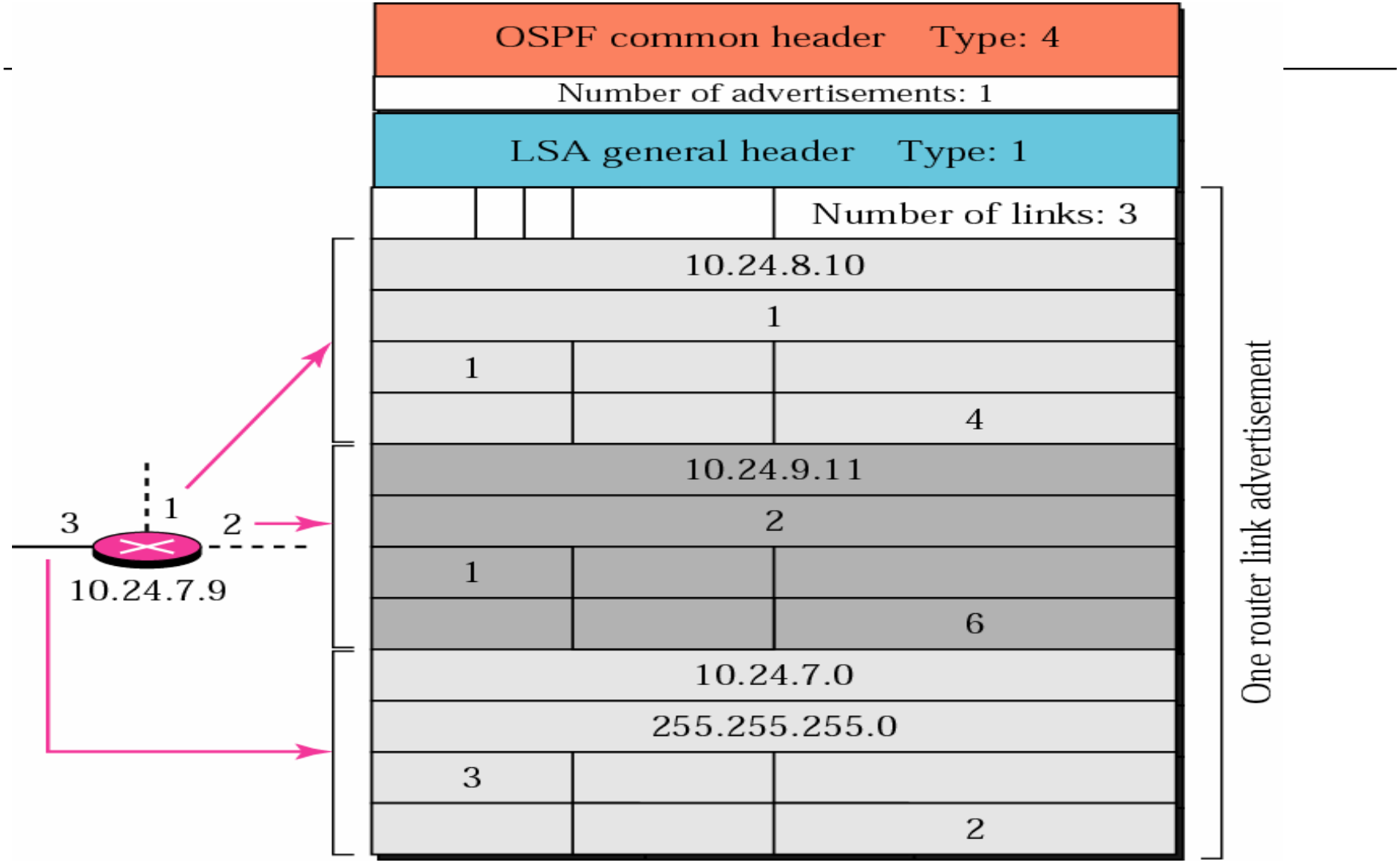


Figure 14.32 *Solution to Example 3*



Network Link LSA

- Define the links of a network
- A ***designed router***, on behalf of the ***transient network***, distributes this type of LSA packet
- Announce the *existence of all of the routers connected to* the network
  - See Fig. 14.33

Network Link LSA (Cont.)

- **Format**
  - Network mask
    - Define the network mask
  - Attached router
    - Define the IP addresses of all attached routers

Figure 14.33 *Network link*

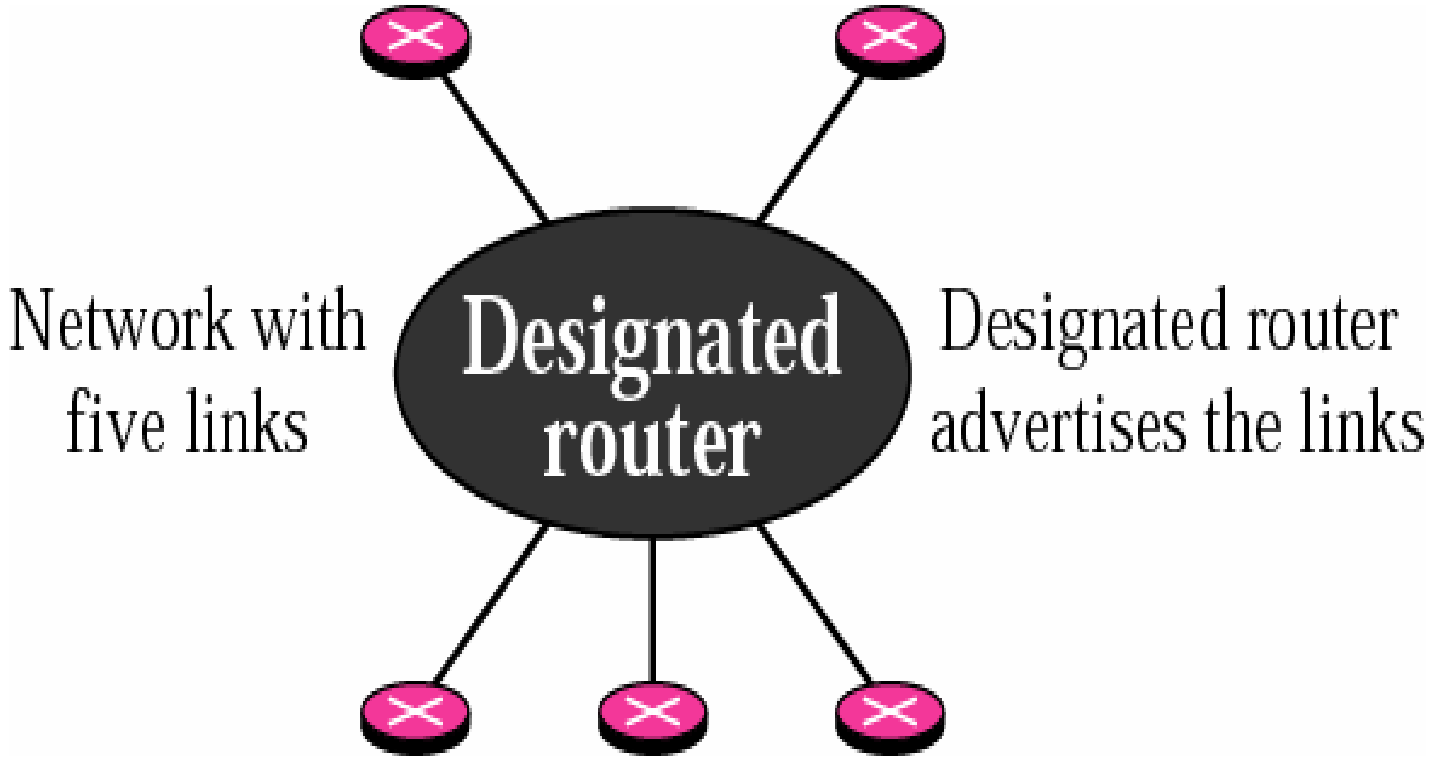
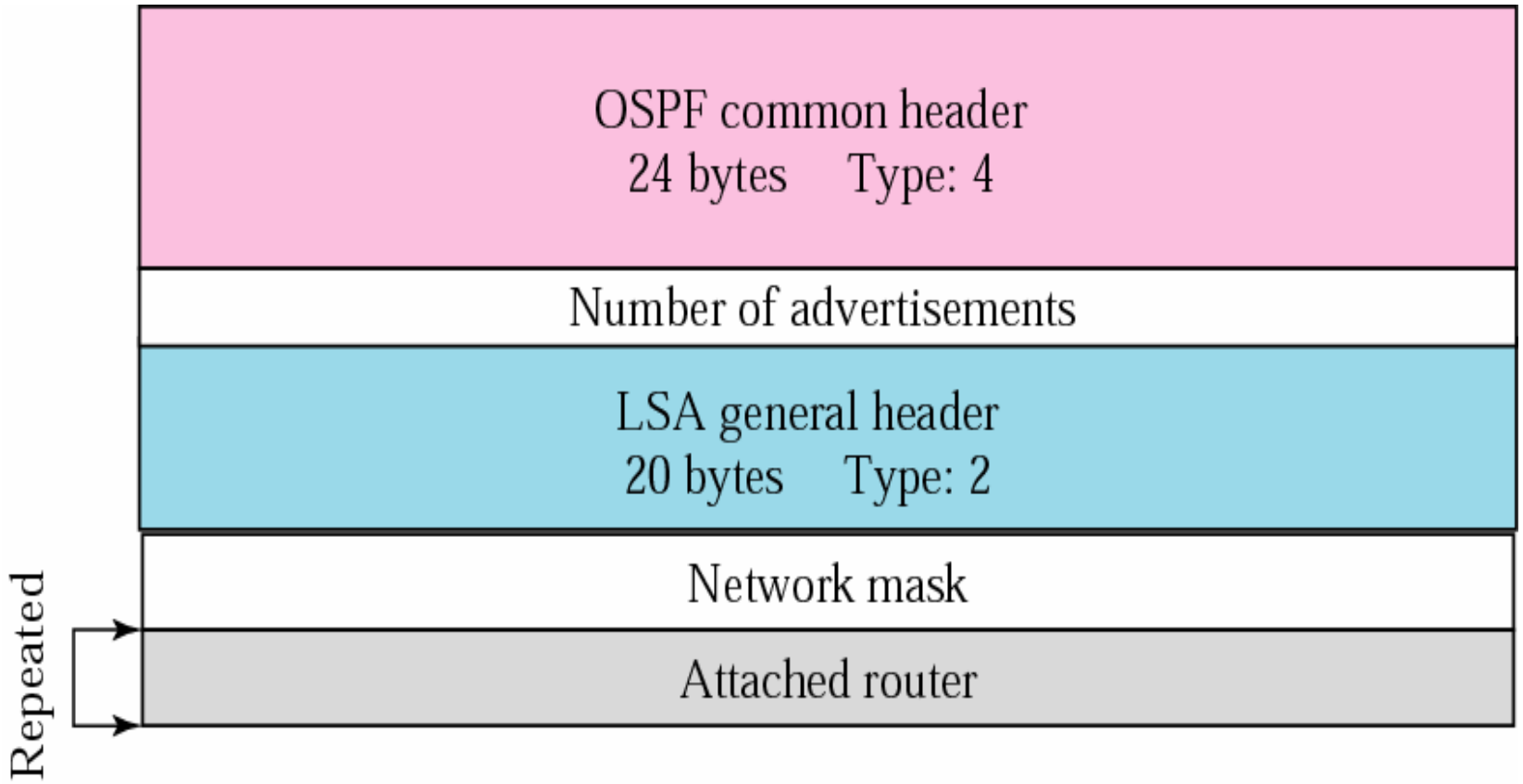
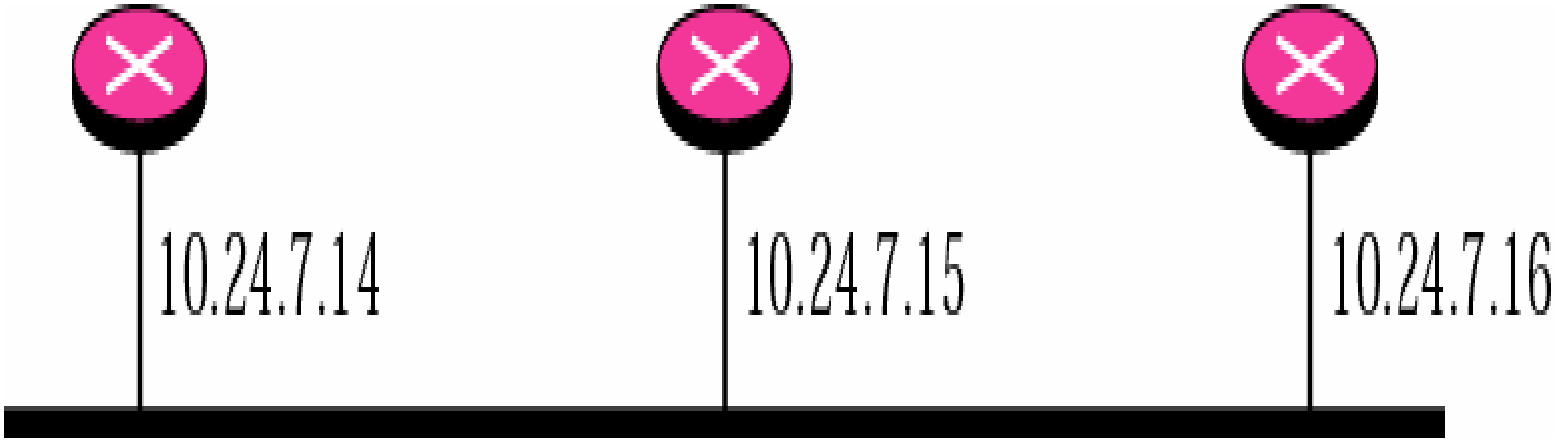


Figure 14.34 *Network link advertisement format*



Example 4

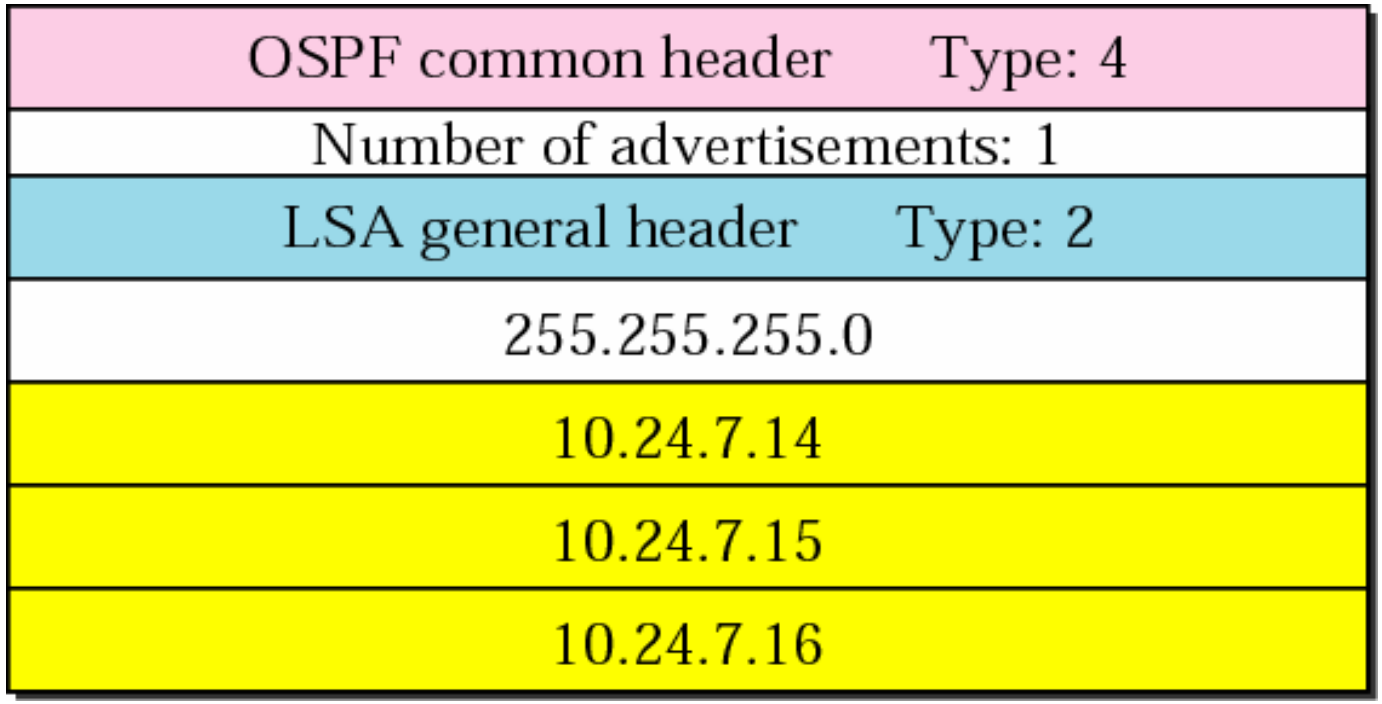
Give the network link LSA in the following Figure.



Solution

The network, for which the network link advertises, has three routers attached. The LSA shows the mask and the router addresses. See Figure 14.36.

Figure 14.36 *Solution to Example 4*



Example 5

In Figure 14.37, which router(s) sends out router link LSAs?

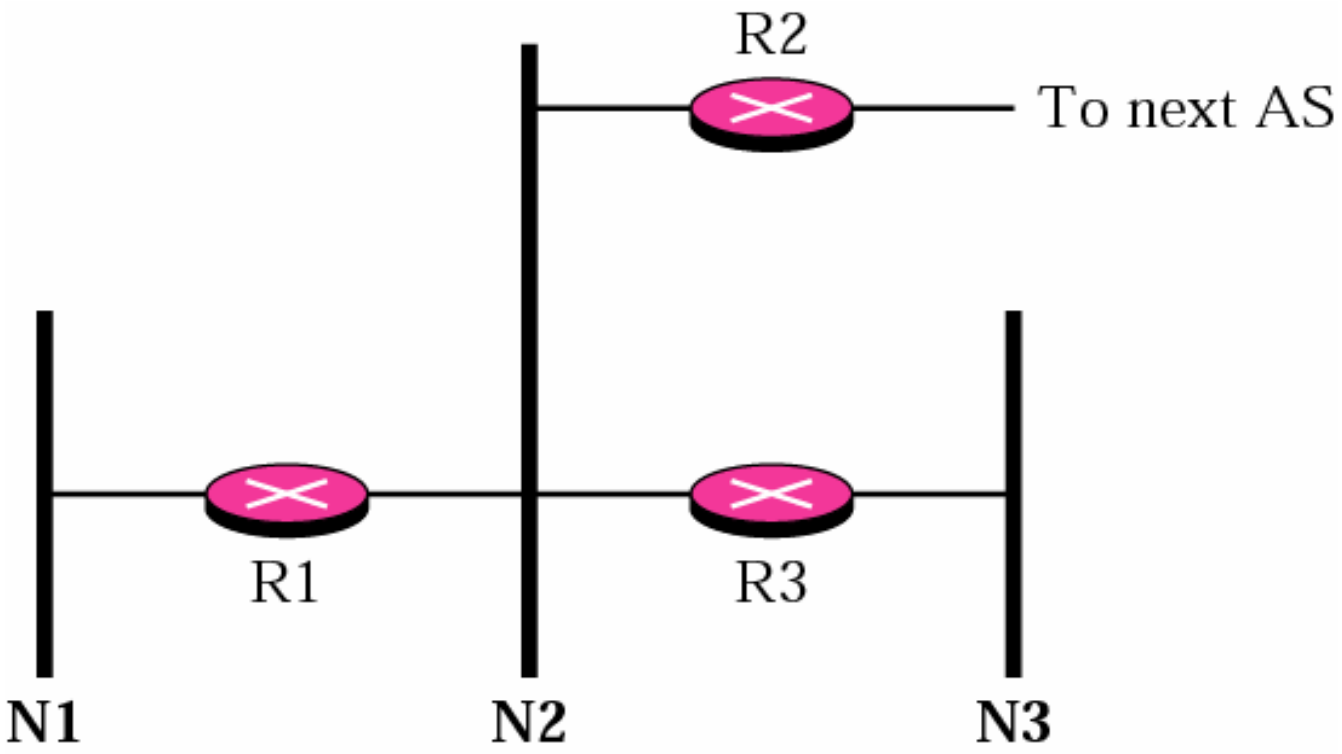
See Next Slide

Solution

All routers advertise router link LSAs.

- *a. R1 has two links, N1 and N2.*
- *b. R2 has one link, N1.*
- *c. R3 has two links, N2 and N3.*

Figure 14.37 *Example 5 and Example 6*



Example 6

In Figure 14.37, which router(s) sends out the network link LSAs?

Solution

All three network must advertise network links:

a. Advertisement for N1 is done by R1 because it is the only attached router and therefore the designated router.

b. Advertisement for N2 can be done by either R1, R2, or R3, depending on which one is chosen as the designated router.

c. Advertisement for N3 is done by R3 because it is the only attached router and therefore the designated router.

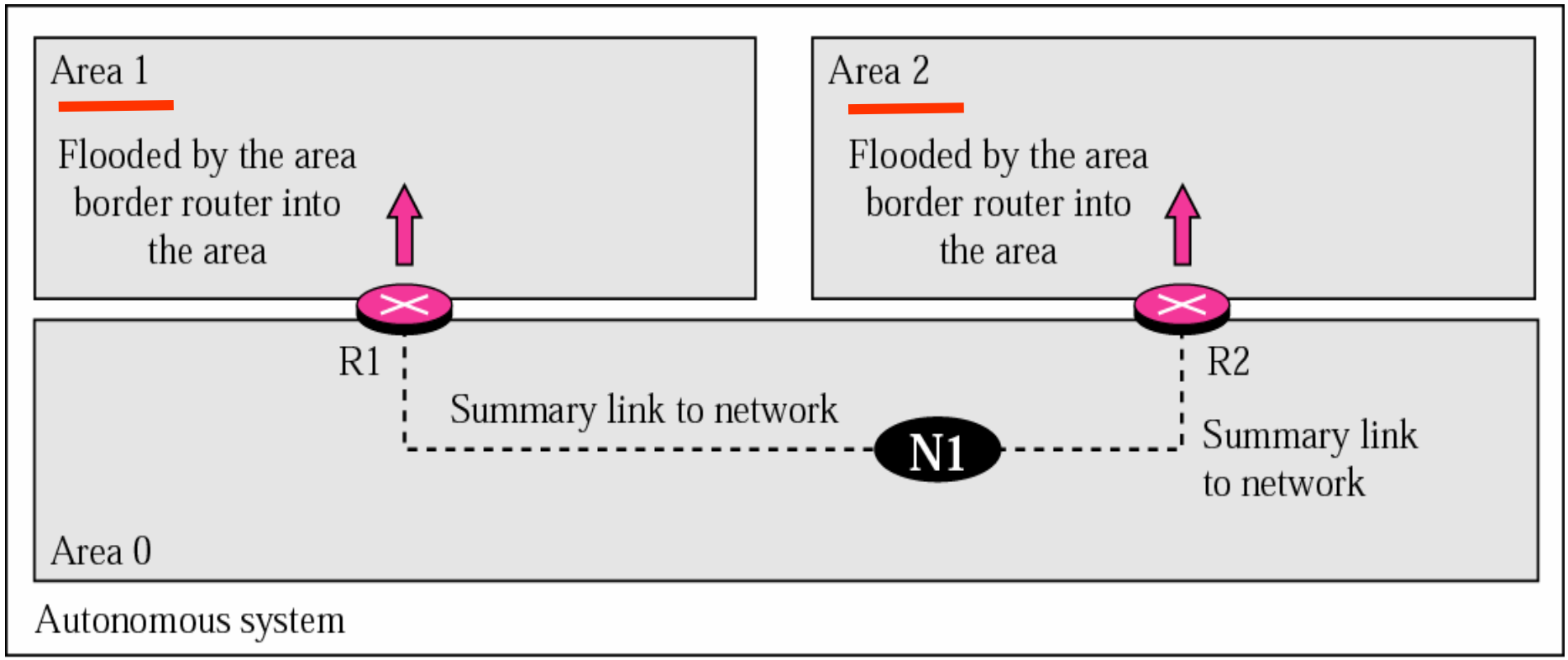
Summary Link to Network LSA

- Router link and network link advertisements
  - Flood the area with information inside an area
- But a router must also know about the networks outside its area
  - The *area border routers* provide this information
- An area border router is active in more than one area
  - Receive *router link* and *network link advertisements*
  - Create a router table for each area
  - Provide one area's information to other areas by the *summary link to network advertisement*

Example

- R1 is an area border router and has two routing tables
  - One for area 1 and one for area 0
- R1 will flood *area 1* with information about how to reach a network located in *area 0*
- R2 plays the same role

Figure 14.38 *Summary link to network*

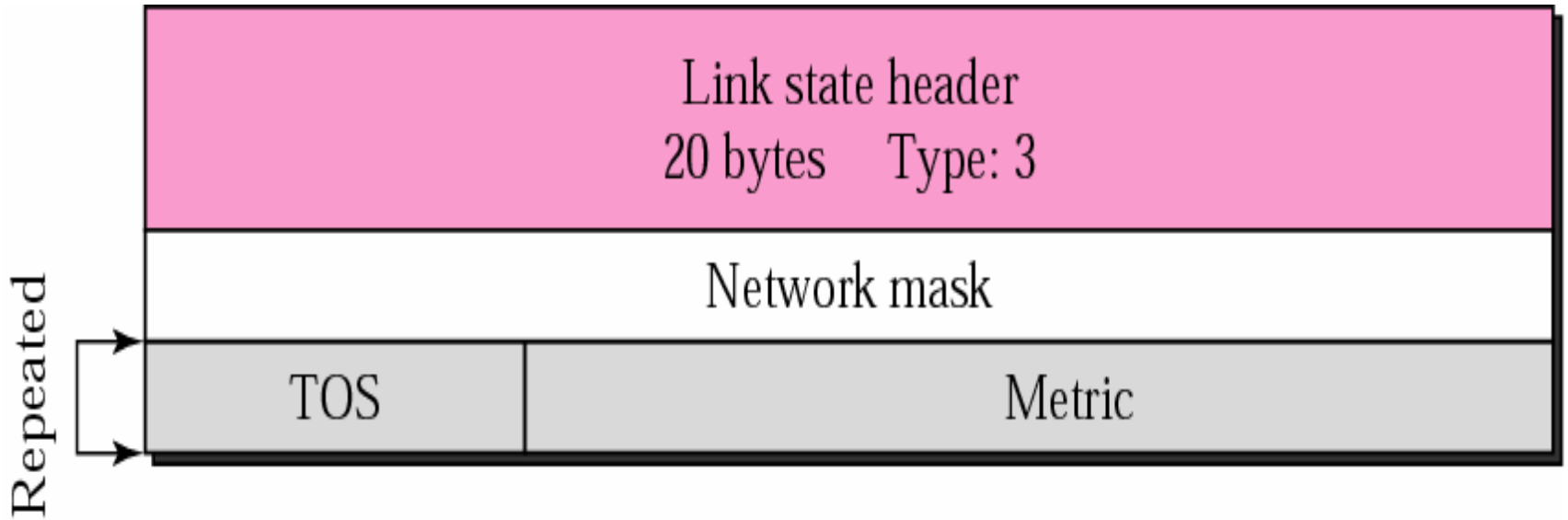


Summary Link to Network LSA (Cont.)

- The LSA consists of only network mask and metric for each type of service
  - Not include the *network address*
  - Since the IP address of the advertising router is in the header
- Each advertisement announces *only one network*
  - If more than one network, a separate advertisement must be issued for each
- Format
  - Network mask
  - TOS:
    - Type of service
  - Metric:
    - Metric for the type of service defined in the TOS field

Figure 13-46

Summary Link to Network LSA

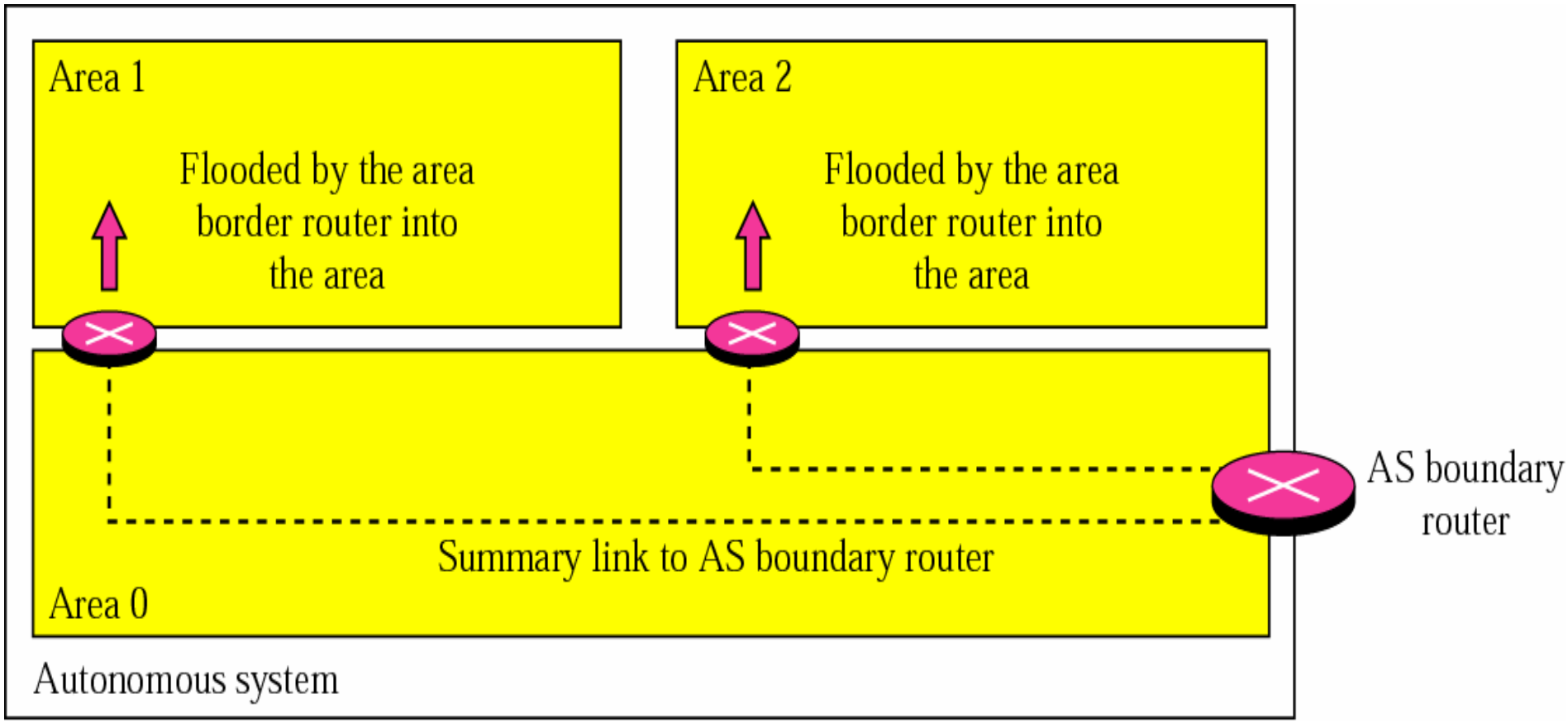


Summary Link to AS Boundary Router LSA

- Previous advertisement lets every router know the cost to reach all of the networks inside the AS
- But, how to reach a network outside an AS?
- A router must know *how to reach the autonomous boundary router first*
- The *summary link to AS boundary router* provides this information
  - The *area border routers* flood their area with this information

Figure 13-29

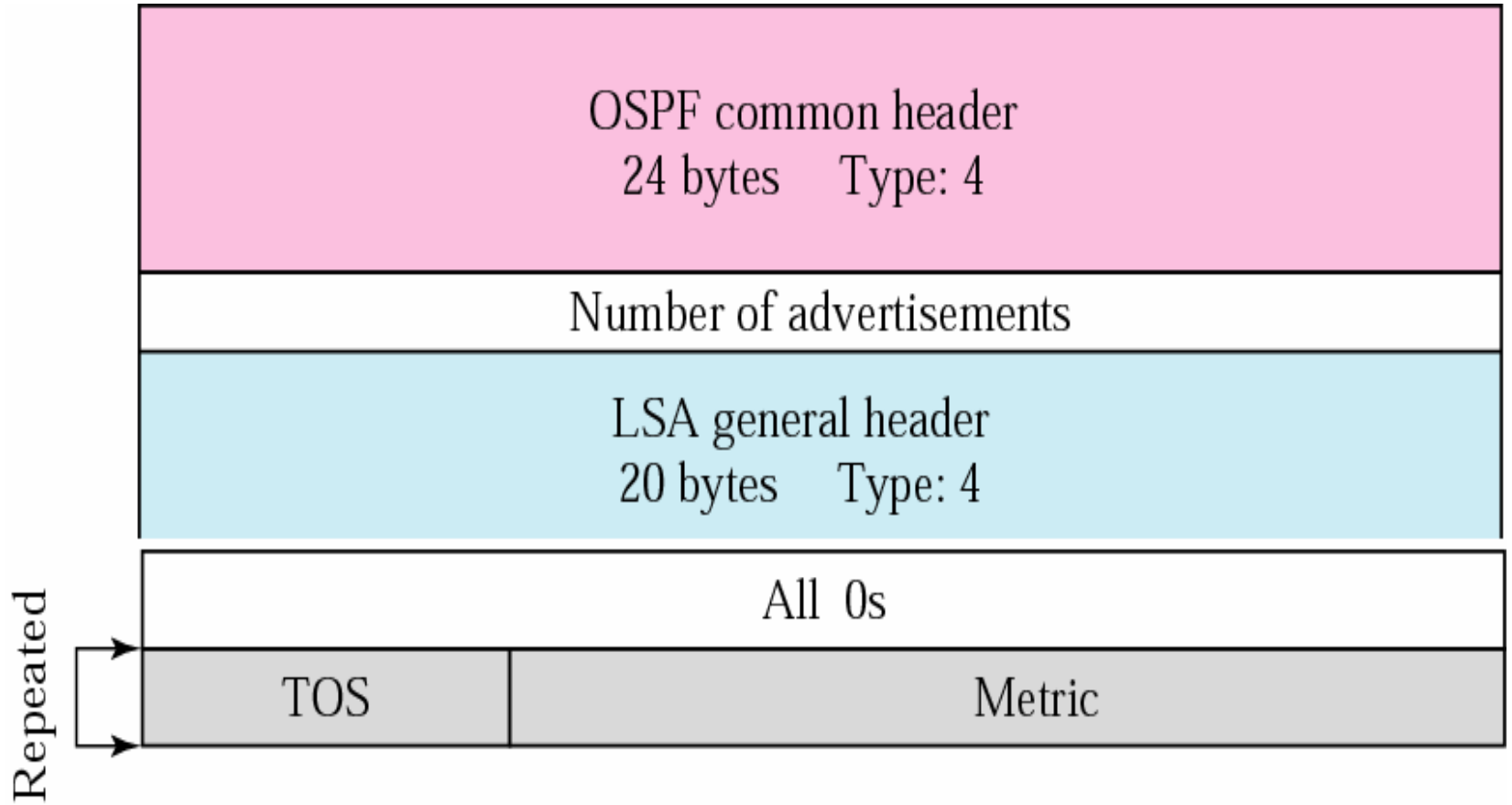
Summary Link to AS Boundary Router LSA



Summary Link to AS Boundary Router LSA (Cont.)

- Announce the *route to an AS boundary router*
  - Define the network to which the AS boundary router is attached
  - The *area border routers* flood their area with this LAS
- Format
  - The same as the summary link to network LSA

Figure 14.41 *Summary link to AS boundary router LSA*



External Link LSA

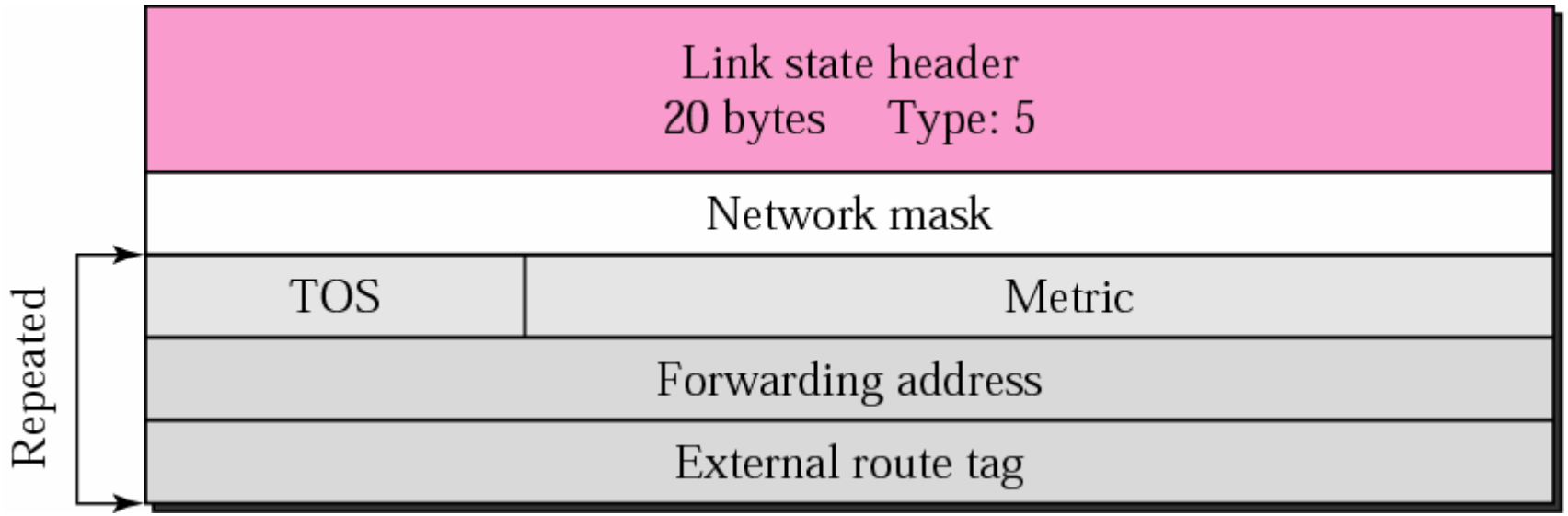
- How a router inside an AS know which networks are available outside the AS ?
  - The ***external link advertisement*** provides this information
- The *AS boundary routers* floods the autonomous system with the cost of each network outside the AS
  - Using a routing table created by an *exterior routing protocol*
- Notably, each advertisement announces one single network
  - Separate announcements are made if more than one network exists Announce all the networks outside the AS

External Link LSA

- Use to announce all of the networks outside the AS
- Format: similar to the summary link to AS boundary router LSA but add two fields
  - *Forwarding address*
    - May define a *forward router* than can provide a better route to the destination
  - *External route tag*
    - Used by other protocol, but not by OSPF

Figure 12.48

External Link LSA



Other Packets

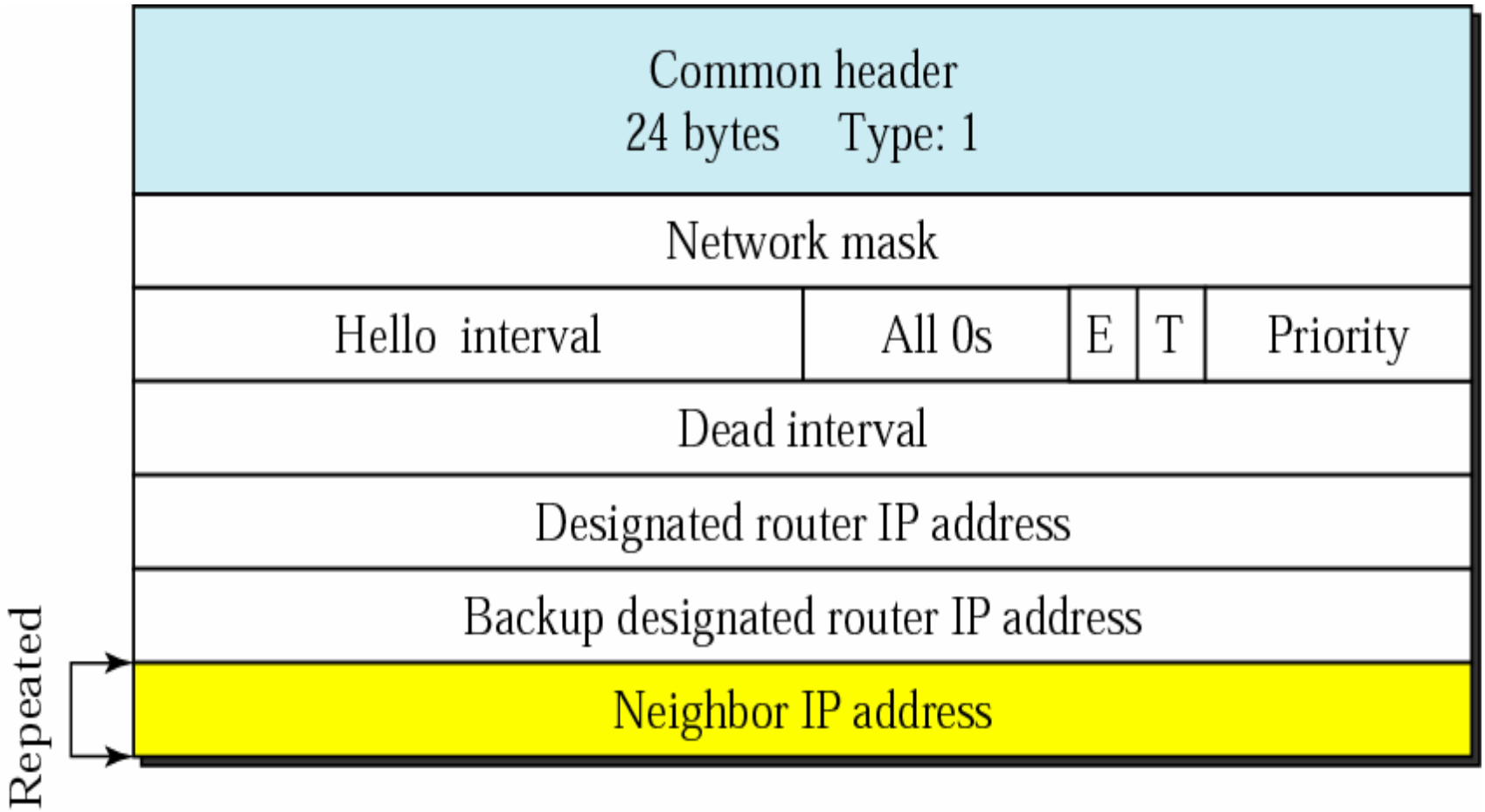
- Not used as LSA but are essential to the OSPF
  - Hello message
  - Database description message
  - Link state request packet
  - Link state acknowledgment packet

Hello Message

- OSPF uses the hello message to
  - Create neighborhood relationships
  - Test the reachability of neighbors
- First step in link state routing
  - It must first greet its neighbors

Figure 12-25

Hello Packet



Hello Packet Format

- Network mask: 32-bit
  - Define the network mask of the network over which the hello message is sent
- Hello interval: 16-bit
  - Define the number of seconds between hello message
- E flag: 1-bit
  - If it is set, the area is a stub area
- T flag: 1-bit
  - If it is set, the router supports multiple metrics

Hello Packet Format (Cont.)

- Priority
  - The priority of the router. Used for the selection of the designated router
  - The router with the highest priority is chosen as the *designated router*
  - The router with the second highest priority is chosen as the *backup designated router*
  - If it is 0, the router never wants to be a designated or backup designated router

Hello Packet Format (Cont.)

- Dead interval: 32-bit
  - The number of seconds before a router assumes that a neighbor is dead
- Designated router IP address: 32-bit
- Backup designated router IP address: 32-bit
- Neighbor IP address: a repeated 32-bit field
  - A current list of all the neighbors from which the sending router has received the hello message

Database Description Message

- When a router is connected to the system *for the first time* or *after a failure*
  - It needs the complete link state database immediately
- Thus, it sends hello packets to greet its neighbors
- If this is the first time that the neighbors hear from the router
  - They send a *database description packet*

Database Description Message (Cont.)

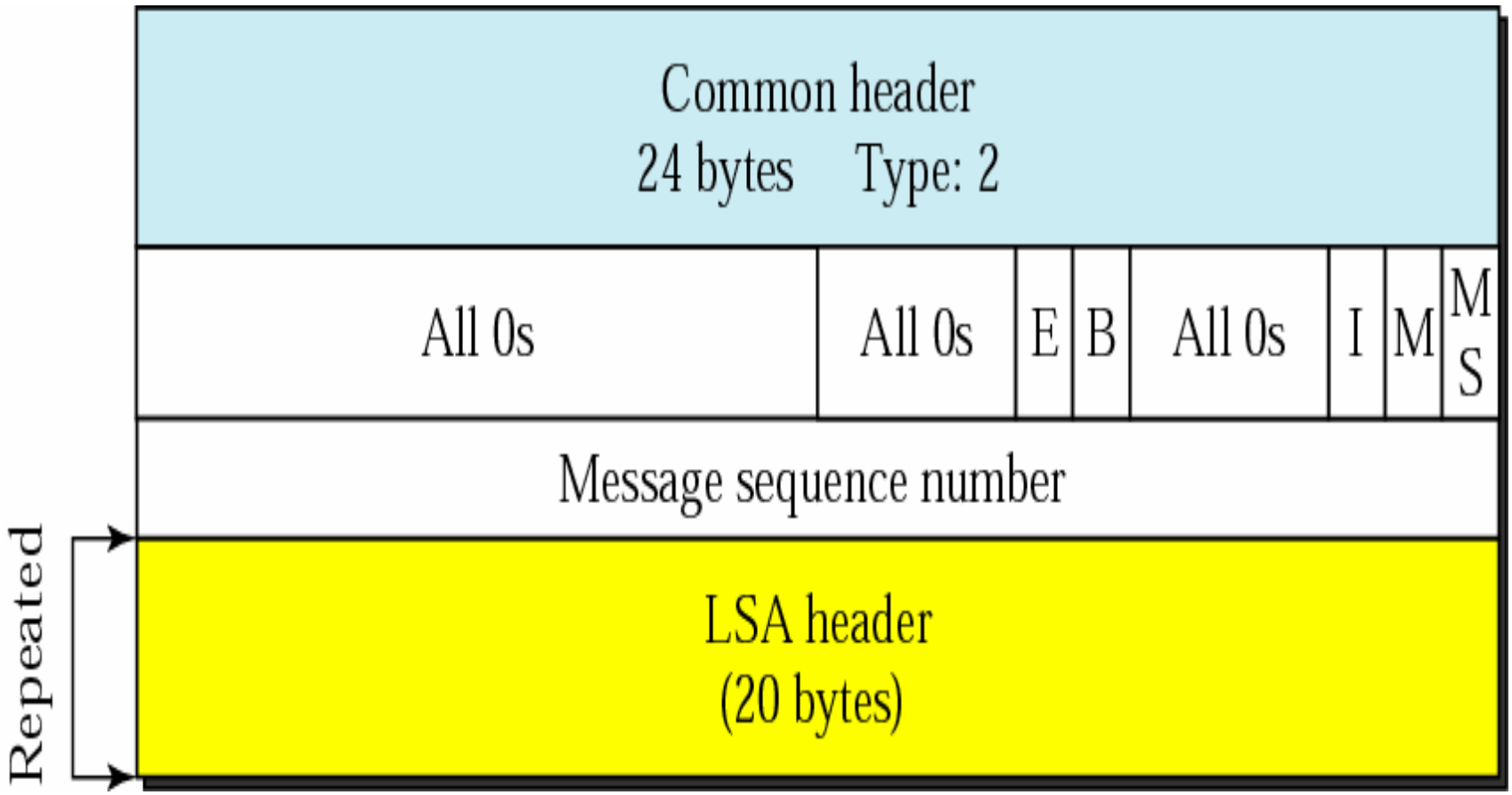
- The database description message does not contain complete database information
  - It only gives an *outline*, the title of each line in the database
- The newly router examines the outline and find out which lines it does not have
  - Send one or more *link state request packets* to get full information about that particular link
  - The content of the database may be divided into several message

Database Description Message (Cont.)

- When two routers want to exchange database description packets
  - One of them acts as mater
  - The other is the slave

Figure 12-26

Database Description Packet



Database Description Message Format

- E flag: 1-bit
  - Set to 1 if the advertising router is an autonomous boundary router
- B flag: 1-bit
  - Set to 1 if the advertising router is an area border router
- I flag: 1-bit, the initialization flag
  - Set to 1 if the message is the first message
- M flag: 1-bit, more flag
  - Set to 1 if this is not the last message

Database Description Message Format (Cont.)

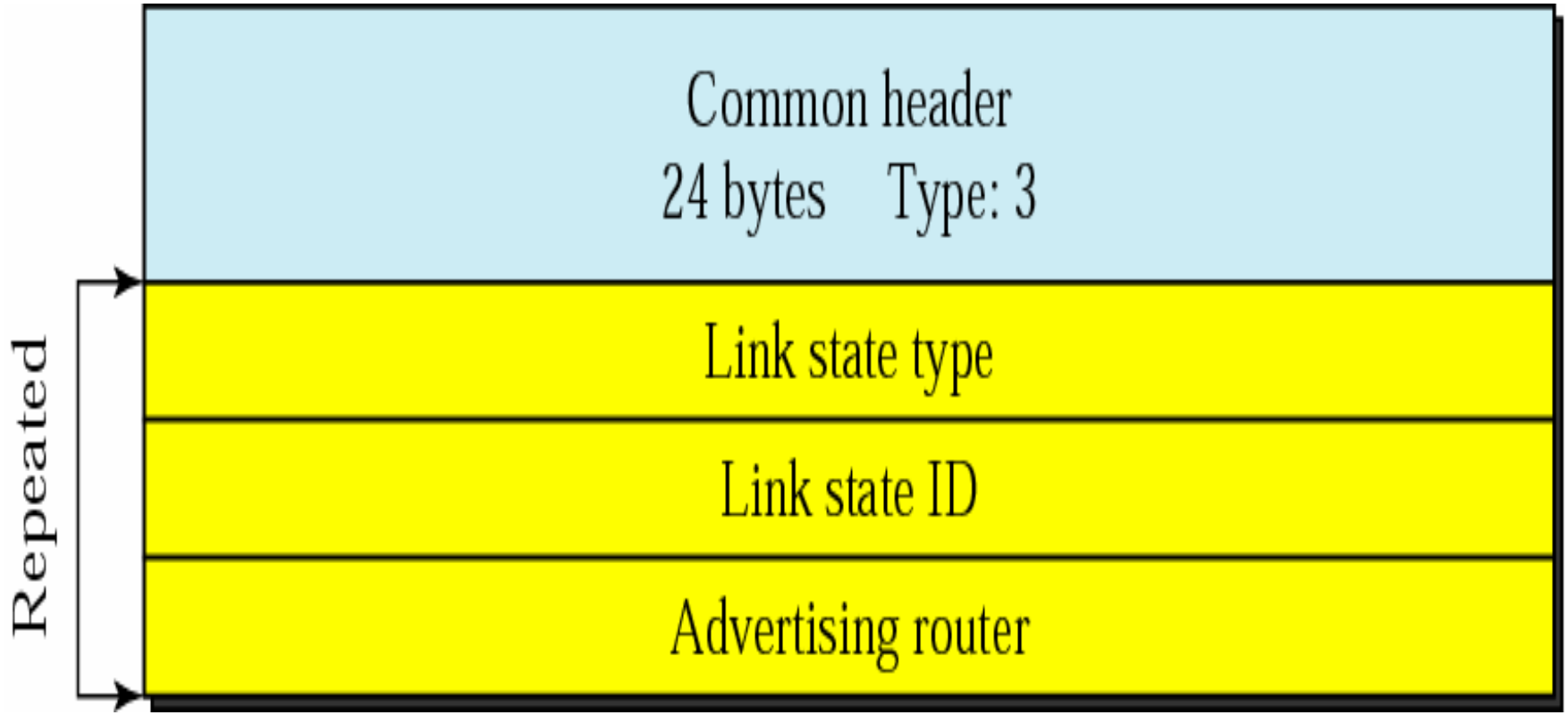
- M/S flag: 1-bit, master/slave flag
  - Indicate the origin of the packet. Master = 1, Slave = 0
- Message sequence number: 32-bit
  - Contain the sequence number of the message
- LSA header: 20-bit
  - Used in each LSA
  - The format of this header is discussed in the *link state update* message
    - Only give the outline of each link
  - It is repeated for each link in the link state database

Link State Request Packet

- Sent by a router that needs information about a specific route or routes
  - Answered with a *link state update packet*
- Used by a newly connected router to request more information after receiving the *database description packet*

Figure 13.27

Link State Request Packet



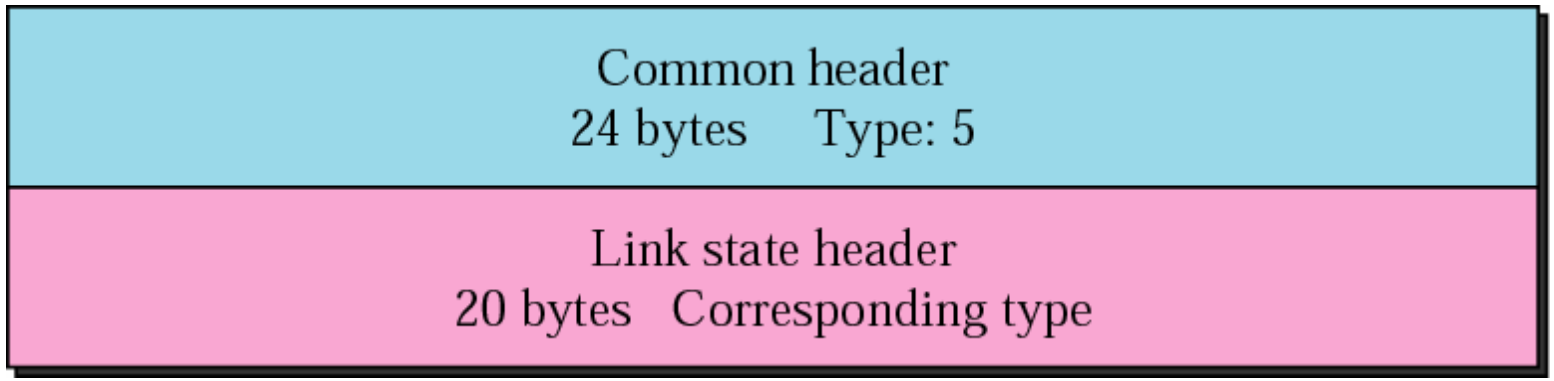
Link State Acknowledgment Packet

- OSPF forces every router to **acknowledge** the receipt of every link state update packet
  - Make routing more reliable

- Format
  - Common header
  - Link state header

Figure 13-49

Link State Acknowledgment Packet



Encapsulation

- OSPF packets are encapsulated in IP datagram
  - OSPF contains the acknowledgment mechanism for flow and error control
  - Doe not need a transport layer protocol to provide these services

14.6 PATH VECTOR ROUTING

Path vector routing is similar to distance vector routing. There is at least one node, called the speaker node, in each AS that creates a routing table and advertises it to speaker nodes in the neighboring ASs..

The topics discussed in this section include:

Initialization
Sharing
Updating

Path Vector Routing

- Why *distance vector routing* and *link state routing* are not good candidates for interdomain routing?

- Mostly because of ***scalability***
  - Intractable when the domain of operation becomes large

Path Vector Routing (Cont.)

- **Distance vector routing**
  - Unstable if there is more than a few hops in the domain
- **Link state routing**
  - An internet is usually too big for this routing method
    - Need a huge amount of resources to calculate routing table
    - Create heavy traffic because of flooding
- **Solution**
  - ***Path vector routing***

Path Vector Routing (Cont.)

- There is one node in each AS to act on behalf of the entire SA
  - Called it ***speaker node***

- The speaker node in an AS
  - Creates a routing table
  - Advertises the routing table to speaker nodes in the neighboring ASs

Path Vector Routing (Cont.)

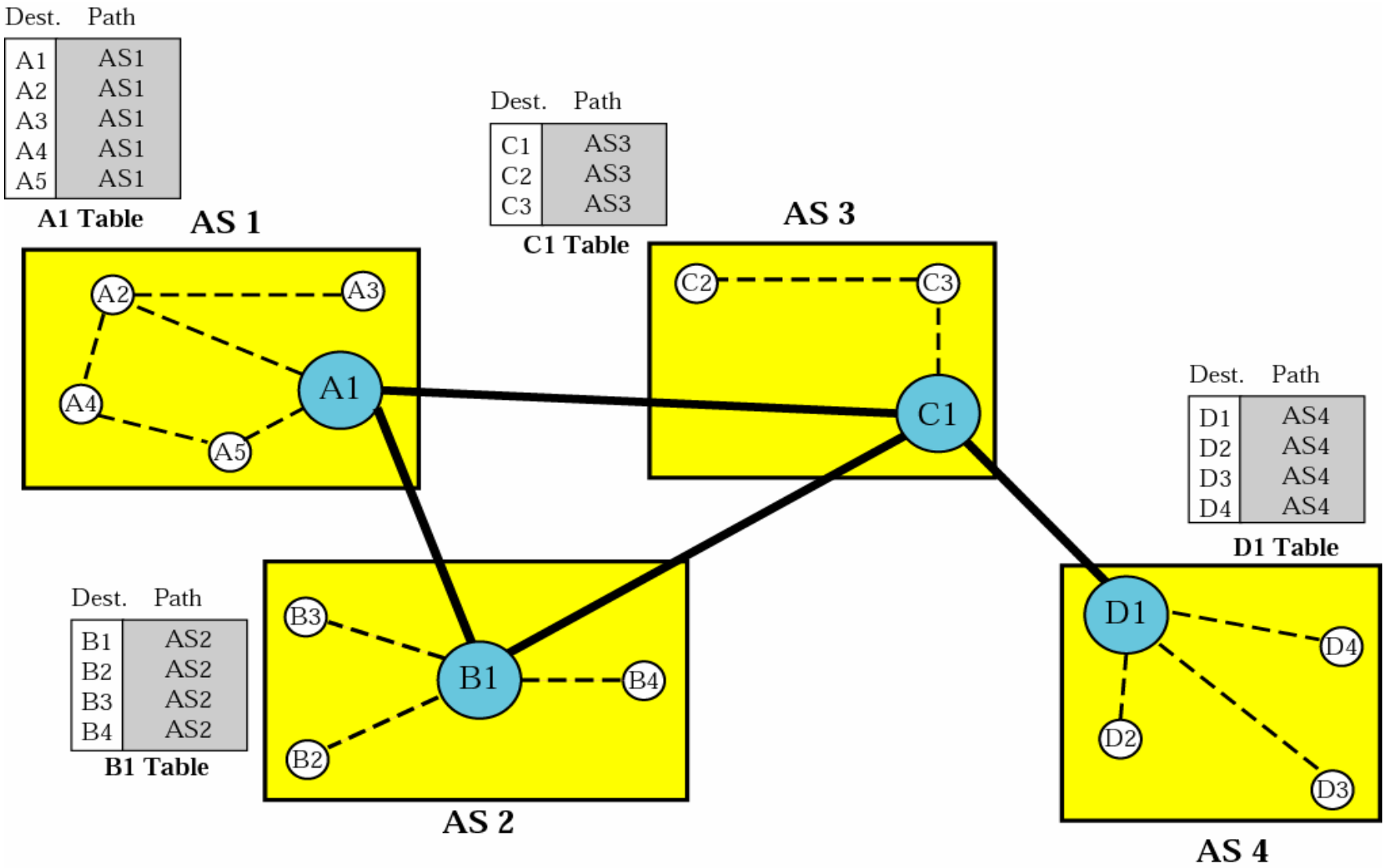
- Similar to distance vector routing
  - Except only speaker nodes in each AS can communicate with each other
  - Furthermore, a speaker node advertises the ***path***
    - Not the metric of the nodes

Initialization

- At the beginning, each speaker node can know only the reachability of nodes inside its AS

- Example
  - See the Following 14.48

Figure 14.48 *Initial routing tables in path vector routing*



Sharing

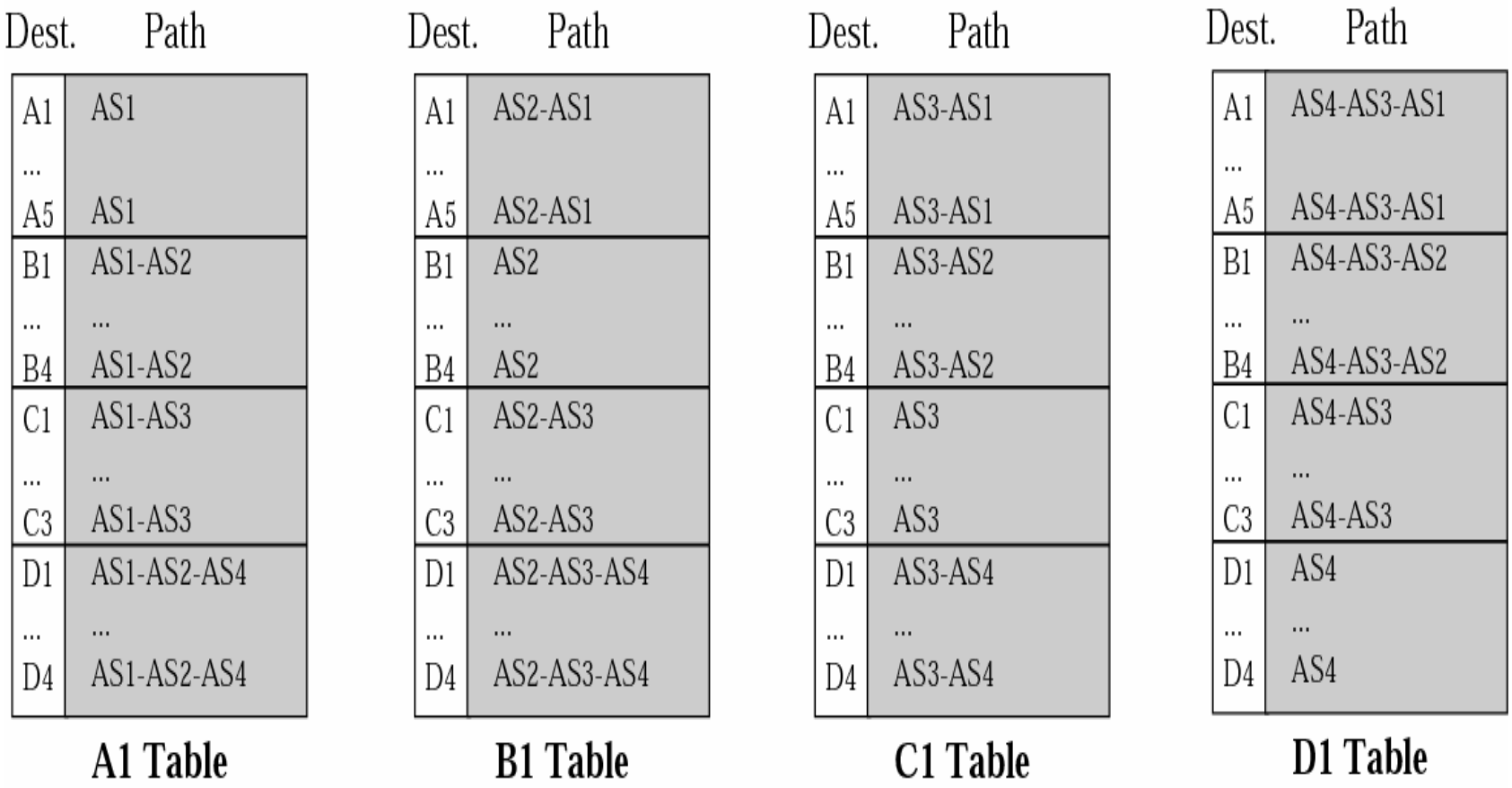
- A speaker in an AS share its table with ***immediate*** neighbors

- Example
  - In Fig. 14.48
  - A1 share with B1 and C1

Updating

- When a speaker node receives a two-column tables, update its table
  - Add the nodes that are not in its routing table
  - Add its ***own AS*** and ***AS that sent the table***
- Example
  - Fig. 14.49 shows a stabilized table
  - If A1 receives a packet for node A3
    - The path is in AS1 (the packet is at home)
  - If D1 receives a packet for node A2
    - The packet should go from AS4 to AS3, and then to AS1

Figure 14.49 *Stabilized tables for four autonomous systems*



Loop Prevention

- Path vector routing can avoid the instability of distance vector routing and the creation of loops

- When a router receives a message
  - Check if its AS is in the path list
  - If yes, looping is involved
    - Drop the message

Policy Routing

- *Policy routing* can be easily implemented through path vector routing
  - Path vector routing lists *all the ASs of each path*
  - Once a router receives a message, it can *check the path*.
  - If one of the AS listed in the path is against its policy,
    - It can ignore that path and that destination
      - Does not update its routing table with this path
      - Does not send this message to its neighbors
- Thus, path vector routing are not based on the *smallest hop count* or *the minimum metric*
  - Based on the *policy* imposed on the router by the administrator

Optimum Path

- The optimum path is the path that fits the organization
  - Criteria may be: hop count, security and safety, reliability

- Path vector routing can achieve optimum path by looking for a path best for the organization
  - Since all the AS are listed in the path

14.7 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 topics discussed in this section include:

Types of Autonomous Systems
Path Attributes
BGP Sessions
External and Internal BGP
Types of Packets
Packet Format
Encapsulation

BGP

- BGP: Border Gateway Protocol
- An inter-autonomous system routing protocol
- Based on the *path vector routing* method

Types of Autonomous Systems

- The Internet is divided into hierarchical domains called autonomous systems (ASs)
  - A large corporation manages its own network is an AS
- AS can be divided into three categories
  - ***Stub***
  - ***Multihomed***
  - ***Transmit***

Stub AS

- Has only one connection to another AS
- Interdomain data traffic is a stub AS can be either ***created*** or ***terminated***
  - Cannot ***pass through*** a stub AS
- A stub AS is either a source or a sink

- Example
  - A small corporation or a small ISP

Multihomed AS

- Has more than one connection to another AS
- But it is still only a source or sink for data traffic
  - There is not transient traffic
  - Does not allow data coming from one AS and going to another AS to pass through
- Example
  - A large corporation connected to more than one regional AS but does not allow transient traffic

Transit AS

- A multihomed AS and allow transient traffic
- Example
  - National and international ISPs

CIDR

- BGP supports Classless Interdomain Routing addresses
- The address and the prefix length are used in updating message

Path Attributes

- In previous example, the path was presented as a list of AS
- Actually, the path was presented as *a list of attributes*
  - The list of attributes help the receiving router make a better decision when applying its policy

Path Attributes (Cont.)

- Attributes are divided into two categories: ***well-known*** and ***optional***
- **Well-known:** one that every BGP router should recognize
  - **Mandatory:** must appear in the description of a route
    - e.g., ORIGIN: the source of the routing information (RIP or OSPF)
    - e.g., AS_PATH: the list of AS through which the destination can be reached
    - e.g., NEXT_HOP: the next router to which data packet should be sent
  - **Discretionary**
    - Must be recognized by each router
    - But is not required to be included in every update message

Path Attributes (Cont.)

- **Optional:** one that need not be recognized by every router
  - **Transitive**
    - Must be passed to the next router by the router that has not implemented this attribute
  - **Nontransitive**
    - One that should be discarded if the receiving router has not implemented it

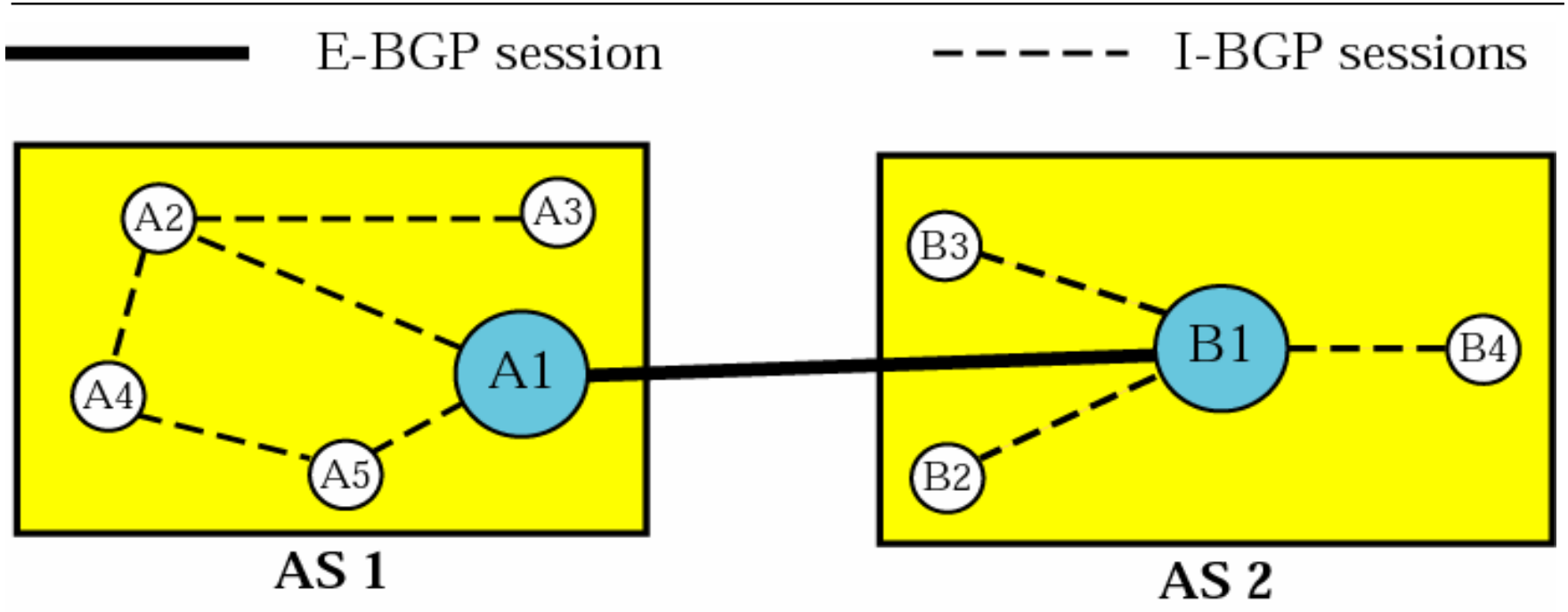
BGP Sessions

- A session is a connection that is established between two BGP routers only for exchanging routing information
  - Use TCP for its reliable environment
- Note, a BGP session can last for a long time until something unusual happens
  - BGP session thus also called ***semi-permanent connections***

External and Internal BGP

- BGP have two types of sessions
  - External BGP (E-BGP)
  - Internal BGP (I-BGP)
- E-BGP
  - Exchange information between two speaker nodes belonging to two different ASs
- I-BGP
  - Exchange information between two routers inside an AS

Figure 14.50 *Internal and external BGP sessions*



AS1 and AS2: E-BGP

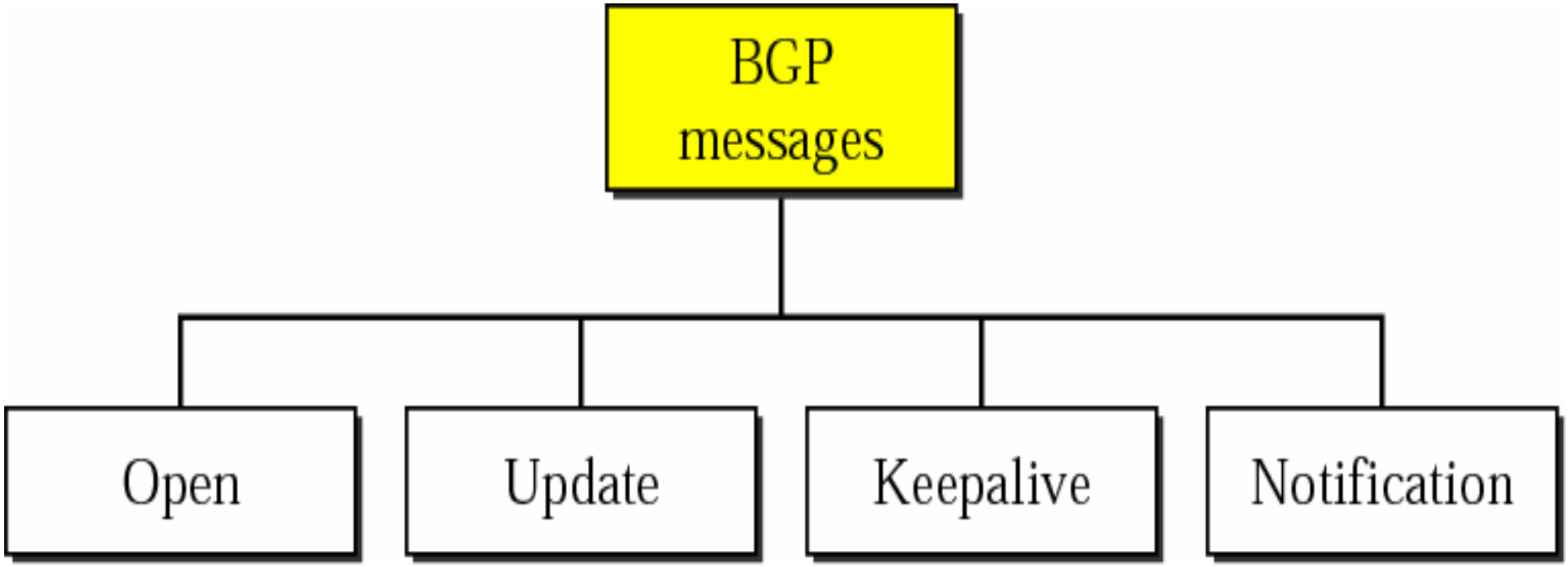
I-BGP: collect information from other routers in their AS

Types of Packets

- BGP uses four different types of messages
  - Open
  - Update
  - Keepalive
  - Notification

Figure 13-51

Types of BGP Messages

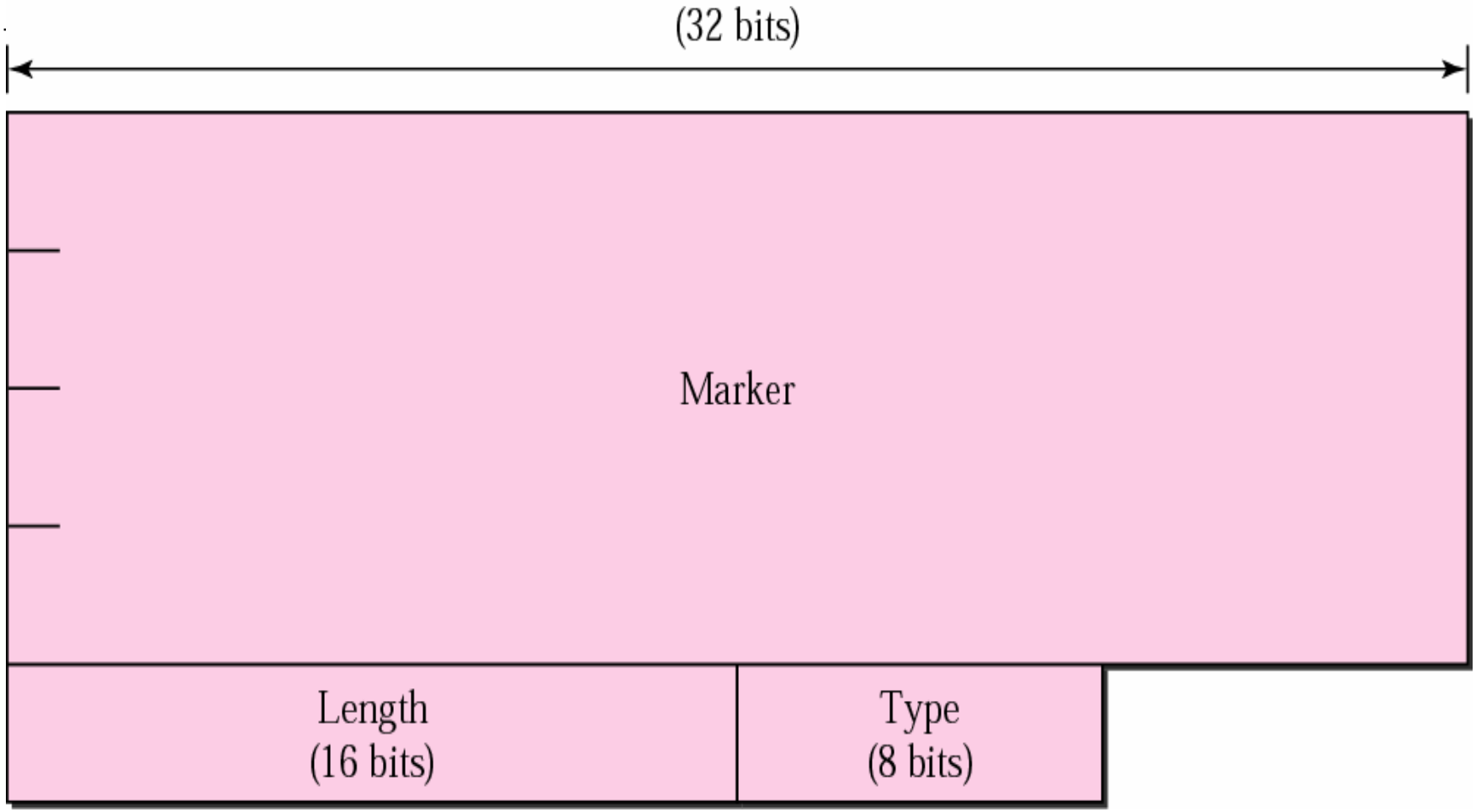


Packet Format

- All BGP packets share the same common header
- Header format
  - **Marker:** 16-bit
    - Reserved for authentication
  - **Length:** 2-bytes
    - Define the length of the total message, including the header
  - **Type:** 1-byte
    - Define the type of the packet

Figure 13-52

BGP Packet Header

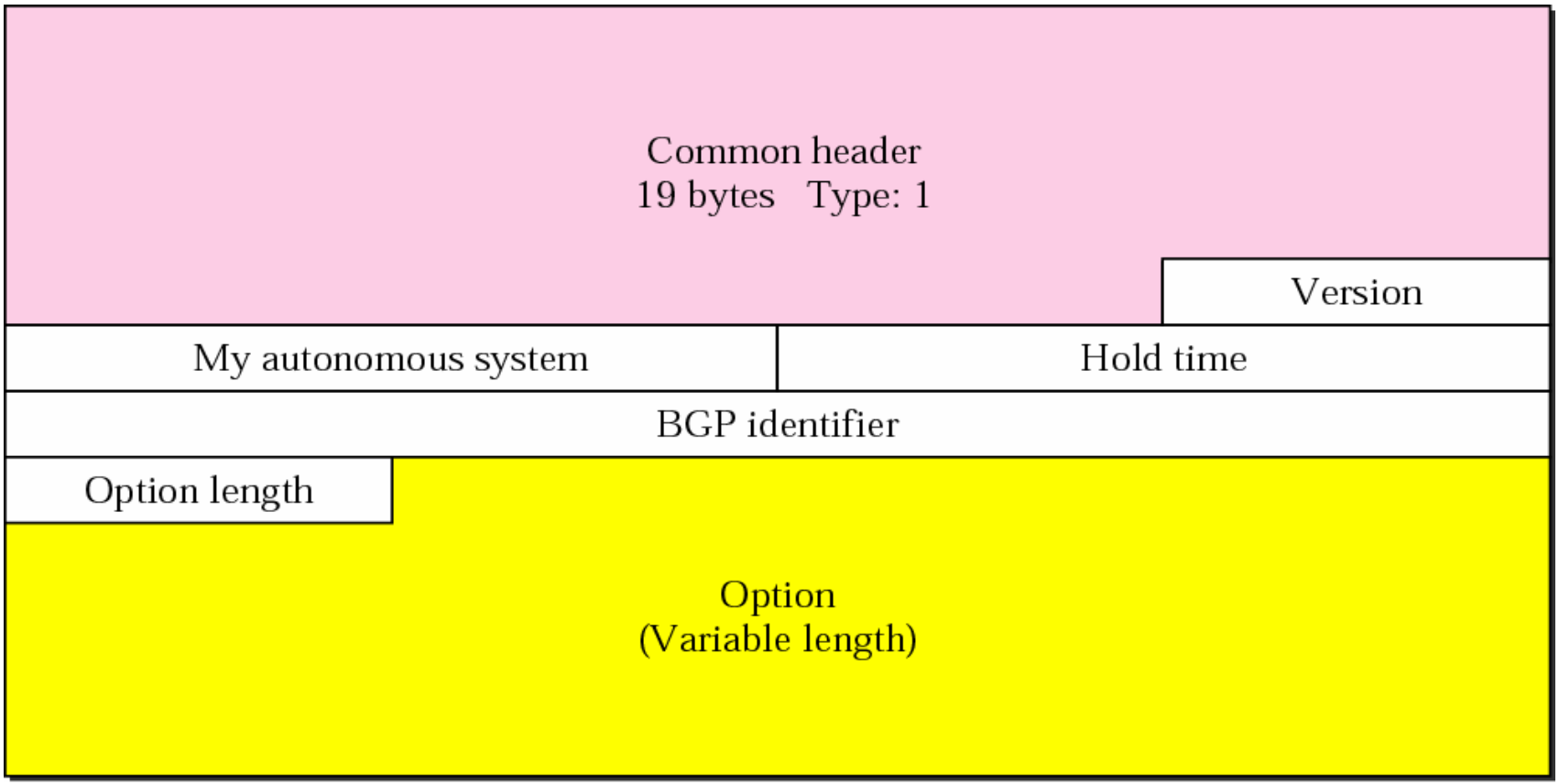


Open Message

- Used to create a neighborhood relationship
- A router running BGP opens a *TCP* connection with a neighbor and sends an *open message*
- If the neighbor accepts
  - It responses with a *keepalive message*
  - The relationship then has been established between the two router

Figure 12-52

Open Message



Open Message Packet Format

- Version: 1-byte
  - Define the version of BGP. The current version is 4
- My autonomous system: 2-byte
  - Define the autonomous system number
- Hold time: 2-byte
  - Define the maximum number of seconds that can elapsed before one of the parties receives a keepalive or update message from the other
  - If a router does not receive one of the messages during the hold period, it considers the other party dead

Open Message Packet Format (Cont.)

- BGP number: 4-byte
  - Define the router that sends the open message
- Option parameter length: 1-byte
  - Define the length of the total option parameters
    - Since open message may also contain some option parameters
- Option parameters
  - The only option parameters defined so far is *authentication*

Update Message

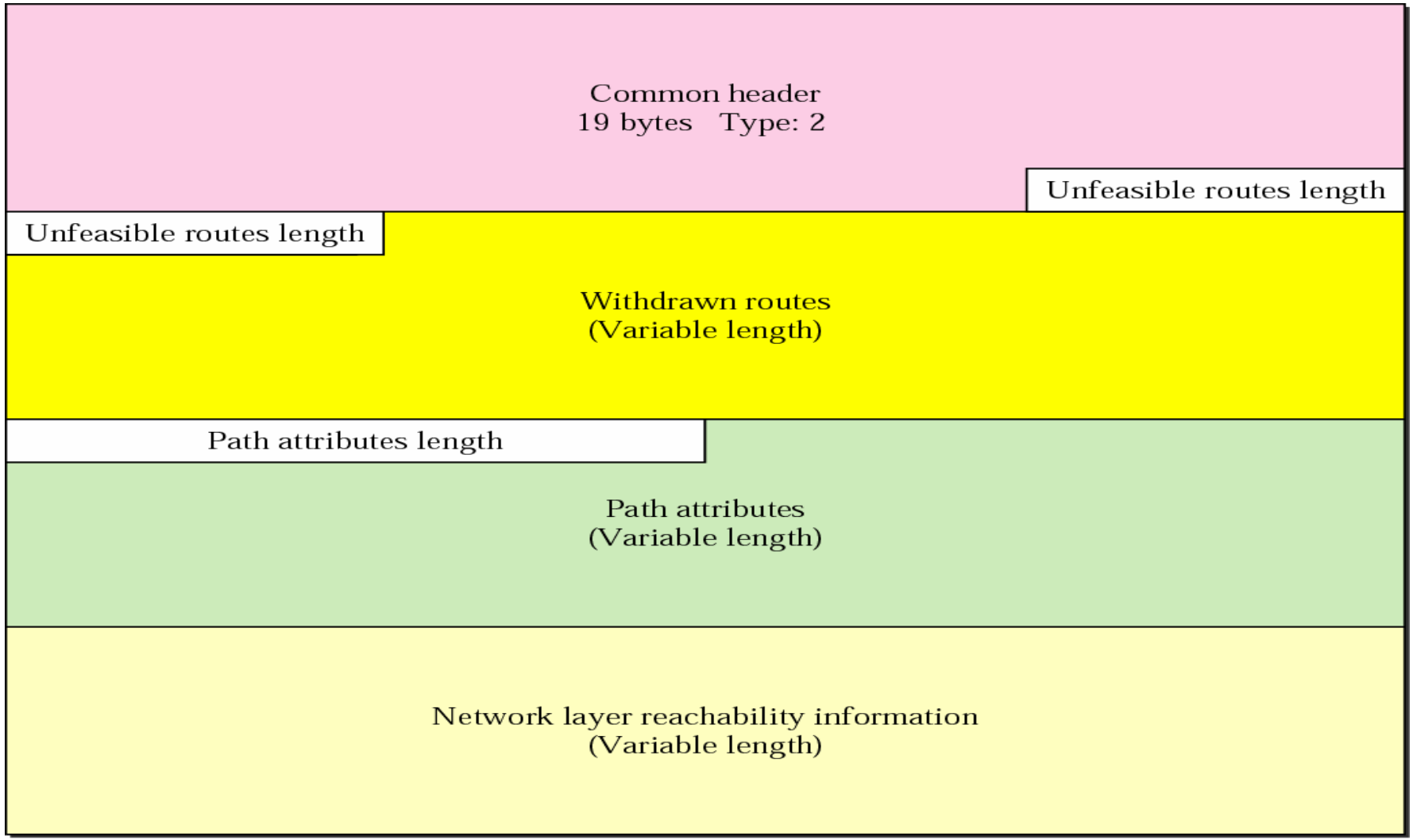
- Used by a router to
  - Withdraw destination that have advertised previously
  - Announce a route to a new destination
- Note, a router can withdraw several destinations in a single update message
  - However, it can only advertise one new destination in a single update message

Update Message Format

- Unfeasible routes length: 2-byte
  - Define the length of the next field
- Withdraw routes
  - List all routes that should be deleted from the previously advertised list
- Path attributes length: 2-byte
  - Define the length of the next field
- Path attributes:
  - Defines *the attributes of the path (route) to the network* whose reachability is being announced in this msssage

Figure 13.54

Update Message

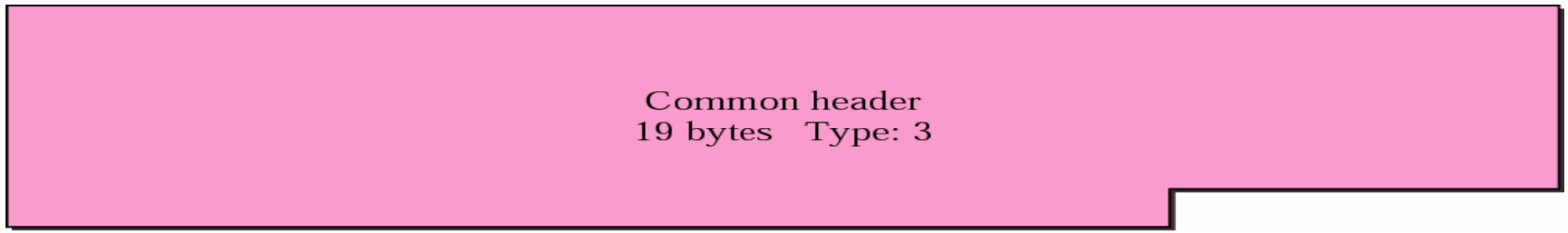


Update Message Format

- Network Layer reachability information (NLRI)
  - Define the network that is actually advertised by this message
  - Has two fields
    - Length: define the number of bits in the prefix
    - IP address prefix: the common part of the network address
    - Example: if a network is 153.18.7.0/24
      - Length = 24
      - IP address prefix = 153.18.7
- Thus, BGP4 supports *classless addressing* and *CIDR*

Keepalive Message

- The BGP routers exchange keepalive message *regularly*
  - Tell each other that they are alive
- Format
  - Consist of only the common header



Keepalive Message

Notification Message

- Sent by a router
  - Whenever an *error condition* is detected
  - A router wants to *close* the connection
- Format
  - Error code: 1-byte, define the category of the error
    - **Message header error**
    - **Open message error**
    - **Update message error**
    - **Hold timer expired**
    - **Finite state machine error**
    - **Cease**
  - Error subcode: 1-byte
    - Furthermore define the type of error in each category
  - Error data
    - Used to give more diagnostic information about the error

Figure 13-56

Notification Message

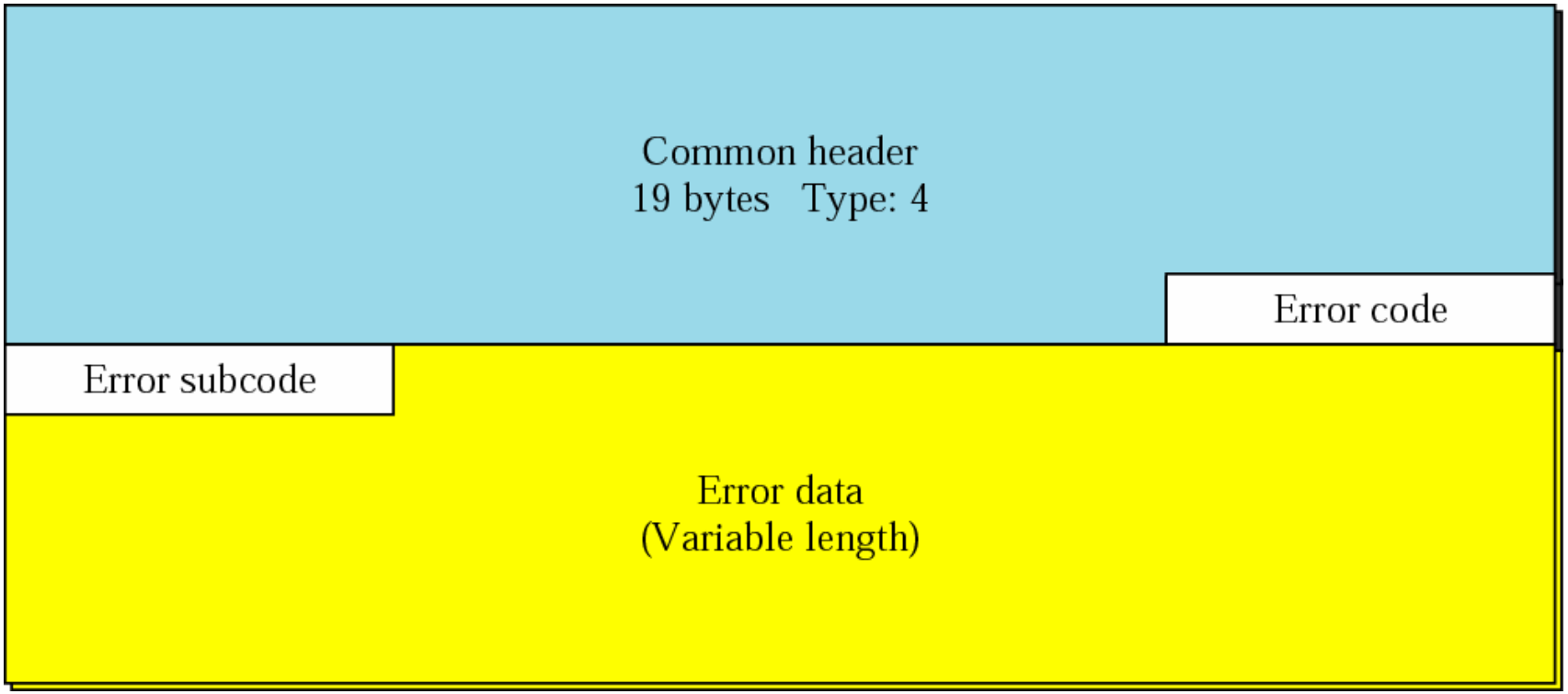


Table 14.3 Error codes

Error Code	Error Code Description	Error Subcode Description
1	Message header error	Three different subcodes are defined for this type of error: synchronization problem (1), bad message length (2), and bad message type (3).
2	Open message error	Six different subcodes are defined for this type of error: unsupported version number (1), bad peer AS (2), bad BGP identifier (3), unsupported optional parameter (4), authentication failure (5), and unacceptable hold time (6).
3	Update message error	Eleven different subcodes are defined for this type of error: malformed attribute list (1), unrecognized well-known attribute (2), missing well-known attribute (3), attribute flag error (4), attribute length error (5), invalid origin attribute (6), AS routing loop (7), invalid next hop attribute (8), optional attribute error (9), invalid network field (10), malformed AS_PATH (11).
4	Hold timer expired	No subcode defined.
5	Finite state machine error	This defines the procedural error. No subcode defined.
6	Cease	No subcode defined.

Encapsulation

- BGP message are encapsulated in TCP segments using the well-known port 179
  - No need for error control and flow control
- Thus, when a TCP connection is opened
  - The exchange of update, keepalive, and notification message is continued
  - Until a *notification message* of type *cease* is sent