# Routing Protocols (RIP, OSPF, and BGP)

# Outline

- INTERIOR AND EXTERIOR ROUTING
- RIP
- OSPF
- 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 equals
    - 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 prococols

# INTERIOR AND EXTERIOR ROUTING

# Interior and Exterior Routing

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

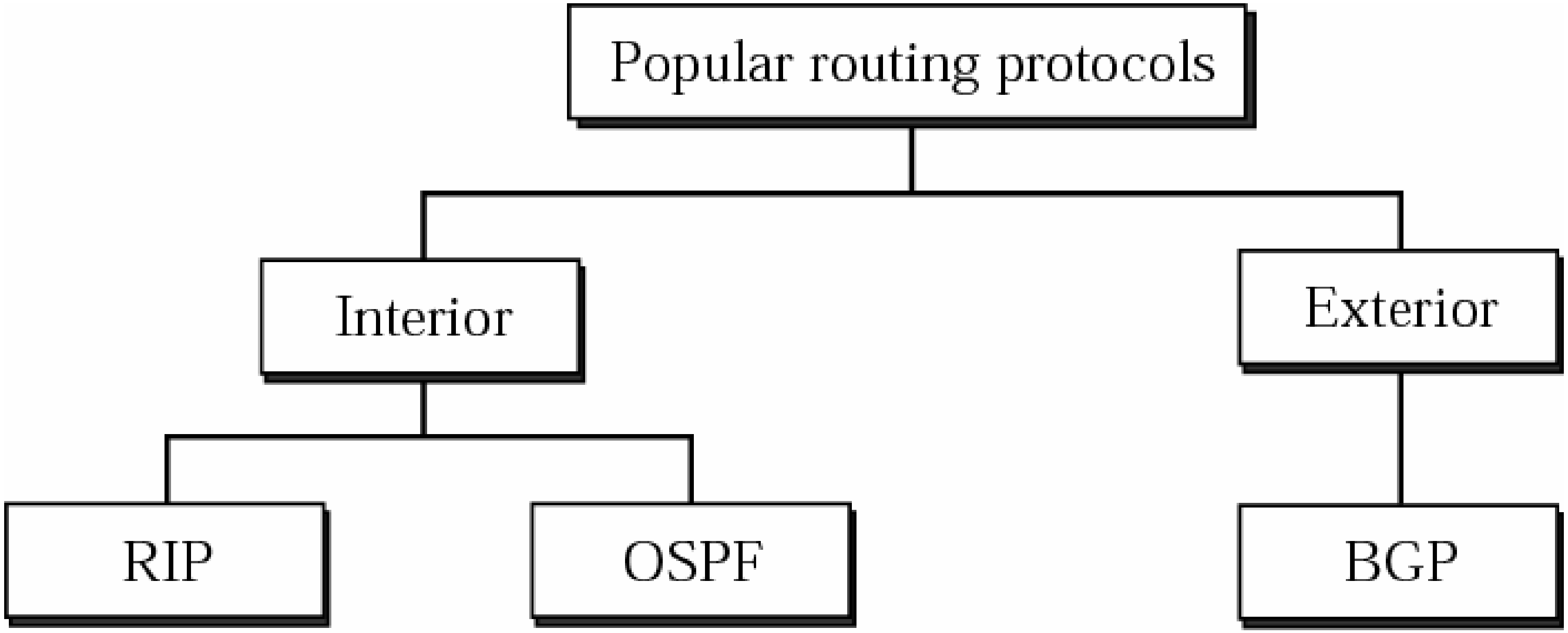
- Thus, an internet is divided into ***autonomous systems (AS)***
  - AS is a group of networks and routers under the authority of a single administration

# Interior and Exterior Routing

- ***Interior routing***
  - Routing inside an autonomous system
  - Each AS can chose its own interior routing protocol
  - Examples: RIP and OSPF
- ***Exterior routing***
  - Routing between autonomous systems
  - Only one exterior routing protocol is usually used for exterior routing
  - Examples: BGP

Figure 13.1

# Popular Routing Protocols



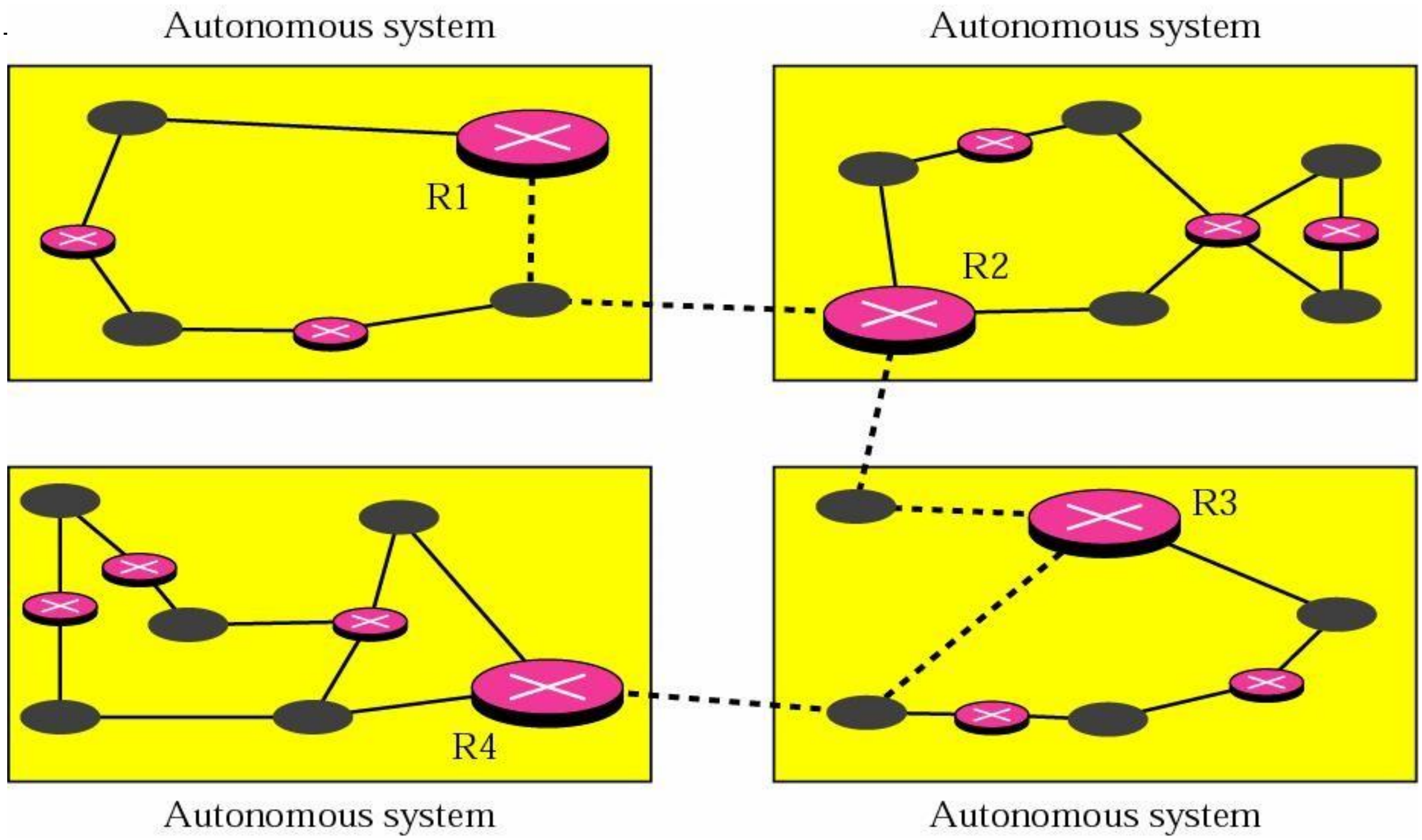
# Example

- R1, R2, R3 and R4 use an interior and an exterior routing protocol

- Solid thin lines
  - Interior routing protocol

- Broken thick lines
  - Exterior routing protocol

# Autonomous Systems



# RIP: Routing Information Protocol

# RIP

- RIP: Routing Information Protocol
  - Based on *distance vector routing*
  - Use the Bellman-Ford algorithm for calculating the routing tables

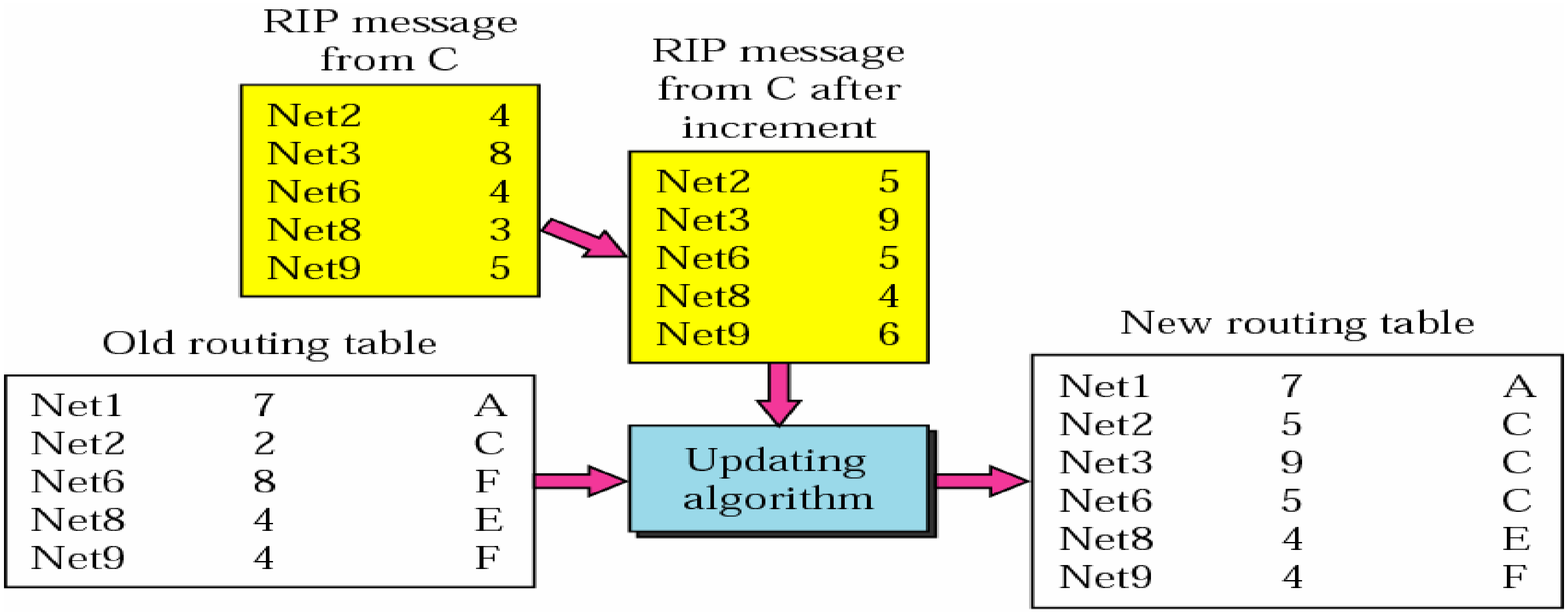
# Distance Vector Routing

- Each router periodically shares its knowledge about the entire internet with its neighbors
  - ***Sharing knowledge about the entire AS***
    - At the start, a router's knowledge may be sparse
    - But, how much it knows is unimportant, it sends whatever it has
  - ***Sharing only with neighbors***
    - Sends its knowledge only to neighbors
  - ***Sharing at regular intervals***

# RIP Updating Information

- Routing table is updated on receipt of a RIP response message
- *Receipt: a response RIP message*
- *Add one hop to the hop count for each advertised destination*
- *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 advertised one*
    - *Else*
      - *If (advertised hop count smaller than one in the table)*
        - *Replace entry in the routing table*
- *Return*

# Example of Updating a Routing Table




Net1: No news, do not change
Net2: Same next hop, replace
Net3: A new router, add
Net6: Different next hop, new hop count smaller, replace
Net8: Different next hop, new hop count the same, do not change
Net9: Different next hop, new hop count larger, do not change

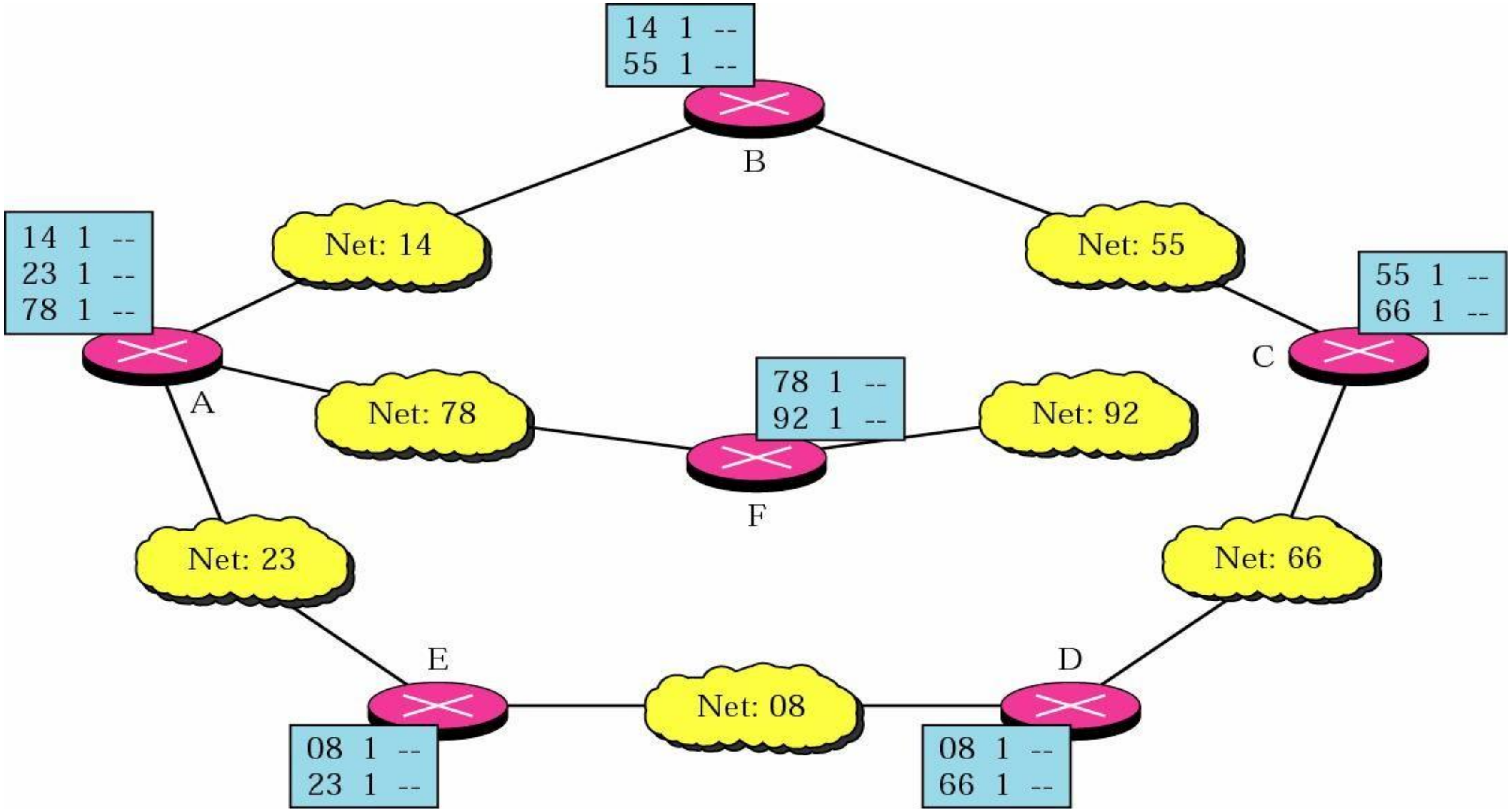
# Initializing the Routing Table

- When a router is added to a network
  - It initializes a routing table for itself using its *configuration file*
  - The table contains only the *directly attached networks* and the *hop count* (= 1)

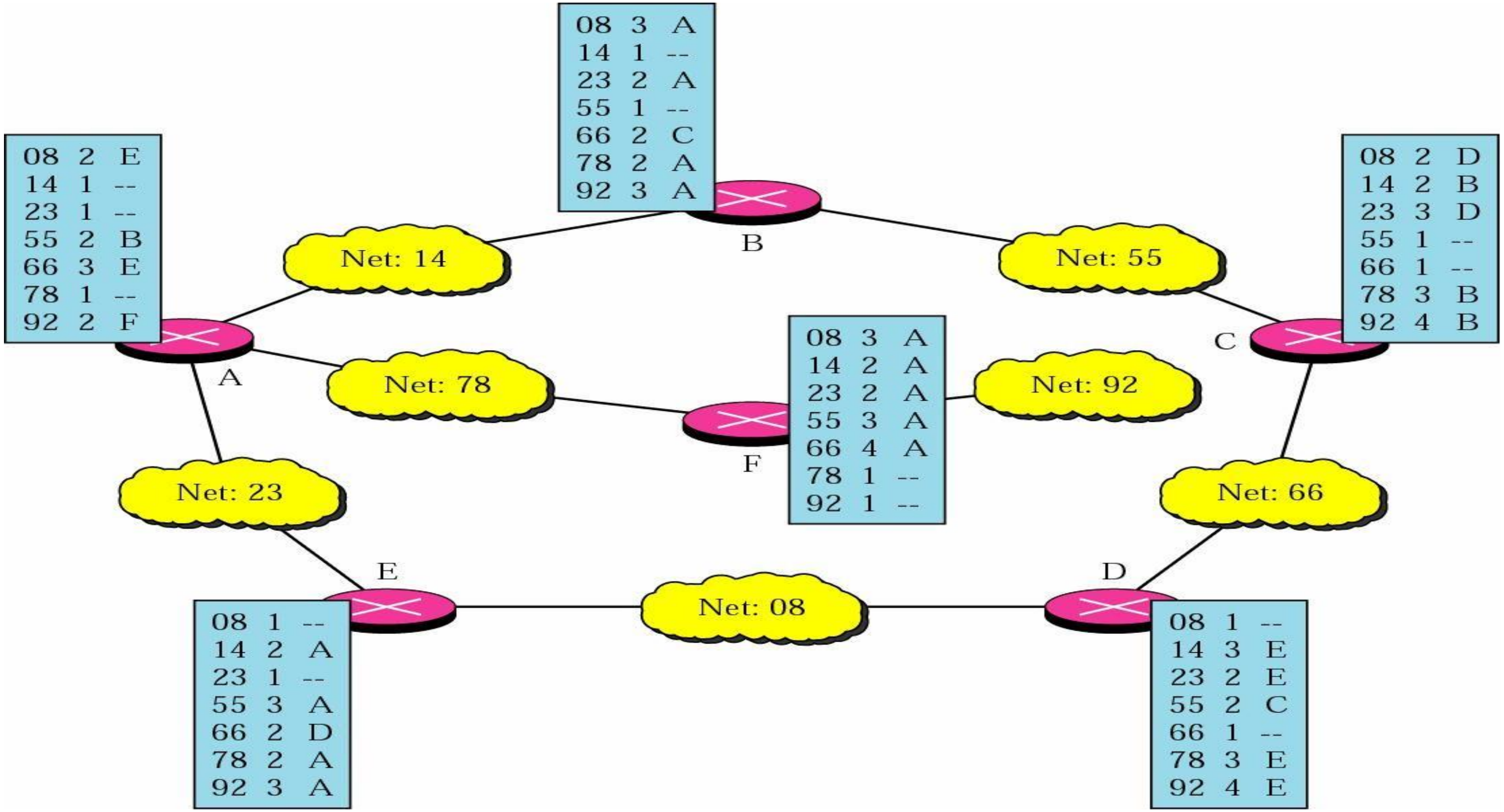
# Updating the Routing Table

- Each routing table is updated upon receipt of RIP message
  - Using the RIP updating message algorithm shown above

# Initial Routing Tables in a Small Autonomous System



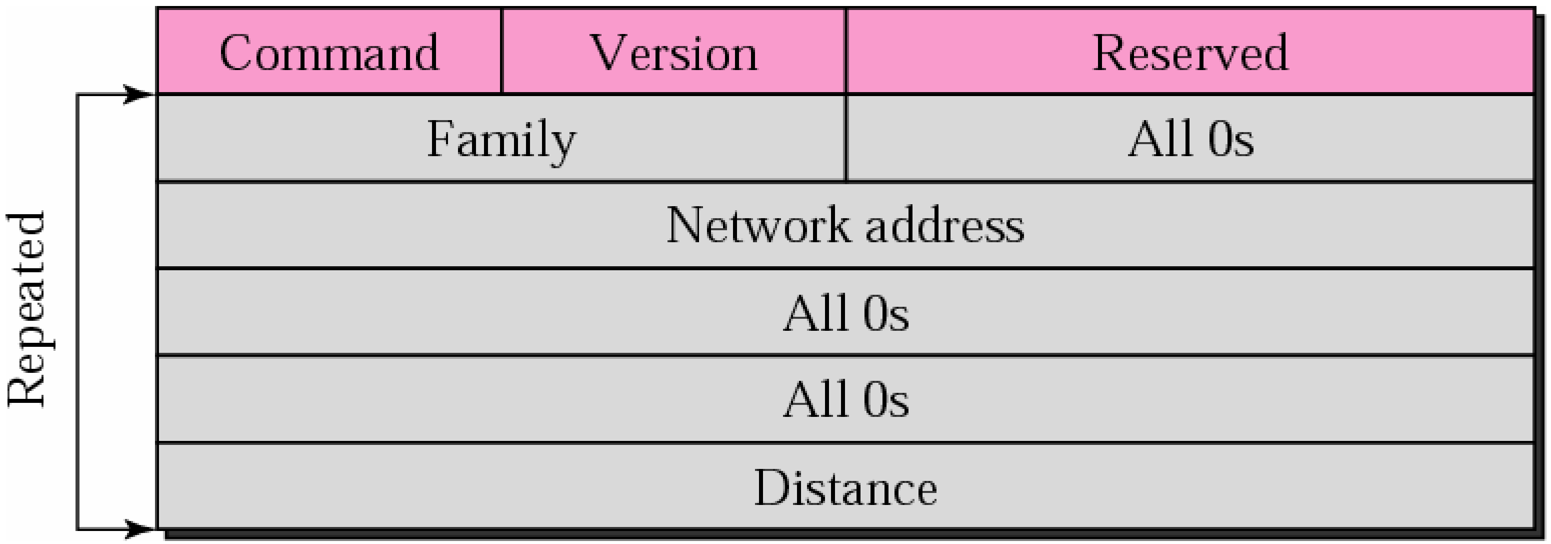
# Final Routing Tables for the Previous Figure



# 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
- 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

# RIP Message Format

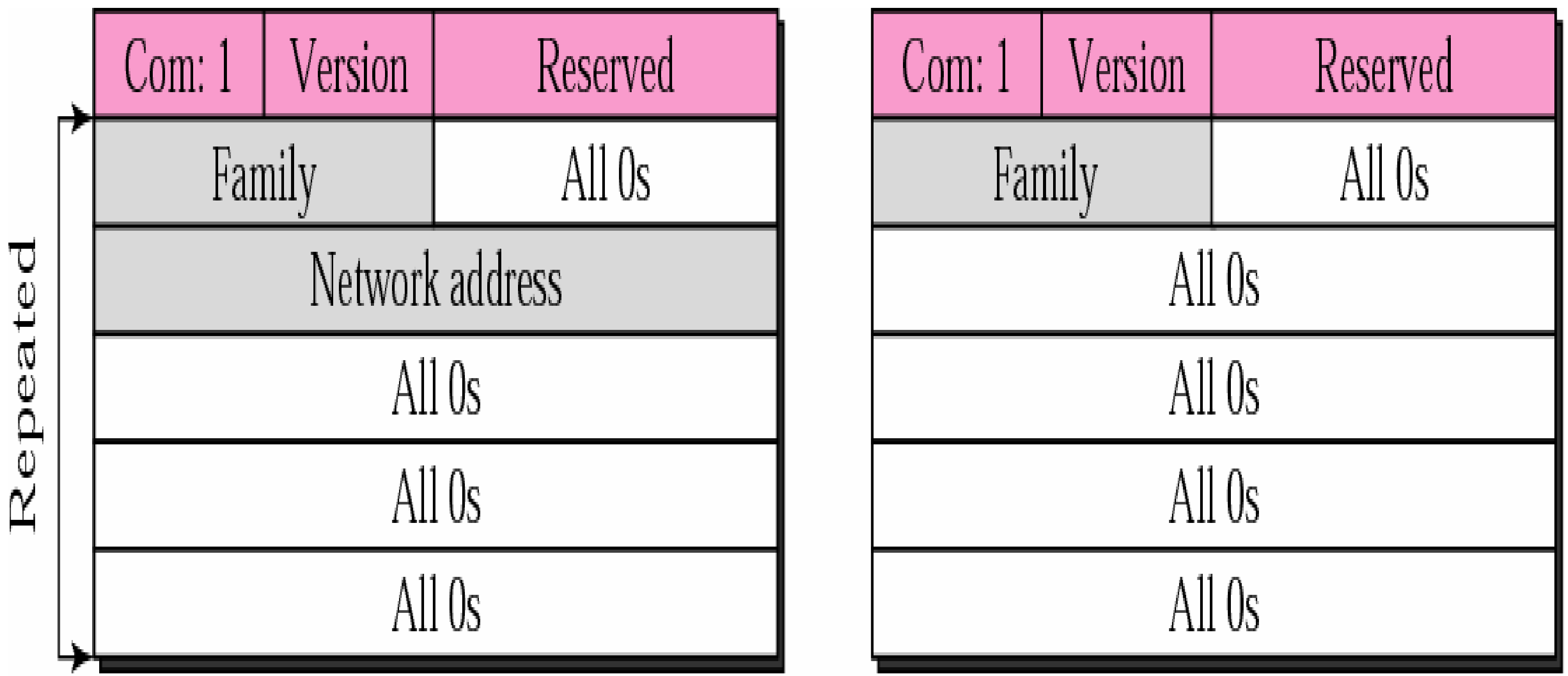


# Requests and Response

- 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*

# Request Messages




a. Request for some

b. Request for all

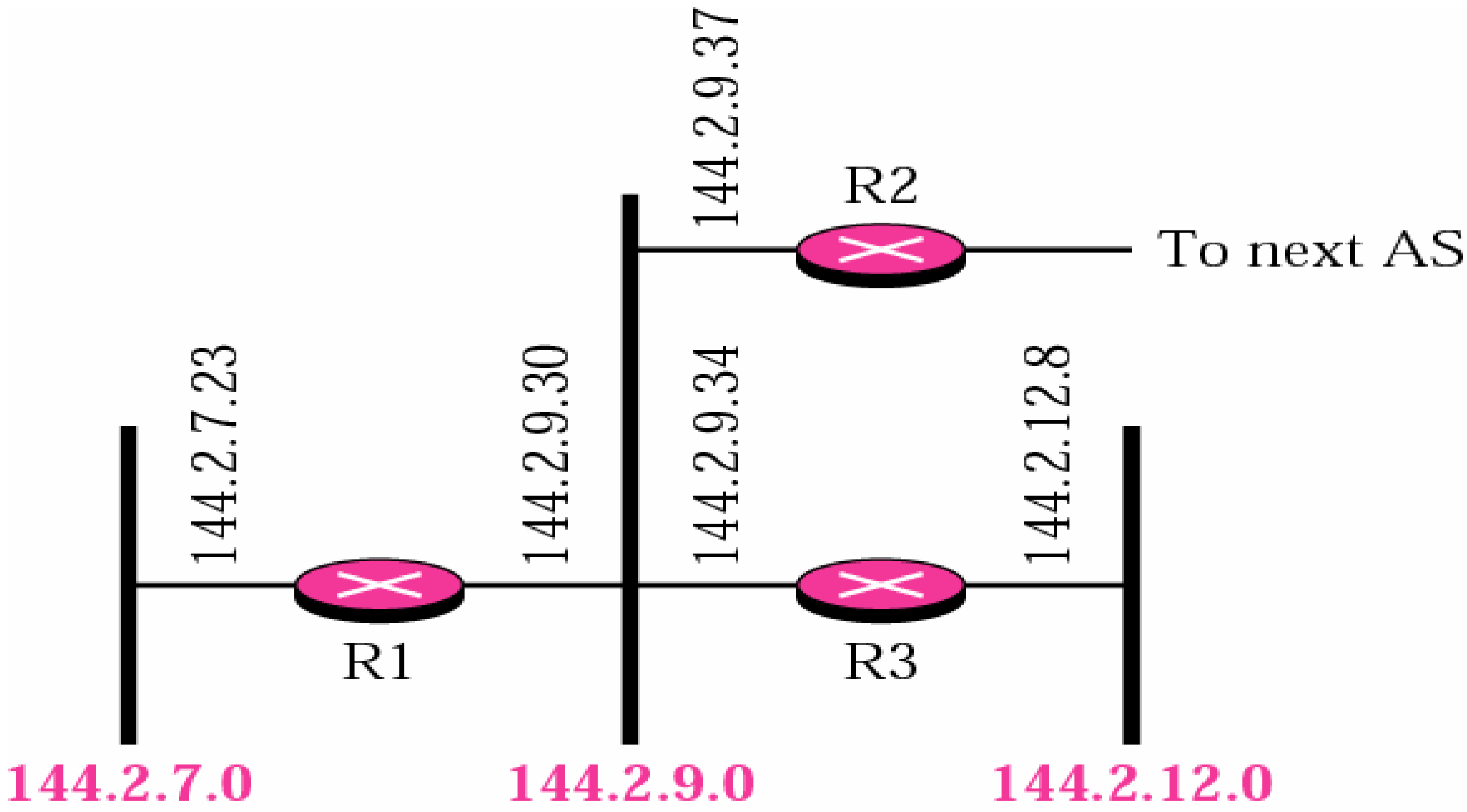
# Requests and Response (Cont.)

- 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*

- What is the periodic response sent by router R1 in the next slide?
- Assume R1 knows about the whole autonomous system.

# Example 1

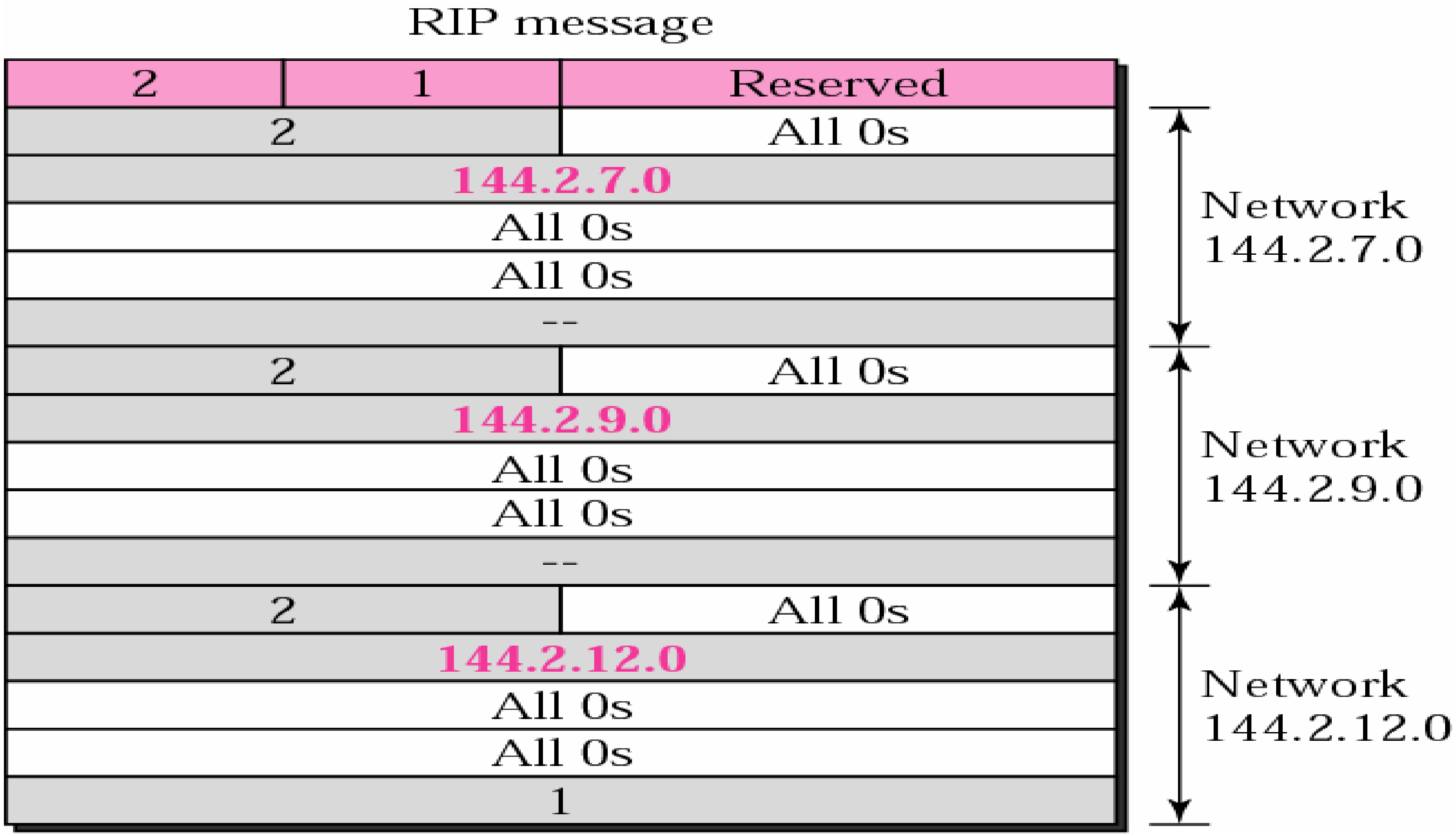


144.2.9.37
R2
To next AS
144.2.7.23
144.2.9.30
144.2.9.34
144.2.12.8
R1
R3
144.2.7.0
144.2.9.0
144.2.12.0

## *Solution*

- R1 can advertise three networks 144.2.7.0, 144.2.9.0, and 144.2.12.0.
- The periodic response (update packet) is shown in the next slide.

# Solution to Example 1

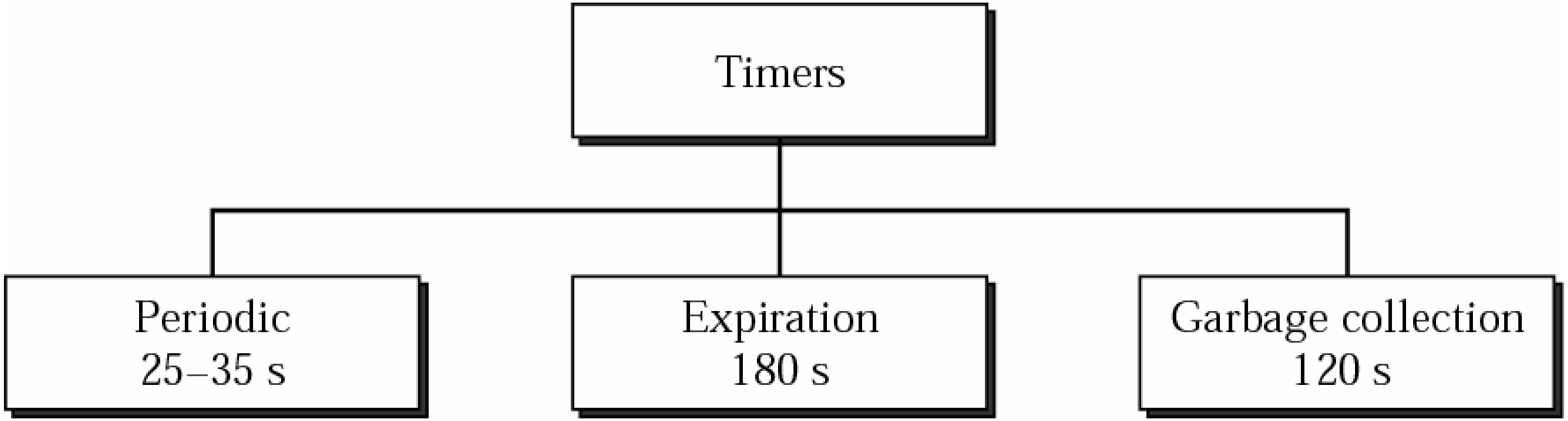


# Timers in RIP

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

Figure 13-10

# 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

# Problems in RIP

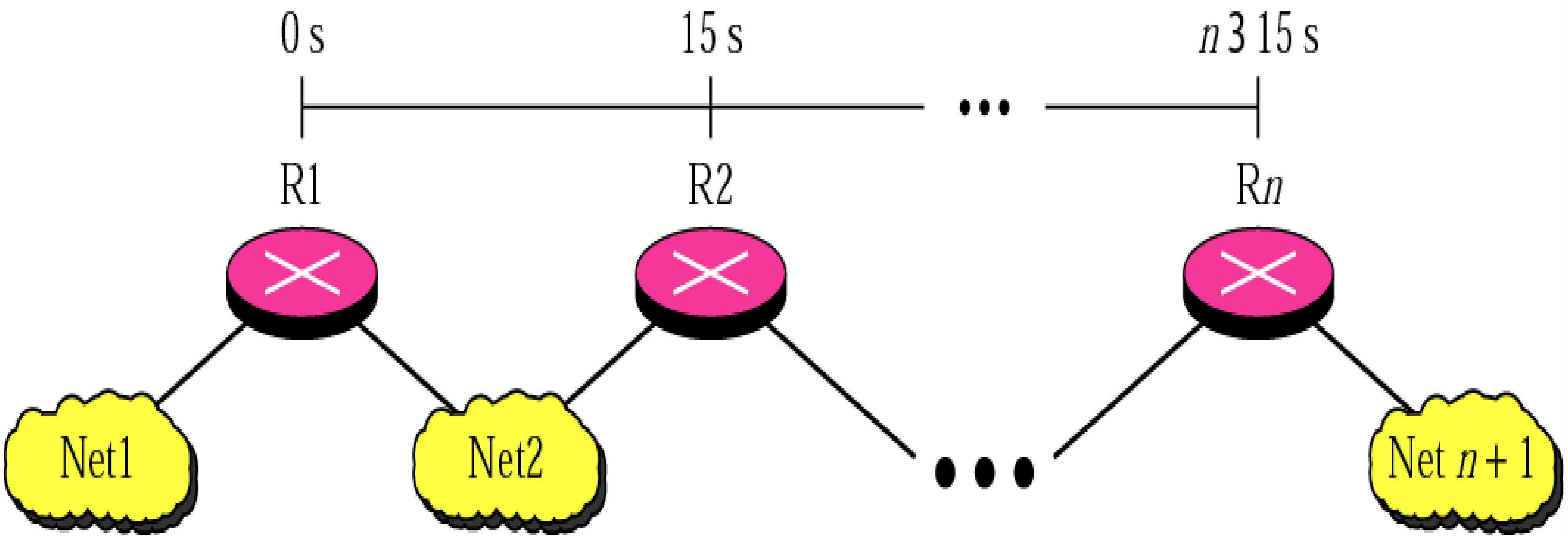
- Slow Convergence
- Instability

# Slow Convergence

- A change somewhere in the internet propagates very slowly through the rest of the internet
- For example
  - A change is Net1, R1 updates itself immediately
  - R1->R2: an average of 15 s
  - R2->R3: an average of 15 s
  - …
  - Thus, R1-Rn: an average of *15* x *n* s

Figure 13.11

# Slow Convergence

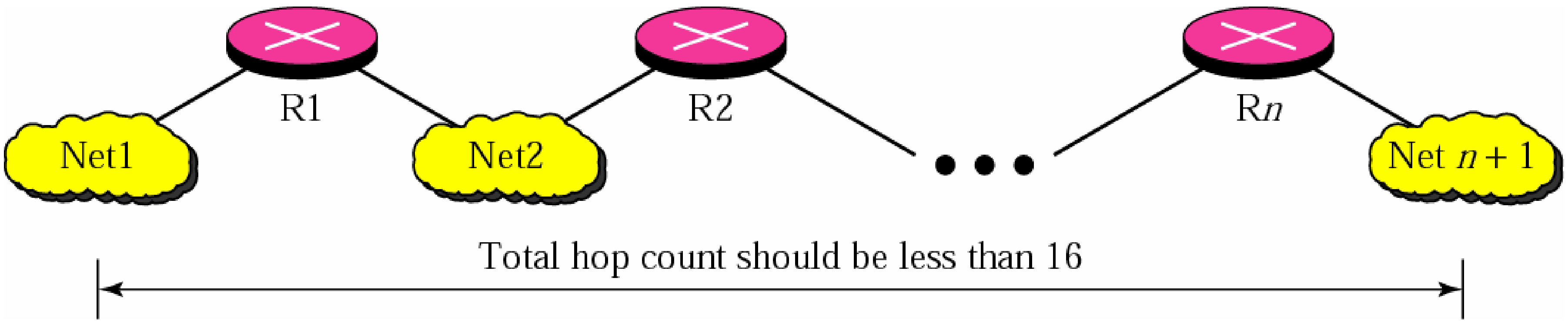


# Solution

- Limit the hop count to 15
  - Prevent data packet from wandering around forever
- Thus, an *autonomous system* using RIP is limited to a diameter of *15*
  - The number 16 is considered infinity and designates an *unreachable network*

Figure 13-13

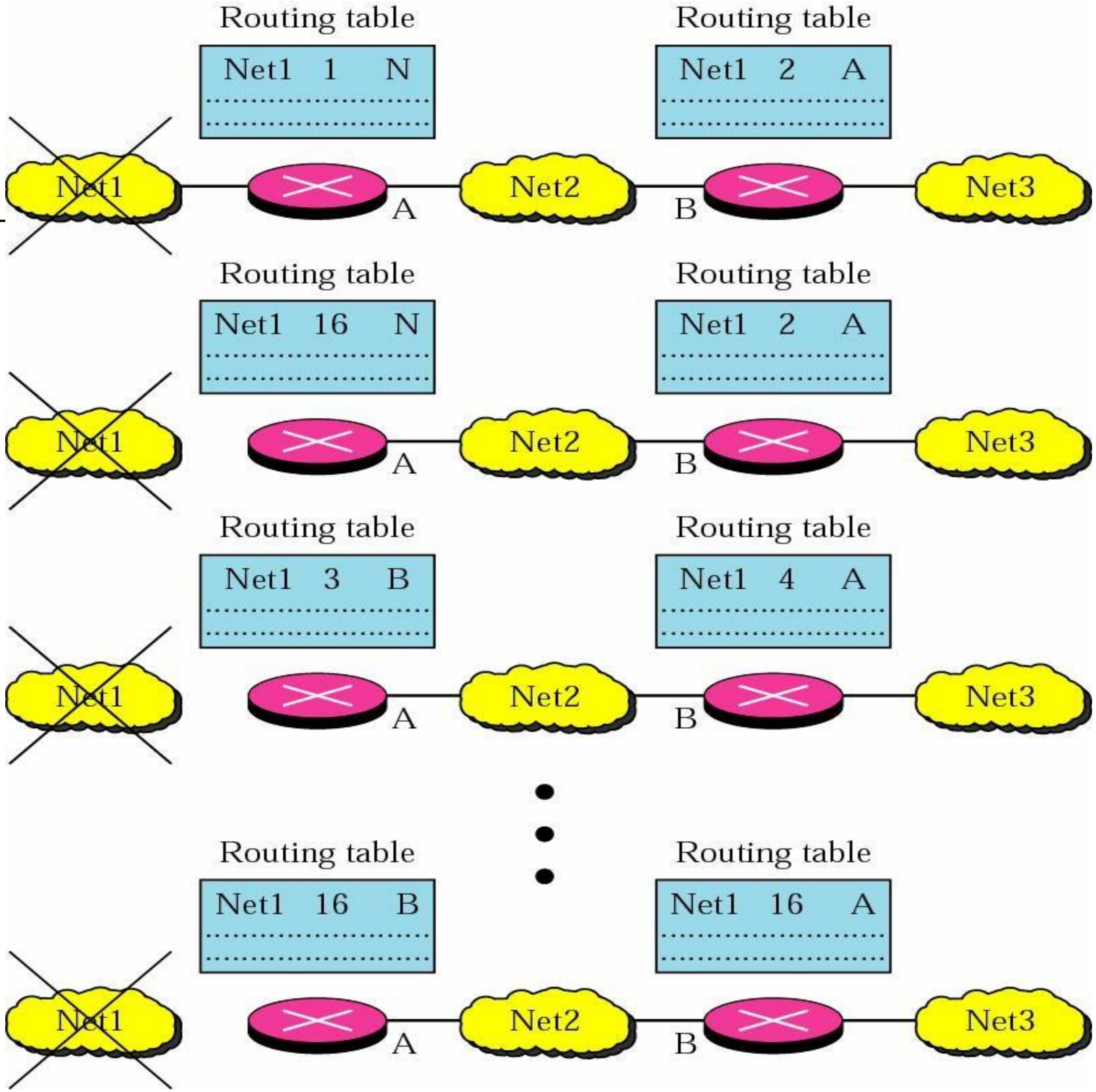
# Hop count



# Instability

- An internet running RIP can become *unstable*
  - A packet could go from one router to another in a loop

# Instability



# Solutions

- Triggered Update
- Split Horizons
- Poison Reverses

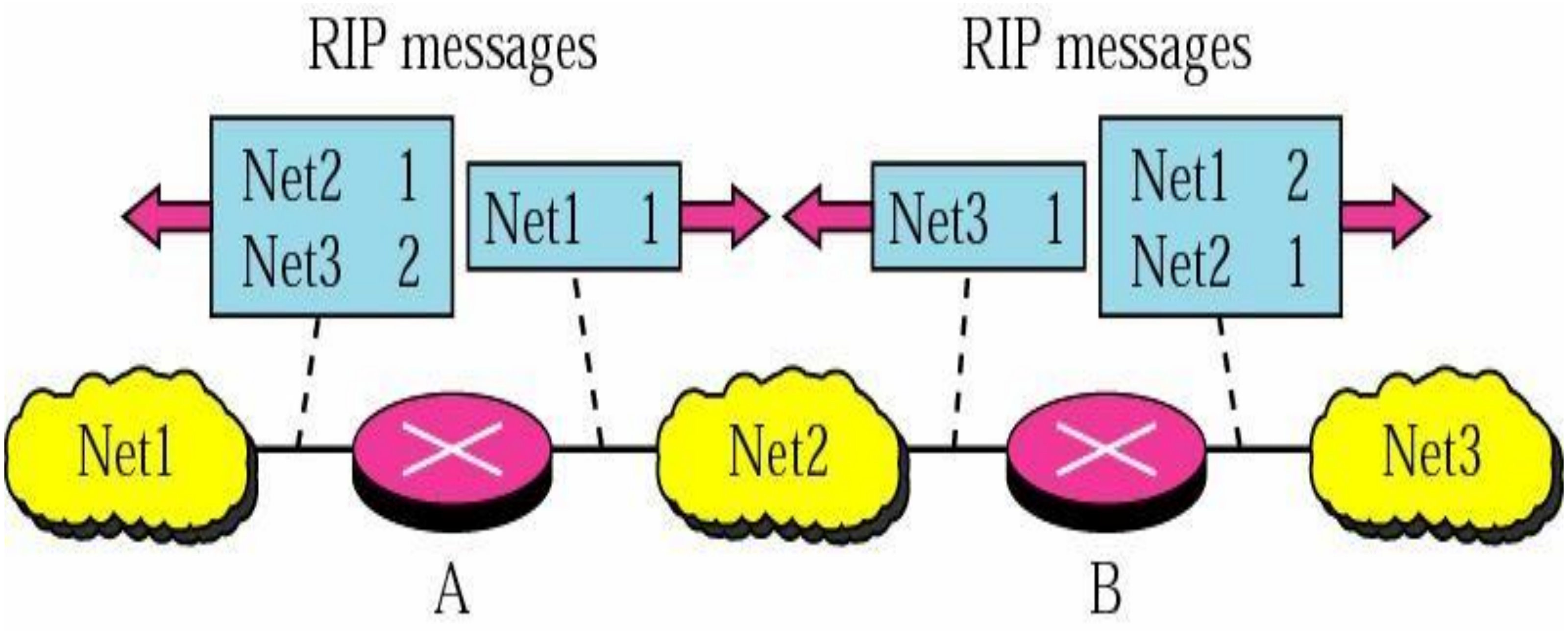
# Triggered Update

- If there is no change, updates are sent at the 30-s intervals
- If there is a change, the routers sends out its new table immediately

# Split Horizons

- 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
- Example
  - 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

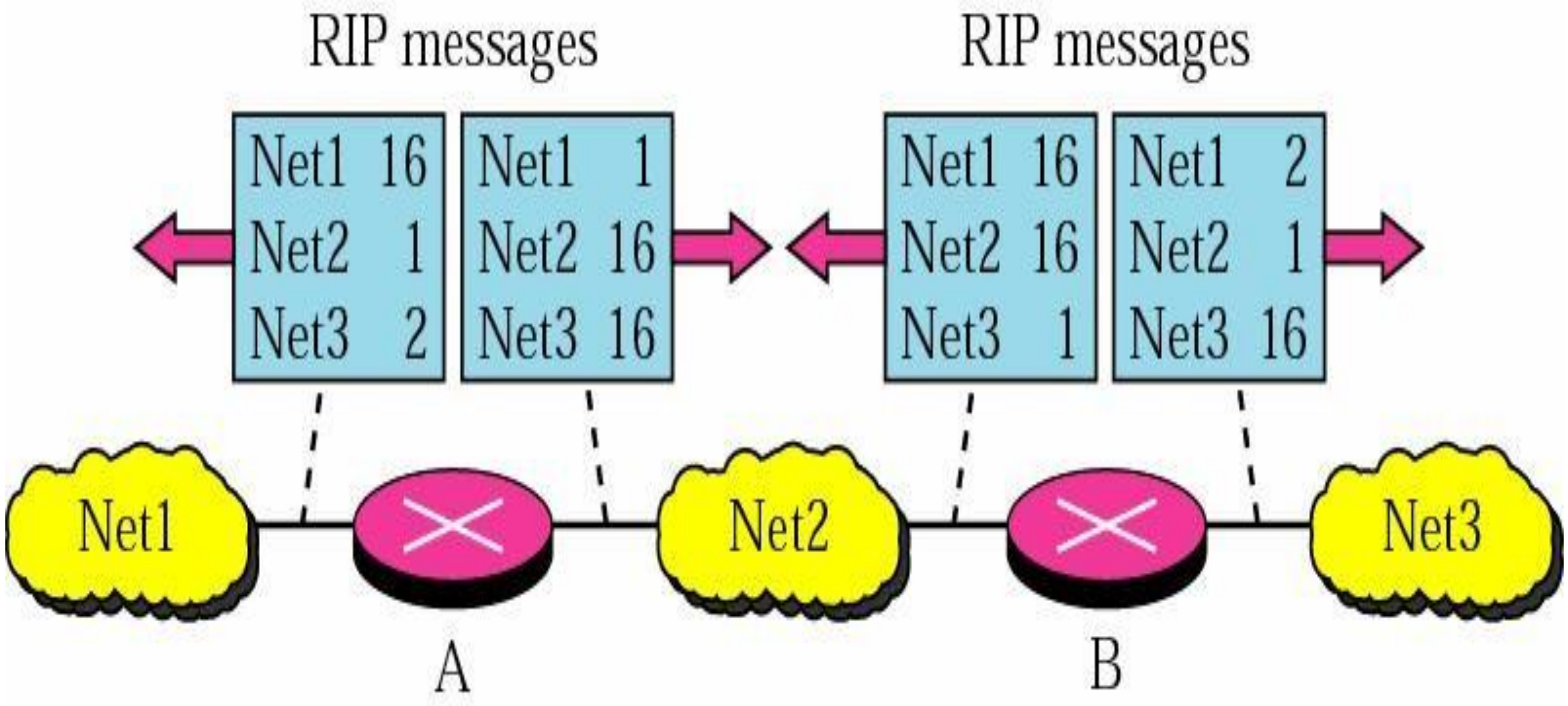


# Poison Reverse

- A variation of split horizons
- Information received is used to update routing table and then passed out to *all* interface
- However, a table entry that has come through one interface is set to a metric of *16 as it goes out through the same interface*
- 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

Figure 13.15

# Poison Reverse



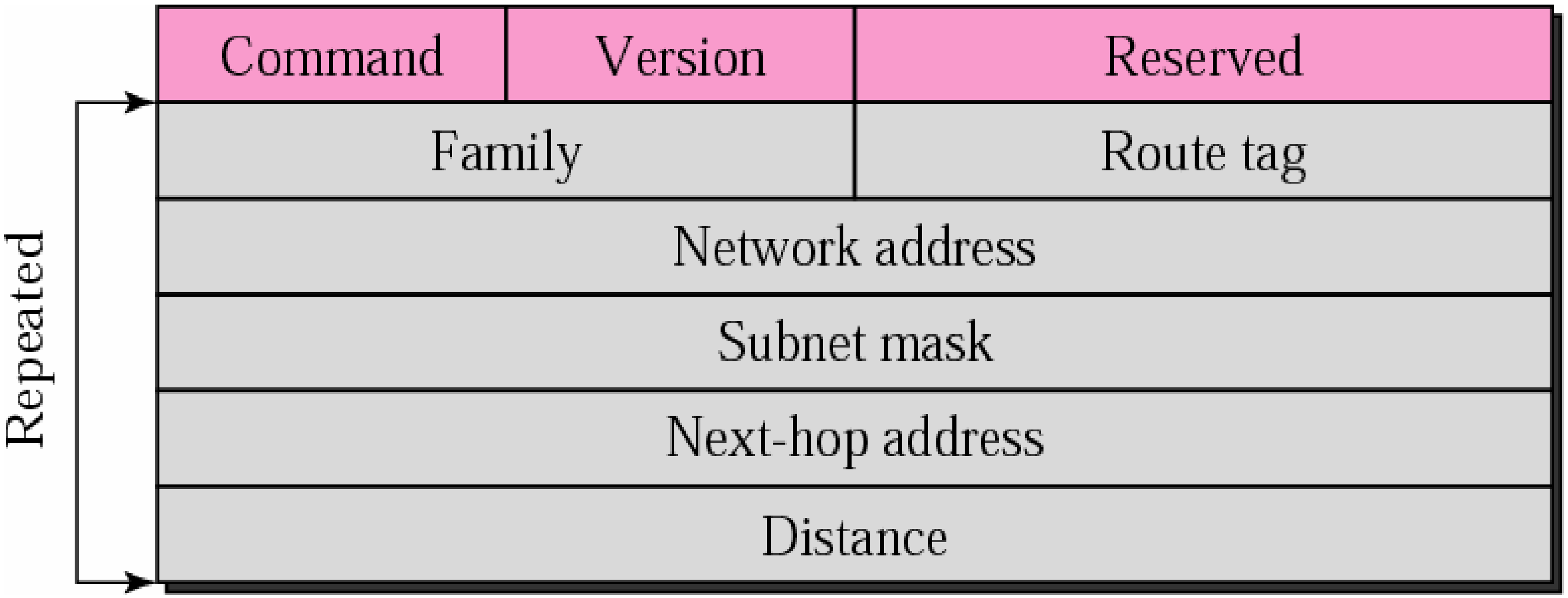
# RIP Version 2

- 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

# RIP-v2 Format



Note

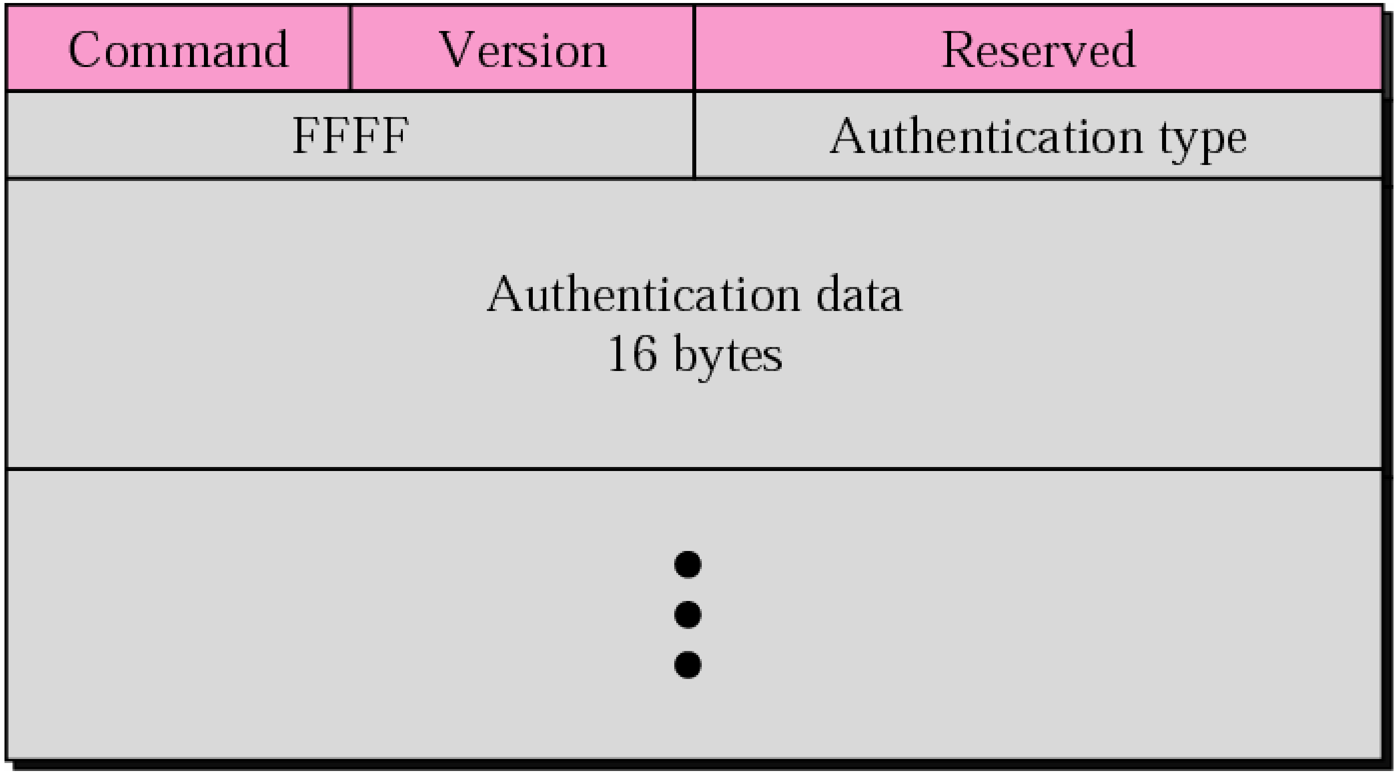
***RIP version 2 supports CIDR.***

# Authentication

- Protect the message against unauthorized advertisement
- The first entry of the message is set aside for authentication information
  - Family field = $FFFF_{16}$

Figure 13.17

# Authentication



# Multicasting

- Version 1 of RIP uses broadcasting to send RIP message to every neighbor
  - All the router 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

Note

***RIP uses the services of UDP on well-known port 520.***

# OSPF:
# Open Shortest Path First

# OSPF

- OSFP: *O*pen *S*hortest *P*ath *F*irst
- Another interior routing protocol
- OSPF divides an autonomous system into *areas*
  - To handle routing efficiently and in a timely manner

# Areas

- 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*

- Each area has a special router called *area border routers*
  - Summarize the information about the area and sent it to other areas

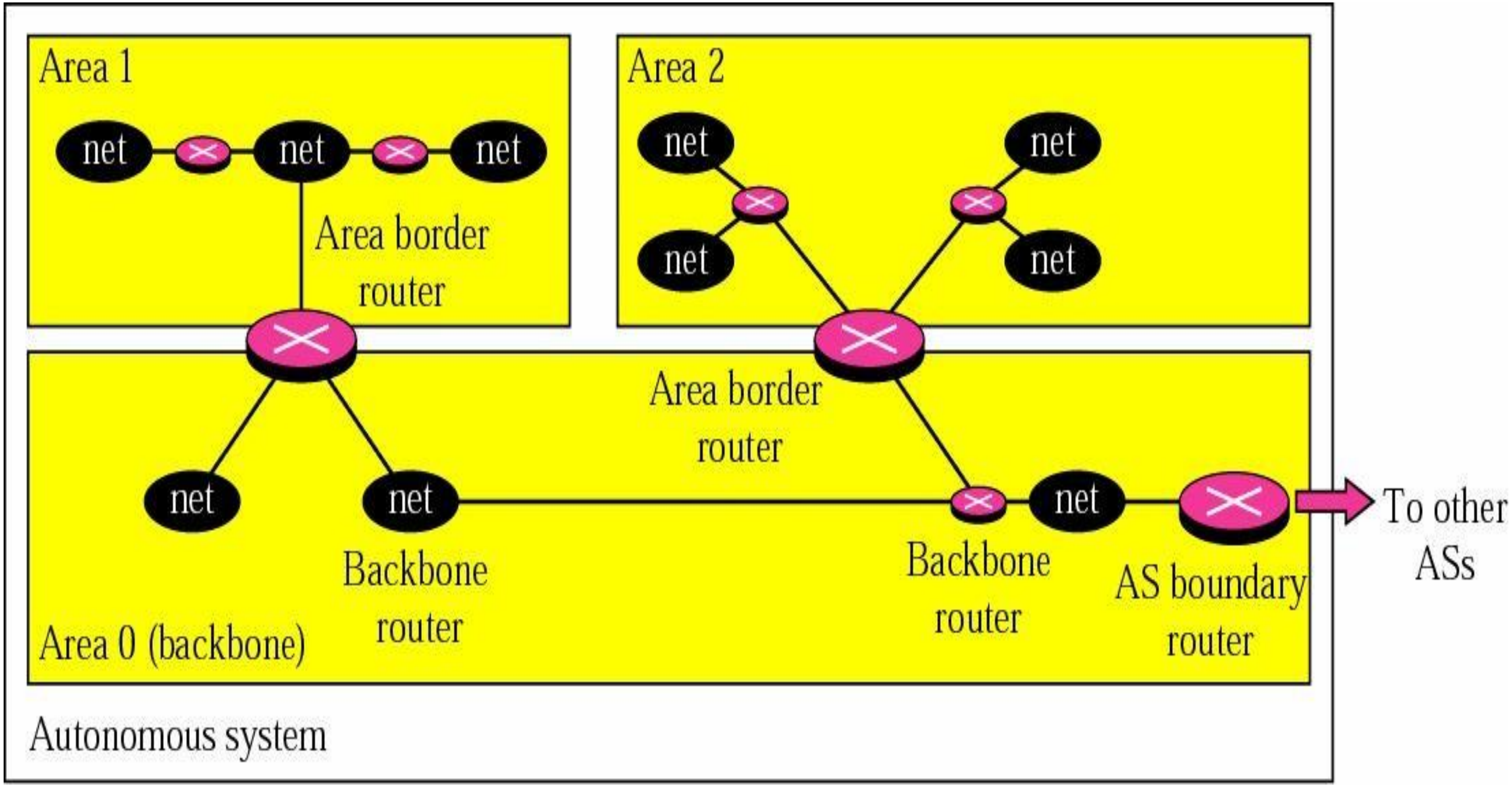
# Areas (Cont.)

- 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

# 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

# Link State Routing

- OSPF uses *link state routing* to update the routing table in an area
- **Link state routing:** a process by which each router shares its knowledge about its neighbor with every router in the area

# Link State Routing (Cont.)

- Three keys
  - **Sharing knowledge about the neighborhood**
    - Each router sends the *state of its neighborhood* to every other router in the area
  - **Sharing with every other router**
    - Each router sends the state of its neighborhood to *every other router in the area* by *flooding*
  - **Sharing when there is a change**
    - Each router shares the state of its neighborhood only *when there is a change*

# Link State Routing (Cont.)

- The idea behind link state routing
  - *Each router should have the exact topology of the internet at every moment*
- Thus, every router should have the whole picture of the internet
  - A router can calculate the shortest path between itself and each network
- But, how to represent an internet by graph?

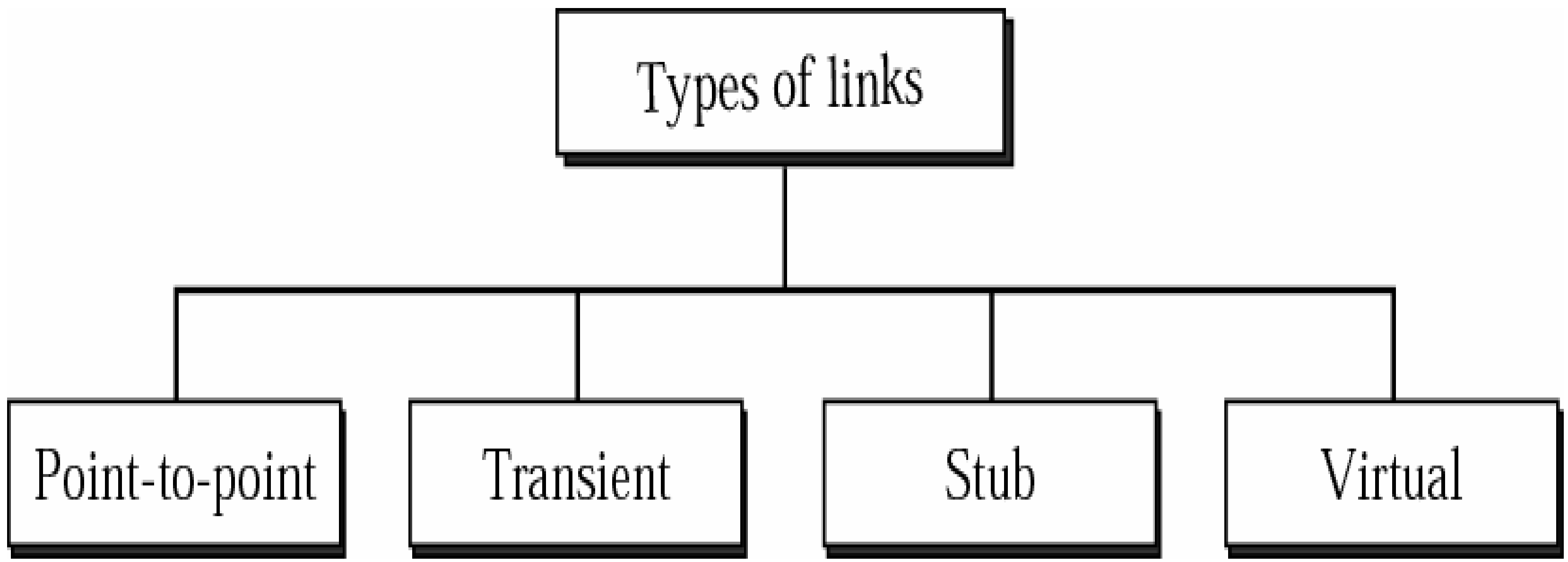
# Types of Links

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

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

Figure 13.10

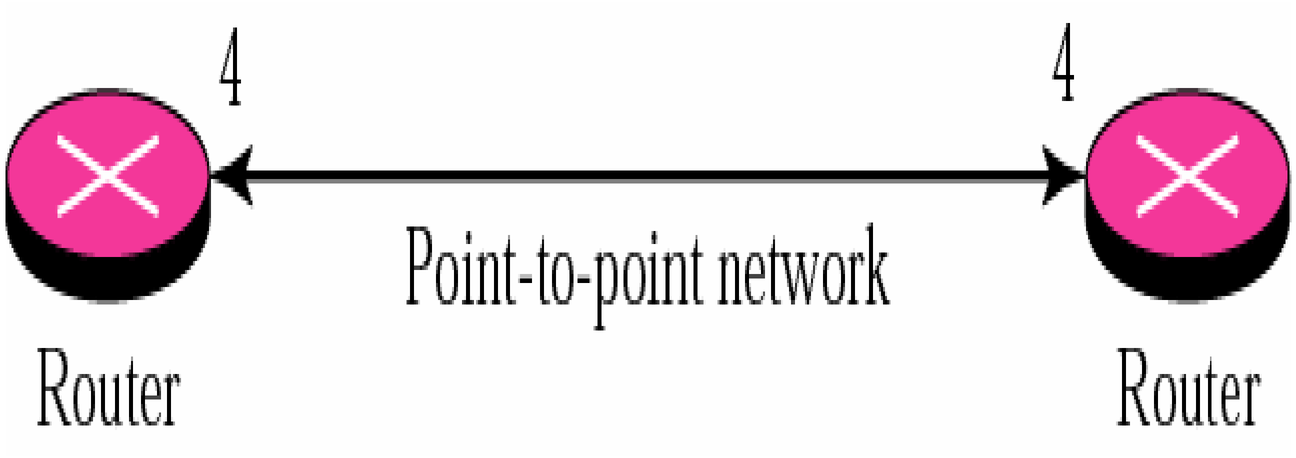
# Types of Links



# Point-to-Point Link

- Connect two routers without any other host or router in between
- 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

# 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

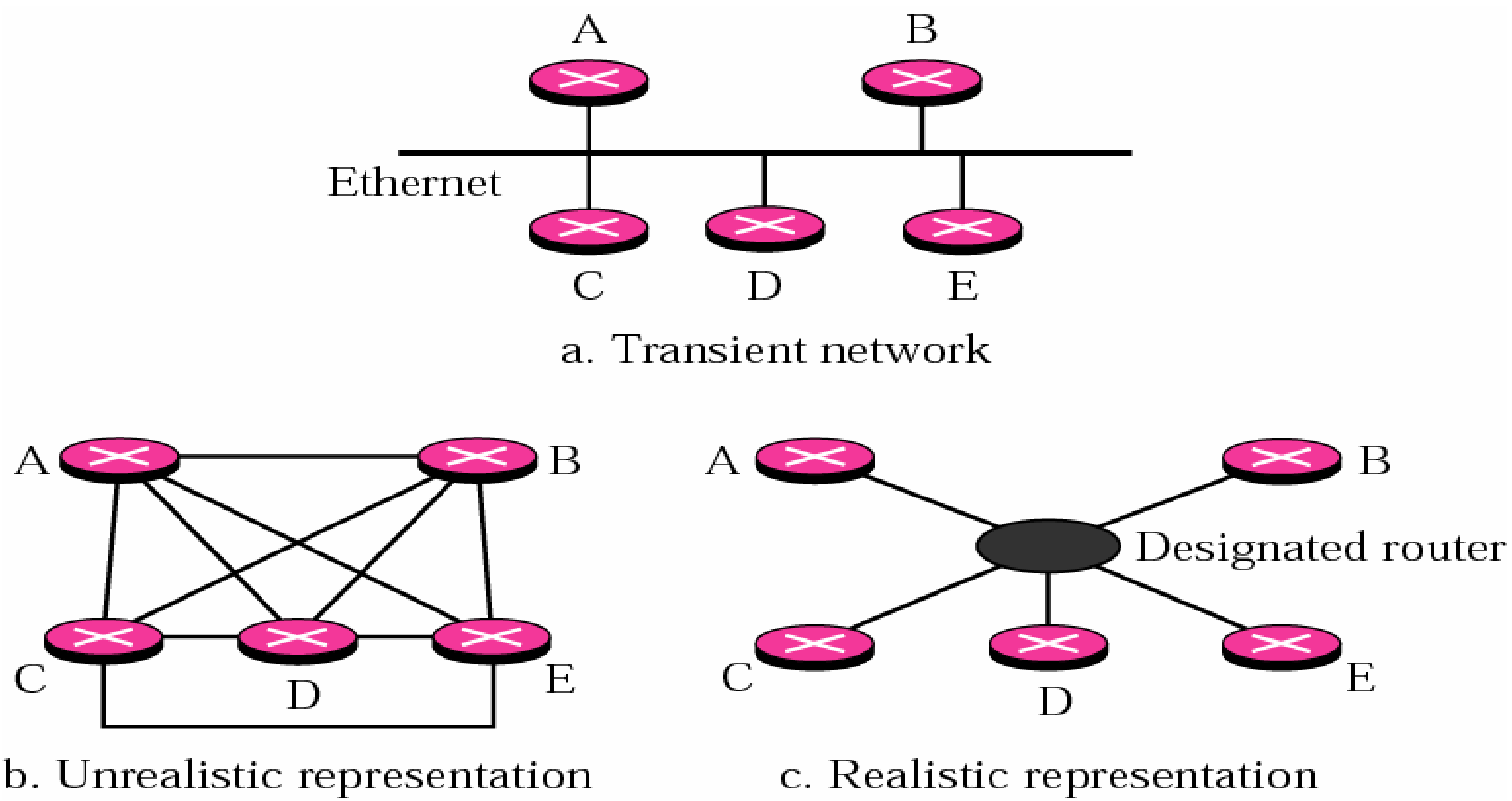
- Example
  - All LANs and some WANs with two or more routers

# 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
- 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 13.31

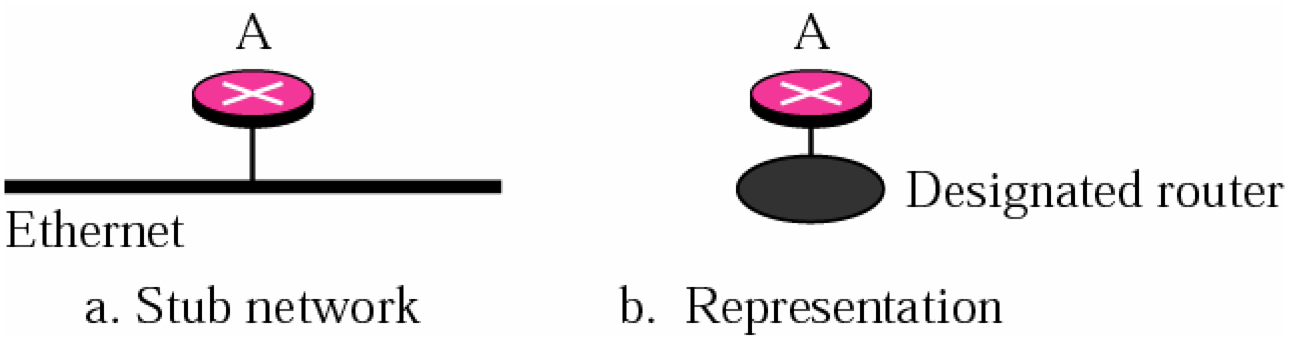
# Transient Link



# 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 designed router as the network
  - Note, the link is only one-directional
    - From the router to the network

# Stub Link

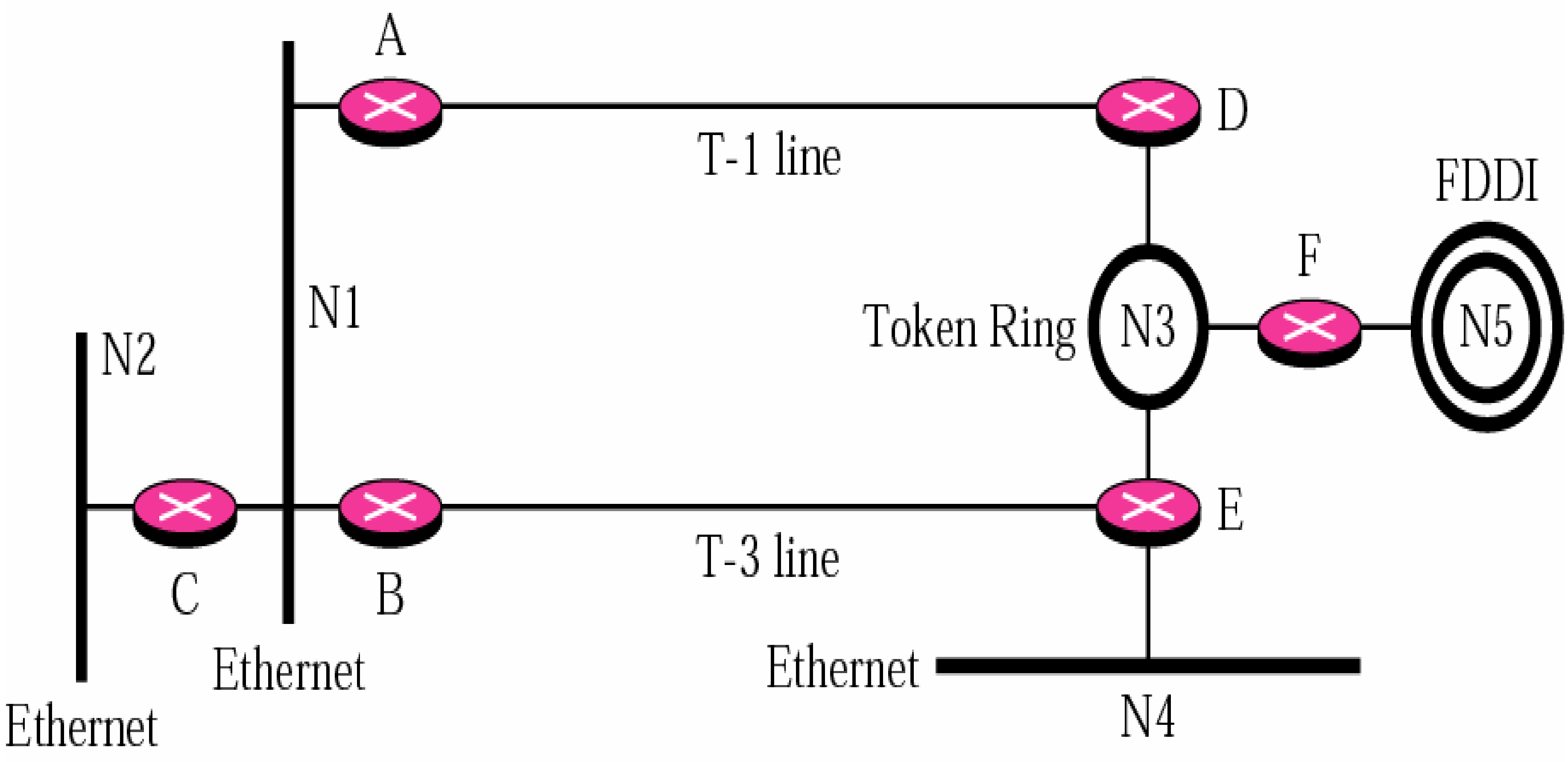

A
Ethernet
a. Stub network
A
Designated router
b. Representation

# Virtual Link

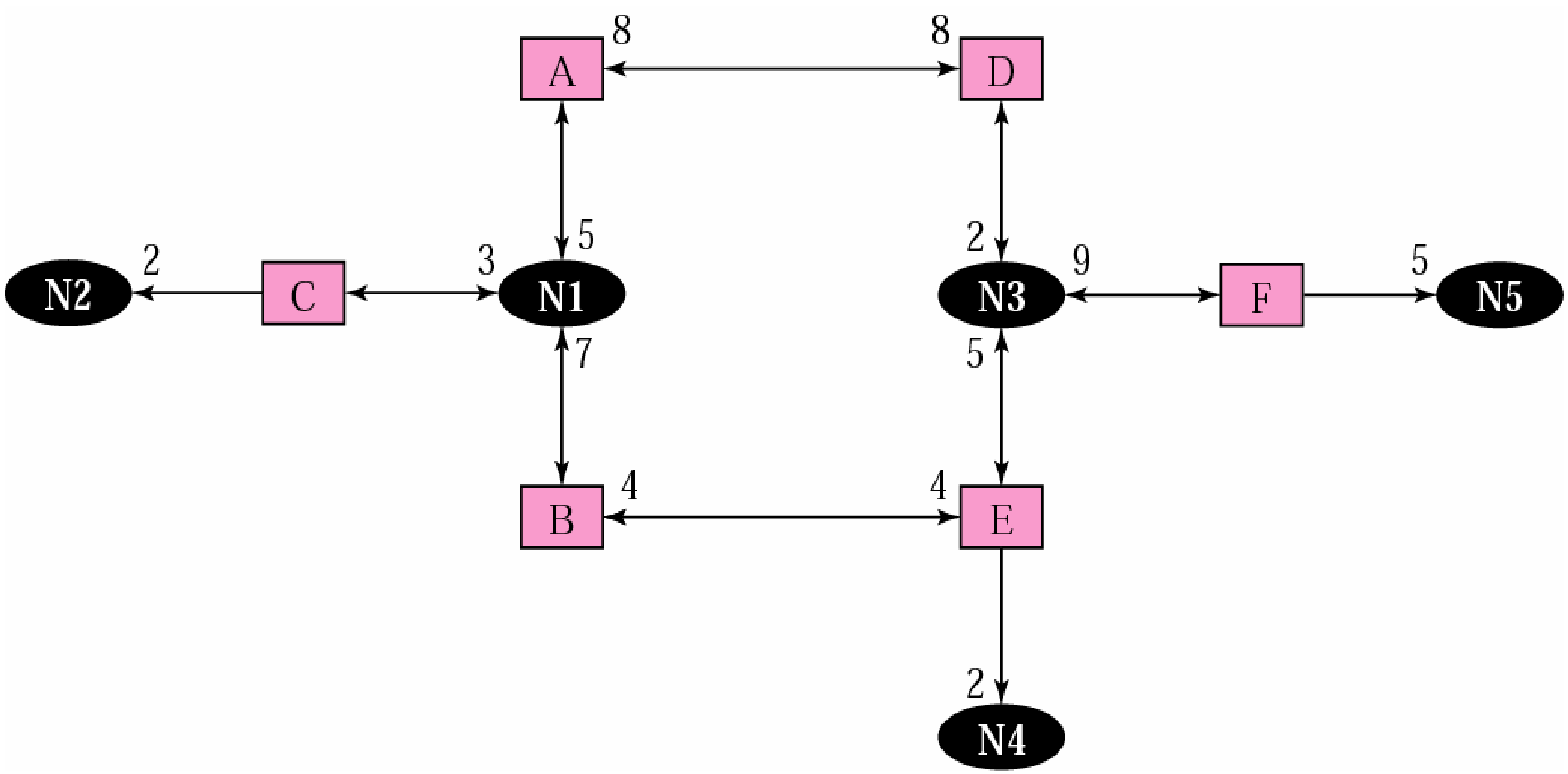
- When the link between two routers is broken
  - The administrator may create a virtual path between them using a longer path and may go through several routers

Figure 13-33

# Example of an internet



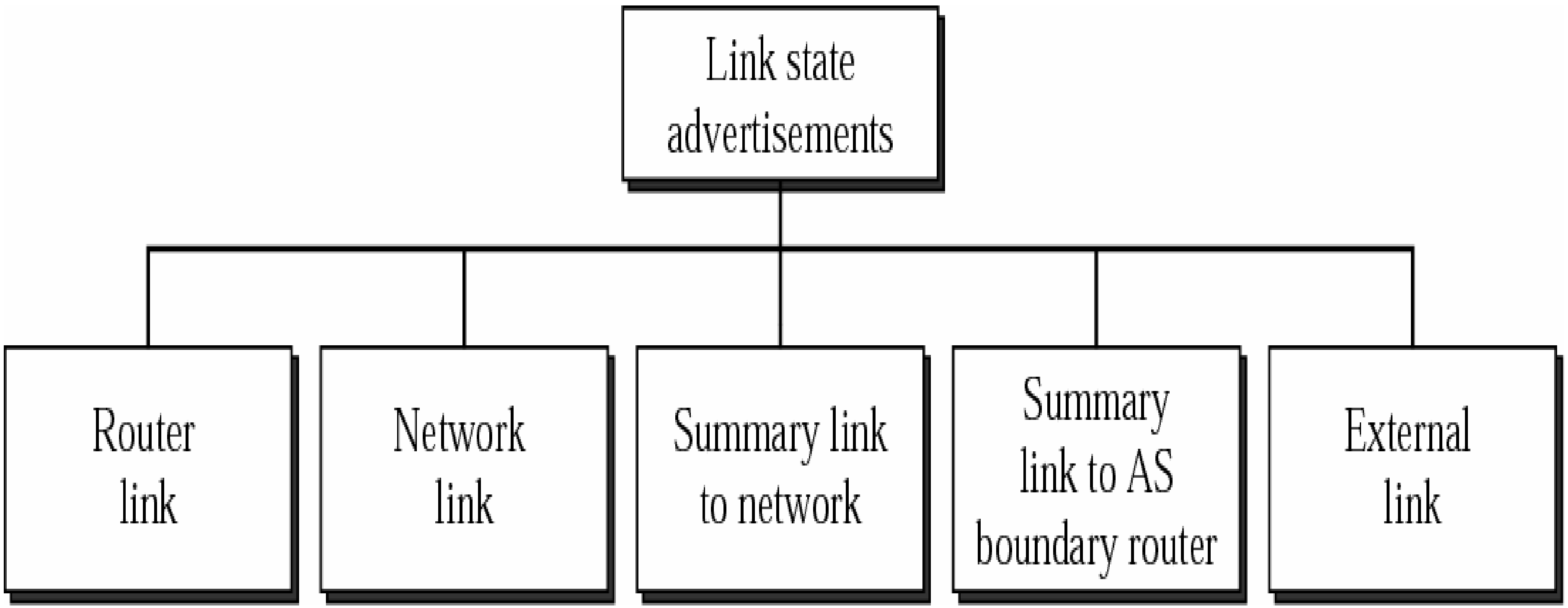
# Graphical Representation of an internet



# Link State Advertisements

- Each entity distributes *Link State Advertisement* to share information about their neighbors
- An LSA announces the states of entity links
- Five LSAs, depend on the type of entity
  - Router link
  - Network link
  - Summary link to network
  - Summary link to AS boundary router
  - External link

# Types of LSAs

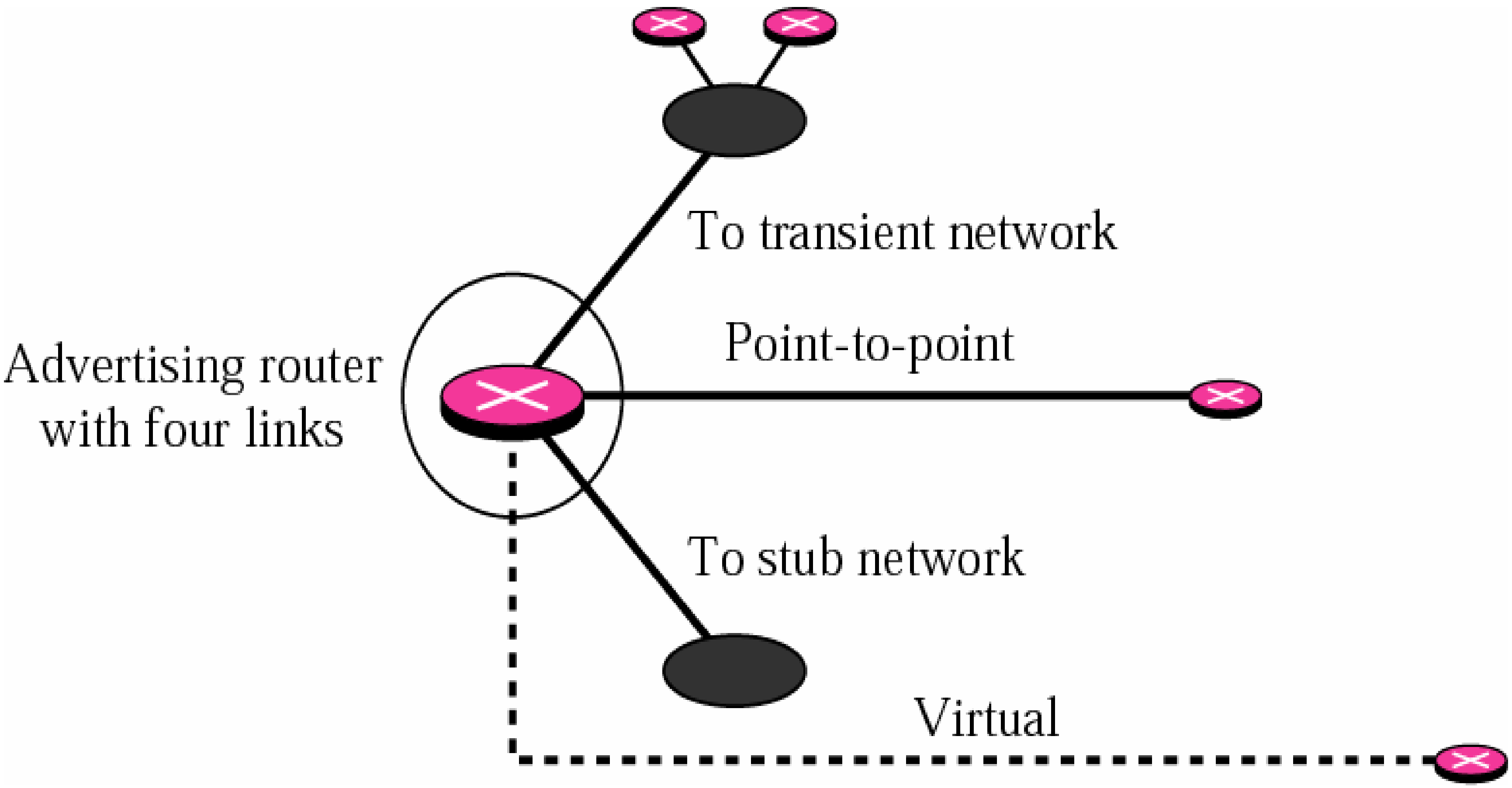


# Router Link

- Define the links 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)

# Router Link



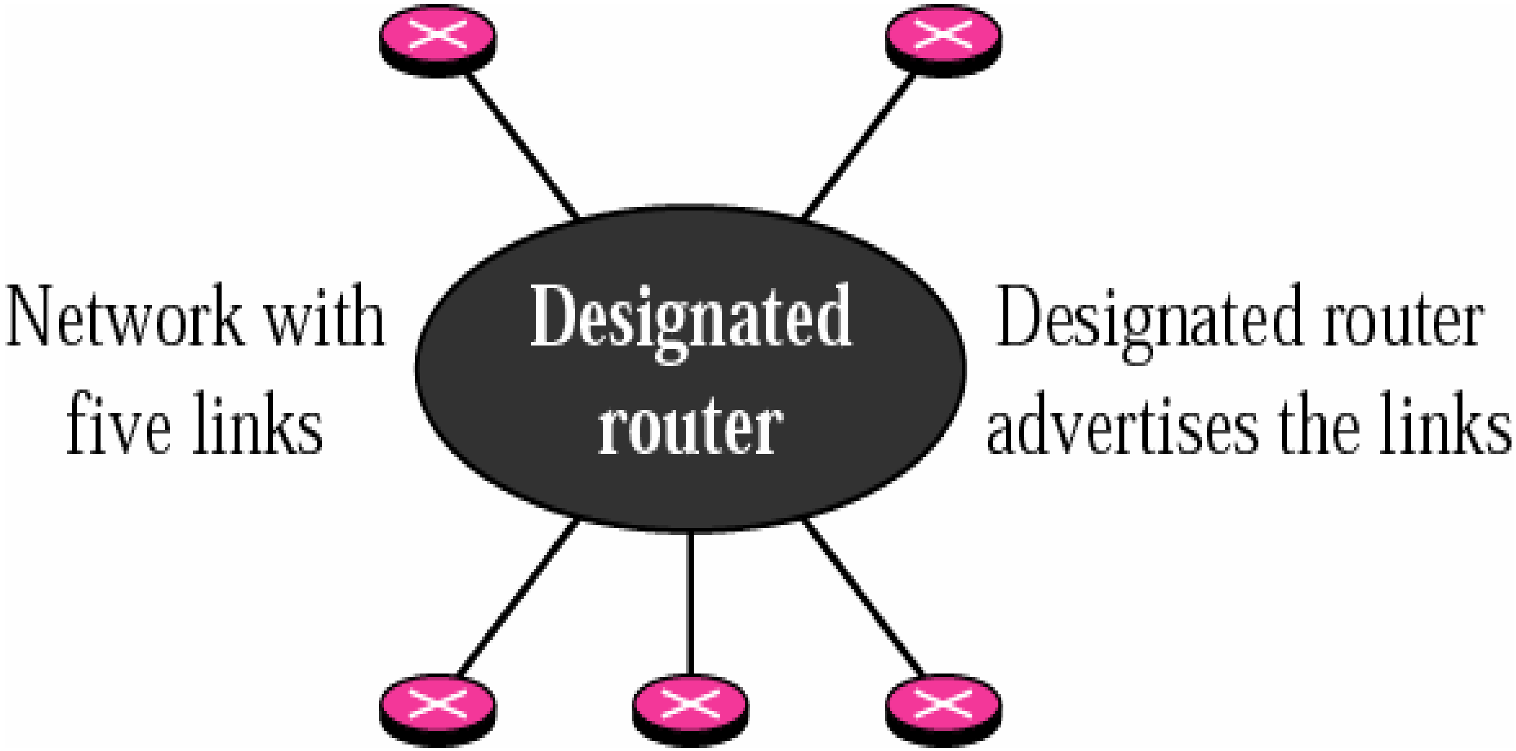
To transient network
Point-to-point
Advertising router
with four links
To stub network
Virtual

# Network Link

- Defines the links of a network
- A designated 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

Figure 13-27

# Network Link



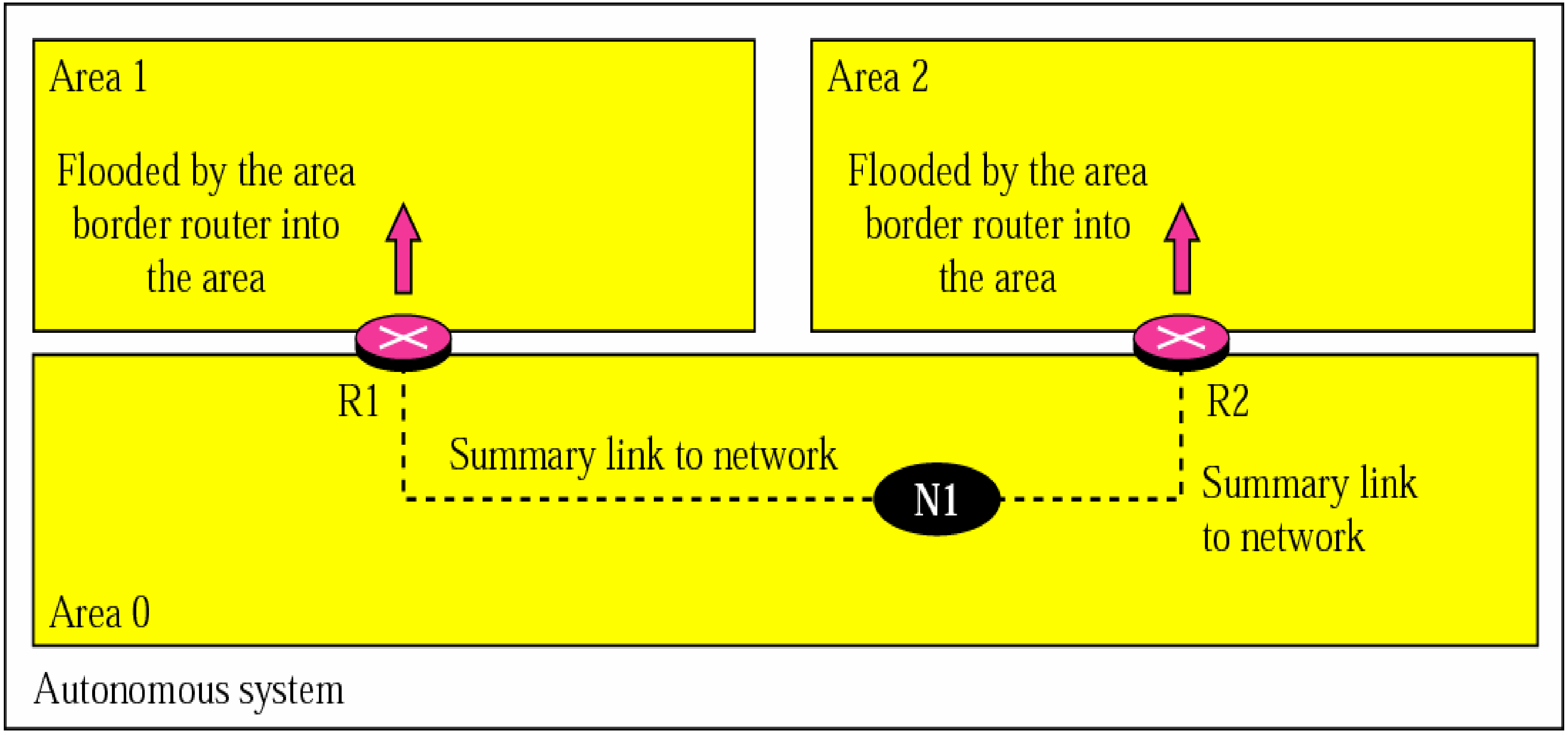
# Summary Link to Network

- 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 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
  - Two routing table: 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

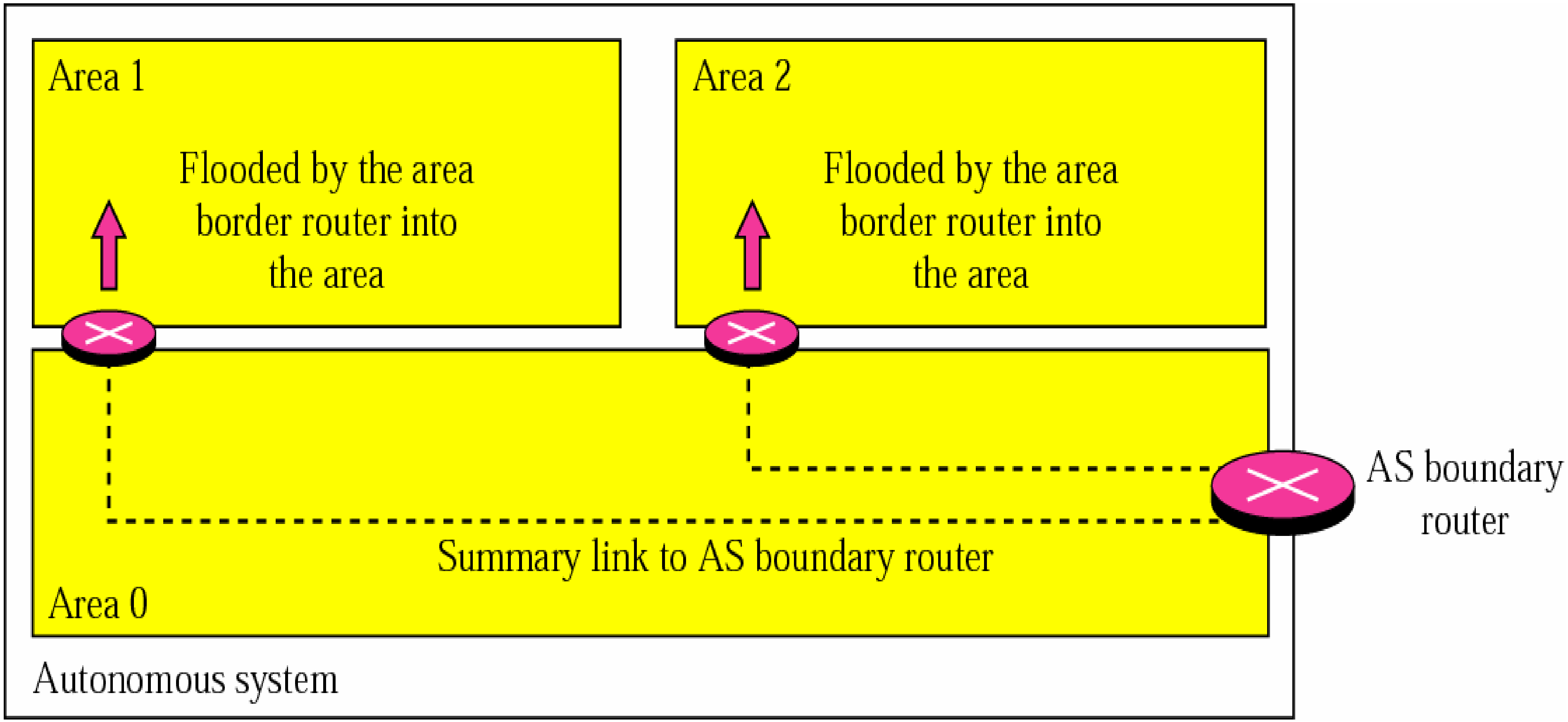
# Summary Link to Network



# Summary Link to AS Boundary Router

- 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

# Summary Link to AS Boundary Router



# External Link

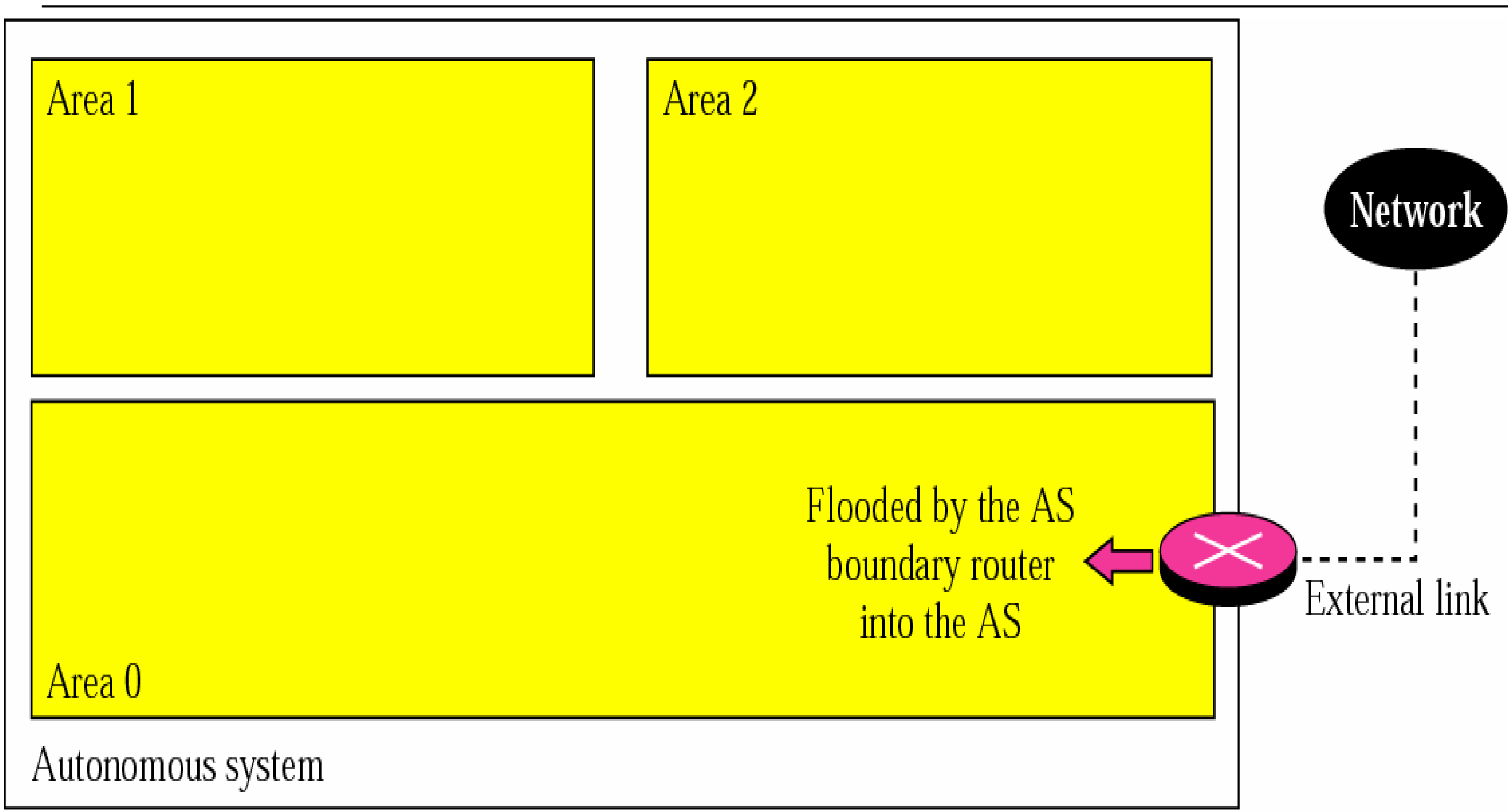
- How a router inside an AS know which networks are available outside the AS ?

- 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*

# External Link (Cont.)

- Notably, each advertisement announces one single network
  - Separate announcements are made if more than one network exists

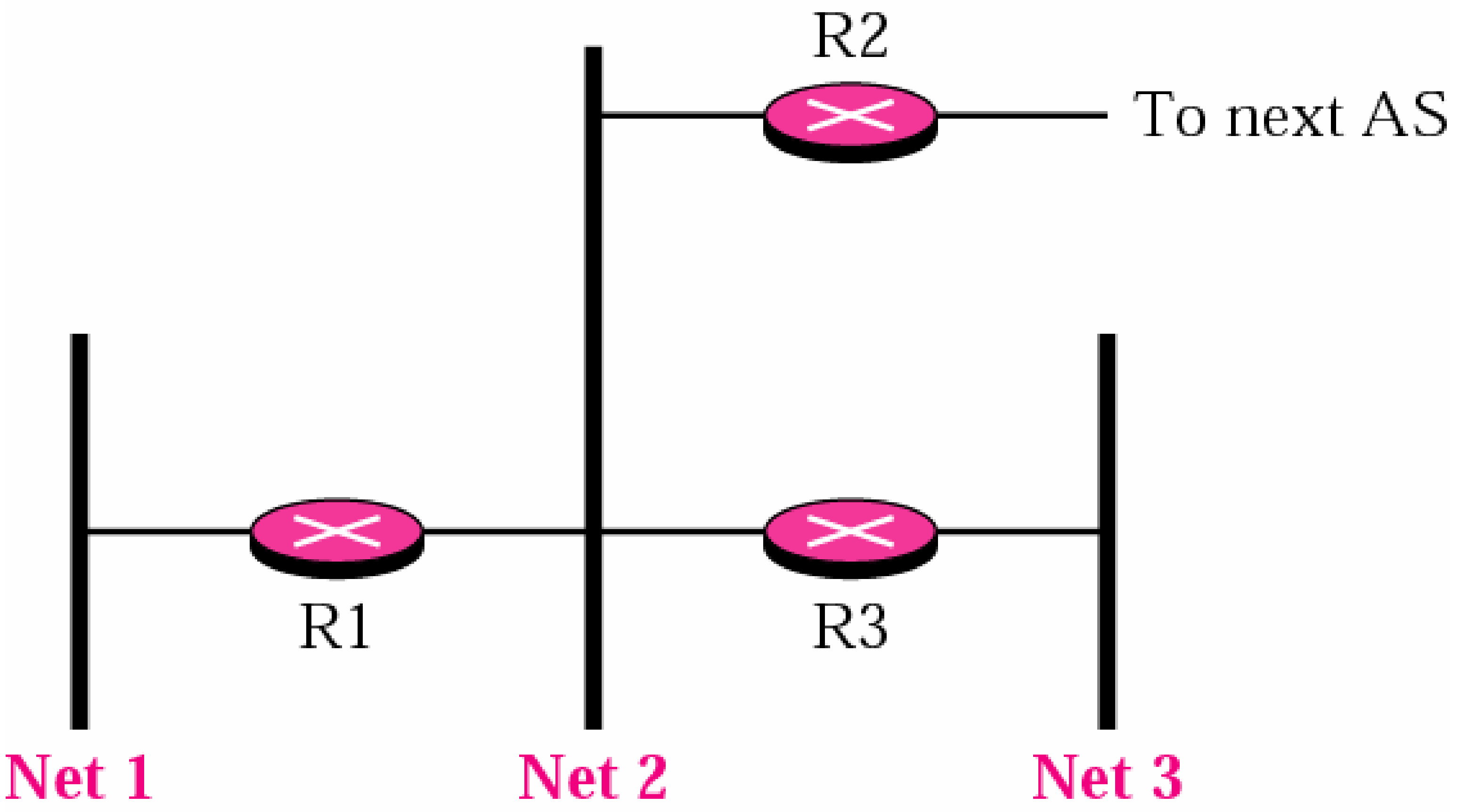
# External Link



## *Example 3*

- In Figure 13.31 (next slide), which router(s) sends out *router link LSAs*?

# Example 3 and Example 4



## *Solution*

All routers advertise router link LSAs.

R1 has two links, Net1 and Net2.

R2 has one link, Net2 in this AS.

R3 has two links, Net2 and Net3.

## *Example 4*

In Figure 13.31, which router(s) sends out the *network link LSAs*?

## *Solution*

All three network must advertise network links:

Advertisement for Net1 is done by R1 because it is the only router and therefore the designated router.

Advertisement for Net2 can be done by either R1, R2, or R3, depending on which one is chosen as the designated router.

Advertisement for Net3 is done by R3 because it is the only router and therefore the designated router.

# Link State Database

- Every router in an area
  - Receive the router link and network link LSA from every other router
  - And form a *link state database*
    - A tabular representation of the topology of the internet *inside an area*
- Notably, every router in the same area has the same link state database

**Note**

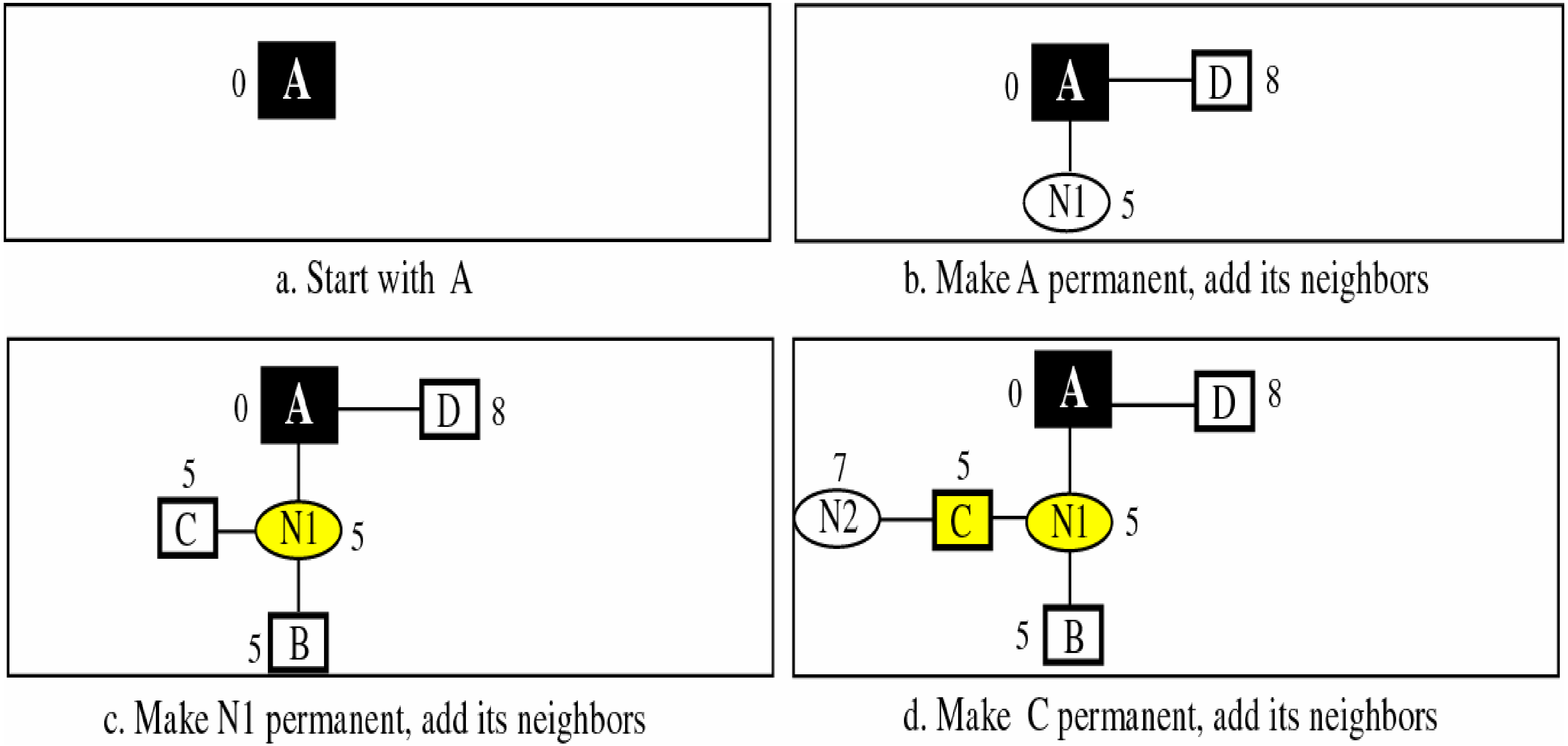
***In OSPF, all routers have the same link state database.***

# Dijkstra Algorithm

- Each router applies the *Dijkstra algorithm* to from its *link state database*

- Dijkstra algorithm
  - Calculate the shortest path between two points

# Shortest Path Calculation



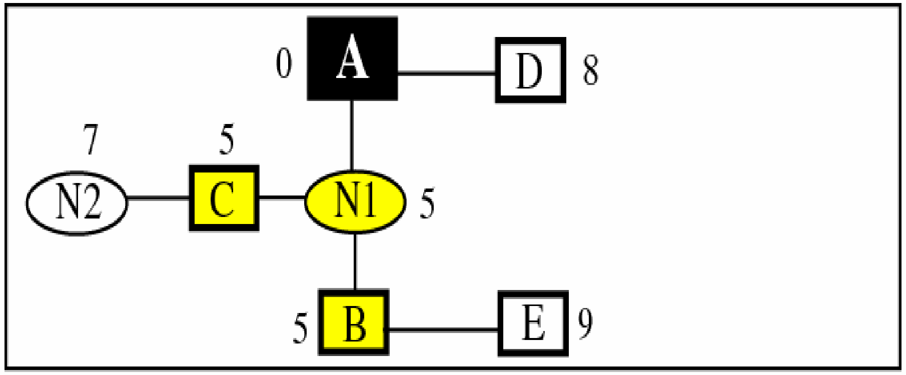

a. Start with A

b. Make A permanent, add its neighbors

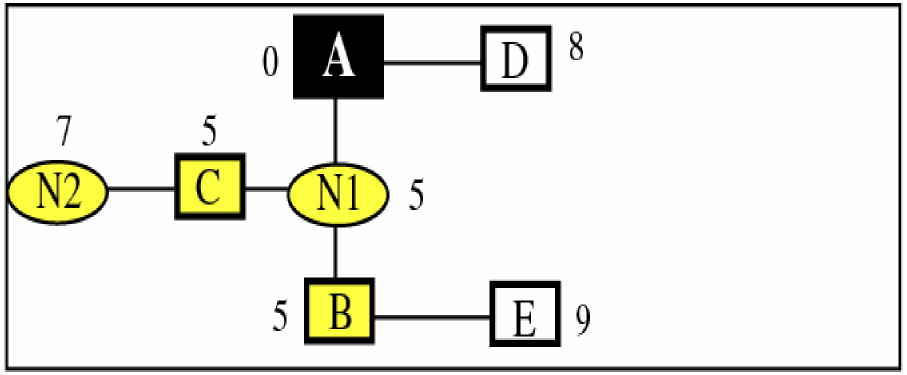
c. Make N1 permanent, add its neighbors

d. Make C permanent, add its neighbors

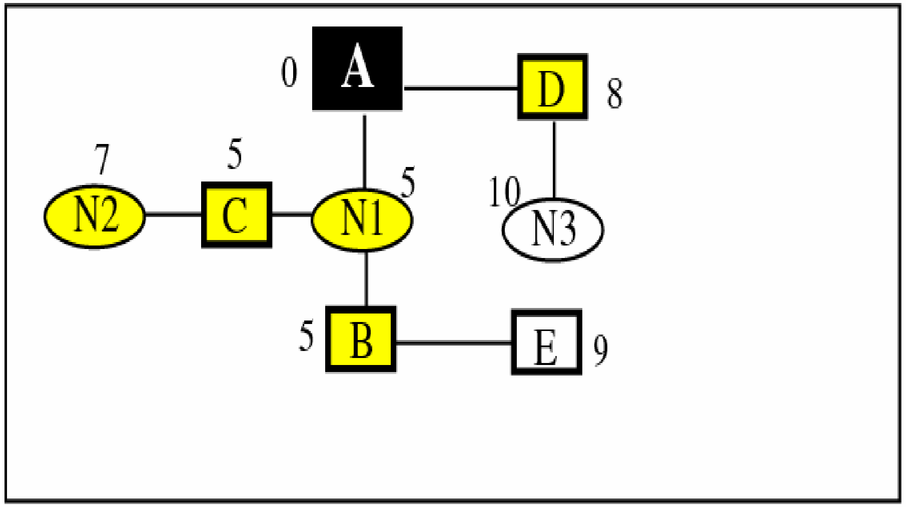
# Shortest Path Calculation



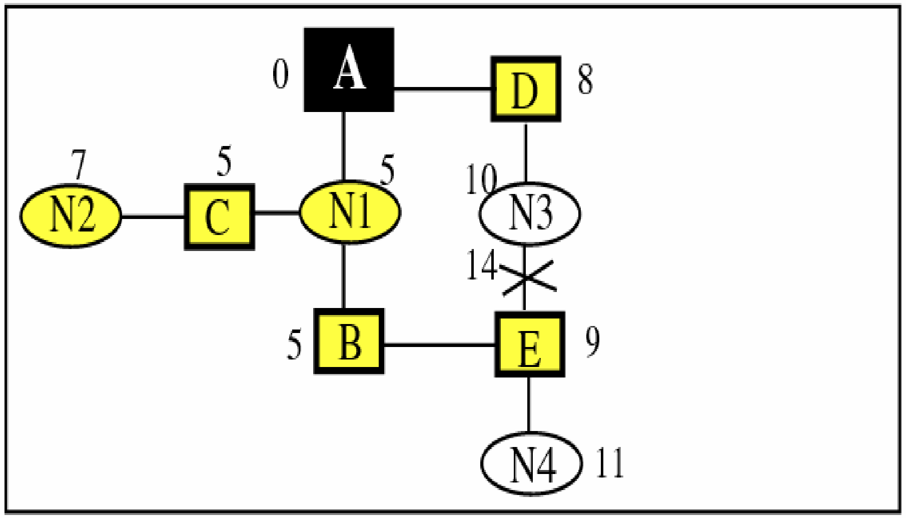

e. Make B permanent, add its neighbors




f. Make N2 permanent

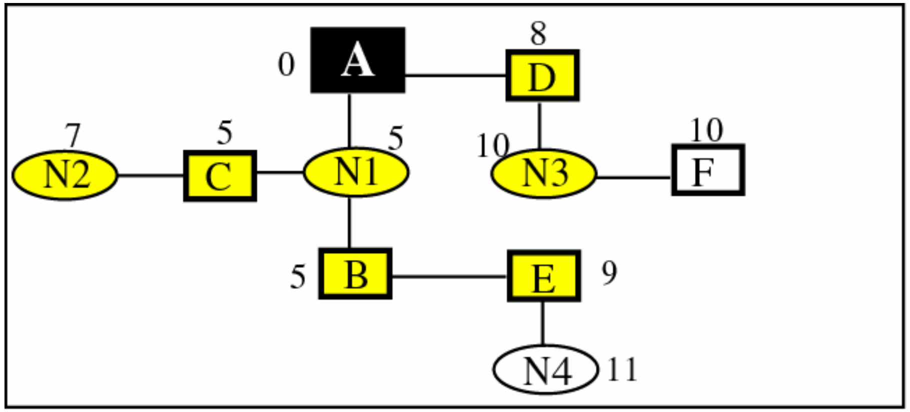



g. Make D permanent, add its neighbors

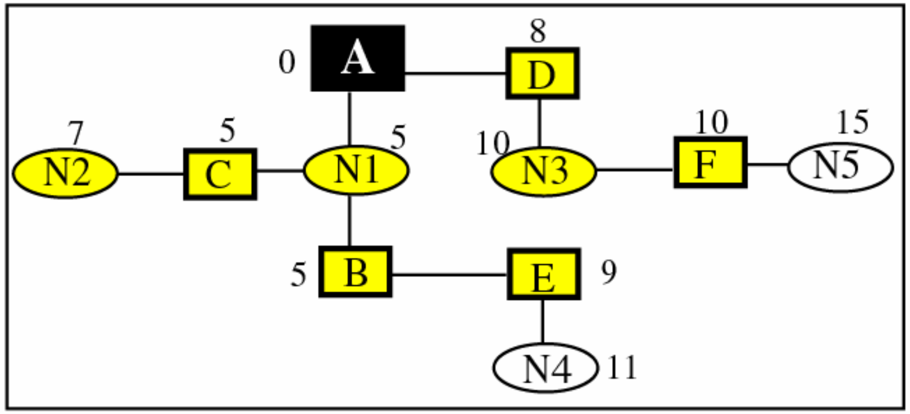



h. Make E permanent, add its neighbors

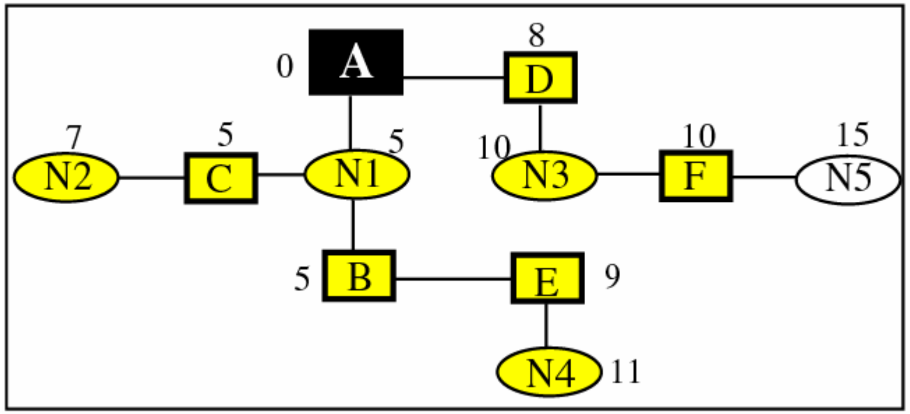
# Shortest Path Calculation



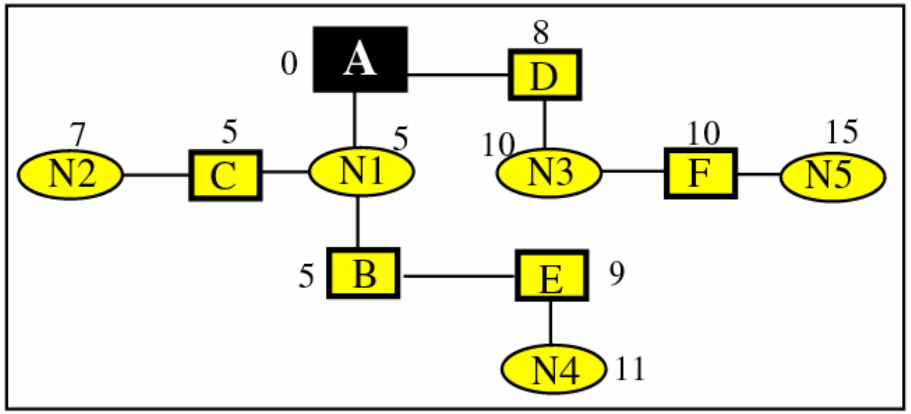

i. Make N3 permanent, add its neighbors




j. Make F permanent, add its neighbors




k. Make N4 permanent




l. Make N5 permanent

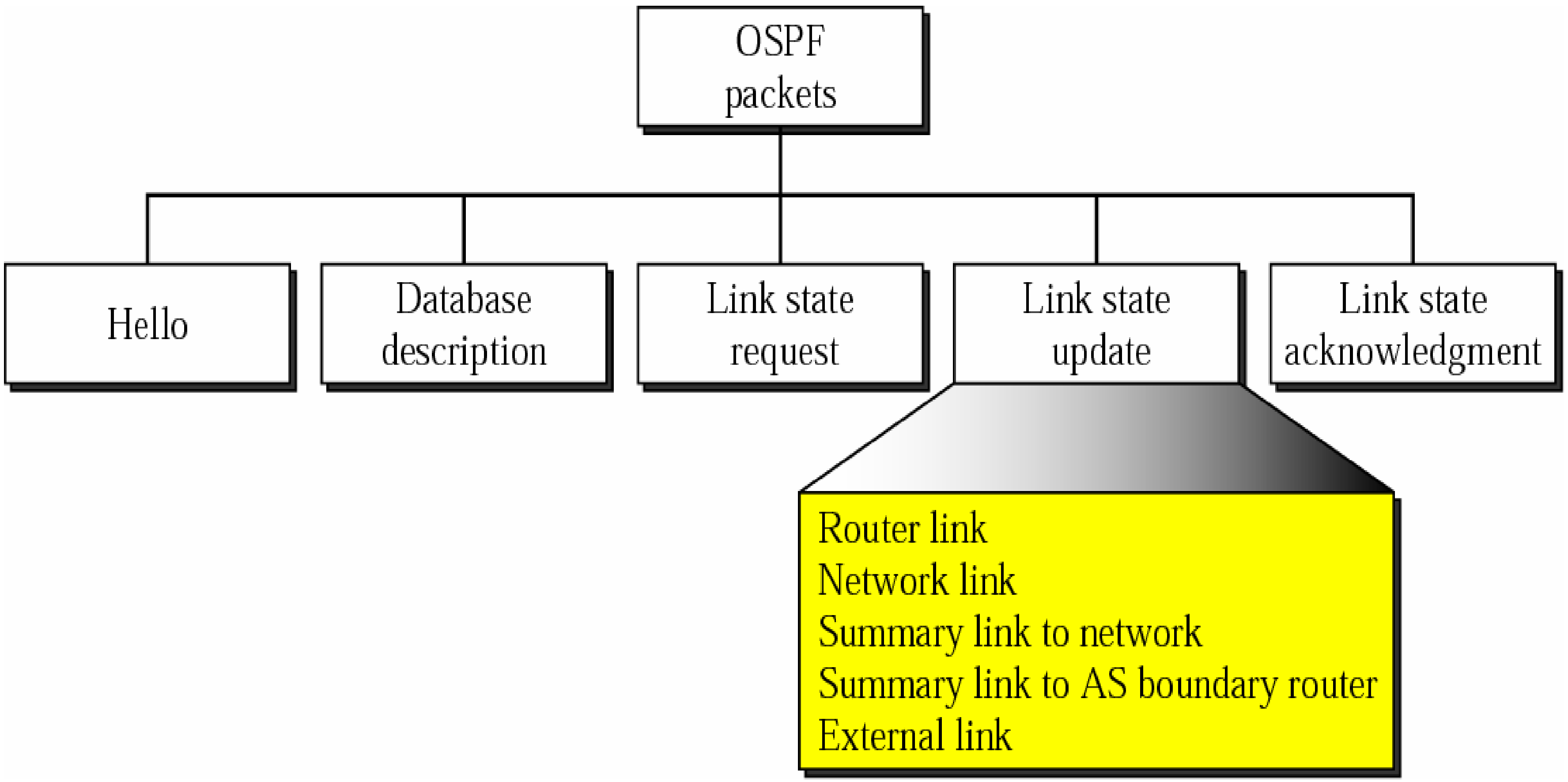
# Routing Table

- Each routers uses the shortest path tree method to construct its routing table
- Routing table in OSPF includes the cost of reaching each network in the area
- To find the cost of reaching network outside of the area, the routers use the
  - Summary link to network
  - Summary link to boundary router
  - External link advertisements

# 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

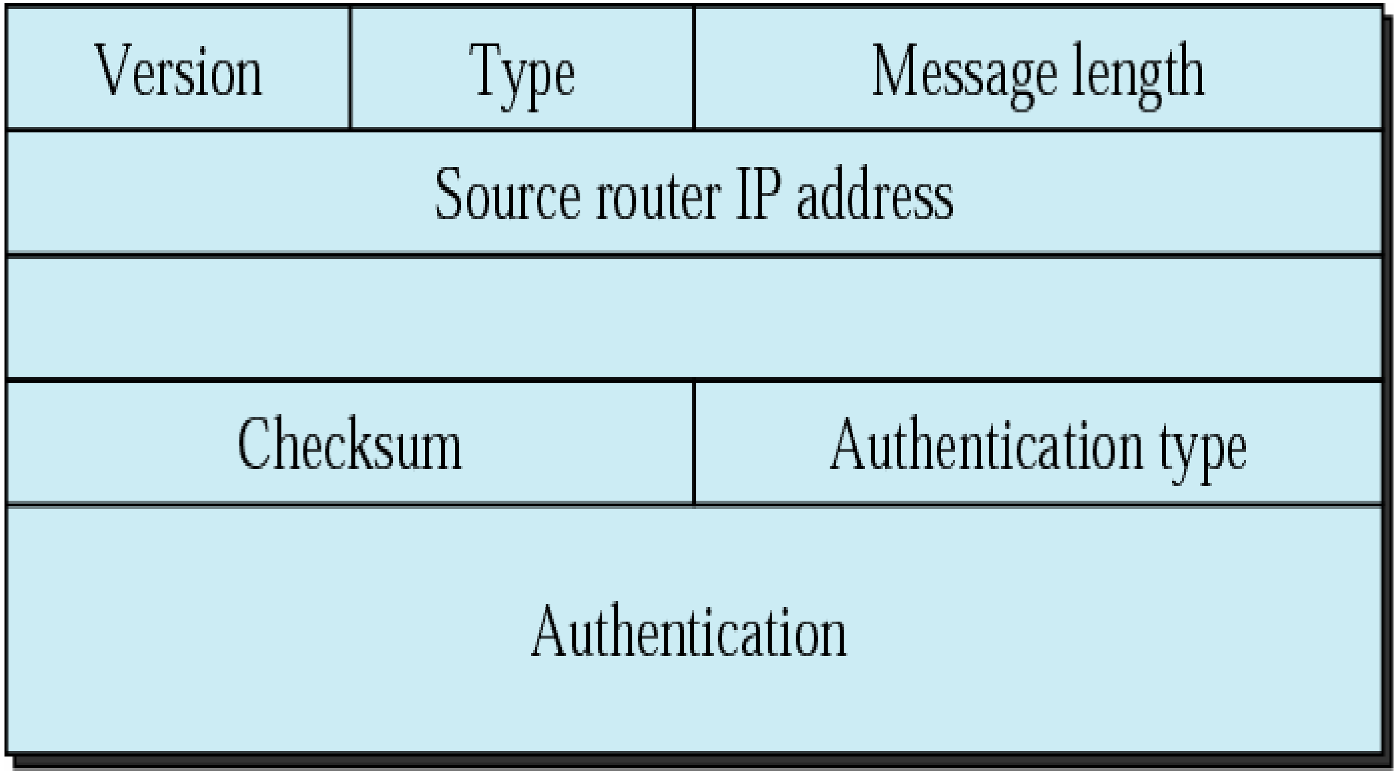
# Types of OSPF Packets



# Packet Format

- 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 route IP address: 32-bit
    - The IP address of the router that sends the packet

# OSPF Packet Header



# Packet Format (Cont.)

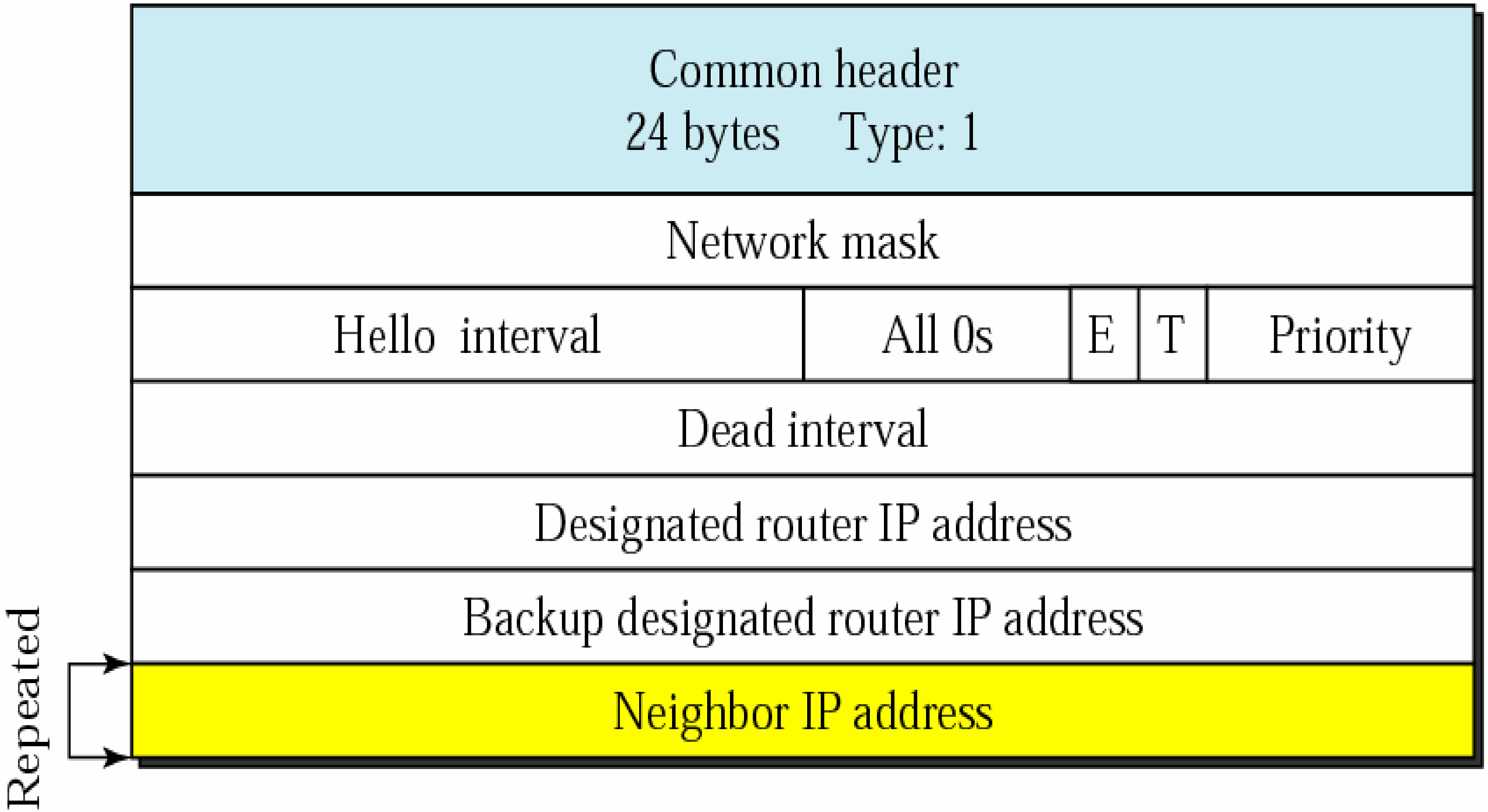
- Area identification: 32-bit
  - The area within which the routing take place
- Checksum: 16-bit
  - Error detection on the entire packet excluding the authentication type and authentication data field
- Authentication type: 16-bit
  - Define the authentication method used in this area
  - 0: none,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

# 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

# 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
- F 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*

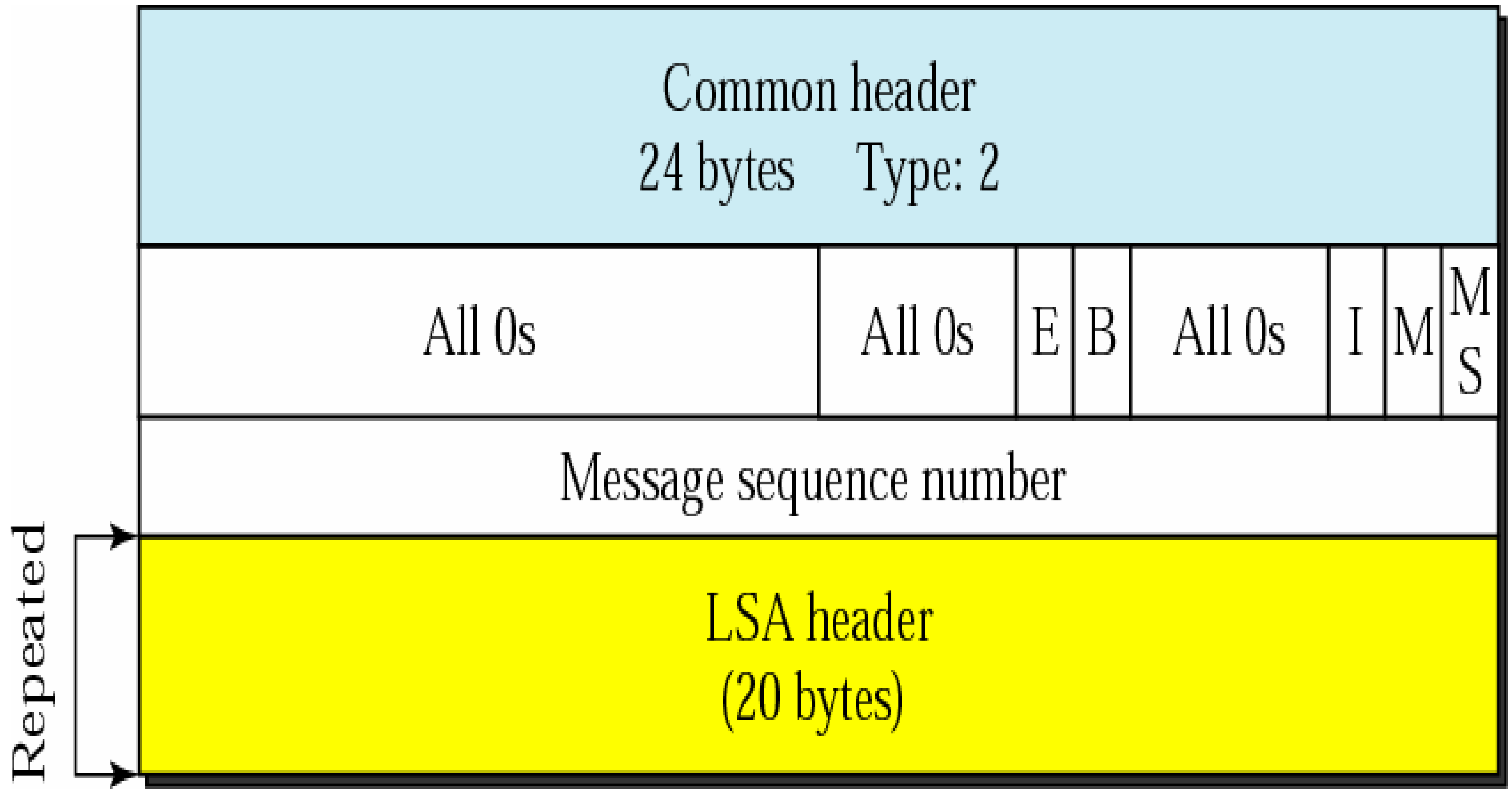
# Database Description Message (Cont.)

- 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

# 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

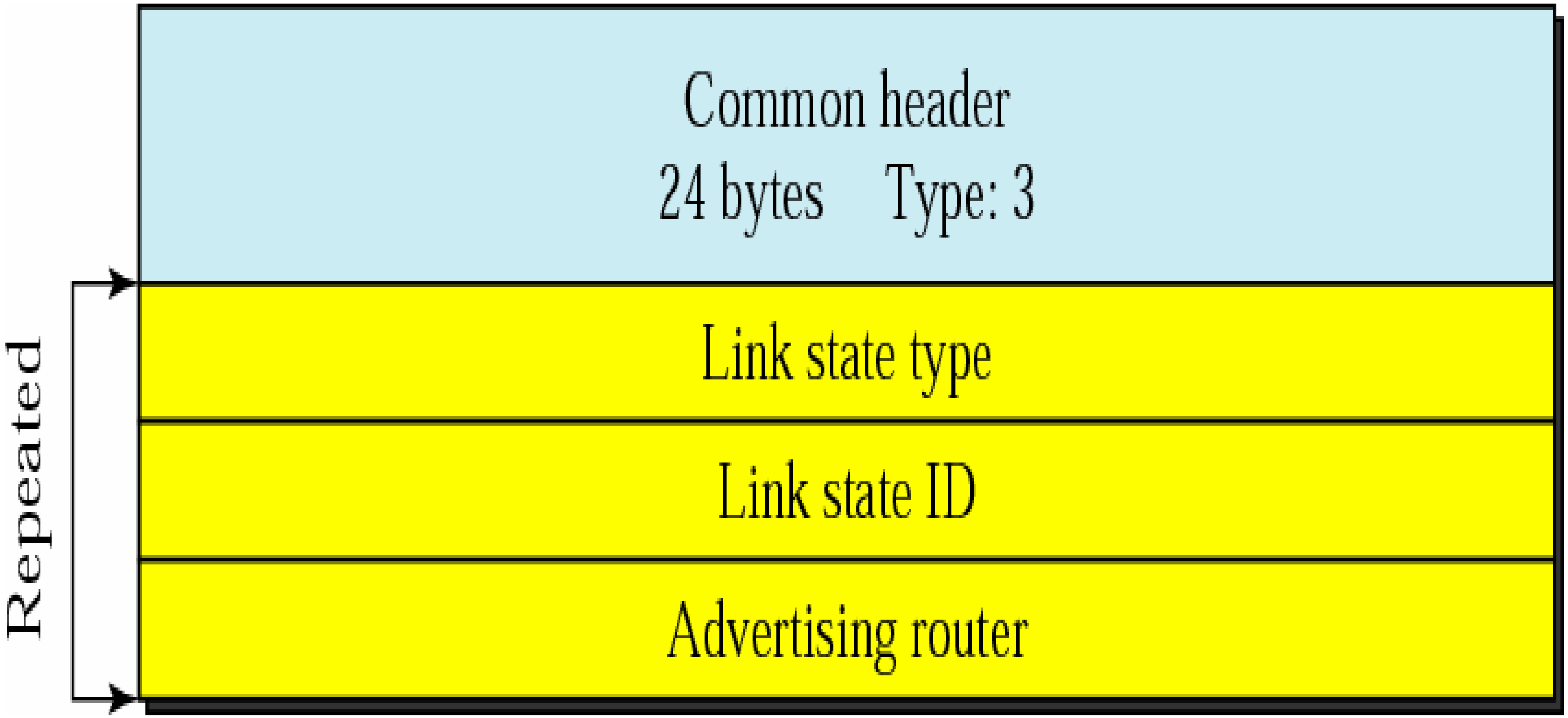
# Database Description Message Format (Cont.)

- 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*

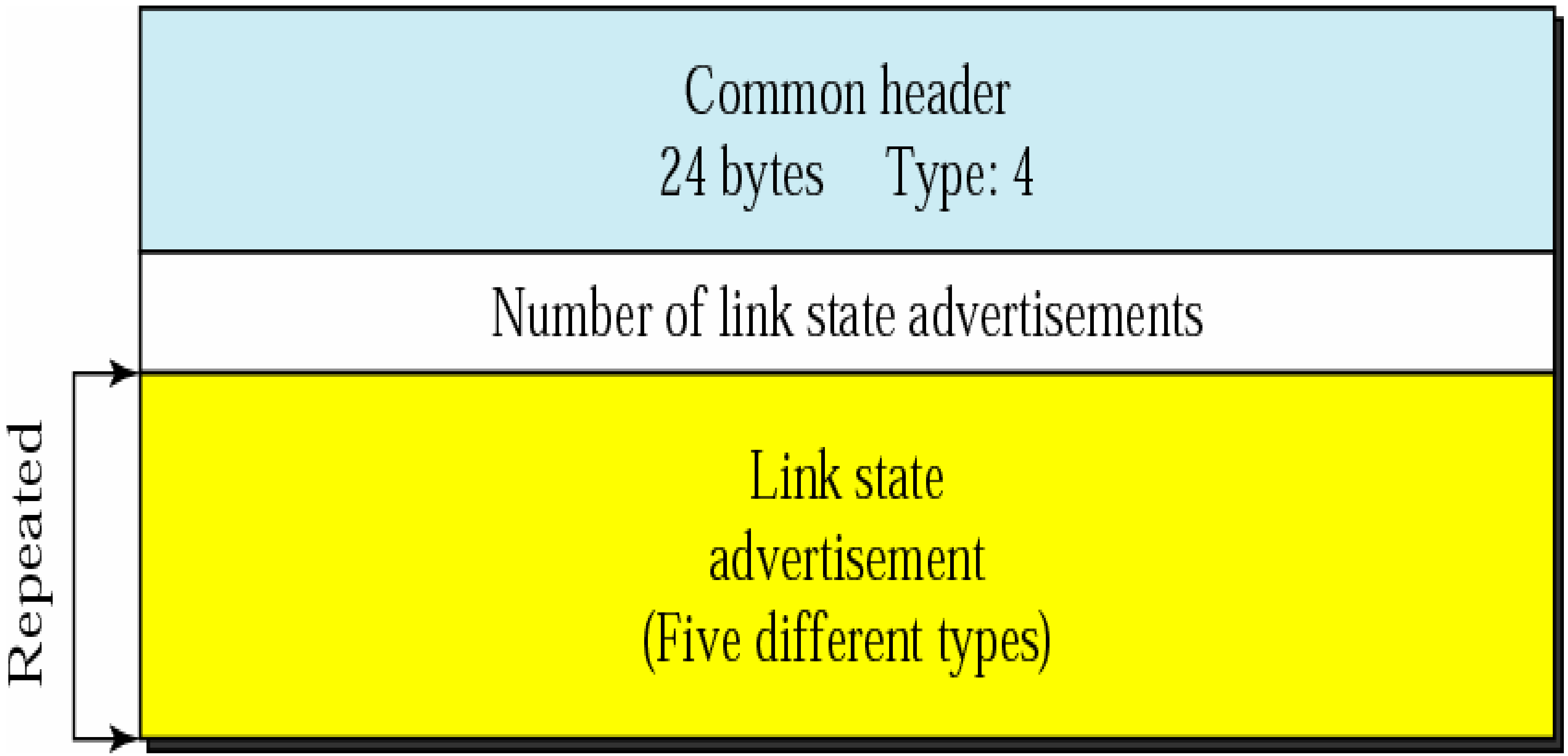
# Link State Request Packet



# Link State Update Packet

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

# Link State Update Packet



# LSA Format

- 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 Format (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 Format (Cont.)

- Link state ID: depend on the type of link
  - Router link: IP address of the router
  - Network link: IP address of the designed 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 Format (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 13-30

# LSA Header

| | | | | |
|---|---|---|---|---|
| Link state age | Reserved | E | T | Link state type |
| Link state ID | | | | |
| Advertising router | | | | |
| Link state sequence number | | | | |
| Link state checksum | | Length | | |

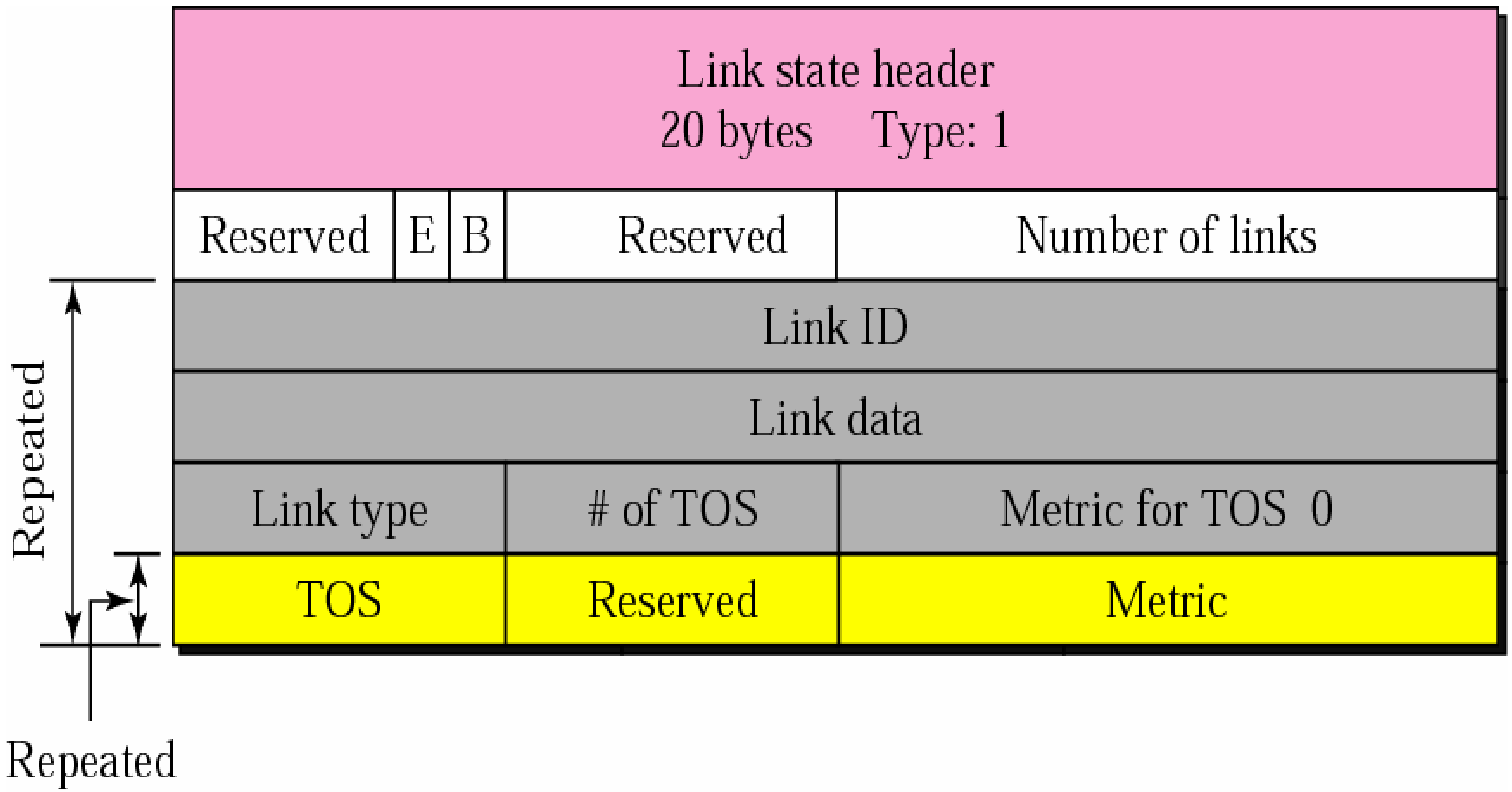
# Router Link LSA

- Advertise all of the links of a router
- Format
  - Link ID:
    - Depend on the type of link, see Table 13.3
  - Link data:
    - Give additional information about the link, also depend on the type of link, see Table 13.3
  - Link type:
    - Four different types of links are defined based on the type of network, see Table 13.3
  - Number of types of services (TOS)
    - The number of type of services announced for each link

# Router Link LSA (Cont.)

- 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

# Router Link LSA



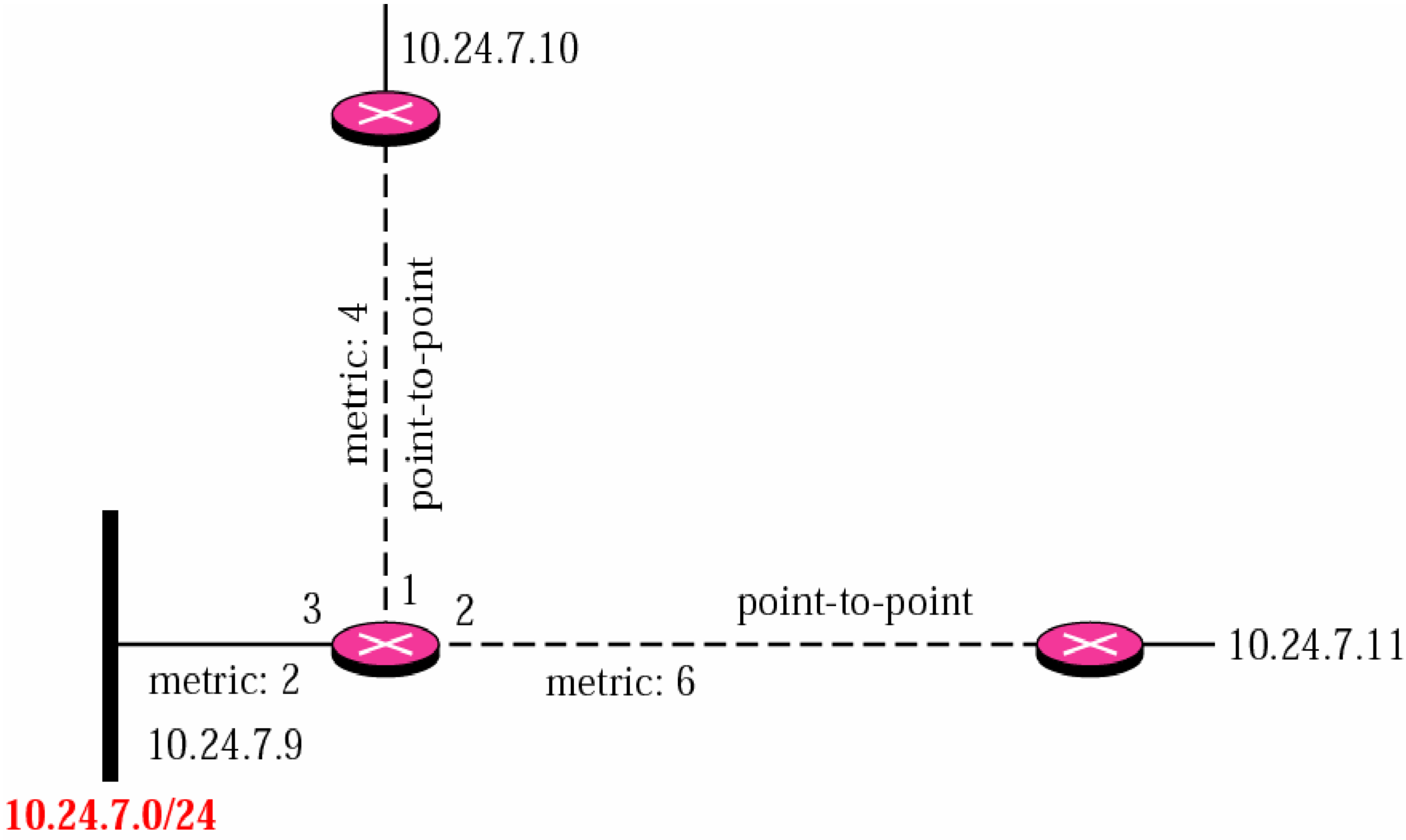
# Table 13.3

| **Link Type** | **Link Identification** | **Link Data** |
|---|---|---|
| Type 1: point-to-point connection to another network | Address of neighbor router | Interface number |
| Type 2: connection to any-to-any network | Address of designed router | Router address |
| Type 3: connection to stub network | Network address | Network mask |
| Type 4: virtual link | Address of neighbor router | Router address |

## *Example 5*

Give the router link LSA sent by router 10.24.7.9 in Figure 13.41.

# Example 5

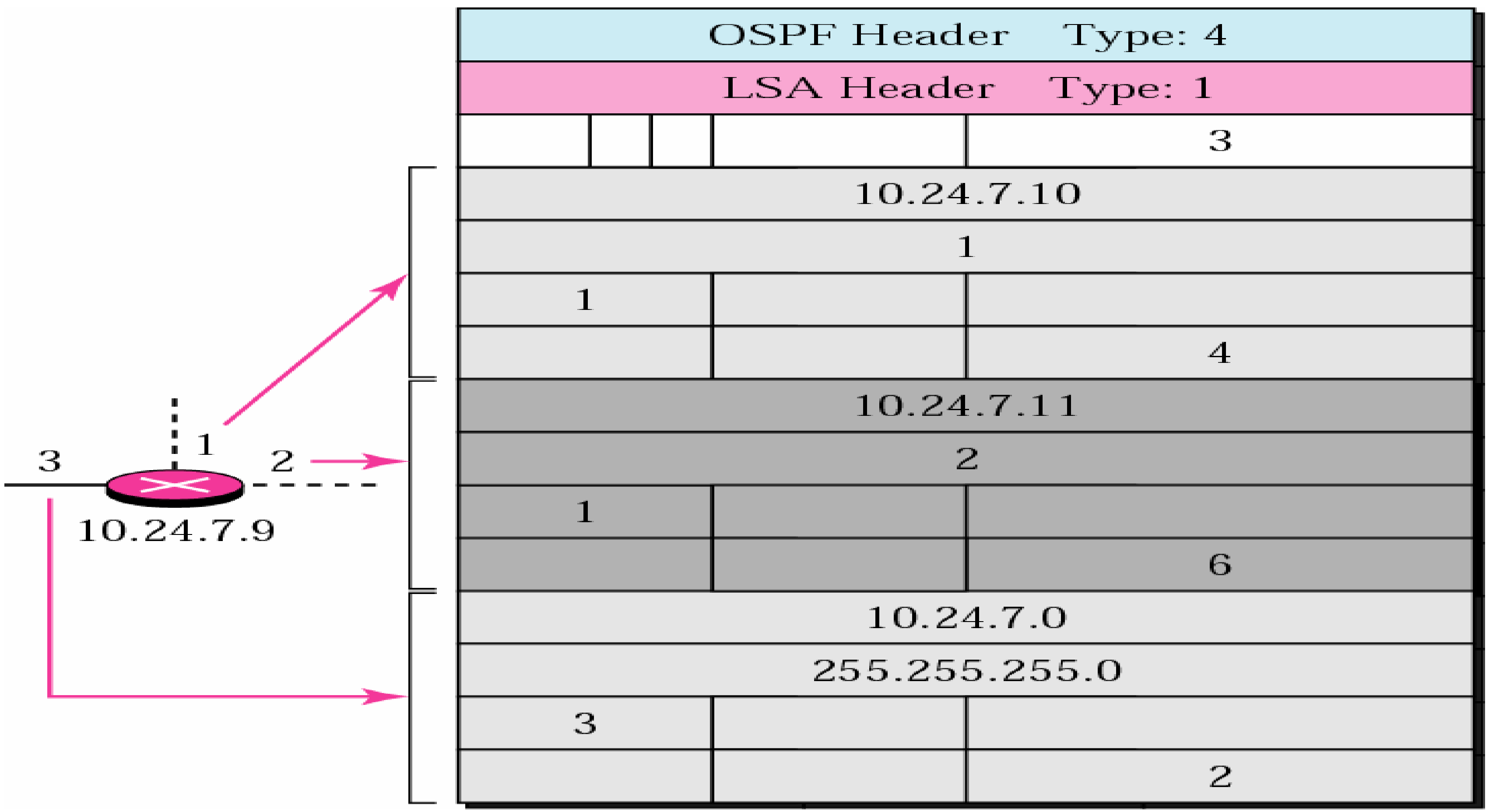

10.24.7.10
metric: 4
point-to-point
1
2
3
point-to-point
10.24.7.11
metric: 2
metric: 6
10.24.7.9
10.24.7.0/24

## *Solution*

- This router has three links
  - Two of type 1 (point-to-point)
  - One of type 3 (stub network)

- Figure 13.42 shows the router link LSA

# Solution to Example 5

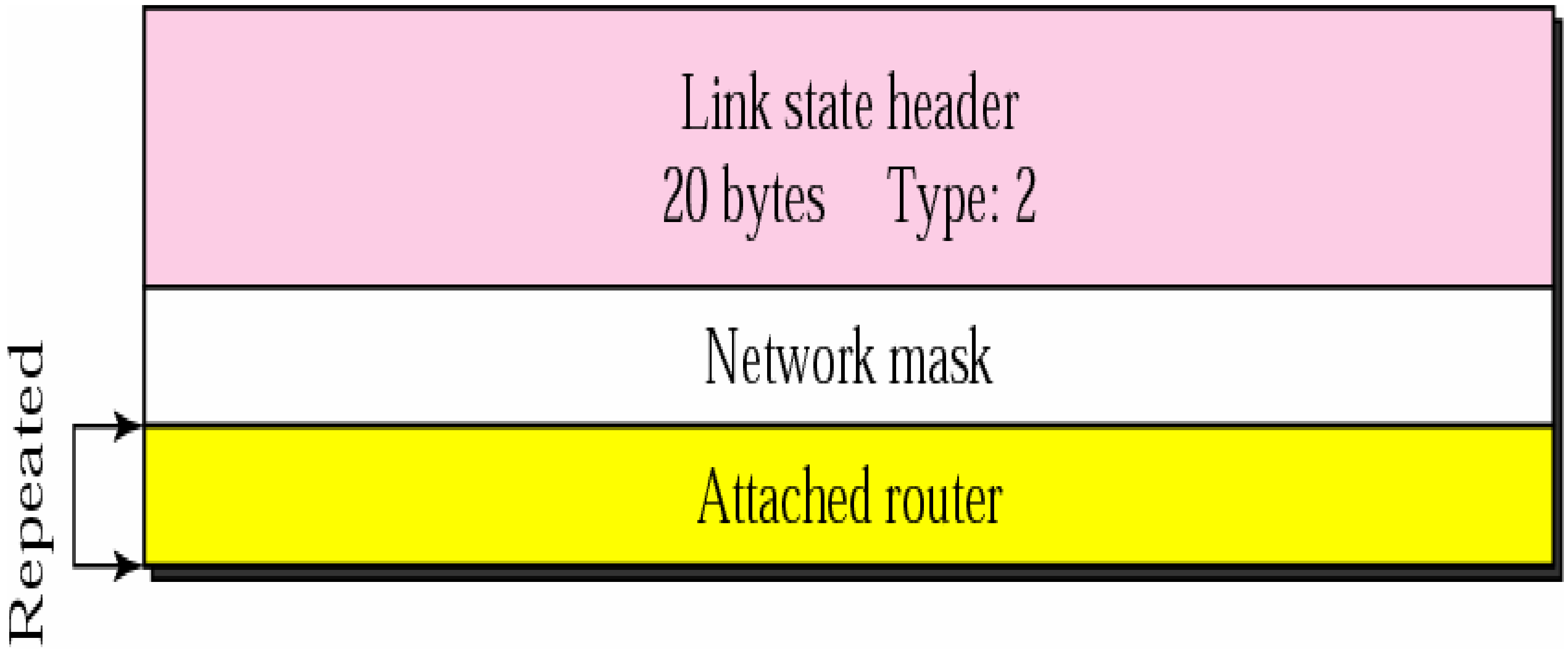


# Network Link LSA

- Announce the links connected to a network
- Format
  - Network mask
    - Define the network mask
  - Attached router
    - Define the IP addresses of all attached routers

Figure 13.42

# Network Link Advertisement Format

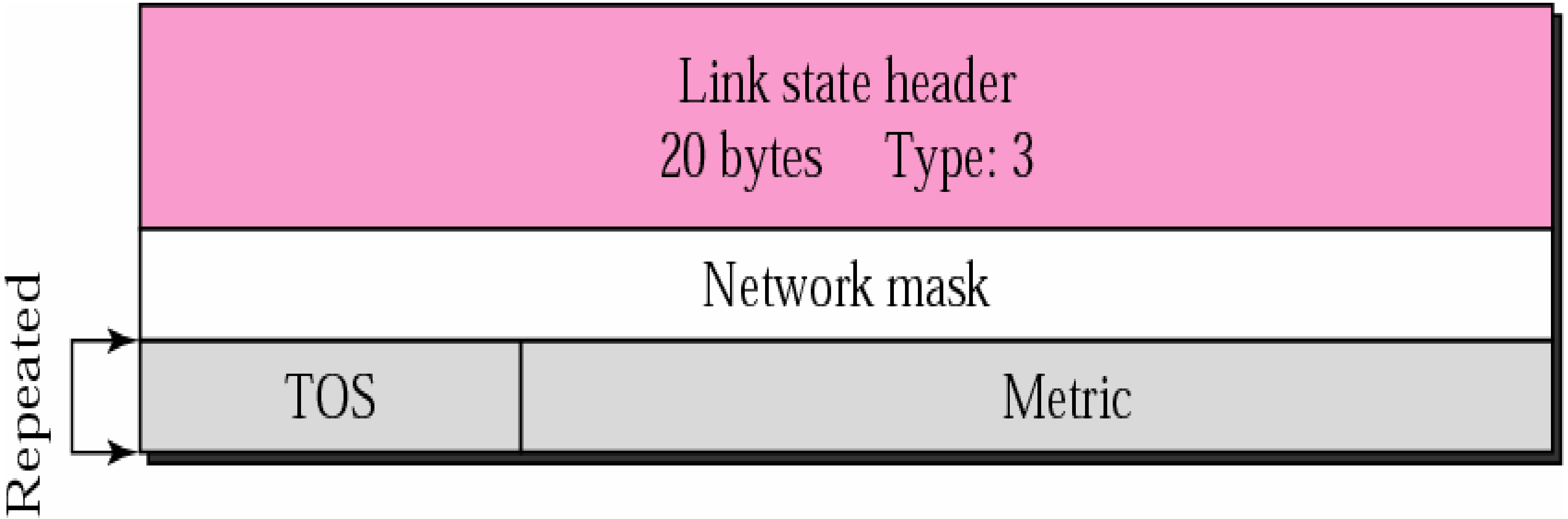


# Summary Link to Network LSA

- Used by the area router to announce the existence of other networks outside the area
- 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

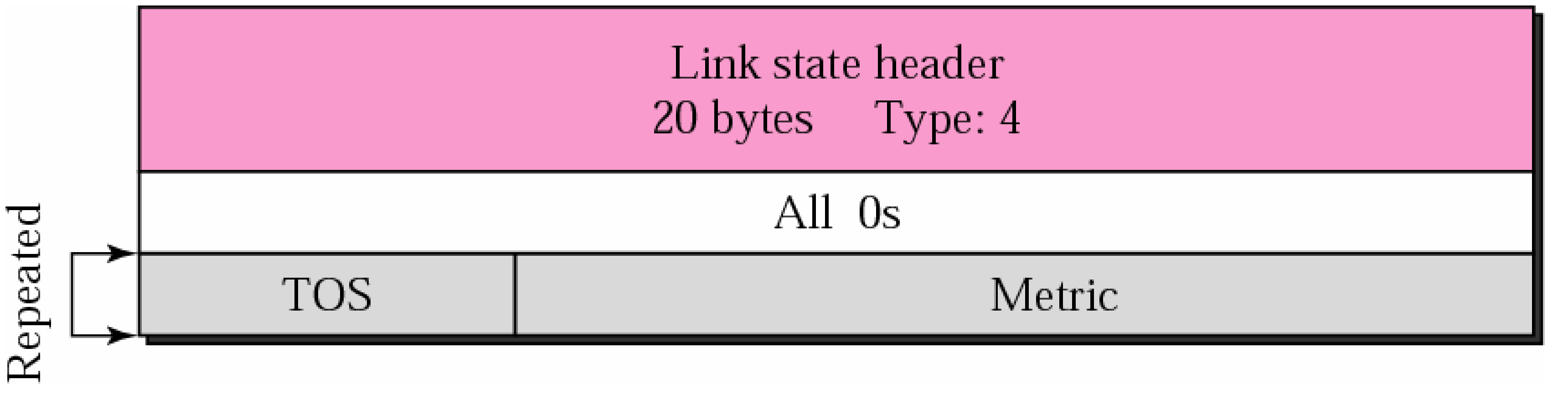


# Summary Link to AS Boundary Router LSA

- Announce the route to an AS boundary router
  - Define the network to which the AS boundary router is attached
- Format
  - The same as the summary link to network LSA

Figure 13.47

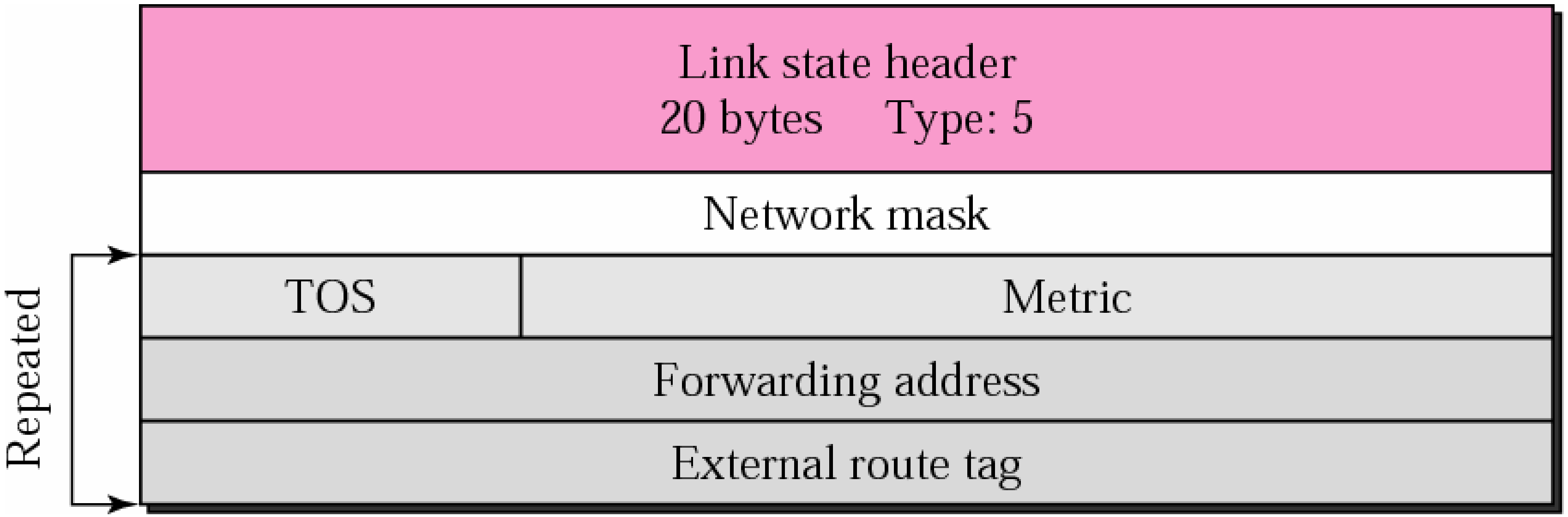
# Summary Link to AS Boundary LSA



# External Link LSA

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

# External Link LSA

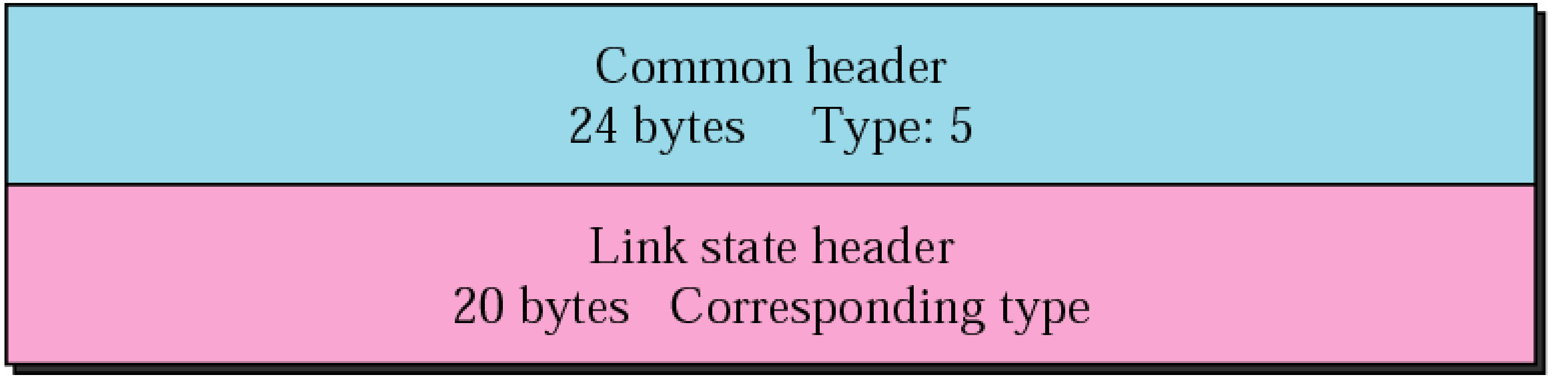


# Link State Acknowledgment Packet

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

- Format
  - Common header
  - Link state header

# 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

Note

***OSPF packets are encapsulated in IP datagrams.***

# BGP:
# Border Gateway Protocol

# BGP

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

# BGP (Cont.)

- Why the *distance vector routing* and *link state routing* are not good candidates for inter-autonomous system routing?

- **Distance vector routing**
  - There are occasions in which the route with the smallest hop count is not the preferred route
    - For example, we may not want a packet to pass through an AS that is not secure even it is the shortest path
  - Unstable as discussed before

# BGP (Cont.)

- **Link state routing**
  - An internet is usually too big for this routing method
  - If used, each router must have a huge link state database
    - It would take a long time for each router to calculate its routing table by Dijkstra algorithm

# Path Vector Routing

- Each entry in the routing table contains
  - The *destination network*
  - The *next router*
  - The *path* to reach the destination, usually defined *as ordered list of ASs*

# Example of a Path Vector Routing Table

| **Network** | **Next Router** | **Path** |
|---|---|---|
| NO1 | R01 | AS14, AS23, AS67 |
| NO2 | R05 | AS22, AS67, AS05, AS89 |
| NO3 | RO6 | AS67, AS89, AS09, AS34 |
| NO4 | R12 | AS62, AS02, AS09 |

# Path Vector Message

- *Autonomous boundary routers* advertise the reachability of networks in their own AS to neighbor autonomous boundary routers

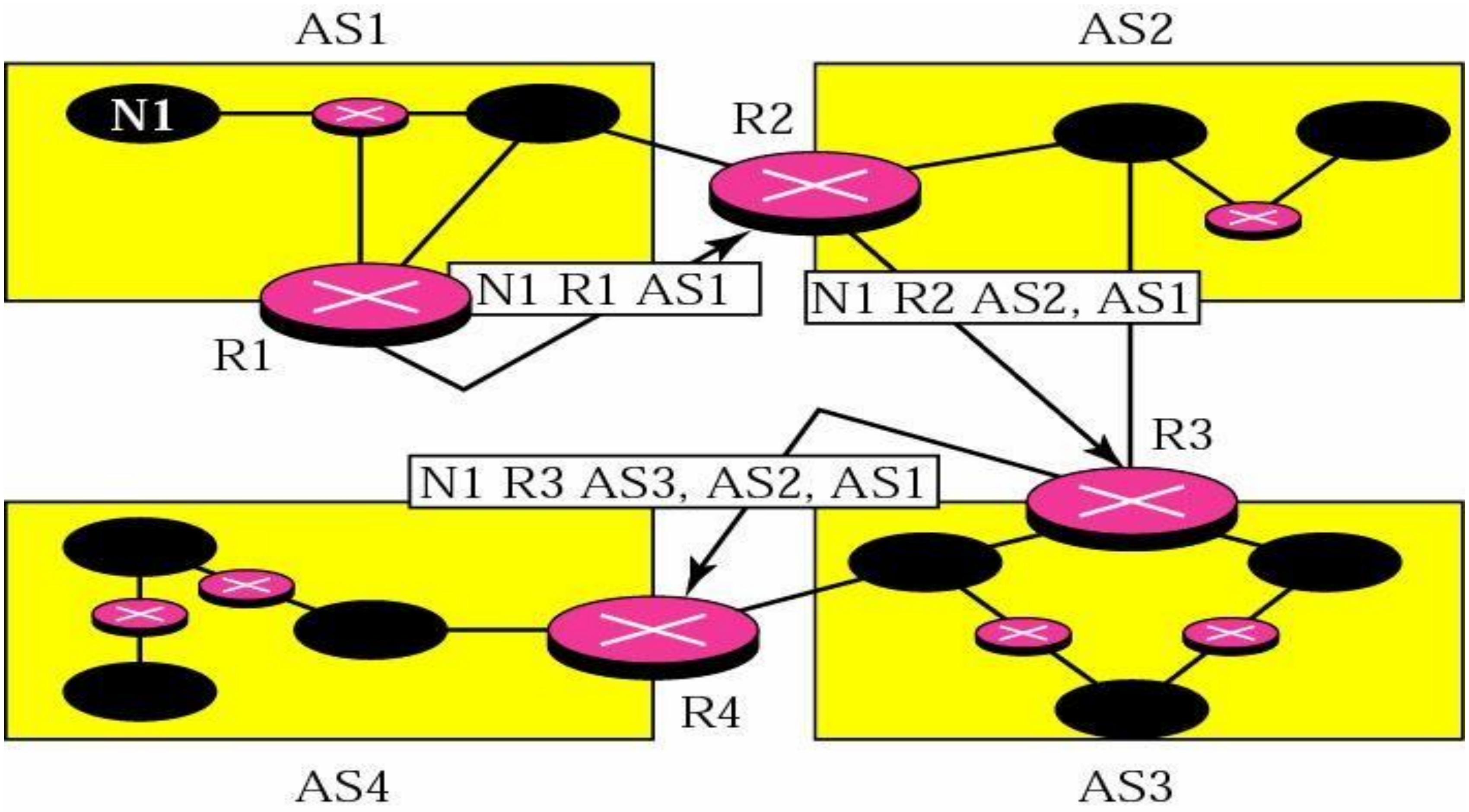
- Each router receives a path vector message
  - Verify the advertised path is in agreement with its policy
  - If yes, update its routing table and modify the message before sending it to the next router
    - Add its AS number to the path and replacing the next router entry with its own identification

# Example

- R1 send a path vector message
  - Advertise the reachability of N1

- R2 receives the message
  - Update it routing table
  - Send the message to R3 with *adding its AS to the path* and *inserting itself as the next router*

# Path Vector Packets



# Loop Prevention

- Path vector routing can solve
  - The instability of distance vector routing
  - The creation of loops
    - When a router receive a message, it check to see if its AS is in the path list or not
      - If yes, looping is involved and the message is ignored

# Policy Routing

- *Policy routing* can be easily implemented through path vector routing
  - 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

# 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

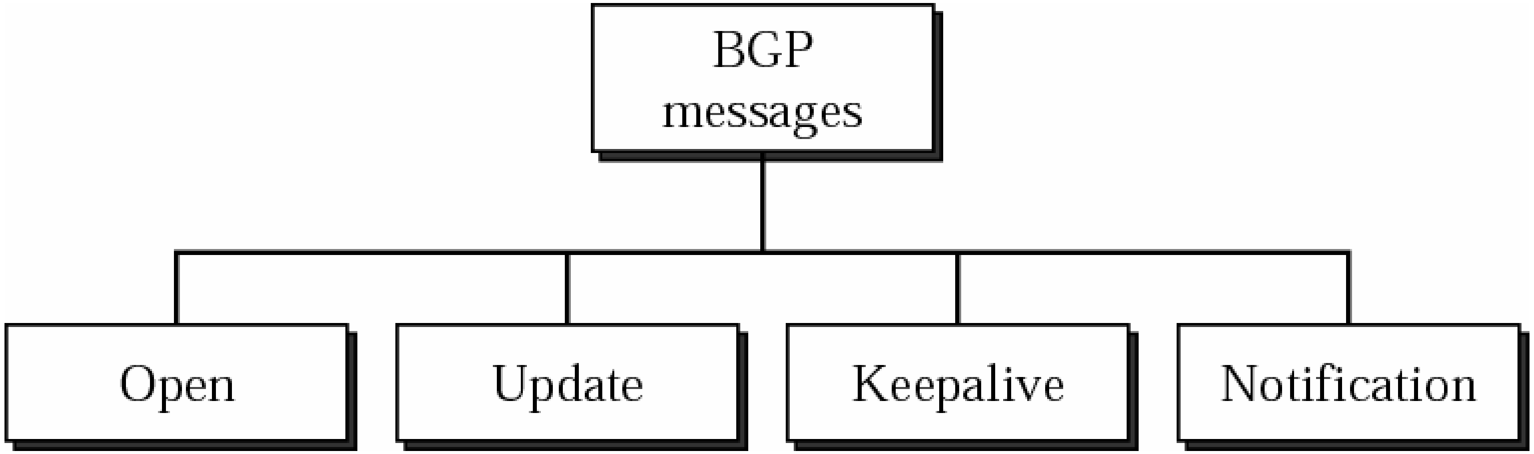
# Path Attributes (Cont.)

- **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

# Types of Packets

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

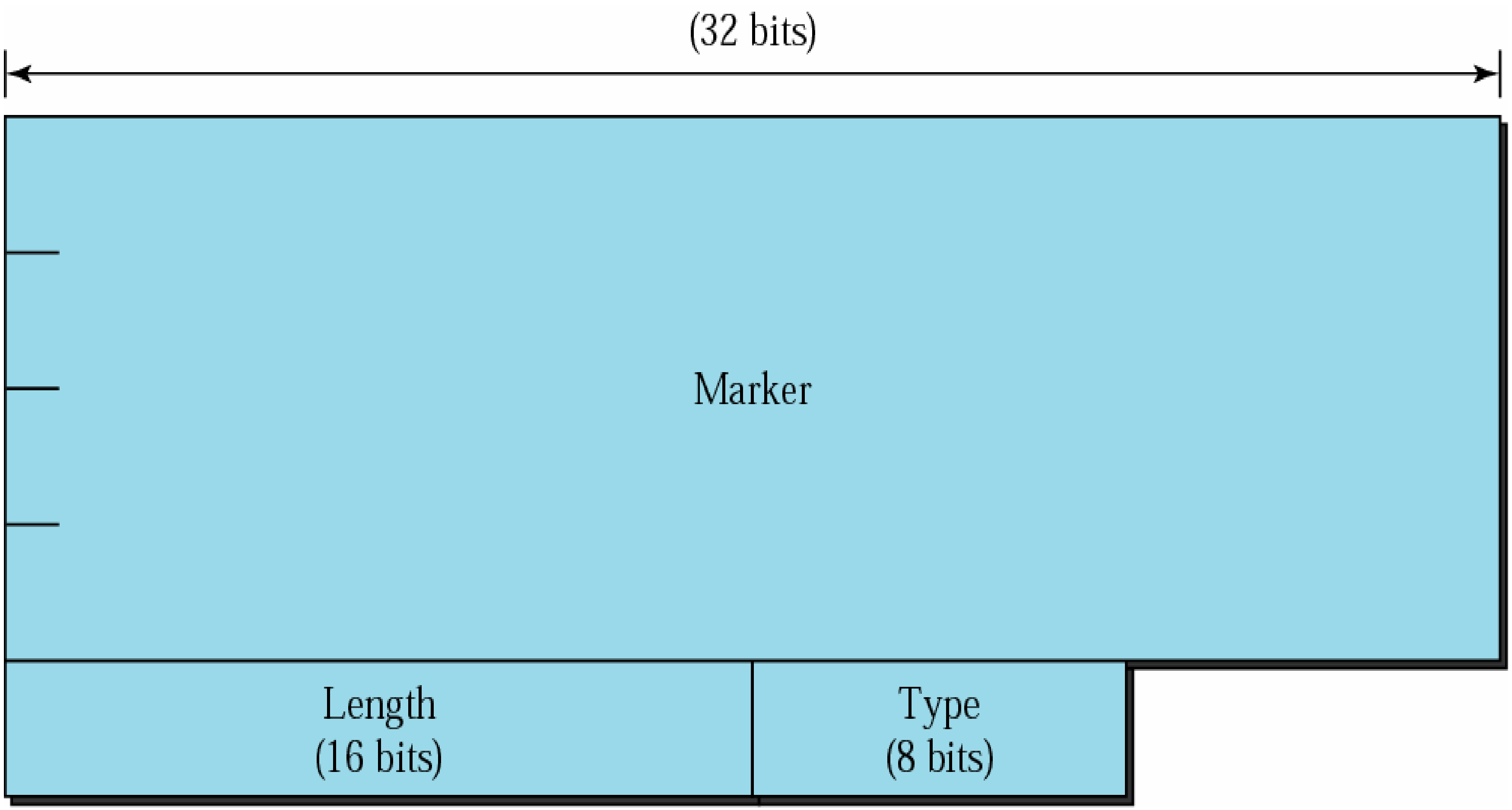
# Types of BGP Messages



# 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

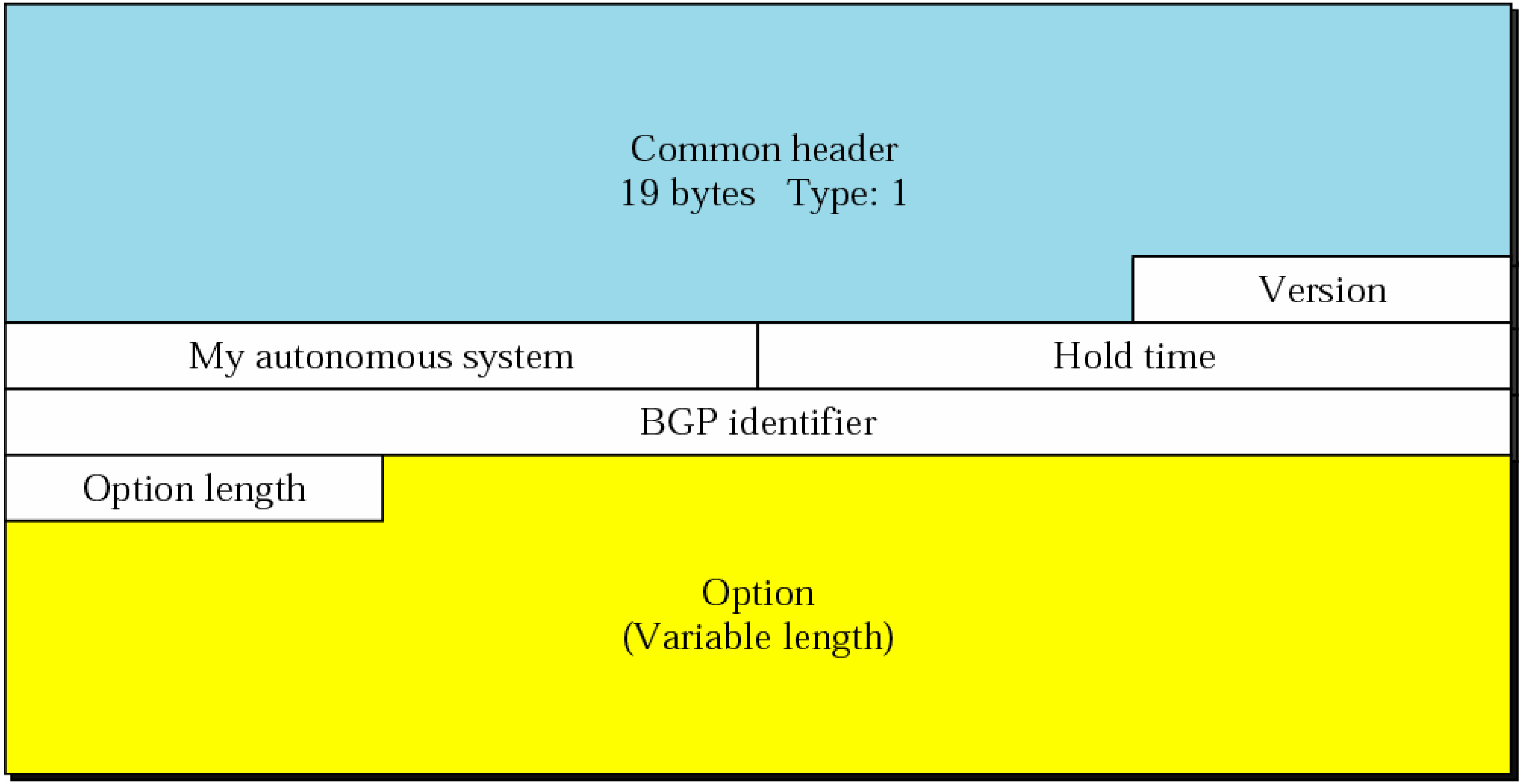
# BGP Packet Header



# 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

# 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*

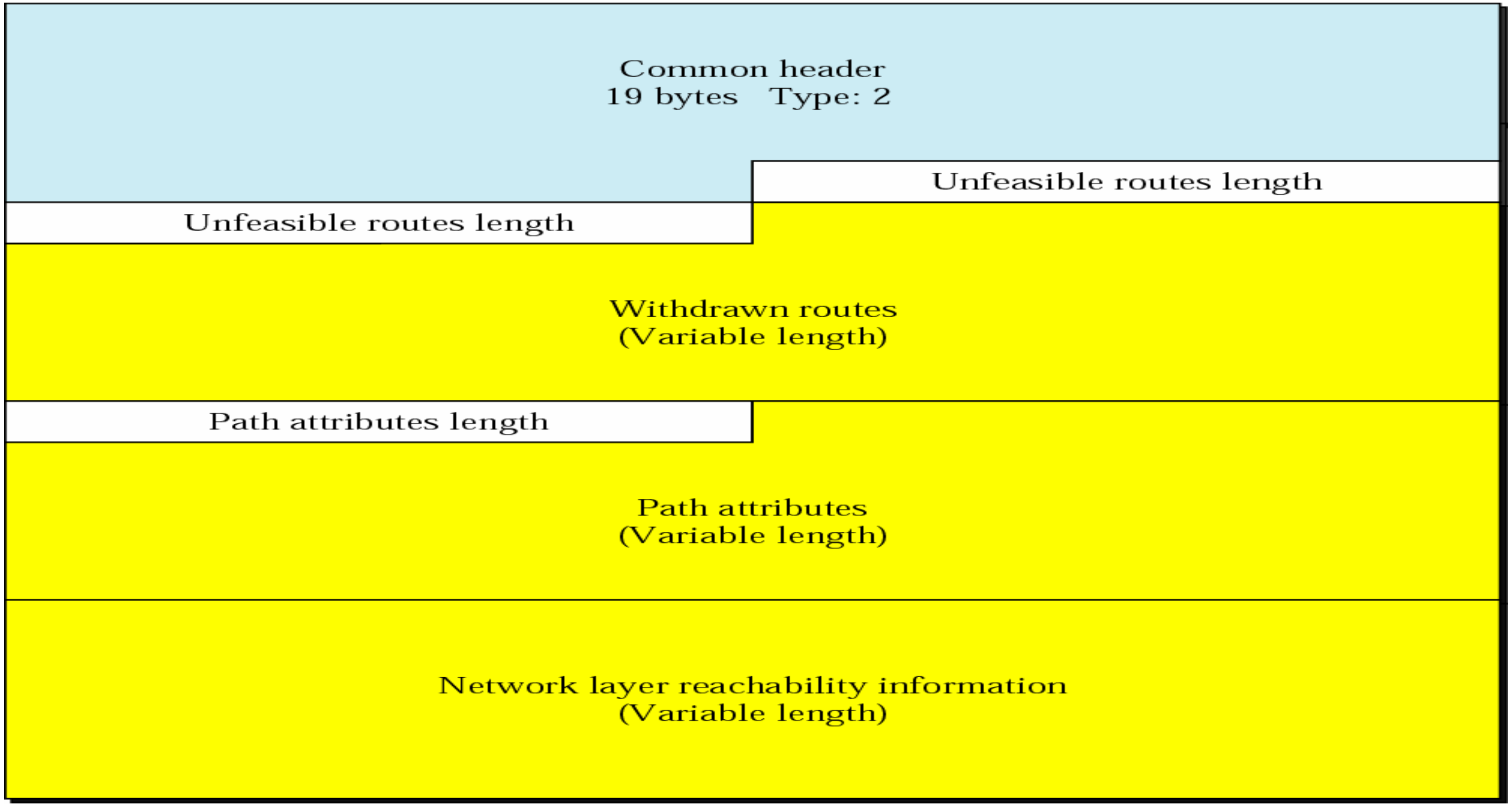
# Update Message

- 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

# 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:
    - Thus, BGP4 supports *classless addressing* and *CIDR*

Note

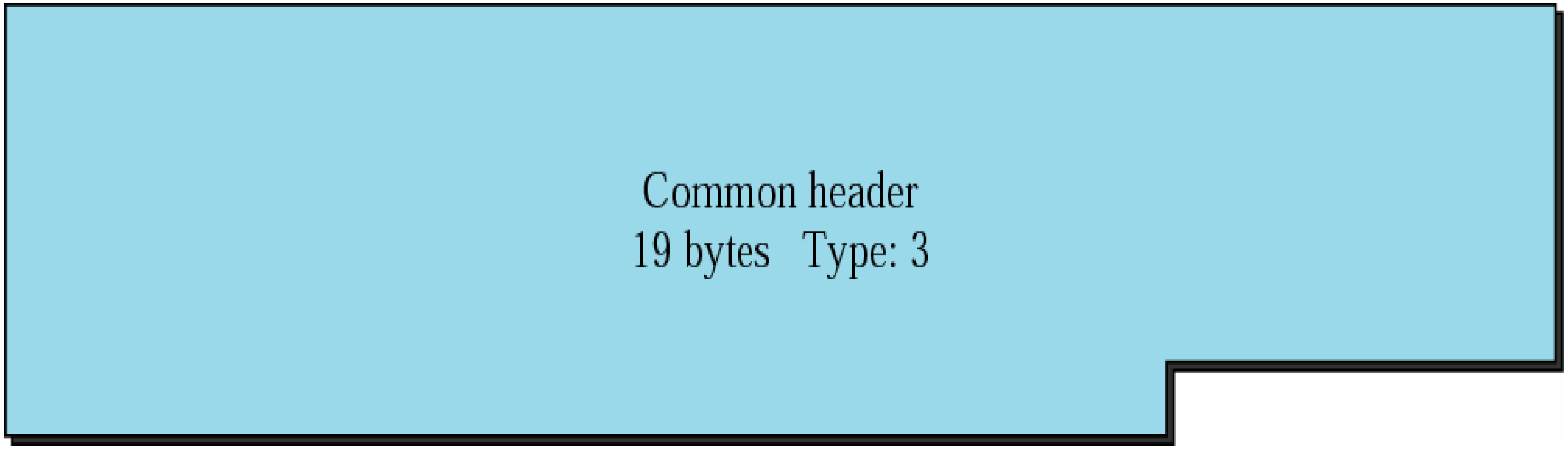
***BGP supports classless addressing and CIDR.***

# Keepalive Message

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

Figure 13.55

# Keepalive Message

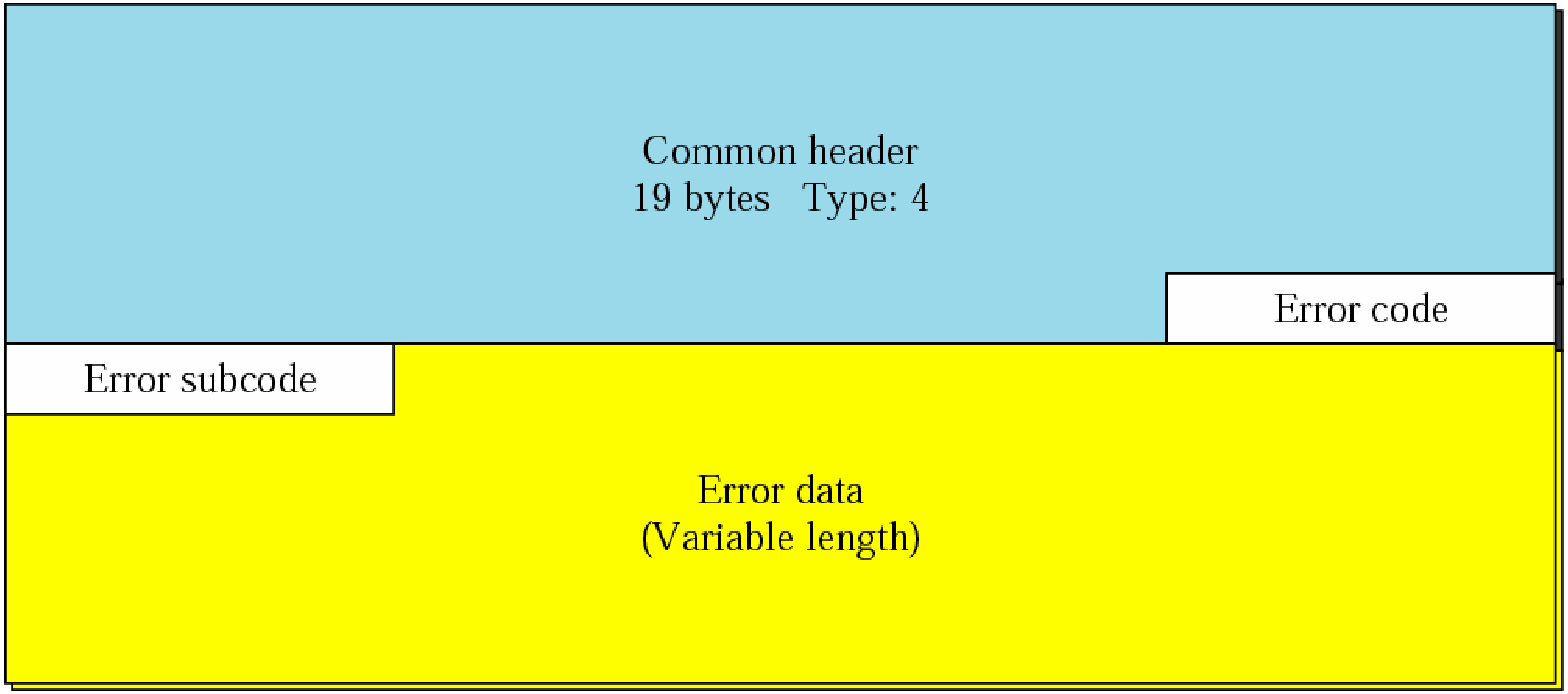


# Notification Message

- 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



# 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

Note

***BGP uses the services of TCP on port 179.***