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***



## 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.***

**Figure 14.1** ***Autonomous systems***

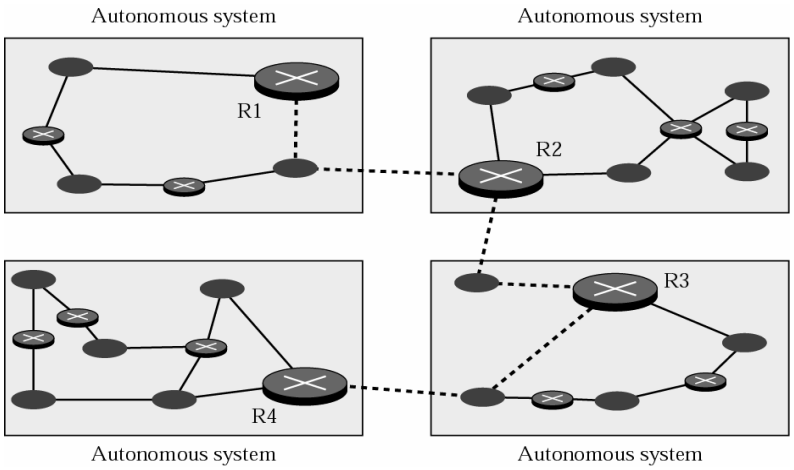
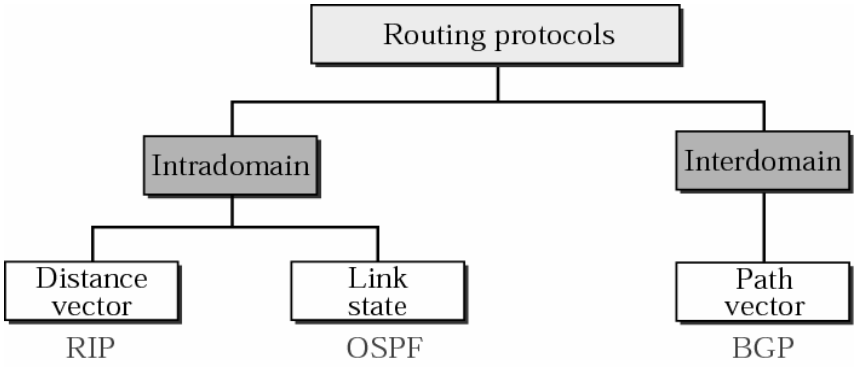





**Figure 14.2** ***Popular routing protocols***



## 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

- Router initialize routing table to contain an entry for each directly connected network
- One router sends list of its routes to another
- List contains pairs of destination network and distance

| Destination | Distance | Route |
|---|---|---|
| Net 1 | 0 | Direct |
| Net 2 | 0 | Direct |

# Distance vector routing (cont)

- When to update?
  - Shorter distance
  - New node joins
  - Topology change

Router K  Router J  Net 24

**(Router K→Net24) = (Router K→ Router J) + (Router J→Net 24)**

**N+1 = 1 + N**



# Routing table update

| Destination | Distance | Route (next hop) |
|---|---|---|
| Net 1 | 0 | direct |
| Net 2 | 0 | direct |
| Net 4 | 8 | Router L |
| Net 17 | 5 | Router M |
| Net 24 | 6 | Router J |
| Net 30 | 2 | Router Q |
| Net 42 | 2 | Router J |

existing routing table for a router K

| Destination | Distance | |
|---|---|---|
| Net 1 | 2 | |
| Net 4 | 3 | Shorter distance |
| Net 17 | 6 | |
| Net 21 | 4 | New node joins |
| Net 24 | 5 | |
| Net 30 | 10 | |
| Net 42 | 3 | Topology change |

incoming update message from router J
(marked items cause change)

**Figure 14.3** *Distance vector routing tables*

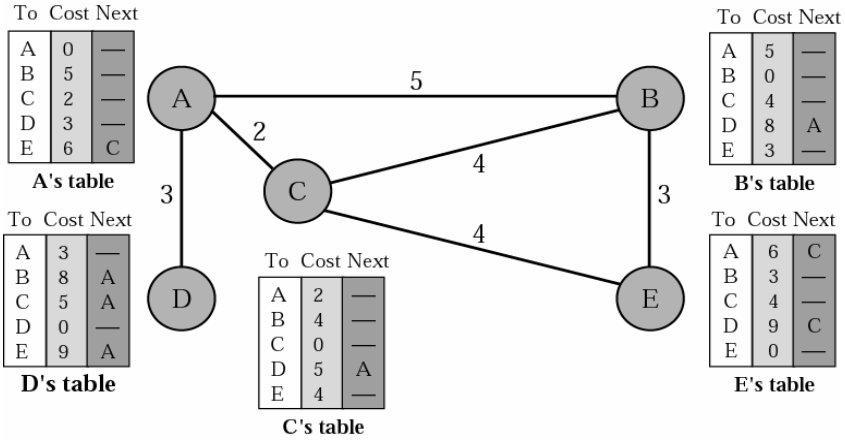
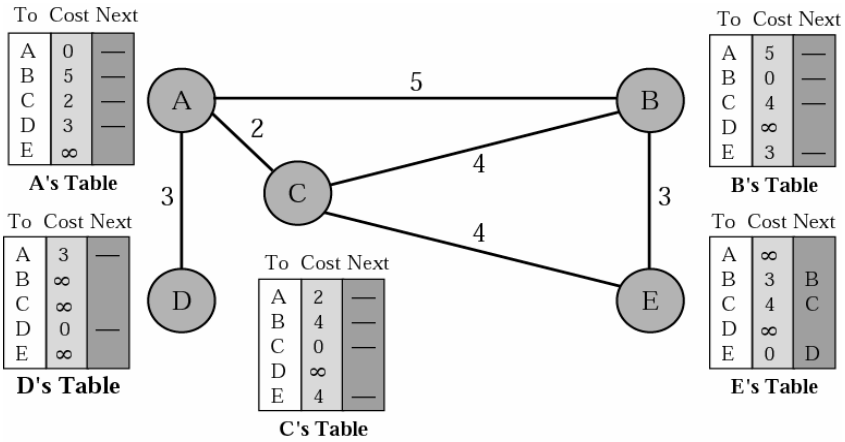
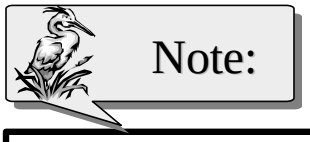





**Figure 14.4** *Initialization of tables in distance vector routing*





Note:

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



**Figure 14.5** *Updating in distance vector routing*

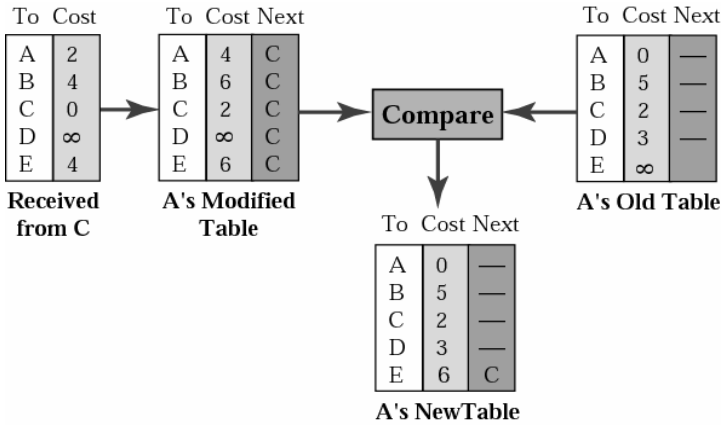
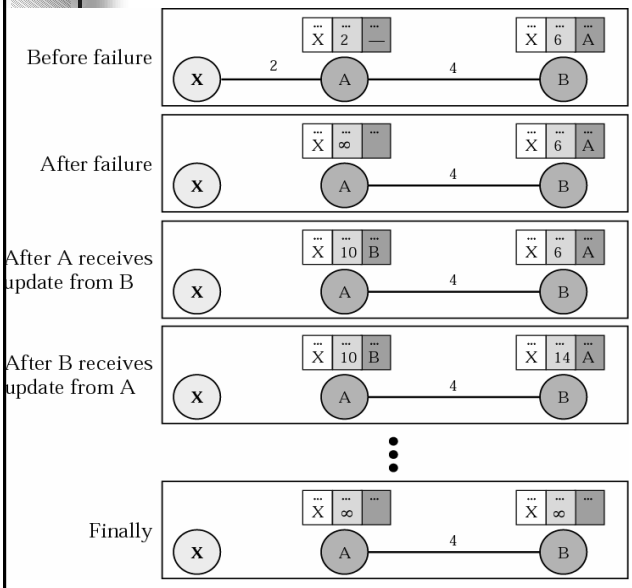


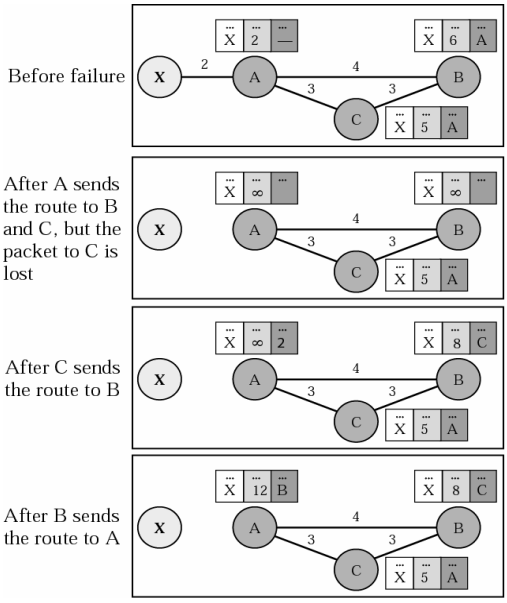
**Figure 14.6** *Two-node instability*




Solution for two-node loop problem
- Defining infinity: define 16 as infinity
- Split horizon: node B thinks that the optimum route to reach X is via A, it does not need to advertise this piece of information to A
- Split horizon and poison reverse: B still advertise the value for X to A, but it replace the distance with infinity



**Figure 14.7** *Three-node instability*



# 14.3 RIP

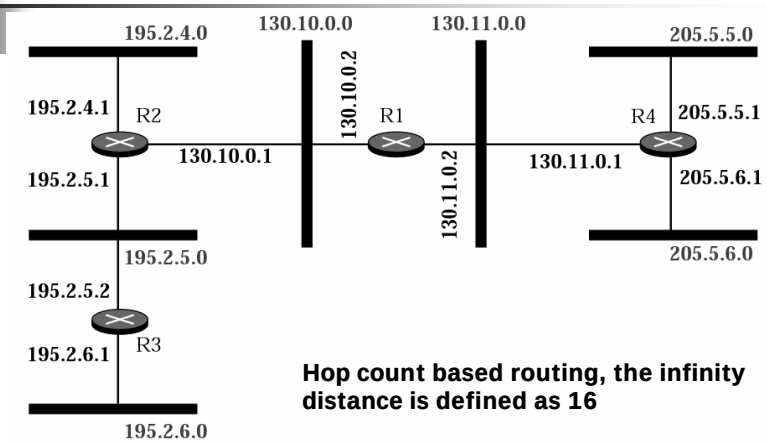
***The Routing Information Protocol (RIP) is an intradomain routing protocol used inside an autonomous system. It is a very simple protocol based on distance vector routing.***

***The topics discussed in this section include:***

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



**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 |

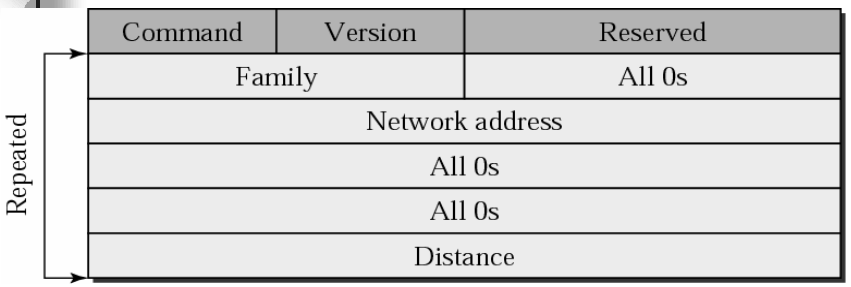
R3 Table

| 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 | ——— |

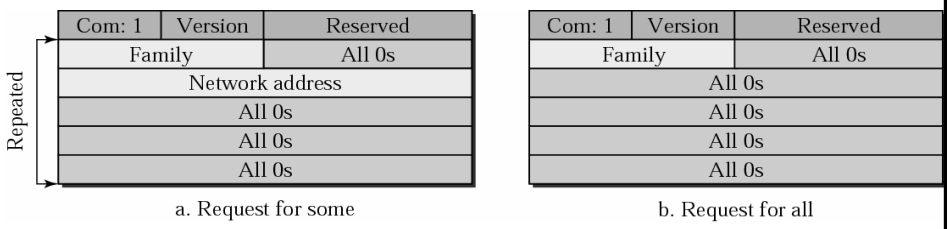
**Figure 14.9** ***RIP message format***




- Command: specifies the type of message: request (1) or response (2)
- Version: 1 or 2
- Family: TCP/IP family (2)
- Network address: destination network address
- Distance: the hop count from the advertising router to the destination network



**Figure 14.10** ***Request messages***




a. Request for some

b. Request for all

- A request message is sent by a router that has just come up or by a router that has some time-out entries
- A request can ask about specific entries or all entries

## Example 1

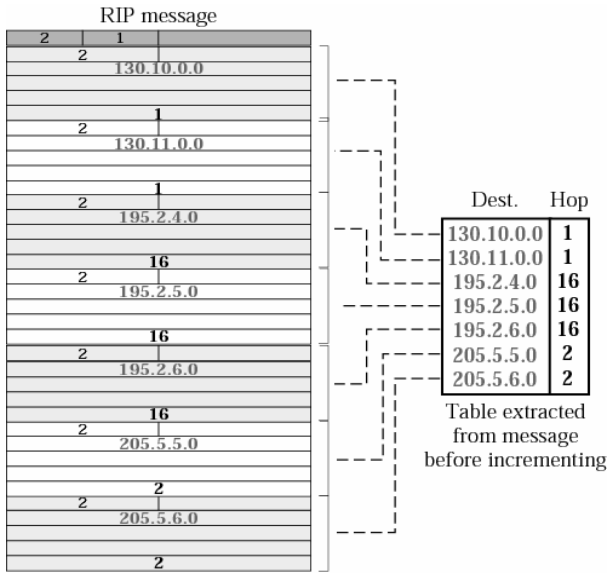
***Figure 14.11 shows the update message sent from router R1 to router R2 in Figure 14.8. The message is sent out of interface 130.10.0.2.***

***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, it replaces 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.***

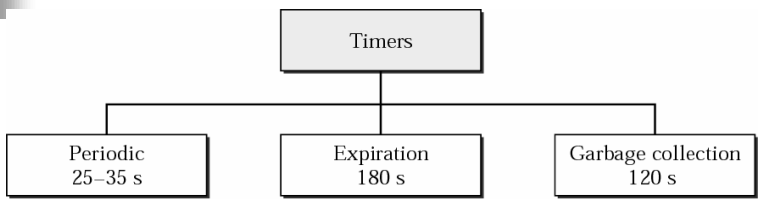
**See Next Slide**



**Figure 14.11** *Solution to Example 1*



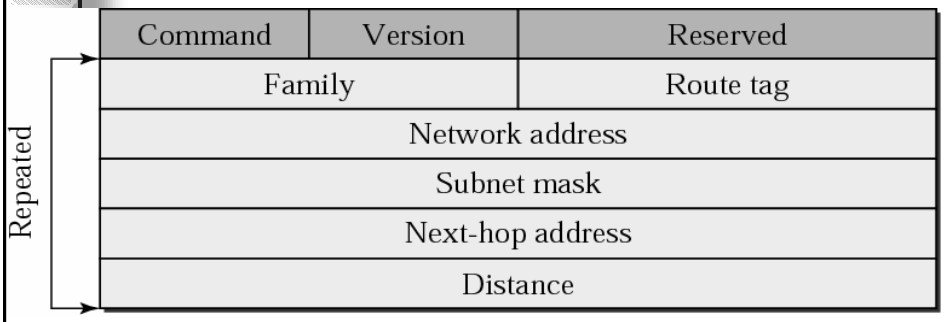
## Timer in *RIP*




- RIP uses three timers to support its operation
  - Periodic timer
    - Advertising of regular update messages
  - Expiration timer
    - Govern the validity of a route. When the time is timeout, the hop count of the route is set to 16
  - Garbage collection timer
    - The invalid route does not immediately purge from the table until the garbage collection timer is timeout.



**Figure 14.13** ***RIP version 2 format***




- Route tag: the autonomous number
  - Can receive interdomain routing information
- Subnet mask: can be used to classless addressing
- Next-hop address: used in the multi-autonomous systems share a same backbone network

Note:

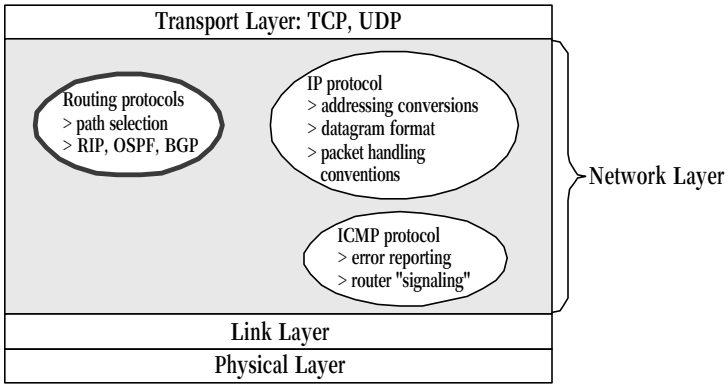
***RIP uses the services of UDP on well-known port 520.***

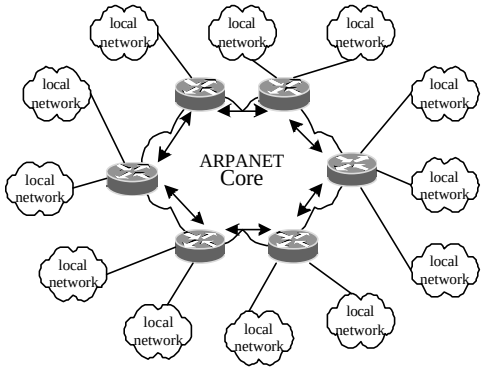
***RIPv1 uses broadcasting to send RIP messages to every neighbor***

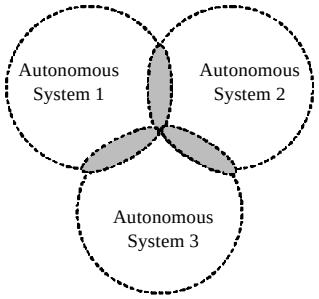
***RIPv2 uses multicast address to send the RIP messages only to RIP routers in the network***



# Network Layer





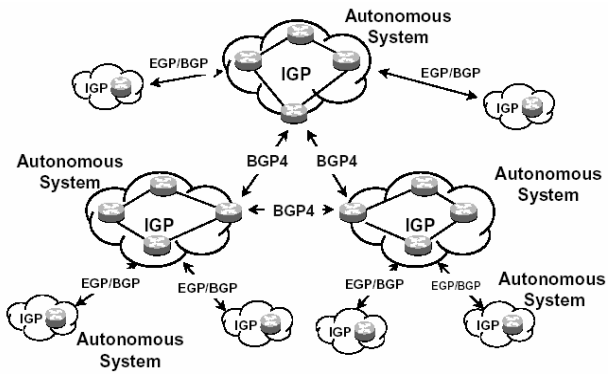




Border areas where EGP routng protocol is running

Core Routers

GGP

TCP/IP Protocol Suite 25




TCP/IP Protocol Suite 26

From: Applied Network Research Group, Department of Computer Engineering, Kasetsart University

# 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*

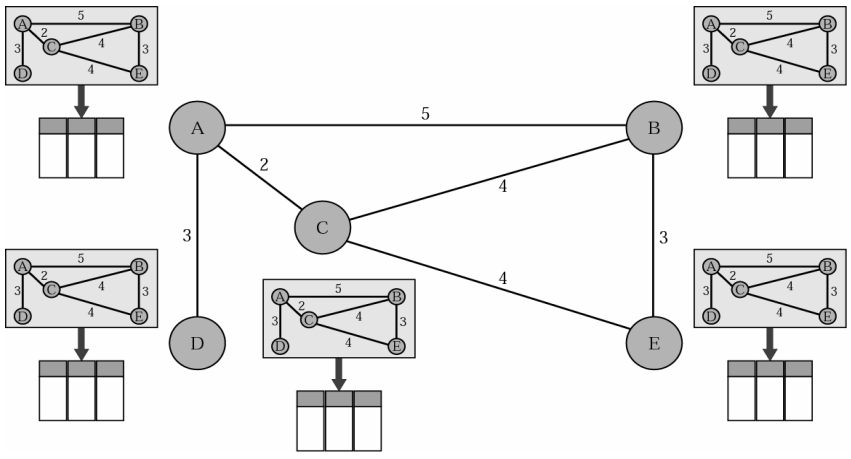
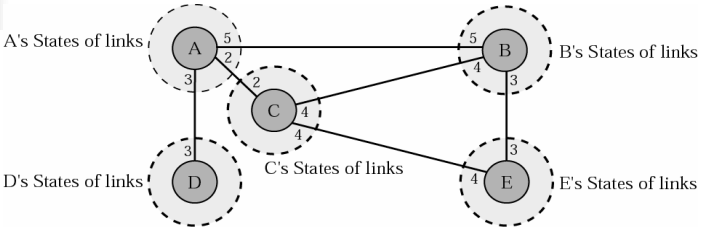


**Figure 14.16** *Link state knowledge*




- Each node knows the state (type, condition, and cost) of its links
- The node floods the state of its links to other nodes
- The whole network topology can be compiled from the partial knowledge of each node



# Building routing tables

- Four sets of actions are required in link state routing
  - Creating of the states of the kinks by each node, called the link state packet or LSP
    - When there is a change in the topology of the domain
    - On a periodic basis (about 60 minutes or 2 hours)
  - Dissemination of LSPs to every other router, called flooding
    - A node that receives an LSP compares it with the copy it may already have.
    - If the newly arrived LSP is older than the one it has, it discards the LSP.
    - If it is newer, the node does
      - Discard the old LSP and keep the new one
      - It sends a copy of it out of each interface except the one from which the packet arrived
  - Formation of a shortest path tree for each node
    - Dijkstra algorithm
  - Calculation of a routing table based on the shortest path tree

**Figure 14.17** ***Dijkstra algorithm***

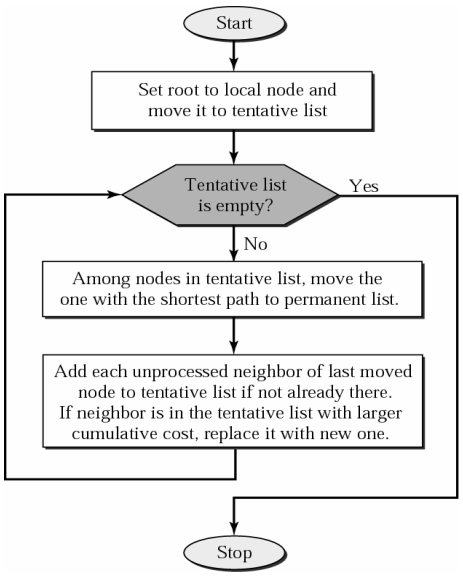
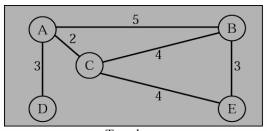


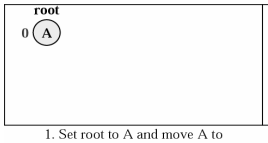



**Figure 14.18** ***Example of formation of shortest path tree***

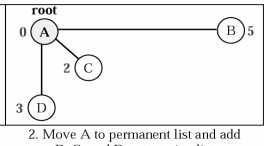



Topology

Permanent list: empty
Tentative list: A(0)



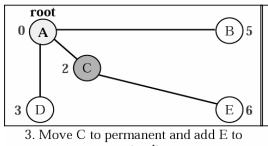

1. Set root to A and move A to tentative list



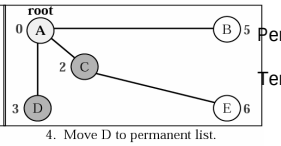

2. Move A to permanent list and add B, C, and D to tentative list

Permanent list: A(0)
Tentative list:
B(5), C(2), D(3)

Permanent list: A(0), C(2)
Tentative list:
B(5), D(3), E(6),



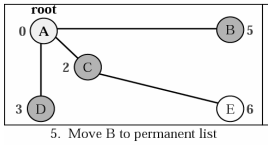

3. Move C to permanent and add E to tentative list



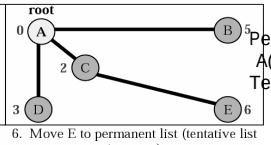

4. Move D to permanent list.

Permanent list:
A(0), C(2), D(3)
Tentative list:
B(5), E(6),

Permanent list:
A(0), C(2), D(3), B(5)
Tentative list:
E(6)




5. Move B to permanent list




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

Permanent list:
A(0), C(2), D(3), B(5), E(6)
Tentative list: empty

# Calculation of routing table from shortest path tree

***Table 14.1 Routing table for node A***

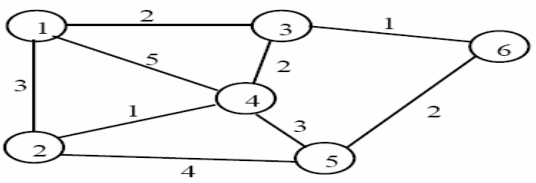
| *Node* | *Cost* | *Next Router* |
|---|---|---|
| A | 0 | — |
| B | 5 | — |
| C | 2 | — |
| D | 3 | — |
| E | 6 | C |



# Dijkstra algorithm

- Initialization :
  - N={s} ; $D_j = C_{sj}, \forall j \neq s$ ; $D_s = 0$
- Finding the next closest node:
  - $D_i = \min_{j \neq N} D_j$.
  - Add i to N , if N contains all the node, stop
- Updating minimum costs:
  - $D_i = \min_{j \neq N}\{D_j, D_i + C_{ij}\}$ ; Go to step 2

# Dijkstra algorithm example

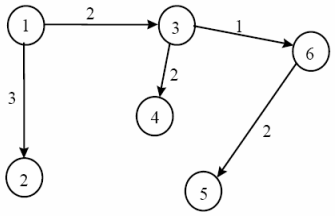



| iteration | N | $D_2$ | $D_3$ | $D_4$ | $D_5$ | $D_6$ |
|---|---|---|---|---|---|---|
| initial | {1} | 3 | 2 | 5 | ∞ | ∞ |
| 1 | {1,3} | 3 | 2 | 4 | ∞ | 3 |
| 2 | {1,2,3} | 3 | 2 | 4 | 7 | 3 |
| 3 | {1,2,3,6} | 3 | 2 | 4 | 5 | 3 |
| 4 | {1,2,3,4,6} | 3 | 2 | 4 | 5 | 3 |
| 5 | {1,2,3,4,5,6} | 3 | 2 | 4 | 5 | 3 |



# Dijkstra algorithm

Shortest-path tree from node 1 to other nodes




Routing table

| Destination | Next node | Cost |
|---|---|---|
| 2 | 2 | 3 |
| 3 | 3 | 2 |
| 4 | 3 | 4 |
| 5 | 3 | 5 |
| 6 | 3 | 3 |

# 14.5 OSPF

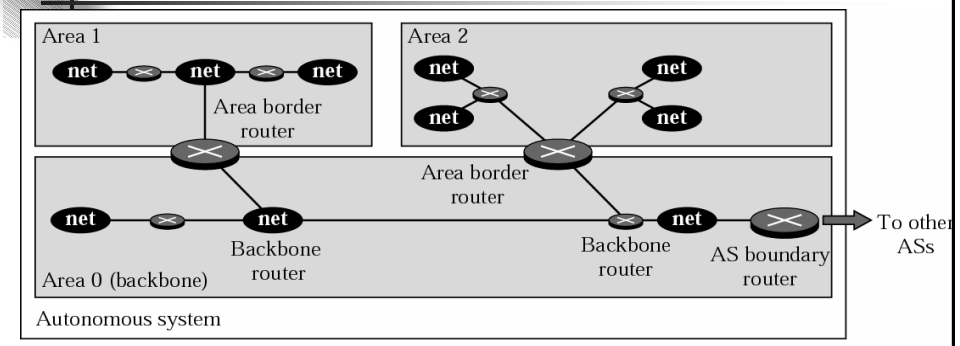
***The Open Shortest Path First (OSPF) protocol is an intradomain routing protocol based on link state routing. Its domain is also an autonomous system.***

***The topics discussed in this section include:***

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



**Figure 14.19** *Areas in an autonomous system*




- Area: OSPF divides an autonomous system into areas
  - An area is a collection of networks, hosts and routers
  - Routers inside an area flood the area with routing information
- Area border routers: summarize the information about the area and send it to other areas
- Backbone: all of the areas inside an autonomous system must be connected to the backbone
  - Backbone area ID is zero

# OSPF Multi-Area

- Router Type in Multi-Area
  - Internal Router
  - Backbone Router
  - Area Border Router, ABR
  - Autonomous System Border Router, ASBR

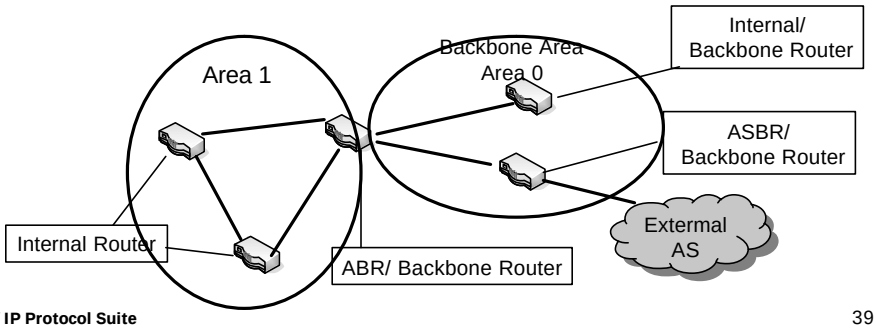





**Figure 14.20** ***Types of links***

- Metric: OSPF allows assign a cost, called the metric, to each route.
  - The metric can be based on a type of service (minimum delay, maximum throughput, and so on)
- Type of links in OSPF
  - Point-to-point: connect two routers without any other host or router in between
  - Transient
  - Stub
  - virtual

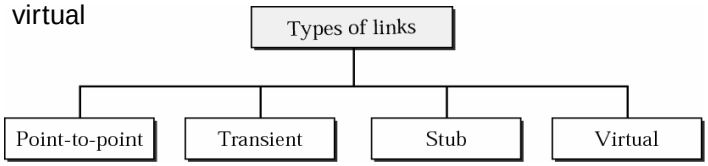


**Figure 14.21** *Point-to-point link*

- Point-to-point: connect two routers without any other host or router in between

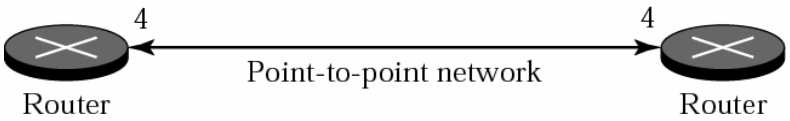
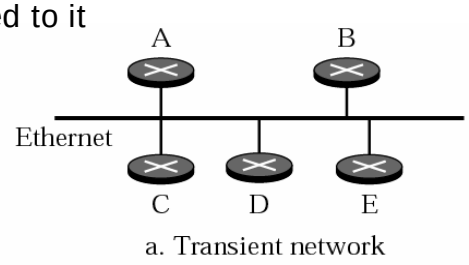


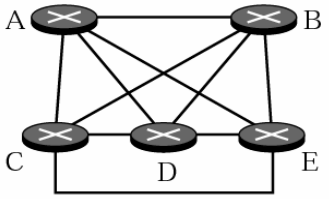



**Figure 14.22** *Transient link*

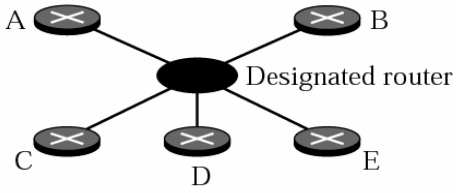
- Transient Link: is a network with several routers attached to it




a. Transient network




b. Unrealistic representation

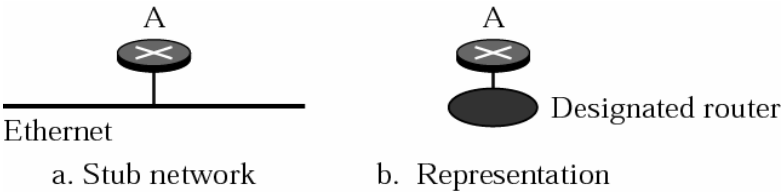



c. Realistic representation

**The network itself is represented by a node**

**Figure 14.23** *Stub link*

- Stub link: is a network that is connected to only one router
  - The data packets enter the network through this single router and leave the network through this same router




**Using the designated router for the network**

- Virtual link: when the link between two routers is broken, the administration may create a virtual link between them using a longer path that probably goes through several routers



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

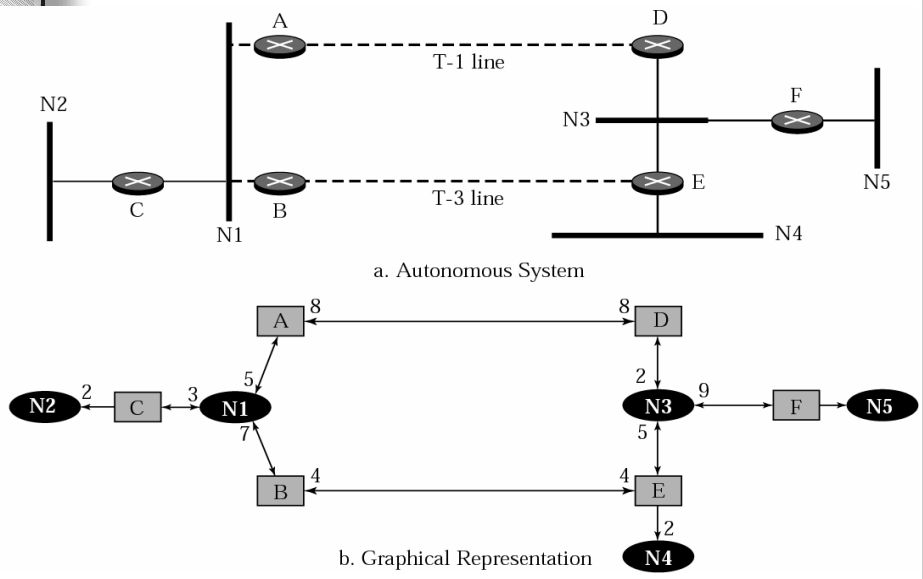
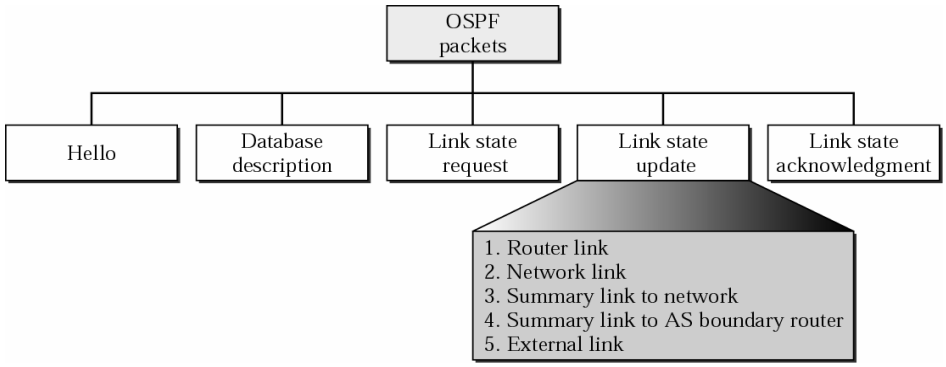


**Figure 14.25** *Types of OSPF packets*






# Encapsulation and Format

- Packet Encapsulation
  - Network Layer

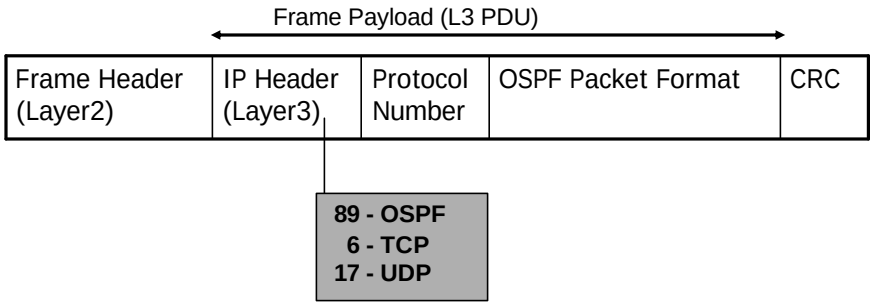
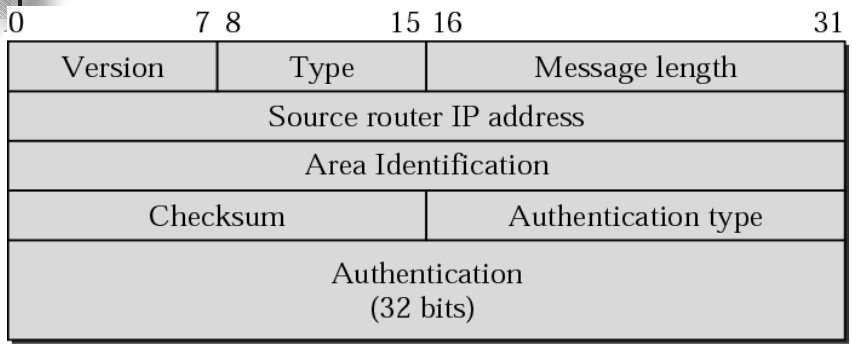


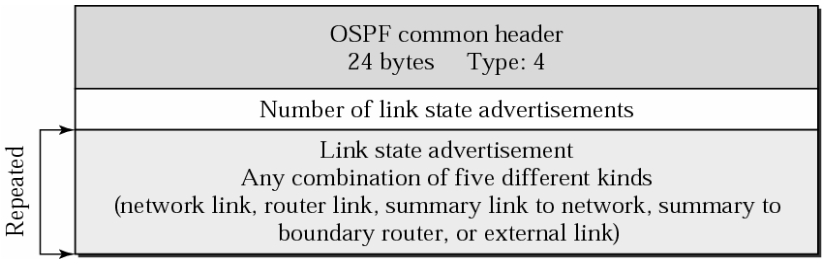
**Figure 14.26** *OSPF* ***common header***




- 24 bytes
- Version: currently version 2
- Type：[1] Hello, [2] DDPs, [3] LSR, [4] LSU, [5] LSAck
- Message length: total message length including the header



**Figure 14.27** ***Link state update packet***




- All five kinds links state update have the same general header

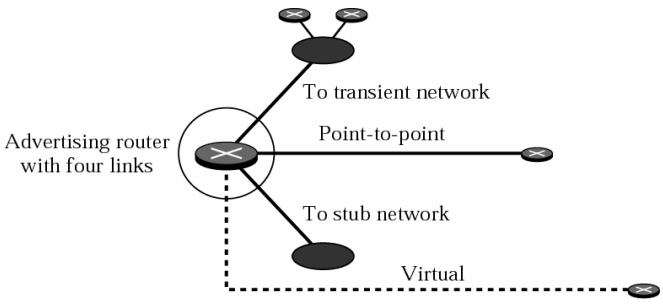
**Figure 14.28** *LSA general header*

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

- Link state age: the number of seconds elapsed since this message was first generated
- E flag: 1 is a stub area (only one path to the backbone area)
- T flag: 1 means the router can handle multiple types of service
- Link state type: defines the LSA type (1:router link, 2: network link, 3: summary link to network, 4: summary link to AS boundary router, 5: external link)
- Link state ID: the value of this field depends on the type of link
- Advertising router: IP address of the router advertising this message
- Link state sequence number:



**Figure 14.29** *Router link*




- Router link LSA
  - A router link defines the links of a true router
  - A true router uses this advertisement to announce information about all of its links and what is at the other side of the link (neightbors)

**Figure 14.30** ***Router link LSA***

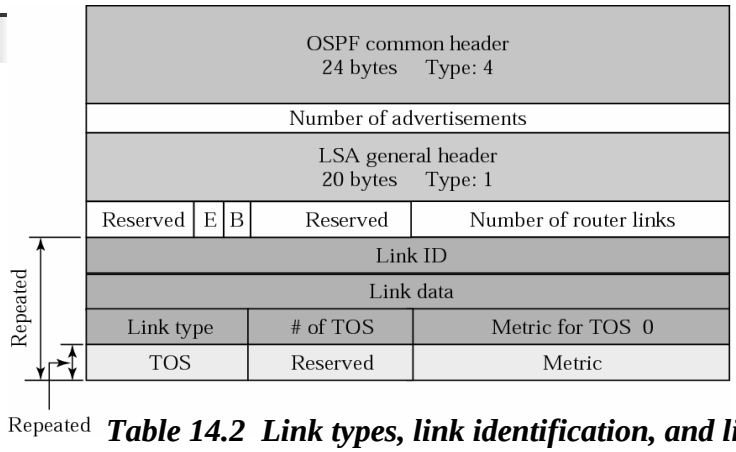



***Table 14.2 Link types, link identification, and link data***

| *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.***

**See Next Slide**

***Solution***

***This router has three links: two of type 1 (point-to-point) and one of type 3 (stub network). Figure 14.32 shows the router link LSA.***

**Figure 14.31** *Example 3*

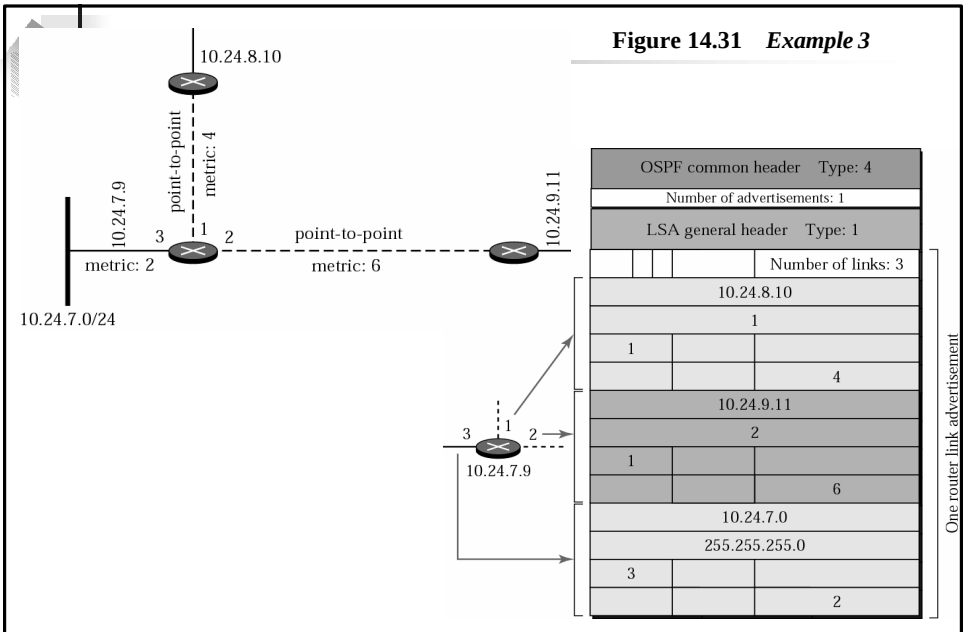
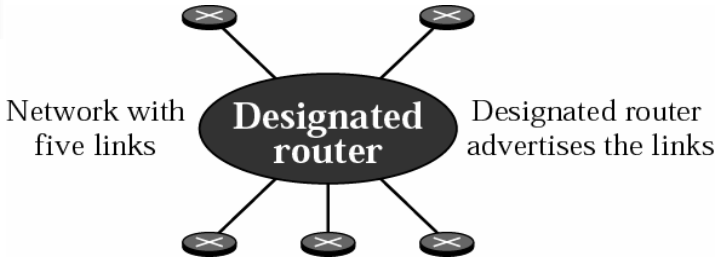


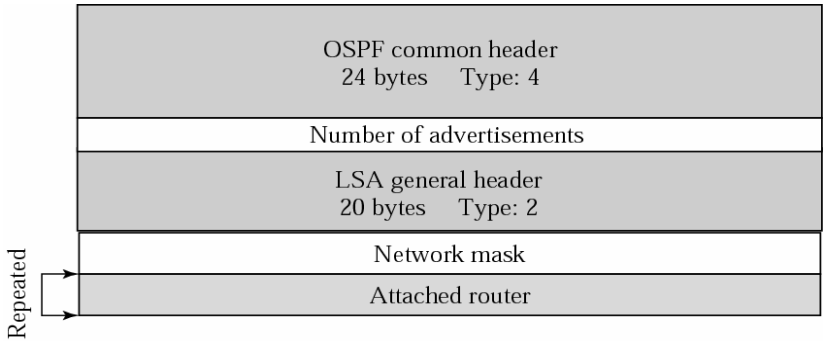



**Figure 14.33** *Network link*




- Network links LSA
  - Define the links of a network
  - A designated router, on behalf of the transient network, distributes this type of LSP packet
  - The packet announces the existence of all of the routers connected to the network

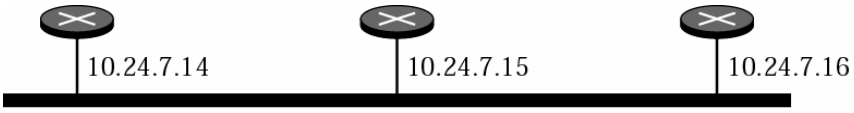
**Figure 14.34** *Network link advertisement format*






## *Example 4*

***Give the network link LSA in Figure 14.35.***

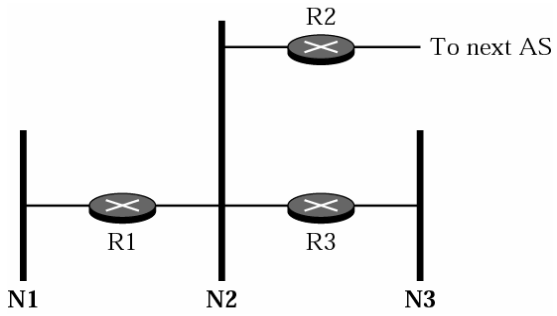



***Solution.***

| OSPF common header Type: 4 |
|---|
| Number of advertisements: 1 |
| LSA general header Type: 2 |
| 255.255.255.0 |
| 10.24.7.14 |
| 10.24.7.15 |
| 10.24.7.16 |

## Example 5

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




***Solution***

***All routers advertise router link LSAs.***

***a. R1 has two links, N1 and N2.***

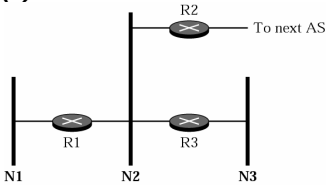
***b. R2 has one link, N1.***

***c. R3 has two links, N2 and N3.***



## Example 6

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




***Solution***

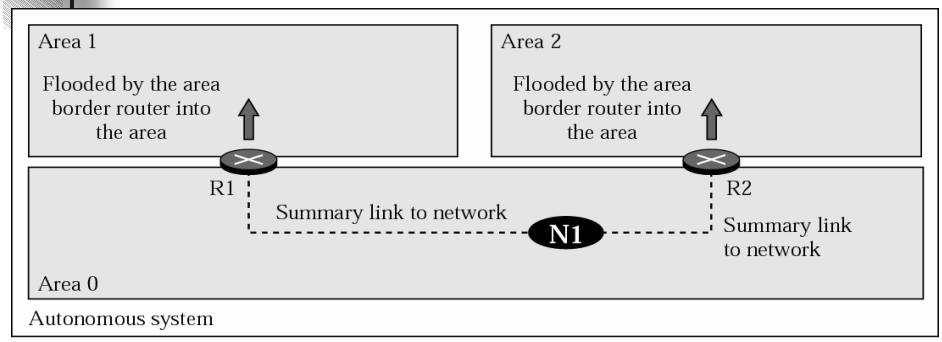
***All three network must advertise network links:***

***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.***

**Figure 14.38** ***Summary link to network***




- Router link and network link advertisements flood the area with information about the router links and network links inside an area
- Summary link to network LSA
  - Used by the area border router to announce the existence of other networks outside the area



**Figure 14.39** ***Summary link to network LSA***

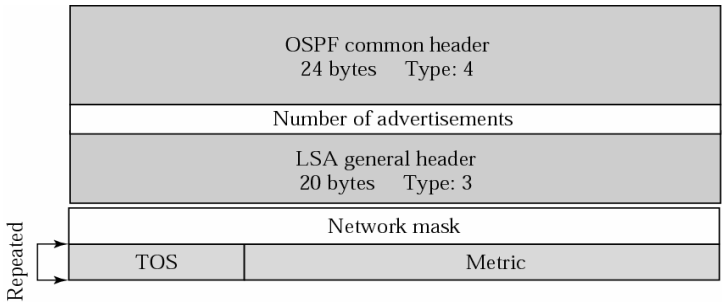
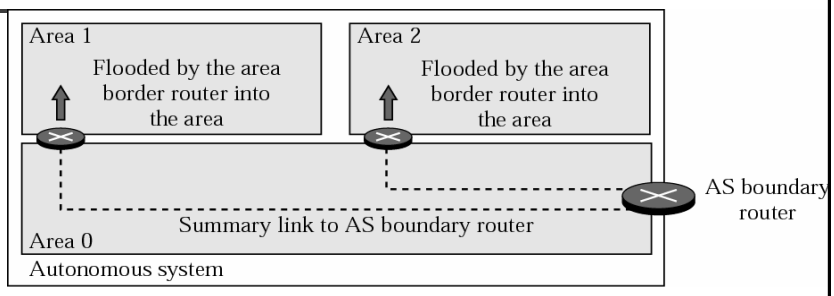


**Figure 14.40** ***Summary link to AS boundary router***




- The summary link to AS boundary router provides the information of what the route to an autonomous boundary router

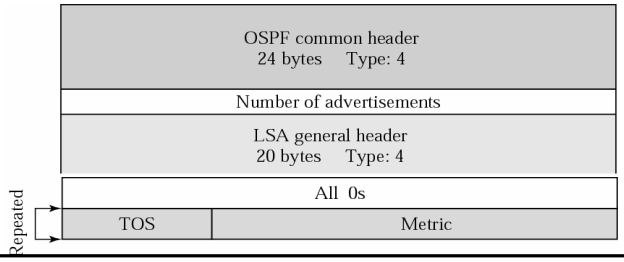
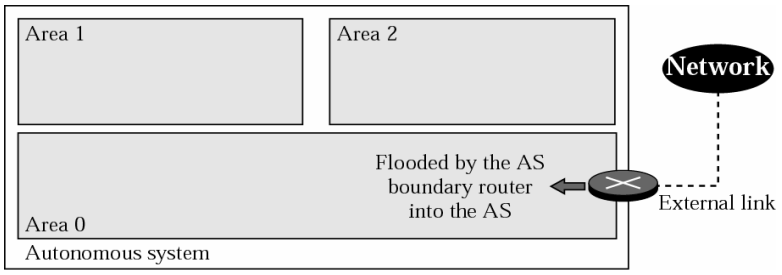


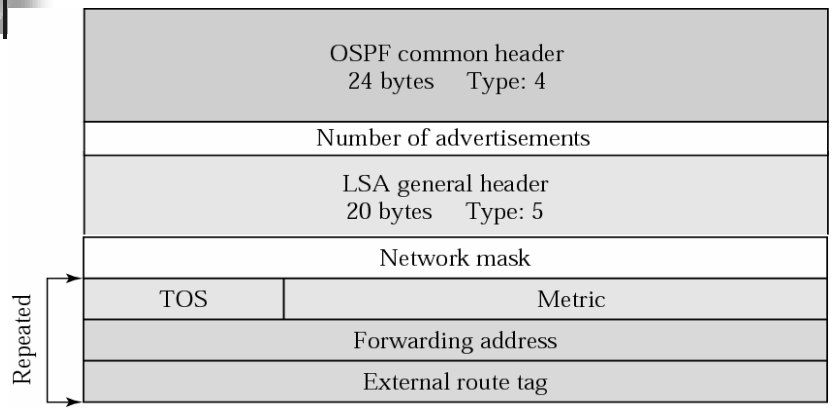



**Figure 14.42** ***External link***




- The external link advertisement provides the information about which networks are available outside the autonomous system
- The AS boundary router floods the autonomous system with the cost of each network outside the autonomous system using a routing table created by an interdomain routing protocol

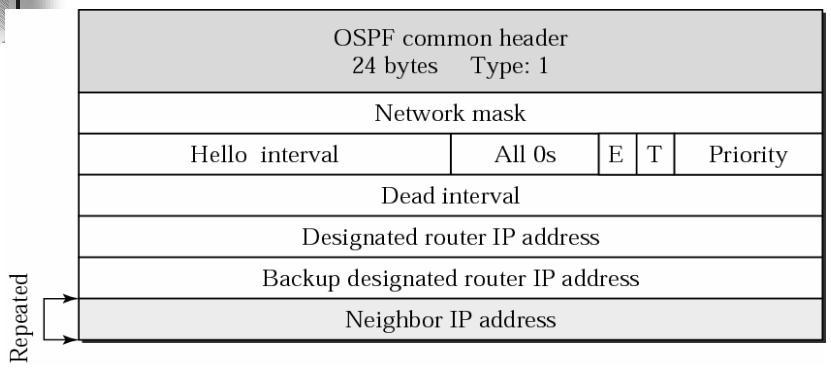
**Figure 14.43** ***External link LSA***




- Forwarding address: forwarding router that can provide a better route to the destination
- Route tag: used by other protocols

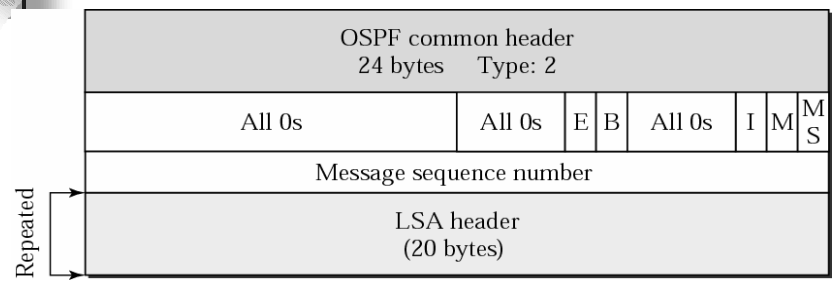


**Figure 14.44** ***Hello packet***




- Hello Message
  - Create neighborhood relationships and to test the reachability of neighbors
  - Before a router can flood all of the other routers with information , it must first greet its neighbors

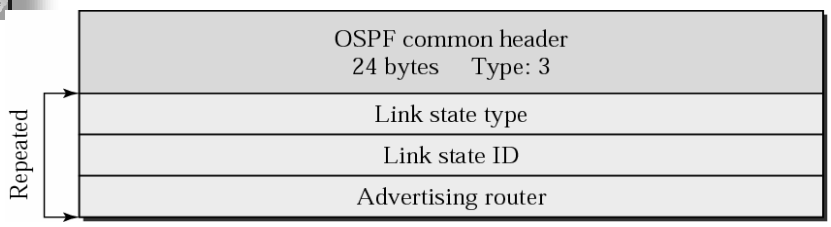
**Figure 14.45** ***Database description packet***




- Database description packet
  - When a router is connected to the system for the first time or after a failure, 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 message
    - The database description packet gives an outline, the title of each line in the database

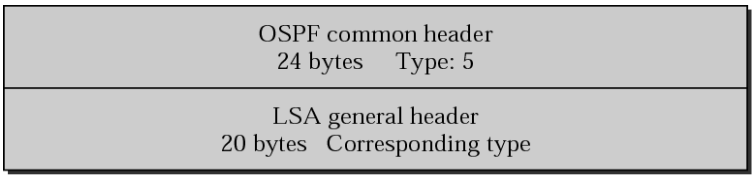


**Figure 14.46** ***Link state request packet***




- Link state request packet
  - Sent by a router that needs information about a specific route or routes
  - It is answered with a link state update packet

**Figure 14.47** *Link state acknowledgment packet*




- Link state acknowledgment packet
  - Make routing more reliable by forcing every router to acknowledge the reeipt of every link state update packet




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

## 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

- Distance vector and link state routing are both intradomain routing protocols
  - Used inside an autonomous system
  - Not suitable for interdomain routing because of scalability
- Path vector routing proved to be useful for interdomain routing
  - Similar to distance vector routing
  - Each autonomous system has a one node called speaker node which acts on behalf of the entire autonomous system
  - The speaker node creates a routing table and advertises it to speaker nodes in the neighboring Ass
  - The speaker node advertises the path, not the metric of the nodes

**Figure 14.48** *Initial routing tables in path vector routing*

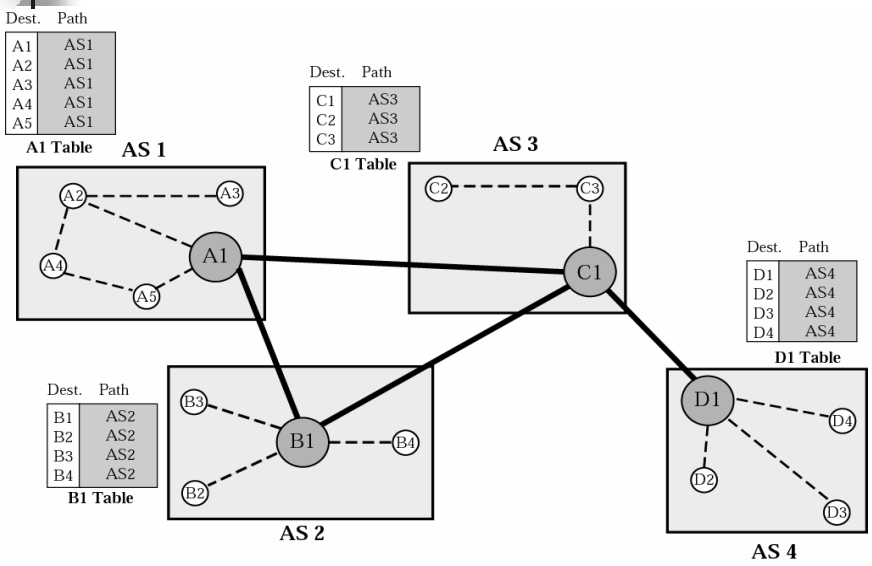
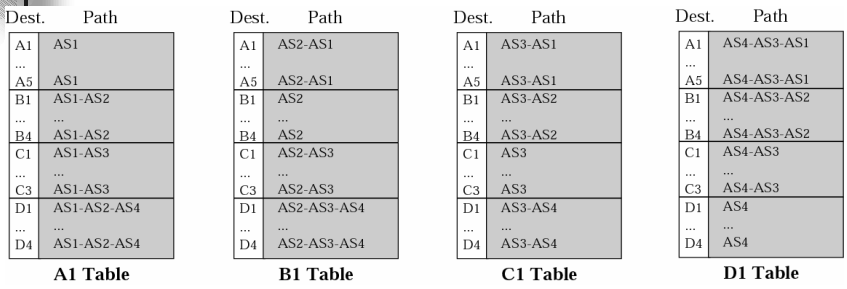





**Figure 14.49** *Stabilized tables for four autonomous systems*




- Loop prevention
  - The instability of distance vector routing and creation of loops can be avoided in path vector routing
  - When a router receives a message, it checks to see if its autonomous system is in the path list to the destination
    - If it is, looping is involved and the message is ignored
- Policy routing
  - Can be easily implemented through path vector routing
- Optimum 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:***

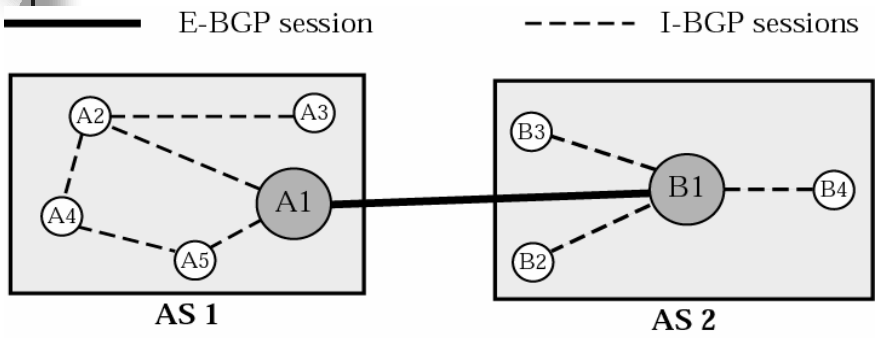
***Types of Autonomous Systems***
***Path Attributes***
***BGP Sessions***
***External and Internal BGP***
***Types of Packets***
***Packet Format***
***Encapsulation***



# Types of autonomous systems

- Stub AS
  - Only one connection to another AS
- Multihomed AS
  - More than one connection to other ASs, but it is still only a source or sink for data traffic
- Transit AS
  - Is a multihomed AS that also allows trnsient traffic (internet backbones)
- BGP uses Classless Interdomain Routing addresses (CIDR, prefix and prefix length)
- BGP sessions
  - BGP uses the services of TCP
  - A session at the BGP level, as an application program, is a connection at the TCP level

**Figure 14.50** *Internal and external BGP sessions*




- External BGP (E-BGP)
  - Use to exchange information between two speaker nodes belonging to two different autonomous systems
- Internal BGP (I-BGP)
  - Use to exchange routing information between two routers inside an autonomous system



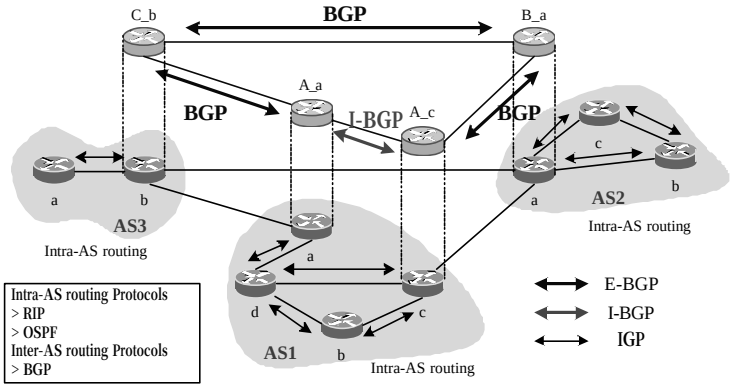
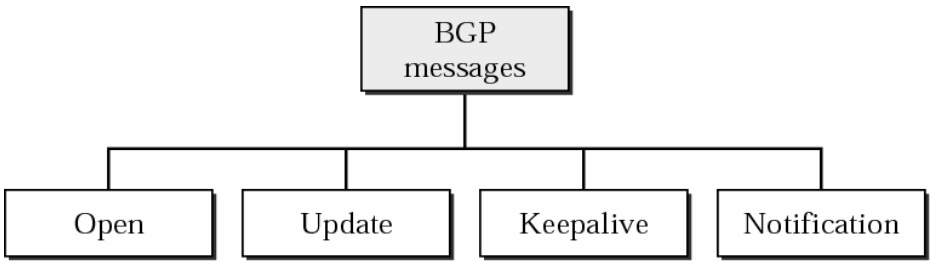


**Figure 14.51** *Types of BGP messages*






# BGP Functionality and message types

- BGP peers perform three functions
  - Initial peer acquisition and authentication
    - OPEN message
  - The primary focus of the protocol
    - UPDATE, NOTIFICATION message
  - Ongoing verification (the peers and the network connection are functioning correctly)
    - KEEPALIVE message

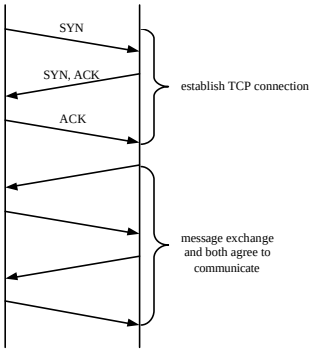
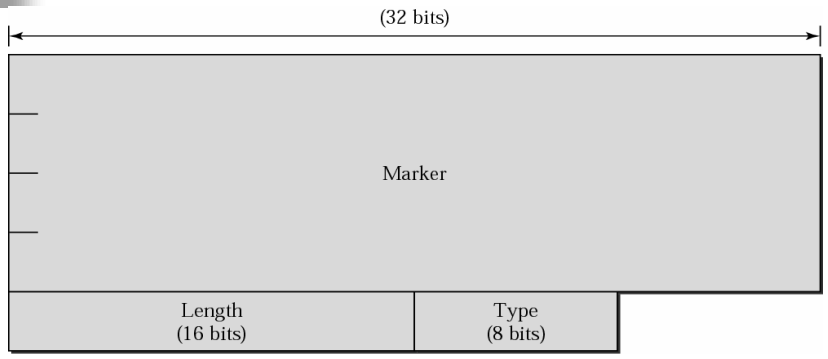


**Figure 14.52** ***BGP packet header***

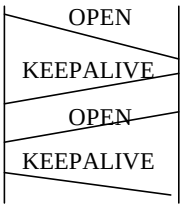



Marker : 16-octets, detect loss of synchronization between a pair of BGP speaker, authenticate incoming BGP messages
Length : 2-octets, total length of the message (including the header)
Type : 1-octet, detect message type
(1:OPEN, 2:UPDATE, 3:NOTIFICATION, 4:KEEPALIVE)



# BGP OPEN Message

- After TCP connection is established
- The first message sent by each side
- If OPEN message is acceptable, a KEEPALIVE message confirming the OPEN is sent back.



# BGP OPEN Message

- Each side must send an OPEN and receive a KEEPALIVE message before they can exchange routing information

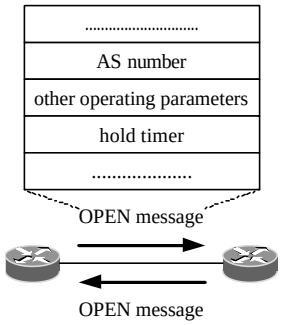
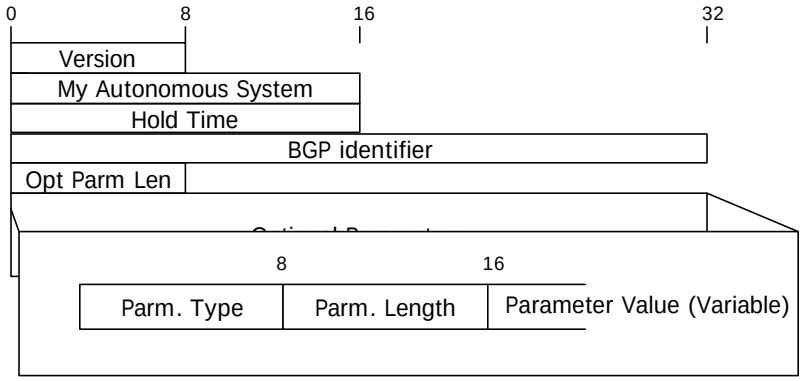





**Figure 14.53** ***Open message***

After a transport protocol connection is established, OPEN message sent each side.



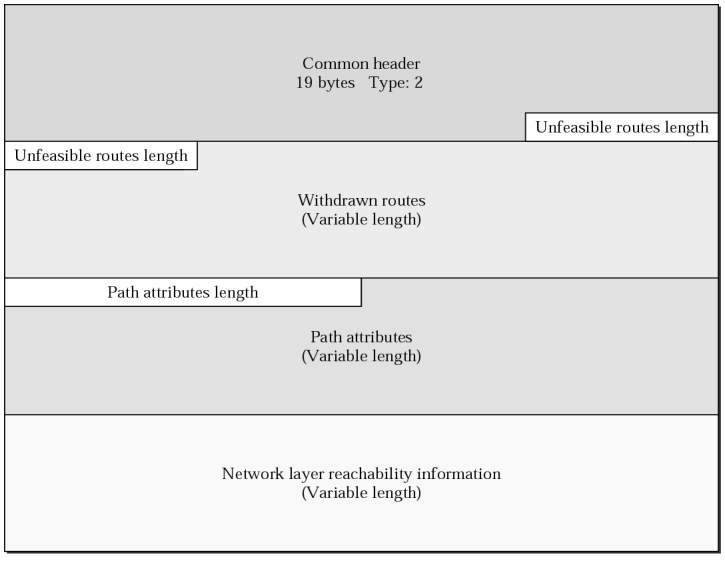

| | |
|---|---|
| Version | : 1-octet, BGP version number |
| My Autonomous System | : 2-octets, Autonomous System number of sender |
| Hold Time | : 2-octets |
| BGP identifier | : 4-octets, BGP identifier of the sender |
| Opt parameter length | : 1-octet, total length of the Optional parameter field in octet |

# BGP UPDATE Message Format

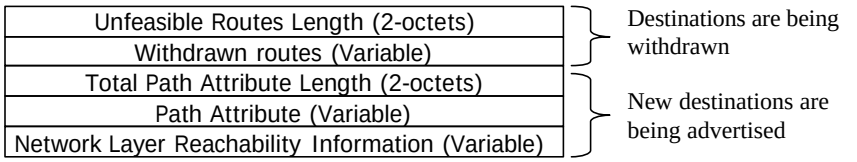
- Used to transfer routing information between BGP peers (to advertise new destination or withdraw unreachable destination)



**Figure 14.54** ***Update message***



# UPDATE Message Format




Unfeasible Routes Length : total length of withdraw routes in octets
Withdrawn route : a list of IP address prefix,2-tuple of the form<length, prefix>
Total Path Attribute Length : total length of path attribute field in octets
Path Attribute : A variable lengths sequence, each path attribute is a triple <attribute type, attribute length, attribute value>
Network Layer Reachability Information
: a list of IP address prefixes, each one is 2-tuple of the form <length, prefix>




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

**Figure 14.55** *Keepalive message*

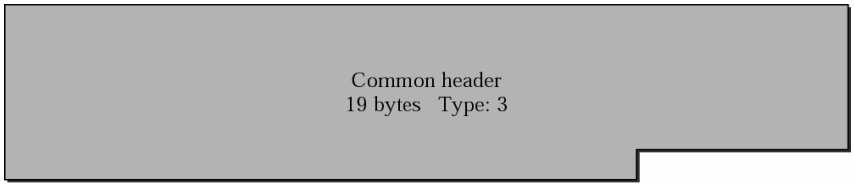





**Figure 14.56** *Notification message*

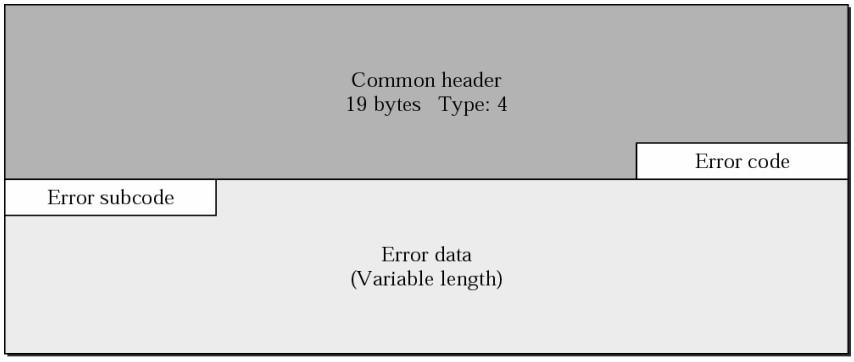


***Table 14.3 Error codes***

| *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. |




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