# Optional Topics

# ADDRESS MAPPING:

- An internet is made of a combination of physical networks connected by internetworking devices such as routers. A packet starting from a source host may pass through several different physical networks before finally reaching the destination host.
- The hosts and routers are recognized at the network level by their logical (IP) addresses. However, packets pass through physical networks to reach these hosts and routers.
- At the physical level, the hosts and routers are recognized by their physical addresses. A physical address is a local address. Its jurisdiction is a local network. It must be unique locally, but is not necessarily unique universally. It is called a *physical* address because it is usually (but not always) implemented in hardware.
- An example of a physical address is the 48-bit MAC address in the Ethernet protocol, which is imprinted on the NIC installed in the host or router.

# ADDRESS MAPPING:

- The physical address and the logical address are two different identifiers. We need both because a physical network such as Ethernet can have two different protocols at the network layer such as IP and IPX (Novell) at the same time. Likewise, a packet at a network layer such as IP may pass through different physical networks such as Ethernet and LocalTalk (Apple).
- This means that delivery of a packet to a host or a router requires two levels of addressing: logical and physical.
- We need to be able to map a logical address to its corresponding physical address and vice versa.
- These can be done by using either static or dynamic mapping.

# ADDRESS MAPPING:

- The physical address and the logical address are two different identifiers. We need both because a physical network such as Ethernet can have two different protocols at the network layer such as IP and IPX (Novell) at the same time. Likewise, a packet at a network layer such as IP may pass through different physical networks such as Ethernet and LocalTalk (Apple).
- This means that delivery of a packet to a host or a router requires two levels of addressing: logical and physical.
- We need to be able to map a logical address to its corresponding physical address and vice versa.
- These can be done by using either static or dynamic mapping.

# Static Mapping

- Static mapping involves in the creation of a table that associates a logical address with a physical address. This table is stored in each machine on the network. Each machine that knows, for example, the IP address of another machine but not its physical address can look it up in the table. This has some limitations because physical addresses may change in the following ways:
  - A machine could change its NIC, resulting in a new physical address.
  - In some LANs, such as LocalTalk, the physical address changes every time the computer is turned on.
  - A mobile computer can move from one physical network to another, resulting in a change in its physical address.
  - To implement these changes, a static mapping table must be updated periodically. This overhead could affect network performance. In dynamic mapping each time a machine knows one of the two addresses (logical or physical), it can use a protocol to find the other one.

# ARP

# ARP:

Topics Covered:

1. Mapping Logical to Physical Address:
2. ARP – Request and Reply
3. ARP-Packet Format
4. Encapsulation:
5. Operation of ARP:
6. Four Cases Using ARP
7. Proxy ARP:

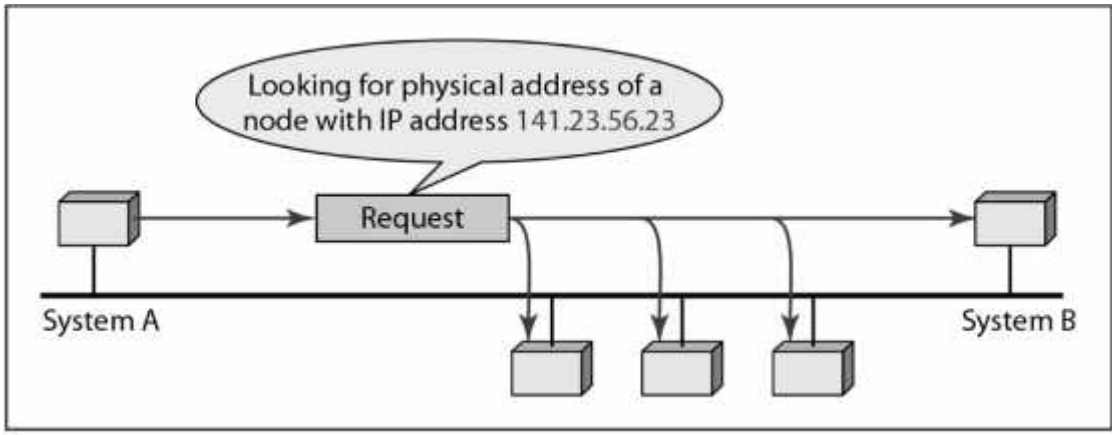
# Mapping Logical to Physical Address: ARP:

- Anytime a host or a router has an IP datagram to send to another host or router, it has the logical (IP) address of the receiver.
- The logical (IP) address is obtained from the DNS  the sender is the host or it is found in a routing table  if the sender is a router.
- But the IP datagram must be encapsulated in a frame to be able to pass through the physical network. This means that the sender needs the physical address of the receiver.
- The host or the router sends an ARP query packet. The packet includes the physical and IP addresses of the sender and the IP address of the receiver.
- Because the sender does not know the physical address of the receiver, the query is broadcast over the network.

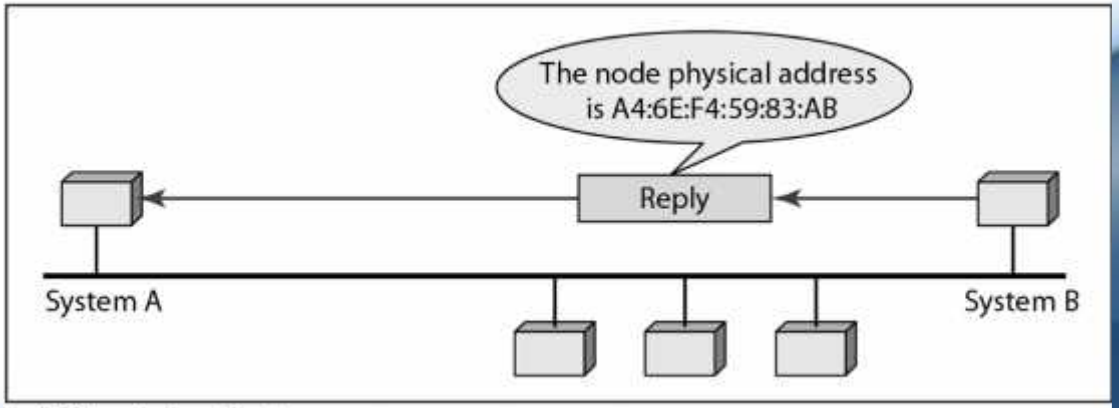
# Mapping Logical to Physical Address: ARP:

- Every host or router on the network receives and processes the ARP query packet, but only the intended recipient recognizes its IP address and sends back an ARP response packet.
- The response packet contains the recipient's IP and physical addresses.
- The packet is unicast directly to the inquirer by using the physical address received in the query packet.

# ARP – Request and Reply




a. ARP request is broadcast

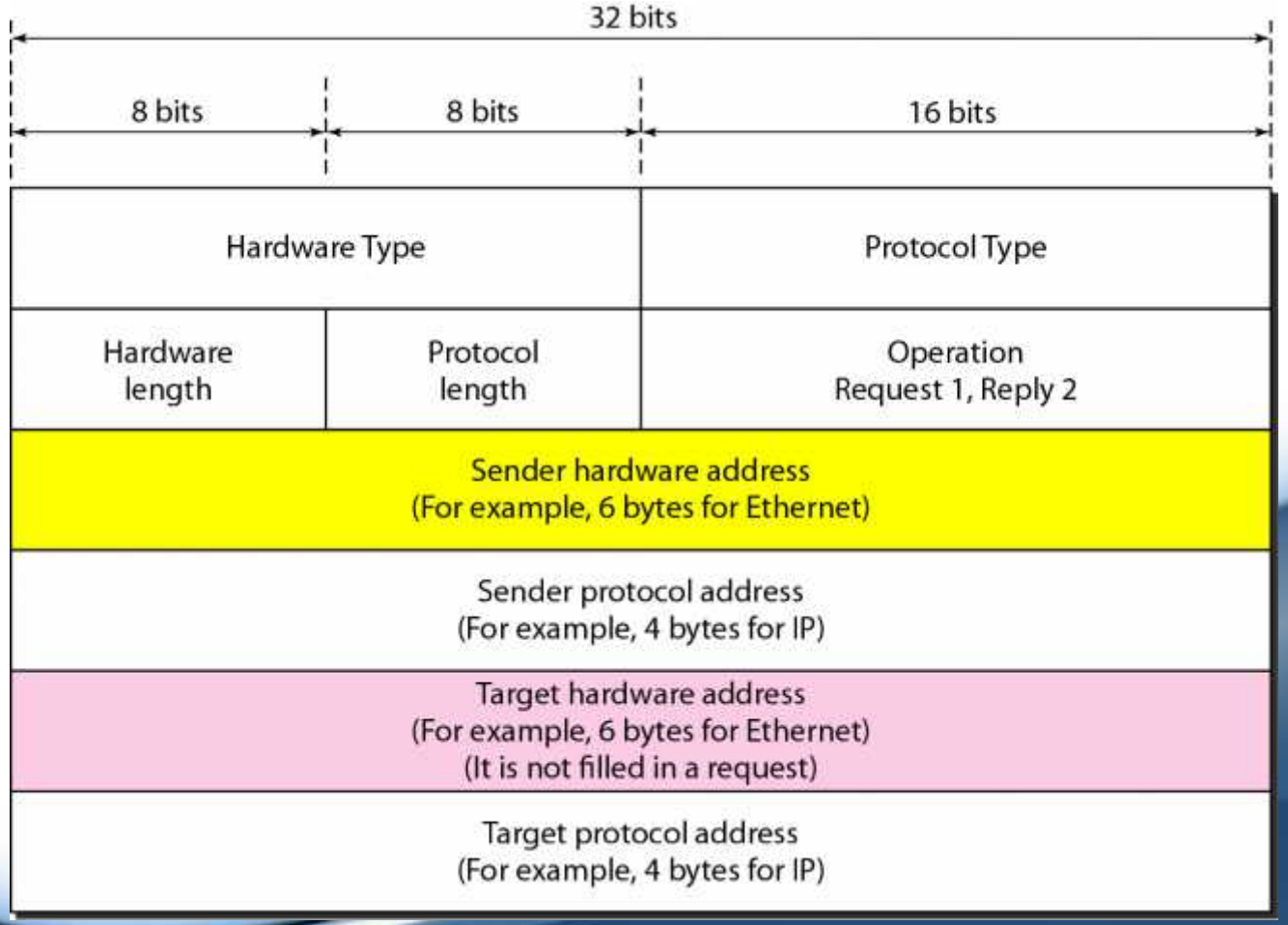



b. ARP reply is unicast

# Cache Memory:

- Using ARP is inefficient if system A needs to broadcast an ARP request for each IP packet it needs to send to system B.
- It could have broadcast the IP packet itself. ARP can be useful if the ARP reply is cached (kept in cache memory for a while) because a system normally sends several packets to the same destination.
- A system that receives an ARP reply stores the mapping in the cache memory and keeps it for 20 to 30 minutes unless the space in the cache is exhausted.
- Before sending an ARP request, the system first checks its cache to see if it can find the mapping

# ARP-Packet Format



# Fields of ARP Packet

- Hardware type:
  - This is a 16-bit field defining the type of the network on which ARP is running. Each LAN has been assigned an integer based on its type. For example, Ethernet is given type 1. ARP can be used on any physical network.
- Protocol type:
  - This is a 16-bit field defining the protocol. For example, the value of this field for the IPv4 protocol is 080016, ARP can be used with any higher-level protocol.

# Fields of ARP Packet

- Hardware length:
  - This is an 8-bit field defining the length of the physical address in bytes. For example, for Ethernet the value is 6.
- Protocol length:
  - This is an 8-bit field defining the length of the logical address in bytes. For example, for the IPv4 protocol the value is 4.
- Operation:
  - This is a 16-bit field defining type of packet. Two packet types are defined: ARP request (1) & ARP reply (2).

# Fields of ARP Packet

- Sender hardware address:
  - This is a variable-length field defining the physical address of the sender. For example, for Ethernet this field is 6 bytes long.

- Sender protocol address:
  - This is a variable-length field defining the logical (for example, IP) address of the sender. For the IP protocol, this field is 4 bytes long.

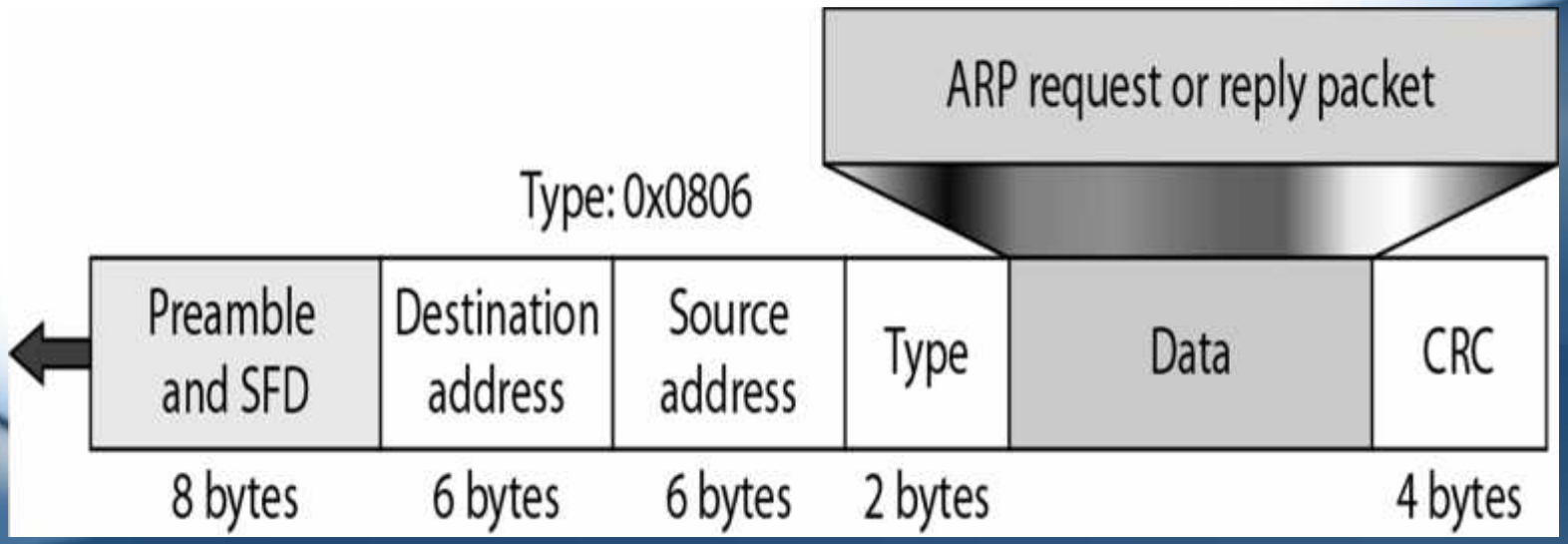
# Fields of ARP Packet

- Target hardware address:
  - This is a variable-length field defining the physical address of the target. For example, for Ethernet this field is 6 bytes long. For an ARP request message, this field is all Os because the sender does not know the physical address of the target.

- Target protocol address:
  - This is a variable-length field defining the logical (for example, IP) address of the target. For the IPv4 protocol, this field is 4 bytes long.

# Encapsulation:

- An ARP packet is encapsulated directly into a data link frame.
- For example, in Figure 21.3 an ARP packet is encapsulated in an Ethernet frame. Note that the type field indicates that the data carried by the frame are an ARP packet



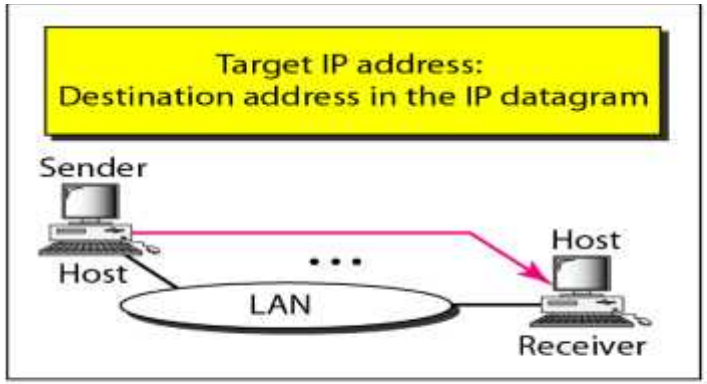
# Operation of ARP:

- The sender knows the IP address of the target. We will see how the sender obtains this shortly.
- IP asks ARP to create an ARP request message, filling in the sender physical address, the sender IP address, and the target IP address. The target physical address field is filled with Os.
- The message is passed to the data link layer where it is encapsulated in a frame by using the physical address of the sender as the source address and the physical broadcast address as the destination address.

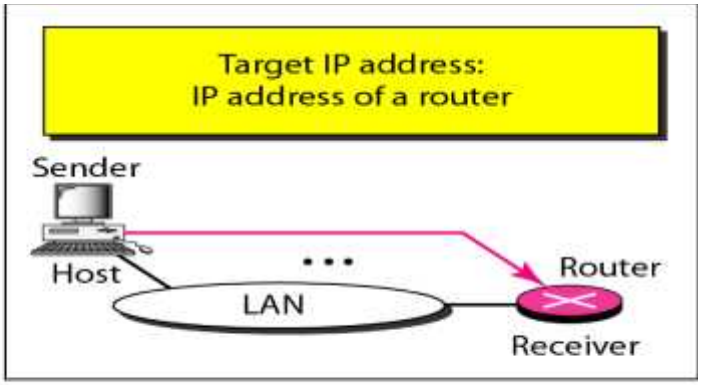
# Operation of ARP:

- Every host or router receives the frame. Because the frame contains a broadcast destination address, all stations remove the message and pass it to ARP. All machines except the one targeted drop the packet. The target machine recognizes its IP address.
- The target machine replies with an ARP reply message that contains its physical address. The message is unicast.
- The sender receives the reply message. It now knows the physical address of the target machine.
- The IP datagram, which carries data for the target machine, is now encapsulated in a frame and is unicast to the destination.

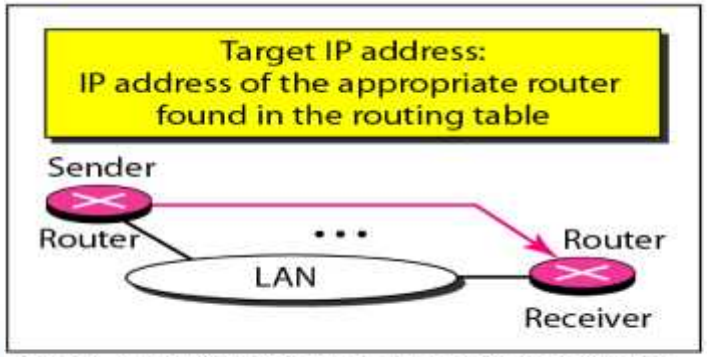
# Four Cases Using ARP



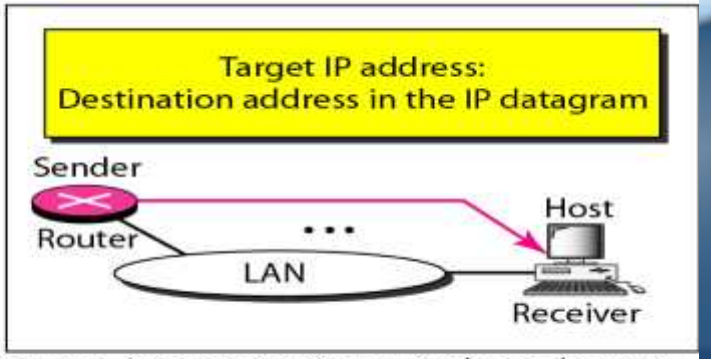

Case 1. A host has a packet to send to another host on the same network.




Case 2. A host wants to send a packet to another host on another network. It must first be delivered to a router.



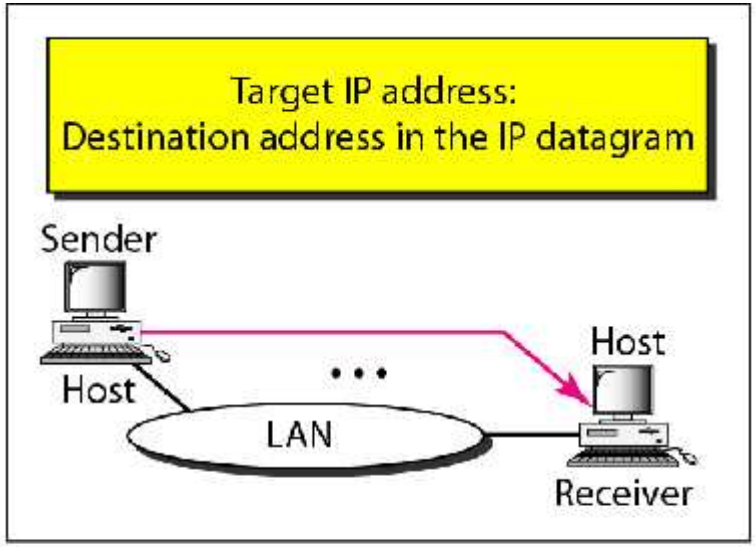

Case 3. A router receives a packet to be sent to a host on another network. It must first be delivered to the appropriate router.




Case 4. A router receives a packet to be sent to a host on the same network.

# Case 1

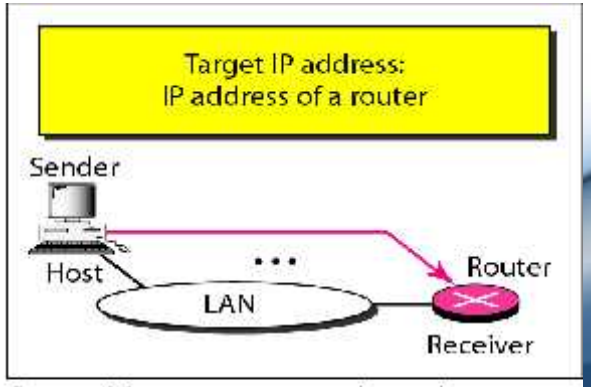
- The sender is a host and wants to send a packet to another host on the same network. In this case, the logical address that must be mapped to a physical address is the destination IP address in the datagram header.




Case 1. A host has a packet to send to another host on the same network.

# Case 2

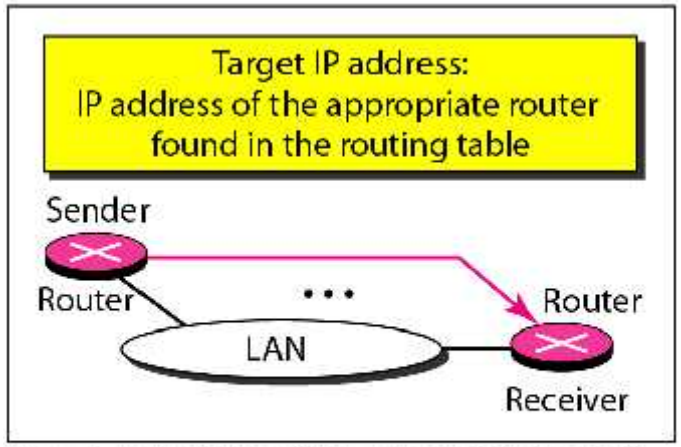
- The sender is a host and wants to send a packet to another host on another network.
- In this case, the host looks at its routing table and finds the IP address of the next hop (router) for this destination. If it does not have a routing table, it looks for the IP address of the default router. The IP address of the router becomes the logical address that must be mapped to a physical address.




Case 2. A host wants to send a packet to another host on another network. It must first be delivered to a router.

# Case 3

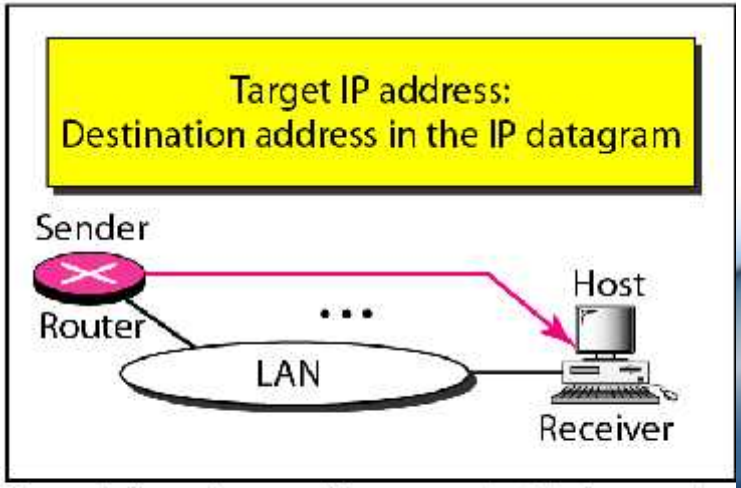
- The sender is a router that has received a datagram destined for a host on another network. It checks its routing table and finds the IP address of the next router.
- The IP address of the next router becomes the logical address that must be mapped to a physical address.




Case 3. A router receives a packet to be sent to a host on another network. It must first be delivered to the appropriate router.

# Case 4

- The sender is a router that has received a datagram destined for a host on the same network.
- The destination IP address of the datagram becomes the logical address that must be mapped to a physical address.




Case 4. A router receives a packet to be sent to a host on the same network.

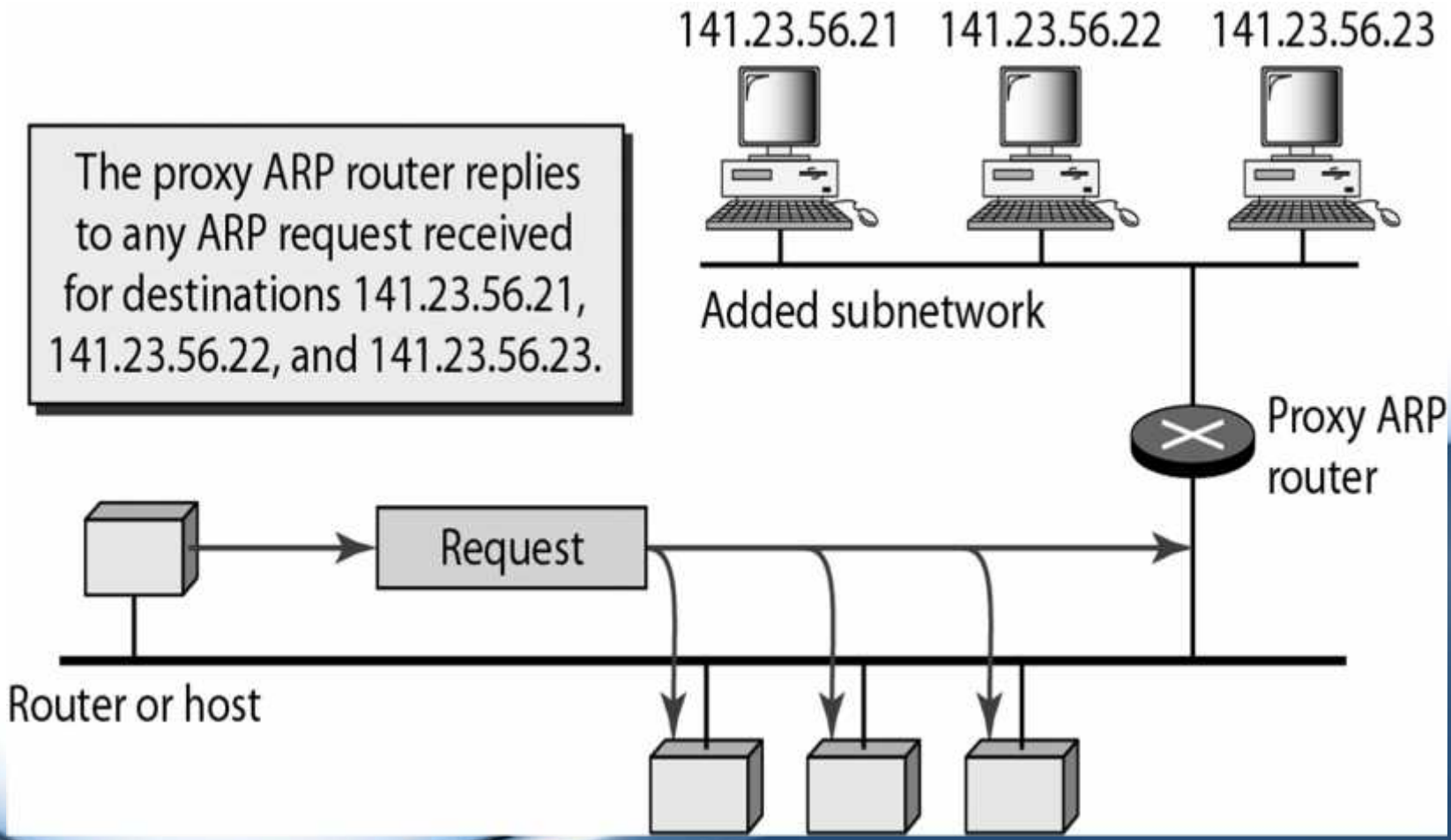
# Proxy ARP:

- A technique called *proxy* ARP is used to create a subnetting effect. A proxy ARP is an ARP that acts on behalf of a set of hosts. Whenever a router running a proxy ARP receives an ARP request looking for the IP address of one of these hosts, the router sends an ARP reply announcing its own hardware (physical) address.
- After the router receives the actual IP packet, it sends the packet to the appropriate host or router. Let us give an example. In Figure 21.6 the ARP installed on the right-hand host will answer only to an ARP request with a target IP address of 141.23.56.23. However, the administrator may need to create a subnet without changing the whole system to recognize subnetted addresses.

# Proxy ARP:

- One solution is to add a router running a proxy ARP. In this case, the router acts on behalf of all the hosts installed on the subnet.
- When it receives an ARP request with a target IP address that matches the address of one of its groups (141.23.56.21, 141.23.56.22, or 141.23.56.23), it sends an ARP reply and announces its hardware address as the target hardware address. When the router receives the IP packet, it sends the packet to the appropriate host.

# Proxy ARP


141.23.56.21
141.23.56.22
141.23.56.23
The proxy ARP router replies to any ARP request received for destinations 141.23.56.21, 141.23.56.22, and 141.23.56.23.
Added subnetwork
Proxy ARP router
Request
Router or host

# Thank You

# ICMP

# ICMP - Intro

- The IP provides unreliable and connectionless datagram delivery. It was designed this way to make efficient use of network resources. The IP protocol is a best-effort delivery service that delivers a datagram from its original source to its final destination. However, it has two deficiencies:
  - lack of error control
  - lack of assistance mechanisms.
- The IP protocol has no error-reporting or error-correcting mechanism.
  - What happens if something goes wrong?
  - What happens if a router must discard a datagram because it cannot find a router to the final destination, or because the time-to-live field has a zero value?
  - What happens if the final destination host must discard all fragments of a datagram because it has not received all fragments within a predetermined time limit?
- These are examples of situations where an error has occurred and the IP protocol has no built-in mechanism to notify the original host.

# ICMP - Intro

- The IP protocol also lacks a mechanism for host and management queries.
- A host sometimes needs to determine if a router or another host is alive. And sometimes a network administrator needs information from another host or router.
- The Internet Control Message Protocol (ICMP) has been designed to compensate for the above two deficiencies. It is a companion to the IP protocol.

# What we Need to study in ICMP

- Types of Messages
- Message Format
- Error Reporting Messages:
  - Destination Unreachable
  - Source Quench
  - Time Excedded
  - Parameter Problem
  - Redirection
- Query Messages:
  - Echo Request and Reply
  - Timestamp Request and Reply
  - Address-Mask Request and Reply
  - Router solicitation and advertisement
- Checksum
- Debugging Tools
  - Ping
  - TraceRoute

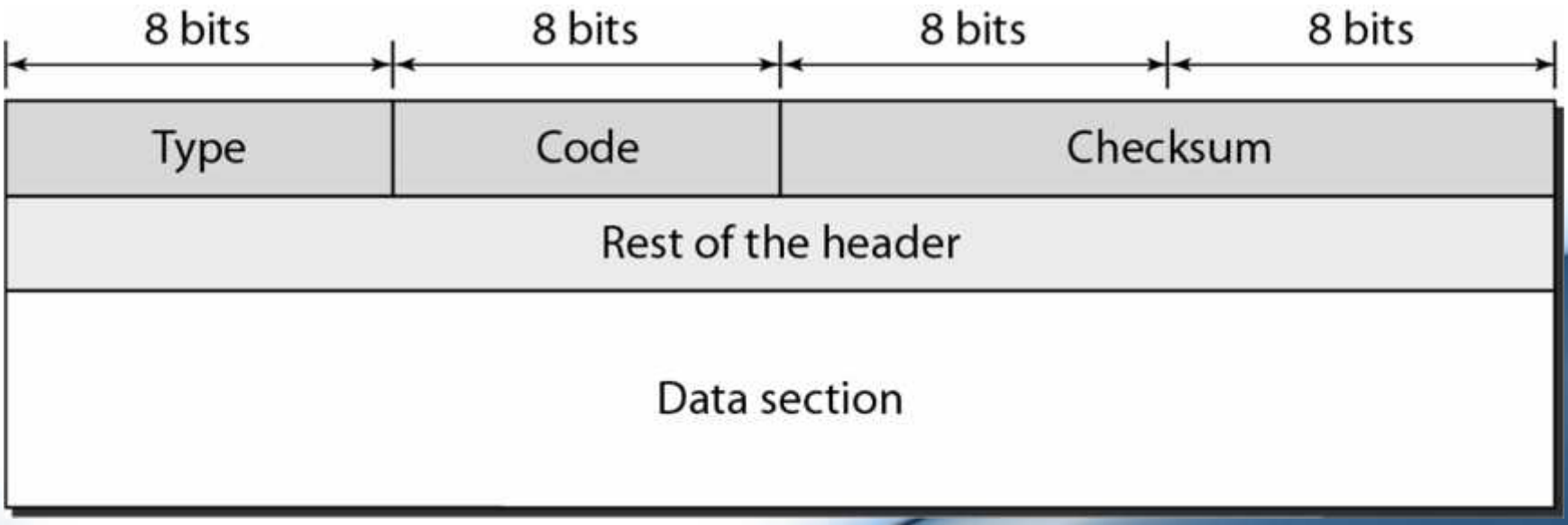
# Types of Messages:

- ICMP messages are divided into two broad categories: error-reporting messages and query messages.
- The error-reporting messages report problems that a router or a host (destination) may encounter when it processes an IP packet.
- The query messages, which occur in pairs, help a host or a network manager get specific information from a router or another host.
  - For example, nodes can discover their neighbors.
  - Also, hosts can discover and learn about routers on their network, and routers can help a node redirect its messages.

# Message Format:

- An ICMP message has an 8-byte header and a variable-size data section.
- Although the general format of the header is different for each message type, the first 4 bytes are common to all.
  - The first field, ICMP type, defines the type of the message.
  - The code field specifies the reason for the particular message type. The last common field is the checksum field.
  - The rest of the header is specific for each message type.
  - The data section in error messages carries information for finding the original packet that had the error.
  - In query messages, the data section carries extra information based on the type of the query.

# Message Format:



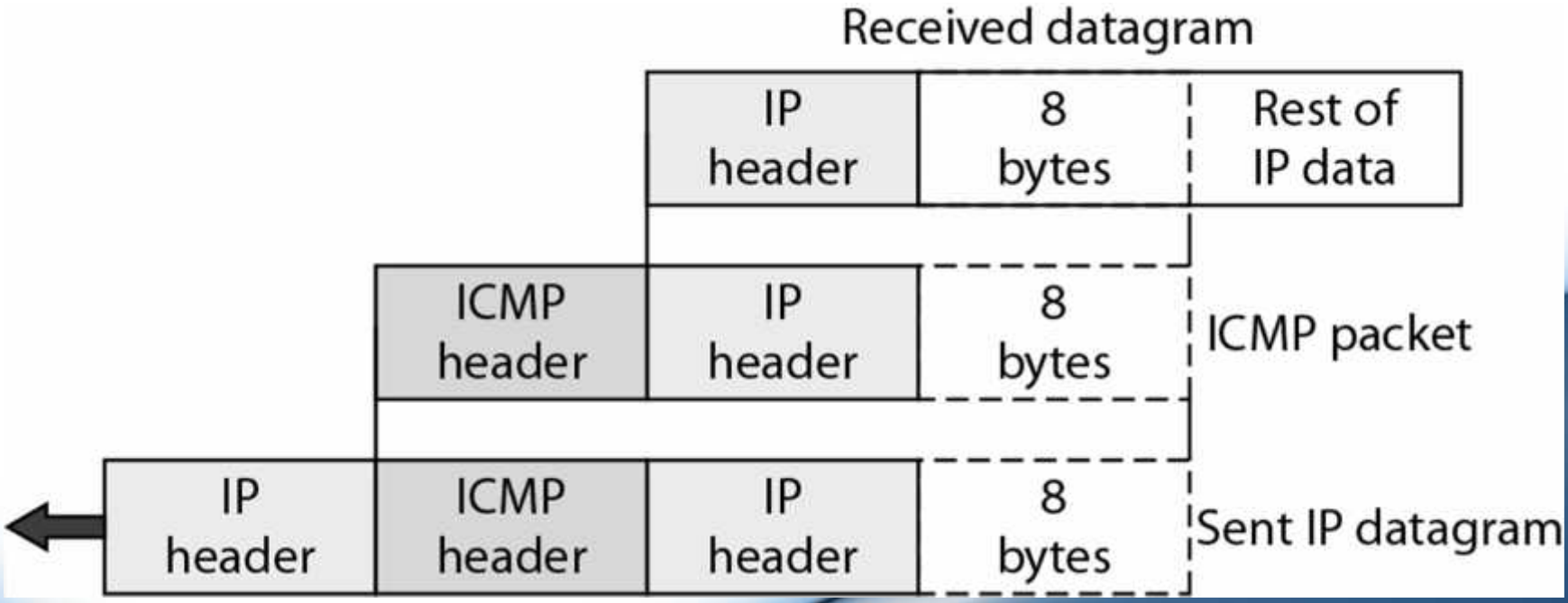
# Error Reporting Messages

- One of the main responsibilities of ICMP is to report errors. Although technology has produced increasingly reliable transmission media, errors still exist and must be handled. IP is an unreliable protocol. This means that error checking and error control are not a concern of IP.
- ICMP was designed, in part, to compensate for this shortcoming. However, ICMP does not correct errors-it simply reports them. Error correction is left to the higher-level protocols.
- Error messages are always sent to the original source because the only information available in the datagram about the route is the source and destination IP addresses.
- ICMP uses the source IP address to send the error message to the source (originator) of the datagram. ICMP always reports error messages to the original source.

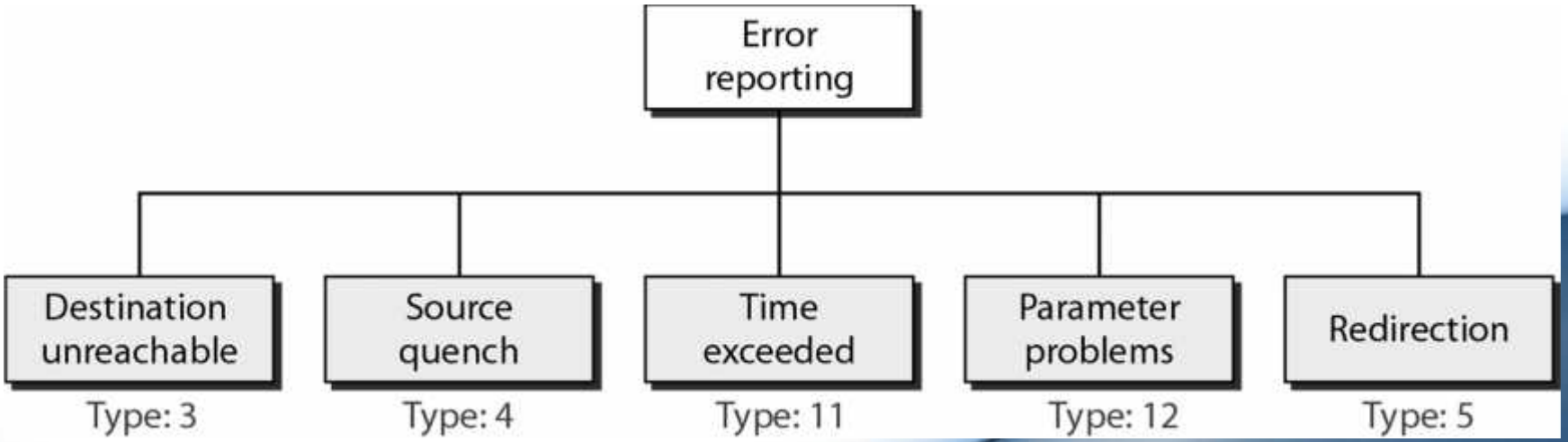
# Important Points

- The following are important points about ICMP error messages:
  - No ICMP error message will be generated in response to a datagram carrying an ICMP error message.
  - No ICMP error message will be generated for a fragmented datagram that is not the first fragment.
  - No ICMP error message will be generated for a datagram having a multicast address.
  - No ICMP error message will be generated for a datagram having a special address such as 127.0.0.0 or 0.0.0.0.

# ICMP – Error Reporting



# Error Reporting Messages



# Destination Unreachable:

- When a router cannot route a datagram or a host cannot deliver a datagram, the datagram is discarded and the router or the host sends a destination-unreachable message back to the source host that initiated the datagram.

- Note that destination-unreachable messages can be created by either a router or the destination host.

# Source Quench:

- The IP protocol is a connectionless protocol.
- There is no communication between the source host, which produces the datagram, the routers, which forward it, and the destination host, which processes it.
- One of the ramifications of this absence of communication is the lack of *flow control.* IP does not have a flow control mechanism embedded in the protocol. The lack of flow control can create a major problem in the operation of IP: congestion.
- The source host never knows if the routers or the destination host has been overwhelmed with datagrams.
- The source host never knows if it is producing datagrams faster than can be forwarded by routers or processed by the destination host.

# Source Quench:

- The lack of flow control can create congestion in routers or the destination host. A router or a host has a limited-size queue (buffer) for incoming datagrams waiting to be forwarded (in the case of a router) or to be processed (in the case of a host).
- If the datagrams are received much faster than they can be forwarded or processed, the queue may overflow. In this case, the router or the host has no choice but to discard some of the datagrams.
- The source-quench message in ICMP was designed to add a kind of flow control to the IP. When a router or host discards a datagram due to congestion, it sends a source-quench message to the sender of the datagram.
- This message has two purposes.
  - First, it informs the source that the datagram has been discarded.
  - Second, it warns the source that there is congestion somewhere in the path and that the source should slow down (quench) the sending process.

# Time Exceeded:

- The time-exceeded message is generated in two cases: routers use routing tables to find the next hop (next router) that must receive the packet.
- If there are errors in one or more routing tables, a packet can travel in a loop or a cycle, going from one router to the next or visiting a series of routers endlessly.
- Each datagram contains a field called *time to live* that controls this situation. When a datagram visits a router, the value of this field is decremented by 1. When the time-to-live value reaches 0, after decrementing, the router discards the datagram.
- However, when the datagram is discarded, a time-exceeded message must be sent by the router to the original source.
- Second, a time-exceeded message is also generated when not all fragments that make up a message arrive at the destination host within a certain time limit.

# Parameter Problem:

- Any ambiguity in the header part of a datagram can Create serious problems as the datagram travels through the Internet.

- If a router or the destination host discovers an ambiguous or missing value in any field of the datagram, it discards the datagram and sends a parameter-problem message back to the source.

# Redirection:

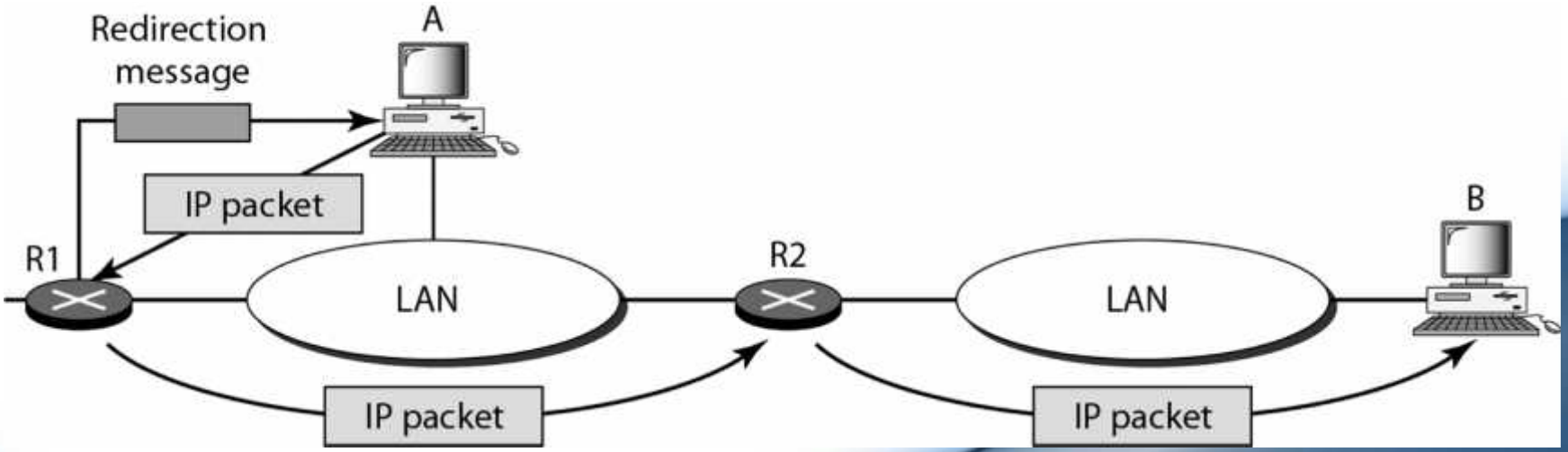
- When a router needs to send a packet destined for another network, it must know the IP address of the next appropriate router. The same is true if the sender is a host. Both routers and hosts, then, must have a routing table to find the address of the router or the next router. Routers take part in the routing update process and are supposed to be updated constantly. Routing is dynamic.

- However, for efficiency, hosts do not take part in the routing update process because there are many more hosts in an internet than routers. Updating the routing tables of hosts dynamically produces unacceptable traffic. The hosts usually use static routing.

# Redirection:

- When a host comes up, its routing table has a limited number of entries. It usually knows the IP address of only one router, the default router. For this reason, the host may send a datagram, which is destined for another network, to the wrong router. In this case, the router that receives the datagram will forward the datagram to the correct router.
- However, to update the routing table of the host, it sends a redirection message to the host.
- Host A wants to send a datagram to host B. Router R2 is obviously the most efficient routing choice, but host A did not choose router R2. The datagram goes to R1 instead. Router R1, after consulting its table, finds that the packet should have gone to R2. It sends the packet to R2 and, at the same time, sends a redirection message to host A. Host A's routing table can now be updated.

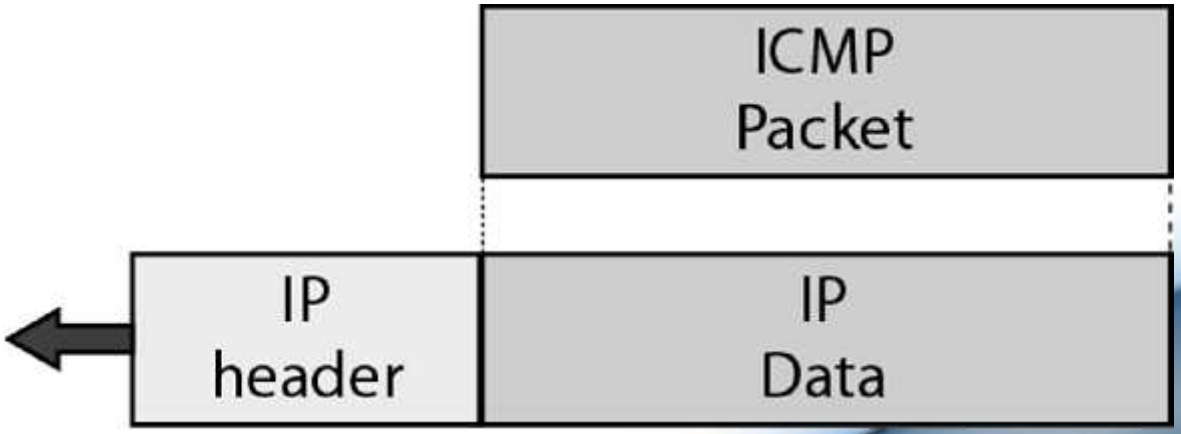
# Redirection:



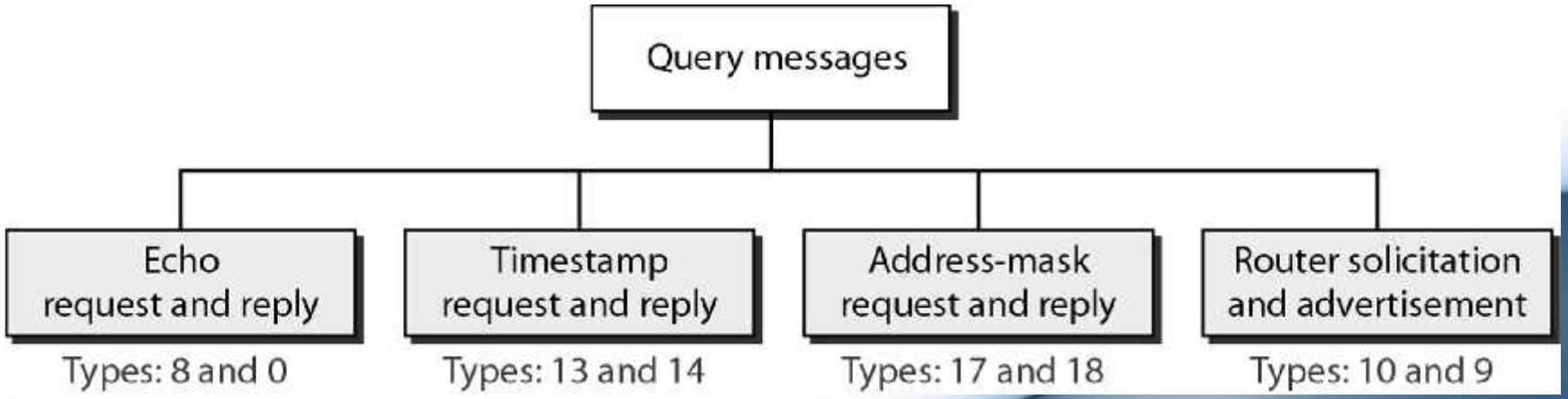
# Query Messages

- In addition to error reporting, ICMP can diagnose some network problems. This is accomplished through the query messages, a group of four different pairs of messages.
- In this type of ICMP message, a node sends a message that is answered in a specific format by the destination node.
- A query message is encapsulated in an IP packet, which in turn is encapsulated in a data link layer frame.
- However, in this case, no bytes of the original IP are included in the message

# Query Messages



# Query Messages



# Echo Request and Reply:

- The echo-request and echo-reply messages are designed for diagnostic purposes. Network managers and users utilize this pair of messages to identify network problems. The combination of echo-request & echo-reply messages determines whether 2 systems (hosts or routers) can communicate.
- The echo-request and echo-reply messages can be used to determine if there is communication at the IP level. Because ICMP messages are encapsulated in IP datagrams, the receipt of an echo-reply message by the machine that sent the echo request is proof that the IP protocols in the sender and receiver are communicating with each other using the IP datagram.
- Also, it is proof that the intermediate routers are receiving, processing, and forwarding IP datagrams.
- Today, most systems provide a version of the *ping* command that can create a series (instead of just one) of echo-request and echo-reply messages, providing statistical information

# Timestamp Request and Reply:

- Two machines (hosts or routers) can use the timestamp request and timestamp reply messages to determine the round-trip time needed for an IP datagram to travel between them.

- It can also be used to synchronize the clocks in two machines.

# Address-Mask Request and Reply:

- A host may know its IP address, but it may not know the corresponding mask. For example, a host may know its IP address as 159.31.17.24, but it may not know that the corresponding mask is /24.
- To obtain its mask, a host sends an address-mask-request message to a router on the LAN.
  - If the host knows the address of the router, it sends the request directly to the router.
  - If it does not know, it broadcasts the message.
- The router receiving the address-mask-request message responds with an address-mask-reply message, providing the necessary mask for the host.
- This can be applied to its full IP address to get its subnet address.

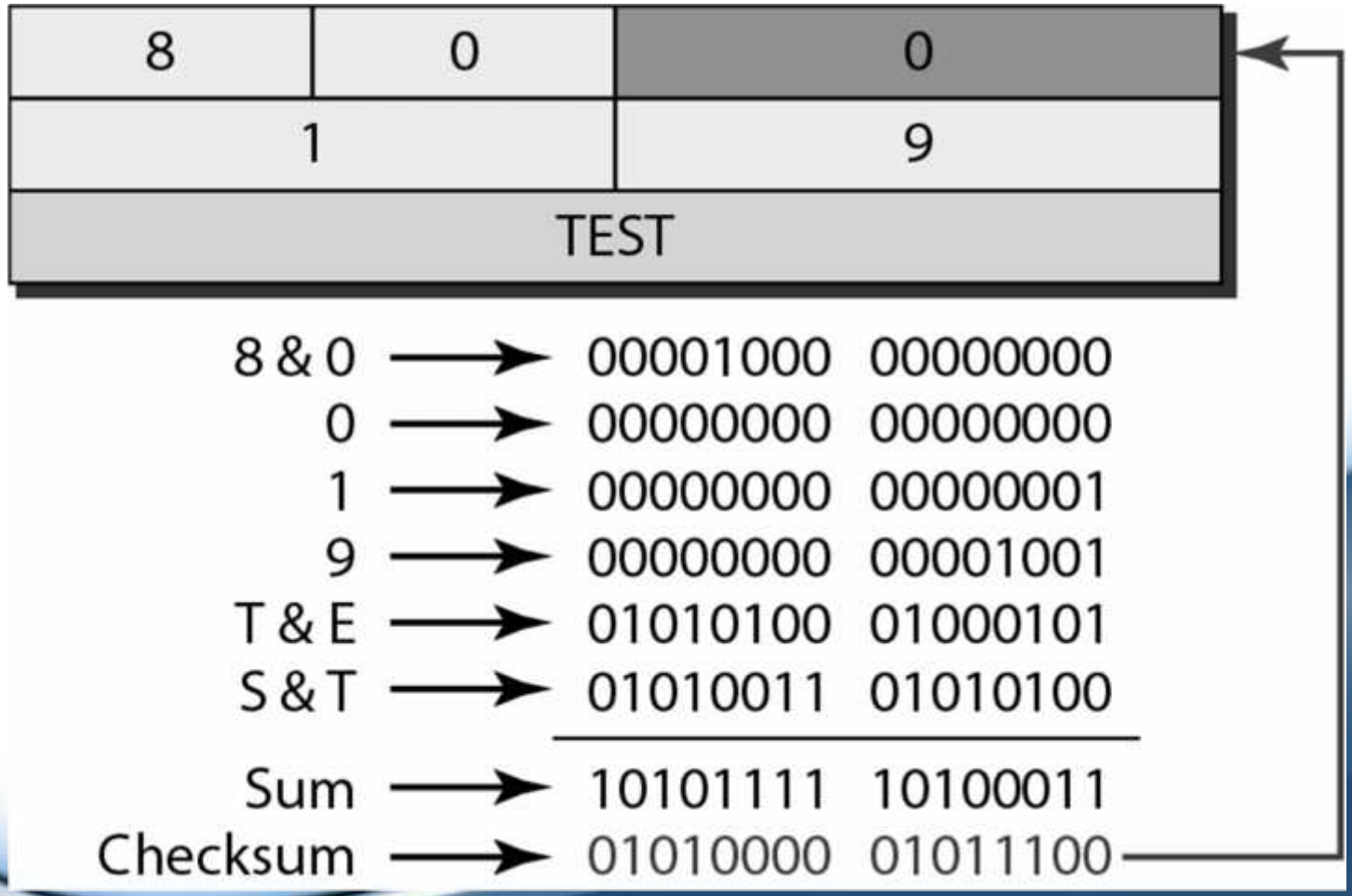
# Router Solicitation and Advertisement:

- As we discussed in the redirection message section, a host that wants to send data to a host on another network needs to know the address of routers connected to its own network.
- Also, the host must know if the routers are alive and functioning. The router-solicitation and router-advertisement messages can help in this situation.
- A host can broadcast (or multicast) a router-solicitation message. The router or routers that receive the solicitation message broadcast their routing information using the router-advertisement message.
- A router can also periodically send router-advertisement messages even if no host has solicited. Note that when a router sends out an advertisement, it announces not only its own presence but also the presence of all routers on the network of which it is aware.

# Checksum

- We learned the concept and idea of the checksum. In ICMP the checksum is calculated over the entire message (header and data).
- Figure shows an example of checksum calculation for a simple echo-request message. We randomly chose the identifier to be 1 and the sequence number to be 9.
- The message is divided into 16-bit (2-byte) words. The words are added and the sum is complemented.
- Now the sender can put this value in the checksum field.

# Checksum


8
0
0
1
9
TEST
8 & 0 → 00001000 00000000
0 → 00000000 00000000
1 → 00000000 00000001
9 → 00000000 00001001
T & E → 01010100 01000101
S & T → 01010011 01010100
Sum → 10101111 10100011
Checksum → 01010000 01011100

# Debugging Tools

- There are several tools that can be used in the Internet for debugging.
- We can determine the viability of a host or router. We can trace the route of a packet.
- We introduce two tools that use ICMP for debugging: *ping* and *trace route*

# Ping

- We can use the *ping* program to find if a host is alive and responding. We use *ping* here to see how it uses ICMP packets. The source host sends ICMP echo-request messages (type: 8, code: 0); the destination, if alive, responds with ICMP echo-reply messages.
- The *ping* program sets the identifier field in the echo-request and echo-reply message and starts the sequence number from 0; this number is incremented by 1 each time a new message is sent.
- Note that *ping* can calculate the round-trip time. It inserts the sending time in the data section of the message. When the packet arrives, it subtracts the arrival time from the departure time to get the round-trip time (RTT). We use the *ping* program to test the server fhda.edu. The result is shown below:

```
$ ping thda.edu
```

# Ping

- The *ping* program sends messages with sequence numbers starting from O. For each probe it gives us the RTT time. The TTL (time to live) field in the IP datagram that encapsulates an ICMP message has been set to 62, which means the packet cannot travel more than 62 hops.
- At the beginning, *ping* defines the number of data bytes as 56 and the total number of bytes as 84. It is obvious that if we add 8 bytes of ICMP header and 20 bytes of IP header to 56, the result is 84.
- However, note that in each probe *ping* defines the number of bytes as 64. This is the total number of bytes in the ICMP packet (56 + 8). The *ping* program continues to send messages, if we do not stop it by using the interrupt key (ctrl + c, for example).
- After it is interrupted, it prints the statistics of the probes. It tells us the number of packets sent, the number of packets received, the total time, and the RTT minimum, maximum, and average. Some systems may print more information.

# Ping

```
$ ping fhda.edu
PING fhda.edu (153.18.8.1)    56 (84)  bytes of data.
64 bytes from tiptoe.fhda.edu (153.18.8.1): icmp_seq=0      ttl=62      time=1.91 ms
64 bytes from tiptoe.fhda.edu (153.18.8.1): icmp_seq=1      ttl=62      time=2.04 ms
64 bytes from tiptoe.fhda.edu (153.18.8.1): icmp_seq=2      ttl=62      time=1.90 ms
64 bytes from tiptoe.fhda.edu (153.18.8.1): icmp_seq=3      ttl=62      time=1.97 ms
64 bytes from tiptoe.fhda.edu (153.18.8.1): icmp_seq=4      ttl=62      time=1.93 ms
64 bytes from tiptoe.fhda.edu (153.18.8.1): icmp_seq=5      ttl=62      time=2.00 ms
64 bytes from tiptoe.fhda.edu (153.18.8.1): icmp_seq=6      ttl=62      time=1.94 ms
64 bytes from tiptoe.fhda.edu (153.18.8.1): icmp_seq=7      ttl=62      time=1.94 ms
64 bytes from tiptoe.fhda.edu (153.18.8.1): icmp_seq=8      ttl=62      time=1.97 ms
64 bytes from tiptoe.fhda.edu (153.18.8.1): icmp_seq=9      ttl=62      time=1.89 ms
64 bytes from tiptoe.fhda.edu (153.18.8.1): icmp_seq=10     ttl=62      time=1.98 ms

--- fhda.edu ping statistics ---
11 packets transmitted, 11 received, 0% packet loss, time 10103ms
      rtt min/avg/max = 1.899/1.955/2.041 ms
```

# Traceroute:

- The *traceroute* program in UNIX or *tracert* in Windows can be used to trace the route of a packet from the source to the destination. We have seen an application of the *traceroute* program to simulate the loose source route and strict source route options of an IP datagram.

- We use this program in conjunction with ICMP packets in this chapter. The program elegantly uses two ICMP messages, time exceeded and destination unreachable, to find the route of a packet. This is a program at the application level that uses the services of UDP. Let us show the idea of the *traceroute* program

# Traceroute:

- Please Refer to Notes

# Thank You

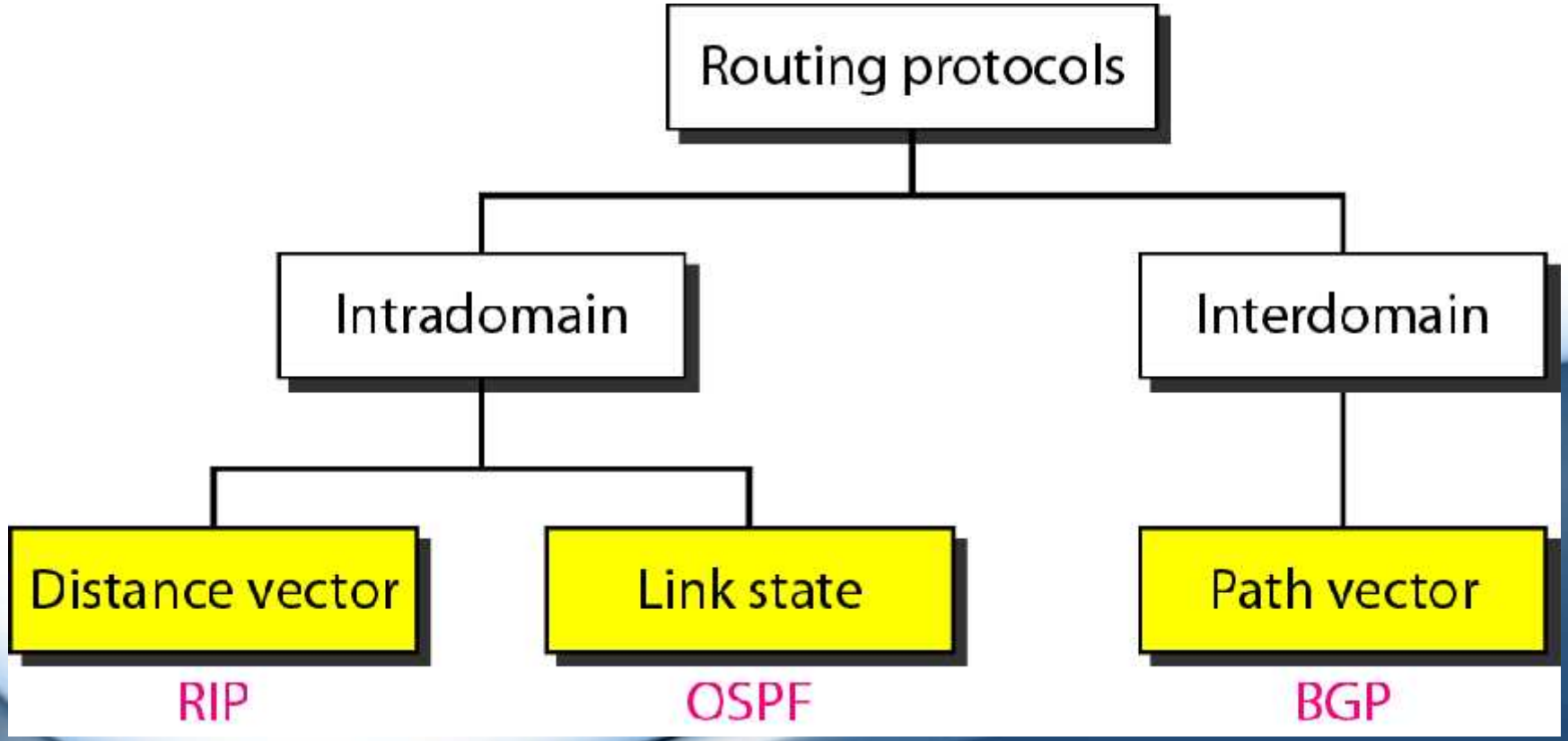
# Unicast Routing Protocols

# Unicast Routing Protocols

Topics Covered:

1. Routing Protocols
2. Intra- and Interdomain Routing
3. Distance Vector Routing
4. RIP

# Routing Protocols



# Intra- and Interdomain Routing :

- Today, an internet can be so large that one routing protocol cannot handle the task of updating the routing tables of all routers.
- For this reason, an internet is divided into autonomous systems.
- An autonomous system (AS) is a group of networks and routers under the authority of a single administration.
  - Routing inside an autonomous system is referred to as intradomain routing.
  - Routing between autonomous systems is referred to as interdomain routing.
- Each autonomous system can choose one or more intradomainrouting protocols to handle routing inside the autonomous system.
- However, only one interdomain routing protocol handles routing between autonomous systems.

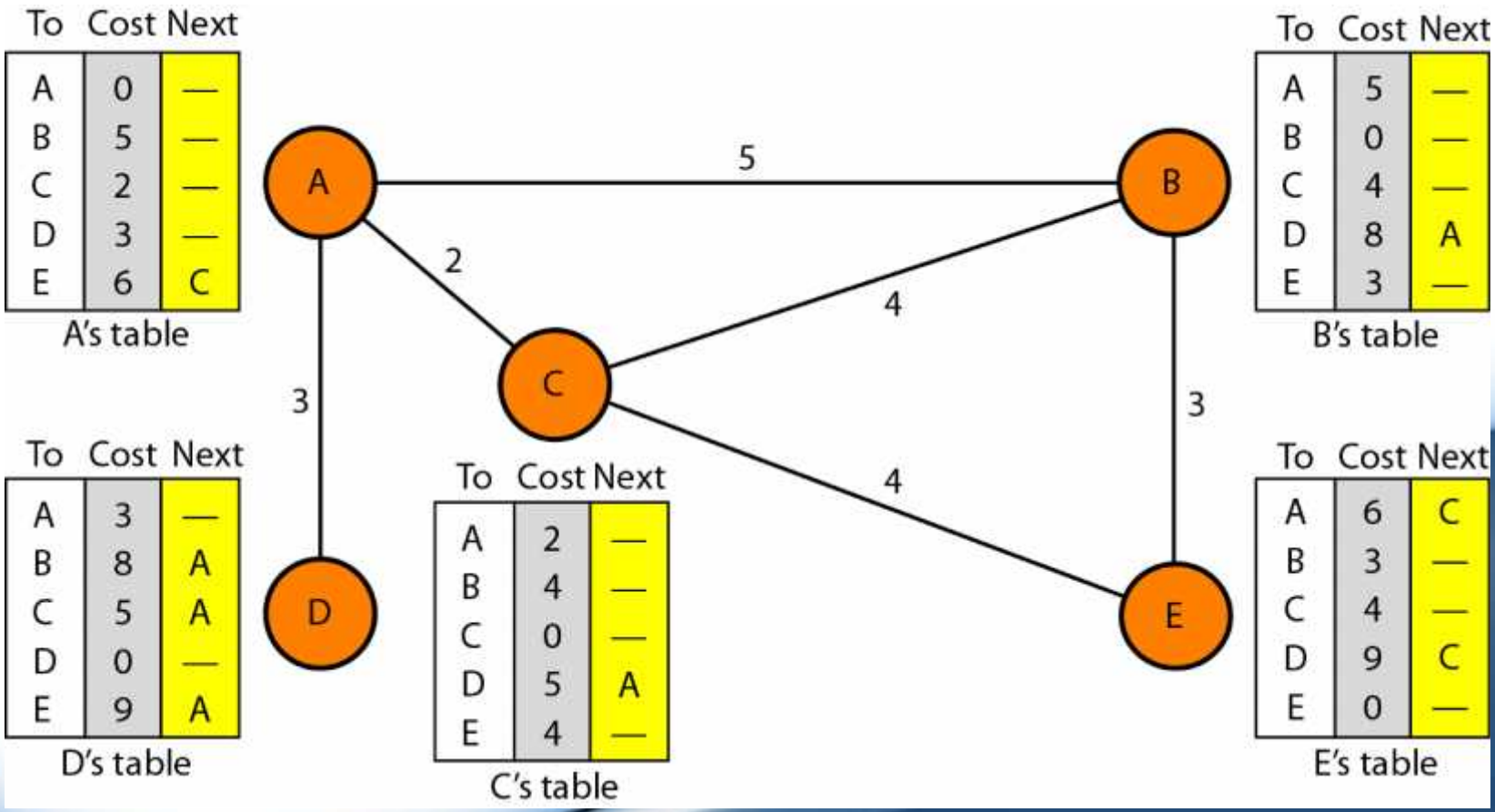
# Distance Vector Routing :

- In distance vector routing, the least-cost route between any two nodes is the route with minimum distance.

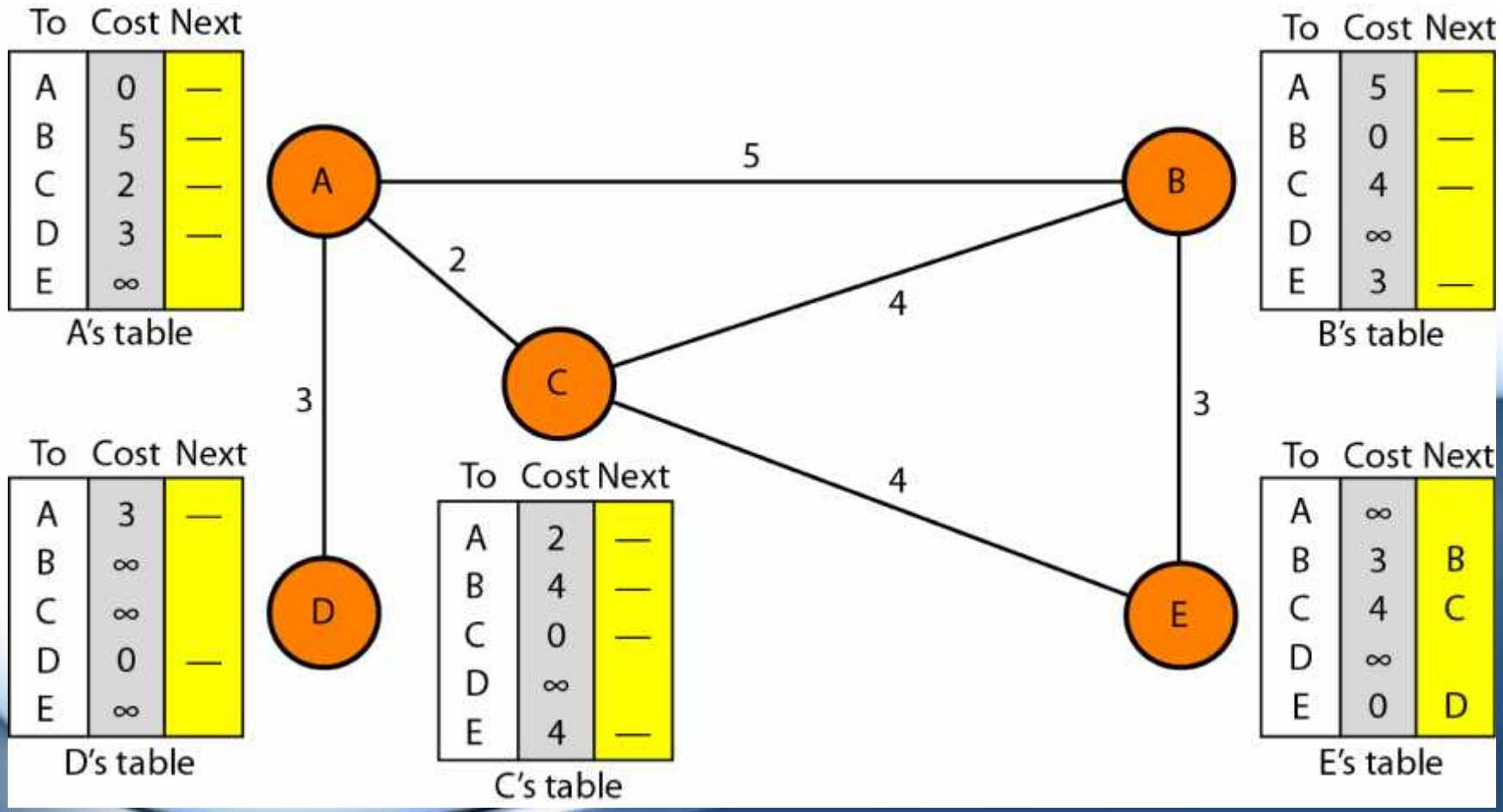
- In this protocol, as the name implies, each node maintains a vector (table) of minimum distances to every node. The table at each node also guides the packets to the desired node by showing the next stop in the route (next-hop routing).

- We can think of nodes as the cities in an area and the lines as the roads connecting them. A table can show a tourist the minimum distance between cities. In Figure 22.14, we show a system of five nodes with their corresponding tables

# Distance Vector Routing tables:



# Initialization



# Sharing :

- The whole idea of distance vector routing is the sharing of information between neighbors. Although node A does not know about node E, node C does. So if node C shares its routing table with A, node A can also know how to reach node E.
- On the other hand, node C does not know how to reach node D, but node A does. If node A shares its routing table with node C, node C also knows how to reach node D. In other words, nodes A and C, as immediate neighbors, can improve their routing tables if they help each other.
- There is only one problem. How much of the table must be shared with each neighbor? A node is not aware of a neighbor's table.
- The best solution for each node is to send its entire table to the neighbor and let the neighbor decide what part to use and what part to discard. However, the third column of a table (next stop) is not useful for the neighbor.

# Sharing :

- When the neighbor receives a table, this column needs to be replaced with the sender's name. If any of the rows can be used, the next node is the sender of the table.

- A node therefore can send only the first two columns of its table to any neighbor. In other words, sharing here means sharing only the first two columns. In distance vector routing, each node shares its routing table with its immediate neighbors periodically and when there is a change.

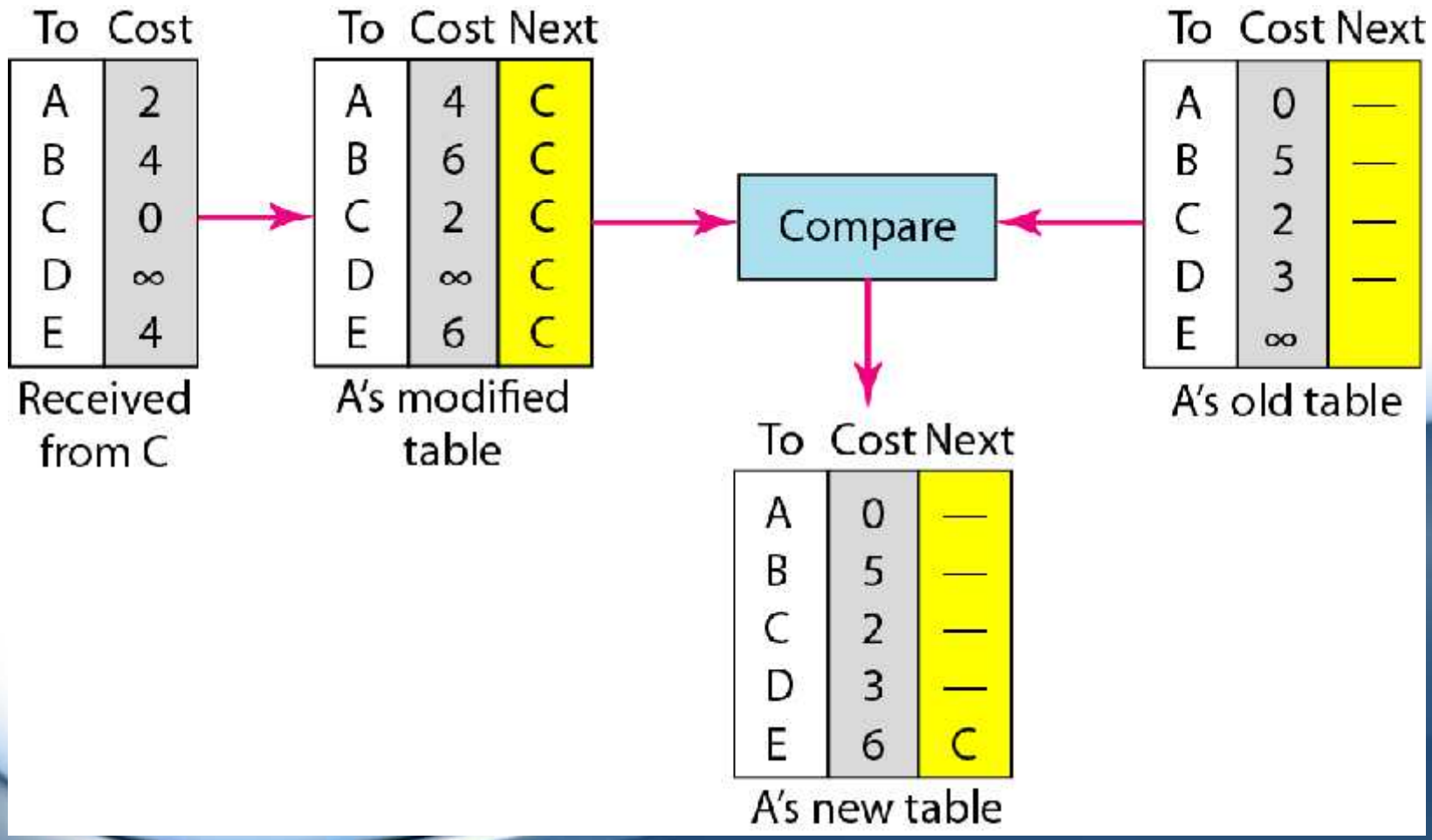
# Updating :

- When a node receives a two-column table from a neighbor, it needs to update its routing table. Updating takes three steps:
- The receiving node needs to add the cost between itself and the sending node to each value in the second column. The logic is clear. If node C claims that its distance to a destination is *x* mi, and the distance between A and C is *y* mi, then the distance between A and that destination, via C, is *x* + *y* mi.
- The receiving node needs to add the name of the sending node to each row as the third column if the receiving node uses information from any row. The sending node is the next node in the route.
- The receiving node needs to compare each row of its old table with the corresponding row of the modified version of the received table.
  - If the next-node entry is different, the receiving node chooses the row with the smaller cost. If there is a tie, the old one is kept.
  - If the next-node entry is the same, the receiving node chooses the new row. For example, suppose node C has previously advertised a route to node X with distance

# Updating :

- Suppose that now there is no path between C and X; node C now advertises this route with a distance of infinity.
- Node A must not ignore this value even though its old entry is smaller.
- The old route does not exist any more.
- The new route has a distance of infinity.

# Updating


To Cost
A 2
B 4
C 0
D ∞
E 4
Received from C
To Cost Next
A 4 C
B 6 C
C 2 C
D ∞ C
E 6 C
A's modified table
Compare
To Cost Next
A 0 —
B 5 —
C 2 —
D 3 —
E ∞
A's old table
To Cost Next
A 0 —
B 5 —
C 2 —
D 3 —
E 6 C
A's new table

# Important Points

- There are several points we need to emphasize here.
  - First, as we know from mathematics, when we add any number to infinity, the result is still infinity.
  - Second, the modified table shows how to reach A from A via C. If A needs to reach itself via C, it needs to go to C and come back, a distance of 4.
  - Third, the only benefit from this updating of node A is the last entry, how to reach E. Previously, node A did not know how to reach E (distance of infinity); now it knows that the cost is 6 via C.

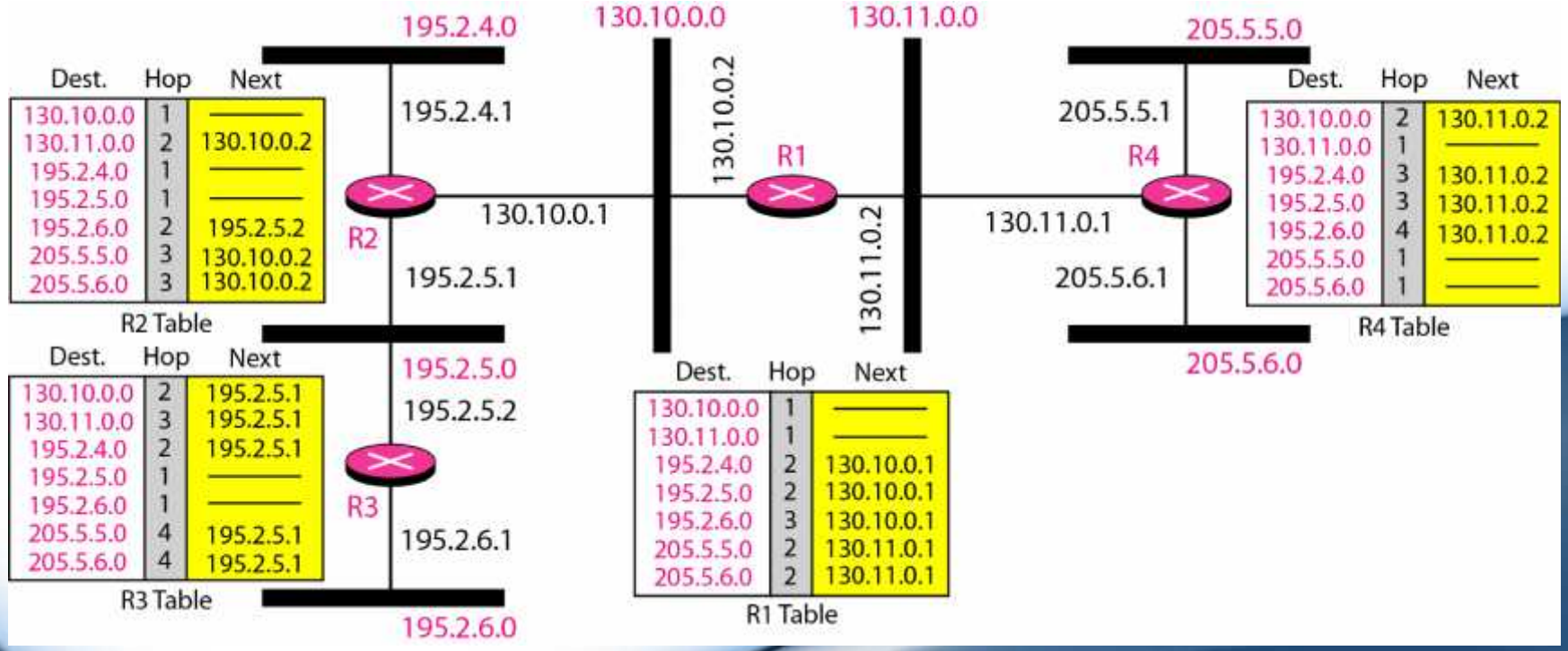
# When to Share :

- When does a node send its partial routing table (only two columns) to all its immediate neighbors? The table is sent both periodically and when there is a change in the table.
  - *Periodic Update* A node sends its routing table, normally every 30 s, in a periodic update. The period depends on the protocol that is using distance vector routing.
  - *Triggered Update* A node sends its two-column routing table to its neighbors anytime there is a change in its routing table. This is called a triggered update. The change can result from the following.
    - A node receives a table from a neighbor, resulting in changes in its own table after updating.
    - A node detects some failure in the neighboring links which results in a distance change to infinity.

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

# A Sample of RIP



# Explanation of Sample

- The table of each router is also shown. Let us look at the routing table for RI.
- The table has seven entries to show how to reach each network in the autonomous system.
- Router RI is directly connected to networks 130.10.0.0 and 130.11.0.0, which means that there are no next-hop entries for these two networks.
- To send a packet to one of the three networks at the far left, router RI needs to deliver the packet to R2.
- The next-node entry for these three networks is the interface of router R2 with IP address 130.10.0.1.
- To send a packet to the two networks at the far right, router RI needs to send the packet to the interface of router R4 with IP address 130.11.0.1.

# Thank You

# Link State Routing and OSPF

# Link State Routing and OSPF

Topics Covered:

1. Link State Routing :
2. OSPF :
    1. Areas
    2. Types of Links

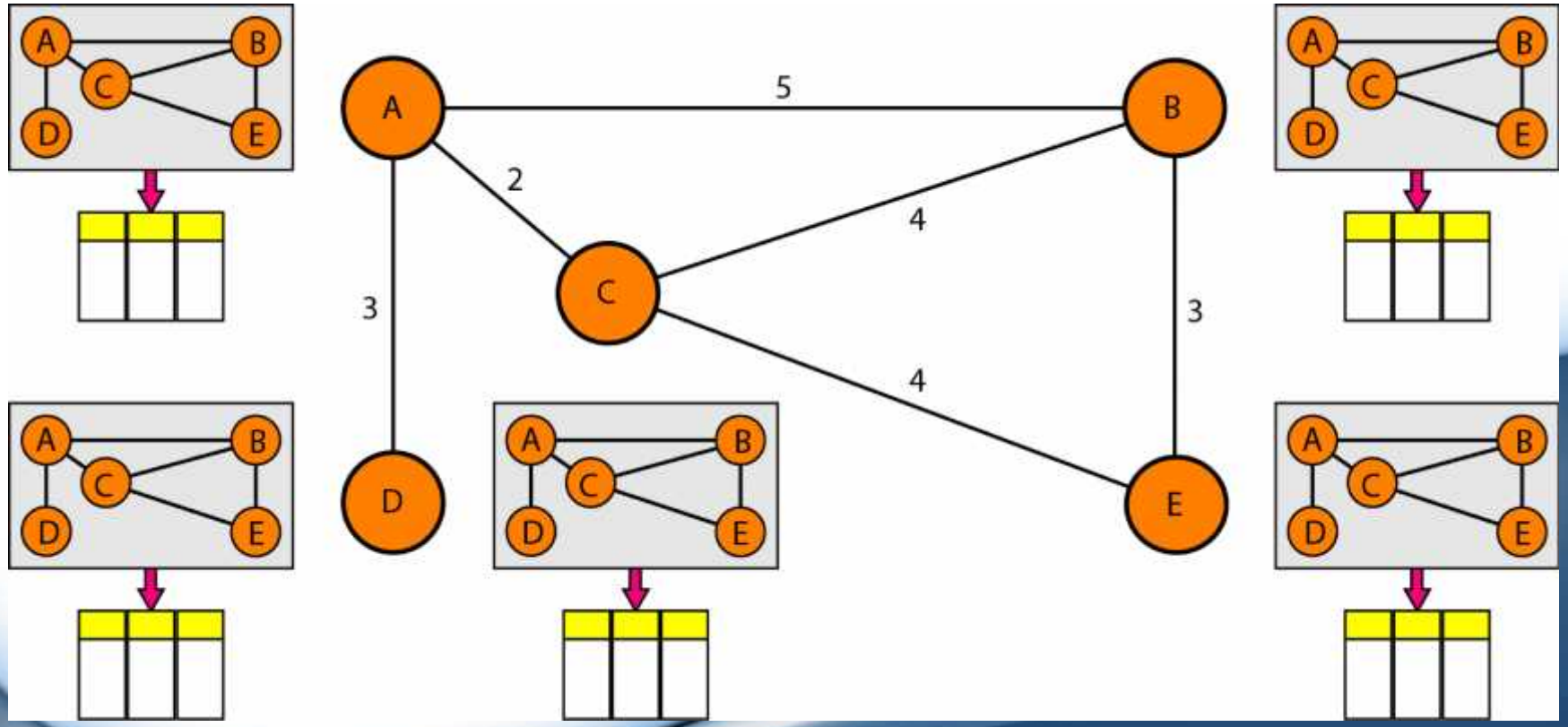
# Link State Routing :

- Link state routing has a different philosophy from that of distance vector routing.

- In link state routing, if each node in the domain has the entire topology of the domain
  - the list of nodes and links
  - how they are connected including the type
  - cost (metric)
  - condition of the links (up or down)

- The node can use Dijkstra's algorithm to build a routing table

# Link State Routing



# Link State Routing

- The figure shows a simple domain with five nodes. Each node uses the same topology to create a routing table, but the routing table for each node is unique because the calculations are based on different interpretations of the topology.

- The topology must be dynamic, representing the latest state of each node and each link. If there are changes in any point in the network (a link is down, for example), the topology must be updated for each node.

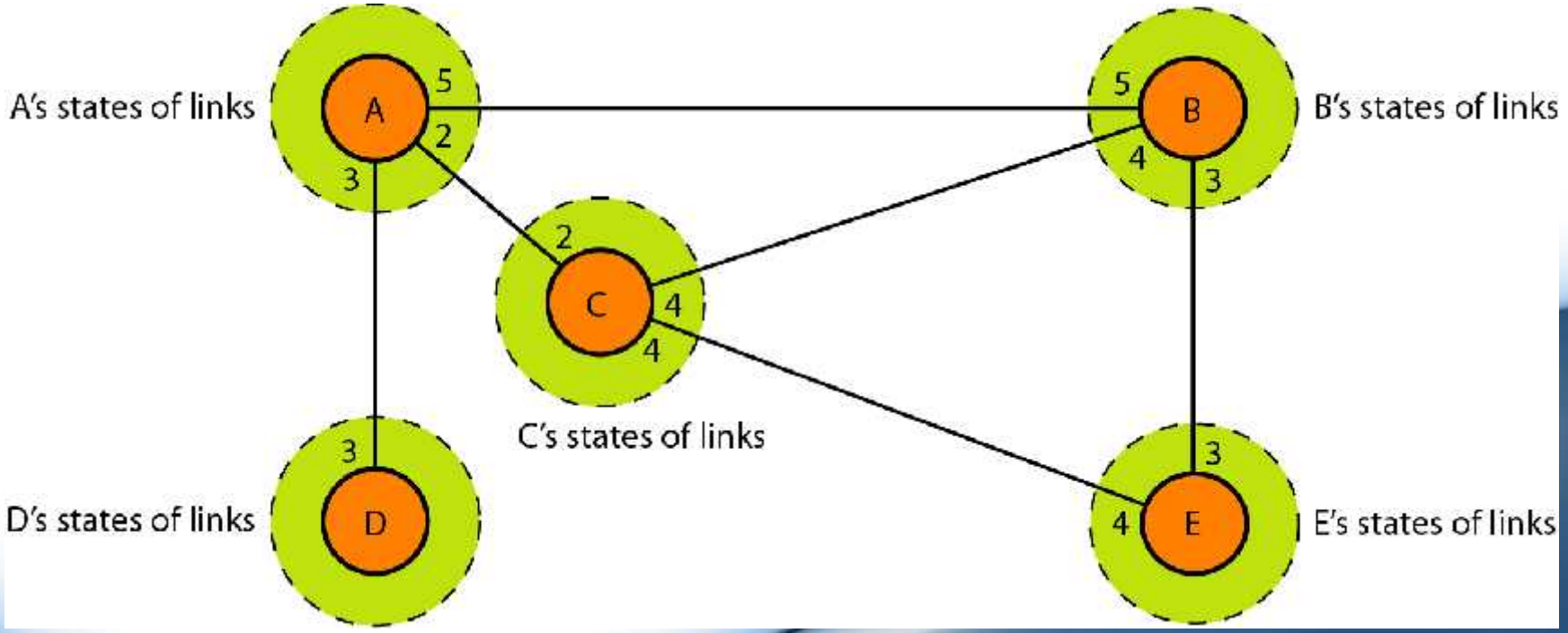
- How can a common topology be dynamic and stored in each node? No node can know the topology at the beginning or after a change somewhere in the network.

# Link State Routing

- Link state routing is based on the assumption that, although the global knowledge about the topology is not clear, each node has partial knowledge: it knows the state (type, condition, and cost) of its links.

- In other words, the whole topology can be compiled from the partial knowledge of each node.

# Link State Routing



# Building Routing Tables :

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

# OSPF :

- The Open Shortest Path First or OSPF protocol is an intradomain routing protocol based on link state routing. Its domain is also an autonomous system.
  - What is an Area?
  - What is a Metric?
  - What is a Link?

# Areas

- To handle routing efficiently and in a timely manner, OSPF divides an autonomous system into areas. An area is a collection of networks, hosts, and routers all contained within an autonomous system.

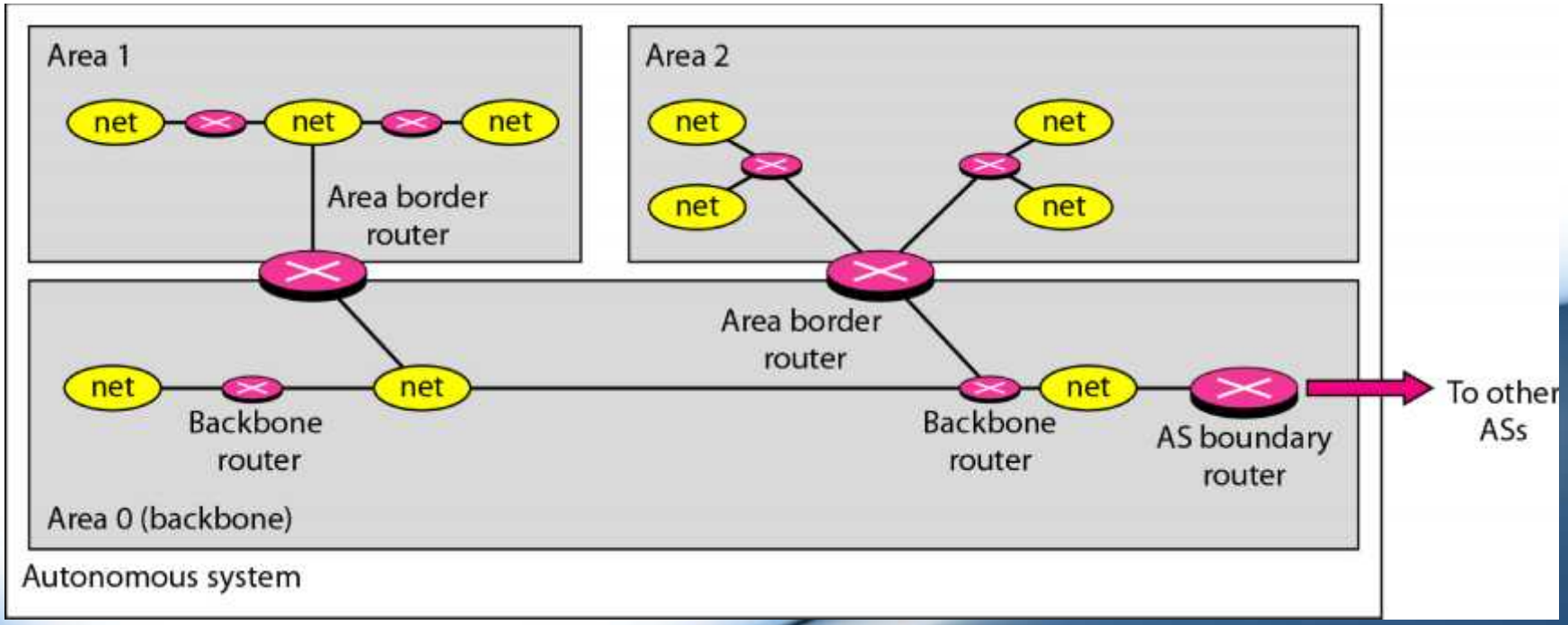
- An autonomous system can be divided into many different areas. All networks inside an area must be connected.
  - Routers inside an area flood the area with routing information.
  - At the border of an area, special routers called area border routers summarize the information about the area and send it to other areas.

# Areas

- Among the areas inside an autonomous system is a special area called the *backbone;* all the areas inside an autonomous system must be connected to the backbone. In other words, the backbone serves as a primary area and the other areas as secondary areas. This does not mean that the routers within areas cannot be connected to each other, however.
- The routers inside the backbone are called the backbone routers. Note that a backbone router can also be an area border router.

- If, because of some problem, the connectivity between a backbone and an area is broken, a virtual link between routers must be created by an administrator to allow continuity of the functions of the backbone as the primary area. Each area has an area identification.

# Areas

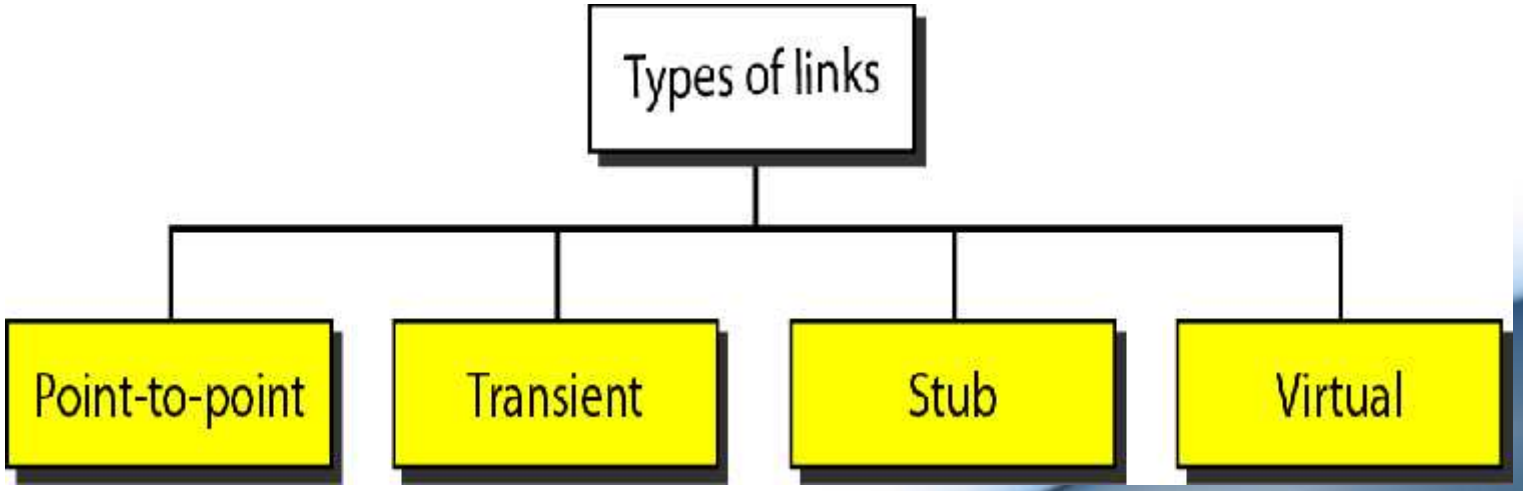

Area 1
net
net
net
Area border router
Area 2
net
net
net
net
Area border router
net
Backbone router
net
Backbone router
net
AS boundary router
To other ASs
Area 0 (backbone)
Autonomous system

# Metric:

- The OSPF protocol allows the administrator to assign a cost, called the metric, to each route.
- The metric can be based on a type of service
  - minimum delay
  - maximum throughput
- As a matter of fact, a router can have multiple routing tables, each based on a different type of service.

# Types of Links

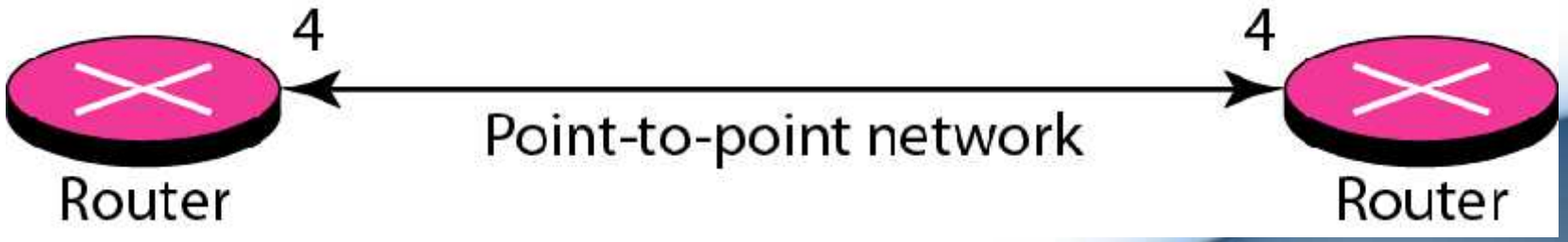
- In OSPF terminology, a connection is called a *link.* Four types of links have been defined: point-to-point, transient, stub, and virtual



# Point - to - Point

- A point-to-point link connects two routers without any other host or router in between. In other words, the purpose of the link (network) is just to connect the two routers.
- An example of this type of link is two routers connected by a telephone line or a T line. There is no need to assign a network address to this type of link.
- Graphically, the routers are represented by nodes, and the link is represented by a bidirectional edge connecting the nodes.
- The metrics, which are usually the same, are shown at the two ends, one for each direction. In other words, each router has only one neighbor at the other side of the link

# Point - to - Point



# A transient link

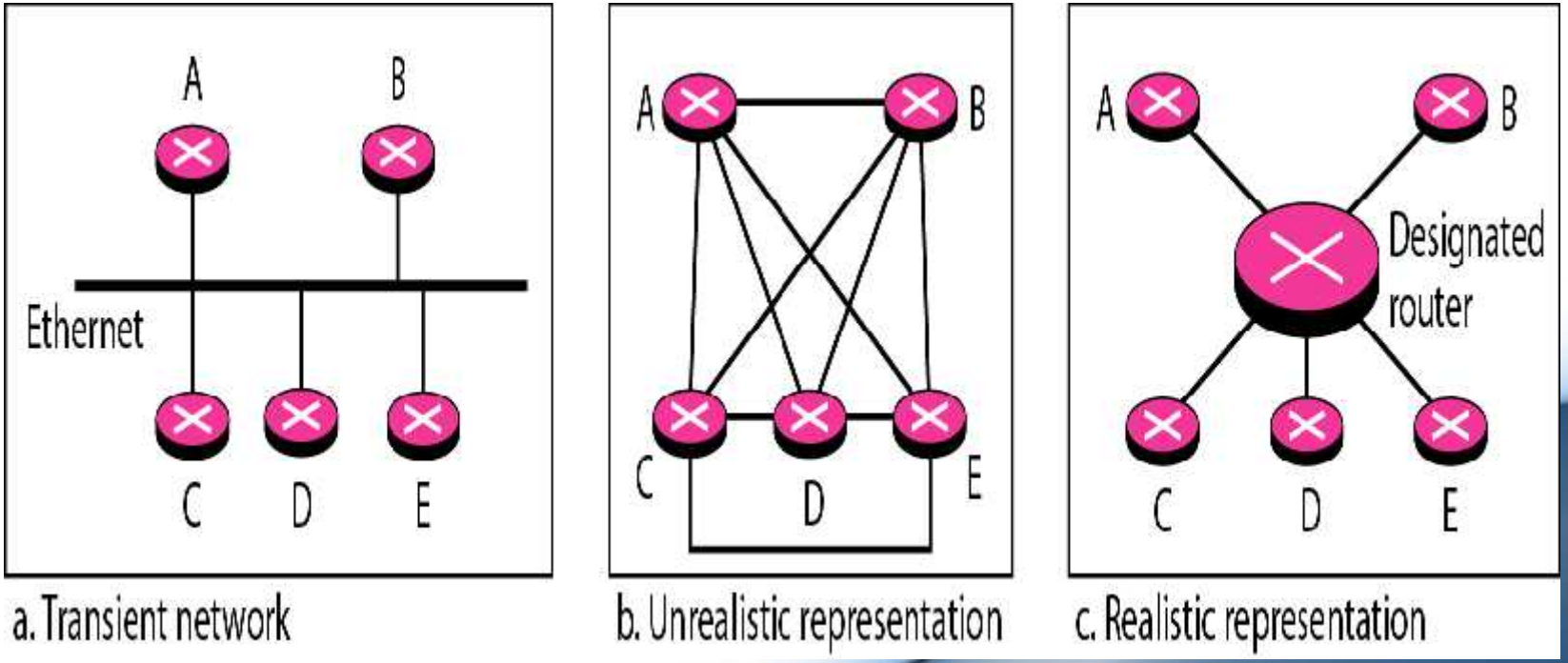
- A transient link is a network with several routers attached to it. The data can enter through any of the routers and leave through any router. All LANs and some WANs with two or more routers are of this type. In this case, each router has many neighbors.
- This is neither efficient nor realistic.
  - It is not efficient because each router needs to advertise the neighborhood to four other routers, for a total of 20 advertisements.
  - It is not realistic because there is no single network (link) between each pair of routers; there is only one network that serves as a crossroad between all five routers.
- To show that each router is connected to every other router through one single network, the network itself is represented by a node. However, because a network is not a machine, it cannot function as a router. One of the routers in the network takes this responsibility. It is assigned a dual purpose; it is a true router and a designated router.

# A transient link

- Now each router has only one neighbor, the designated router (network). On the other hand, the designated router (the network) has five neighbors. We see that the number of neighbor announcements is reduced from 20 to 10. Still, the link is represented as a bidirectional edge between the nodes.

- However, while there is a metric from each node to the designated router, there is no metric from the designated router to any other node. The reason is that the designated router represents the network. We can only assign a cost to a packet that is passing through the network. We cannot charge for this twice. When a packet enters a network, we assign a cost; when a packet leaves the network to go to the router, there is no charge.

# A transient link




a. Transient network

b. Unrealistic representation

c. Realistic representation

# A stub link

- A 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. This is a special case of the transient network.

- We can show this situation using the router as a node and using the designated router for the network. However, the link is only one-directional, from the router to the network (see Figure 22.28).

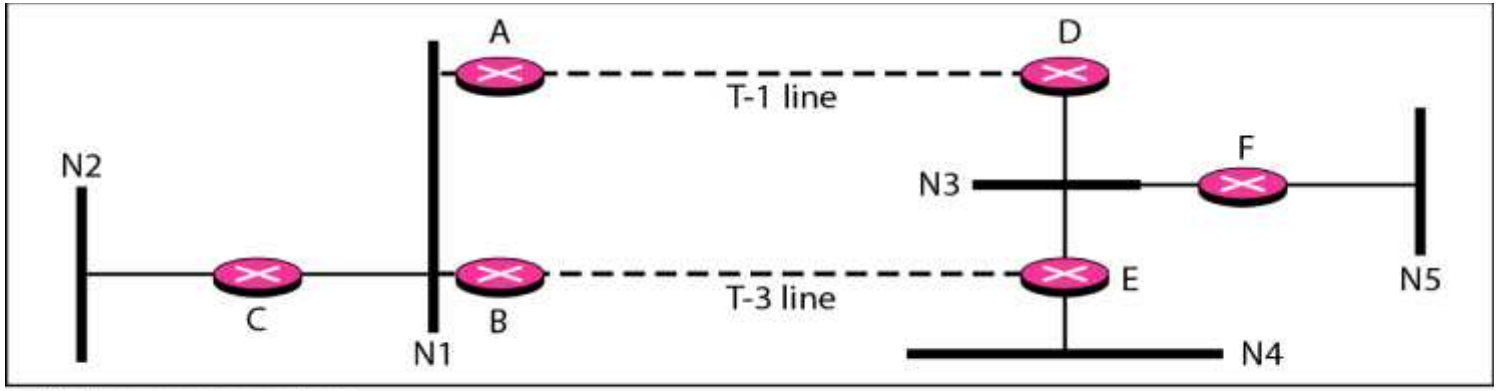
- When the link between two routers is broken, the administration may create a virtual link between them, using a longer path that probably goes through several routers.

# A stub link

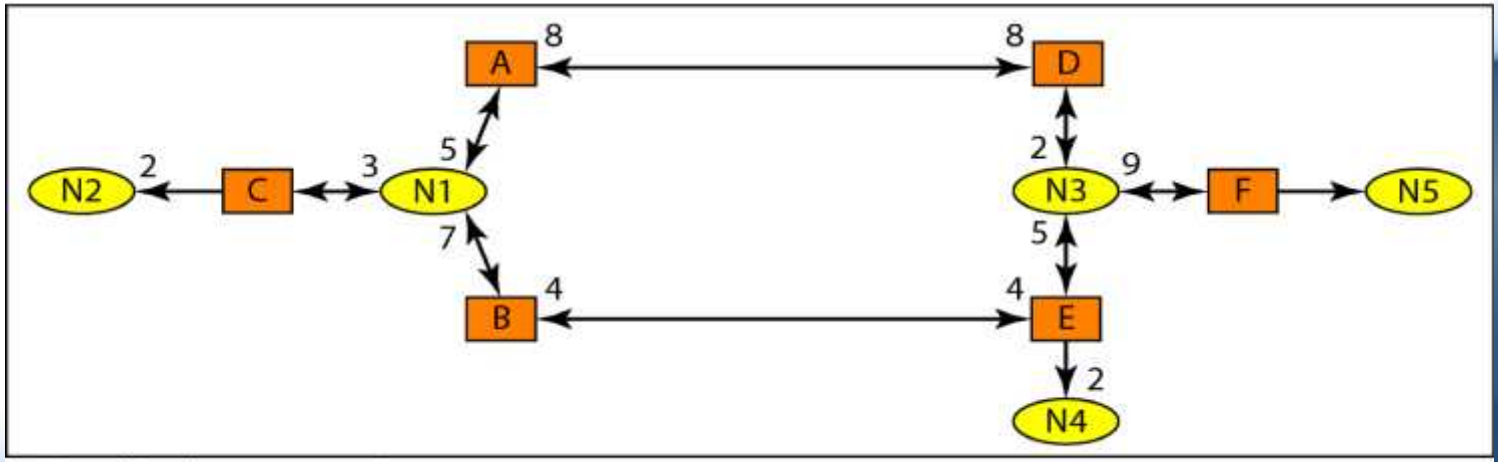
- **Graphical Representation:** Let us now examine how an AS can be represented graphically. Figure 22.29 shows a small AS with seven networks and six routers.

- Two of the networks are point-to-point networks. We use symbols such as Nl and N2 for transient and stub networks. There is no need to assign an identity to a point-to-point network. The figure also shows the graphical representation of the AS as seen by OSPF. We have used square nodes for the routers and ovals for the networks (represented by designated routers). However, OSPF sees both as nodes. Note that we have three stub networks.

# A Stub Link




a. Autonomous system




b. Graphical representation

# Thank You

# Path Vector Routing and BGP

# Path Vector Routing and BGP

Topics Covered:

1. Path Vector Routing
2. BGP
    1. Types of AS
    2. Path Attributes
    3. BGP Sessions

# Path Vector Routing

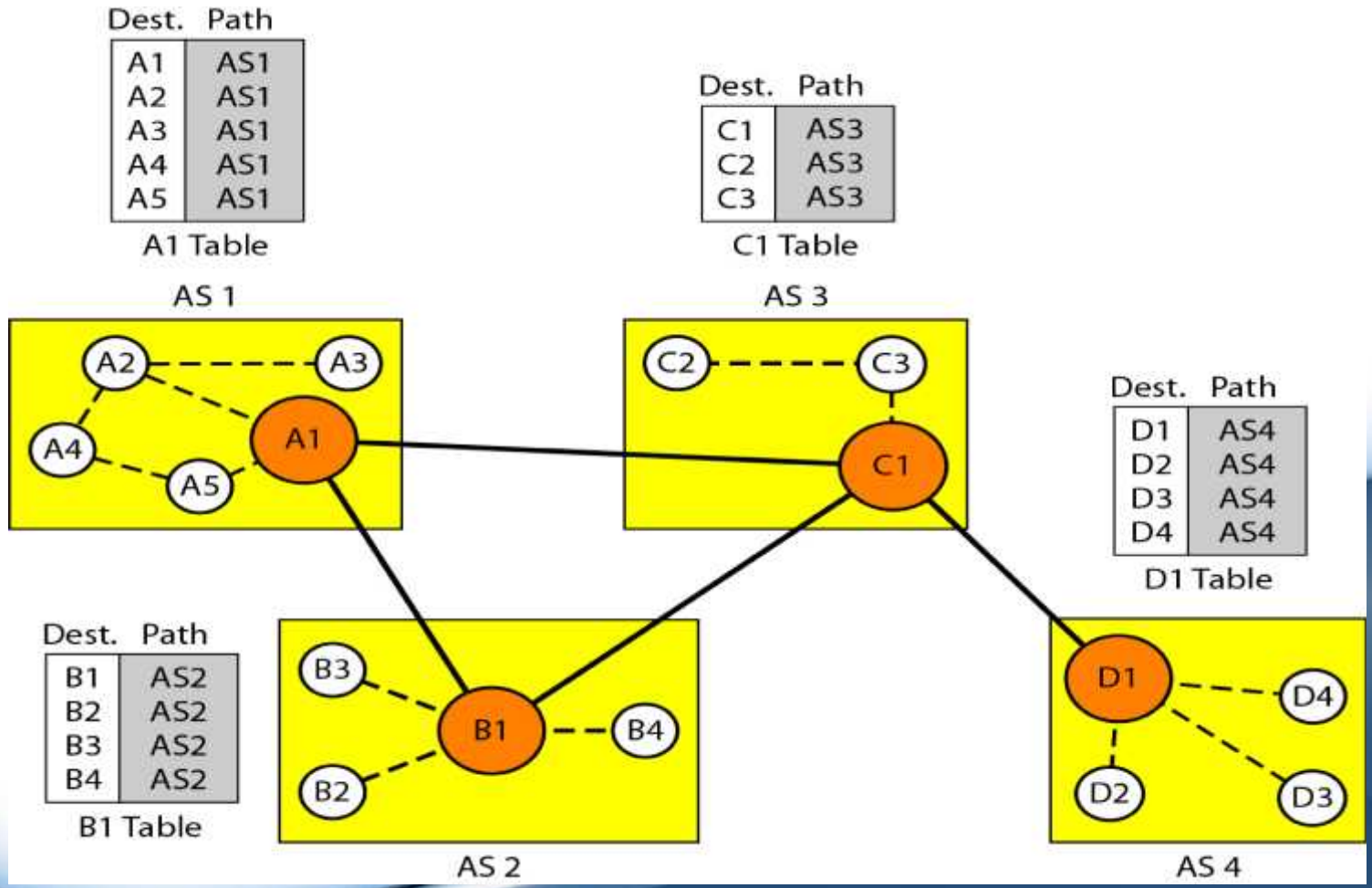
- Distance vector and link state routing are both intradomain routing protocols.
- They can be used inside an autonomous system, but not between autonomous systems. These two protocols are not suitable for interdomain routing mostly because of scalability.
- Distance vector routing is subject to instability if there are more than a few hops in the domain of operation.
- Link state routing needs a huge amount of resources to calculate routing tables. It also creates heavy traffic because of flooding.
- There is a need for a third routing protocol which we call path vector routing.

# Initialization :

- At the beginning, each speaker node can know only the reachability of nodes inside its autonomous system. Figure 22.30 shows the initial tables for each speaker node in a system made of four ASs.

- Node Al is the speaker node for ASl, Bl for AS2, Cl for AS3, and Dl for AS4. Node Al creates an initial table that shows Al to A5 are located in ASI and can be reached through it. Node Bl advertises that Bl to B4 are located in AS2 and can be reached through Bl. And so on.

# Initialization



# Sharing

- Just as in distance vector routing, in path vector routing, a speaker in an autonomous system shares its table with immediate neighbors.

- In Figure 22.30, node Al shares its table with nodes Bl and Cl. Node Cl shares its table with nodes Dl, Bl, andAl. Node Bl shares its table with Cl andAl. Node Dl shares its table with Cl.

# Updating

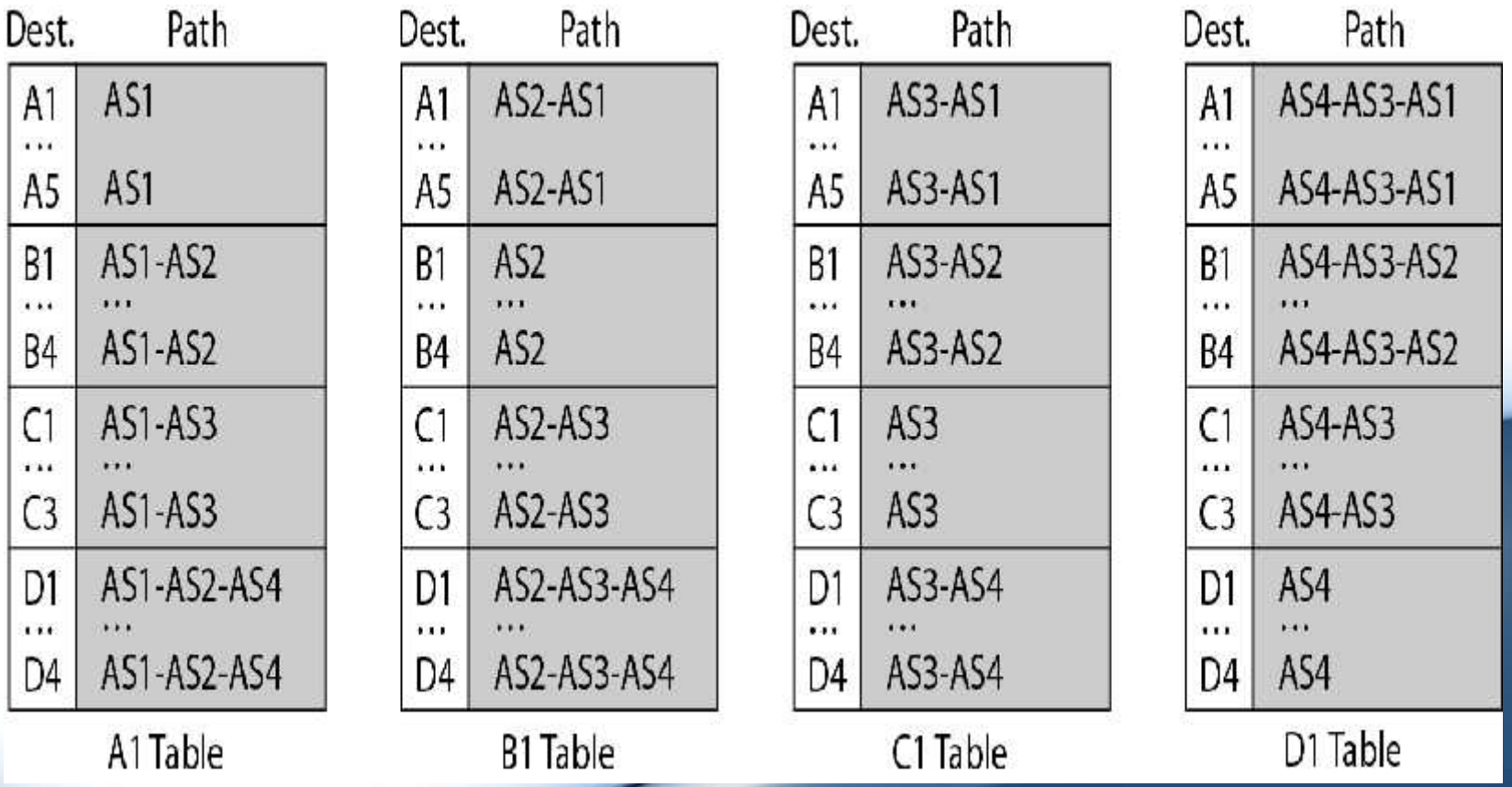
- When a speaker node receives a two-column table from a neighbor, it updates its own table by adding the nodes that are not in its routing table and adding its own autonomous system and the autonomous system that sent the table. After a while each speaker has a table and knows how to reach each node in other ASs. Figure 22.31 shows the tables for each speaker node after the system is stabilized.

- According to the figure, if router Al receives a packet for nodes A3, it knows that the path is in ASI (the packet is at home); but if it receives a packet for Dl, it knows that the packet should go from ASl, to AS2, and then to AS3. The routing table shows the path completely. On the other hand, if node Dl in AS4 receives a packet for node A2, it knows it should go through AS4, AS3, and AS 1.

# Updating

- **Loop prevention.**
  - The instability of distance vector routing and the 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.**
  - Policy routing can be easily implemented through path vector routing. When a router receives a message, it can check the path. If one of the autonomous systems listed in the path is against its policy, it can ignore that path and that destination. It does not update its routing table with this path, and it does not send this message to its neighbors.

# Updating



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

A1 Table

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

B1 Table

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

C1 Table

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

D1 Table

# 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.

# Types of AS

- **Stub AS:**
  - A stub AS has only one connection to another AS. The interdomain data traffic in a stub AS can be either created or terminated in the AS. The hosts in the AS can send data traffic to other ASs. The hosts in the AS can receive data coming from hosts in other ASs. Data traffic, however, cannot pass through a stub AS. A stub AS is either a source or a sink. A good example of a stub AS is a small corporation or a small local ISP.

# Types of AS

- **Multihomed AS:**
  - A multihomed AS has more than one connection to other ASs, but it is still only a source or sink for data traffic. It can receive data traffic from more than one AS. It can send data traffic to more than one AS, but there is no transient traffic. It does not allow data coming from one AS and going to another AS to pass through. A good example of a multihomed AS is a large corporation that is connected to more than one regional or national AS that does not allow transient traffic.

- **Transit AS:**
  - A transit AS is a multihomed AS that also allows transient traffic. Good examples of transit ASs are national and international ISPs (Internet backbones).

# Path Attributes

- Attributes are divided into two broad categories:
  - Well Known
  - Optional

# Well Known

- A well known attribute is one that every BGP router must recognize. An optional attribute is one that needs not be recognized by every router.
- Well-known attributes are themselves divided into two categories: mandatory and discretionary.
  - A *well-known mandatory attribute* is one that must appear in the description of a route.
  - A *well-known discretionary attribute* is one that must be recognized by each router, but is not required to be included in every update message.
  - One wellknown mandatory attribute is ORIGIN. This defines the source of the routing information (RIP, OSPF, and so on).

# Well Known

- Another well-known mandatory attribute is AS_PATH. This defines the list of autonomous systems through which the destination can be reached. Still another well-known mandatory attribute is NEXT-HOP, which defines the next router to which the data packet should be sent.

# Optional

- The optional attributes can also be subdivided into two categories:
  - transitive and
  - non-transitive.

- An *optional transitive attribute* is one that must be passed to the next router by the router that has not implemented this attribute.
- An *optional nontransitive attribute* is one that must be discarded if the receiving router has not implemented it.

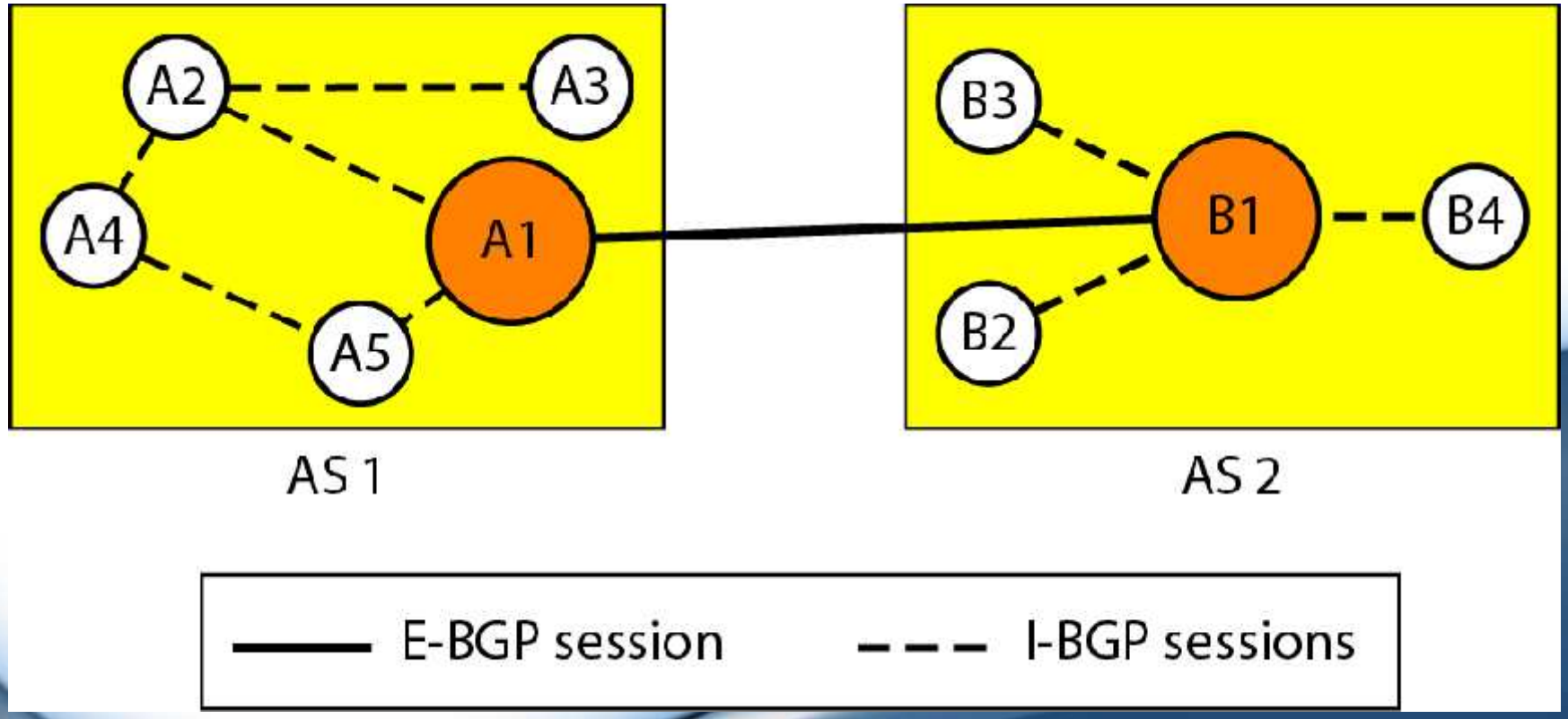
# BGP Sessions

- The exchange of routing information between two routers using BGP takes place in a session. A session is a connection that is established between two BGP routers only for the sake of exchanging routing information.
- To create a reliable environment, BGP uses the services of TCP. In other words, a session at the BGP level, as an application program, is a connection at the TCP level.
- However, there is a subtle difference between a connection in TCP made for BGP and other application programs. When a TCP connection is created for BGP, it can last for a long time, until something unusual happens.
- For this reason, BGP sessions are sometimes referred to as *semipermanent connections.*

# External and Internal BGP

- If we want to be precise, BGP can have two types of sessions: external BGP (E-BGP) and internal BGP (I-BGP) sessions.
  - The E-BGP session is used to exchange information between two speaker nodes belonging to two different autonomous systems.
  - The I-BGP session, on the other hand, is used to exchange routing information between two routers inside an autonomous system.
- The session established between AS1 and AS2 is an E-BOP session. The two speaker routers exchange information they know about networks in the Internet.
- However, these two routers need to collect information from other routers in the autonomous systems. This is done using I-BOP sessions

# External and Internal BGP



# Thank You