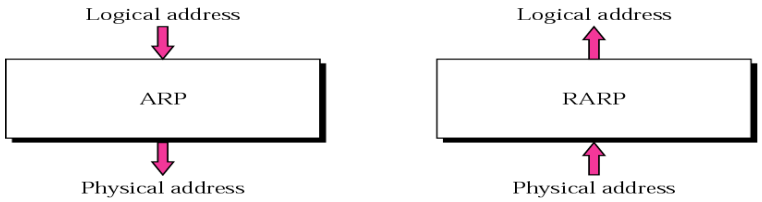
Unit - II

UNIT – II : Address Resolution Protocol (ARP)

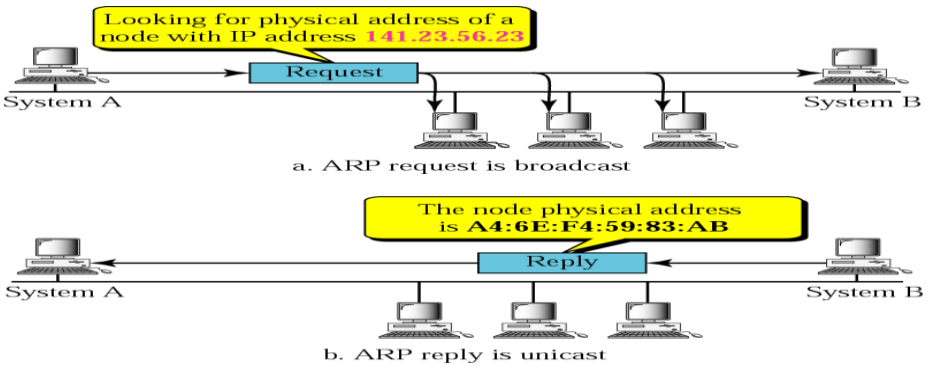
Address mapping - The ARP protocol - ATMARP - ARP package - Internet control message protocol Version 4 - Introduction - Messages - Debugging tools - ICMP package - Unicast routing protocols (RIP, OSPE and BGP) - Introduction - Intra and inter domain routing - Distance vector routing - RIP - Link state routing - OSPF - Path vector routing - BGP

ADDRESS MAPPING:

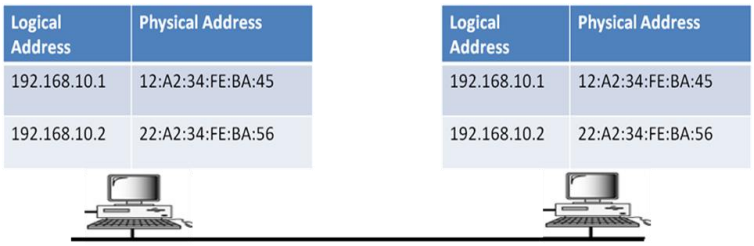
- ✓ The delivery of a packet to a host or a router requires two levels of addressing: *logical* and *physical*.
- ✓ We need to be able to map a logical address to its corresponding physical address and vice versa.
- ✓ These can be done using either *static* or *dynamic* mapping.



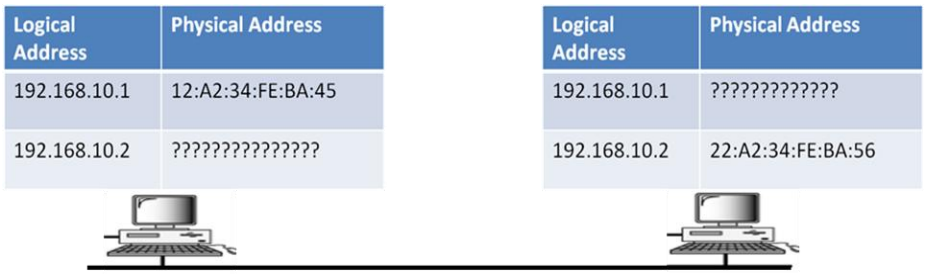
- ✓ Example:



- ✓ **Static Mapping**
  - Static mapping means creating 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.
  - **Limitations:** Table must be updated periodically. This overhead could affect network performance.
    - 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.

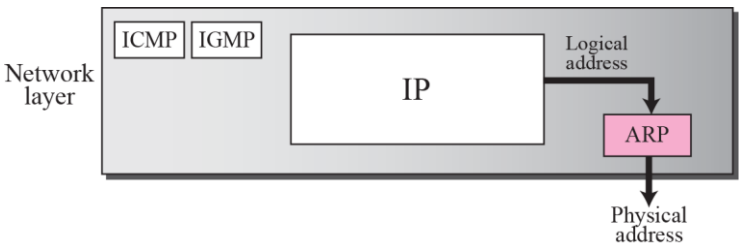


- ✓ **Dynamic Mapping**
  - In dynamic mapping, each time a machine knows the logical address of another machine.
  - Need protocol to find the physical address.
  - Two protocols have been designed to perform dynamic mapping:
    - Address Resolution Protocol (ARP)
      - maps a logical address to a physical address
    - Reverse Address Resolution Protocol (RARP)
      - maps a physical address to a logical address.

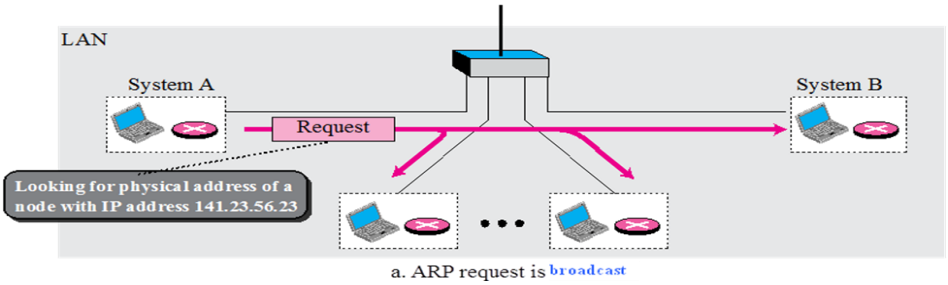


THE ARP PROTOCOL:

- ✓ 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. 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.
- ✓ A mapping corresponds a logical address to a physical address. ARP accepts a logical address from the IP protocol, maps the address to the corresponding physical address and pass it to the data link layer.
- ✓ Figure shows the position of the ARP in the TCP/IP protocol suite.

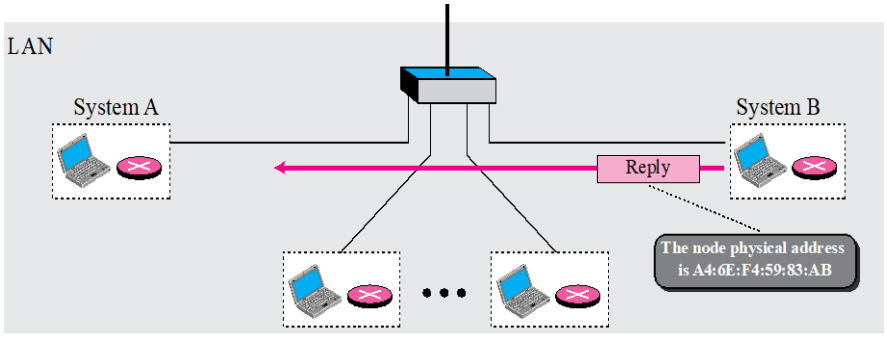


- ✓ **ARP operation:**
  - Anytime a host, or a router, needs to find the physical address of another host or router on its network, it sends an ARP query packet. The packet includes the physical and IP addresses of the sender and the IP address of the receiver. Because the sender does not know the physical address of the receiver, the query is broadcast over the network.



a. ARP request is broadcast

- 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 using the physical address received in the query packet.



b. ARP reply is unicast

✓ **Packet Format**

Hardware Type		Protocol Type
Hardware length	Protocol length	Operation Request 1, Reply 2
Sender hardware address (For example, 6 bytes for Ethernet)		
Sender protocol address (For example, 4 bytes for IP)		
Target hardware address (For example, 6 bytes for Ethernet) (It is not filled in a request)		
Target protocol address (For example, 4 bytes for IP)		

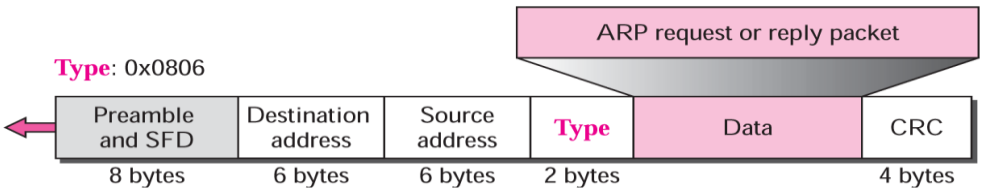
- Figure shows the format of an ARP packet. The fields are as follows:
- Hardware type:
  - This is a 16-bit field defining the type of the network on which ARP is running. Each LAN has been assigned an integer based on its type. For example, Ethernet is given the type 1. ARP can be used on any physical 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.

- 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 the type of packet. Two packet types are defined: ARP request (1), ARP reply (2).

- 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.
- 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 0s 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 an ARP packet is encapsulated in an Ethernet frame.

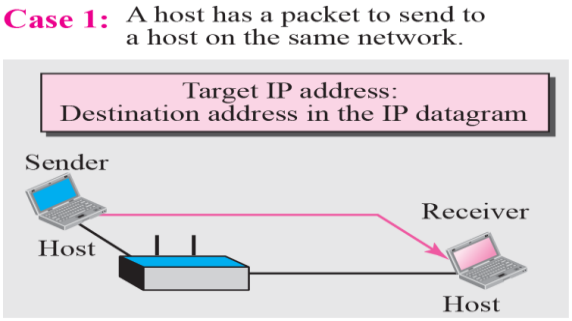


✓ **Operation:**

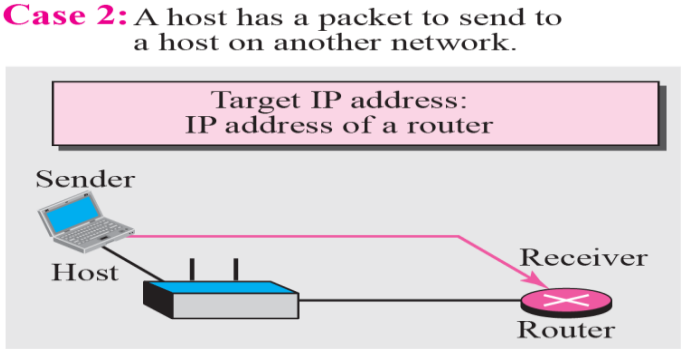
- Let us see how ARP functions on a typical internet. First we describe the steps involved. Then we discuss the four cases in which a host or router needs to use ARP.
- These are seven steps involved in an ARP process:
  1. The sender knows the IP address of the target.
  2. IP asks ARP to create an 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 0s.
  3. The message is passed to the data link layer where it is encapsulated in a frame using the physical address of the sender as the source address and the physical broadcast address as the destination address.
  4. 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 the IP address.

5. The target machine replies with an ARP reply message that contains its physical address. The message is unicast.
6. The sender receives the reply message. It now knows the physical address of the target machine.
7. The IP datagram, which carries data for the target machine, is now encapsulated in a frame and is unicast to the destination.

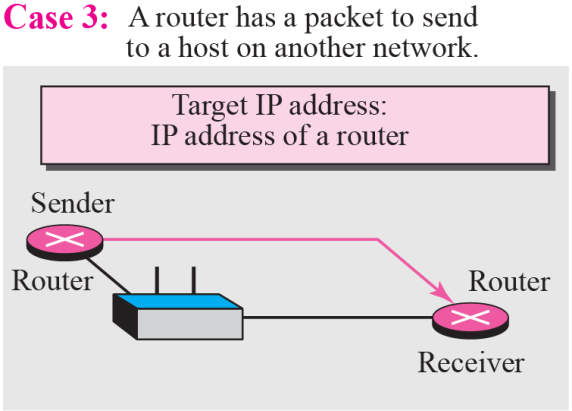
- Four Different Cases:
  - The following are four different cases in which the services of ARP can be used.
  - Case 1:
    - The sender is a host and wants to send a packet to another host on the same network. In this case, the logical address that must be mapped to a physical address is the destination IP address in the datagram header.



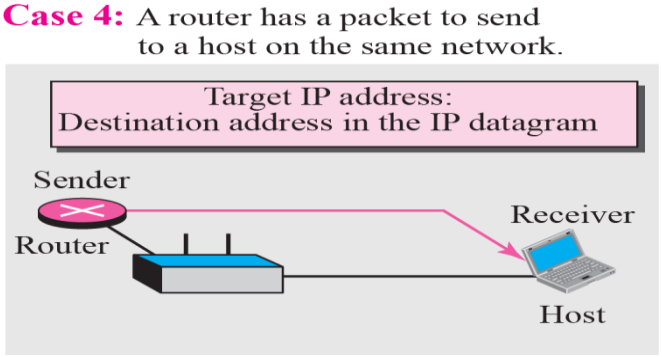
  - Case 2:
    - The sender is a host and wants to send a packet to another host on another network. In this case, the host looks at its routing table and finds the IP address of the next hop (router) for this destination. If it does not have a routing table, it looks for the IP address of the default router. The IP address of the router becomes the logical address that must be mapped to a physical address.



  - Case 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.

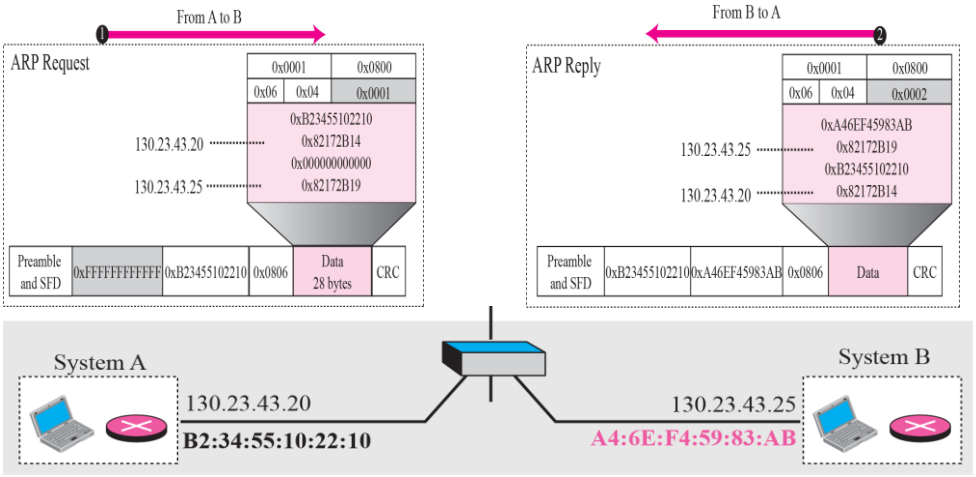


- Case 4:
  - The sender is a router that has received a datagram destined for a host in the same network. The destination IP address of the datagram becomes the logical address that must be mapped to a physical address.



✓ **Example**

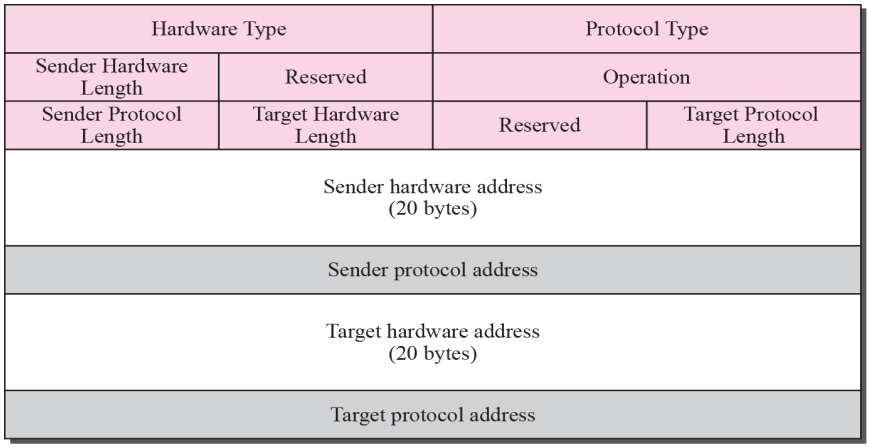
- A host with IP address 130.23.43.20 and physical address B2:34:55:10:22:10 has a packet to send to another host with IP address 130.23.43.25 and physical address A4:6E:F4:59:83:AB. The two hosts are on the same Ethernet network. Show the ARP request and reply packets encapsulated in Ethernet frames.



ATM ARP

- ✓ When IP packet are moving through an ATMWAN, a mechanism protocol is needed to find (map) the physical address of the exiting-point router in the ATM WAN given the IP address of the router. This is the same task performed by ARP on a LAN. However, there is a difference between a LAN and an ATM network. A LAN is a broadcast network (at the data link layer); ARP uses the broadcasting capability of a LAN to send (broadcast) an ARP request.

- ✓ **Packet Format**



  - The format of an ATMARP packet, which is similar to the ARP packet, is shown in Figure. The fields are as follows:
    - **Hardware type (HTYPE) :** The 16-bit HTYPE field defines the type of the physical network. Its value is 0013
    - 16 for an ATM network.
    - **Protocol type (PTYPE):** The 16-bit PTYPE field defines the type of the protocol. For IPv4 protocol the value is 080016.
    - **Sender hardware length (SHLEN):** The 8-bit SHLEN field defines the length of the sender's physical address in bytes. For an ATM network the value is 20.
    - **Operation (OPER):** The 16-bit OPER field defines the type of the packet. Five packet types : 1 – Request, 2- Reply, 8- Inverse Request, 9 – Inverse Reply, 10. NACK
    - **Sender protocol length (SPLEN):** The 8-bit SPLEN field defines the length of the address in bytes. For IPv4 the value is 4 bytes.
    - **Target hardware length (TLEN):** The 8-bit TLEN field defines the length of the receiver's physical address in bytes. For an ATM network the value is 20.
    - **Target protocol length (TPLEN):** The 8-bit TPLEN field defines the length of the address in bytes. For IPv4 the value is 4 bytes.
    - **Sender hardware address (SHA):** The variable-length SHA field defines the physical address of the sender. For ATM networks defined by the ATM Forum, the length is 20 bytes.
    - **Sender protocol address (SPA):** The variable-length SPA field defines the address of the sender. For IPv4 the length is 4 bytes.
    - **Target hardware address (THA):** The variable-length THA field defines the physical address of the receiver. For ATM networks defined by the ATM Forum, the length is 20 bytes. This field is left

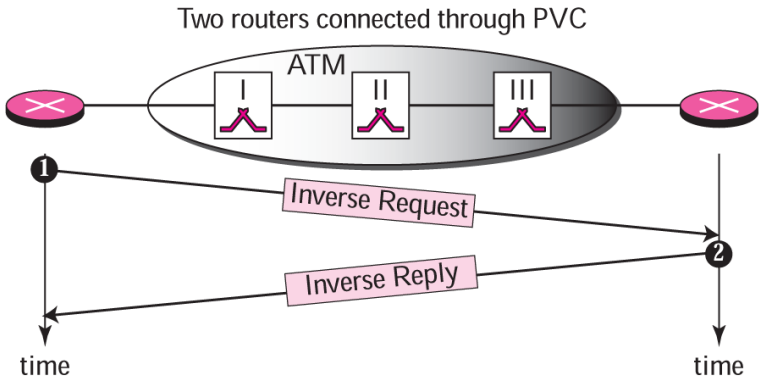
empty for request messages and filled in for reply and NACK messages.

- **Target protocol address (TPA):** The variable-length TPA field defines the address of the receiver. For IPv4 the length is 4 bytes.

✓ **ATMARP Operation**

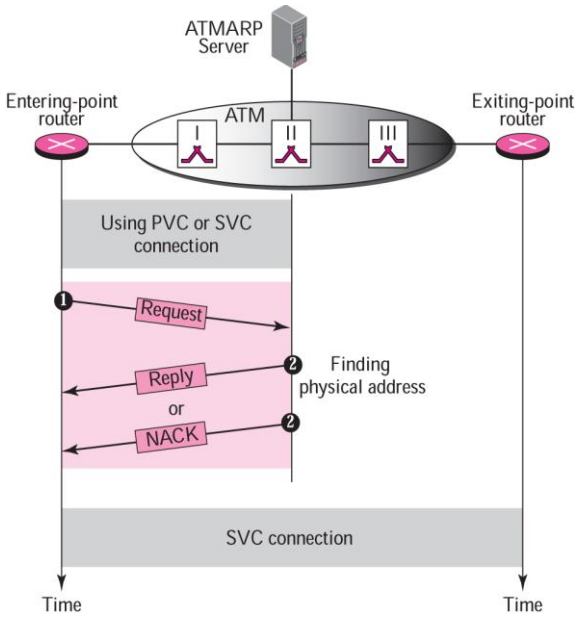
- There are two methods to connect two routers on an ATM network:
  - through a permanent virtual circuit (PVC) or
  - through a switched virtual circuit (SVC).
- The operation of ATMARP depends on the connection method.

- ***PVC Connection & Binding with PVC***
  - A permanent virtual circuit (PVC) connection is established between two end points by the network provider.
  - The VPIs and VCIs are defined for the permanent connections and the values are entered in a table for each switch.
  - If a PVC is established between two routers, there is no need for an ATMARP server. However, the routers must be able to bind a physical address to an IP address. The inverse request message and inverse reply message can be used for the binding.



- ***SVC Connection***
  - In a switched virtual circuit (SVC) connection, each time a router wants to make a connection with another router (or any computer), a new virtual circuit must be established.
  - However, the virtual circuit can be created only if the entering-point router knows the physical address of the exiting-point router
  - To map the IP addresses to physical addresses, each router runs a client ATMARP program, but only one computer runs an ATMARP server program.
  - ATM is a non-broadcast network; an ATMARP request cannot reach all routers connected to the network.
  - The process of establishing a virtual connection requires three steps: connecting to the server, receiving the physical address, and establishing the connection.

- ***Binding with ATMARP***

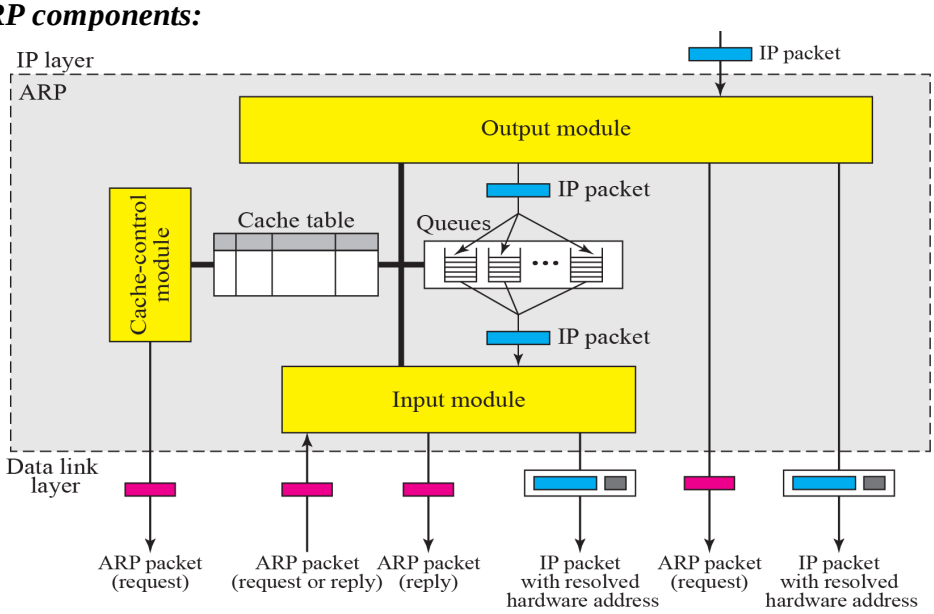


- Connecting to the Server
  - Normally, there is a permanent virtual circuit established between each router and the server. If there is no PVC connection between the router and the server, the server must at least know the physical address of the router to create an SVC connection just for exchanging ATMARP request and reply messages.
- Receiving the Physical Address
  - When there is a connection between the entering point router and the server, the router sends an ATMARP request to the server. The server sends back an ATMARP reply if the physical address can be found or an ATMARP NACK otherwise. If the entering-point router receives a NACK, the datagram is dropped.
- Establishing Virtual Circuits
  - After the entering-point router receives the physical address of the exiting-point router, it can request an SVC between itself and the exiting point router. The ATM network uses the two physical addresses to set up a virtual circuit which lasts until the entering-point router asks for disconnection. In this step, each switch inside the network adds an entry to its tables to enable them to route the cells carrying the IP datagram.

ARP PACKAGE

- ✓ ARP package involves five components: a **cache table, queues, an output module, an input module, and a cache-control module**.
- ✓ The package receives an IP datagram that needs to be encapsulated in a frame that needs the destination physical (hardware) address.
- ✓ If the ARP package finds this address, it delivers the IP packet and the physical address to the data link layer for transmission.

- ✓ ***ARP components:***



- ○ ***Cache Table:***
  - ▪ A sender usually has more than one IP datagram to send to the same destination. It is inefficient to use the ARP protocol for each datagram destined for the same host or router. The solution is the cache table. When a host or router receives the corresponding physical address for an IP datagram, the address can be saved in the cache table. This address can be used for the datagrams destined for the same receiver within the next few minutes. However, as space in the cache table is very limited, mappings in the cache are not retained for an unlimited time.
  - ▪ The cache table is implemented as an array of entries. In our package, each entry contains the following fields:
    - State, Hardware type, Protocol type, Hardware length, Protocol length, Interface number, Queue number, Attempts, Time-out, Hardware address and Protocol address.
- ○ ***Queues:***
  - ▪ Our ARP package maintains a set of queues, one for each destination, to hold the IP packets while ARP tries to resolve the hardware address.

- ○ ***Output Module:***
  - ▪ The output module sends unresolved packets into the corresponding queue.
  - ▪ The output module in pseudocode.

```
ARP_Output_Module ( )
  {
  Sleep until an IP packet is received from IP software.
  Check cache table for an entry corresponding to the
  destination of IP packet.
     If (entry is found)
     {
              If (the state is RESOLVED)
              {
              Extract the value of the hardware address from the entry.
               Send the packet and the hardware address to data
               link layer.
              Return
              } // end if
               If (the state is PENDING)
              {
               Enqueue the packet to the corresponding queue.
              Return
               }//end if
    }//end if
    If (entry is not found)
    {
              Create a cache entry with state set to PENDING and
               ATTEMPTS set to 1.
              Create a queue.
               Enqueue the packet.
              Send an ARP request.
              Return
    }//end if
 } //end module
```

- ***Input Module:***
  - The input module removes a packet from a queue and sends it, with the resolved physical address, to the data link layer for transmission.
  - The input module in pseudocode.

```
ARP_Input_Module ( )
   {
   Sleep until an ARP packet (request or reply) arrives.
   Check the cache table to find the corresponding entry.
   If (found)
          {
          Update the entry.
          If (the state is PENDING)
                 {
                 While (the queue is not empty)
                        {
                        Dequeue one packet.
                        Send the packet and the hardware address.
                        }//end while
                 }//end if
          }//end if
```

```
If (not found)
   {
        Create an entry.
        Add the entry to the table.
   }//end if
 If (the packet is a request)
   {
   Send an ARP reply.
   }//end if
Return
}//end module
```

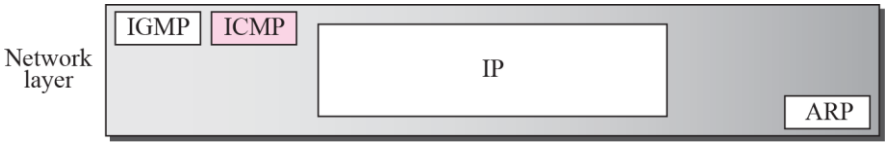
- ***Cache-Control Module:***
  - The cache-control module is responsible for maintaining the cache table. It periodically (for example, every 5 s) checks the cache table, entry by entry.
    - If the state of the entry is FREE, it continues to the next entry.
    - If the state is PENDING, the module increments the value of the attempts field by 1. It then checks the value of the attempts field. If this value is greater than the maximum number of attempts allowed, the state is changed to FREE and the corresponding queue is destroyed. However, if the number of attempts is less than the maximum, the module creates and sends another ARP request.
    - If the state of the entry is RESOLVED, the module decrements the value of the time-out field by the amount of time elapsed since the last check. If this value is less than or equal to zero, the state is changed to FREE and the queue is destroyed.
  - The cache-control module in pseudocode.

```
ARP_Cache_Control_Module ( )
  {
  Sleep until the periodic timer matures.
  Repeat for every entry in the cache table
  {
  If (the state is FREE)
  {
  Continue.
  }//end if
    If (the state is PENDING)
    {
Increment the value of attempts by 1.
    If (attempts greater than maximum)
    {
    Change the state to FREE.
    Destroy the corresponding queue.
    }// end if
    else
    {
    Send an ARP request.
```

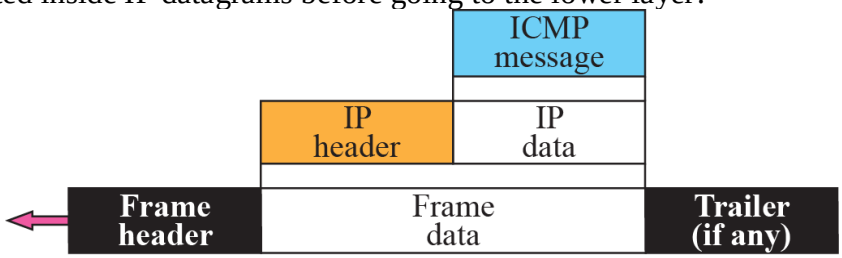
```
}//end else
continue.
}//end if
If (the state is RESOLVED)
{
Decrement the value of time-out.
If (time-out less than or equal 0)
{
Change the state to FREE.
Destroy the corresponding queue.
}//end if
}//end if
}//end repeat
Return.
}//end module
```

Internet Control Message Protocol Version 4 (ICMPv4)

- ✓ The IP protocol has no error-reporting or error correcting mechanism.
- ✓ **Deficiencies:**
  - IP protocol has no built-in mechanism to notify the original host.
  - The IP protocol lacks a mechanism for host and management queries.
    - A host sometimes needs to determine if a router or another host is alive.
    - sometimes a network manager 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.
- ✓ Position of ICMP in the network layer



- ✓ ICMP itself is a network layer protocol. However, its messages are not passed directly to the data link layer as would be expected. Instead, the messages are first encapsulated inside IP datagrams before going to the lower layer.



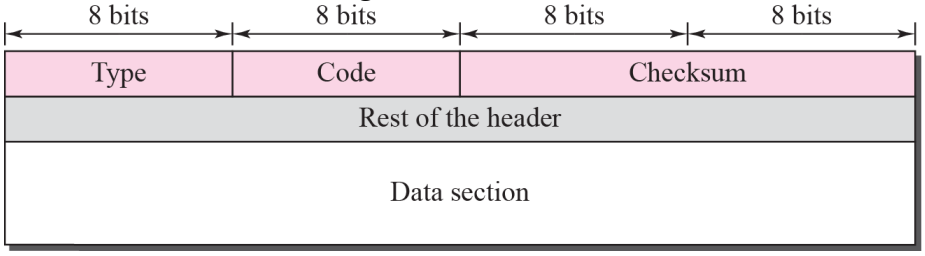
- ✓ The value of the protocol field in the IP datagram is 1 to indicate that the IP data is an ICMP message.

MESSAGES

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

Category	*Type*	*Message*
Error-reporting messages	3	Destination unreachable
	4	Source quench
	11	Time exceeded
	12	Parameter problem
	5	Redirection
Query messages	8 or 0	Echo request or reply
	13 or 14	Timestamp request or reply

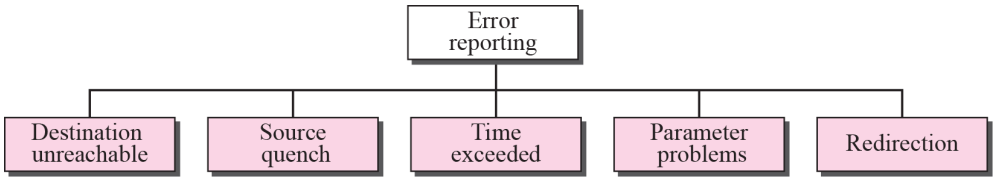
✓ General format of ICMP messages



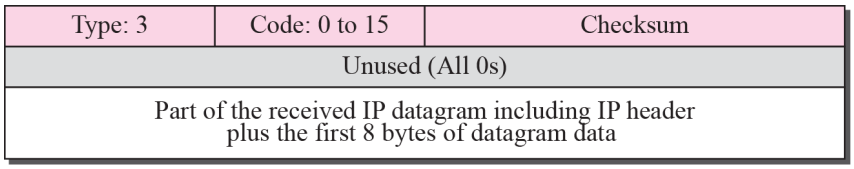
- An ICMP message has an 8-byte header and a variable-size data section. Although the general format of the header is different for each message type, the first 4 bytes are common to all. As Figure shows,
  - The first field, ICMP type, defines the type of the message.
  - The code field specifies the 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.

✓ **Error Reporting Messages**

- One of the main responsibilities of ICMP is to report errors.
- 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.
- **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.**



- **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.



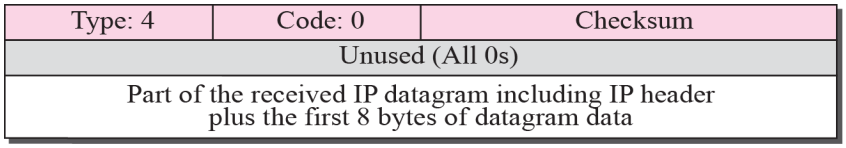
  - The code field for this type specifies the reason for discarding the datagram:
    - Code 0: The network is unreachable, possibly due to hardware failure.
    - Code 1: The host is unreachable. This can also be due to hardware failure.
    - Code 2: The protocol is unreachable. An IP datagram can carry data belonging to higher-level protocols such as UDP, TCP, and OSPF. If the destination host receives a datagram that must be delivered, for example, to the TCP protocol, but the TCP protocol is not running at the moment, a code 2 message is sent.
    - Code 3: The port is unreachable. The application program (process) that the datagram is destined for is not running at the moment.
    - Code 4: Fragmentation is required, but the DF (do not fragment) field of the datagram has been set. In other words, the sender of the datagram has specified that the datagram not be fragmented, but routing is impossible without fragmentation.
    - Code 5: Source routing cannot be accomplished. In other words, one or more routers defined in the source routing option cannot be visited.
    - Code 6: The destination network is unknown. This is different from code 0. In code 0, the router knows that the destination network exists, but it is unreachable at the moment. For code 6, the router has no information about the destination network.
    - Code 7: The destination host is unknown. This is different from code 1. In code 1, the router knows that the destination host exists, but it is unreachable at the moment. For code 7, the router is unaware of the existence of the destination host.
    - Code 8: The source host is isolated.
    - Code 9: Communication with the destination network is administratively prohibited.
    - Code 10: Communication with the destination host is administratively prohibited.
    - Code 11: The network is unreachable for the specified type of service. This is different from code 0. Here the router can route

the datagram if the source had requested an available type of service.

- Code 12: The host is unreachable for the specified type of service. This is different from code 1. Here the router can route the datagram if the source had requested an available type of service.
- Code 13: The host is unreachable because the administrator has put a filter on it.
- Code 14: The host is unreachable because the host precedence is violated. The message is sent by a router to indicate that the requested precedence is not permitted for the destination.
- Code 15: The host is unreachable because its precedence was cut off. This message is generated when the network operators have imposed a minimum level of precedence for the operation of the network, but the datagram was sent with precedence below this level.

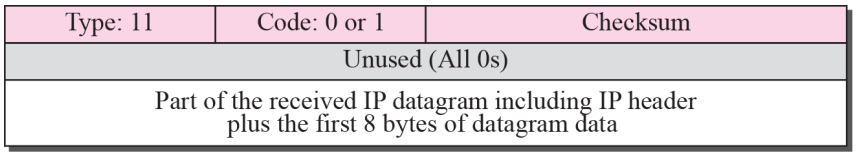
- Destination-unreachable messages with codes 2 or 3 can be created only by the destination host. Other destination-unreachable messages can be created only by routers.
- A router cannot detect all problems that prevent the delivery of a packet.

- **Source Quench**
  - The IP protocol is a connectionless protocol.
  - There is no flow-control or congestion-control mechanism in the IP protocol.
  - The source-quench message in ICMP was designed to add a kind of flow control and congestion 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.
  - The source-quench format is shown in Figure

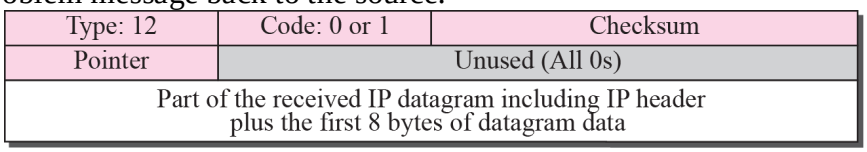


- **Time Exceeded**
  - Whenever a router decrements a datagram with a time-to-live value to zero, it discards the datagram and sends a time-exceeded message to the original source.
  - When the final destination does not receive all of the fragments in a set time, it discards the received fragments and sends a time-exceeded message to the original source.

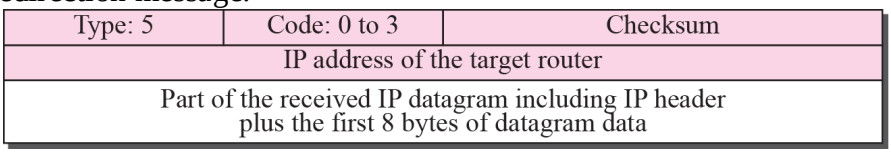
- In a time-exceeded message, code 0 is used only by routers to show that the value of the time-to-live field is zero. Code 1 is used only by the destination host to show that not all of the fragments have arrived within a set time.



- **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.



  - Figure shows the format of the parameter-problem message. The code field in this case specifies the reason for discarding the datagram:
    - Code 0: There is an error or ambiguity in one of the header fields. In this case, the value in the pointer field points to the byte with the problem. For example, if the value is zero, then the first byte is not a valid field.
    - Code 1: The required part of an option is missing. In this case, the pointer is not used.
- **Redirection**
  - A host usually starts with a small routing table that is gradually augmented and updated. One of the tools to accomplish this is the redirection message.



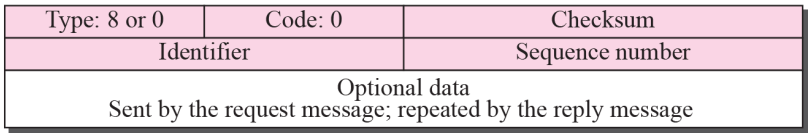
  - A redirection message is sent from a router to a host on the same local network.
  - The format of the redirection message is shown in Figure.
  - The code field for the redirection message narrows down the redirection:
    - Code 0: Redirection for a network-specific route.
    - Code 1: Redirection for a host-specific route.
    - Code 2: Redirection for a network-specific route based on a specified type of service.
    - Code 3: Redirection for a host-specific route based on a specified type of service.

✓ **Query Messages**

- **Echo Request and Reply**

- An echo-request message can be sent by a host or router. An echo-reply message is sent by the host or router that receives an echo-request message.
- Echo-request and echo-reply messages can be used by network managers to check the operation of the IP protocol.
- Echo-request and echo-reply messages can test the reachability of a host. This is usually done by invoking the ping command.

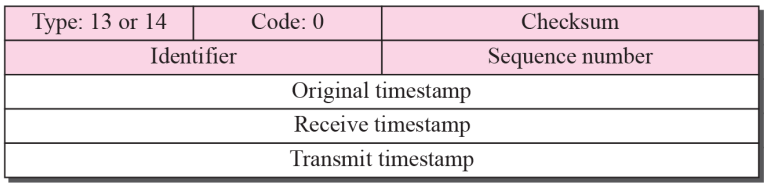
Type 8: Echo request
Type 0: Echo reply



Type: 8 or 0	Code: 0	Checksum
Identifier		Sequence number
Optional data Sent by the request message; repeated by the reply message		

- **Timestamp Request and Reply**
  - Timestamp-request and timestamp-reply messages can be used to calculate the round-trip time between a source and a destination machine even if their clocks are not synchronized.
  - The timestamp-request and timestamp-reply messages can be used to compute the one-way or round-trip time required for a datagram to go from a source to a destination and then back again. The formulas are
    - sending time = receive timestamp - original timestamp
    - receiving time = returned time - transmit timestamp
    - round-trip time = sending time + receiving time

Type 13: request
Type 14: reply



Type: 13 or 14	Code: 0	Checksum
Identifier		Sequence number
Original timestamp		
Receive timestamp		
Transmit timestamp		

DEBUGGING TOOLS

- ✓ There are several tools that can be used in the Internet for debugging. We can find if a host or router is alive and running.
- ✓ We can trace the route of a packet. We introduce two tools that use ICMP for debugging:
  - **Ping :** The ping program to find if a host is alive and responding.

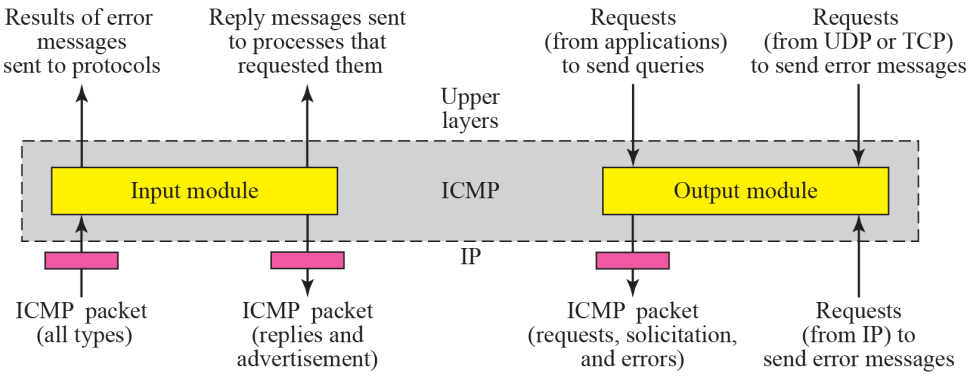
```
$ ping fhda.edu
PING fhda.edu (153.18.8.1)   56 (84)  bytes of data.
64 bytes from tiptoe.fhda.edu (153.18.8.1): icmp_seq=0    ttl=62    time=1.91 ms
64 bytes from tiptoe.fhda.edu (153.18.8.1): icmp_seq=1    ttl=62    time=2.04 ms
64 bytes from tiptoe.fhda.edu (153.18.8.1): icmp_seq=2    ttl=62    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 10103 ms
rtt min/avg/max = 1.899/1.955/2.041 ms
```

- **Traceroute:** It can be used to trace the route of a packet from the source to the destination.
- We use the traceroute program to find the route from the computer voyager.deanza.edu to the server fhda.edu. The following shows the result.

```
$ traceroute fhda.edu
traceroute to fhda.edu      (153.18.8.1), 30 hops max, 38 byte packets
1 Dcore.fhda.edu            (153.18.31.25)      0.995 ms      0.899 ms      0.878 ms
2 Dbackup.fhda.edu          (153.18.251.4)      1.039 ms      1.064 ms      1.083 ms
3  tiptoe.fhda.edu          (153.18.8.1)        1.797 ms      1.642 ms      1.757 ms
```

ICMP PACKAGE

- ✓ ICMP package made of two modules: an input module and an output module. Figure 9.16 shows these two modules.



- ✓ Input Module:
  - The input module handles all received ICMP messages. It is invoked when an ICMP packet is delivered to it from the IP layer. If the received packet is a request, the module creates a reply and sends it out. If the received packet is a redirection message, the module uses the information to update the routing table. If the received packet is an error message, the module informs the protocol about the situation that caused the error. The pseudocode is shown below:

```
ICMP_Input_module (ICMP_Packet)
  {
  If (the type is a request)
  {
  Create a reply
  Send the reply
  }
  If (the type defines a redirection)
  {
   Modify the routing table
   }
   If (the type defines other error messages)
   {
   Inform the appropriate source protocol
   }
   Return
 }
```

- ✓ Output Module:
  - The output module is responsible for creating request, solicitation, or error messages requested by a higher level or the IP protocol. The module receives a demand from IP, UDP, or TCP to send one of the ICMP error messages.
  - The pseudocode is shown in below:

```
ICMP_Output_Module (demand)
  {
  If (the demand defines an error message)
  {
        If (demand comes from IP AND is forbidden)
```

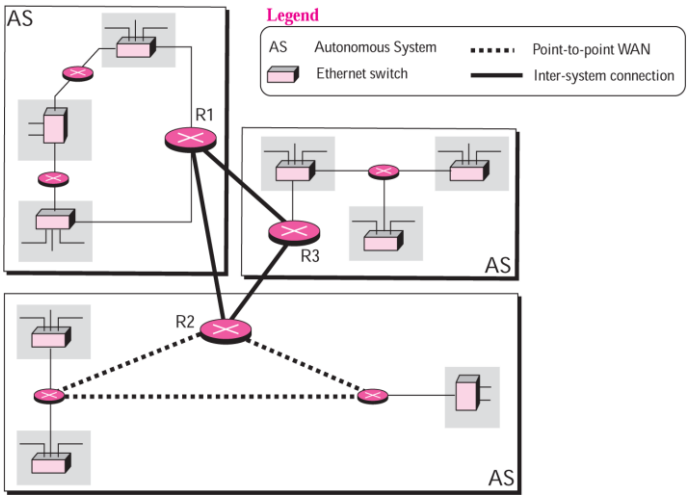
```
        {
        Return
        }
        If (demand is a valid redirection message)
          {
          Return
          }
          Create an error message
          If (demand defines a request)
          {
          Create a request message
          }
          Send the message
          Return
}
```

Unicast Routing Protocols (RIP, OSPF, and BGP)

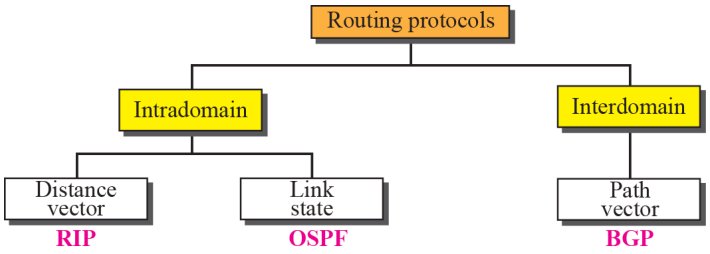
- ✓ Routing is the act of moving information from a source to a destination in an internetwork.
  - An internet is a combination of networks connected by routers. When a datagram goes from a source to a destination, it will probably pass through many routers until it reaches the router attached to the destination network.
- ✓ Routing protocol specifies how routers communicate with each other, disseminating information that enables them to select routes between any two nodes on a network.

INTER- AND INTRA-DOMAIN 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 called intra-domain routing.
  - Routing between autonomous systems is called inter-domain routing.

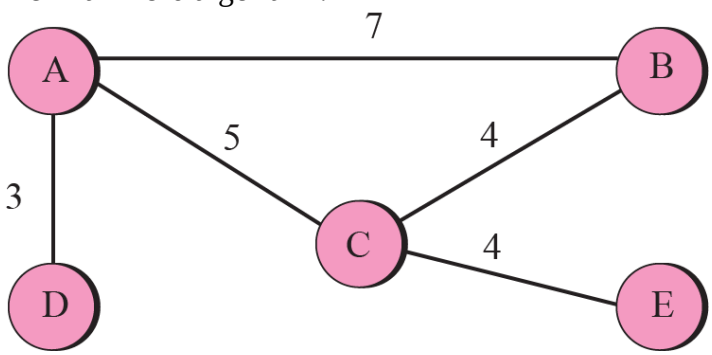


- ✓ Popular routing protocols



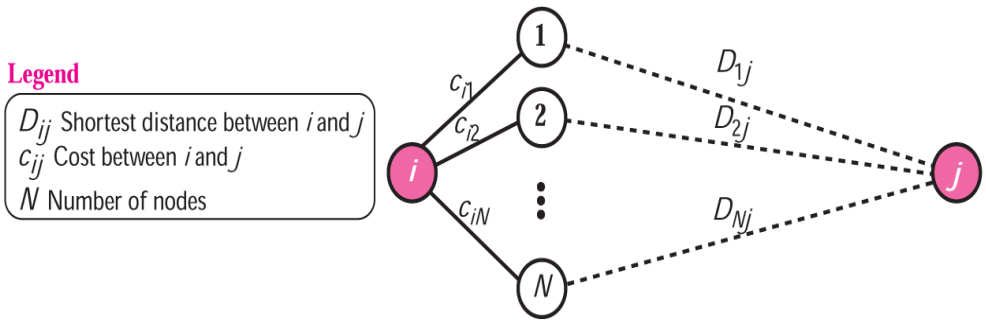
DISTANCE VECTOR ROUTING

- ✓ Distance-vector routing protocols use the Bellman–Ford algorithm.
  - To find the shortest path between nodes in a graph given the distance between nodes.
  - We first discuss this algorithm before we see how it can be modified to be used for updating routing tables in a distance vector routing.
- ✓ Bellman–Ford algorithm **to find the least cost (shortest path) between any two nodes.**
- ✓ A graph for the Bellman-Ford algorithm:



- ✓ The fact behind Bellman-Ford algorithm:

$$D_{ij} = \text{minimum } \{(c_{i1} + D_{1j}), (c_{i2} + D_{2j}), \ldots (c_{iN} + D_{Nj})\}$$



- ✓ Create a shortest distance table (vector) for each node using the following steps:
  1. The shortest distance and the cost between a node and itself is initialized to 0.
  2. The shortest distance between a node and any other node is set to infinity. The cost between a node and any other node should be given (can be infinity if the nodes are not connected).
  3. The algorithm repeat until there is no more change in the shortest distance vector.

✓ Bellman-Ford Algorithm:

```
Bellman_Ford ( )
{
   // Initialization
   for (i = 1 to N; for j = 1 to N)
   {
       if(i == j)   D_ij = 0   c_ij = 0
       else         D_ij = ∞   c_ij = cost between i and j
   }
   // Updating
   repeat
   {
      for (i = 1 to N; for j = 1 to N)
      {
          D_ij ← minimum [(c_i1 + D_1j)  ... (c_iN + D_Nj)]
      } // end for
   } until (there was no change in previous iteration)
} // end Bellman-Ford
```

✓ **DISTANCE VECTOR ROUTING**

1. In distance vector routing, the cost is normally hop counts (how many networks are passed before reaching the destination). So the cost between any two neighbors is set to 1.
2. Each router needs to update its routing table asynchronously, whenever it has received some information from its neighbors.
3. After a router has updated its routing table, it should send the result to its neighbors so that they can also update their routing table.
4. Each router should keep at least three pieces of information for each route: destination network, the cost, and the next hop.
5. We refer to information about each route received from a neighbor as R (record), which has only two pieces of information: R.dest and R.cost. The next hop is not included in the received record because it is the source address of the sender.

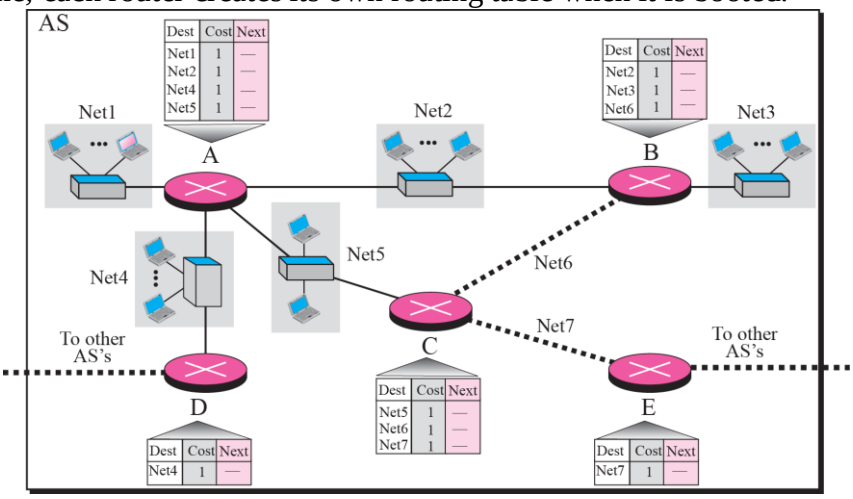
```
Distance_Vector_Algorithm ( )
{
    // At startup
   for (i = 1 to N)           // N is number of ports
   {
       Table_i.dest = address of the attached network
       Table_i.cost = 1
       Table_i.next = ——       // Means at home
       Send a record R about each row to each neighbor
   } // end for loop

     // Updating
   repeat (forever)
   {
       Wait for arrival of a record R from a neighbor
       Update (R, T)           // Call update module
       for (i = 1 to N)        // N is the current table size
```

```
        {
              Send a record R about each row to each neighbor
        }
      }  // end repeat

}  // end Distance_Vector
```

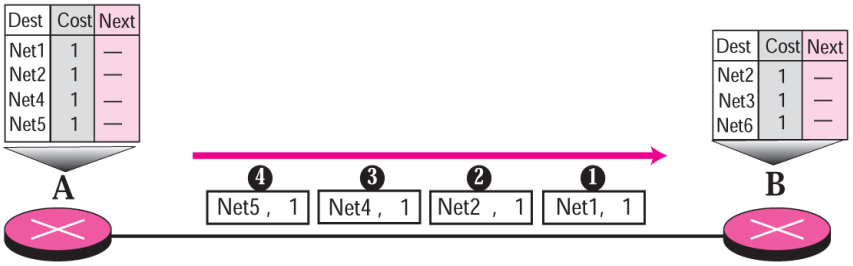
```
Update (R, T)           // Update module
{
    Search T for a destination matching the one in R
    if (destination is found in row i)
    {
        if (R.cost + 1 < T_i.cost    or    R.next == T_i.next)
        {
              T_i.cost = R.cost + 1
              T_i.next = Address of sending router
        }
     else  discard  the  record     // No change is needed
     }
     else
      // Insert the new router
     {
         T_N+1.dest = R.dest
         T_N+1.cost = R.cost + 1
         T_N+1.next = Address of sending router
         Sort the table according to destination address
     }
}  // end of Update module
```

- Updating Routing 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
    - i.e. the sender of the new row is the provider of the old entry
    - The receiving node chooses the new row, even though the new value is infinity.
  - Periodic Update
    - A node sends its routing table, normally 30 seconds, in a periodic update
  - Triggered Update
    - A node sends its routing table to its neighbors any time when there is a change in its routing table
    - 1. After updating its routing table, or
    - 2. Detects some failure in the neighboring links

- Figure shows the initial routing table for an AS. Note that the figure does not mean that all routing tables have been created at the same time; each router creates its own routing table when it is booted.



- Now assume router A sends four records to its neighbors, routers B, D, and C. Figure shows the changes in B's routing table when it receives these records. We leave the changes in the routing tables of other neighbors as exercise.



Routing Table B — After receiving record 1

Dest	Cost	Next
Net1	2	A
Net2	1	—
Net3	1	—
Net6	1	—

Routing Table B — After receiving record 2

Dest	Cost	Next
Net1	2	A
Net2	1	—
Net3	1	—
Net6	1	—

Routing Table B — After receiving record 3

Dest	Cost	Next
Net1	2	A
Net2	1	—
Net3	1	—
Net4	2	A
Net6	1	—

Routing Table B — After receiving record 4

Dest	Cost	Next
Net1	2	A
Net2	1	—
Net3	1	—
Net4	2	A
Net5	2	A
Net6	1	—

- Figure shows the final routing tables for each routers

A

Dest	Cost	Next
Net1	1	—
Net2	1	—
Net3	2	B
Net4	1	—
Net5	1	—
Net6	2	C
Net7	2	C

B

Dest	Cost	Next
Net1	2	A
Net2	1	—
Net3	1	—
Net4	2	A
Net5	2	A
Net6	1	—
Net7	2	C

C

Dest	Cost	Next
Net1	2	A
Net2	2	A
Net3	2	B
Net4	2	A
Net5	1	—
Net6	1	—
Net7	1	—

D

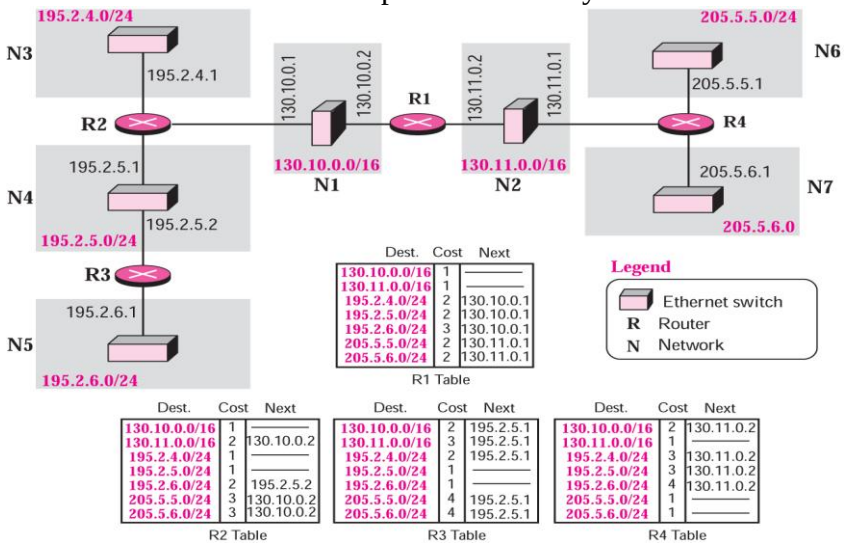
Dest	Cost	Next
Net1	2	A
Net2	2	A
Net3	3	A
Net4	1	—
Net5	1	A
Net6	3	A
Net7	3	A

E

Dest	Cost	Next
Net1	3	C
Net2	3	C
Net3	3	C
Net4	3	C
Net5	2	C
Net6	2	C
Net7	1	—

Routing Information Protocol (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 directly with some considerations:
  1. In an autonomous, we are dealing with routers and networks (links). The routers have routing tables, networks don't
  2. The destination in a routing table is a network, which means the first column defines a network address.
  3. The metric used by RIP is the number of links that have to be used to reach the destination which is called hop count.
  4. Infinity is defined as 16.
  5. The next node column defines the address of the router to which the packet is to be sent to reach its destination.
- ✓ Figure shows an autonomous system with seven networks and four routers.
  - Router R1 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 R1 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 R1 needs to send the packet to the interface of router R4 with IP address 130.11.0.1. The other tables can be explained similarly.



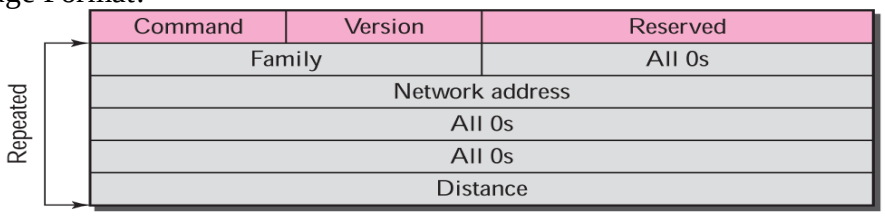
Dest.	Cost	Next
130.10.0.0/16	1	———
130.11.0.0/16	1	———
195.2.4.0/24	2	130.10.0.1
195.2.5.0/24	2	130.10.0.1
195.2.6.0/24	3	130.10.0.1
205.5.5.0/24	2	130.11.0.1
205.5.6.0/24	2	130.11.0.1

Dest.	Cost	Next
130.10.0.0/16	1	———
130.11.0.0/16	2	130.10.0.2
195.2.4.0/24	1	———
195.2.5.0/24	1	———
195.2.6.0/24	2	195.2.5.2
205.5.5.0/24	3	130.10.0.2
205.5.6.0/24	3	130.10.0.2

Dest.	Cost	Next
130.10.0.0/16	2	195.2.5.1
130.11.0.0/16	3	195.2.5.1
195.2.4.0/24	2	195.2.5.1
195.2.5.0/24	1	———
195.2.6.0/24	1	———
205.5.5.0/24	4	195.2.5.1
205.5.6.0/24	4	195.2.5.1

Dest.	Cost	Next
130.10.0.0/16	2	130.11.0.2
130.11.0.0/16	1	———
195.2.4.0/24	3	130.11.0.2
195.2.5.0/24	3	130.11.0.2
195.2.6.0/24	4	130.11.0.2
205.5.5.0/24	1	———
205.5.6.0/24	1	———

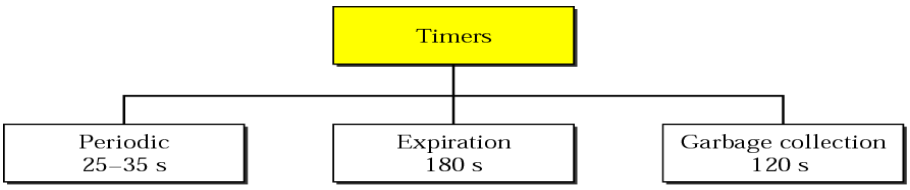
- ✓ RIP Message Format:



- ✓ Command: This 8-bit field specifies the type of message: request (1) or response (2)
- ✓ Version: This 8-bit defines the version. In the textbook, we use version 1
- ✓ Family: This 16-bit field defines the family of the protocol used. For TCP/IP the value is 2.
- ✓ Network address: RIP has allocated 14 bytes for this field to be applicable to any protocol.
- ✓ Distance: This 32-bit field defines the hop count from the advertising router to the destination network.

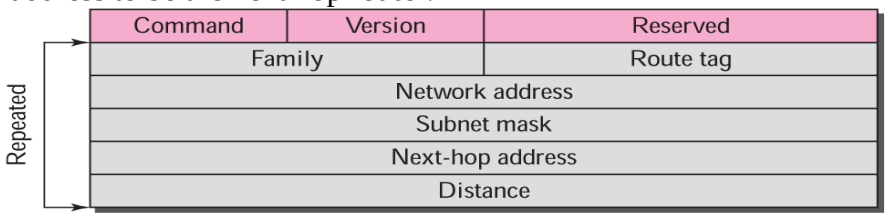
- ✓ Request and Response:
  - **Request**. A request message is sent by a router that has just come up or by a router that has some time-out entries. A request can ask about specific entries or all entries.
  - **Response**. A solicited response is sent only in answer to a request. It contains information about the destination specified in the corresponding request. An unsolicited response is sent periodically, every 30 s or when there is a change in the routing table**.**

- ✓ Timers in RIP:
  - RIP uses 3 timers. The periodic timer controls the sending of messages, the expiration timer governs the validity of a route, and the garbage collection timer advertises the failure of a route**.**



  - Periodic Timer:
    - The periodic timer controls the advertising of regular update messages
    - The working model uses a random number between 25 and 30 s
    - This is to prevent any possible synchronization and therefore overload on an internet if routers update simultaneously
  - Expiration Timer:
    - The expiration timer governs the validity of a route
    - When a router receives update information for a route, the expiration timer is set to 180 s for that particular route
    - Every time a new update for the route is received, the timer is reset
    - If the timer is expired, the hop count of the route is set to 16, which means the destination is unreachable
  - Garbage Collection Timer:
    - When the information about a route becomes invalid, the router does not immediately purge that route from its table
    - Instead, it continues to advertise the route with a metric value of 16
    - At the same time, the garbage collection timer is set to 120 s for that route
    - When the count reaches zero, the route is purged from the table
- ✓ **RIP Version 2:**
  - RIP version 2 was designed to overcome some of the shortcomings of version 1.
  - The designers of version 2 have not augmented the length of the message for each entry.
  - They have only replaced those fields in version 1 that were filled with 0s for the TCP/IP protocol with some new fields.
  - Message Format Updates:

- Route tag. This field carries information such as the autonomous system number. It can be used to enable RIP to receive information from an interdomain routing protocol.
- Subnet mask. This is a 4-byte field that carries the subnet mask. This means that RIP2 supports classless addressing and CIDR.
- Next-hop address. If the sending router want to specify another router IP address to be the next hop router.



Link state routing - OSPF - Path vector routing - BGP

Refer the Book