Unit I

UNIT – 1: The OSI Model and The TCP/IP Protocol Suite

TCP/IP protocol suite - Addressing- Internet protocol version 4 (IPv4) - Datagrams - Fragmentation - Options - Checksum - IPv4 addresses - Introduction - Classful addressing - Classless addressing - Special addresses - NAT

Comparison of OSI and TCP/IP Model:

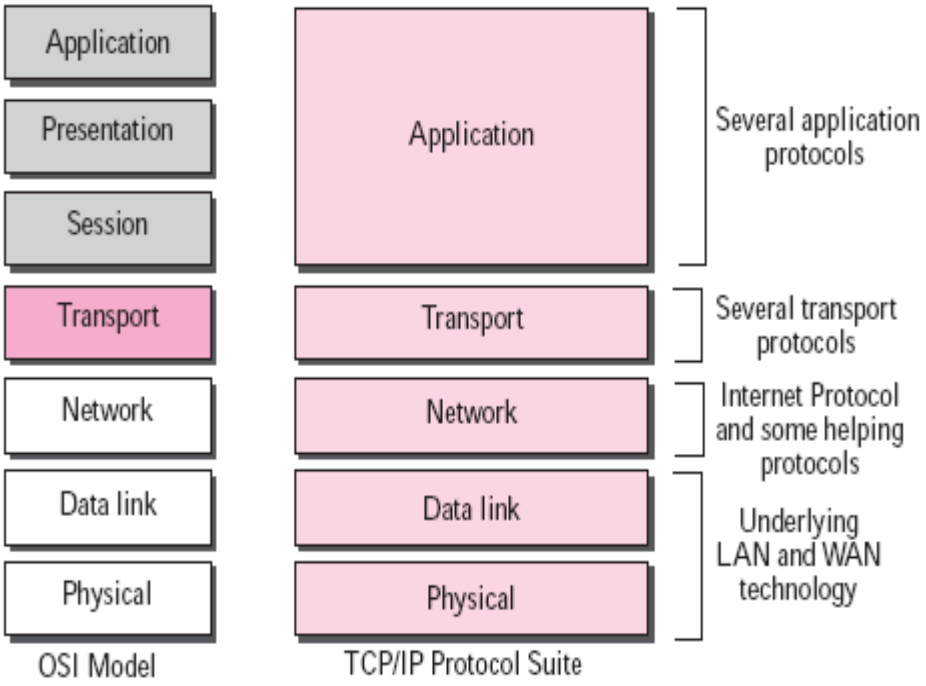


Fig: OSI and TCP/IP Model's

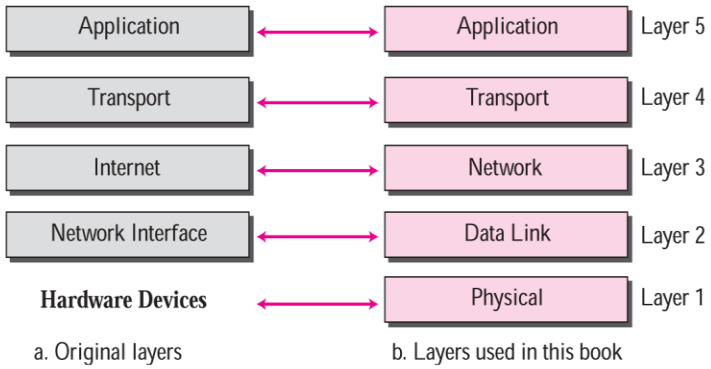
- ✓ When we compare the two models, we find that two layers, session and presentation are missing from the TCP/IP protocol suite.
- ✓ These two layers were not added to the TCP/IP protocol suite after the publication of the OSI model.
- ✓ The application layer in the suite is usually considered to be the combination of three layers in the OSI model, as shown in above Fig.

- ✓ Two reasons were mentioned for this decision.
  1) TCP/IP has more than one transport-layer protocol. Some of the functionalities of the session layer are available in some of the transport layer protocols.

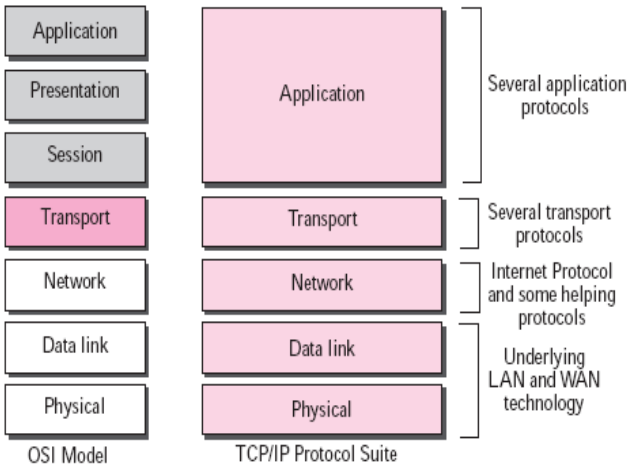
  2) The application layer is not only one piece of software. Many applications can be developed at this layer. If some of the functionalities mentioned in the session and presentation are needed for a particular application, it can be included in the development of that piece of software.

TCP/IP Protocol Suite:

- ✓ It was developed prior to the OSI Model.
- ✓ The layers in the TCP/IP protocol suite do not match exactly with those in the OSI model.
- ✓ The original TCP/IP protocol suite was defined as four software layers built upon the hardware.
- ✓ Now, TCP/IP is thought of as a five-layer model with the layers named similarly to the ones in the OSI model.



OSI and TCP/IP Model



Layers in the TCP/IP Protocol Suite

- ✓ In this section, we briefly discuss the purpose of each layer in the TCP/IP protocol suite. When we study the purpose of each layer, it is easier to think of a private internet, instead of the global Internet.
- ✓ Figure below shows our imaginary internet that is used to show the purpose of each layer. We have six links (LANS) and four routers (R1 to R4). We have shown only two computers, A and B.

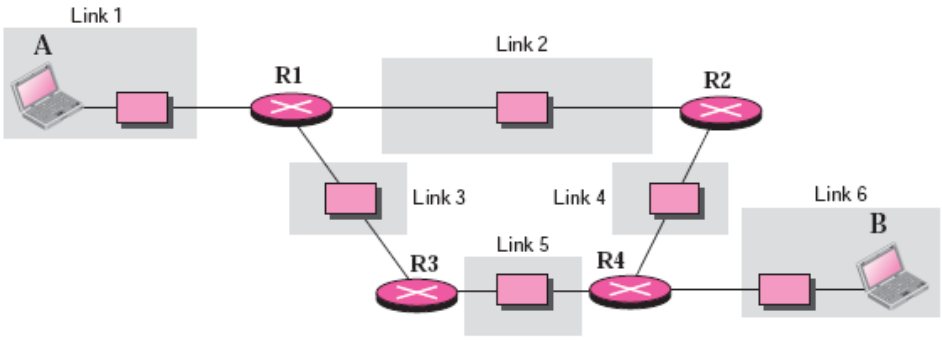


Fig : A Private Internet

Physical Layer

- ✓ TCP/IP does not define any specific protocol for the physical layer.
- ✓ It supports all of the standard and proprietary protocols.
- ✓ The communication is between two hops or nodes, either a computer or router.
- ✓ The unit of communication is a single bit.
- ✓ When the connection is established between the two nodes, a stream of bits is flowing between them. The physical layer, however, treats each bit individually.

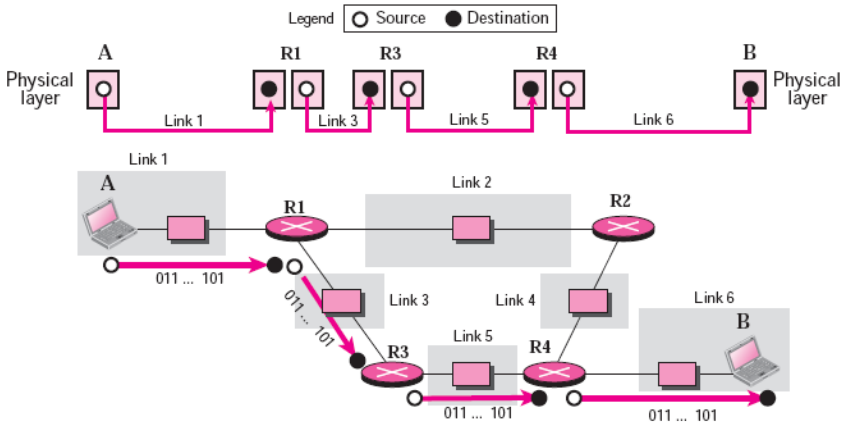


Fig : Communication at the Physical Layer

Above figure shows the communication between nodes. We are assuming that at this moment the two computers have discovered that the most efficient way to communicate with each other is via routers R1, R3, and R4.

- The figure, however, shows only physical layers involved in the communication.
- The journey of bits between computer A and computer B is made of four independent short trips.
- Computer A sends each bit to router R1 in the format of the protocol used by link 1. Router 1 sends each bit to router R3 in the format dictated by the protocol used by link 3. And so on. Router R1 has two three physical layers (two are shown in our scenario).
- The layer connected to link 1 receives bits according to the format of the protocol used by link 1; the layer connected to link 3 sends bits according to the format of the protocol used by link 3. It is the same situation with the other two routers involved in the communication.

Data Link Layer

- ✓ TCP/IP does not define any specific protocol for the data link layer either.
- ✓ It supports all of the standard and proprietary protocols. At this level, the communication is also between two hops or nodes.
- ✓ The unit of communication however, is a packet called a frame.
- ✓ A frame is a packet that encapsulates the data received from the network layer with an added header and sometimes a trailer.
- ✓ The head, among other communication information, includes the source and destination of frame.
    - The destination address is needed to define the right recipient of the frame because many nodes may have been connected to the link.
    - The source address is needed for possible response or acknowledgment as may be required by some protocols.

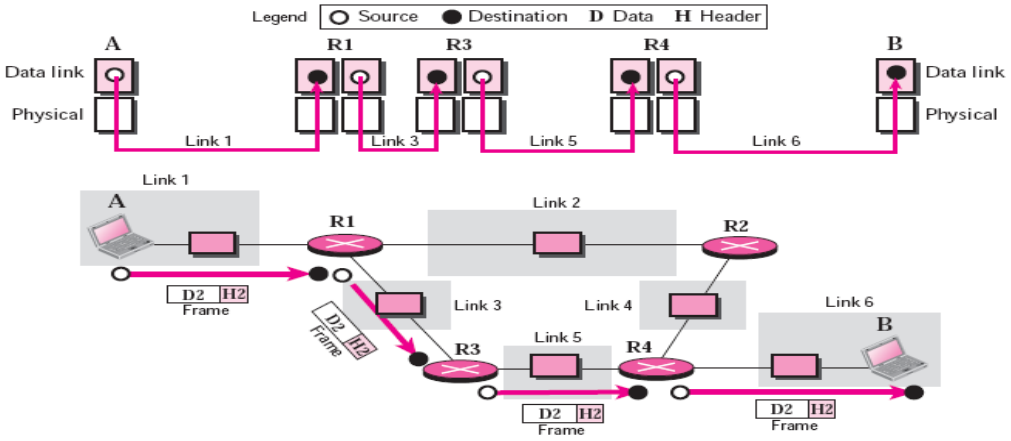


Fig: Communication at the Data link Layer

- ✓ Note that the frame that is travelling between computer A and router R1 may be different from the one travelling between router R1 and R3. When the frame is received by router R1, this router passes the frame to the data link layer protocol shown at the left. The frame is opened, the data are removed. The data are then passed to the data link layer protocol shown at the right to create a new frame to be sent to the router R3. The reason is that the two links, link 1 and link 3, may be using different protocols and require frames of different formats.

Network Layer

- ✓ At the network layer, TCP/IP supports the Internet Protocol (IP).
- ✓ The Internet Protocol (IP) is the transmission mechanism used by the TCP/IP protocols.
- ✓ IP transports data in packets called datagrams, each of which is transported separately.
- ✓ Datagrams can travel along different routes and can arrive out of sequence or be duplicated.
- ✓ IP does not keep track of the routes and has no facility for reordering datagrams once they arrive at their destination.

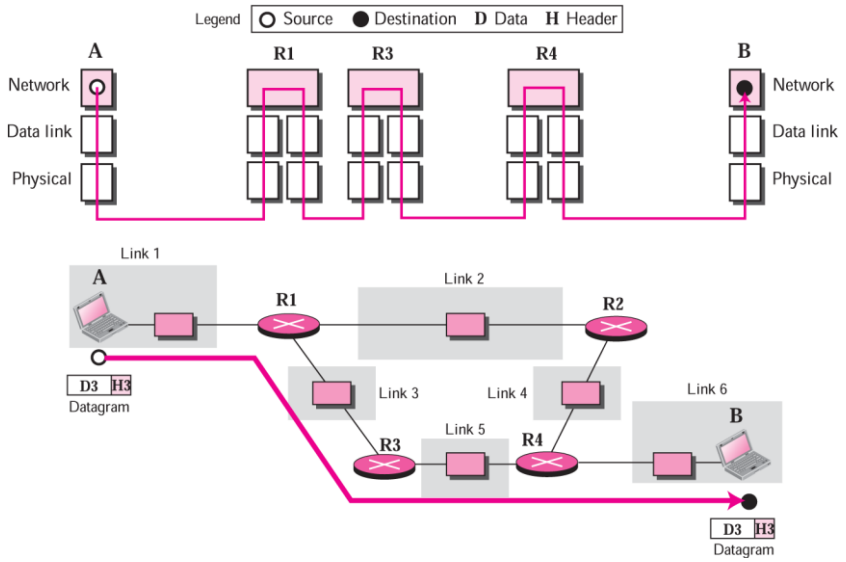


Fig: Communication at the Network Layer

- ✓ Communication at the network layer is end to end while the communications at the other two layers are node to node. The datagram started at computer A is the one that reaches computer B.

Transport Layer

- ✓ There is a main difference between the transport layer and the network layer. Although all nodes in a network need to have the network layer, only the two end computers need to have the transport layer.
- ✓ The unit of communication at the transport layer is a segment, user datagram, or a packet, depending on the specific protocol used in this layer.

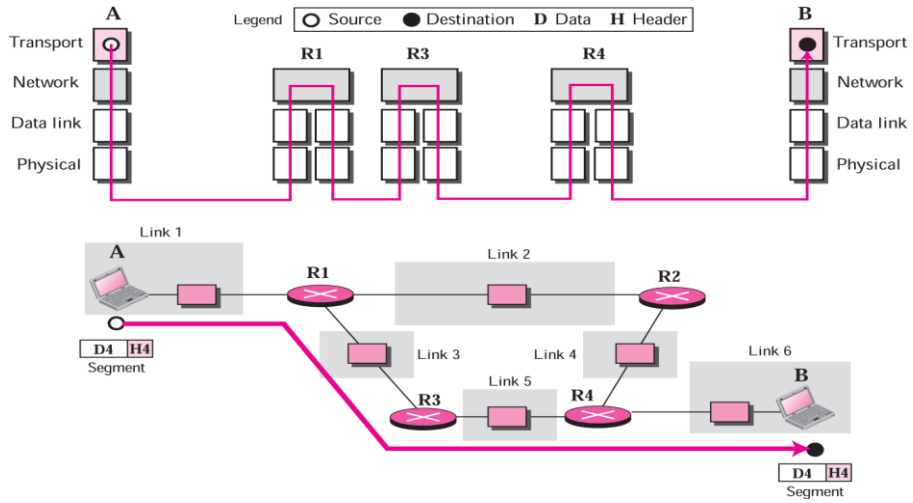


Fig: Communication at the Transport Layer

- ✓ The network layer is responsible for sending individual datagrams from computer A to computer B; the transport layer is responsible for delivering the whole message, which is called a segment, a user datagram, or a packet, from A to B.
- ✓ A segment may consist of a few or tens of datagrams.
- ✓ The segments need to be broken into datagrams and each datagram has to be delivered to the network layer for transmission.
- ✓ Internet defines a different route for each datagram, the datagrams may arrive out of order and may be lost. The transport layer at computer B needs to wait until all of these datagrams to arrive, assemble them and make a segment out of them.
- ✓ Traditionally, the transport layer was represented in the TCP/IP suite by two protocols:
  - User Datagram Protocol (UDP)
  - Transmission Control Protocol (TCP)
- ✓ A new protocol called
  - Stream Control Transmission Protocol (SCTP)

Application Layer

- ✓ The application layer in TCP/IP is equivalent to the combined session, presentation, and application layers in the OSI model.
- ✓ The application layer allows a user to access the services of our private internet or the global Internet.
- ✓ Many protocols are defined at this layer to provide services such as electronic mail, file transfer, accessing the World Wide Web, and so on.
- ✓ The unit of communication at the application layer is a message.

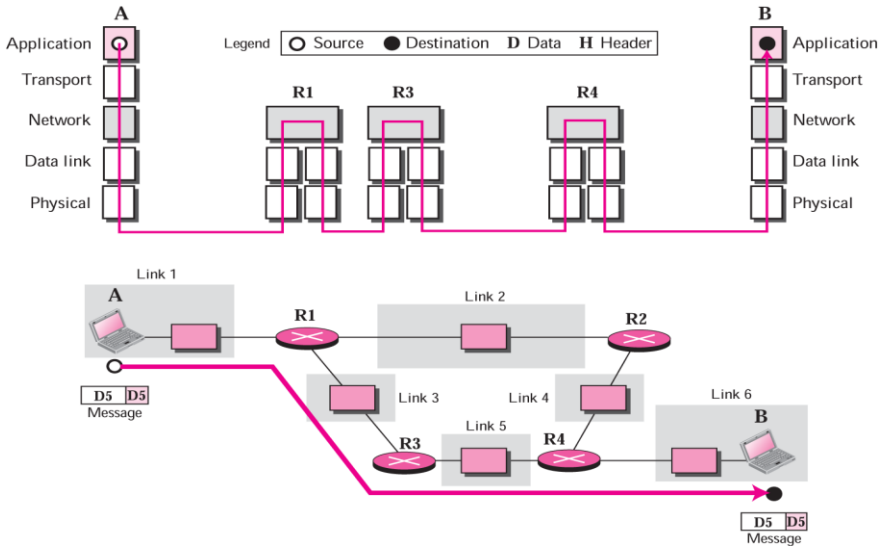


Fig: Communication at the Application Layer

- ✓ Note that the communication at the application layer, like the one at the transport layer, is end to end. A message generated at computer A is sent to computer B without being changed during the transmission.

ADDRESSING

- ✓ Four levels of addresses are used in an internet employing the TCP/IP protocols:
  - Physical address,
  - Logical address,
  - Port address, and
  - Application-specific address.
- ✓ Each address is related to a one layer in the TCP/IP architecture, as shown in Figure shown below.

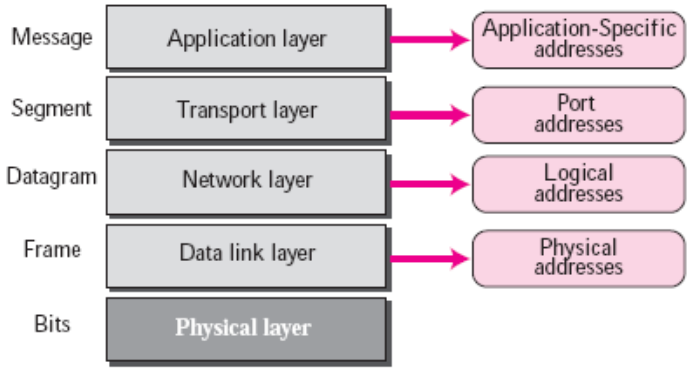


Fig: Addresses in the TCP/IP protocol suite.

Physical Addresses

- ✓ The physical address is also called as the link address, is the address of a node as defined by its LAN or WAN.
- ✓ It is included in the frame used by the data link layer. It is the lowest-level address.
- ✓ The size and format of these addresses vary depending on the network.
- ✓ For example,
  - Ethernet uses a 6-byte (48-bit) physical address that is imprinted on the network interface card (NIC).

- LocalTalk (Apple), however, has a 1-byte dynamic address that changes each time the station comes up.

✓ Most local area networks use a 48-bit (6-byte) physical address written as 12 hexadecimal digits; every byte (2 hexadecimal digits) is separated by a colon, as shown below:

07:01:02:01:2C:4B

A 6-byte (12 hexadecimal digits) physical address.

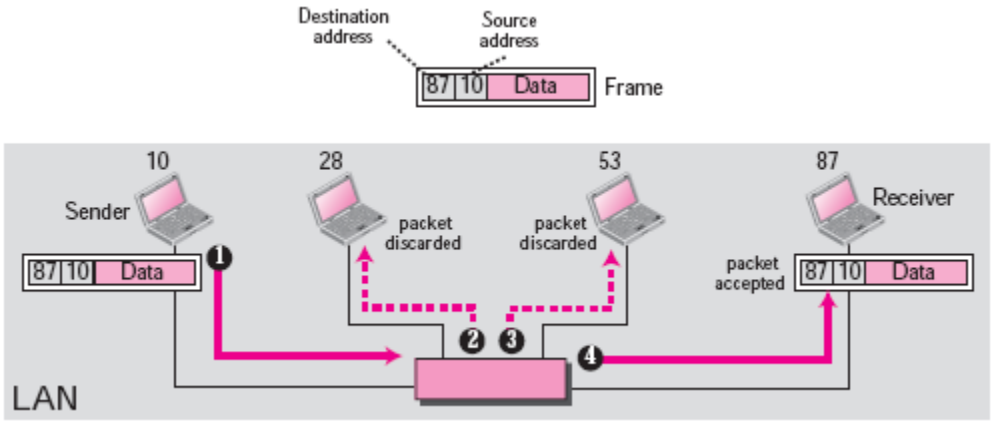


Fig: Example for Physical Addresses

✓ Physical addresses can be either unicast (one single recipient), multicast (a group of recipients), or broadcast (to be received by all systems in the network). Some networks support all three addresses.

Logical Addresses

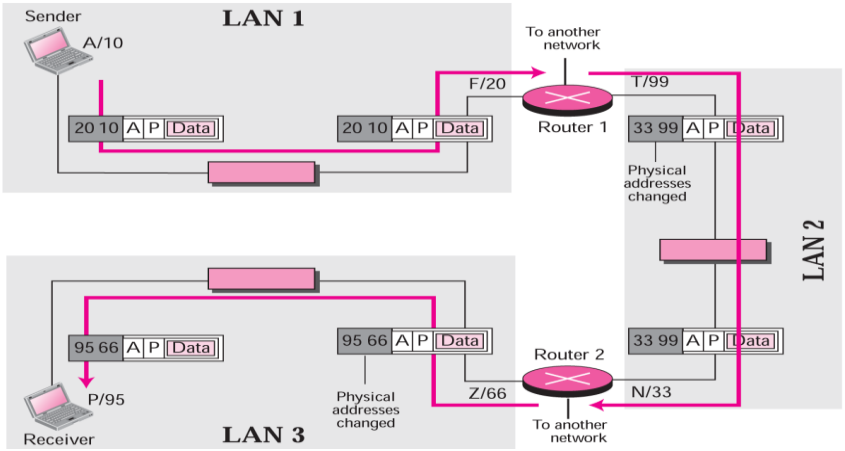
✓ Logical addresses are necessary for universal communications that are independent of underlying physical networks.

✓ A universal addressing system is needed in which each host can be identified uniquely, regardless of the underlying physical network. The logical addresses are designed for this purpose.

✓ A logical address in the Internet is currently a 32-bit address that can uniquely define a host connected to the Internet.

✓ The Network is not permitted two hosts as same IP address.

✓ The physical addresses will change from hop to hop, but the logical addresses remain the same.

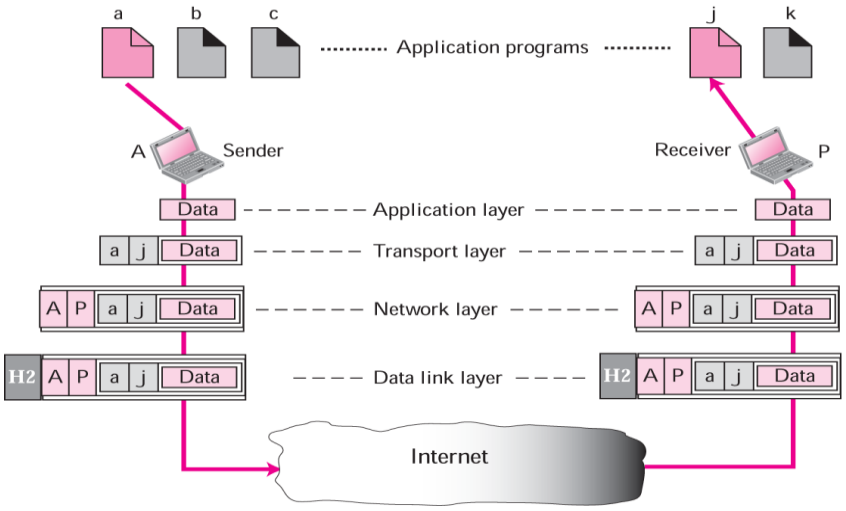


✓ Logical Address example

- 193.168.0.1

Port Addresses

- ✓ The IP address and the physical address are necessary for a quantity of data to travel from a source to the destination host.
- ✓ However, arrival at the destination host is not the final objective of data communications on the Internet.
- ✓ Today, computers are devices that can run multiple processes at the same time.
- ✓ The end objective of Internet communication is a process communicating with another process.
- ✓ **The physical addresses change from hop to hop, but the logical and port addresses usually remain the same.**
- ✓ For example,
  - computer A can communicate with computer C by using TELNET. At the same time, computer A communicates with computer B by using the File Transfer Protocol (FTP).
  - For these processes to receive data simultaneously, we need a method to label the different processes. In other words, they need addresses.
  - In the TCP/IP architecture, the label assigned to a process is called a port address.
- ✓ A port address in TCP/IP is 16 bits in length.
- ✓ Portl Address example
  - 753



  - Figure shows two computers communicating via the Internet. The sending computer is running three processes at this time with port addresses a, b, and c. The receiving computer is running two processes at this time with port addresses j and k. Process a in the sending computer needs to communicate with process j in the receiving computer. Note that although both computers are using the same application, FTP.

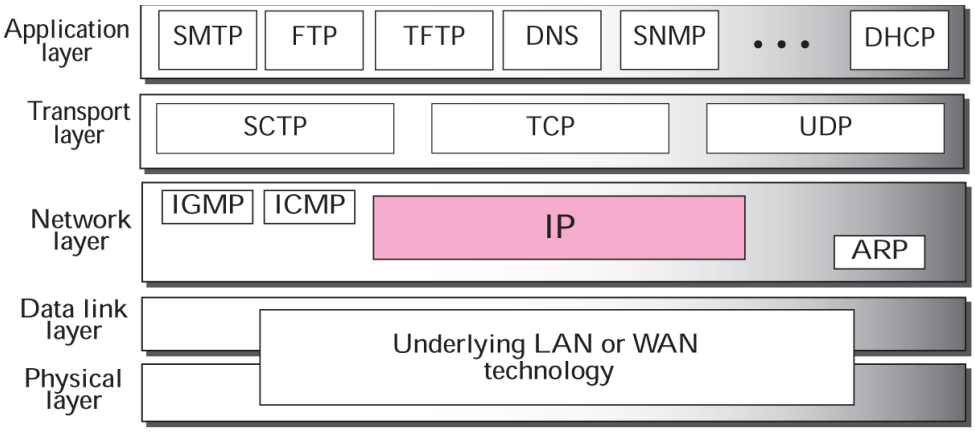
Application-Specific Addresses

- ✓ Some applications have user-friendly addresses that are designed for that specific application.
- ✓ Examples
  - e-mail address (for example, forouzan@fhda.edu)

- the recipient of an e-mail
- The Universal Resource Locator (URL) (for example, www.mhhe.com).
  - used to find a document on the world Wide Web.

Internet protocol version 4 (IPv4)

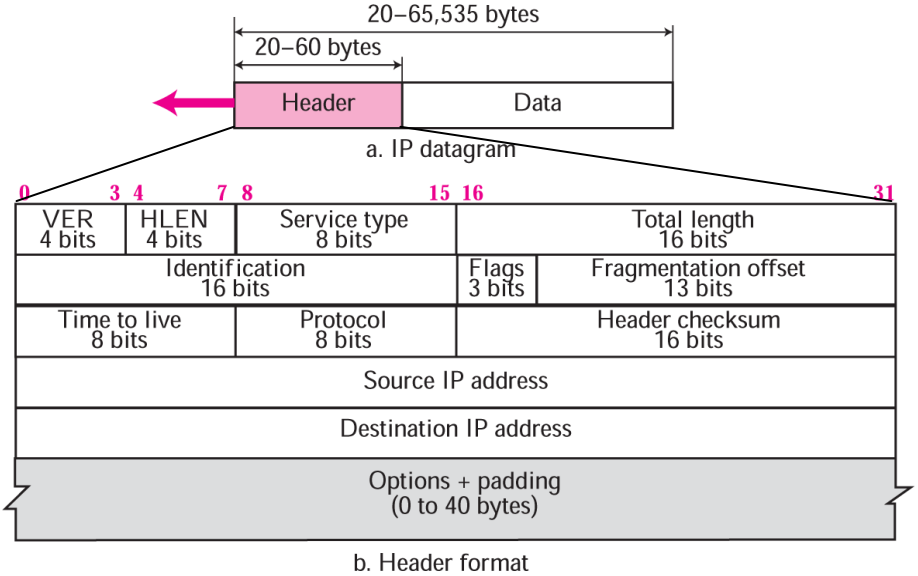
- ✓ The Internet Protocol (IP) is the transmission mechanism used by the TCP/IP protocols at the network layer.
- ✓ Figure shows the position of IP in the suite.



- ✓ IP is an unreliable and connectionless datagram protocol.
- ✓ It is a best-effort delivery service. The term best-effort means that IP packets can be corrupted, lost, arrive out of order, or delayed and may create congestion for the network.
- ✓ If reliability is important, IP must be paired with a reliable protocol such as TCP.
- ✓ IP is also a connectionless protocol for a packet switching network that uses the datagram approach.

Datagrams

- ✓ Packets in the network (internet) layer are called datagrams.
- ✓ A datagram is a variable-length packet consisting of two parts:
  - Header - It is 20 to 60 bytes in length and contains information essential to routing and delivery.
  - Data – Payload data.
- ✓ Figure shows the IP datagram format.



- ✓ **Version (VER) :**

- This 4-bit field defines the version of the IP protocol. Currently the version is 4. This field tells the IP software running in the processing machine that the datagram has the format of version 4.

✓ **Header length (HLEN):**
- This 4-bit field defines the total length of the datagram header in 4-byte words. This field is needed because the length of the header is variable (between 20 and 60 bytes).
  - When there are no options, the header length is 20 bytes.
  - When the option field is at its maximum size 60 bytes.

✓ **Service type:**
- In the original design of IP header, this field was referred to as type of service (TOS)
- TOS component is used to determine the type of service that must be provided by the Internet layer depending on the type of application for which the data transfer needs to be done.
- It has 8 bits field.
  - The first three bits filed are known as precedence bits (ignored as today).
  - The next 4 bits represent type of service
    - TOS are :
      - 0000 - Normal
      - 0001 - Minimizing Cost
      - 0010 - Maximize reliability.
      - 0100 - Maximize throughput
      - 1000 - Minimize delay.
  - Last bit is unused.

✓ **Total length:**
- This is a 16-bit field that defines the total length (header plus data) of the IP datagram in bytes.
- To find the length of the data coming from the upper layer, subtract the header length from the total length.

  Length of data = total length - header length
- The header length can be found by multiplying the value in the HLEN field by four.

✓ **Identification:**
- This 16-bit field identifies a datagram originating from the source host. The combination of the identification and source IP address must uniquely define a datagram as it leaves the source host.
- All fragments of a datagram contain the same identification value.
- This allows the destination host to determine which fragment belongs to which datagram.

✓ **Flags:**
- This is a three-bit field.

  - The first bit is reserved (not used).
  - The second bit (D) is called the **do not fragment bit.**
    - If its value is 1, the machine must not fragment the datagram
    - If its value is 0, the datagram can be fragmented if necessary.

- The third bit (M) is called the **more fragment bit.**
  - If its value is 1, it means the datagram is not the last fragment; there are more fragments after this one.
  - If its value is 0, it means this is the last or only fragment.

✓ **Fragmentation offset:**
- This 13-bit field shows the relative position of this fragment with respect to the whole datagram.
- It is the offset of the data in the original datagram measured in units of 8 bytes.

✓ **Time to live:**
- A datagram has a limited lifetime in its travel through an internet.
- This field was originally designed to hold a timestamp, which was decremented by each visited router.
- The datagram was discarded when the value became zero.

✓ **Protocol:**
- This 8-bit field defines the higher-level protocol that uses the services of the IP layer.
- An IP datagram can encapsulate data from several higher level protocols such as TCP, UDP, ICMP, and IGMP.
- This field specifies the final destination protocol to which the IP datagram should be delivered.
- Some of the value of this field for different higher-level protocols

Value	*Protocol*	*Value*	*Protocol*
1	ICMP	17	UDP
2	IGMP	89	OSPF
6	TCP		

✓ **Checksum:**
- To provide basic protection against corruption in transmission.
- Example – CRC

✓ **Source address:**
- This 32-bit field defines the IP address of the source. This field must remain unchanged during the time the IP datagram travels from the source host to the destination host.

✓ **Destination address:**
- This 32-bit field defines the IP address of the destination. This field must remain unchanged during the time the IP datagram travels from the source host to the destination host.

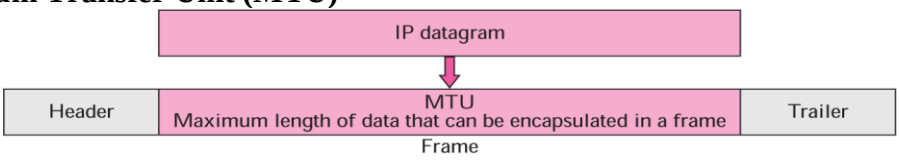
✓ **Options:**
- One or more several types of options may be included after the standard header in certain IP datagrams.

✓ **Padding:**
- The variable part comprises the options, which can be a maximum of 40 bytes. Options, as the name implies, are not required for a datagram. They can be used for network testing and debugging.

✓ **Data:**
- The data to be transmitted in the datagram.

FRAGMENTATION

- ✓ A datagram can travel through different networks. Each router decapsulates the IP datagram from the frame it receives, processes it, and then encapsulates it in another frame. The format and size of the received frame depend on the protocol used by the physical network through which the frame has just traveled. The format and size of the sent frame depend on the protocol used by the physical network through which the frame is going to travel.
- ✓ **Maximum Transfer Unit (MTU)**



  - Each data link layer protocol has its own frame format in most protocols. One of the fields defined in the format is the maximum size of the data field. In other words, when a datagram is encapsulated in a frame, the total size of the datagram must be less than this maximum size, which is defined by the restrictions imposed by the hardware and software used in the network.
  - The value of the MTU differs from one physical network protocol to another. For example, the value for the Ethernet LAN is 1500 bytes, for FDDI LAN is 4352 bytes, and for PPP is 296 bytes.
  - In order to make the IP protocol independent of the physical network, the designers decided to make the maximum length of the IP datagram equal to 65,535 bytes.
  - This makes transmission more efficient if we use a protocol with an MTU of this size.
  - However, for other physical networks, we must divide the datagram to make it possible to pass through these networks. This is called **fragmentation.**

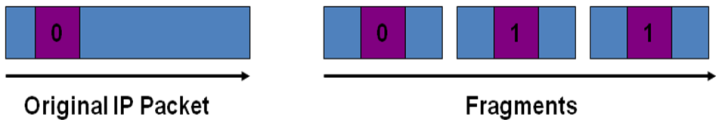
- ✓ **Fields Related to Fragmentation**
  - **Identification:**
    - This 16-bit field identifies a datagram originating from the source host. The combination of the identification and source IP address must uniquely define a datagram as it leaves the source host.
    - All fragments of a datagram contain the same identification value.
    - This allows the destination host to determine which fragment belongs to which datagram.

  - **Flags:**
    - This is a three-bit field.

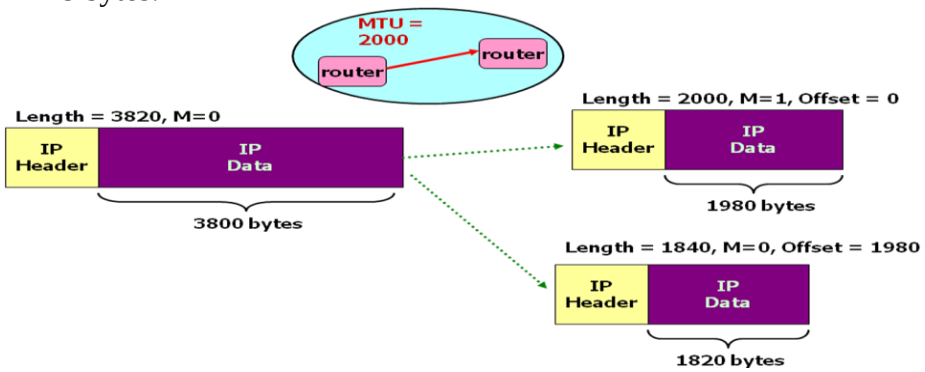
      - The first bit is reserved (not used).
      - The second bit (D) is called the **do not fragment bit.**
        - If its value is 1, the machine must not fragment the datagram

- If its value is 0, the datagram can be fragmented if necessary.

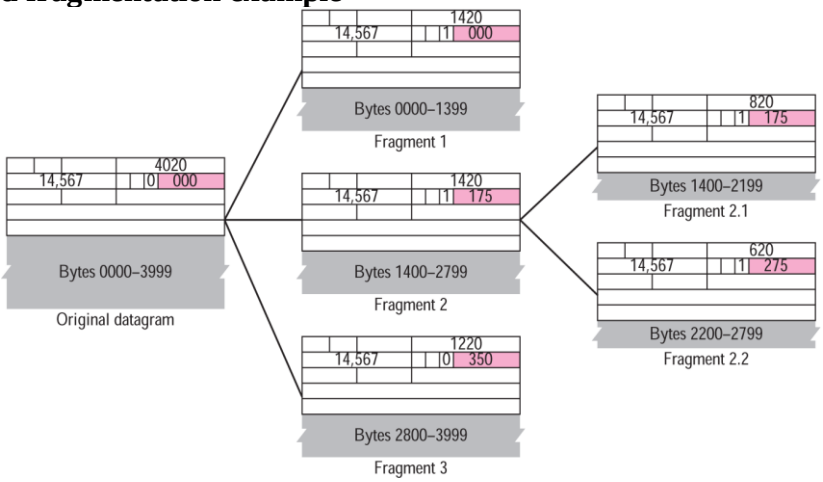
- The third bit (M) is called the **more fragment bit.**
  - If its value is 1, it means the datagram is not the last fragment; there are more fragments after this one.
  - If its value is 0, it means this is the last or only fragment.



- **Fragmentation offset:**
  - This 13-bit field shows the relative position of this fragment with respect to the whole datagram.
  - It is the offset of the data in the original datagram measured in units of 8 bytes.



- **Detailed fragmentation example**



- The figure also shows what happens if a fragment itself is fragmented. In this case the value of the offset field is always relative to the original datagram. For example, in the figure, the second fragment is itself fragmented later to two fragments of 800 bytes and 600 bytes, but the offset shows the relative position of the fragments to the original data. It is obvious that even if each fragment follows a different path and arrives out of order, the final destination host can reassemble the original datagram from the fragments received (if none of them is lost) using the following strategy:

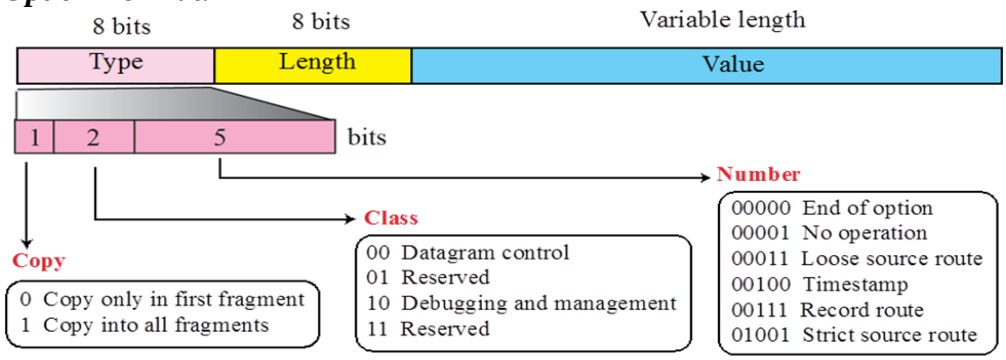
1. The first fragment has an offset field value of zero.

2. Divide the length of the first fragment by 8. The second fragment has an offset value equal to that result.
3. Divide the total length of the first and second fragment by 8. The third fragment has an offset value equal to that result.
4. Continue the process. The last fragment has a more bit value of 0.

OPTIONS

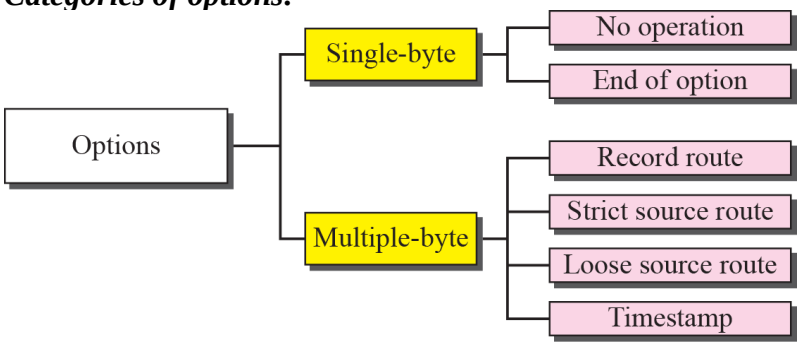
- ✓ The header of the IP datagram is made of two parts: a fixed part and a variable part. The fixed part is 20 bytes long and was discussed in the previous section. The variable part comprises the options, which can be a maximum of 40 bytes.
- ✓ Options, as the name implies, are not required for a datagram. They can be used for network testing and debugging.
- ✓ Options are not a required part of the IP header.
- ✓ Option processing is required of the IP software.

- ✓ ***Option Format:***

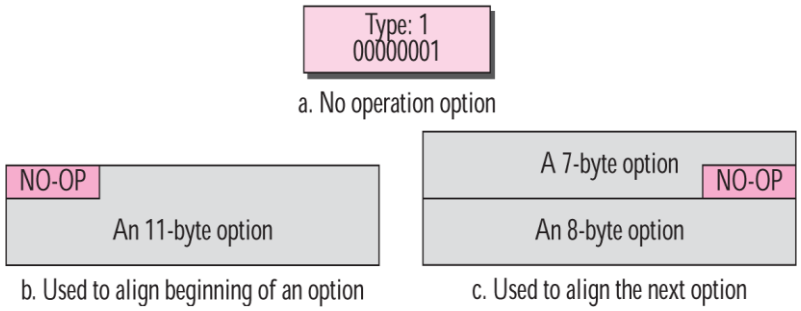


Copy - controls the presence of the option in fragmentation.
Class - defines the general purpose of the option.
Number – defines the type of option.
Length – defines the total length of the option.
(this field is not present in all of the option types.)
Value – contains the data that specific options require.

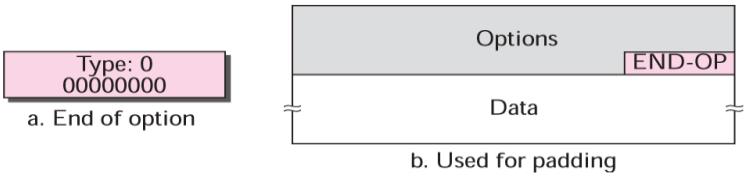
- ✓ ***Categories of options:***



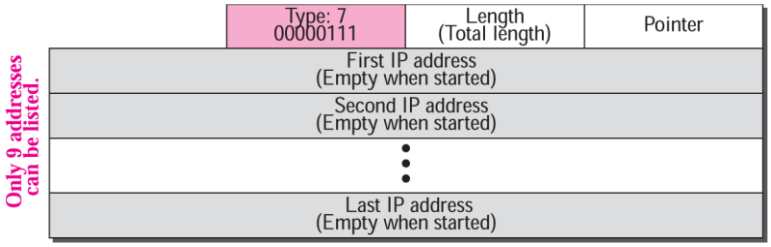
  - ***No Operation:***
    - It is used as a filter between options (or)
    - A "dummy option" used as "internal padding" to align certain options on a 32-bit boundary when required.
    - For example
    - It can be used to align the next option on a 16 bit or 32 bit boundary.



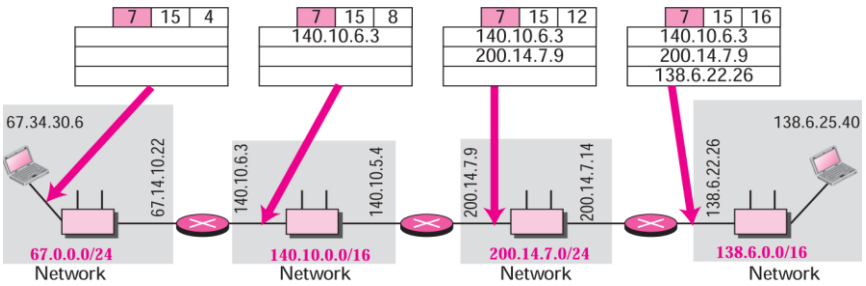
- ***End-of-option option:***
  - It is used to mark the end of a list of options (used as a last option).
  - After this options, the receiver looks for payload data.



- ***Record-route option:***
  - It is used to record the Internet routers that handle the datagram.
  - It can list up to 9 router IP addresses, since the maximum size of the header is 60 bytes, which must include 20 bytes for the base header. This implies that only 40 bytes are left over for the option part.
  - The source creates placeholder fields in the option to be filled by the visited routers.



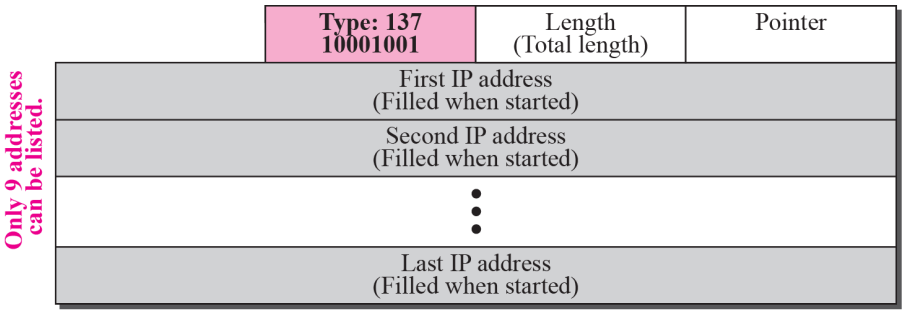
  - The pointer field is an offset integer field containing the byte number of the first empty entry. In other words, it points to the first available entry.
  - ***Record-route concept example***



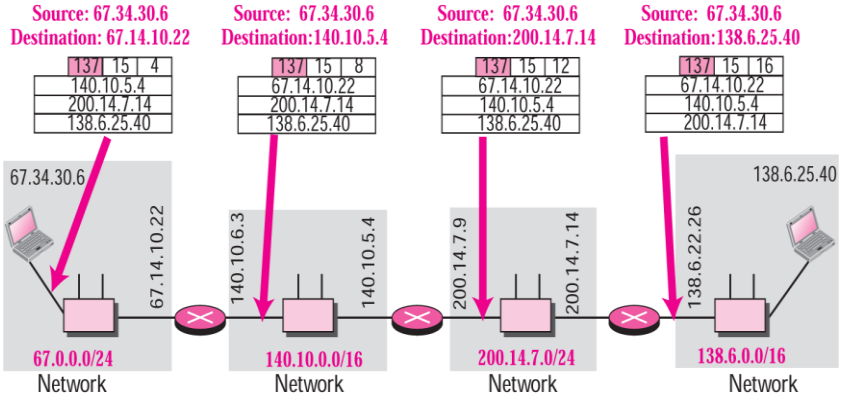
  - *The source creates empty fields for the IP addresses in the data field of the option. When the datagram leaves the source, all of the fields are empty. The pointer field has a value of 4, pointing to the first empty field.*

- *When the datagram is traveling, each router that processes the datagram compares the value of the pointer with the value of the length.*
- *If value of pointer > value of length, the option is full*
- *If value of pointer ! > value of length, the router inserts its outgoing IP address in the next empty field.*
- *In this case, the router adds the IP address of its interface from which the datagram is leaving. The router then increments the value of the pointer by 4.*

- ***Strict-source-route option:***
  - It is used by the source to predetermine a route for the datagram as it travels through the Internet.
  - The sender can choose a route with a specific type of service, such as minimum delay or maximum throughput.
  - If a datagram specifies a strict source route, all of the routers defined in the option must be visited by the datagram.
  - If the datagram visits a router that is not on the list, the datagram is discarded and an error message is issued.
  - If the datagram arrives at the destination and some of the entries were not visited, it will also be discarded and an error message issued.



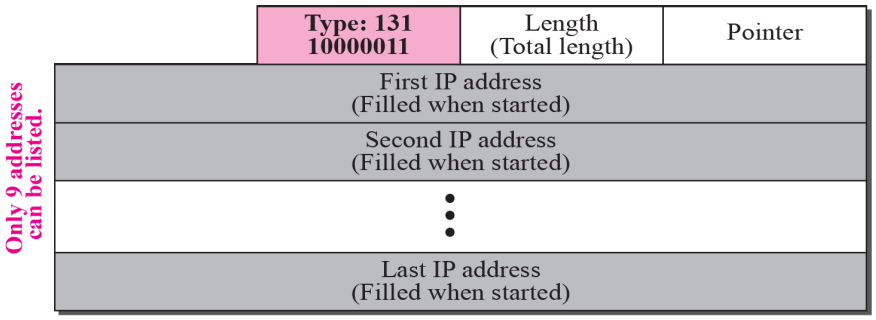
  - The format is similar to the record route option with the exception that all of the IP addresses are entered by the sender.



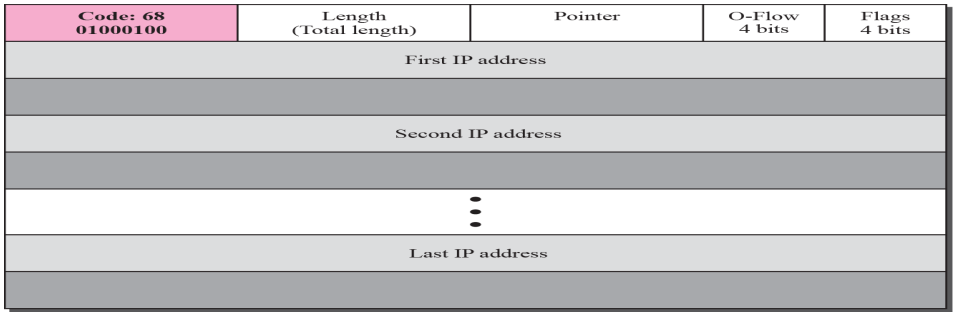
  - When the datagram is traveling, each router that processes the datagram compares the value of the pointer with the value of the length.

- If value of pointer > value of length, the datagram has visited all of the predefined routers. The datagram cannot travel anymore; it is discarded and an error message is created.
- If value of pointer ! > value of length, the router compares the destination IP address with its incoming IP address:
    - If they are equal, it processes the datagram, swaps the IP address pointed by the pointer with the destination address, increments the pointer value by 4, and forwards the datagram.
    - If they are not equal, it discards the datagram and issues an error message.

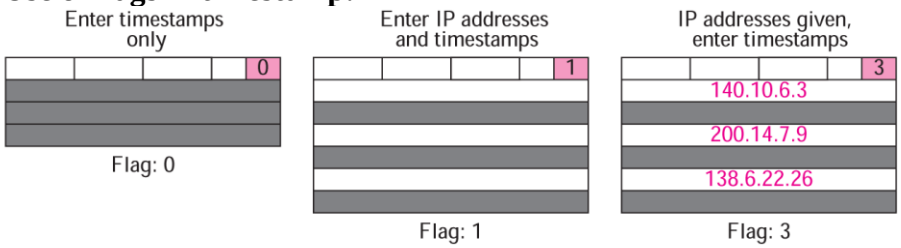
- ***Loose-source-route option:***
  - It is similar to the strict source route, but it is more relaxed.
  - Each router in the list must be visited, but the datagram can visit other routers as well.



- ***Time-stamp option:***
  - It is used to record the time of datagram processing by a router. The time is expressed in milliseconds from midnight, Universal Time.
  - Knowing the time a datagram is processed can help users and managers track the behavior of the routers in the Internet. We can estimate the time it takes for a datagram to go from one router to another.

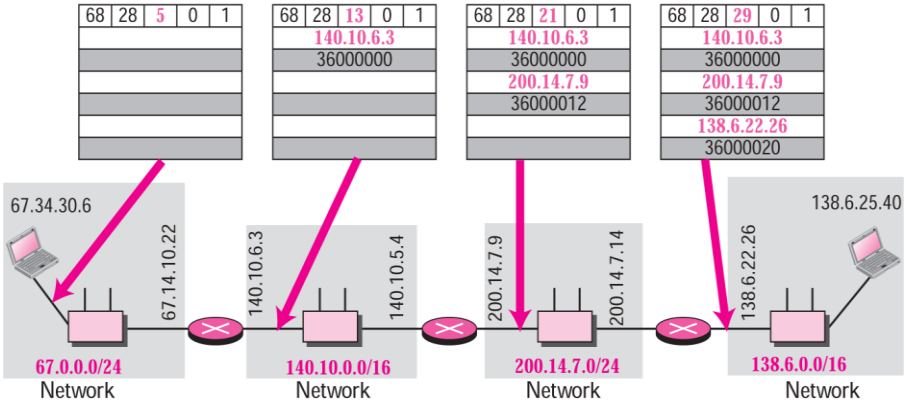


  - The overflow field records the number of routers that could not add their timestamp because no more fields were available.
  - The flags field specifies the visited router responsibilities.
  - **Use of flags in timestamp:**



    - If the flag value is 0, each router adds only the timestamp in the provided field.
    - If the flag value is 1, each router must add its outgoing IP address and the timestamp.
    - If the value is 3, the IP addresses are given, and each router must check the given IP address with its own incoming IP address.
      - If there is a match, the router overwrites the IP address with its outgoing IP address and adds the timestamp.

- ***Timestamp concept***
  - Figure shows the actions taken by each router when a datagram travels from source to destination. The figure assumes a flag value of 1.

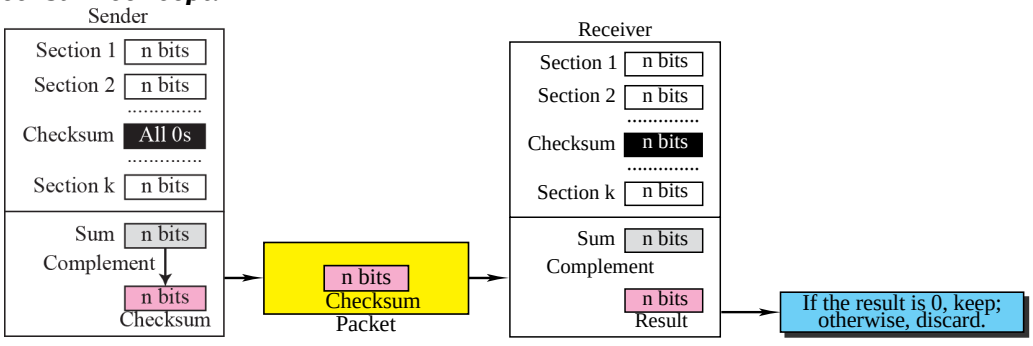


CHECKSUM

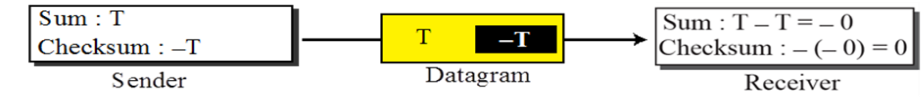
- The error detection method used by most TCP/IP protocols is called the checksum.
- The checksum protects against the corruption that may occur during the transmission of a packet.
- It is redundant information added to the packet.
- The checksum is calculated at the sender and the value obtained is sent with the packet.
- The receiver repeats the same calculation on the whole packet including the checksum. If the result is satisfactory, the packet is accepted; otherwise, it is rejected.

- **Checksum Calculation at the Sender**
  - The packet is divided into k sections, each of n bits (n is usually 16).
  - All sections are added together using one's complement arithmetic.
  - The final result is complemented to make the checksum.
- **Checksum Calculation at the Receiver**
  - The receiver divides the received packet into k sections and adds all sections.
  - It then complements the result. If the final result is 0, the packet is accepted; otherwise, it is rejected.

- ***Checksum concept:***



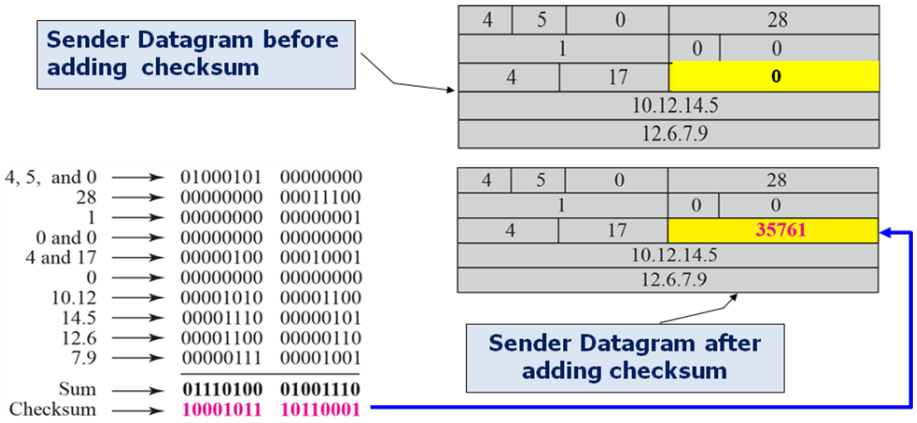
Checksum in one's complement arithmetic



- *Checksum in IP covers only the header, not the data.*

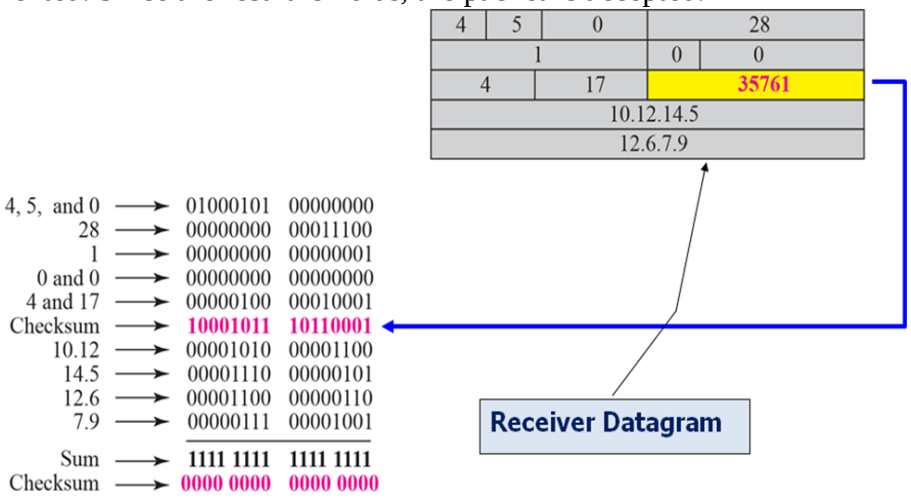
- ***Example of checksum calculation at the sender***

  - Figure shows an example of a checksum calculation at the sender site for an IP header without options. The header is divided into 16-bit sections. All the sections are added and the sum is complemented. The result is inserted in the checksum field.



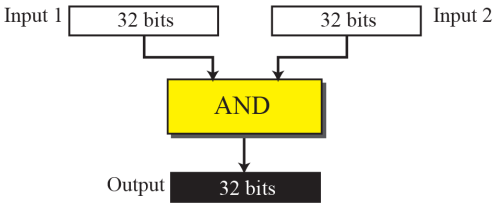
- ***Example of checksum calculation at the receiver***

  - Figure shows the checking of checksum calculation at the receiver site (or intermediate router) assuming that no errors occurred in the header. The header is divided into 16-bit sections. All the sections are added and the sum is complemented. Since the result is 16 0s, the packet is accepted.



IP ADDRESS

- ✓ An IPv4 address is 32 bits long.
- ✓ The IPv4 addresses are unique and universal.
- ✓ Address Space : The address space of IPv4 is 232 or 4,294,967,296.
- ✓ Notation :
  - Binary notation (base 2) – In binary notation, an IPv4 address is displayed as 32 bits. To make the address more readable, one or more spaces is usually inserted between each octet (8 bits). Each octet is often referred to as a byte.
    - 01110101 10010101 00011101 11101010
  - Dotted-decimal notation (base 256)- IPv4 address is usually written in decimal form with a decimal point (dot) separating the bytes.
    - 192.168.10.1
  - Hexadecimal notation (base 16) - Each hexadecimal digit is equivalent to four bits. This means that a 32-bit address has 8 hexadecimal digits.
    - 0X810B0BEF
- ✓ Operations :
  - Bitwise AND Operation:
    - The bitwise AND operation is a binary operation; it takes two inputs.
    - The AND operation compares the two corresponding bits in two inputs and selects the smaller bit from the two.
    - When the numbers are represented in dotted-decimal notation, we can use two short cuts.
    - When at least one of the two bytes is 0 or 255, the OR operation selects the smaller byte (or one of them if equal).
    - When none of the two bytes is 0 or 255, we can write each byte as the sum of eight terms, where each term is a power of 2. We then select the smaller term in each pair (or one of them if equal) and add them to get the result of OR operation.



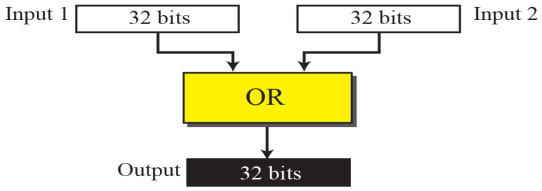
AND

Input 1	Input 2	Output
0	0	0
0	1	0
1	0	0
1	1	1

Operation for each bit

  - Bitwise OR Operation:
    - The bitwise OR operation is a binary operation; it takes two inputs.
    - The OR operation compares the two corresponding bits in two inputs and selects the larger bit from the two.
    - When the numbers are represented in dotted-decimal notation, we can use two short cuts.
    - When at least one of the two bytes is 0 or 255, the OR operation selects the larger byte (or one of them if equal).
    - When none of the two bytes is 0 or 255, we can write each byte as the sum of eight terms, where each term is a power of 2. We then select the larger

term in each pair (or one of them if equal) and add them to get the result of OR operation.

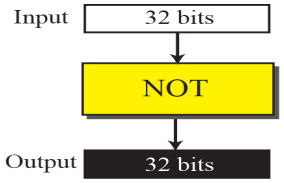


OR

Input 1	Input 2	Output
0	0	0
0	1	1
1	0	1
1	1	1

Operation for each bit

- Bitwise NOT Operation:
  - The bitwise NOT operation is a unary operation; it takes one input.
  - The NOT operation, when applied to a 32-bit number in binary format, inverts each bit. Every 0 bit is changed to a 1 bit; every 1 bit is changed to a 0 bit.
  - when the number is represented as a four-byte dotted-decimal notation, we can use a short cut; we can subtract each byte from 255.



NOT operation

Input	Output
0	1
1	0

Operation for each bit

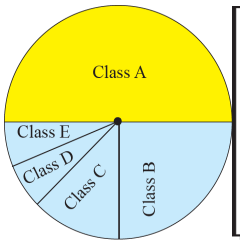
CLASSFUL ADDRESSES

✓ IP addresses, when started a few decades ago, used the concept of classes. This architecture is called Classful addressing.

✓ In the mid-1990s, a new architecture, called classless addressing, was introduced that supersedes the original architecture.

✓ Classes:

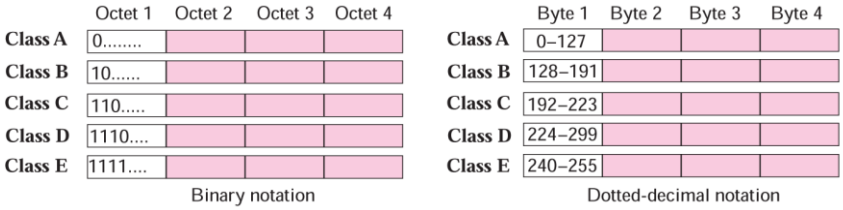
- In Classful addressing, the IP address space is divided into five classes: A, B, C, D, and E. Each class occupies some part of the whole address space.



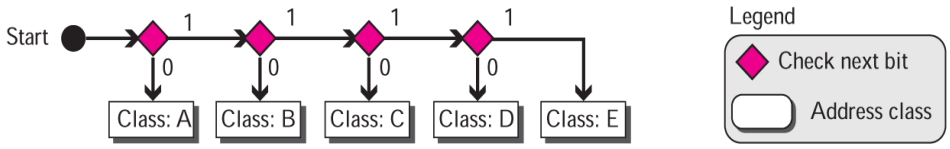
Class A: 2^{31} = 2,147,483,648 addresses, 50%
Class B: 2^{30} = 1,073,741,824 addresses, 25%
Class C: 2^{29} = 536,870,912 addresses, 12.5%
Class D: 2^{28} = 268,435,456 addresses, 6.25%
Class E: 2^{28} = 268,435,456 addresses, 6.25%

✓ Finding the class of address

- Find the class of an address when the address is given either in binary or dotted-decimal notation.
- In the binary notation, the first few bits can immediately tell us the class of the address.
- In the dotted-decimal notation, the value of the first byte can give the class of an address.



✓ Finding the class of an address using continuous checking



Example

1. Find the class of each address:

a) 00000001 00001011 00001011 11101111
b) 11000001 10000011 00011011 11111111
c) 10100111 11011011 10001011 01101111
d) 11110011 10011011 11111011 00001111

Solution:

a) The first bit is 0. This is a class A address.
b) The first 2 bits are 1; the third bit is 0. This is a class C address.
c) The first bit is 1; the second bit is 0. This is a class B address.
d) The first 4 bits are 1s. This is a class E address.

2. Find the class of each address:

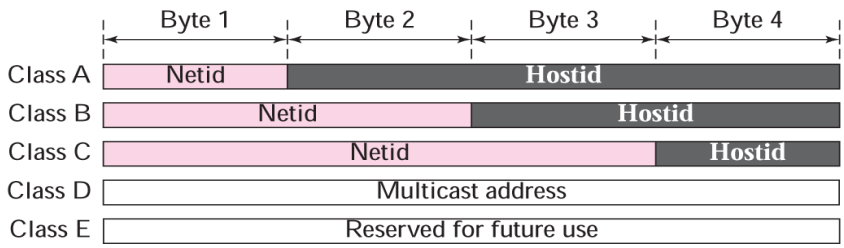
e) 227.12.14.87
f) 193.14.56.22
g) 14.23.120.8
h) 252.5.15.111

Solution:

e) The first byte is 227 (between 224 and 239); the class is D.
f) The first byte is 193 (between 192 and 223); the class is C.
g) The first byte is 14 (between 0 and 127); the class is A.
h) The first byte is 252 (between 240 and 255); the class is E.

✓ Netid and Hostid

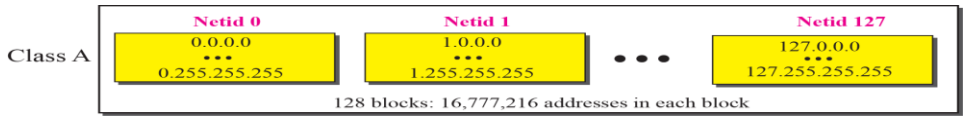
- In classful addressing, an IP address in classes A, B, and C is divided into netid and hostid. These parts are of varying lengths, depending on the class of the address. Figure shows the netid and hostid bytes.
- Classes D and E are not divided into netid and hostid.



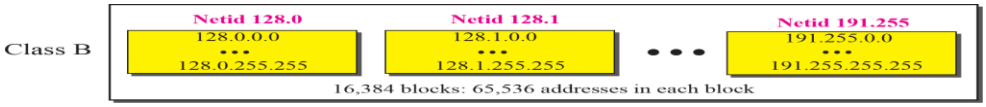
- In class A, 1 byte defines the netid and 3 bytes define the hostid.
- In class B, 2 bytes define the netid and 2 bytes define the hostid.
- In class C, 3 bytes define the netid and 1 byte defines the hostid.

✓ Classes and Blocks

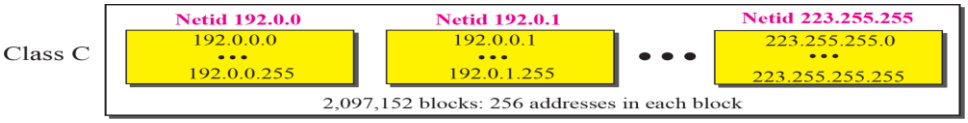
- Each class is divided into a fixed number of blocks with each block having a fixed size.
- **Class A:**
  - Only 1 byte in class A defines the netid and the leftmost bit should be 0, the next 7 bits can be changed to find the number of blocks in this class.
  - Therefore, class A is divided into 2^7 = 128 blocks that can be assigned to 128 organizations.
  - Each block in this class contains 16,777,216 addresses.
  - Many addresses are wasted in this class.



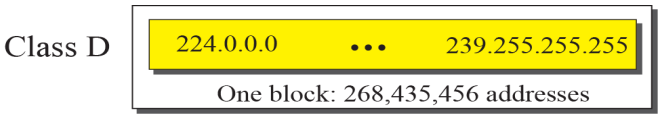
- **Class B:**
  - Two bytes in class B defines the netid and the leftmost bit should be 10, the next 14 bits can be changed to find the number of blocks in this class.
  - Therefore, class B is divided into 2^14 = 16,384 blocks that can be assigned to 16,384 organizations.
  - Each block in this class contains 65,536 addresses.
  - Many addresses are wasted in this class



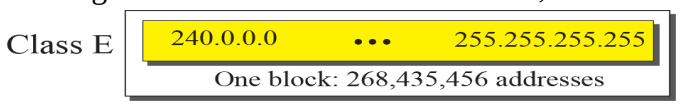
- **Class C**
  - Three bytes in class C defines the netid and the leftmost bit should be 110, the next 21 bits can be changed to find the number of blocks in this class.
  - Therefore, class B is divided into 2^21 = 2,097,152 blocks that can be assigned to 2,097,152 organizations.
  - Each block in this class contains 256 addresses.



- **Class D**
  - One block of class D addresses.
  - It is designed for multicasting.
  - Each address in this class is used to define one group of hosts on the Internet.
  - When a group is assigned an address in this class, every host that is a member of this group will have a multicast address in addition to its normal (unicast) address.

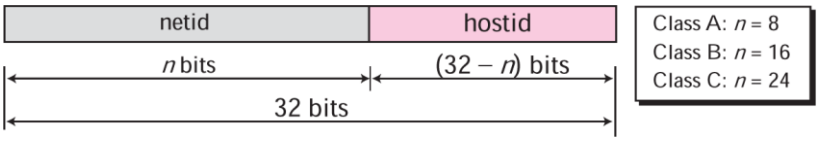


- **Class E**
  - One block of class E addresses.
  - It was designed for use as reserved addresses,

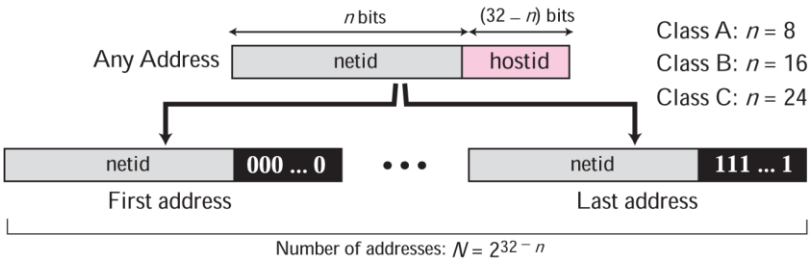


✓ Two-Level Addressing

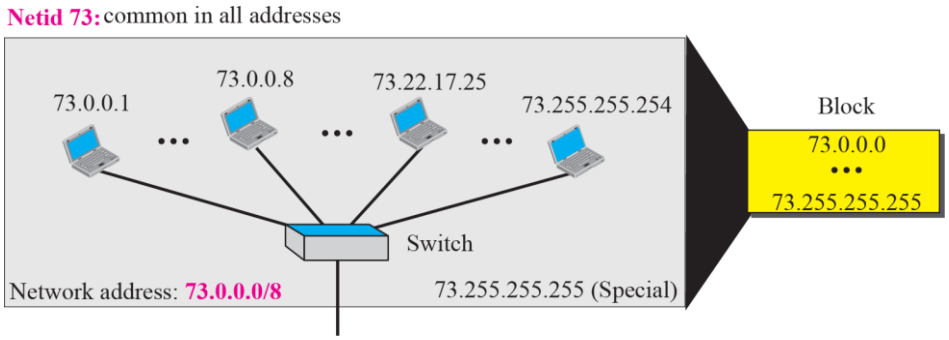
- The whole purpose of IPv4 addressing is to define a destination for an Internet packet.
- When classful addressing was designed, it was assumed that the whole Internet is divided into many networks and each network connects many hosts.
- A network was normally created by an organization that wanted to be connected to the Internet.
- The Internet authorities allocated a block of addresses to the organization (in class A, B, or C).
- Each address in classful addressing contains two parts: netid and hostid.
  - The netid defines the network;
  - The hostid defines a particular host connected to that network.



- *Information extraction in Classful addressing:*
  - A block is a range of addresses. Given any address in the block, we normally like to know three pieces of information about the block: the number of addresses, the first address, and the last address.
  - We can now find these three pieces of information as shown in Figure
    1. The number of addresses in the block, N, can be found using $N = 2^{32-n}$.
    2. To find the first address, we keep the n leftmost bits and set the (32 -n) rightmost bits all to 0s.
    3. To find the last address, we keep the n leftmost bits and set the (32-n) rightmostbits all to 1s.



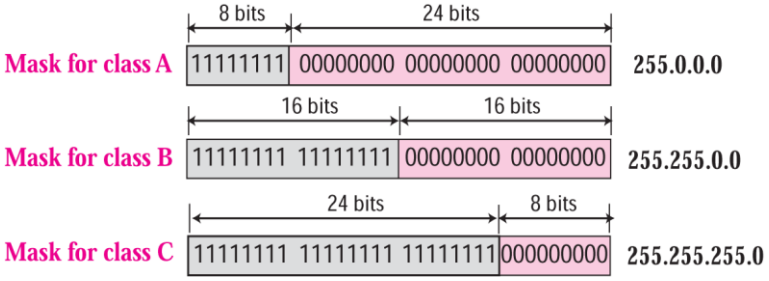
- *Example Problem:*
  - *An address in a block is given as 73.22.17.25. Find the number of addresses in the block, the first address, and the last address.*



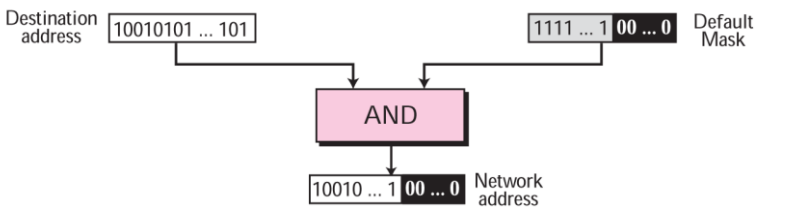
  - *Solution:*
    Figure shows a possible configuration of the network that uses this block.
    1. The number of addresses in this block is N = 232−n = 16,777,216.
    2. To find the first address, we keep the leftmost 8 bits and set the rightmost 24 bits all to 0s. The first address is 73.0.0.0/8, in which 8 is the value of n.
    3. To find the last address, we keep the leftmost 8 bits and set the rightmost 24 bits all to 1s. The last address is 73.255.255.255.

✓ Network Mask

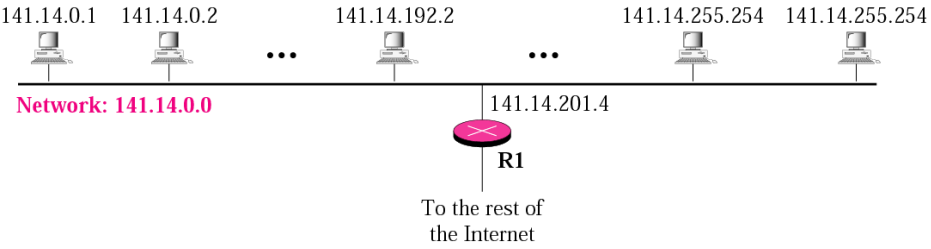
- The methods we described previously for extracting the network address are mostly used to show the concept. The routers in the Internet normally use an algorithm to extract the network address from the destination address of a packet. To do this, we need a network mask. A network mask or a default mask in classful addressing is a 32-bit number with n leftmost bits all set to 1s and (32 − n) rightmost bits all set to 0s.



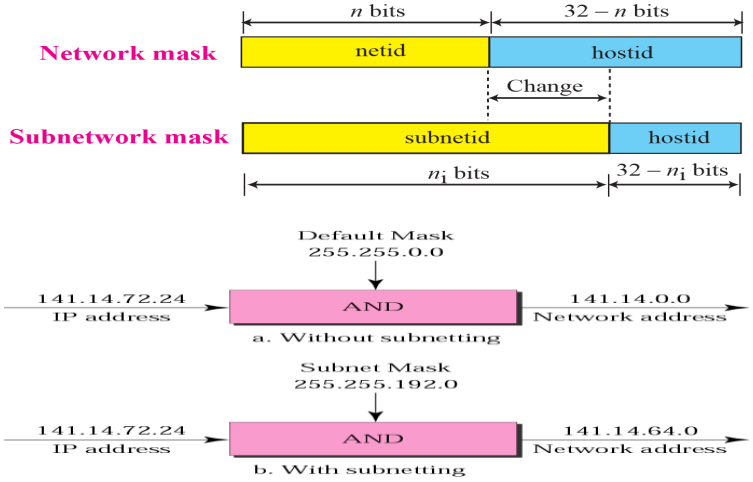
- To extract the network address from the destination address of a packet, a router uses the AND operation.



- ✓ Subnetting
  - A network is divided into several smaller subnetworks (subnets) with each subnetwork having its own subnetwork address.
  - A network with two levels of hierarchy (not subnetted)



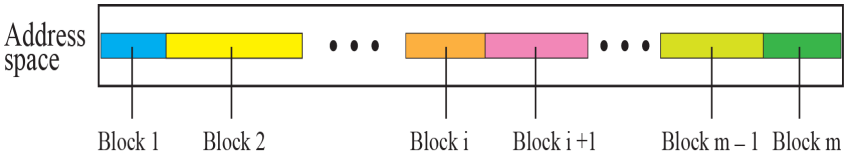
  - Default mask and subnet mask
    - The network mask is used when a network is not subnetted.
    - When we divide a network to several subnetworks, we need to create a subnetwork mask (or subnet mask) for each subnetwork



CLASSLESS ADDRESSING

- ✓ Subnetting and supernetting in classful addressing did not really solve the address depletion problem. With the growth of the Internet, it was clear that a larger address space was needed as a long-term solution. Although the long-range solution has already been devised and is called IPv6, a short-term solution was also devised to use the same address space but to change the distribution of addresses to provide a fair share to each organization. The short-term solution still uses IPv4 addresses, but it is called classless addressing.
- ✓ Variable-Length Blocks
  - In classless addressing, the whole address space is divided into variable length blocks.
- ✓ Number of Addresses in a Block
  - There is only one condition on the number of addresses in a block; it must be a power of 2 (2, 4, 8, . . .).
    - A household may be given a block of 2 addresses.
    - A small business may be given 16 addresses.

- A large organization may be given 1024 addresses.



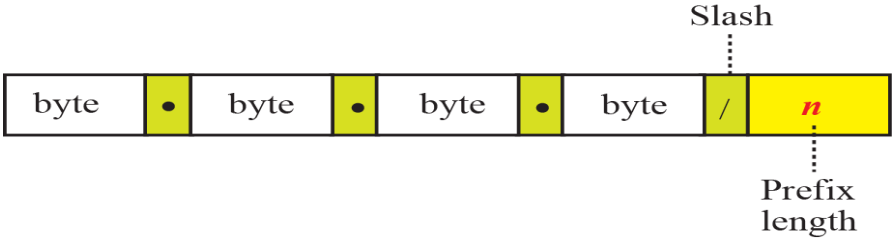
✓ Two-Level Addressing

- In classful addressing, two-level addressing was provided by dividing an address into *netid and hostid.*
  - The netid defined the network
  - The hostid defined the host in the network.
- In classless addressing, the block is actually divided into two parts, the prefix and the suffix.
  - The prefix plays the same role as the netid
  - The suffix plays the same role as the hostid.
- All addresses in the block have the same prefix; each address has a different suffix.

- The prefix length in classless addressing can be 1 to 32.

✓ **Slash notation**

- In classless addressing, we need to include the prefix length to each address if we need to find the block of the address.
- In this case, the prefix length, n, is added to the address separated by a slash. The notation is informally referred to as slash notation.
- The slash notation is formally referred to as classless interdomain routing or CIDR notation.



✓ **Example**

1. A small organization is given a block with the beginning address and the prefix length 205.16.37.24/29 (in slash notation). What is the range of the block?

Solution

The beginning address is 205.16.37.24. To find the last address we keep the first 29 bits and change the last 3 bits to 1s.

Beginning:11001111 00010000 00100101 00011000

Ending : 11001111 00010000 00100101 00011111

There are only 8 addresses in this block.

2. What is the network address if one of the addresses is 167.199.170.82/27?

Solution

The prefix length is 27, which means that we must keep the first 27 bits as it is and change the remaining bits (5) to 0s. The 5 bits affect only the last byte. The last byte is 01010010. Changing the last 5 bits to 0s, we get 01000000 or 64. The network address is 167.199.170.64/27.

SPECIAL ADDRESSES

All-Zero's Address

- The block 0.0.0.0/32, which contains only one single address, is reserved for communication when a host needs to send an IPv4 packet but it does not know its own address.
- This is normally used by a host at bootstrap time when it does not know its IPv4 address.
- The host sends an IPv4 packet to a bootstrap server {called DHCP(Dynamic Host Configuration Protocol) server} using this address as the source address and a **limited broadcast address** as the destination address to find its own address

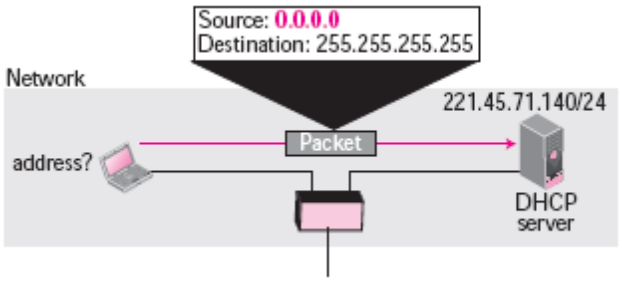


Fig: Example of using All-Zero's Address

All-One's Address: Limited Broadcast Address

- The block 255.255.255.255/32, which contains one single address, is reserved for limited broadcast address in the current network.
- A host that wants to send a message to every other host can use this address as a destination address in an IPv4 packet.
- However, a router will block a packet having this type of address to confine the broadcasting to the local network.
- A host sends a datagram using a destination IPv4 address consisting of all 1s.
- All devices on this network receive and process this datagram.

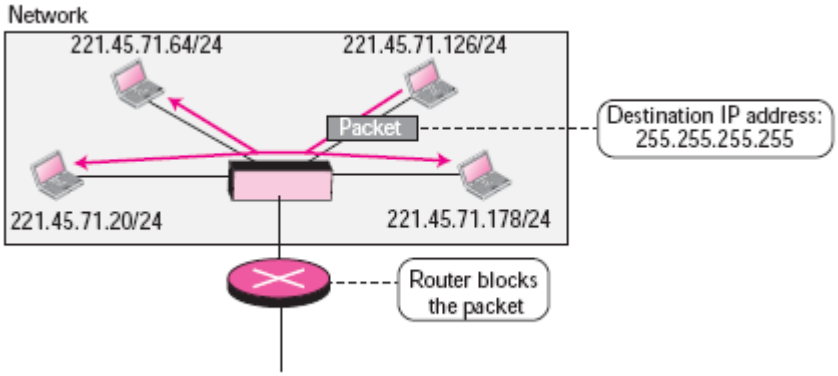


Fig: Example of All-One's Address

Loopback Addresses

- The block 127.0.0.0/8 is used for the **loopback address,** which is an address used to test the software on a machine.
- When this address is used, a packet never leaves the machine; it simply returns to the protocol software. It can be used to test the IPv4 software.
- For example, an application such as "ping" can send a packet with a loopback address as the destination address to see if the IPv4 software is capable of receiving and processing a packet.
- As another example, the loopback address can be used by a *client process* (a running application program) to send a message to a server process on the same machine. Note that this can be used only as a destination address in an IPv4 packet.

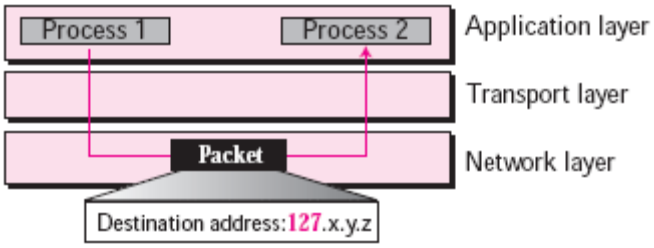


Fig: Example of Loopback Address

Private Addresses

- A number of blocks are assigned for private use.
- They are not recognized globally.
- These blocks are depicted shown in the Table below.
- These addresses are used either in isolation or in connection with network address translation techniques(NAT).

Block	*Number of addresses*	*Block*	*Number of addresses*
10.0.0.0/8	16,777,216	192.168.0.0/16	65,536
172.16.0.0/12	1,047,584	169.254.0.0/16	65,536

Table: Addresses for Private Networks

Multicast Addresses

- The block 224.0.0.0/4 is reserved for multicast communication.