Chapter 2: Internet Protocol Overview

A protocol is a set of rules that make communication on a network more efficient. For example, while flying an airplane, pilots obey very specific rules for communication with other airplanes and with air traffic control.

OSI Model

An architectural model for open networking systems that was developed by the International Organization for Standardization (ISO) in Europe in 1974. The Open Systems Interconnection (OSI) reference model was intended as a basis for developing universally accepted networking protocols, but this initiative essentially failed for the following reasons.

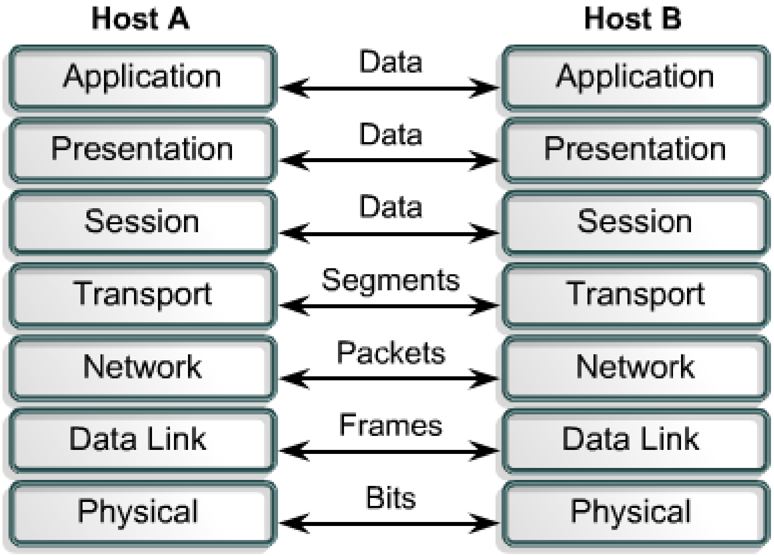
- The standards process was relatively closed compared with the open standards process used by the Internet Engineering Task Force (IETF) to develop the TCP/IP protocol suite.
- The model was overly complex. Some functions (such as connectionless communication) were neglected, while others (such as error correction and flow control) were repeated at several layers.
- The growth of the Internet and TCP/IP-a simpler, real-world protocol model-pushed the OSI reference model out.

Benefits of OSI Model:

- It breaks network communication into smaller, more manageable parts.
- It standardizes network components to allow multiple vendor development and support.
- It allows different types of network hardware and software to communicate with each other.
- It prevents changes in one layer from affecting other layers.
- It divides network communication into smaller parts to make learning it easier to understand.

Peer-to-Peer Communication:

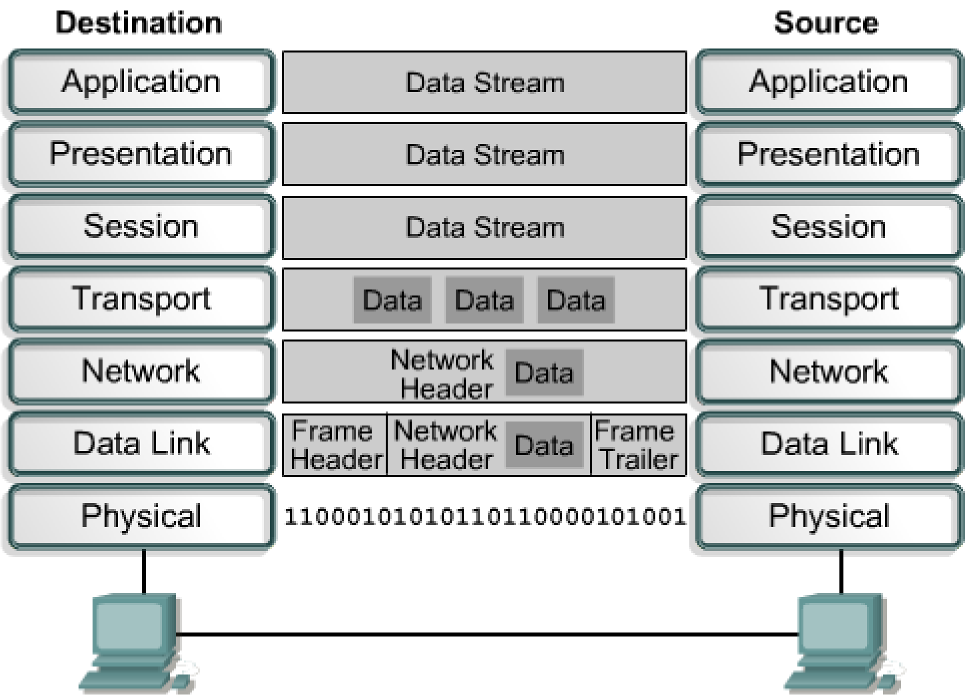
In order for data to travel from the source to the destination, each layer of the OSI model at the source must communicate with its peer layer at the destination. This form of communication is referred to as peer-to-peer. During this process, the protocols of each layer exchange information, called protocol data units (PDUs). Each layer of communication on the source computer communicates with a layer specific PDU, and with its peer layer on the destination computer as illustrated in Figure below.



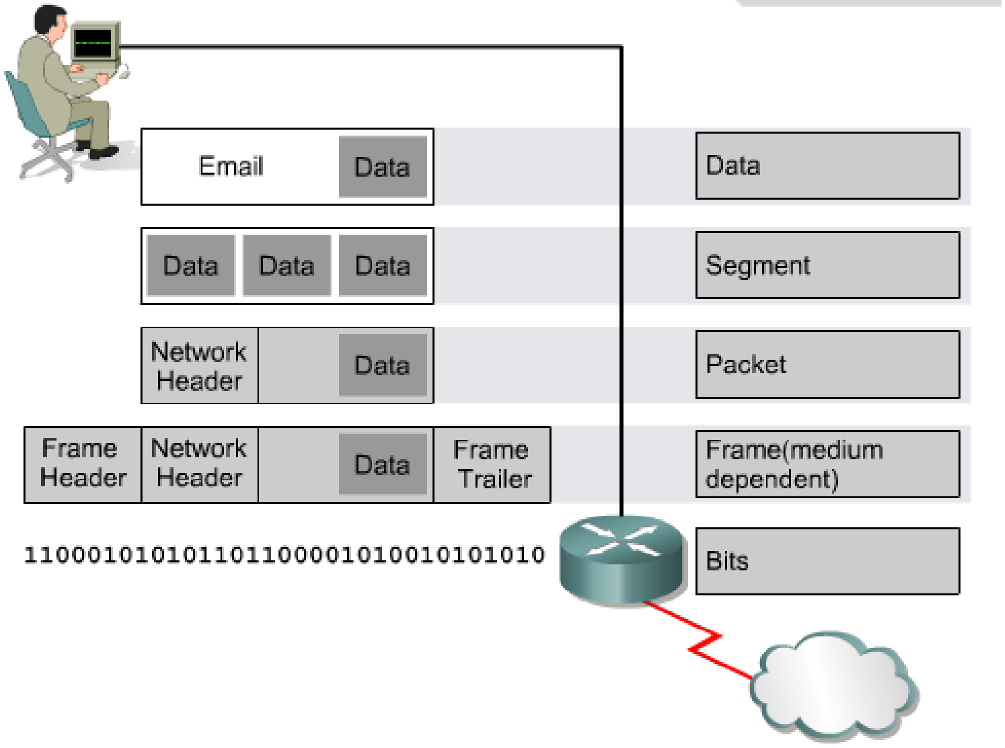
Data packets on a network originate at a source and then travel to a destination. Each layer depends on the service function of the OSI layer below it. To provide this service, the lower layer uses encapsulation to put the PDU from the upper layer into its data field. Then it adds whatever headers and trailers the layer needs to perform its function. Next, as the data moves down through the layers of the OSI model, additional headers and trailers are added.

Data Encapsulation:

All communications on a network originate at a source, and are sent to a destination. The information sent on a network is referred to as data or data packets. If one computer (host A) wants to send data to another computer (host B), the data must first be packaged through a process called encapsulation. Encapsulation wraps data with the necessary protocol information before network transit. Therefore, as the data packet moves down through the layers of the OSI model, it receives headers, trailers, and other information.



Data Encapsulation example:



Network performs the following five conversion steps in order to encapsulate the data

1. Build the data.
2. Package the data for end-to-end transport.
3. Add the network IP address to the header.
4. Add the data link layer header and trailer.
5. Convert to bits for transmission.

Physical Layer:
Physical layer is the bottom layer of the OSI reference model. The physical layer has four important characteristics.

- **Mechanical:** Relates to the physical properties of the interface to a transmission medium. Typically, the specification is of a pluggable connector that joins one or more signal conductors, called circuits.

- **Electrical:** Relates to the representation of bits (e.g., in terms of voltage levels) and the data transmission rate of bits. It defines the voltage, current, modulation, bit synchronization, connection activation and deactivation, and various electrical characteristics for the transmission media (such as unshielded or shielded twisted-pair cabling, coaxial cabling, and fiber-optic cabling).

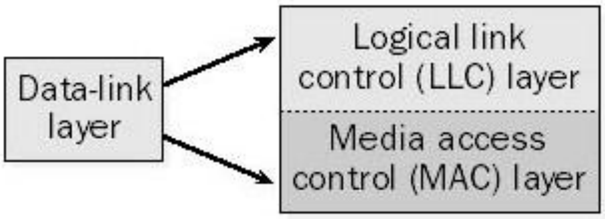
- **Functional:** Specifies the functions performed by individual circuits of the physical interface between a system and the transmission medium.

- **Procedural:** Specifies the sequence of events by which bit streams are exchanged across the physical medium.

Data Link Layer:
The physical layer provides only a raw bit-stream service, the data link layer attempts to make the physical link reliable while providing the means to activate, maintain, and deactivate the link.

For LANs, the Project 802 standards of the Institute of Electrical and Electronics Engineers (IEEE) separate the data-link layer into two sub layers:

- **The logical link control (LLC) layer,** the upper of the two layers, which is responsible for flow control, error correction, and re-sequencing functions for connection-oriented communication, but which also supports connectionless communication
- **The media access control (MAC) layer,** the lower of the two layers, which is responsible for providing a method for stations to gain access to the medium


Data-link layer
Logical link control (LLC) layer
Media access control (MAC) layer

Functions:

- **Framing:** The data link layer divides the stream of bits received from the network layer into manageable data units called frames.

- **Physical addressing:** If frames are to be distributed to different systems on the network, the data link layer adds a header to the frame to define the sender and/or receiver of the frame. If the frame is intended for a system outside the sender's network, the receiver address is the address of the device that connects the network to the next one.

- **Flow control:** If the rate at which the data are absorbed by the receiver is less than the rate at which data are produced in the sender, the data link layer imposes a flow control mechanism to avoid overwhelming the receiver.

- **Error control**: The data link layer adds reliability to the physical layer by adding mechanisms to detect and retransmit damaged or lost frames. It also uses a mechanism to recognize duplicate frames. Error control is normally achieved through a trailer added to the end of the frame.

- **Access control:** When two or more devices are connected to the same link, data link layer protocols are necessary to determine which device has control over the link at any given time.

Examples of data-link protocols for local area networking include the following:

- IEEE 802.3, which provides the Carrier Sense Multiple Access with Collision Detection(CSMA/CD) access method for baseband Ethernet networks
- IEEE 802.5, which provides the token-passing access method for baseband token ring implementations

For WANs, data-link layer protocols encapsulate LAN traffic into frames suitable for transmission over WAN links. Common data-link encapsulation methods for WAN transmission include the following:

- Point-to-point technologies such as Point-to-Point Protocol (PPP) and High-level Data Link Control (HDLC) protocol
- Multipoint technologies such as frame relay, Asynchronous Transfer Mode (ATM), Switched Multimegabit Data Services (SMDS), and X.25

Network Layer:

Layer 3 of the Open Systems Interconnection (OSI) reference model for networking. The network layer is responsible for functions such as the following:

- Logical addressing and routing of packets over the network
- Establishing and releasing connections and paths between two nodes on a network
- Transferring data, generating and confirming receipts, and resetting connections

The network layer also supplies connectionless and connection-oriented services to the transport layer above it. The network layer functions closely with the physical layer (layer 1) and data-link layer (layer 2) in most real-world network protocol implementations.

On TCP/IP-based networks, IP addresses and network numbers are used at the network layer, and IP routers perform their routing functions at this layer. An example of an OSI model network layer protocol is the X.25 packet-switching network layer protocol, which is built on the X.21 physical layer protocol.

Transport Layer:

Layer 4 of the Open Systems Interconnection (OSI) reference model. The transport layer is responsible for providing reliable transport services to the upper-layer protocols. These services include the following:

- Flow control to ensure that the transmitting device does not send more data than the receiving device can handle.
- Packet sequencing for segmentation of data packets and remote reassembly.
- Error handling and acknowledgments to ensure that data is retransmitted when required.
- Multiplexing for combining data from several sources for transmission over one data path.
- Virtual circuits for establishing sessions between communicating stations.

The Transmission Control Protocol (TCP) of the TCP/IP protocol suite resides at the transport layer.

A connection between two devices that acts as though it's a direct connection even though it may physically be circuitous. The term is used most frequently to describe connections between two hosts in a packet-switching network. In this case, the two hosts can communicate as though they have a dedicated connection even though the packets might actually travel very different routes before arriving at their destination. An X.25 connection is an example of a virtual circuit. Virtual circuits can be either permanent (called PVCs) or temporary (called SVCs).

Session Layer:

Layer 5 of the Open Systems Interconnection (OSI) reference model, which enables sessions between computers on a network to be established and terminated. The session layer does not concern itself with issues such as the reliability and efficiency of data transfer between stations because these functions are provided by the first four layers of the OSI reference model.

Functions:

- **Dialog control:** The session layer allows two systems to enter into a dialog. It allows the communication between two processes to take place in either half- duplex (one way at a time) or full-duplex (two ways at a time) mode.

- **Synchronization:** The session layer allows a process to add checkpoints, or synchronization points, to a stream of data. For example, if a system is sending a file of 2000 pages, it is advisable to insert checkpoints after every 100 pages to ensure that each 100-page unit is received and acknowledged independently. In this case, if a crash happens during the transmission of page 523, the only pages that need to be resent after system recovery are pages 501 to 523. Pages previous to 501 need not be resent.

Presentation Layer:
The presentation layer is concerned with the syntax and semantics of the information exchanged between two systems.

Specific responsibilities of the presentation layer include the following:

- **Translation:** The processes (running programs) in two systems are usually exchanging information in the form of character strings, numbers, and so on. The information must be changed to bit streams before being transmitted. Because different computers use different encoding systems, the presentation layer is responsible for interoperability between these different encoding methods. The presentation layer at the sender changes the information from its sender-dependent format into a common format. The presentation layer at the receiving machine changes the common format into its receiver-dependent format.

- **Encryption:** To carry sensitive information, a system must be able to ensure privacy. Encryption means that the sender transforms the original information to another form and sends the resulting message out over the network. Decryption reverses the original process to transform the message back to its original form.

- **Compression:** Data compression reduces the number of bits contained in the information. Data compression becomes particularly important in the transmission of multimedia such as text, audio, and video.

Application layer:
Layer 7 of the Open Systems Interconnection (OSI) reference model, in which network-aware, user controlled software is implemented—for example, e-mail, file transfer utilities, and terminal access. The application layer represents the window between the user and the network. Examples of protocols that run at the application layer include File Transfer Protocol (FTP), Hypertext Transfer Protocol (HTTP), telnet, and similar protocols that can be implemented as utilities the user can interface with.

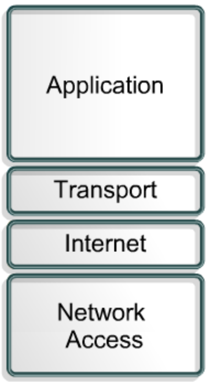
File transfer, access, and management: This application allows a user to access files in a remote host (to make changes or read data), to retrieve files from a remote computer for use in the local computer, and to manage or control files in a remote computer locally.

Mail services: This application provides the basis for e-mail forwarding and storage.

Directory services: This application provides distributed database sources and access for global information about various objects and services.

TCP/IP Model:

The U.S. Department of Defense (DOD) created the TCP/IP reference model because it wanted a network that could survive any conditions



Application Layer:
The application layer handles high-level protocols, representation, encoding, and dialog control. The TCP/IP protocol suite combines all application related issues into one layer. It ensures that the data is properly packaged before it is passed on to the next layer.
TCP/IP includes Internet and transport layer specifications such as IP and TCP as well as specifications for common applications. TCP/IP has protocols to support file transfer, e-mail, and remote login, in addition to the following:

- **File Transfer Protocol (FTP)** – FTP is a reliable, connection-oriented service that uses TCP to transfer files between systems that support FTP. It supports bidirectional binary file and ASCII file transfers.

- **Trivial File Transfer Protocol (TFTP)** – TFTP is a connectionless service that uses the User Datagram Protocol (UDP). TFTP is used on the router to transfer configuration files and Cisco IOS images, and to transfer files between systems that support TFTP. It is useful in some LANs because it operates faster than FTP in a stable environment.

- **Network File System (NFS)** – NFS is a distributed file system protocol suite developed by Sun Microsystems that allows file access to a remote storage device such as a hard disk across a network.

- **Simple Mail Transfer Protocol (SMTP) –** SMTP administers the transmission of e-mail over computer networks. It does not provide support for transmission of data other than plain text.

- **Telnet** – Telnet provides the capability to remotely access another computer. It enables a user to log into an Internet host and execute commands. A Telnet client is referred to as a local host. A Telnet server is referred to as a remote host.

- **Simple Network Management Protocol (SNMP) – SNMP** is a protocol that provides a way to monitor and control network devices. SNMP is also used to manage configurations, statistics, performance, and security.

- **Domain Name System (DNS)** – DNS is a system used on the Internet to translate domain names and publicly advertised network nodes into IP addresses.

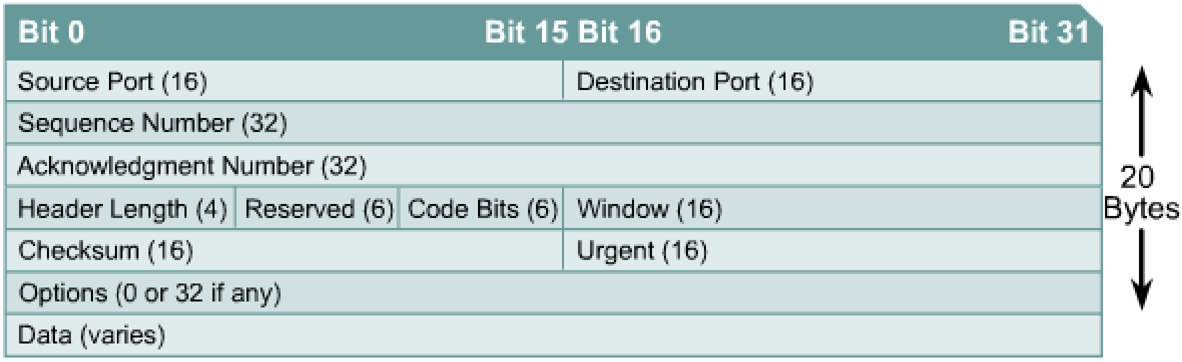
Transport Layer:
The transport layer provides a logical connection between a source host and a destination host. Transport protocols segment and reassemble data sent by upper-layer applications into the same data stream, or logical connection, between end points.

- Creates packet from bytes stream received from the application layer.
- Uses port number to create process to process communication.
- Uses a sliding window protocol to achieve flow control.
- Uses acknowledgement packet, timeout and retransmission to achieve error control.

The primary duty of the transport layer is to provide end-to-end control and reliability as data travels through this cloud. This is accomplished through the use of sliding windows, sequence numbers, and acknowledgments. The transport layer also defines end-to-end connectivity between host applications. Transport layer protocols include TCP and UDP.

TCP is a connection-oriented transport layer protocol that provides reliable full-duplex data transmission. TCP is part of the TCP/IP protocol stack. In a connection-oriented environment, a connection is established between both ends before the transfer of information can begin. TCP breaks messages into segments, reassembles them at the destination, and resends anything that is not received. TCP supplies a virtual circuit between end-user applications.

TCP Header Format:
TCP uses only a single type of protocol data unit, called a **TCP segment**. The header is shown in Figure below. Because one header must serve to perform all protocol mechanisms, it is rather large, with a minimum length of 20 octets.



Bit 0		Bit 15	Bit 16 Bit 31
Source Port (16)			Destination Port (16)
Sequence Number (32)			
Acknowledgment Number (32)			
Header Length (4)	Reserved (6)	Code Bits (6)	Window (16)
Checksum (16)			Urgent (16)
Options (0 or 32 if any)			
Data (varies)			

The following protocols use TCP:

- FTP
- HTTP
- SMTP
- Telnet

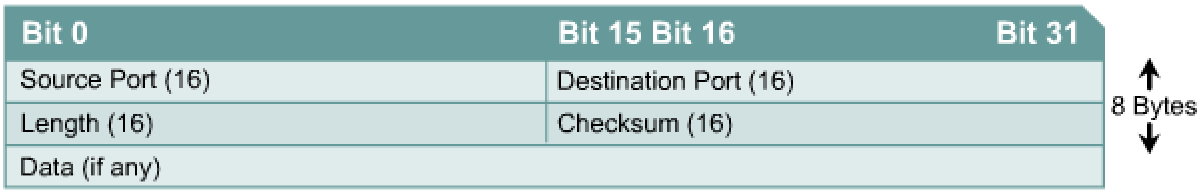
The following are the definitions of the fields in the TCP segment:

- **Source port** – Number of the port that sends data
- **Destination port** – Number of the port that receives data
- **Sequence number** – Number used to ensure the data arrives in the correct order
- **Acknowledgment number** – Next expected TCP octet
- **HLEN** – Number of 32-bit words in the header
- **Reserved** – Set to zero
- **Code bits** – Control functions, such as setup and termination of a session
- **Window** – Number of octets that the sender will accept
- **Checksum** – Calculated checksum of the header and data fields
- **Urgent pointer** – Indicates the end of the urgent data
- **Option** – One option currently defined, maximum TCP segment size
- **Data** – Upper-layer protocol data

Code Bits or Flags (6 bits).

- URG: Urgent pointer field significant.
- ACK: Acknowledgment field significant.
- PSH: Push function.
- RST: Reset the connection.
- SYN: Synchronize the sequence numbers.
- FIN: No more data from sender.

UDP (User Datagram Protocol): UDP is the connectionless transport protocol in the TCP/IP protocol stack. UDP is a simple protocol that exchanges datagrams without guaranteed delivery. It relies on higher-layer protocols to handle errors and retransmit data.



Bit 0	Bit 15 Bit 16 Bit 31
Source Port (16)	Destination Port (16)
Length (16)	Checksum (16)
Data (if any)	

Fig: UDP Datagram

UDP does not use windows or ACKs. Reliability is provided by application layer protocols. UDP is designed for applications that do not need to put sequences of segments together.

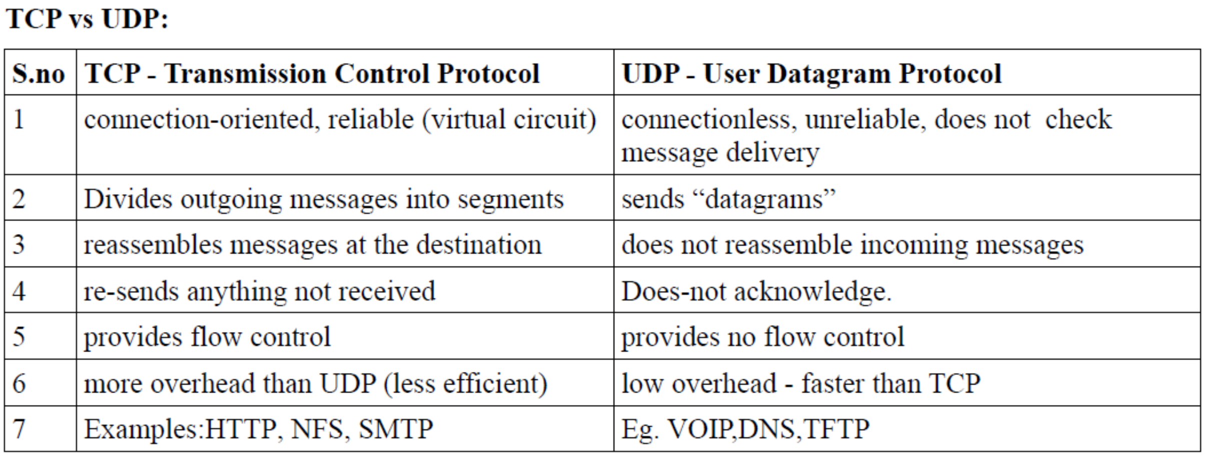
The following protocols use UDP:

- TFTP
- SNMP
- DHCP
- DNS

The following are the definitions of the fields in the UDP segment:

- **Source port** – Number of the port that sends data
- **Destination port** – Number of the port that receives data
- **Length** – Number of bytes in header and data

- **Checksum** – Calculated checksum of the header and data fields
- **Data** – Upper-layer protocol data



TCP vs UDP:

S.no	TCP - Transmission Control Protocol	UDP - User Datagram Protocol
1	connection-oriented, reliable (virtual circuit)	connectionless, unreliable, does not check message delivery
2	Divides outgoing messages into segments	sends "datagrams"
3	reassembles messages at the destination	does not reassemble incoming messages
4	re-sends anything not received	Does-not acknowledge.
5	provides flow control	provides no flow control
6	more overhead than UDP (less efficient)	low overhead - faster than TCP
7	Examples:HTTP, NFS, SMTP	Eg. VOIP,DNS,TFTP

Internet Layer:

The purpose of the Internet layer is to select the best path through the network for packets to travel. The main protocol that functions at this layer is IP. Best path determination and packet switching occur at this layer.

The following protocols operate at the TCP/IP Internet layer:

- IP provides connectionless, best-effort delivery routing of packets. IP is not concerned with the content of the packets but looks for a path to the destination.
- Internet Control Message Protocol (ICMP) provides control and messaging capabilities.
- Address Resolution Protocol (ARP) determines the data link layer address, or MAC address, for known IP addresses.
- Reverse Address Resolution Protocol (RARP) determines the IP address for a known MAC address.

IP performs the following operations:

- Defines a packet and an addressing scheme
- Transfers data between the Internet layer and network access layer
- Routes packets to remote hosts

Network Access Layer:

The network access layer allows an IP packet to make a physical link to the network media. It includes the LAN and WAN technology details and all the details contained in the OSI physical and data link layers.

Drivers for software applications, modem cards, and other devices operate at the network access layer. The network access layer defines the procedures used to interface with the network hardware and access the transmission medium. Modem protocol standards such as

Serial Line Internet Protocol (SLIP) and Point-to-Point Protocol (PPP) provide network access through a modem connection. Many protocols are required to determine the hardware, software, and transmission-medium specifications at this layer. This can lead to confusion for users. Most of the recognizable protocols operate at the transport and Internet layers of the TCP/IP model. Network access layer protocols also map IP addresses to physical hardware addresses and encapsulate IP packets into frames. The network access layer defines the physical media connection based on the hardware type and network interface.

OSI Model and TCP/IP Model:

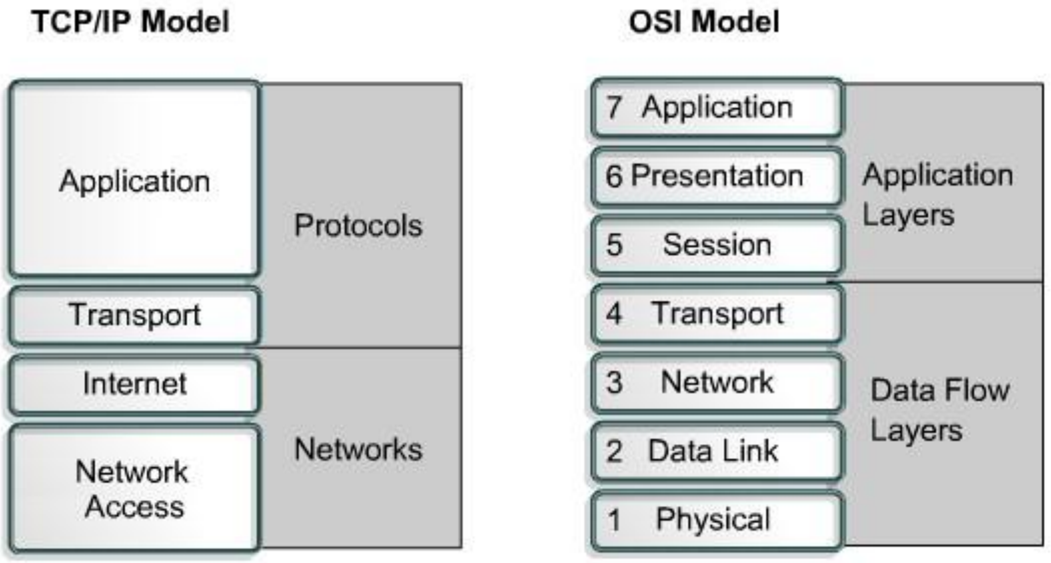
The OSI and TCP/IP models have many similarities:

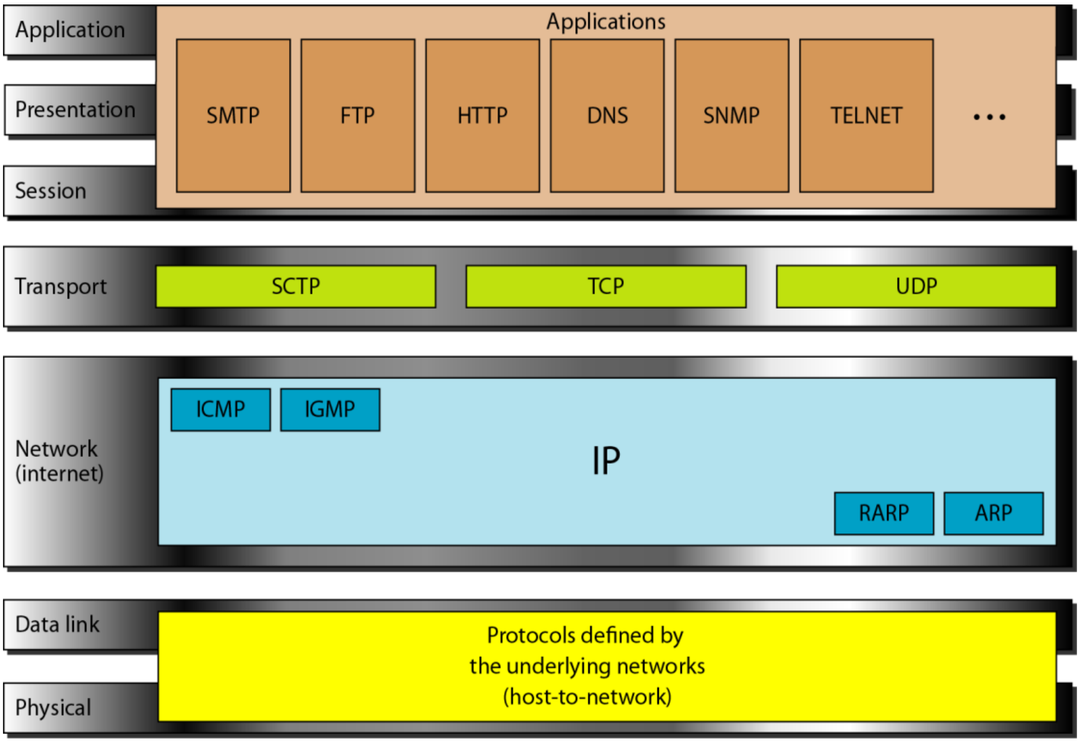
- Both have layers.
- Both have application layers, though they include different services.
- Both have comparable transport and network layers.
- Both use packet-switched instead of circuit-switched technology.
- Networking professionals need to know both models.

Here are some differences of the OSI and TCP/IP models:

- TCP/IP combines the OSI application, presentation, and session layers into its application layer.
- TCP/IP combines the OSI data link and physical layers into its network access layer.
- TCP/IP appears simpler because it has fewer layers.
- When the TCP/IP transport layer uses UDP it does not provide reliable delivery of packets. The transport layer in the OSI model always does.

The Internet was developed based on the standards of the TCP/IP protocols. The TCP/IP model gains credibility because of its protocols. The OSI model is not generally used to build networks. The OSI model is used as a guide to help students understand the communication process.





IP

IP provides communication between hosts on different kinds of networks (i.e., different data-link implementations such as Ethernet and Token Ring). It is a connectionless, unreliable packet delivery service. Connectionless means that there is no handshaking, each packet is independent of any other packet. It is unreliable because there is no guarantee that a packet gets delivered; higher level protocols must deal with that.

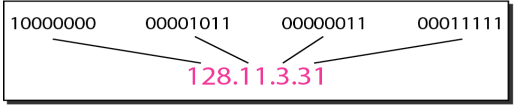
IP Addresses

IP defines an addressing scheme that is independent of the underlying physical address (e.g., 48-bit MAC address. The designers of the Internet Protocol defined an IP address as a 32-bit number and this system, known as Internet Protocol Version 4 (IPv4), is still in use today. However, due to the enormous growth of the Internet and the predicted depletion of available addresses, a new version of IP (IPv6), using 128 bits for the address, was developed in 1995. IPv6 was standardized as RFC 2460 in 1998, and its deployment has been ongoing since the mid-2000s. Each packet sent across the internet contains the IP address of the source of the packet and the IP address of its destination.

For routing efficiency, the IP address is considered in two parts: the prefix which identifies the physical network, and the suffix which identifies a computer on the network. A unique prefix is needed for each network in an internet. For the global Internet, network numbers are obtained from Internet Service Providers (ISPs). ISPs coordinate with a central organization called the Internet Assigned Number Authority (IANA).

IPv4

- Foundational protocol of the current internet.
- An IPv4 address is 32 bits long.
- The IPv4 addresses are unique and universal.
- The address space of IPv4 is 2^{128} = 4,294,967,296 (4.3 billion).
- Dotted-decimal notation and binary notation for an IPv4 address is look like,



IPv4 Address Classes

- An IPv4 address is a 32-bit quantity.
    - Expressed as a dotted-decimal notation W.X.Y.Z, where dots are used to separate each of the four octets of the address.
    - Consists of the two logical parts:
        - A network number
        - A host number
    - This partition defines the *IP address classes.*
- A computer on the internet is addressed using a two-tuple:
    - **The network number**
        - Assigned and managed by central authority.
    - **The host number**
        - Assigned and managed by local network administrator.
- When routing a packet to the destination network, only the network number is looked at.
- There are five defined IP address classes.

Class A	Un icast
Class B	Unicast
Class C	Unicast
Class D	Multicast
Class E	Reserved

- Identified by the first few bits in the IP address.
- There also exist some special-purpose IP addresses.
- The class-based addressing is also known as the *classful model.*
    - Different network classes represent different network-to-hosts ratio.
    - Lend themselves to different network configurations.

Class A address

0	Network	Host	Host	Host

- Network bits: 7
    - Number of Networks = 2^7-1 =127
- Host bits: 24
    - Number of hosts =2^{24}-2 =16,777,214
- Address range:

  - ➢ 0.0.0.0 to 127.255.255.255

Class B address

10	Network	Network	Host	Host

- Network bits: 14
  - ➢ Number of Networks = 2^{14}-1 =16,383
- Host bits: 16
  - ➢ Number of hosts =2^{16}-2 =65,534
- Address range:
  - ➢ 128.0.0.0 to 191.255.255.255

Class C address

110	Network	Network	Network	Host

- Network bits: 21
  - ➢ Number of Networks = 2^{21}-1 =2,097,151
- Host bits: 8
  - ➢ Number of hosts =2^{8}-2 =254
- Address range:
  - ➢ 192.0.0.0 to 223.255.255.255

Class D address

1110	Multicast Address

- Address range:
  - ➢ 224.0.0.0 to 239.255.255.255

Class E address

1111	Reserved Address

- Address range:
  - ➢ 240.0.0.0 to 255.255.255.255

Special-purpose IP Addresses

- Reserved for private use
  - ➢ 10.x.x.x (Class A)
  - ➢ 172.16.x.x – 172.31.x.x (Class B)
  - ➢ 192.168.x.x (Class B)
- Loopback/ local address
  - ➢ 127.0.0.0-127.222.255.255
- Default network

  - 0.0.0.0

- Limited broadcast
  - 255.255.255.255

IPv4 Header Format

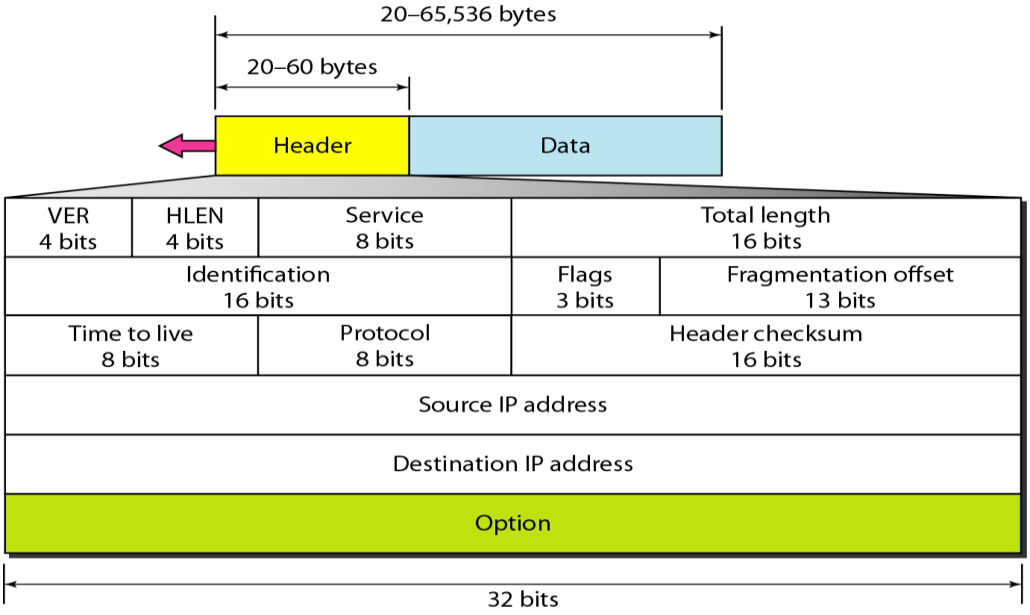


Fig.: IPv4 header format

Version (4 bits): Indicates the version number, to allow evolution of the protocol.

Internet Header Length (IHL 4 bits): Length of header in 32 bit words. The minimum value is five for a minimum header length of 20 octets.

Type-of-Service (8 bits): The Type-of-Service field contains an 8-bit binary value that is used to determine the priority of each packet. This value enables a Quality-of-Service (QoS) mechanism to be applied to high priority packets, such as those carrying telephony voice data. The router processing the packets can be configured to decide which packet it is to forward first based on the Type-of-Service value.

Total length (16 bits): total datagram length, in octets.

Identifier (16 bits): A sequence number that, together with the source address, destination address, and user protocol, is intended to uniquely identify a datagram. Thus, the identifier should be unique for the datagram's source address, destination address, and user protocol for the time during which the datagram will remain in the internet.

Fragment Offset: A router may have to fragment a packet when forwarding it from one medium to another medium that has a smaller Maximum Transmission Unit (MTU). MTU is the size (in bytes) of the largest protocol data unit that the layer can pass onwards. When fragmentation occurs, the IPv4 packet uses the Fragment Offset field and the MF flag in the IP header to reconstruct the packet when it arrives at the destination host. The fragment offset field identifies the order in which to place the packet fragment in the reconstruction.

Flags (3 bits): Only two of the bits are currently defined: MF (More Fragments) and DF (Don't Fragment).

More Fragments flag (MF): The More Fragments (MF) flag is a single bit in the Flag field used with the Fragment Offset for the fragmentation and reconstruction of packets. The More Fragments flag bit is set; it means that it is not the last fragment of a packet. When a receiving host sees a packet arrive with the MF = 1, it examines the Fragment Offset to see where this fragment is to be placed in the reconstructed packet. When a receiving host receives a frame with the MF = 0 and a non-zero value in the Fragment offset, it places that fragment as the last part of the reconstructed packet. An unfragmented packet has all zero fragmentation information (MF = 0, fragment offset=0).

Don't Fragment flag (DF): The Don't Fragment (DF) flag is a single bit in the Flag field that indicates that fragmentation of the packet is not allowed. If the Don't Fragment flag bit is set, then fragmentation of this packet is NOT permitted. If a router needs to fragment a packet to allow it to be passed downward to the Data Link layer but the DF bit is set to 1, then the router will discard this packet.

IP Destination Address (32 bits): The IP Destination Address field contains a 32-bit binary value that represents the packet destination Network layer host address.

IP Source Address (32 bits): The IP Source Address field contains a 32-bit binary value that represents the packet source Network layer host address.

Time-to-Live (8 bits): The Time-to-Live (TTL) is an 8-bit binary value that indicates the remaining "life" of the packet. The TTL value is decreased by at least one each time the packet is processed by a router (that is, each hop). When the value becomes zero, the router discards or drops the packet and it is removed from the network data flow. This mechanism prevents packets that cannot reach their destination from being forwarded indefinitely between routers in a routing loop. If routing loops were permitted to continue, the network would become congested with data packets that will never reach their destination. Decrementing the TTL value at each hop ensures that it eventually becomes zero and that the packet with the expired TTL field will be dropped.

Protocol (8 bits): This 8-bit binary value indicates the data payload type that the packet is carrying. The Protocol field enables the Network layer to pass the data to the appropriate upper-layer protocol.

Example values are:
01 ICMP
06 TCP
17 UDP

Header checksum (16 bits): An error-detecting code applied to the header only. Because some header fields may change during transit (e.g., time to live, segmentation-related fields), this is reverified and recomputed at each router. The checksum field is the 16-bit one's complement addition of all 16-bit words in the header. For purposes of computation, the checksum field is itself initialized to a value of zero.

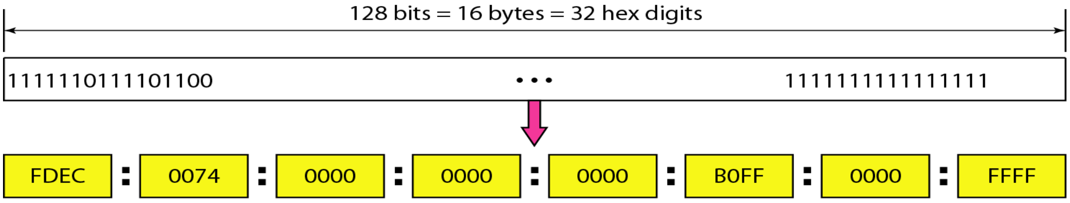
Options (variable): Encodes the options requested by the sending user.

Padding (variable): (Sometimes embedded with option field). Used to ensure that the datagram header is a multiple of 32 bits.

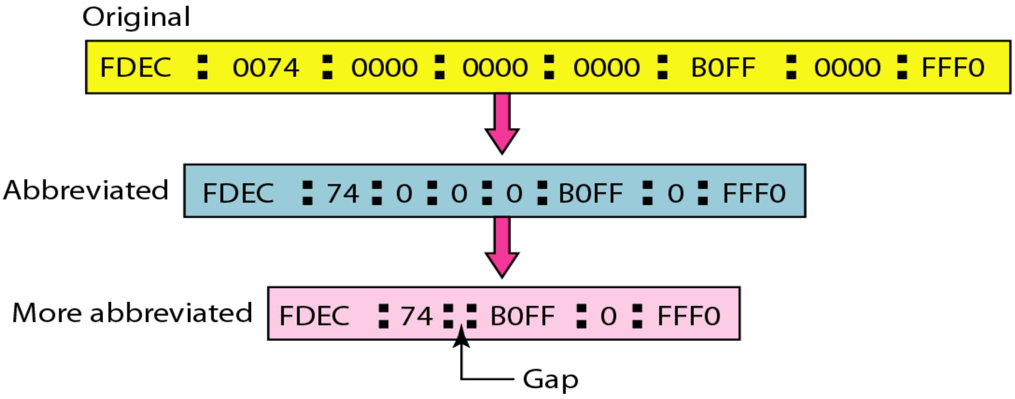
Data (variable): The data field must be an integer multiple of 8 bits. The maximum length of the datagram (data field plus header) is 65,535 octets.

IPv6

- An IPv6 address is 128 bits long.
- The address space of IPv4 is $2^{128} = 3.4\times10^{38}$.
- So large that everyone on earth will have enough addresses to have their own internets with as many addresses as the current Internet has.
- So large that there would be 10^{24} internet addresses per each square meter on earth
- If addresses are assigned at the rate of 1,000,000 every microsecond (1/1,000,000th of a second), it would take more than 10^{20} years to assign all possible addresses
- IPv6 address in binary and hexadecimal colon notation is looks like,



- Abbreviated IPv6 addresses


Original
FDEC : 0074 : 0000 : 0000 : 0000 : B0FF : 0000 : FFF0
Abbreviated
FDEC : 74 : 0 : 0 : 0 : B0FF : 0 : FFF0
More abbreviated
FDEC : 74 :: B0FF : 0 : FFF0
Gap

- **IPv6 address Components**
  - High order 64 bits
    - Network and subnet id
  - low order 64 bits
    - host id
    - Can be derived from MAC address

- **Basic IPv6 Address Types**
  - **Unicast**
    - Identify a specific host across entire Internet.
    - Globally routable.
    - Highest order bits are 001.
    - Destination address specifies a single computer. Route datagram along shortest path.
  - **Anycast**
    - Deliver to Any one of a group of hosts.
    - i.e., Destination is a set of computers, possibly at different locations, that all share a single address. Route datagram along shortest path and deliver to exactly one member of the group (i.e. closest member)
  - **Multicast**
    - Deliver to an entire group of hosts.
    - Start with octets 0xFF.
    - i.e., Destination is a set of computers, possibly at different locations. One copy of the datagram will be delivered to each member of the group using hardware multicast or broadcast if viable.

- **Advantages of IPv6**
  - **Larger address space**
  - **Better header format**

    IPv6 uses new header format in which options are separated from the base header and inserted, when needed, between the base header and upper-layer data. This simplifies and speeds up the routing process because most of the options do not need to be checked by router.
  - **New options**

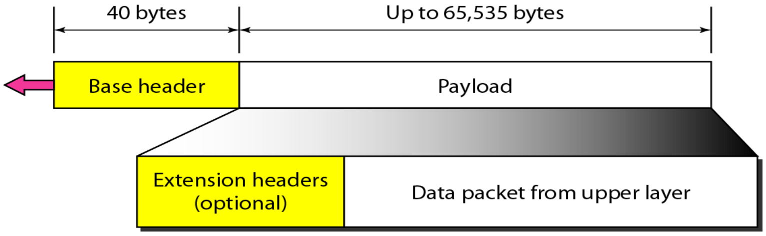
    IPV6 has new options to allow for additional functionalities.
  - **Allowance for extension**

    IPv6 is designed to allow the extension of the protocol if required by new technologies or applications.
  - **Support for resource allocation**

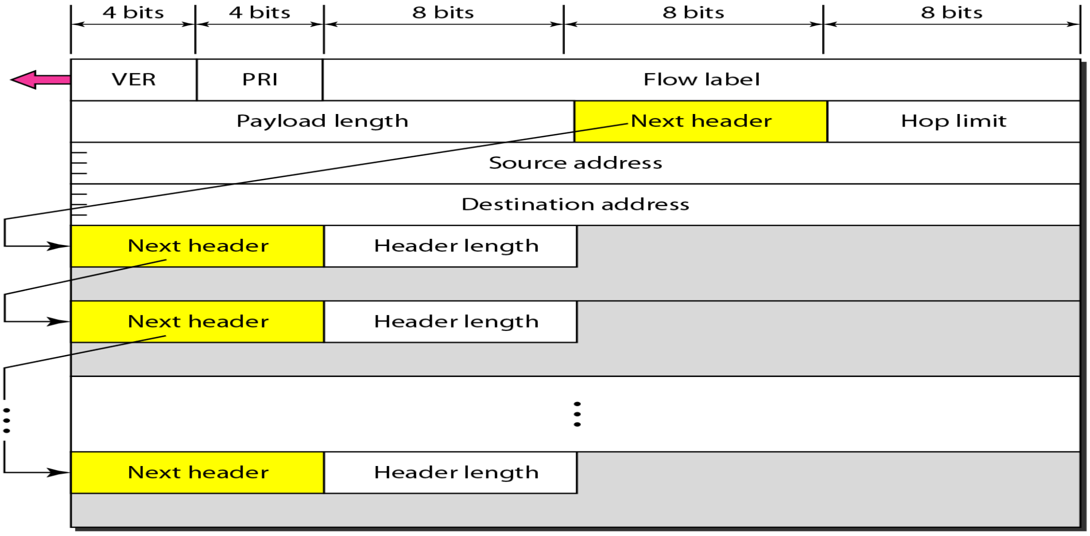
    In IPv6, the type-of-service field has been removed, but a mechanism (called flow label) has been added to enable the source to request special handling of the packet. This mechanism can be used to support traffic such as real-time audio and video.
  - **supports for more security**

The encryption and authentication options in IPv6 provide confidentiality and integrity of the packet.

➢ General Form of IPv6 Datagram



➢ Format of an IPv6 base header

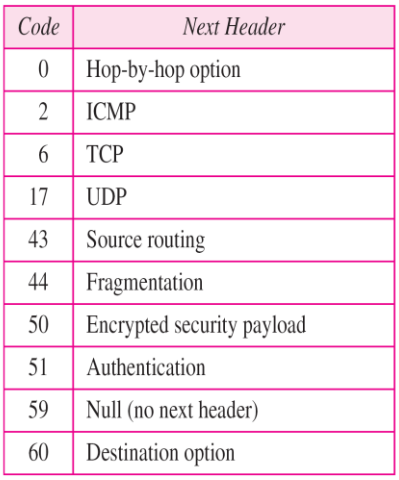


Fields defined in the IPv6 base header are:

- ***Version:*** This 4-bit field identifies the IP version number. For IPv6, it is 6.
- ***Priority*****:** This 4-bit priority field defines the priority of the packet with respect to traffic congestion.
- ***Flow label*****:** This 24-bit field that is designed to provide special handling for the particular flow of the data.
- ***Payload length:*** This 16-bit field defines the length of the IP datagram excluding the base header.
- ***Next header***: This 8-bit selector identifies the type of header immediately following the IPv6 header. The next header is either one of the optional extension headers used by IP or the header of an encapsulated packet such as UDP or TCP. Each extension header also contains this field. In IP version 4 this field is called *protocol*.
- ***Hop limit***: This 8-bit serves the same purpose as the TTL field in IPV4. Decremented by 1 by each node that forwards the packet. The packet is discarded if Hop Limit is decremented to zero.

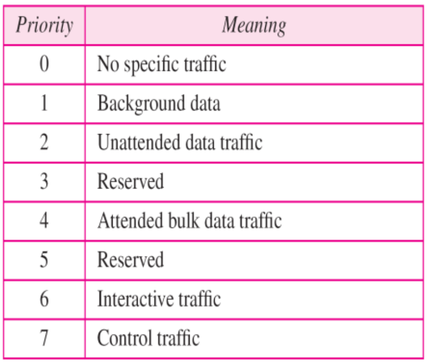
- ***Source Address***: 128-bit address of the originator of the packet.
- ***Destination Address:*** 128-bit address of the intended recipient of the packet.

Next header codes for IPV6



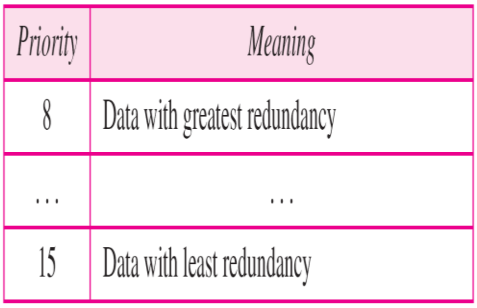
Code	*Next Header*
0	Hop-by-hop option
2	ICMP
6	TCP
17	UDP
43	Source routing
44	Fragmentation
50	Encrypted security payload
51	Authentication
59	Null (no next header)
60	Destination option

Priority (Priorities for congestion-controlled traffic)



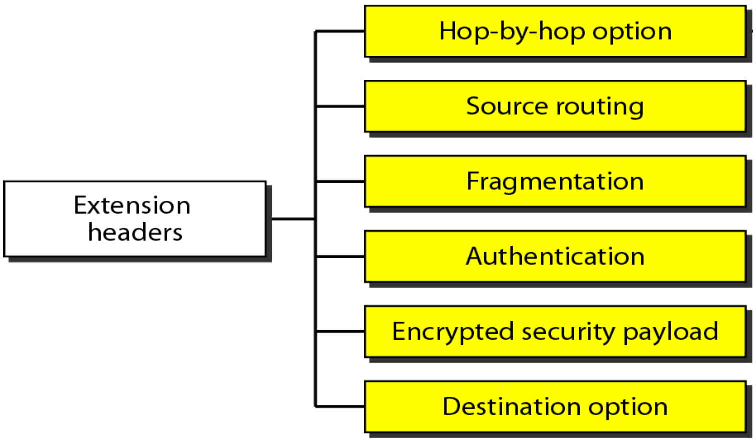
Priority	*Meaning*
0	No specific traffic
1	Background data
2	Unattended data traffic
3	Reserved
4	Attended bulk data traffic
5	Reserved
6	Interactive traffic
7	Control traffic

Priority (Priorities for noncongestion-controlled traffic)



Priority	*Meaning*
8	Data with greatest redundancy
. . .	. . .
15	Data with least redundancy

Extension header types



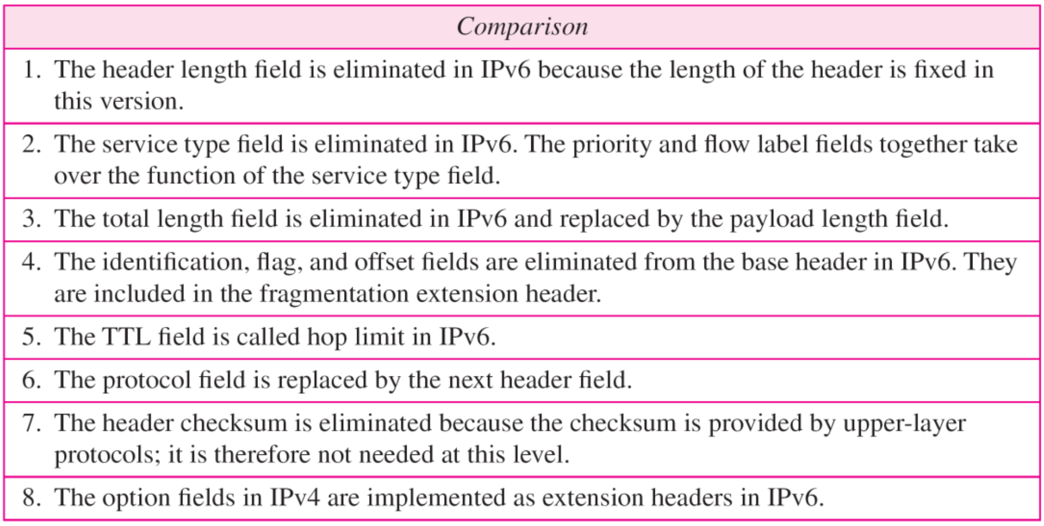
- **Hop-by-Hop Option**
  - ➢ Used when the source needs to pass information to all routers visited by the datagram.
- **Source Routing**
  - ➢ Combines the concept of strict source route and the loose source route options of IPv4.
- **Fragmentation**
- **Authentication**
  - ➢ Validates the message sender and ensures the integrity of the data.
- **Encrypted Security Payload (ESP)**
  - ➢ Provides confidentiality and guards against eavesdropping.
- **Destination Option**
  - ➢ Used when source needs to pass information to the destination only.
  - ➢ Intermediate routers are not permitted access to this information.

Transition from IPv4 to IPv6

Three alternate transition strategies:

1. **Dual stack:** Both IPv4 and IPv6 protocol stacks supported in the gateway.
2. **Tunneling:** an IPv6 datagram flows through an intermediate IPv44 network by encapsulating the whole IPv6 packet as payload.
3. **Header translation:** An IPV4 address is translated into a IPv6 address, and vice versa.

Comparison between IPv4 and IPv6 packet headers



Comparison
1. The header length field is eliminated in IPv6 because the length of the header is fixed in this version.
2. The service type field is eliminated in IPv6. The priority and flow label fields together take over the function of the service type field.
3. The total length field is eliminated in IPv6 and replaced by the payload length field.
4. The identification, flag, and offset fields are eliminated from the base header in IPv6. They are included in the fragmentation extension header.
5. The TTL field is called hop limit in IPv6.
6. The protocol field is replaced by the next header field.
7. The header checksum is eliminated because the checksum is provided by upper-layer protocols; it is therefore not needed at this level.
8. The option fields in IPv4 are implemented as extension headers in IPv6.

The Internet Society

- Internet Architecture Board (IAB)
- Internet Engineering Task Force (IETF)
- Internet Engineering Steering Group (IESG)

Internet RFCs

Request for Comments (**RFC**) is a written proposal published by the Internet Engineering Task Force (IETF) describing methods, behaviors, research, or innovations applicable to the working of the Internet and Internet-connected systems.

Through the Internet Society, engineers and computer scientists may publish discourse in the form of an RFC, either for peer review or simply to convey new concepts, information, or (occasionally) engineering humor. The IETF adopts some of the proposals published as RFCs as Internet standards.

Request For Comments documents were invented by Steve Crocker in 1969 to help record unofficial notes on the development of the ARPANET. They have since become the official record for Internet specifications, protocols, procedures, and events.

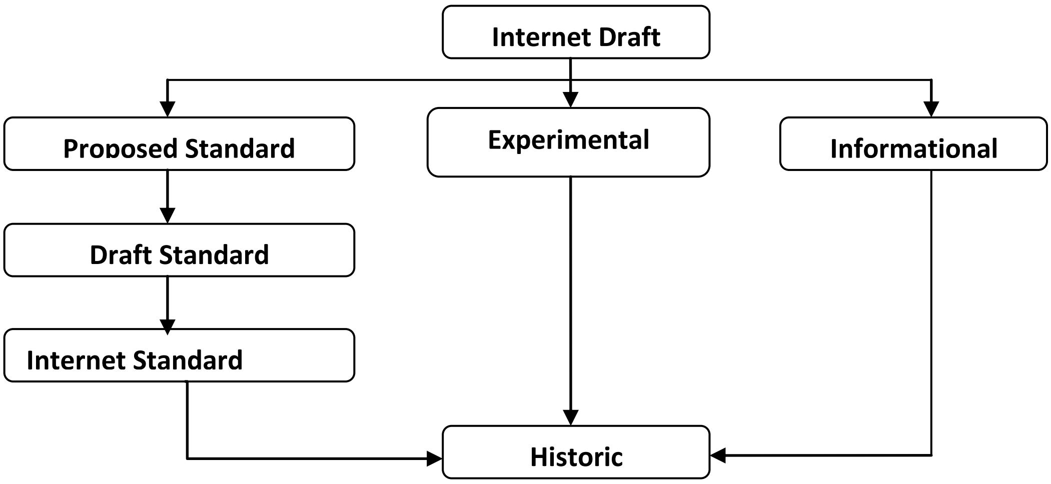
Process involved in RFC Publication

The actual development of new standards is carried out by working groups chartered by IETF.

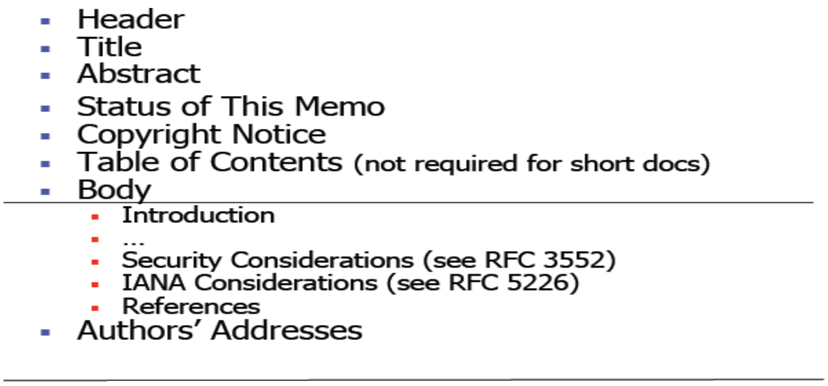
- The working group makes a draft version of the document.
- Place it in the “Internet Draft” online directory.
- Kept there for six month, and review and comments on the draft obtained.

- The IESG may approve the publication of the draft as an RFC during this period.
    - or else it is withdrawn from the directory.
- The working group may subsequently publish a revised version of the draft.

RFC Publication Process



Contents of Internet-Draft/RFC



- Header
- Title
- Abstract
- Status of This Memo
- Copyright Notice
- Table of Contents (not required for short docs)
- Body
    - Introduction
    - ...
    - Security Considerations (see RFC 3552)
    - IANA Considerations (see RFC 5226)
    - References
- Authors' Addresses

Important RFCs

RFC821: Simple Mail Transfer Protocol
RFC791: Internet Protocol
RFC793: Transmission Control Protocol
RFC2616: Hypertext Transfer Protocol 1.1
RFC2045: MIME
RFC1321: MD5 Message Digest Algorithm
RFC1866: Hypertext Markup Language 2.0
RFC2437: RSA Crypto Specifications 2.0
RFC2631: Diffie-Hellman Key Agreement

Where to find the RFCs

http://www.faqs.org/rfcs/ http://www.ietf.org/rfc.html/ http://www.rfc.net