Unit 2

Internet Protocol Overview

2.1 TCP/IP and the IP Layer overview

Network Protocol

In a computer network, entities in different hosts need to communicate. For two entities/ peers for successfully communicate, they must "speak the same language". What is communicated must conform to same manually accepted set of conventions is referred to as protocol, which may be defined as a set of rules governing the exchange of data between two entities

Layered Architecture

A Network is a conceptual framework that describes how data and network information are communicated from an application on one computer through network media to an application on other computers in terms of different layers. Network architecture is also known as Reference model.

To tackle with the design complexity most of the networks are organize as a set of layers or levels. The fundamental idea of layered architecture is to divide the design into small pieces. The layering provides modularity to the network design. The main duty of each layer is to provide offer services to higher layers, and provide abstraction. The main benefits of layered architecture are modularity and clear interfaces.

Layered architectures have several advantages. Some of them are,

- Modularity and clear interface
- Provide flexibility to modify network services
- Ensure independence of layers
- Management of network architecture is easy
- Each layer can be analyzed and tested independent of other layers
- Standardization

Both the **OSI** and the **TCP/IP** architectures are layered architectures, that is the functionality of the network is decomposed into layers, where a higher level layer uses the services provided by the layer immediately below it.

The OSI, or Open System Interconnection, model defines a networking framework to implement protocols in seven layers. Control is passed from one layer to the next, starting at the application layer in one station, and proceeding to the bottom layer, over the channel to the next station and back up the hierarchy.

Application Layer (Layer 7)
This layer supports application and end-user processes. It consists of protocols that focus on process-to-process communication across an IP network and provides a firm communication interface and end-user services. This layer provides application services for file transfers, e-mail, and other network software services. Telnet and FTP are applications that exist entirely in the application level. It may be convenient to think of the application layer as the high-level set-up services for the application program or an interactive user.

Presentation Layer (Layer 6)
The presentation layer is responsible for the delivery and formatting of information to the application layer for further processing or display. The presentation layer works to transform data into the form that the application layer can accept. This layer formats and encrypts data to be sent across a network, providing freedom from compatibility problems. It is sometimes called the syntax layer.

Session Layer (Layer 5)
The session layer provides the mechanism for opening, closing and managing a session between end-user application processes. The services offered by the session layer are generally implemented in application environments using remote procedure calls (RPCs). The session layer is also responsible for synchronizing information from different sources. For example, sessions are implemented in live television programs in which the audio and video streams emerging from two different sources are merged together. This avoids overlapping and silent broadcast time.

For an example, our web browser (an application layer object) opens a web page. That page contains text, graphics, Macromedia Flash objects and perhaps a Java applet. The graphics, the Flash object and the Java applet are all stored as separate files on the web server. To access them, a separate download must be started. Our web browser opens a separate session to the web server to download each of the individual files. The session layer keeps track of which packets and data belong to which file and keeps track of where they go (in this case, to ourweb browser).
In most modern Internet applications, the session, presentation and application layers are usually rolled together inside the application itself, thus, your web browser performs all functions of the session, presentation and application layers.

Transport Layer (Layer 4)

The Transport layer ensures the reliable arrival of messages and provides error checking mechanisms and data flow controls. The Transport layer provides services for both "connection-oriented" transmissions and for "connectionless" transmissions. For connection-mode transmissions, a transmission may be sent or arrive in the form of packets that need to be reconstructed into a complete message at the other end.

It provides logical communication between application processes running on different hosts within a layered architecture of protocols and other network components.The transport layer is also responsible for the management of error correction, providing quality and reliability to the end user. This layer enables the host to send and receive error corrected data, packets or messages over a network and is the network component that allows multiplexing.

Transport layers work transparently within the layers above to deliver and receive data without errors. The sending side breaks application messages into segments and passes them on to the network layer. The receiving side then reassembles segments into messages and passes them to the application layer

Network Layer (Layer 3)
This layer provides switching and routing technologies, creating logical paths, known as virtual circuits, for transmitting data from node to node. Routing and forwarding are functions of this layer, as well as addressing, internetworking, error handling,congestion control and packet sequencing.

Functions of the network layer include:

- Connection model
  IP is connectionless protocol, in that a datagram can travel from a sender to a recipient without the recipient having to send an acknowledgement. Connection-oriented protocols exist at other, higher layers of the OSI model.

- Host addressing
  Every host in the network must have a unique address that determines where it is. So, this layer function is to address every hosts uniquely in the subnet in the form of IP Address.

- Message forwarding
  Since many networks are partitioned into sub networks and connect to other networks for wide-area communications, networks use specialized hosts, called gateways or routers, to forward packets between networks. This is also of interest to mobile applications, where a user may move from one location to another, and it must be arranged that his messages follow him.

Data Link Layer (Layer 2)

The Data-Link layer is the protocol layer in a program that handles the moving of data in and out across a physical link in a network. The data link layer is concerned with local delivery of frames between devices on the same LAN. Data-link frames, as these protocol data units are called, do not cross the boundaries of a local network. Inter-network routing and global addressing are higher layer functions, allowing data-link protocols to focus on local delivery, addressing, and media arbitration.

The data link thus provides data transfer across the physical link. That transfer can be reliable or unreliable; many data-link protocols do not have acknowledgments of successful frame reception and acceptance, and some data-link protocols might not even have any form of checksum to check for transmission errors. In those cases, higher-level protocols must provide flow control, error checking, and acknowledgments and retransmission.

Physical Layer (Layer 1)

The physical layer consists of the basic networking hardware transmission technologies of a network. It is a fundamental layer underlying the logical data structures of the higher level functions in a network. The physical layer defines the means of transmitting raw bits rather than logical data packets over a physical link connecting network nodes. The bit stream may be grouped into code words or symbols and converted to a physical signal that is transmitted over a hardware transmission medium. The physical layer provides an electrical, mechanical, and procedural interface to the transmission medium. Ethernet cabling, Token Ring network technology and SCSI all function at the Physical layer of the OSI model. Hubs and other repeaters are standard network devices that function at the Physical layer. Cables and connectors also are a part of the Physical layer.

Internet Protocol layer

The internet layer or IP layer is a group of internetworking methods, protocols, and specifications in the Internet protocol suite that are used to transport datagrams (packets) from the originating host across network boundaries, if necessary, to the destination host specified by a network address (IP address) which is defined for this purpose by the Internet Protocol (IP). The internet layer derives its name from its function of forming an internet , or facilitating internetworking, which is the concept of connecting multiple networks with each other through gateways.

The Internet Protocol (IP) is a network-layer (Layer 3) protocol that contains addressing information and some control information that enables packets to be routed. IP is documented in RFC 791 and is the primary network-layer protocol in the Internet protocol suite. Along with the Transmission Control Protocol (TCP), IP represents the heart of the Internet protocols.

The internet layer has three (3) basic functions:

1) For outgoing packets, select the next-hop host (gateway) and transmit the packet to this host by passing it to the appropriate link layer implementation;

2) For incoming packets, capture packets and pass the packet payload up to the appropriate transport-layer protocol, if appropriate.

3) In addition it provides error detection and diagnostic capability.

2.2 IPv4 and IPV6

Internet Protocol Version 4 (IPv4) is the fourth revision of the IP and a widely used protocol in data communication over different kinds of networks. IPv4 is a connectionless protocol used in packet-switched layer networks, such as Ethernet. It provides the logical connection between network devices by providing identification for each device. There are many ways to configure IPv4 with all kinds of devices - including manual and automatic configurations - depending on the network type.

IPv4 is based on the best-effort model. This model guarantees neither delivery nor avoidance of duplicate delivery; these aspects are handled by the upper layer transport. IPv4 uses 32-bit addresses for Ethernet communication in five classes, named A, B, C, D and E. Classes A, B and C have a different bit length for addressing the network host. Class D addresses are reserved for multicasting, while class E addresses are reserved for future use.

Internet Protocol version 6 (IPv6) is the latest revision of the Internet Protocol (IP), the communications protocol that provides an identification and location system for computers on networks and routes traffic across the Internet. IPv6 was developed by the Internet Engineering Task Force (IETF) to deal with the long-anticipated problem of IPv4 address exhaustion.

IPv6 uses a 128-bit address, allowing 2^128, or approximately 3.4×1038 addresses, or more than 7.9×1028 times as many as IPv4, which uses 32-bit addresses. IPv4 allows only approximately 4.3 billion addresses. The two protocols are not designed to be interoperable, complicating the transition to IPv6.

IPv6 addresses are represented as eight groups of four hexadecimal digits separated by colons, for example 2001:0db8:85a3:0042:1000:8a2e:0370:7334, but methods of abbreviation of this full notation exist.

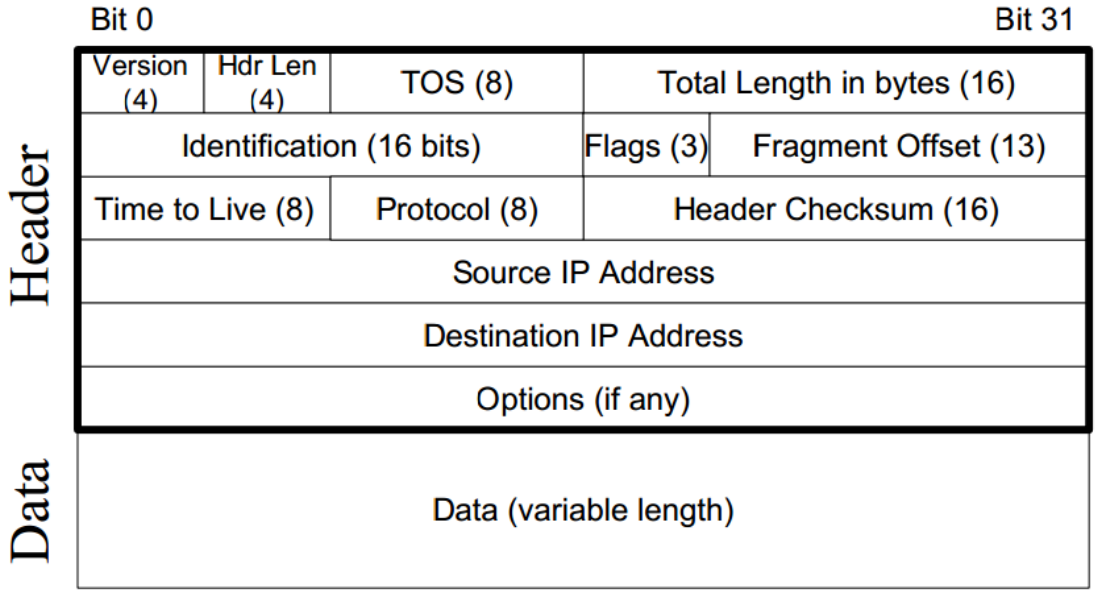
IPv6 is an Internet Protocol (IP) for packet-switched internetworking that specifies the format of packets (also called datagrams) and the addressing scheme across multiple IP networks. In comparing the two protocols IPv6 expands upon the addressing and routing capabilities of IPv4 in a number of ways including:

- In IPv6 the IP address size is increased from 32 bits to 128 bits
- IPv6 supports a greater number of addressable nodes
- IPv6 provides more levels of addressing hierarchy
- IPv6 offers simpler auto-configuration of addresses
- Ipv6 also supports simplified header format

IPV4 Vs IPV6

IPv4	IPv6
Addresses are 32 bits (4 bytes) in length.	Addresses are 128 bits (16 bytes) in length
IPSec is optional and should be supported externally	IPSec support is not optional
Header does not identify packet flow for QoS handling by routers	Header contains Flow Label field, which Identifies packet flow for QoS handling by router.
Both routers and the sending host fragment packets.	Routers do not support packet fragmentation. Sending host fragments packets
Header includes a checksum.	Header does not include a checksum.
Header includes options.	Optional data is supported as extension headers.
ARP uses broadcast ARP request to resolve IP to MAC/Hardware address.	Multicast Neighbor Solicitation messages resolve IP addresses to MAC addresses.
Broadcast addresses are used to send traffic to all nodes on a subnet.	IPv6 uses a link-local scope all-nodes multicast address.
Configured either manually or through DHCP.	Does not require manual configuration or DHCP.
Must support a 576-byte packet size (possibly fragmented).	Must support a 1280-byte packet size (without fragmentation).

IPV4 Frame Format



Field Name	Size (bytes)	Description
Version	1/2 (4 bits)	Version: Identifies the version of IP used to generate the datagram. For IPv4, this is of course the number 4. The purpose of this field is to ensure compatibility between devices that may be running different versions of IP. In general, a device running an older version of IP will reject datagrams created by newer implementations, under the assumption that the older version may not be able to interpret the newer datagram correctly.
IHL	1/2 (4 bits)	Internet Header Length (IHL): Specifies the length of the IP header, in 32-bit words. This includes the length of any options fields and padding. The normal value of this field when no options are used is 5 (5 32-bit words = 5*4 = 20 bytes). Contrast to the longer Total Lengthfield below.
TOS	1	Type Of Service (TOS): A field designed to carry information to provide quality of service features, such as prioritized delivery, for IP datagrams. It was never widely used as originally defined, and its

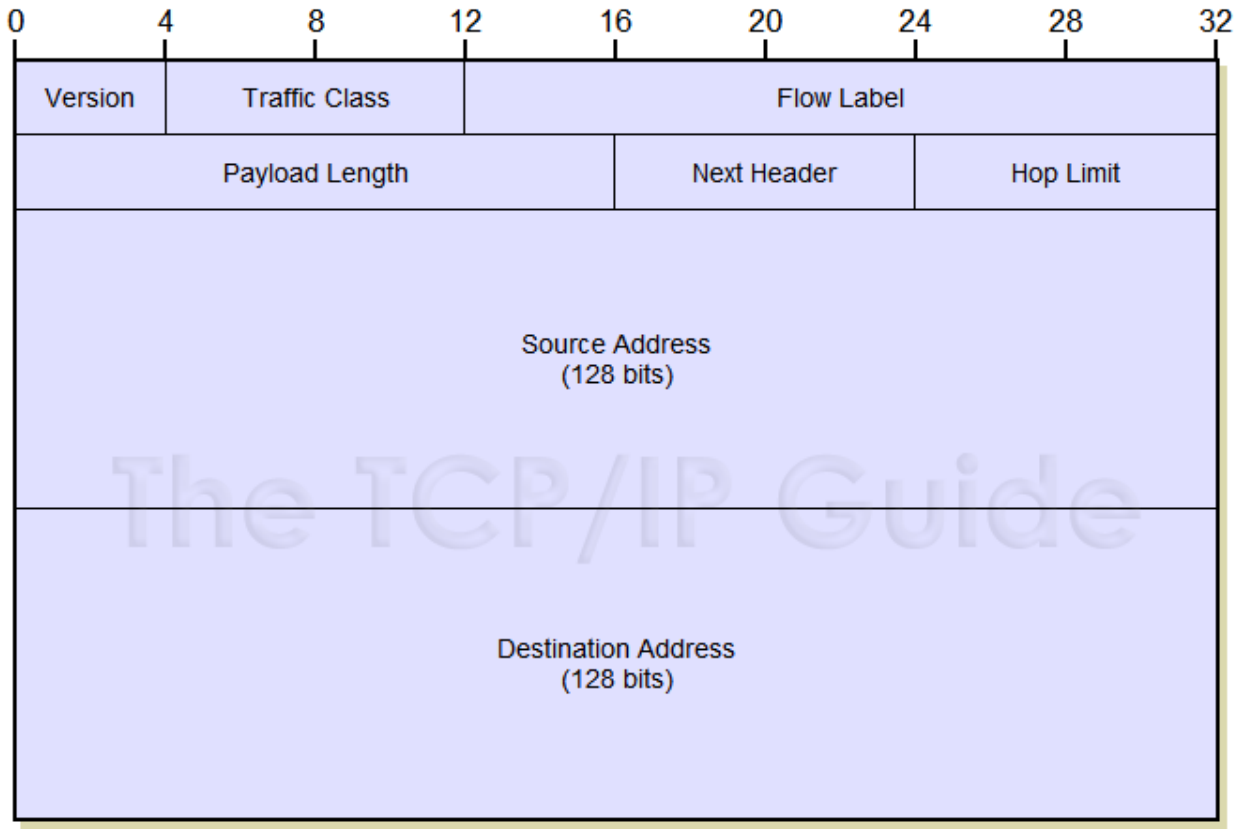
<table>
<tr><td></td><td></td><td>meaning has been subsequently redefined for use by a technique calledDifferentiated Services (DS). See below for more information.</td></tr>
<tr><td>TL</td><td>2</td><td>Total Length (TL): Specifies the total length of the IP datagram, in bytes. Since this field is 16 bits wide, the maximum length of an IP datagram is 65,535 bytes, though most are much smaller.</td></tr>
<tr><td>Identification</td><td>2</td><td>Identification: This field contains a 16-bit value that is common to each of the fragments belonging to a particular message; for datagrams originally sent unfragmented it is still filled in, so it can be used if the datagram must be fragmented by a router during delivery. This field is used by the recipient to reassemble messages without accidentally mixing fragments from different messages. This is needed because fragments may arrive from multiple messages mixed together, since IP datagrams can be received out of order from any device. See the discussion of IP message fragmentation.</td></tr>
<tr><td>Flags</td><td>3/8
(3 bits)</td><td>Flags: Three control flags, two of which are used to manage fragmentation (as described in the topic on fragmentation), and one that is reserved:
<table>
<tr><th>Subfield Name</th><th>Size (bytes)</th><th>Description</th></tr>
<tr><td>Reserved</td><td>1/8 (1 bit)</td><td>Reserved: Not used.</td></tr>
<tr><td>DF</td><td>1/8 (1 bit)</td><td>Don't Fragment: When set to 1, specifies that the datagram should not be fragmented. Since the fragmentation process is generally "invisible" to higher layers, most protocols don't care about this and don't set this flag. It is, however, used for testing the maximum transmission unit (MTU) of a link.</td></tr>
<tr><td>MF</td><td>1/8 (1 bit)</td><td>More Fragments: When set to 0, indicates the last fragment in a message; when set to 1, indicates that more fragments are yet to come in the fragmented message. If no fragmentation is used for a message, then of course there is only one "fragment" (the whole message), and this flag is 0. If fragmentation is used, all fragments but the last set this flag to 1 so the recipient knows when all fragments have been sent.</td></tr>
</table></td></tr>
<tr><td>Fragment Offset</td><td>1 5/8
(13 bits)</td><td>Fragment Offset: When fragmentation of a message occurs, this field specifies the offset, or position, in the overall message where the data in this fragment goes. It is specified in units of 8 bytes (64 bits). The first fragment has an offset of 0. Again, see the discussion of fragmentation for a description of how the field is used.</td></tr>
<tr><td>TTL</td><td>1</td><td>Time To Live (TTL): Short version: Specifies how long the datagram is allowed to "live" on the network, in terms of router hops. Each router</td></tr>
</table>

<table>
<tr><td></td><td></td><td>decrements the value of the TTL field (reduces it by one) prior to transmitting it. If the TTL field drops to zero, the datagram is assumed to have taken too long a route and is discarded.

See below for the longer explanation of TTL.</td></tr>
<tr><td>Protocol</td><td>1</td><td>Protocol: Identifies the higher-layer protocol (generally either a transport layer protocol or encapsulated network layer protocol) carried in the datagram. The values of this field were originally defined by the IETF "Assigned Numbers" standard, RFC 1700, and are now maintained by the Internet Assigned Numbers Authority (IANA):
<table>
<tr><th>Value (Hexadecimal)</th><th>Value (Decimal)</th><th>Protocol</th></tr>
<tr><td>00</td><td>0</td><td>Reserved</td></tr>
<tr><td>01</td><td>1</td><td>ICMP</td></tr>
<tr><td>02</td><td>2</td><td>IGMP</td></tr>
<tr><td>03</td><td>3</td><td>GGP</td></tr>
<tr><td>04</td><td>4</td><td>IP-in-IP Encapsulation</td></tr>
<tr><td>06</td><td>6</td><td>TCP</td></tr>
<tr><td>08</td><td>8</td><td>EGP</td></tr>
<tr><td>11</td><td>17</td><td>UDP</td></tr>
<tr><td>32</td><td>50</td><td>Encapsulating Security Payload (ESP) Extension Header</td></tr>
<tr><td>33</td><td>51</td><td>Authentication Header (AH) Extension Header</td></tr>
</table>
Note that the last two entries are used when IPSec inserts additional headers into the datagram: the AH or ESP headers.</td></tr>
<tr><td>Header Checksum</td><td>2</td><td>Header Checksum: A checksum computed over the header to provide basic protection against corruption in transmission. This is not the more complex CRC code typically used by data link layer technologies such as Ethernet; it's just a 16-bit checksum. It is calculated by dividing the header bytes into words (a word is two bytes) and then adding them together. The data is not checksummed, only the header. At each hop the device receiving the datagram does the same checksum calculation and on a mismatch, discards the datagram as damaged.</td></tr>
<tr><td>Source Address</td><td>4</td><td>Source Address: The 32-bit IP address of the originator of the datagram. Note that even though intermediate devices such as routers may handle the datagram, they do not normally put their address into this field—it is always the device that originally sent the datagram.</td></tr>
</table>

Destination Address	4	Destination Address: The 32-bit IP address of the intended recipient of the datagram. Again, even though devices such as routers may be the intermediate targets of the datagram, this field is always for the ultimate destination.
Options	Variable	Options: One or more of several types of options may be included after the standard headers in certain IP datagrams. I discuss them in the topic that follows this one.
Padding	Variable	Padding: If one or more options are included, and the number of bits used for them is not a multiple of 32, enough zero bits are added to "pad out" the header to a multiple of 32 bits (4 bytes).
Data	Variable	Data: The data to be transmitted in the datagram, either an entire higher-layer message or a fragment of one.

IPV6 Frame Format



Source address (128 bits) The 128-bit source address field contains the IPv6 address of the originating node of the packet. It is the address of the originator of the IPv6 packet.

Destination address (128 bits) The 128-bit contains the destination address of the recipient node of the IPv6 packet. It is the address of the intended recipient of the IPv6 packet.

Version/IP version (4-bits) The 4-bit version field contains the number 6. It indicates the version of the IPv6 protocol. This field is the same size as the IPv4 version field that contains the number 4. However, this field has a limited use because IPv4 and IPv6 packets are not distinguished based on the value in the version field but by the protocol type present in the layer 2 envelope.

Packet priority/Traffic class (8 bits) The 8-bit Priority field in the IPv6 header can assume different values to enable the source node to differentiate between the packets generated by it by associating different delivery priorities to them. This field is subsequently used by the originating node and the routers to identify the data packets that belong to the same traffic class and distinguish between packets with different priorities.

Flow Label/QoS management (20 bits) The 20-bit flow label field in the IPv6 header can be used by a source to label a set of packets belonging to the same flow. This large field was created to provide additional support for real-time datagram delivery and quality of service features. A unique flow label is used to identify all the datagrams in a particular flow, so that routers between the source and destination all handle them the same way, to help ensure uniformity in how the datagrams in the flow are delivered. For example, if a video stream is being sent across an IP internetwork, the datagrams containing the stream could be identified with a flow label to ensure that they are delivered with minimal latency.

Payload length in bytes(16 bits) The 16-bit payload length field contains the length of the data field in octets/bits following the IPv6 packet header. The 16-bit Payload length field puts an upper limit on the maximum packet payload to 64 kilobytes. In case a higher packet payload is required, a Jumbo payload extension header is provided in the IPv6 protocol. A Jumbo payload (Jumbogram) is indicated by the value zero in the Payload Length field. Jumbograms are frequently used in supercomputer communication using the IPv6 protocol to transmit heavy data payload.

Next Header (8 bits) The 8-bit Next Header field identifies the type of header immediately following the IPv6 header and located at the beginning of the data field (payload) of the IPv6 packet. This field usually specifies the transport layer protocol used by a packet's payload. The two most common kinds of Next Headers are TCP (6) and UDP (17), but many other headers are also possibl. In case of IPv6 protocol, the Next Header field is similar to the IPv4 Protocol field.

Time To Live (TTL)/Hop Limit (8 bits) The 8-bit Hop Limit field is decremented by one, by each

node (typically a router) that forwards a packet. If the Hop Limit field is decremented to zero, the packet is discarded. The main function of this field is to identify and to discard packets that are stuck in an indefinite loop due to any routing information errors. The 8-bit field also puts an upper limit on the maximum number of links between two IPv6 nodes. In this way, an IPv6 data packet is allowed a maximum of 255 hops before it is eventually discarded. An IPv6 data packet can pass through a maximum of 254 routers before being discarded.