2.1. TCP/IP and the IP Layer overview
2.2. IPv4 and IPv6 Address Types and Formats
2.3. IPv4 and IPv6 Header Structure
2.4. Internet RFCs

The **Internet Protocol (IP)** is the method or protocol or rule by which data is sent from one computer to another on the Internet. Each computer (known as a host) on the Internet has at least one IP address that uniquely identifies it from all other computers on the Internet.

When you send or receive data (for example, an e-mail note or a Web page), the message gets divided into little chunks called packets. Each of these packets contains both the sender's Internet address and the receiver's address. Any packet is sent first to a gateway computer that understands a small part of the Internet. The gateway computer reads the destination address and forwards the packet to an adjacent gateway that in turn reads the destination address and so forth across the Internet until one gateway recognizes the packet as belonging to a computer within its immediate neighborhood or domain. That gateway then forwards the packet directly to the computer whose address is specified.

Because a message is divided into a number of packets, each packet can, if necessary, be sent by a different route across the Internet. Packets can arrive in a different order than the order they were sent in. *The Internet Protocol just delivers them.* It's up to another protocol, the Transmission Control Protocol (TCP) to put them back in the right order.

IP provides several services:

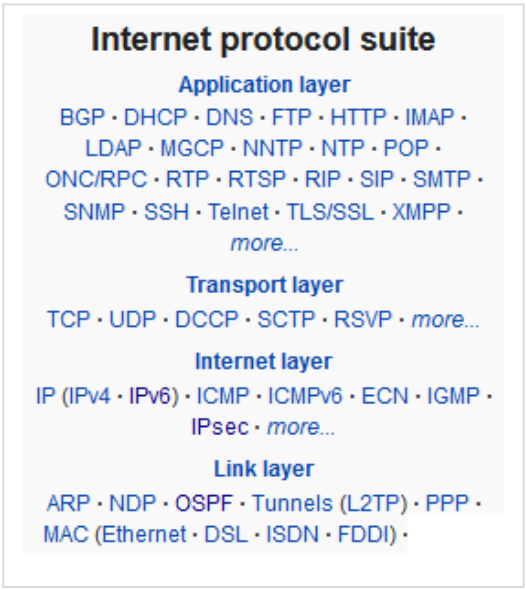
- **Addressing.** IP headers contain 32-bit addresses which *identify the sending and receiving hosts.* These addresses are used by intermediate routers to select a path through the network for the packet.
- **Fragmentation.** *IP packets may be split, or fragmented, into smaller packets.* This permits a large packet to travel across a network which can only handle smaller packets. *IP fragments and reassembles packets transparently.*
- **Packet timeouts.** Each IP packet contains a Time To Live (TTL) field, which is decremented every time a router handles the packet. If TTL reaches zero, the packet is discarded, *preventing packets from running in circles forever and flooding a network.*
- **Type of Service.** IP *supports traffic prioritization* by allowing packets to be labeled with an abstract type of service.
- **Options.** IP provides several optional features, *allowing a packet's sender to set requirements on the path* it **takes** through the network (source routing), **trace** the route a packet takes (record route), and **label** packets with security features.

Relationship between TCP and IP

- **IP** – Layer 3 protocol for logical addressing but, **TCP** - Layer 4 protocol which ensures reliability and is connection oriented.
- The source packet has destination address for its destination. TCP works with this logical address and helps the packets to reach their destinations, and provides acknowledgement when packet reached to its destination.
- **IP** : The forwarding service. It (unreliably) reloads messages from one wire onto another, so nodes can send messages to nodes they are not physically connected with. **TCP** : Kind of a wrapper around IP. Utilizes IP's messaging service in order to provide connections between processes running on different nodes, which are reliable (requests retransmissions if messages get lost), avoid congestion on the communication path and won't overcome receiver
- **IP** : From one end to another (remote device or connected device). **TCP** : From one process to another (process running on the two ends)

2.1. TCP/IP and the IP Layer overview

*The **Internet protocol suite** is the conceptual model and set of communications protocols used on the Internet and similar computer networks. The Internet protocol suite provides end-to-end data communication specifying how data should be packetized, addressed, transmitted, routed and received.*



Layer 4. Application Layer : Application layer is the top most layer of four layer TCP/IP model. Application layer is present on the top of the Transport layer. *Application layer defines TCP/IP application protocols and how host programs interface with Transport layer services to use the network.* Protocols are all *Higher-level protocols like DNS, HTTP , Telnet, SSH, FTP, TFTP (Trivial File Transfer Protocol), SNMP (Simple Network Management Protocol), , DHCP (Dynamic Host Configuration Protocol), X Windows, RDP (Remote Desktop Protocol)* etc.

Layer 3. Transport Layer : Transport Layer is the third layer of the four layer TCP/IP model. The position of the Transport layer is between Application layer and Internet layer. *The purpose of Transport layer is to permit devices on the source and destination hosts to carry on a conversation. Transport layer defines the level of service and status of the connection used when transporting data.*

The main protocols included at Transport layer are *TCP (Transmission Control Protocol) and UDP (User Datagram Protocol).*

Layer 2. Internet Layer : Internet Layer is the second layer of the four layer TCP/IP model. The position of Internet layer is between Network Access Layer and Transport layer.

Internet layer pack data into data packets known as IP datagrams, which contain source and destination address (logical address or IP address) information that is used to forward the datagrams between hosts and across networks. The Internet layer is also responsible for routing of IP datagrams.

The main protocols included at Internet layer *are IP (Internet Protocol), ICMP (Internet Control Message Protocol), ARP (Address Resolution Protocol), RARP (Reverse Address Resolution Protocol) and IGMP (Internet Group Management Protocol).*

Layer 1. Network Access Layer : Network Access Layer is the first layer of the four layer TCP/IP model. *Network Access Layer defines details of how data is physically sent through the network, including how bits are electrically or optically signaled by hardware devices that interface directly with a network medium, such as coaxial cable, optical fiber, or twisted pair copper wire.*

The protocols included in Network Access Layer are Ethernet, Token Ring, FDDI, X.25, Frame Relay etc.

2.2. IPv4 and IPv6 Address Types and Formats

IPv4 is 32 bits long and offers around 4,294,967,296 (2^{32}) addresses.

An Internet Protocol address (IP address) is a numerical label assigned to each device (e.g., computer, printer) participating in a computer network that uses the Internet Protocol for communication. *An IP address serves two principal functions: host or network interface **identification and location addressing**. Its role has been characterized as follows: "A **name** indicates what we seek. An **address** indicates where it is. A **route** indicates how to get there."*

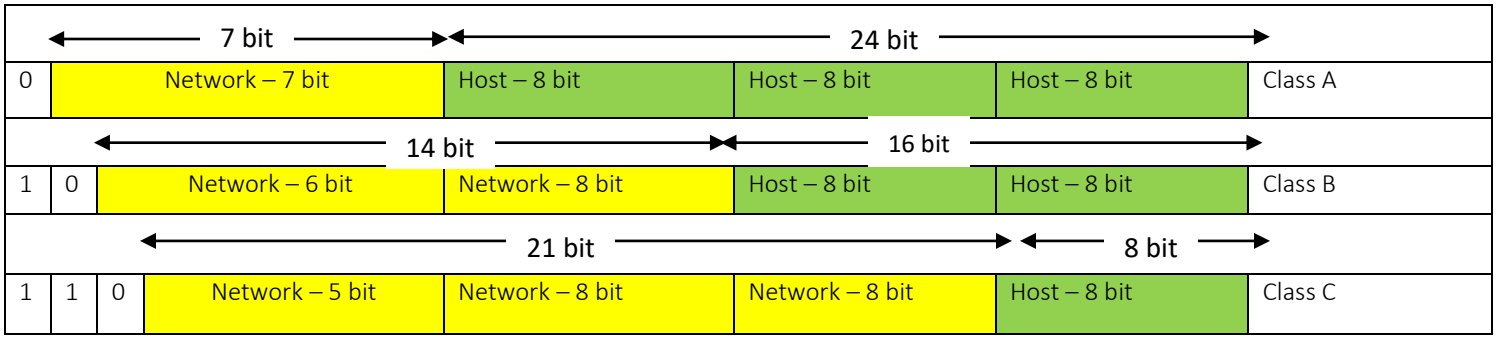
-Number of Networks : $2^{Network\_bits}$

-Number of Host : $2^{Host\_bits}-2$ *(2 IP addresses cannot be assigned to hosts, i.e. the first IP of a network is network number and the last IP is reserved for Broadcast IP.)*

IPv4 Address Class

Class	First bit of first 8 bit	IP Range	Subnet Bits	No. of Bits Network/ Host Bits	Max Hosts	Max Networks
Class A	0	1.x.x.x to 127.x.x.x	1	8/ 24	1,67,77,214 (2^{24}-2)	126 (2^{8-1})
Class B	10	128.0.x.x to 191.255.x.x	2	16/ 16	65,534 (2^{16}-2)	16,384 (2^{16-2})
Class C	110	192.0.0.x to 223.255.255.x	3	24/ 8	254 (2^{8}-2)	20,97,152 (2^{24-3})
Class D	1110	224.0.0.0 to 239.255.255.255.	4	Class D is reserved for Multicasting. Does not have any subnet mask.		
Class E	11110	240.0.0.0 to 255.255.255.254	5	reserved for experimental purposes only for R&D or Stud		

IPv4 Address Format



Private IP Address

- A private IP address is an IP address that's reserved for internal use behind a router or other Network Address Translation (NAT) device, apart from the public.
- Private IP addresses are in contrast to public IP addresses, which are public and cannot be used within a home or business network.
- Sometimes a private IP address is also referred to as a *local IP address*.
- The Internet Assigned Numbers Authority (IANA) reserves the following IP address blocks for use as private IP addresses:
  i. **10.0.0.0 to 10.255.255.255**, allows over 16 million addresses
  ii. **172.16.0.0 to 172.31.255.255**, allows over 1 million addresses
  iii. **192.168.0.0 to 192.168.255.255**, allows over 65,000 addresses

RFC1918 name	IP address range	host id	mask bits	number of addresses	*classful* description	largest CIDR block (subnet mask)
24-bit block	10.0.0.0 - 10.255.255.255 11111111.x.x.x	24 bits	8 bits	16,777,216	single class A network	10.0.0.0/8 (255.0.0.0)

20-bit block	172.16.0.0 - 172.31.255.255 11111111.1111xxxx.x.x	20 bits	12 bits	1,048,576	16 contiguous class B networks	172.16.0.0/12 (255.240.0.0)
16-bit block	192.168.0.0 - 192.168.255.255 11111111.11111111.x.x	16 bits	16 bits	65,536	256 contiguous class C networks	192.168.0.0/16 (255.255.0.0)

Reserved IP Address

- Another set of IP addresses that are restricted even further are called *reserved* IP addresses.
- These are similar to private IP addresses in the sense that they can't be used for communicating on the greater internet, but they're even more restrictive than that.
- The most famous reserved IP is 127.0.0.1. This address is called the loopback address and is used to test the network adapter or integrated chip. No traffic addressed to 127.0.0.1 is sent over the local network or public internet.
- Technically, the entire range from **127.0.0.0 to 127.255.255.255** is reserved for loopback purposes but you'll almost never see anything but 127.0.0.1 used in the real world.
- The range from **0.0.0.0 to 0.255.255.255** are also reserved but don't do anything at all.

***IPv6**

Internet Protocol Version 6 (IPv6) is a network layer protocol that enables data communications over a packet switched network. Packet switching involves the sending and receiving of data in packets between two nodes in a network.
IPv6 was planned to replace the widely-used Internet Protocol Version 4 (IPv4) that is considered the backbone of the modern Internet. IPv6 is often referred to as the "next generation Internet" *because of its expanded capabilities and its growth through recent large scale deployments.*
IPv4 is out of IP addresses. IPv4 has only 4.3 billion addresses, and with PCs, smartphones, tablets, gaming systems, and just about everything else connecting to the Internet we've tapped the system dry. IPv6 uses 128-bit addresses and is capable of 340 undecillion addresses. That is 340 times 10 to the 36th power, or 340 trillion trillion trillion possible IP addresses.
An IPv6 address can have either of the following two formats:

- Normal - Pure IPv6 format
- Dual - IPv6 plus IPv4 formats

An **IPv6 (Normal)** address has the following format: y : y : y : y : y : y : y : y where *y* is called a *segment* and can be any hexadecimal value between 0 and FFFF. The segments are separated by colons - not periods. An IPv6 normal address must have eight segments, however a short form notation can be used in the Tape Library Specialist Web interface for segments that are zero, or those that have leading zeros. The short form notation can not be used from the operator panel.
he following list shows examples of valid IPv6 (Normal) addresses:

- 2001 : db8: 3333 : 4444 : 5555 : 6666 : 7777 : 8888
- 2001 : db8 : 3333 : 4444 : CCCC : DDDD : EEEE : FFFF
- : : (implies all 8 segments are zero)
- 2001: db8: : (implies that the last six segments are zero)
- : : 1234 : 5678 (implies that the first six segments are zero)
- 2001 : db8: : 1234 : 5678 (implies that the middle four segments are zero)
- 2001:0db8:0001:0000:0000:0ab9:C0A8:0102 (This can be compressed to eliminate leading zeros, as follows: 2001:db8:1::ab9:C0A8:102)

An **IPv6 (Dual)** address combines an IPv6 and an IPv4 address and has the following format: y : y : y : y : y : y : x . x . x . x. The IPv6 portion of the address (indicated with y's) is always at the beginning, followed by the IPv4 portion (indicated with x's).

- In the IPv6 portion of the address, y is called a segment and can be any hexadecimal value between 0 and FFFF. The segments are separated by colons - not periods. The IPv6 portion of the address must have six segments but there is a short form notation for segments that are zero.
- In the IPv4 portion of the address x is called an octet and must be a decimal value between 0 and 255. The octets are separated by periods. The IPv4 portion of the address must contain three periods and four octets.

The following list shows examples of valid IPv6 (Dual) addresses:

- 2001 : db8: 3333 : 4444 : 5555 : 6666 : 1 . 2 . 3 . 4
- : : 11 . 22 . 33 . 44 (implies all six IPv6 segments are zero)
- 2001 : db8: : 123 . 123 . 123 . 123 (implies that the last four IPv6 segments are zero)
- : : 1234 : 5678 : 91 . 123 . 4 . 56 (implies that the first four IPv6 segments are zero)
- : : 1234 : 5678 : 1 . 2 . 3 . 4 (implies that the first four IPv6 segments are zero) 2001 : db8: : 1234 : 567

IPv6 - Address Types & Formats

IPv6 Addressing Modes

addressing mode refers to the mechanism of hosting an address on the network. IPv6 offers several types of modes by which a single host can be addressed. More than one host can be addressed at once or the host at the closest distance can be addressed.

- *Unicast*—An identifier for a single interface. A packet sent to a unicast address is delivered to the interface identified by that address. E.g. sending a letter to a friend or phoning them.
- *Multicast*—An identifier for a set of interfaces that typically belong to different nodes.
- *Anycast*—An identifier for a set of interfaces that typically belong to nearest nodes provides same service. A packet sent to an anycast address is delivered to the nearest interface *(in terms of routing distance)* in the anycast group
  Example: when you're in Europe, the 8.8.8.8 server will be a close by European server. When you're in Japan, that same IP address(8.8.8.8) will be a close by Asian server. *So, Normally, you want to reach one particular server, anycast wants an answer from "any" server . Anycast can be used for load balancing purposes.*

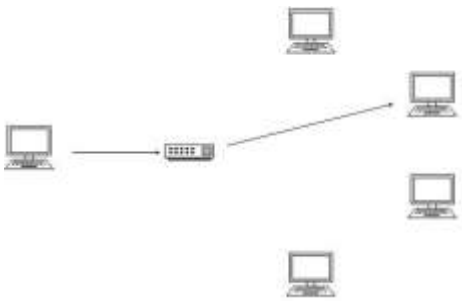


Fig1. Unicast

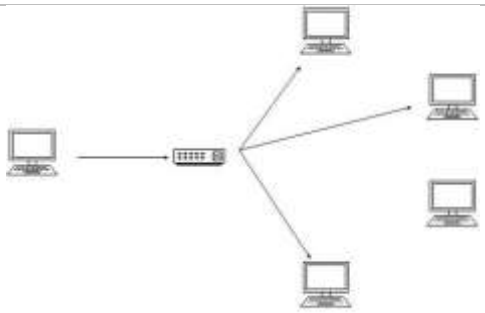


Fig2. Multicast

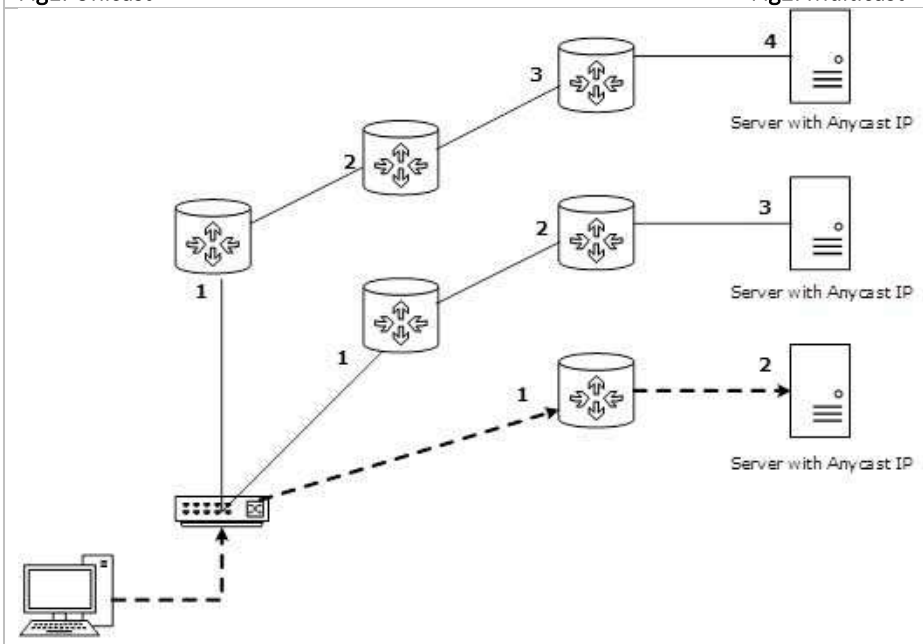


Fig3. Anycast

IPv6 Address Format

- Unicast addresses. *A packet is delivered to one interface*. In this case there is just one sender, and one receiver.(**one-to-one**) e.g. 3731:54:65fe:2::a7 , 0:0:0:0:0:0:0:1 is called the loopback address. It may be used by a node to send an IPv6 packet to itself. Unicast address are:-
  i. **Interface Identifiers**: used to identify interfaces on a link. They are required to be unique within a subnet prefix. It is recommended that the same interface identifier not be assigned to different nodes on a link. They may also be unique over a broader scope.
  ii. **Unspecified Address**- (`::/128`) The address 0:0:0:0:0:0:0:0 is called the unspecified address. It must never be assigned to any node. It indicates the absence of an address.
  iii. **Loopback Address**:- (`::1/128`) The unicast address 0:0:0:0:0:0:0:1 is called the loopback address. It may be used by a node to send an IPv6 packet to itself. It must not be assigned to any physical interface. It is treated as having Link-Local scope, and may be thought of as the Link-Local unicast address of a virtual interface (typically called the "loopback interface") to an imaginary link that goes nowhere.
  iv. **Global unicast addresses**, which are conventional, publicly routable address, just like conventional IPv4 publicly routable addresses.

3 Bits	45 Bits	16 Bits	64 Bits
001	Global Routing Prefix	Subnet ID	Interface ID

- **GRP(**Global Routing Prefix) is used to identify a address type like multicast or an address range assigned to a site.
- **Subnet ID** is used to identify subnets within a site, used within a organization's site.
- **Interface ID** is used to identify an interface on a specific subnet within the site. Its size is 64 bits. It is known Node ID or Host ID in IPv4.

v. **Link-local addresses (FE80::/10 e.g. fe80::200:5aee:feaa:20a2)** are akin to the private, non-routable addresses in IPv4 (10.0.0.0/8, 172.16.0.0/12, 192.168.0.0/16). They are not meant to be routed, but confined to a single network segment. Link-local addresses mean you can easily throw together a temporary LAN, such as for conferences or meetings, or set up a permanent small LAN the easy way.

10 Bits	54 Bits	64 Bits
1111 1110 10	000 … 000	Interface ID

vi. **Site-Local IPv6 Unicast Addresses**: Site-Local addresses were originally designed to be used or addressing inside of a site without the need for a global prefix. **e.g FEC0::/10**

10 Bits	54 Bits	64 Bits
1111 1110 11	000 … 000	Interface ID

vii. **Transition Address**

- **IPv4-mapped address** (**::ffff/96 e.g. ::ffff:192.0.2.47**) Two types of IPv6 addresses are defined that carry an IPv4 address in the low-order 32 bits of the address. Is used to represent an IPv4 as IPv6 address.
- **IPv4 compatible address**- **::192.172.12.3 –** communicating with IPv6 over an IPv4 infrastructure that uses public IPv4 address.
- **6to4 address**- 2002:WWXX:YYZZ:SubnetID:InterfaceID is assigned a node for the 6to4 transition technology.
- **Teredo** address: 2001::/32 is assigned to a node for the Teredo IPv6 transition technology

- **Multicast addresses.** *A* packet is delivered from one or more points to multiple or a set of interfaces.(**one-to-many or many-to-many**). These addresses are used to identify multicast groups. They should only be used as destination addresses, never as source addresses. e.g. addresses fall under the range ff00::/8 , FF01:0:0:0:0:0:0:1

The main goal of multicasting is having an efficient network to save bandwidth on links by optimizing the number of packets exchanged between nodes. Multicast implies the concept of a group:

- Any node can be a member of a multicast group
- A source node may send packets to a multicast group
- All members of a multicast group get packets that are sent to the group

Note : IPv6 does not use broadcast messages.

Multicast Address e.g. ff00::/8

1111 1111	Flags	Scope	Group Identifier
8 bits	4 bits	4 bits	112 bits

For IPv6 multicast addresses, the first eight bits are reserved as 1111 1111. Thus, the prefix of an IPv6 multicast address is ff00::/8. Similar to IPv6 Link Local addresses, it is easy to identify an IPv6 multicast address, because IPv6 multicast addresses have left most hexadecimal digits as "FF"

• After the leftmost 8 bits which are reserved as "1111 1111", the next four bits are known as flags. Only 3 of the 4 flag bits in the flags field are defined currently. The most significant bit in the 4 bits flags field is reserved for future use. The remaining three flags are known as R, P and T.

4 Bits			
0	1	1	0
0	1	2	3

4 Bits inside flags field	Flag name	When "0" set	When "1" set
0 (Most Significant Bit)	Currently not in use	Currently not in use	Currently not in use
1	R (Rendezvous)	When R flag set to 0, the multicast rendezvous point is not embedded with multicast address	When R flag set to 1, the multicast rendezvous point is embedded with multicast address

2	P (Prefix)	When P flag set to 0, the multicast address is not based on network prefix	When P flag set to 1, the multicast address based on network prefix
3 (Least Significant Bit)	T (Transient)	When T flag set to 0, the multicast address is a permanently assigned (well-known) multicast IPv6 address	When T flag set to 1, the multicast address is a transient (Dynamically assigned) multicast address

• After the leftmost 8 bits which are reserved as "1111 1111", and the next four flag bits, the next four bits are defined as the Scope bits. Scope bits (4 bits) are used to indicate the scope of delivery of IPv6 multicast traffic.
The following table lists the values possible currently for the scope field. E.g. **FF02:: – link-local scope**.

Hex Value	Scope	Meaning
0	Reserved	Currently not in use
1	Interface-local scope	The Interface-local scope is limited for a local single interface only. Useful only for loopback delivery of multicasts within a node.
2	Link-local scope	Link-local scope is defined for the local link. The traffic with the multicast address of FF02::2 is limited to local link scope. An IPv6 router will never forward the multicast traffic destined to FF02::2 beyond the local link.
3	Subnet-local scope	Subnet-local scope ranges subnets on multiple links.

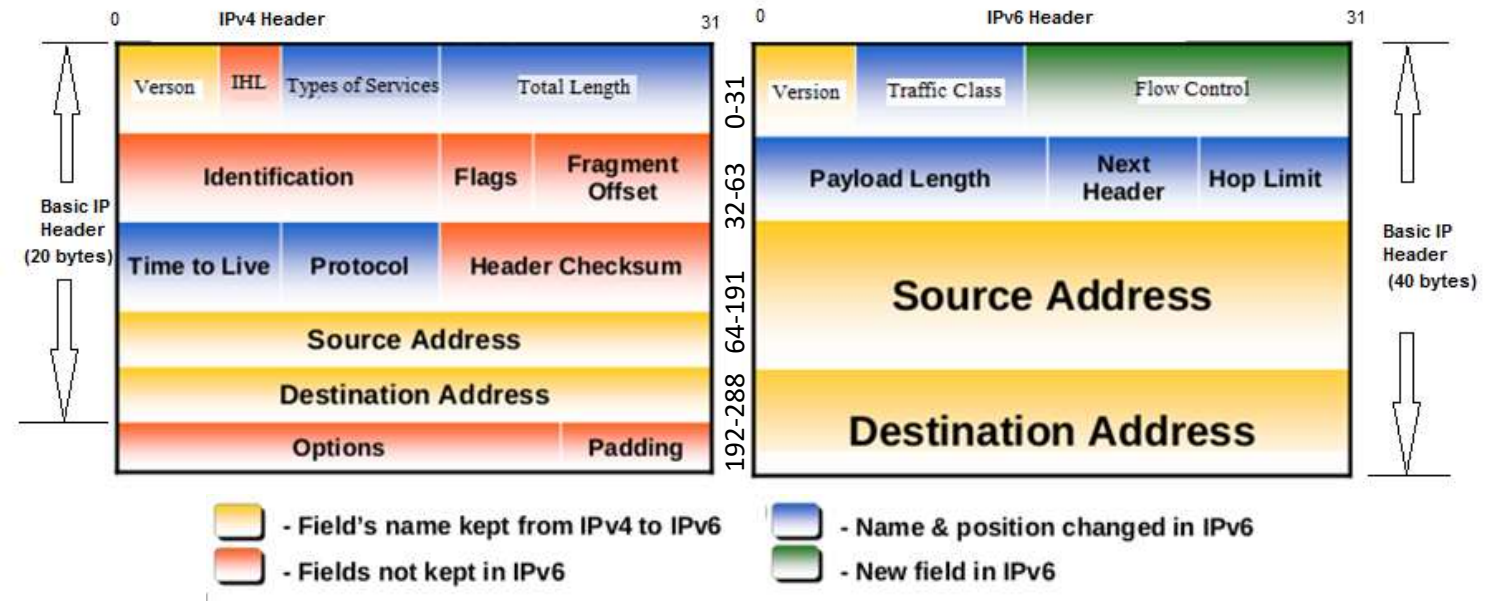
Group ID: identifies the multicast group and is unique within the scope. Its size is 112 bits.

- **Anycast addresses.** *A* packet is delivered to the nearest of multiple interfaces (the "nearest" one, according to the routing protocols' measure of distance*).*(**one-to-any**) e.g. FF01:0:0:0:0:0:0:1 (IPv4 - 224.0.0.0/4)

Anycast address are Link-Local (FE80::/10), Site-Local(FEC0::/10), Aggrigatable Global(2001::/16, 2002::/16,3FFE::/16) - Anycast addresses use aggregatable global unicast addresses. They can also use site-local or link-local addresses. Note that it is impossible to distinguish an anycast address from a unicast address.

An IPv6 anycast address is an identifier for a set of interfaces (typically belonging to different nodes) - A packet sent to an anycast address is delivered to one of the interfaces identified by that address (the “nearest” one, according to the routing protocol’s measure of distance).

2.3. IPv4 and IPv6 Header Structure



IPv6 header design is focused mainly on simplicity - to keep the datagram as simple as possible and to keep the size of the headers fixed. The reason for this was to increase processing performance - simple constant size headers can be processed quickly, at or very close to wire-speed. Following are the main comparison between IPv4 header and IPv6 header.

• IPv6 header is much simpler than IPv4 header.

• The size of IPv6 header is much bigger than that of IPv4 header, because of IPv6 address size. IPv4 addresses are 32bit binary numbers and IPv6 addresses are 128 bit binary numbers.

• In IPv4 header, the source and destination IPv4 addresses are 32 bit binary numbers. In IPv6 header, source and destination IPv6 addresses are 128 bit binary numbers.

• IPv4 header includes space for IPv4 options. In IPv6 header, we have a similar feature known as extension header. IPv4 datagram headers are normally 20-byte in length. But we can include IPv4 option values also along with an IPv4 header. In IPv6 header we do not have options, but have extension headers.
• The fields in the IPv4 header such as IHL (Internet Header Length), identification, flags are not present in IPv6 header.
• Time-to-Live (TTL), a field in IPv4 header, typically used for preventing routing loops, is renamed to it's exact meaning, "Hop Limit".

IPv6 Header Fields –

Version (4-bits): It represents the version of Internet Protocol, i.e. 6=0110.
Traffic Class (8-bits): replaces the *Type Of Service (TOS)* field in the IPv4 header, These 8 bits are divided into two parts. The most significant 6 bits are used for Type of Service to let the Router Known what services should be provided to this packet. Classifies traffic for QoS - minimize delay, maximize throughput, maximize reliability and minimize monetary cost.. The least significant 2 bits are used for Explicit Congestion Notification (ECN).
Flow Label (20-bits): This label is used to maintain the unique and sequential flow, delivery of the packets belonging to a communication between a 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 (16-bits): replaces the *Total Length* field from the IPv4 header, measures the length of the datagram This field is used to tell the routers how much information a particular packet contains in its payload. Payload is composed of Extension Headers and Upper Layer data.
Next Header (8-bits): This field replaces the *Protocol* field. This field is used to indicate either the type of Extension Header, or if the Extension Header is not present then it indicates the Upper Layer PDU. The values for the type of Upper Layer PDU are same as IPv4's. - **Hop-by-hop option header :** Next header value is 0, read by all devices in transit network - **Destination Options Header :** Next header value is 60, read by destination devices - **Routing Header :** Next header value is 43, Contains methods to support making routing decision - **Fragment Header :** Next header value is 44, contains parameters of datagram fragmentation and reassembly - **Authentication Header :** Next header value is 51, information regarding Integrity and authentication, security - **Encapsulation Security Payload Header :** Next header value is 50, encryption information, Confidentiality
Hop Limit (8-bits): This is same as TTL in IPv4. This field is used to stop packet to loop in the network infinitely. The value of Hop Limit field is decremented by 1 as it passes a link (router/hop). When the field reaches 0 the packet is discarded.
Source Address (128-bits): This field indicates the address of originator or sender of the packet.
Destination Address (128-bits): This field provides the address of intended recipient or receiver of the packet.

Details -

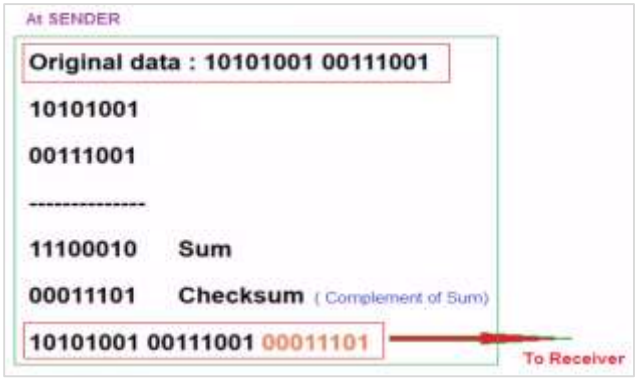
- The total length of the datagram header doubled (from 20 bytes to 40 bytes) although the IPv6 addresses are four times as long.
- In IPv6 just a subset of IPv4 header fields have been adopted.
- The whole second line of the IPv4 datagram, designed for fragmentation, has been moved to an extension header in IPv6.
- The CRC (cyclic redundancy check) has been skipped for two good reasons: First, frame consistency is checked in lower layers, so it is largely redundant.
- Second, CRC decelerates the datagram processing – every forwarding node decreases the datagram lifetime, so it changes the header and must recalculate the CRC.
- Thanks to the constant header length the corresponding header length field is not necessary anymore.

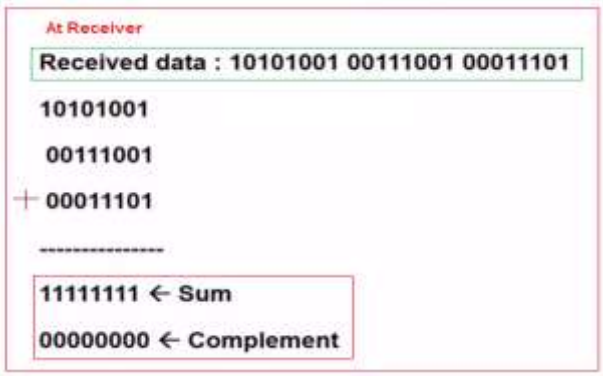
IP V4 Header Fields -

- ***Protocol Version(4 bits)*** : This is the first field in the protocol header. This field occupies 4 bits. *This signifies the current IP protocol version being used.* Most common version of IP protocol being used is version 4 while version 6 is out in market and fast gaining popularity.
- ***Internet Header Length-IHL(4 bits)*** *: This field provides the length of the IP header*. The length of the header is represented in 32 bit words. This length also includes IP options (if any). Since this field is of 4 bits so the maximum header length allowed is 60 bytes. Usually when no options are present then the value of this field is 5. Here 5 means five 32 bit words ie 5 *4 = 20 bytes.
- ***Type of service(8 bits)*** : It is used tell the network how to treat the IP packet. These bits are generally used to indicate the Quality of Service (QoS) for the IP Packet are *:* minimize delay, maximize throughput, maximize reliability and minimize monetary cost.
- ***Total length(16 bits)***: *This represents the total IP datagram length* including header and data *in bytes.* Since the header length (described above) gives the length of header and this field gives total length so the length of data and its starting point can easily be calculated using these two fields. Since this is a 16 bit field and it represents length of IP datagram so the maximum size of IP datagram can be 2^{16} = 65535 bytes.
- ***Identification(16 bits)***: *This field is used for uniquely identifying the IP datagrams.* This value is incremented every-time an IP datagram is sent from source to the destination is used for reassembling the packet at the destination.
- ***Flags(3 bits)***: This It indicates if the IP packet can be further fragmented or not and if the packet is the last fragment or not of a larger transfer.

- bit 0: Reserved; must be zero.
- bit 1: Don't Fragment (DF)
- bit 2: More Fragments (MF)

- ***Fragment offset(13 bits)***: In case of fragmented IP data grams, this field contains the offset(in terms of 8 bytes units) from the start of IP datagram. So again, this field is used in reassembly process of fragmented IP datagrams.
- ***Time to live(8 bits)*** : This value field helps prevent datagrams from persisting (e.g. going in circles) on an internet. *The value of this field in the beginning is set to be around 32 or 64 (lets say) but at every hop over the network this field is decremented by one. When this field becomes zero, the data gram is discarded.*
- ***Protocol(8 bits)*** : This field *represents the transport layer protocol TCP, UDP* that handed over data to IP layer. This field comes in handy when the data is demultiplex-ed at the destination as in that case IP would need to know which protocol to hand over the data to.
- ***Header Checksum(16 bits)*** : used for error-checking of the header.





- ***Source and destination IP(32 bits each)*** : These *fields store the source and destination address* respectively. Since size of these fields is 32 bits each so an IP address os maximum length of 32 bits can be used. So we see that this limits the number of IP addresses that can be used. To counter this problem, IP V6 has been introduced which increases this capacity.
- ***Options(Variable length)*** : This field represents a list of options that are active for a particular IP datagram. This is an optional field that could be or could not be present. If any option is present in the header then the first byte is represented as show in table 1 :

In the description above, the 'copy flag' means that copy this option to all the fragments in case this IP datagram gets fragmented. The 'option class' represents the following values : 0 -> control, 1-> reserved, 2 -> debugging and measurement, and 3 -> reserved. Some of the options are shown in table 2:

- **Padding** : Variable size bit field. Used to ensure that the datagram header is a multiple of 32 bits in length.
- ***Data***: This *field contains the data from the protocol layer that has handed over the data to IP layer*. Generally this data field contains the header and data of the transport layer protocols. Please note that each TCP/IP layer protocol attaches its own header at the beginning of the data it receives from other layers in case of source host and in case of destination host each protocol strips its own header and sends the rest of the data to the next layer.

class	number	length	description
0	0	–	end of option list
0	1	–	no operation
0	2	11	security
0	3	var.	loose source routing
0	9	var.	strict source routing
0	7	var.	record route
0	8	4	stream id
2	4	var.	INTERNET time stamp

Table 2 : Options(Variable length)

IPv4 vs IPv6

- The 128-bits in the IPv6 address are eight 16-bit hexadecimal blocks separated by colons. For example, 2dfc:0:0:0:0217:cbff:fe8c:0.
- IPv4 addresses are divided into "classes" with Class A networks for a few huge networks, Class C networks for thousands of small networks, and Class B networks that are in between. IPv6 uses subnetting to adjust network sizes with a given address space assignment.
- IPv4 uses class-type address space for multicast use (224.0.0.0/4). IPv6 uses an integrated address space for multicast, at FF00::/8.
- IPv4 uses "broadcast" addresses that forced each device to stop and look at packets. IPv6 uses multicast groups.
- IPv4 uses 0.0.0.0 as an unspecified address, and class-type address (127.0.0.1) for loopback. IPv6 uses :: and ::1 as unspecified and loopback address respectively.
- IPv4 uses globally unique public addresses for traffic and "private" addresses. IPv6 uses globally unique unicast addresses and local addresses (FD00::/8).
- IPv4 has lack of security. IPv6 has a built-in strong security : Encryption and Authentication.
- IPv4 enabled clients can be configured manually or they need some address configuration mechanism. It does not have a mechanism to configure a device to have globally unique IP address.

Fragmentation Process

- In TCP/IP, fragmentation refers to the process of breaking packets into the smallest maximum size packet data unit (PDU) supported by any of the underlying networks, IN OSI Model referred as *segmentation.*
- The Maximum Transmission Unit (MTU) is the largest size of IP datagram which may be transferred using a specific data link connection The MTU value is a design parameter of a LAN and is a mutually agreed value (i.e. both ends of a link agree to use the same specific value) for most WAN links.
- When a datagram is fragmented, either by the originating device or by one or more routers transmitting the datagram, it becomes multiple fragment datagrams. The destination of the overall message must collect these fragments and then *reassemble* them into the original message in correct order.
  - **Fragmentation**: a technique to limit datagram size to MTU of any network.
  - IP uses fragmentation – split datagrams into pieces to fit in network with small MTU
  - Router detects datagram larger than network MTU - Splits into pieces called fragments - Each piece smaller than output network MTU
  - Each fragment has datagram header and is sent separately, Ultimate destination reassembles fragments

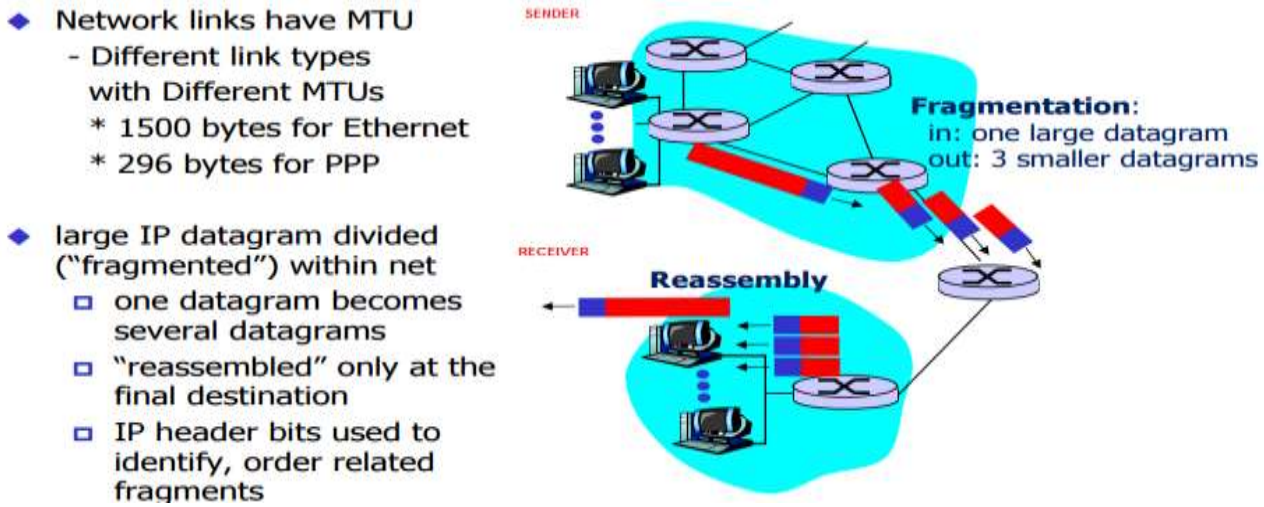


Fig. Fragmentation and Reassembly

Comparisons between fragmentation process between IPv6 and IPv4

1. Fields for handling fragmentation are not in the basic IPv6 header but put into an extension header if fragmentation is required, this makes IPv6 fragmentation slim because this fragmentation extension header is only inserted if the packet if fragmentation needs to be done. In IPv4 flags field handle fragmentation e.g. 0=reserved, 1=do not fragment, 2= more fragment
2. IPv6 routers do not fragment anymore. Fragmentation has to be done by source host. Source will evaluate the packet size by using path MTU discovery.
3. Length of header in IPv4 is 20 bytes, in IPv6 is 40 bytes.

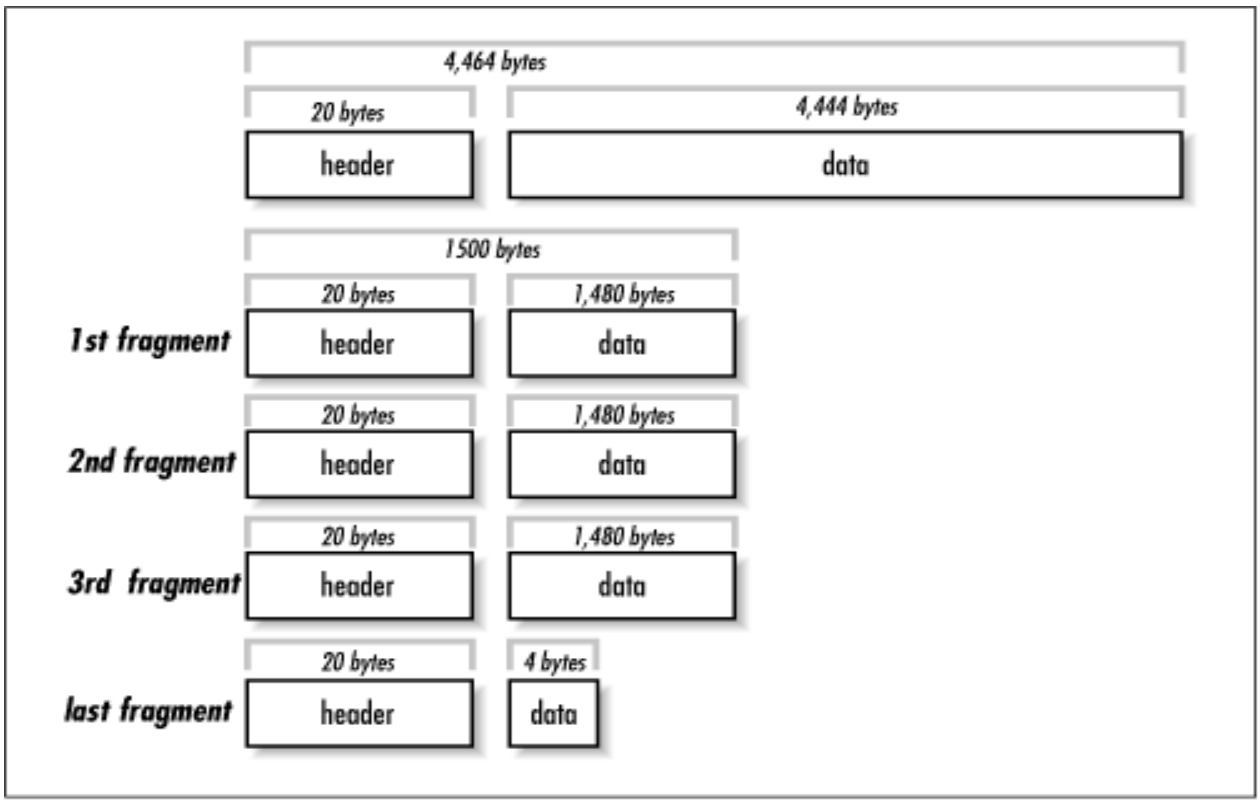
Example : Fragmentation Process in IPv4

Suppose, total datagram length is 4464 Bytes (Including Header), MTU is 1500

- We know for IPv4, Header length = 20 Bytes
- Total possible fragments = (4464-20) / (1500-20) = 3.0027 i.e. approximate are 4.

- For 1st ,2nd , 3rd fragment, each datagram length(including header length) = MTU = 1500,
- Data (excluding header length) = 1500 - 20 = 1480

- For 4th fragment, data length = 4444 – 3 x 1480 = 4
- More Fragment, MF=0



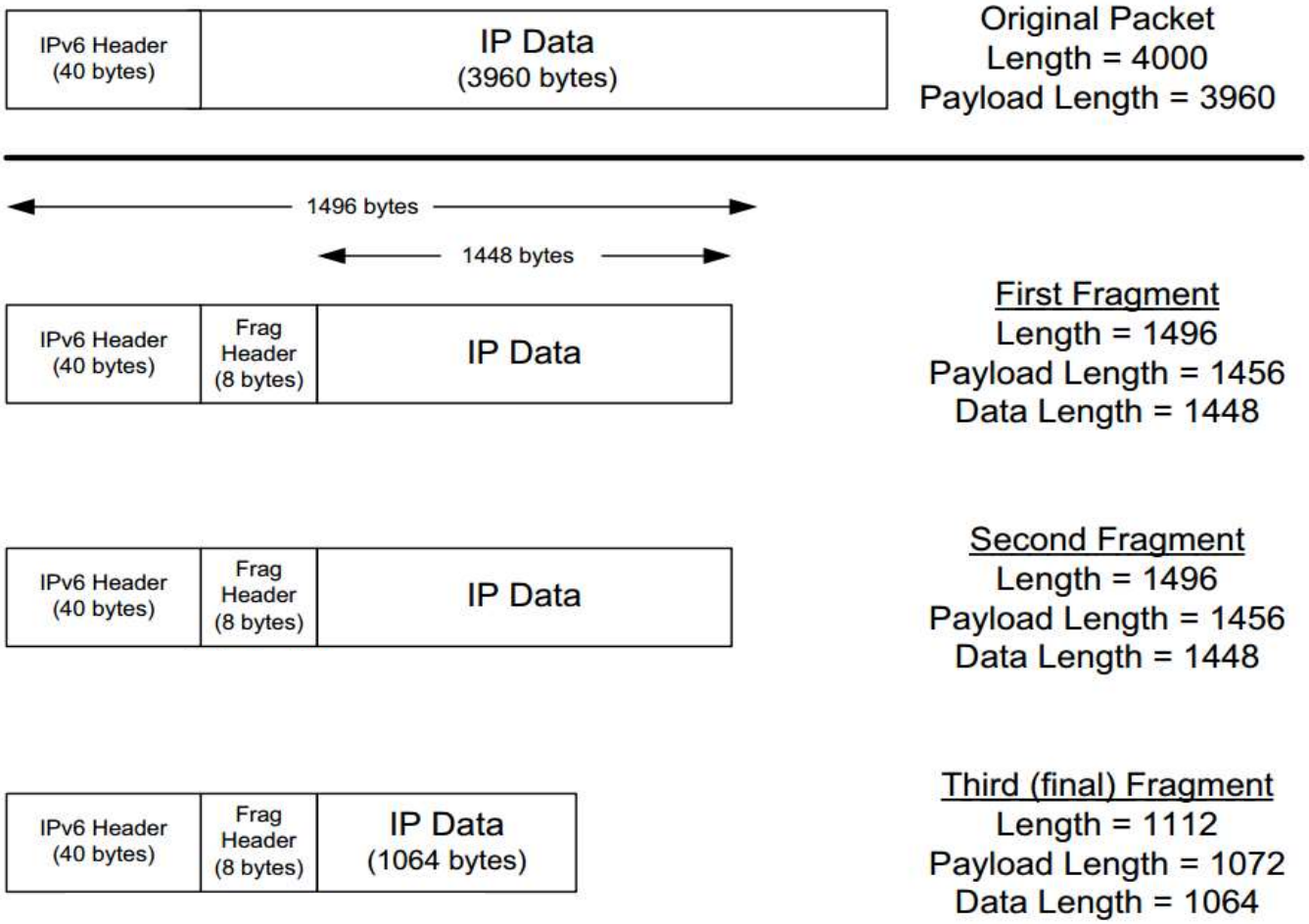
Example : Fragmentation Process in IPv6

Suppose, total datagram length is 4000 Bytes (Including Header), MTU is 1496

- We know for IPv6, Header length = 40 Bytes and Fragment Header = 8 Bytes
- Total possible fragments = (4000-40) / (1496-8-40) = 2.7348 i.e. approximate are 3.

- For **1st** and **2nd** fragment, each datagram length (including IPv6 header + Fragment Header) = MTU = 1496,
- Payload Length (excluding IPv6 header) = 1496 – 40 = 1456
- Data (excluding IPv6 header + Fragment Header) = 1496 – 40 - 8 = 1448
- More Fragment, MF=1
- Offset **1st** = 0

- For **3rd** fragment, data length = 3960 – 2 x 1448= 1064
- Payload length = 1064 + 8 = 1072
- Datagram or Fragment length = 1072 + 40 = 1112



Assignment : Compare the fragmentation process for IPv4 and IPv6 with suitable example.

2.4. Internet RFCs (Request for Comments) - rfc825

- A Request for Comments (RFC) is a formal document from the Internet Engineering Task Force (IETF) that is the result of committee drafting and subsequent review by interested parties
- RFC documents were invented by Steve Crocker in 1969 to help record unofficial notes on the development of ARPANET. *RFCs have since become official documents of Internet specifications, communications protocols, procedures, and events*
- RFCs are a collection of documents which describe various actual and suggested practices relevant to the Internet.
- A RFC is a type of *publication* from the Internet Engineering Task Force (IETF) and the Internet Society (ISOC), *the principal technical development and standards-setting bodies for the Internet*.
- An RFC is *authored by engineers and computer scientists in the form of a* memorandum describing methods, behaviors, research, or innovations applicable to the working of the Internet and Internet-connected systems. It is submitted 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.
- Most RFCs deal with technical arrangements and conventions, often called **protocols**.
- RFCs are the **publications** of the (proposed) standards. **Without** publication and broad distribution, standards are pretty much useless.

Why are RFCs important for the Internet standards and history?

- Because a subset of the RFCs (IETF documents) specify the protocol standards for TCP/IP and The Internet. If you intend to write software to implement application or network protocols to be used on the Internet, you'll be reading relevant RFCs to comply with the on-the-wire bit protocols.
- The whole body of RFCs constitute a primary source for History of the Internet in that it's what we who were developing the net were writing to each other, along with archives of the Internet Engineering Task Force (IETF) (electronic) mailing lists. If you read them in order, you'll have a good view of how the Internet came to be the way it is.
- For those studying Computer Networking, quite a number of the RFCs will teach practical things that the typical academic textbook will not cover.
- The terms and descriptions used in the RFCs to discuss a technology or practice often become the standard terminology as well.
- RFCs are not standards or authoritative - until they are promoted to "**STD**"s.
- Understanding the relationships between the RFCs can help you better understand the limitations of particular technologies or protocols or practices.

The RFC series contains three sub-series for IETF RFCs:

1. **BCP:** Best Current Practice; mandatory IETF RFCs not on standards track, see below.
2. **FYI:** For Your Information; informational RFCs promoted by the IETF as specified in RFC 1150 (FYI 1). In 2011, RFC 6360 obsoleted FYI 1 and concluded this sub-series.
3. **STD:** Standard; this used to be the third and highest maturity level of the IETF standards track specified in RFC 2026

Fundamental Internet protocols are listed below, together with the RFC's that describe them.

Protocol	Acronym	Purpose	RFC
Internet Protocol	IP	Physical network	RFC-791
Internet Control Message Protocol	ICMP	Status messaging	RFC-792
Transmission Control Protocol	*TCP*	Guaranteed delivery	RFC-793
User Datagram Protocol	UDP	Coordination, Audio	RFC-768
Telnet Protocol	TELNET	Remote login	RFC-764
File Transfer Protocol	FTP	Network utility	RFC-765
Simple Mail Transfer Protocol	SMTP	*Email*	RFC-788
Network News Transfer Protocol	NNTP	*Usenet*	RFC-977
Hypertext Transfer Protocol	HTTP	*Web*	RFC-2068

RFC Status

Not all RFCs are standards. Each RFC is assigned a description with regard to status **within** the Internet standardization process. This status is one of the following: *Informational*, *Experimental*, *Best Current Practice*, *Standards Track*, or *Historic*.

Each RFC is static; if the document is changed, it is submitted again and assigned a new RFC number.

(i)"Standards Track" :

- Standards-track documents are further divided into *Proposed Standard*, *Draft Standard*, and *Internet Standard* documents.
- Only the IETF, represented by the Internet Engineering Steering Group (IESG), can approve standards-track RFCs.
- If an RFC becomes an Internet Standard (STD), it is assigned an STD number but retains its RFC number. The definitive list of Internet Standards is the Official Internet Protocol Standards. Previously STD 1 used to maintain a snapshot of the list.[14]

(ii)"Informational" :

An *informational* RFC can be nearly anything from April 1 jokes to widely recognized essential RFCs like Domain Name System Structure and Delegation (RFC 1591). Some informational RFCs formed the FYI sub-series.

(iii)"Experimental":

An *experimental* RFC can be an IETF document or an individual submission to the 'RFC Editor'. A draft is designated experimental if it is unclear the proposal will work as intended or unclear if the proposal will be widely adopted. An experimental RFC may be promoted to standards track if it becomes popular and works well.

(iv)"Best Current Practice":

- The Best Current Practice subseries collects administrative documents and other texts which are considered as official rules and not only *informational*, but which do not affect *over the wire data*. The border between standards track and BCP is often unclear. If a document only affects the Internet Standards Process, like BCP 9,[16] or IETF administration, it is clearly a BCP. If it only defines rules and regulations for Internet Assigned Numbers Authority (IANA) registries it is less clear; most of these documents are BCPs, but some are on the standards track.
- The BCP series also covers technical recommendations for how to practice Internet standards; for instance the recommendation to use source filtering to make DoS attacks more difficult (RFC 2827: "*Network Ingress Filtering: Defeating Denial of Service Attacks which employ IP Source Address Spoofing*") is BCP 38.

(v)"Historic":

A *historic* RFC is one that the technology defined by the RFC is no longer recommended for use, which differs from "Obsoletes" header in a replacement RFC. For example, RFC 821 (SMTP) itself is obsoleted by various newer RFCs, but SMTP itself is still "current technology", so it is not in "Historic" status.[17] On the other hand, since BGP version 4 has entirely superseded earlier BGP versions, the RFCs describing those earlier versions (e.g. RFC 1267) have been designated historic.

(vi)"Unknown":

Status *unknown* is used for some very old RFCs, where it is unclear which status the document would get if it were published today. Some of these RFCs would not be published at all today; an early RFC was often just that: a simple request for comments, not intended to specify a protocol, administrative procedure, or anything else for which the RFC series is used today.

RFC Streams

(1) IETF,

The **Internet Engineering Task Force** (**IETF**) is an open standards organization, which develops and promotes voluntary Internet standards, in particular the standards that comprise the Internet protocol suite (TCP/IP).[2] It has no formal membership or membership requirements. All participants and managers are volunteers, though their work is usually funded by their employers or sponsors.

(2) IRTF,

The **Internet Research Task Force** (**IRTF**) focuses on longer term research issues related to the Internet while the parallel organization, the Internet Engineering Task Force (IETF), focuses on the shorter term issues of engineering and standards making. The Internet Research Task Force (IRTF) promotes research of importance to the evolution of the Internet by creating focused, long-term research groups working on topics related to Internet protocols, applications, architecture and technology.

(3) IAB,

a committee of the Internet Engineering Task Force (IETF) and an advisory body of the Internet Society (ISOC). Its responsibilities include architectural oversight of IETF activities, Internet Standards Process oversight and appeal, and the appointment of the Request for Comments (RFC) Editor. The IAB is also responsible for the management of the IETF protocol parameter registries.

(4) *independent submission*

Only the IETF creates BCPs and RFCs on the standards track. An *independent submission* is checked by the IESG for conflicts with IETF work; the quality is assessed by an *independent submission editorial board*. In other words, IRTF and *independent* RFCs are supposed to contain relevant info or experiments for the Internet at large not in conflict with IETF work