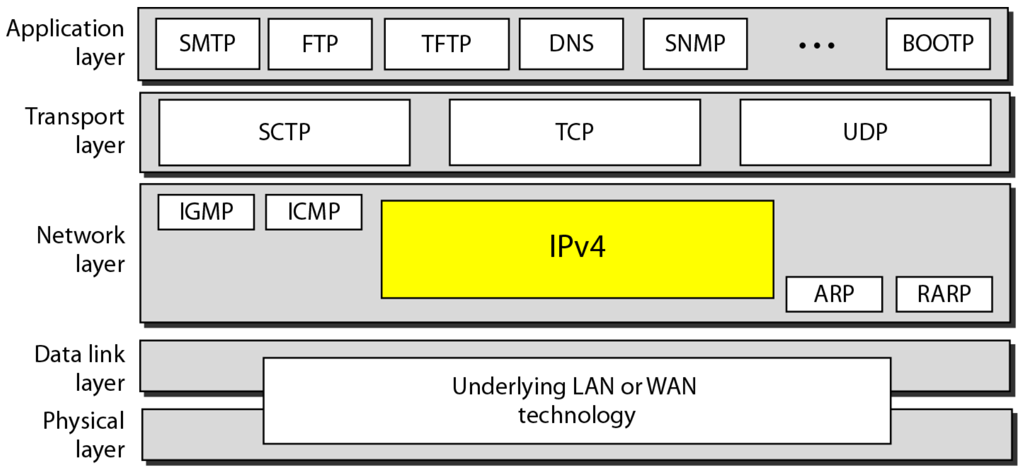
# Network Layer: Part 3

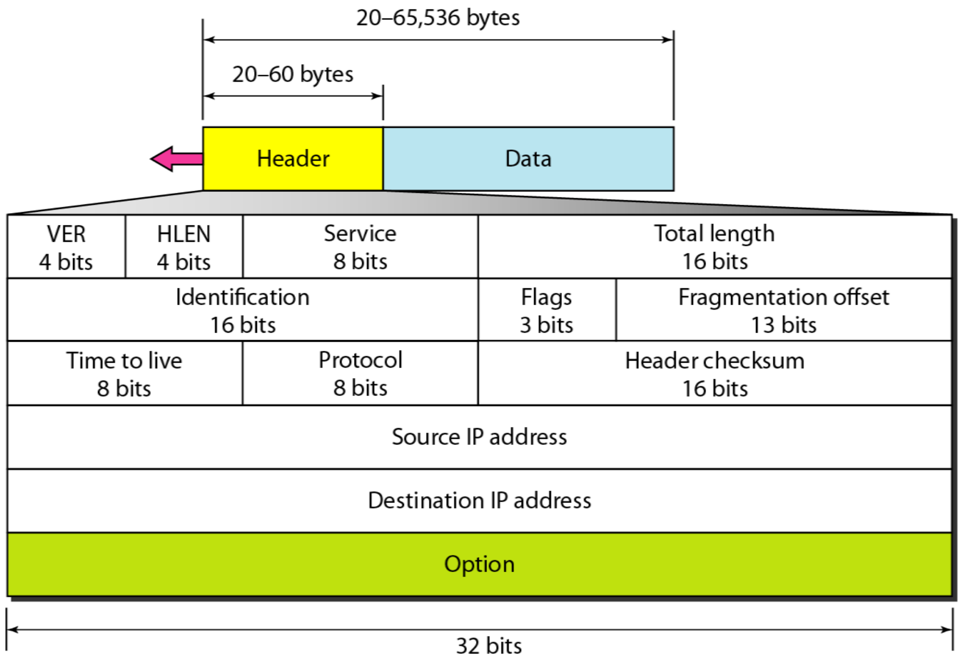
## TCP / IP Suite




***Position of IPv4 in TCP/IP protocol suite***

## IPv4 datagram format / IP Packet format

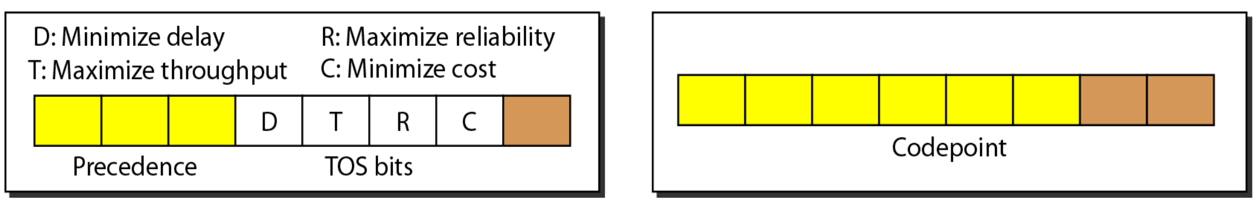
The Internet Protocol version 4 (IPv4) is the delivery mechanism used by the TCP/IP protocols. IPv4 is an unreliable and connectionless datagram protocol-a best-effort delivery service. The term *best-effort* means that IPv4 provides no error control or flow control (except for error detection on the header). IPv4 assumes the unreliability of the underlying layers and does its best to get a transmission through to its destination, but with no guarantees. If reliability is important, IPv4 must be paired with a reliable protocol such as TCP.




**Version (VER)**.This 4-bit field defines the version of the IPv4 protocol. Currently the version is 4. However, version 6 (or IPng) may totally replace version 4 in the future.

**Header length (HLEN)**.This 4-bit field defines the total length of the datagram header in 4 byte words. The header length is 20 bytes, and the value of this field is 5 (5 x 4 = 20). When the option field is at its maximum size, the value of this field is 15 (15 x 4 = 60).

**Services:** This is 8 bit fields.




Service type

Differentiated services

Types of service TOS Bits Description
0000 Normal (default)
0001 Minimize cost
0010 Maximize reliability
0100 Maximize throughput
1000 Minimize delay

**Total length**: This is 16 bit field. The total length field defines the total length of the datagram including the header. The header length can be found by multiplying the value in the HLEN field by 4.

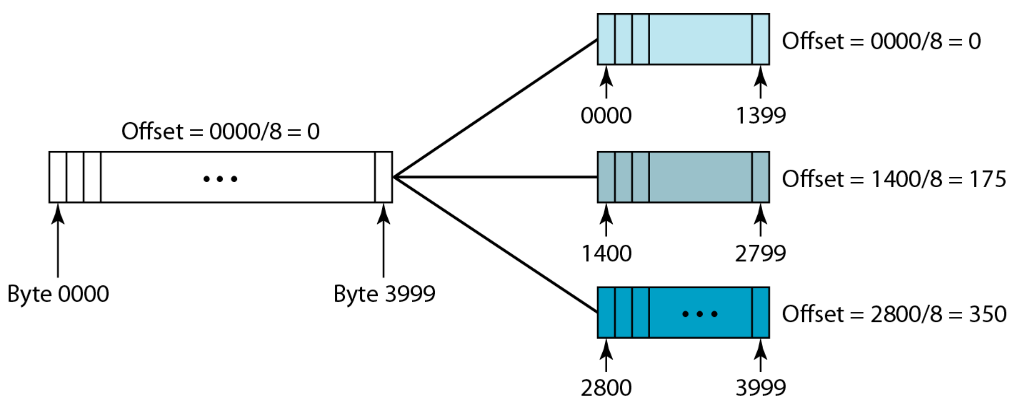
Length of data = total length - header length

**Identification**. This 16-bit field identifies a datagram originating from the source host. The combination of the identification and source IPv4 address must uniquely define a datagram as it leaves the source host. When a datagram is fragmented, the value in the identification field is copied to all fragments. In other words, all fragments have the same identification number, the same as the original datagram.

**Flag:** This is a 3-bit field. The first bit is reserved. The second bit is called the *do not fragment* bit. If its value is 1, the machine must not fragment the datagram.The third bit is called the *more fragment* bit. If its value is 1, it means the datagram is not the last fragment; there are more fragments after this one. If its value is 0, it means this is the last or only fragment.

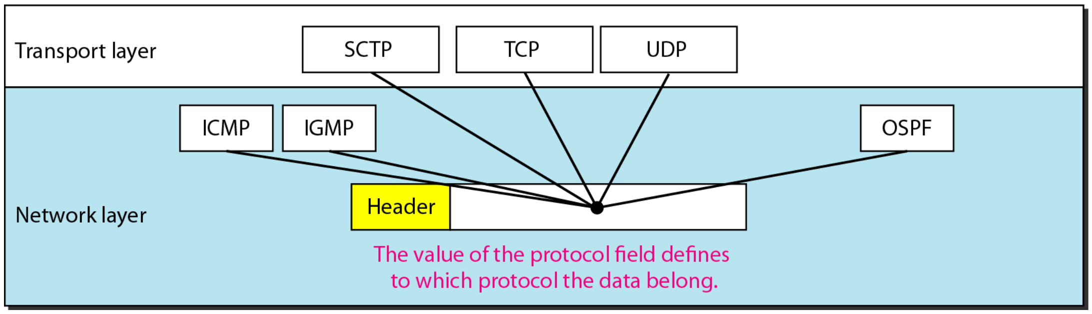

**Fragmentation offset**. This 13-bit field shows the relative position of this fragment with respect to the whole datagram.

The bytes in the original datagram are numbered 0 to 3999. The first fragment carries bytes 0 to 1399. The offset for this datagram is 0/8 =O. The second fragment carries bytes 1400 to 2799; the offset value for this fragment is 1400/8 = 175. Finally, the third fragment carries bytes 2800 to 3999. The offset value for this fragment is 2800/8 =350.




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

**Protocol**. This 8-bit field defines the higher-level protocol that uses the services of the IPv4 layer.



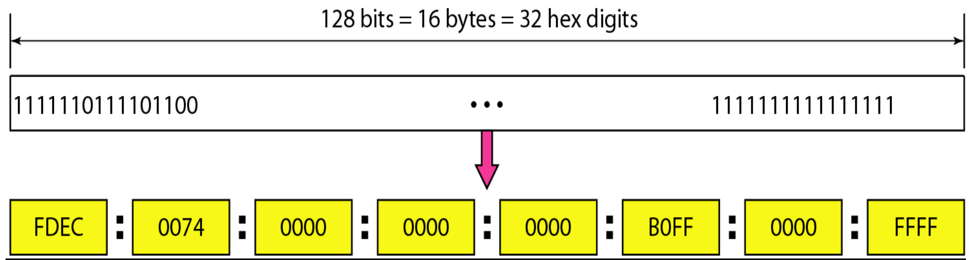

**Source address**. This 32-bit field defines the IPv4 address of the source. This field must remain unchanged during the time the IPv4 datagram travels from the source host to the destination host.

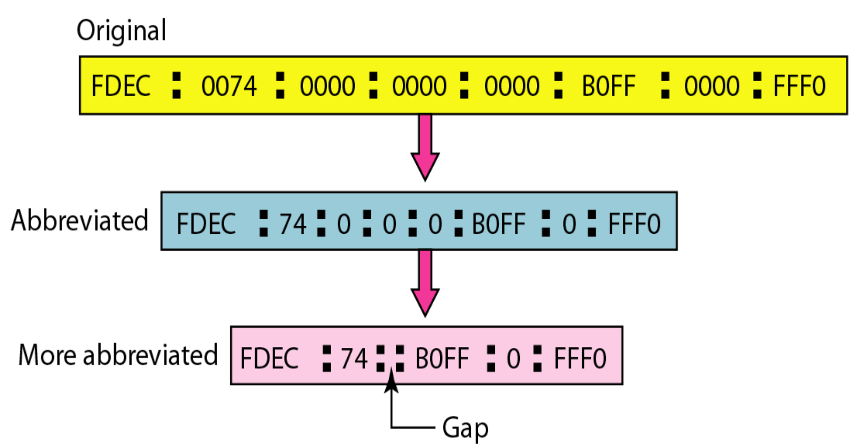
**Destination address**. This 32-bit field defines the IPv4 address of the destination.

## IPv6 Addresses

An IPv6 address consists of 16 bytes ,it is 128 bits long.

In this notation, 128 bits is divided into eight sections, each 2 bytes in length. Two bytes in hexadecimal notation requires four hexadecimal digits. Therefore, the address consists of 32 hexadecimal digits, with every four digits separated by a colon.



**Question:**
Expand the address **0:15::1:12:1213** to its original.

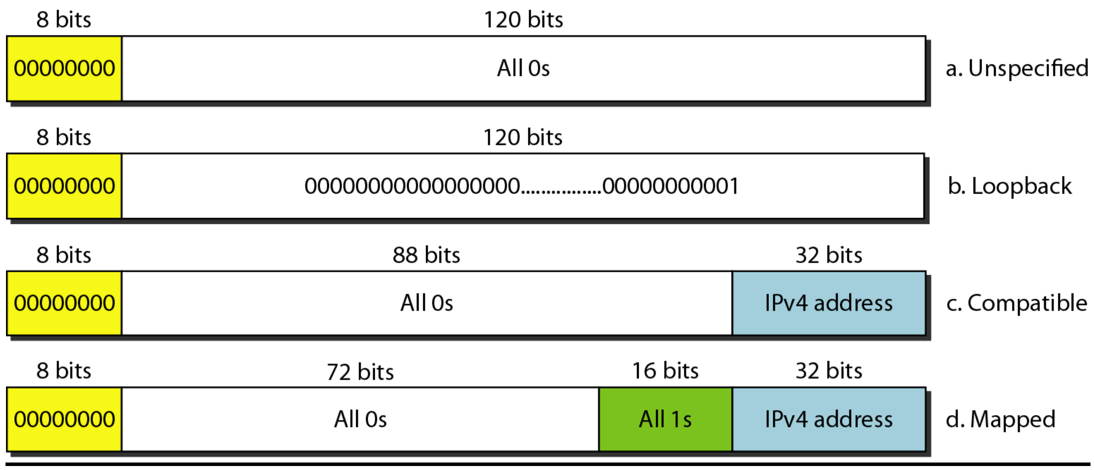
**Answer:** The original address is **0000:0015:0000:0000:0000:0001:0012:1213**

## Address Space

IPv6 has a much larger address space; $\mathbf{2^{128}}$ addresses are available. The designers of IPv6 divided the address into several categories. A few leftmost bits, called the *type prefix*, in each address define its category. The type prefix is variable in length, but it is designed such that no code is identical to the first part of any other code. In this way, there is no ambiguity;

| *Type Prefix* | *Type* | *Fraction* |
|---|---|---|
| 0000 0000 | Reserved | 1/256 |
| 0000 0001 | Unassigned | 1/256 |
| 0000 001 | ISO network addresses | 1/128 |
| 0000 010 | IPX (Novell) network addresses | 1/128 |
| 0000 011 | Unassigned | 1/128 |
| 0000 1 | Unassigned | 1/32 |
| 0001 | Reserved | 1/16 |
| 001 | Reserved | 1/8 |
| **010** | **Provider-based unicast addresses** | **1/8** |

**Reserverd Address**



## Advantages of IPv 6

**IPv6 has some advantages over IPv4 that can be summarized**as follows:

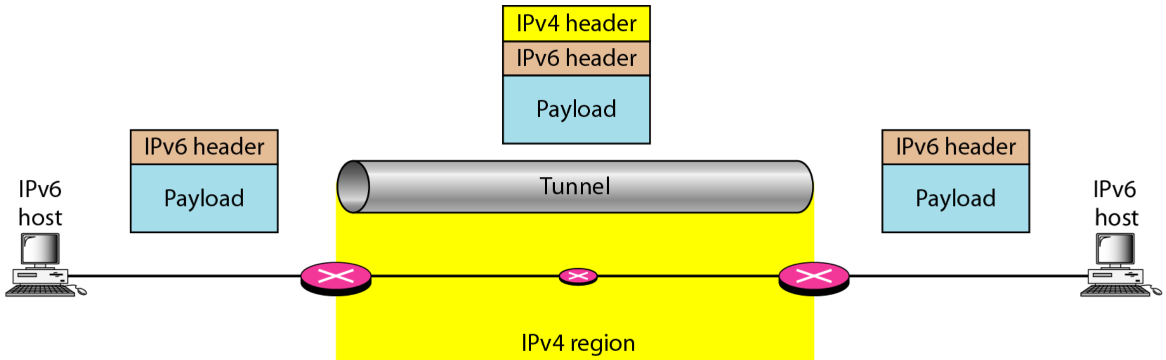
- Larger address space. An IPv6 address is 128 bits long, Compared with the 32-bit address of IPv4
- Better header format. IPv6 uses a new header format in which options are separated from the base header and inserted, when needed, between the base heade rand the upper-layer data. This simplifies and speeds up the routing process because most of the options do not need to be checked by routers.
- 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 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.
- Support for more security. The encryption and authentication options in IPv6 provide confidentiality and integrity of the packet.

### Comparison between IPv4 and IPv6 packet headers

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

## Tunneling

Tunnelingis a strategy used when two computers using IPv6 want to communicate with each other and the packet must pass through a region that uses IPv4. To pass through this region, the packet must have an IPv4 address. So the IPv6 packet is encapsulated in an IPv4 packet when it enters the region, and it leaves its capsule when it exits the region. It seems as if the IPv6 packet goes through a tunnel at one end and emerges at the other end. To make it clear that the IPv4 packet is carrying an IPv6 packet as data, the protocol value is set to 41.

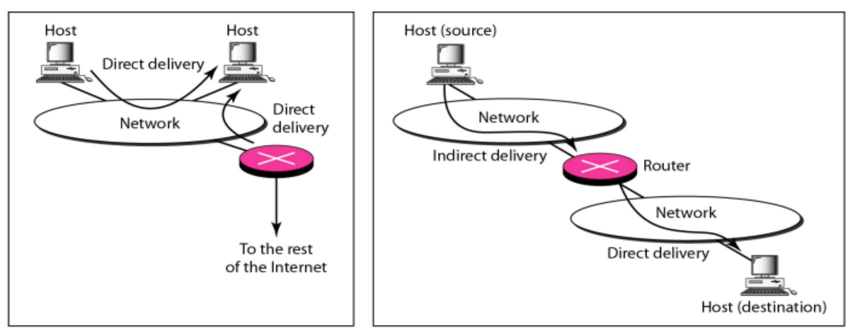



## Delivery

The network layer supervises the handling of the packets by the underlying physical networks. We define this handling as the delivery of a packet. It can be direct or indirect delivery.

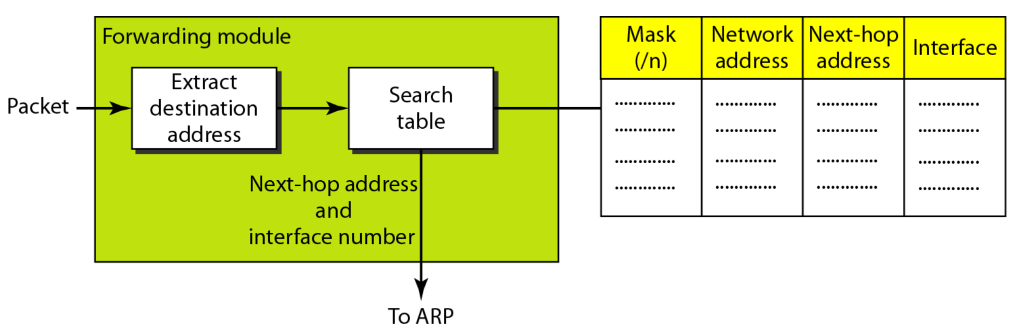
*Direct Delivery:* In a direct delivery, the final destination of the packet is a host connected to the same physical network as the deliverer.

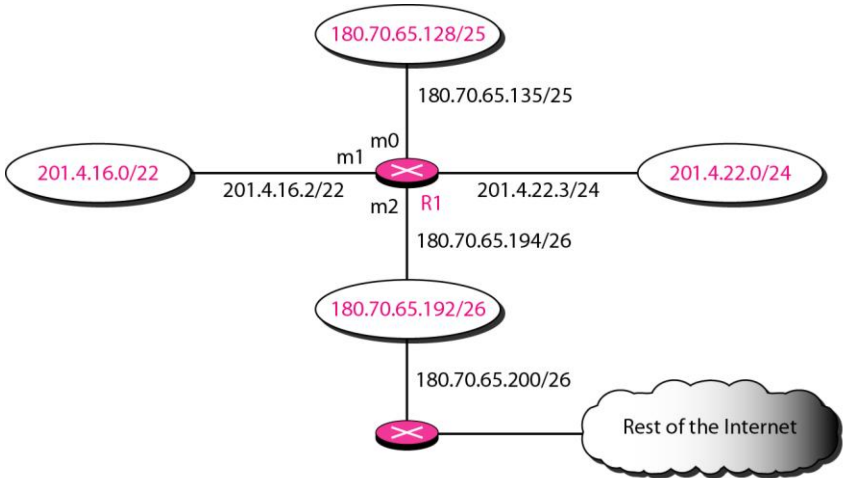
*Indirect Delivery:* If the destination host is not on the same network as the deliverer, the packet is delivered indirectly.




a. Direct delivery        b. Indirect and direct delivery

## In Classless Addressing we need atleast 4 column in the routing table






\

| *Mask* | *Network Address* | *Next Hop* | *Interface* |
|---|---|---|---|
| /26 | 180.70.65.192 | — | m2 |
| /25 | 180.70.65.128 | — | m0 |
| /24 | 201.4.22.0 | — | m3 |
| /22 | 201.4.16.0 | .... | m1 |
| Any | Any | 180.70.65.200 | m2 |