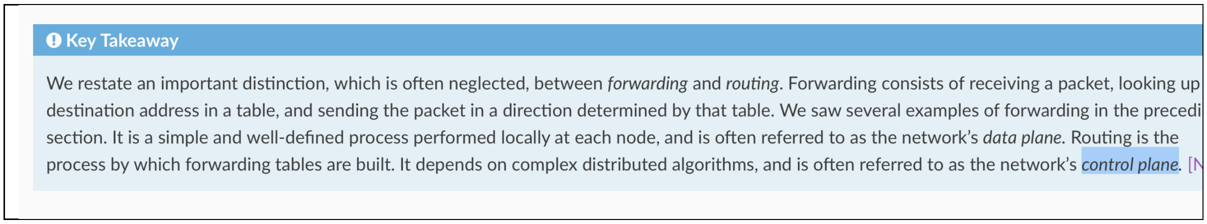
Class Notes

The Network Layer (Data Plane) (Chapter 4)

1. Difference between data plane and control plane:

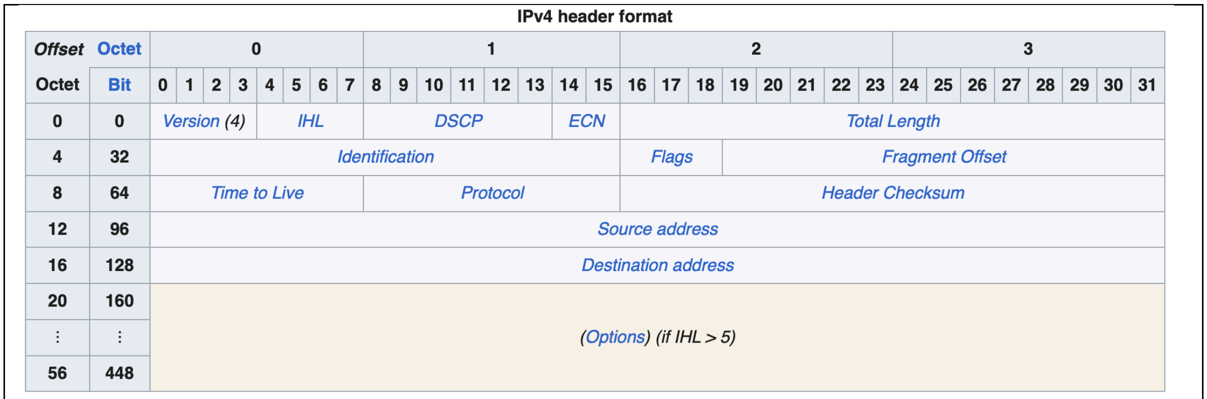


Key Takeaway

We restate an important distinction, which is often neglected, between *forwarding* and *routing*. Forwarding consists of receiving a packet, looking up destination address in a table, and sending the packet in a direction determined by that table. We saw several examples of forwarding in the precedi section. It is a simple and well-defined process performed locally at each node, and is often referred to as the network's *data plane*. Routing is the process by which forwarding tables are built. It depends on complex distributed algorithms, and is often referred to as the network's *control plane*. [N

Packet Structure of IPv4 Datagram

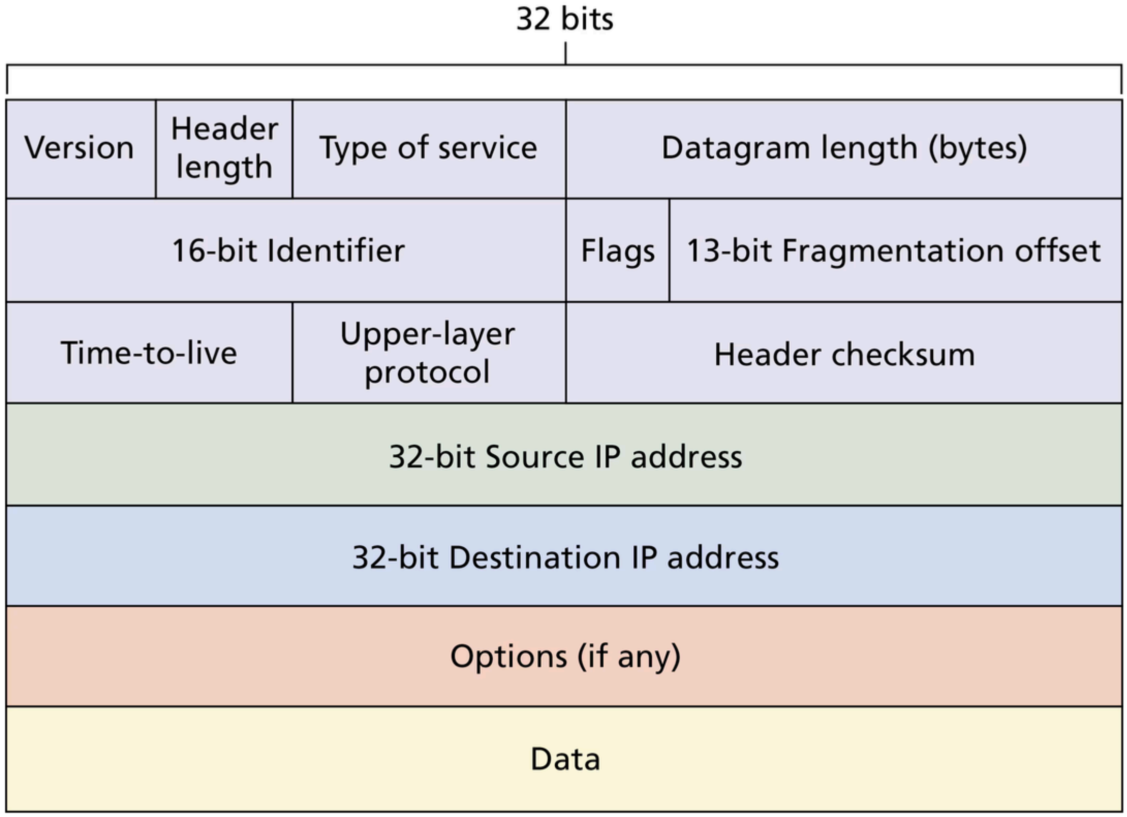
1. Following is the structure of IPv4 datagram.



IPv4 header format

Offset	Octet	0								1								2								3							
Octet	**Bit**	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
0	0	*Version (4)*				*IHL*				*DSCP*						*ECN*		*Total Length*															
4	32	*Identification*																*Flags*			*Fragment Offset*												
8	64	*Time to Live*								*Protocol*								*Header Checksum*															
12	96	*Source address*																															
16	128	*Destination address*																															
20	160	*(Options) (if IHL > 5)*																															
⋮	⋮																																
56	448																																

Source: Wikipedia

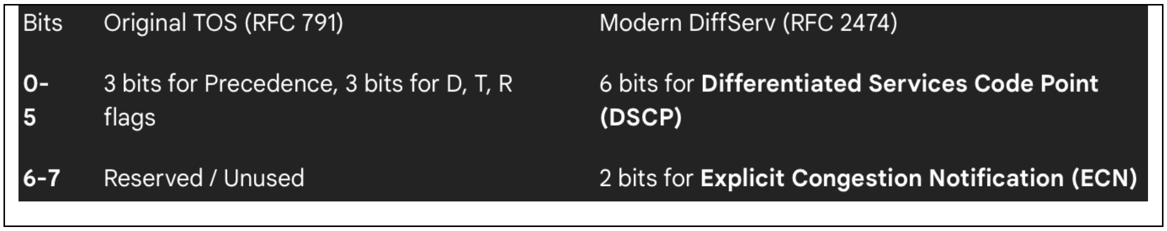


Source: Textbook

2. Version:
    a) First field of the packet so that a router could quickly decide is it IPv4 or IPv6 packet.
    b) Has 4 bits (0 to 15 values possible.). IP versions 0, 1, 2, and 3 were experimental during the days of Arpanet (Current Internet's ancestor). IP version 5 was another experimental version. Today IP versions 4 and 6 are active. Intention was that IPv6 will replace IPv4 but that didn't happen due to factors such as economic (ISPs and other stakeholders have long-term investments in iPv4 infrastructure and software), social (we can't have a flag day on the Internet as big as it is today where after a certain day and time everyone will shift from IPv4 to IPv6), and other reasons.
3. Protocol as demultiplexing key:
    a) At the network layer of the final destination host, network layer needs to decide to whom to hand over the payload of IP packet to. If protocol value is 6 that means data is a TCP segment, if protocol value is 17 that means IP payload is for UDP segment. If protocol value is 1, that means payload data is related to ICMP. See IANA website for a complete list of protocol numbers: https://www.iana.org/assignments/protocol-numbers/protocol-numbers.xhtml
4. Header Length (4 bits, Units = Words, 1 Word = 4 Bytes) and Datagram Length (Units = Bytes, Includes length of header + payload):
    a) Header of an IPv4 datagram is not of fixed size because IP datagram can have options fields. A typical IPv4 header size is of 20 bytes (no options).
    b) Datagram length – (header length*4) = payload size in Bytes

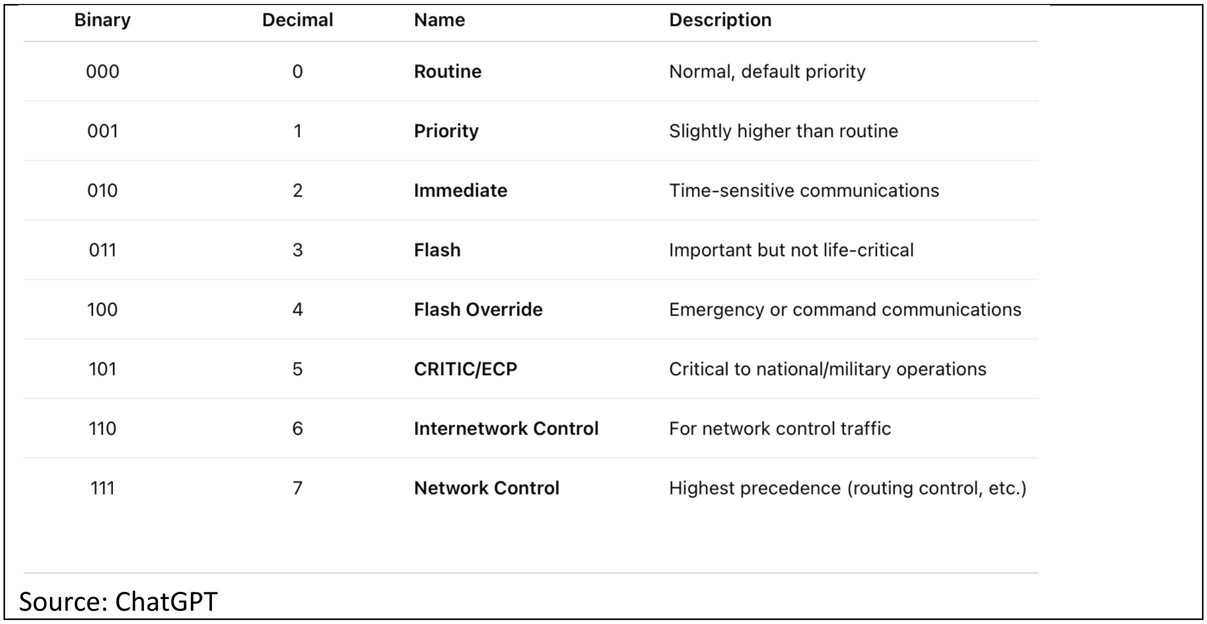
   c) Datagram length = 16-bit Theoretical max IP datagram = 2^16 Bytes in size
   d) Question: Why IPv4 header has fields for header length and full datagram length? TCP header only had one field for header length.
   e) IPv6 has fixed header length of size 40 Bytes.

5. ECN two bits
   a) We already discussed it in the context of TCP explicit congestion notification.
6. Type of service (8 bits)
   a) TOS: D = Low Delay, T = High Throughput, R = High Reliability



Bits	Original TOS (RFC 791)	Modern DiffServ (RFC 2474)
0-5	3 bits for Precedence, 3 bits for D, T, R flags	6 bits for **Differentiated Services Code Point (DSCP)**
6-7	Reserved / Unused	2 bits for **Explicit Congestion Notification (ECN)**

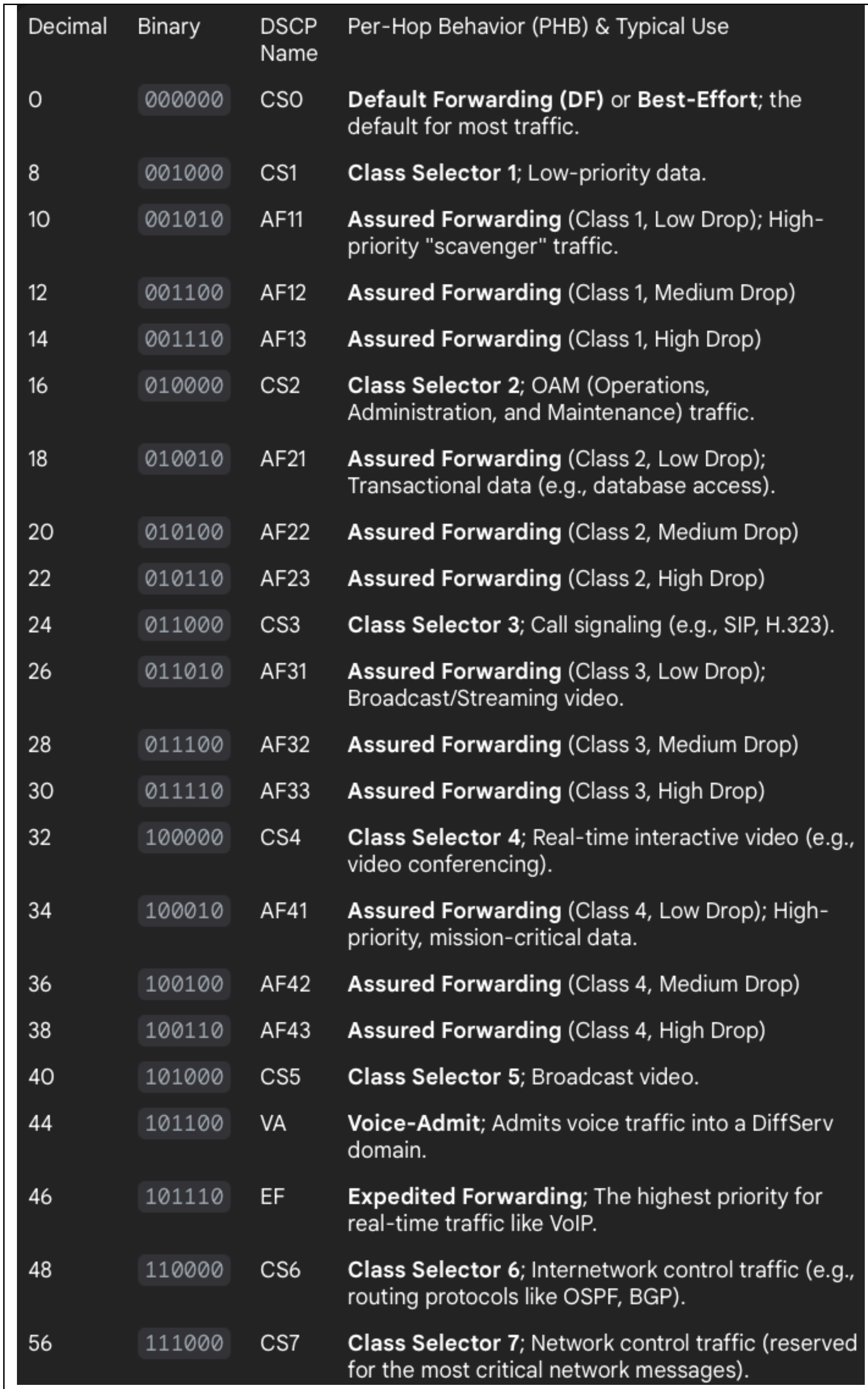
7. TOS precedence values:



Binary	Decimal	Name	Description
000	0	**Routine**	Normal, default priority
001	1	**Priority**	Slightly higher than routine
010	2	**Immediate**	Time-sensitive communications
011	3	**Flash**	Important but not life-critical
100	4	**Flash Override**	Emergency or command communications
101	5	**CRITIC/ECP**	Critical to national/military operations
110	6	**Internetwork Control**	For network control traffic
111	7	**Network Control**	Highest precedence (routing control, etc.)

Source: ChatGPT

8. DSCP values usually used:



Decimal	Binary	DSCP Name	Per-Hop Behavior (PHB) & Typical Use
0	`000000`	CS0	**Default Forwarding (DF)** or **Best-Effort**; the default for most traffic.
8	`001000`	CS1	**Class Selector 1**; Low-priority data.
10	`001010`	AF11	**Assured Forwarding** (Class 1, Low Drop); High-priority "scavenger" traffic.
12	`001100`	AF12	**Assured Forwarding** (Class 1, Medium Drop)
14	`001110`	AF13	**Assured Forwarding** (Class 1, High Drop)
16	`010000`	CS2	**Class Selector 2**; OAM (Operations, Administration, and Maintenance) traffic.
18	`010010`	AF21	**Assured Forwarding** (Class 2, Low Drop); Transactional data (e.g., database access).
20	`010100`	AF22	**Assured Forwarding** (Class 2, Medium Drop)
22	`010110`	AF23	**Assured Forwarding** (Class 2, High Drop)
24	`011000`	CS3	**Class Selector 3**; Call signaling (e.g., SIP, H.323).
26	`011010`	AF31	**Assured Forwarding** (Class 3, Low Drop); Broadcast/Streaming video.
28	`011100`	AF32	**Assured Forwarding** (Class 3, Medium Drop)
30	`011110`	AF33	**Assured Forwarding** (Class 3, High Drop)
32	`100000`	CS4	**Class Selector 4**; Real-time interactive video (e.g., video conferencing).
34	`100010`	AF41	**Assured Forwarding** (Class 4, Low Drop); High-priority, mission-critical data.
36	`100100`	AF42	**Assured Forwarding** (Class 4, Medium Drop)
38	`100110`	AF43	**Assured Forwarding** (Class 4, High Drop)
40	`101000`	CS5	**Class Selector 5**; Broadcast video.
44	`101100`	VA	**Voice-Admit**; Admits voice traffic into a DiffServ domain.
46	`101110`	EF	**Expedited Forwarding**; The highest priority for real-time traffic like VoIP.
48	`110000`	CS6	**Class Selector 6**; Internetwork control traffic (e.g., routing protocols like OSPF, BGP).
56	`111000`	CS7	**Class Selector 7**; Network control traffic (reserved for the most critical network messages).

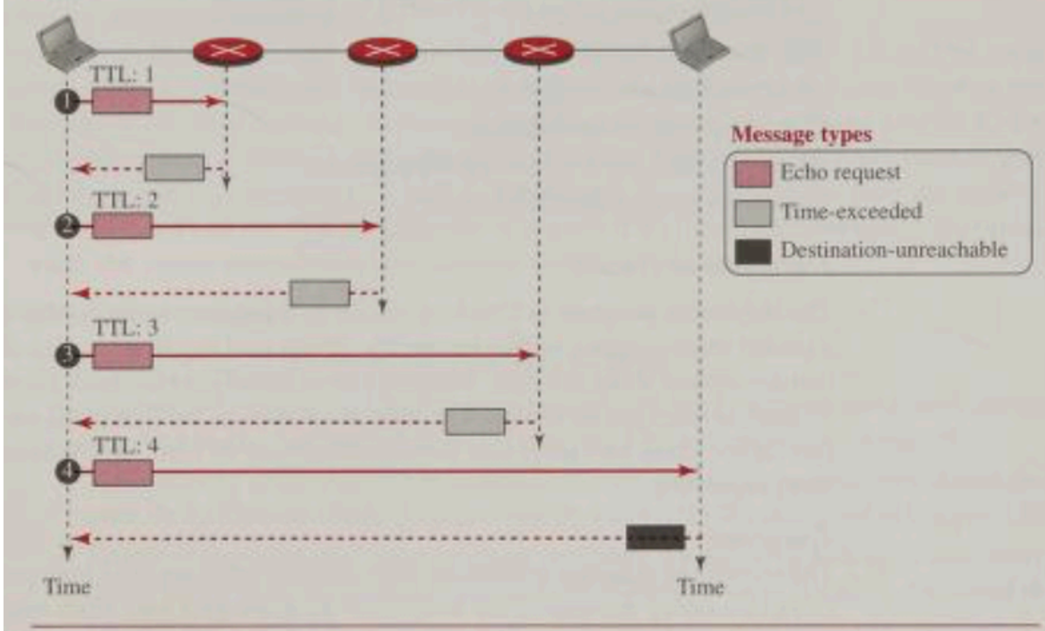
Source: Google Gemini

9. Header checksum (16 bits)

   a) Internet checksum on IPv4 header fields only. Each router needs to re-calculate it for verification and because each router changed another header field (TTL) and hence checksum needs to be recalculated.
   b) IPv6 header does not have checksum field

10. Time to live (8 bits)
    a) Reason: To guard against routing loops in the network topology. If such a loop exists, packets will loop forever on the Internet wasting resources.
    b) Initially the semantics of this field were that each router will decrement this time by the amount a packet waited inside a router. But later, for implementation efficiency, it was changed to **<u>hop count</u>**. Now each router decrements it by one and if its value becomes 0 after decrement, router drops such an IP packet and sends a ICMP message to inform, about this event to the source. Note that the network layer at the destination does not decrement TTL because it will not be forwarding that packet.
    c) When TTL changes in the IP header, router needs to recalculate the checksum.
    d) In IPv6, there is no checksum and one of the reasons is to avoid doing this work.
    e) Traceroute program uses TTL in a clever way to print network path taken by the packets.
        - Destination-unreachable means: port used in UDP packet for traceroute is not bound to any application at the destination.

Figure 19.10 *Use of ICMPv4 in traceroute*



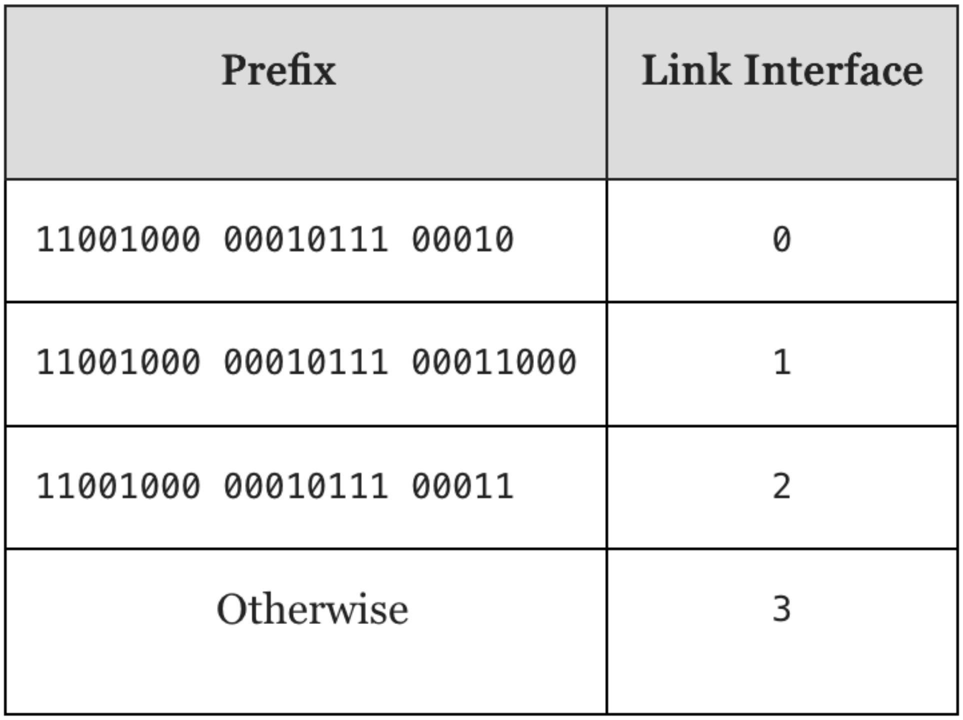
Source: Data Communications and Networking, 5th edition by Behrouz A Forouzan

11. Source and destination addresses
    a) In IPv4, both source and destination addresses are 32 bit long. Consider them as names or unique identification of hosts on the network. 2^32 possible addresses **<u>does not have a flat hierarchy</u>**. Rather there is a structure to it that we will explore later.

b) IPv4 are usually written as dotted quads such as: 10.1.2.3 where each number has 8 bits and hence can have value from 0 to 255

c) IP addresses are assigned to network interfaces. IPs have network portion and the host portion to it. For example, 10.0.0.0/8 means that first 8 bits (from the left to right) are part of the network (or subnetwork) identifier. Remaining 24 bits are host identifiers.

d) Traditional routers use **longest prefix matching on the destination IP addresses** to index their forwarding tables. That means for an incoming IP packet, routers will pick its destination address and will find what entries match the destination address. If more than one entry match, they pick the one with the longest match. For example, if destination IP is: `11001000 00010111 00010110 10100001` then packet will be forwarded to interface 0. If destination IP was `11001000 00010111 00011000 10101010` it matches both second and third entry, but the second entry is the longest one. So that will be used.

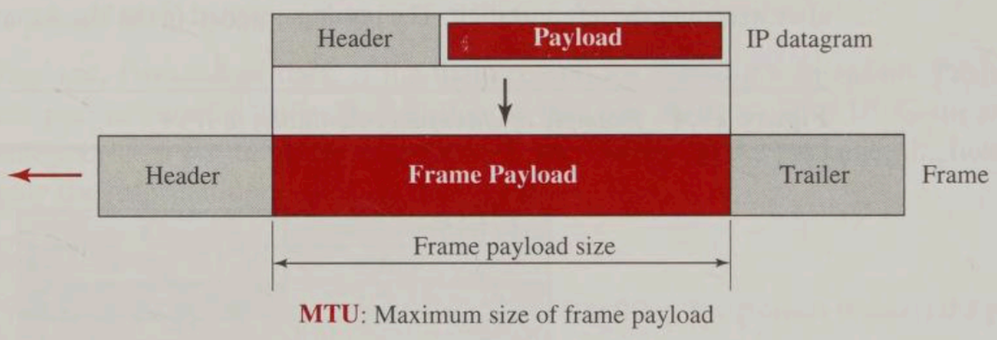


Prefix	Link Interface
11001000 00010111 00010	0
11001000 00010111 00011000	1
11001000 00010111 00011	2
Otherwise	3

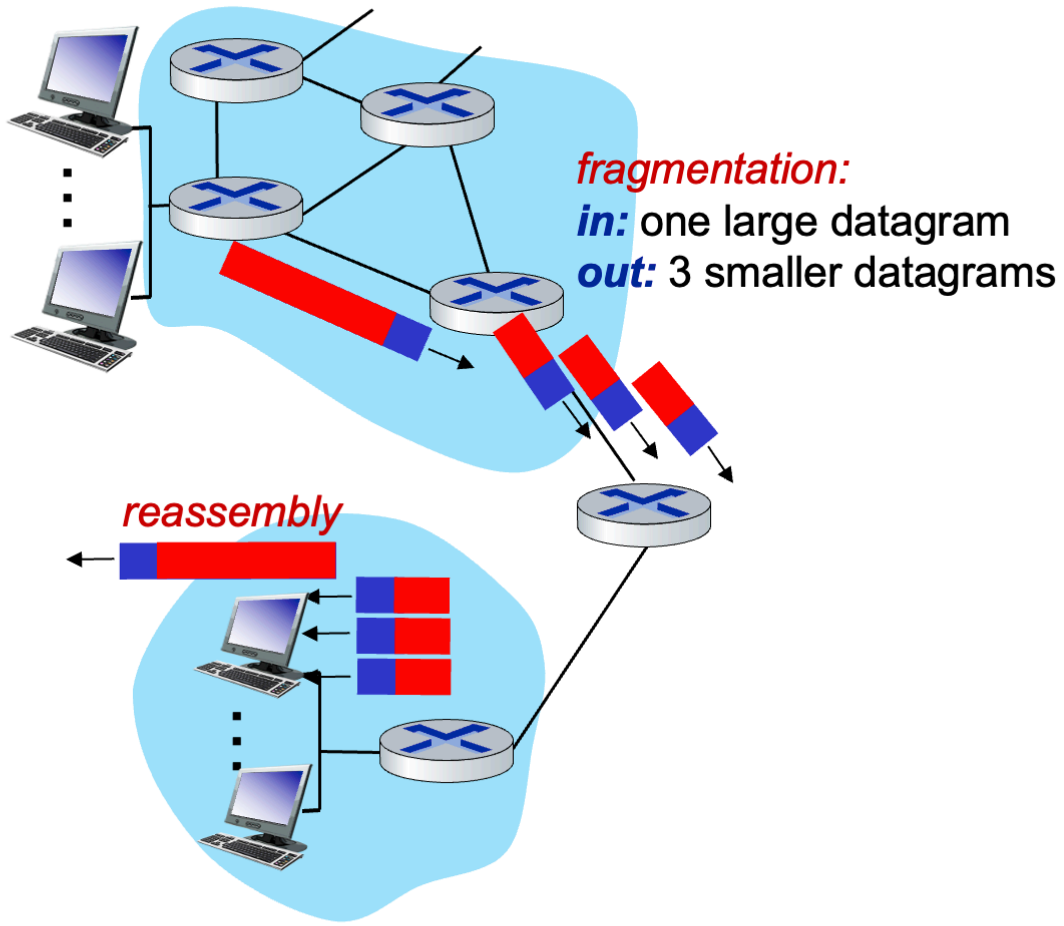
Source: Textbook

12. Options field:
    a) To reduce the size of the core IPv4 header, optional options facility is there.
    b) Usually not used over the Internet because some firewalls can drop such IP packets.
    c) Those are optional.
13. IPv4 Packet Fragmentation & Reassembly (Identifier, flags, and fragment offset):
    a) Identifier = 16 bits long
    b) Flags = 3 bits (Reserved Bit (should be 0), Do Not Fragment (DF), More Fragments (MF)
        - If DF = 1 => Do not fragment this IP packet
        - If MF = 1 => More sub-packets (fragments) are coming

Figure 19.5 *Maximum transfer unit (MTU)*



Source: Data Communications and Networking, 5th edition by Behrouz A Forouzan

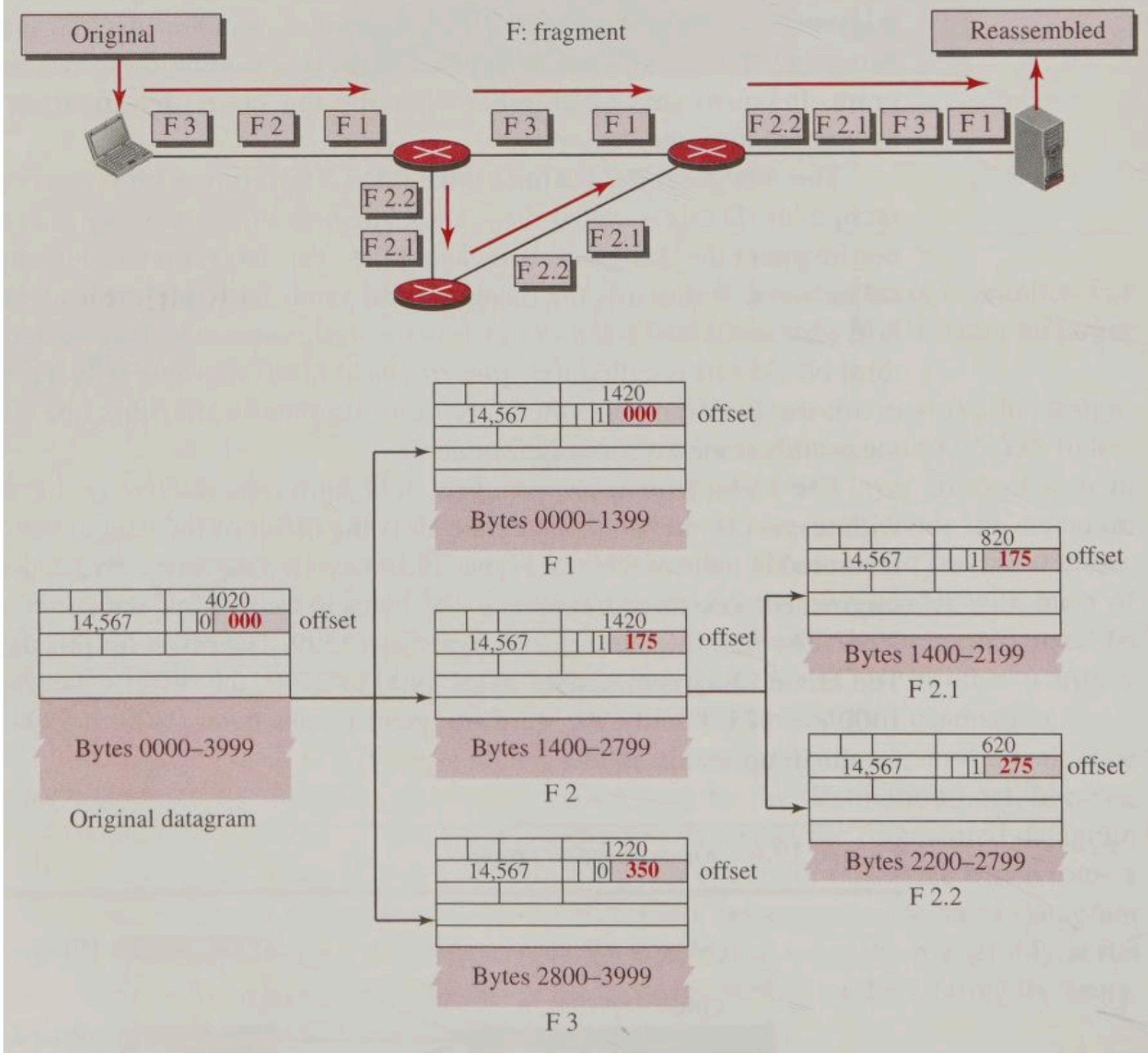


fragmentation:
in: one large datagram
out: 3 smaller datagrams

reassembly

Source: Textbook Slides

Figure 19.7 *Detailed fragmentation example*



Source: Data Communications and Networking, 5th edition by Behrouz A Forouzan

Towards the end of the textbook slides for chapter 4, there is another example for fragmentation and reassembly that you should practice.

Caution about fragmented payload size and offset number

MTU in Bytes – 20 Bytes (IPv4 headear) = X
X must be divisible by 8
If X is not divisible by 8, we need to use the largest interger small than X which is also divisible by 8.
See an example below:
Reason: The offset field measure blocks of 8 Bytes.

Example Walkthrough

Let's say a router needs to send a large packet through a link with an **MTU of 1499 bytes**.

1. **Calculate Maximum Payload:**
   - The maximum total size is 1499 bytes.
   - Subtracting the 20-byte IP header gives a theoretical max payload of bytes.
2. **Check for Divisibility:**
   - The router checks if 1479 is divisible by 8.
   - . It's not.
3. **Adjust the Payload Size:**
   - The router must find the largest number less than or equal to 1479 that is a multiple of 8.
   - It does this by taking the integer part of the division (184) and multiplying it by 8: bytes.
4. **Create the Fragment:**
   - The router will create a fragment with a **1472-byte payload**.
   - The total size of this fragment will be bytes, which is safely under the 1499-byte MTU.

The next fragment will then start at an offset of , perfectly aligning with the 8-byte rule. A few bytes of potential payload space are sacrificed in each fragment to maintain the protocol's integrity, ensuring the destination can reassemble the pieces correctly

Example source: Google Gemini

How IPs are assigned

IANA is actually the central and primary entity involved in IP address allocation.

The **Internet Corporation for Assigned Names and Numbers (ICANN)** is responsible for operating IANA. So, while you hear about ICANN, the specific function of allocating IP address blocks to registries is performed by IANA.

Here's the hierarchy and the reasoning behind this multi-layered system.

The IP Address Allocation Hierarchy

The distribution of IP addresses works like a pyramid, flowing from a single global source down to individual users.

1. **IANA (Internet Assigned Numbers Authority):** At the absolute top is IANA. It manages the entire global pool of unallocated IP addresses (both IPv4 and IPv6).[3] Think of it as the central bank for IP addresses.

2. Regional Internet Registries (RIRs): IANA does not give addresses to countries or ISPs directly. Instead, it allocates large blocks of addresses to five massive, non-profit RIRs. Each RIR is responsible for a specific geographic region of the world.4

   *

   - **AFRINIC:** Africa
   - **APNIC:** Asia/Pacific
   - **ARIN:** Canada, United States, and some Caribbean islands
   - **LACNIC:** Latin America and the Caribbean
   - **RIPE NCC:** Europe, the Middle East, and Central Asia
3. **Local Internet Registries (LIRs) & ISPs:** The RIRs then distribute smaller blocks of addresses to LIRs within their regions.[10] These are typically Internet Service Providers (ISPs), large enterprises, or academic institutions.
4. **End Users:** Finally, the ISPs assign individual IP addresses to their customers, like you and me.

Why Use This Hierarchical System?

So, why doesn't IANA just hand out addresses to anyone who asks? The system is designed this way for crucial reasons of scalability, efficiency, and policy.

1. Scalability and Administrative Efficiency

It would be a logistical nightmare for one central organization (IANA) to handle millions of requests from every ISP and company around the globe. Delegating this responsibility to regional bodies makes the entire system manageable. The RIRs handle the administrative overhead of verification, justification, and distribution for their specific regions.[11]

2. Regional Policy and Needs

Different parts of the world have vastly different internet landscapes, growth rates, and needs. The RIR model allows for the development of regional policies that are tailored to local conditions.[12] The internet community in each region collectively decides on the rules for how IP addresses are distributed and managed by their RIR. This ensures fairness and responsiveness to local needs.

3. Efficient Routing (Route Aggregation)

This is a critical technical reason. By allocating large, contiguous blocks of IP addresses to a single region, this system helps keep the global internet routing tables efficient. Routers on the internet's backbone don't need to know the specific path to every single IP address. Instead, they just need to know, for example, that a huge block of addresses "lives" in the Asia-Pacific region and can be reached via the

APNIC network infrastructure. This process, called **route aggregation** or **supernetting**, is vital for keeping the internet fast and stable.[13]

Source: Google Gemini