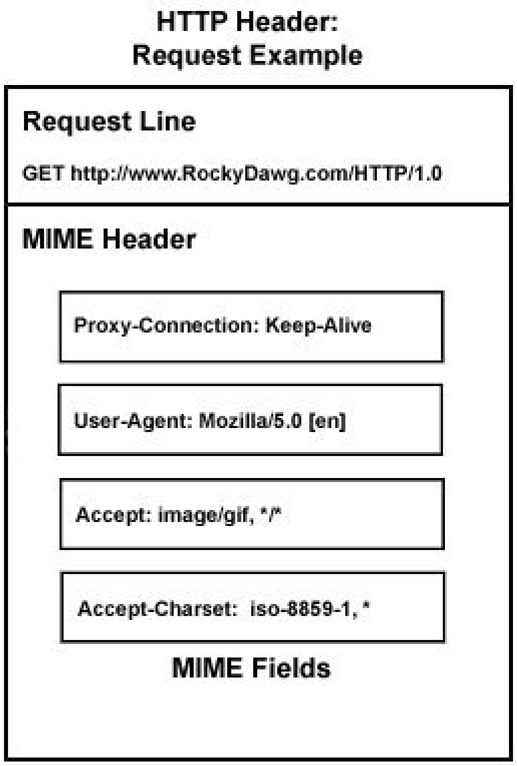
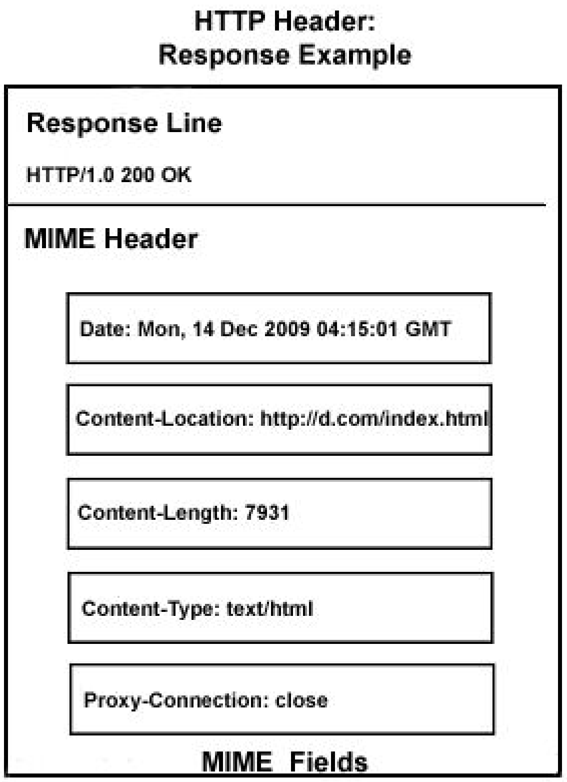
Structural Design and Implementation details

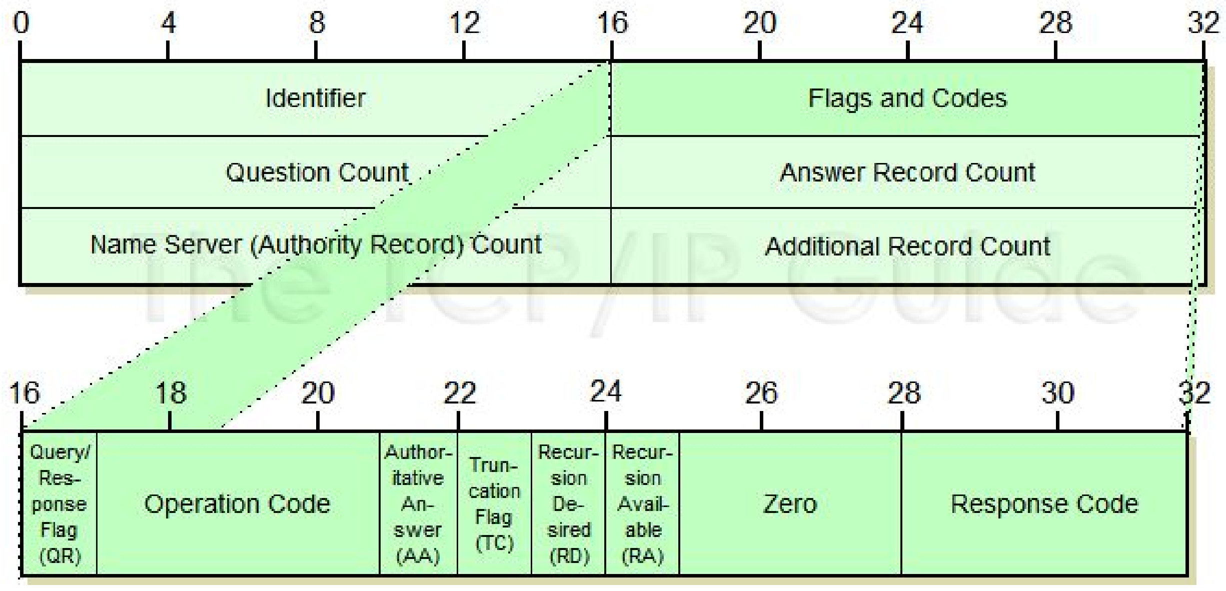
Application layer protocols:

HTTP


HTTP Header:
Request Example
Request Line
GET http://www.RockyDawg.com/HTTP/1.0
MIME Header
Proxy-Connection: Keep-Alive
User-Agent: Mozilla/5.0 [en]
Accept: image/gif, */*
Accept-Charset: iso-8859-1, *
MIME Fields



DNS



To parse DNS packet we have struct named "header" which contains all the above shown 6 fields.
The Flags and codes field is further enumerated into several parts which signifies many flags like Operation code, Autorititative answer , truncation flag, recursion denied , recursion available and Response code as shown above.
After this , we have the Questions i.e the actual website names which are to be converted to IP addresses.
In query message, we have fields till questions only, but in the case of response messages we have Answers field and Authoritative records also.
Answers field contains the IP addresses of the asked websites and Authoritative records contains the name of the Name servers which tells the IP address of the corresponding website.

DHCP

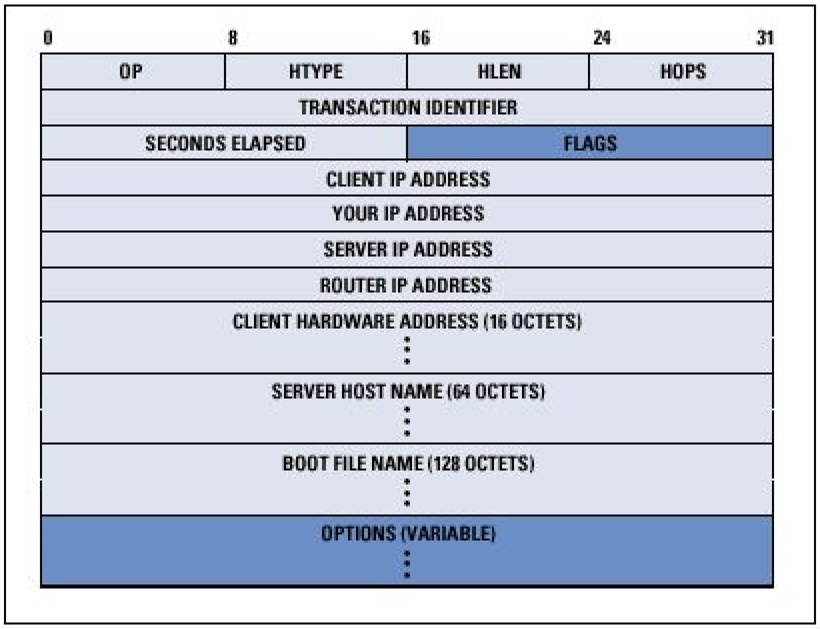
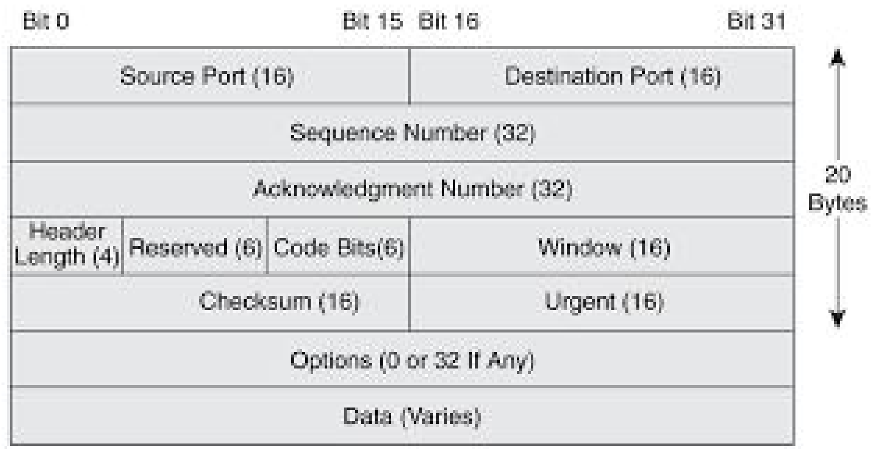


Figure 2: DHCP Message Format

Since there is no struct or header information stored in any header file we have to parse the DHCP message according to the above sown fields.

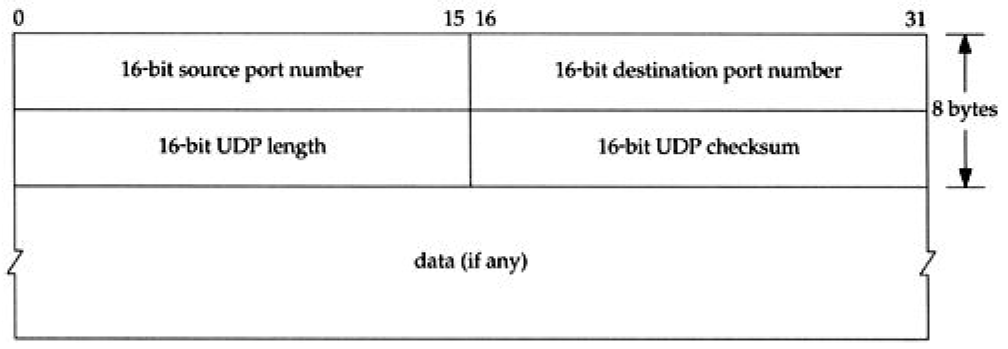
Transport layer protocols:

TCP



We have a struct named “tcphdr” in the <netinet/tcp.h> file which contains all the above shown fields.
The minimum size header is 5 words and the maximum is 15 words thus giving the minimum size of **20 bytes** and maximum of **60 bytes**, allowing for up to **40 bytes** of options in the header.
We use the port number to know about the Application layer protocol used.
Ex: 80 for HTTP, 53 for DNS, 67/68 for DHCP.

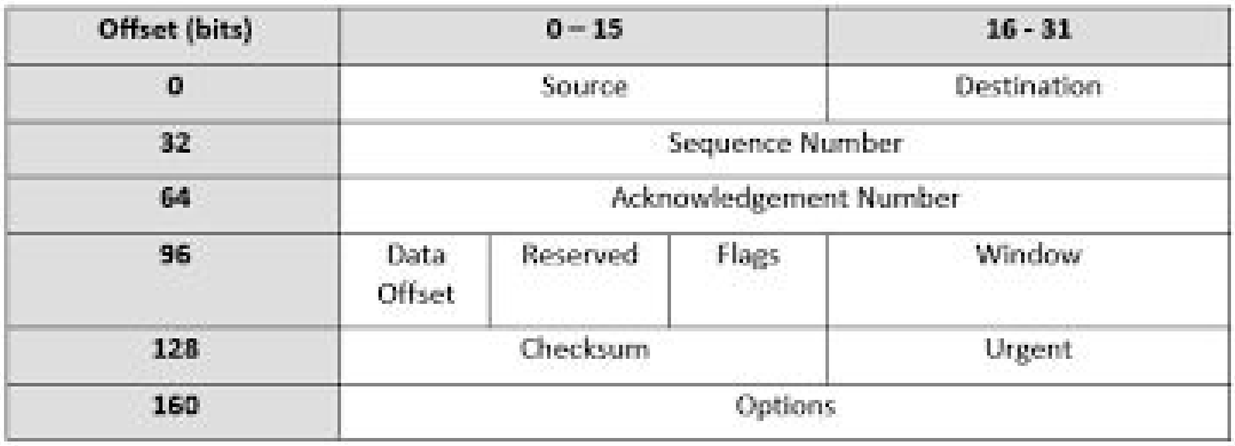
UDP



We have a struct named “udphdr” in the <netinet/udp.h> file which contains all the above shown fields.
The size of an IPv4 header is at least **20 bytes**, the size of an IPv6 header at least **40 bytes**. The payload of an IP packet is typically a TCP segment or a UDP datagram. A UDP datagram consists of a UDP header and the transported data. The size of a UDP header is **8 bytes**.
We use the port number to know about the Application layer protocol used.
Ex: 80 for HTTP, 53 for DNS, 67/68 for DHCP.

DCCP

We have a struct named “dccp_hdr” in the <linux/dccp.h> file which contains all the below shown fields.
The standard size of a DCCP packet header is usually **12 bytes**.



Offset (bits)	0 – 15			16 - 31
0	Source			Destination
32	Sequence Number			
64	Acknowledgement Number			
96	Data Offset	Reserved	Flags	Window
128	Checksum			Urgent
160	Options			

Network layer protocols:

ICMP

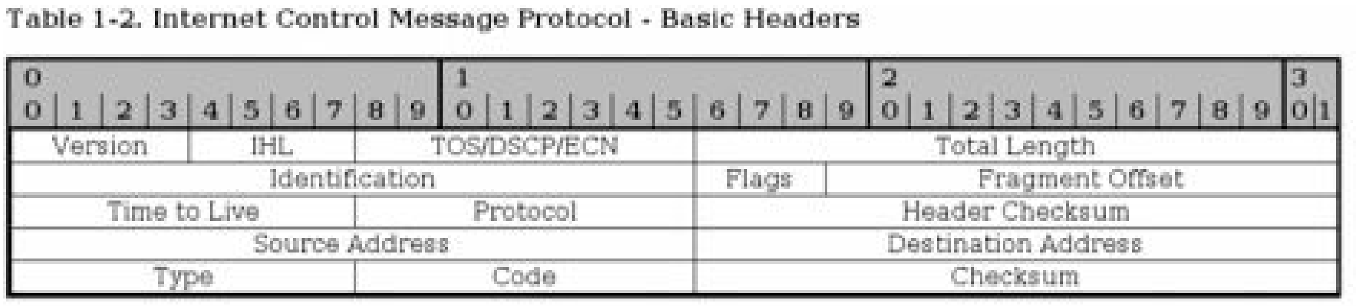


Table 1-2. Internet Control Message Protocol - Basic Headers

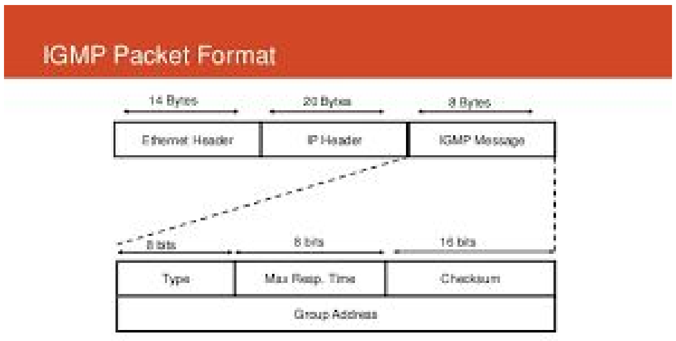
0 0 1 2 3	4 5 6 7	8 9 1 0 1 2 3 4 5	6 7 8	9 2 0 1 2 3 4 5 6 7 8 9 3 0 1
Version	IHL	TOS/DSCP/ECN	Total Length	
Identification			Flags	Fragment Offset
Time to Live		Protocol	Header Checksum	
Source Address			Destination Address	
Type		Code	Checksum	

We have a struct named “icmphdr” in the <netinet/ip_icmp.h> file which contains all the above shown fields.
The standard size of a ICMP packet header is usually **8 bytes** which includes Type , Code and Checksum.

There are various types like “Echo reply, Redirect, Echo , Router Advertisement etc” which are implemented in the code.

After the ICMP header information we have the data field.

IGMP

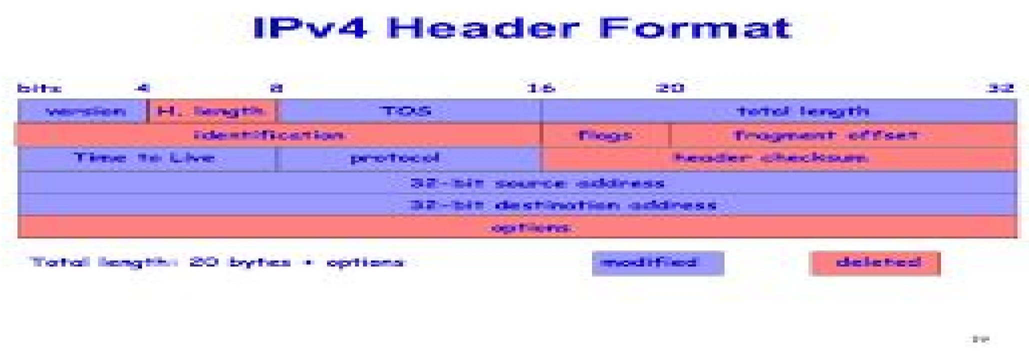


The Internet Group Management Protocol (**IGMP**) is a communications protocol used by hosts and adjacent routers on IPv4 networks to establish multicast group memberships.
There are two versions of IGMP, default version 2 and IGMPv3.
We have implemented both of them case by case.

All the group addresses and number of sources data are present directly in the struct named "igmpv3" which is present in the file <netinet/igmp.h>.

Link layer protocols:

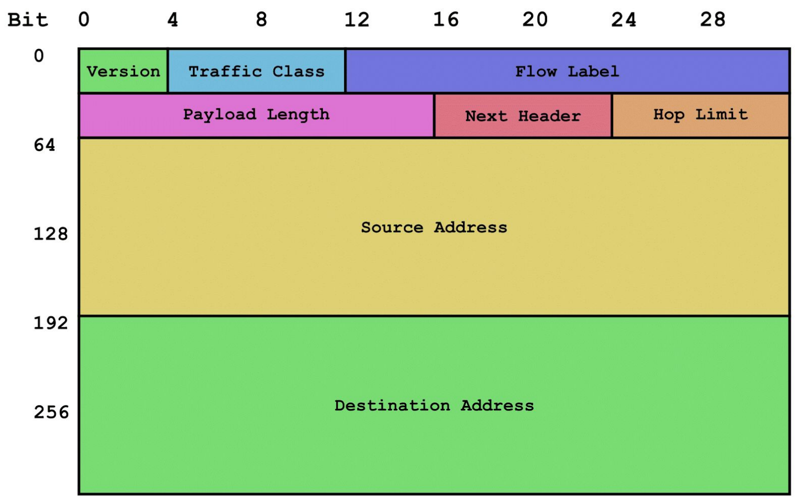
IPv4



We have a struct named "iphdr" in the <netinet/ip.h> file which contains all the above shown fields.
Options field is present in the header iff the IHL length is greater than 5 bytes.
The protocol value gives us the information about the type of transport layer protocol.
Ex: 6 for TCP, 17 for UDP, 1 for ICMP etc.

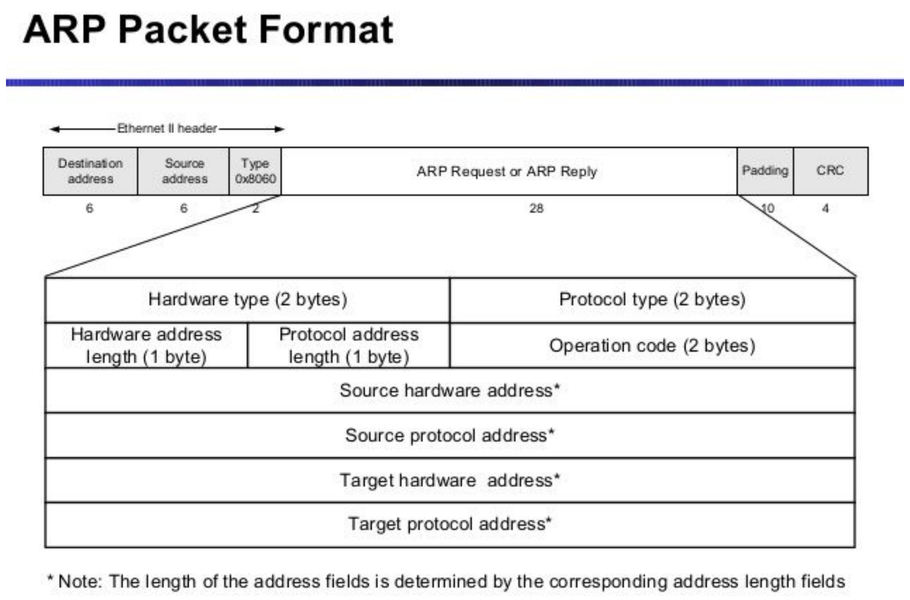
IPv6

The size of IPv6 header is much bigger than that of IPv4 header, because of IPv6 address size.
In the case of IPv6 , the source and destination IP addresses are 128 bit binary number.
In IPv6 we have extension header which is similar to the space field in IPv4 header.
TTL in IPv4 is renamed as Hop limit in IPv6.



ARP

Using ether_arp structure which contains all the fields which are shown in the format below we can parse the ARP packet.
This contains the Source MAC address , Dest. MAC address etc.


ARP Packet Format
Ethernet II header
Destination address
Source address
Type 0x8060
ARP Request or ARP Reply
Padding
CRC
6
6
2
28
10
4
Hardware type (2 bytes)
Protocol type (2 bytes)
Hardware address length (1 byte)
Protocol address length (1 byte)
Operation code (2 bytes)
Source hardware address*
Source protocol address*
Target hardware address*
Target protocol address*
* Note: The length of the address fields is determined by the corresponding address length fields