Ethernet II frame (rfc894):

destination address 6 bytes	source address 6 bytes	type 2	*data* *46 – 1500 bytes*	CRC 4 bytes

type value	data content
≤ `0x05dc`	(IEEE 802.3 frame)
`0x0800`	IP datagram
`0x0806`	ARP request/reply
`0x8025`	RARP request/reply

PPP frame (rfc1661/1662):

flag `0x7e` 1 byte	addr `0xff` 1 byte	control `0x03` 1 byte	protocol 2 byte	*data* *up to 1500 bytes*	CRC 2 bytes	flag `0x7e` 1 byte

protocol value	data content
`0x0021`	ip datagram
`0xc021`	link control data
`0x8021`	network control data (ip in this case)

ARP packet (28 bytes total, rfc826):

hard- ware 2 bytes	protocol type 2 bytes	hard size 1	prot size 1	oper- ation 2 byte	sender ethernet address 6 bytes	sender ip address 4 bytes	target ethernet address 6 bytes	target ip address 4 bytes

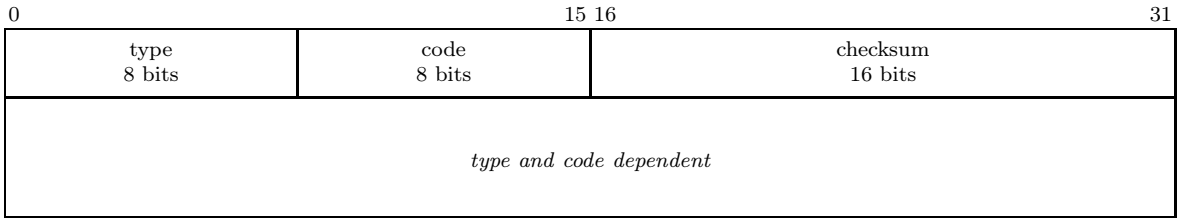
field name	comments
hardware type	1 = ethernet
protocol type	same value as type in ethernet header (0x0800 = ip)
hardware size	ethernet address size in bytes (6)
protocol size	ip address size in bytes (4)
operation	1 = arp req., 2 = arp reply, 3 = rarp req. and 4 = rarp reply

IP header (rfc791):

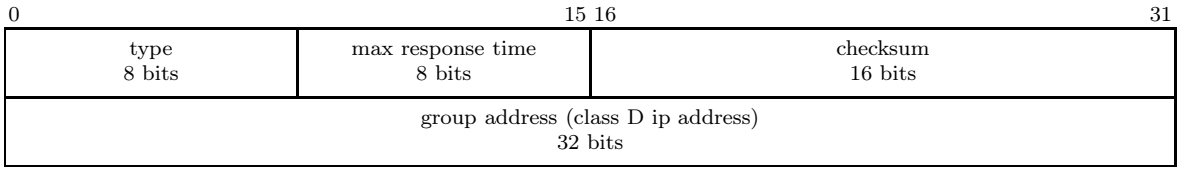
	0			15 16			31
1	version 4 bits	header length 4 bits	type of service 8 bits	total length (bytes) 16 bits			
2	identification 16 bits			0	D F	M F	fragment offset 13 bits
3	time to live 8 bits		protocol 8 bits	header checksum 16 bits			
4	source ip address 32 bits						
5	destination ip address 32 bits						
	options *0 to 10 * 32 bits*						
	data						

field name	comments
version	4 = current version (IPv4), 6 = new version (IPv6)
header length	number of 32 bit words in header (valid values 5 – 15)
type of service	(TOS) is often ignored, default settings as in rfc1700
word 2	is used for (de-)fragmentation (identification, Don't Fragment, More Fragments, fragment offset)
time to live	(TTL) decremented by 1 by each router, packet is discarded when value is zero
protocol	1 = ICMP, 2 = IGMP, 4 = IP tunneling (rfc2003), 6 = TCP, 8 = EGP, 17 = UDP (see rfc1700 under "PROTOCOL NUMBERS")

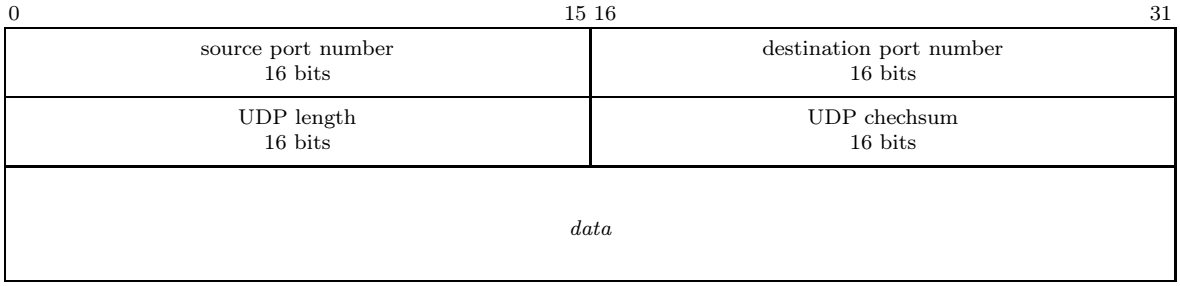
ICMP message (rfc792):



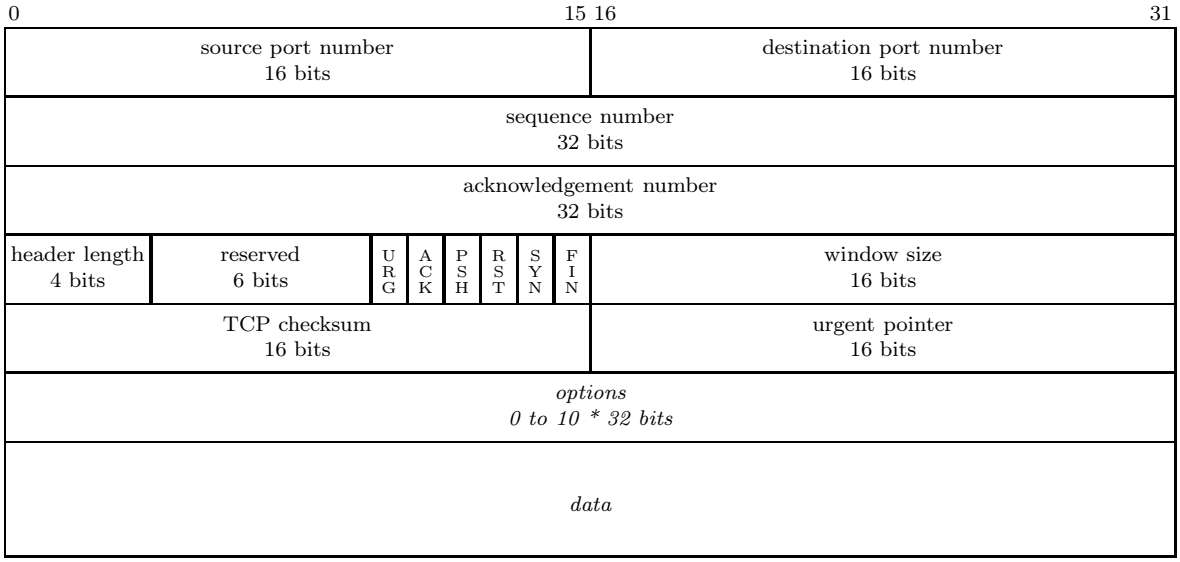
IGMP message (rfc2236):



UDP header (rfc768):



TCP header (rfc793, host requirements rfc):



field name	comments
sequence number	so receiver (of this packet) can deliver data (to the application) in the rigth order data flows in both directions, and both direction has its own sequence numbering
acknowledgement number	the sender (of this packet) has got all data with seq.no (of other direction) $<$ this ack.no
window size	how much data the sender will accept, also used for flow control