ECE 358: Computer Networks

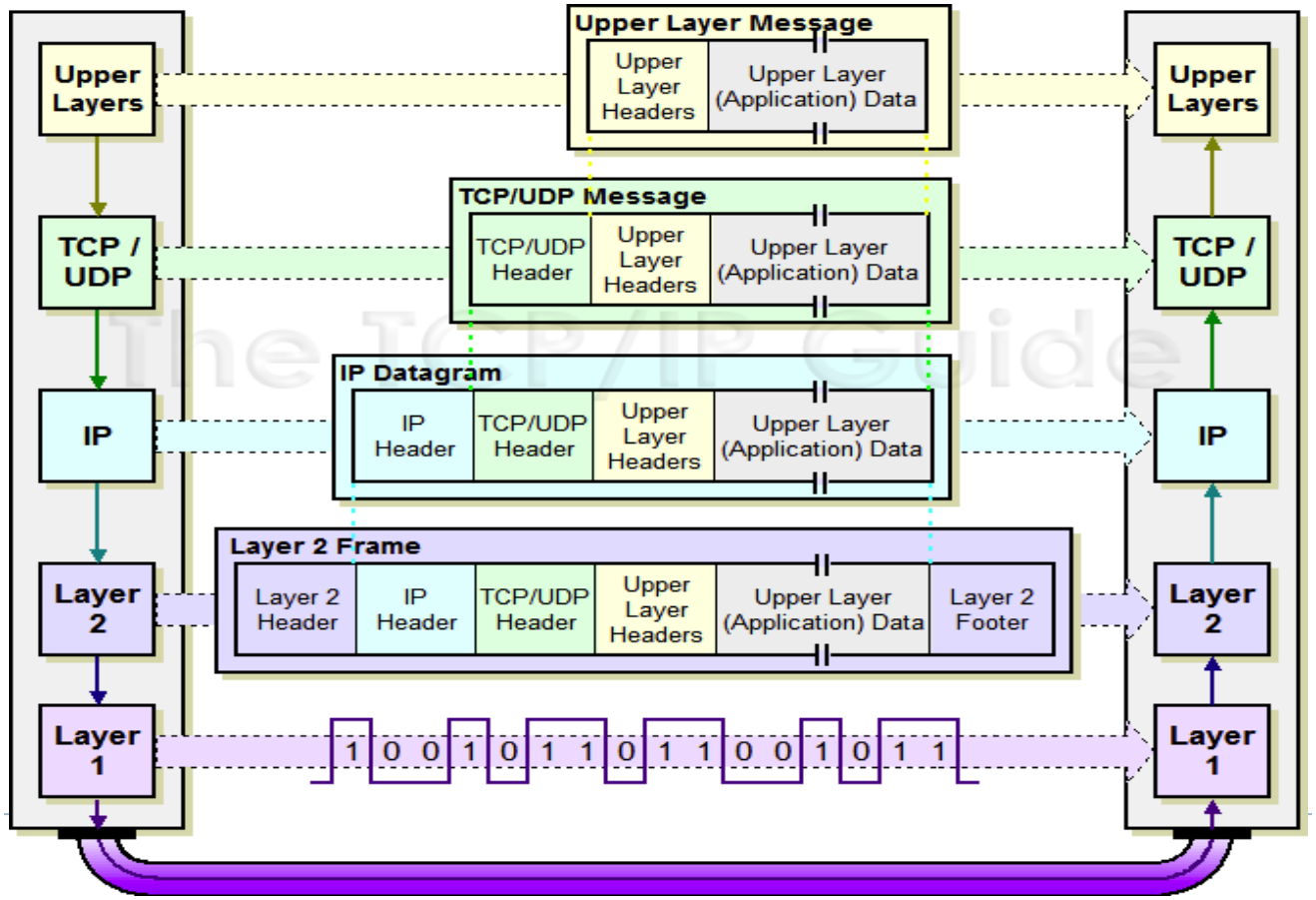
Fall 2023

LAB3: Encapsulation and Network Utilities

Originally created by: Dr. Albert Wasef
Presented by Hamidreza Nafissi
ECE 358

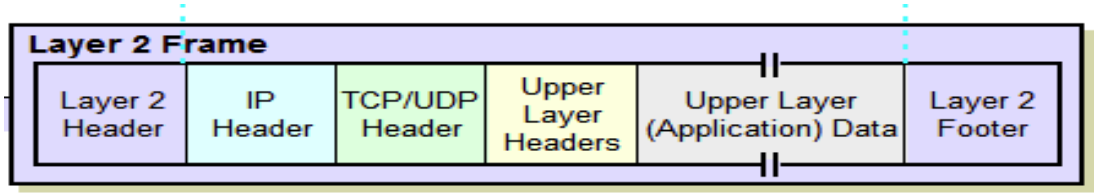
Overview

- Each layer in the OSI model adds its own header and/or footer.

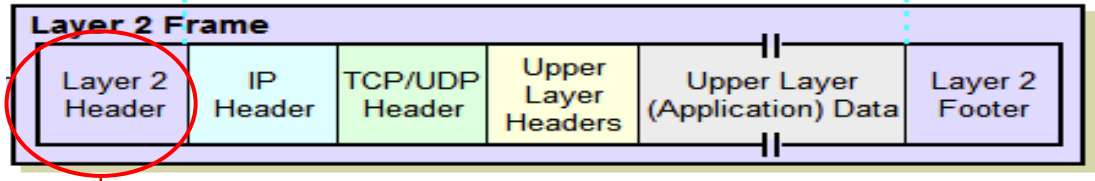


Overview

- In this lab, you are going to
  - Interpret all the encapsulated headers of captured Ethernet frames.
  - Use some network utilities to get an idea about the performance of the network.
- In this presentation we are going to focus only on the interpretation of the encapsulated headers of Ethernet frames.



Ethernet Frame

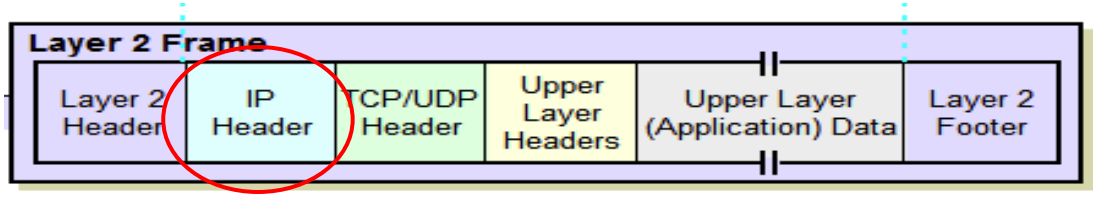


MAC header

Destination Address (6 Bytes)	Source Address (6 Bytes)	Type (2 Bytes)	Data	CRC

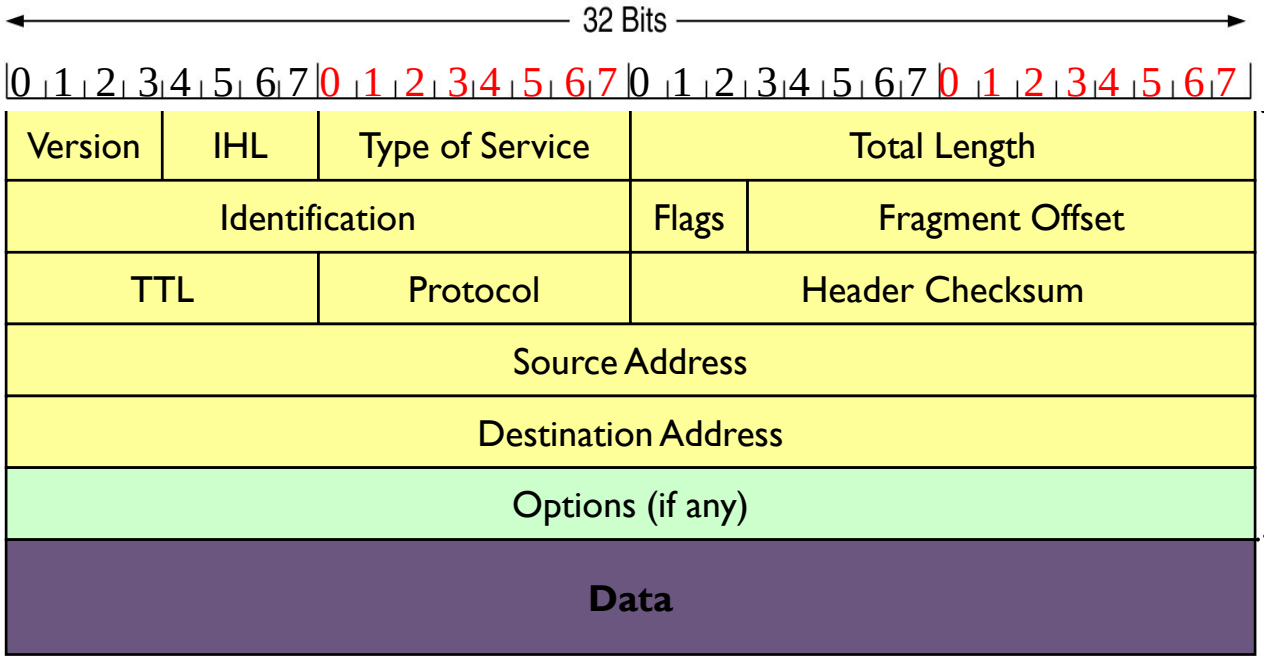
- The figure shows the format of Ethernet frames sent and received by the MAC layer.
- Destination address is either single address or group address (broadcast = 111…111)
- Type of payload
  - 0x0800 IPv4
  - 0x0806 ARP (Address Resolution Protocol)
  - 0x86dd IPv6

IPv4 Header



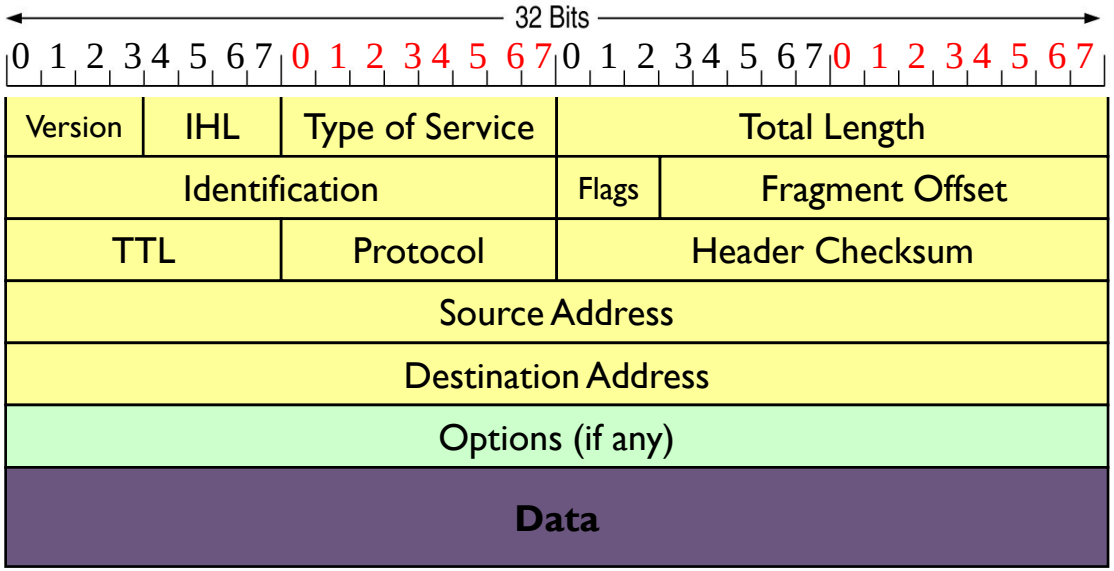
- A summary of the IP header is shown in the figure below.
- The number on the top is the bit number and each row is four bytes long (i.e. 32 bits).

IPv4 Packet Format



IPv4 Header

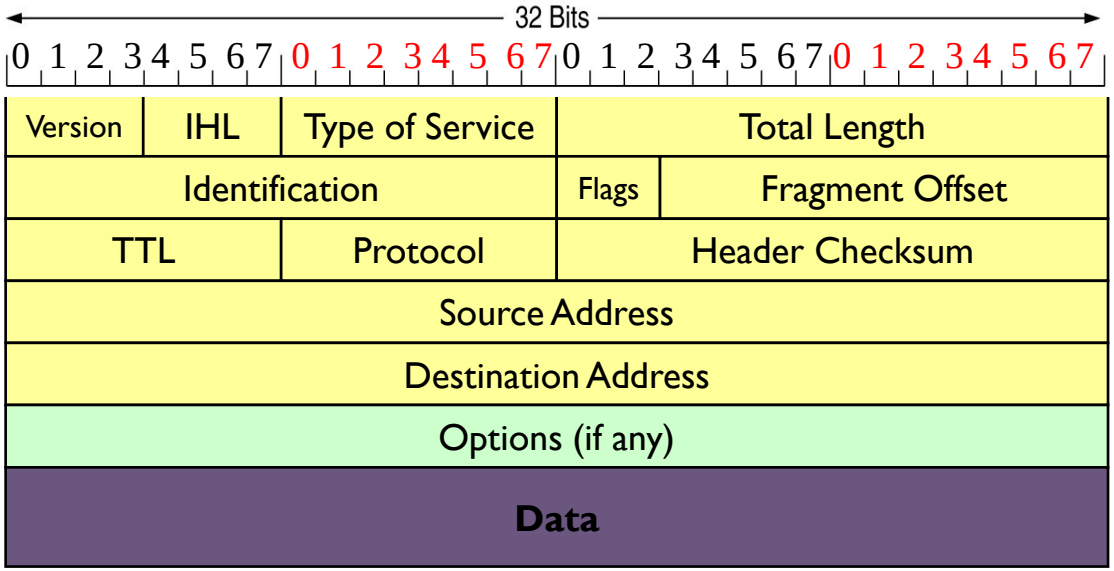
- Version (4 bits): IP Version
  - 4 for IPv4
  - 6 for IPv6
- IHL (4 bits): Internet Header Length
  - 32-bit words (typically 5)
- Type of Service (8 bits)
  - Priority information



  - Bits 0-2: Precedence (111 - Network Control 110 - Internetwork Control
    101 - CRITIC/ECP (Critical and Emergency Call Processing)
    100 - Flash Override 011 - Flash 010 – Immediate
    001 – Priority 000 – Routine)
  - Bit 3: 0 = Normal Delay, 1 = Low Delay.
  - Bits 4: 0 = Normal Throughput, 1 = High Throughput.
  - Bits 5: 0 = Normal Reliability, 1 = High Reliability.
  - Bit 6-7: Reserved for Future Use.
- Total Length (16 bits): Bytes (including header)

IPv4 Header

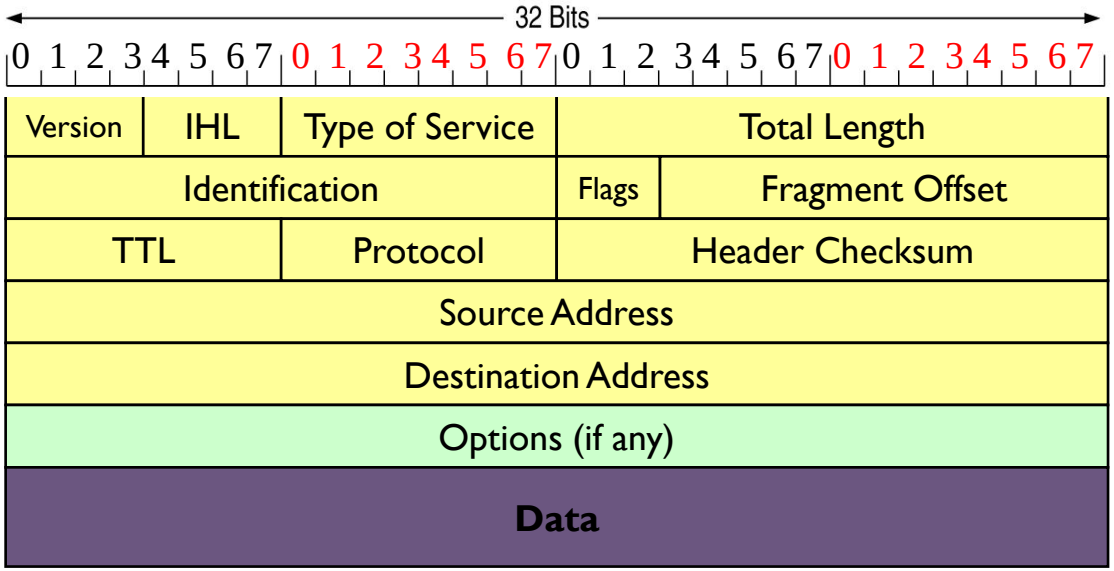
- Identification(16 bits): Identity assigned by the sender to aid in assembling the fragments in a datagram.

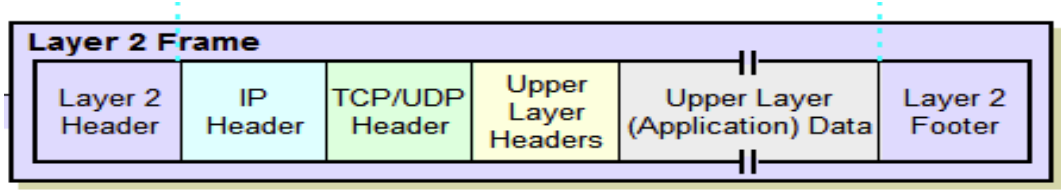


- Flags (3 bits):
  - Bit 0: reserved, must be zero
  - Bit 1: (DF) 0 = May Fragment, 1 = Don't Fragment.
  - Bit 2: (MF) 0 = Last Fragment, 1 = More Fragments.
-
- Fragment offset (13 bits): Used primarily for fragmentation. It indicates where in the datagram this fragment belongs.

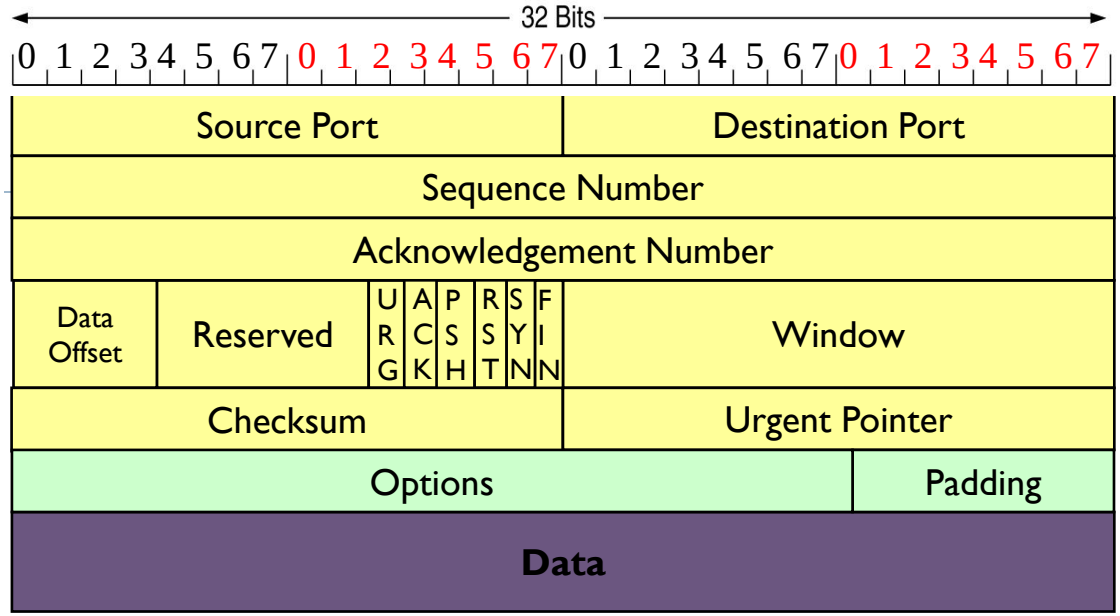
IPv4 Header

- Time to live (8 bits)
  - Must be decremented at each router
  - Packets with TTL=0 are thrown away
  - Ensure packets exit the network
- Protocol (8 bits)
  - Higher layer protocols
  - TCP = 6, ICMP = 1, UDP = 17…
- Header checksum (16 bits)
  - Ensures some degree of header integrity
- Source Address
  - 32-bit IP address of sender
- Destination Address
  - 32-bit IP address of destination

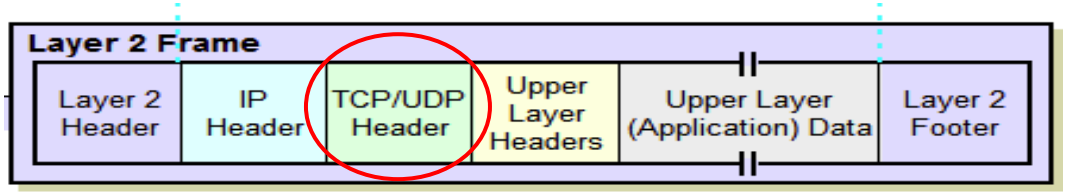




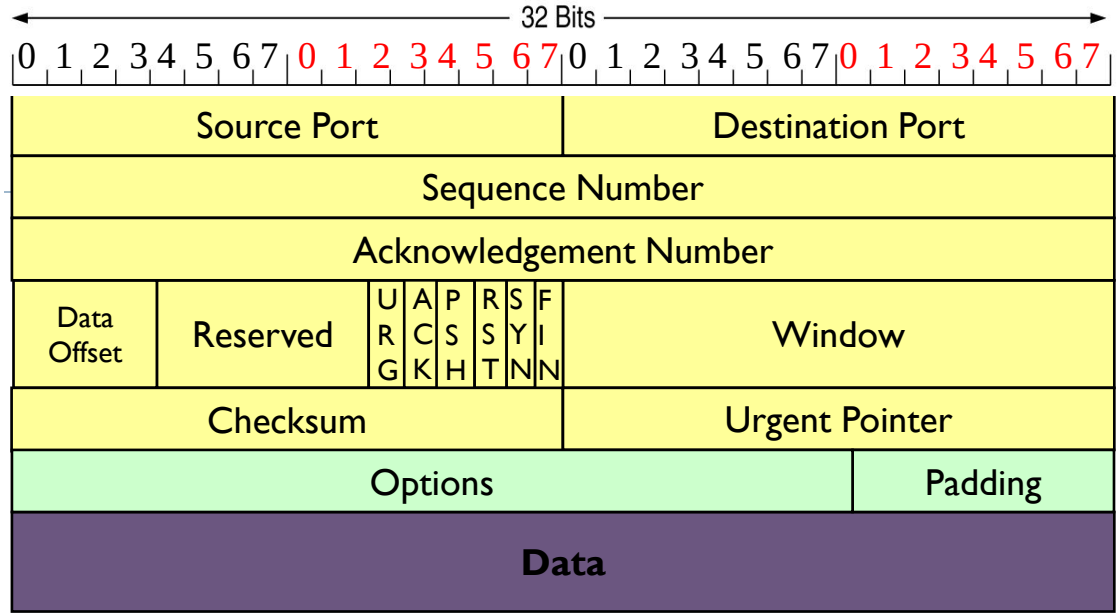
TCP Header



- Source Port: 16 bits
  - The source port number.
- Destination Port: 16 bits
  - The destination port number.
- Sequence Number: 32 bits
  - The sequence number of the first data octet in this segment.
- Acknowledgment Number: 32 bits
  - If the ACK control bit is set, this field contains the value of the next sequence number.



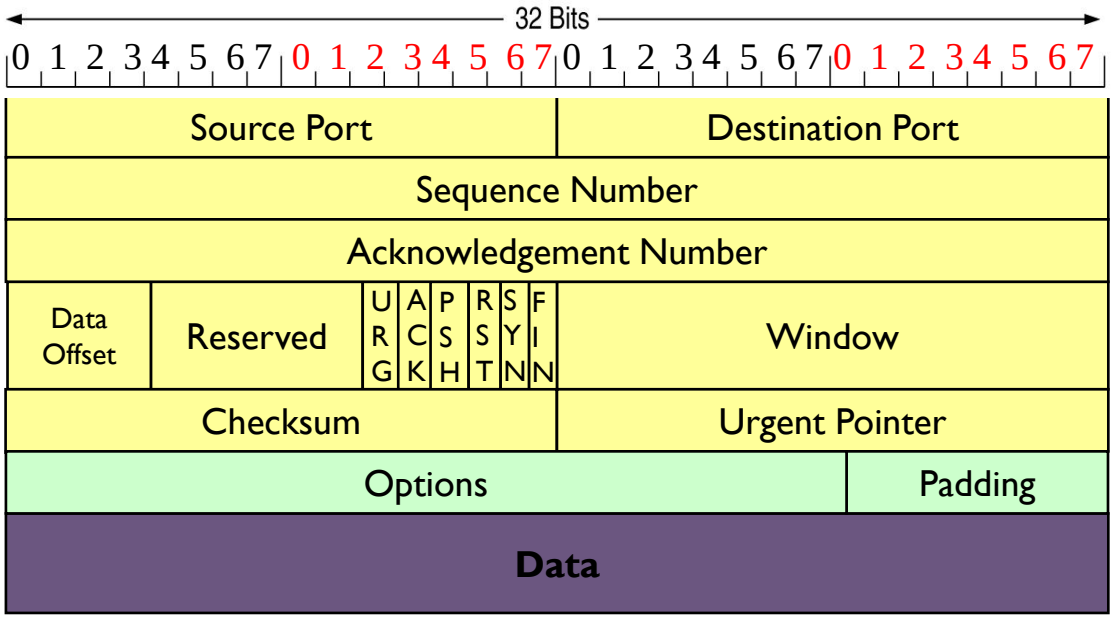
TCP Header



- Data Offset: 4 bits
  - The number of 32 bit words in the TCP Header. This indicates where the data begins.
- Reserved: 6 bits
  - Reserved for future use. Must be zero.
- Control Bits: 6 bits

  URG: Urgent Pointer field
  ACK: Acknowledgment field
  PSH: Push Function
  RST: Reset the connection
  SYN: Synchronize sequence numbers
  FIN: No more data from sender
- Window: 16 bits
  - The number of data octets beginning with the one indicated in the acknowledgment field which the sender of this segment is willing to accept.

TCP Header



- Checksum: 16 bits
- Urgent Pointer: 16 bits
  - The urgent pointer points to the sequence number of the octet following the urgent data. This field is only be interpreted in segments with the URG control bit set.

Protocol Header Analysis

Sample frame:

```
00 00 0c d9 fa 88 00 00 b4 a0 15 c1 08 00 45 00
00 28 04 04 40 00 80 06 42 a0 80 d3 a0 3c 80 0a
13 14 04 3a 00 15 54 f1 f2 09 d6 7d df 9d 50 10
40 5a b9 e8 00 00
```

- Ethernet header:

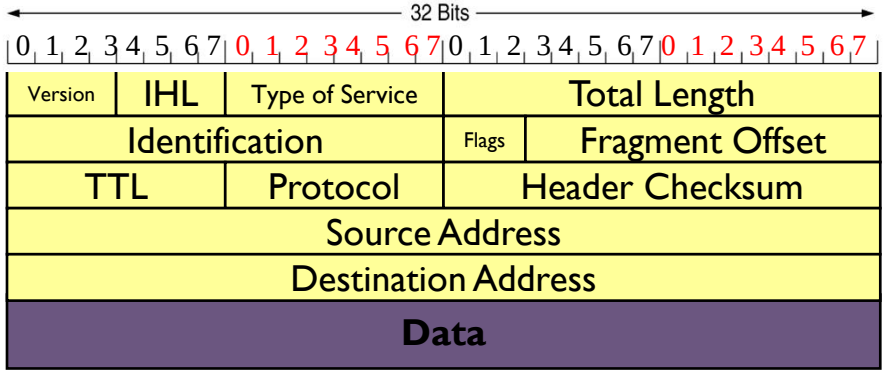
Destination Address (6 Bytes)	Source Address (6 Bytes)	Type (2 Bytes)	Data	CRC

- 00 00 0c d9 fa 88: Ethernet destination address is 00 00 0c d9 fa 88 (unicast).
- 00 00 b4 a0 15 c1: Ethernet source address: 00 00 b4 a0 15 c1 (unicast).
- 08 00: The payload type is IP (0x0800). (Note: 0x0806 is ARP.)

Protocol Header Analysis

Sample frame:

```
00 00 0c d9 fa 88 00 00 b4 a0 15 c1 08 00 45 00
00 28 04 04 40 00 80 06 42 a0 80 d3 a0 3c 80 0a
13 14 04 3a 00 15 54 f1 f2 09 d6 7d df 9d 50 10
40 5a b9 e8 00 00
```



IP header:

- 45: This is an IP version 4 datagram,
- 45: The header length is 5x4 = 20 bytes. (There is no *options* field in the given IP header).
- 00 : Type of Service
  - (0 0 0 0 0 0 0 0 in binary): This datagram has routine precedence (the lowest).
  - (0 0 0 0 0 0 0 0 in binary): the 3 type of service (ToS) bits

    0 0 0 *Normal delay*

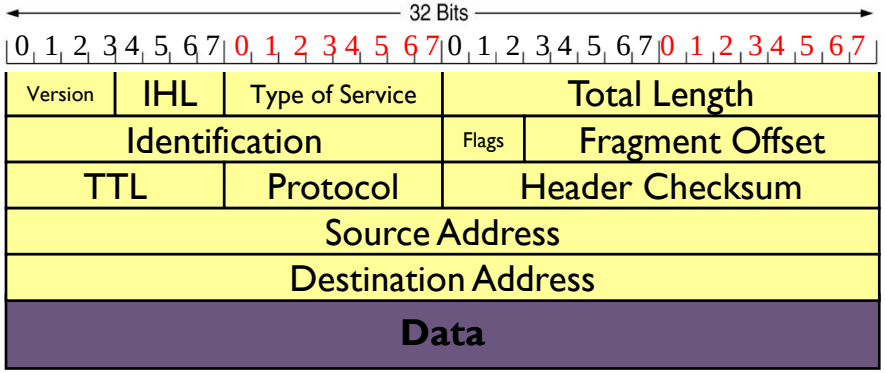
    0 0 0 *Normal throughput*

    0 0 0 *Normal Reliability*
- (0 0 0 0 0 0 0 0 in binary): The last two bits must be zero (for future use).

Protocol Header Analysis

Sample frame:

```
00 00 0c d9 fa 88 00 00 b4 a0 15 c1 08 00 45 00
00 28 04 04 40 00 80 06 42 a0 80 d3 a0 3c 80 0a
13 14 04 3a 00 15 54 f1 f2 09 d6 7d df 9d 50 10
40 5a b9 e8 00 00
```



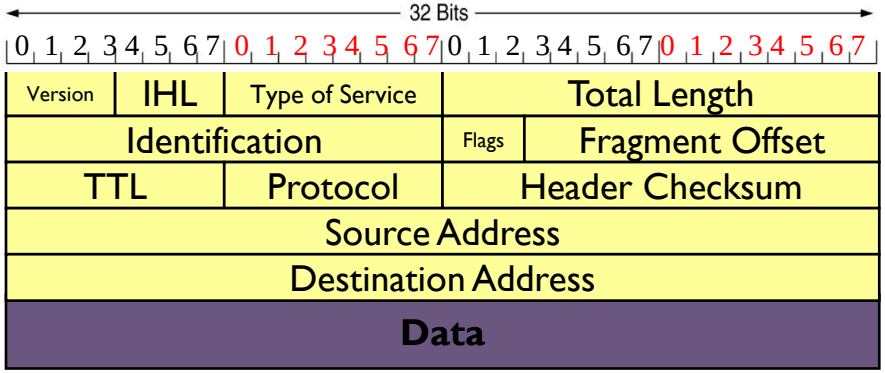
IP header:

- 00 28: Total length of the IP datagram is 40 (0x0028) bytes.
- 04 04: The identification of this datagram is 0x0404 (for fragmentation purpose).
- 40 00: (0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0):
  - 1 Don't Fragment flag set
  - 0 More Fragment flag unset
  - The Fragment offset is 0.
- This means that the datagram cannot be fragmented, and there are no fragments after this datagram. With a fragment offset equals to zero, we know that this is the only fragment of a datagram.

Protocol Header Analysis

Sample frame:

```
00 00 0c d9 fa 88 00 00 b4 a0 15 c1 08 00 45 00
00 28 04 04 40 00 80 06 42 a0 80 d3 a0 3c 80 0a
13 14 04 3a 00 15 54 f1 f2 09 d6 7d df 9d 50 10
40 5a b9 e8 00 00
```



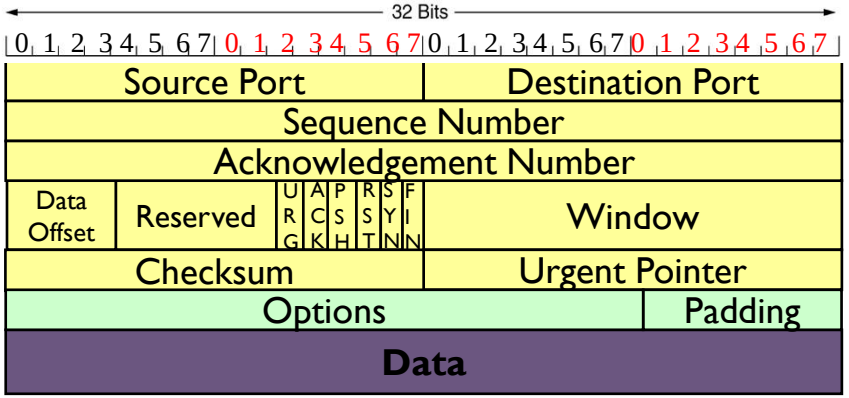
IP header:

- 80: Time to live = 128 (0x80), meaning the datagram may exist for *at most* 128 more hops.
- 06: The Protocol on top is TCP (0x06) (Note: 0x01 is ICMP and 0x11 is UDP).
- 42 a0: This is the checksum of the datagram.
- 80 d3 a0 3c: Source IP address is 128.211.160.60.
- 80 0a 13 14: Destination IP address is 128.10.19.20.

Protocol Header Analysis

Sample frame:

```
00 00 0c d9 fa 88 00 00 b4 a0 15 c1 08 00 45 00
00 28 04 04 40 00 80 06 42 a0 80 d3 a0 3c 80 0a
13 14 04 3a 00 15 54 f1 f2 09 d6 7d df 9d 50 10
40 5a b9 e8 00 00
```



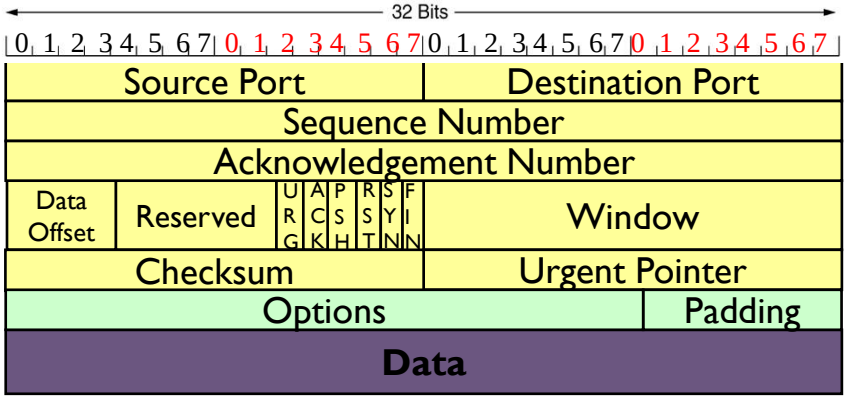
TCP header:

- 04 3a: The Source port is 1082, which is an arbitrarily port number assigned by the operating system.
- 00 15: The Destination port is 21, which is the well-known port for FTP (File Transfer Protocol).
- 54 f1 f2 09: The Seq. no. is 1425142281.
- d6 7d df 9d: The Ack no. is 3598573469.
- 50: Data offset is 20 (5 x 4) bytes. This is the length of the TCP header.

Protocol Header Analysis

Sample frame:

```
00 00 0c d9 fa 88 00 00 b4 a0 15 c1 08 00 45 00
00 28 04 04 40 00 80 06 42 a0 80 d3 a0 3c 80 0a
13 14 04 3a 00 15 54 f1 f2 09 d6 7d df 9d 50 10
40 5a b9 e8 00 00
```



TCP header:

- 10 (0 0 0 1 0 0 0 0):
  - Flags: URG 0 ACK 1 PSH 0
    RST 0 SYN 0 FIN 0

Only the ACK flag is set, meaning that the value carried in the acknowledgement field is valid.
(You should comment on all the flags that are set, i.e., equal to 1)

- 40 5a: the receiver window size is 16474 (0x405a) bytes.
- b9 e8: Checksum of the whole TCP segment.
- 00 00: Urgent pointer (Not used in this segment).

- Data: none

Notes for Protocol Analysis

- You are going to parse two frames according to the first digit of your Group Number, i.e., if the lowest digit of your group number = *i*, do frame *i* and frame *i*+*10*.
- Using the example given in the lab manual as a template, parse the frames in a human-readable format and comment.
- For example, write an IP address in the dotted decimal notation and header length as a positive integer.
- Also, color (or, underline) the different parts of the frames to indicate their layers: 2, 3, 4, or app data and indicate the name of the highest layer protocol.

Network Utilities

1. *arp*
2. *ifconfig*
3. *netstat*
4. *nslookup*
5. *ping*
6. *traceroute*