

# CS 168 Reference Card

Version 1.1  
Revised Mar 26, 2025

## Bit: IPv4 Header

0	Version (4)	Hdr len* (4)	Type of Service (8)	Total Length in Bytes (16)	
32	Identification (16)			Flags (3)	Fragment Offset (13)
64	TTL (8)		Protocol (8)	Header Checksum (16)	
96	Source IP Address (32)				
128	Destination IP Address (32)				
160	Options (if any)				
...	Payload				

IPv4 protocol numbers: 1=ICMP 6=TCP 17=UDP

## Bit: IPv6 Header

0	Version (4)	Traffic Class (8)	Flow Label (20)		
32	Payload Length in Bytes (16)			Next Header (8)	Hop Limit (8)
64	Source IP Address (128)				
192	Destination IP Address (128)				

## Bit: UDP Header

0	Source Port (16)		Destination Port (16)	
32	Total Length in Bytes (16)		Checksum (16)	
...	Payload			

## Bit: TCP Header

0	Source Port (16)		Destination Port (16)	
32	Sequence Number (32)			
64	Acknowledgment Number (32)			
96	Hdr Len* (4)	Unused (4)	Flags (8)	Advertised Window (16)
128	Checksum (8)		Urgent Pointer (8)	
160	Options (variable-length)			
...	Payload			

\*Note: Header length is measured in 4-byte words.

Bit:

### ICMP Header

0	Type (8)	Code (8)	Checksum (16)
32	Rest of header, all 0 for this class (32)		
...	Payload		

ICMP Port Unreachable: Type=3, Code=3

ICMP TTL Exceeded: Type=11, Code=0

### Ethernet Frame

Preamble (7)	SFD (1)	Destination MAC (6)	Source MAC (6)	Type (2)	Payload	FCS (4)	IPG (12)
--------------	---------	---------------------	----------------	----------	---------	---------	----------

Size of each field measured in bytes, not bits.

Preamble, SFD, and IPG separate packets on the wire. FCS is checksum.

### SI Prefixes

Size	Prefix	Symbol	Size	Prefix	Symbol
10 <sup>-3</sup>	milli-	m	10 <sup>3</sup>	kilo-	k
10 <sup>-6</sup>	micro-	μ	10 <sup>6</sup>	mega-	M
10 <sup>-9</sup>	nano-	n	10 <sup>9</sup>	giga-	G
10 <sup>-12</sup>	pico-	p	10 <sup>12</sup>	tera-	T

bps = Bits per second. B/s = Bytes per second.

$$\text{TCP Throughput} = \sqrt{\frac{3}{2}} \times \frac{\text{MSS}}{\text{RTT} \sqrt{p}} \quad \text{where } p \text{ is packet drop probability.}$$