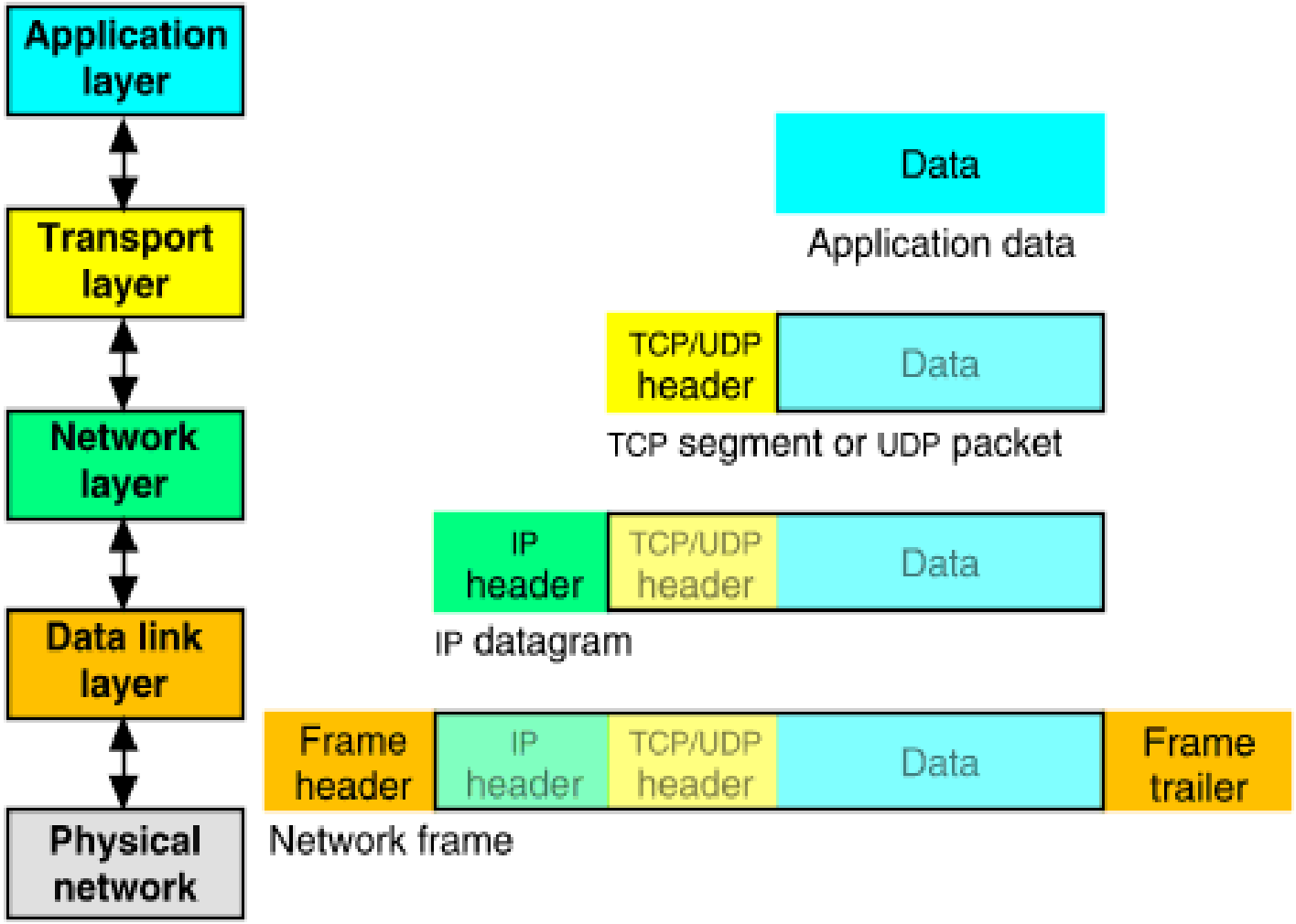
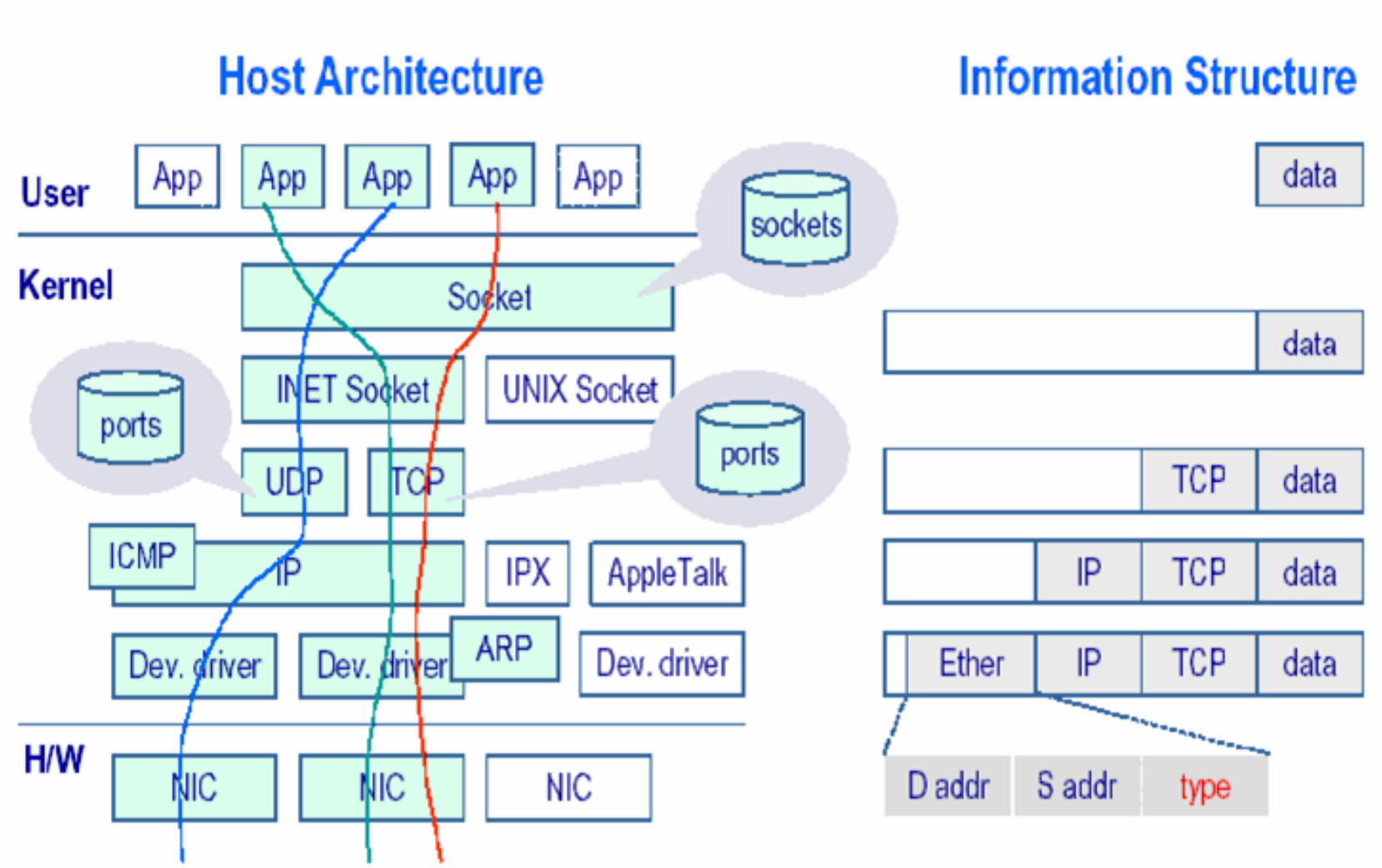
Raw Sockets

Network Packet Encapsulation



Path & Headers



Standard Sockets

- Can only receive frames destined to:
  - Specific address
  - Broadcast
  - Multicast
- Headers (Ethernet, IP, TCP, etc) are stripped by the network stack.
- Packet headers cannot be modified before send.

Advanced Functions

- Promiscuous mode
  - receive all frames in broadcast domain
- Raw Sockets:
  - Receive complete packets, including headers
  - Inject packets with custom headers and data into the network

Promiscuous Mode

- It is the “See All, Hear All” Wizard mode ☺
- Tells the network driver to accept all packets irrespective of whom the packets are addressed to.
- Used for Network Monitoring – both legal and illegal monitoring ☺
- We can do this by programmatically setting the IFF_PROMISC flag or by using the ifconfig utility (ifconfig eth0 promisc)

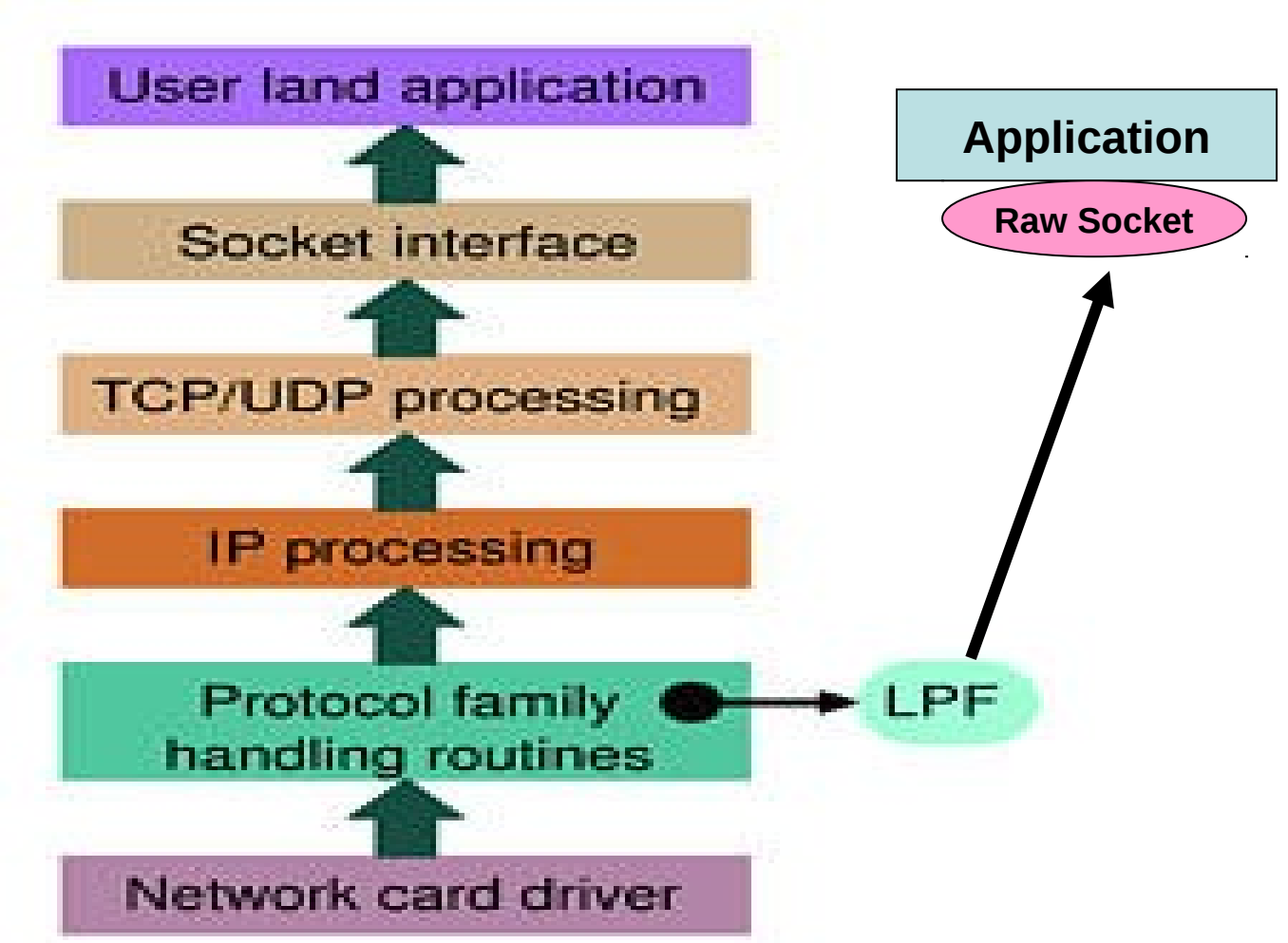
Getting all headers - Sniffing

- Once we set the interface to promiscous mode we can get “full packets” with all the headers.
- We can process these packets and extract data from it.
- Note we are receiving packets meant for all hosts => see what your neighbors are doing in the lab ☺

Sending arbitrary packets – Packet Injection

- We “manufacture” our own packets and send it out on the network.
- Absolute power – total network stack bypass
- Most active network monitoring tools and hacking tools use this
  - Dos attacks
  - Syn Floods
  - IP Spoofs

Raw Sockets – a closer look



Raw sockets

- Provide a way to bypass the whole network stack
- Deliver a packet directly to an application.

PF_PACKET

- It is a software interface to send/receive packets at layer 2 of the OSI i.e. device driver.
- All packets received will be complete with all headers and data.
- All packets sent will be transmitted without modification by the kernel to the medium.
- Supports filtering using Berkley Packet Filters.

Creating a Raw Socket

- Call socket() with appropriate arguments.

  Socket(PF_PACKET, SOCK_RAW, int protocol)

  Protocol is ETH_P_IP for IP networks. It is mostly used as a filter. To receive all types of packets ETH_P_IP is used.

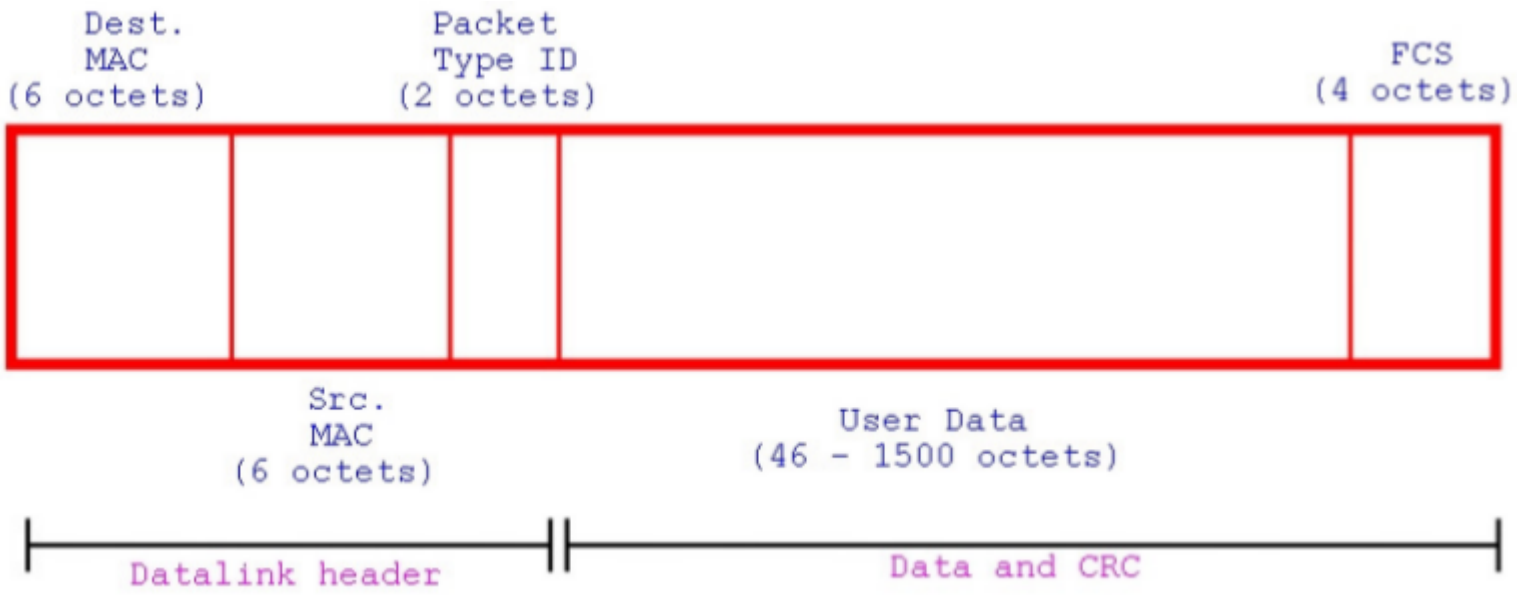
The making of a Sniffer

- Create Raw socket – socket()
- Set interface you want to sniff on in promiscous mode.
- Bind Raw socket to this interface – bind()
- Receive packets on the socket – recvfrom()
- Process received packets
- Close the raw socket().

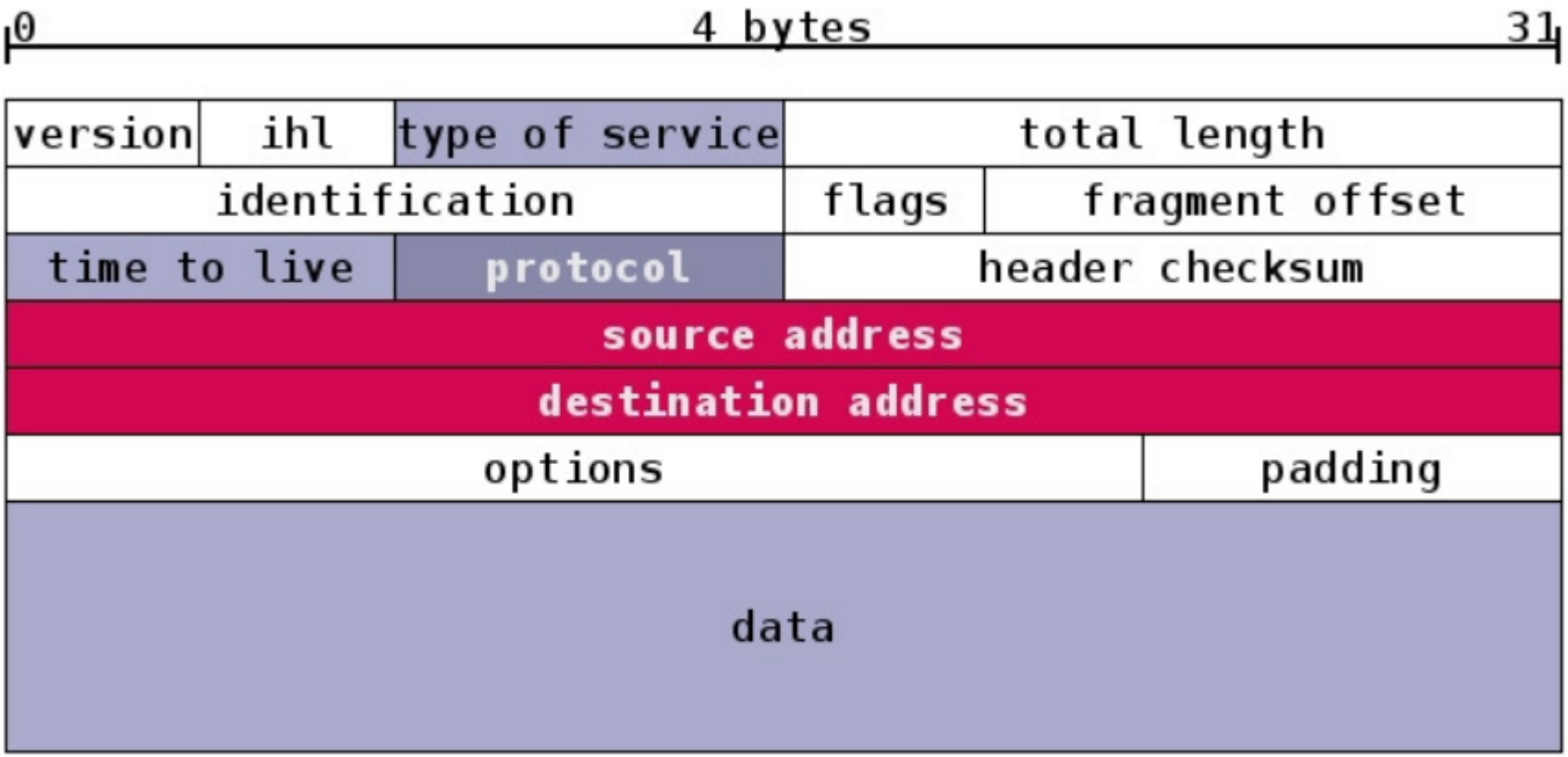
The making of a Packet Injector

- Create a raw socket – socket()
- Bind socket to the interface you want to send packets onto – bind()
- Create a packet
- Send the packet – sendto()
- Close the raw socket – close()

Ethernet Frame

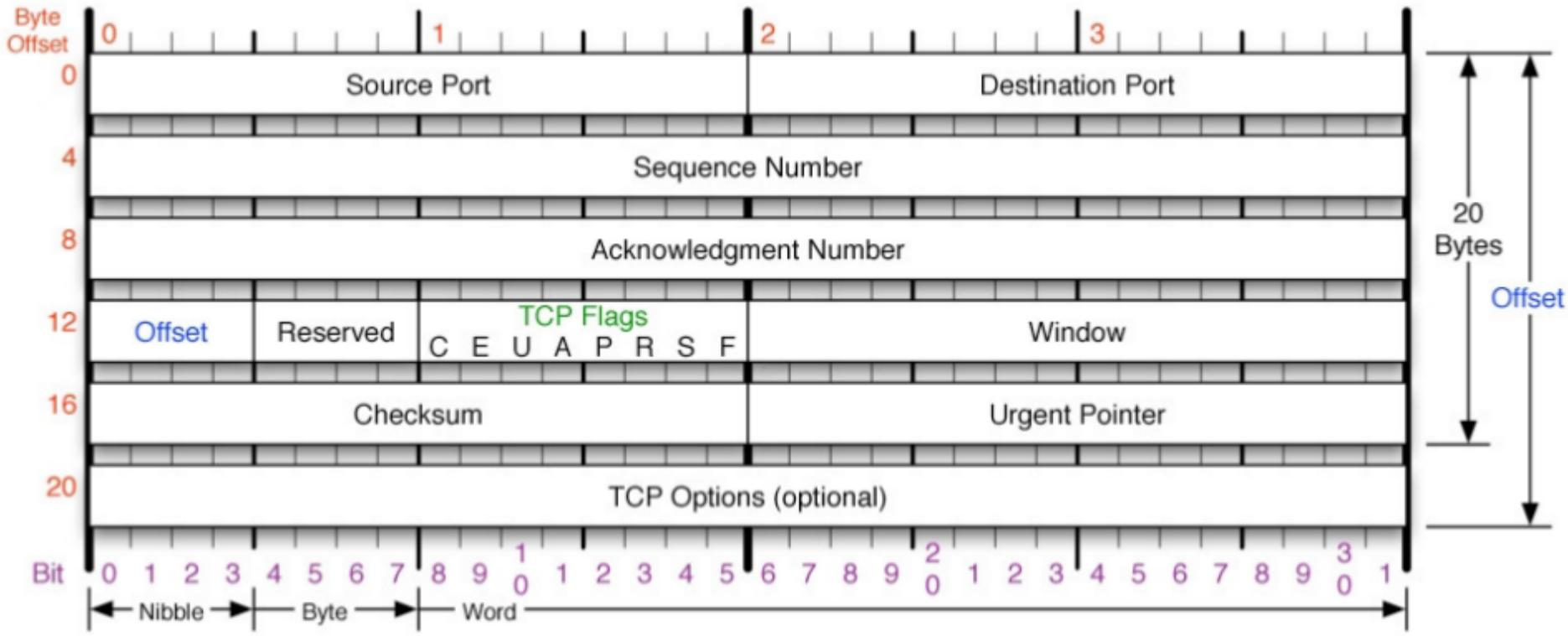


IP Frame


0
4 bytes
31
version
ihl
type of service
total length
identification
flags
fragment offset
time to live
protocol
header checksum
source address
destination address
options
padding
data

UDP Frame

TCP Frame


Byte Offset
0
1
2
3
0
Source Port
Destination Port
4
Sequence Number
8
Acknowledgment Number
12
Offset
Reserved
TCP Flags
C E U A P R S F
Window
16
Checksum
Urgent Pointer
20
TCP Options (optional)
20 Bytes
Offset
Bit
0 1 2 3 4 5 6 7 8 9 10 1 2 3 4 5 6 7 8 9 20 1 2 3 4 5 6 7 8 9 30 1
Nibble
Byte
Word