Intro to Computer Networking

John Rodriguez

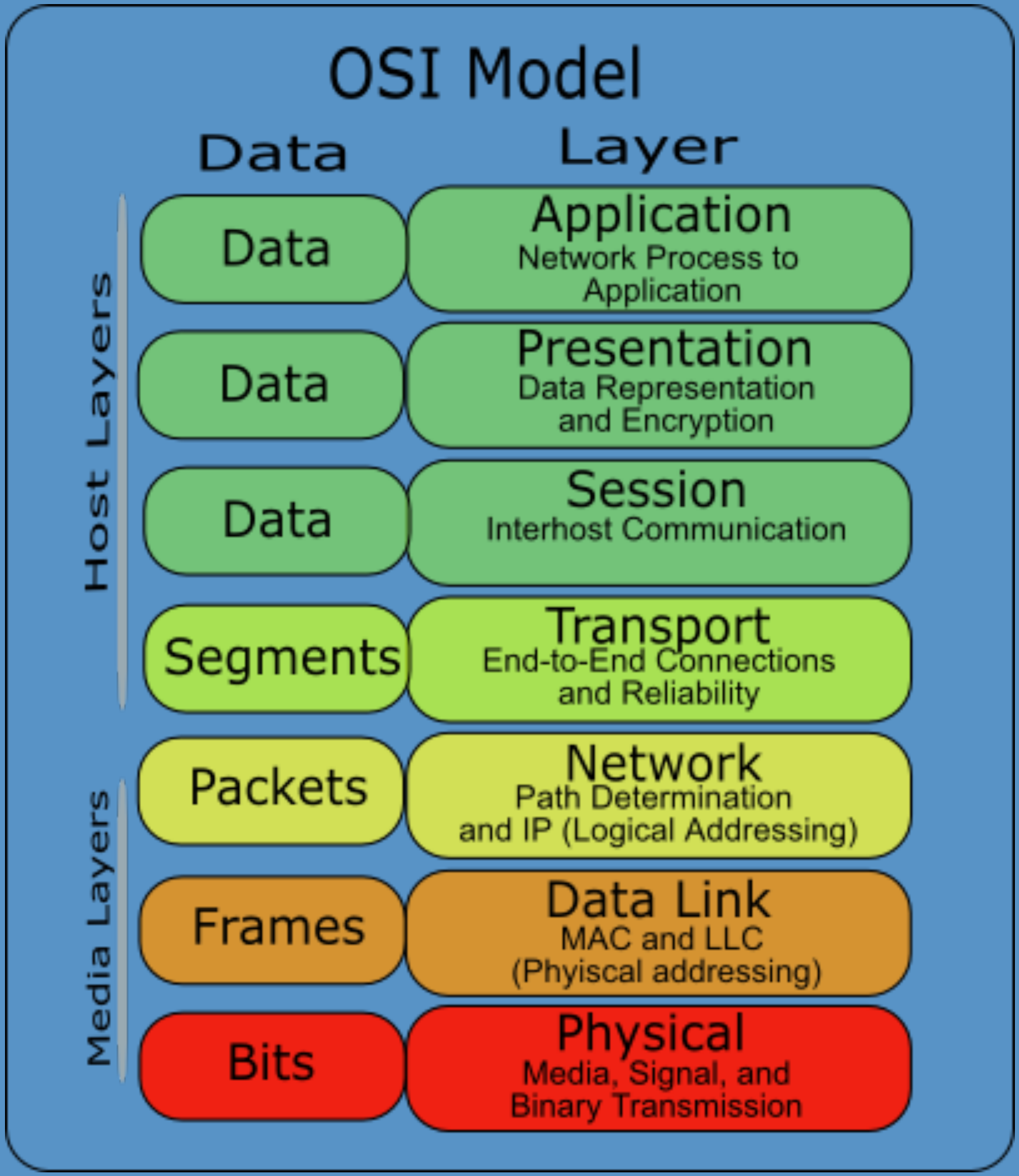
Networking

- *an infrastructure allowing computers to exchange data*
- At some point, computers were siloed devices, limited to word processing, local audio/image processing, games, etc. People did not collaborate via machines.

OSI Model

- a conceptual model that standardizes the functions of a telecommunication or computing system without regard of their underlying internal structure and technology
- goal: the ***interoperability*** of diverse communication systems with standard protocols
- A layer serves the layer above it and is served by the layer below it
- The original version of this model defined seven layers

OSI Model

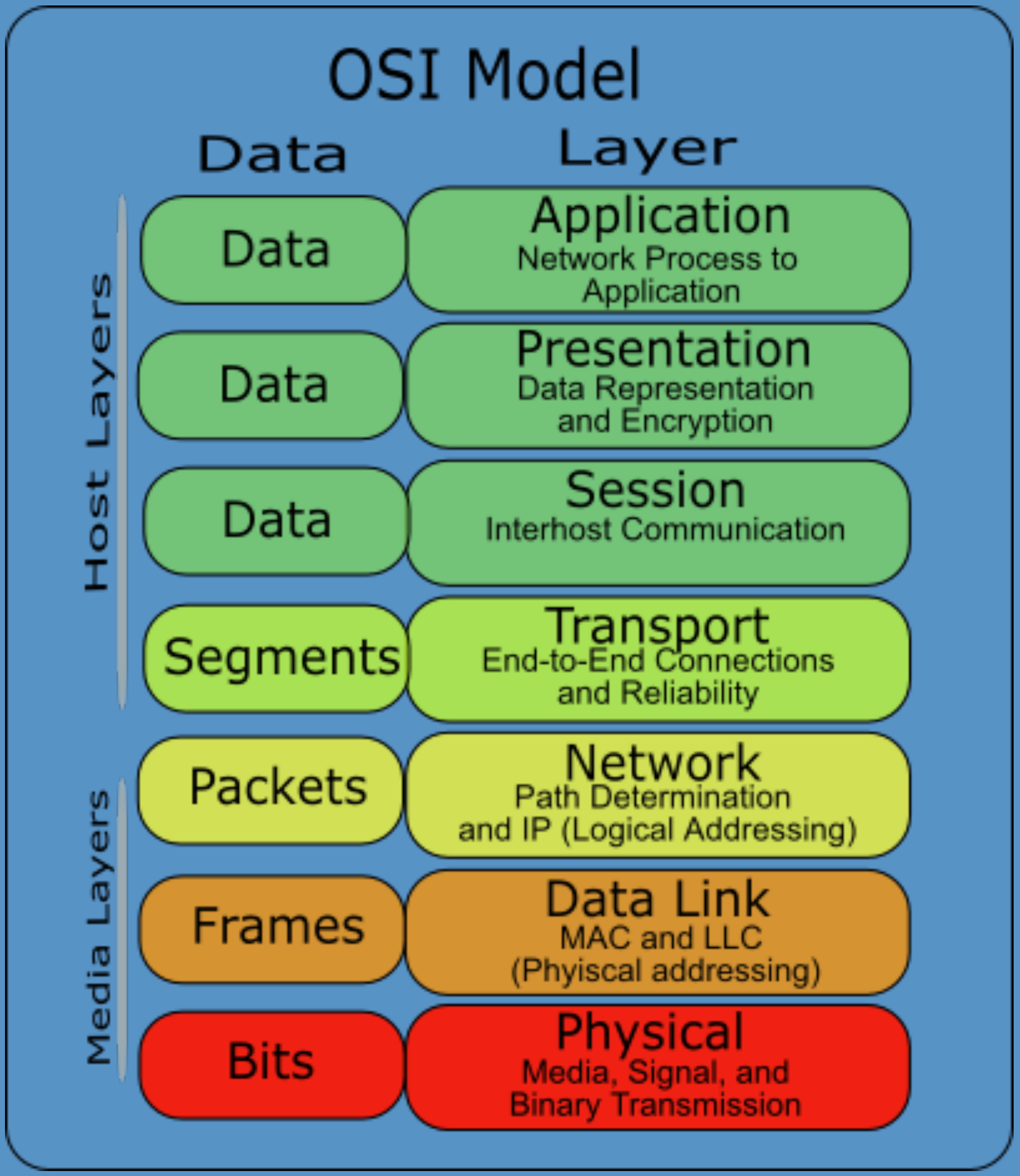


OSI Model
Data
Layer
Host Layers
Media Layers
Data
Application
Network Process to Application
Data
Presentation
Data Representation and Encryption
Data
Session
Interhost Communication
Segments
Transport
End-to-End Connections and Reliability
Packets
Network
Path Determination and IP (Logical Addressing)
Frames
Data Link
MAC and LLC (Phyiscal addressing)
Bits
Physical
Media, Signal, and Binary Transmission

OSI Model

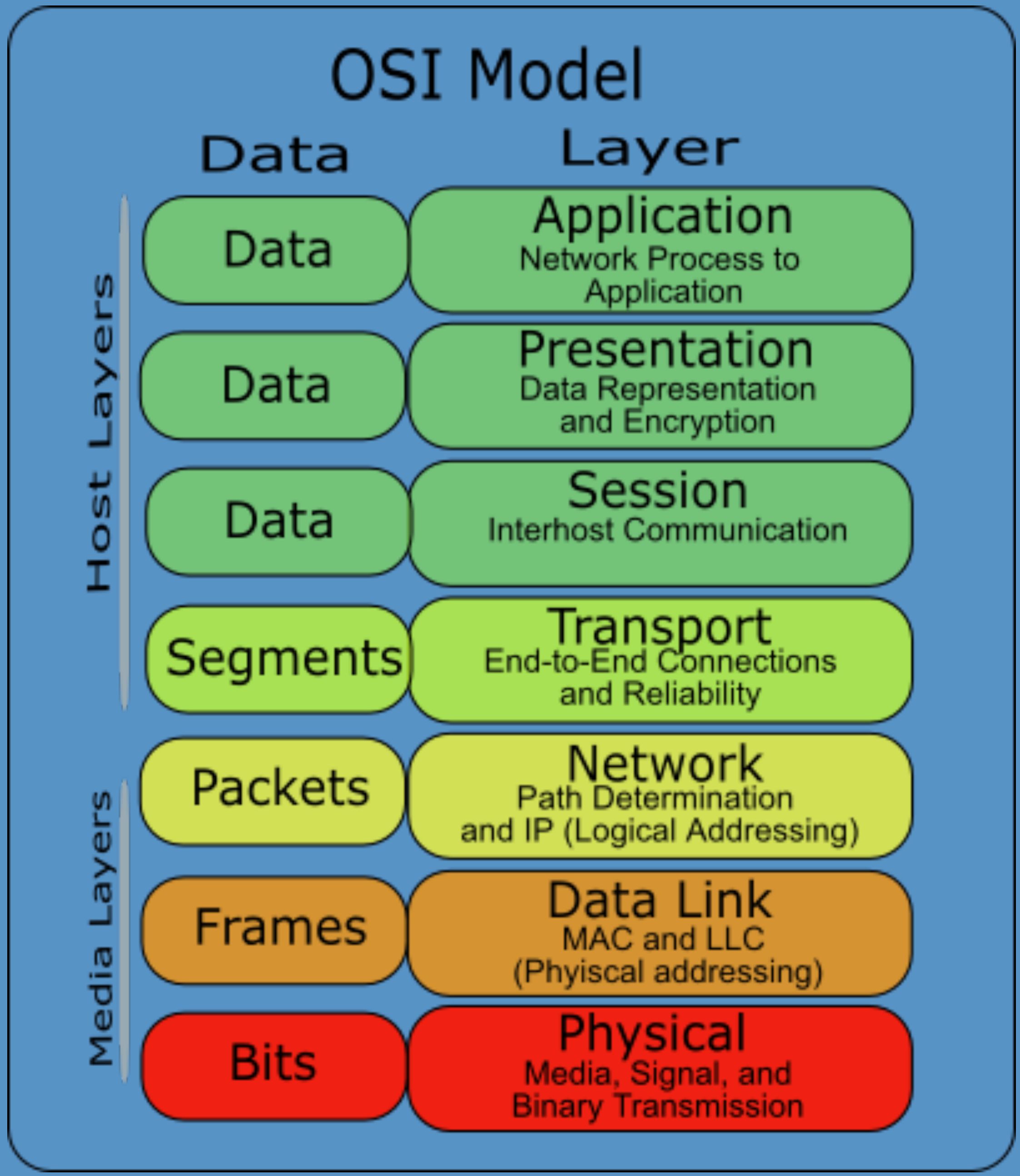
App 1

↓



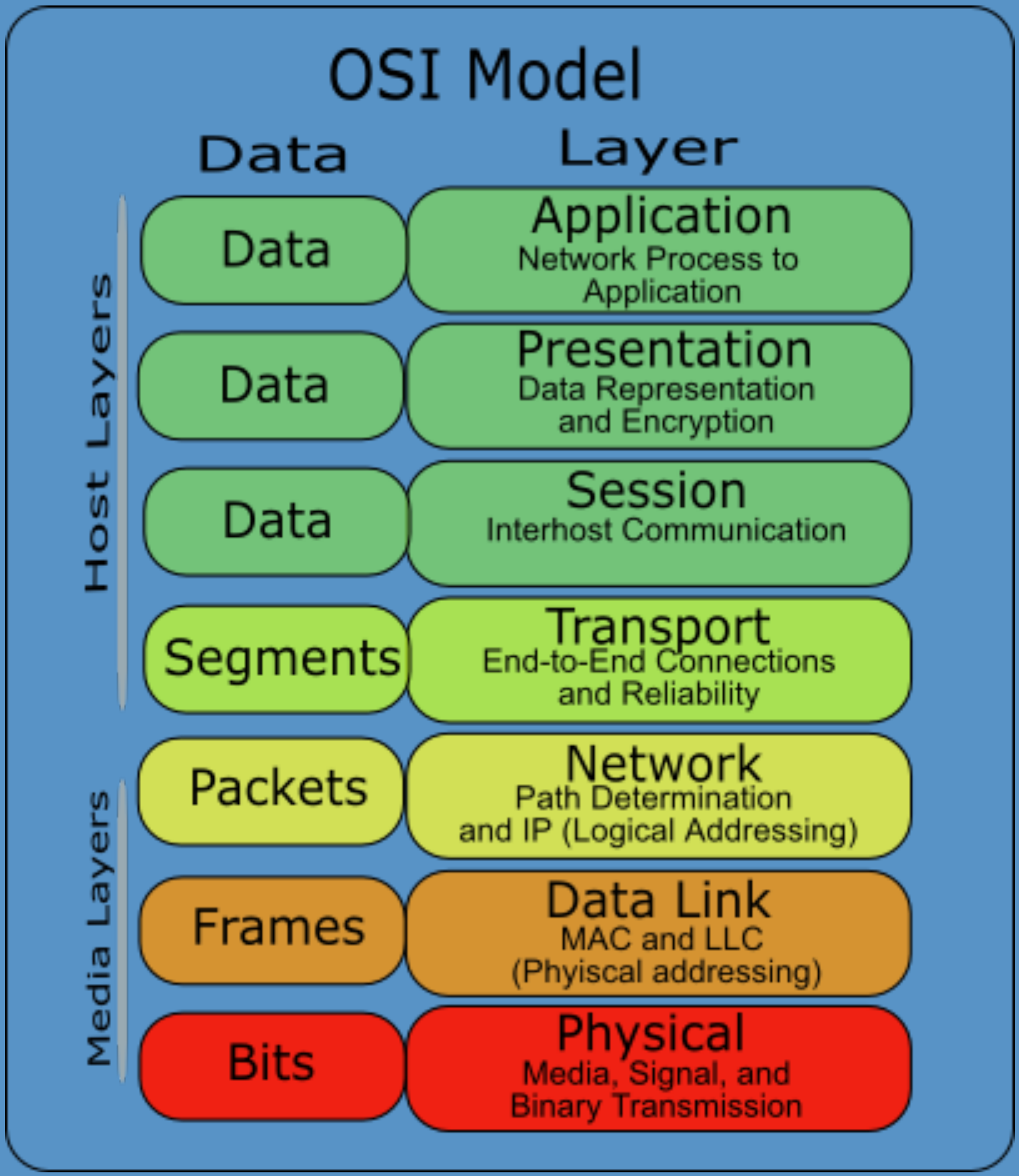
OSI Model

App 1



OSI Model

App 1

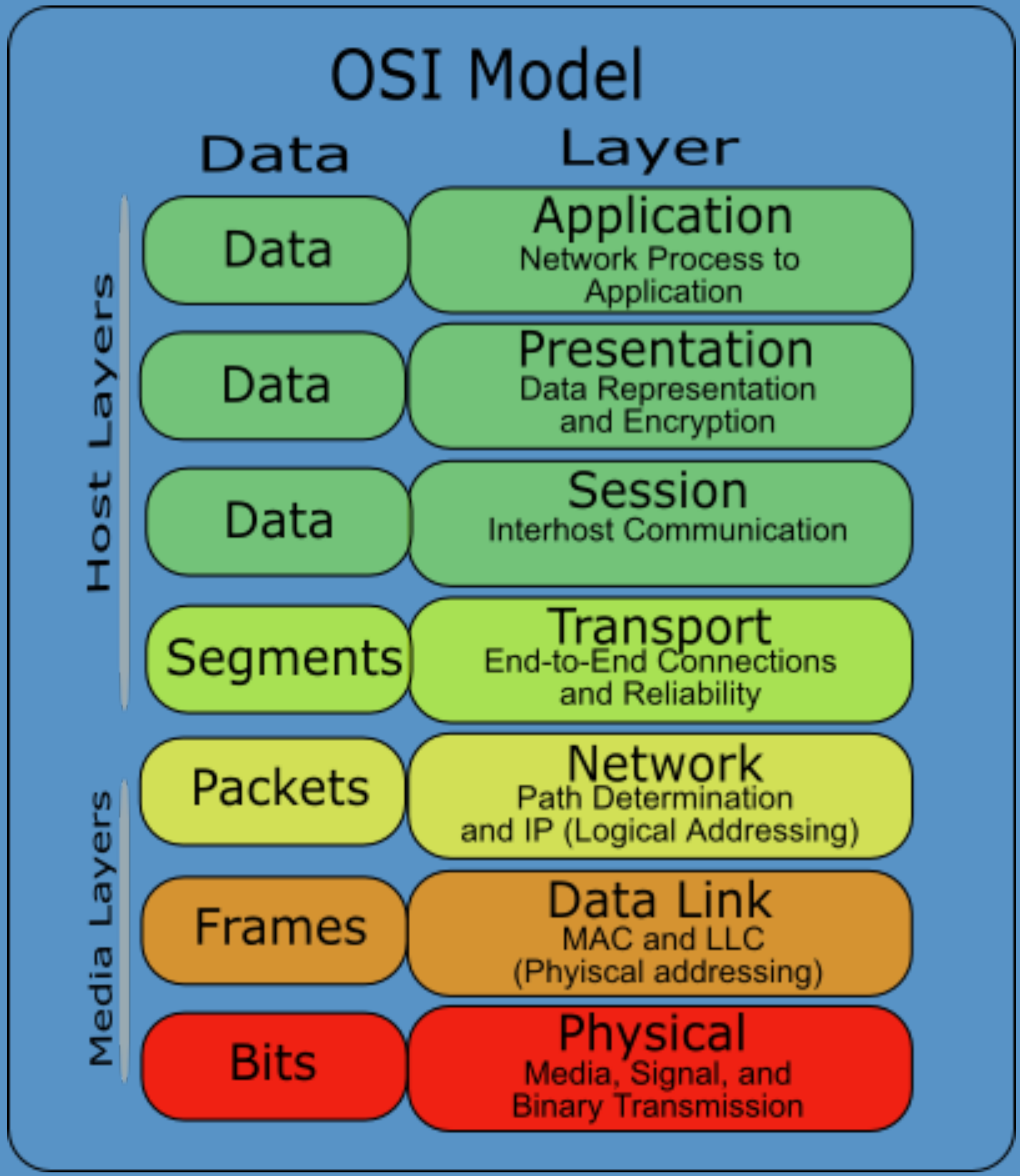


App 2

OSI Model

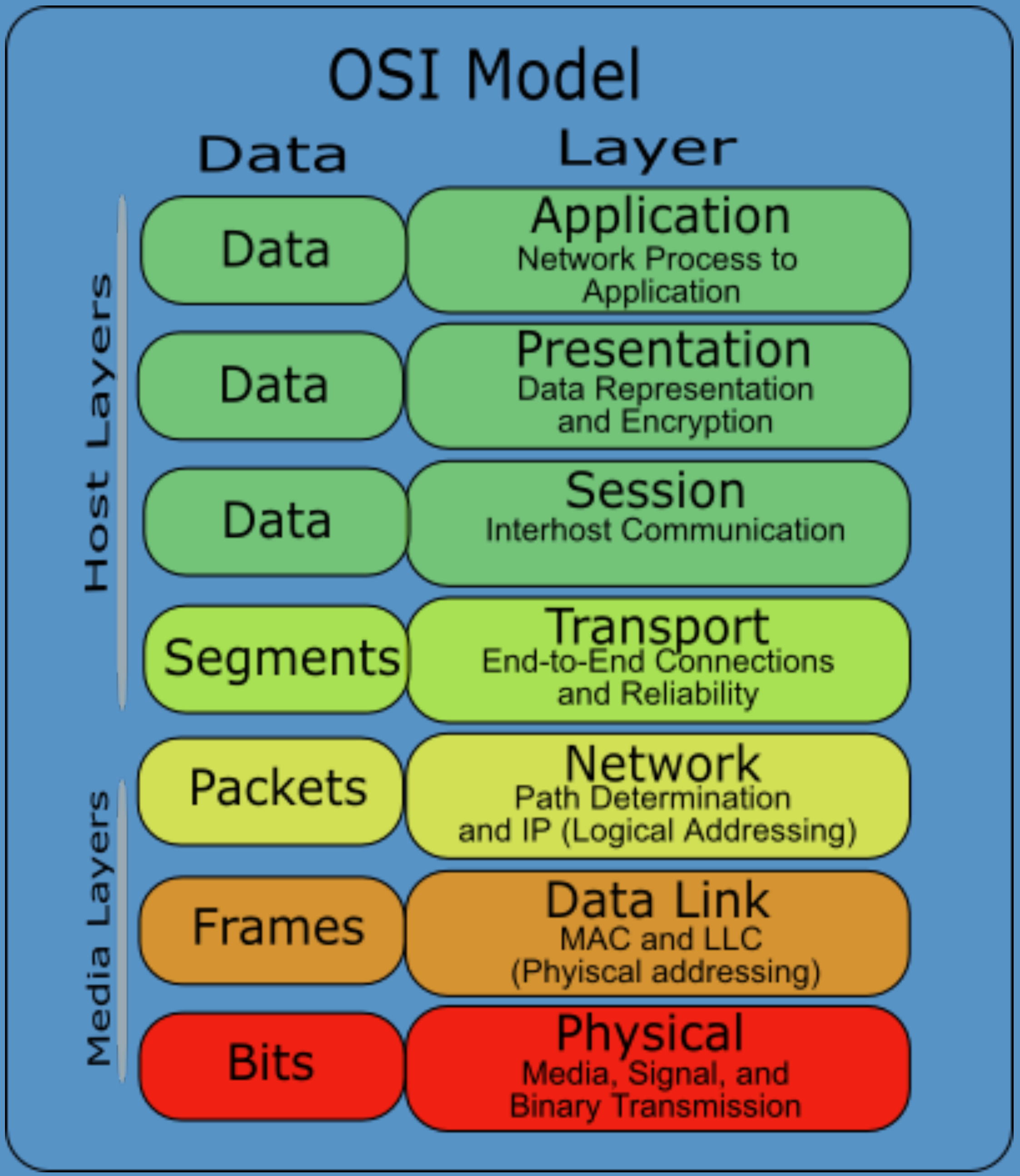
App 1

App 2

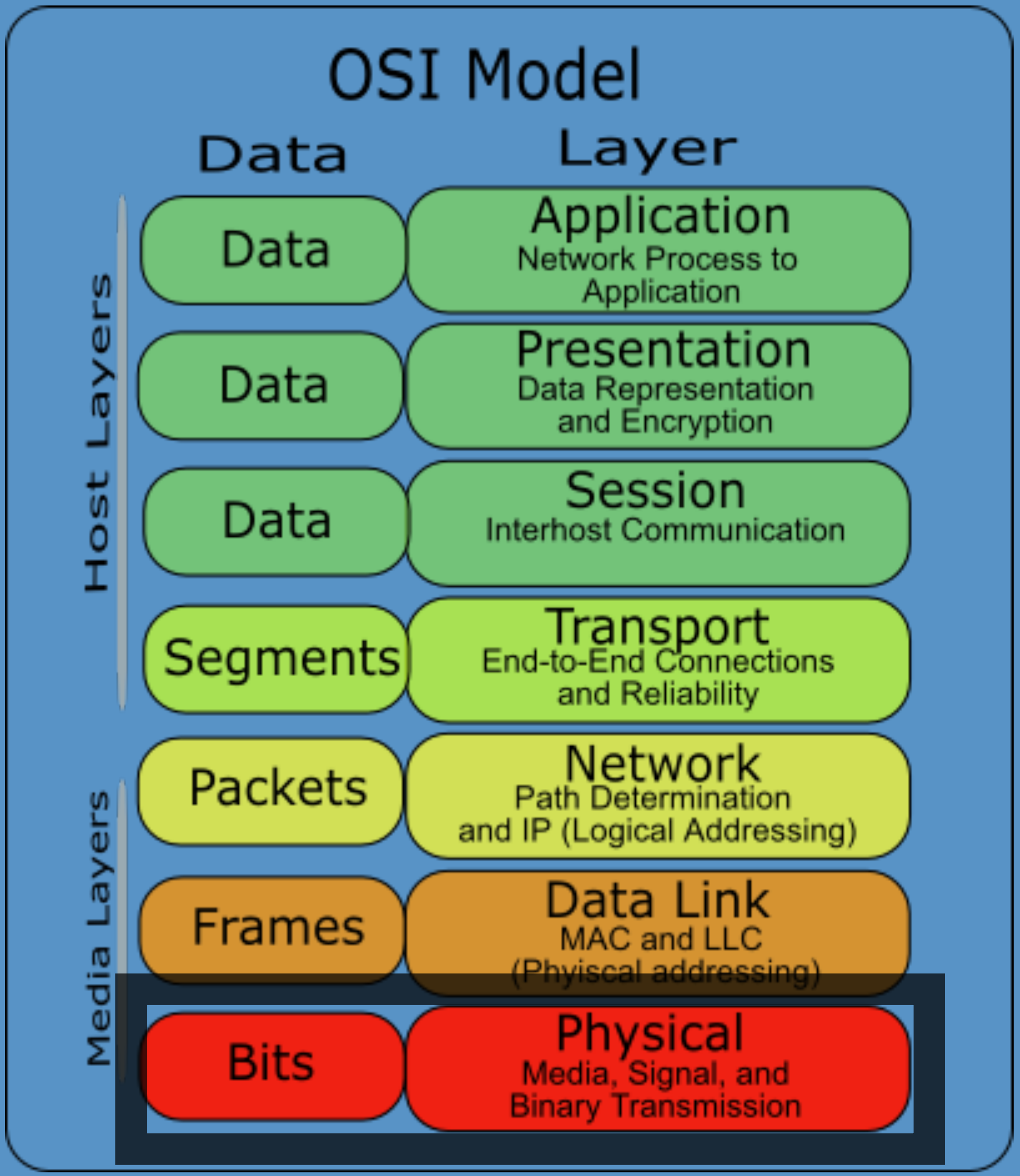


Layer 1

Layer 1


OSI Model
Data
Layer
Host Layers
Media Layers
Data
Application
Network Process to Application
Data
Presentation
Data Representation and Encryption
Data
Session
Interhost Communication
Segments
Transport
End-to-End Connections and Reliability
Packets
Network
Path Determination and IP (Logical Addressing)
Frames
Data Link
MAC and LLC (Phyiscal addressing)
Bits
Physical
Media, Signal, and Binary Transmission

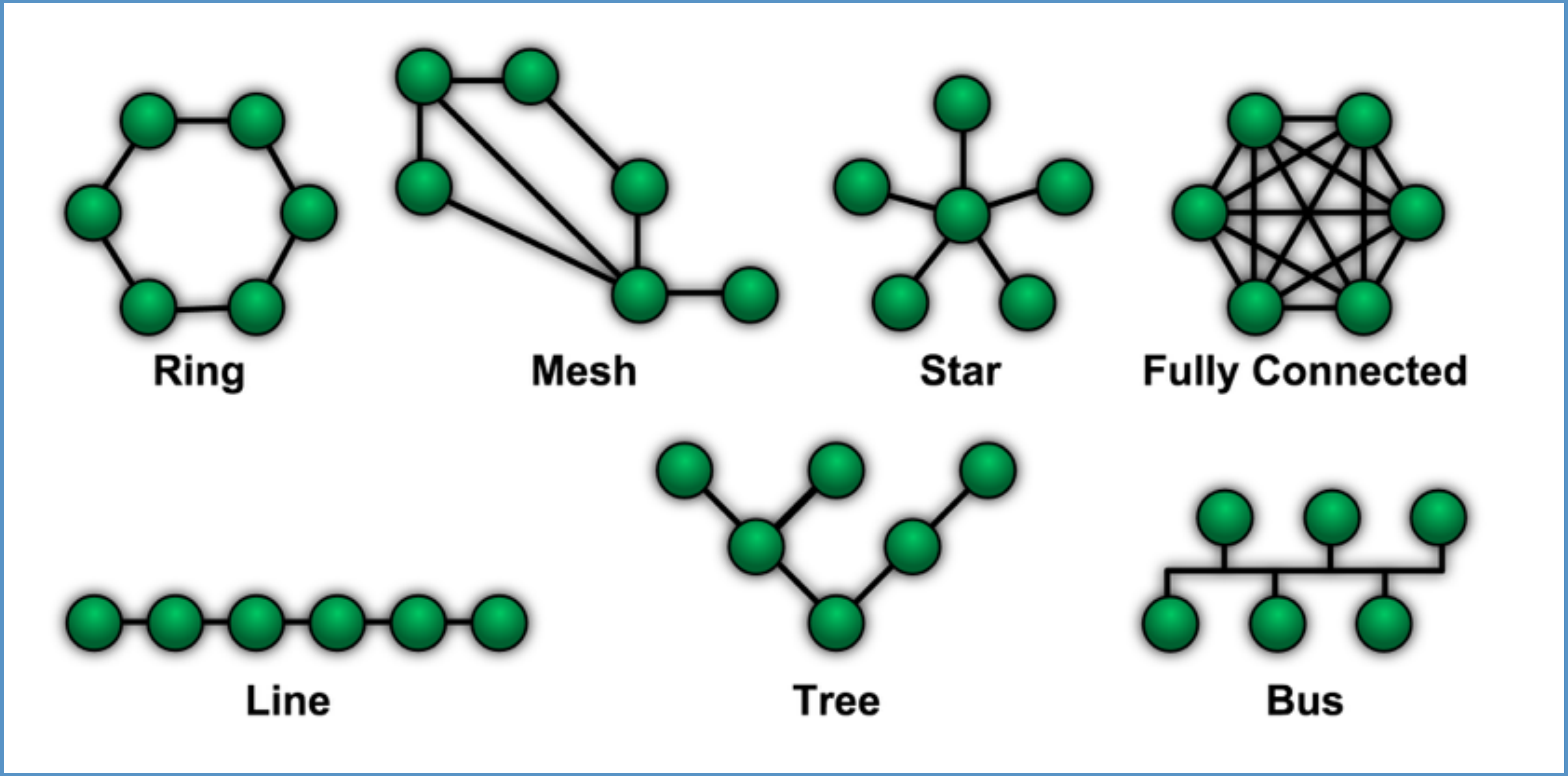
Layer 1


OSI Model
Data
Layer
Host Layers
Data
Application
Network Process to Application
Data
Presentation
Data Representation and Encryption
Data
Session
Interhost Communication
Segments
Transport
End-to-End Connections and Reliability
Media Layers
Packets
Network
Path Determination and IP (Logical Addressing)
Frames
Data Link
MAC and LLC (Phyiscal addressing)
Bits
Physical
Media, Signal, and Binary Transmission

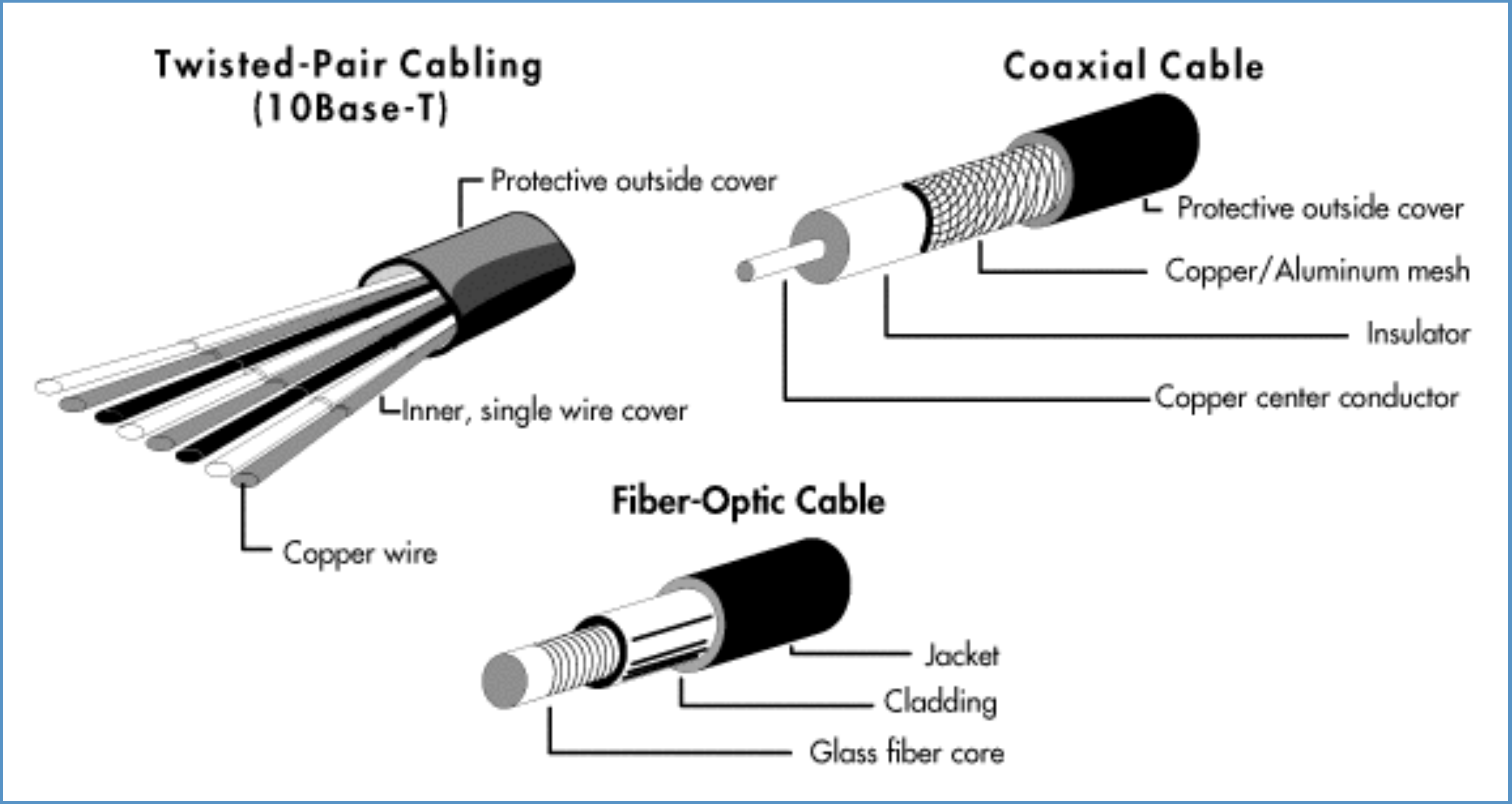
Layer 1: Physical Layer

- defines the means of transmitting raw bits rather than logical data packets over a physical link connecting network nodes
- defines the relationship between a device and a physical transmission medium (e.g., a copper or fiber optical cable, radio frequency)
- defines the transmission mode: simplex, half duplex, full duplex
- defines the network topology: bus, mesh, ring
- Transmission media: air, coaxial cable, category 5E cable, USB cable, fiber cable

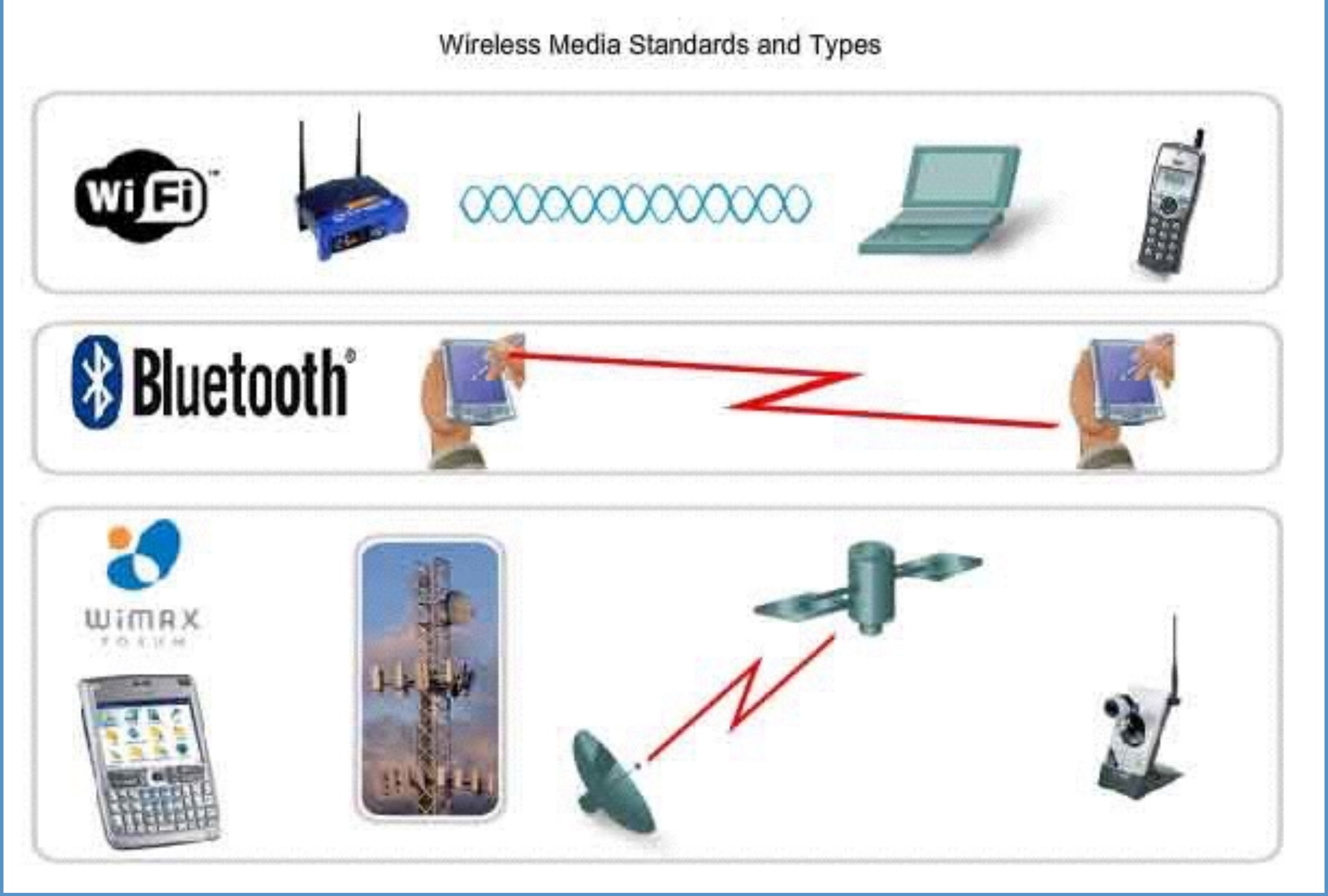
Network Topology



Transmission Media



Transmission Media

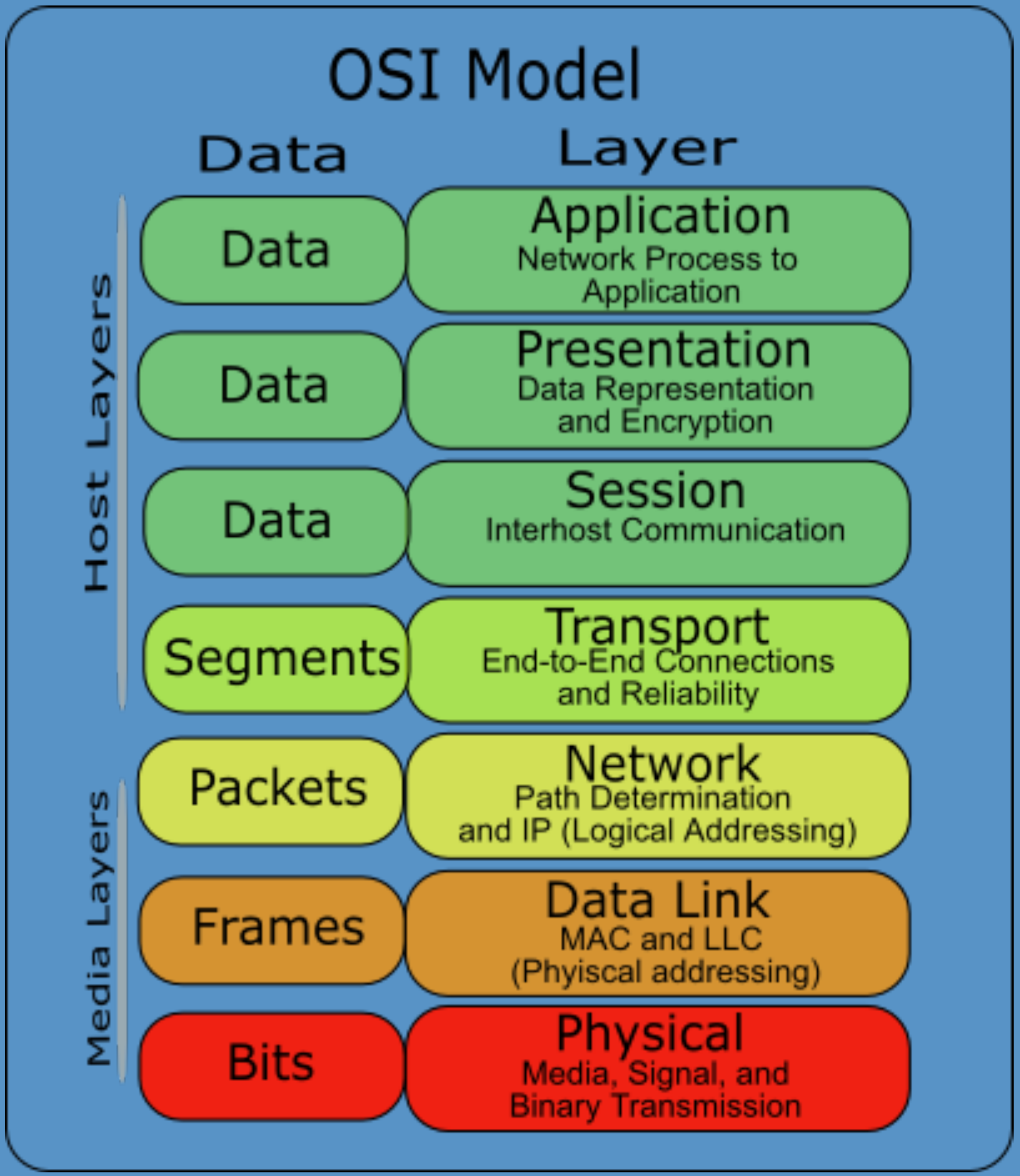


Layer 1: Examples

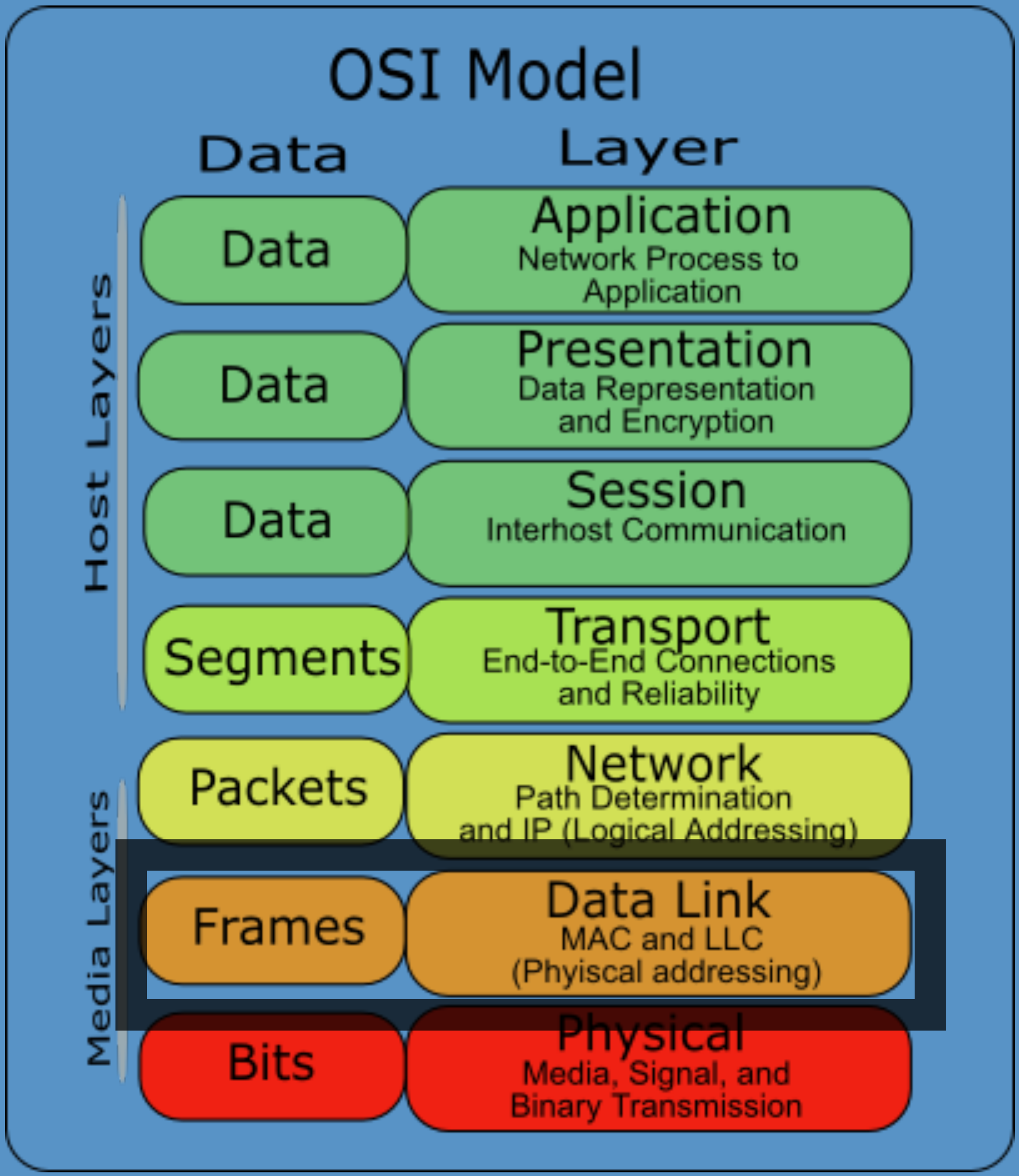
- Digital Subscriber Line (DSL)
- Integrated Services Digital Network (ISDN)
- Infrared Data Association (scanners)
- Universal Serial Bus (USB)
- Bluetooth
- Ethernet

Layer 2

Layer 2


OSI Model
Data
Layer
Host Layers
Media Layers
Data
Application
Network Process to Application
Data
Presentation
Data Representation and Encryption
Data
Session
Interhost Communication
Segments
Transport
End-to-End Connections and Reliability
Packets
Network
Path Determination and IP (Logical Addressing)
Frames
Data Link
MAC and LLC (Phyiscal addressing)
Bits
Physical
Media, Signal, and Binary Transmission

Layer 2

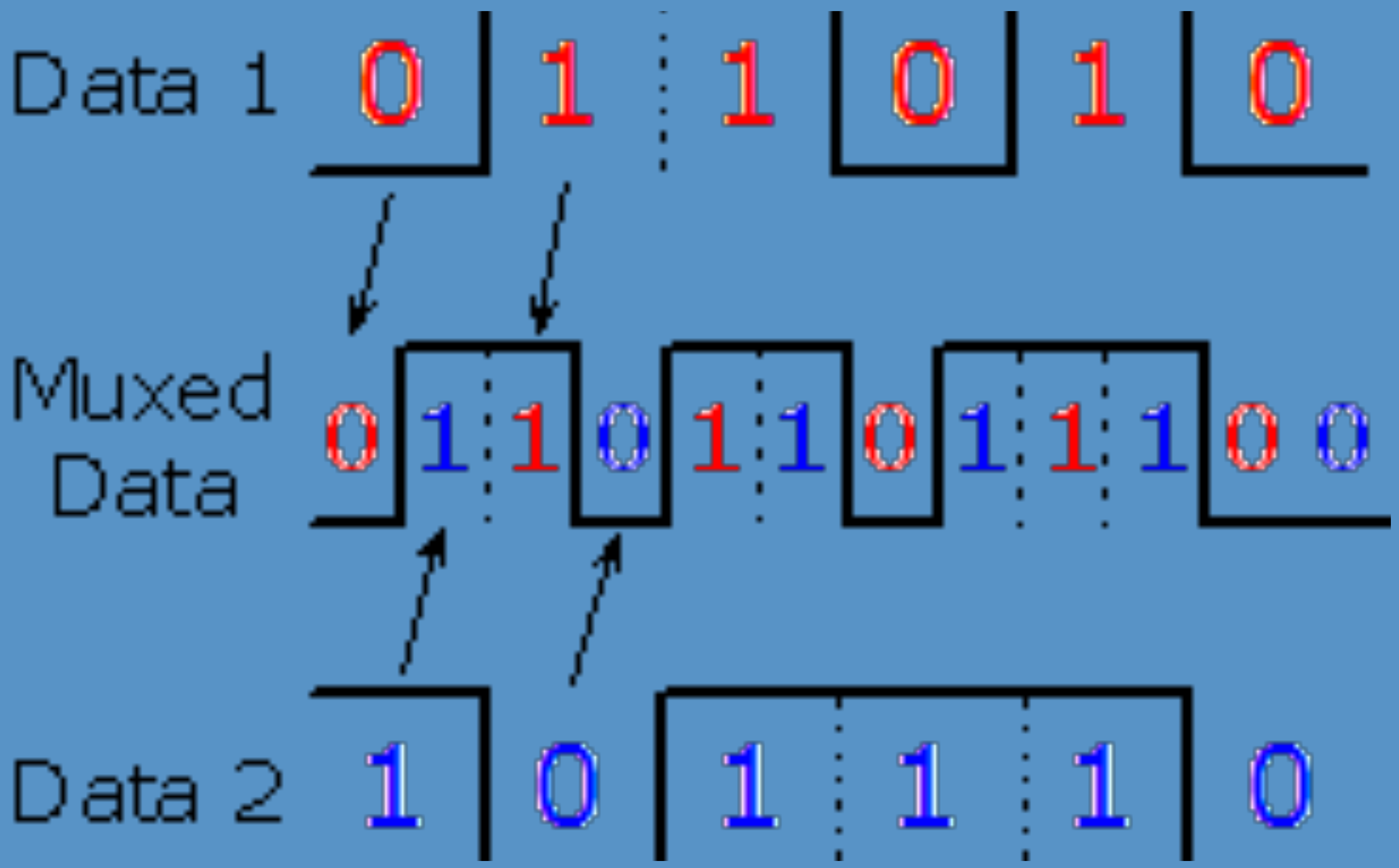


OSI Model
Data
Layer
Data
Application
Network Process to Application
Data
Presentation
Data Representation and Encryption
Data
Session
Interhost Communication
Segments
Transport
End-to-End Connections and Reliability
Packets
Network
Path Determination and IP (Logical Addressing)
Frames
Data Link
MAC and LLC (Phyiscal addressing)
Bits
Physical
Media, Signal, and Binary Transmission
Host Layers
Media Layers

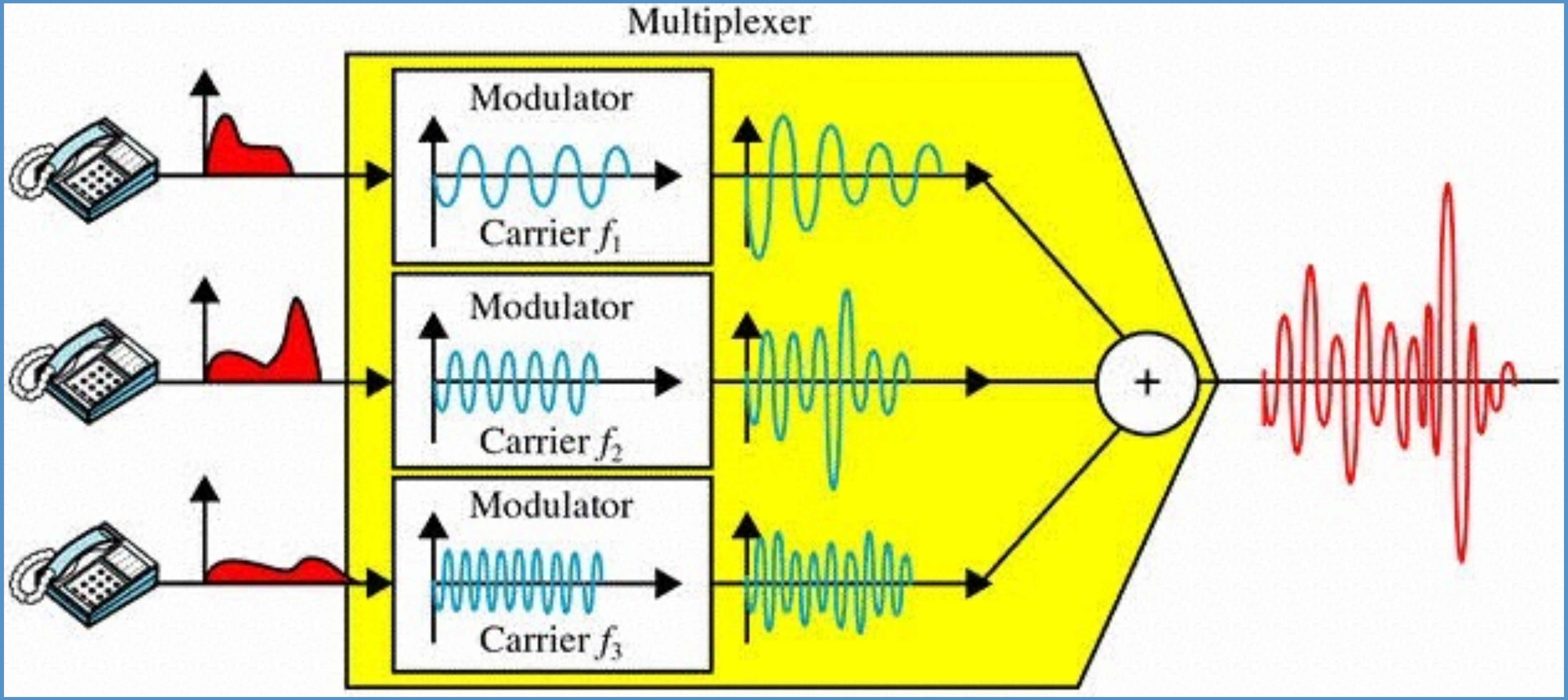
Layer 2: Link Layer

- provides node-to-node transfer of “frames”
- defines the protocol to establish and terminate a connection between two physically connected devices and the protocol for flow control between them
- detects and possibly corrects errors that may occur in the physical layer
- aka the Media Access Control (MAC) layer

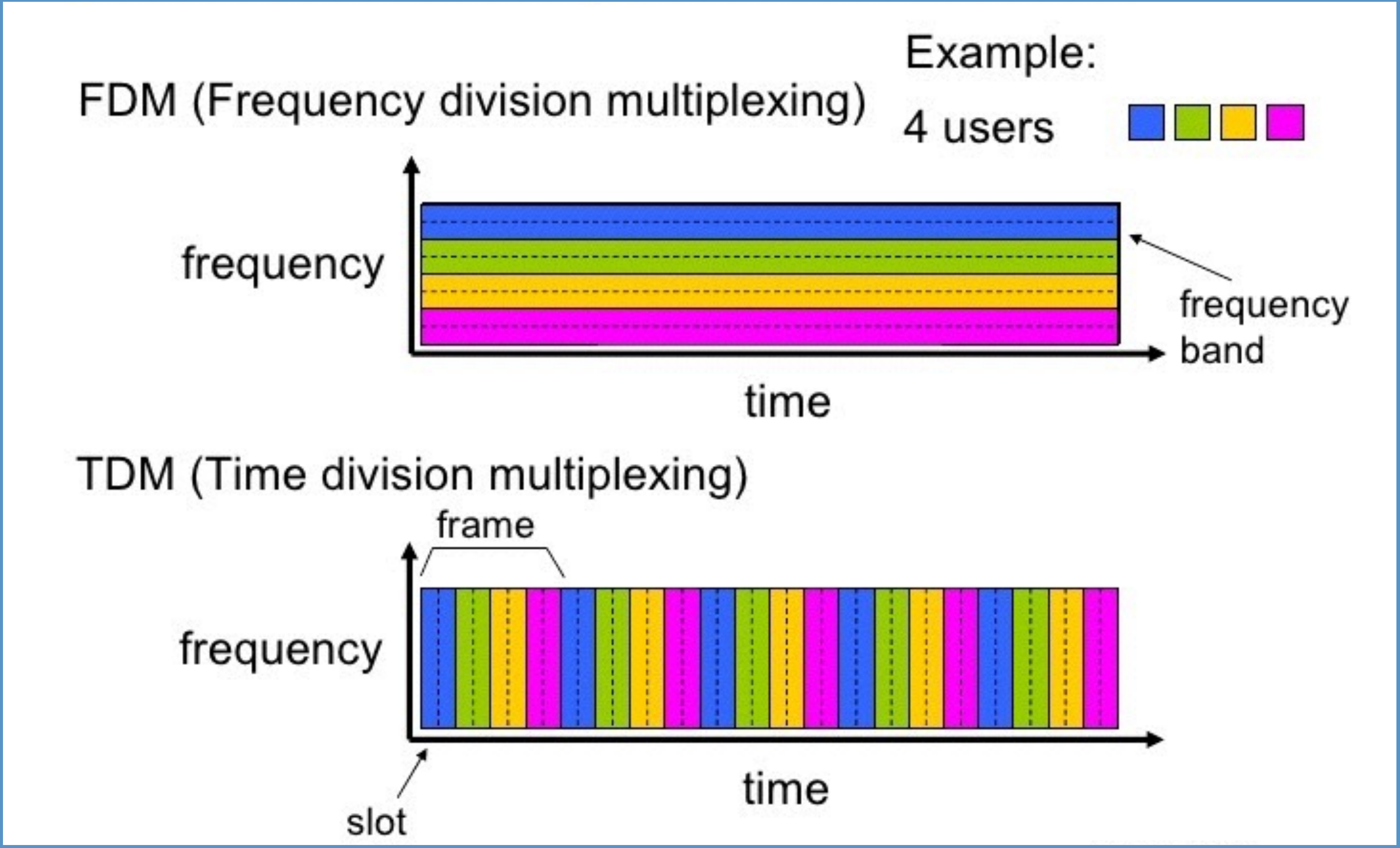
Time Division



Frequency Division



Time vs Frequency DM



Error Detection: Parity Bit

7 bits of data	(count of 1 bits)	8 bits including parity	
		even	odd
0000000	0	0000000**0**	0000000**1**
1010001	3	1010001**1**	1010001**0**
1101001	4	1101001**0**	1101001**1**
1111111	7	1111111**1**	1111111**0**

Error Detection: Parity Bit

Type of bit parity	Successful transmission scenario
Even parity	A wants to transmit: 1001 A computes parity bit value: 1+0+0+1 (mod 2) = 0 A adds parity bit and sends: 10010 B receives: 10010 B computes parity: 1+0+0+1+0 (mod 2) = 0 B reports correct transmission after observing expected even result.

Error Detection: Parity Bit

Type of bit parity error	Failed transmission scenario
Even parity Error in the second bit	A wants to transmit: 1001 A computes parity bit value: 1^0^0^1 = 0 A adds parity bit and sends: 10010 **...TRANSMISSION ERROR...** B receives: 1**1**010 B computes overall parity: 1^1^0^1^0 = 1 B reports incorrect transmission after observing unexpected odd result.

Error Detection: Parity Bit

Type of bit parity error	Failed transmission scenario
Even parity Two corrupted bits	A wants to transmit: 1001 A computes even parity value: 1^0^0^1 = 0 A sends: 10010 **...TRANSMISSION ERROR...** B receives: 1**1**01**1** B computes overall parity: 1^1^0^1^1 = 0 B reports correct transmission though actually incorrect.

Error Correction: (3,1) Repetition

Triplet received	Interpreted as
000	0 (error free)
001	0
010	0
100	0
111	1 (error free)
110	1
101	1
011	1

Layer 2: MAC Address

- Each Network Interface Controller (NIC) has its own 6-byte MAC address which physically identifies itself
- Although intended to be a permanent and globally unique identification, it is possible to change the MAC address on most modern hardware
- Changing MAC addresses is necessary in network virtualization
- It can also be used in the process of exploiting security vulnerabilities. This is called MAC spoofing.
- According to Edward Snowden, the NSA has a system that tracks the movements of everyone in a city by monitoring the MAC addresses of their electronic devices. As a result of users being trackable by their devices' MAC addresses.

Layer 2: MAC Address

via 'ifconfig'

```
~ ifconfig
lo0: flags=8049<UP,LOOPBACK,RUNNING,MULTICAST> mtu 16384
        options=3<RXCSUM,TXCSUM>
        inet6 ::1 prefixlen 128
        inet 127.0.0.1 netmask 0xff000000
        inet6 fe80::1%lo0 prefixlen 64 scopeid 0x1
        nd6 options=1<PERFORMNUD>
gif0: flags=8010<POINTOPOINT,MULTICAST> mtu 1280
stf0: flags=0<> mtu 1280
en0: flags=8863<UP,BROADCAST,SMART,RUNNING,SIMPLEX,MULTICAST> mtu 1500
        ether a4:5e:60:e9:a7:85
        inet6 fe80::a65e:60ff:fee9:a785%en0 prefixlen 64 scopeid 0x4
        inet 172.19.131.92 netmask 0xffffff00 broadcast 172.19.131.255
        nd6 options=1<PERFORMNUD>
        media: autoselect
        status: active
en1: flags=8963<UP,BROADCAST,SMART,RUNNING,PROMISC,SIMPLEX,MULTICAST> mtu 1500
        options=60<TSO4,TSO6>
```

Layer 2: Exercise

- The following bit sequence was sent over an unreliable channel using a (3,1) repetition sequence:

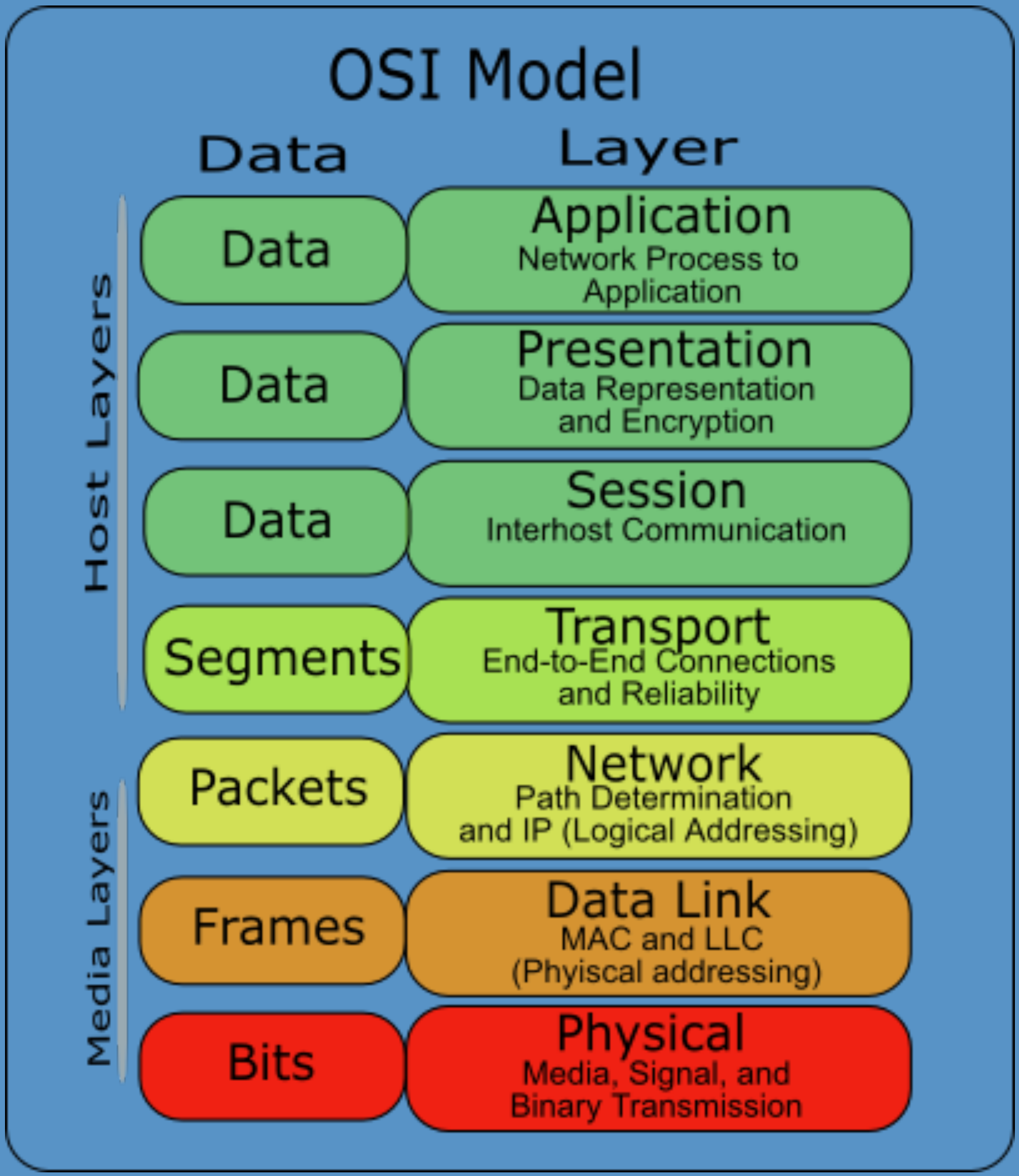
```
001 111 000 000 000 100 101 111

000 000 110 111 000 111 001 010

000 111 000 011 000 000 100 111
```

Recover the transmitted bit sequence and convert using ASCII:

http://www.binaryhexconverter.com/binary-to-ascii-text-converter

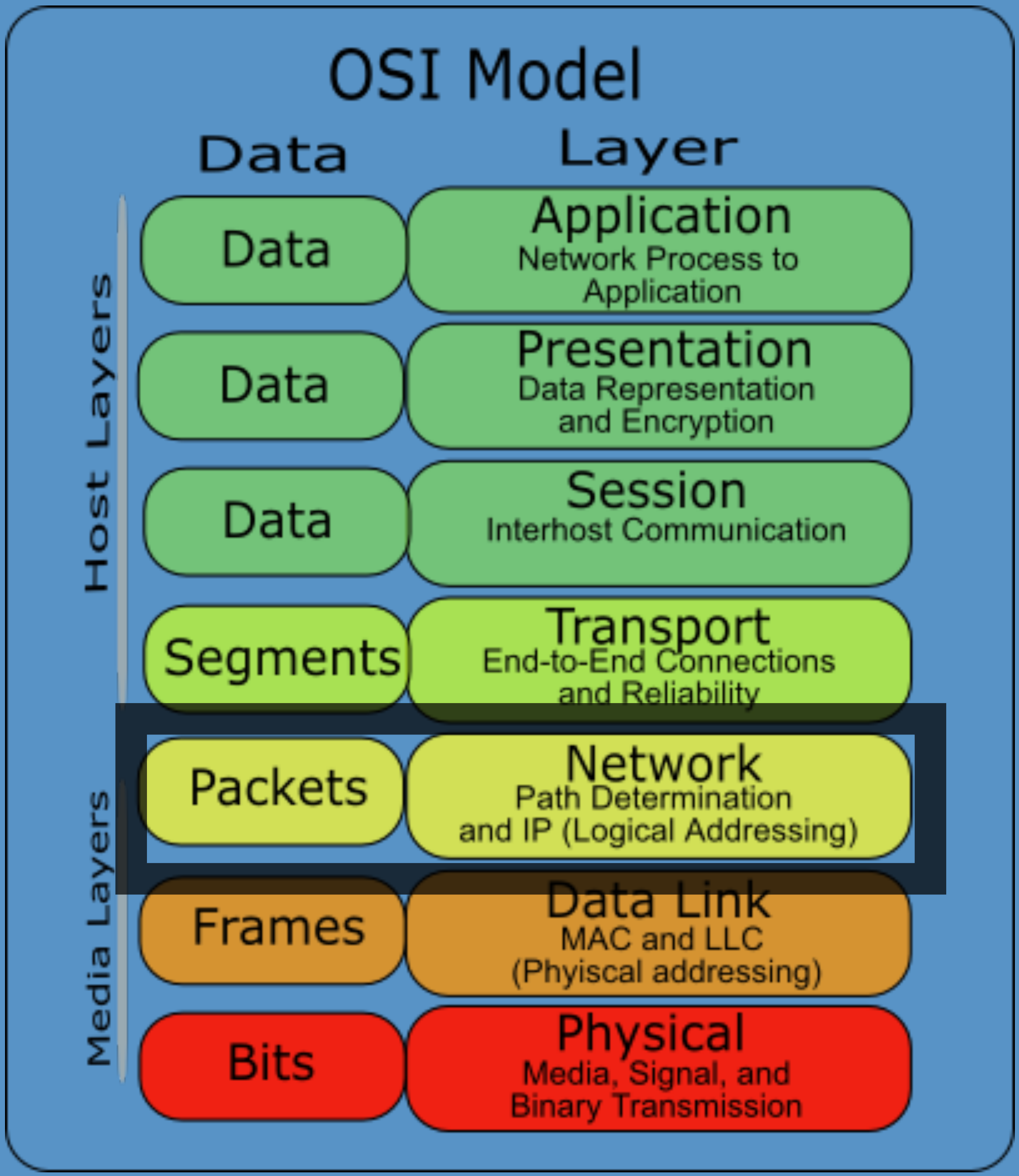
Layer 3

Layer 3



OSI Model
Data
Layer
Host Layers
Media Layers
Data
Application
Network Process to Application
Data
Presentation
Data Representation and Encryption
Data
Session
Interhost Communication
Segments
Transport
End-to-End Connections and Reliability
Packets
Network
Path Determination and IP (Logical Addressing)
Frames
Data Link
MAC and LLC (Phyiscal addressing)
Bits
Physical
Media, Signal, and Binary Transmission

Layer 3

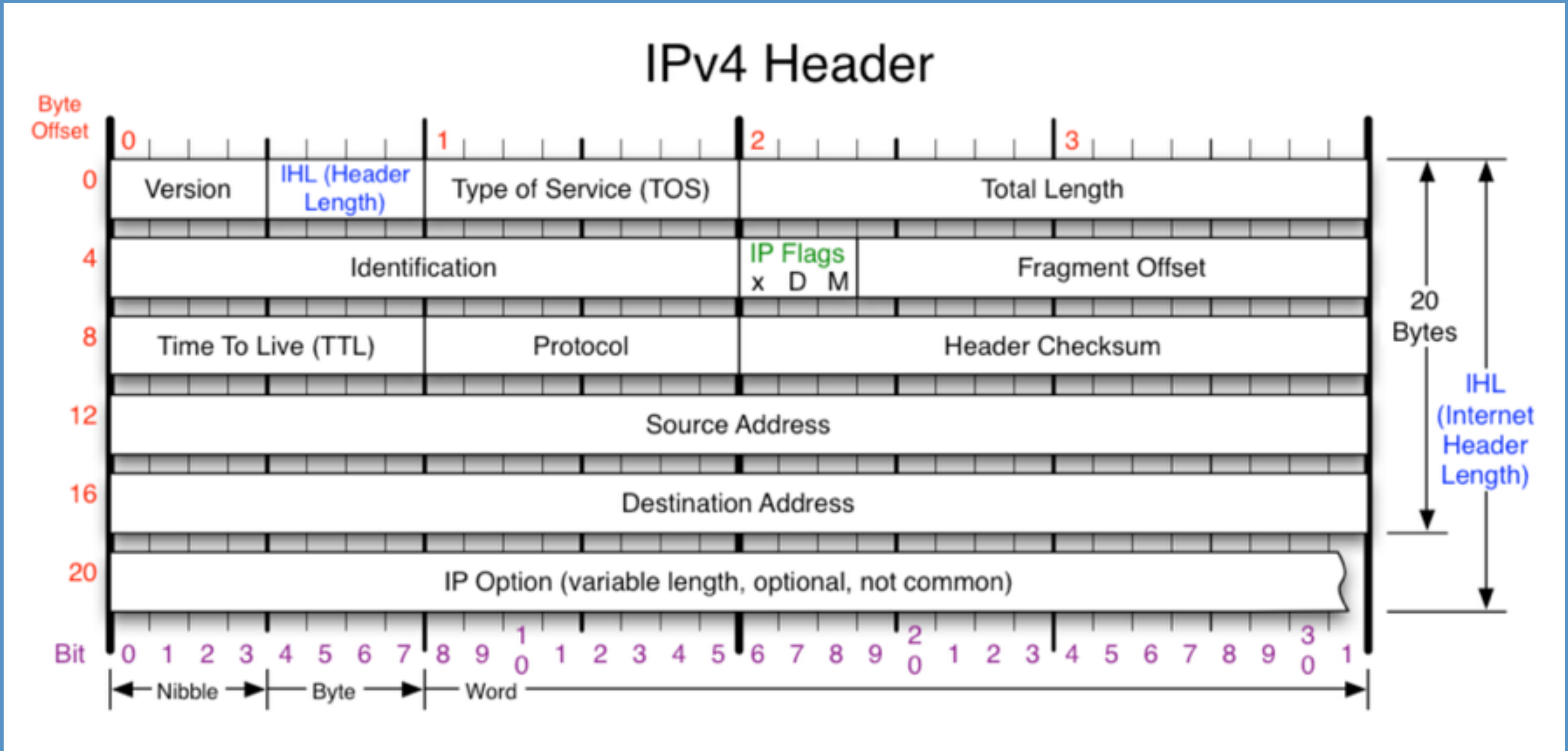


OSI Model
Data
Layer
Host Layers
Media Layers
Data
Application
Network Process to Application
Data
Presentation
Data Representation and Encryption
Data
Session
Interhost Communication
Segments
Transport
End-to-End Connections and Reliability
Packets
Network
Path Determination and IP (Logical Addressing)
Frames
Data Link
MAC and LLC (Phyiscal addressing)
Bits
Physical
Media, Signal, and Binary Transmission

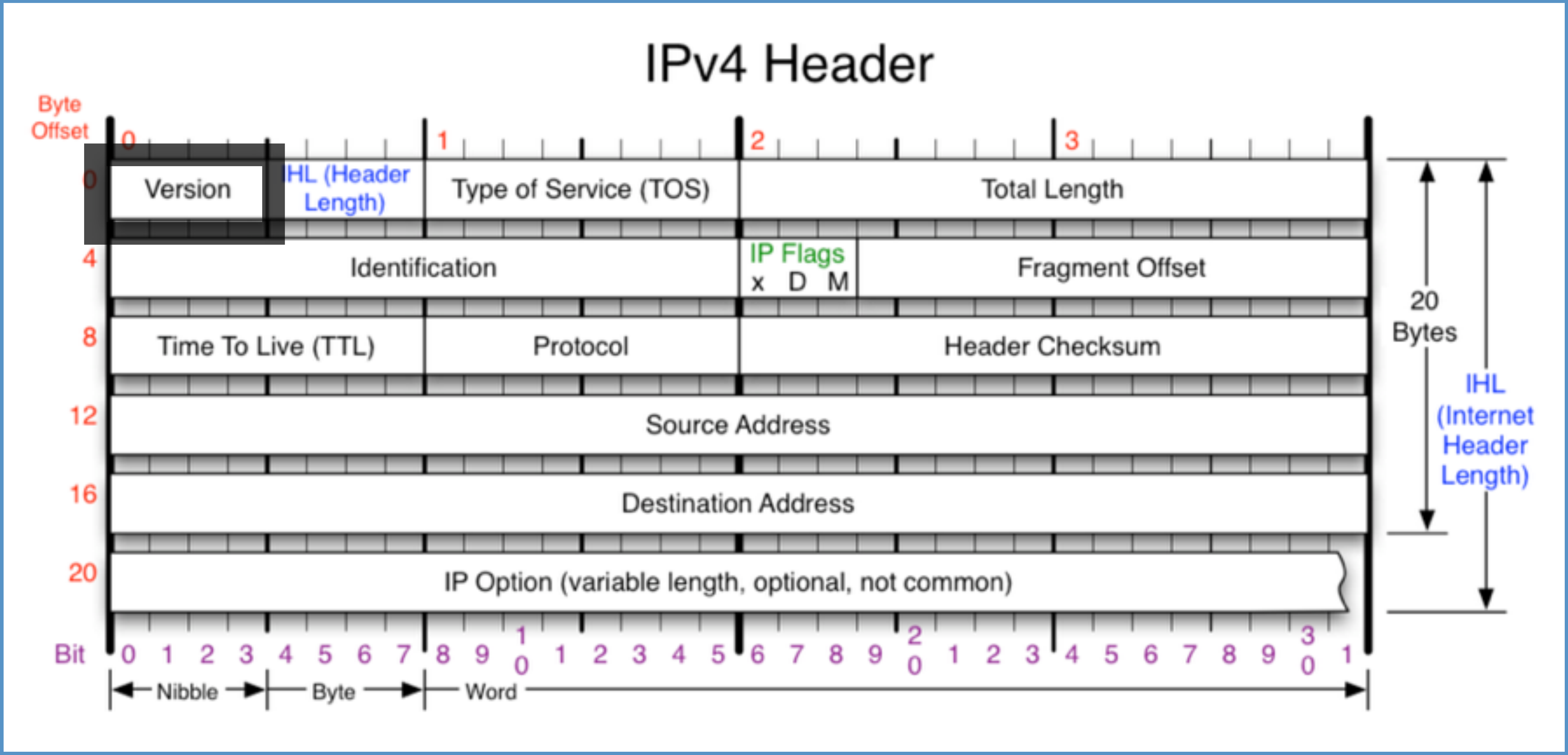
Layer 3: Network Layer

- Enables the routing of “packets” from one system to system
- IP Addresses (IPv4, IPv6)
- “Best Effort” but not guaranteed
- IPv4 uses 32-bit (4-byte) addresses (2^{32} = 4,294,967,296)

Layer 3: Network Layer

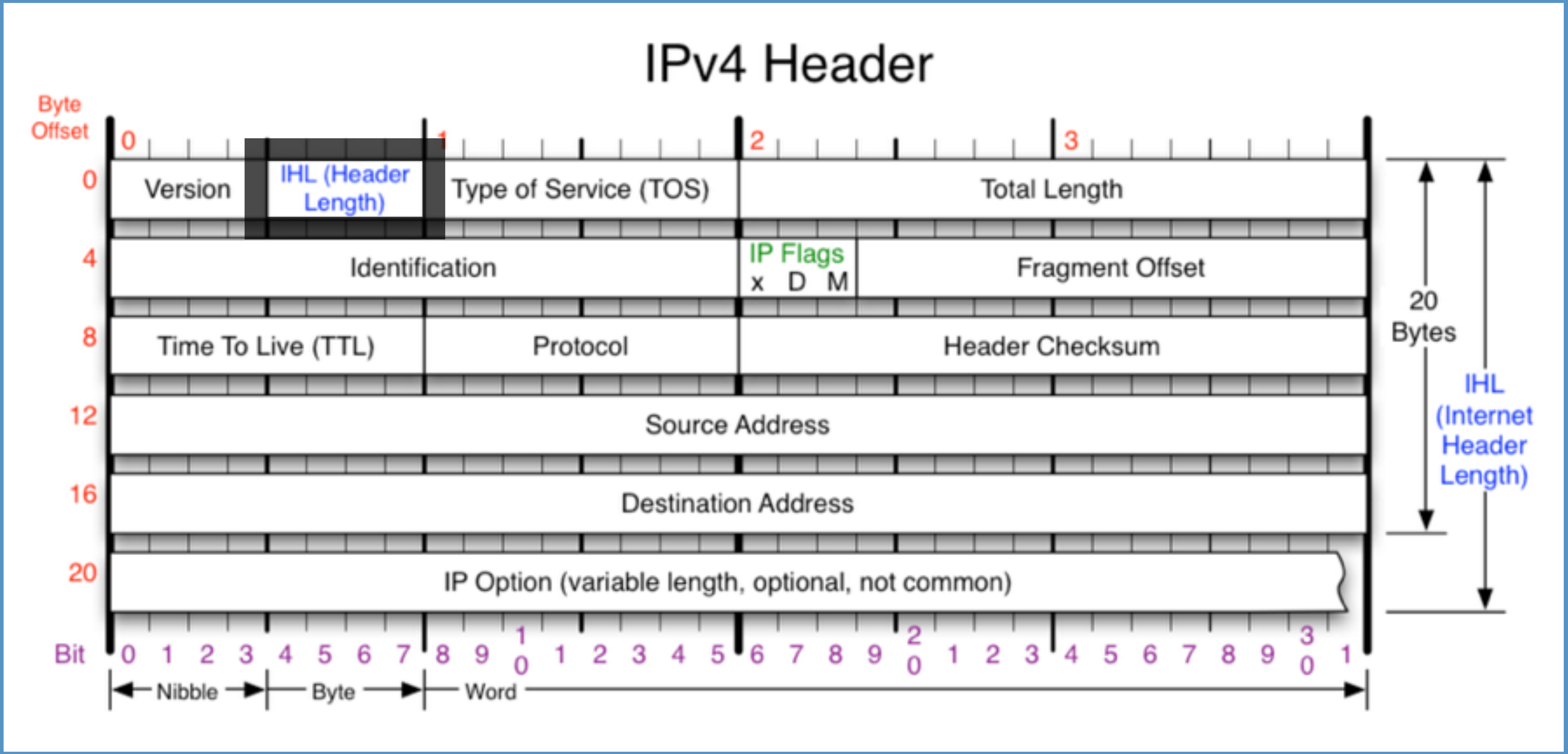


Layer 3: Network Layer



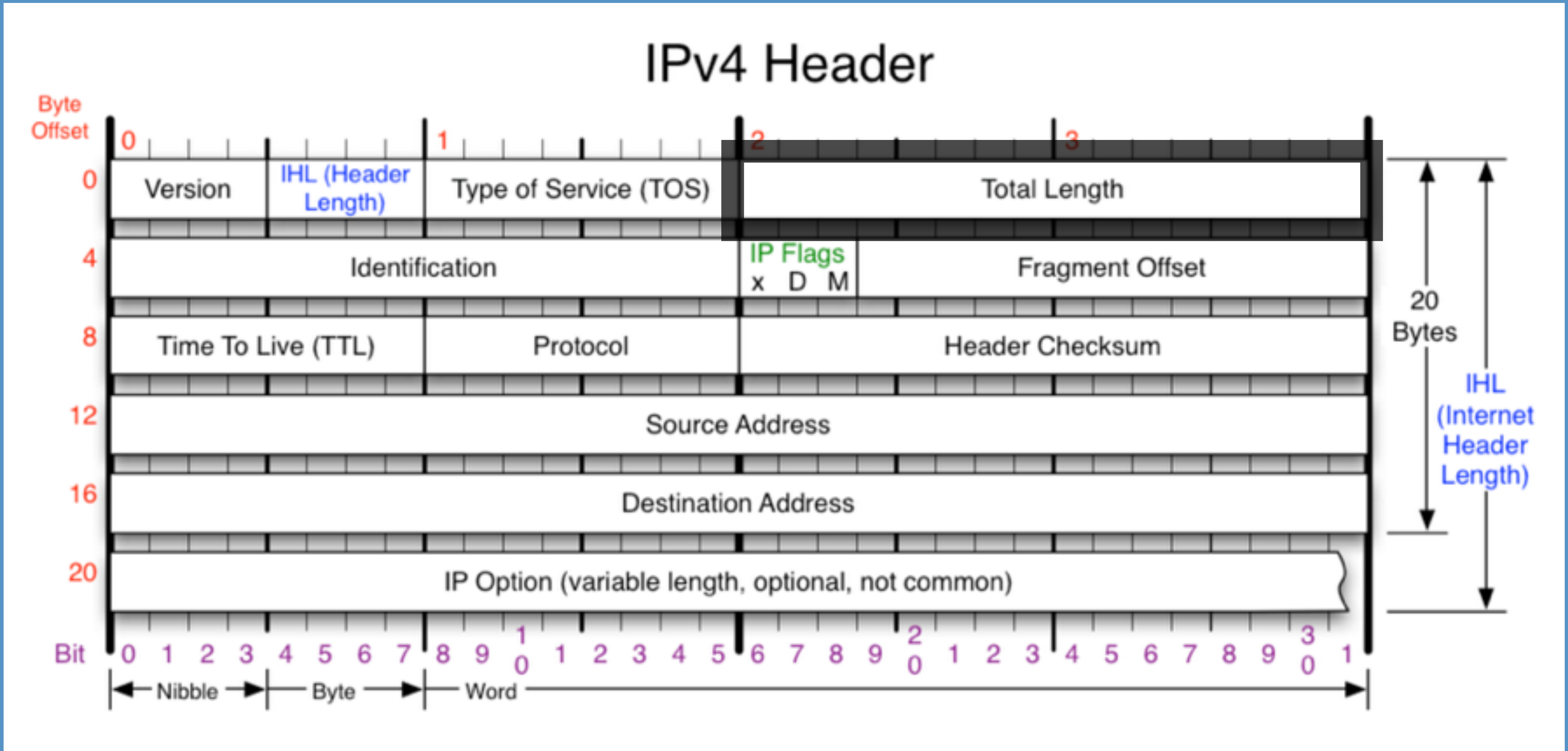
Version: 4 (IPv4) or 6 (IPv6)

Layer 3: Network Layer



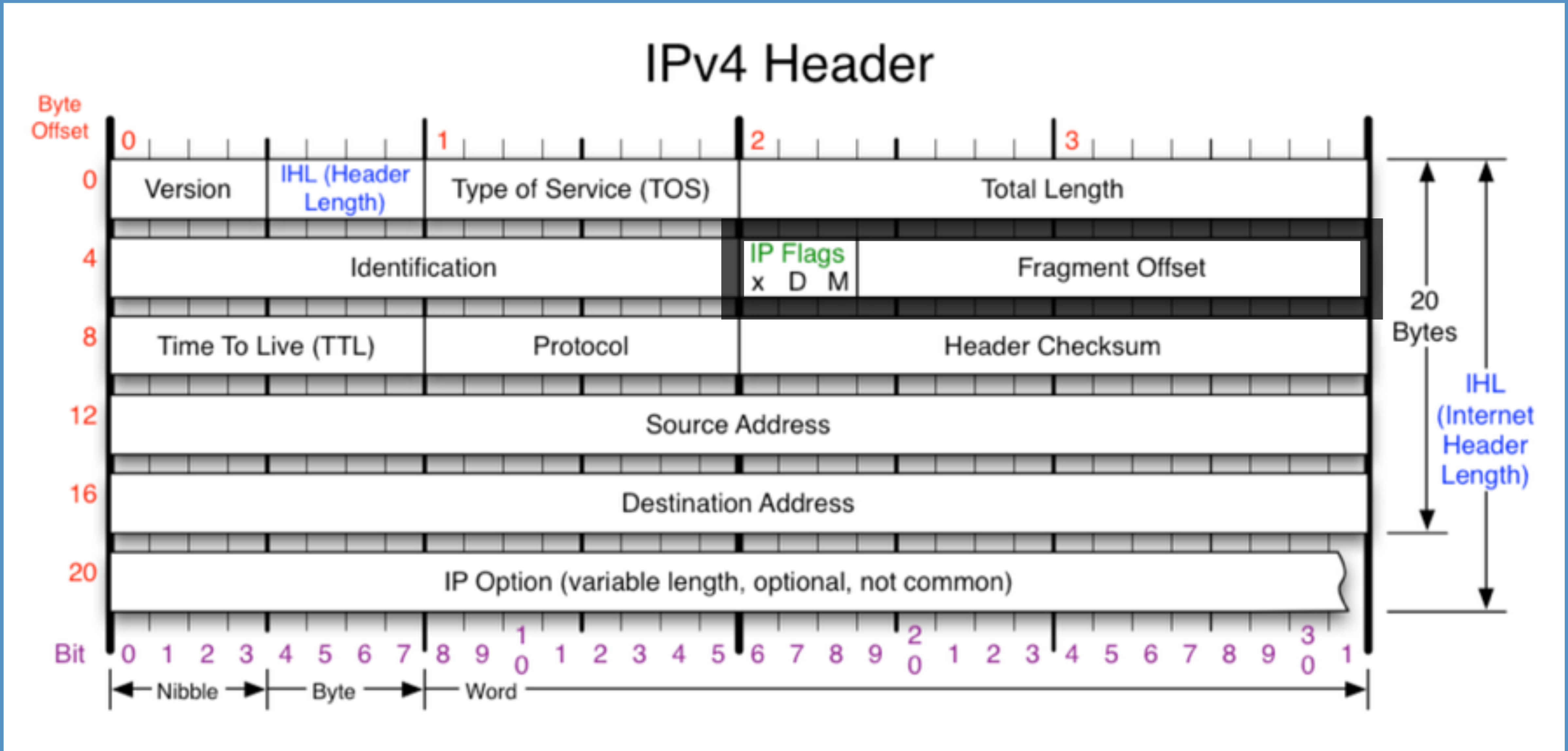
Internet Header Length: # of 32-bit *words* in header

Layer 3: Network Layer



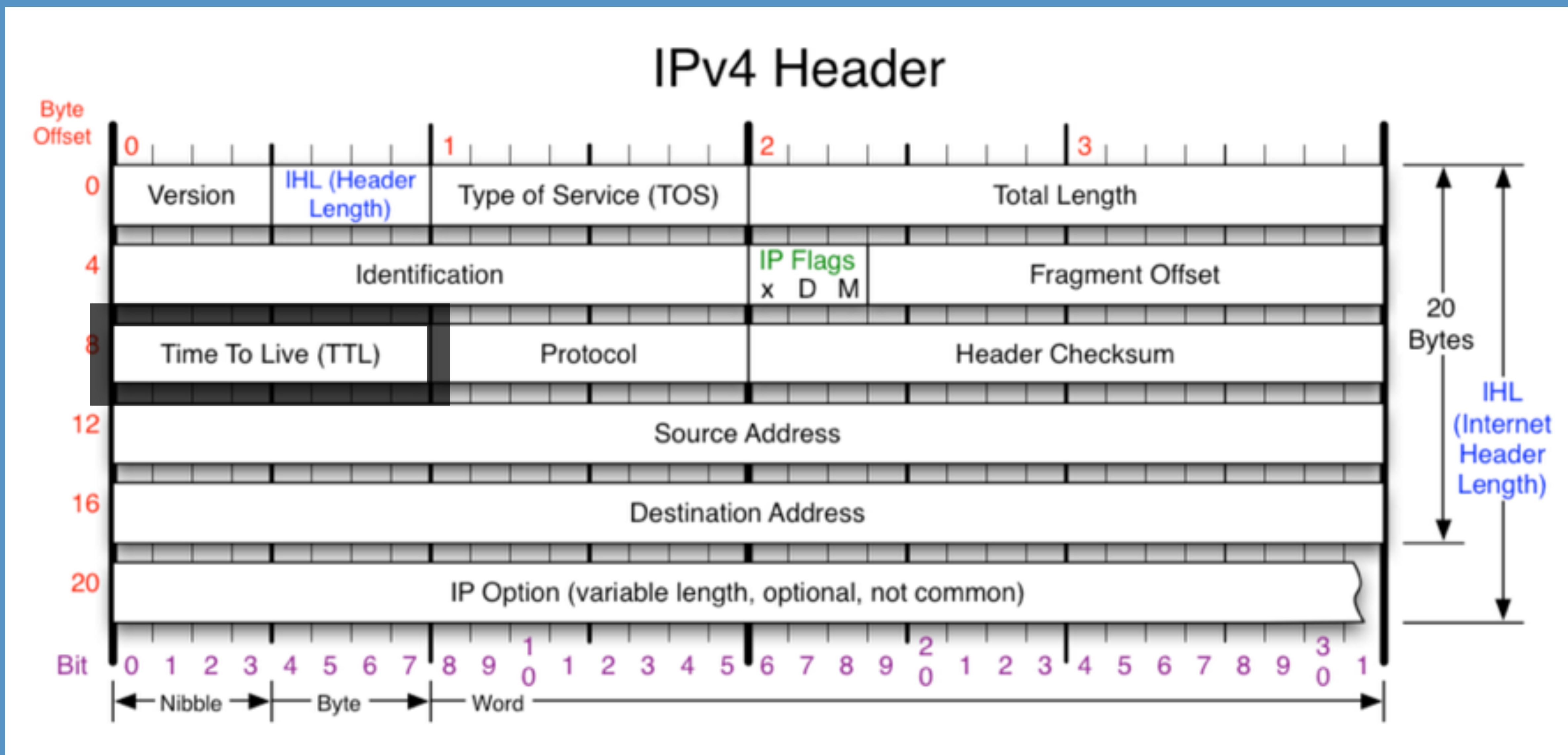
Total Length: entire packet size, *in bytes*

Layer 3: Network Layer



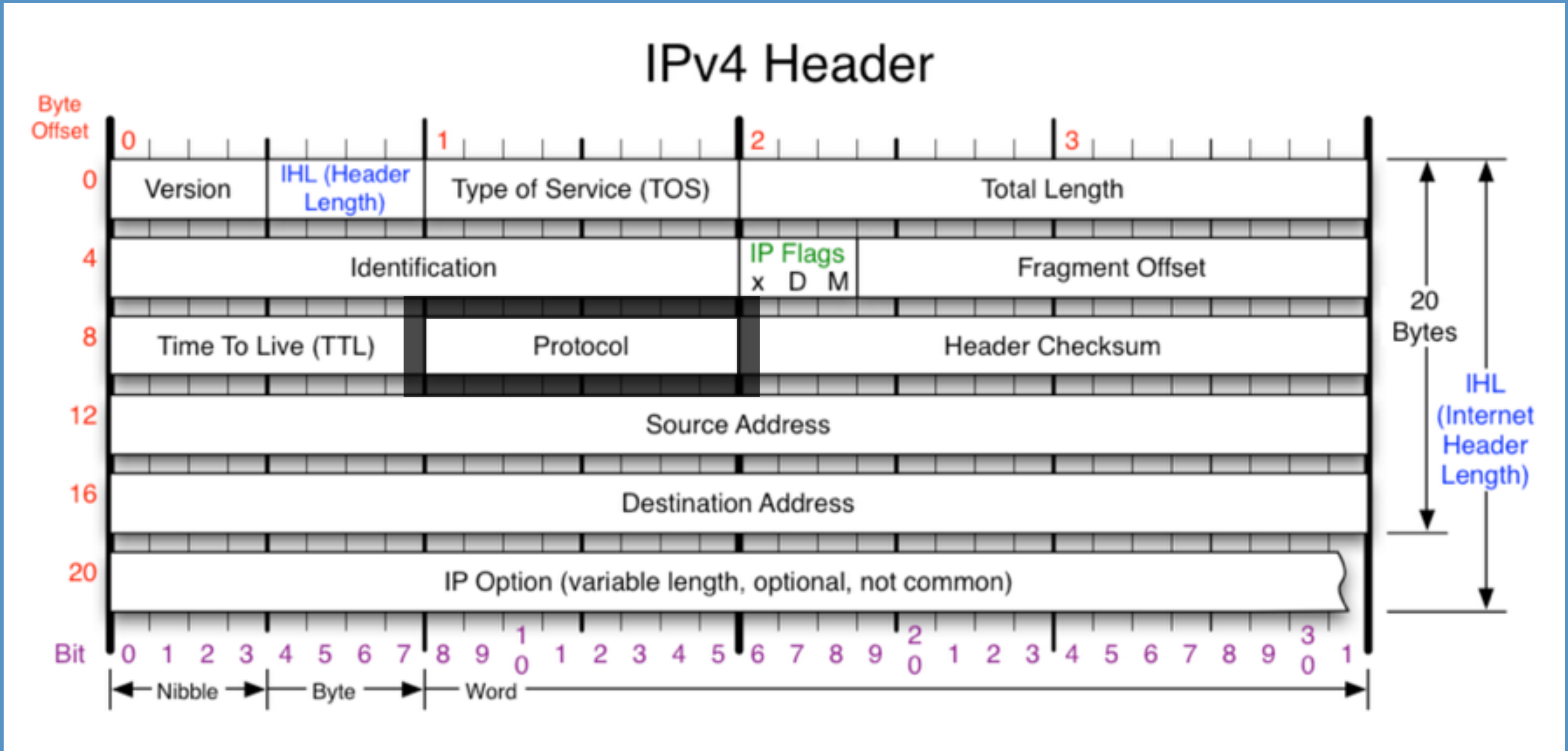
Fragmentation: *more on this*

Layer 3: Network Layer



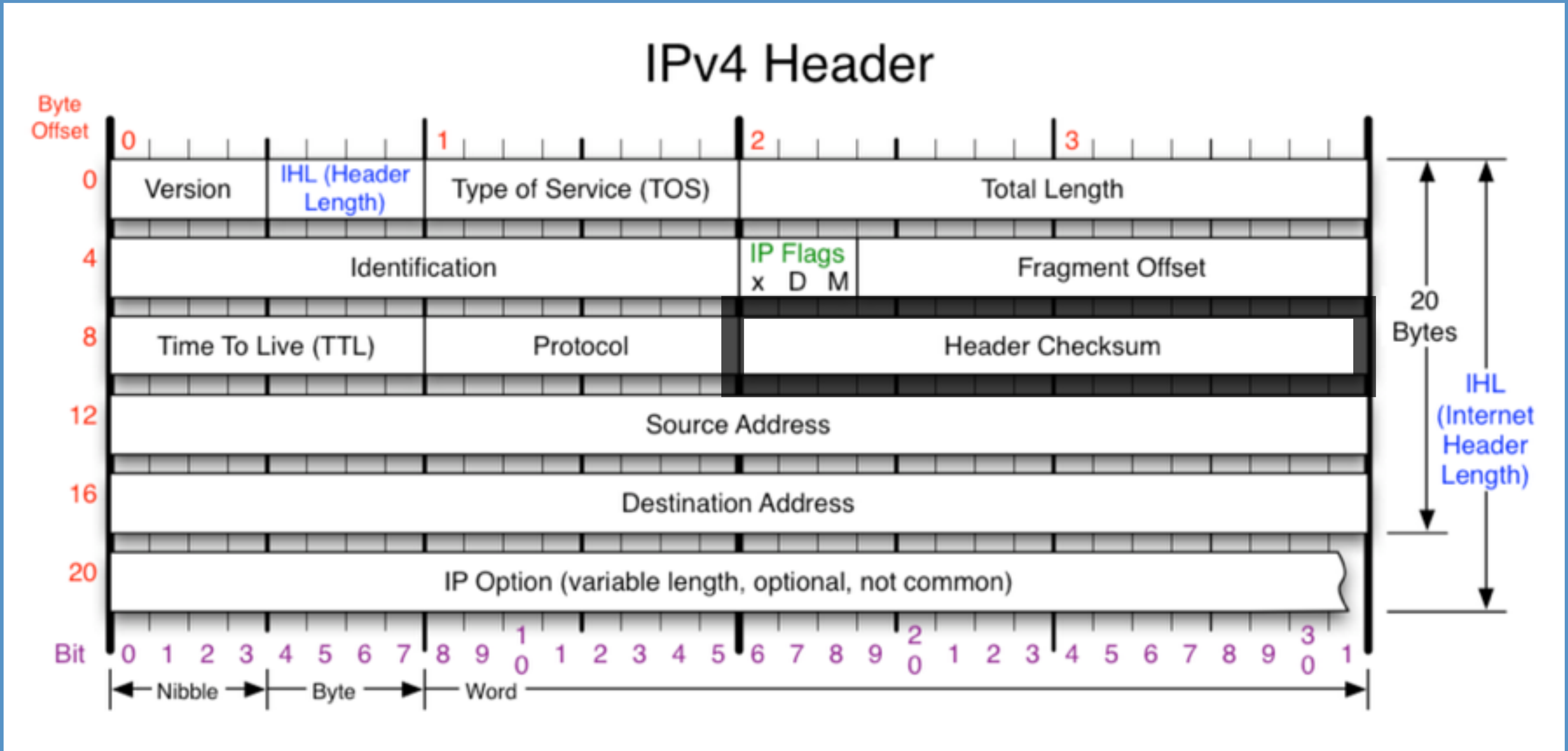
Time to Live: every hop on network decrements by 1

Layer 3: Network Layer



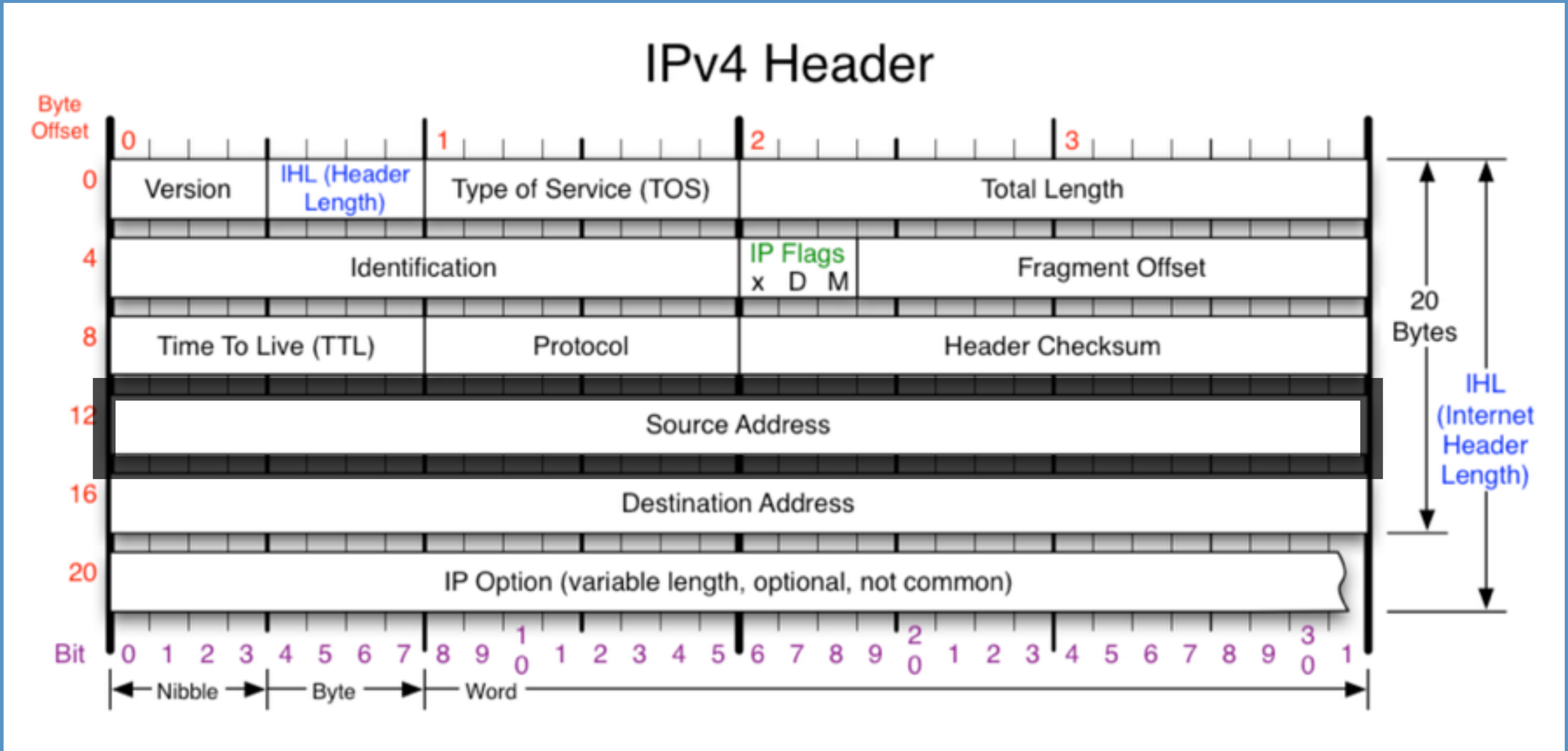
Protocol: defines the next layer in the ISO model…

Layer 3: Network Layer



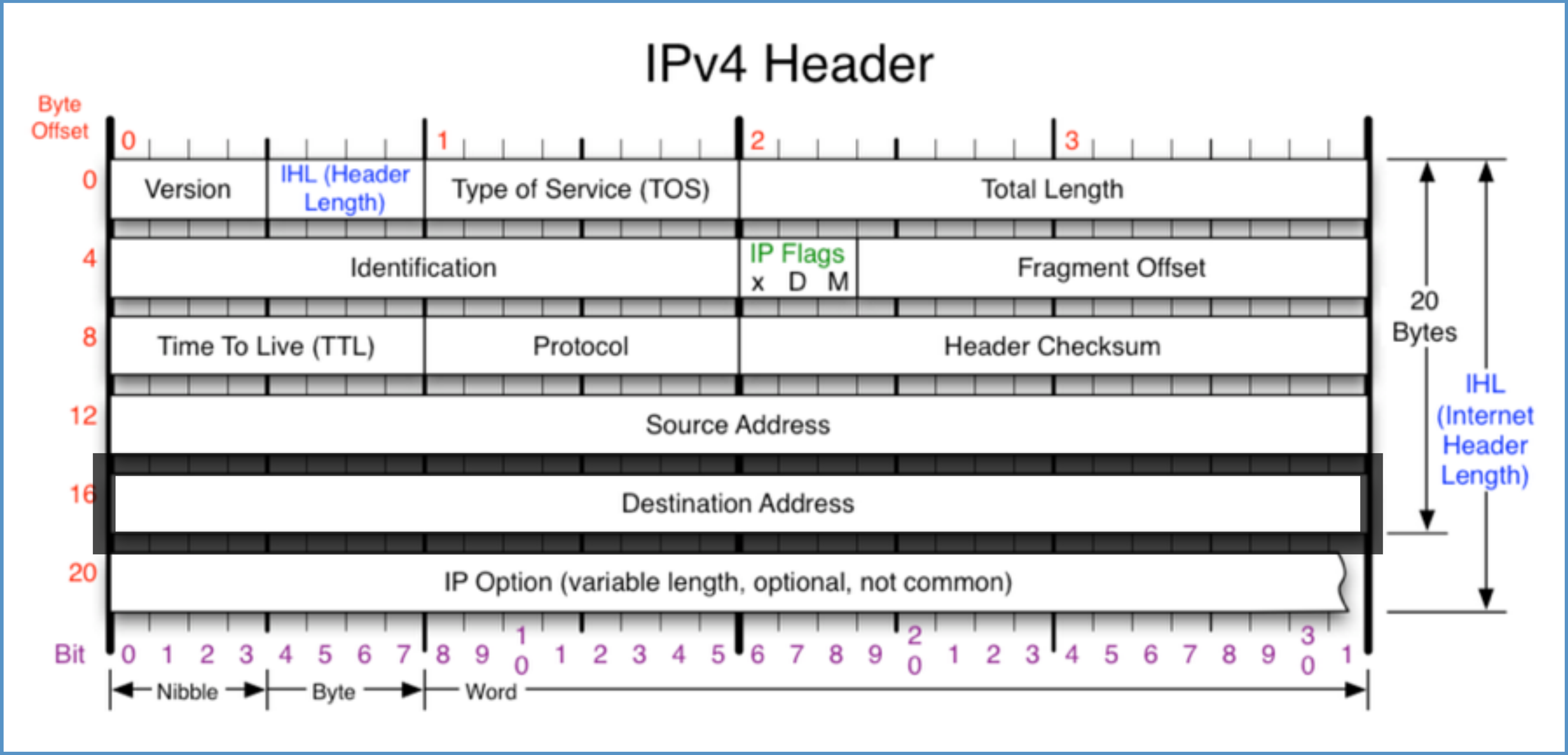
Checksum: only for *header*, upper layer handles *data*

Layer 3: Network Layer



Source: IP address of sender (may change)

Layer 3: Network Layer



Destination: IP address of receiver (may change)

IPv4 vs IPv6

- IPv4 first deployed as part of the ARPANET in 1983
- It was apparent that the pool of available IPv4 addresses was being depleted at a rate that was not initially anticipated in the original design of the network address system
- Some factors which led to IPv4 address exhaustion:
  - Rapidly growing number of Internet users
  - Always-on devices — ADSL modems, cable modems
  - Mobile devices — laptop computers, PDAs, mobile phones

IPv4 vs IPv6

- The threat of exhaustion was the motivation for remedial technologies. Included are:
  - Network address translation (NAT)
  - Dynamic Host Configuration Protocol (DHCP)
  - Tighter control by regional Internet registries over the allocation of addresses to local Internet registries
  - Network renumbering to reclaim large blocks of address space allocated in the early days of the Internet

IPv4 vs IPv6

- This limitation stimulated the development of IPv6 in the 1990s, which has been in commercial deployment since 2006
- IPv4 uses 32-bit (4-byte) addresses (2^{32} = 4,294,967,296)
- IPv6 uses 128-bit (16-byte) addresses (2^{128} = 3.4 x 10^{38})
- IPv4 finally suffered exhaustion on February 3, 2011

IPv4 vs IPv6

via ‘ifconfig’

```
~ ifconfig
lo0: flags=8049<UP,LOOPBACK,RUNNING,MULTICAST> mtu 16384
        options=3<RXCSUM,TXCSUM>
        inet6 ::1 prefixlen 128
        inet 127.0.0.1 netmask 0xff000000
        inet6 fe80::1%lo0 prefixlen 64 scopeid 0x1
        nd6 options=1<PERFORMNUD>
gif0: flags=8010<POINTOPOINT,MULTICAST> mtu 1280
stf0: flags=0<> mtu 1280
en0: flags=8863<UP,BROADCAST,SMART,RUNNING,SIMPLEX,MULTICAST> mtu 1500
        ether a4:5e:60:e9:a7:85
        inet6 fe80::a65e:60ff:fee9:a785%en0 prefixlen 64 scopeid 0x4
        inet 172.19.131.92 netmask 0xffffff00 broadcast 172.19.131.255
        nd6 options=1<PERFORMNUD>
        media: autoselect
        status: active
en1: flags=8963<UP,BROADCAST,SMART,RUNNING,PROMISC,SIMPLEX,MULTICAST> mtu 1500
        options=60<TSO4,TSO6>
```

IPv4 vs IPv6

via ‘ifconfig’

```
~ ifconfig
lo0: flags=8049<UP,LOOPBACK,RUNNING,MULTICAST> mtu 16384
        options=3<RXCSUM,TXCSUM>
        inet6 ::1 prefixlen 128
        inet 127.0.0.1 netmask 0xff000000
        inet6 fe80::1%lo0 prefixlen 64 scopeid 0x1
        nd6 options=1<PERFORMNUD>
gif0: flags=8010<POINTOPOINT,MULTICAST> mtu 1280
stf0: flags=0<> mtu 1280
en0: flags=8863<UP,BROADCAST,SMART,RUNNING,SIMPLEX,MULTICAST> mtu 1500
        ether a4:5e:60:e9:a7:85
        inet6 fe80::a65e:60ff:fee9:a785%en0 prefixlen 64 scopeid 0x4
        inet 172.19.131.92 netmask 0xffffff00 broadcast 172.19.131.255
        nd6 options=1<PERFORMNUD>
        media: autoselect
        status: active
en1: flags=8963<UP,BROADCAST,SMART,RUNNING,PROMISC,SIMPLEX,MULTICAST> mtu 1500
        options=60<TSO4,TSO6>
```

Subnets

- IP addresses can be divided into ranges
- Each address in the range is composed of a *network prefix* and a *host identifier.* A *subnet mask* designates which bits belong to the network prefix
- Network Prefix = IP address & subnet mask
- Host Identifier = IP address & ~subnet mask
- Format:
  - 192.168.5.130/24

Subnets

- IP addresses can be divided into ranges
- Each address in the range is composed of a *network prefix* and a *host identifier.* A *subnet mask* designates which bits belong to the network prefix
- Network Prefix = IP address & subnet mask
- Host Identifier = IP address & ~subnet mask
- Common format: Mask
  - 192.168.5.130/**24**

Example

Applying the Subnet Mask

A device with address 192.0.0.1 belongs to network 192.0.0.0

	High order bits Prefix /16		Low order bits	
	192 .	0 .	0 .	1
Host Address	11000000	00000000	00000000	00000001
Subnet Mask	255	255	0	0
	11111111	11111111	00000000	00000000
Network Address	11000000	00000000	00000000	00000000
Network	192 .	0 .	0 .	0

Example

via ‘ifconfig’

```
~ ifconfig
lo0: flags=8049<UP,LOOPBACK,RUNNING,MULTICAST> mtu 16384
        options=3<RXCSUM,TXCSUM>
        inet6 ::1 prefixlen 128
        inet 127.0.0.1 netmask 0xff000000
        inet6 fe80::1%lo0 prefixlen 64 scopeid 0x1
        nd6 options=1<PERFORMNUD>
gif0: flags=8010<POINTOPOINT,MULTICAST> mtu 1280
stf0: flags=0<> mtu 1280
en0: flags=8863<UP,BROADCAST,SMART,RUNNING,SIMPLEX,MULTICAST> mtu 1500
        ether a4:5e:60:e9:a7:85
        inet6 fe80::a65e:60ff:fee9:a785%en0 prefixlen 64 scopeid 0x4
        inet 172.19.131.92 netmask 0xffffff00 broadcast 172.19.131.255
        nd6 options=1<PERFORMNUD>
        media: autoselect
        status: active
en1: flags=8963<UP,BROADCAST,SMART,RUNNING,PROMISC,SIMPLEX,MULTICAST> mtu 1500
        options=60<TSO4,TSO6>
```

Special Addresses

- Loopback
  - 127.0.0.0/8 (IPv4)
  - ::1 (IPv6)
- Private Addresses
  - 192.168.0.0/16 (IPv4)
  - fc00::/7 (IPv6)
- Broadcast
  - 255.255.255.255 (IPv4)
  - N/A (IPv6)

Exercise

- Using the output in *ifconfig*, determine the valid IPv4 address range for the CQ4 subnet

DNS

- IP address => routing and network interface identification
- Domain name => host on the Internet, e.g., www.example.com
- The use of domain names requires translating, called resolving, them to addresses and vice versa
- The translation between addresses and domain names is performed by the Domain Name System (DNS)
- DNS is a hierarchical, distributed naming system which allows for sub-delegation of name spaces to other DNS servers

ARP

- maps IP to MAC address
- sender: whose MAC address belongs to this IP?
- “going down” the ISO model

ping

- uses Internet Control Message Protocol (ICMP)
- sender: ICMP ECHO (notes local time)
- receiver: ICMP ECHO REPLY
- sender: (computes time difference)

traceroute

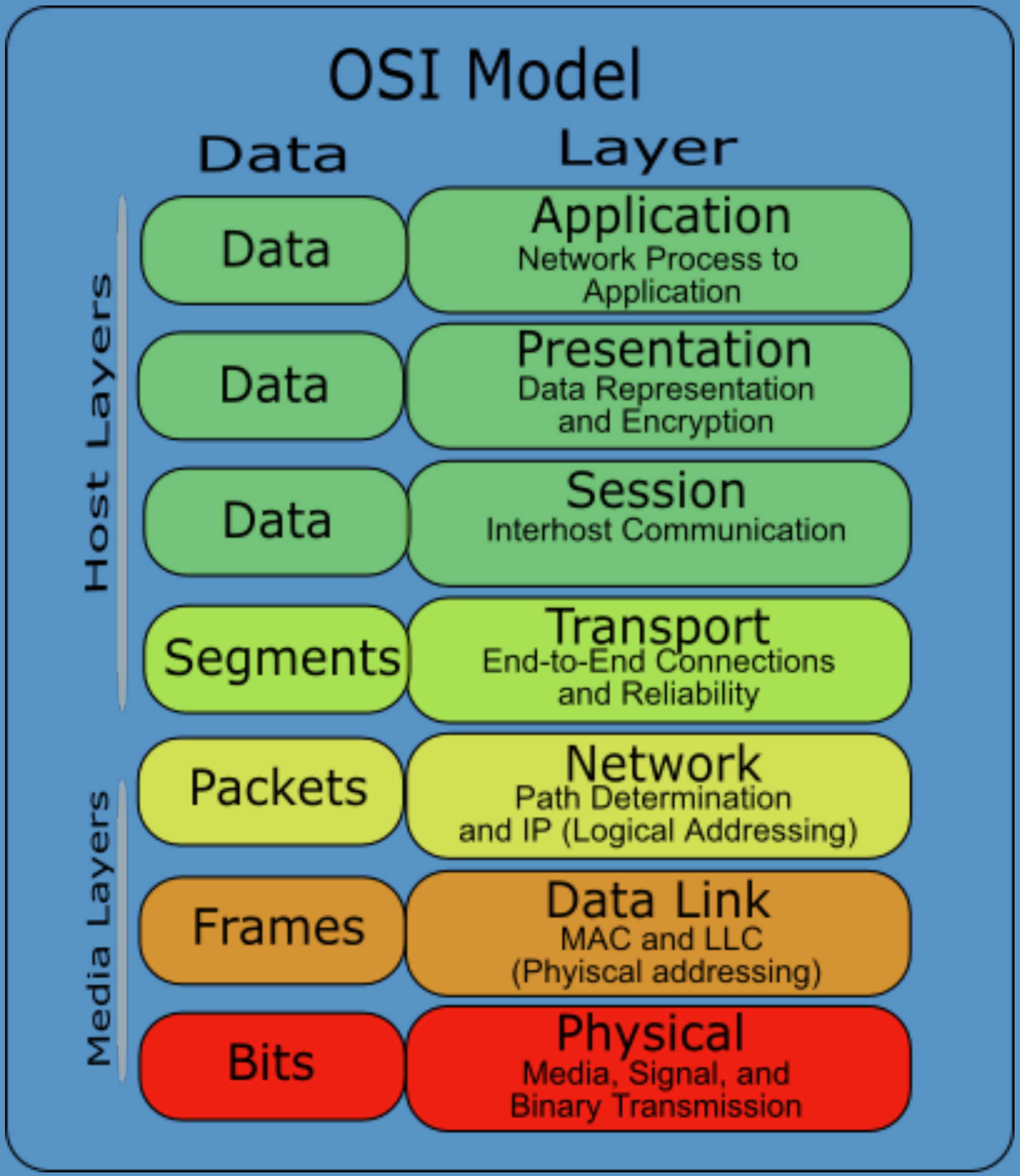
- uses TTL field in packet
- each “hop” on network reduces TTL
- when 0, packet is dropped and ICMP Time Exceeded sent back to sender
- else if destination, packet dropped

Exercises

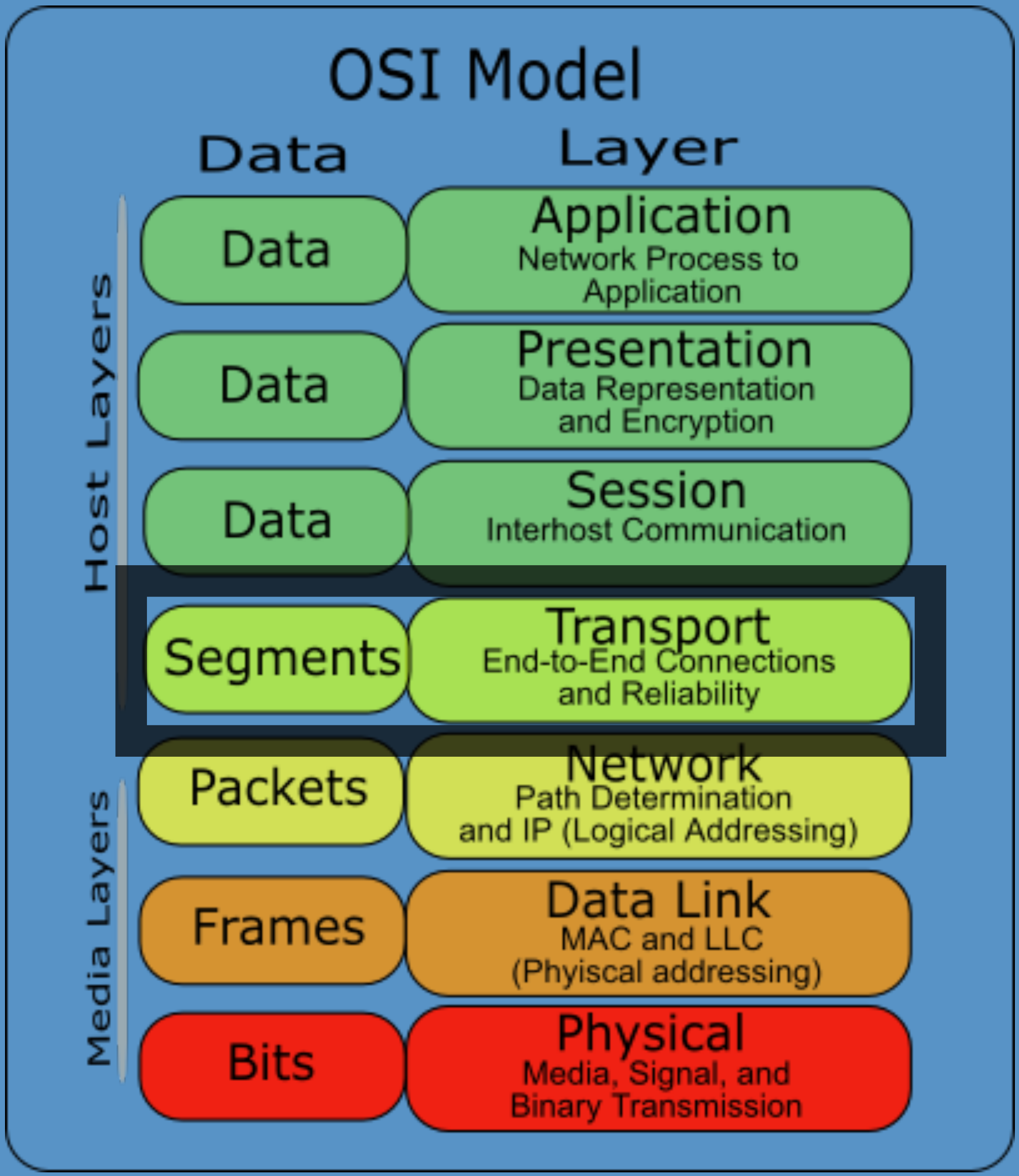
- Using the output in *dig*, resolve ‘c4q.nyc’ to its IP address
  - Hint: A records map to IPv4 records
- Using the output of “*arp -an*”, determine whose IP and MAC addresses are in your ARP cache
- Ping www.google.com and www.tajmahal.gov.in. Note the difference in roundtrip times.
- Traceroute www.stanford.edu, www.tajmahal.gov.in, www.google.com and other sites and note the number of hops.

Layer 4

Layer 4


OSI Model
Data
Layer
Host Layers
Media Layers
Data
Application
Network Process to Application
Data
Presentation
Data Representation and Encryption
Data
Session
Interhost Communication
Segments
Transport
End-to-End Connections and Reliability
Packets
Network
Path Determination and IP (Logical Addressing)
Frames
Data Link
MAC and LLC (Phyiscal addressing)
Bits
Physical
Media, Signal, and Binary Transmission

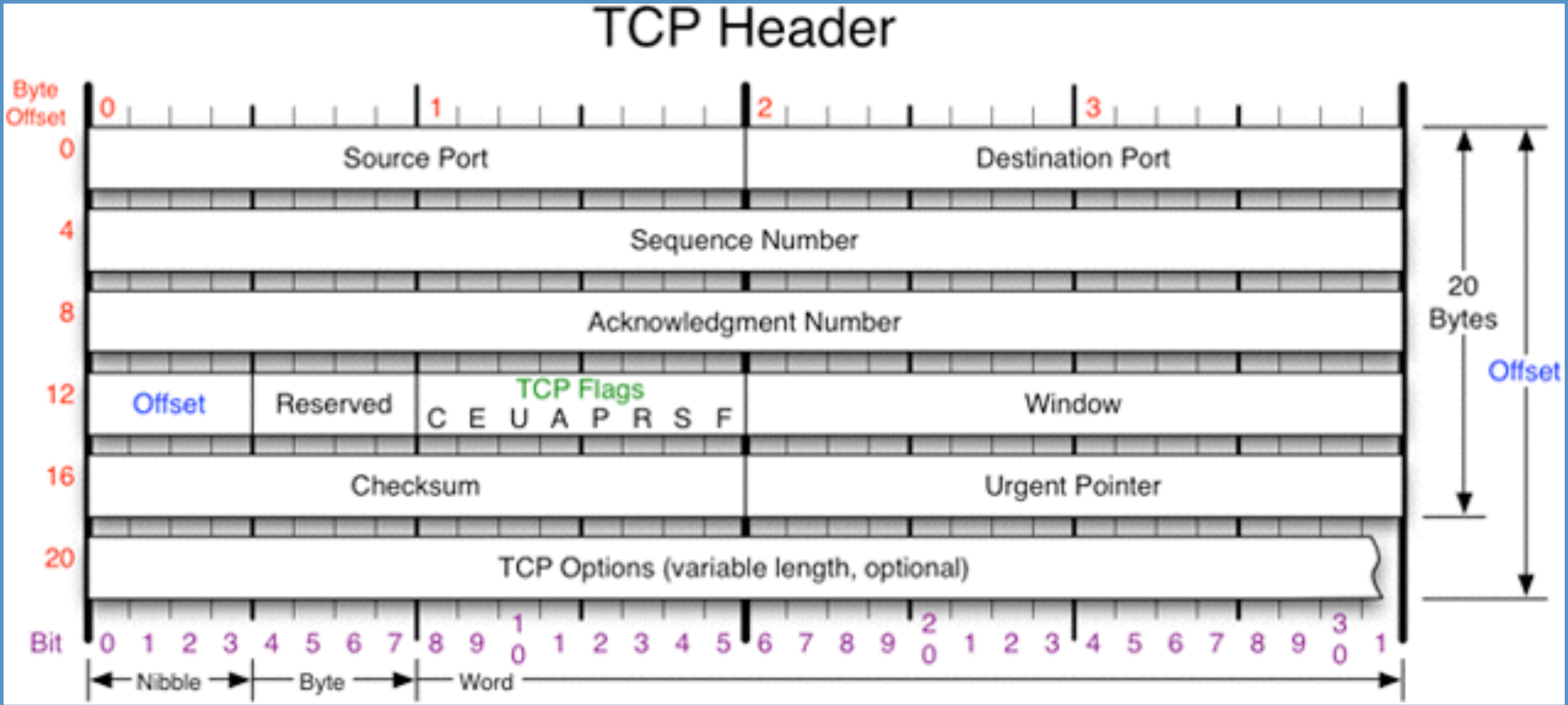
Layer 4


OSI Model
Data
Layer
Host Layers
Media Layers
Data
Application
Network Process to Application
Data
Presentation
Data Representation and Encryption
Data
Session
Interhost Communication
Segments
Transport
End-to-End Connections and Reliability
Packets
Network
Path Determination and IP (Logical Addressing)
Frames
Data Link
MAC and LLC (Phyiscal addressing)
Bits
Physical
Media, Signal, and Binary Transmission

Layer 4: Transport Layer

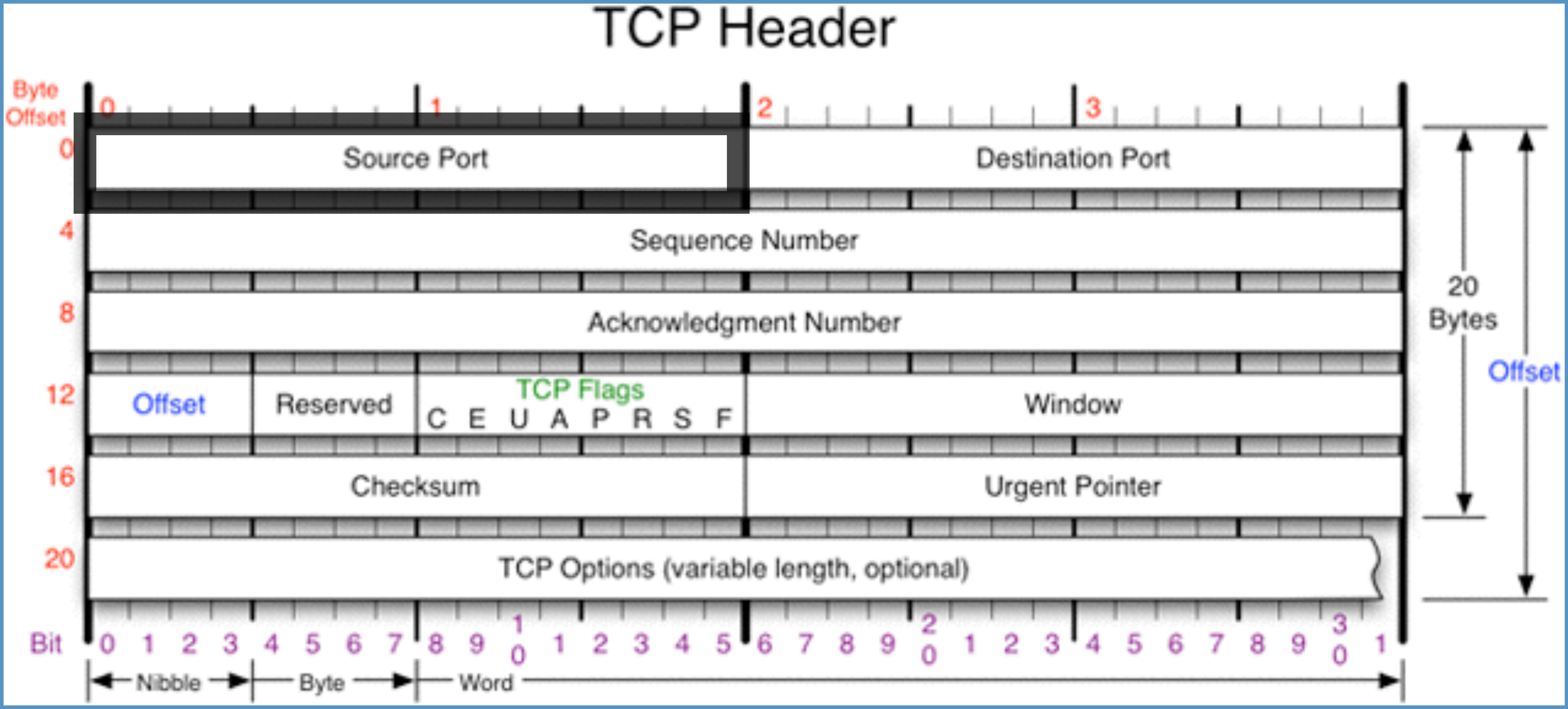
- Now that we can route from machine to machine, what about process to process?
- defines “segments” between different processes of different hosts
- port - a machine-specific number assignment to an application connecting to the Internet
- TCP: reliable messaging
- UDP: unreliable

TCP



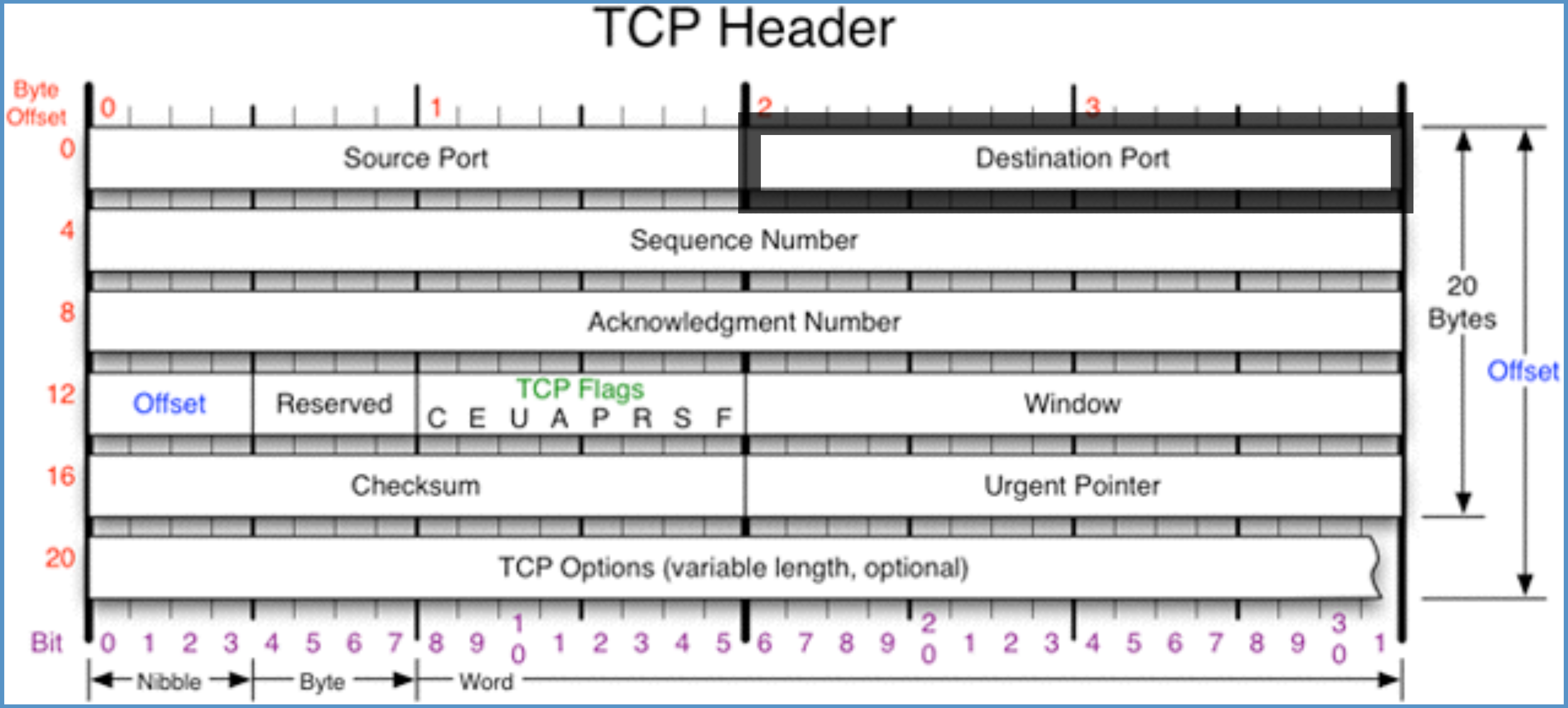
TCP Header
Byte Offset
0
4
8
12
16
20
0
1
2
3
Source Port
Destination Port
Sequence Number
Acknowledgment Number
Offset
Reserved
TCP Flags
C E U A P R S F
Window
Checksum
Urgent Pointer
TCP Options (variable length, optional)
20 Bytes
Offset
Bit
0 1 2 3 4 5 6 7 8 9 10 1 2 3 4 5 6 7 8 9 20 1 2 3 4 5 6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

TCP



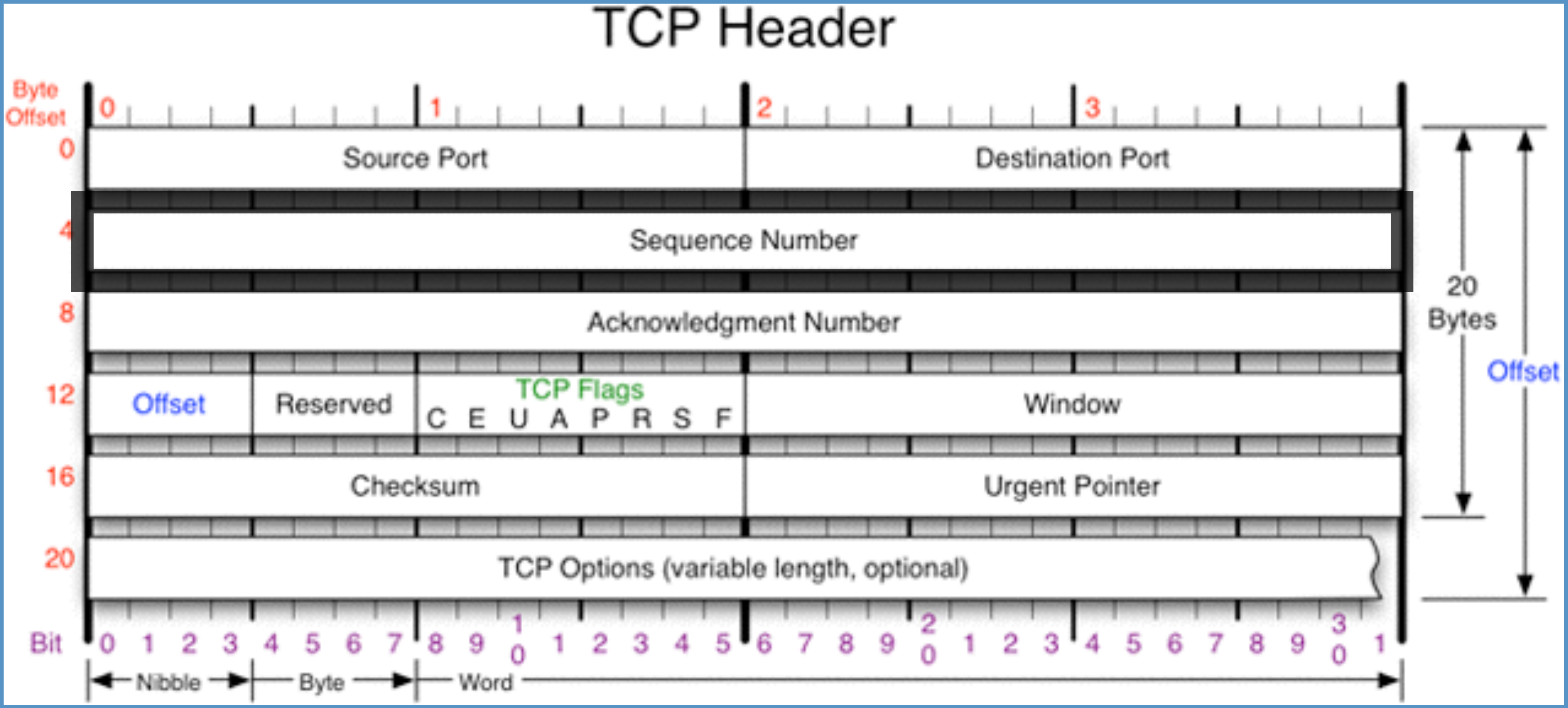
Source: the port assigned to sending application

TCP



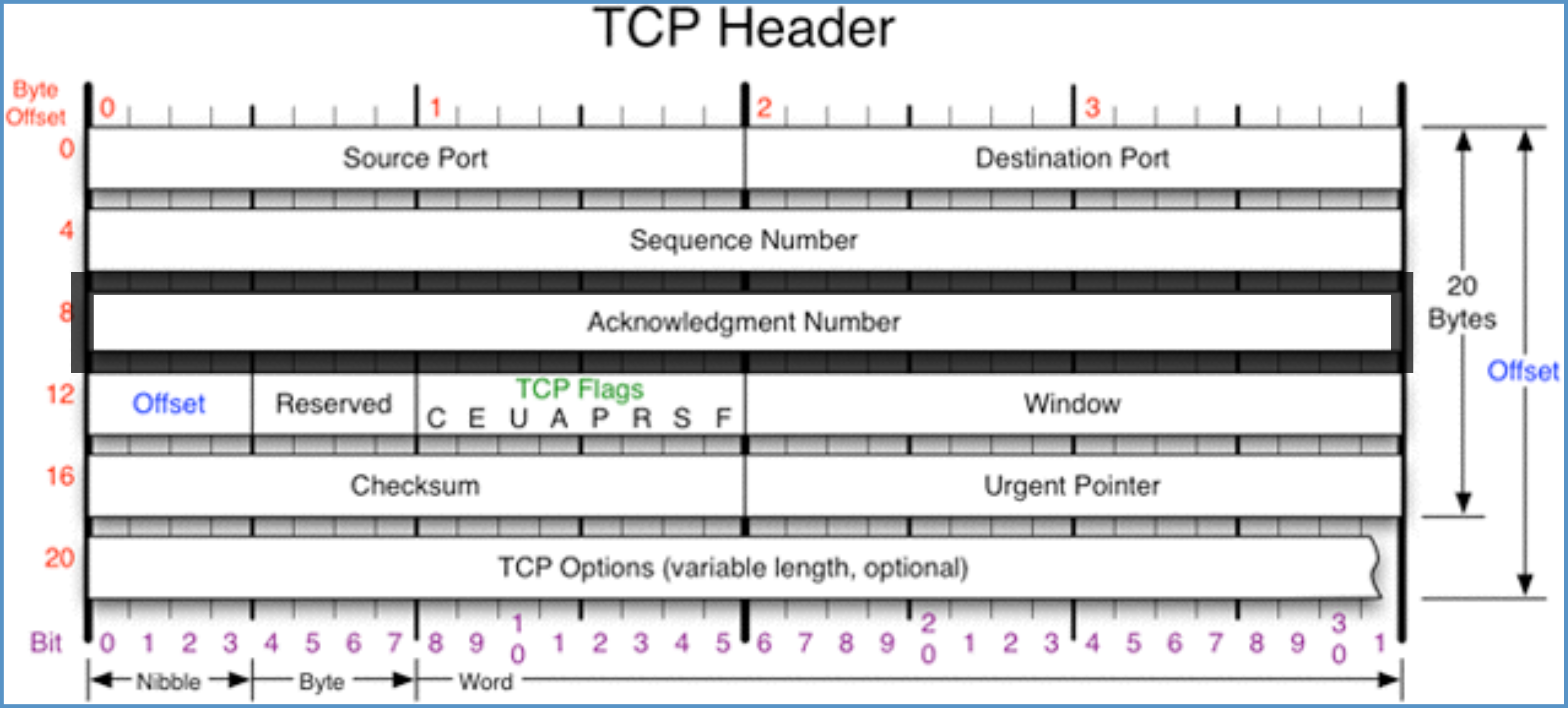
Destination: the port assigned to receiving application

TCP



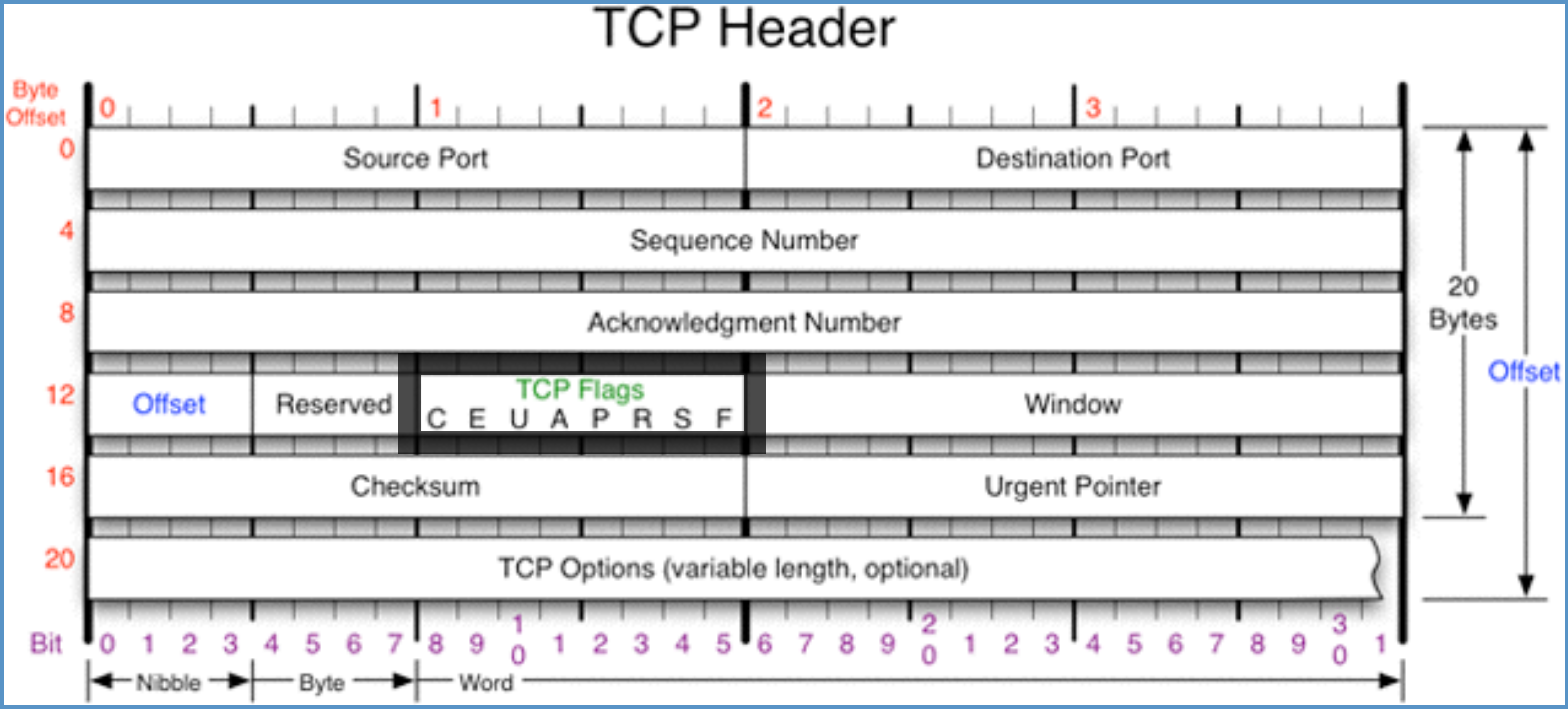
Sequence #: the "tag" assigned to this segment

TCP



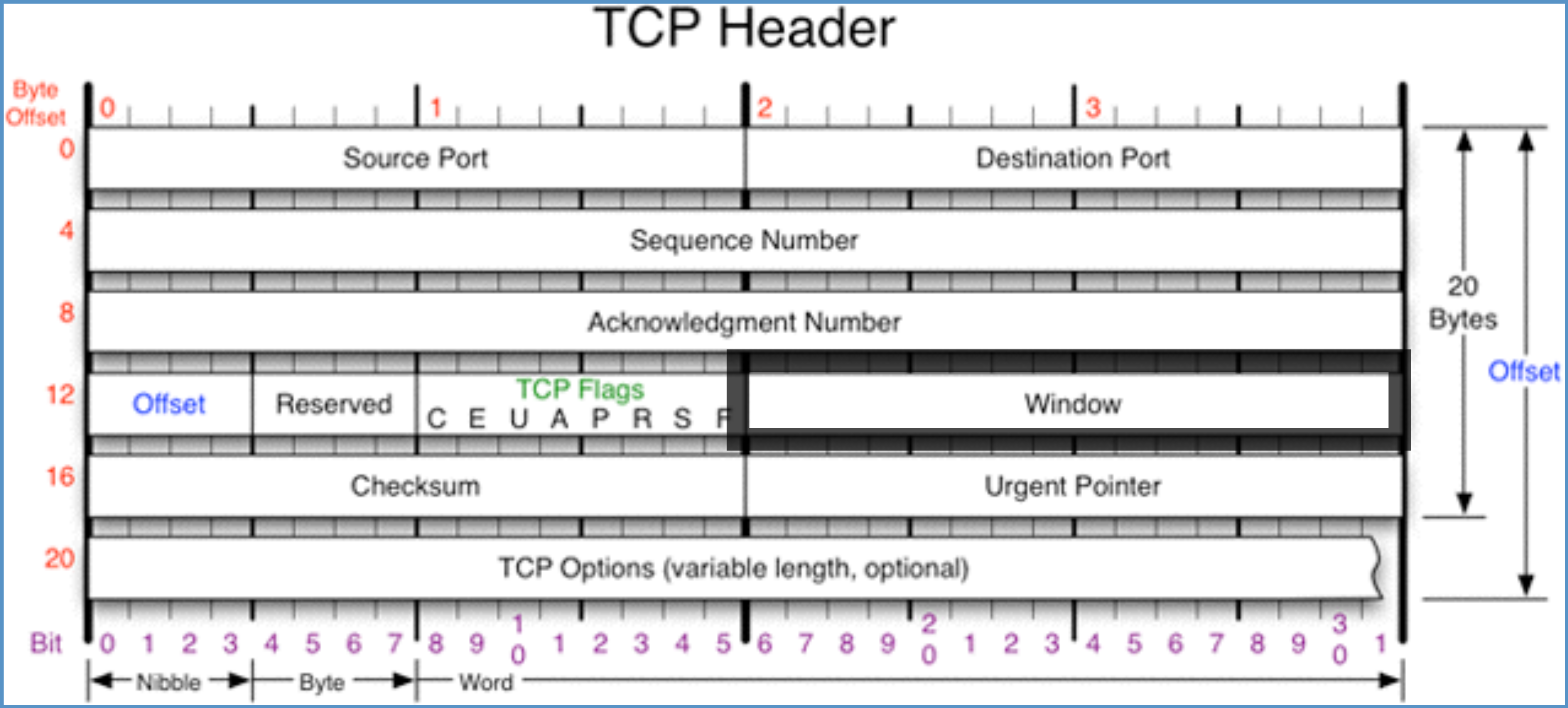
Ack #: the *next* sequence # expected by receiver

TCP



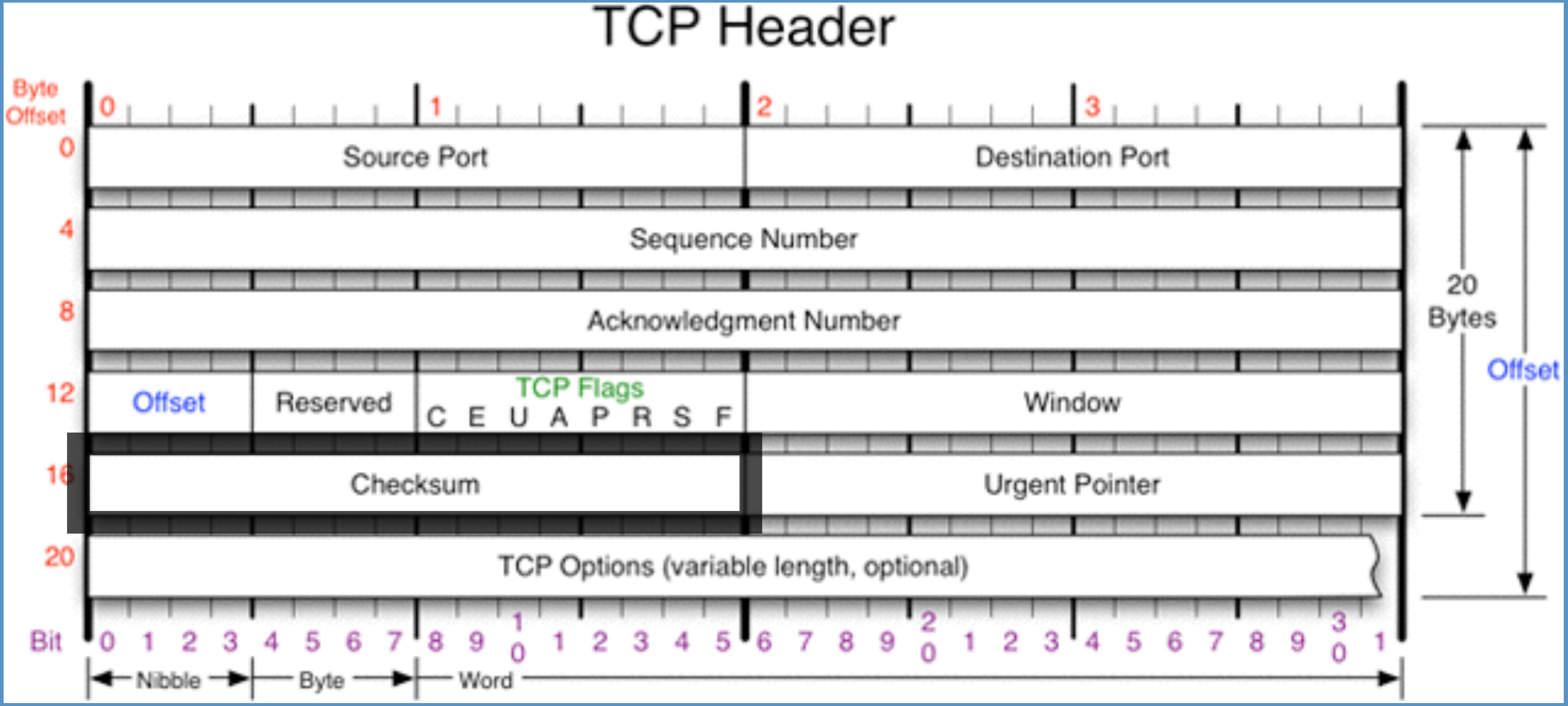
Flags: SYN, ACK, FIN…

TCP



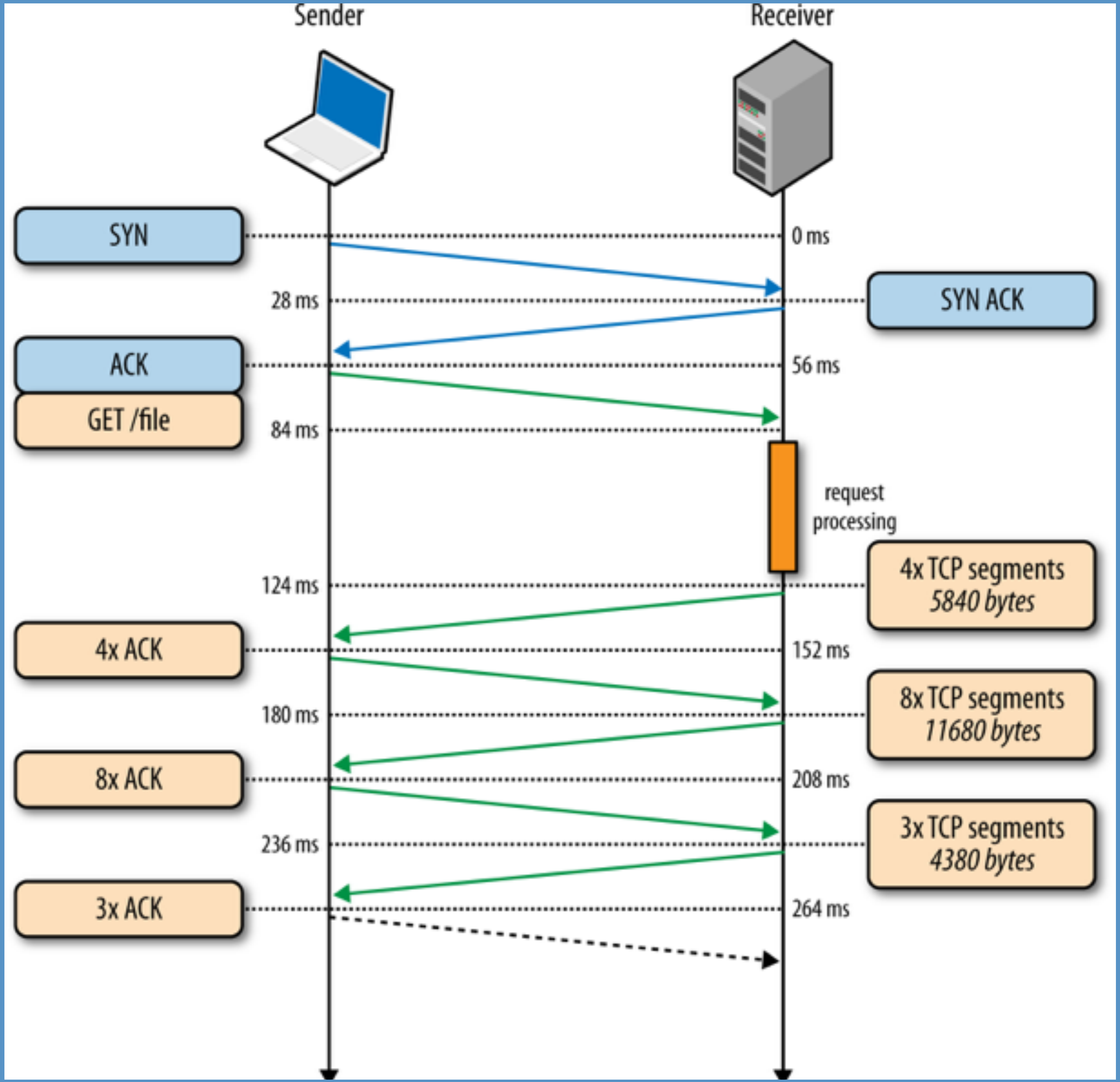
Window: size of the “*receive window*”, dynamic!

TCP

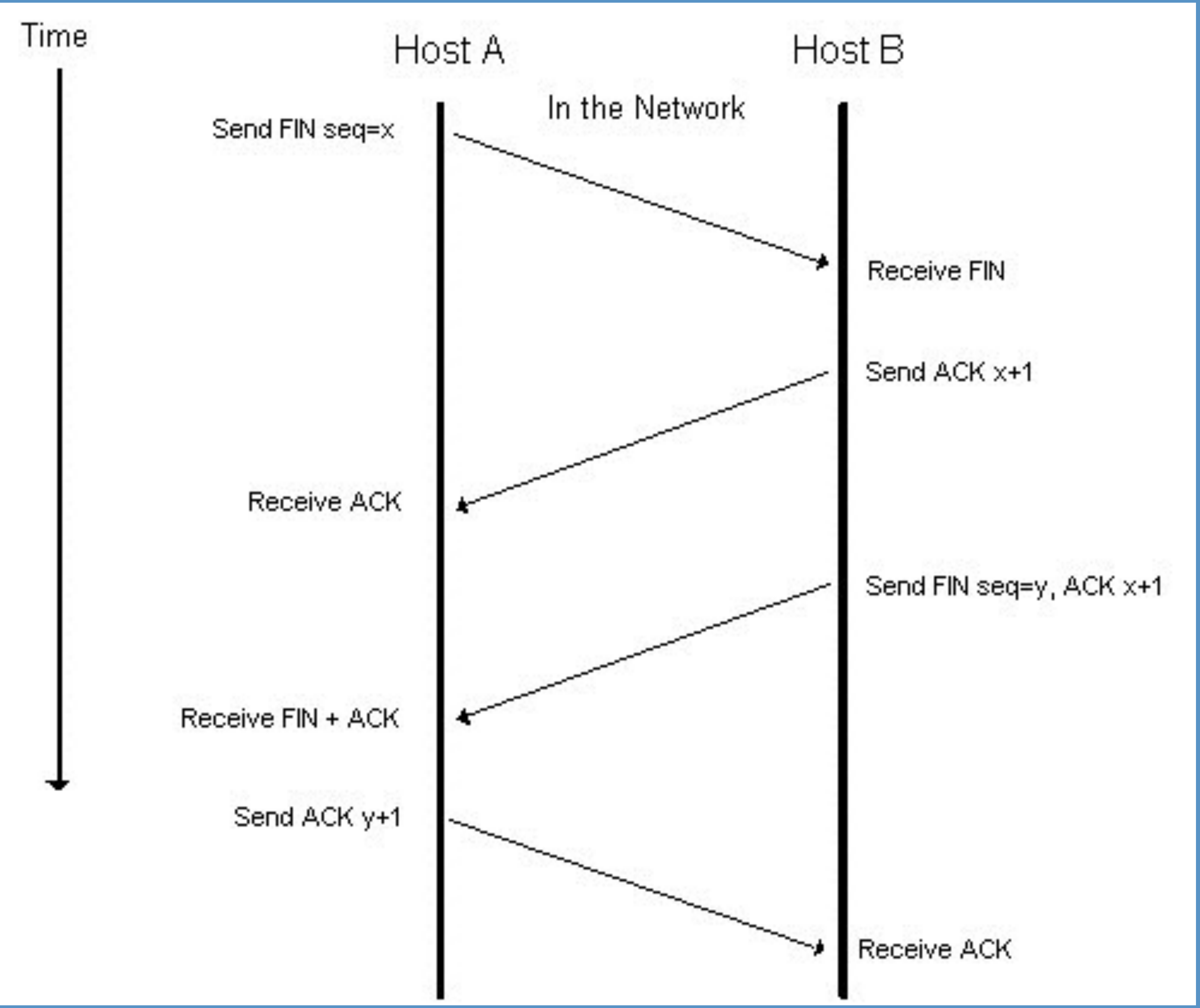


Checksum: data checksum

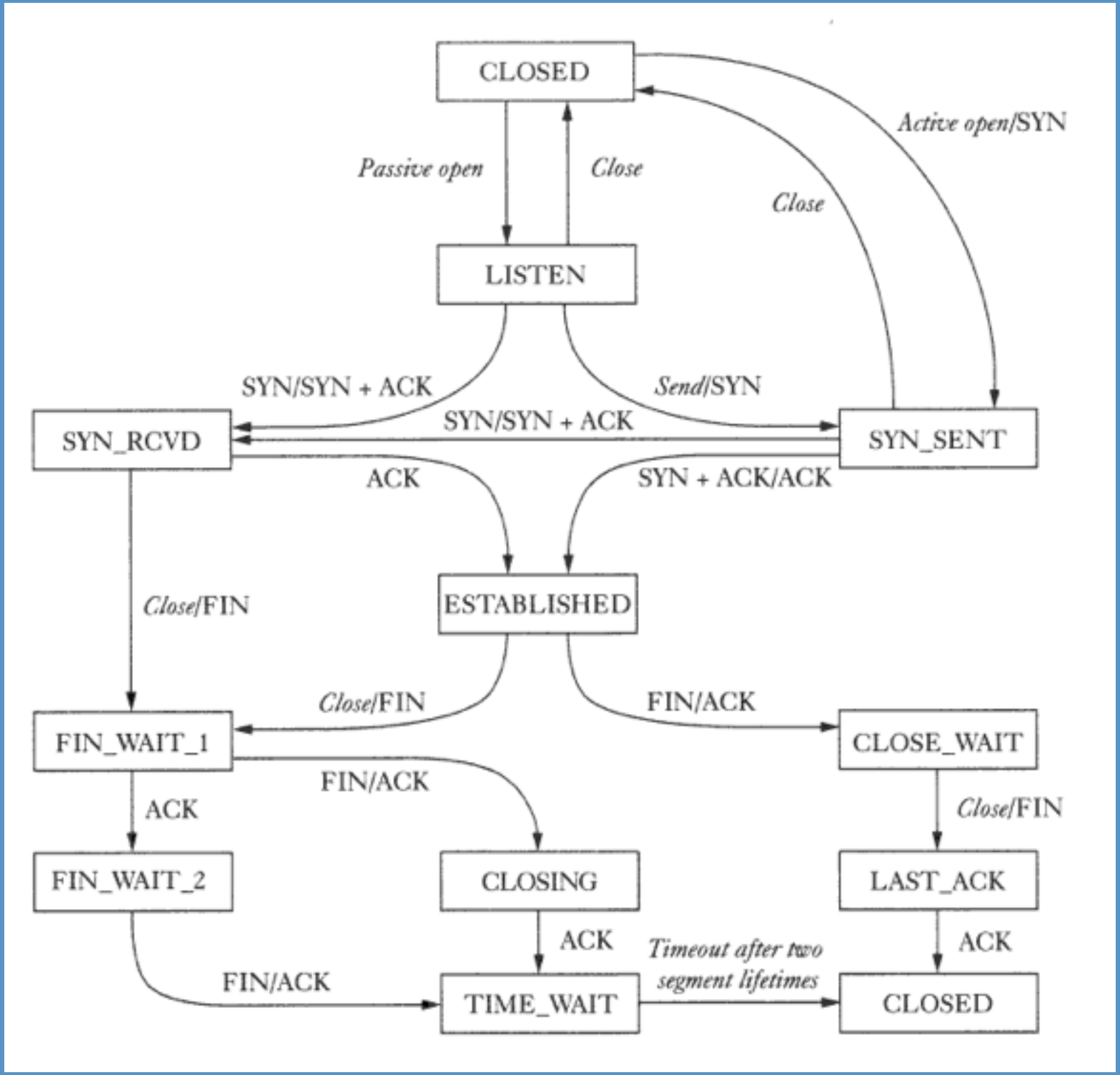
TCP: Connection Start



TCP: Connection End



TCP States


CLOSED
Passive open
Close
Close
Active open/SYN
LISTEN
SYN/SYN + ACK
Send/SYN
SYN/SYN + ACK
SYN_RCVD
SYN_SENT
ACK
SYN + ACK/ACK
Close/FIN
ESTABLISHED
Close/FIN
FIN/ACK
FIN_WAIT_1
CLOSE_WAIT
FIN/ACK
ACK
Close/FIN
FIN_WAIT_2
CLOSING
LAST_ACK
ACK
Timeout after two
segment lifetimes
ACK
FIN/ACK
TIME_WAIT
CLOSED

Wireshark

- Graphical front-end open-source packet analyzer
- Great tool for learning networking concepts!
- Can capture and filter packets
- Command line equivalent (powerful!): tcpdump

Exercise

- Use “lsof -i TCP -n -P” to list the TCP connections currently taking place on your laptop

Sockets

- Socket => file descriptor
- TCP (UDP, etc.) uses ports, OS uses sockets
- How an application communicates over the internet

References

- https://github.com/revmischa/learn-software-engineering/wiki/How-The-Internet-Works

Homework

- Create a server that accepts a UTC offset and returns the time in ISO 8601 format. Write a client that prompts the user to test.

Exit Ticket

- See Slack channel