TCP/IP And Unix Network Tools

TCP/IP Networking Review
Unix Network Tools

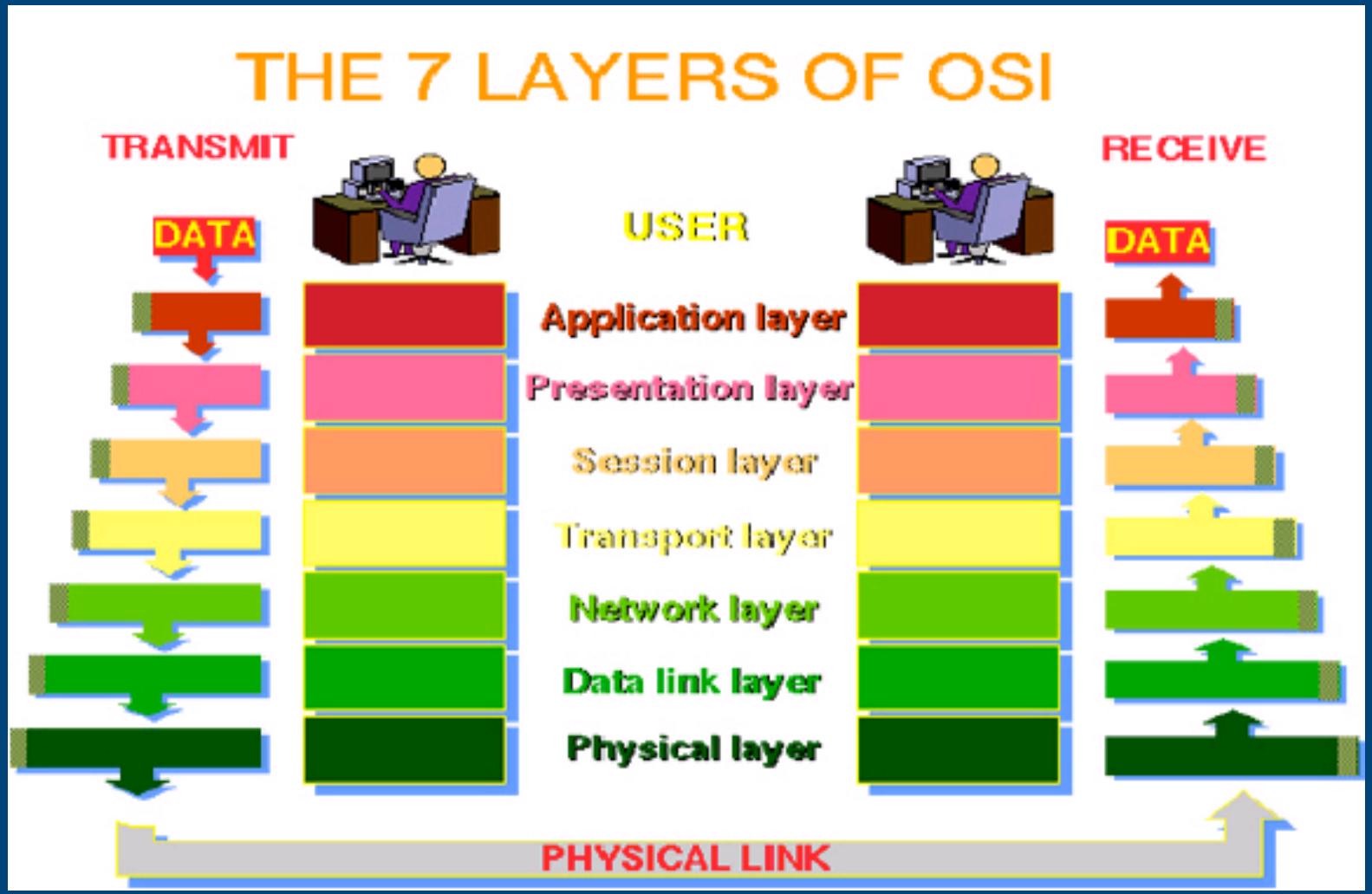
History of the Internet

- U.S. ARPANet (Advanced Research Projects Agency) in the Defense Department
- Design a Reliable, Robust Network Syste
- Detailed History
  - http://www.computerhistory.org/internet_history/
- Some Milestones
  - Arpanet Interface Message Processor RFC1 1969
  - Ethernet - 1973
  - Transmission Control Protocol - 1974
  - Internet Protocol - 1981
  - WWW Hypertext Protocol - 1989

Layers

- Complex problems can be solved using the common divide and conquer principle. In this case the internals of the Internet are divided into separate layers.
  - Makes it easier to understand
  - Developments in one layer need not require changes in another layer
  - Easy formation (and quick testing of conformation to) standards
- Two main models of layers are used:
  - OSI (Open Systems Interconnection)
  - TCP/IP

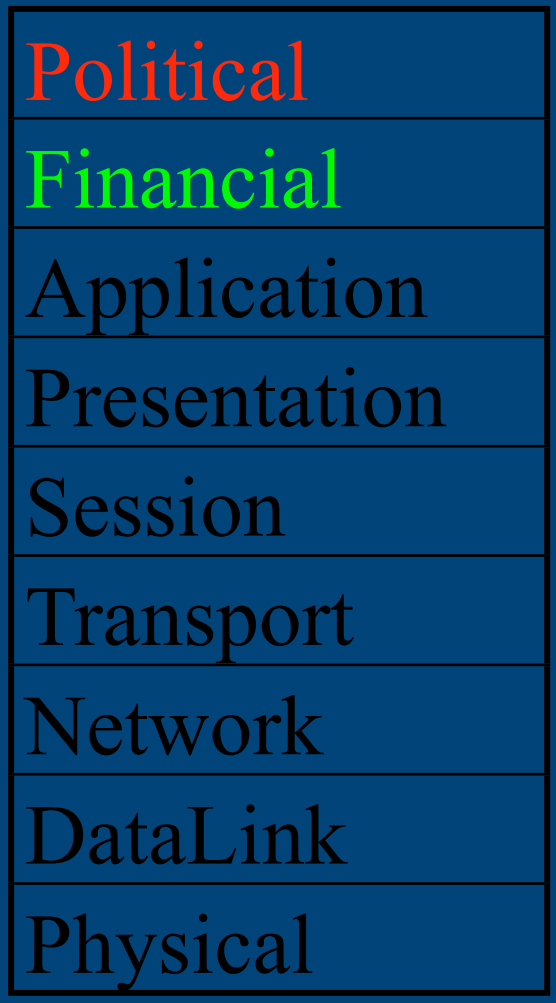
OSI Model


THE 7 LAYERS OF OSI
TRANSMIT
RECEIVE
DATA
USER
DATA
Application layer
Presentation layer
Session layer
Transport layer
Network layer
Data link layer
Physical layer
PHYSICAL LINK

OSI

- Conceptual model composed of seven layers, developed by the International Organization for Standardization (ISO) in 1984.
  - Layer 7 – Application (servers and clients etc web browsers, httpd)
  - Layer 6 – Presentation (file formats e.g pdf, ASCII, jpeg etc)
  - Layer 5 – Session (conversation initialisation, termination,)
  - Layer 4 – Transport (inter host comm – error correction, QOS)
  - Layer 3 – Network (routing – path determination, IP[x] addresses etc)
  - Layer 2 – Data link (switching – media acces, MAC addresses etc)
  - Layer 1 – Physical (signalling – representation of binary digits)
- Acronym: All People Seem To Need Data Processing

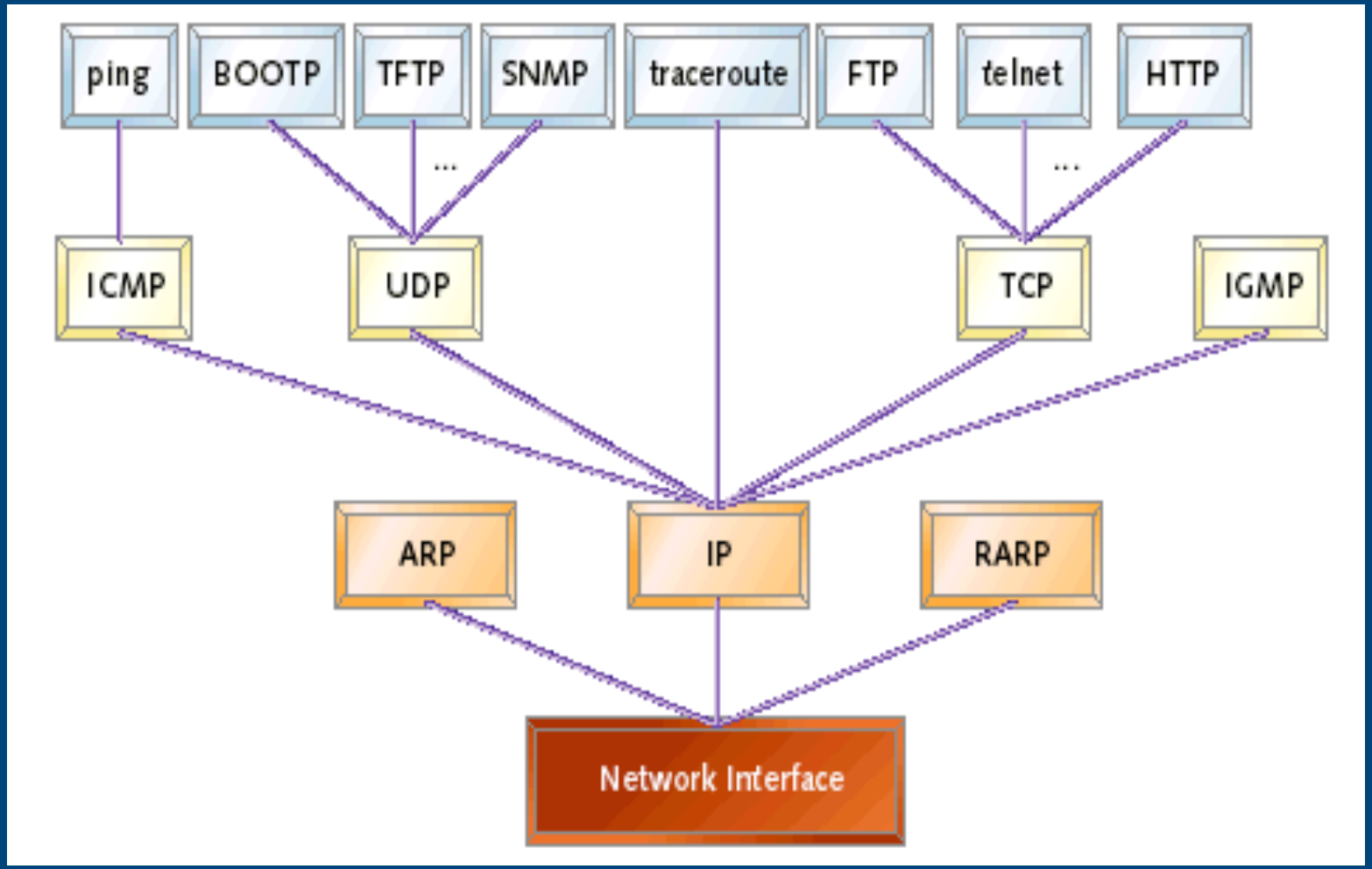
Two Other Layers To Be Aware Of



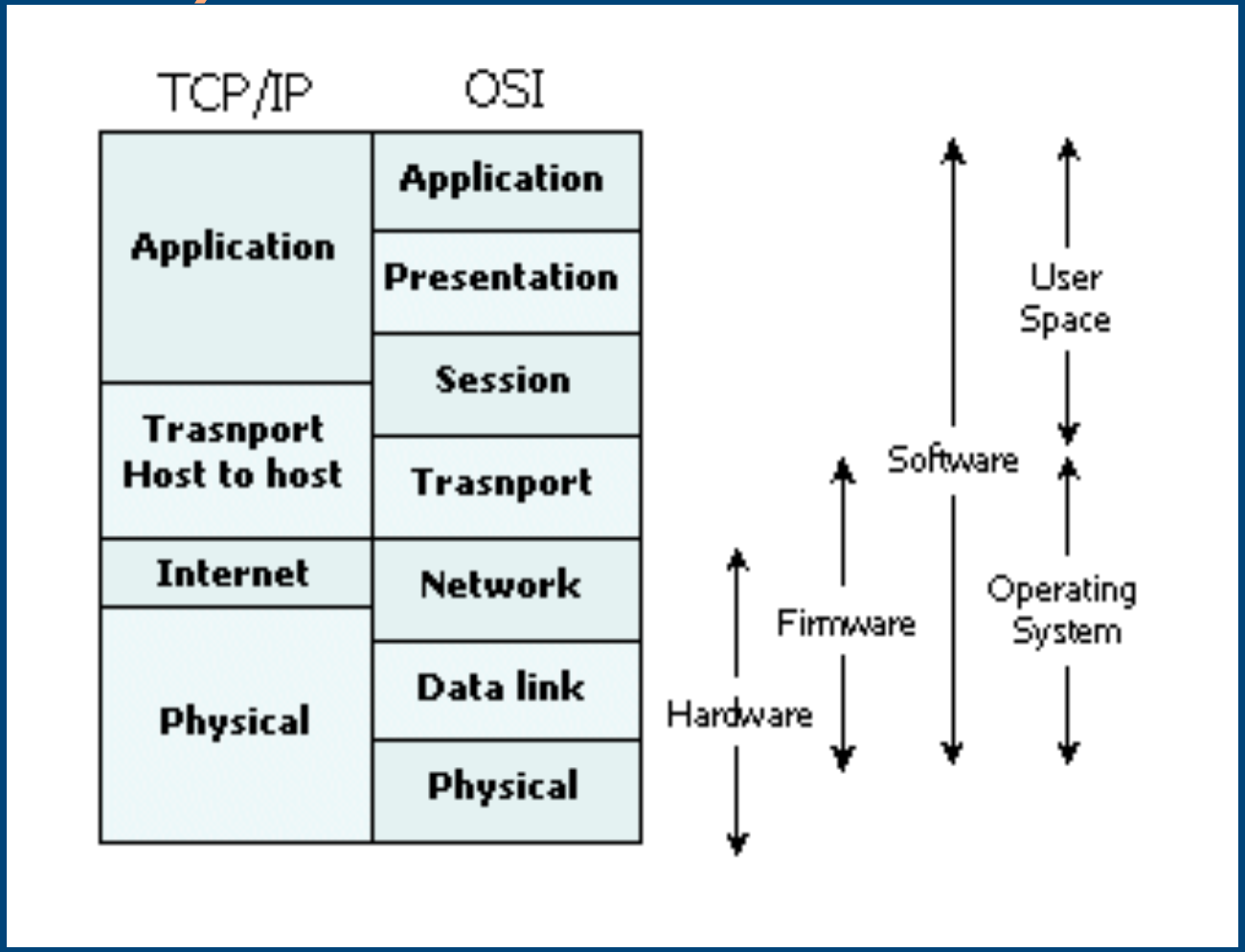
TCP/IP

- Generally, TCP/IP (Transmission Control Protocol/Internet Protocol) is described using three to five functional layers. We have chosen the common DoD reference model, which is also known as the Internet reference model.
  - Process/Application Layer consists of applications and processes that use the network.
  - Host-to-host transport layer provides end-to-end data delivery services.
  - Internetwork layer defines the datagram and handles the routing of data.
  - Network access layer consists of routines for accessing physical networks.

TCP/IP diagram

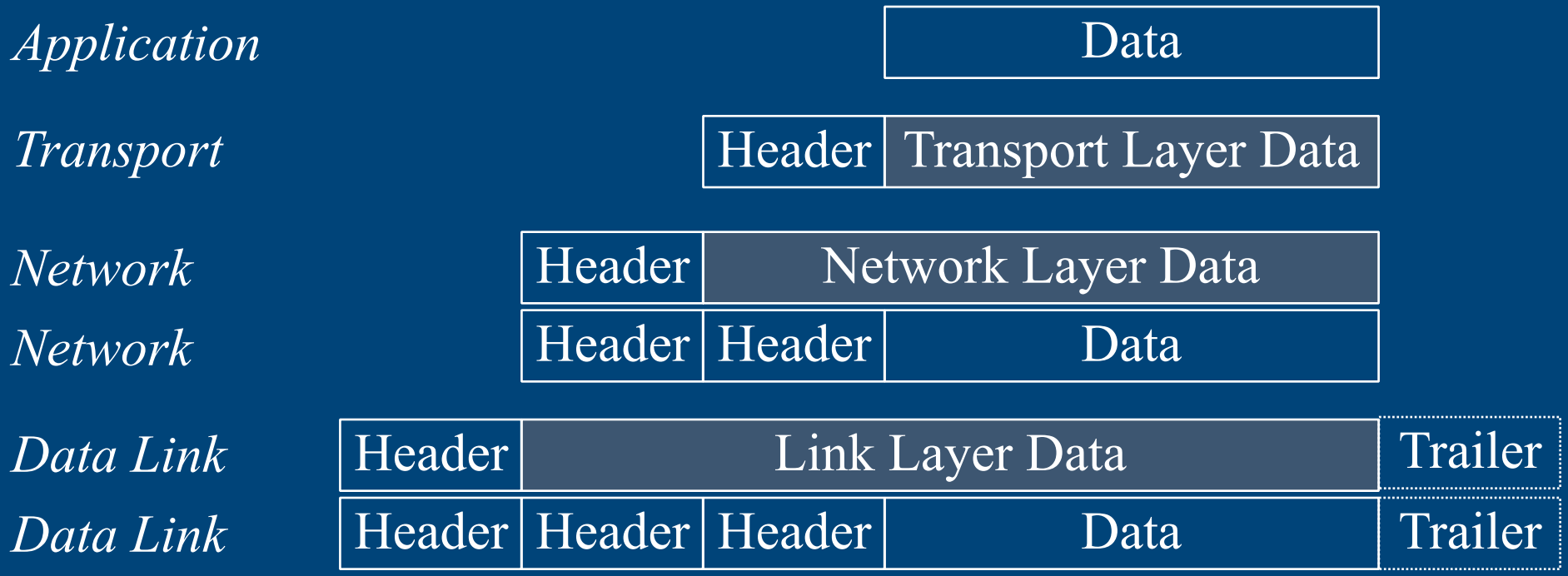


OSI and TCP/IP



Encapsulation & Decapsulation

- Lower layers add headers (and sometimes trailers) to upper layers packets



Frame, Datagram, Segment, Packet

- Different names for packets at different layers
  - Ethernet (link layer) frame
  - IP (network layer) datagram
  - TCP (transport layer) segment
- Terminology is not strictly followed
  - we often just use the term “packet” at any lay

IP Internet Protocol

- RFC 791
  - http://www.faqs.org/rfcs/rfc791.html
- Connectionless, Best-Effort Protocol
- Supports Fragmentation and Reassembly
- IP Header is Checksummed At Each Hop
- TTL Counter Decremented at Each Routing Hop
- 32-bit Globally Unique Addresses
- Grouped Into Networks: Address & Netmask
- Addresses Allocated By Registrars

IP Internet Protocol

+	Bits 0–3	4–7	8–15	16–18	19–31
0	Version	Header length	Type of Service (now DiffServ and ECN)	Total Length	
32	Identification			Flags	Fragment Offset
64	Time to Live		Protocol	Header Checksum	
96	Source Address				
128	Destination Address				
160	Options				
160 or 192+	Data				

So what is an IP address anyway?

- 32 bit number (4 octet number) can be represented in lots of ways:

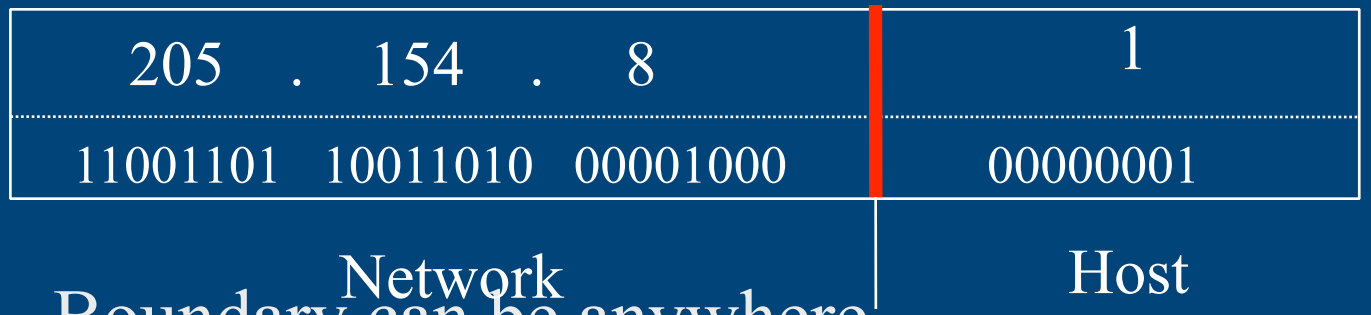
133	27	162	125

10000101	00011011	10100010	01111101

85	1B	A2	7D

More to the structure

- Hierarchical Division in IP Address:
  - Network Part (Prefix)
    - describes which physical network
  - Host Part (Host Address)
    - describes which host on that network



  - Boundary can be anywhere
    - very often NOT at a multiple of 8 bits

Network Masks

- Network Masks help define which bits are used to describe the Network Part and which for hosts
- Different Representations:
  - decimal dot notation: 255.255.224.0
  - binary: 11111111 11111111 11100000 00000000
  - hexadecimal: 0xFFFFE000
  - number of network bits: /19
- Binary AND of 32 bit IP address with 32 bit netmask yields network part of address

Sample Netmasks

137.158.128.0/**17** (netmask **255.255.128.0**)

1111 1111	**1111 1111**	**1** **000 0000**	**0000 0000**
1000 1001	1001 1110	1 000 0000	0000 0000

198.134.0.0/**16** (netmask **255.255.0.0**)

1111 1111	**1111 1111**	**0000 0000**	**0000 0000**
1100 0110	1000 0110	0000 0000	0000 0000

205.37.193.128/**26** (netmask **255.255.255.192**)

1111 1111	**1111 1111**	**1111 1111**	**11** **00 0000**
1100 1101	0010 0101	1100 0001	10 00 0000

Special IP Addresses

- All 0's in host part: Represents Network
  - e.g. 193.0.0.0/24
  - e.g. 138.37.128.0/17
- All 1's in host part: Broadcast
  - e.g. 137.156.255.255 (137.156.0.0/16)
  - e.g. 134.132.100.255 (134.132.100.0/24)
  - e.g. 190.0.127.255 (190.0.0.0/17)
- 127.0.0.0/8: Loopback address (127.0.0.1)
- 0.0.0.0: Various special purposes

Allocating IP addresses

- The subnet mask is used to define size of a network
- E.g a subnet mask of 255.255.255.0 or /24 implies 32-24=8 host bits
  - 2^8 minus 2 = 254 possible hosts
- Similarly a subnet mask of 255.255.255.224 or /27 implies 32-27=5 hosts bits
  - 2^5 minus 2 = 30 possible hosts

Private Addresses

- RFC1918 Addresses
- Private IP address ranges:
  - 10/8 (10.0.0.0 – 10.255.255.255)
  - 192.168/16 (192.168.0.0 – 192.168.255.255)
  - 172.16/12 (172.16.0.0 – 172.31.255.255)
- Not Routed To The Global Internet
- Often Used In Firewall and VPN Networks
- NAT - Network Address Translation Used to Connect Public Address(es) to A Private Segment

UDP - User Datagram Protocol

- Transport Layer - Layer 4
- Encapsulated Within IP
- RFC 768
  - http://www.faqs.org/rfcs768.html
- Connectionless, Unreliable
- 16-bit Source, 16-bit Destination Ports
- Examples: DNS, TFTP

UDP - User Datagram Protocol

+	Bits 0 - 15	16 - 31
0	Source Port	Destination Port
32	Length	Checksum
64	Data	

TCP - Transmission Control Protocol

- Encapsulated within IP
- Connection Oriented, Reliable Protocol
- RFC793
  - http://www.faqs.org/rfcs/rfc793.html
- Ordered by Sequence/Acknowledge Numbers
- Checksum over both the Header and the Data
- A “Well Behaved Protocol
  - Retransmit, Slow Start, Windows, Congestion Avoid
- Examples: HTTP, SMTP, FTP, SSH, TELNET

TCP - Transmission Control Protocol

Bit offset	Bits 0–3	4–7	8–15								16–31
0	Source port										Destination port
32	Sequence number										
64	Acknowledgment number										
96	Data offset	Reserved	CWR	ECE	URG	ACK	PSH	RST	SYN	FIN	Window Size
128	Checksum										Urgent pointer
160	Options (optional)										
160/192+	Data										

TCP - Open and Close

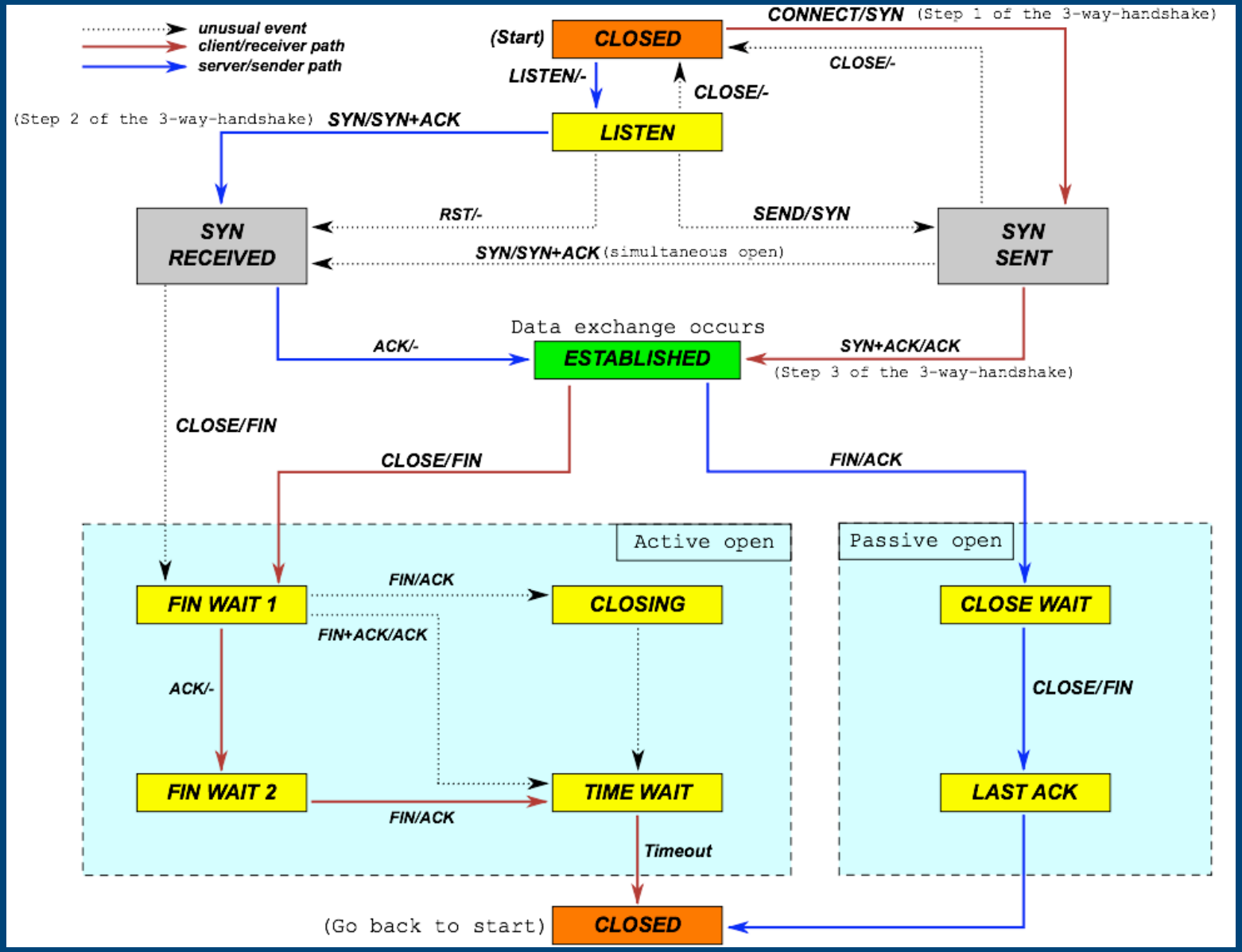
- Connection Setup Involves Send/Receive/Send

```
                  CLIENT                          SERVER                 STATE
    CLOSED                                                               LISTEN
1.  SYN-SENT     --> (SEQ=100)(CTL=SYN)                          --> SYN-RECEIVED
2.  ESTABLISHED <-- (SEQ=300)(ACK=101)(CTL=SYN,ACK)          <-- SYN-RECEIVED
3.  ESTABLISHED --> (SEQ=101)(ACK=301)(CTL=ACK)              --> ESTABLISHED
    ESTABLISHED --> (SEQ=101)(ACK=301)(CTL=ACK)(DATA)        --> ESTABLISHED
```

- Connection Teardown

```
FIN          -->
             <--      FIN/ACK

             <--      FIN
FIN/ACK      -->
```


unusual event
client/receiver path
server/sender path
(Start)
CLOSED
CONNECT/SYN (Step 1 of the 3-way-handshake)
CLOSE/-
LISTEN/-
CLOSE/-
(Step 2 of the 3-way-handshake) SYN/SYN+ACK
LISTEN
SYN RECEIVED
RST/-
SEND/SYN
SYN/SYN+ACK (simultaneous open)
SYN SENT
Data exchange occurs
ACK/-
ESTABLISHED
SYN+ACK/ACK
(Step 3 of the 3-way-handshake)
CLOSE/FIN
CLOSE/FIN
FIN/ACK
Active open
Passive open
FIN/ACK
FIN WAIT 1
CLOSING
CLOSE WAIT
FIN+ACK/ACK
ACK/-
CLOSE/FIN
FIN WAIT 2
TIME WAIT
LAST ACK
FIN/ACK
Timeout
(Go back to start)
CLOSED

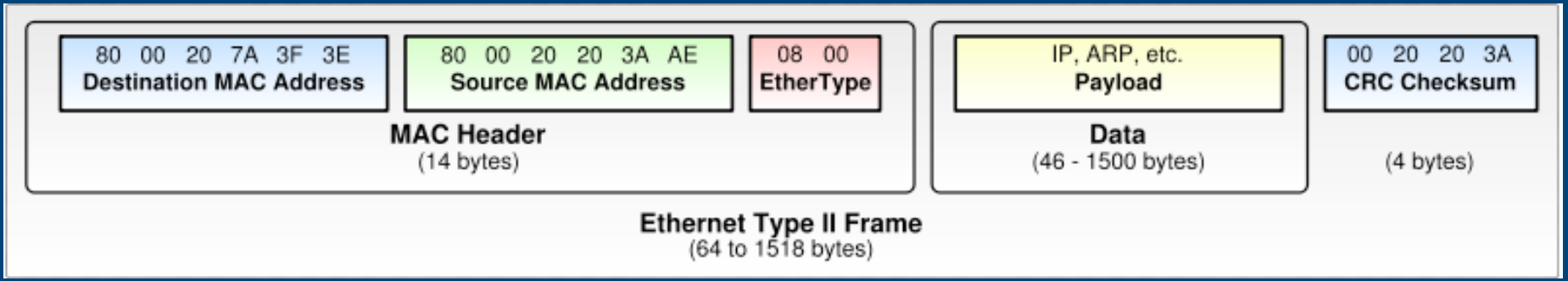
ICMP - Internet Control Message Protocol

- Error Message Reporting for IP Networks
- RFC 792
  - http://www.faqs.org/rfcs/rfc791.html
- Messages Contain a “Type” and “Code”
- Ping Tool
  - Send Echo Request: Type 8, Code 0
  - Receive Echo Reply: Type 0, Code 0
- Many Other Uses
  - Destination Unreachable / Port Unreachable
  - Fragmentation Needed
  - TTL Exceeded

Ethernet

- Layer 2, Link Layer
- Most Common Carrier For IP
- IEEE 802.3 Standards
- 48-bit Addresses (6 Bytes)
- “Unique” Addresses, 1st 3 Bytes is the Vendor OUI: Organizationally Unique Identifier
- CSMACD, Carrier Sense Multiple Access Collision Detect

Ethernet



- 1500 Byte Data Payload (MaxTransmissionUnit)
- 6+6+2+4 = 1518 Byte Frame
- IEEE 802.1q VLANS = 1522 Byte Frame
- Broadcast Address FF:FF:FF:FF:FF:FF

ARP - Address Resolution Protocol

- Connects Layer 2 (Ethernet) to Layer 3 (IP)
- Simple Request/Reply Protocol
- Initial Request is Broadcast to FF:FF:FF:FF:FF:FF
- Packet contains Source IP and MAC
- Reply contains Reply IP and MAC
- Clients maintain a local ARP Cache

ARP - Address Resolution Protocol

- Broadcast ARP request

- Machine with that IP responds

- Now send the IP datagram

ARP - Issues

- Gratuitous ARP
  - Clients may broadcast “I AM HERE!” at Any Time
  - Other Clients may choose to Update ARP Cache
- Forged ARP
  - MAC addresses can be Forged “ARP Spoofing”
  - ARP messages can be Forged “ARP Poisoning”
- Clients can configure Static ARP Entries

Communication Steps

- Sequence of Events in Most Communication Involves Many Steps
  - Interface Configuration: IP, Netmask, Broadcast
  - DNS Resolver Configuration
  - DNS Server Response
  - Default Route
  - ARP Request or ARP Cache Lookup
  - Application Server

Communication Steps

- Example: Web to http://www.ubuntu.org/
  - lookup hostname www.ubuntu.org
  - lookup address of nameserver
  - arp for nameserver or default route
  - arp for default router
  - send DNS request
  - wait for DNS reply
  - arp for www.ubuntu.org IP or default route
  - TCP 3-way Handshake with www.ubuntu.org

Unix Tools

- Network Interface
  - ifconfig - set/check interface configuration
  - netstat - check network statistics
- Layer 2, Link Layer
  - arp - set/check ARP cache
  - ip neighbor - set/check ARP cace
  - arping - send/receive ARP
- Layer 3, Network Layer
  - ping
  - traceroute
  - mtr
- Layer 4, Transport Layer
  - tcptraceroute