

Networking Fundamentals

Network Startup Resource Center
www.nsrc.org



These materials are licensed under the Creative Commons Attribution-NonCommercial 4.0 International license
(<http://creativecommons.org/licenses/by-nc/4.0/>)

Objectives

- Introduce Core Concepts & Terminology
 - Layers & Layer Models
 - Encapsulation
 - Frames, Datagrams, Segments, Packets
 - TCP/IP Protocol Suite
 - IP Addressing
 - IP Routing
 - Basic Linux Network Commands

Layers

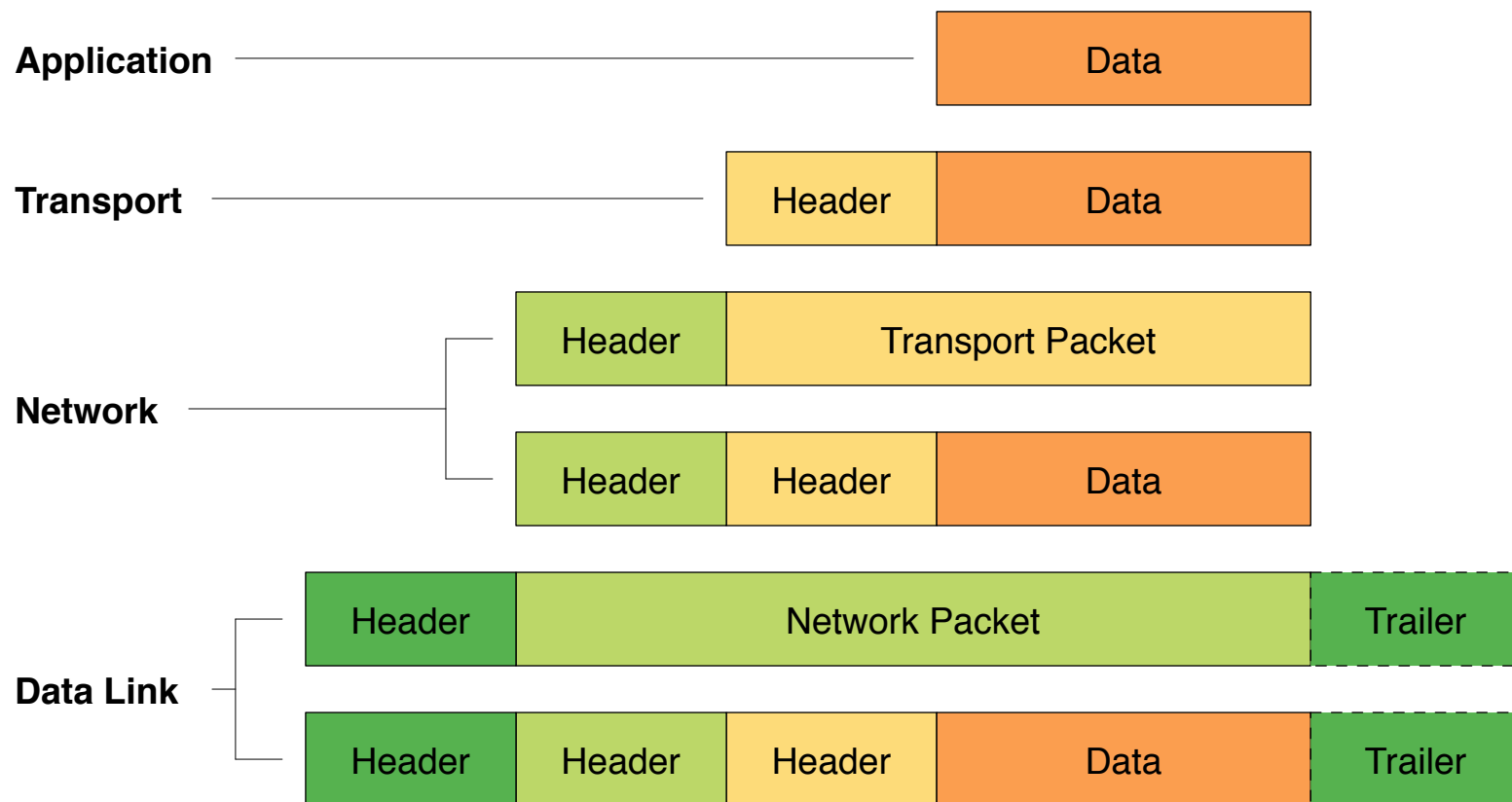
- Internet functions can be divided in layers
 - Easy to understand
 - Easy to program for
 - Change one layer without changing other layers
 - Easy to write standards for and test
- Two main models of layers are used:
 - OSI (Open Systems Interconnection) Layers
 - TCP/IP Layers
- We'll cover both models in detail

Layers & Encapsulation

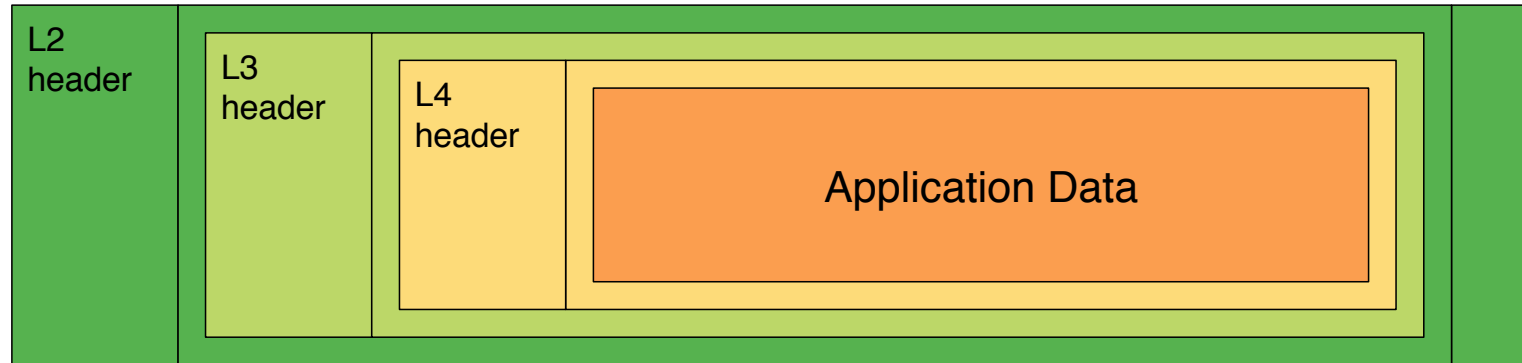
- Each layer provides services to the layer above
- Each layer makes use of the layer below
- Data from one higher layers is encapsulated in frames of the layer below

Encapsulation & Decapsulation

Lower layers add headers (& trailers) to upper layer packets



Encapsulation in Action



- L4 segment contains part of stream of application protocol
- L3 datagram contains L4 segment
- L2 frame has L3 datagram in data portion

Frame, Datagram, Segment, Packet

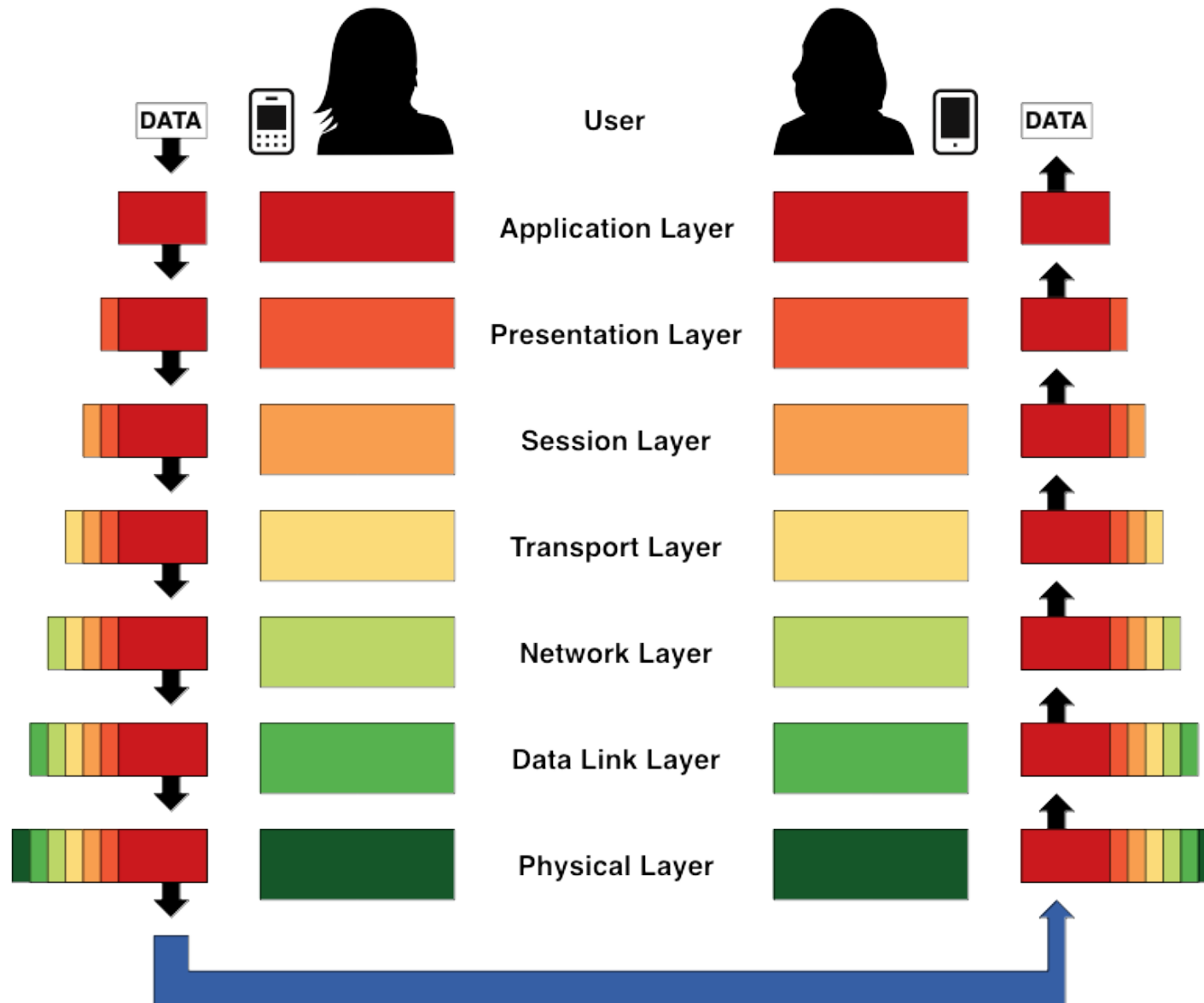
- Different names for packets at different layers
 - Link Layer = Ethernet Frame
 - Network Layer = IP Packet
 - Transport Layer = TCP Segment
- Terminology is not strictly followed
 - We often just use the term “packet” at any layer

OSI Seven Layer Model

Conceptual model developed by the International Organization for Standardization (ISO) in 1984

- Layer 7 – Application (servers & clients, web browsers, httpd)
- Layer 6 – Presentation (file formats, e.g. PDF, ASCII, JPEG)
- Layer 5 – Session (conversation initialization, termination)
- Layer 4 – Transport (inter host comm – error correction)
- Layer 3 – Network (routing – path determination, IP addresses)
- Layer 2 – Data link (switching – media access, MAC addresses)
- Layer 1 – Physical (signaling – representation of binary digits)

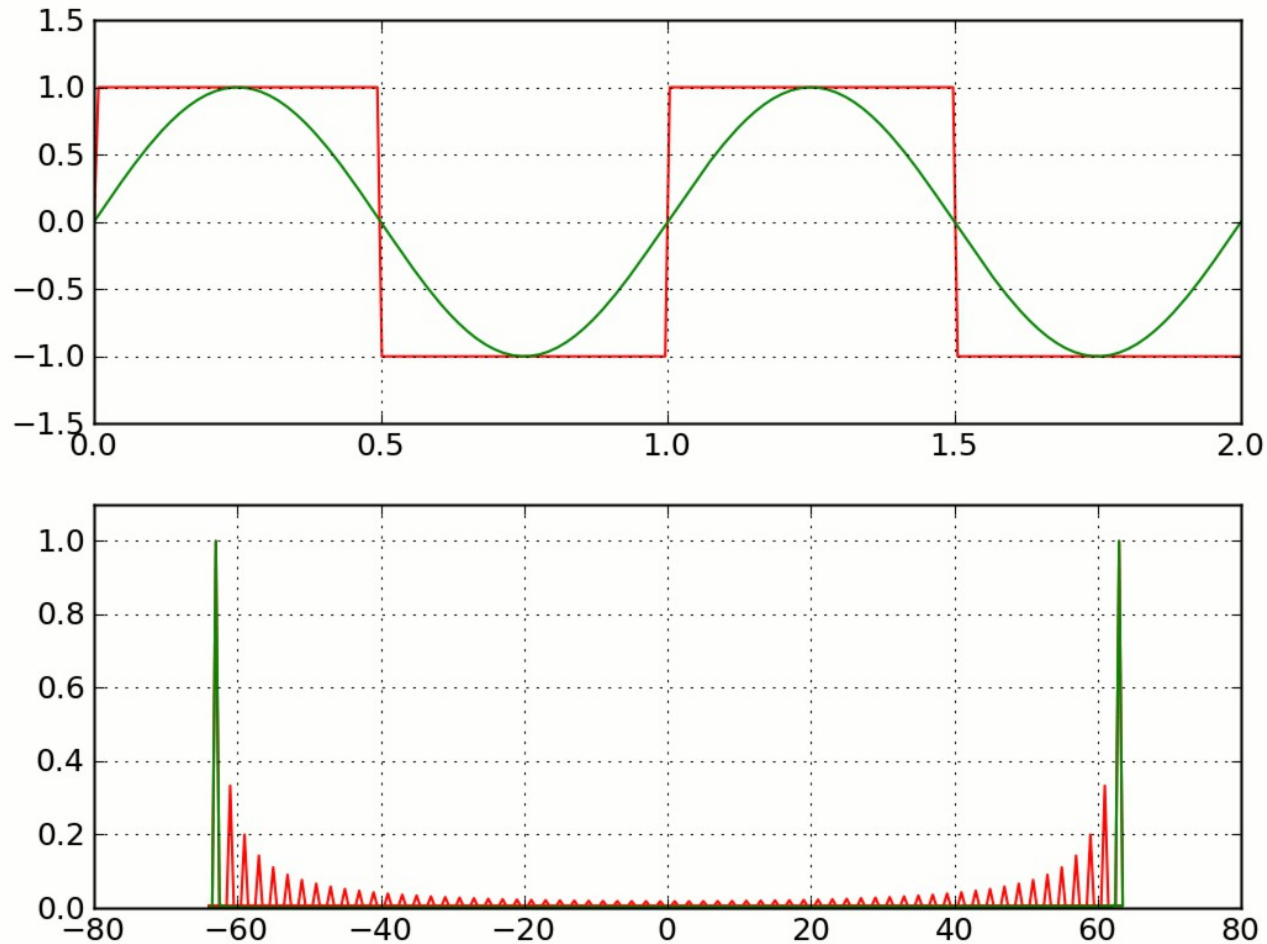
OSI Model



Layer 1: Physical Layer

- Transfers a stream of bits
- Defines physical characteristics
 - Connectors, pinouts
 - Cable types, voltages, modulation
 - Fibre types, lambdas
 - Transmission rate (bits per second)
- No knowledge of bytes or frames

Layer 1: Physical Layer

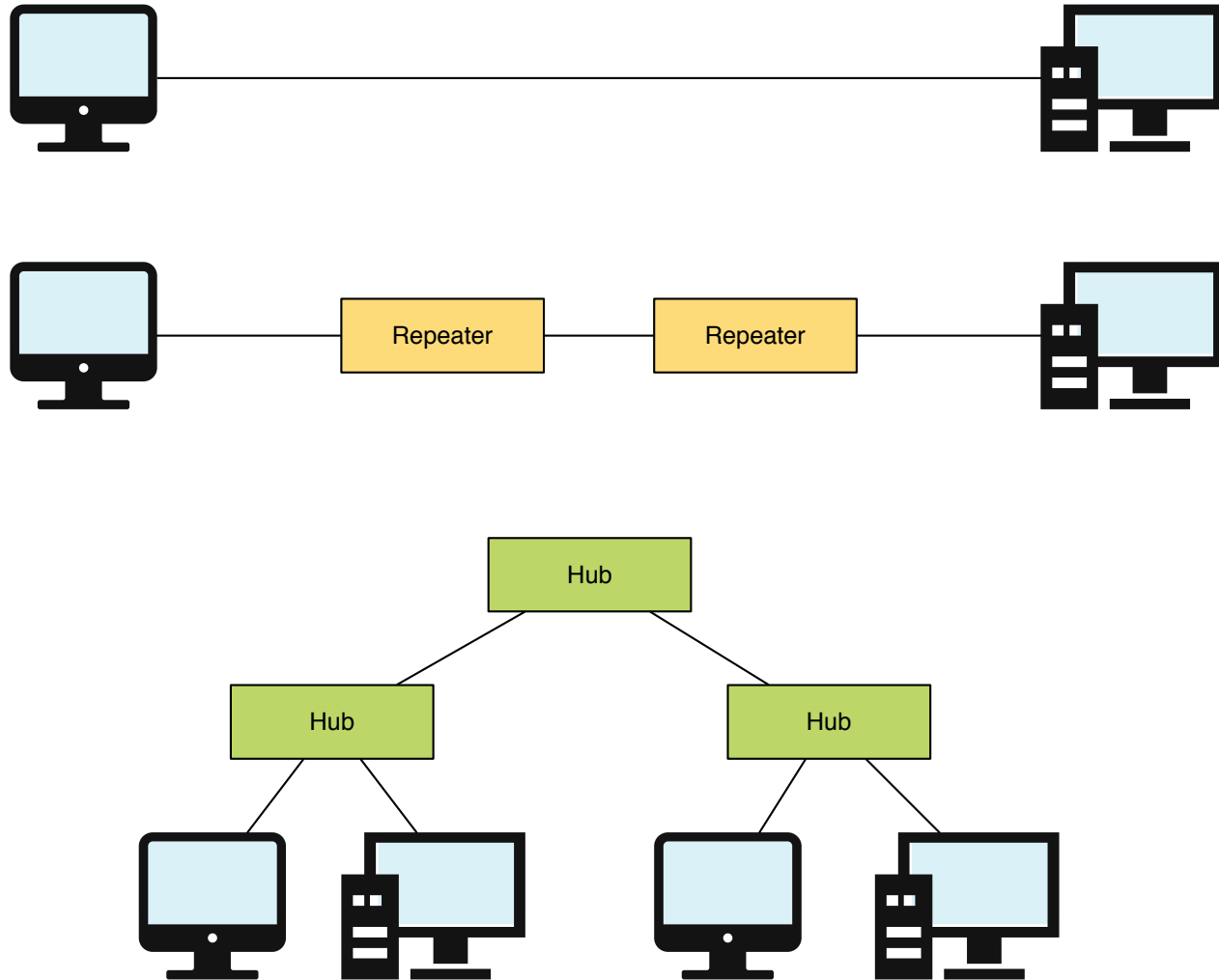


By Peretuset (Own work) CC BY 3.0 (<http://creativecommons.org/licenses/by/3.0>), via Wikimedia Commons

Layer 1: Physical Layer

- Types of Equipment
 - Hub
 - Repeater
 - Media Converter
- Works at the level of individual bits
- All data sent out of all ports
- Data may go where it is not needed

Building Networks at Layer 1



Layer 2: Data Link Layer

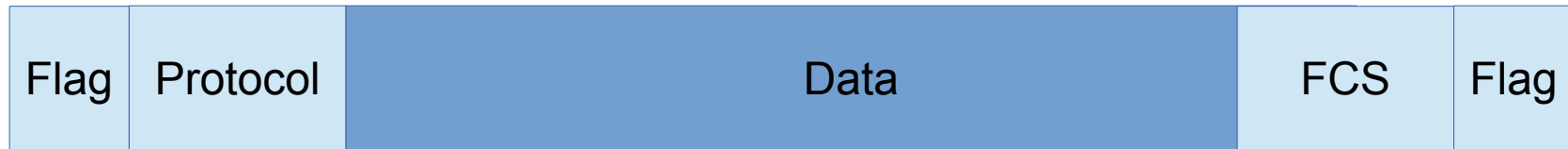
- Organizes data into frames
- May detect transmission errors (corrupt frames)
- May support shared media
 - Access control, collision detection
 - *Carrier Sense, Multiple Access, Collision Detect*
 - Addressing (who should receive the frame)
 - Unicast, Multicast
- Usually identifies the L3 protocol carried

Layer 2 Example: SLIP



That's it!

Layer 2 Example: PPP



- Also includes link setup and negotiation
 - Agree link parameters (LCP)
 - Authentication (PAP/CHAP)
 - Layer 3 Settings (IPCP)

Layer 2 Example: Ethernet

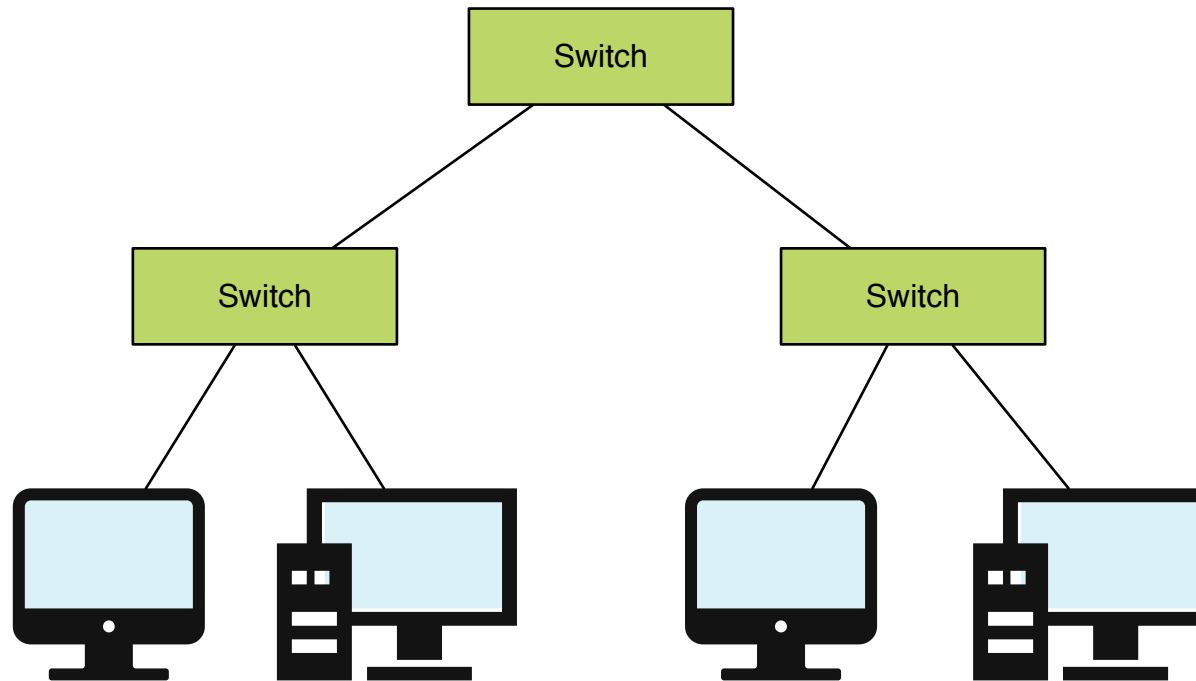


- MAC addresses
- Protocol: 2 bytes
 - e.g. 0800 = IPv4, 0806 = ARP, 86DD = IPv6
- Preamble: Carrier Sense, Collision Detection

Layer 2: Data Link Layer

- Types of Equipment
 - Switch
 - Bridge
- Receives L2 frames & selectively retransmits
- Learns which MAC addresses on which ports
- MAC known: only sends traffic to correct port
- MAC unknown: broadcast to all ports (like hub)
- Doesn't inspect packet beyond L2 header

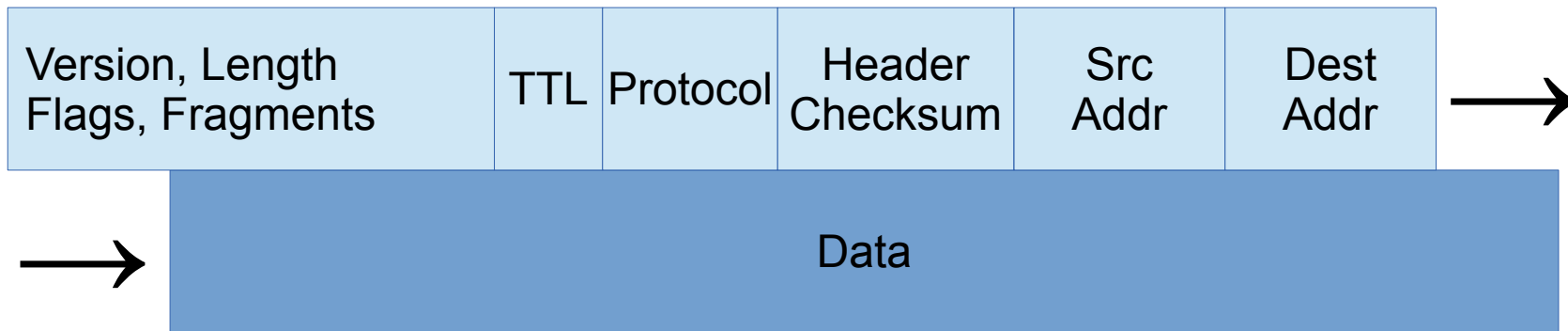
Building Networks at Layer 2



Layer 3: Network Layer

- Connects Layer 2 networks together
 - Forwards data from one network to another
- Universal frame format (datagram)
- Unified addressing scheme
 - Independent of underlying L2 networks
 - Globally organized and managed addresses
- Identifies the Layer 4 protocol being carried
- Handles fragmentation and reassembly of packets

Layer 3 Example: IPv4 Datagram

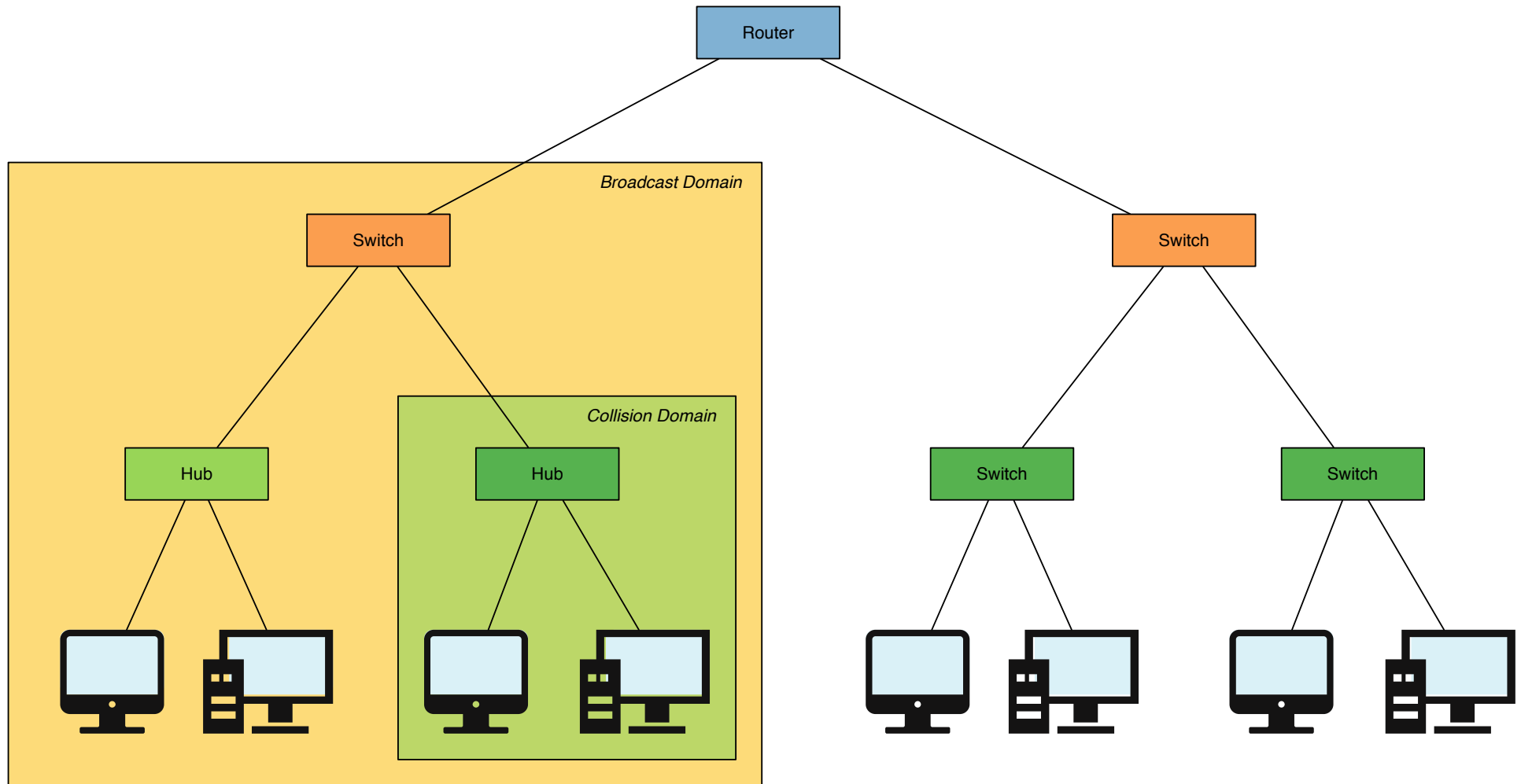


- Source, Destination: IPv4 addresses
- Protocol: 1 byte
 - e.g. 6 = TCP, 17 = UDP
 - see /etc/protocols

Layer 3: Network Layer

- Types of Equipment
 - Router
 - ~~Layer 3 Switch~~ Router
- Routers managed together are an Autonomous System
- Routers look at Destination IP in Forwarding Table
- Forwarding table can be static or dynamic
 - Static is built by hand, or scripted externally
 - Dynamic within an AS: Interior Gateway Protocol (IGP)
 - Dynamic between AS: Exterior Gateway Protocol (EGP)

Building Networks At Layer 3



Layer 4: Transport Layer

- Identifies the endpoint process
 - Another level of addressing (port numbers)
- May provide reliable delivery
 - Streams of unlimited size
 - Error correction & retransmission
 - In-sequence delivery
 - Flow control
- Might just be unreliable datagram transport

Layer 4: User Datagram Protocol



- System (Well-Known) Ports < 1024
 - 53, 69, 161, 162
- User (Registered) Ports 1024 - 49151
- Dynamic (Ephemeral) Ports
 - IANA Recommends ≥ 49152 , Linux uses ≥ 32768
 - Typically used for temporary, one session only
 - Other end of a conversation with a well-known port

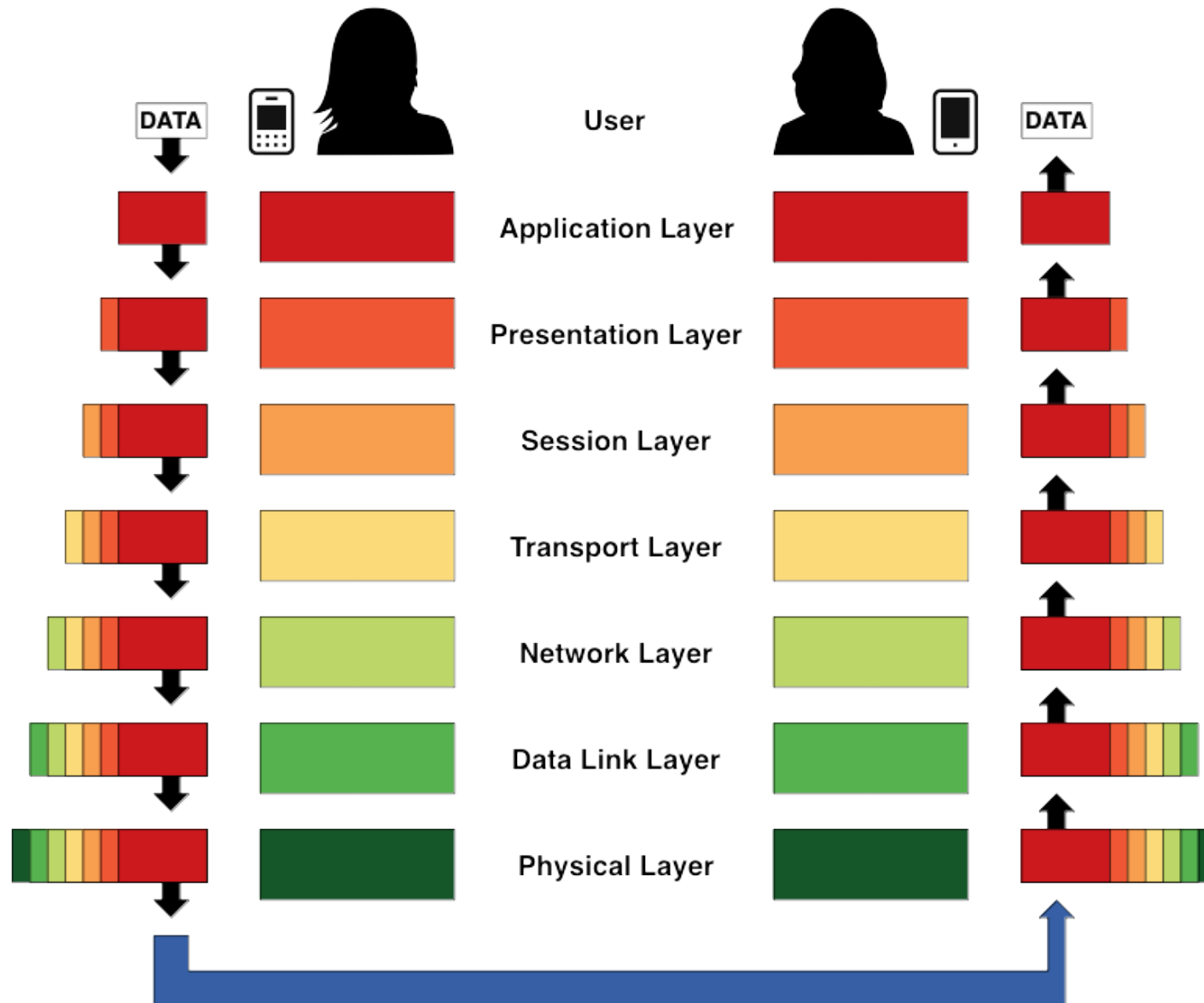
Layers 5+6: Session & Presentation

- Session Layer: long-lived sessions
 - Re-establish transport connection if it fails
 - Multiplex data across multiple connections
- Presentation Layer: data reformatting
 - Character set translation
- Neither exist in the TCP/IP suite
 - Application is responsible for these functions

Layer 7: Application Layer

- The actual work you want to do
- Protocols specific to each application
- Examples?
 -
 -
 -
 -

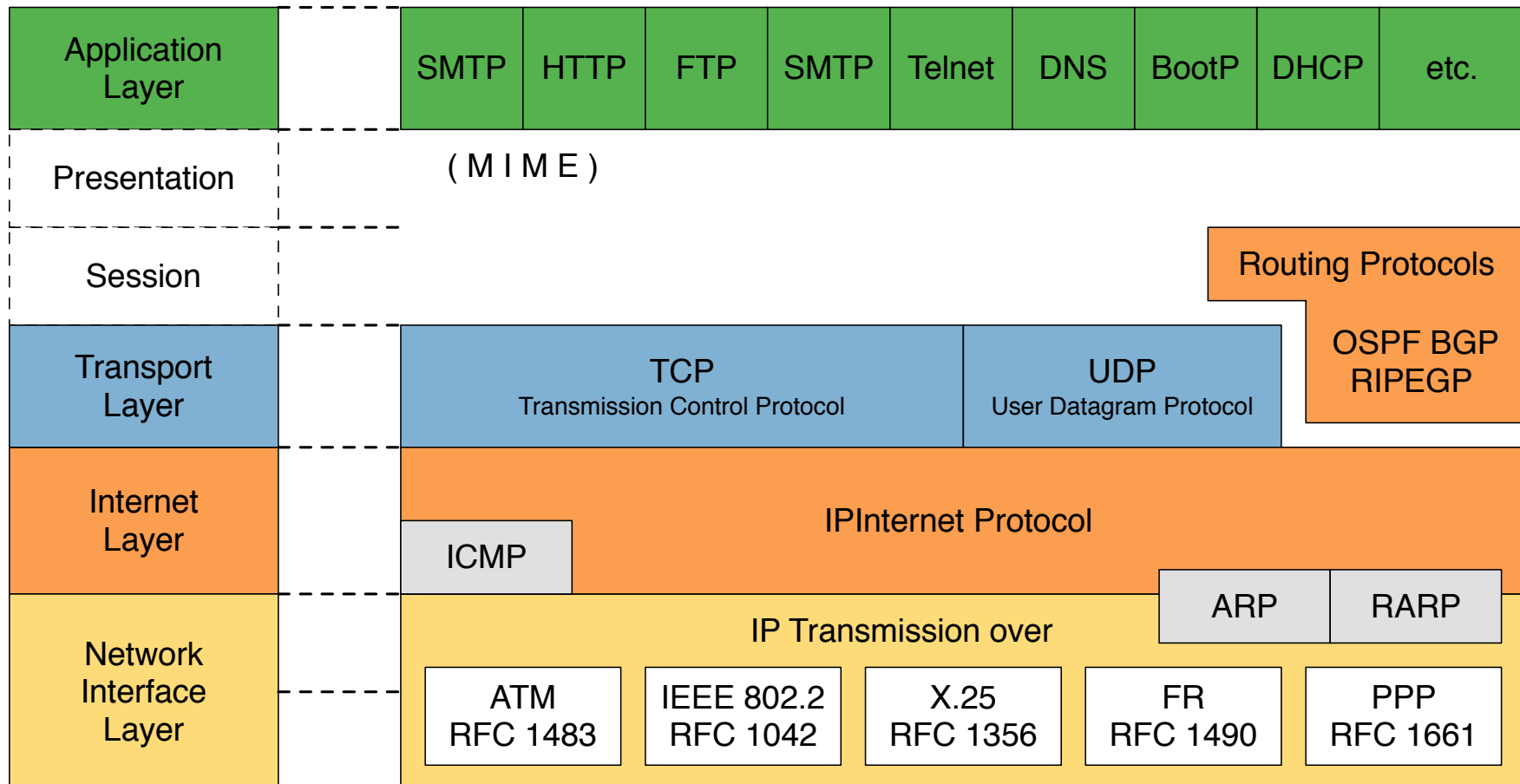
OSI Model



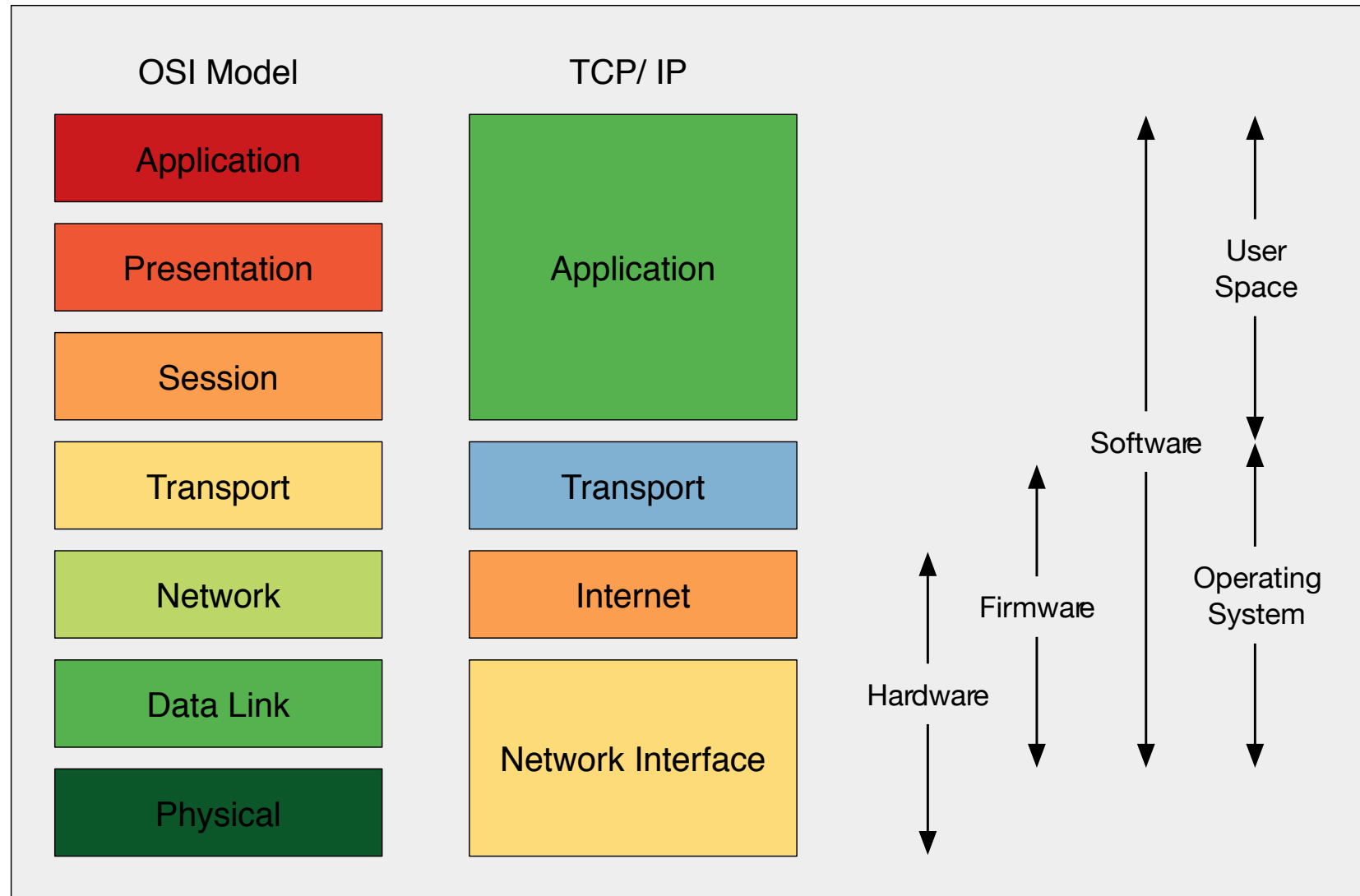
Other Layer Models

- The OSI Model is most used... but
- The DoD Model also applicable... and
- The Hourglass Model is realistic & appropriate

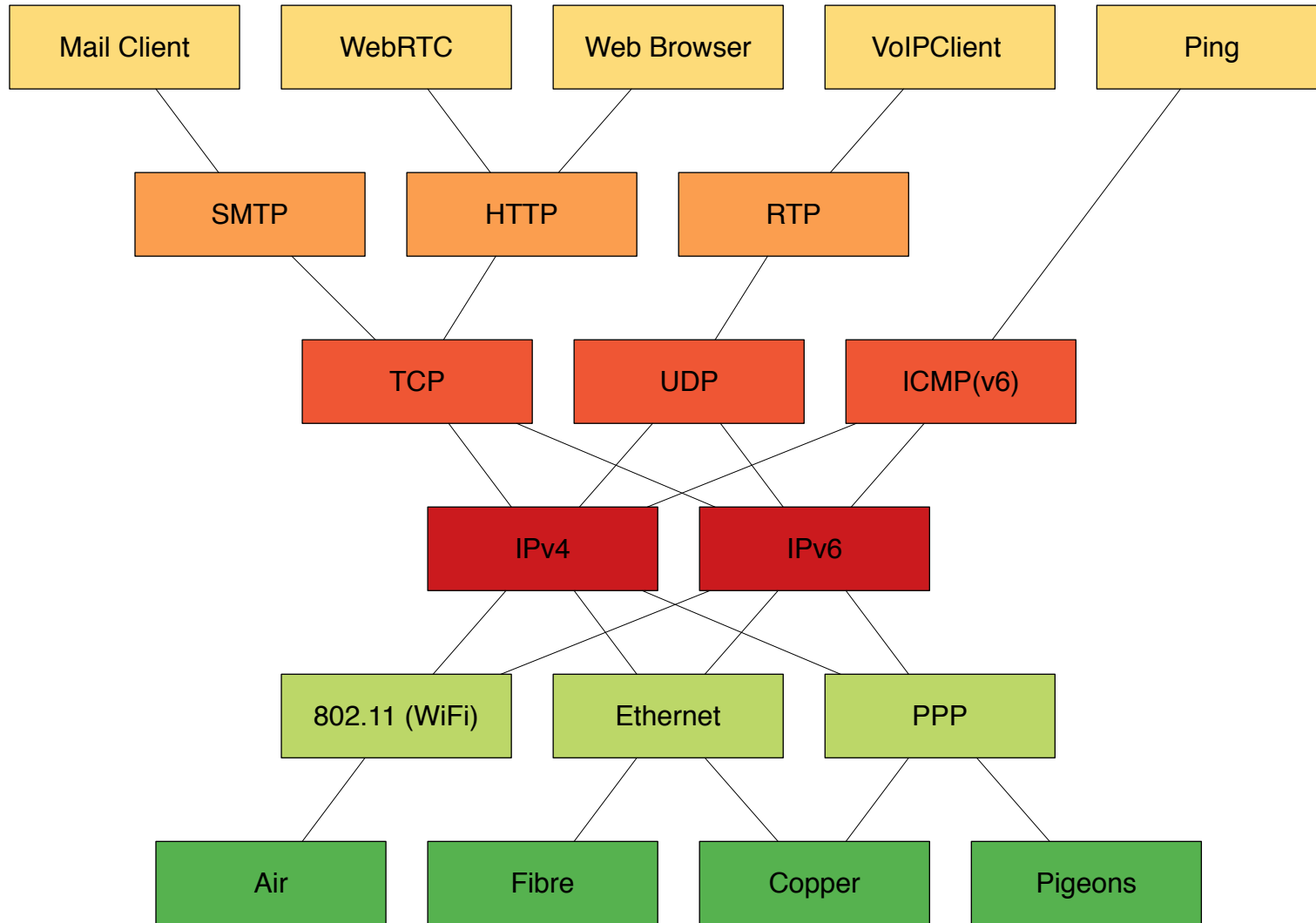
DoD Four-Layer Model



OSI vs DoD Layer Models



TCP/IP “Hourglass” Model



Layers Summary

- Networks designed & described with layers
 - OSI Model: 7 Layers
 - DoD: 3-5 Layer Model
 - Hourglass Model: 6 Layers
- Each layer encapsulates the layer below
- Receiving hosts reverse the process

IP Addresses for IPv4

- 32 bit addresses
- Calculating in decimal, bin and hex
- Hierarchical division in IP Addresses
- Network Masks
- Allocating Addresses
- Special IP addresses
- Subnetting / Supernetting

What's an IPv4 Address?

- 32 bit number
 - 4 octet number
- Can be represented in several ways:

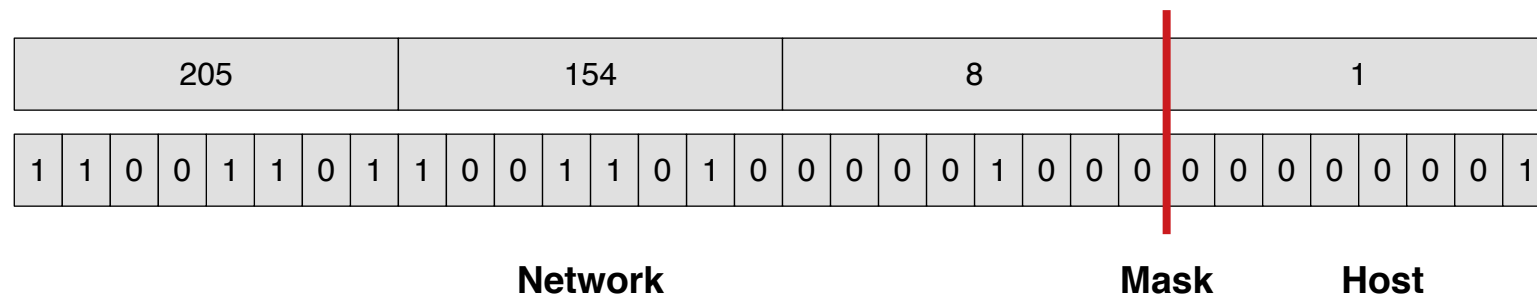
133	27	162	125
85	1B	A2	7D
1 0 0 0 0 1 0 1	0 0 0 1 1 0 1 1	1 0 1 0 0 0 1 0	0 1 1 1 1 1 0 1

Calculating dec, hex, bin

- ipcalc is your friend!
- Try this: ipcalc 133.27.162.125
- The Linux command line is your friend!
- Try this:
 - echo 'ibase=10;obase=16;27' | bc
 - echo 'ibase=10;obase=2;27' | bc
 - echo 'ibase=16;obase=A;1B' | bc

Hierarchical Division in IP Addresses

- Network Part (Prefix): Describes which network
- Host Part (Host Address): Describes which host
- Boundary can be anywhere!
 - Used to be a multiple of 8, but not required today



Network Masks

- Help define which bits are used for the Network
- And which bits are used for the hosts
- Different Representations Exist:
 - Decimal dot notation
 - Binary notation
 - Number of network bits
- Binary AND of 32 bit IP address with 32 bit netmask = network part of address

Sample Netmasks

137.158.128.0/17 (netmask 255.255.128.0)

1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
137								158								128								0							
1	0	0	1	1	0	0	1	1	0	0	1	1	1	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0		

198.134.0.0/16 (netmask 255.255.0.0)

1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0																			
198															134															0										0									
1	1	0	0	0	1	1	0	1	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0																			

205.37.193.128/26 (netmask 255.255.255.192)

1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	
205								37								193								128							
1	1	0	0	1	1	0	1	0	0	1	0	0	1	0	1	1	1	0	0	0	0	0	1	1	0	0	0	0	0	0	

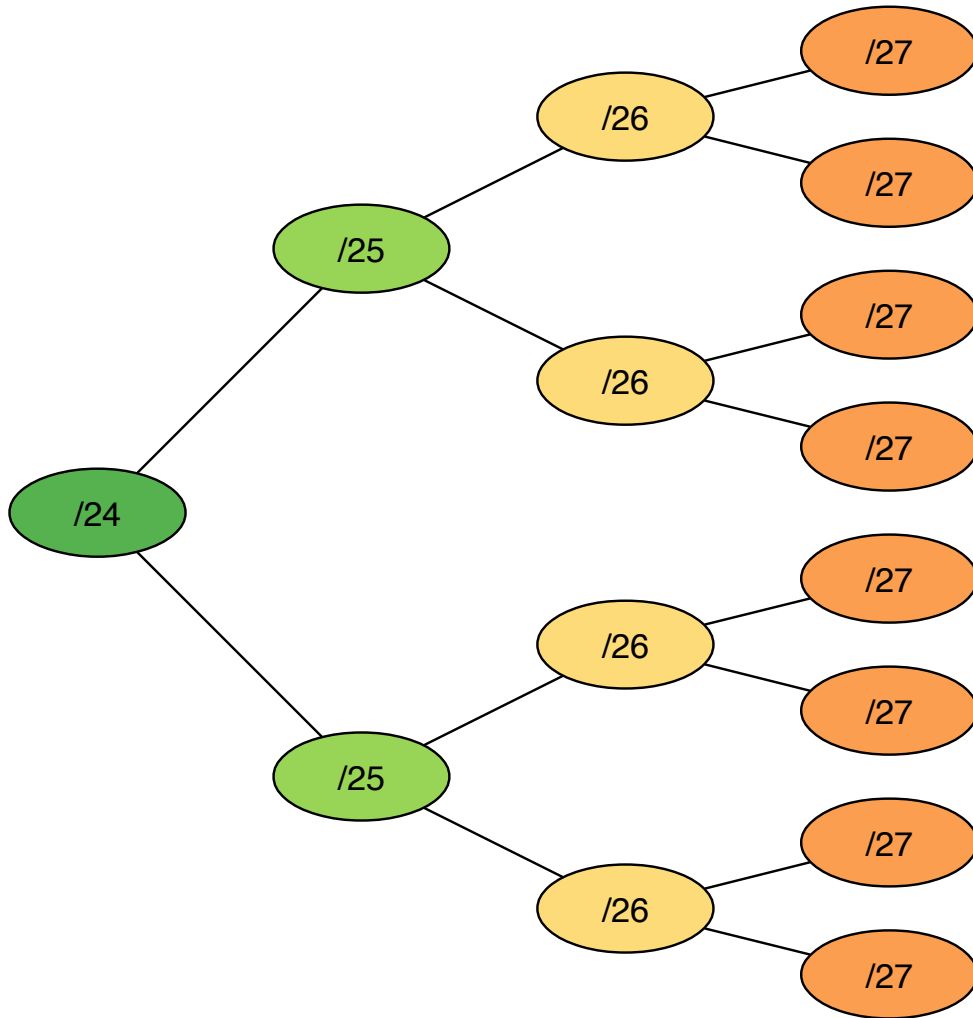
Allocating IP addresses

- The subnet mask defines the network size
- A mask of /24 or 255.255.255.0
 - $32 - 24 = 8$ host bits
 - $2^8 - 2 = 254$ possible hosts
- A mask of /27 or 255.255.255.224
 - $32 - 27 = 5$ host bits
 - $2^5 - 2 = 30$ possible hosts

Special IP Addresses

- Network Address: All 0s in the host part
 - e.g. 193.0.0.0/24
 - e.g. 138.37.128.0/17
- Broadcast Address: All 1s in the host part
 - e.g. 137.156.255.255
 - e.g. 134.132.100.255
- Loopback Address
 - 127.0.0.0/8
- Special Addresses: 0.0.0.0
 - e.g. DHCP

Networks: Subnetting



Add one bit to the netmask
& the network divides in two
This is subnetting

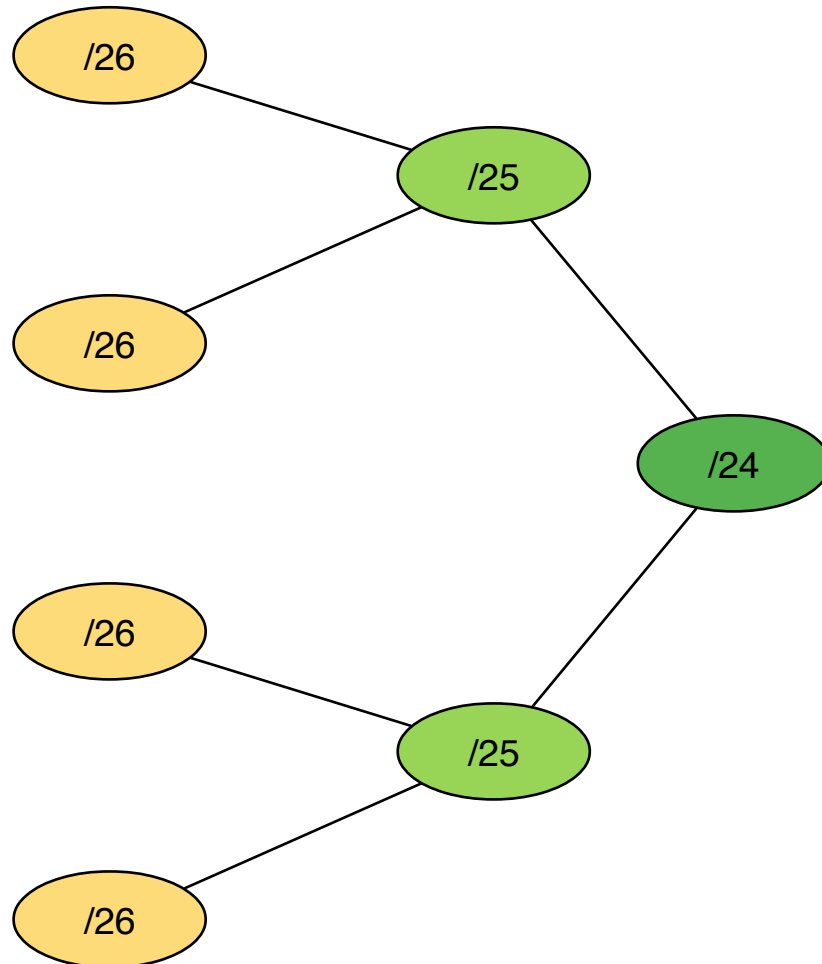
Example: 192.168.10.0/25

Subnets to:

192.168.10.0/26

192.168.10.64/26

Networks: Supernetting



Subtract one bit from netmask
& the network joins together
This is supernetting

192.168.10.0/26

192.168.10.64/26

join together to form:

192.168.10.0/25

IP Numbers

- Public address space available from your NIC
 - AFRINIC, APNIC, ARIN, LACNIC, RIPE NCC
- Private address space available to anyone
 - RFC 1918 Ranges for private networks
 - 10/8, 172.16/12, 192.168/16
 - RFC 6598 for large scale NAT
 - 100.64.0.0/10

Routing

- Every host on the Internet needs a way to get packets to other hosts outside its own subnet
- Hosts that can move packets between subnets are called routers
- Packets can pass through many routers before reaching their destination

The Route Table

- All hosts (including routers) have a route table
- Route tables show which networks are connected
- They specify how to forward packets to other networks
- “ip -B route” on Linux to see v4 & v6 routes
- “netstat -rn -46” on Linux see v4 & v6 routes
- “netstat -rn” on BSD/Mac to see v4 & v6 routes

IPv4 routing table entries

Kernel IP routing table

Destination	Gateway	Genmask	Flags	MSS	Window	irtt	Iface
0.0.0.0	192.168.2.1	0.0.0.0	UG	0	0	0	eth0
192.168.2.0	0.0.0.0	255.255.255.0	U	0	0	0	eth0

root@librenms:~#

- Destination is a network address
- Gateway is a router that can forward packets
- Iface is the network interface the route will use

The default route

Kernel IP routing table

Destination	Gateway	Genmask	Flags	MSS	Window	irtt	Iface
0.0.0.0	192.168.2.1	0.0.0.0	UG	0	0	0	eth0
192.168.2.0	0.0.0.0	255.255.255.0	U	0	0	0	eth0

root@librenms:~#

- Note the first entry in this route table
- It matches every possible IP address
- This is the default route
- It must be a router capable of forwarding traffic

More Complex Routing

- A router's route table could look like this
- Note multiple interfaces & multiple networks
- There's also a gateway for other networks

Destination	Gateway	Genmask	Flags	Interface
192.168.0.0	0.0.0.0	255.255.255.0	U	eth0
192.168.1.0	0.0.0.0	255.255.255.0	U	eth1
192.168.2.0	0.0.0.0	255.255.254.0	U	eth2
192.168.4.0	0.0.0.0	255.255.252.0	U	eth3
0.0.0.0	192.168.1.1	0.0.0.0	UG	eth0

Summary

- Layers & Layer Models
- Encapsulation
- Frames, Datagrams, Segments, Packets
- TCP/IP Protocol Suite
- IP Addressing
- IP Routing