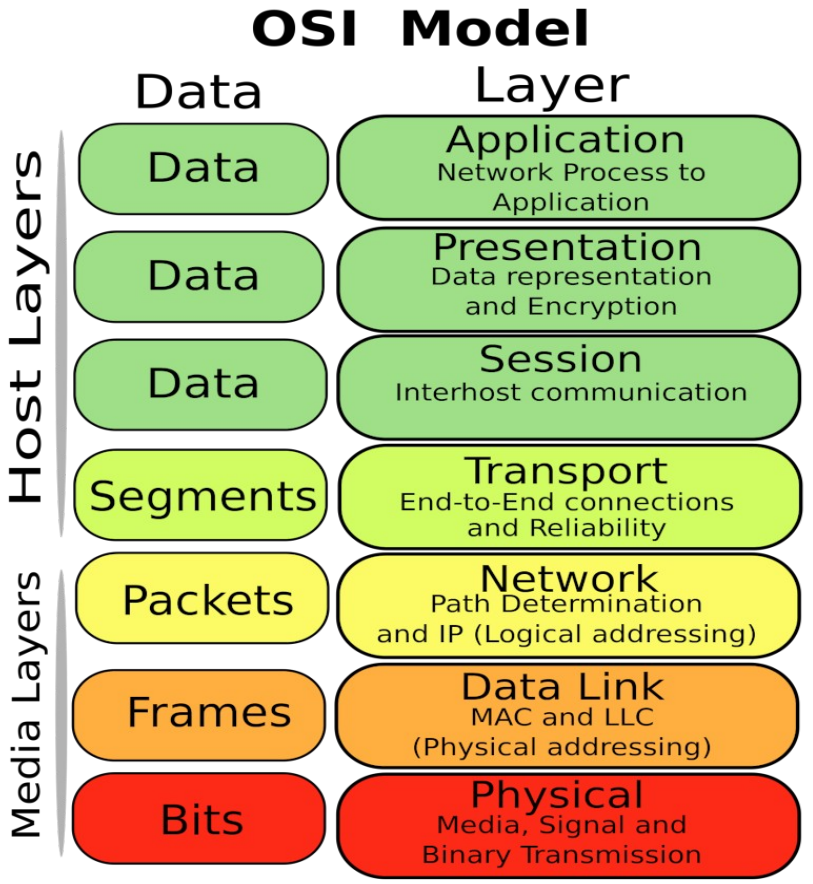
# Networking

Linux System Administration
Fall 2025

# Practical Networking

- Most network traffic is IPv4 or IPv6
  - TCP or UDP
- Most traffic is carried by Ethernet, Fiber or Microwave
  - 802.*



# Evi's Extended OSI Model

- Networking requires many policy decisions
  - Security vs. convenience
  - Interfacing with other networks and groups
  - Common protocols



# Common IP Tools

- `ping`
  - Tests end-to-end connectivity
- `traceroute` or (or `tracert` on Windoze)
  - Shows the route the packet takes
- `ip addr` or `ifconfig`
  - IP configuration
- `ip route` or `route` or `netstat -nr`
  - Configure or show routing

# Ethernet

- Developed in 1970s
  - Commercialized 1980s
  - Shared medium ("ether")
  - Retransmit on collision
- Layer 2
  - 48bit MAC address
  - Data frames
- Clear winner over competitors
  - Marketing over performance
- Initially used taps on coaxial cable with terminators
  - 10base2 & 10base5
- Largely replaced by copper unshielded twisted pair
  - 10baseT, 100baseT, 1000baseT (RJ45)
  - >1G typically fiber
  - Current max 400Gb/s

# Generic Network Hardware

- hub, repeater, concentrator
  - Retransmits all frames on all ports
  - No configuration
- switch, bridge
  - Retransmits frames based on MAC address (layer 2)
  - Self-configuration (Spanning Tree Protocol, ...)
- router
  - Routes packets based on IP header (layer 3)
  - Requires configuration

# Copper (Unshielded Twisted Pair)

- RJ-45
- Cat5, Cat6, Cat7, ...
- T568B colors
- Straight/Cross-over
- Max length ~100m
- Easy field termination
- Power-over-ethernet
- Autonegotiation
- Failure modes
  - Lightning & RF
  - Rodents

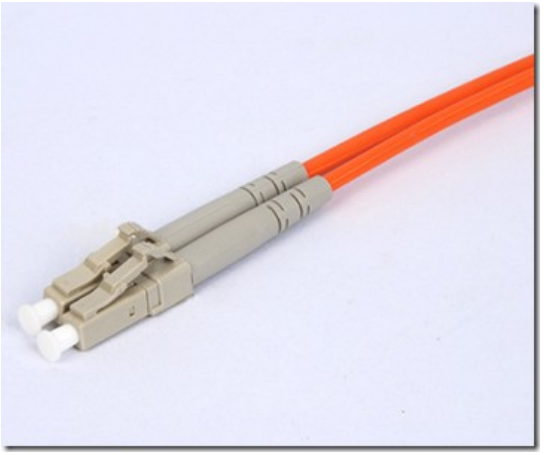


# Fiber Optics

- LC, SC, ST, ...
- Single/multi mode
  - OS1,OM1,OM2,OM3
  - Color denotes type
- Limited bending radius
- Many km runs
- Difficult field termination

- Failure modes
  - Backhoe
  - Bend radius





# Radio Frequency (UHF & microwave)

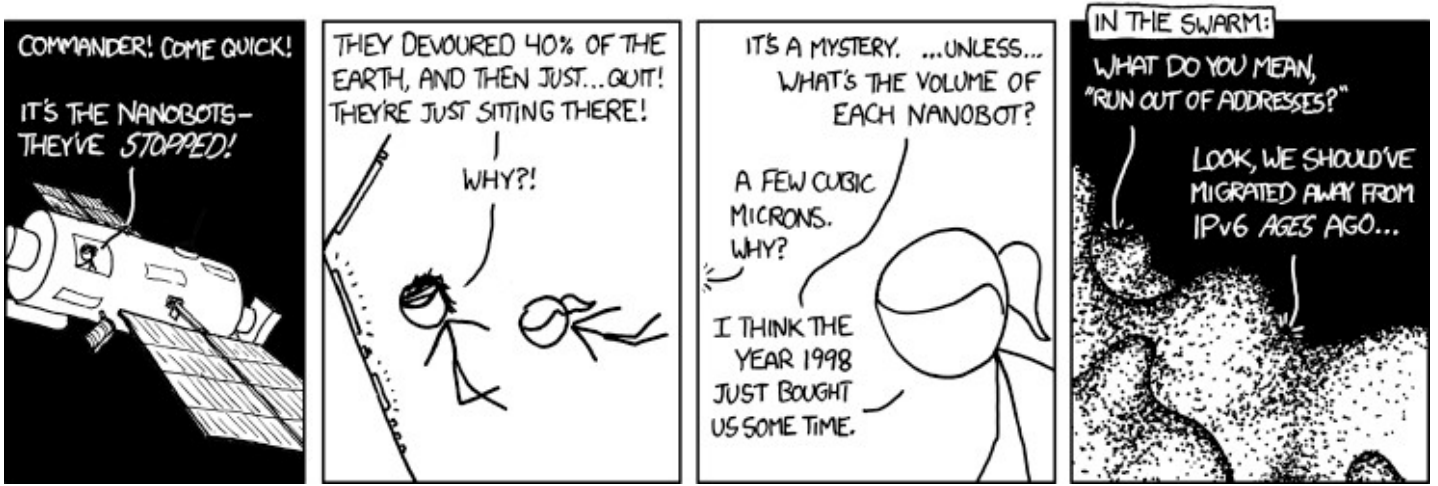
- 300MHz-1GHz
  - << 1Mb/s speeds
  - Beyond line of sight
- 2.4 GHz & 5GHz WiFi
  - Unlicensed
- 3-30 GHz licensed
  - Usually point-to-point
- 60 GHz point-to-point
- Generally 802.11*
  - Speed of light matters in very long shots
  - SSID, modes, modulation...
- Antennas
  - Polarization
  - Omni, Sector, High gain
- Noise and fade

# Internet Protocol (IP)

- Packet switched network
  - Introduced in 1981
  - Replaced older circuit switched networks
- Advantages
  - Decentralized, hierarchical configuration
  - Robust, dynamic routing
  - IPv4 supports 4,294,967,296 unique addresses

# IP Versions

- IPv1, IPv2, IPv3 - Early development versions (messing with TCP/IP)
- IPv4 - What most networks are based on (32 bit, 4 billion addresses)
- IPv5 - Experimental Quality of Service addition
- IPv6 - Next generation IP (128 bit addresses)



# IPv4 Addresses

- Every device must have a unique address
- IPv4 use 32 bits organized as 4 octets
  - 01100100 01000000 00000000 00101010 = 100.64.0.42
  - subnet part host part
  - All 32 bits are used to identify the device
  - The subnet part determines what can be reached directly
  - Typically written as 100.64.0.42/24

# What is a subnet?

- Part of a greater (interconnected) network
- Group of IP addresses that can directly communicate
  - Can find device at Layer 2 via Ethernet or WiFi
  - Extend using hub, switch, bridge
  - Same leading bits (subnet address) but unique hosts bits
- Subnets are connected by routers
  - Determines forwarding based in IP address

# Classfull Inter-Domain Routing

- Prior to 1993 routers assumed classes
  - Class A = 1.x.x.x to 126.x.x.x
    - 126 nets of 16,777,326 hosts each
  - Class B = 128.0.x.x 191.255.x.x
    - 16,384 nets of 65,536 hosts each
  - Class C = 192.0.0.x – 223.255.255.x
    - 2,752,512 nets of 256 hosts each
  - Class D = 224.x.x.x-239.x.x.x
    - Multicast
  - Special cases 0.x.x.x, 127.x.x.x, 100.64.x.x

# Classless Inter-Domain Routing (CIDR)

- Breaks network/host at any bit position
- Example:  10.30.20.0/24
  - Class A address (10.x.x.x)
  - /24 makes it a class C (256 addresses)
- Example: 10.30.20.96/29
  - No classed equivalent
  - 8 addresses
  - Netmask 11111111 11111111 11111111 11111000 = 255.255.255.248
- Extended the life of IPv4
  - Many more unique networks

# Netmask

- Mask used to isolate subnet and host parts
  - All 1 bits must be left of all 0 bits

```
/16 11111111 11111111 00000000 00000000 = 255.255.0.0
/24 11111111 11111111 11111111 00000000 = 255.255.255.0
/29 11111111 11111111 11111111 11111000 = 255.255.255.248
```

- Network address = Host Address & Netmask
  - Sets host bits to all zeroes
- Hosts on same subnet can communicate directly
  - All others need to be routed

# Special host part addresses

- Network address (host part all zeroes)
  - 100.64.42.0/24 01100100 01000000 00101010 00000000
- Broadcast address (host part all ones)
  - 100.64.42.255/24 01100100 01000000 00101010 11111111
- Hosts cannot be assigned network or broadcast address
  - except /31 and /32
- Other *special* addresses are just by convention
  - Gateway typically the lowest address or highest valid address
    - 100.64.0.254/24 or 100.64.0.1/24
- `ipcalc` is convenient for doing network calculations

# Address Resolution Protocol (ARP)

- Layer 2 - Layer 3 address translation
  - IPv4 address to MAC address
  - Requires no user configuration
- Contains Layer 2 & 3 addresses
  - OPER field determines action
    - request:  What is the MAC address for this IP?
      Sent to broadcast MAC address
    - reply: This is the MAC address for this IP.
      Sent to request MAC address
  - Exchange requires just one packet each way
    - Other hosts may learn IPv4:MAC from the exchange

# Common IP Protocols

- `ICMP` **I**nternet **C**ontrol **M**essage **P**rotocol
  - Diagnostic  - echo, unreachable, TTL exceeded
  - Control - redirect
- `UDP` **U**ser **D**atagram **P**rotocol
  - Unreliable, unordered packets (datagrams)
  - Connection-less between IP:port pairs
- `TCP` **T**ransmission **C**ontrol **P**rotocol
  - Reliable, ordered, error-checked continuous stream
  - Connection oriented between IP:port pairs

# Ports (added by TCP & UDP)

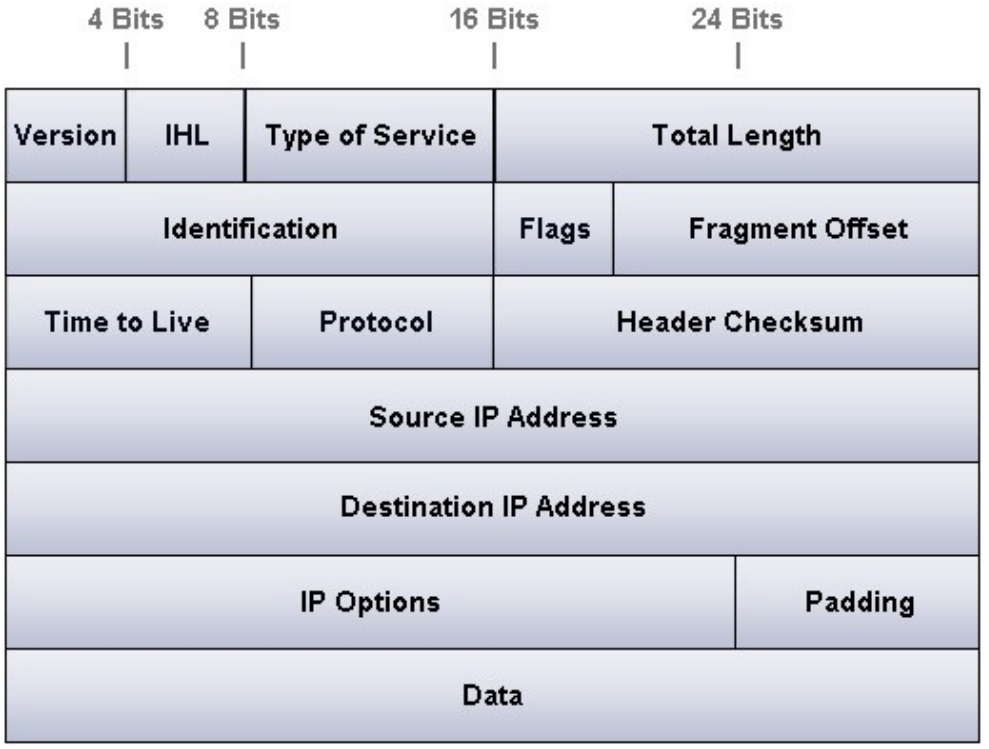
- 16 bit number 1-65535
- Defines a process to talk to
- Ports 1-1023 are *well-known* ports
  - Defined in `/etc/services`
  - Used by programs to give ports symbolic names
- Ephemeral ports (>1023) mostly used for user programs
- Notation  IP:port
  - `100.64.0.42:22` means `100.64.0.42` port `22`
    - TCP or UDP

# Example /etc/services

```
ftp-data 20/tcp
ftp      21/tcp
ssh      22/tcp
telnet   23/tcp
smtp     25/tcp
domain   53/tcp
domain   53/udp
http     80/tcp
pop3     110/tcp
ntp      123/udp
imap2    143/tcp
snmp     161/udp
snmp     161/tcp
snmp-trap 161/tcp
snmp-trap 161/ucp
https    443/tcp
syslog   514/udp
imaps    993/tcp
pop3s    995/tcp
nfs      2049/tcp
nfs      2049/udp
mysql    3306/tcp
```

# Anatomy of an IP packet

- Always big-endian
- Version=4
- IHL=IP Hdr Len
- Type of Service
  - Min delay
  - Max throughput
- Flags & Frag Off
  - Large packets
- Protocols add additional header in data section

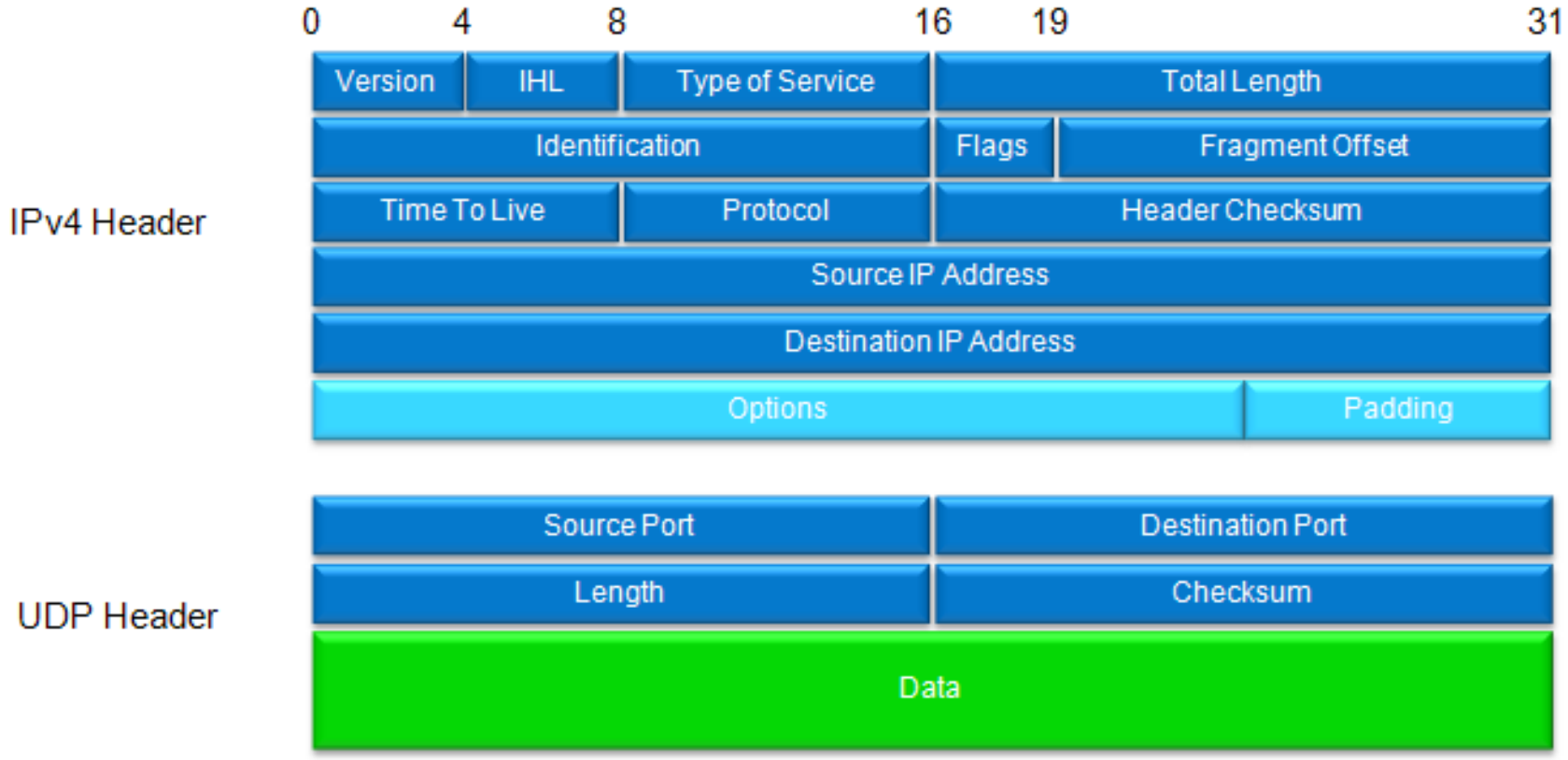


# Anatomy of an ICMP Packet

**IP Datagram**

| | **Bits 0–7** | **Bits 8–15** | **Bits 16–23** | **Bits 24–31** |
|---|---|---|---|---|
| **IP Header (20 bytes)** | Version/IHL | Type of service | Length | |
| | Identification | | *flags* and *offset* | |
| | Time To Live (TTL) | Protocol | Checksum | |
| | Source IP address | | | |
| | Destination IP address | | | |
| **ICMP Header (8 bytes)** | Type of message | Code | Checksum | |
| | Header Data | | | |
| **ICMP Payload (*optional*)** | Payload Data | | | |

# Anatomy of a UDP Packet



# UDP Applications

- Applications that transmit discrete chunks
  - DNS - Domain Name Service
  - NTP - Network Time Protocol
- Applications with time sensitive information
  - Voice over IP (VoIP)
- Applications that add robustness
  - NFSv2, mosh

# Transmission Control Protocol (TCP)

- Bidirectional virtual circuit
  - Input = Output
  - Traffic arrives in order
  - Retransmit, congestion,etc happens invisibly
  - No inherent framing
- Used by many higher level protocols
  - Terminal connections (ssh, telnet)
  - Data transfer (HTTP, FTP, scp, rsync)

# TCP stuff you should know

- TCP exchanges
  - 3-way handshake: SYN, SYN-ACK, ACK
  - Termination:  FIN, FIN-ACK
  - Reset: RST, RST-ACK
- Flow control and sequence numbers
- Keep-alive
- TCP based protocols need to define segments
  - Example:  HTTP uses `\r\n`

# Configuring an IPv4 interface

- Must have
  - IP address
  - Netmask
  - Default gateway
- Nice to have
  - DNS server address
  - NTP server address

- Example:
  - `IP 100.64.0.42/24`
  - `GW 100.64.0.1`
  - `DNS1 128.138.240.1`
    `DNS2 128.138.130.30`
  - `NTP 132.163.97.1`
    - `time-a.www.nist.gov`

# Manual Interface Configuration

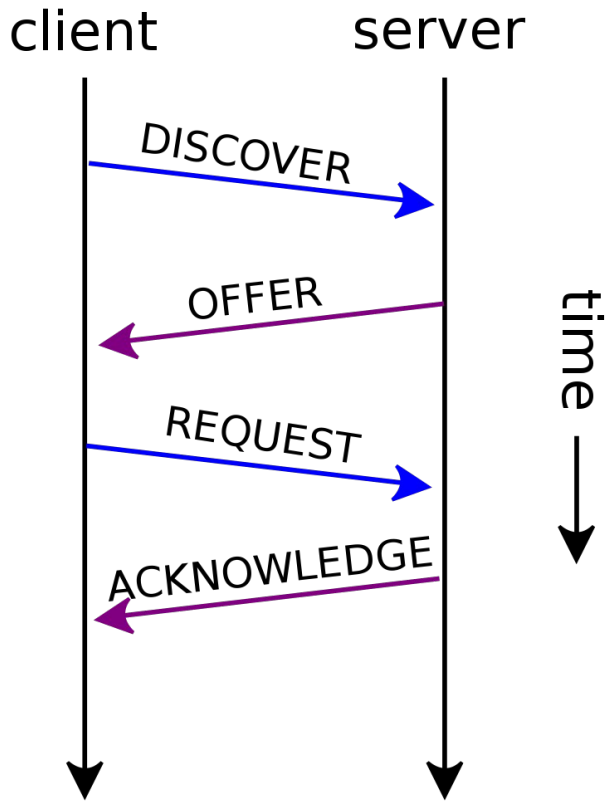
- `ip link`
  - Show available interfaces
- `ip addr add 100.64.42.6/24 dev ens192`
  - Sets IP address for `ens192` to `100.64.42.6`
  - Adds route to `100.64.42.0/24` via `ens192`
- `ip route add default via 100.64.42.1`
  - Adds default route via gateway `100.64.42.1`
  - Reachable on `100.64.42.0/24` using `ens192`
- `ip neigh` shows IP and MAC addresses

# DHCP
# (Dynamic Host Configuration Protocol)

- Server assigns unique IP address
  - Centralizes configuration of IP address space
  - Straddles Layer 2 and Layer 3
  - IP from pool or reservation by MAC address
    - IP, netmask and default gateway
- Typically also sets DNS, NTP, etc.

# DHCP Exchange

- Broadcast
  - > DISCOVER
    - I need an IP address
- Unicast
  - < OFFER
    - You may use this IP
  - > REQUEST
    - I want to take you up on that
  - < ACKNOWLEDGE
    - You got it



# Linux interface names

- Traditional names `eth0`, `eth1`, `eth2`, ...
  - Changes depending order of discovery
- Predictable names
  - `eno*` (onboard) e.g. `eno1`
  - `ens*` (hotplug) e.g. `ens192`
  - `enp*` (bus) e.g. `enp2s0`
  - `enx*` (MAC) e.g. `enx525400d5e0fb`
- Names set by `udev`

# Persistent Interface Configuration

- Redhat
  - Set in `/etc/sysconfig/network-scripts`
  - Configuration by interface, e.g. `ifcfg-ens192`
- Debian
  - Set in `/etc/network/interfaces`
  - All interfaces configured in one file

# RedHat Interface Example

- /etc/sysconfig/network-scripts/ifcfg-ens192

```
DEVICE=ens192
ONBOOT=true
BOOTPROTO=static
IPADDR=100.64.42.6
PREFIX=24
GATEWAY=100.64.42.1
```

# Debian Interface Example

- /etc/network/interfaces

```
auto lo ens192

iface lo inet loopback

allow-hotplug ens192
iface ens192 inet static
  address 100.64.42.6
  netmask 255.255.255.0
  gateway 100.64.42.1
```

# NetworkManager

- Developed by RedHat to handle laptops and in particular wireless networking
  - GUI, `nmtui` and `nmcli`
- Getting harder to remove
  - Honors `/etc/sysconfig/network-scripts`
  - RHEL 9 prioritize key file format configurations in `/etc/NetworkManager/system-connections/`

# RedHat Key File Example

- /etc/NetworkManager/system-connections/ens192.nmconnection

```
[connection]
id=ens192
type=ethernet
interface-name=ens192

[ipv4]
method=manual
address1=100.64.42.6/24,100.64.42.1
```

# Routing

- Router connects multiple subnets
- The next hop must be reachable at Layer 2
- Each subnet has an associated target
  - via (IP address of gateway)
    - Packet sent to the MAC address of the gateway
  - interface
    - Packet sent to the MAC address of destination
- Selects **best** match out of all possibilities