CS 447/647

TCP/IP Networking

What are the core Internet Protocols? (IP, ICMP, UDP, and ARP)

How is the Internet governed? (ICANN, ISOC and IGF)

How are standards developed? (RFC)

What are the 5 layers of the OSI Model?

What are the Protocol Data Units for the 5 layers?

What are some common protocols for each layer?

Common network utilities. (dig, telnet, nmap, ip, and nc)

References

Goralski, W. (2017). The Illustrated Network: How TCP/IP works in a modern network. Amsterdam: Elsevier.

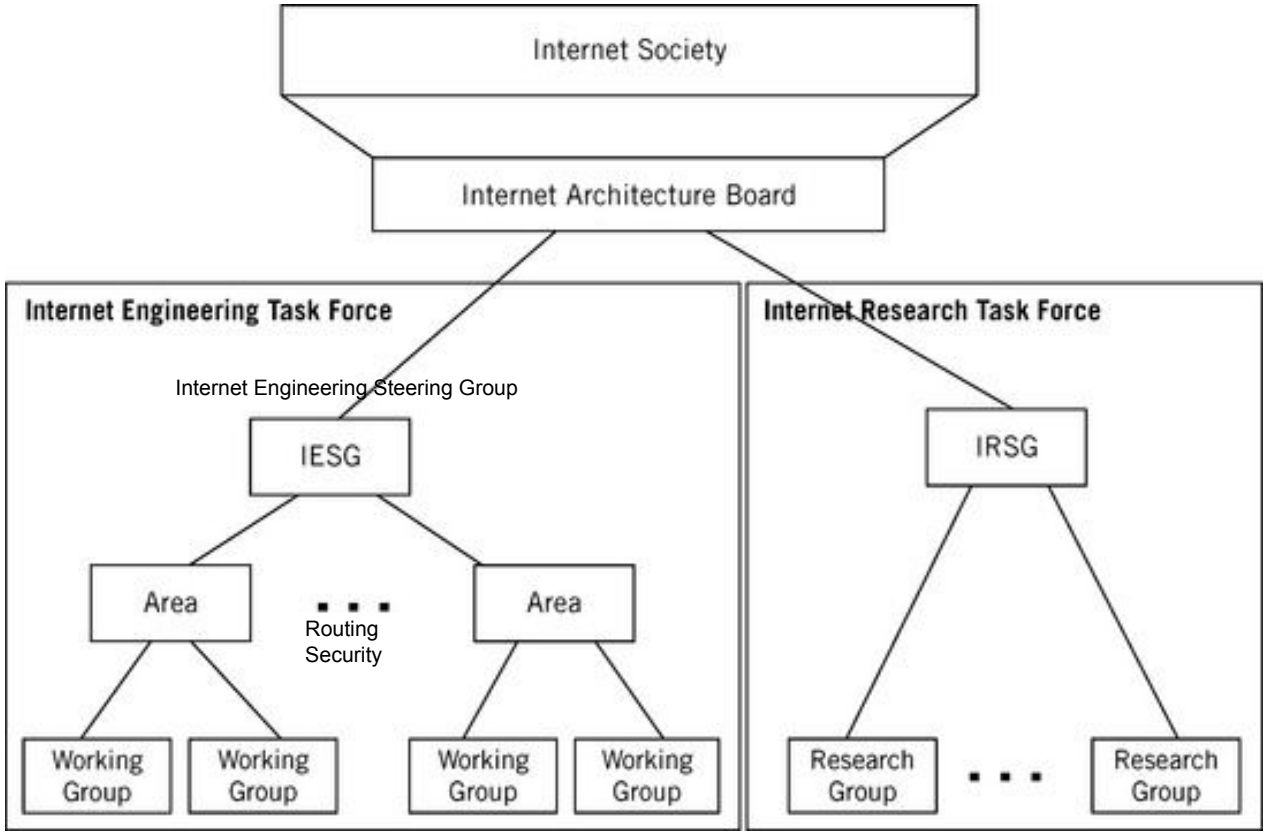
https://learning.oreilly.com/library/view/the-illustrated-network/9780128110287

TCP/IP Networking

- Networking is important
- TCP/IP underpins the Internet
  - Web
  - Email
  - Zoom (UDP)
- TCP/IP is flexible
  - OS Independent
  - Hardware Independent
  - Works on any size or topology

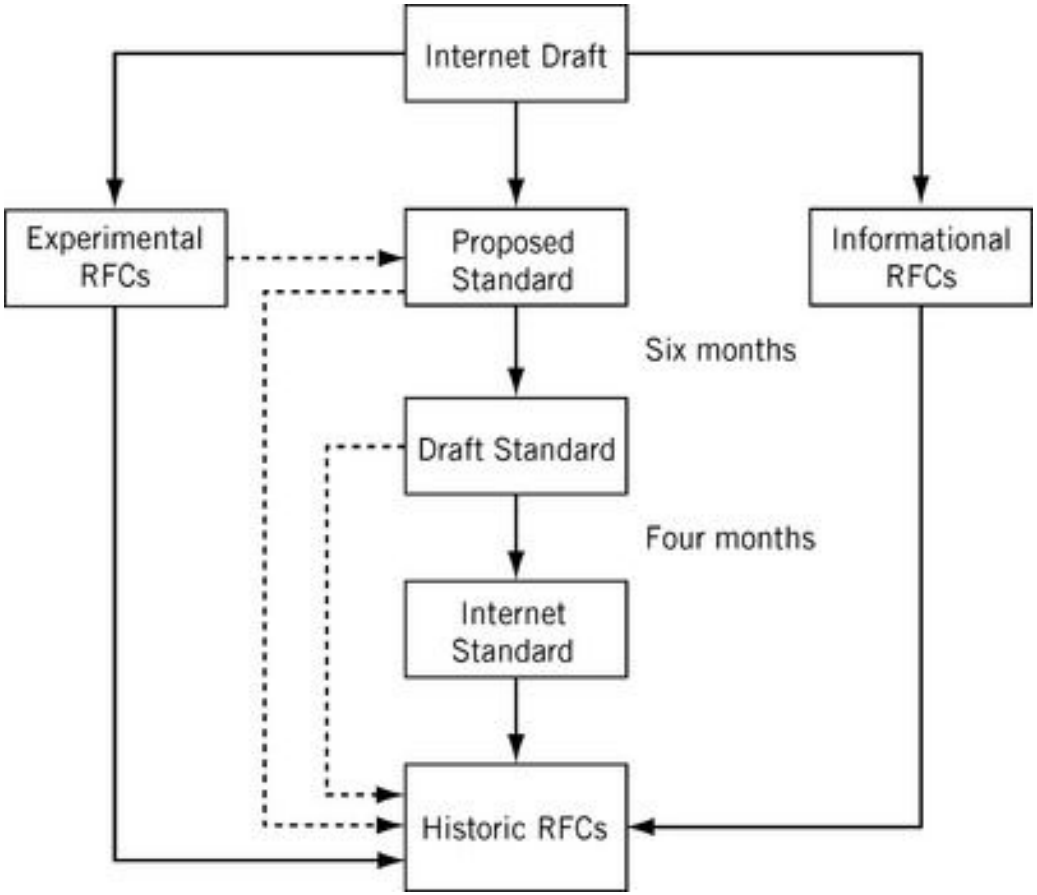
TCP/IP Networking & The Internet

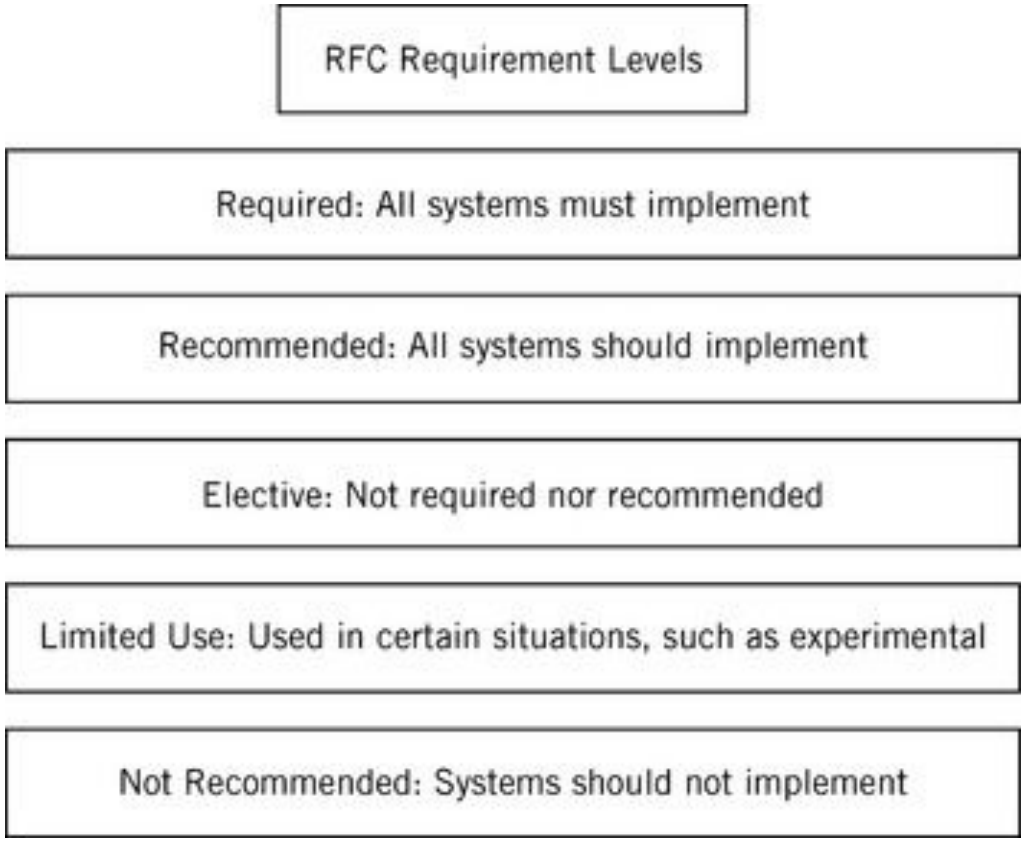
- TCP/IP and the Internet have a shared history
  - TCP was created in 1974 by Vint Cerf
    - https://www.cs.princeton.edu/courses/archive/fall06/cos561/papers/cerf74.pdf
- Progenitor was a network called ARPANET in 1969
  - In the 1980's in transitioned into the commercial Internet
- Collaboratively Managed
  - ICANN - Internet Corporation for Assigned Names and Numbers
    - Enforcement Capabilities
    - Controls the allocation of IPs, domains and protocol ports.
  - ISOC - Internet Society
    - Technical development through IETF - Internet Engineering Task Force
  - IGF - Internet Governance Forum
    - Created by the UN.
    - Used for policy-based discussions


Internet Society
Internet Architecture Board
Internet Engineering Task Force
Internet Research Task Force
Internet Engineering Steering Group
IESG
IRSG
Area
Area
Routing
Security
Working Group
Working Group
Working Group
Working Group
Research Group
Research Group

Network standards and documentation

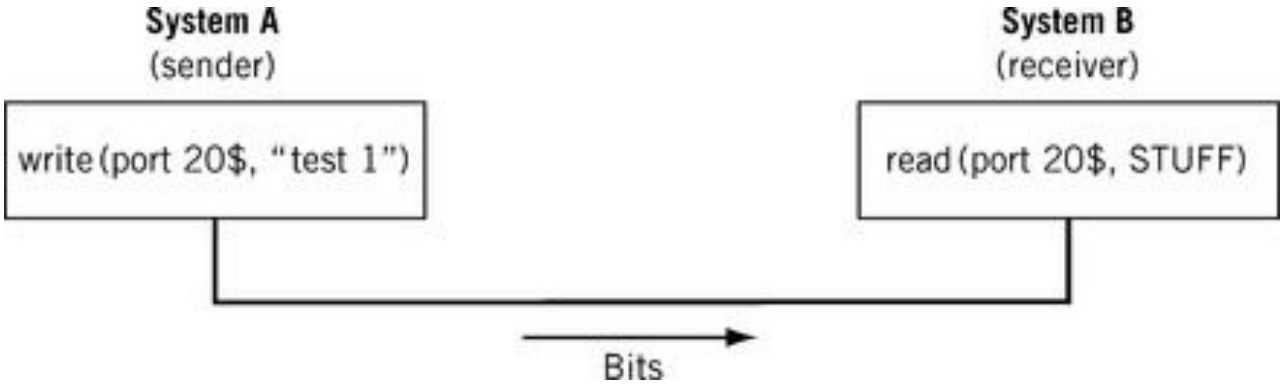
- RFCs - Request for Comments
  - Protocol Standards
    - TFTP, SMTP, HTTP, DNS, etc.
  - Proposed Changes
    - SMTP Require TLS Option - REQUIRETLS
      - https://www.rfc-editor.org/rfc/rfc8689.txt
  - Informational Bulletins
    - 50 years of RFCs
      - https://www.rfc-editor.org/rfc/rfc8700.txt
- RFCs can be assigned
  - BCP - Best Current Practice
    - Network Time Protocol Best Current Practices
  - STD - Standard
  - FYI - For Your Information


Internet Draft
Experimental RFCs
Proposed Standard
Informational RFCs
Six months
Draft Standard
Four months
Internet Standard
Historic RFCs

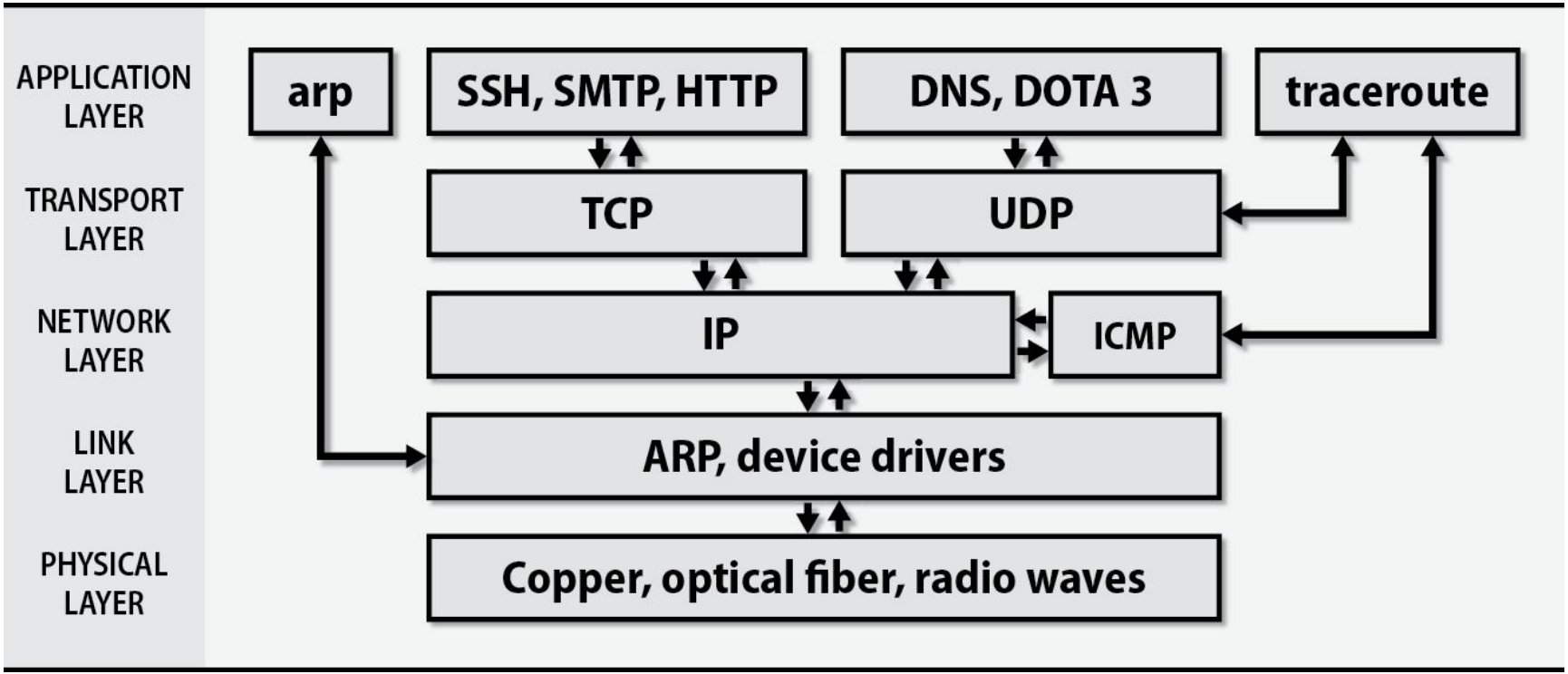

RFC Requirement Levels
Required: All systems must implement
Recommended: All systems should implement
Elective: Not required nor recommended
Limited Use: Used in certain situations, such as experimental
Not Recommended: Systems should not implement

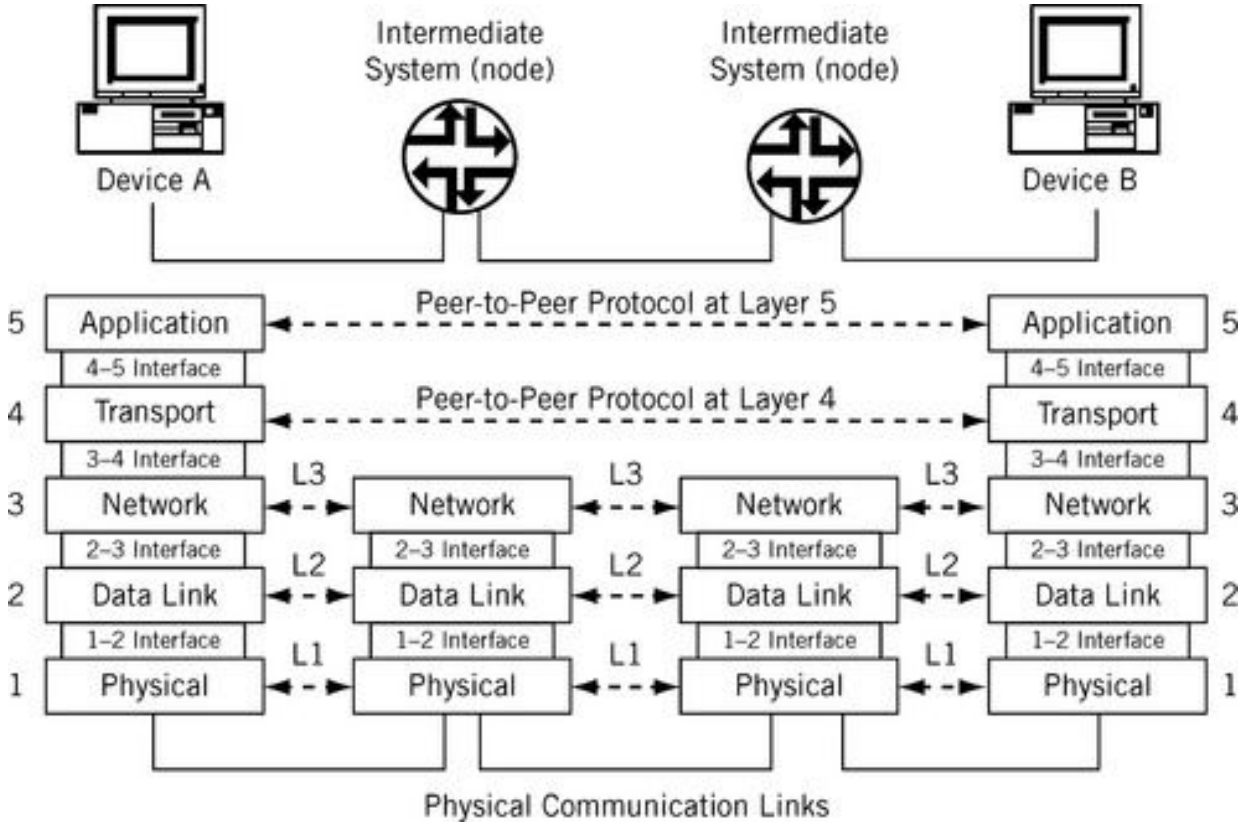
Networking Basics

- IP - Internet Protocol
  - routes data from one machine to another
- ICMP - Internet Control Message Protocol
  - Low-level support for IP error message, routing and debugging
  - ping, traceroute
- ARP - Address Resolution Protocol
  - Translates IP to hardware address (MAC)
- UDP - User Datagram Packet
  - Unreliable one-way delivery
- TCP - Transmission Control Protocol
  - Reliable full-duplex and error corrected conversations


System A
(sender)
write (port 20$, "test 1")
System B
(receiver)
read (port 20$, STUFF)
Bits

TCP/IP Layering model



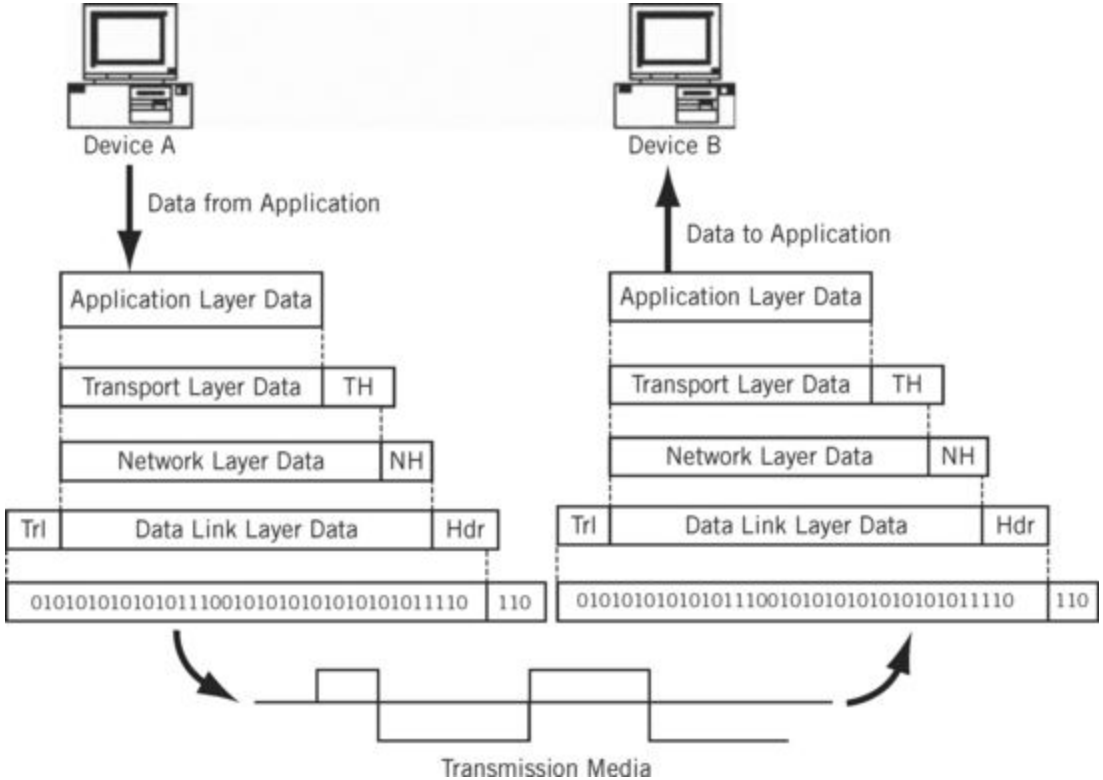

Intermediate System (node)
Intermediate System (node)
Device A
Device B
5
Application
Peer-to-Peer Protocol at Layer 5
Application
5
4–5 Interface
4–5 Interface
4
Transport
Peer-to-Peer Protocol at Layer 4
Transport
4
3–4 Interface
3–4 Interface
L3
L3
L3
3
Network
Network
Network
Network
3
2–3 Interface
2–3 Interface
2–3 Interface
2–3 Interface
L2
L2
L2
2
Data Link
Data Link
Data Link
Data Link
2
1–2 Interface
1–2 Interface
1–2 Interface
1–2 Interface
L1
L1
L1
1
Physical
Physical
Physical
Physical
1
Physical Communication Links

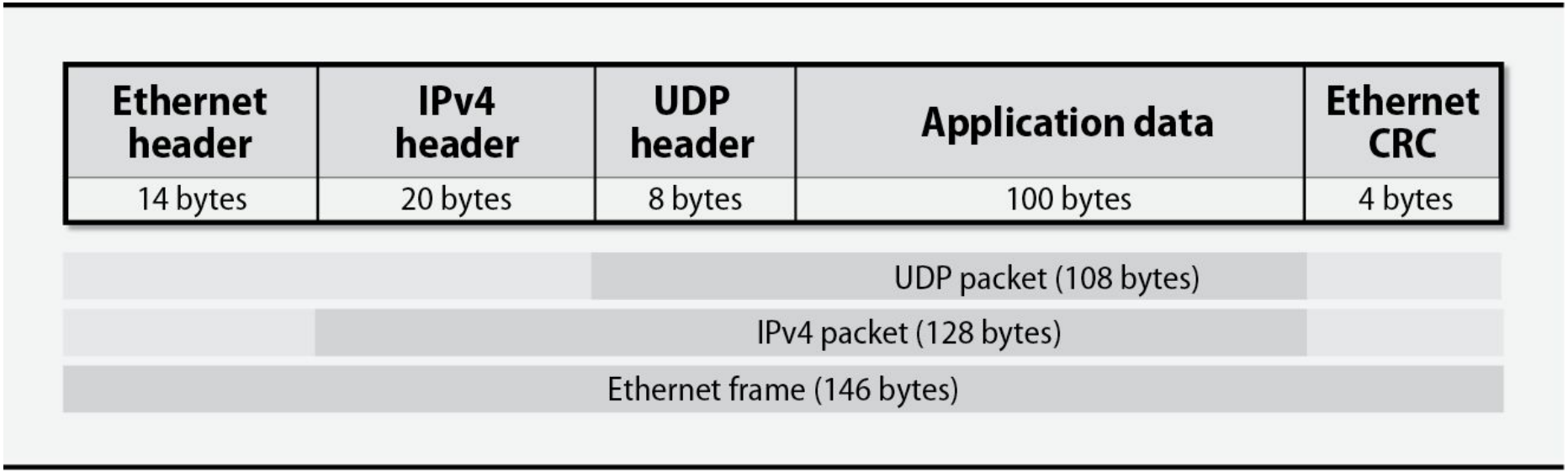
IPv4 and IPv6

- IPv4
  - 32 bit addresses
  - 4,294,967,296 addresses
  - NAT - Network Address Translation
- IPv6
  - 128 bit addresses
  - IPsec - built in authentication and encryption
  - No checksum
  - 30% of google.com visits
- Adoption of IPv6 is slow
  - Amazon, Bing, Wordpress, craigslist
  - Waiting for services to be IPv6 only
  - Cleaned up version of IPv4
    - Python2
      - 2010 to 2020

Packet Encapsulation

- Hardware
  - Ethernet, token ring, Infiniband, Omni-path
- Data travels as packets
  - Max length is dictated by the link layer (2)
  - Packet header has source and destination
  - Checksums, protocol options
  - Handling instructions (TTL)
  - Payload
- Encapsulation
  - Packets are added to by each layer by the sender
  - Each layer is removed by the receiver

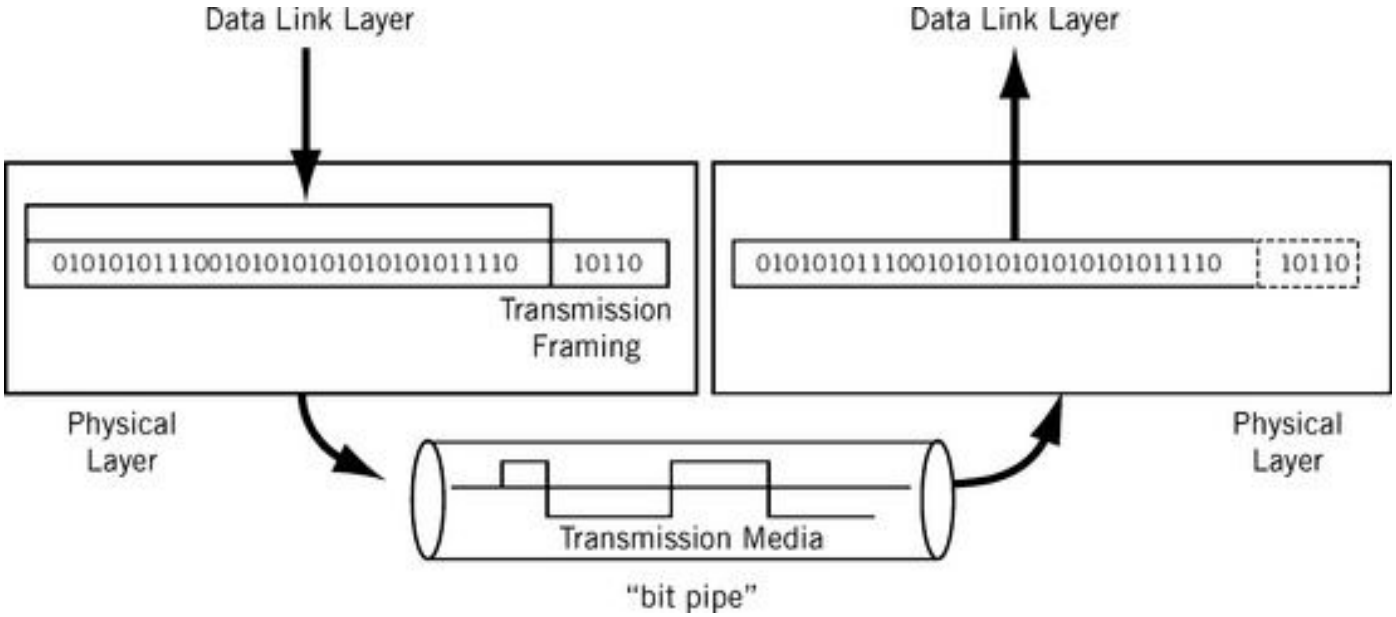

Device A
Device B
Data from Application
Data to Application
Application Layer Data
Transport Layer Data
TH
Network Layer Data
NH
Trl
Data Link Layer Data
Hdr
0101010101010111001010101010101011110
110
Transmission Media


Ethernet header
IPv4 header
UDP header
Application data
Ethernet CRC
14 bytes
20 bytes
8 bytes
100 bytes
4 bytes
UDP packet (108 bytes)
IPv4 packet (128 bytes)
Ethernet frame (146 bytes)

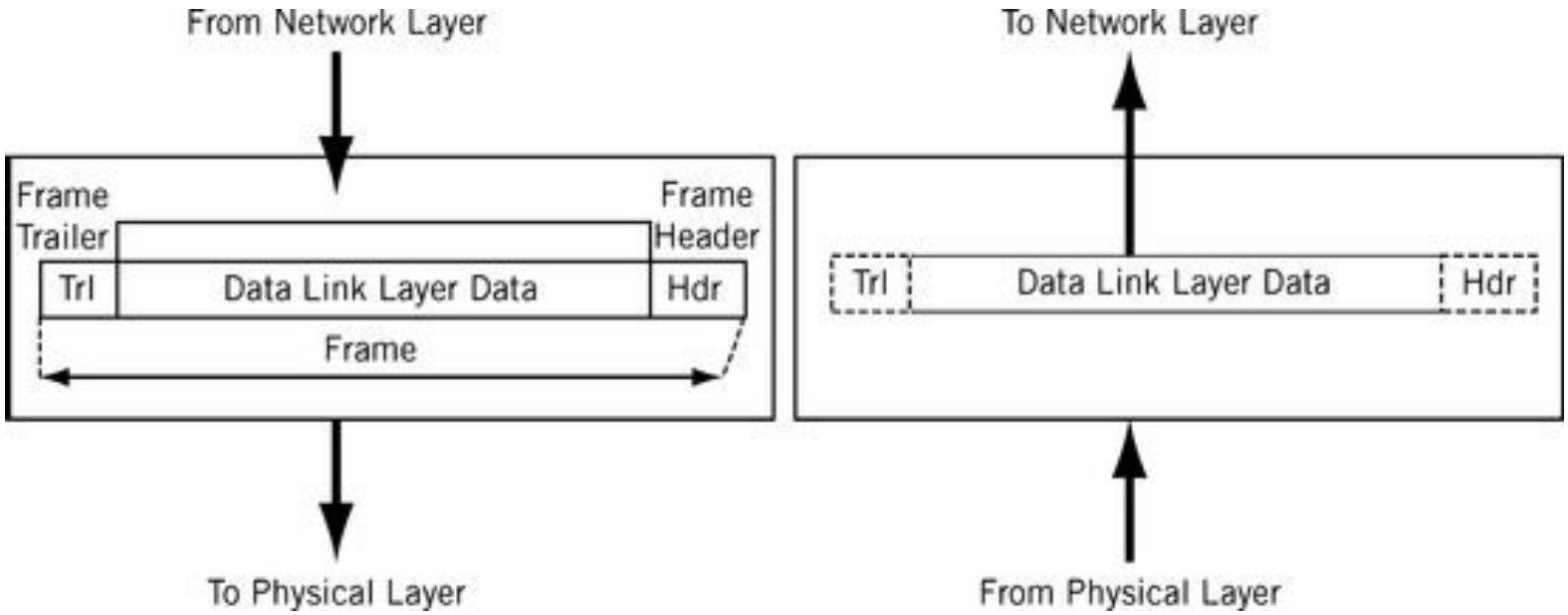
Ethernet Framing

- Adding extra bits to a packet
- Link layer adds headers to packets
  - Header contains addresses
  - Checksums
- Link layer adds separators between packets
- Two parts of link layer
  - Media Access Control - Deals with hardware, puts packets onto the wire
  - Logical Link Control - Ethernet framing

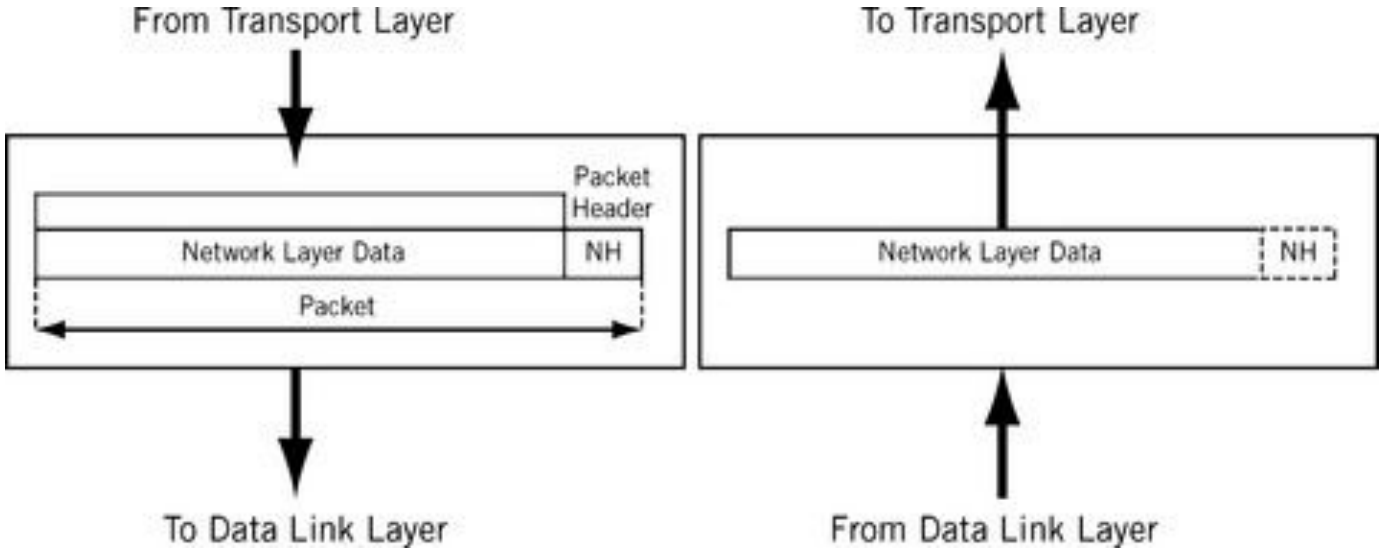
Physical


Data Link Layer
Data Link Layer
01010101110010101010101010111110
10110
Transmission
Framing
01010101110010101010101010111110
10110
Physical
Layer
Physical
Layer
Transmission Media
"bit pipe"

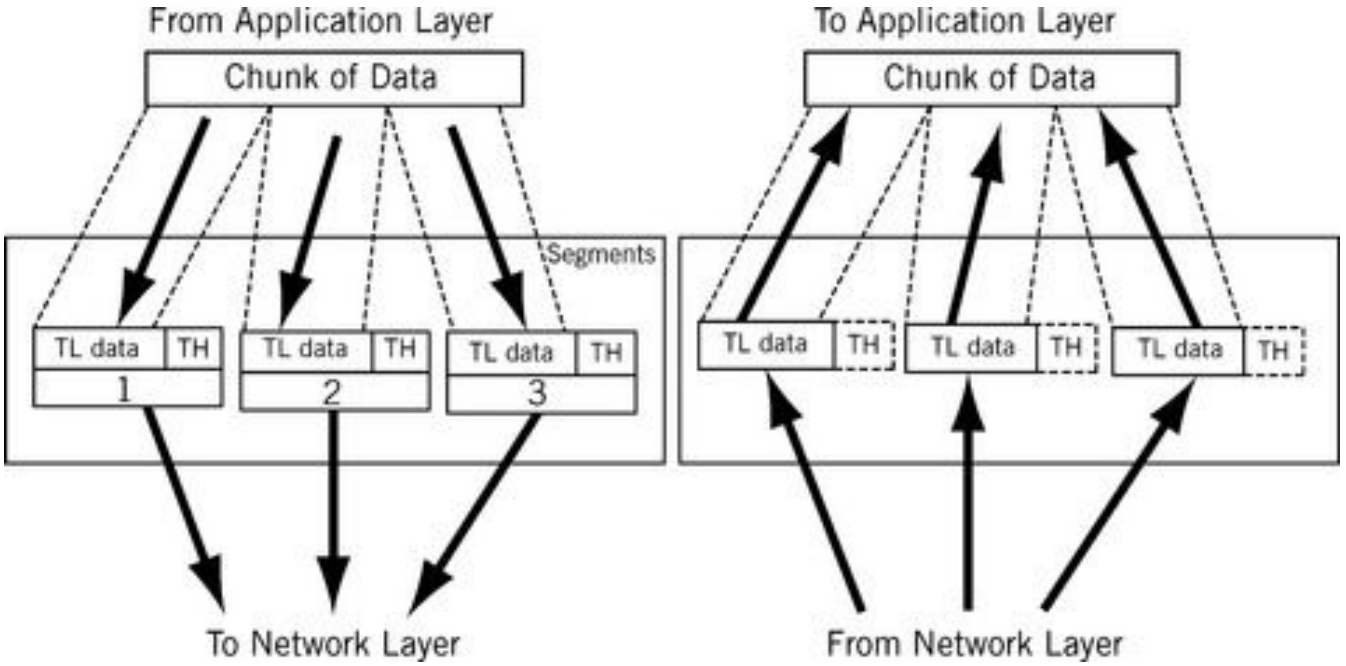
Data Link


From Network Layer
Frame
Trailer
Frame
Header
Trl
Data Link Layer Data
Hdr
Frame
To Physical Layer
To Network Layer
Trl
Data Link Layer Data
Hdr
From Physical Layer

Network


From Transport Layer
Packet Header
Network Layer Data
NH
Packet
To Data Link Layer
To Transport Layer
Network Layer Data
NH
From Data Link Layer

Transport


From Application Layer
Chunk of Data
Segments
TL data
TH
1
TL data
TH
2
TL data
TH
3
To Network Layer
To Application Layer
Chunk of Data
TL data
TH
TL data
TH
TL data
TH
From Network Layer

Maximum Transfer Unit

- Packet size is limited
  - Hardware
    - E1000 - 16,298 bytes
  - Protocol
    - Ethernet - 1500 bytes

Network type	Maximum transfer unit
Ethernet	1,500 bytes (1,492 with 802.2 framing)
IPv6 (all hardware)	At least 1,280 bytes at the IP layer
Token ring	Configurable [a]
Point-to-point WAN links (T1, T3)	Configurable, often 1,500 or 4,500 bytes

a. Common values are 552; 1,064; 2,088; 4,508; and 8,232. Sometimes 1,500 to match Ethernet.

MTU

- IPv4 Packets are split to conform to the MTU
  - Test with
    - ping -s 4500 cs447.cse.unr.edu
- Fragmentation happens in-flight by routers
  - IPv6 moves this to the sender
    - 1,280 bytes
- Lowest MTU link can be found with “do not fragment” flag
  - ICMP error response
    - Contains network info for lowest-MTU link
- TCP does automatic MTU discover
  - UDP does not

Packet Addressing

- MAC Address
  - Hardware
- IPv4 and IPv6 addresses
  - Software
- **Hostnames**
  - Humans

Hostnames

- Domain Name Server
  - A - IPv4
  - AAAA - IPv6
  - PTR - IP to Hostname aka. Reverse lookup
- /etc/hosts
  - *IP hostname hostname1 hostname2*
- Lookup with dig
  - dig A google.com
  - dig A cs447.cse.unr.edu @8.8.8.8
  - dig A cs477.cse.unr.edu @134.197.5.1

Ports

- IP is an address. IE: 127.0.0.1
- Port is a communication channel for an application
  - 1 - 65,535
- IP + Port = Socket
  - 127.0.0.1:80 = HTTP
- /etc/services defines common network services
  - grep daytime /etc/services
- Ports < 1024 reserved for root

IPv4 Address Classes

Class	1st byte[a]	Format	Comments
A	1-127	N.H.H.H	Very early networks, or reserved for DoD
B	128-191	N.N.H.H	Large sites, usually subnetted, were hard to get
C	192-223	N.N.N.H	Were easy to get, often obtained in sets
D	224-239	–	Multicast addresses, not permanently assigned
E	240-255	–	Experimental addresses

a. The value 0 is special and is not used as the first byte of regular IP addresses. The value 127 is reserved for the loopback address.

IPv4 Subnetting

CIDR

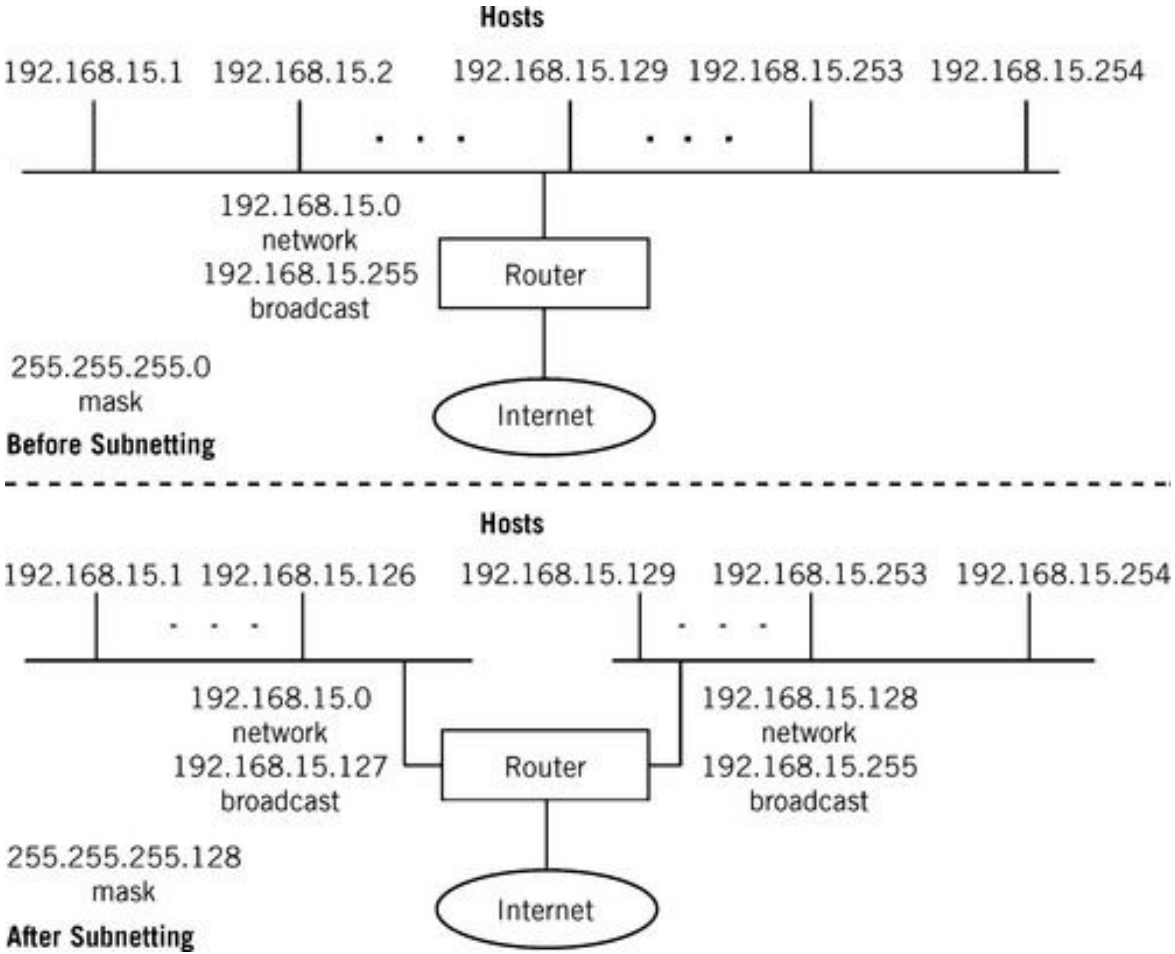
network*

	Netwok	Host	Host	Host	
Class A Subnet Mask	255	0	0	0	/8
Class B Subnet Mask	255	255	0	0	/16
Class C Subnet Mask	255	255	255	0	/24

https://www.bogotobogo.com/DevOps/AWS/aws-VPC-Virtual-Private-Cloud-1-netmast-subnet-default-gateway-CIDR.php

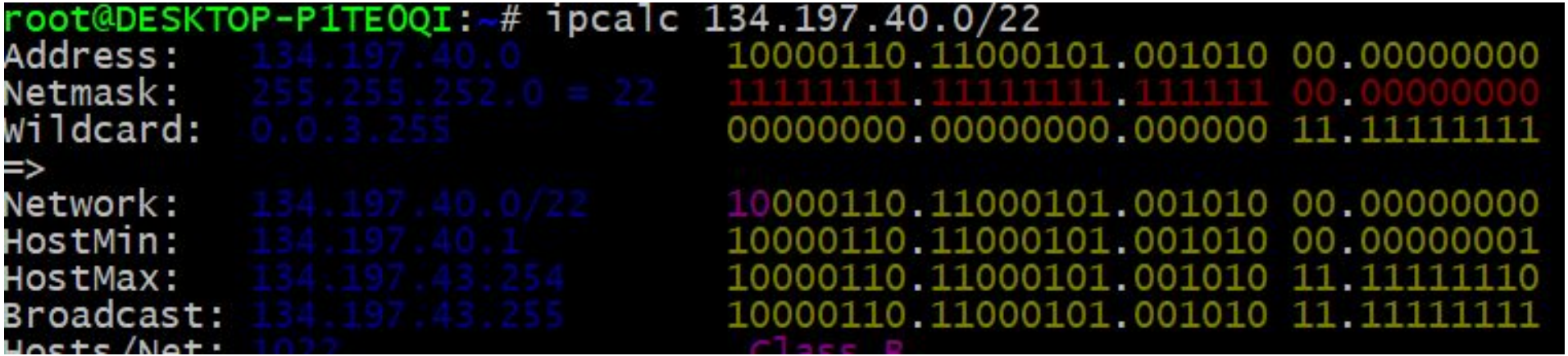
IP address	Netmask	Network	Broadcast
128.138.243.100/16	255.255.0.0	128.138.0.0	128.138.255.255
128.138.243.100/24	255.255.255.0	128.138.243.0	128.138.243.255
128.138.243.100/26	255.255.255.192	128.138.243.64	128.138.243.127

last netmask byte = 256 – net size


Hosts
192.168.15.1
192.168.15.2
192.168.15.129
192.168.15.253
192.168.15.254
192.168.15.0
network
192.168.15.255
broadcast
Router
255.255.255.0
mask
Internet
Before Subnetting
Hosts
192.168.15.1
192.168.15.126
192.168.15.129
192.168.15.253
192.168.15.254
192.168.15.0
network
192.168.15.127
broadcast
192.168.15.128
network
192.168.15.255
broadcast
Router
255.255.255.128
mask
Internet
After Subnetting

IPv4 Subnetting

apt install ipcalc



Classless Inter-Domain Routing (CIDR)

- Splitting networks for routing purposes
- Example

Site has been given a block of eight class C addresses numbered 192.144.0.0 through 192.144.7.0

- 1 network of length /21 with 2,046 hosts, netmask 255.255.248.0
- 8 networks of length /24 with 254 hosts each, netmask 255.255.255.0
- 16 networks of length /25 with 126 hosts each, netmask 255.255.255.128
- 32 networks of length /26 with 62 hosts each, netmask 255.255.255.192

One route 192.144.0.0/21

Address Allocation

Name	Site	Region covered
ARIN	arin.net	North America, part of the Caribbean
APNIC	apnic.net	Asia/Pacific region, including Australia and New Zealand
AfriNIC	afrinic.net	Africa
LACNIC	lacnic.net	Central and South America, part of the Caribbean
RIPE NCC	ripe.net	Europe and surrounding areas

Special forms of IPv4 Addressing

Special Address	NetID	HostID	Example	Use
Network itself	Non-0	All zeros (0s)	192.168.14.0	Used by routers: on a host, means "some host," but it is not used.
Directed broadcast	Non-0	All ones (1s)	192.168.14.255	Destination only: used by routers to send to all host on this network.
Limited broadcast	All 1s	All 1s	225.255.255.255	Destination only: direct broadcast when NetID is not known.
This host on this network	All 0s	All 0s	0.0.0.0	Source only: used when host does not know its IPv4 address.
Specific host on this network	All 0s	Non-0	0.0.0.46	Destination only: defined, but not used
Loopback	127	Any	127.0.0.0	Destination only: packet is not sent out onto network.

Network Address Translation

- Made to deal with IPv4 exhaustion
- Private address spaces
- Border router translates between private and public

IP class	From	To	CIDR range
Class A	10.0.0.0	10.255.255.255	10.0.0.0/8
Class B	172.16.0.0	172.31.255.255	172.16.0.0/12
Class C	192.168.0.0	192.168.255.255	192.168.0.0/16

Routing

- Direct a packet to its destination
  - To reach network A
    - Send packets through machine C
  - Default route
    - Often the gateway assigned by DHCP

ip route show

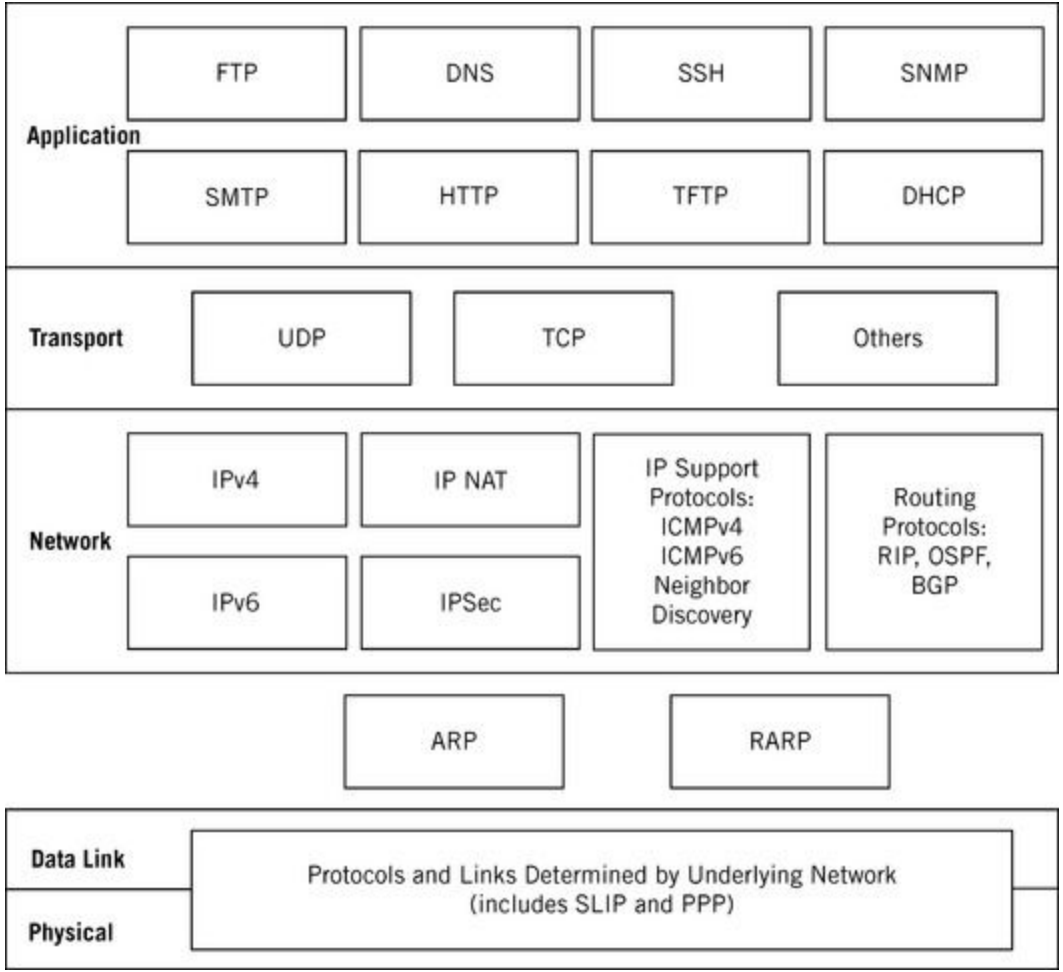
```
redhat$ netstat -rn
Destination      Genmask          Gateway          Fl  MSS   Iface
132.236.227.0    255.255.255.0    132.236.227.93   U   1500  eth0
default          0.0.0.0          132.236.227.1    UG  1500  eth0
132.236.212.0    255.255.255.192  132.236.212.1    U   1500  eth1
132.236.220.64   255.255.255.192  132.236.212.6    UG  1500  eth1
127.0.0.1        255.255.255.255  127.0.0.1        U   3584  lo
```

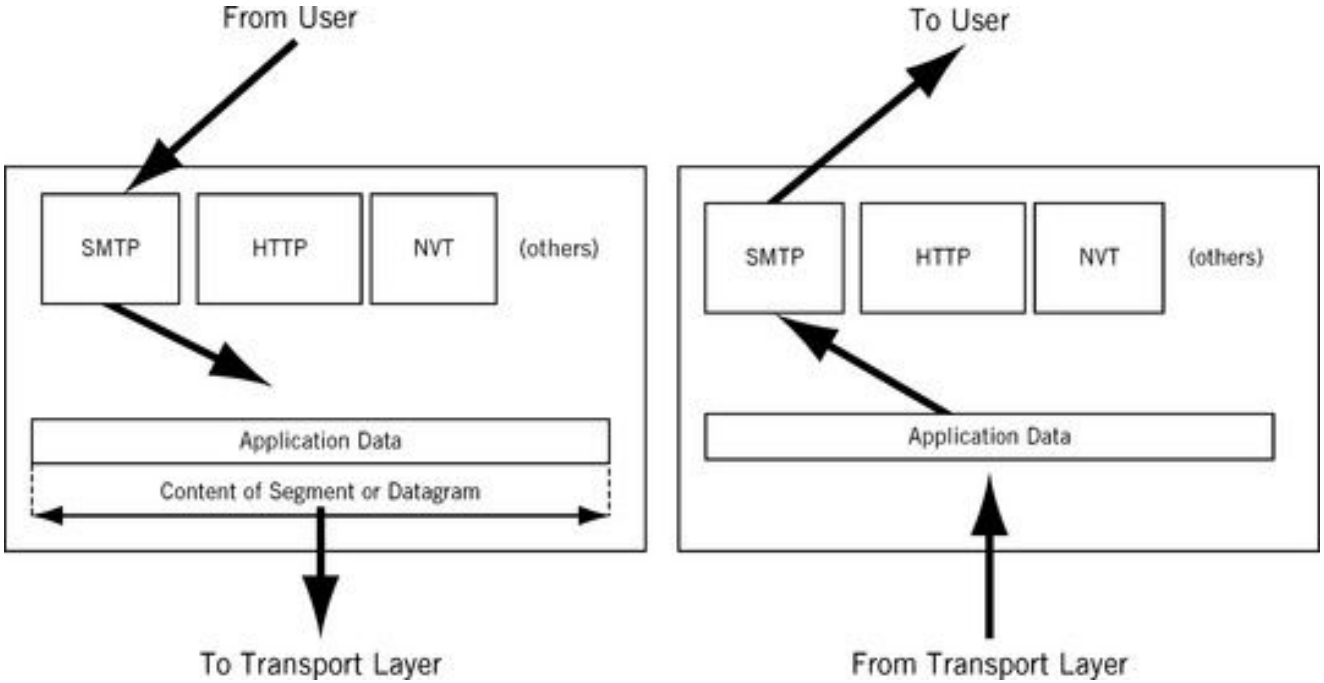
```
ip route add 132.236.220.64/26 via 132.236.212.6 dev eth1
ip route add default via 132.236.227.1 dev eth0
```

Dynamic Host Control Protocol

- Enables automatic
  - IP Address
  - Netmask
  - Gateway
  - DNS Server configuration
- Offers a lease
  - Expires after a configurable amount of time
  - Must be renewed
- Software
  - Isc-dhcp-server
  - dnsmasq - simple

```
# global options

option domain-name "synack.net";
option domain-name-servers gw.synack.net;
option subnet-mask 255.255.255.0;
default-lease-time 600;
max-lease-time 7200;

subnet 192.168.1.0 netmask 255.255.255.0 {
    range 192.168.1.51 192.168.1.60;
    option broadcast-address 192.168.1.255;
    option routers gw.synack.net;
}

subnet 209.180.251.0 netmask 255.255.255.0 {
}

host gandalf {
    hardware ethernet 08:00:07:12:34:56;
    fixed-address gandalf.synack.net;
}
```


Application
FTP
DNS
SSH
SNMP
SMTP
HTTP
TFTP
DHCP
Transport
UDP
TCP
Others
Network
IPv4
IP NAT
IP Support Protocols: ICMPv4 ICMPv6 Neighbor Discovery
Routing Protocols: RIP, OSPF, BGP
IPv6
IPSec
ARP
RARP
Data Link
Physical
Protocols and Links Determined by Underlying Network (includes SLIP and PPP)


From User
SMTP
HTTP
NVT
(others)
Application Data
Content of Segment or Datagram
To Transport Layer
To User
SMTP
HTTP
NVT
(others)
Application Data
From Transport Layer

Basic Network Configuration

ip link set eth0 up

ip add 192.168.47.20/24 dev eth0

ip route add default via 192.168.47.1

Setup DNS Modify

/etc/resolv.conf

/etc/systemd/resolved.conf

Exercise

ifconfig # Old

ip addr # New

ping -r -c 5000 alpine.cse.unr.edu # Useful for checking for packet loss

apt install -y nmap telnet

nmap ponderosa.cse.unr.edu # Check for services

telnet mail.cse.unr.edu 25

telnet google.com 80

Exercise

#Working with application level protocols
#Daytime Protocol
#https://tools.ietf.org/html/rfc867 Port 13 UDP/TCP
#Time Protocol
#https://tools.ietf.org/html/rfc868 Port 37 UDP/TCP
#NIST maintains a set of servers. https://tf.nist.gov/tf-cgi/servers.cgi

apt install netcat #arbitrary TCP and UDP connections and listens

nc time-d-wwv.nist.gov <<< “”

ifconfig/arp/route Command	ip Command
ifconfig	ip addr show #more information ip link show #less information Note: The **show** is the default on most versions of the **ip** command, so **ip link** should do the same thing as **ip link show**.
ifconfig eth0 192.168.1.16 netmask 255.255.255.0 broadcast 192.168.2.255	ip addr add 192.168.1.16 /24 broadcast 192.168.2.255 dev eth0
ifconfig eth0 promisc	ip link set eth0 promisc on
ifconfig eth0 -promisc	ip link set eth0 promisc off
ifconfig eth0 -arp	ip link set eth0 arp off
ifconfig eth0 arp	ip link set eth0 arp on
arp	ip neigh show #neigh = neighbor
arp -i eth0 -d 192.169.1.11	ip neigh del 192.168.1.11 dev eth0
route	ip route show
route add -net 192.168.2.0 netmask 255.255.255.0 gw 192.168.1.100	ip route add 192.168.2.0/24 via 192.168.1.100
route del -net 192.168.2.0 netmask 255.255.255.0 gw 192.168.1.100	ip route del 192.168.2.0/24 via 192.168.1.100
route del default	ip route del default
route add default gw 192.168.1.1	ip route add default via 192.168.1.1