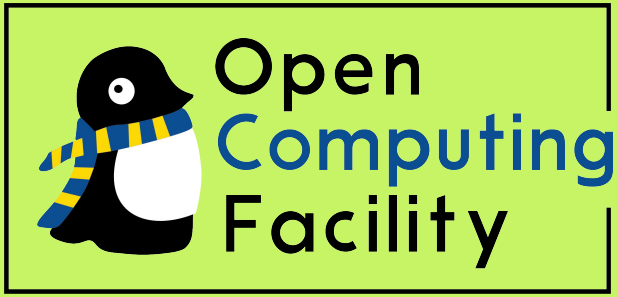


Lecture 5

Networking 101

ncograin

(slide credits abizer, longlian, night)

Course Resources

- Your facilitators!
- Ed, Gradescope
- OCF Slack (ocf.io/slack) or Discord (ocf.io/discord) #decal-general
- All materials available at decal.ocf.io
- Ask questions / work on lab with us during lab sessions! (Tuesday 8-9pm in OCF Lab)

Outline

- What makes up a network system?
- Addressing & Routing
- Protocols
- SysAdmin Tools

What makes up a network system?

Big Question: Why do we need networks?

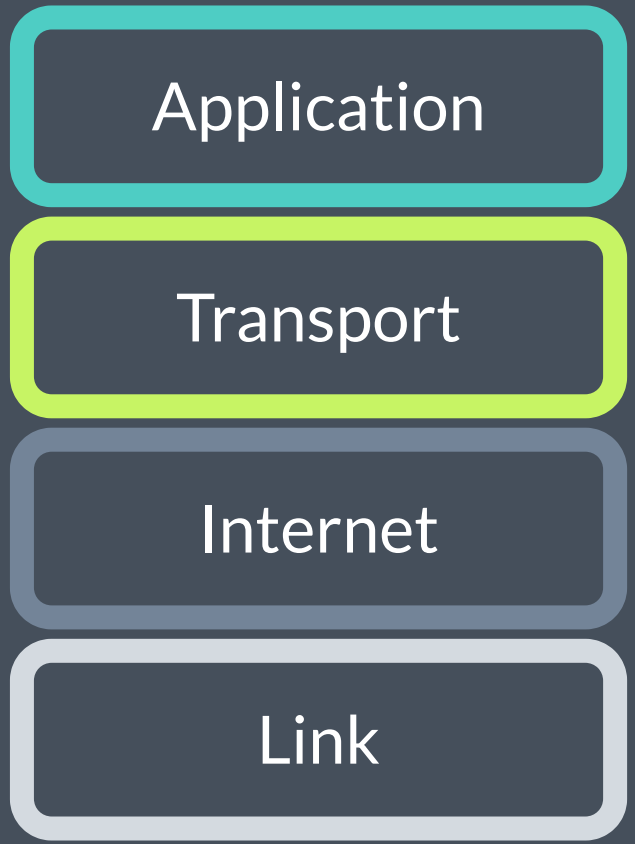
- We want easy communication!

What do we need for easy communication?

- Low latency
- High bandwidth
- Ability to easily identify a computer
- Fault-tolerance

Conceptual Model: TCP/IP

- 4-layer model
- Loosely defined (functions can be in-between levels)
- Higher layers abstract lower layers



Link Layer

- Communication between physically connected nodes
- MAC Addresses
- ARP/NDP
- Underlying hardware implementations are abstracted outside of TCP/IP model
- “Who can I talk to without going through a router?”

Internet Layer

- Transmitting data to different networks
- Global Routing (IP Addresses)
- No guarantees on reliability (just like real life!)

Transport Layer

- Defines protocols for communication between computers
- Dealing with unreliability of sending packets
- TCP/UDP

Application Layer

- Application-specific, actually deals with interpreting the data being sent
- Most “user-facing” functions live here
- File sharing, message passing, database access
- Examples: HTTP/FTP/DNS

Addressing & Routing

Media Access Control (MAC) Addresses

- Looks like this: `00:14:22:01:2a:c5`
- 48 bits, divided into 6 octets
- First 3 octets identify the interface manufacturer
- Unique to each device
- Used to identify devices on local network
- How do we talk to devices outside the local network?

IP Addresses (IPv4)

- IP addresses uniquely identify a host
- IP addresses can be changed
- 32 bits long, divided into 4 octets, look like: 123.123.123.123
- IPv4 octets are commonly expressed in decimal notation, although alternative formats, though rarely employed, are also permissible.

A note on bits, decimal and hexadecimal

- Different ways to write a number!
  - Decimal: uses 0 through 9
    - Add another digit after reaching 9
  - Hex: 0 through 9, a through f
    - Add another digit after reaching f
  - Binary: uses bits (0 or 1)
    - Add another digit... every two numbers... :(
- . or : in addresses are just separators. Make it easy to read

Example: Sixty

Decimal:

60

Used in IPv4

Hex:

3c

(often written 0x3c)

Used in MAC addresses, IPv6

Binary:

0b111100

It’s more convenient to use hex instead!

Subnets

- Remember that IP addresses are represented with binary
- Sometimes, you'll see 192.168.1.0/24
- The /24 is a subnet mask
- Subnet: a range of IP addresses (e.g. 192.168.1.0-255)
- The first 24 bits of of the address: network prefix
- The last 8 bits: host identifier

IPv6

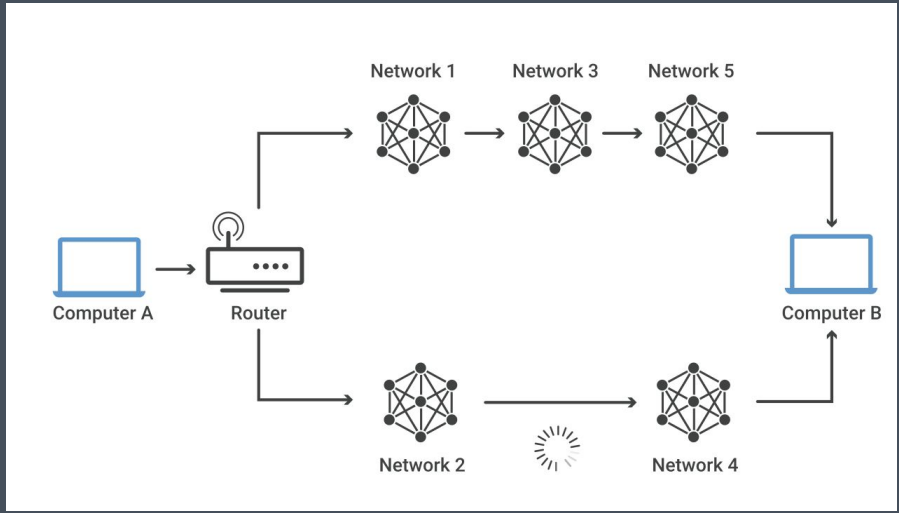
- There are only 4.3 billion IPv4 addresses, but almost 8 billion people in the world - what do we do?
- IPv6 addresses contain 128 bits
- Formatted as follows: 1234:5678:89:0:ab:cd:ef:beef
- We have a million billion IPv6 addresses for each cell in every human on the planet (enough?)

Address Resolution Protocol (ARP)

- Translates IP address to a specific MAC address
- ARP Request: asks devices on local network who an IP address belongs to
- Kernel caches values in an “ARP table”

Routing

- How do we decide where to jump to?
- Routers maintain routing tables, which tell us where to hop to next



169.229.226.0/24	ocf network
169.229.0.0/16	berkeley network
128.32.0.0/16	berkeley network
0.0.0.0/0	via 128.32.0.39

OCF router routing table

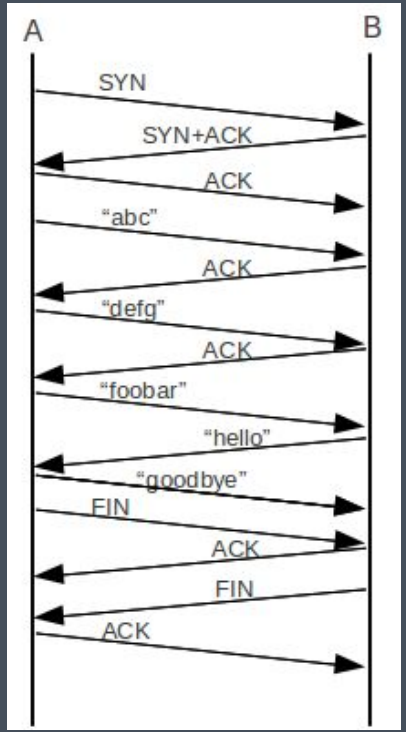
Domain Name System (DNS)

- Translates domains (as found in URLs) to IP addresses
- Can manually set domain -> IP mappings, or can use external DNS resolver

Protocols

Transmission Control Protocol (TCP)

- Ensures reliable transmission
- Connection-oriented: need to initiate a connection before sending data
- Used when reliability is more important than speed
- Examples: website loading, file transfer



User Datagram Protocol (UDP)

- No reliability guarantees
- Does not establish a connection
- Good for real-time applications where some loss is acceptable
- Examples: (lossy) streaming, gaming

Internet Control Message Protocol (ICMP)

- Not used to transmit data
- Technically not a “transport protocol”
- Used to transmit error messages and status info
- Used by diagnostic tools

Internet Control Message Protocol (ICMP)

Type	Code	Status	Description
0 – Echo Reply[6]:14	0		Echo reply (used to ping)
3 – Destination Unreachable[6]:4[9]	0		Destination network unreachable
	1		Destination host unreachable
	3		Destination port unreachable
8 – Echo Request	0		Echo request (used to ping)

SysAdmin Tools

ping

ping

```
~ ❯ ping -c 5 google.com
PING google.com(sfo07s26-in-x0e.1e100.net (2607:f8b0:4005:802::200e)) 56 data bytes
64 bytes from sfo07s26-in-x0e.1e100.net (2607:f8b0:4005:802::200e): icmp_seq=1 ttl=114 time=3.15 ms
64 bytes from sfo07s26-in-x0e.1e100.net (2607:f8b0:4005:802::200e): icmp_seq=2 ttl=114 time=3.13 ms
64 bytes from sfo07s26-in-x0e.1e100.net (2607:f8b0:4005:802::200e): icmp_seq=3 ttl=114 time=3.18 ms
64 bytes from sfo07s26-in-x0e.1e100.net (2607:f8b0:4005:802::200e): icmp_seq=4 ttl=114 time=3.11 ms
64 bytes from sfo07s26-in-x0e.1e100.net (2607:f8b0:4005:802::200e): icmp_seq=5 ttl=114 time=3.16 ms

--- google.com ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4007ms
rtt min/avg/max/mdev = 3.111/3.144/3.176/0.022 ms
```

traceroute

traceroute

```
~ ❯ traceroute google.com                                                    lmathias@tsunami
traceroute to google.com (142.250.189.206), 30 hops max, 60 byte packets
 1  vlan635.inr-350-reccev.berkeley.edu (169.229.226.1)  0.552 ms  0.511 ms  0.457 ms
 2  reccev-cev-cr1--xe-2-0-9.net.berkeley.edu (128.32.0.168)  1.445 ms  1.277 ms  1.278 ms
 3  sut-mdc-sr4--irb-117.net.berkeley.edu (128.32.82.177)  0.949 ms  0.823 ms  0.760 ms
 4  firewall--ethernet1-21-595.ocf.berkeley.edu (128.32.82.178)  0.824 ms  0.856 ms  0.813 ms
 5  sut-mdc-sr5--irb-116.net.berkeley.edu (128.32.82.173)  1.184 ms  1.149 ms  1.117 ms
 6  sut-mdc-cr1--et-0-1-1.net.berkeley.edu (128.32.255.40)  1.751 ms reccev-cev-cr1--et-0-1-1.net.berkeley.edu (128.32.255.42
)  1.796 ms sut-mdc-cr1--et-0-1-1.net.berkeley.edu (128.32.255.40)  1.743 ms
 7  reccev-cev-br1--et-1-1-0.net.berkeley.edu (128.32.0.36)  1.710 ms  1.533 ms  1.466 ms
 8  oak-agg8--ucb--100g.cenic.net (137.164.3.26)  3.319 ms  3.288 ms  2.413 ms
 9  74.125.48.172 (74.125.48.172)  2.566 ms 142.250.164.8 (142.250.164.8)  2.732 ms  2.691 ms
10  108.170.243.1 (108.170.243.1)  4.148 ms 108.170.242.225 (108.170.242.225)  2.792 ms 108.170.243.1 (108.170.243.1)  4.071
ms
11  142.251.224.175 (142.251.224.175)  2.837 ms 142.251.224.173 (142.251.224.173)  2.683 ms 142.251.224.175 (142.251.224.175)
  3.104 ms
12  sfo03s25-in-f14.1e100.net (142.250.189.206)  2.889 ms  2.857 ms  2.848 ms
```

arp

```
~ ❯ arp
Address                  HWtype  HWaddress           Flags Mask            Iface
dhcp-169-229-226-182.OC  ether   7e:e7:6d:6e:4d:cd   C                     ens3
supernova.OCF.Berkeley.  ether   52:54:00:c9:60:55   C                     ens3
anthrax.OCF.Berkeley.ED  ether   52:54:00:c3:f9:d8   C                     ens3
dataloss.OCF.Berkeley.E  ether   ac:1f:6b:1a:8e:92   C                     ens3
dementors.OCF.Berkeley.  ether   52:54:00:a6:16:f6   C                     ens3
dhcp-169-229-226-178.OC  ether   6e:74:84:51:0f:d8   C                     ens3
bigbang.OCF.Berkeley.ED  ether   d0:17:c2:d2:08:62   C                     ens3
blackhole.OCF.Berkeley.  ether   00:1c:73:3d:2f:c2   C                     ens3
dhcp-169-229-226-143.OC  ether   ba:fc:93:84:3c:61   C                     ens3
vlan635.inr-350-reccev.  ether   88:e6:4b:0d:cc:6e   C                     ens3
dhcp-169-229-226-142.OC  ether   bc:d0:74:12:a3:e2   C                     ens3
dhcp-169-229-226-171.OC  ether   04:ea:56:de:e3:56   C                     ens3
dhcp-169-229-226-145.OC  ether   3c:22:fb:0f:d1:22   C                     ens3
flood.OCF.Berkeley.EDU   ether   52:54:00:34:09:e6   C                     ens3
```

dig

dig

```
~ ❯ dig google.com

; <<>> DiG 9.16.37-Debian <<>> google.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 20879
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 1232
; COOKIE: 7b5c9f7901f27f540100000063feda0345fb6daf74a8b74b (good)
;; QUESTION SECTION:
;google.com.                    IN      A

;; ANSWER SECTION:
google.com.             202     IN      A       142.251.46.174

;; Query time: 0 msec
;; SERVER: 2607:f140:8801::1:22#53(2607:f140:8801::1:22)
;; WHEN: Tue Feb 28 20:52:19 PST 2023
;; MSG SIZE  rcvd: 83
```

ip

Can do many things, including displaying/modifying ip addresses and routing info. Example: ip addr

```
@ ~ ❯ ip addr
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: ens3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq state UP group default qlen 1000
    link/ether 52:54:00:c9:60:55 brd ff:ff:ff:ff:ff:ff
    inet 169.229.226.36/24 brd 169.229.226.255 scope global ens3
       valid_lft forever preferred_lft forever
    inet6 2607:f140:8801::1:36/64 scope global
       valid_lft forever preferred_lft forever
    inet6 fe80::5054:ff:fec9:6055/64 scope link
       valid_lft forever preferred_lft forever
3: docker0: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc noqueue state DOWN group default
    link/ether 02:42:48:6d:20:27 brd ff:ff:ff:ff:ff:ff
    inet 172.17.0.1/16 brd 172.17.255.255 scope global docker0
       valid_lft forever preferred_lft forever
```