

SNOWCAST DUE
THURS 2/12

TODAY

- LINK LAYER WRAPUP
- NETWORK LAYER
- INTRO TO FORWARDING

THIS WEEK

- HW1 OUT TODAY
- IP PARTNER FORM
OUT TODAY, DUE THURS
- IP GEARUP THURS
5-7PM CIT 165

Administivia

- Snowcast: now due Thursday (2/12)
 - Look for an announcement with updates to tester, tips for debugging
- IP project: out Thursday
 - Team form: out TODAY, due THURSDAY 2/12
 - Gearup: THURSDAY 2/12 5-7pm CIT 165
- HW1: out later today, due next Thurs
 - Some practice for IP!

Recap: the link layer

(HOME, OFFICE)

Goal: How to connect hosts on a "small" network

- Hosts connect to network via interfaces
- Every interface has a link-layer address
 - Ethernet/Wifi: MAC address (0c:45:22:c1:be:03)

(More info next page)

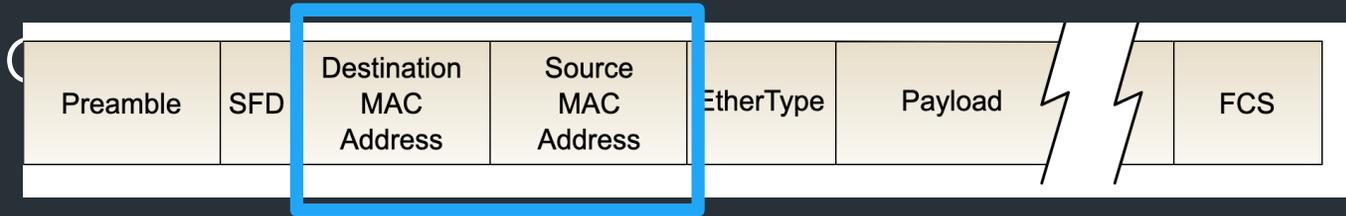
Conceptually, all hosts are connected to all other hosts

If you know the mac address of the host you want to reach, link layer can get the packets

Switch/Wifi AP: move packets between nodes, based on the mac address



Ethernet Addresses (mac addresses)



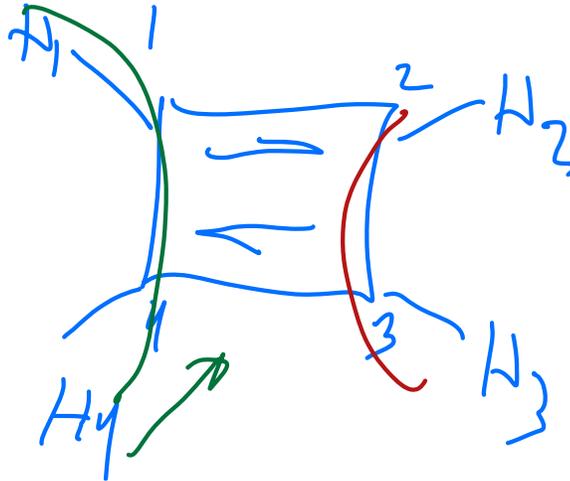
Globally unique, 48-bit address per interface

00:1c:43:00:3d:09 (Samsung adapter)

First 24 bits: [Registered to manufacturers](#)

=> Other protocols have adopted this address format (eg. Wifi, Bluetooth, ...)

=> Nowadays, we call them "mac addresses" or "hardware addresses"



Switch has table (destination MAC address => port)

Has queues and buffering for when multiple packets going to same port

Switches are very fast, very parallel

=> Have some state (the table), and they can do a bit of computation

In general, packets are sent to one host at a time (unicast)

Also special address (ff:ff:ff:ff:ff:ff) is the **broadcast address** => used to send a packet to all host

Recap: the link layer

Goal: How to connect hosts on a "small" network

- Hosts connect to network via interfaces
- Every interface has a link-layer address
 - Ethernet/Wifi: MAC address (**0c:45:22:c1:be:03**)

Mental model for the link layer

- Every host connected to every other host (at least logically)
- Given a link-layer address, know how to reach host **on your network**

=> Devices: Switches, Wifi APs: forward packets between nodes on same network

Example: Ethernet switch MAC table

```
R6#sh mac-address-table
```

```
EHWIC: 0
```

```
Destination Address Address Type VLAN Destination Port
```

```
-----  
5c45.27e0.8383          Dynamic          1 GigabitEthernet0/1/3  
7641.7b63.584a          Dynamic          20 GigabitEthernet0/1/3  
5c45.27e0.8381          Dynamic          10 GigabitEthernet0/1/3  
0000.5e00.0101          Dynamic          10 GigabitEthernet0/0/1  
ca3f.aee3.e3e6          Dynamic          20 GigabitEthernet0/1/3  
644b.f012.7f75          Dynamic          20 GigabitEthernet0/1/3  
f018.9815.8eb8          Dynamic          20 GigabitEthernet0/1/3  
ecb5.fa13.4677          Dynamic          20 GigabitEthernet0/0/2  
a0a4.c5c2.4165          Dynamic          20 GigabitEthernet0/0/1  
4c71.0c92.4f10          Dynamic          10 GigabitEthernet0/1/3  
12d3.acae.bbc0          Dynamic          20 GigabitEthernet0/0/1  
04d4.c448.9cf7          Dynamic          20 GigabitEthernet0/1/3
```

PORTS

MAC ADDR

ONE PER HOST

Does it scale? 🤔

Why doesn't it scale?

Physically don't have enough space for ports, hardware gets more complicated

Lots of different device types--not everything is ethernet

Limitations on distance for cables

Enter: IP

Application

Service: user-facing application.
Application-defined messages

Transport

Service: multiplexing applications
Reliable byte stream to other node (TCP),
Unreliable datagram (UDP)

Network

Service: move packets to any other node in the network
Internet Protocol (IP)

Link

Service: move frames to other node across link.
May add reliability, medium access control

Physical

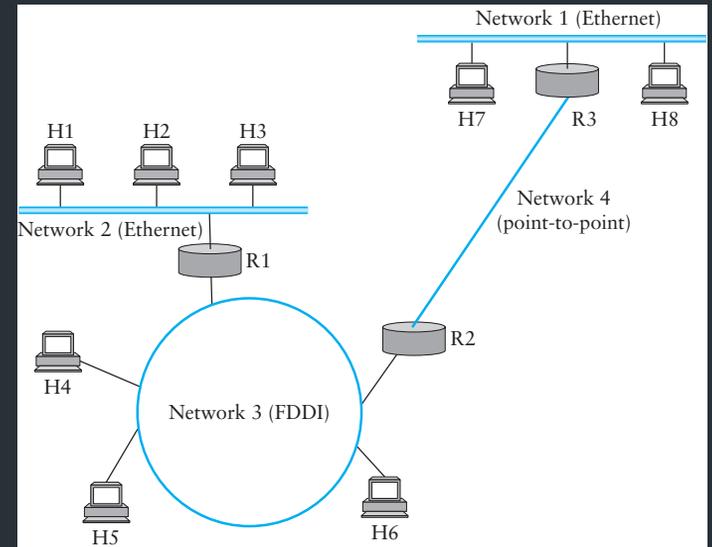
Service: move bits to other node across link

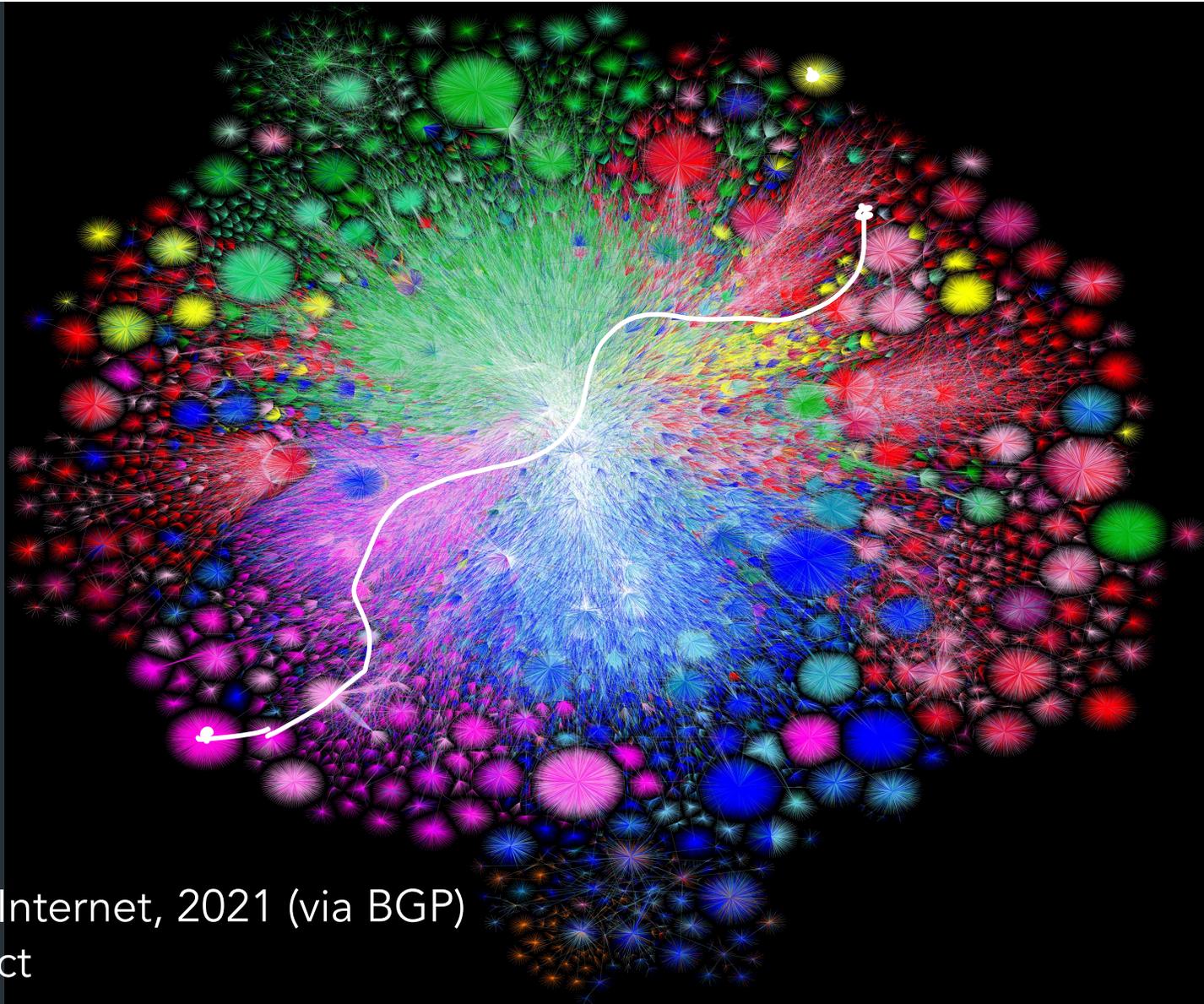
Internet Protocol (IP) Goals

How to connect *everyone*?

- Glue lower-level networks together
- A network of networks!
- Router: device that forwards packets between *networks*

=> Doesn't that sound like switching?





Color Chart

North America (ARIN)

Europe (RIPE)

Asia Pacific (APNIC)

Latin America (LANIC)

Africa (AFRINIC)

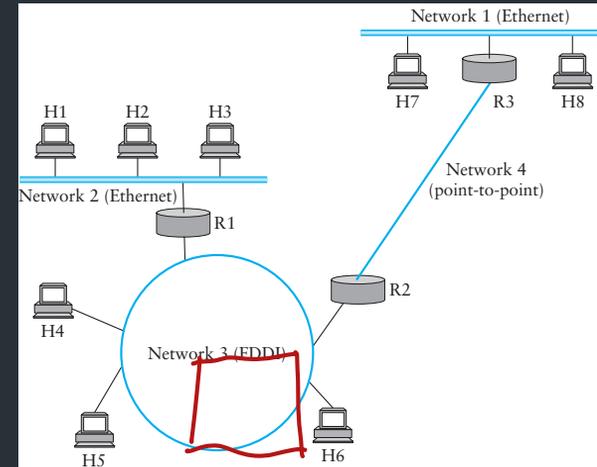
Backbone

US Military

Map of the Internet, 2021 (via BGP)
OPTE project

New Challenges

- Networks are heterogeneous (eg. Wifi vs. Ethernet)
 - Different frame formats
 - Some are more reliable than others
 - Different packet sizes/bandwidths
- Scaling: link-layer strategies don't work!



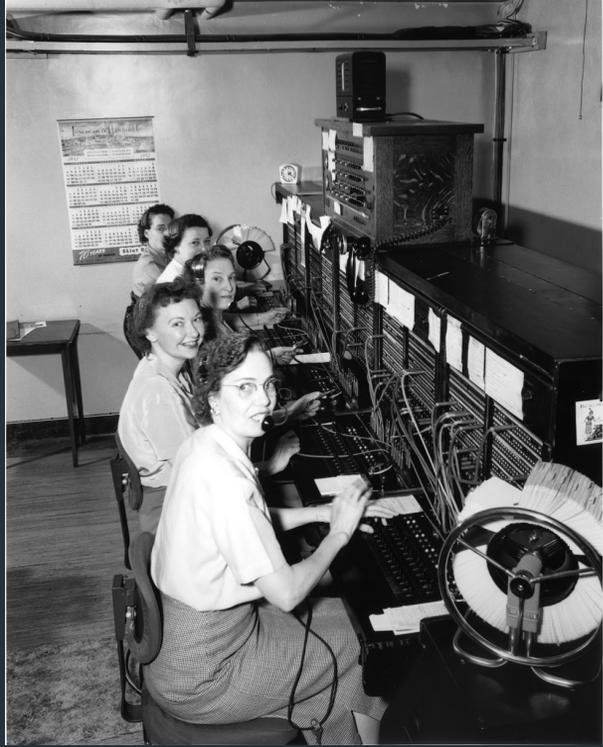
What came before the Internet?

The (landline) telephone network

(Plain Old Telephone Service (POTS))



A Host

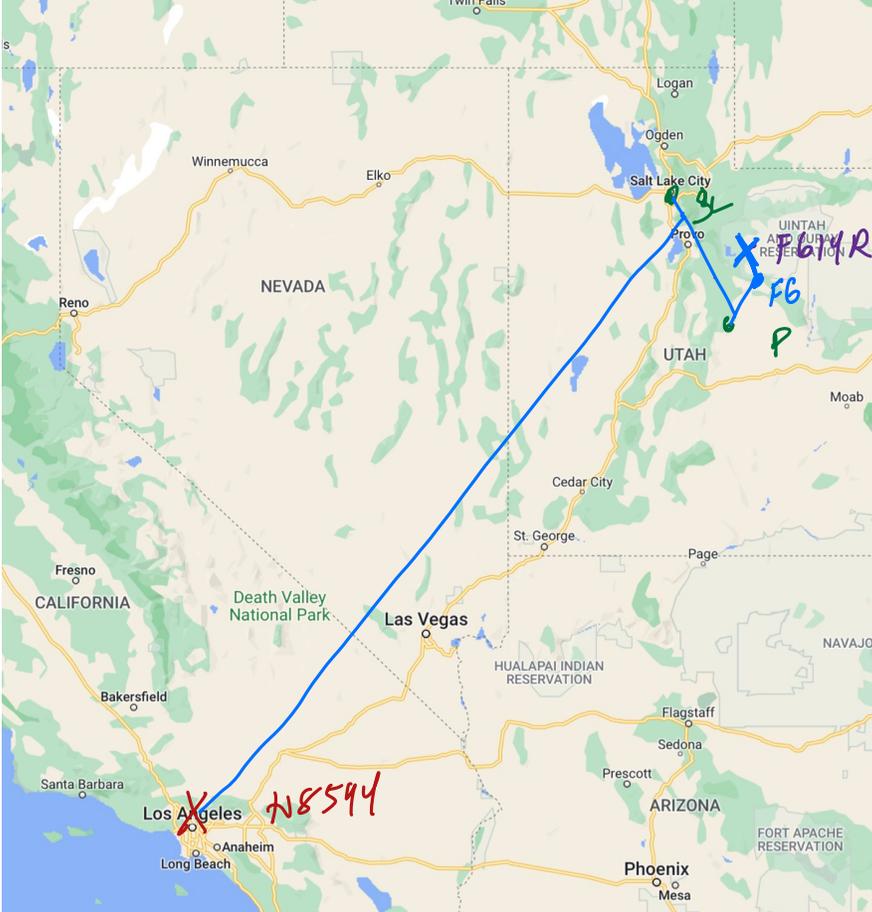




A large telephone exchange, 1943

Example: long distance telephone call

*Dramatization from an episode of
radio program "Dragnet", 1949*



TRANSLATES
TO IP?

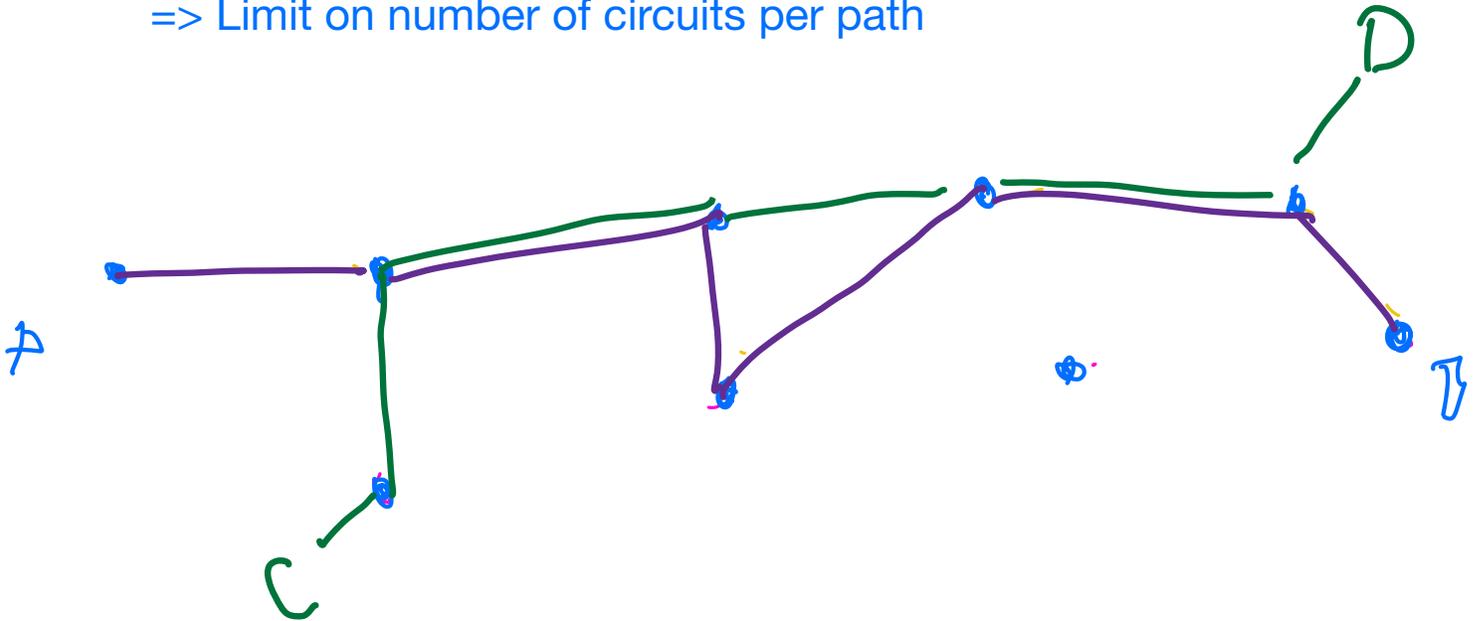
- Initiating operator on call whole time
- Routing to "big" destination first before getting more local
- "Rate operator" knows the whole path at the start
- All operators can listen in

X
✓
X
MAYBE...

Telephone way: circuit switching: operator sets up path for call, keep same path until done

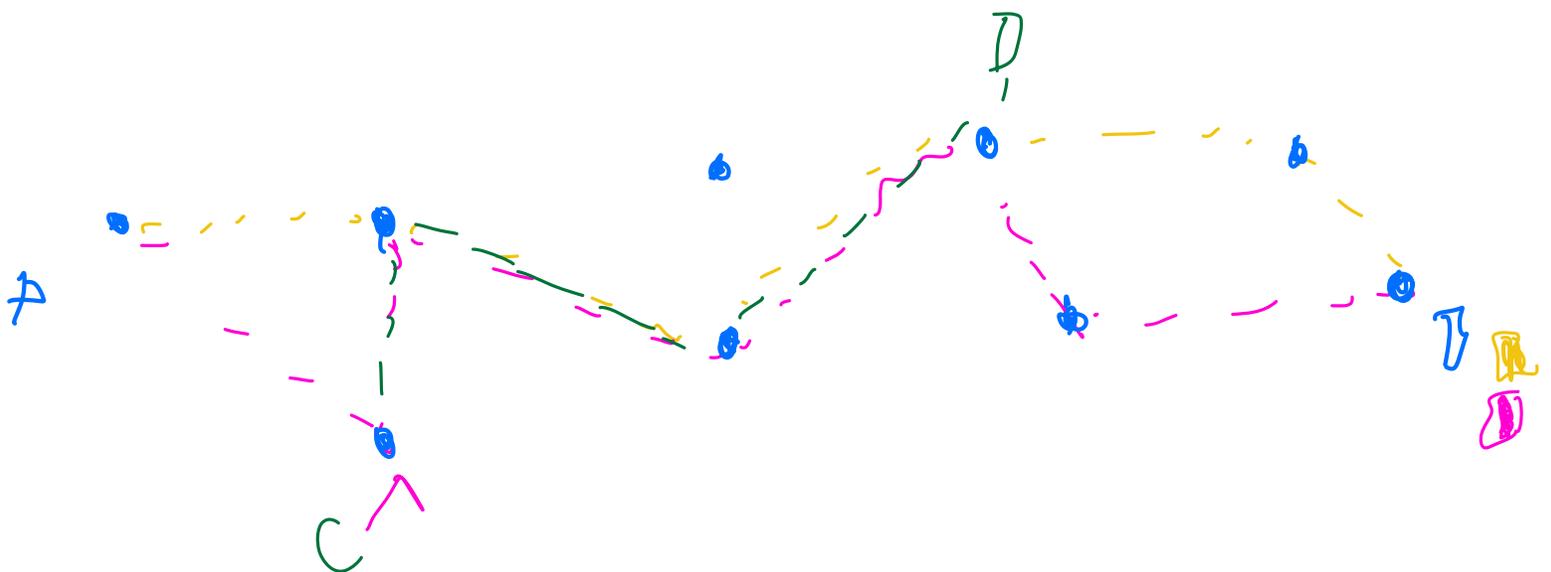
=> circuit needs to be set up ahead of time

=> Limit on number of circuits per path



Internet way: packet switching: divide up data into small chunks ("packets" or "datagrams", that are forwarded around the network separately

- Routers just need to pick destination for next packet
- No initial setup
- Messages must get divided up into small units (and reassembled at other end)
- Packets may take any path (can be different for each packet) 🟡 🟣
- Fairness/access is more "random", which scales better than circuits



Early telephone systems

- Circuit switching: set up whole path for call beforehand

Does it scale?

Early Internet goals

- ⇒ Build system that can connect different *networks*
- ⇒ Operate over long distances
- ⇒ Managed by different entities

Need: *devices* and *protocols* to make this work

Design questions

- How to deal with heterogeneous networks?
- How to find hosts?
- Should messages be reliable or unreliable?
- What to do when a device joins/leaves?
- ...

A Bit of History

Early Packet switched networks: Arpanet's IMPs

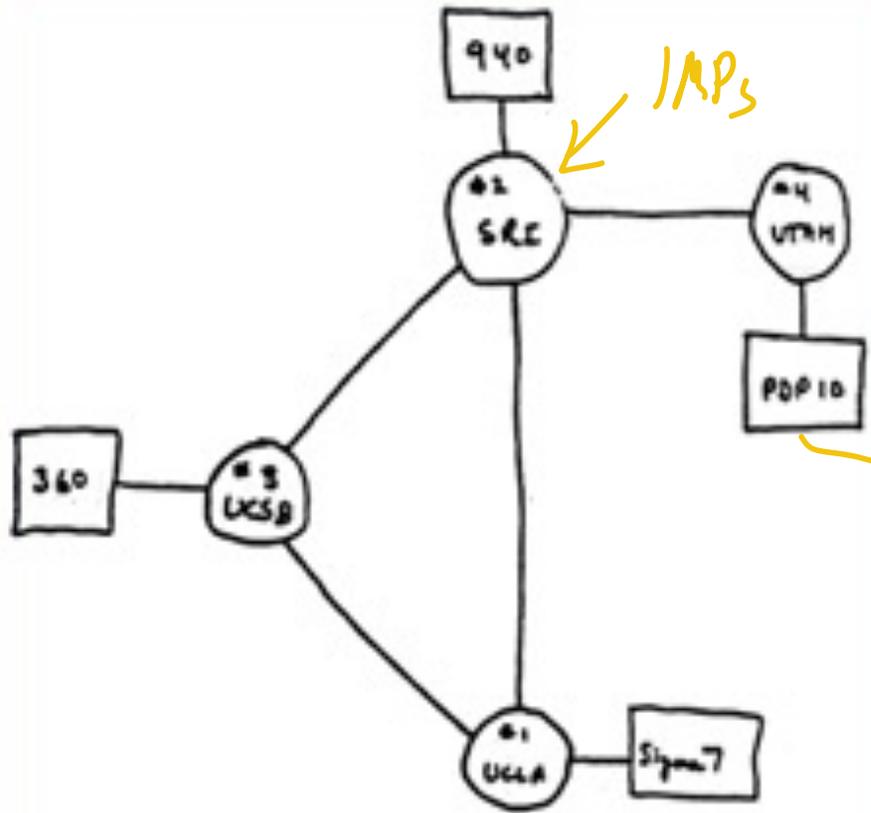
- Late 1960's => RFC 1, 1969!
- Reliable network with many features we know today

Initial version: Network Control Program (NCP)

- Assumed IMPs were reliable

What about when network isn't reliable?





IAPs

SINGLE
COMPUTERS

THE ARPA NETWORK
DEC 1969

How to make such a protocol?

- How to deal with heterogeneous networks?
- How to find hosts?
- Should messages be reliable or unreliable?
- What to do when a device joins/leaves?
- ...

"LEAST COMMON
DENOMINATOR
ON FUNCTION
ALITY."



Big concerns

⇒ Not every application needs all features

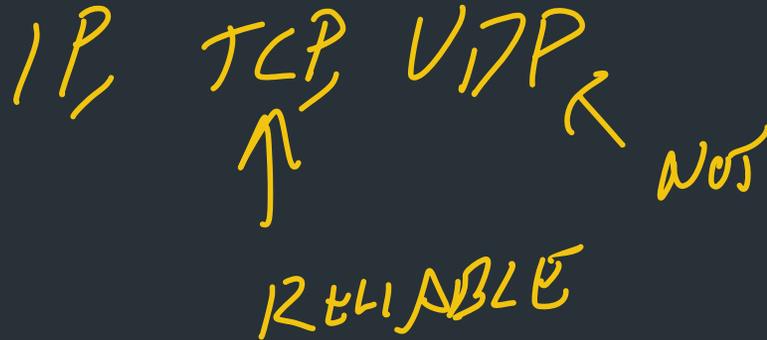
⇒ Can't assume much functionality from (heterogeneous link layer)

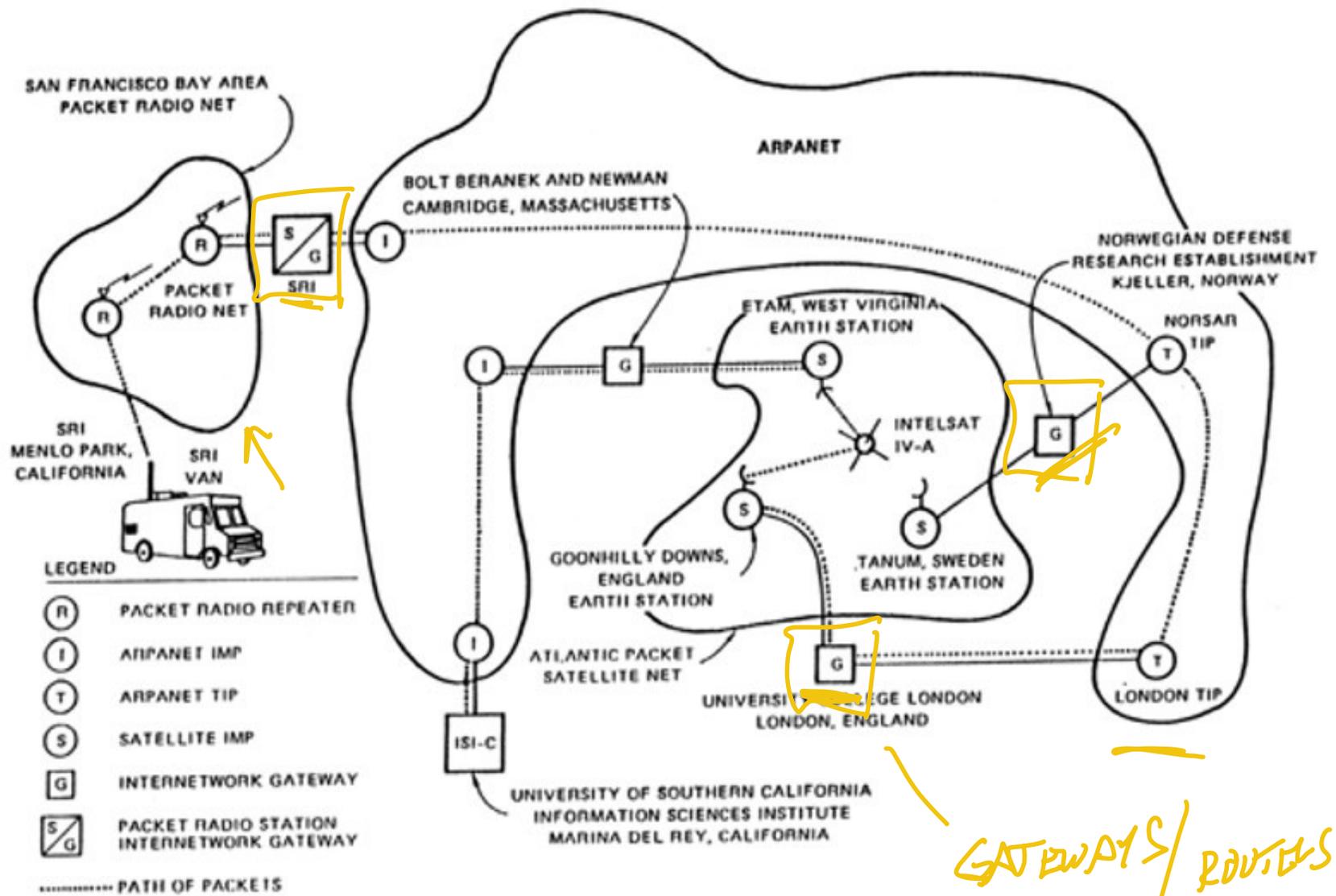
1974: TCP/IP Introduced

- Vint Cerf, Robert Kahn build protocol to replace NCP
- Initial design: single protocol providing a reliable pipe

Eventually, separated into different protocols we know today

IP provides a way to send
"datagrams" between hosts





IP's Decisions

- Connectionless, packet-switched network
 - => Routers are "simple" => no connection state
- "Best-effort" service: other layers add reliability if you need it
 - => Packets might be dropped, reordered, delayed, ...

How to reach hosts?

- Common message format: IP header
- Every host identified by an IP address

An excellent read

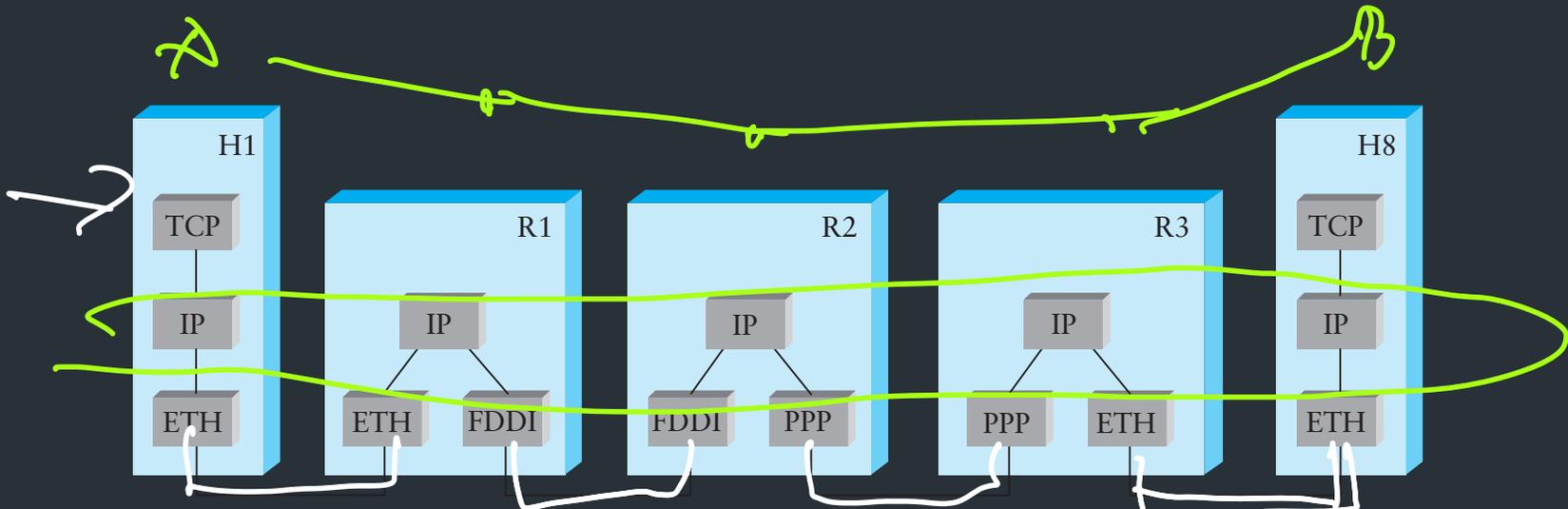
David D. Clark, "The design Philosophy of the DARPA Internet Protocols", 1988

Primary goal: multiplexed utilization of existing interconnected networks

- Other goals:
 - Communication continues despite loss of networks or gateways
 - Support a variety of communication services
 - Accommodate a variety of networks
 - Permit distributed management of its resources
 - Be cost effective
 - Low effort for host attachment
 - Resources must be accountable

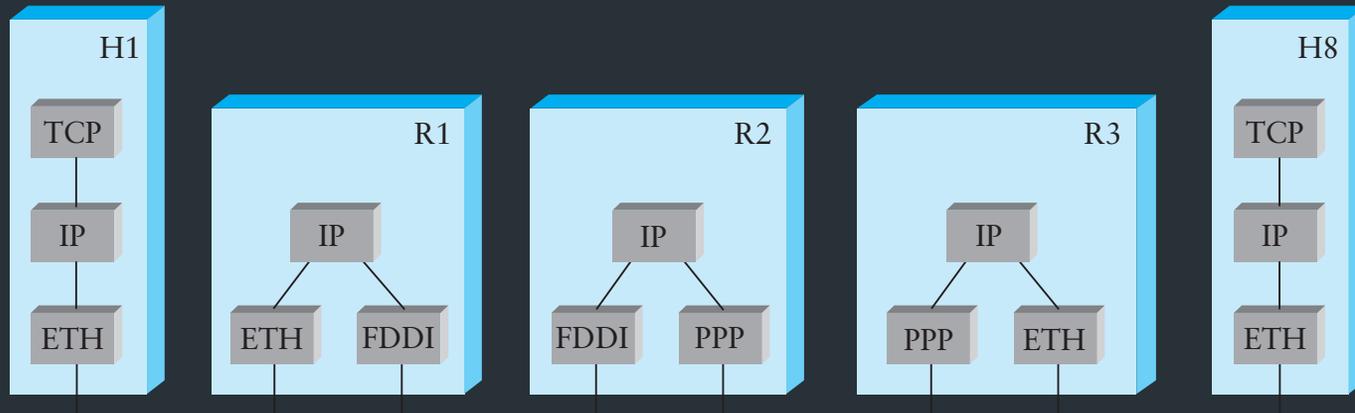
The Internet Protocol (IP): Runs on all hosts and routers

- Addressing: how we name the nodes
- Forwarding: how routers move packets across the network -
- Routing: how the routers figure out "rules" for forwarding



The Internet Protocol (IP): Runs on all hosts and routers

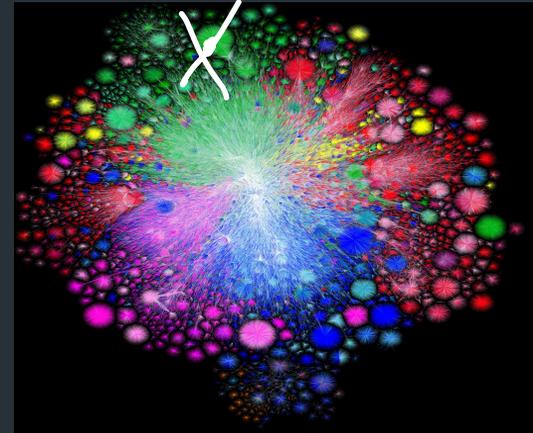
- Addressing: how we name nodes in an IP network
- Provides forwarding: how routers move packets based on the destination address
- (later) Routing: how routers build forwarding rules



IP Addressing

What's an IP address

- Unique number to identify “all” hosts on the Internet
- A number with **structure** => the number tells the network where the host is



Example: phone numbers



FG19R2
≡

+ 1 401 863 1000

COUNTRY
CODE

AREA CODE
(RI)

BROWN
UNIVERSITY

Analogy: back to phones

Telephone numbers have a structure to them

+ 1 401 863 1000
 ↖
 RI

+1 212 555 4253

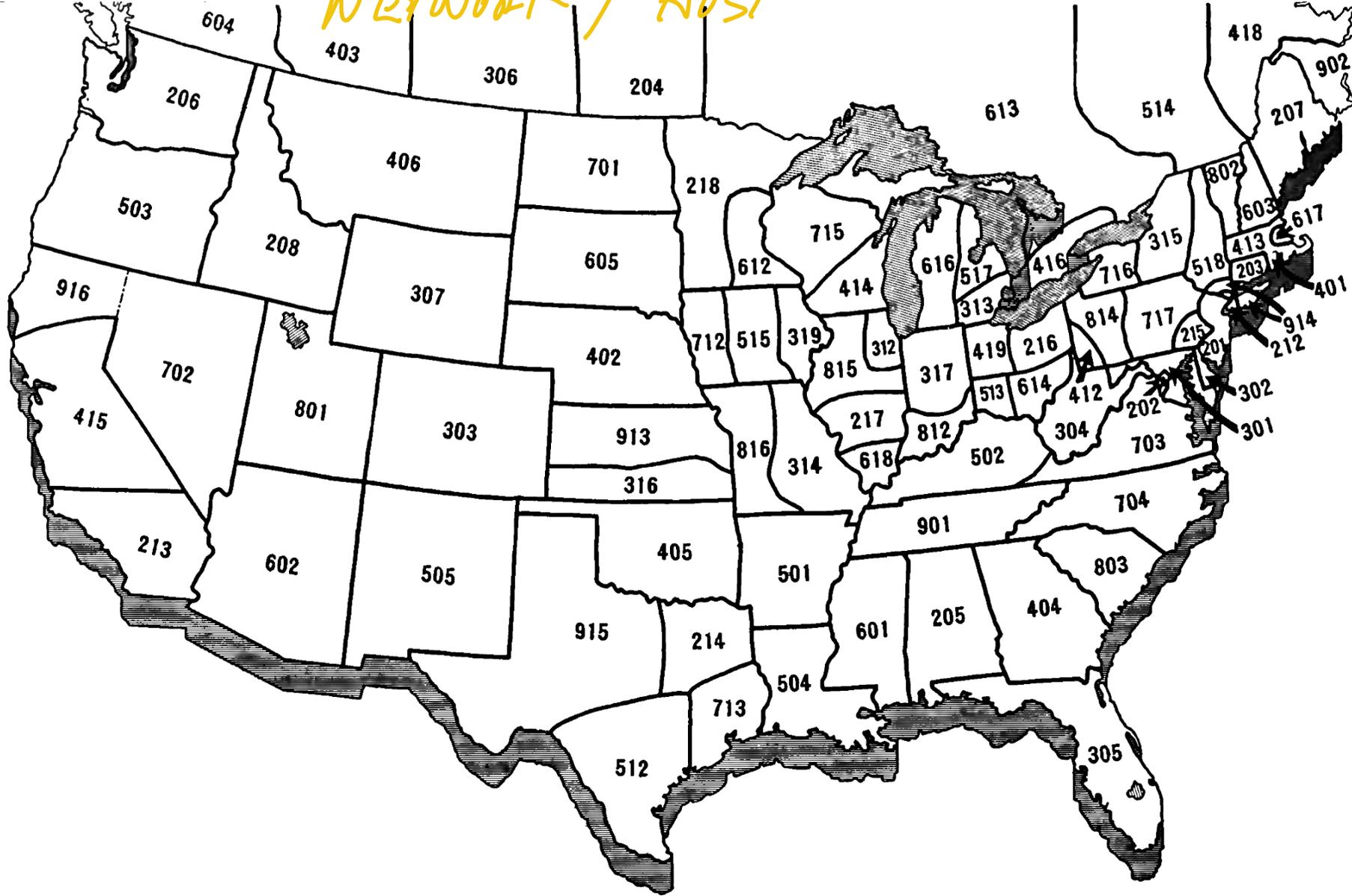
↖
— MANHATTAN



Part of the number tells where you are!
(or at least it did before cell phones)

1115555555 / 11057

NETWORK / NCSI





... and not all numbers are the same length!

IP Addressing



128.148.16.7

IP Version 4: Each address is a 32-bit number:



10000000 10010100 00010000 00000111

Notation

- Write each byte ("octet") as a decimal number
- This is called "dotted decimal" or "dotted quad" notation

2^{32} ≈ 4 BILLION ADDRESSES.

IP Addressing



128.148.16.7

IP Version 4: Each address is a 32-bit number:

128.148.16.7

10000000 10010100 00010000 00000111

32 bits => 2^{32} possible addresses...
problem?

Notation

- Write each byte ("octet") as a decimal number
- This is called "dotted decimal" or "dotted quad" notation

Conceptually: an IP address has two parts

=> Network part: identifies this network to the Internet (FG, "Brown university" => routers use this to get packet to a certain network

=> Host part: identifies the individual host on that network

NETWORK / HOST

128.148.16.7

Size of network vs. host part
changes based on the
network

10000000 10010100 00010000 00000111

Conceptually: an IP address has two parts

=> Network part: identifies this network to the Internet

=> Host part: identifies hosts on that network

128.148.16.7

10000000 10010100 00010000 00000111

Size of host part vs. network part can vary (more on this later)

IP Addressing

Brown owns the range:

128.148.xxx.xxx

16 BITS
 2^{16} POSSIBLE
UNIQUE HOSTS
≈ 65K IPs

10000000 10010100

XXXXXXXXXX XXXXXXXXXXXX

16 BITS

INDIVIDUAL HOSTS

Network part

Host part

Identifies Brown (to the Internet)

Denotes individual hosts
within the Brown Network

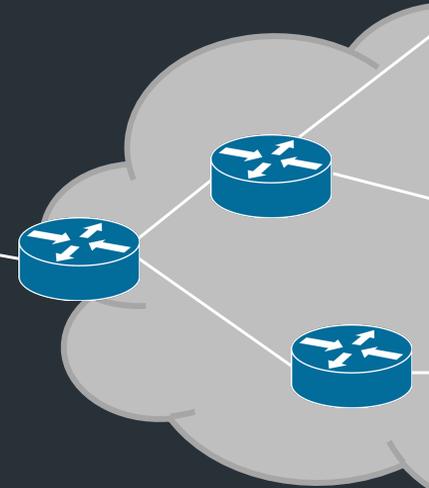
Formal way to write this:

128.148.0.0/16 (16 bits is the network part)

.100.5

Assigning numbers

- Networks are allocated ranges of IPs by global authority (ICANN)
 - Further subdivided by regions, ISPs, ...



- Some IPs have special uses (eg. 127.0.0.1) 128.148.16.7

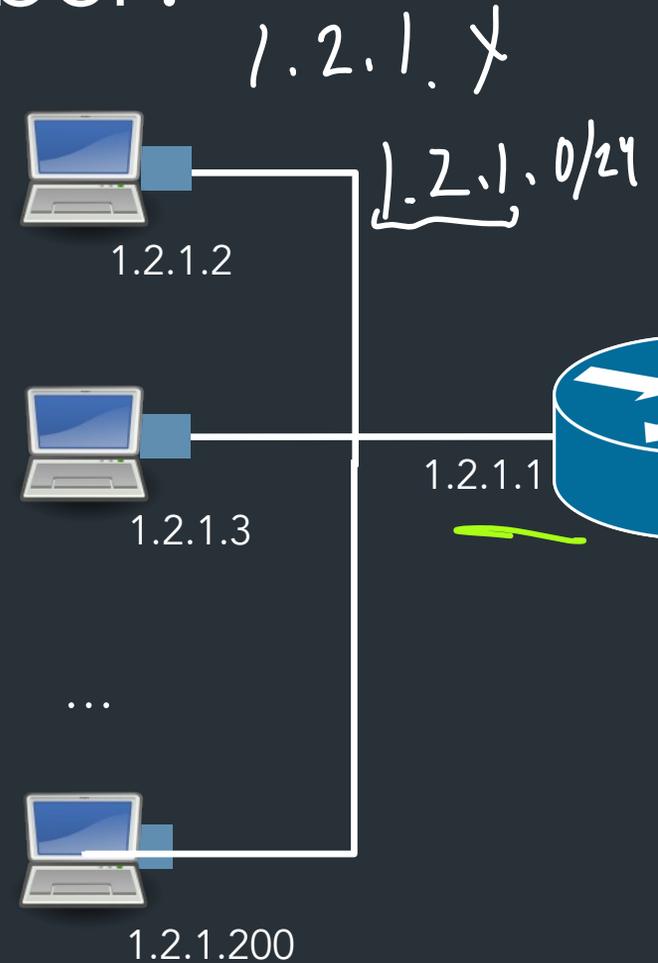
eg. Brown owns 128.148.xxx.xxx, 138.16.xxx.xxx

What do we do with this number?

Link layer: know how to communicate to other devices on your networks

If IP you're trying to send to isn't on your local network, send it to router

Router: device that knows about multiple networks



What do we do with this number?

- Link layer: know how to communicate with devices on “your network”
- Routers: know about multiple networks => Use address to decide how to forward packets between them



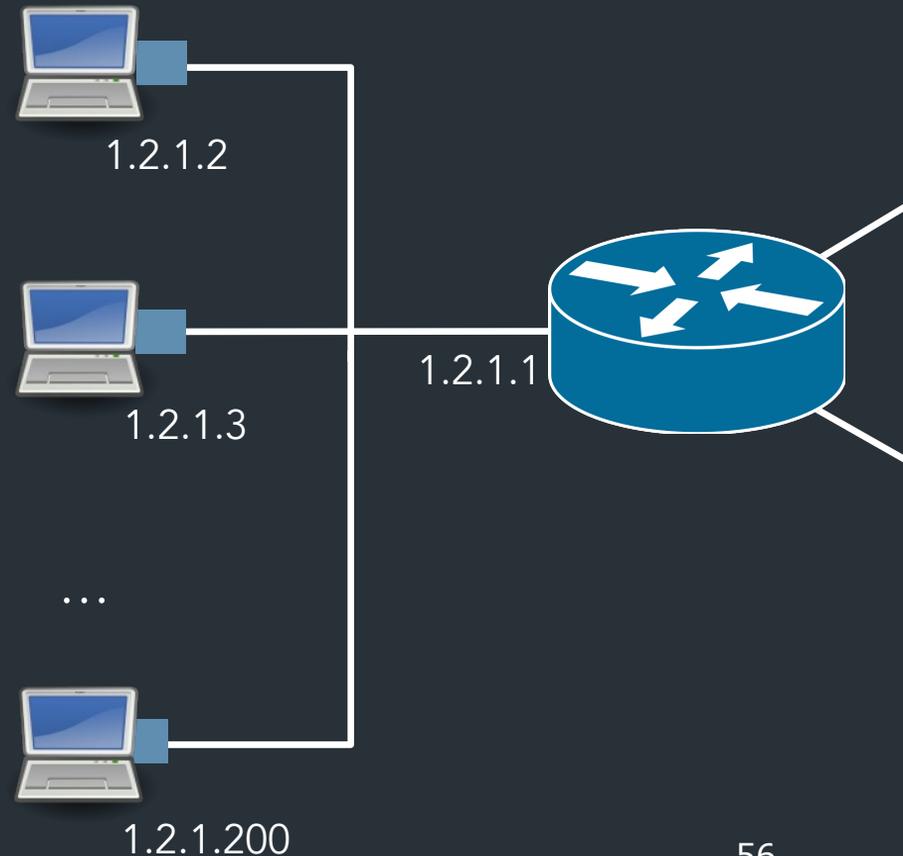
How IP forwarding works

Assume:

- Communicating on same network is easy—this is the link-layer's job!
- Can map IP addresses to MAC addresses (more on this later)

How to reach an address *outside* this network?

Send packets to a router, which forwards IP packets to other networks



A typical configuration

