Discovery Protocols & End-to-End Communication

CPSC 433/533, Spring 2021

Anurag Khandelwal

Administrivia

- Project 2 due on Thursday
- Homework 3 release Today

Last Time

- Frames and framing
- Addressing
- Routing
- Forwarding

Today

- Frames and framing
- Addressing
- Routing
- Forwarding
- Discovery: Bootstrapping end-to-end communication

Today

- **Missing pieces & putting the pieces together**
  - Discovery: Bootstrapping end-to-end communication (ARP)
  - How it all "fits"
- **You may not realize this, but you have learnt a LOT**
- **Must discover lots of information before it can communicate with a remote host B**
  - Connect everything you've learnt so far in the course
  - You will feel great, I promise

Discovery

- A host is "born" knowing only its MAC address

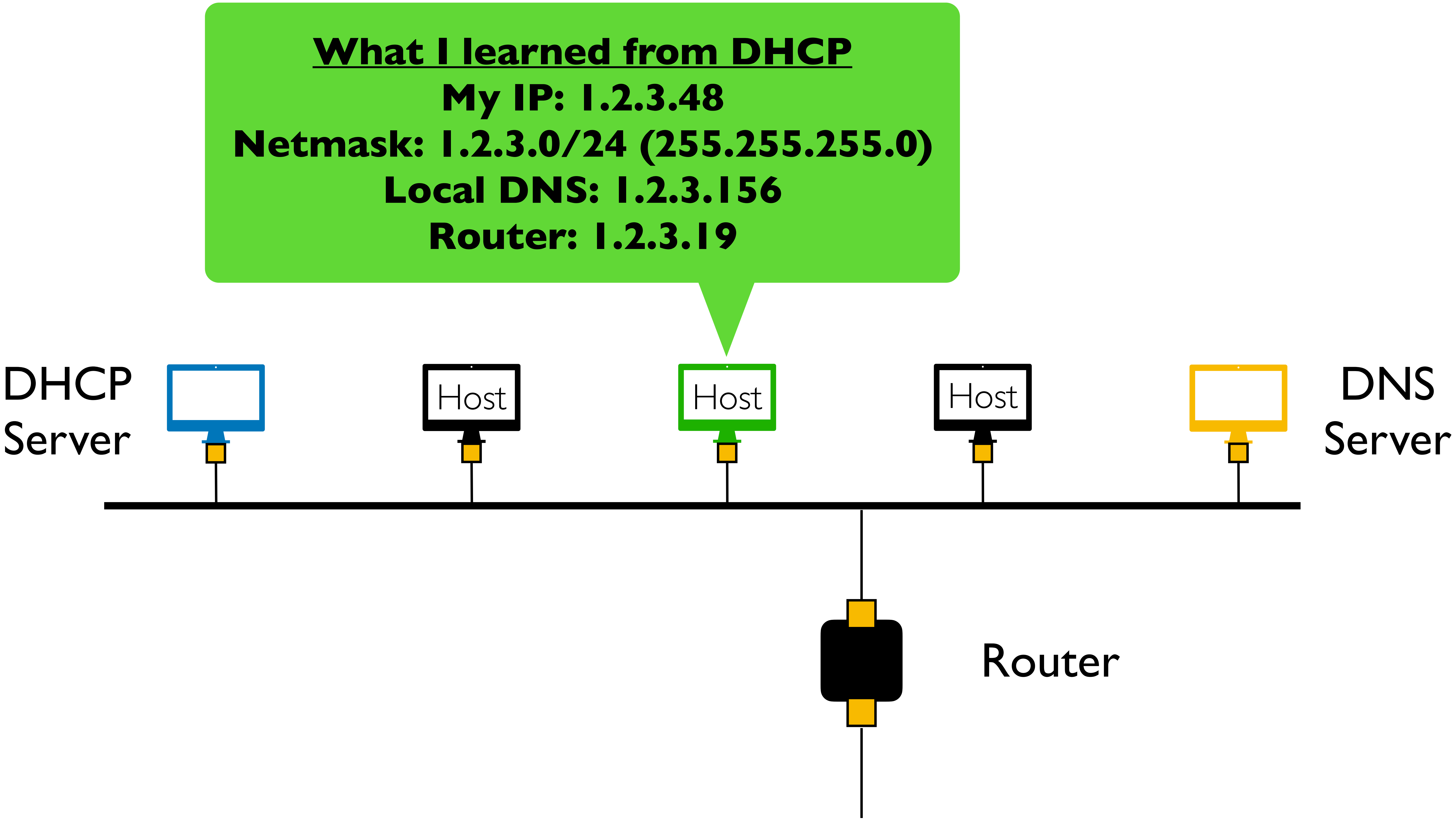
- Must discover lots of information before it can communicate with a remote host B
  - What is my IP address?
  - What is B's IP address?
  - What is B's MAC address? (if B is local)
  - What is my first-hop router's MAC address? (if B is not local)
  - …

DHCP

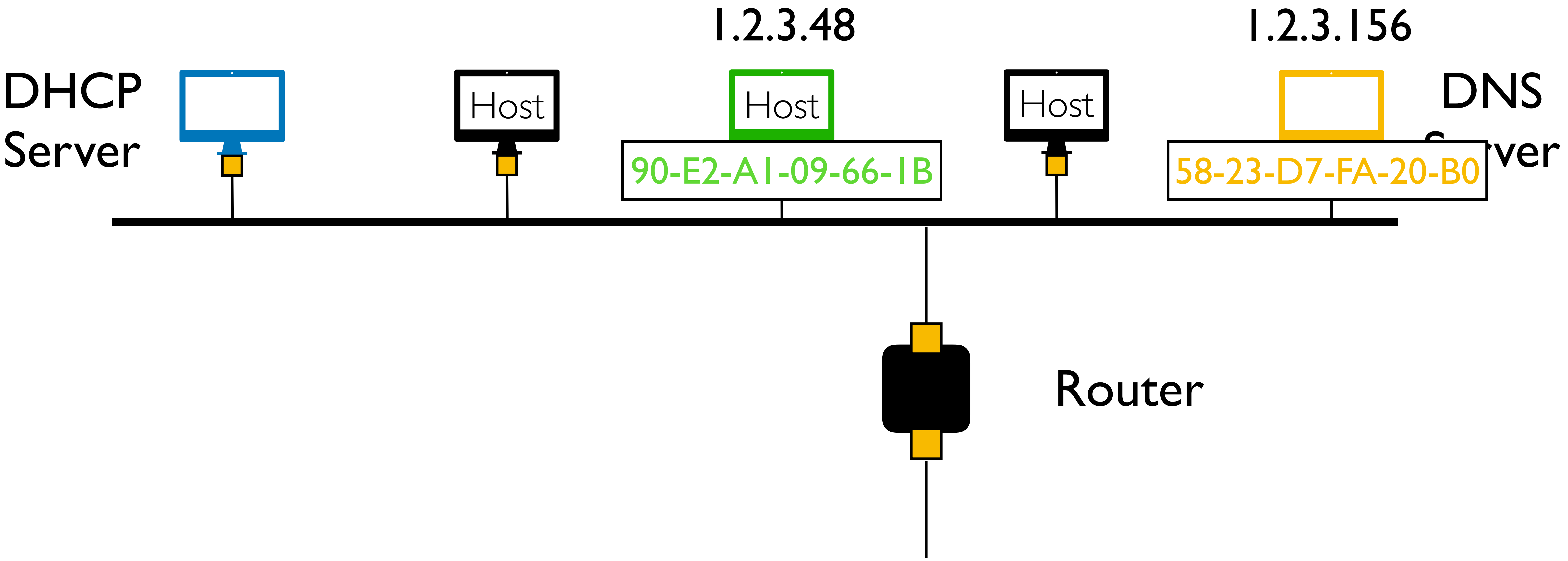
- “Dynamic Host Configuration Protocol”
  - Defined in RFC 2131

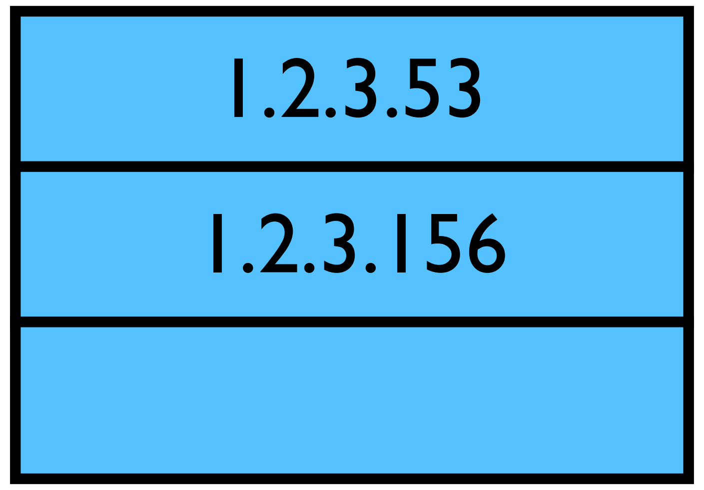
- A host uses DHCP to discover
  - Its own IP address
  - Its netmask
  - IP address(es) for its local DNS name server(s)
  - IP address(es) for its first-hop “default” router(s)

Are we there yet?



Sending packets over Link-Layer





- Encapsulate the IP packet in a link-level (Ethernet) frame
- What's missing?
- **Link layer only understands MAC addresses**
  - Translate the destination IP address to MAC address

ARP: Address Resolution Protocol

- **Every host maintains an ARP table**
  - List of (IP address → MAC address) pairs

- **Consult the table when sending a packet**
  - Map destination IP address to destination MAC address
  - Enscapsulate the (IP) data packet with MAC header; transmit

- **But: what if IP address not in the table**
  - Sender broadcasts: *"Who has IP address 1.2.3.156?"*
  - Receiver responds: *"MAC address 58-23-D7-FA-20-B0"*
  - Sender caches result in its ARP table

What if the destination is remote?

- Look up the MAC address of the first hop router
  - 1.2.3.48 uses ARP to find MAC address for first-hop router 1.2.3.19 rather than the ultimate destination IP address
- How does the green host know the destination is not local?
  - Uses netmask (discovered via DHCP)
- How does the green host know about 1.2.3.19 (router)?
  - Also DHCP

ARP Header


= 1 for Ethernet
= 6 for Ethernet
= 4 for IPv4
= 0x0800 for IPv4
1= request; 2 = reply
Hardware Type (2 bytes)
Protocol Type (2 bytes)
Hardware Addr. Length (1 byte)
Protocol Addr. Length (1 Byte)
Operation (2 Bytes)
Sender Hardware Address (6 bytes)
Sender IP address (4 Butes)
Target Hardware Address (6 Bytes)
Traget IP address (4 Bytes)

Key Ideas in ARP & DHCP

- **Broadcasting: Can use broadcast to make contact**
    - Scalable because of limited size
- **Caching: remember the past for a while**
    - Store the information you learn to reduce overhead
- **Soft state: eventually forget the past**
    - Associate a *time-to-live* field with the information
    - … and either refresh or discard the information
    - Key for *robustness* in the face of unpredictable change

Taking Stock: Naming

Layer	Examples	Structure	Configuration	Resolution Service
Application Layer	cpsc.yale.edu	Organizational Hierarchy	~ Manual	DNS
Network Layer	123.45.6.78	Topological Hierarchy	DHCP	ARP
Link Layer	45-CC-4E-12-F0-97	Vendor (flat)	Hard-coded	

Discovery Mechanisms

We've seen two approaches:

- Broadcast (ARP, DHCP)
  - Flooding doesn't scale
  - No centralized point of failure
  - Zero configuration
- Directory service (DNS)
  - No flooding / scalable
  - Root of the directory is vulnerable (caching is key)
  - Needs configuration to bootstrap (local, root servers, etc.)

Open question: can we get Internet-scale yet zero configuration?

Questions?

Putting Everything Together: Steps in End-to-End Communication

What do hosts need to know?
And how do they find out?

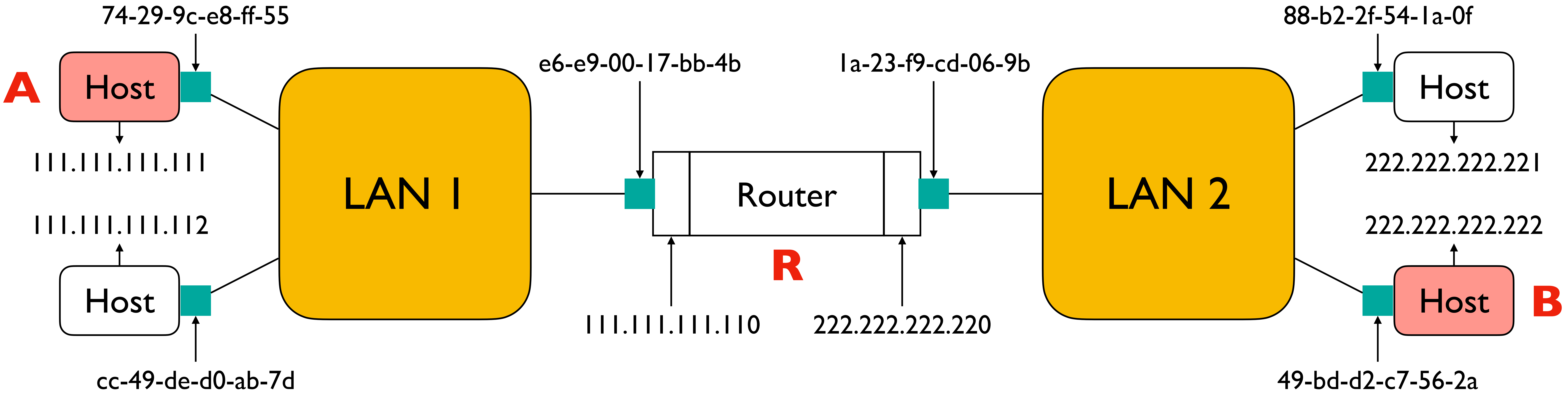
Steps in reaching a destination

- First lookup destination's IP address
  - *DNS*
- Need to know where my local DNS server is
  - *DHCP*
- Also need to know my own IP address
  - *DHCP*

Sending a Packet

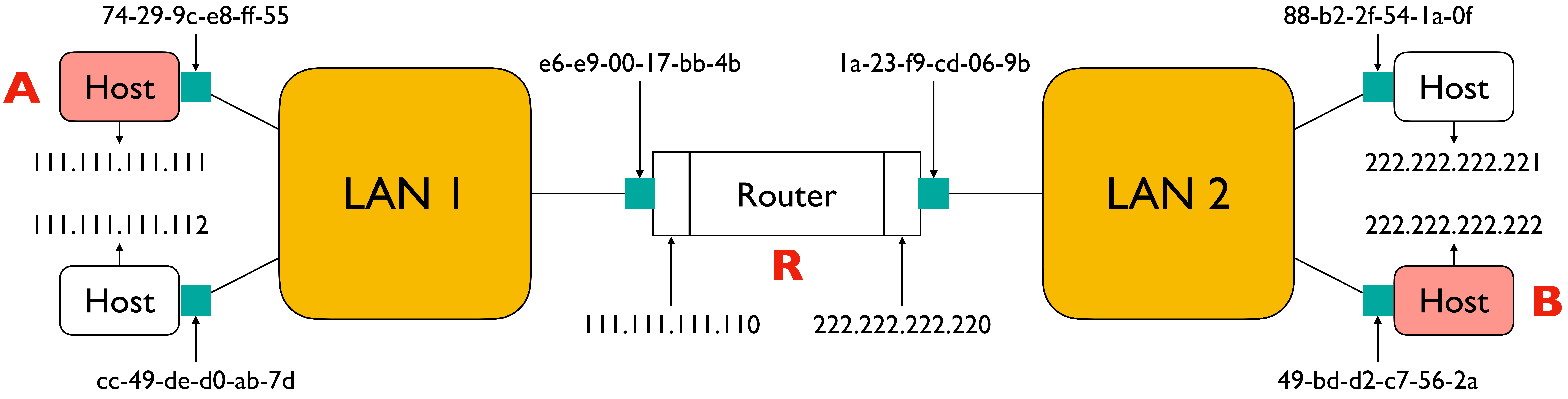
- **On the same subnet:**
  - Use MAC address of destination
  - *ARP*
- **On some other subnet**
  - Use MAC address of first-hop router
  - *DHCP + ARP*
- **And how can a host tell whether destination is on same or other subnet?**
  - Use the netmask
  - *DHCP*

Example: A sending a packet to B



How does host A send an IP packet to host B?

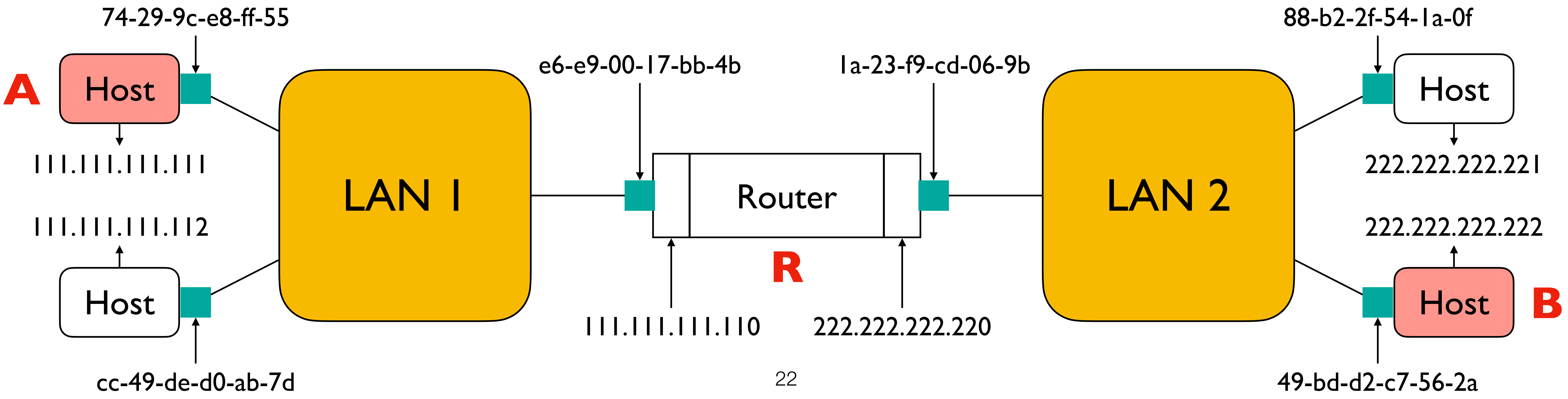
Example: A sending a packet to B



1. **A** sends packet to **R**
2. **R** sends packet to **B**

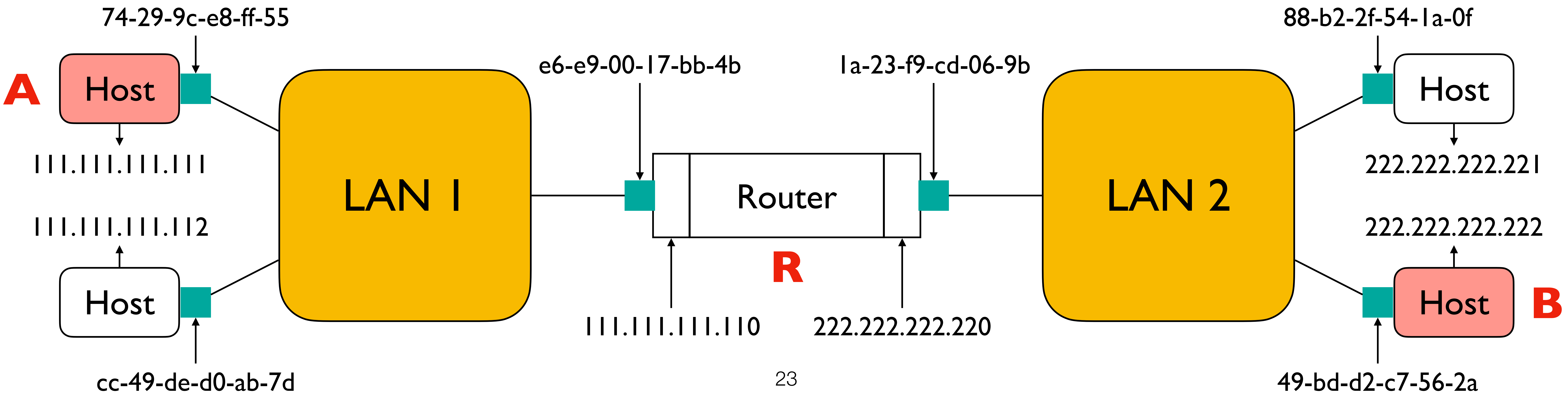
A sends packet through router R

- Host A constructs an IP packet to send to B
  - Source 111.111.111.111, destination 222.222.222.222
- Host A has a gateway router R
  - Used to reach destinations outside of 111.111.111.0/24
  - Address 111.111.111.110 for R learned via <u>*DHCP*</u>



A sends packet through router R

- Host A learns the MAC address of R's interface
  - *ARP* request: broadcast request for 111.111.111.110
  - *ARP* response: R responds with e6-e9-00-17-bb-4b
- Host A encapsulates the IP packet for B, and sends to R

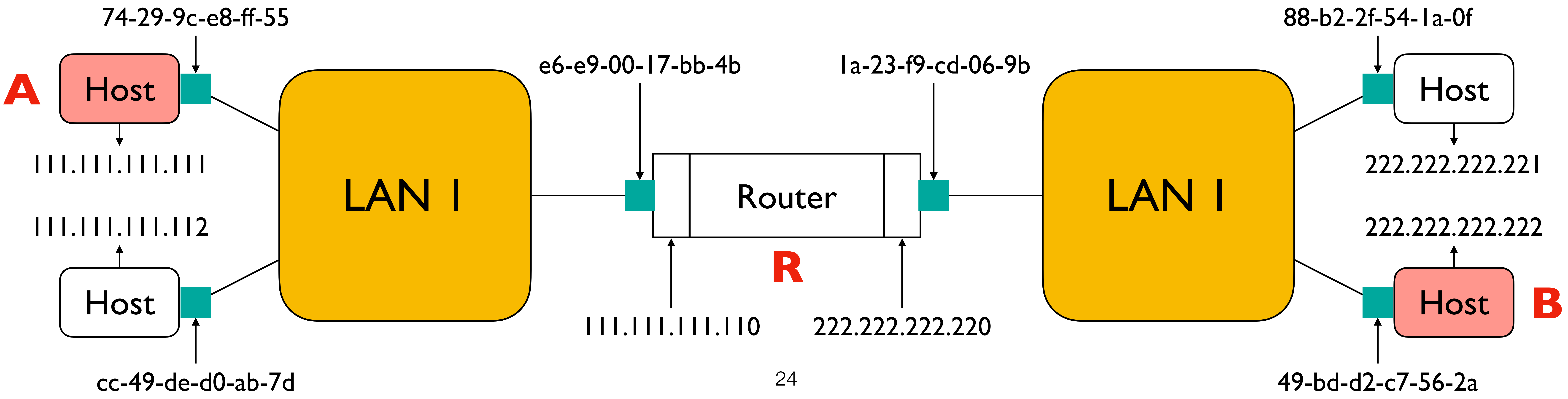


R decides how to forward packet

Two points:

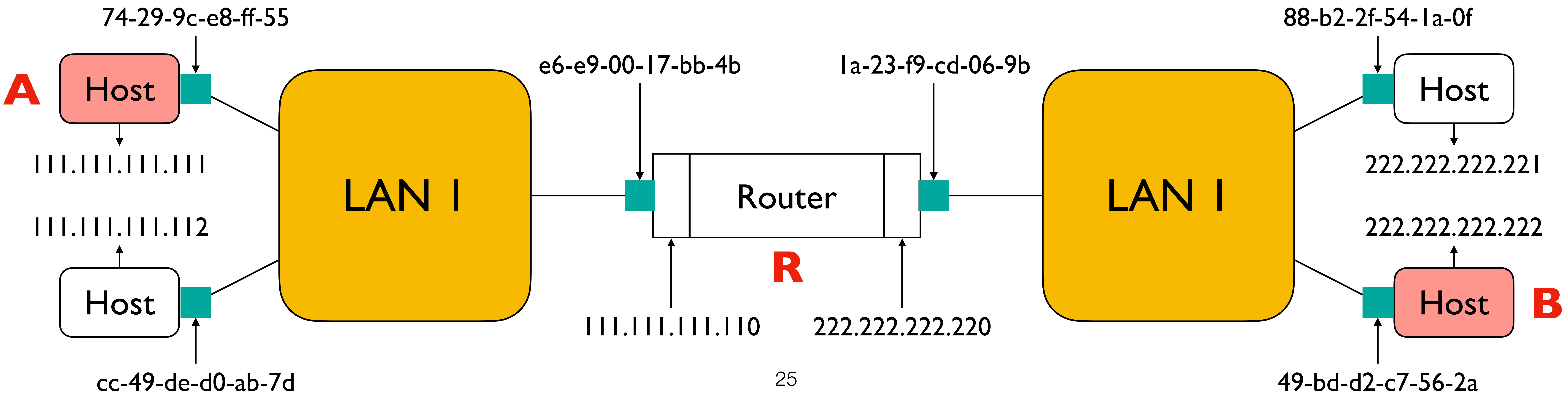
- IP routing table points to this port
- Destination address is within mask of port's address (i.e., local)

- Packet matches 222.222.222.0/24 via other adapter (port)



R sends packet to B

- Router R learns the MAC address of host B
  - *ARP* request: broadcast request for 222.222.222.222
  - *ARP* response: B responds with 49-bd-d2-c7-56-2a
- Router R encapsulates the packet and sends it to B



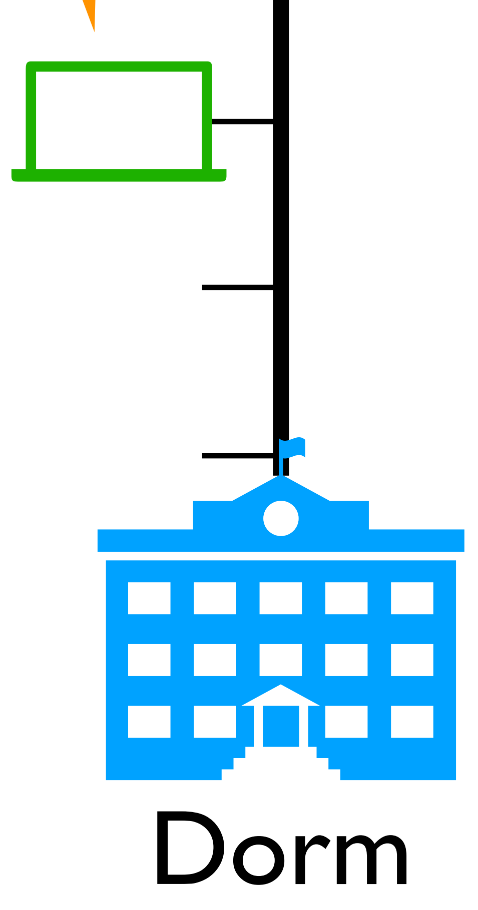
Questions?

Are we there yet?

Yes!

Putting the pieces together

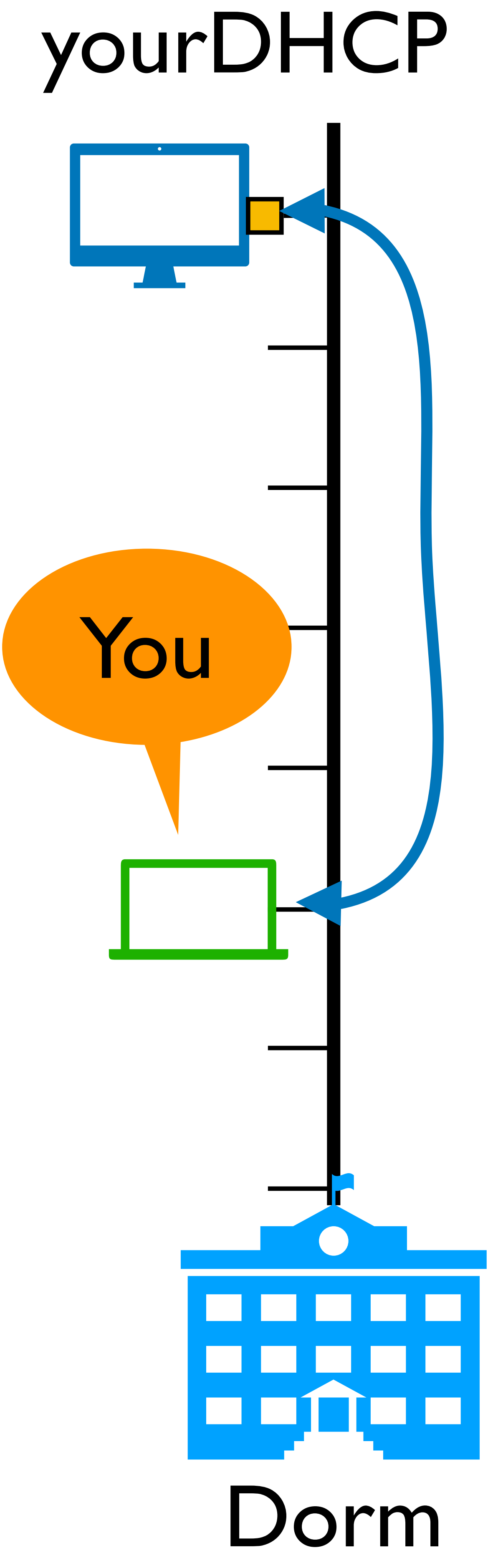
Walk through the steps required to download www.google.com/index.html from your laptop



Count the number of protocols that come into play!

- Assume: 'cold-start' — nothing cached anywhere
- Assume: yourDNS on a different subnet from yourDHCP
- Ignore intra- and inter-domain routing protocols

Step 1: Self Discovery



- You use DHCP to discover bootstrap parameters
  - Your IP address (u.u.u.u)
  - Your DNS server's IP (u.dns.ip.addr)
  - R's IP address (r.r.r.r)
  - …
- Exchange between you and yourDHCP

Ethernet	IP	UDP	DHCP

- Protocol count = 4

Next…

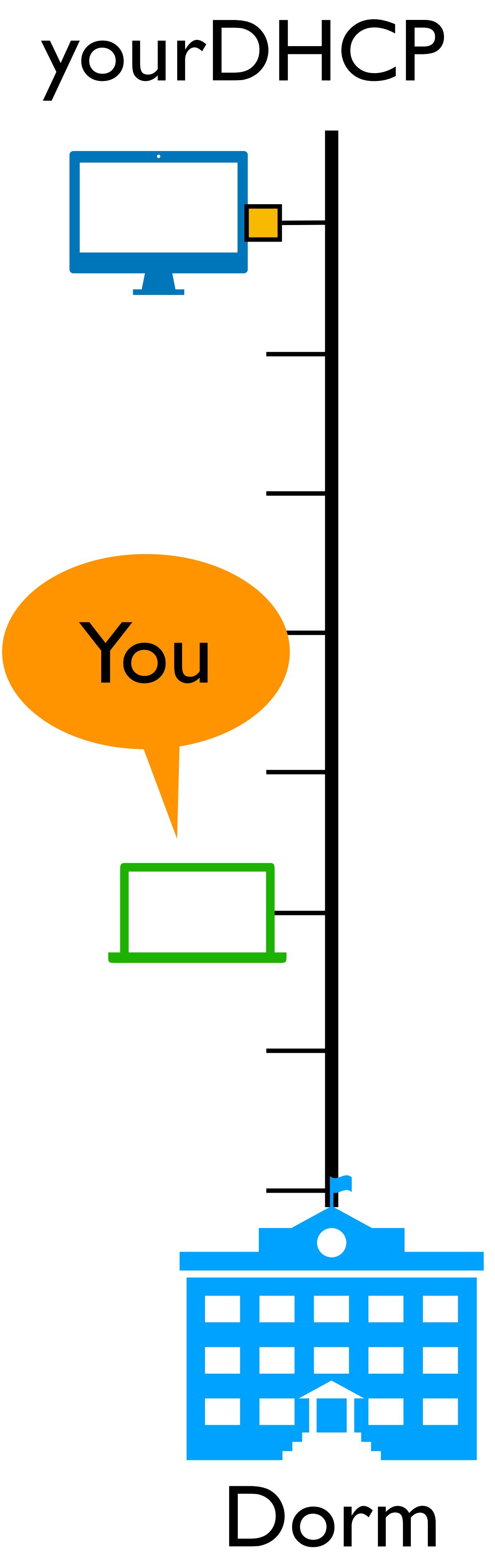
- **You are ready to contact www.google.com**
  - Need an IP address for www.google.com
  - Need to ask Google's DNS server
  - Need to ask yourDNS server to ask Google's DNS
  - I know my DNS server's IP address is u.dns.ip.addr
    - Create a packet to send…

Source: u.u.u.u
Dst: u.dns.ip.addr

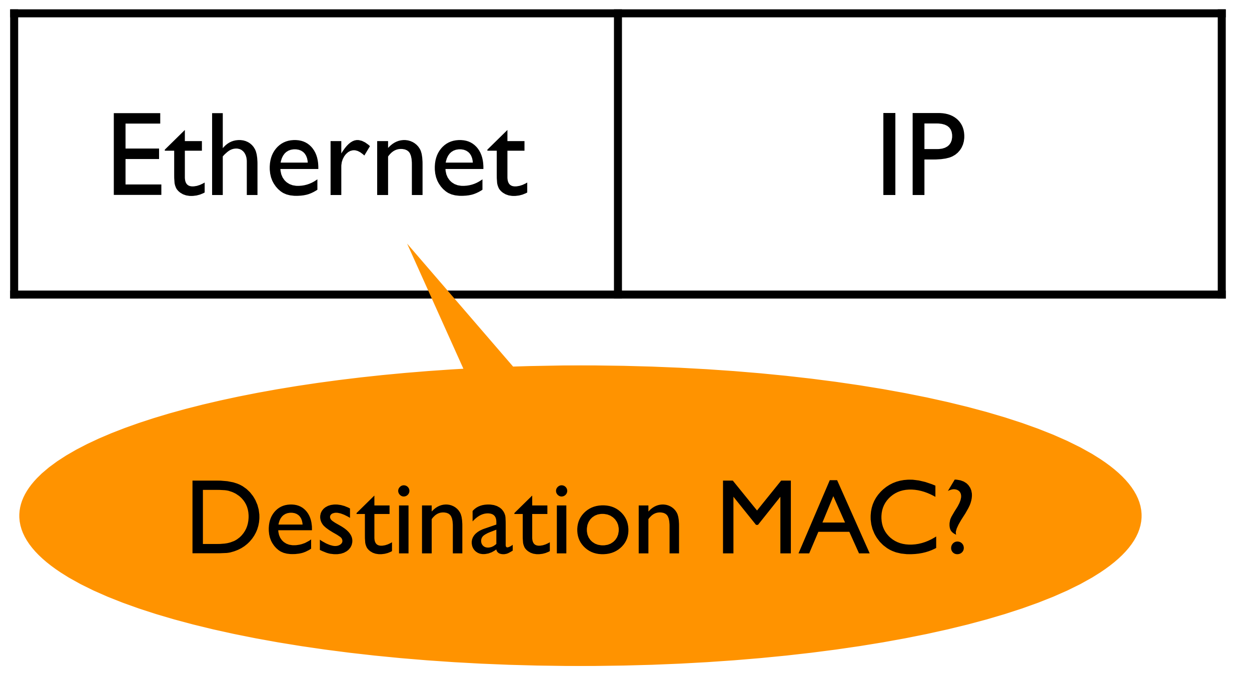
Ethernet	IP	UDP	DNS

Destination MAC?

Step 2: Getting out the door

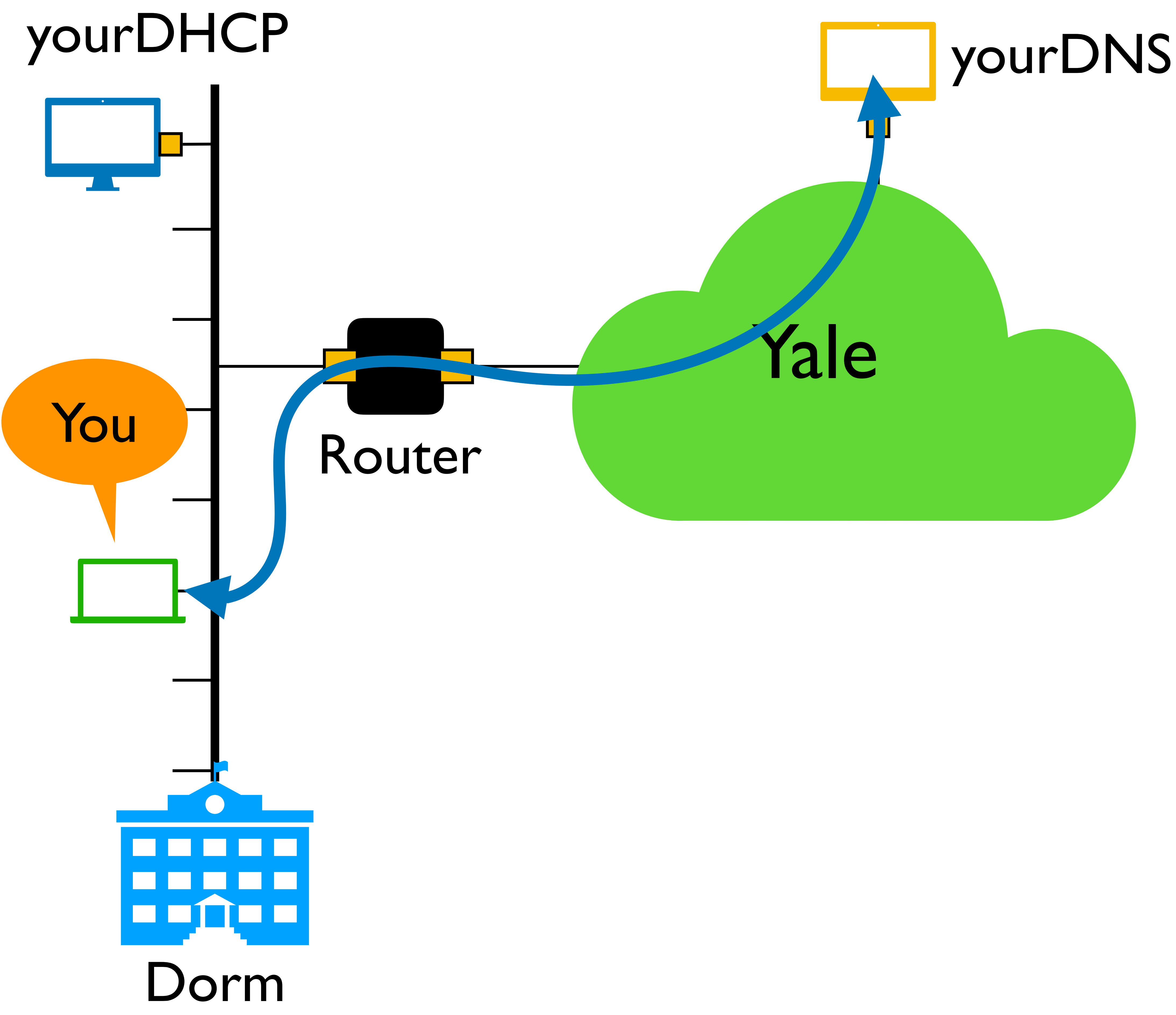


- You use ARP to discover the MAC address of R
- Exchange between you and R



- Protocol count = 5

Step 3: Send a DNS request



- Exchange between you and yourDNS
- Now ready to send that packet

Ethernet	IP	UDP	DNS

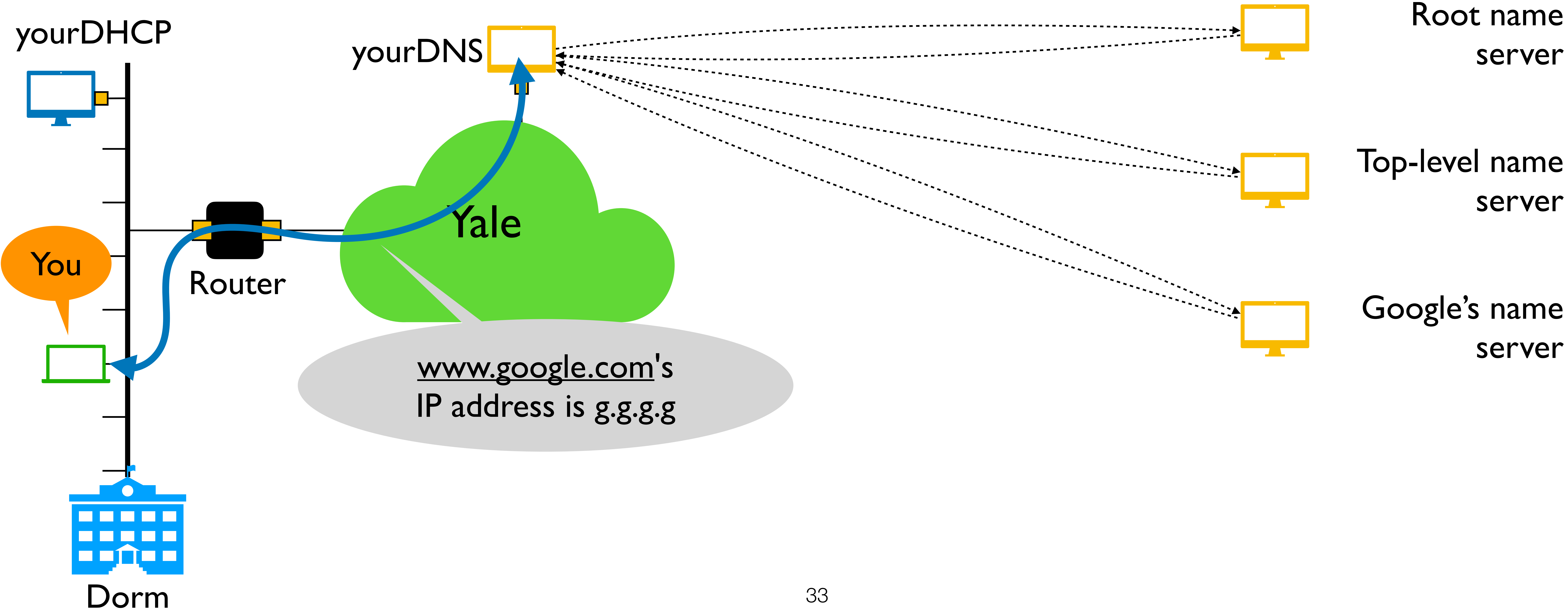
R's MAC?

Source: u.u.u.u
Dst: u.dns.ip.addr

- Protocol count = 6

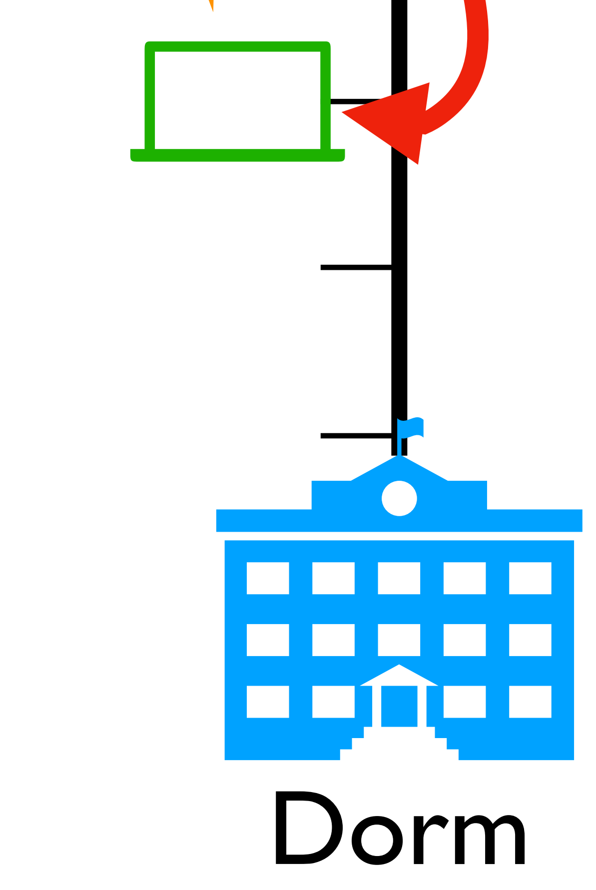
Step 4: yourDNS does its thing

- yourDNS resolves www.google.com

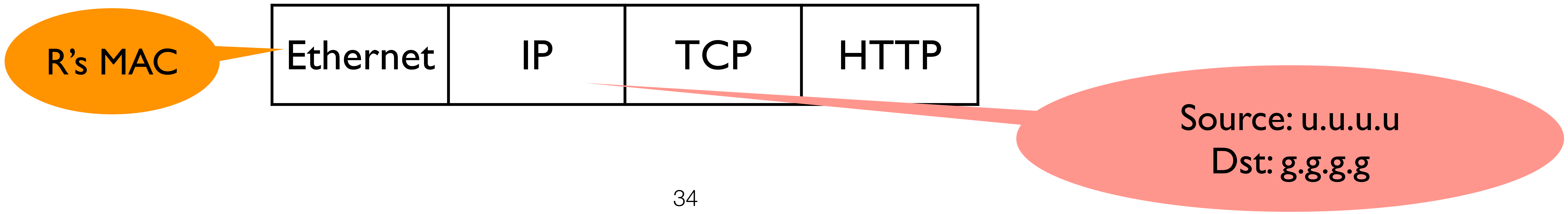


Step 5: Getting the content (at last)

Final protocol count = 8



- Exchange between you and Google's server at g.g.g.g



Recap: Name discovery/resoulution

- **MAC addresses?**
  - My own: hardcoded
  - Others: *ARP* (given IP address)

- **IP addresses?**
  - My own: *DHCP*
  - Others: *DNS* (given domain name)
    - How do I bootstrap DNS communication? (*DHCP*)

- **Domain names?**
  - Search engines

Questions?

Wireless Networks

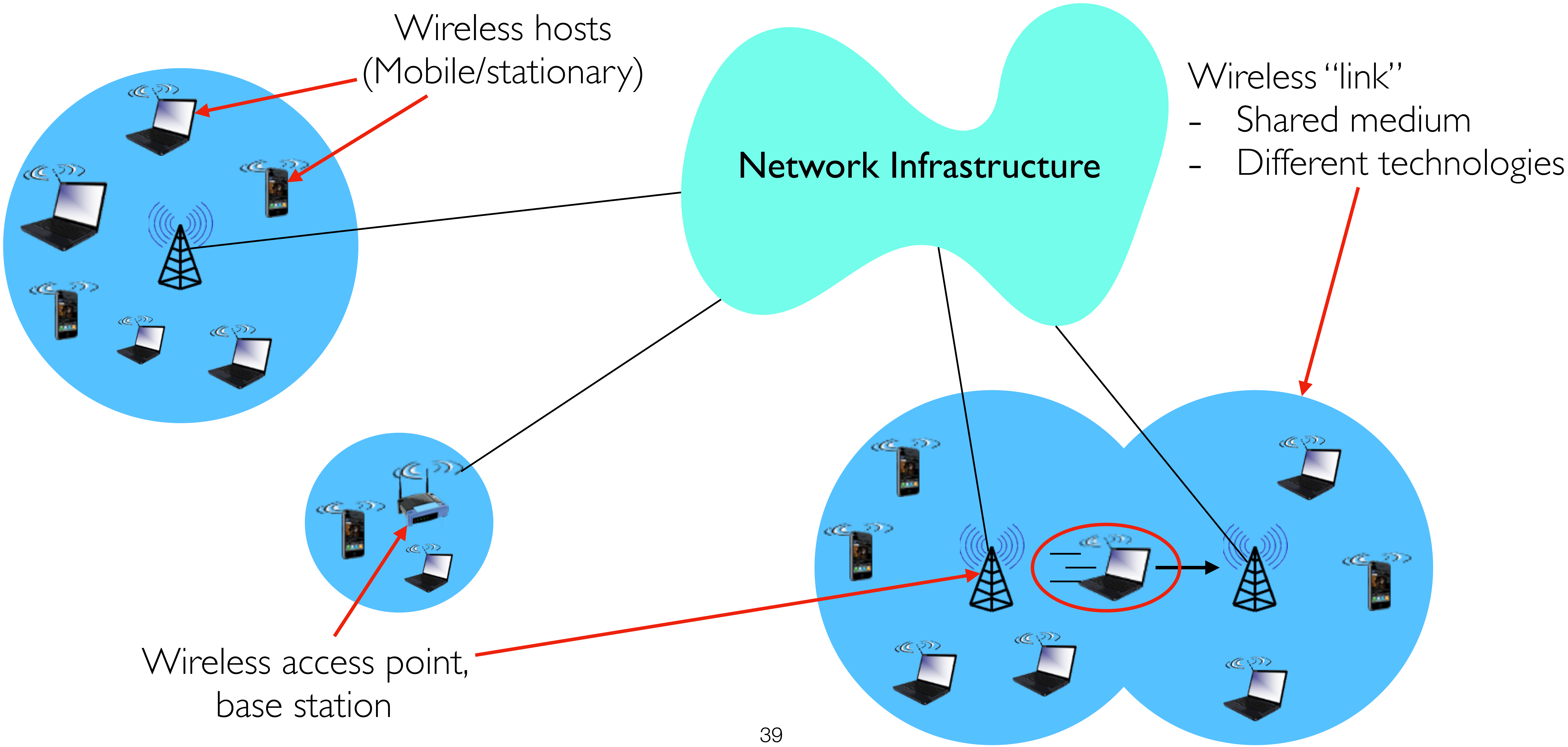
CPSC 433/533, Spring 2021
Anurag Khandelwal

Wireless Networks

- Albert Einstein, when asked to describe radio:

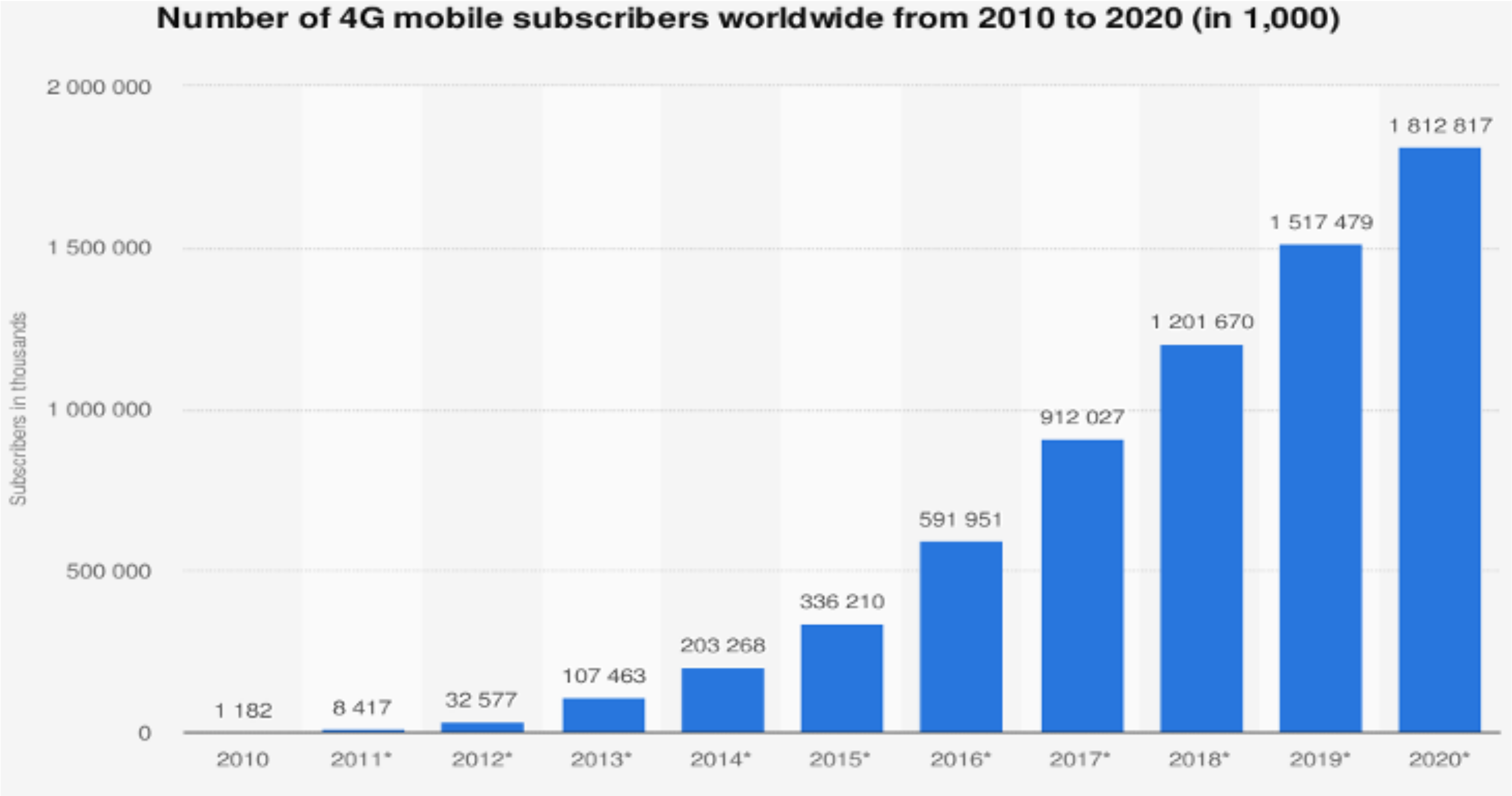
"You see, wire telegraph is a kind of a very, very long cat. You pull his tail in New York, and his head is meowing in Los Angeles. And radio operates exactly the same way… the only difference is that ***there is no cat****."*

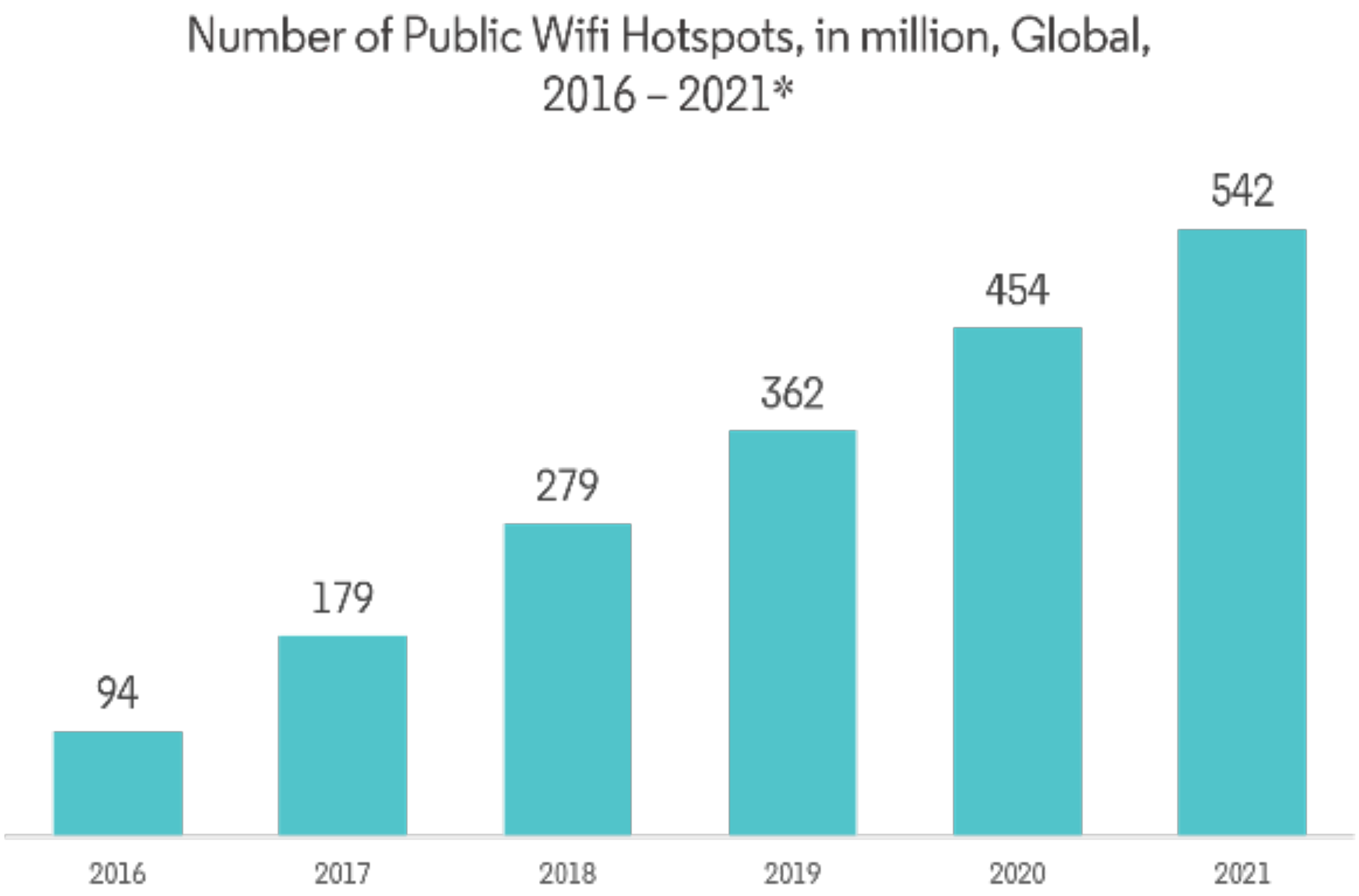
Elements of a Wireless Network



Widespread Deployment

- Worldwide 4G subscribers
  - 2010: ~1.2 million
  - 2015: ~336.2 million
  - 2020: ~1.8 billion





- Wireless LANs
  - Origins in ALOHANet, 1975
  - Now close to half a billion hotspots!

Wireless Technologies

Comparing Wireless Technologies

- Bitrate or Bandwidth
- Range - PAN (personal), LAN (local), MAN (metropolitan), WAN (wide)
- Stationary vs. Mobile
- Two-way vs. One-way
- Digital vs. Analog
- Multi-access vs. Point-to-point
- Frequency or Wavelength

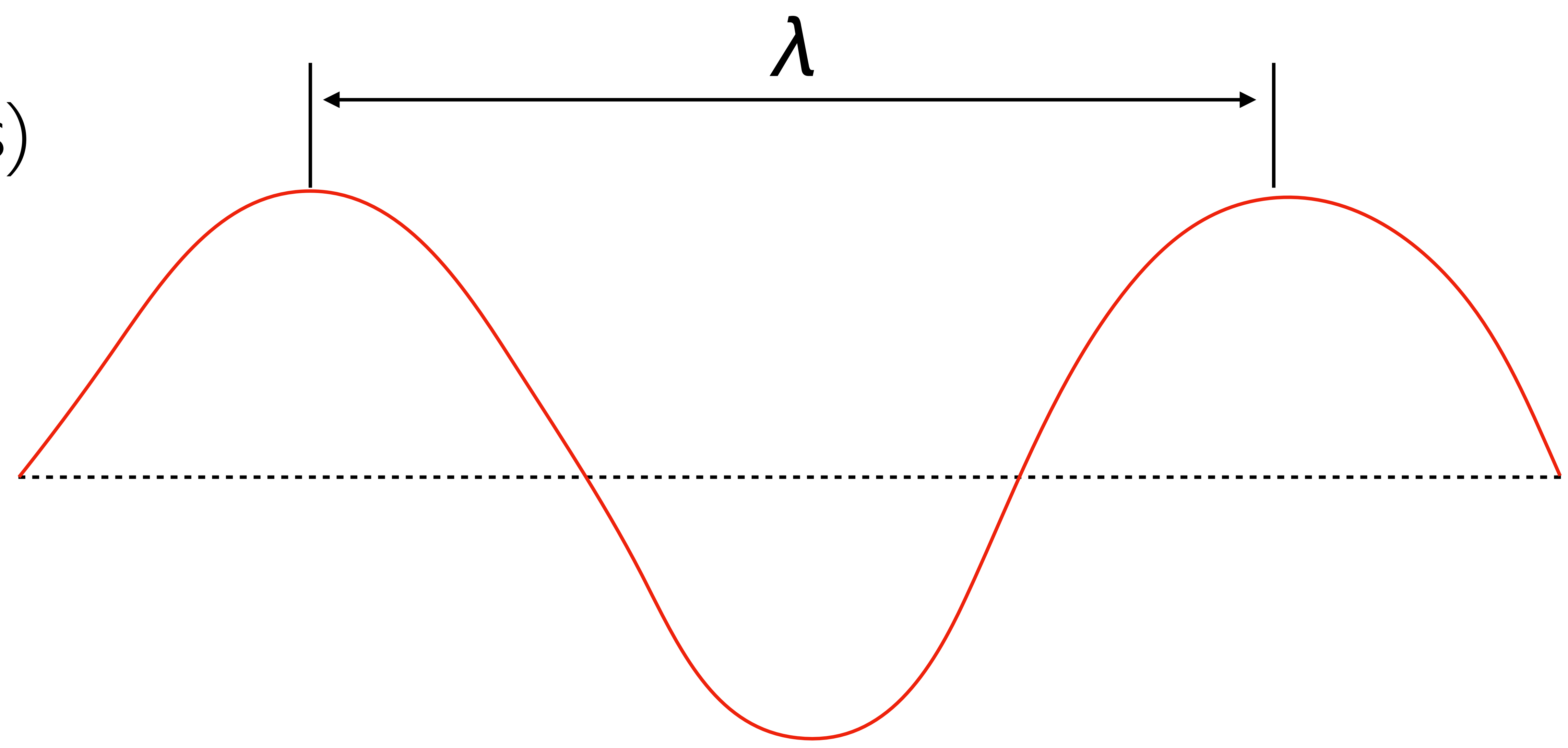
Comparing Wireless Technologies

- Bitrate or Bandwidth
- Range - PAN (personal), LAN (local), MAN (metropolitan), WAN (wide)
- Stationary vs. Mobile
- Two-way vs. One-way
- Digital vs. Analog
- Multi-access vs. Point-to-point
- Frequency or Wavelength

Frequency or Wavelength

Characteristics of a wireless signal

- Frequency: number of cycles per second
- Wavelength: length of each cycle (in meters)

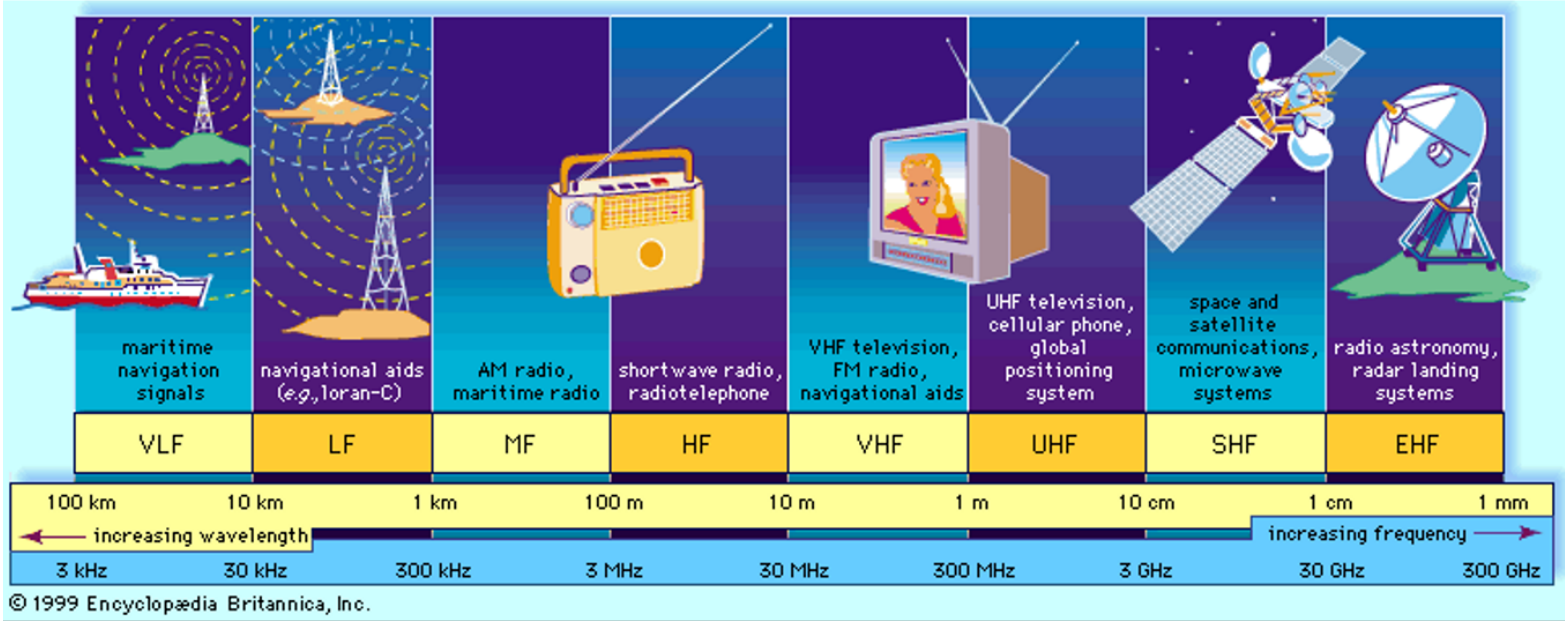


- c is the speed of light
- *f* is frequency
- λ is wavelength

$$\boxed{\lambda = \frac{c}{f}}$$

- Q: How do we share the wireless "medium"?
- A: Frequency division
  - Divide frequency spectrum into frequency bands
  - Allocate frequency band to different wireless technologies

The Wireless "Spectrum"

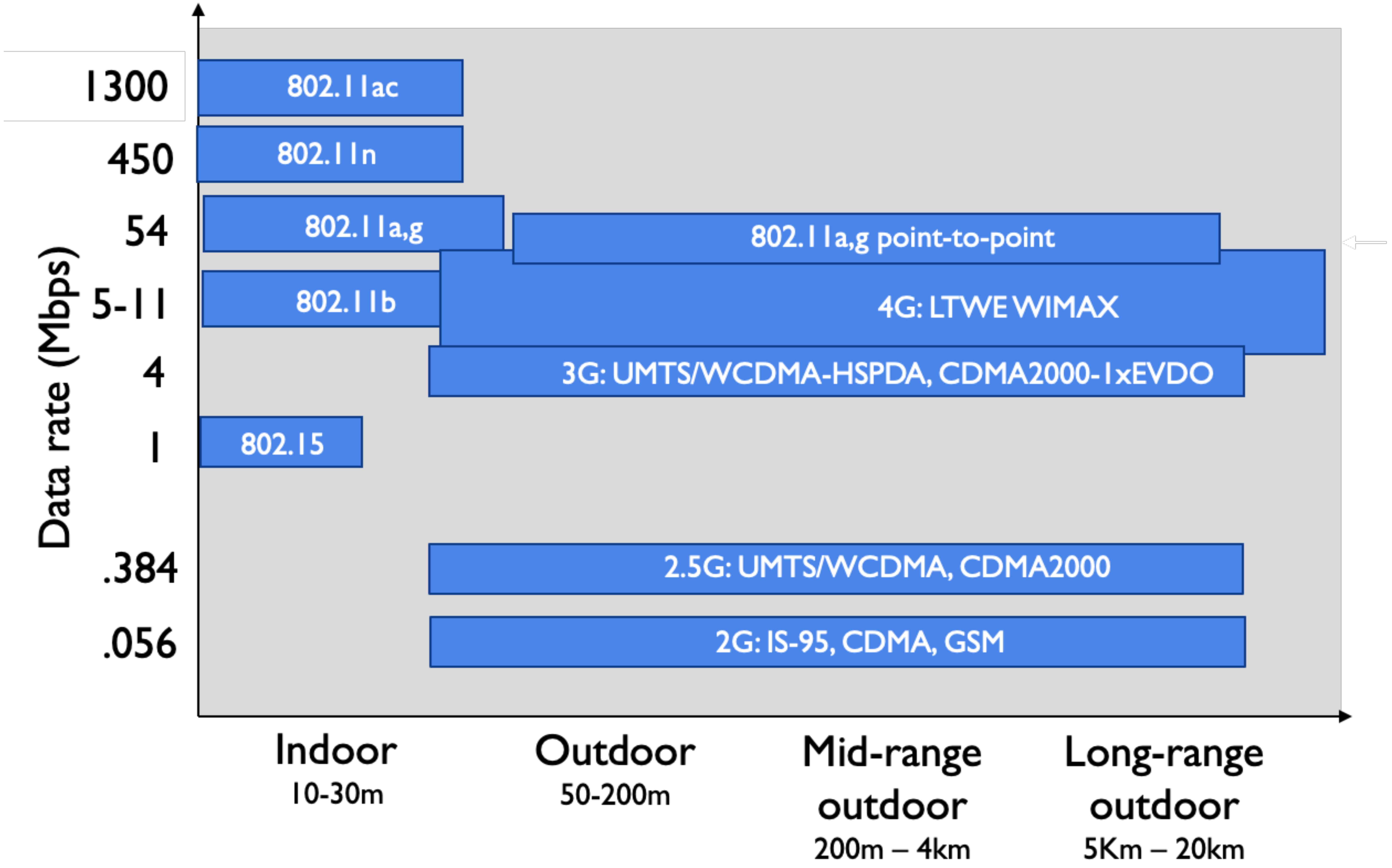


US Frequency Allocations

Common Wireless Standards

- **Cellular (Typically 800/900/1800/1900/3300/4200 Mhz)**
  - 2G: GSM/GPRS/EDGE/CDMA/CDMA2000
  - 3G: UMTS/HSDPA/EVDO; 4G: LTE, WiMax; 5G: LTE
- **IEEE 802.11 (aka WiFi):**
  - b: 2.4Ghz band, 11 Mbps (~4.5 Mbps operating rate)
  - g: 2.4Ghz, 54-108Mbps (~19 Mbps operating rate)
  - a: 5Ghz band, 54-108Mbps (~19 Mbps operating rate)
  - n: 2.4/5Ghz, 150-600Mbps (4x4 MIMO)
  - ac: 2.4/5Ghz, >1Gbps (4x4 MIMO) (wide channels)
- **IEEE 802.15 — lower power wireless:**
  - 802.15.1: 2.4Ghz, 2.1 Mbps (Bluetooth)
  - 802.15.4: 2.4Ghz, 250 Kbps (Sensor networks)

Wireless “Link” Characteristics



Questions?

What makes wireless different?

- **Consider specific frequency band / wireless technology**
  - What makes it different from wired technologies?
- **A lot really…**
- **Broadcast medium**
  - Anybody in proxy can hear & interfere
- **Cannot receive while transmitting**
  - Our own (or nearby) transmitter is deafening our receiver
- **Signals from sender not always intact at receiver**
  - Complicated physics involved, which won't discuss
  - But what can go wrong?

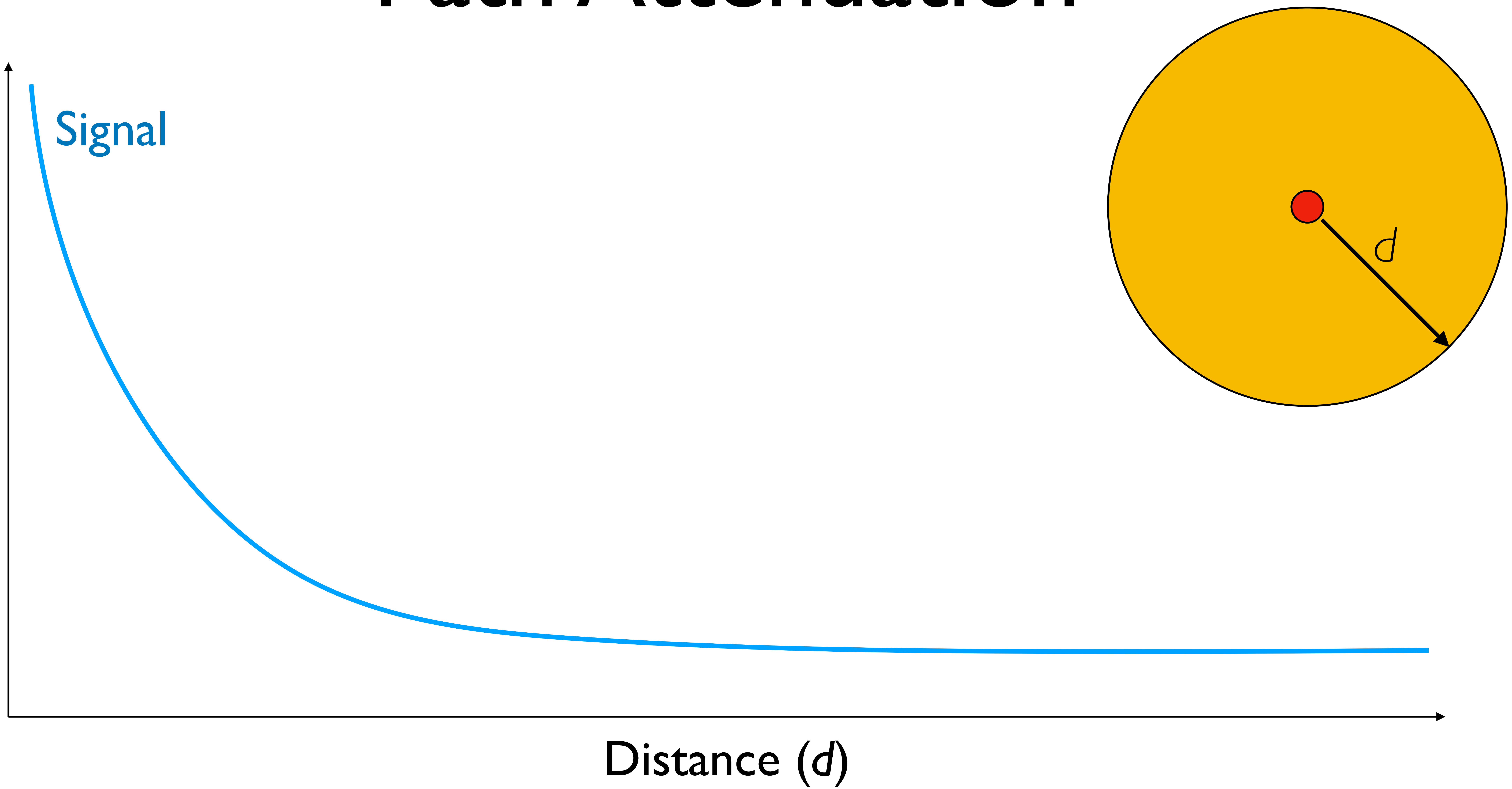
Path Loss / Path Attenuation

- Free space path loss: Reduction in power density of an EM wave as it propagates through space
  - d = distance
  - c is the speed of light
  - f is frequency
  - λ is wavelength

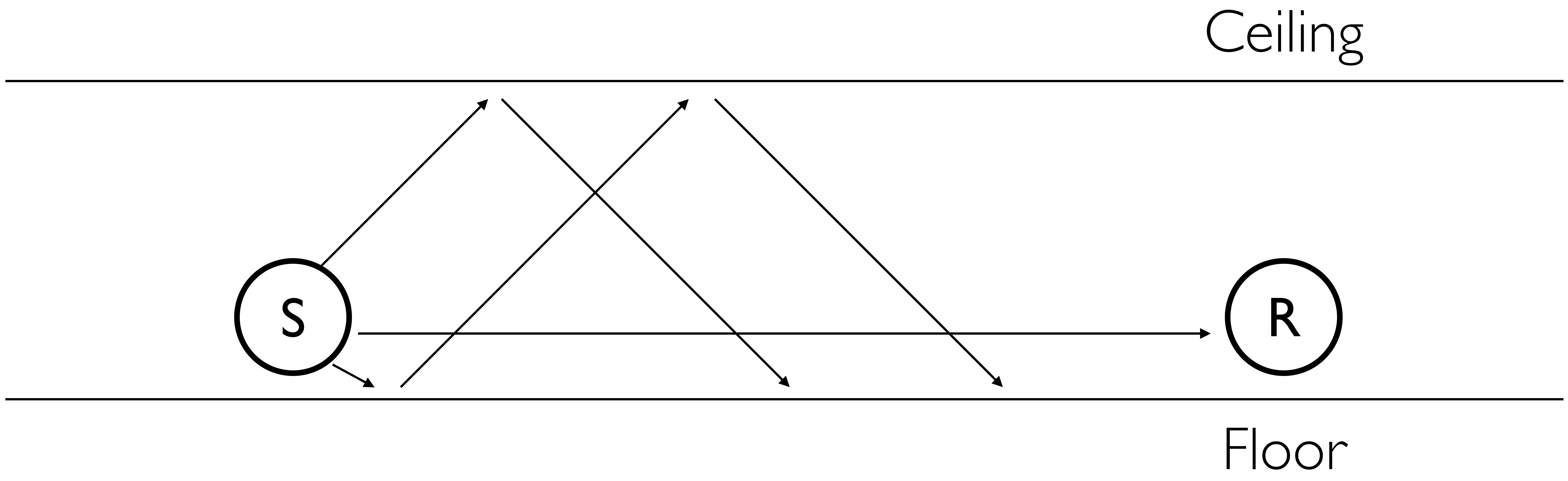
$$\text{FSPL} = \left(\frac{4\pi d}{\lambda}\right)^2 = \left(\frac{4\pi df}{c}\right)^2$$

- Other reasons (not include in FSPL)
  - Reflection, Diffraction, Absorption
  - Terrain counters (urban, rural, vegetation)
  - Humidity
  - …

Path Attenuation



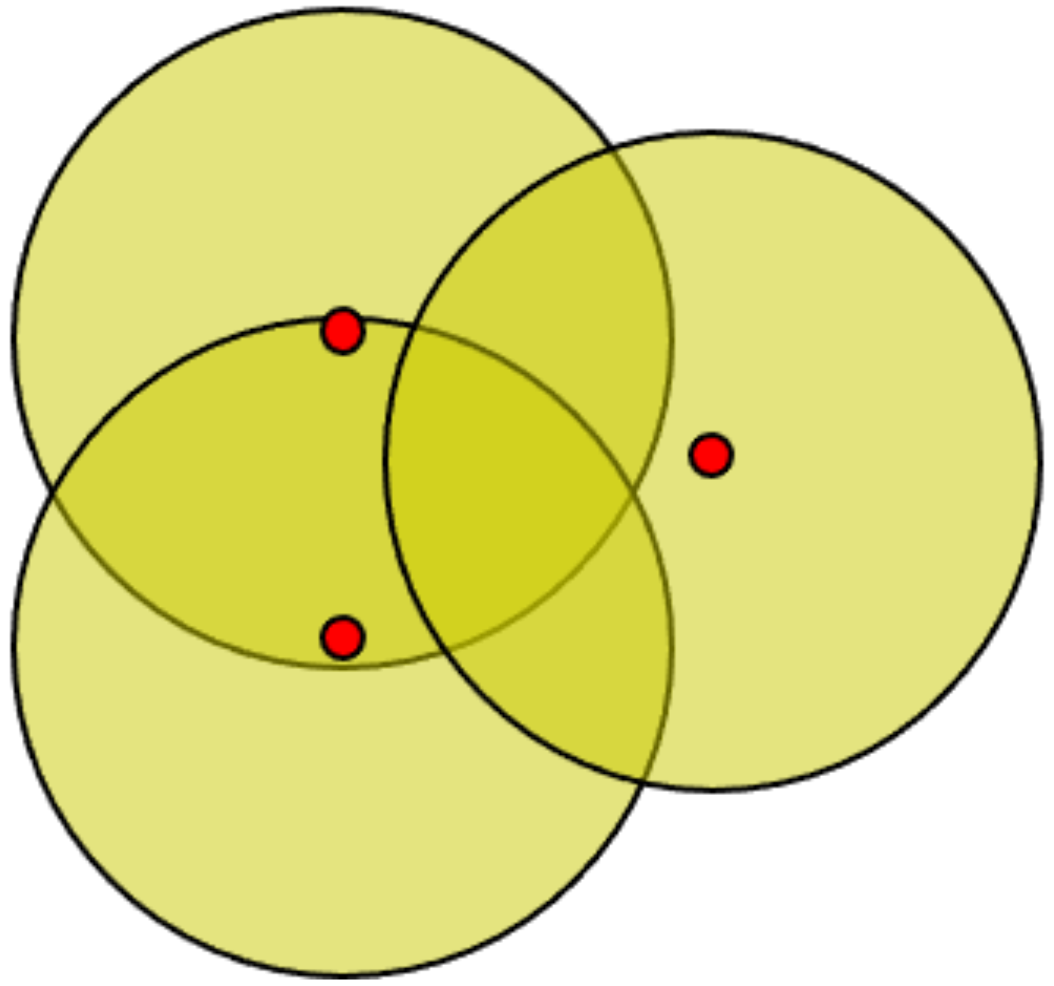
Multipath Effects



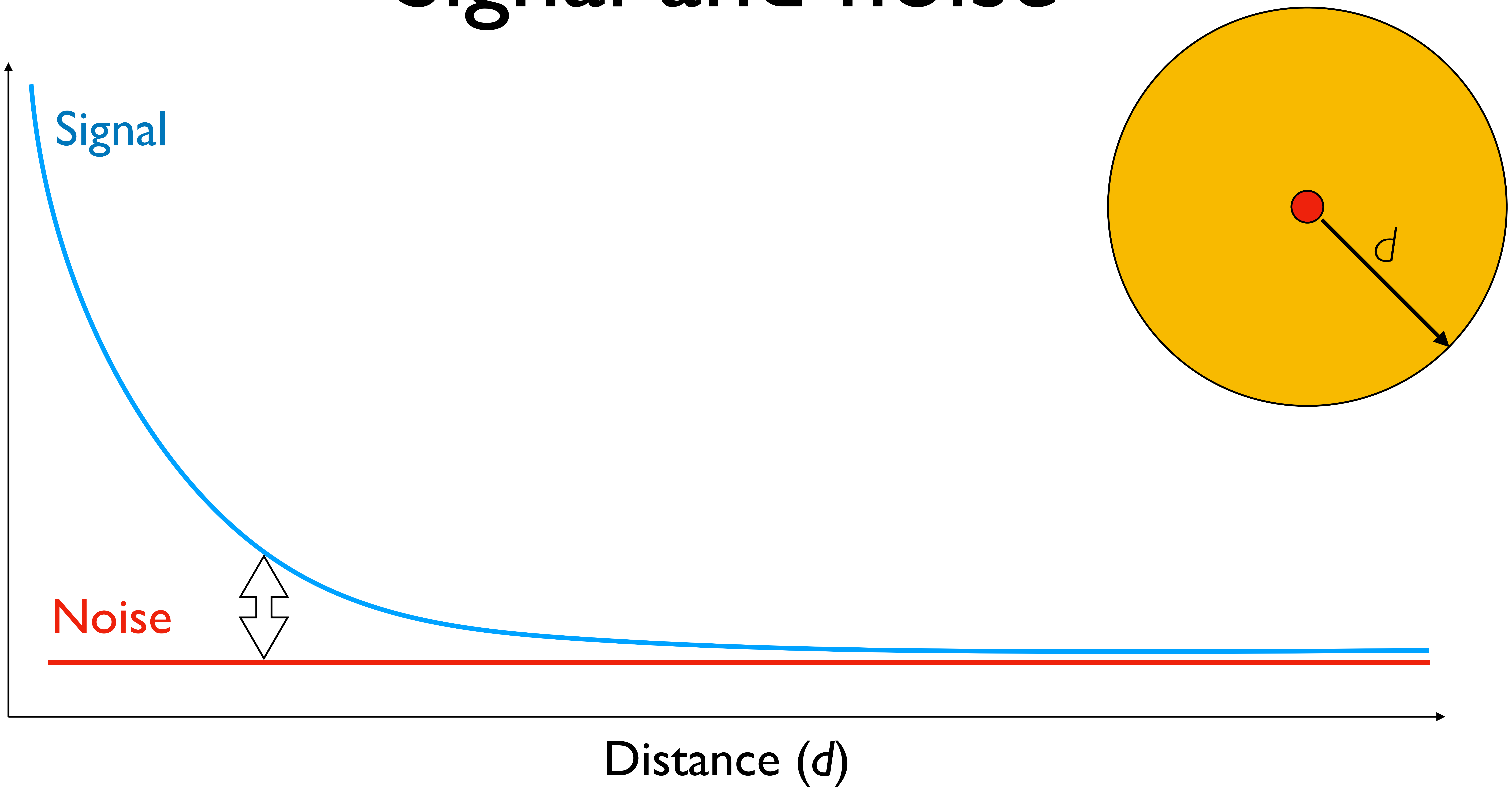
- Signals bounce off surface and interfere with one another
- Self-interference

Interference

- **External Interference**
  - E.g., microwave oven blocks your signal
- **Internal Interference**
  - Nodes of the name network within range of each other collide with one another's transmissions



Signal and noise

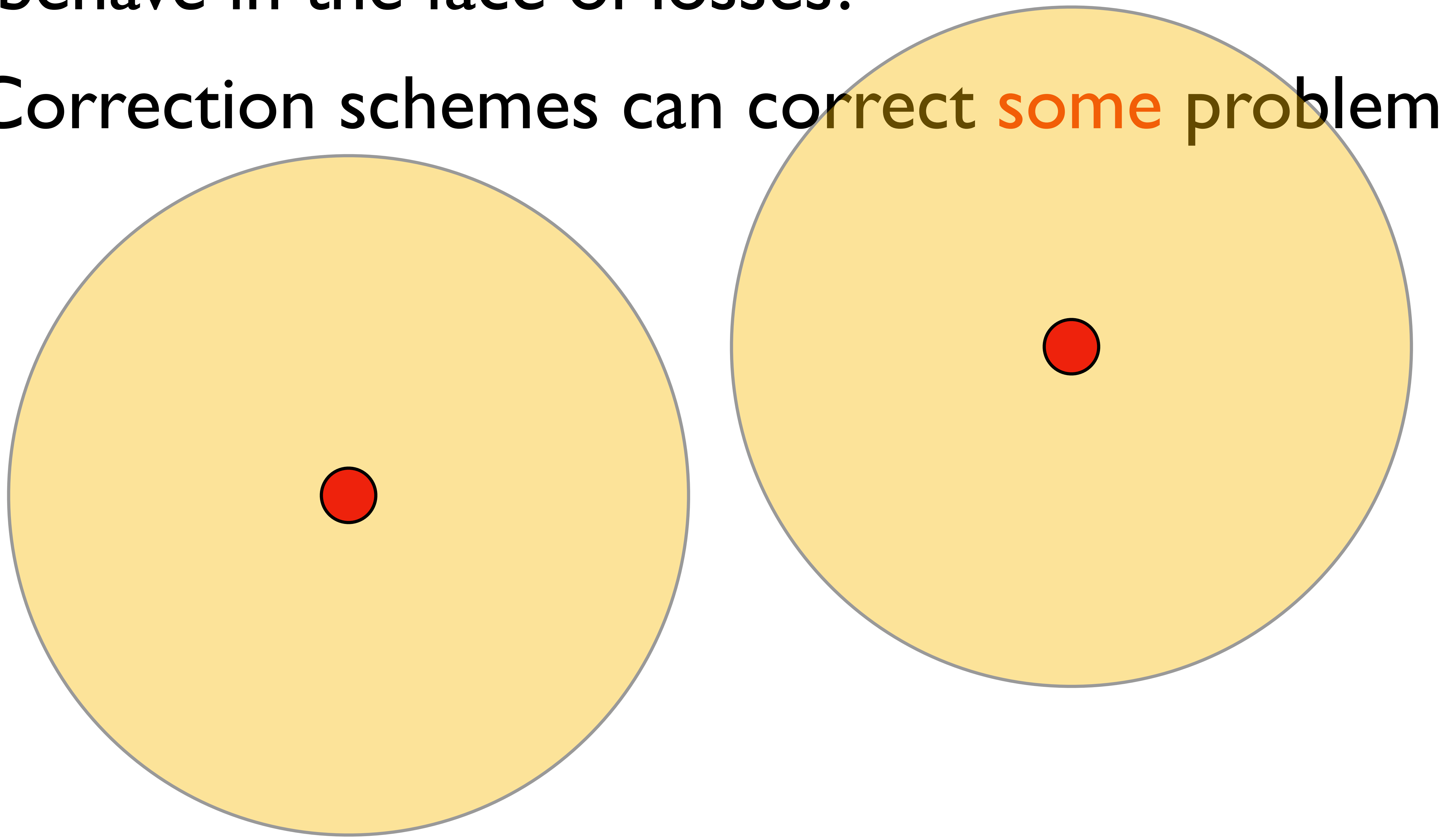


Signal-to-noise Ratio (SNR)

- Relative ratio of:
  - Strength of received signal, and,
  - Background noise
- Larger SNR = easier to extract signal from noise
- Lower SNR = higher Bit Error Rate (BER)
- Can't we make the signal stronger then?
- Not always a good idea…

Signal-to-noise Ratio (SNR)

- **Why isn't increasing SNR always a good idea?**
  - Increased signal strength requires more power
  - Increases the interference range of the sender, so you interfere with more nodes around you
- **How would TCP behave in the face of losses?**
- **Link-layer Error Correction schemes can correct some problems**



Questions?