

Discovery Protocols & End-to-End Communication

CPSC 433/533, Spring 2021

Anurag Khandelwal

Administrivia

Administrivia

- Project 2 due on Thursday

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”

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

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

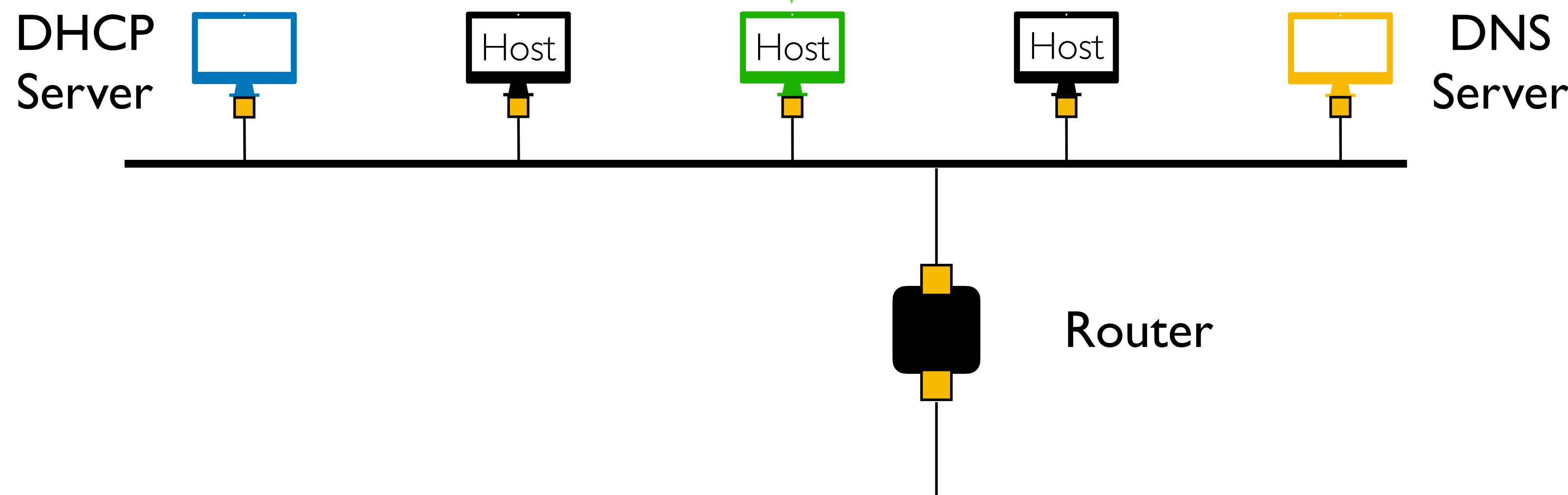
- A host is “born” knowing only its MAC address
- Must discover lots of information before it can communicate with a remote host B
 - What is my IP address?
 - What is B's IP address?
 - What is B's MAC address? (if B is local)
 - What is my first-hop router's MAC address? (if B is not local)
 - ...

DHCP

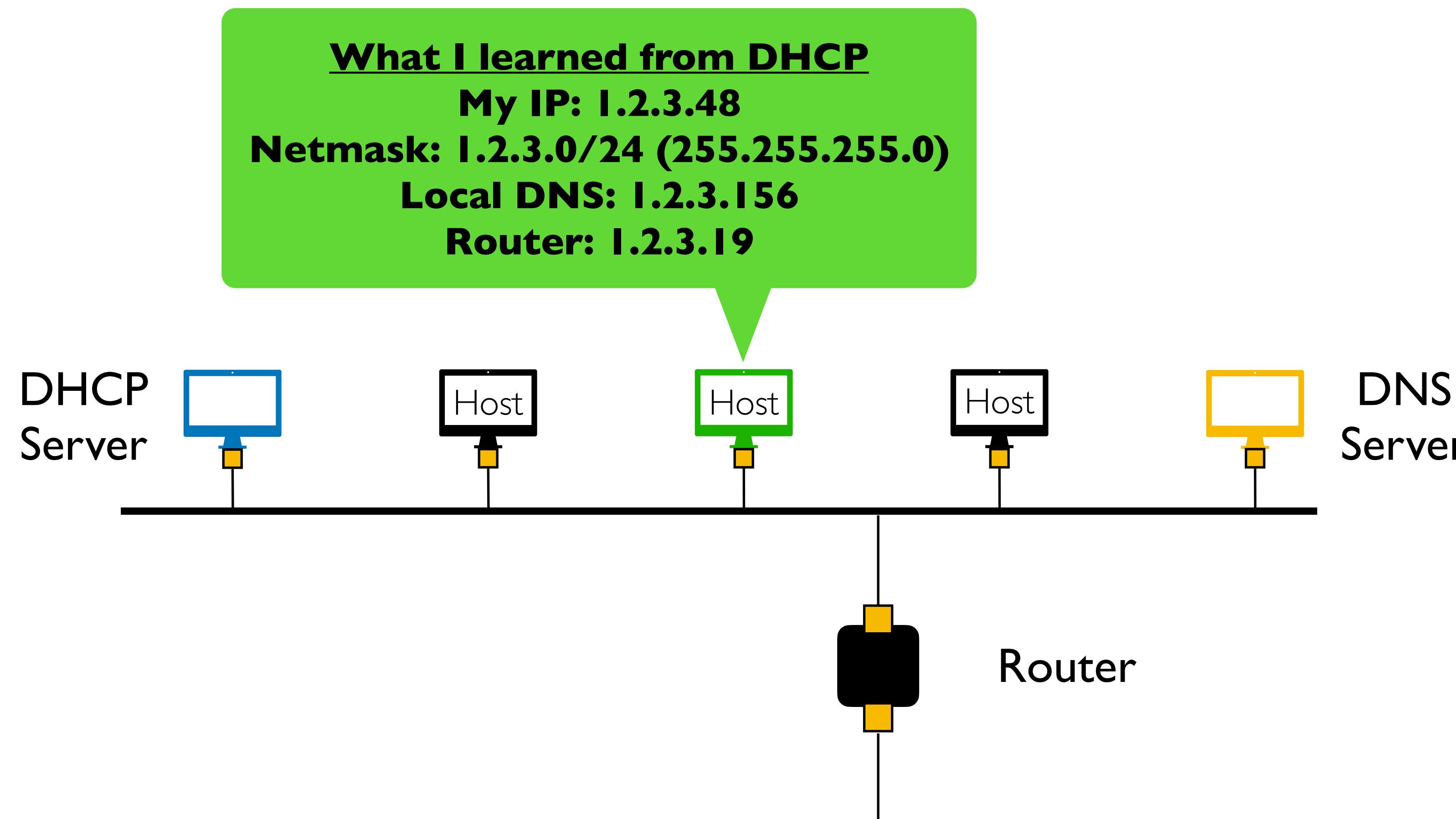
- “Dynamic Host Configuration Protocol”
 - Defined in RFC 2131
- A host uses DHCP to discover
 - Its own IP address
 - Its netmask
 - IP address(es) for its local DNS name server(s)
 - IP address(es) for its first-hop “default” router(s)

What I learned from DHCP

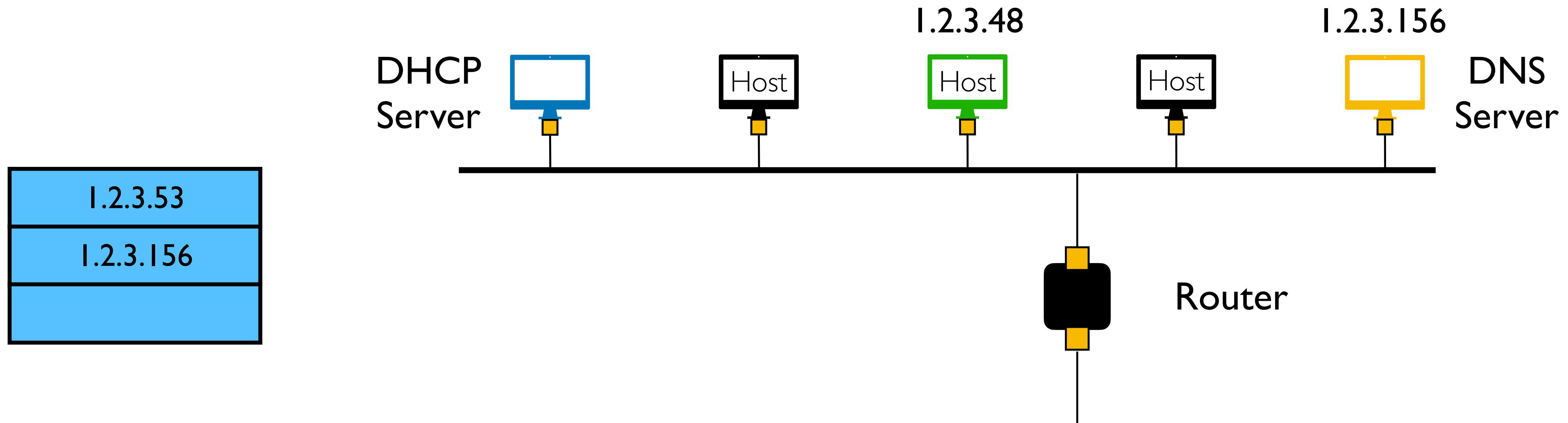
My IP: 1.2.3.48
Netmask: 1.2.3.0/24 (255.255.255.0)
Local DNS: 1.2.3.156
Router: 1.2.3.19



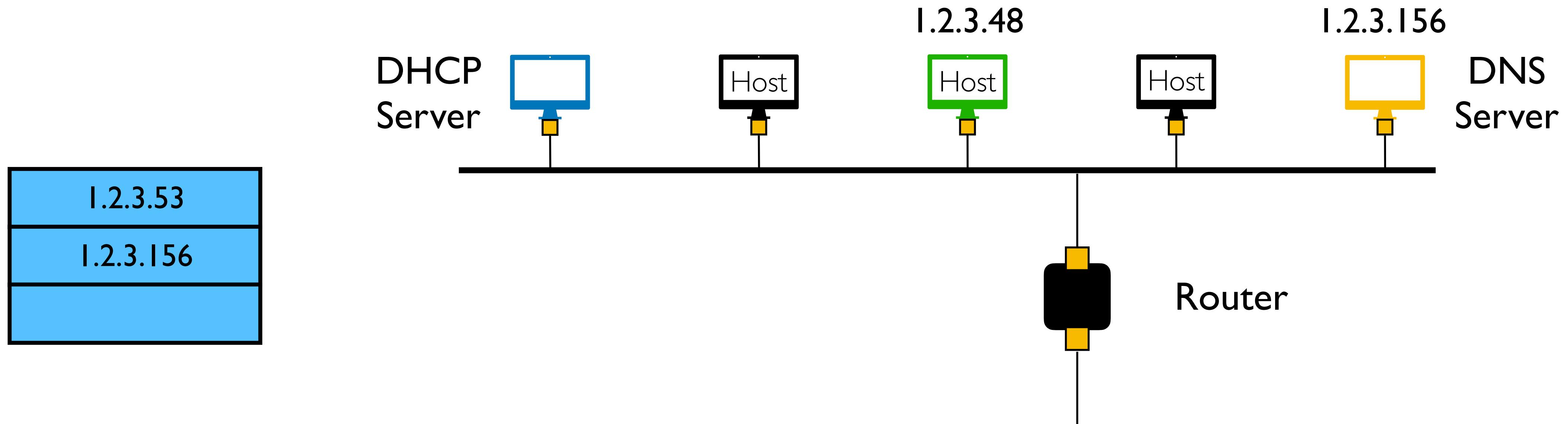
Are we there yet?



Sending packets over Link-Layer

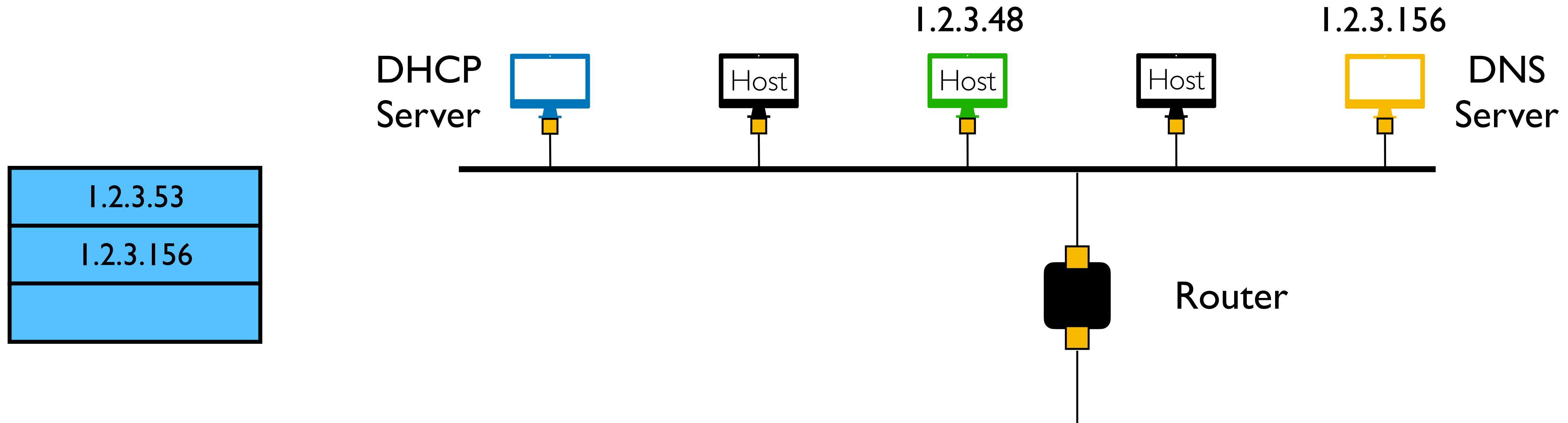


Sending packets over Link-Layer



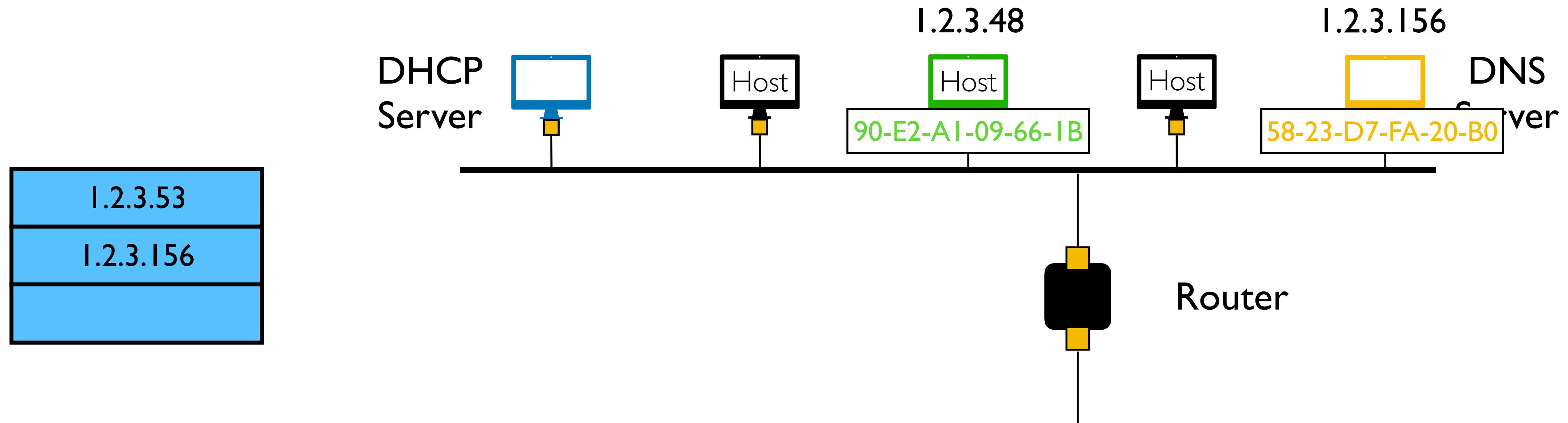
- Encapsulate the IP packet in a link-level (Ethernet) frame

Sending packets over Link-Layer



- Encapsulate the IP packet in a link-level (Ethernet) frame
- What's missing?

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

ARP: Address Resolution Protocol

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

ARP: Address Resolution Protocol

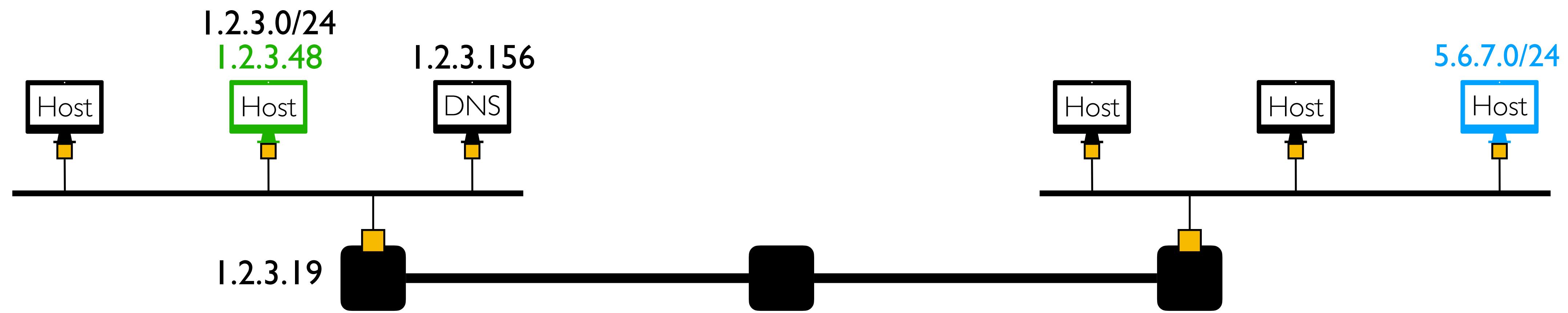
- Every host maintains an ARP table
 - List of (IP address → MAC address) pairs
- Consult the table when sending a packet
 - Map destination IP address to destination MAC address
 - Encapsulate the (IP) data packet with MAC header; transmit

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
 - Encapsulate 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?

What if the destination is remote?

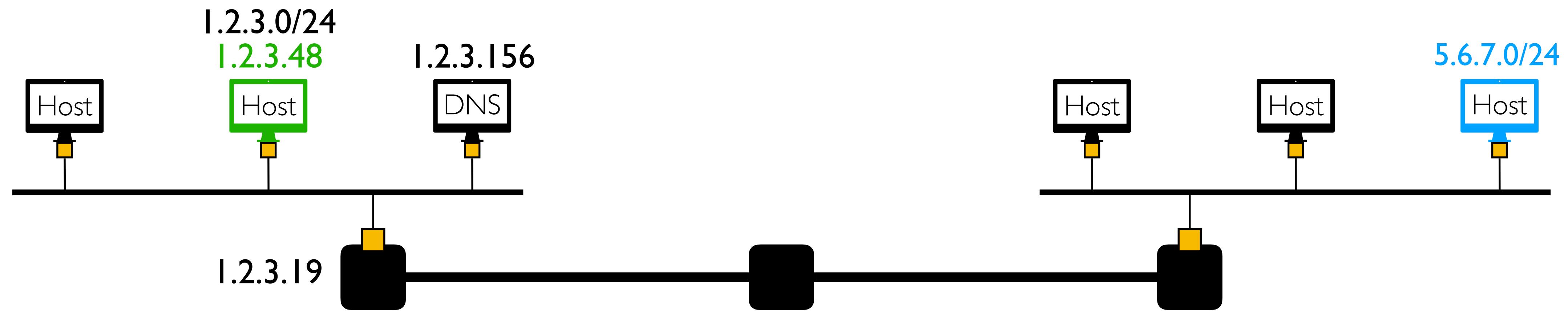


What if the destination is remote?



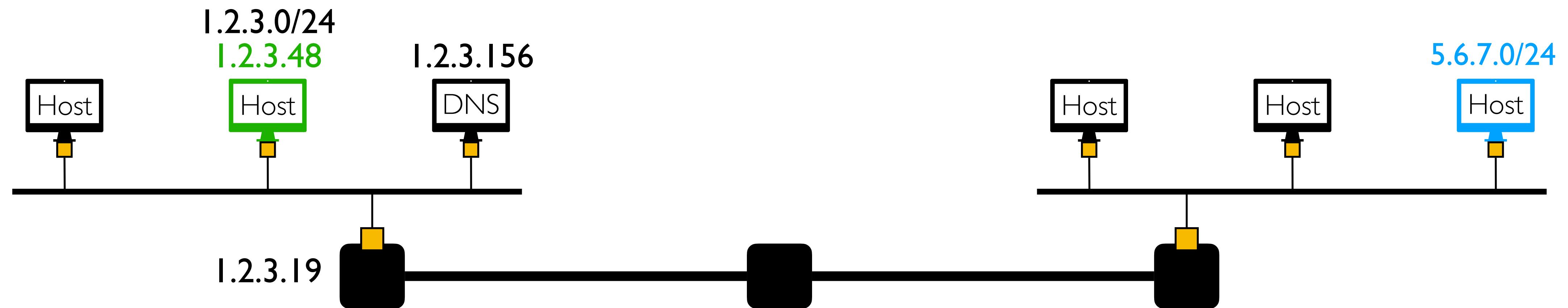
What if the destination is remote?

- Look up the MAC address of the first hop router



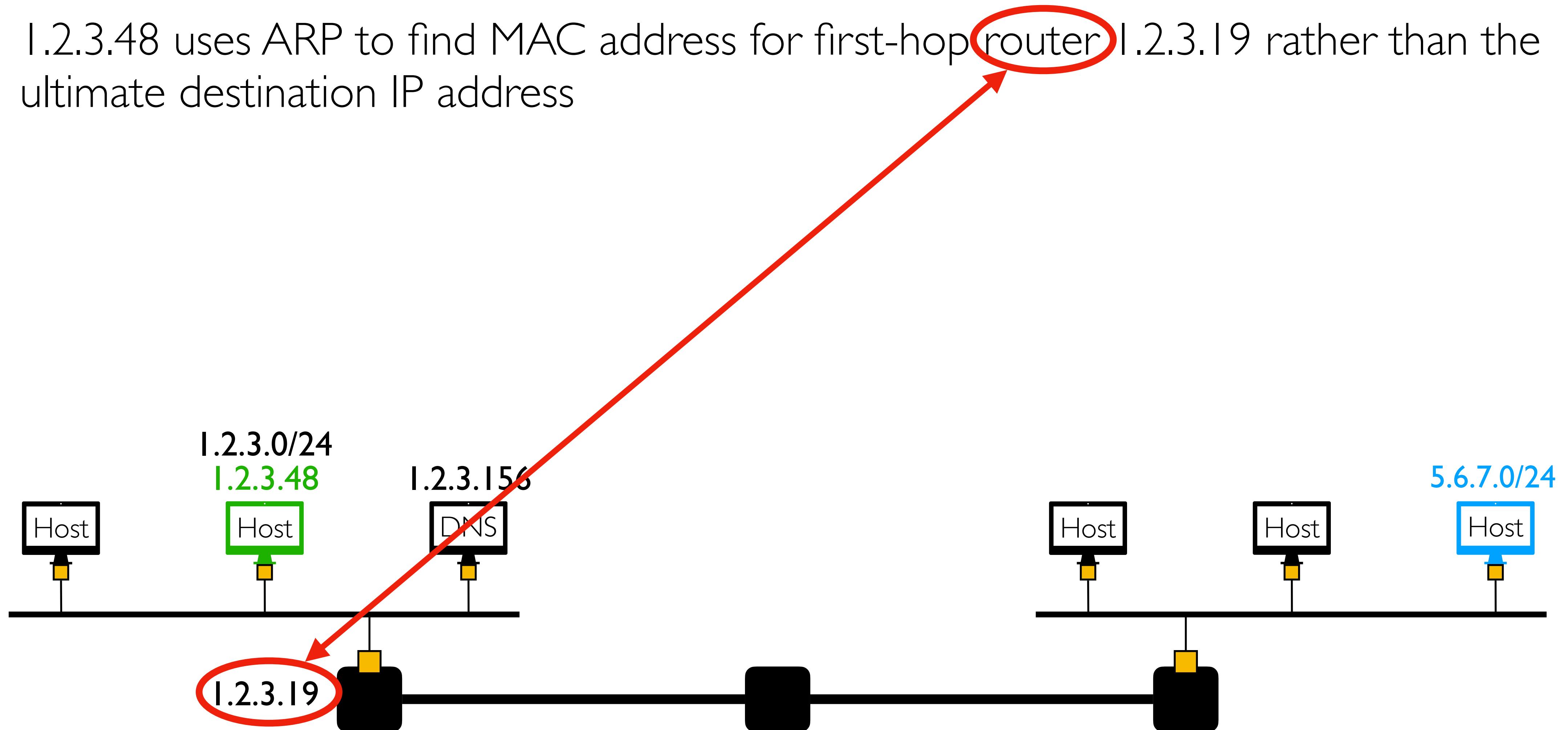
What if the destination is remote?

- Look up the MAC address of the first hop router
 - I.2.3.48 uses ARP to find MAC address for first-hop router I.2.3.19 rather than the ultimate destination IP address



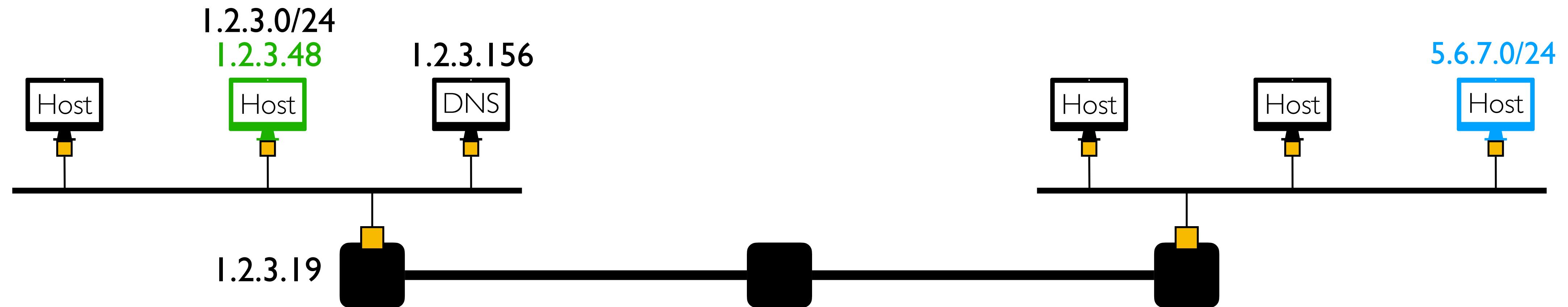
What if the destination is remote?

- Look up the MAC address of the first hop router
 - I.2.3.48 uses ARP to find MAC address for first-hop **router** I.2.3.19 rather than the ultimate destination IP address



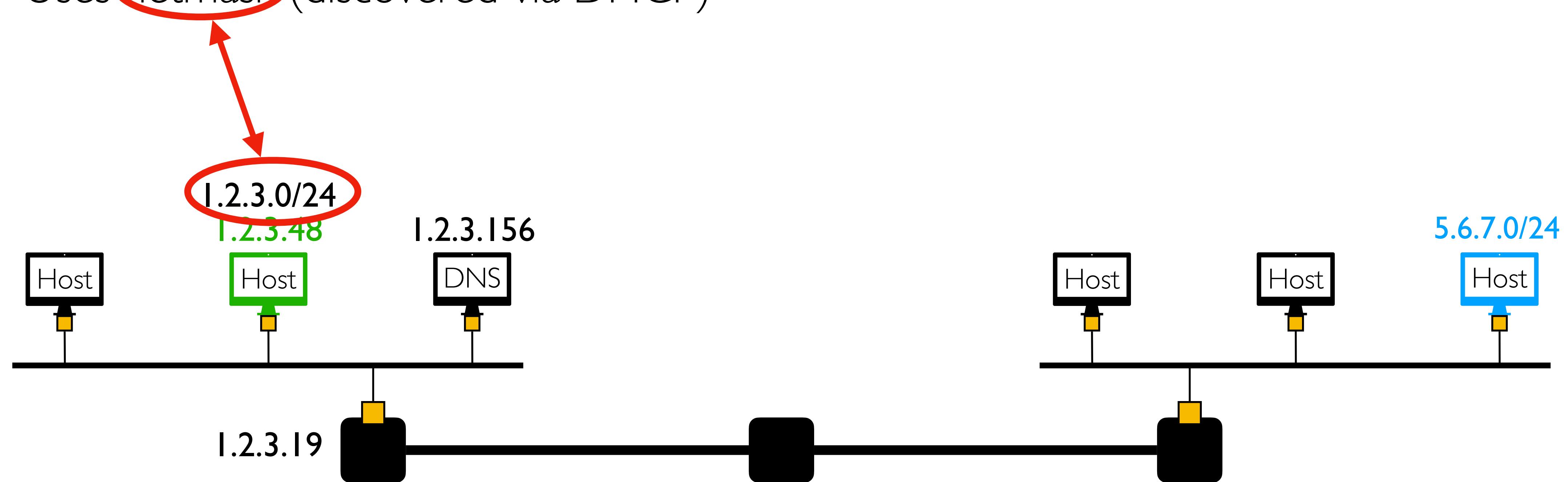
What if the destination is remote?

- Look up the MAC address of the first hop router
 - I.2.3.48 uses ARP to find MAC address for first-hop router I.2.3.19 rather than the ultimate destination IP address
- How does the green host know the destination is not local?



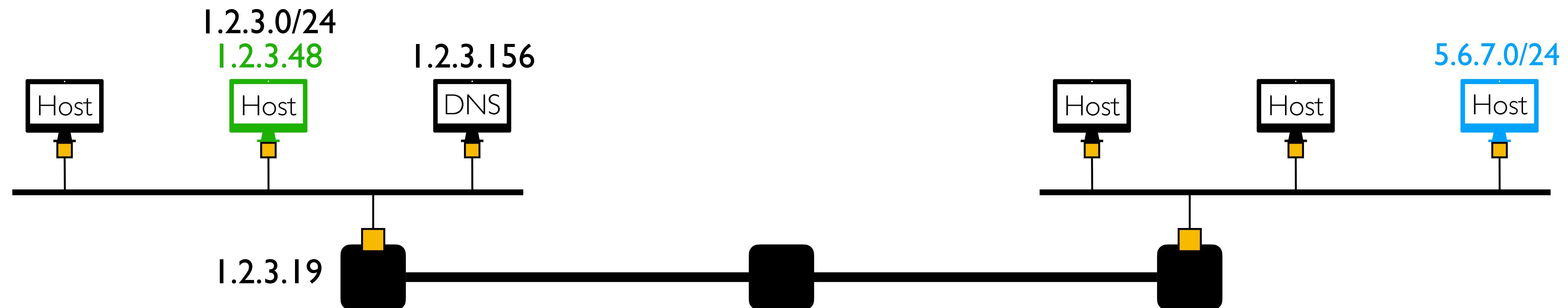
What if the destination is remote?

- 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)



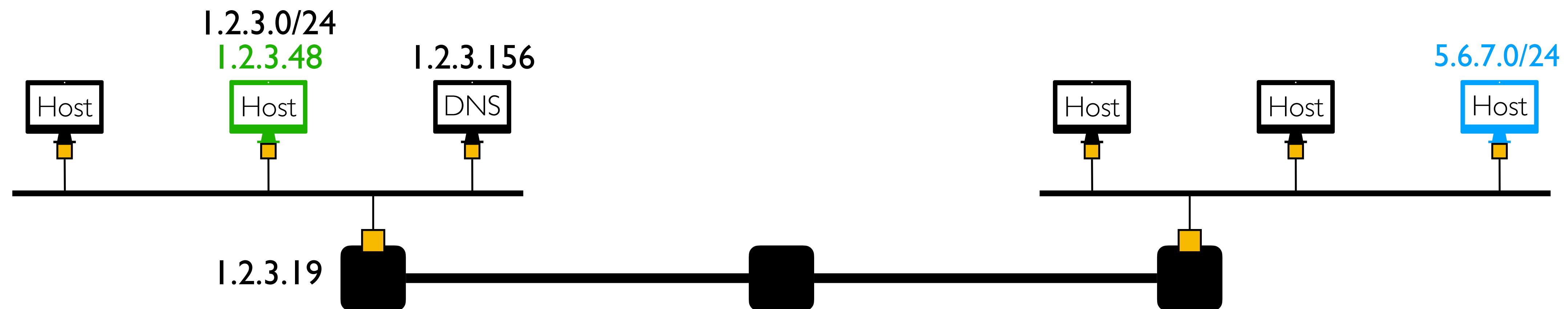
What if the destination is remote?

- 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)?



What if the destination is remote?

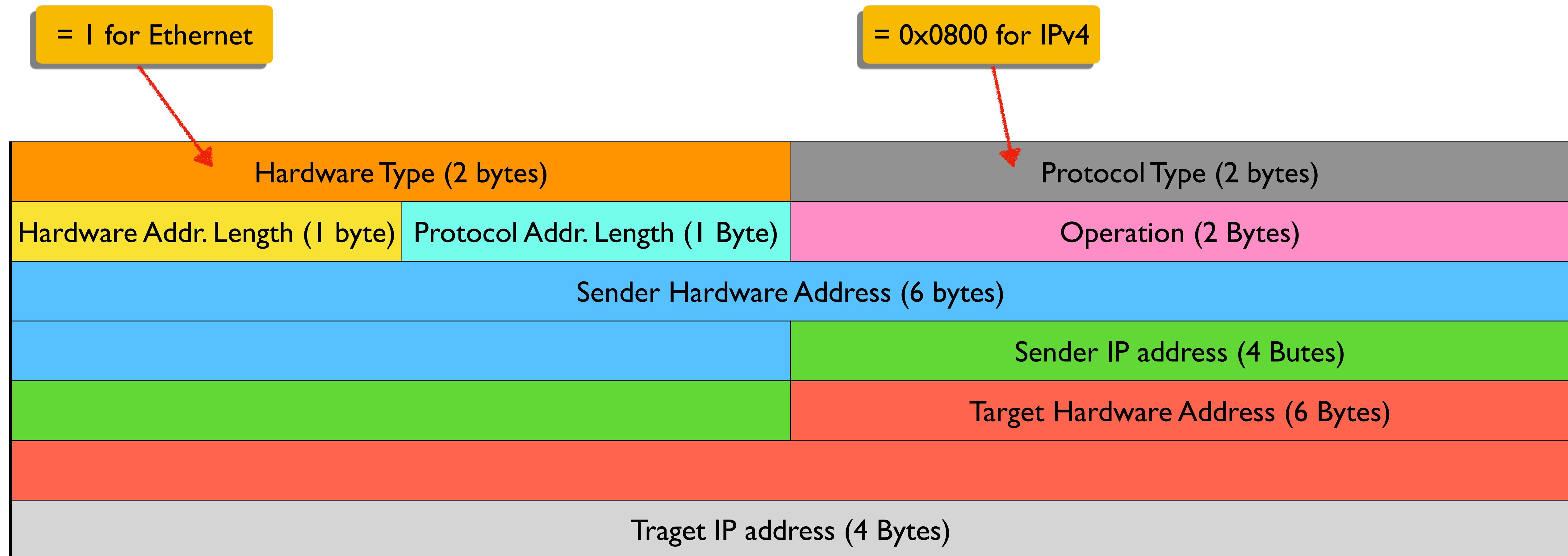
- Look up the MAC address of the first hop router
 - 1.2.3.48 uses ARP to find MAC address for first-hop router 1.2.3.19 rather than the ultimate destination IP address
- How does the green host know the destination is not local?
 - Uses netmask (discovered via DHCP)
- How does the green host know about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



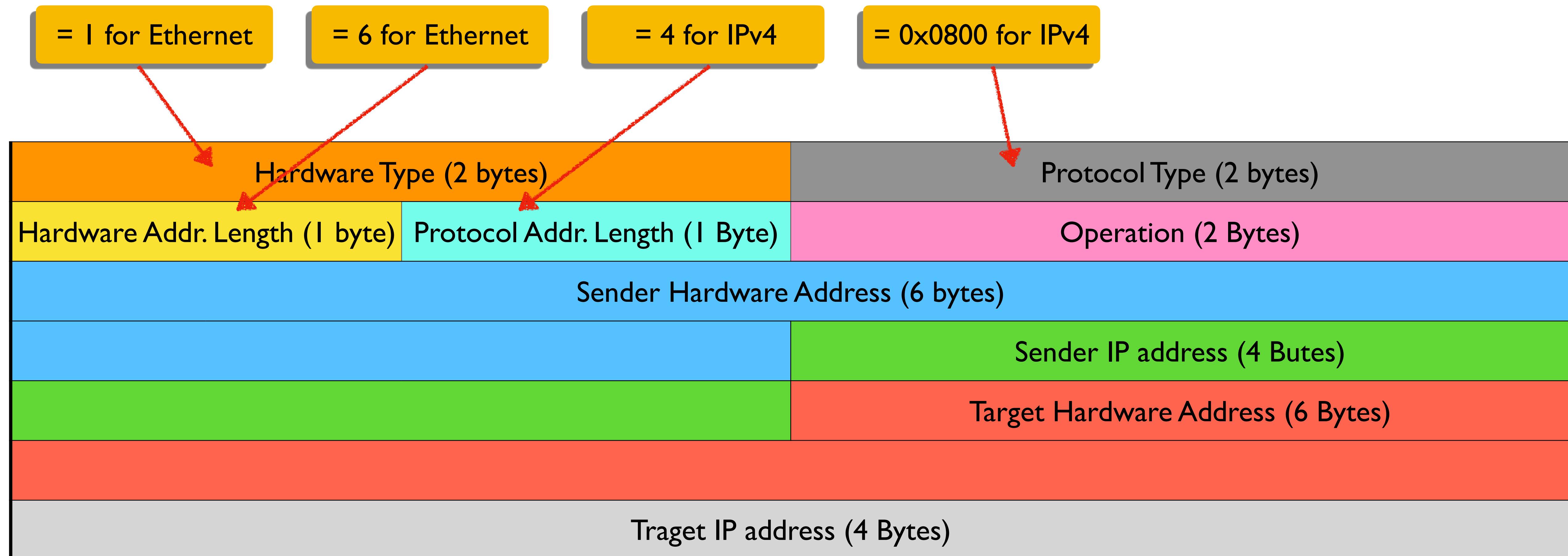
ARP Header

Hardware Type (2 bytes)	Protocol Type (2 bytes)
Hardware Addr. Length (1 byte)	Protocol Addr. Length (1 Byte)
Sender Hardware Address (6 bytes)	
	Sender IP address (4 Bytes)
	Target Hardware Address (6 Bytes)
Traget IP address (4 Bytes)	

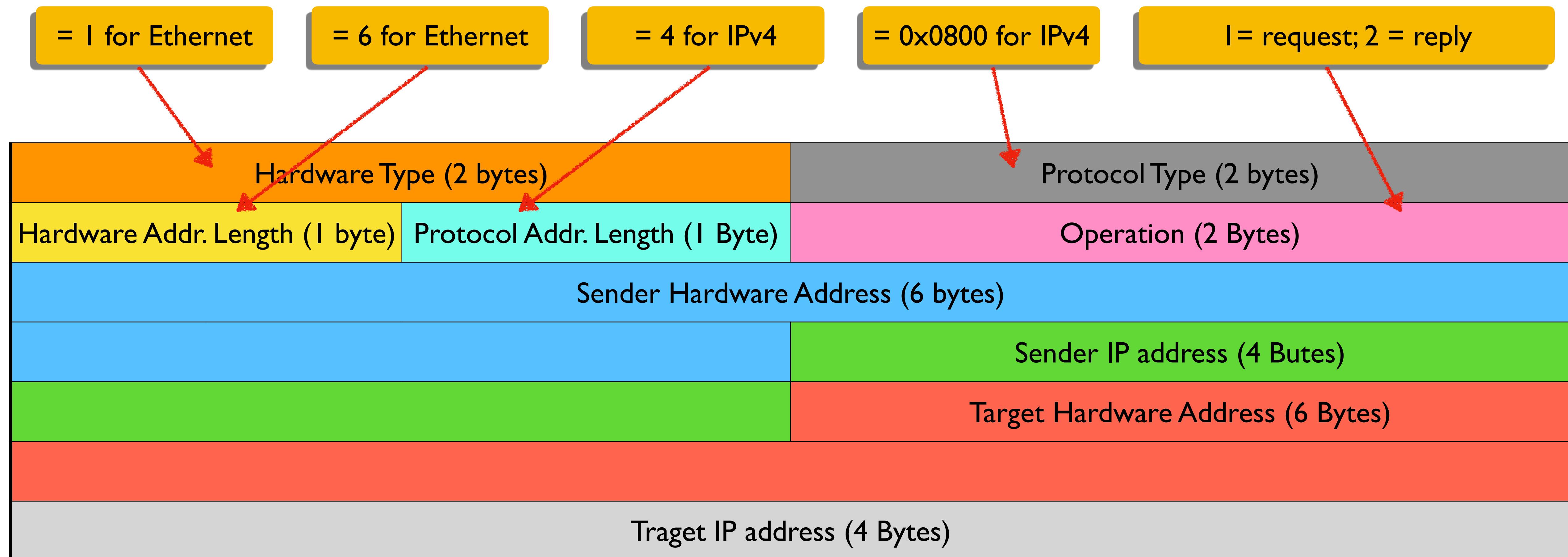
ARP Header



ARP Header



ARP Header



Key Ideas in ARP & DHCP

Key Ideas in ARP & DHCP

- Broadcasting: Can use broadcast to make contact
 - Scalable because of limited size

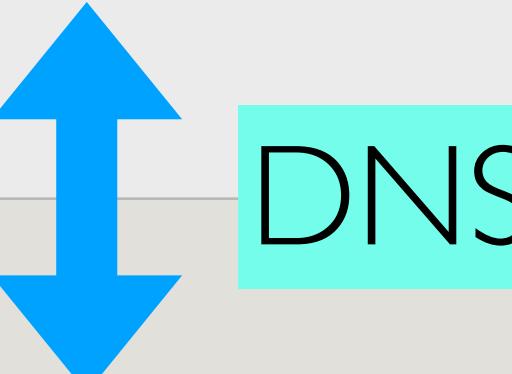
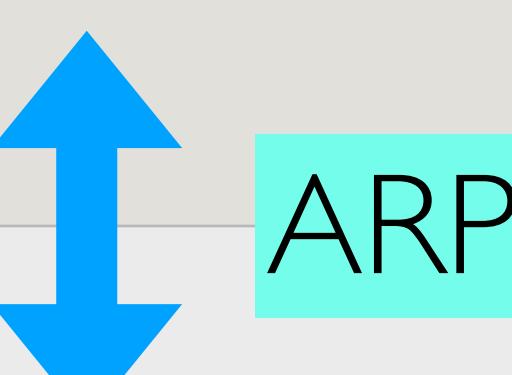
Key Ideas in ARP & DHCP

- **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

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	<u>cpsc.yale.edu</u>	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

Discovery Mechanisms

We've seen two approaches:

- Broadcast (ARP, DHCP)
- Directory service (DNS)

Discovery Mechanisms

We've seen two approaches:

- Broadcast (ARP, DHCP)
 - Flooding doesn't scale
- Directory service (DNS)
 - No flooding / scalable

Discovery Mechanisms

We've seen two approaches:

- Broadcast (ARP, DHCP)
 - Flooding doesn't scale
 - No centralized point of failure
- Directory service (DNS)
 - No flooding / scalable
 - Root of the directory is vulnerable (caching is key)

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.)

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

Steps in reaching a destination

- First lookup destination's IP address

Steps in reaching a destination

- First lookup destination's IP address
 - DNS

Steps in reaching a destination

- First lookup destination's IP address
 - DNS
- Need to know where my local DNS server is

Steps in reaching a destination

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

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

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

Sending a Packet

- On the same subnet:

Sending a Packet

- On the same subnet:
 - Use MAC address of destination

Sending a Packet

- On the same subnet:
 - Use MAC address of destination
 - ARP

Sending a Packet

- On the same subnet:
 - Use MAC address of destination
 - ARP
- On some other subnet

Sending a Packet

- **On the same subnet:**
 - Use MAC address of destination
 - ARP
- **On some other subnet**
 - Use MAC address of first-hop router

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

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?

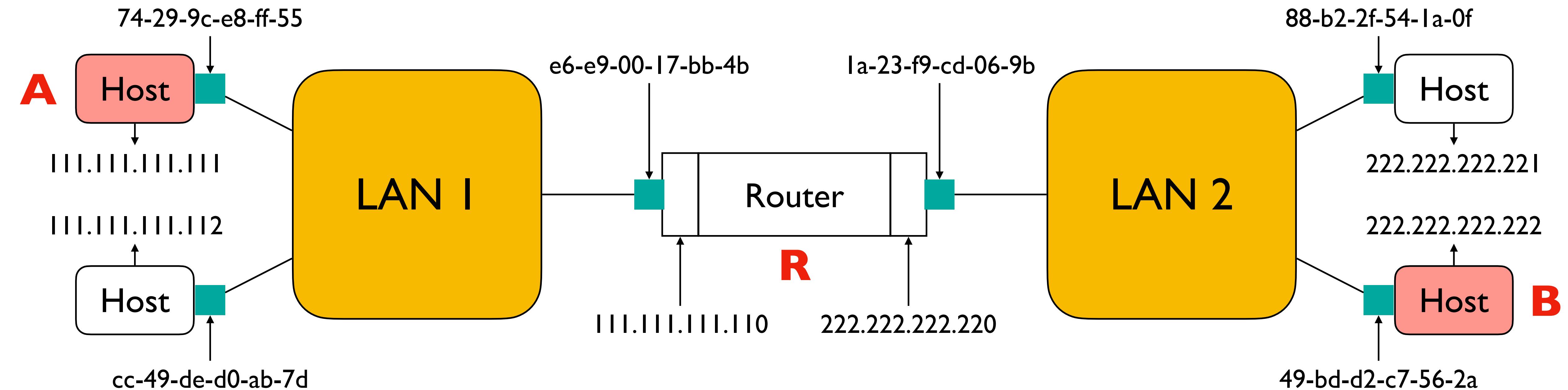
Sending a Packet

- **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

Sending a Packet

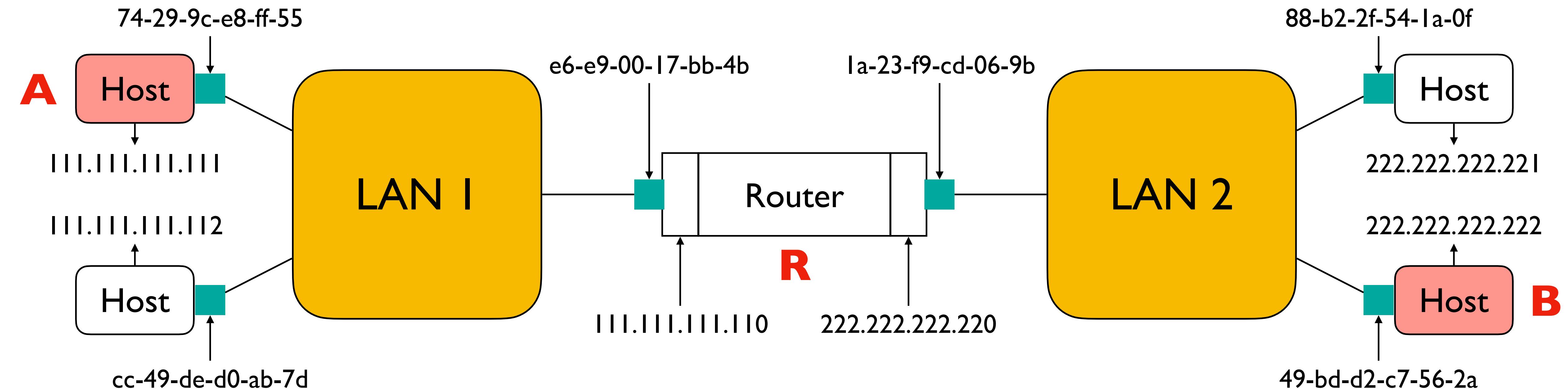
- **On the same subnet:**
 - Use MAC address of destination
 - ARP
- **On some other subnet**
 - Use MAC address of first-hop router
 - DHCP + ARP
- **And how can a host tell whether destination is on same or other subnet?**
 - Use the netmask
 - DHCP

Example: **A** sending a packet to **B**



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

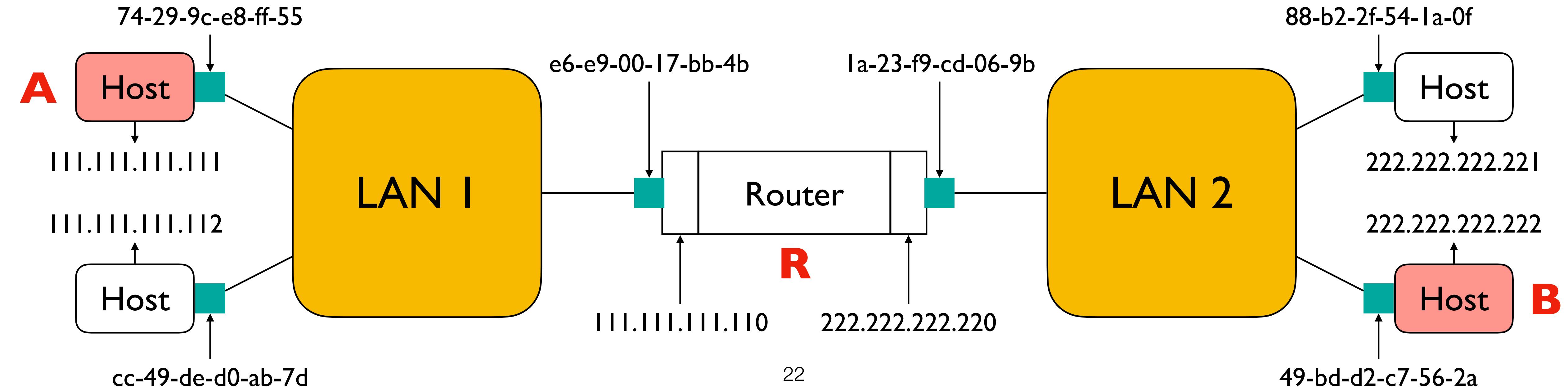
Example: **A** sending a packet to **B**



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

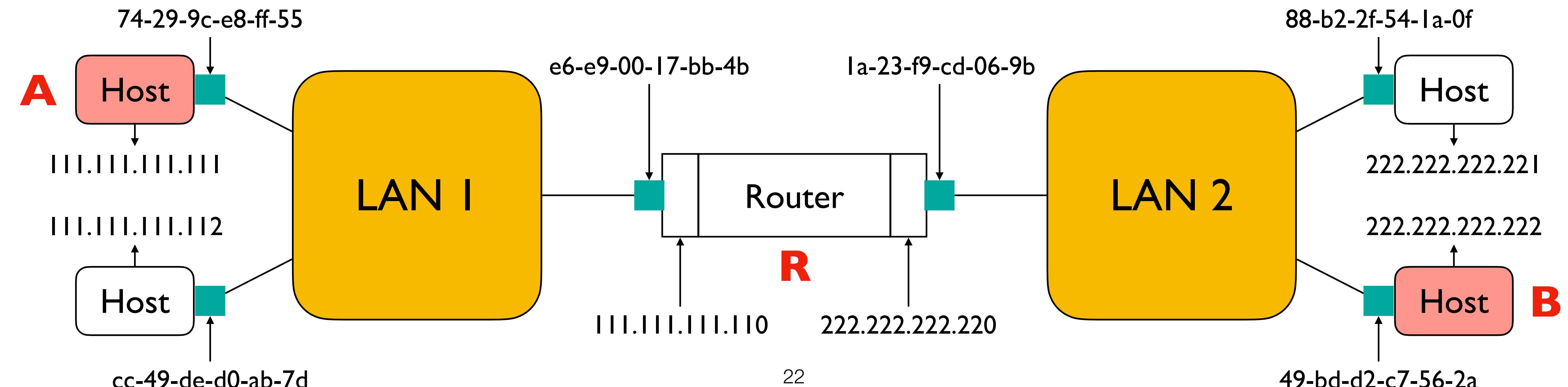
A sends packet through router R

- Host A constructs an IP packet to send to B
 - Source 111.111.111.111, destination 222.222.222.222



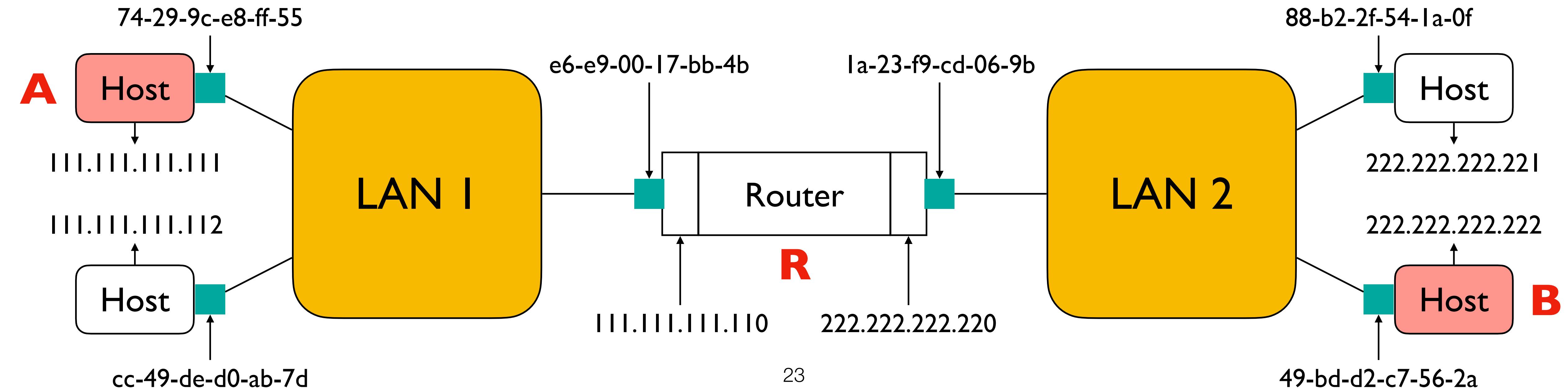
A sends packet through router R

- 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 DHCP



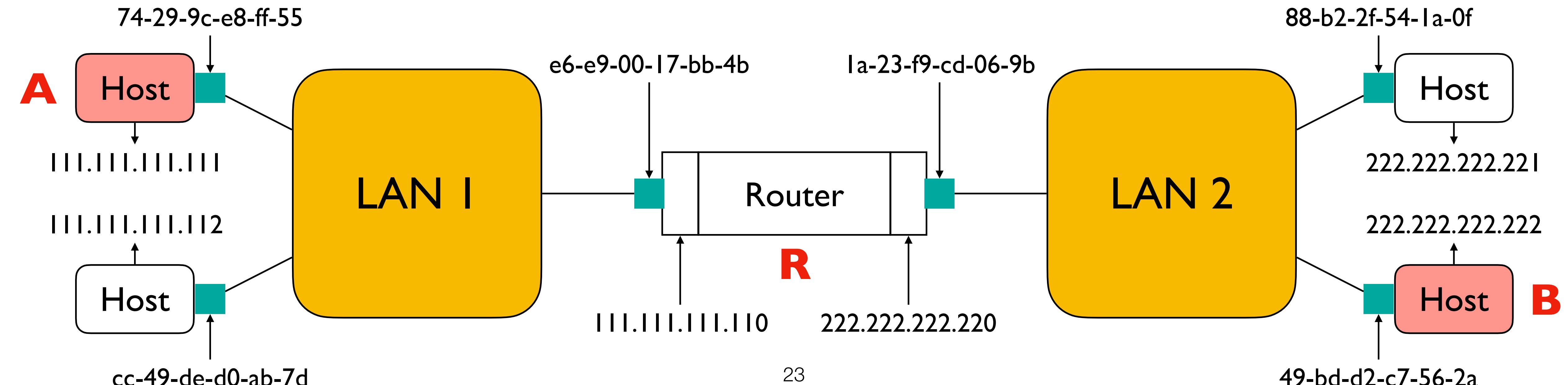
A sends packet through router **R**

- 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



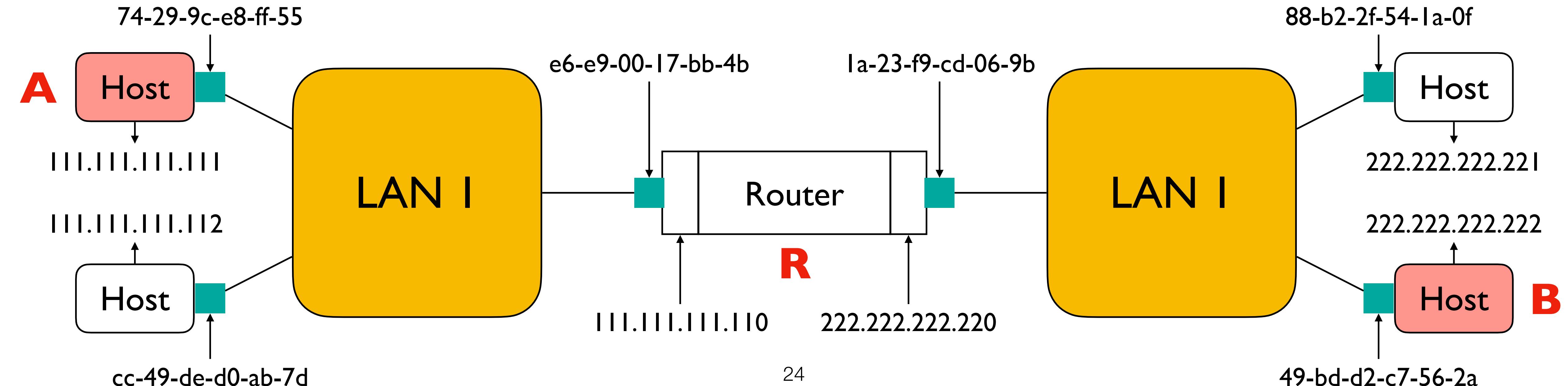
A sends packet through router R

- 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



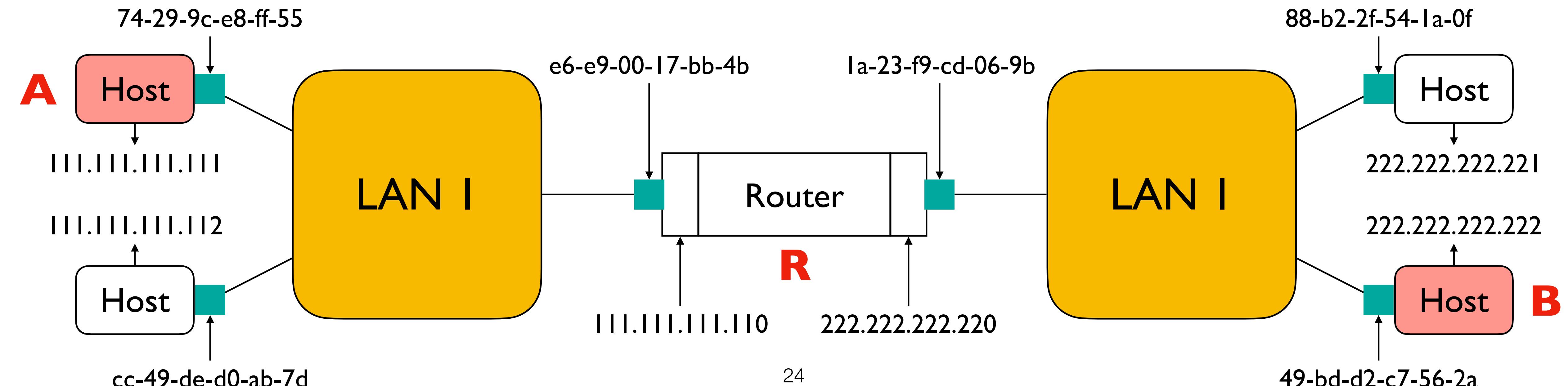
R decides how to forward packet

- Router R's adapter receives the packet
 - R extracts the IP packet from the Ethernet frame
 - R sees the IP packet is destined to 222.222.222.222



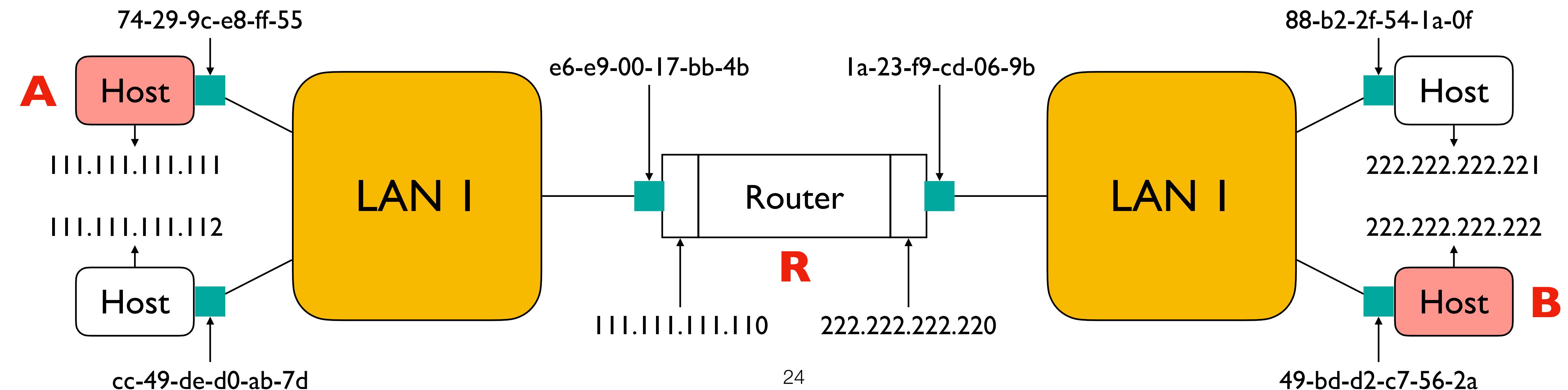
R decides how to forward packet

- Router R's adapter receives the packet
 - R extracts the IP packet from the Ethernet frame
 - R sees the IP packet is destined to 222.222.222.222
- Router R consults its forwarding table
 - Packet matches 222.222.222.0/24 via other adapter (port)



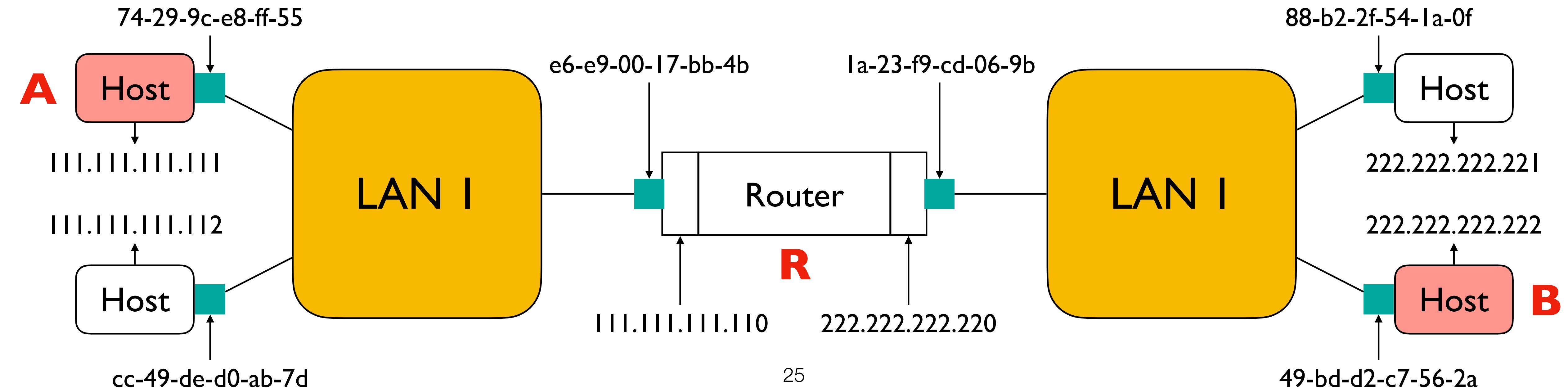
R decides how to forward packet

- 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)



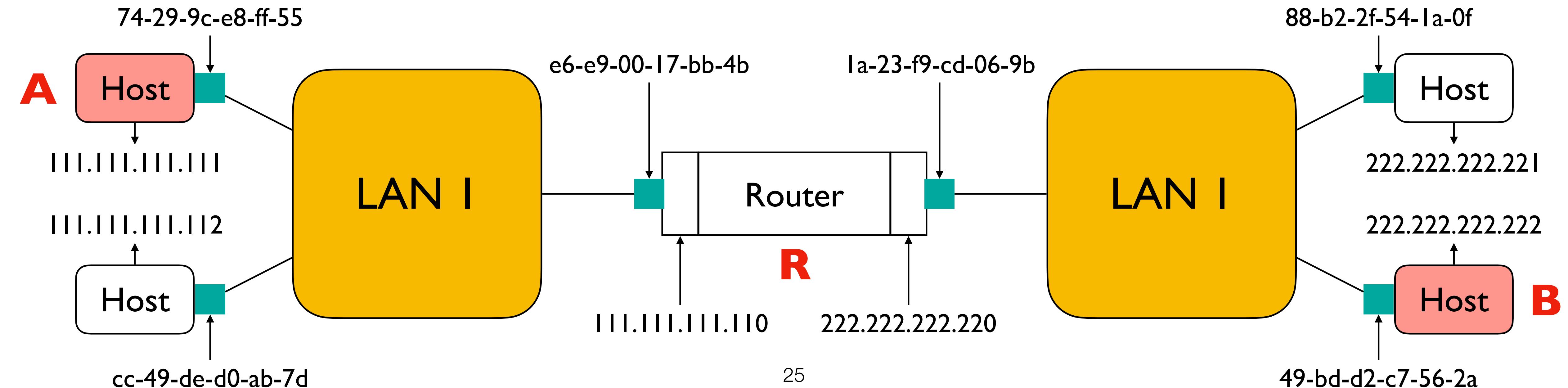
R sends packet to B

- 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



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?

Are we there yet?

Yes!

Putting the pieces together

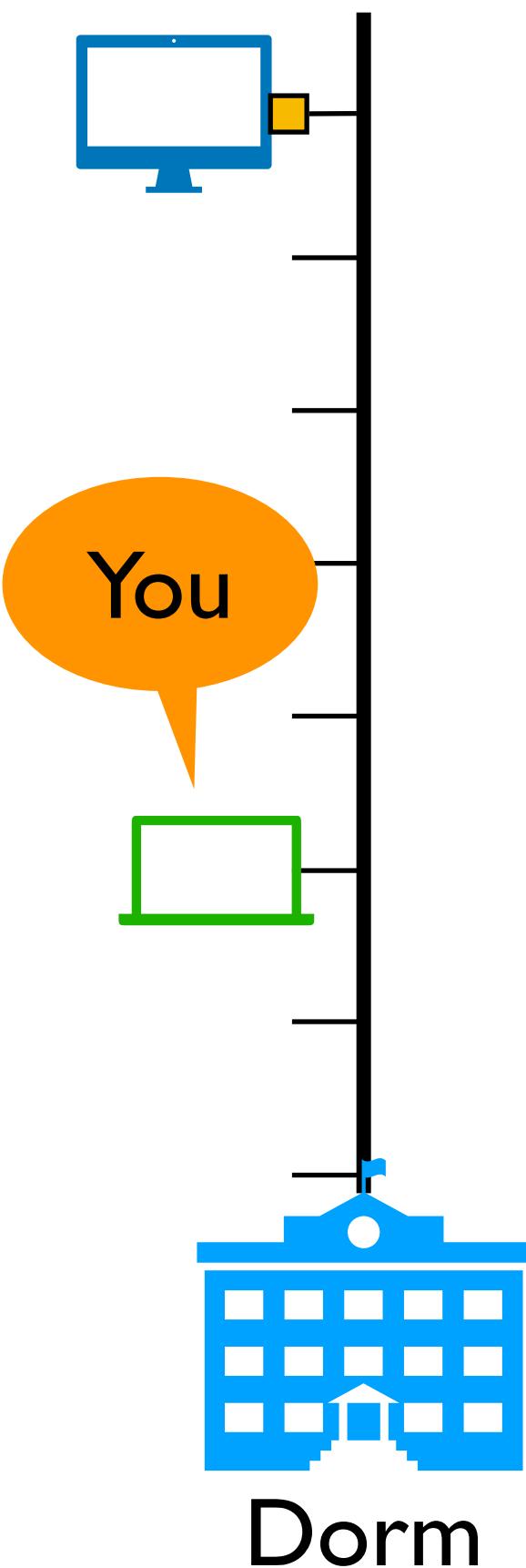
Putting the pieces together

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

Putting the pieces together

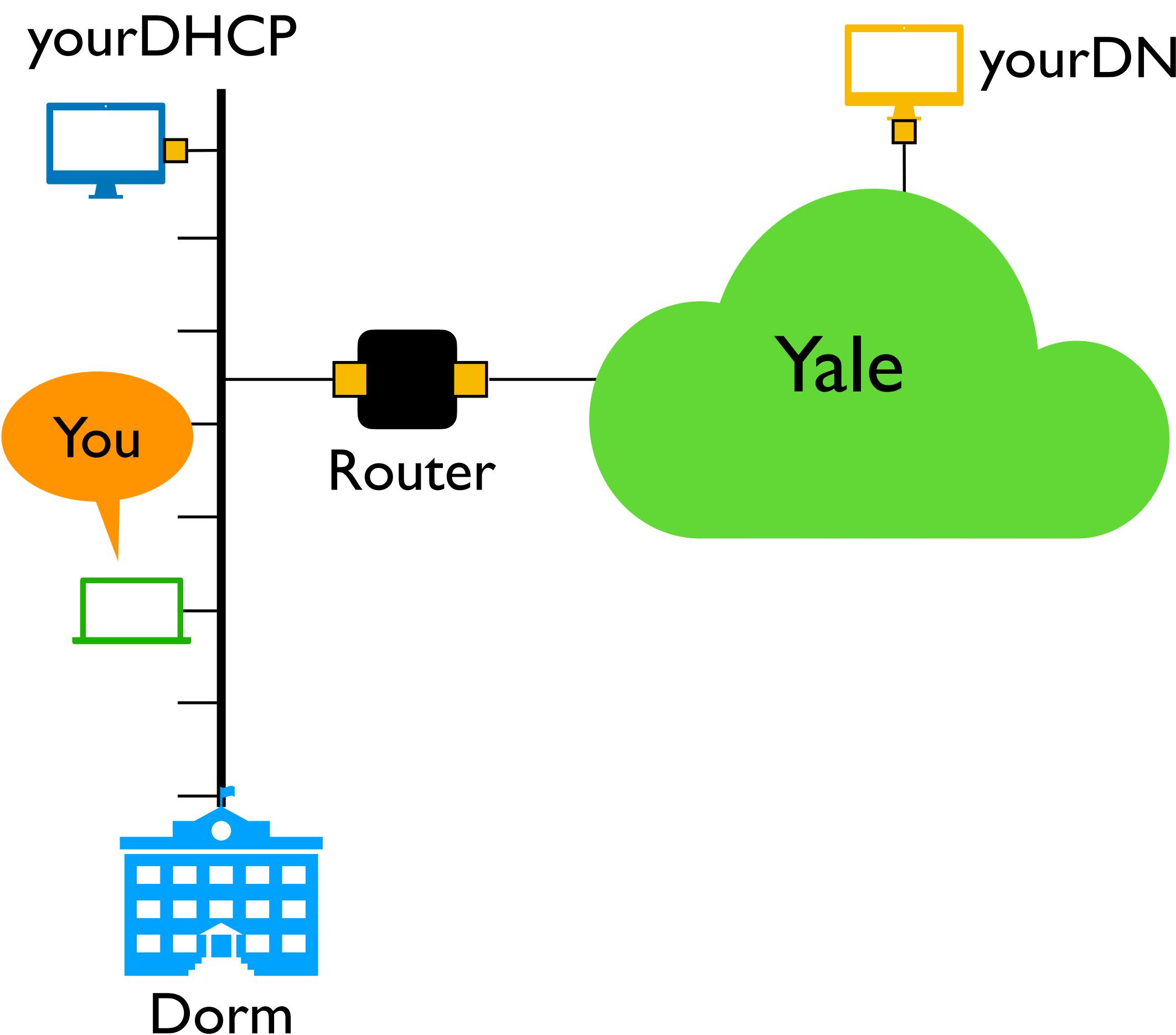
Walk through the steps required to download
www.google.com/index.html from your laptop

yourDHCP



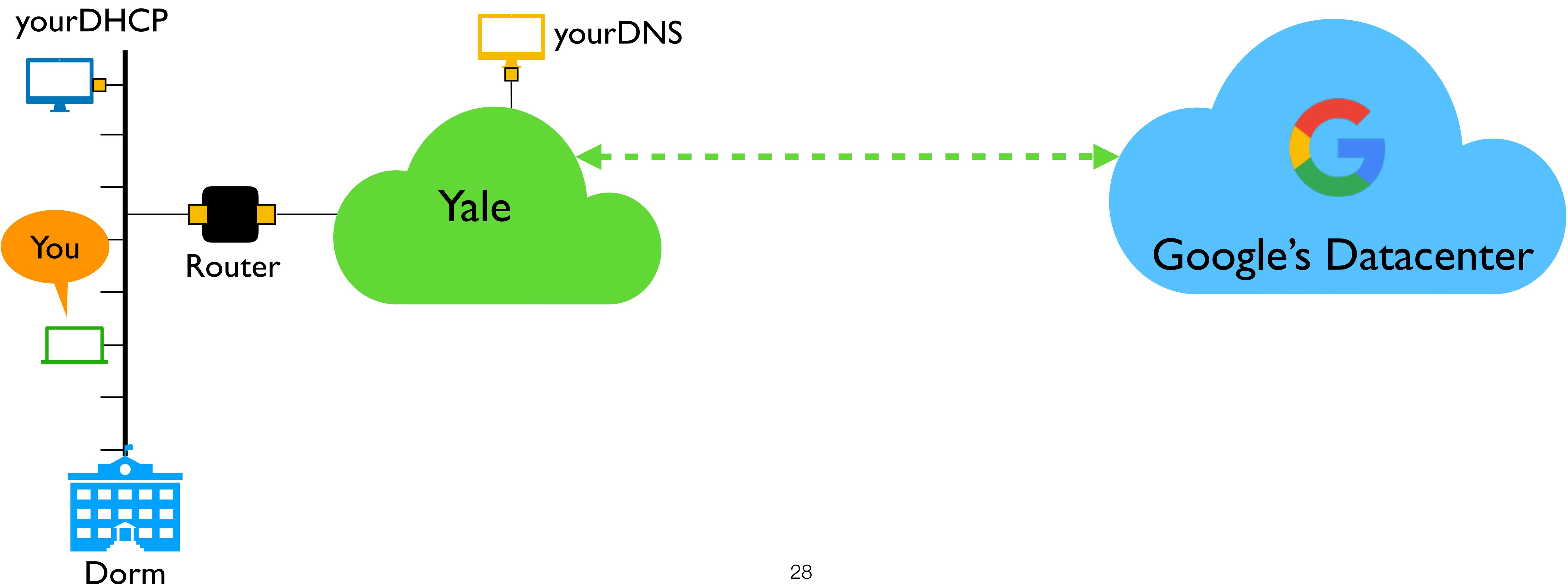
Putting the pieces together

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



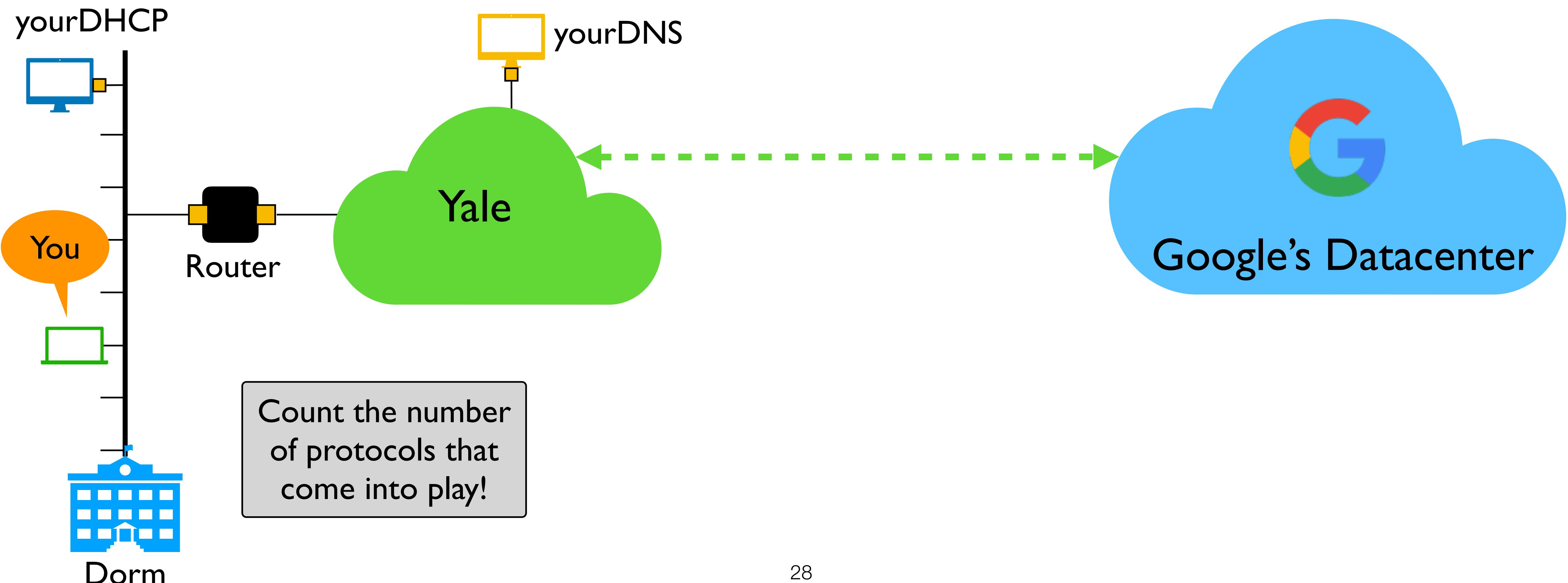
Putting the pieces together

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



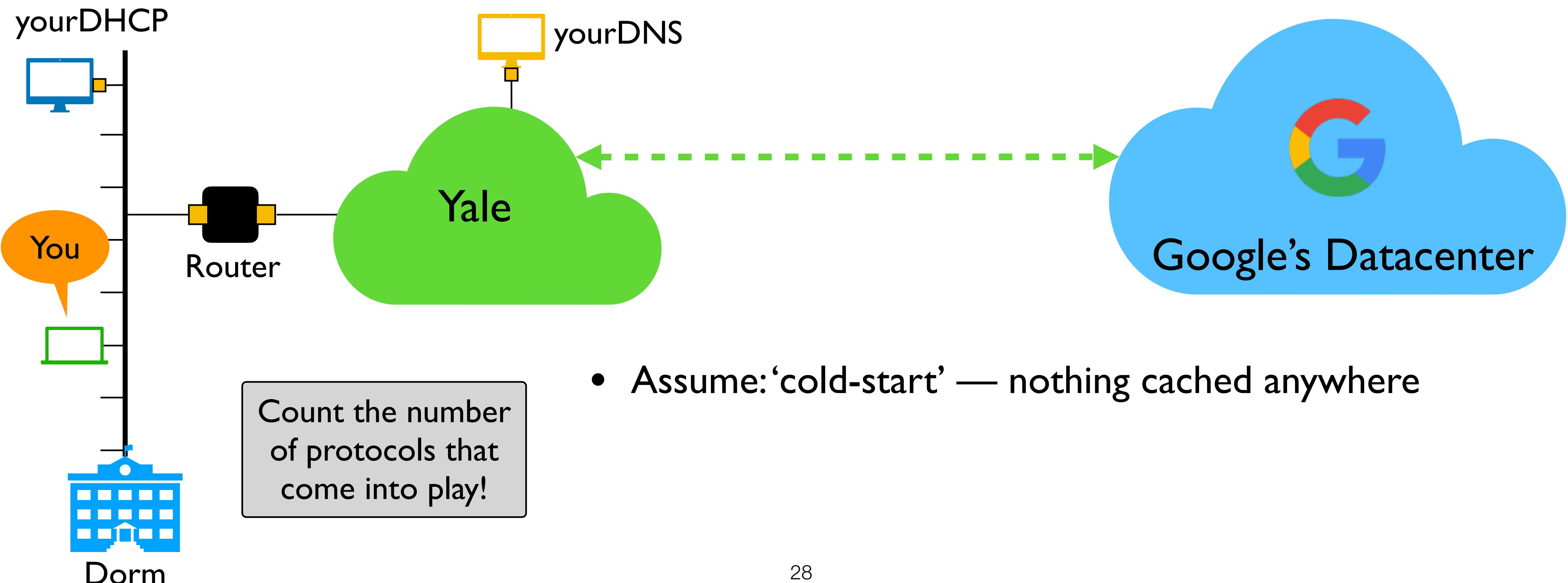
Putting the pieces together

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



Putting the pieces together

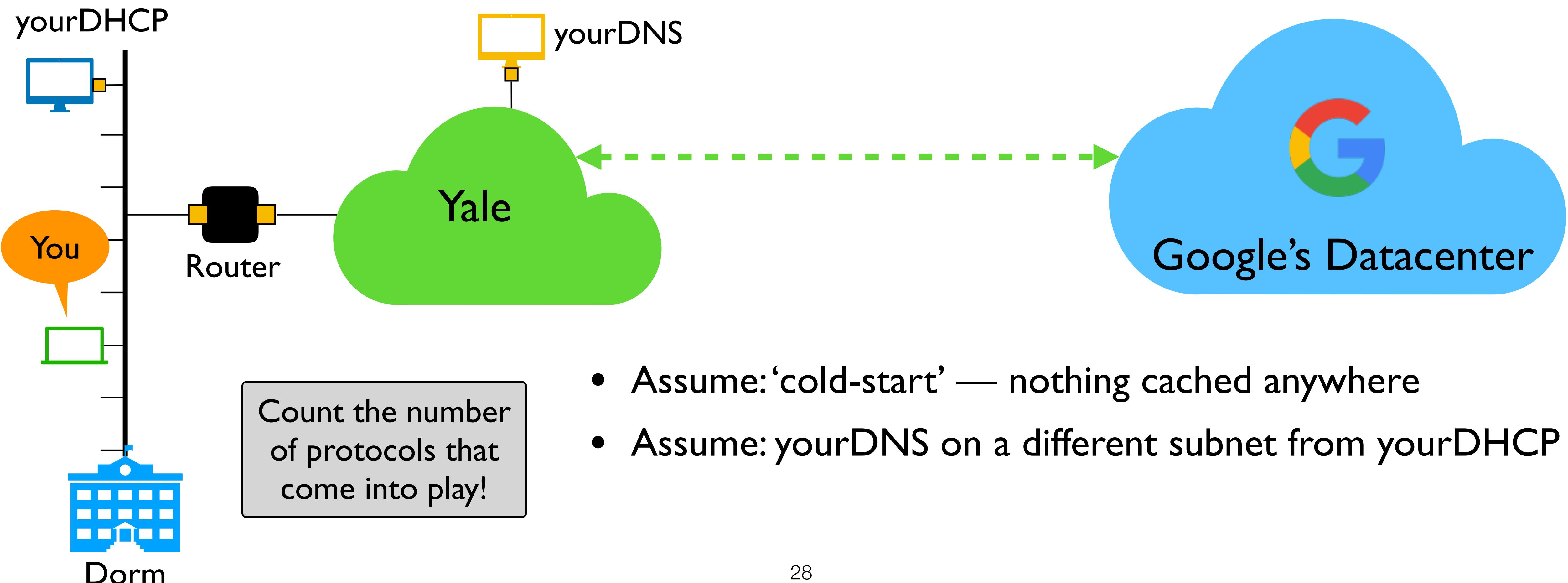
Walk through the steps required to download
www.google.com/index.html from your laptop



- Assume: 'cold-start' — nothing cached anywhere

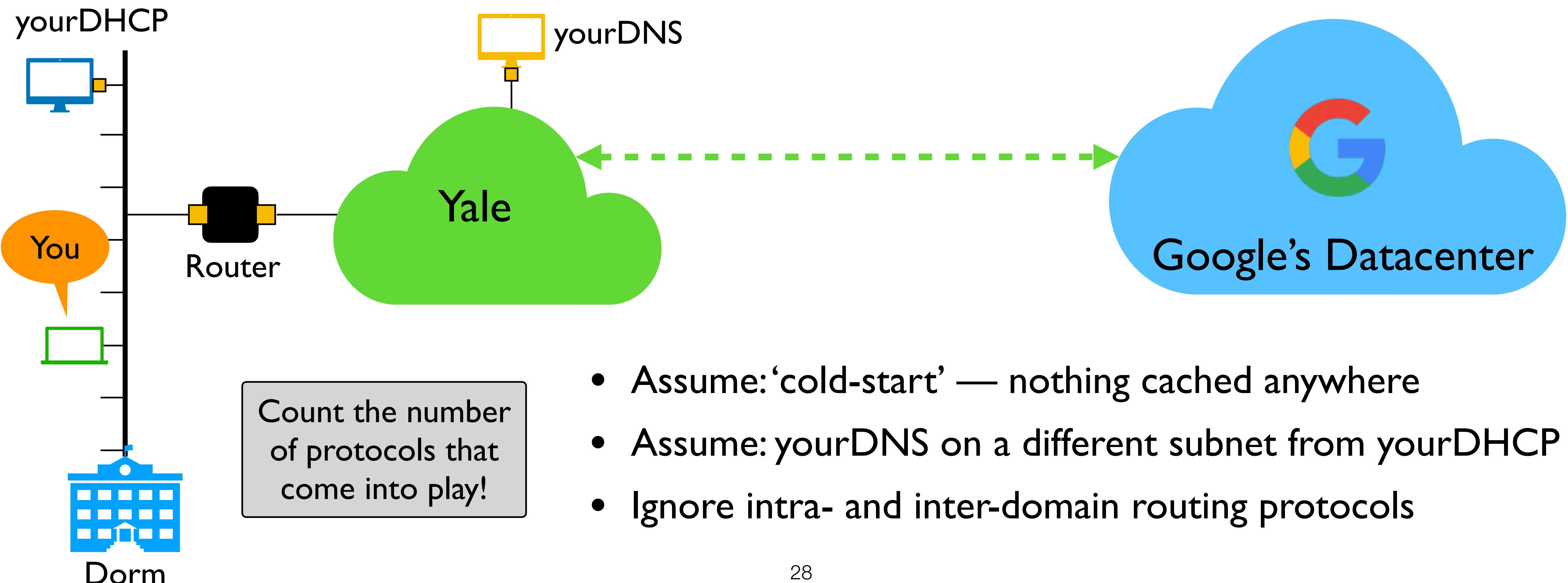
Putting the pieces together

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

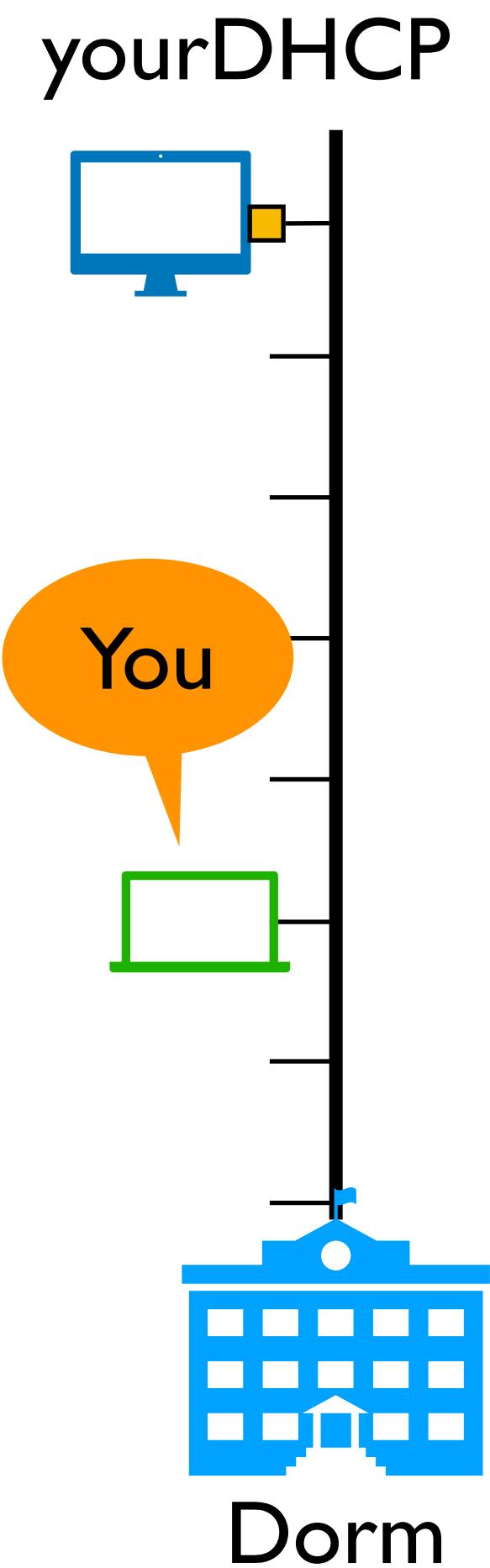


Putting the pieces together

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

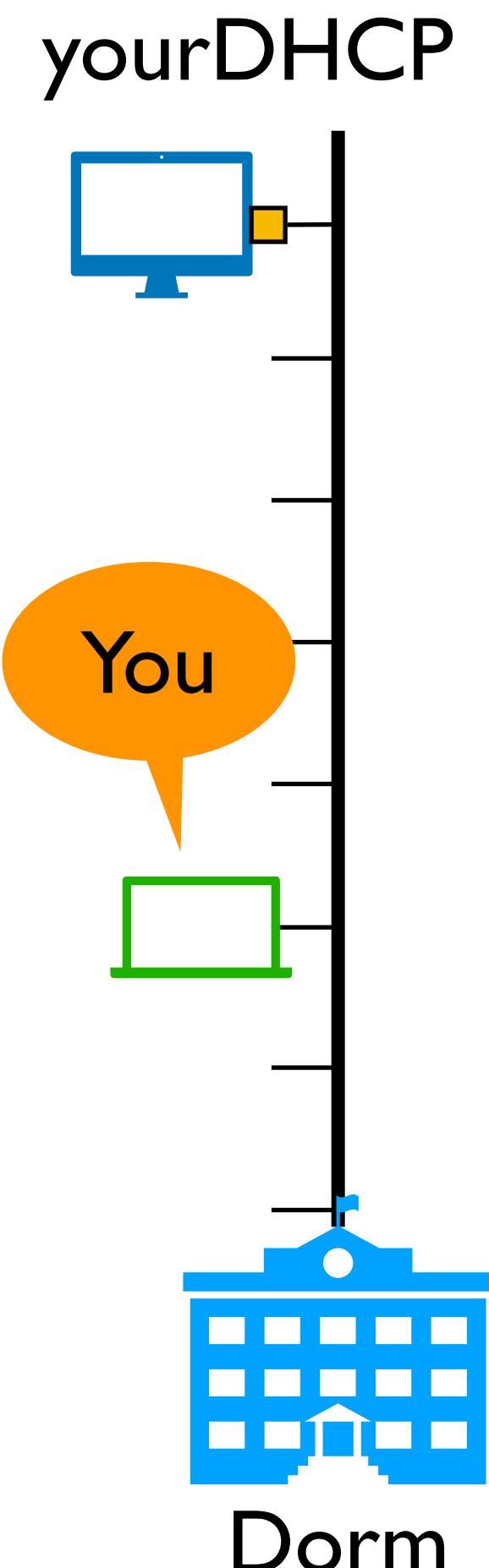


Step I: Self Discovery



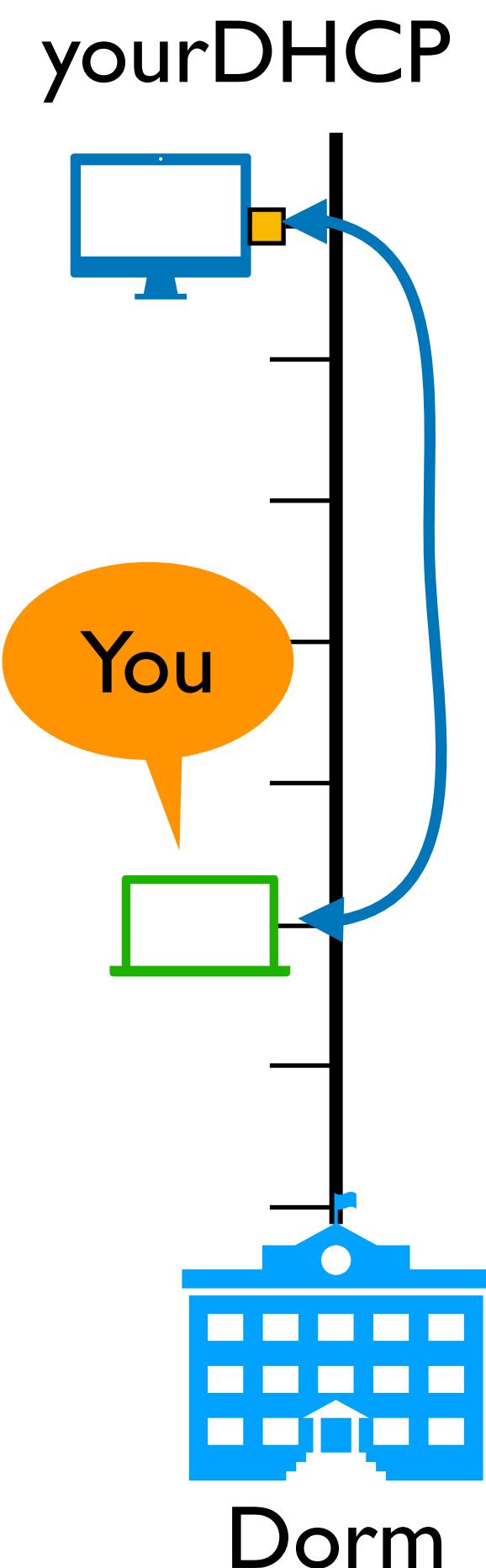
Step I: Self Discovery

- 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)
 - ...



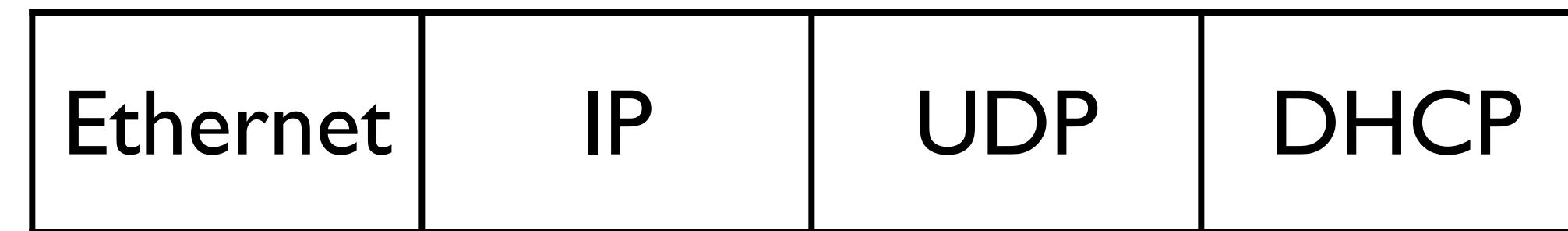
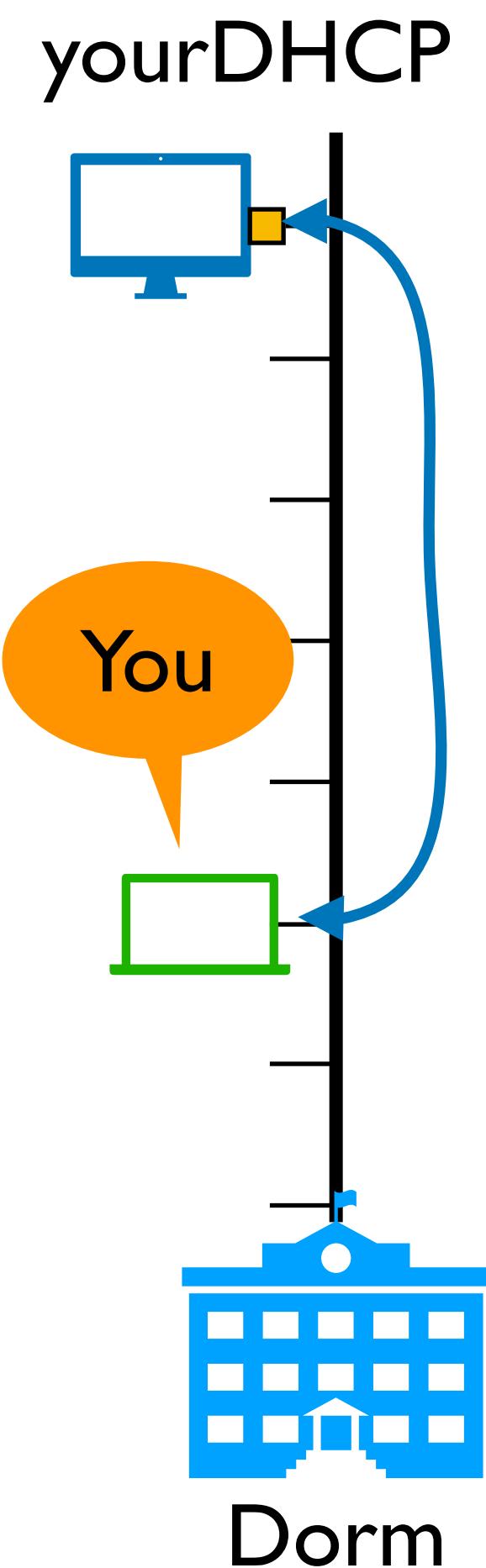
Step I: Self Discovery

- 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



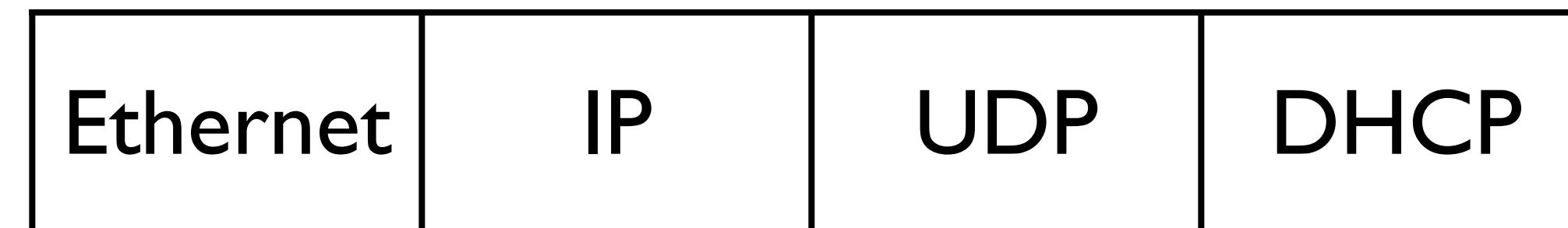
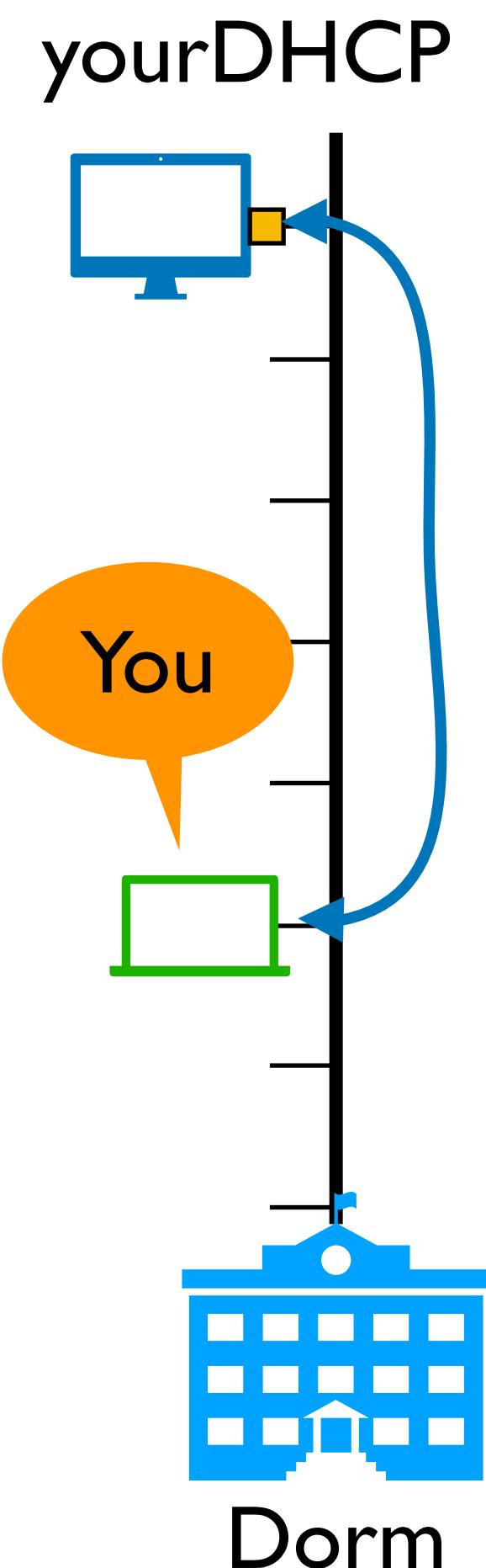
Step I: Self Discovery

- 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



Step I: 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



- Protocol count = 4

Next...

Next...

- You are ready to contact www.google.com

Next...

- You are ready to contact www.google.com
 - Need an IP address for www.google.com

Next...

- You are ready to contact www.google.com
 - Need an IP address for www.google.com
 - Need to ask Google's DNS server

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

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 your DNS server to ask Google's DNS
 - I know my DNS server's IP address is u.dns.ip.addr

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 your DNS server to ask Google's DNS
 - I know my DNS server's IP address is u.dns.ip.addr
 - Create a packet to send...

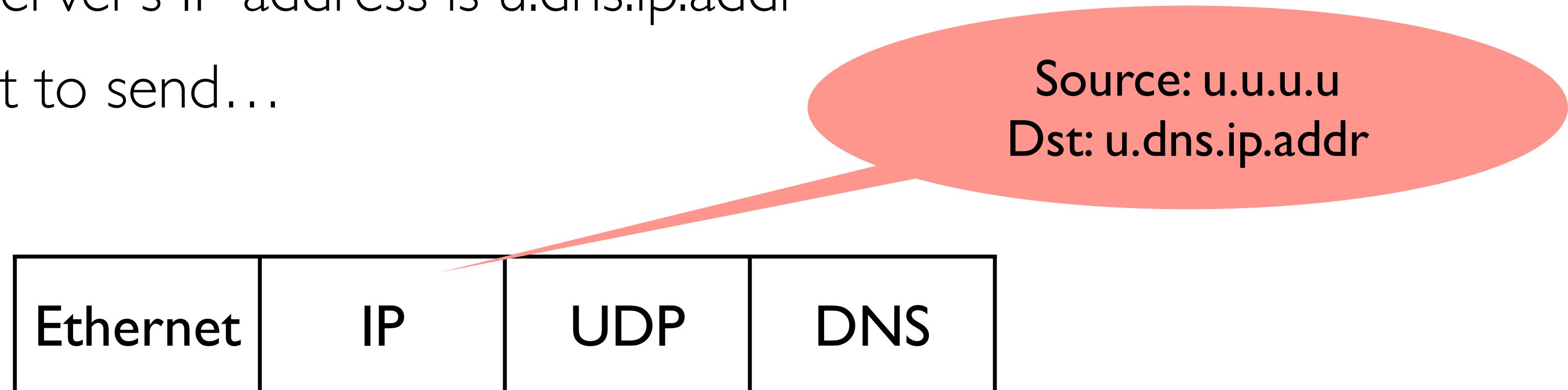
Next...

- 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 your DNS server to ask Google's DNS
 - I know my DNS server's IP address is u.dns.ip.addr
 - Create a packet to send...



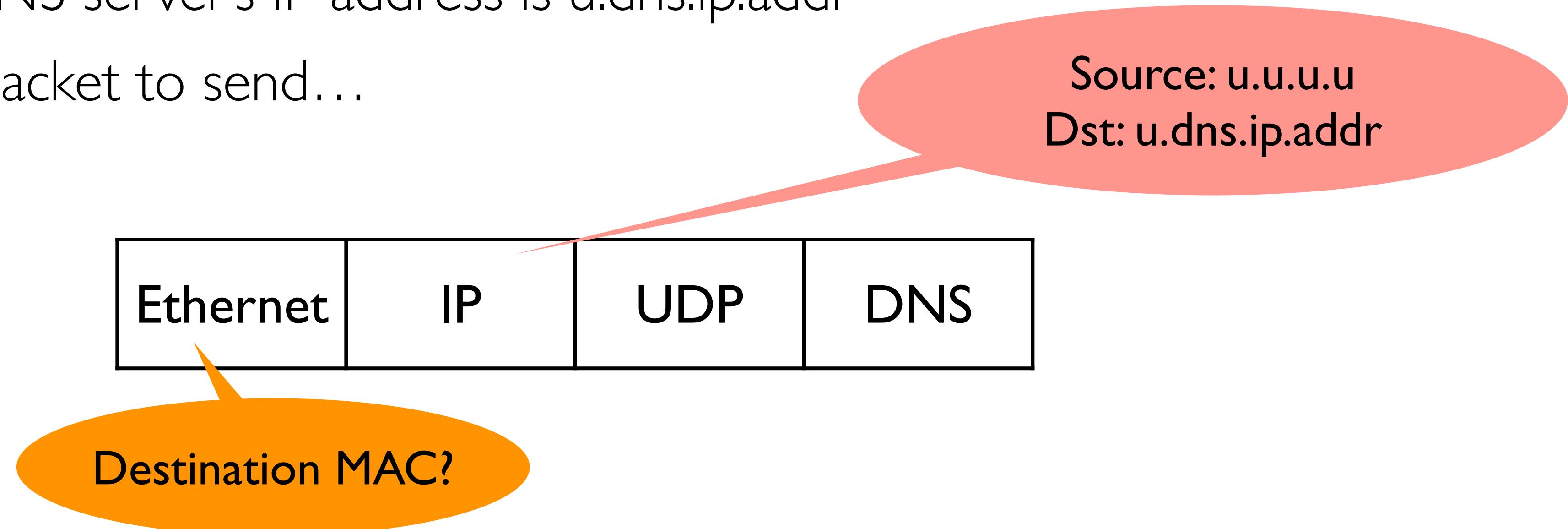
Next...

- 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 your DNS server to ask Google's DNS
 - I know my DNS server's IP address is u.dns.ip.addr
 - Create a packet to send...

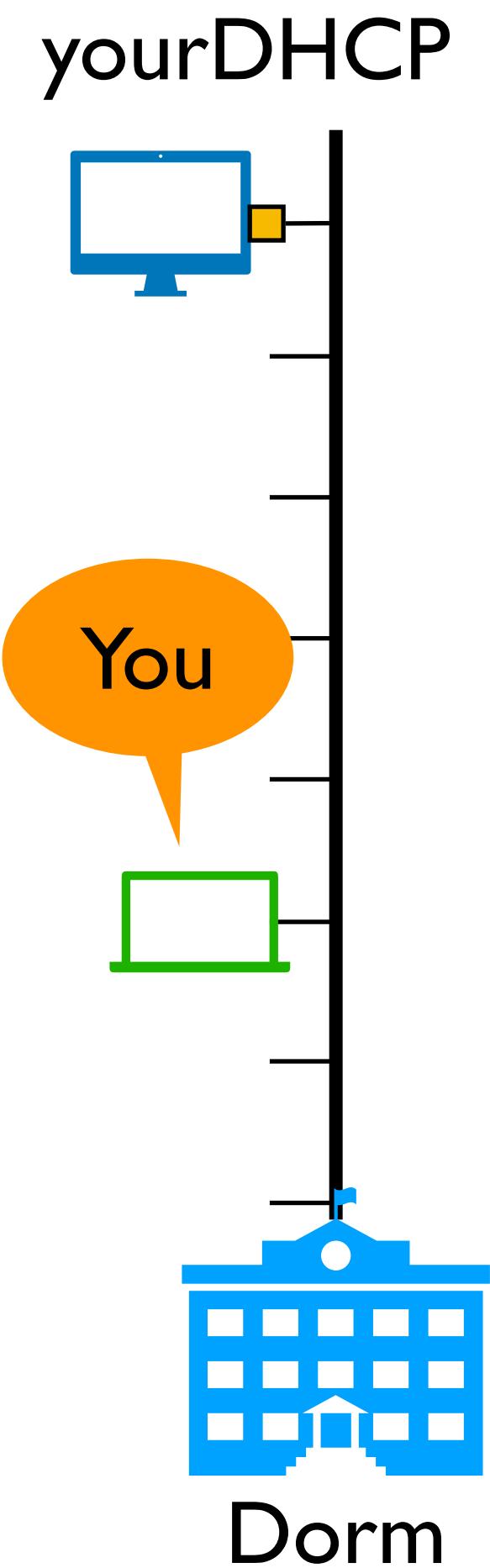


Next...

- 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 your DNS server to ask Google's DNS
 - I know my DNS server's IP address is u.dns.ip.addr
 - Create a packet to send...

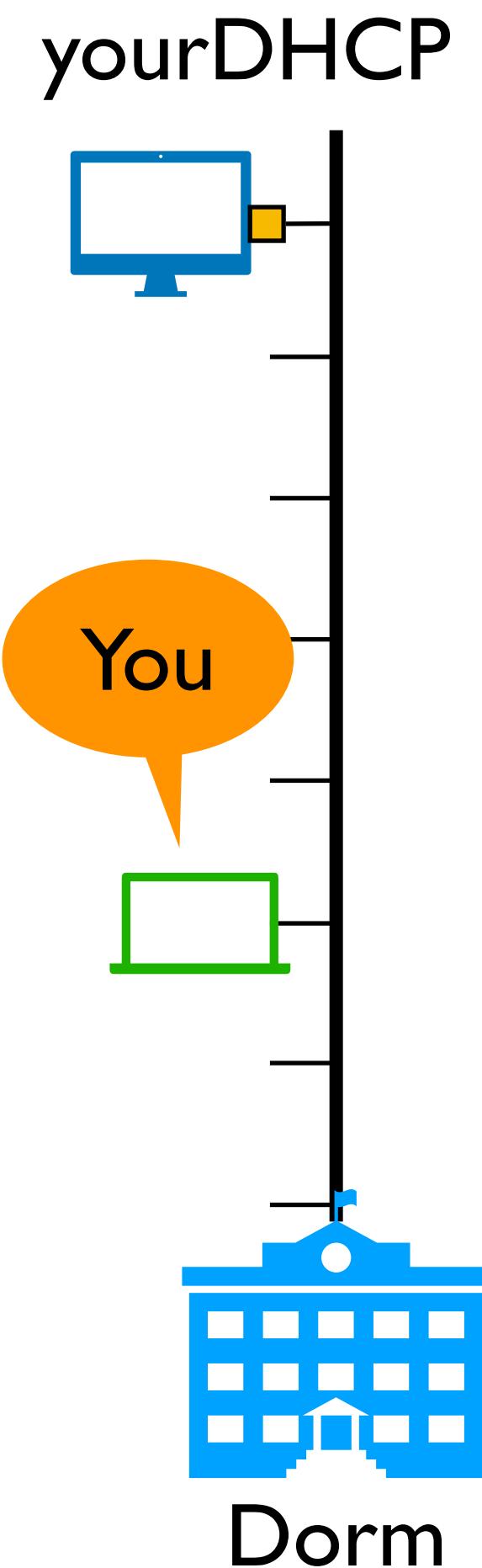


Step 2: Getting out the door



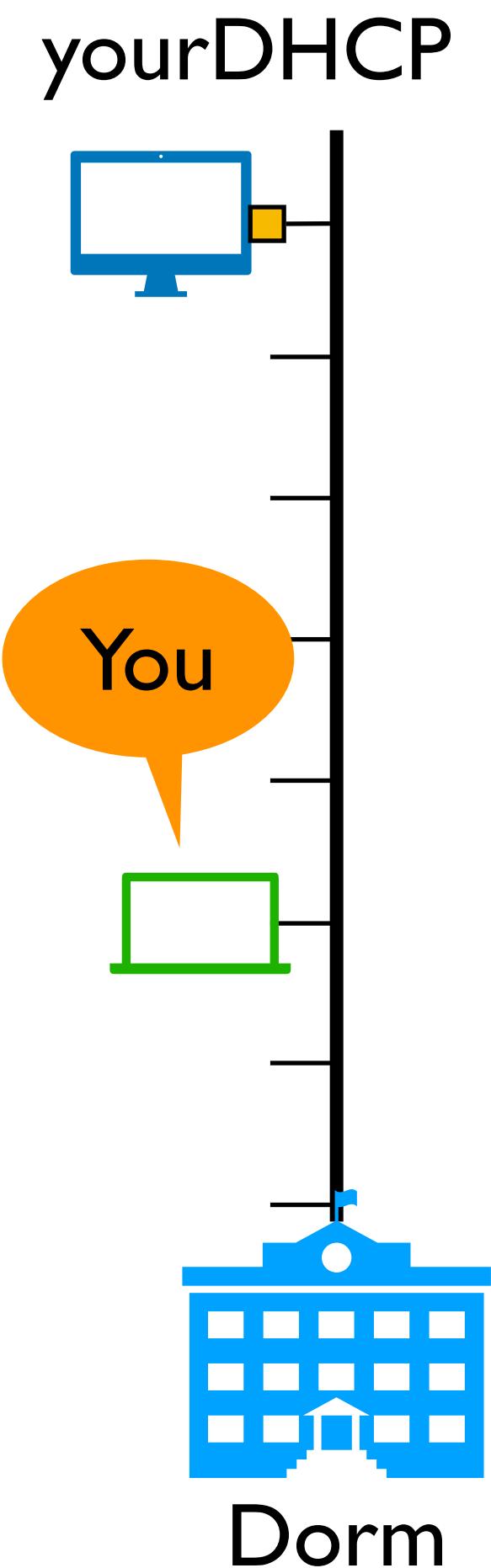
Step 2: Getting out the door

- You use ARP to discover the MAC address of R



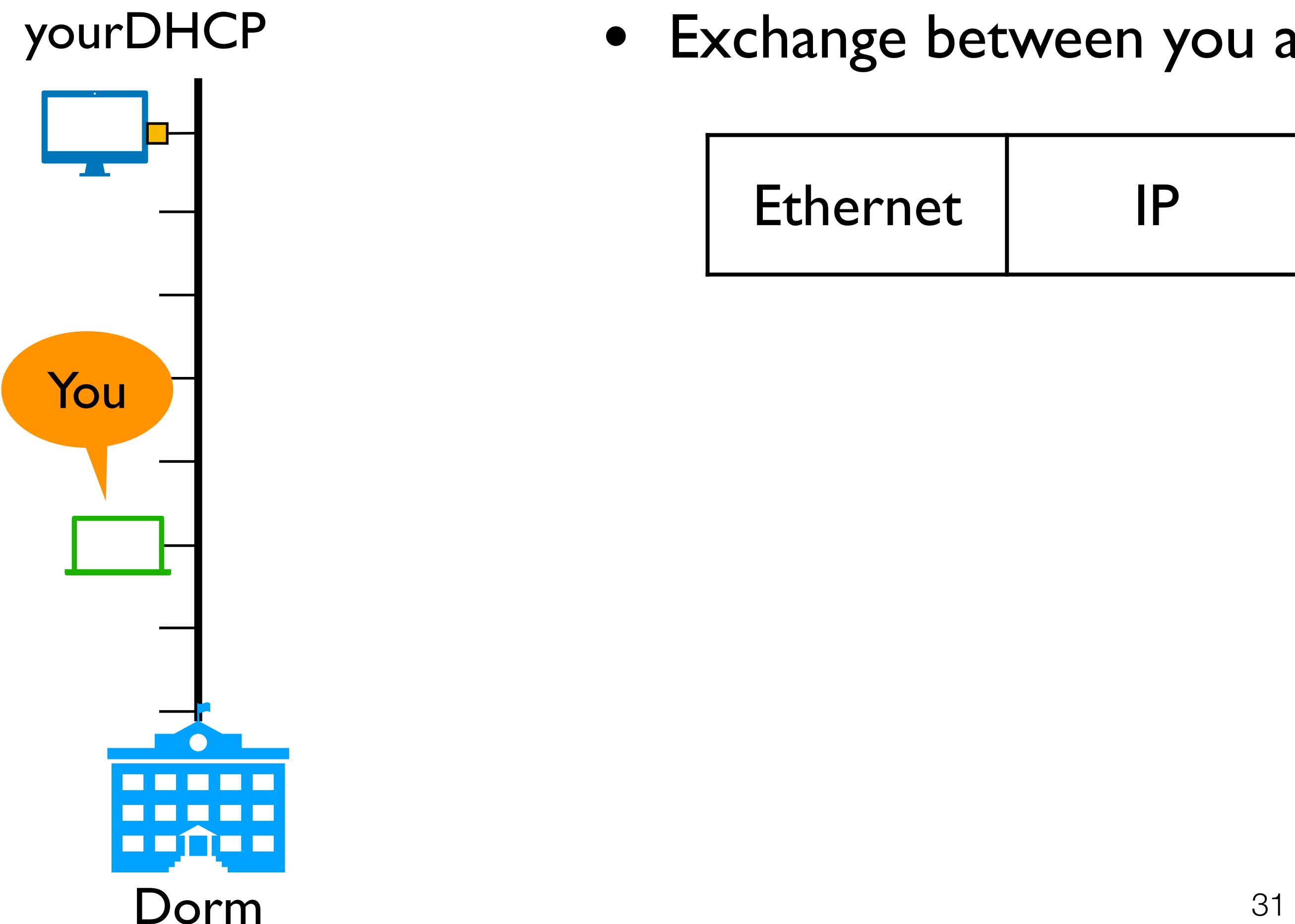
Step 2: Getting out the door

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



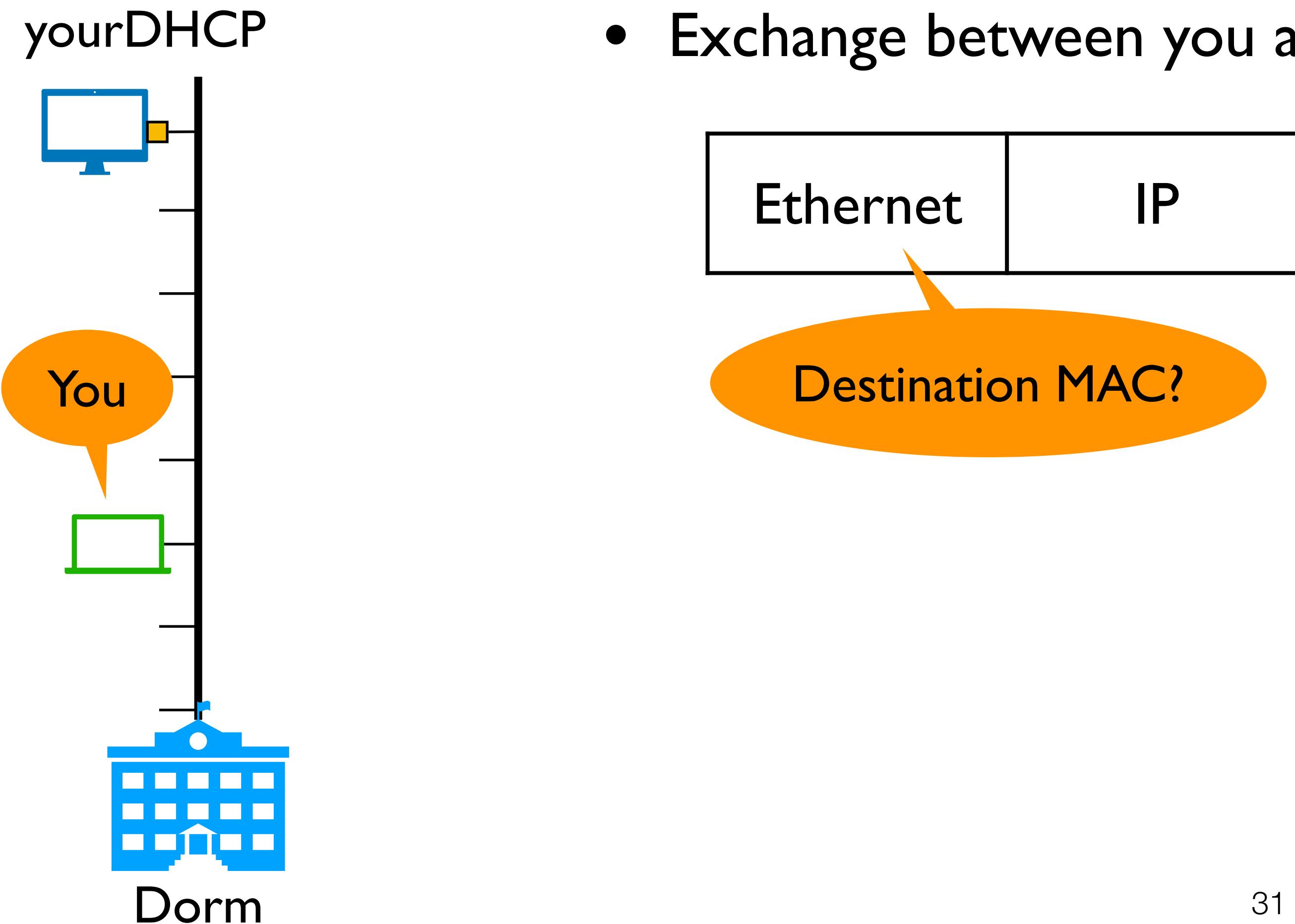
Step 2: Getting out the door

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



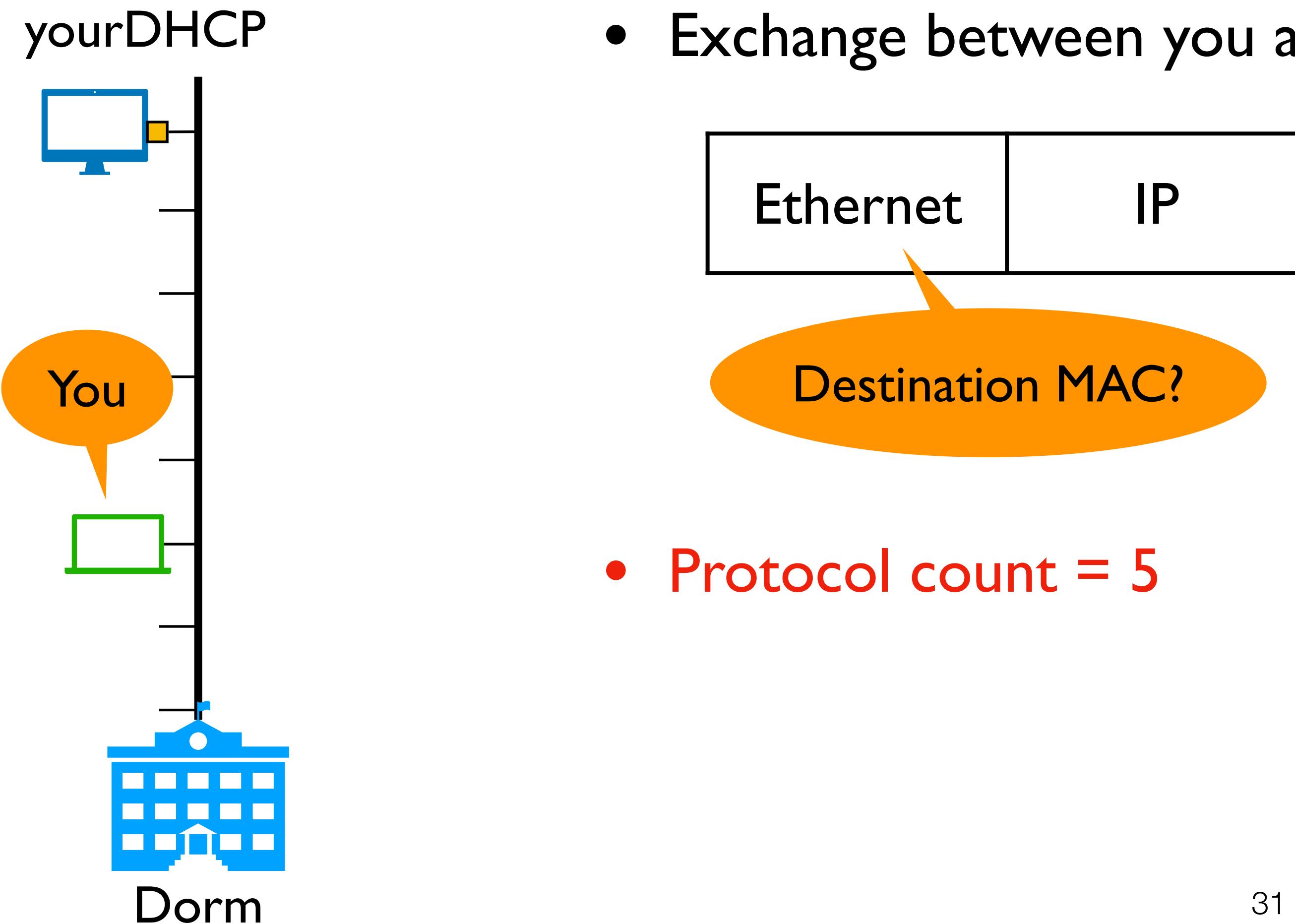
Step 2: Getting out the door

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



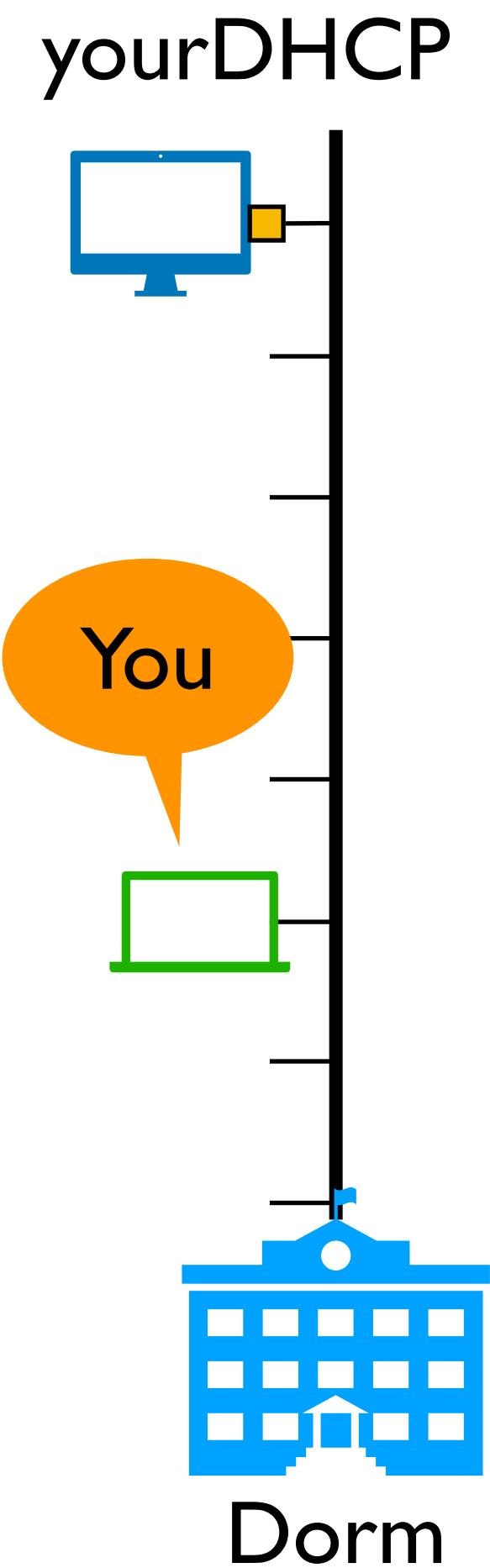
Step 2: Getting out the door

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

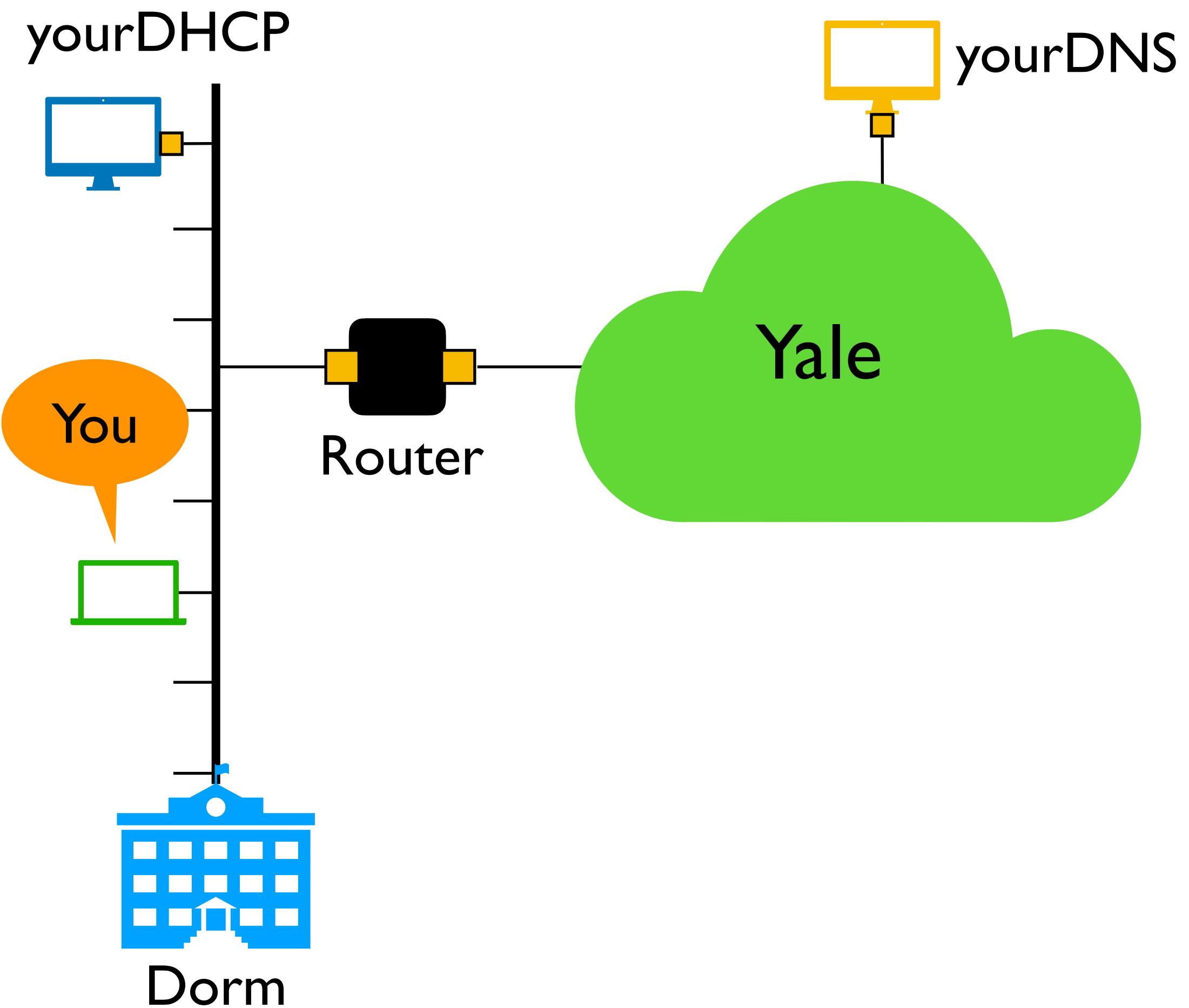


- Protocol count = 5

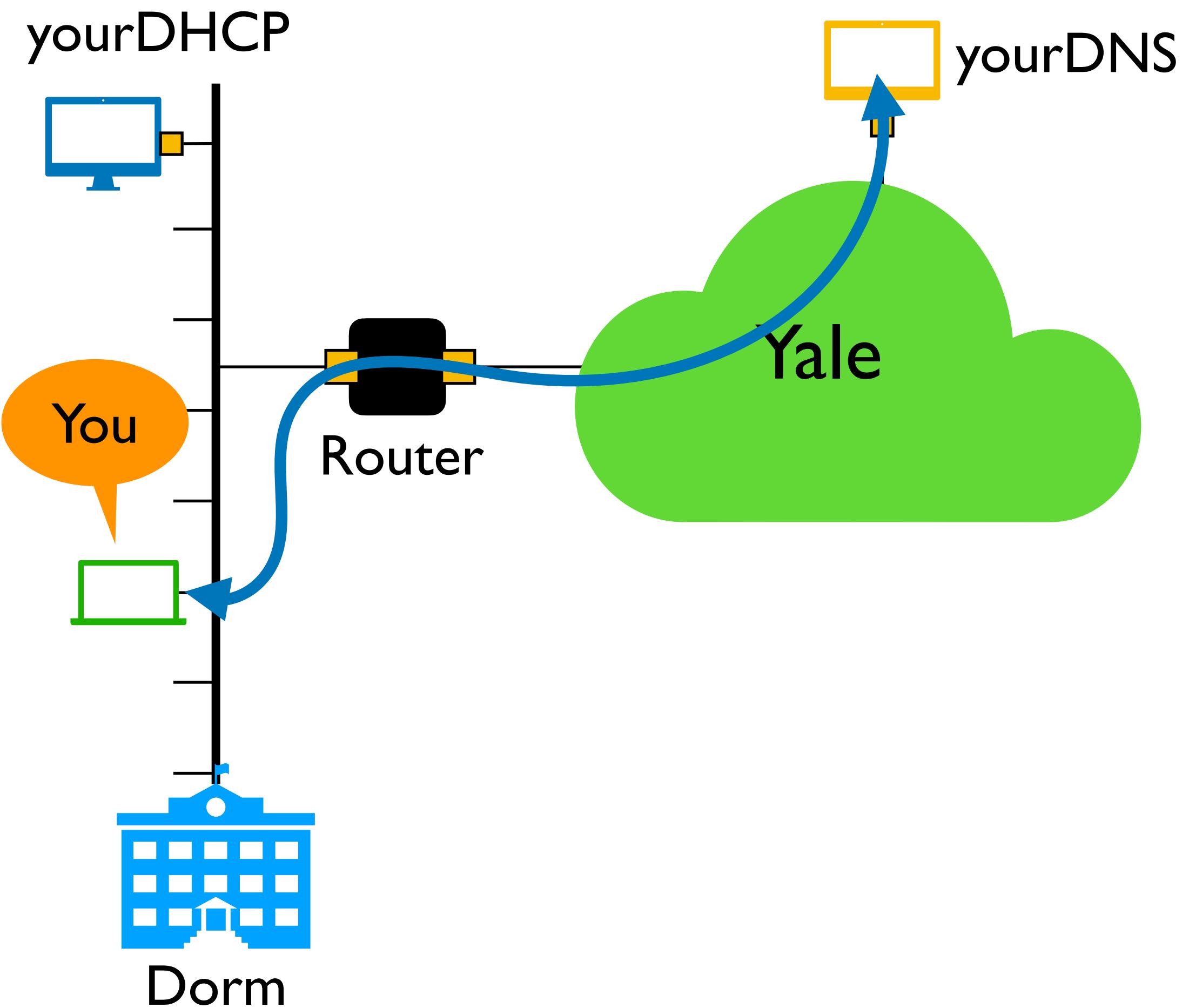
Step 3: Send a DNS request



Step 3: Send a DNS request

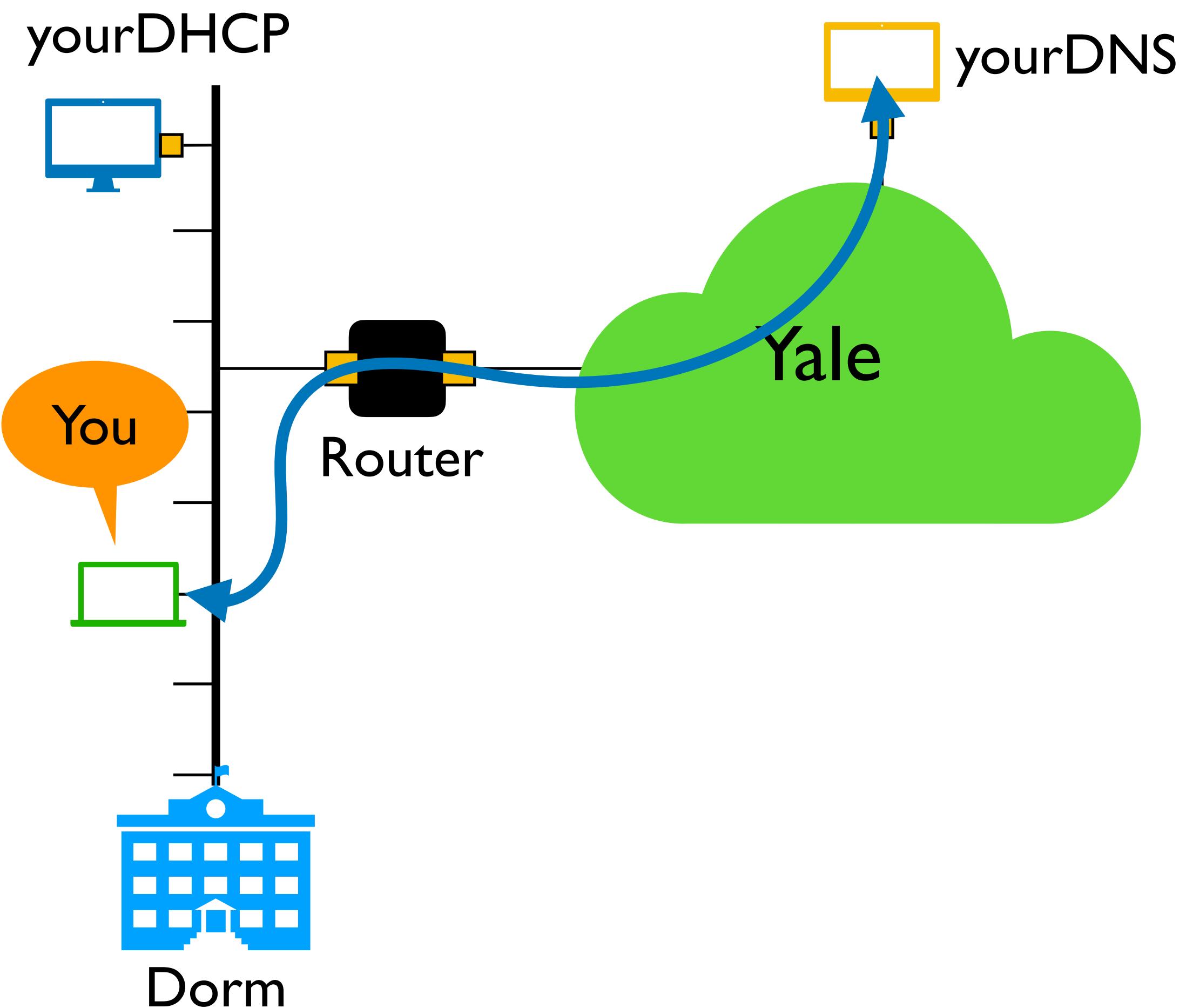


Step 3: Send a DNS request



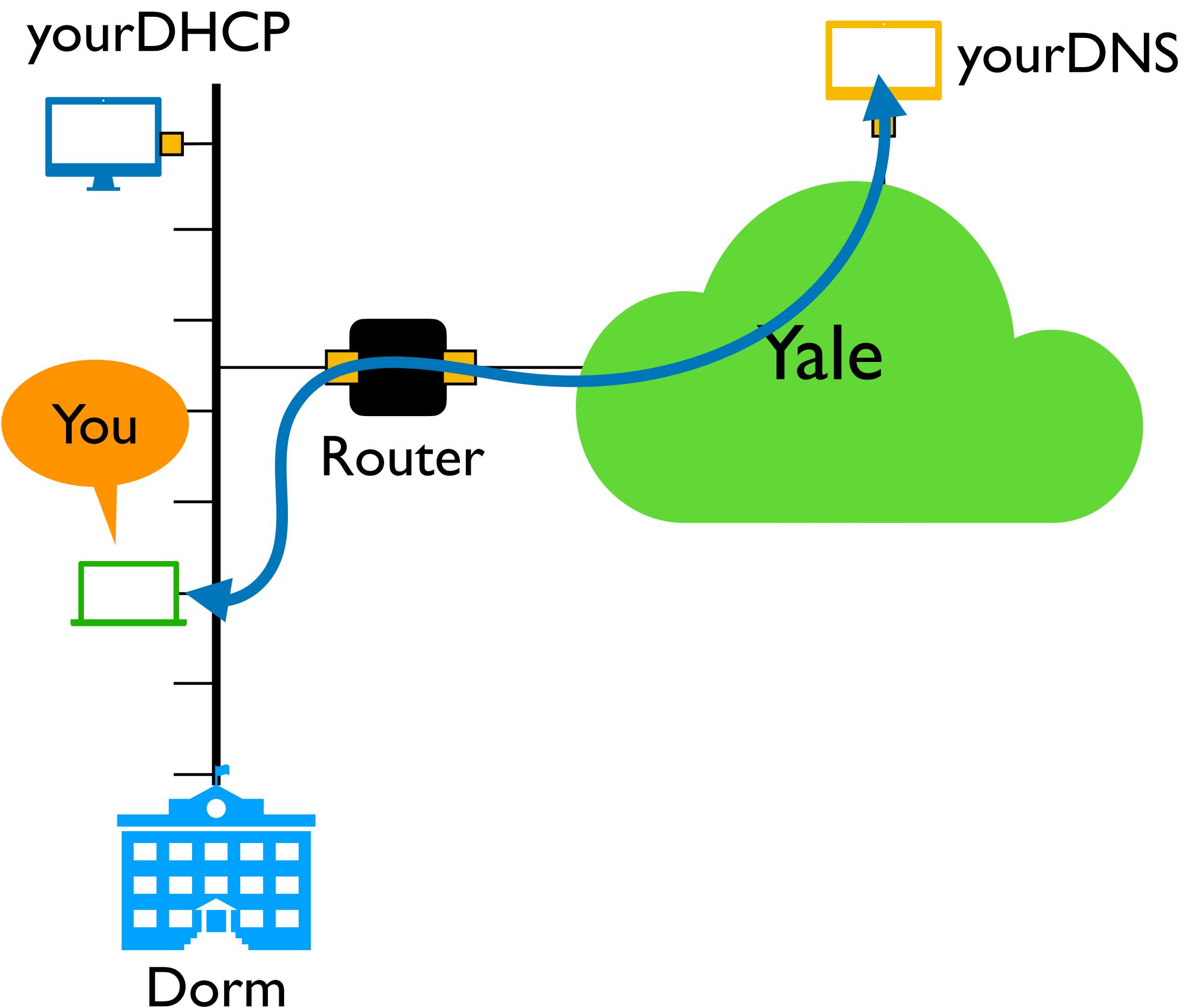
Step 3: Send a DNS request

- Exchange between you and yourDNS



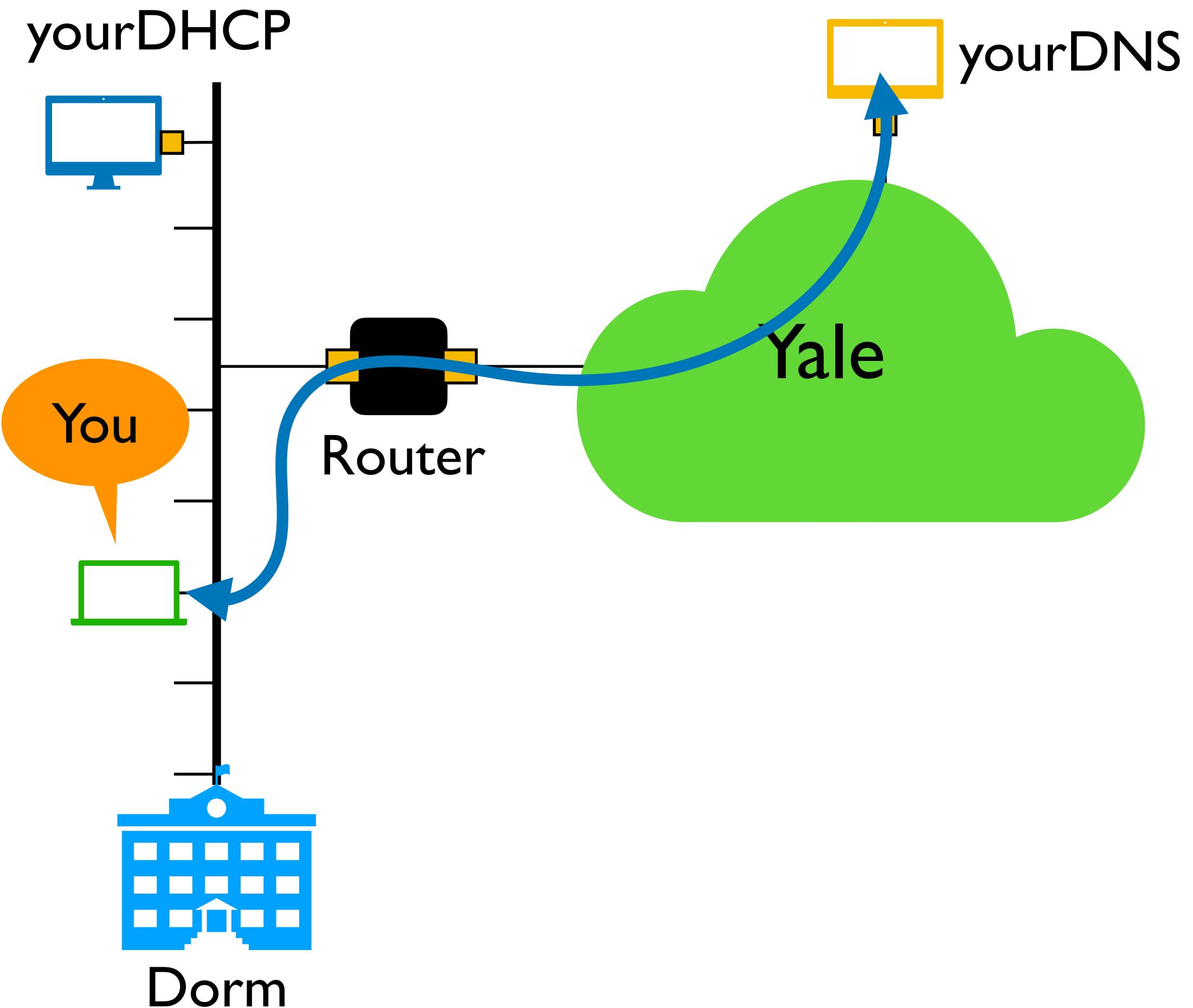
Step 3: Send a DNS request

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



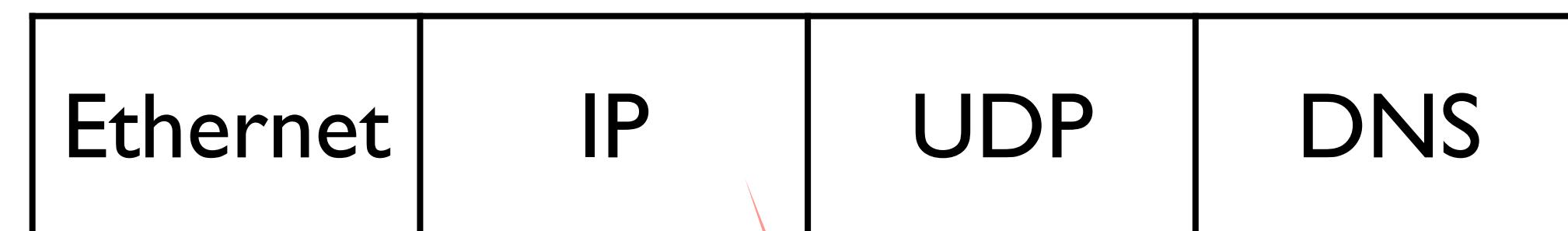
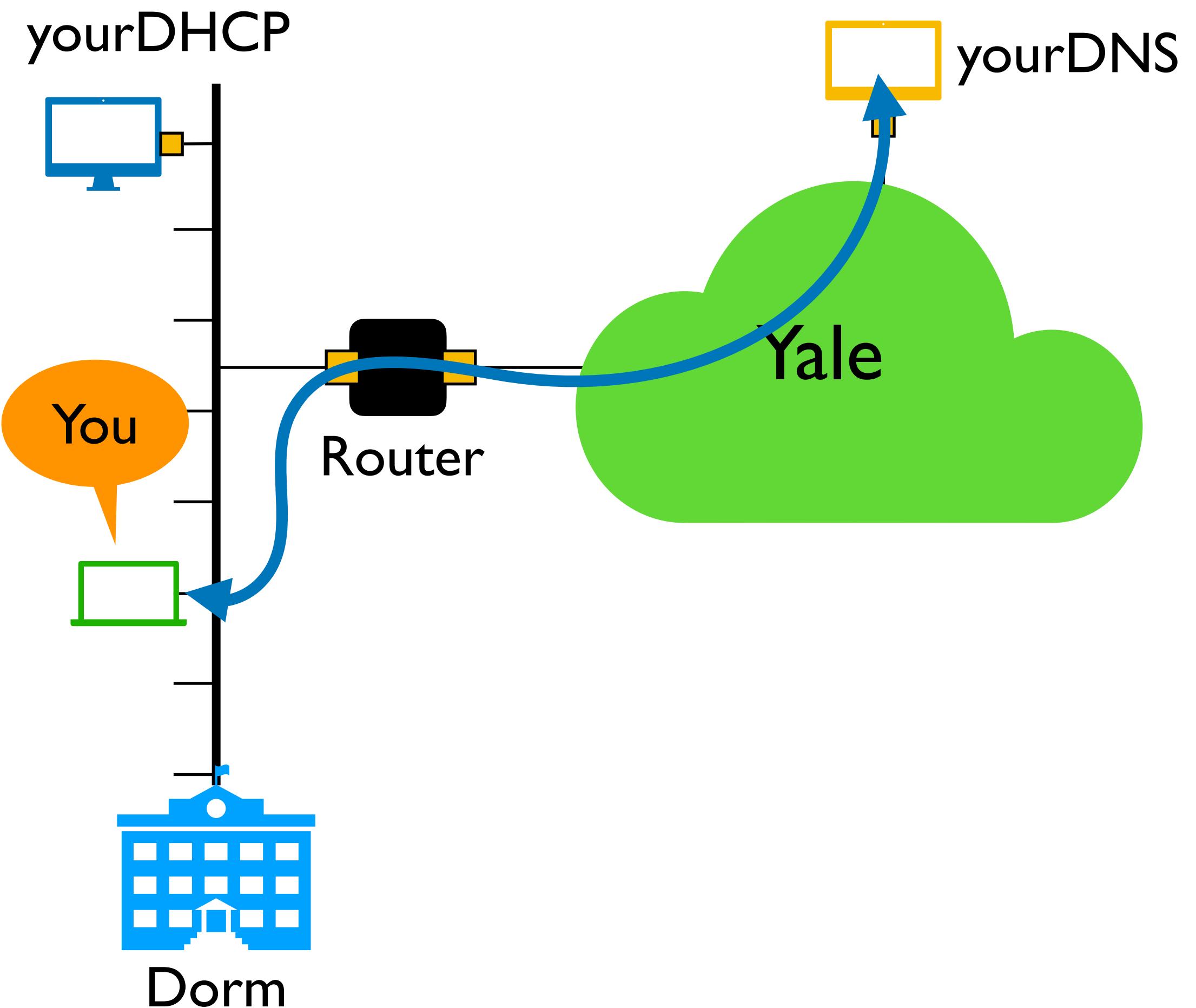
Step 3: Send a DNS request

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



Step 3: Send a DNS request

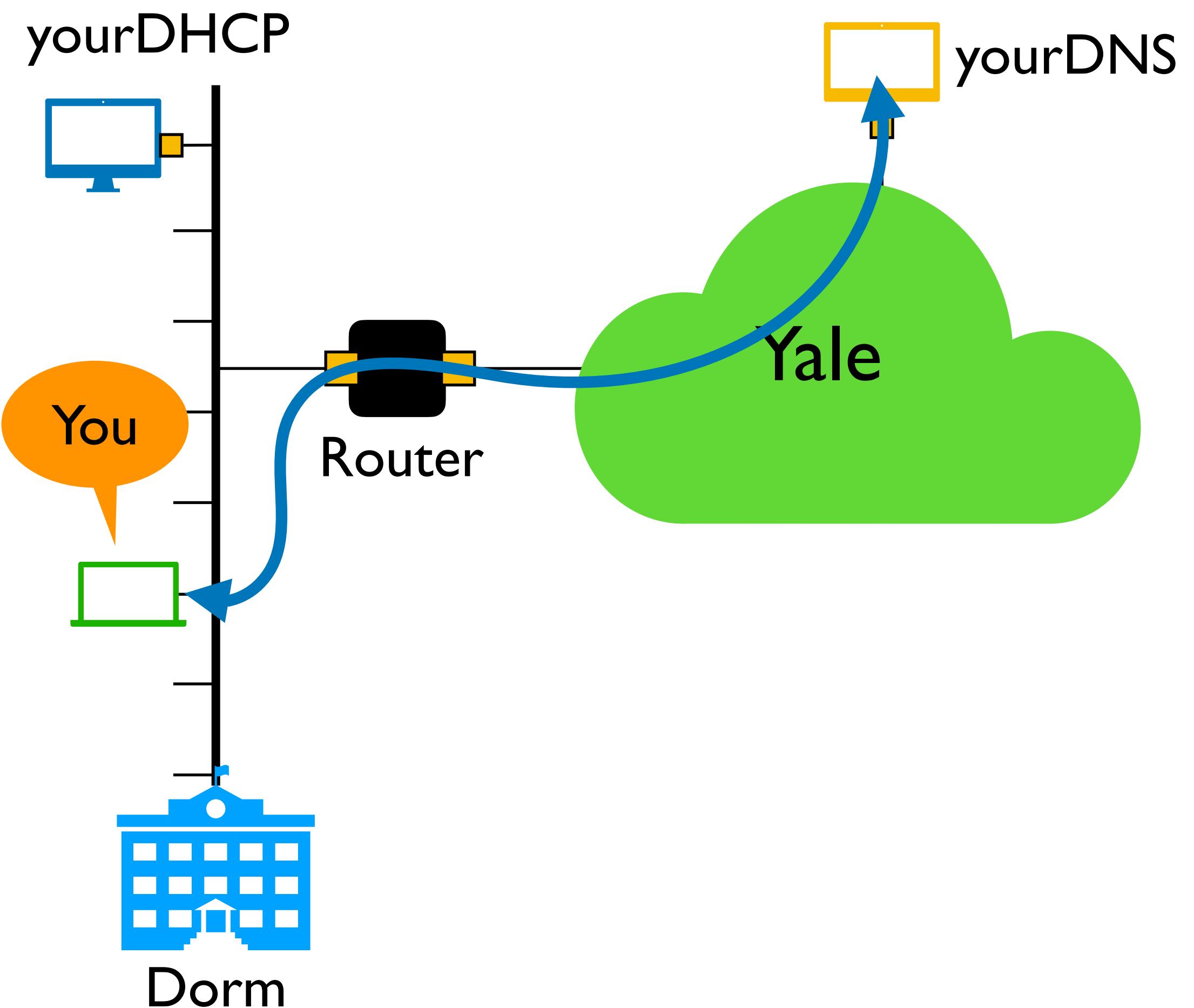
- Exchange between you and yourDNS
- Now ready to send that packet



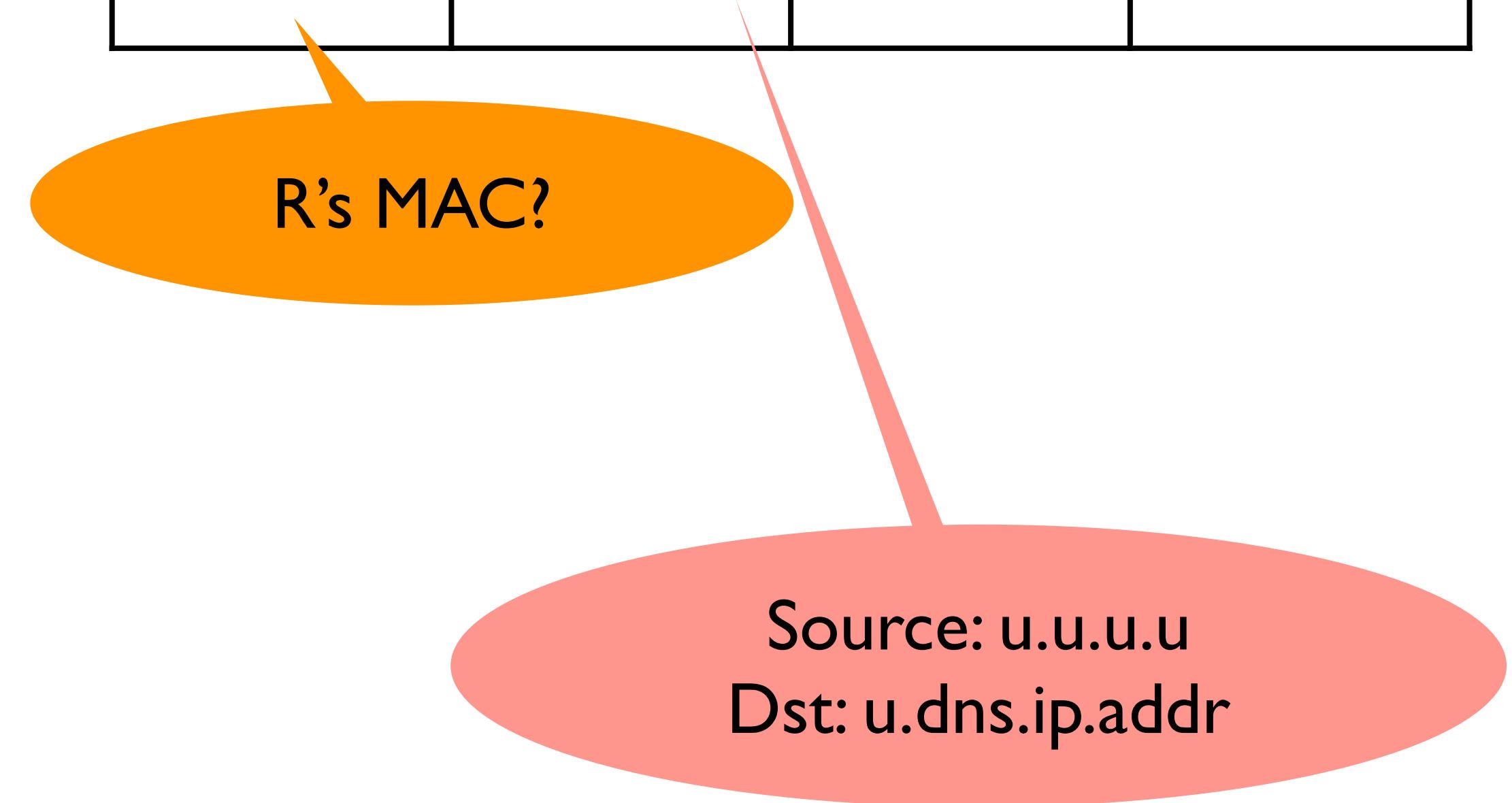
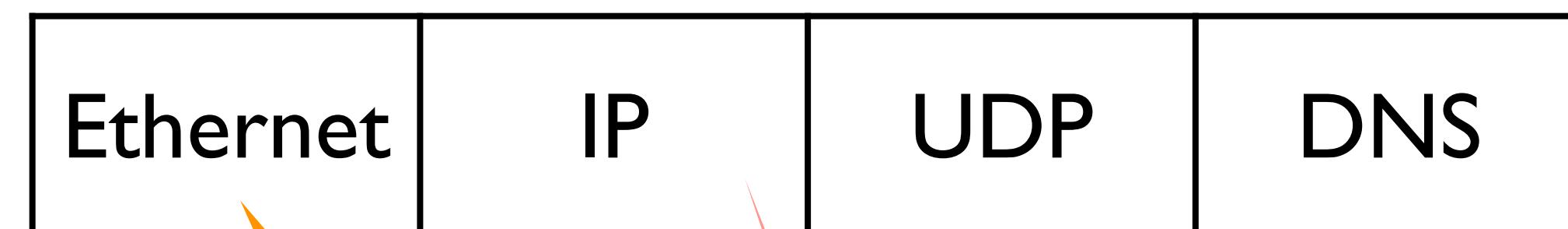
Source: u.u.u.u
Dst: u.dns.ip.addr

Step 3: Send a DNS request

- Exchange between you and yourDNS

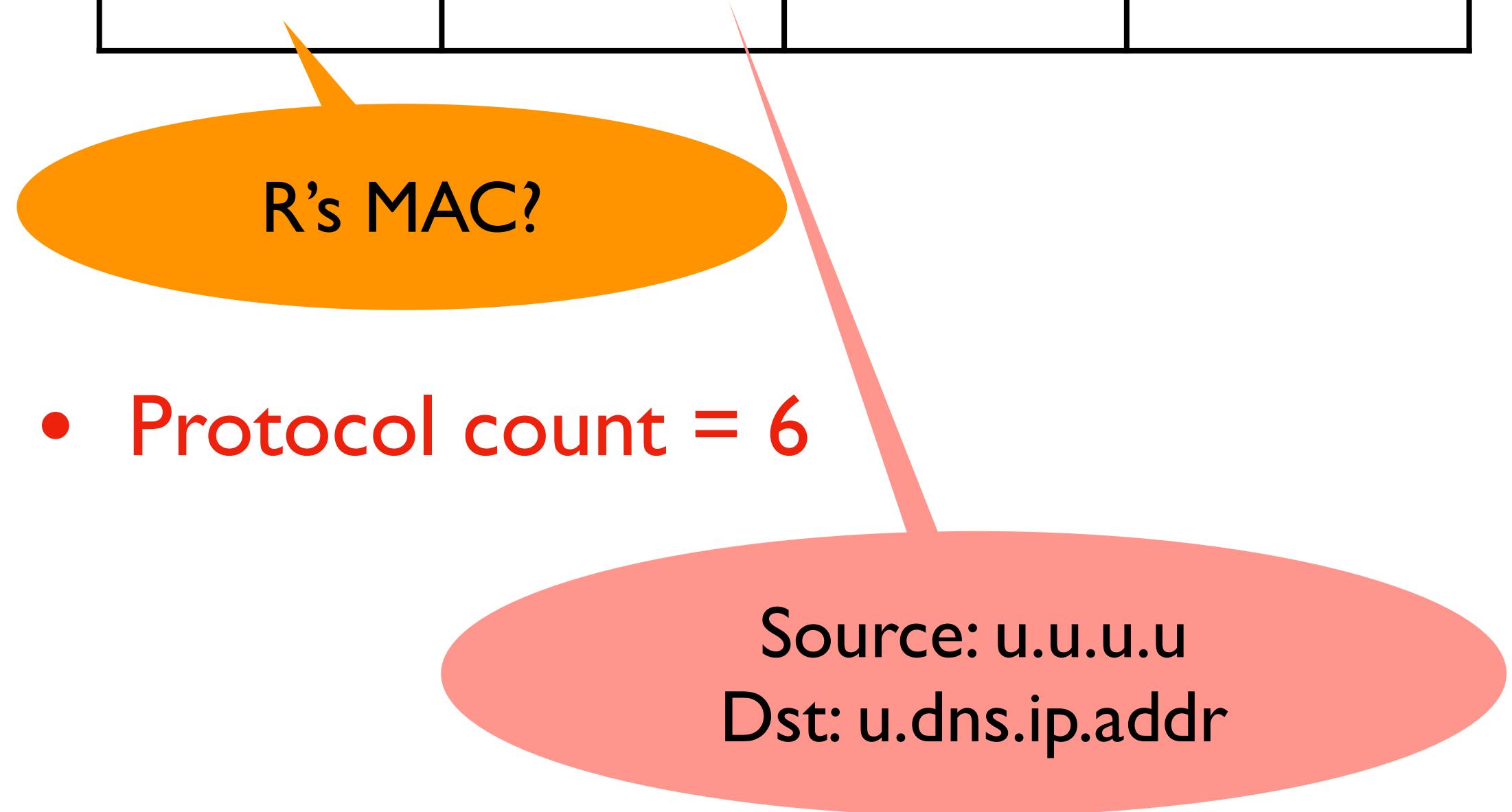
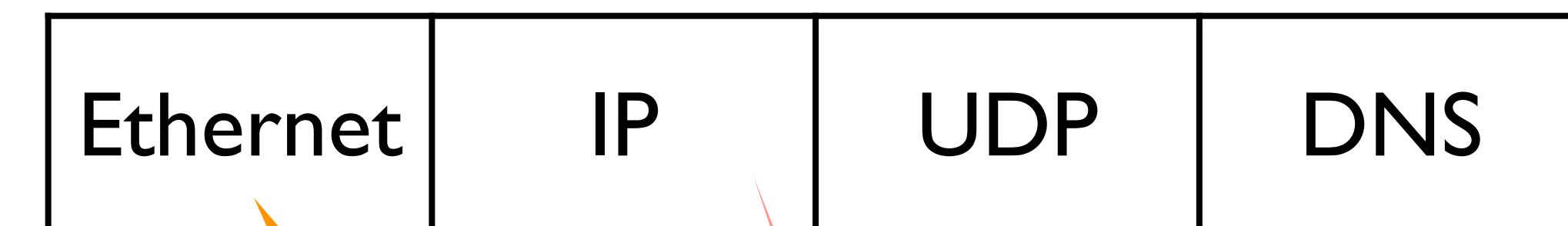
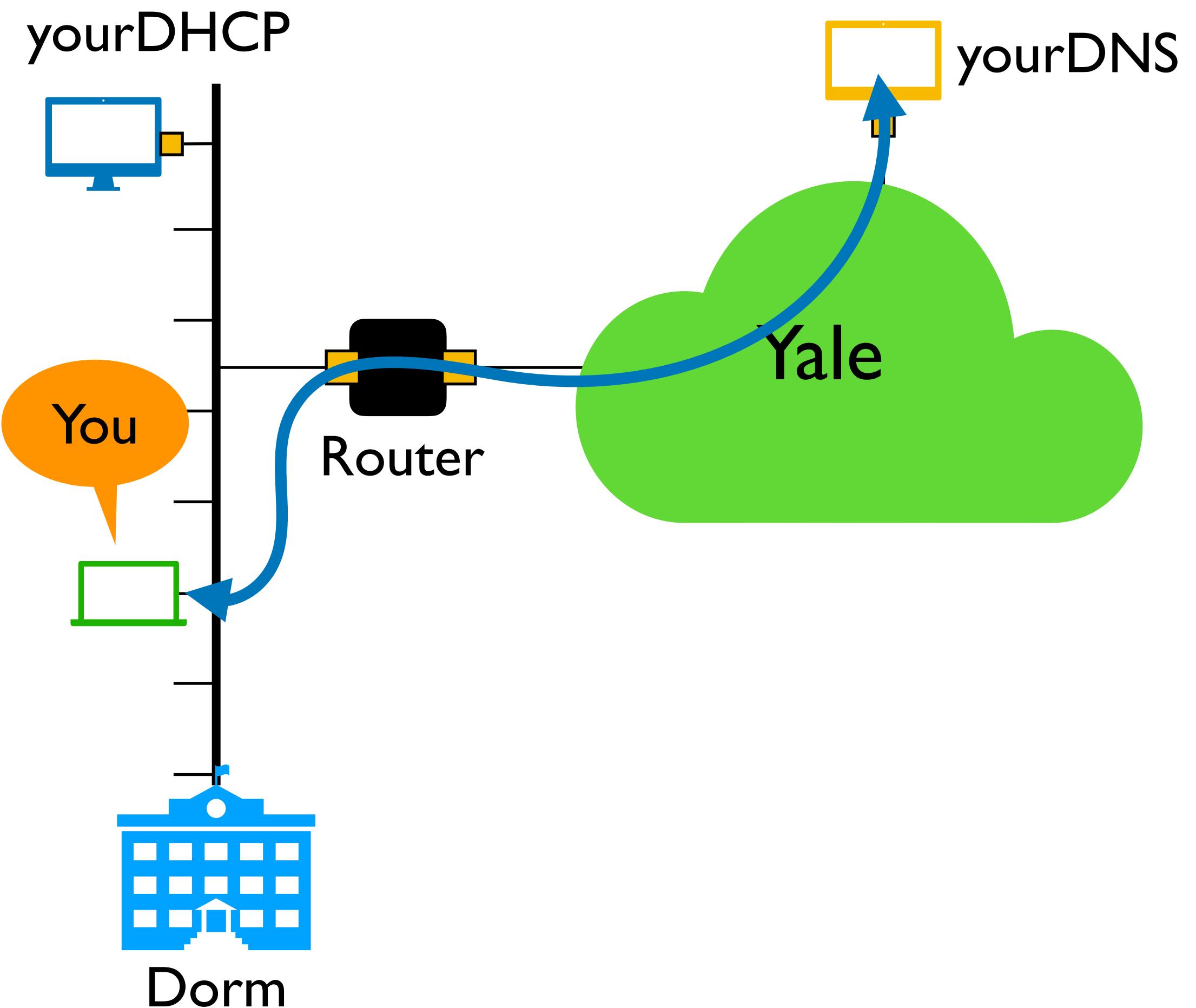


- Now ready to send that packet



Step 3: Send a DNS request

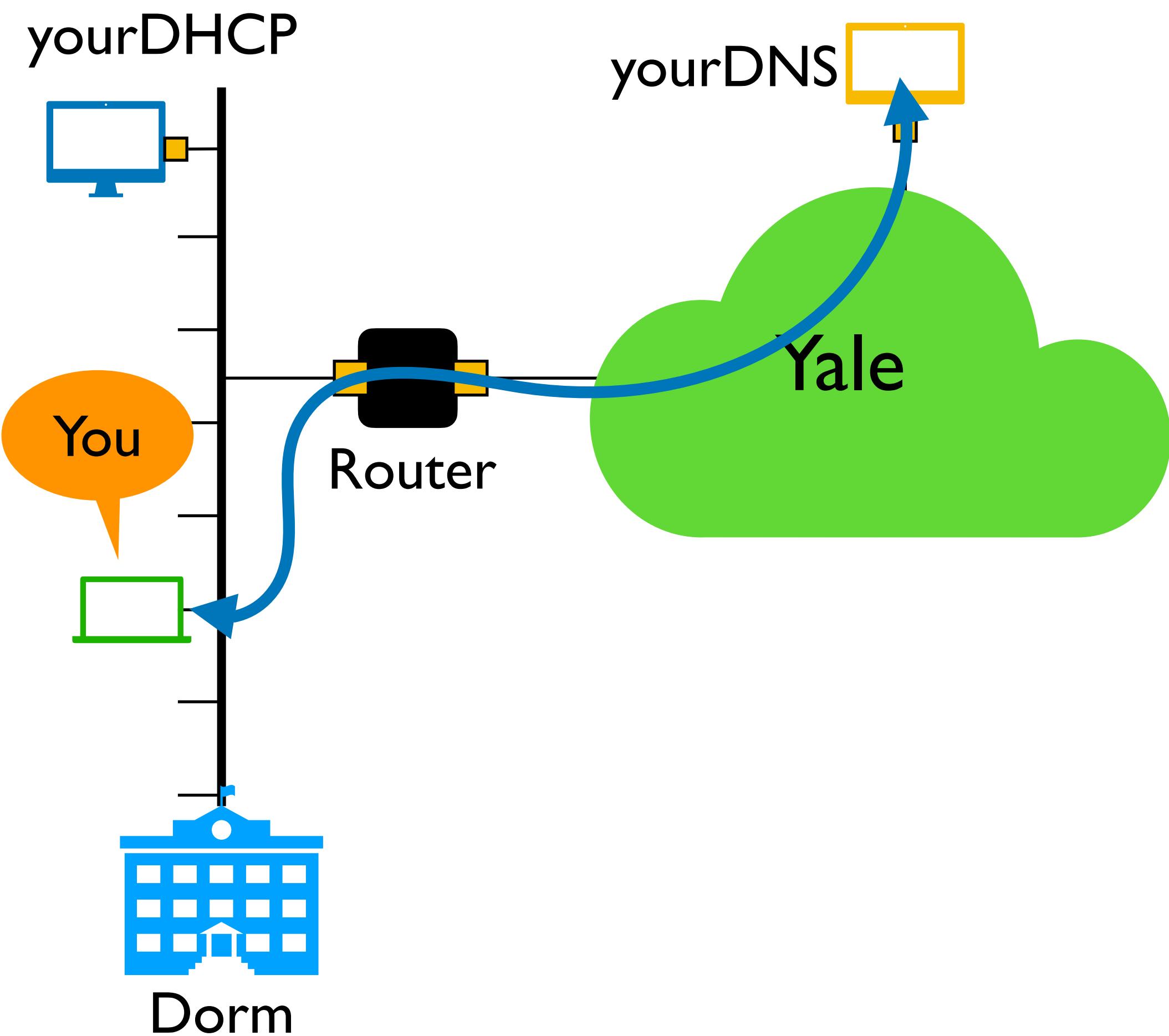
- Exchange between you and yourDNS
- Now ready to send that packet



- Protocol count = 6

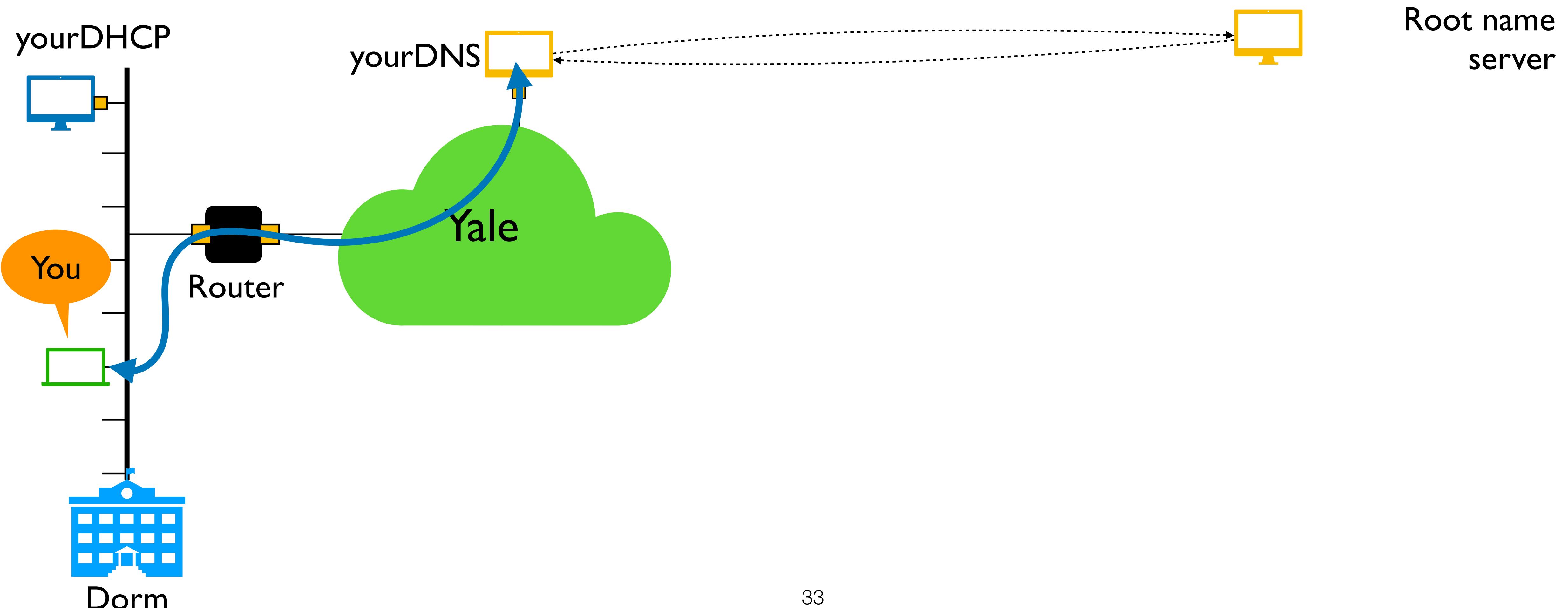
Step 4: yourDNS does its thing

- yourDNS resolves www.google.com



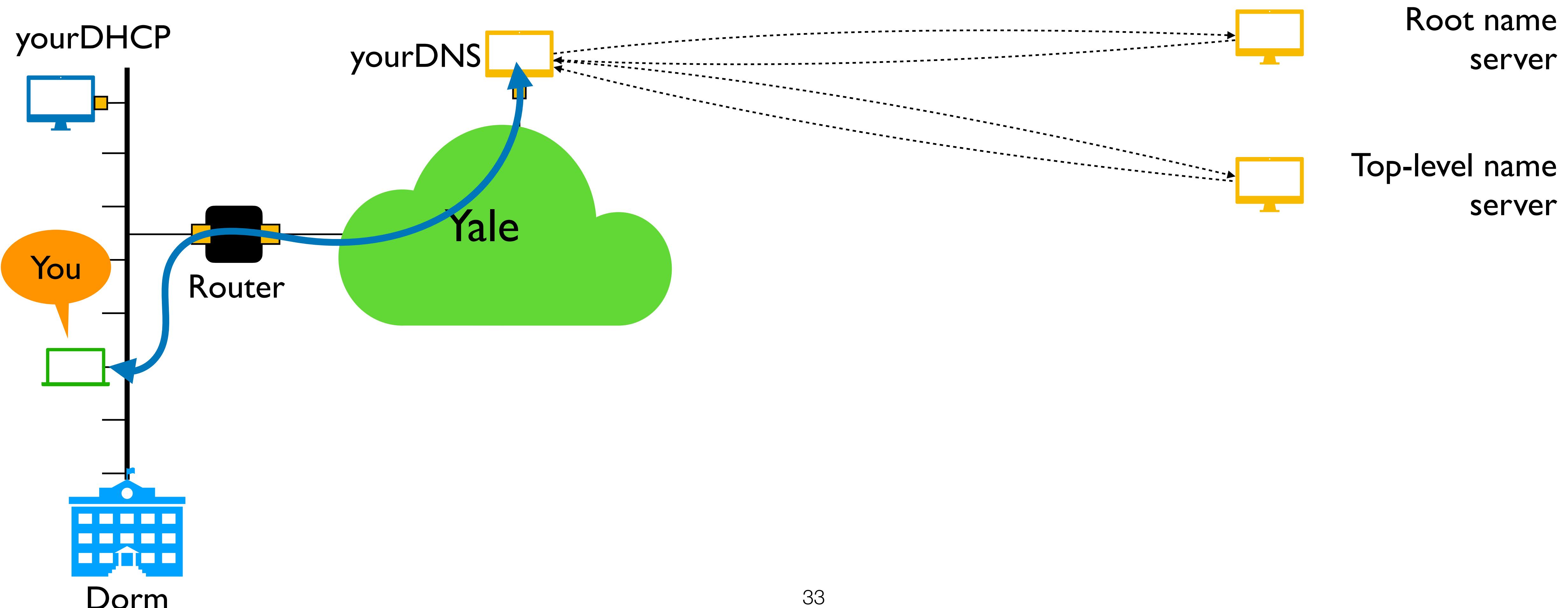
Step 4: yourDNS does its thing

- yourDNS resolves www.google.com



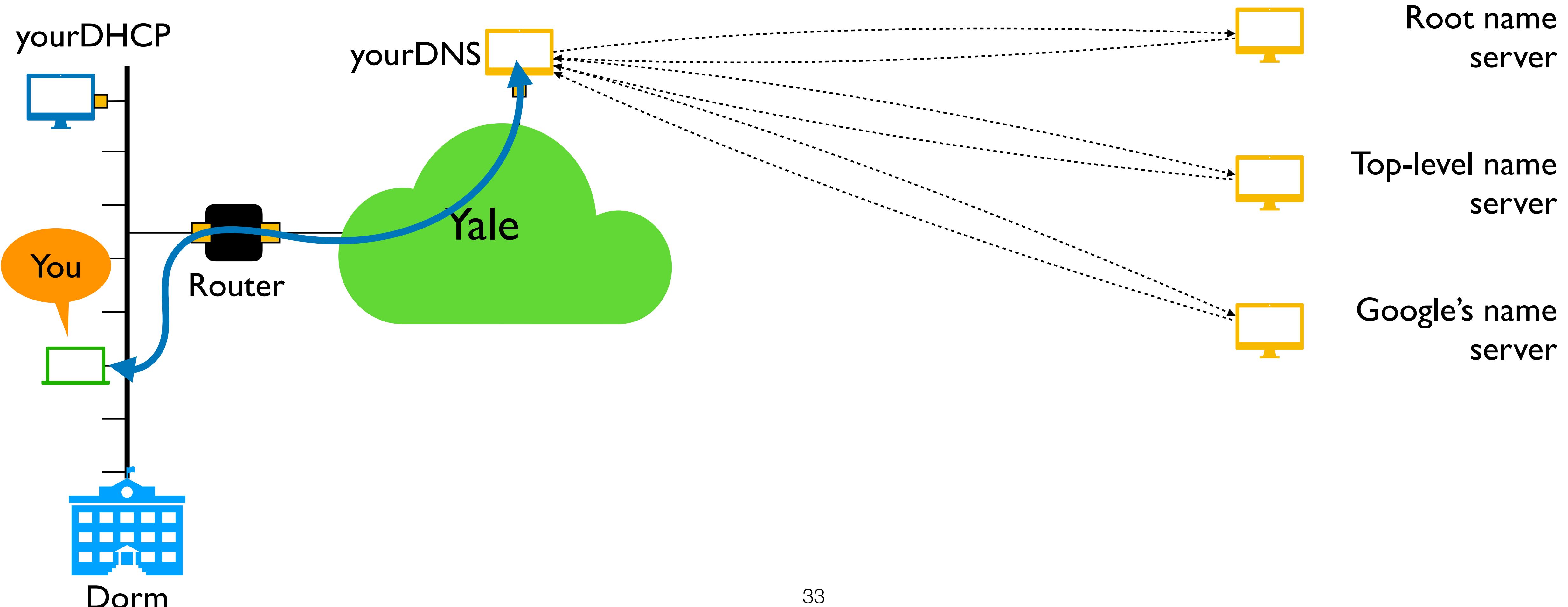
Step 4: yourDNS does its thing

- yourDNS resolves www.google.com



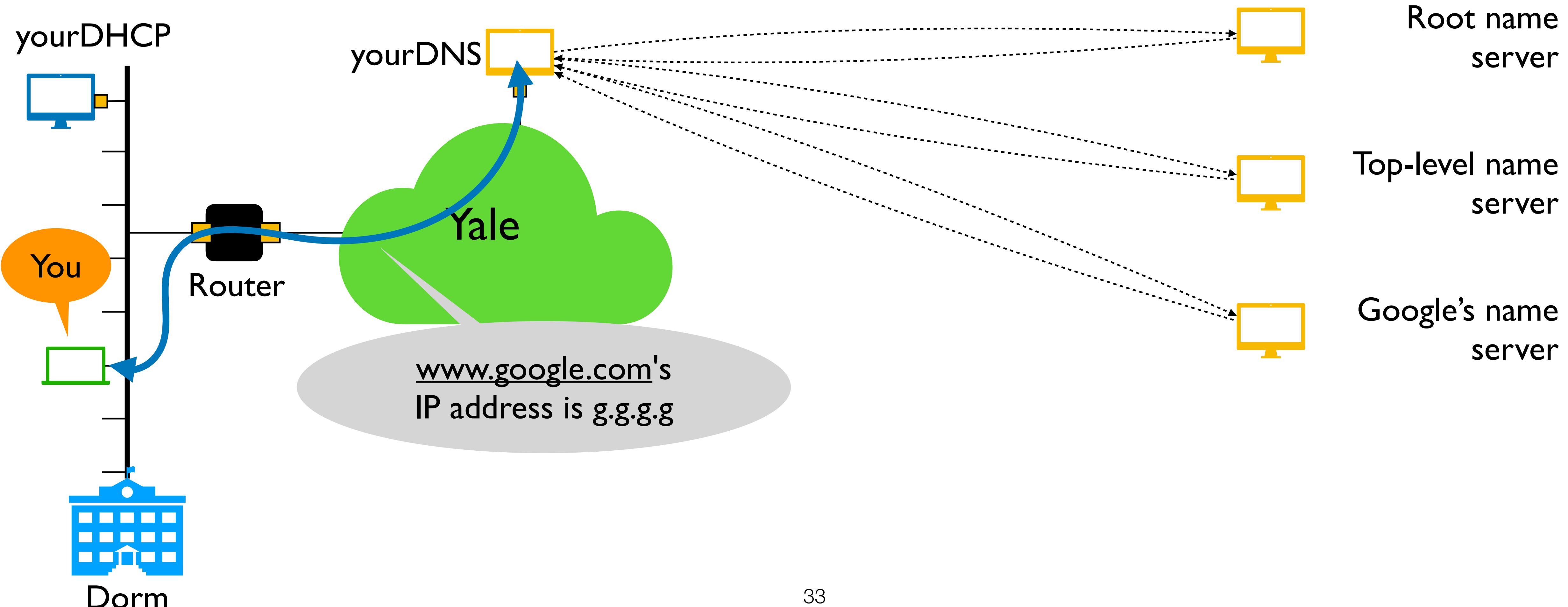
Step 4: yourDNS does its thing

- yourDNS resolves www.google.com

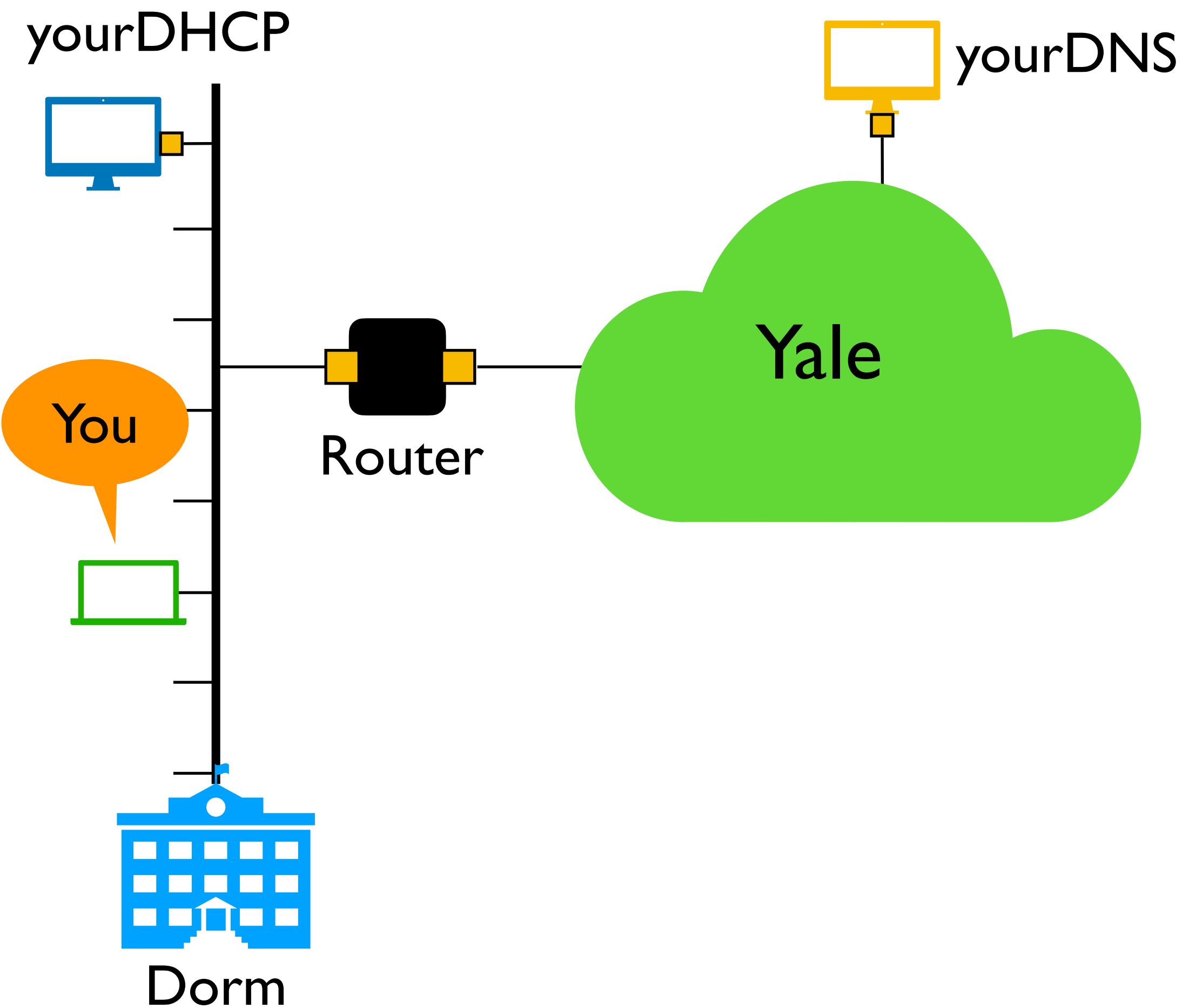


Step 4: yourDNS does its thing

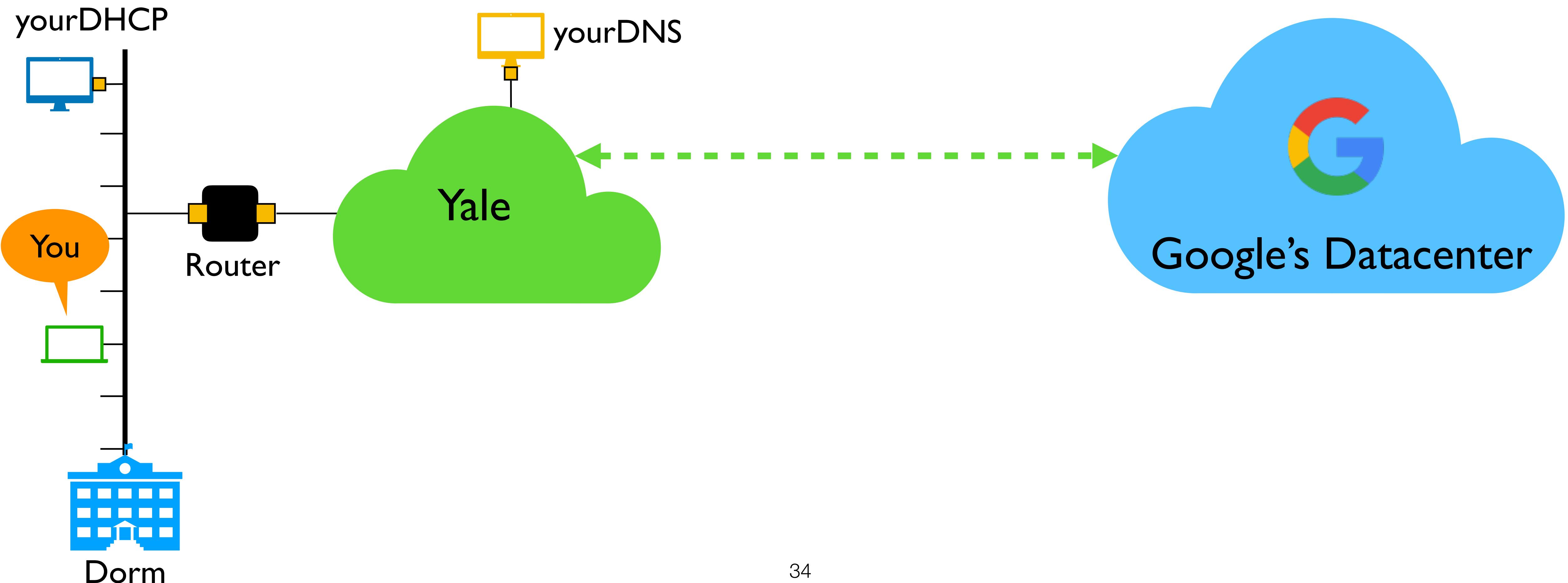
- yourDNS resolves www.google.com



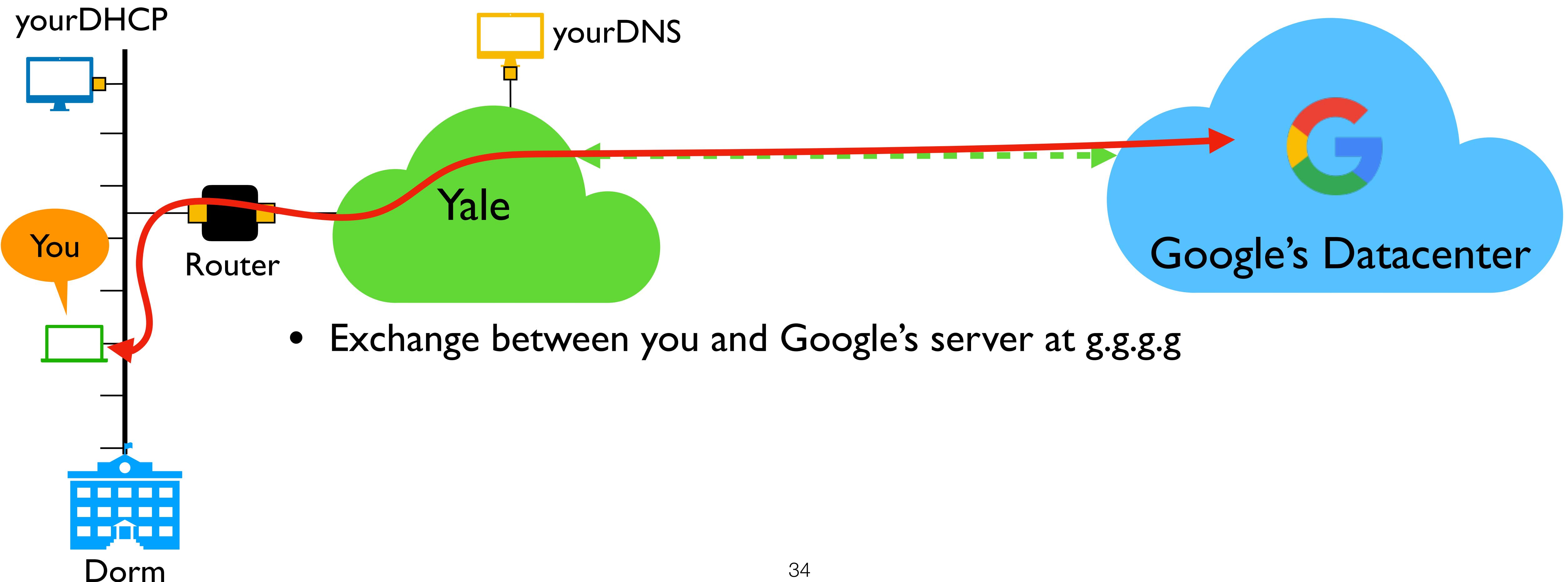
Step 5: Getting the content (at last)



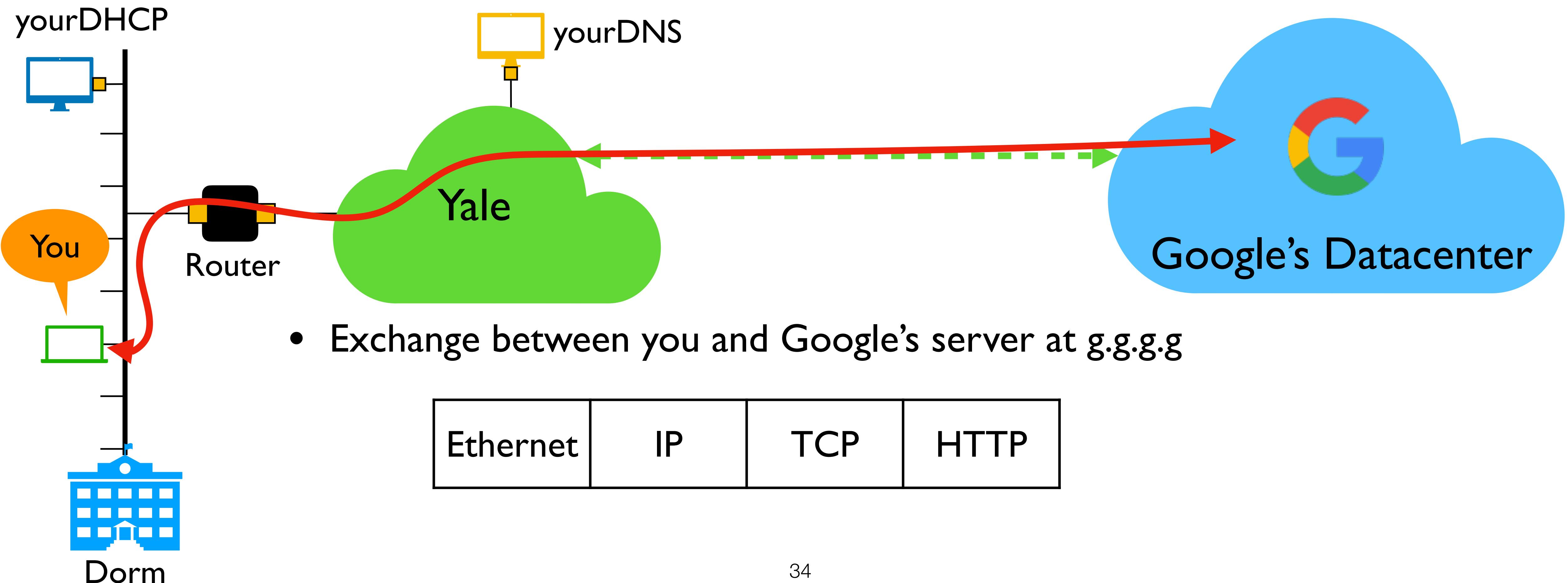
Step 5: Getting the content (at last)



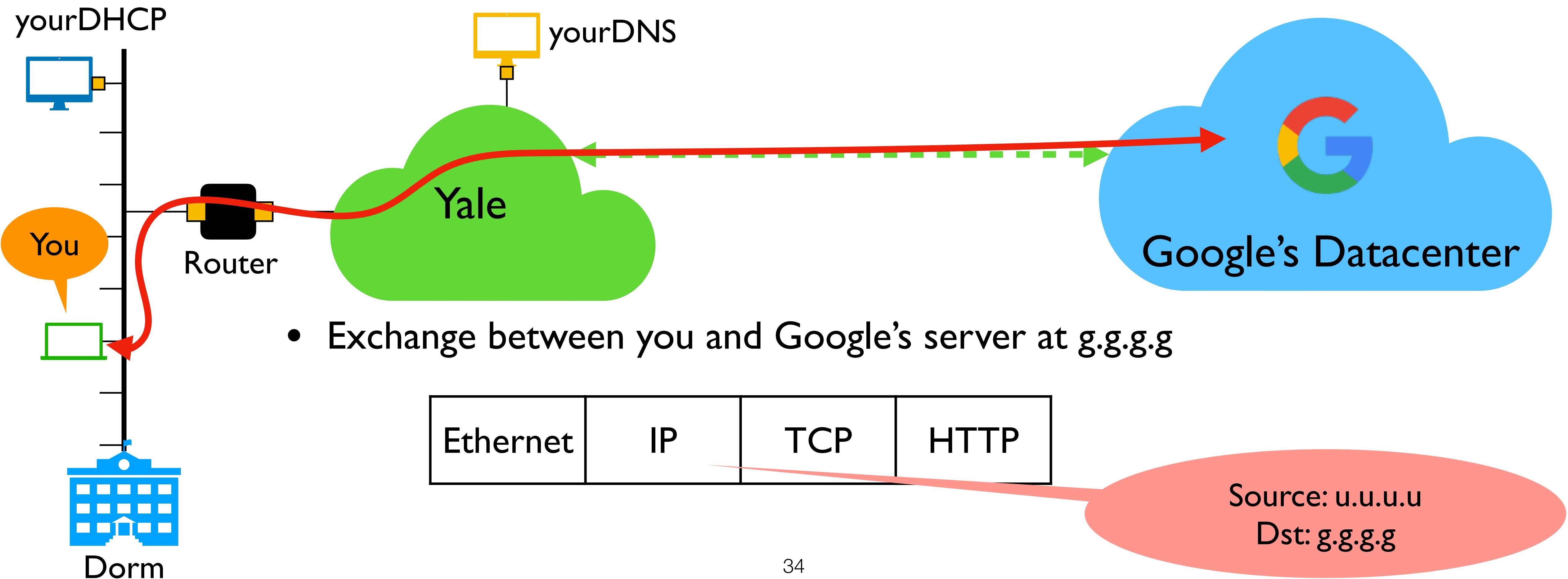
Step 5: Getting the content (at last)



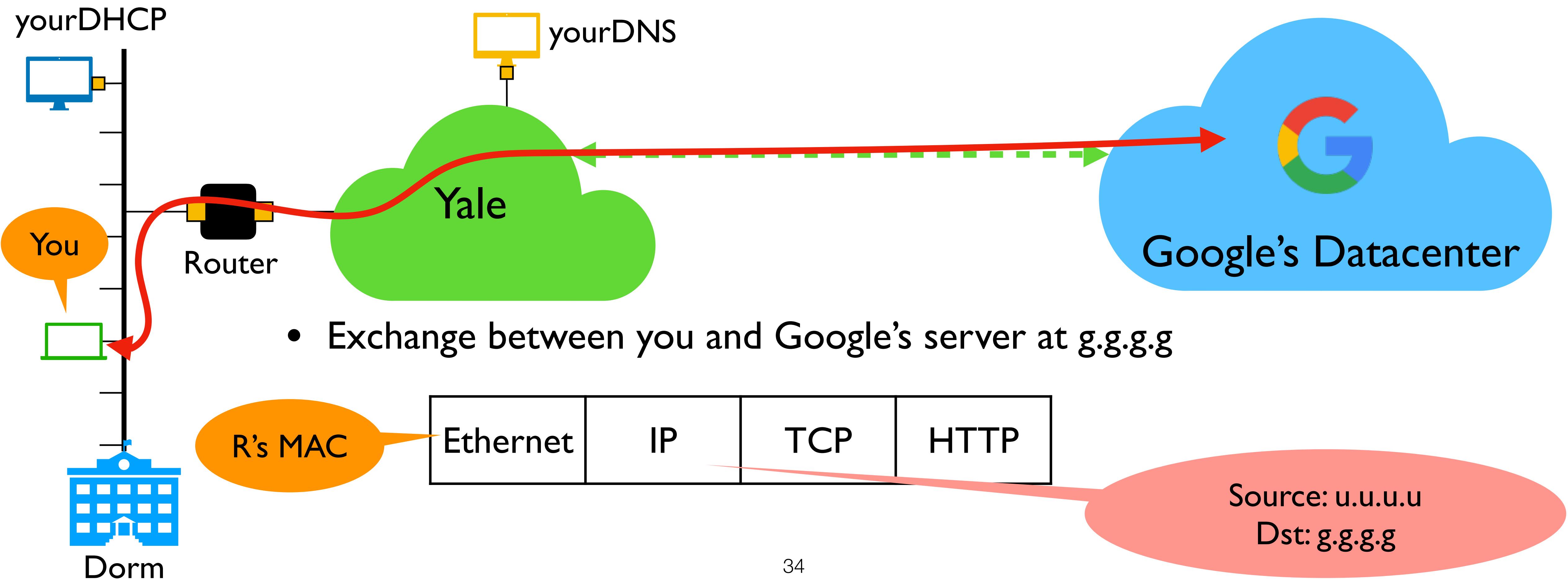
Step 5: Getting the content (at last)



Step 5: Getting the content (at last)

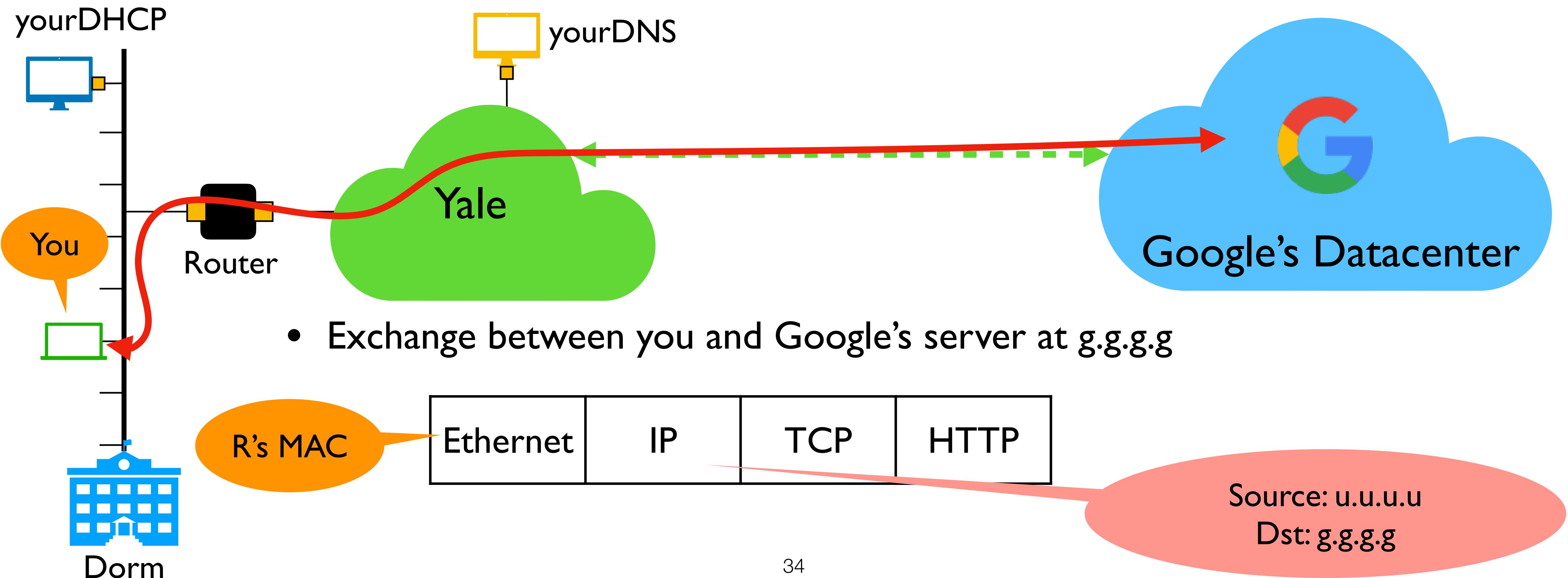


Step 5: Getting the content (at last)



Step 5: Getting the content (at last)

Final protocol count = 8



Recap: Name discovery/resoulution

Recap: Name discovery/resoulution

- MAC addresses?

Recap: Name discovery/resoulution

- MAC addresses?
 - My own: hardcoded

Recap: Name discovery/resoulution

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

Recap: Name discovery/resoulution

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

Recap: Name discovery/resoulution

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

Recap: Name discovery/resoulution

- MAC addresses?
 - My own: hardcoded
 - Others: ARP (given IP address)
- IP addresses?
 - My own: DHCP
 - Others: DNS (given domain name)

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)

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?

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

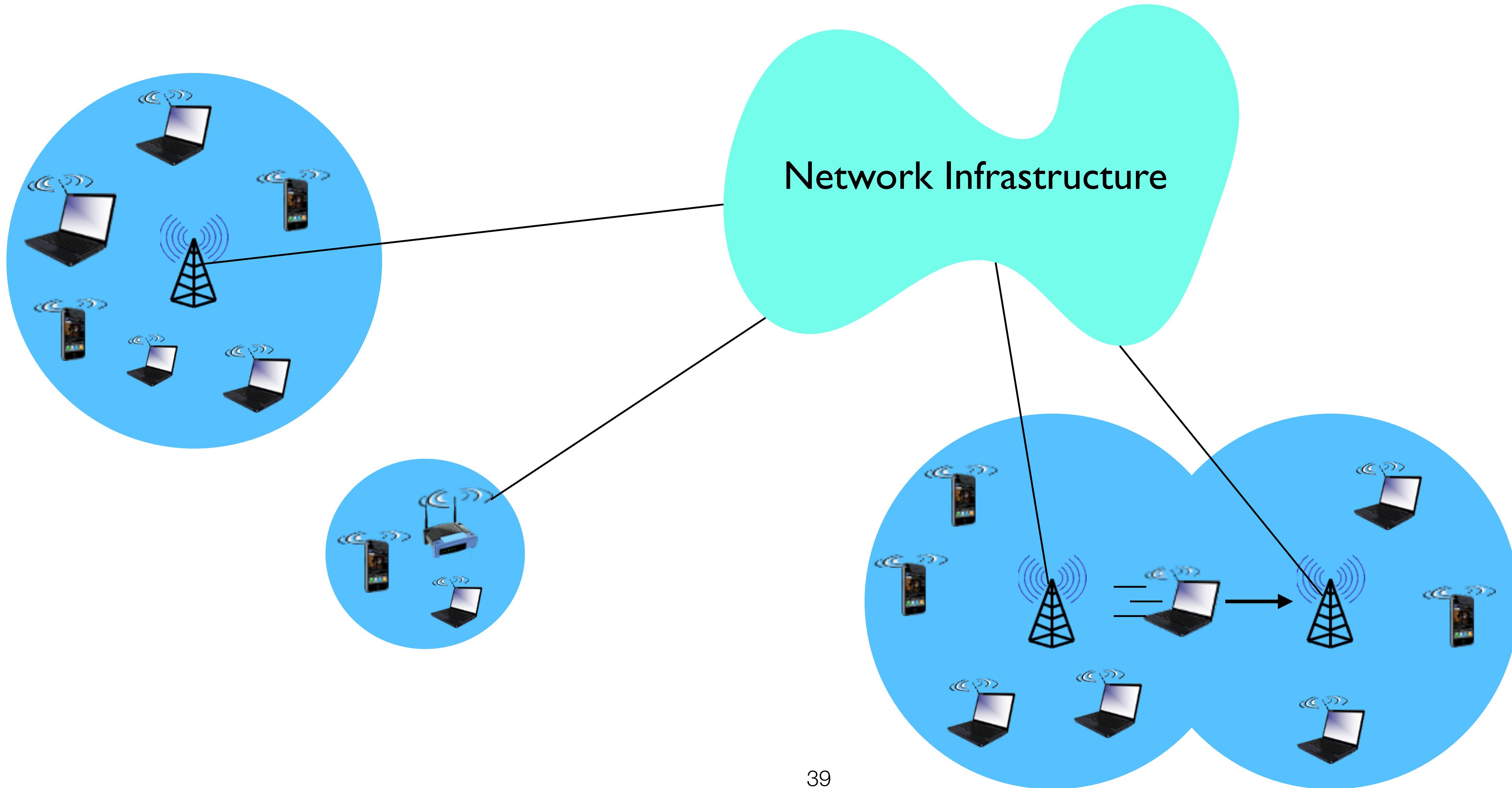
Wireless Networks

Wireless Networks

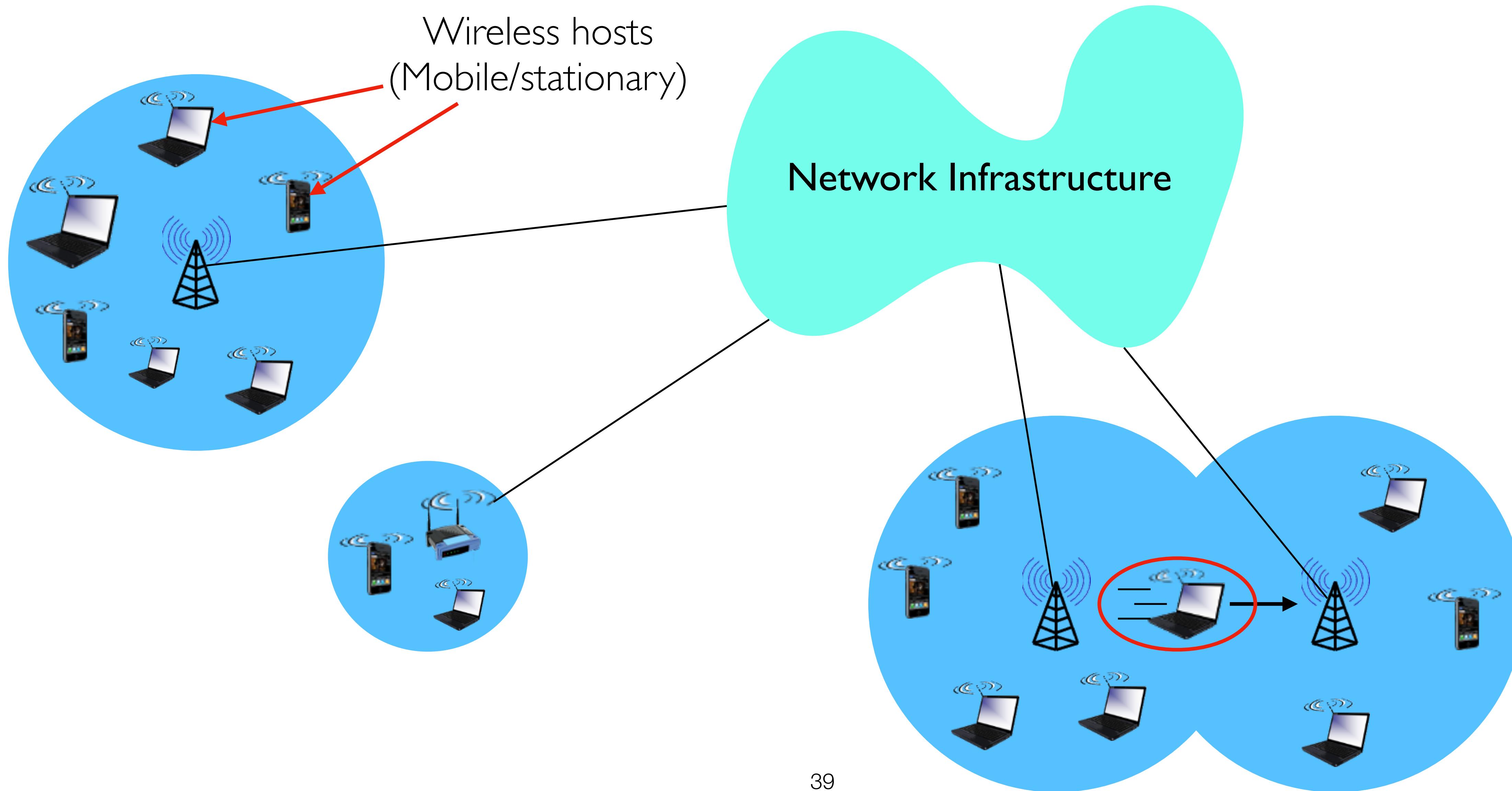
- Albert Einstein, when asked to describe radio:

*“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.**”*

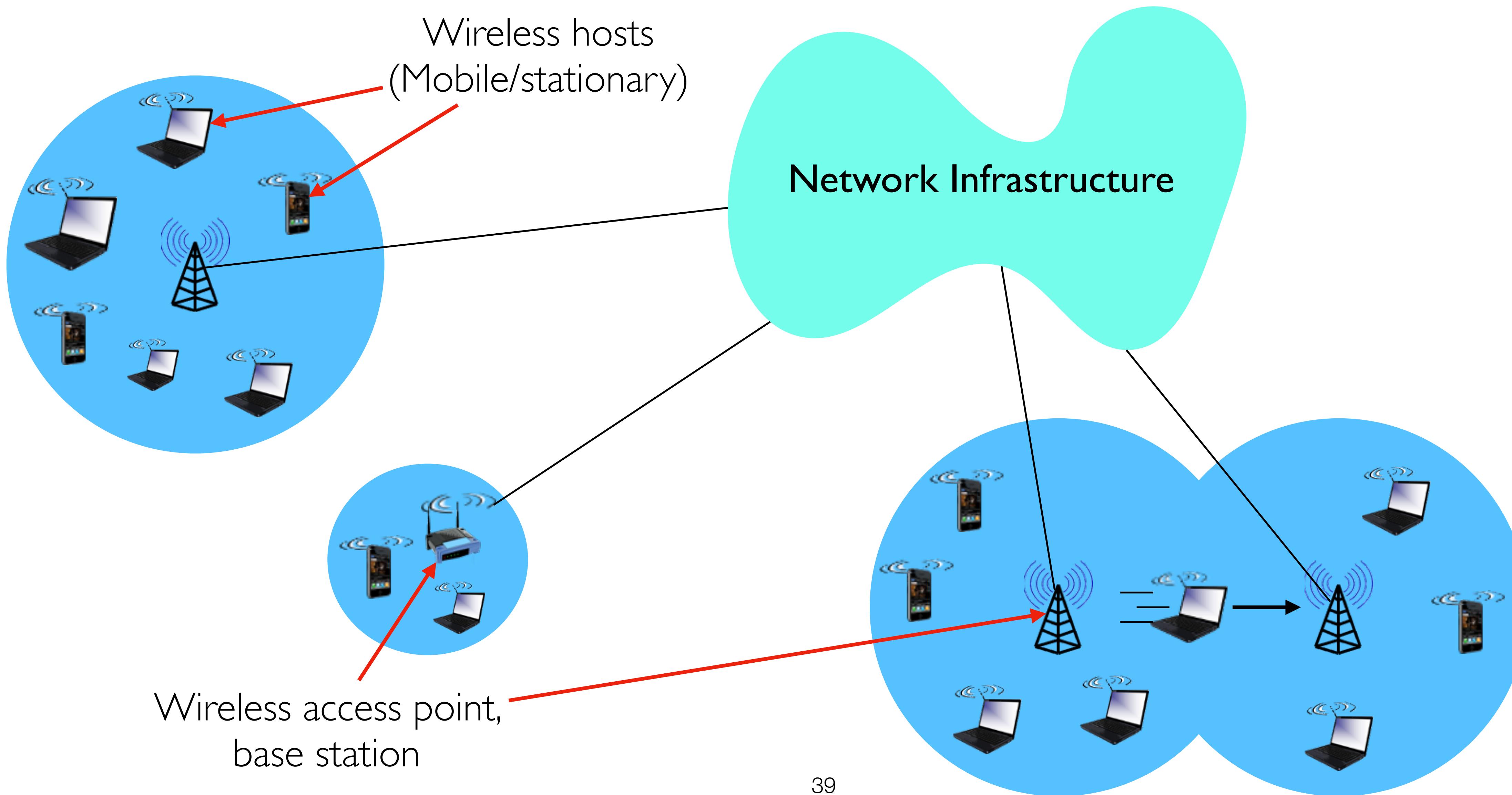
Elements of a Wireless Network



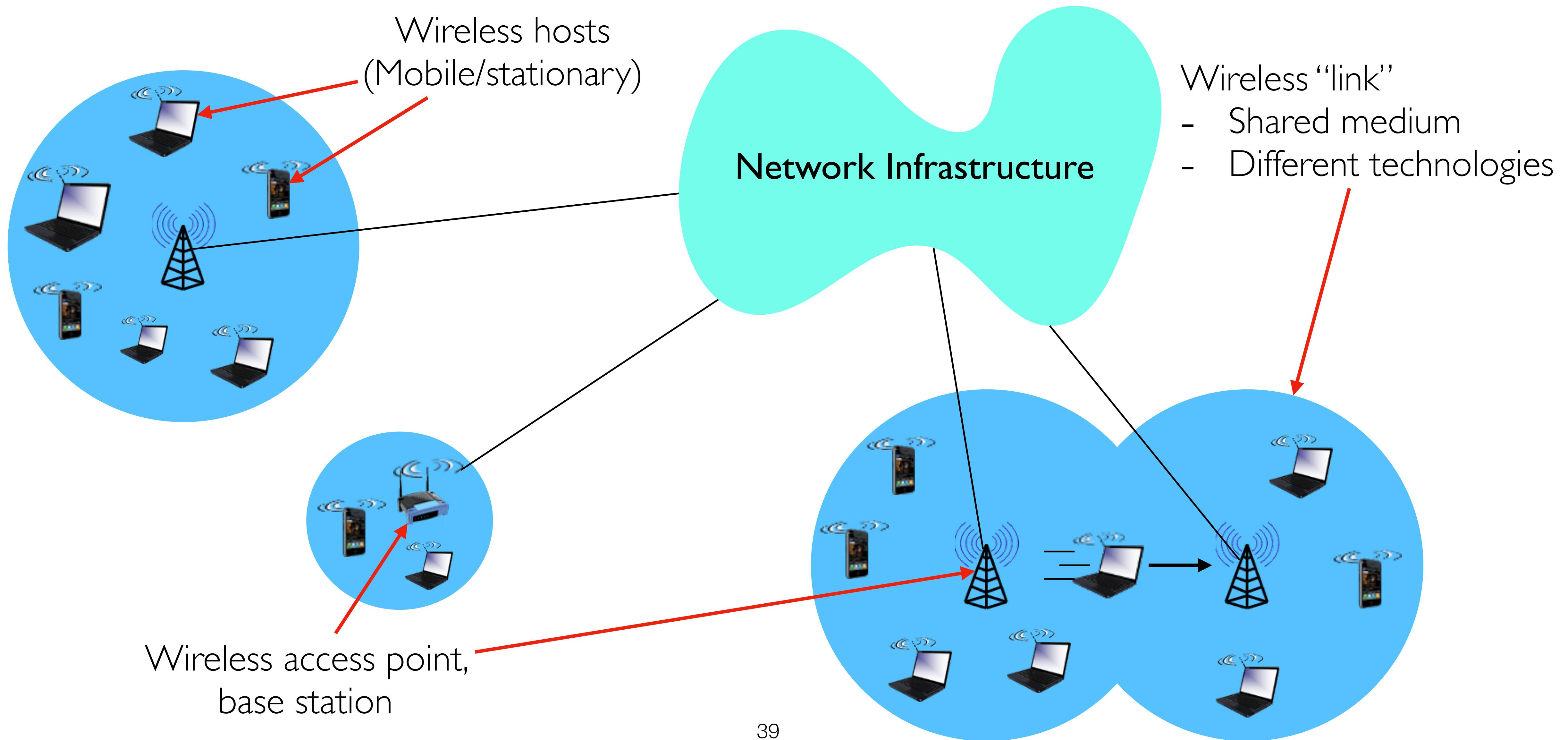
Elements of a Wireless Network



Elements of a Wireless Network



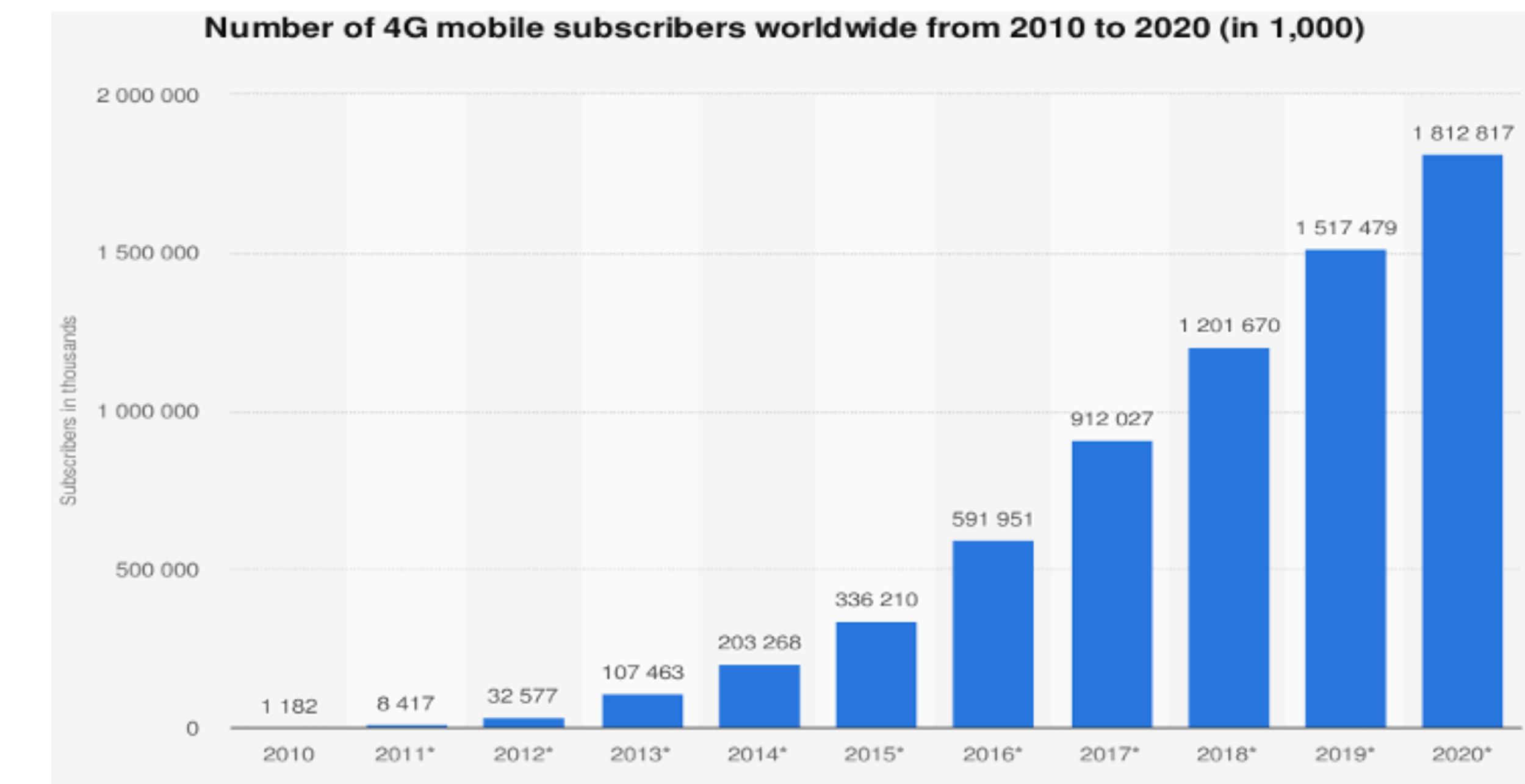
Elements of a Wireless Network



Widespread Deployment

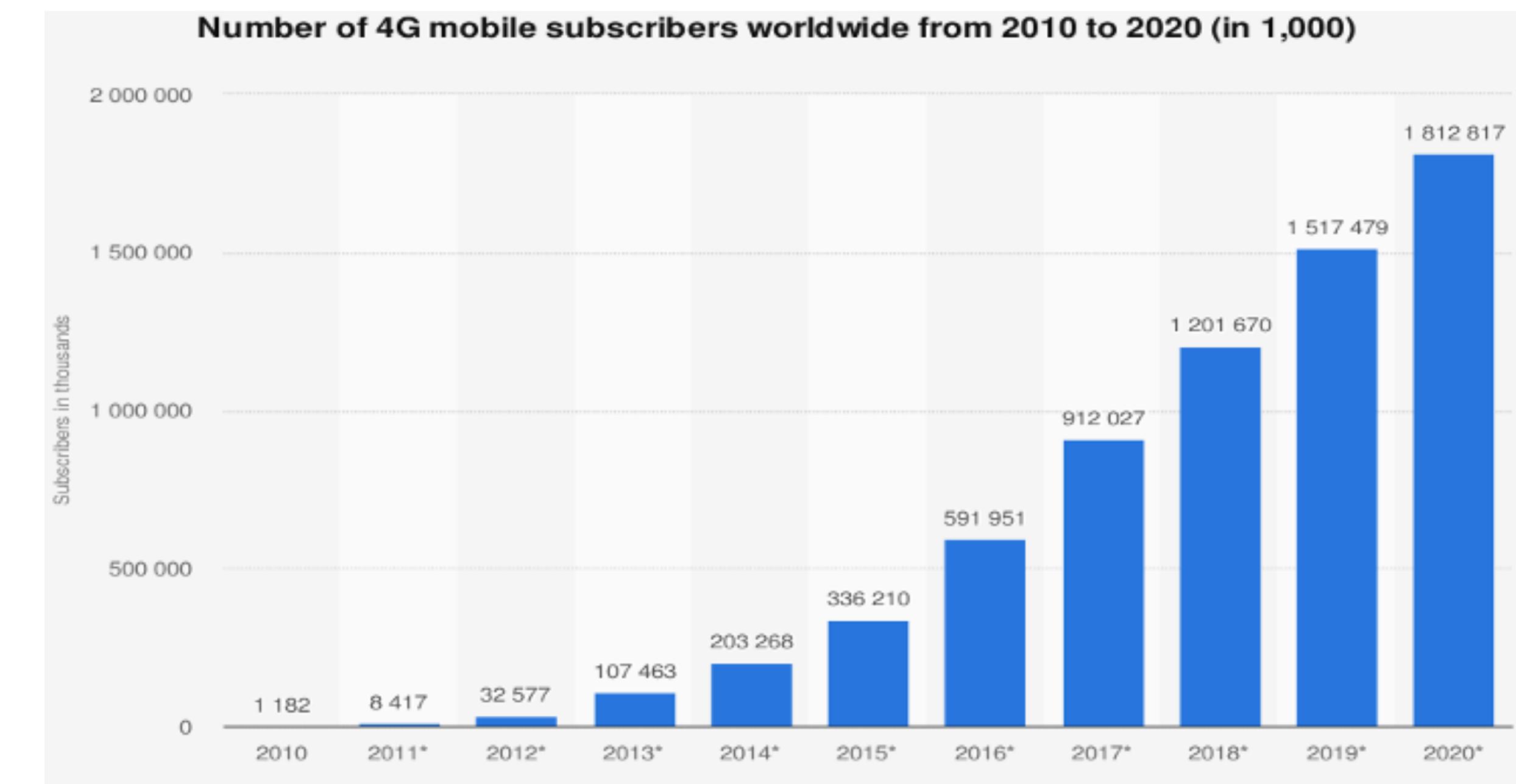
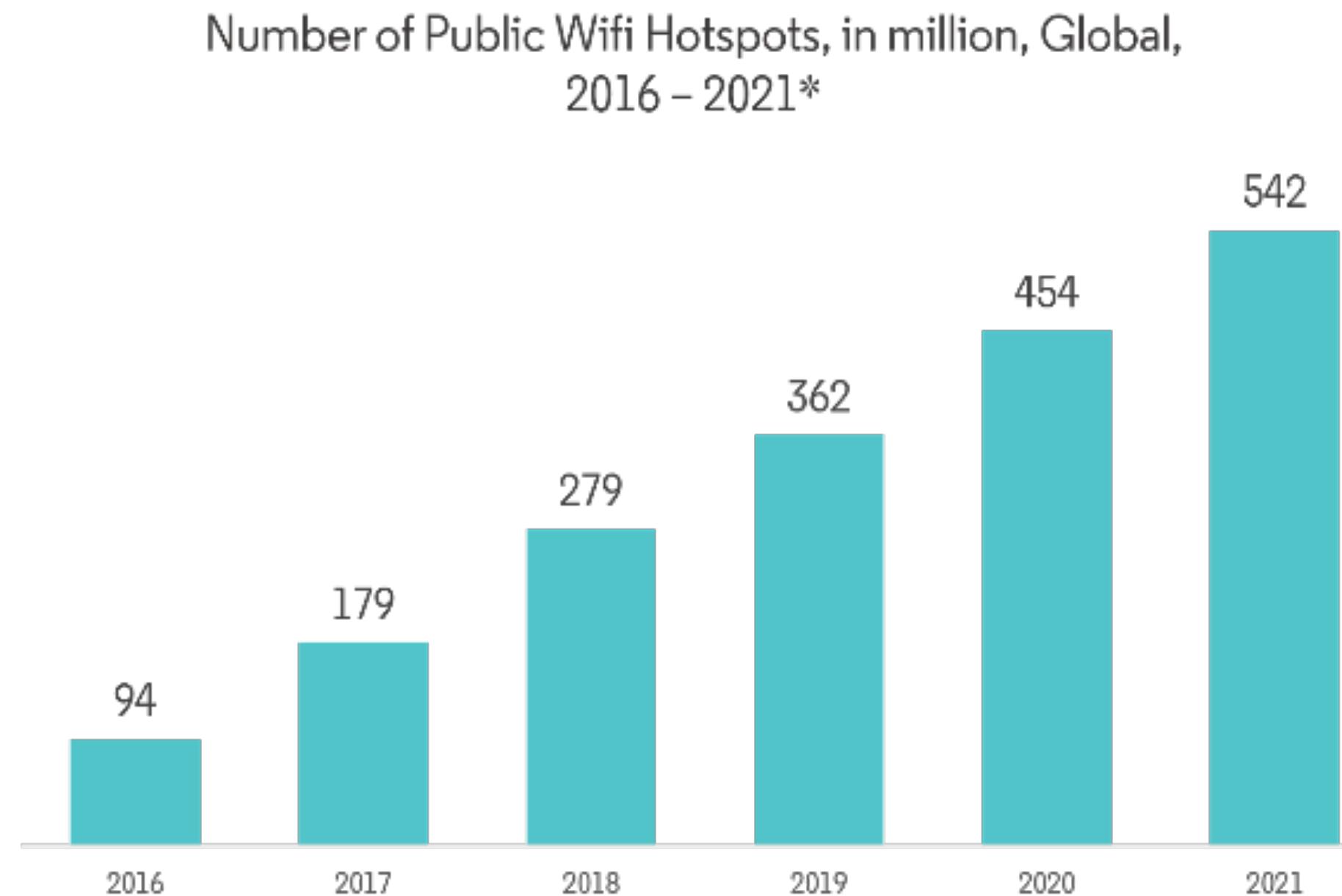
Widespread Deployment

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



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

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

Frequency or Wavelength

Characteristics of a wireless signal

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

Frequency or Wavelength

Characteristics of a wireless signal

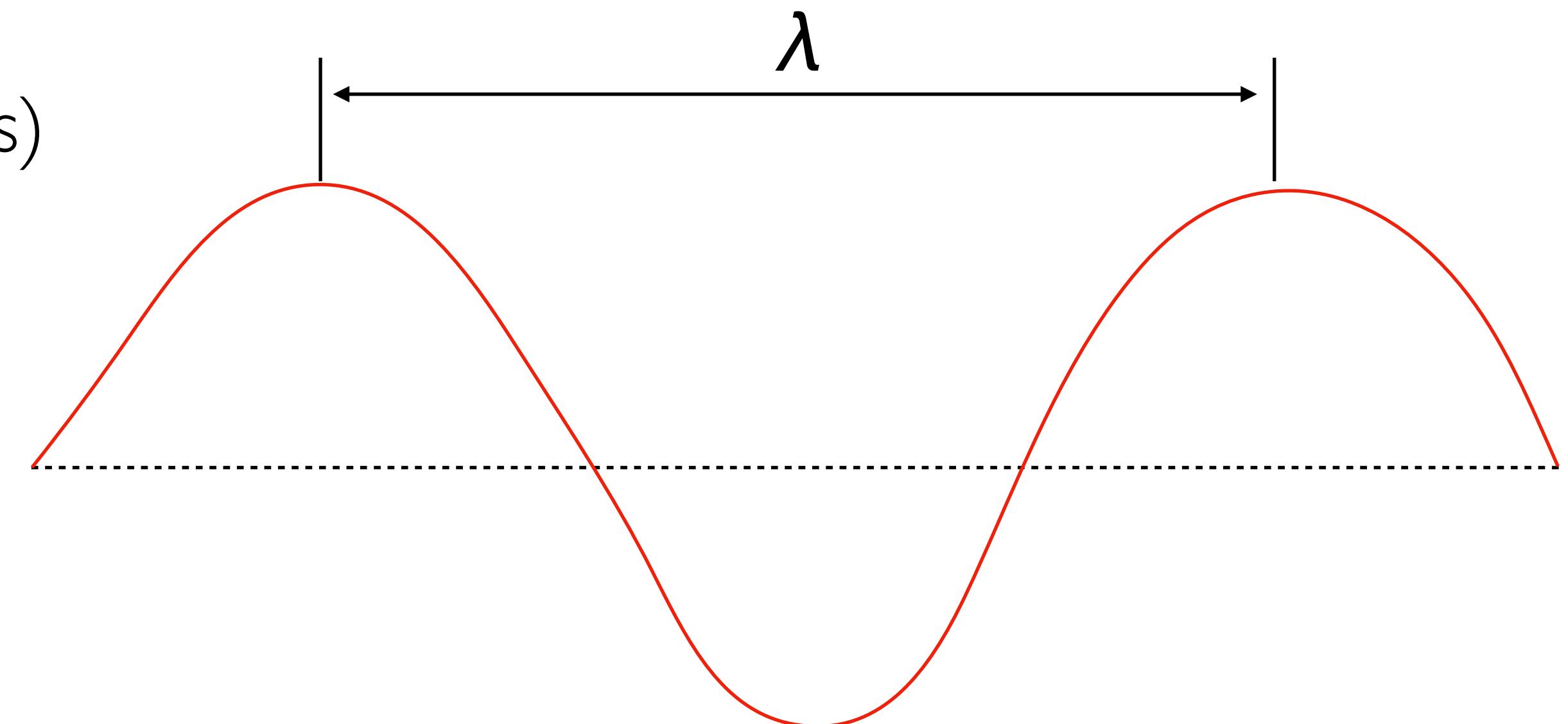
- Frequency: number of cycles per second
 - Wavelength: length of each cycle (in meters)
-
- c is the speed of light
 - f is frequency
 - λ is wavelength

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**

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

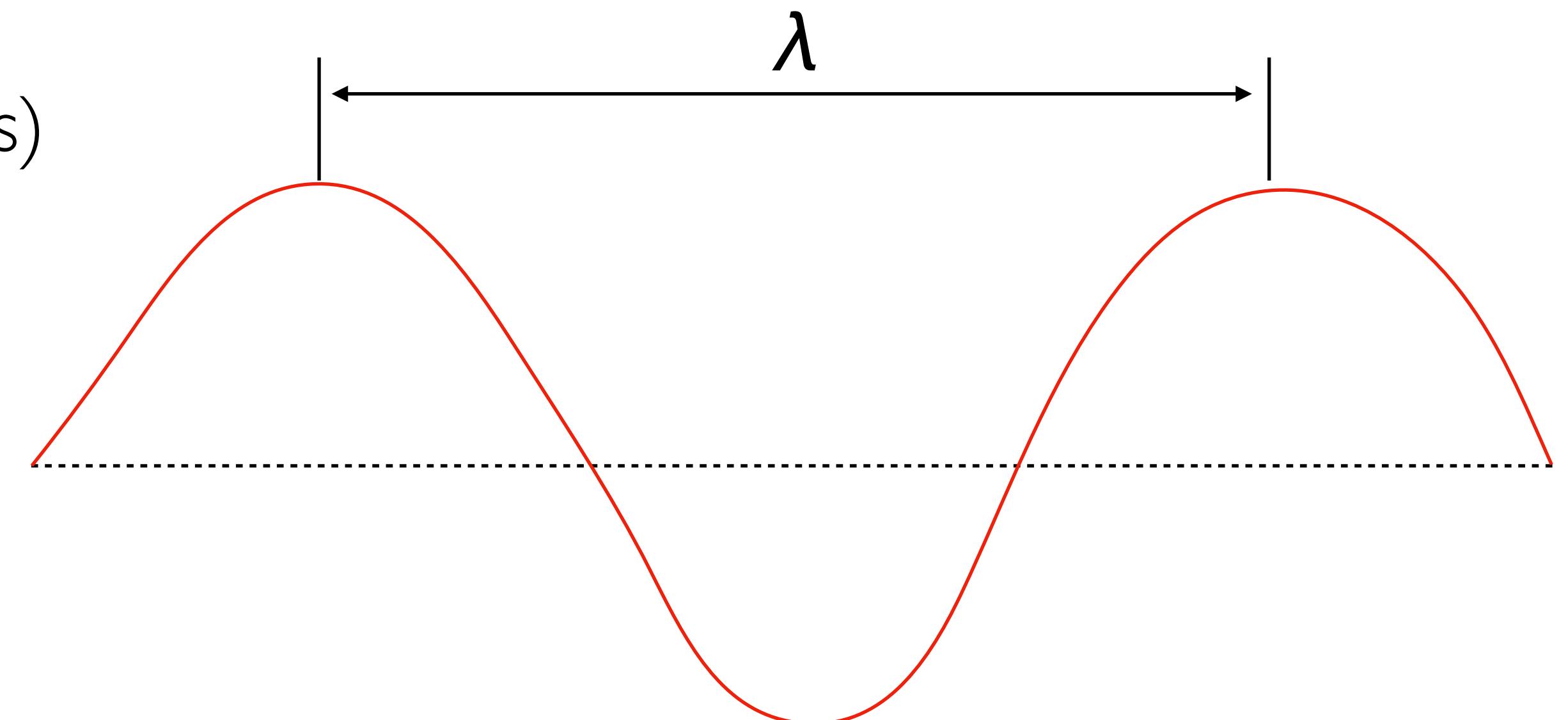


Frequency or Wavelength

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**
- Q: How do we share the wireless “medium”?

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

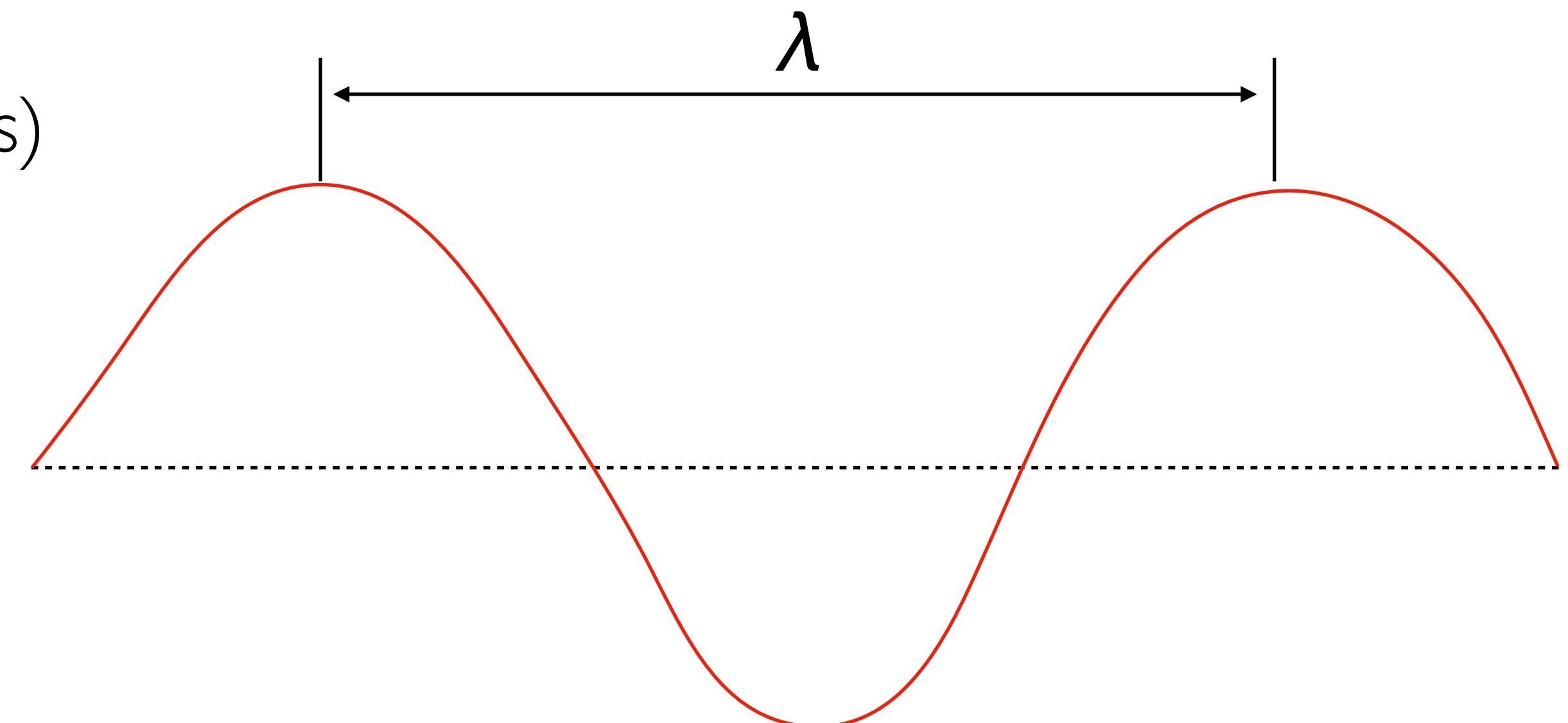


Frequency or Wavelength

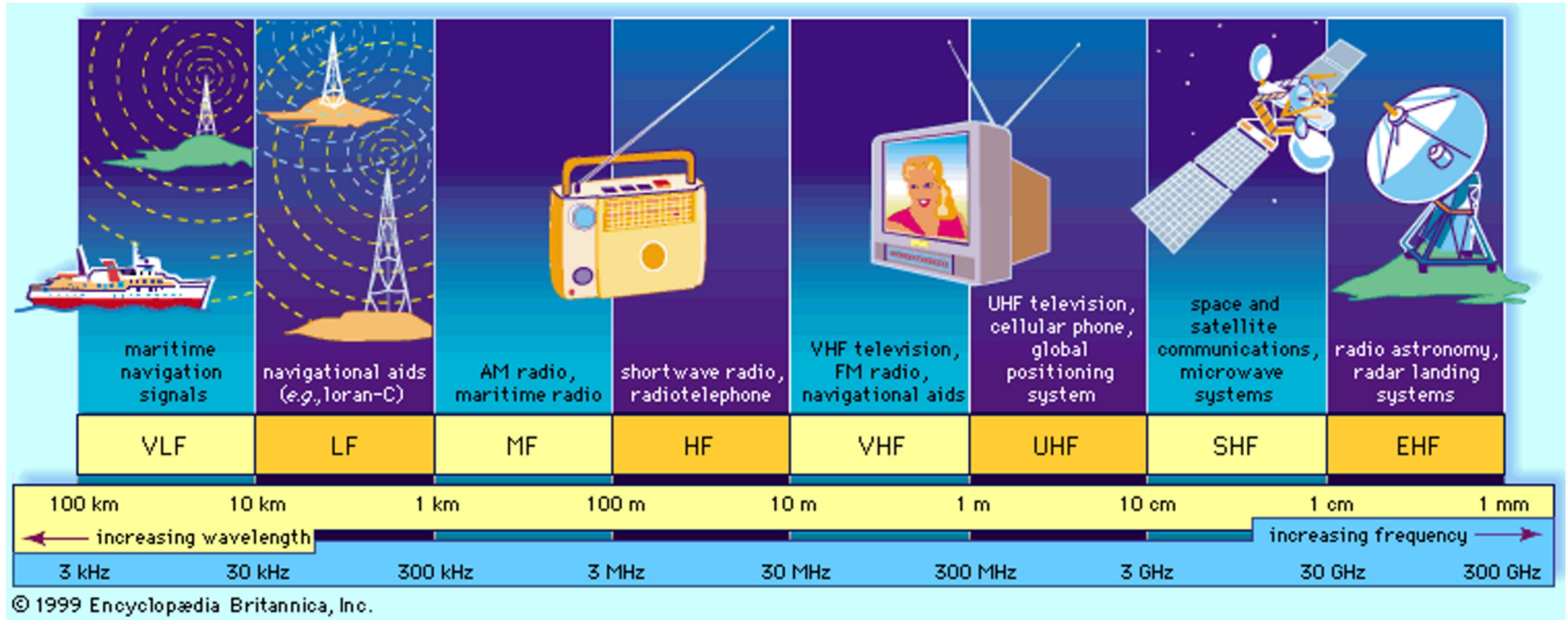
Characteristics of a wireless signal

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

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



The Wireless “Spectrum”



UNITED STATES FREQUENCY ALLOCATIONS

THE RADIO SPECTRUM



This chart is a graphic simplification of the Table of Frequency Allocations used by the FCC and NTIA. As such, it is not complete in its entirety and must change yearly to the Table of Frequency Allocations. Therefore, for complete information, users should consult the Table to determine the most current information.



Common Wireless Standards

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

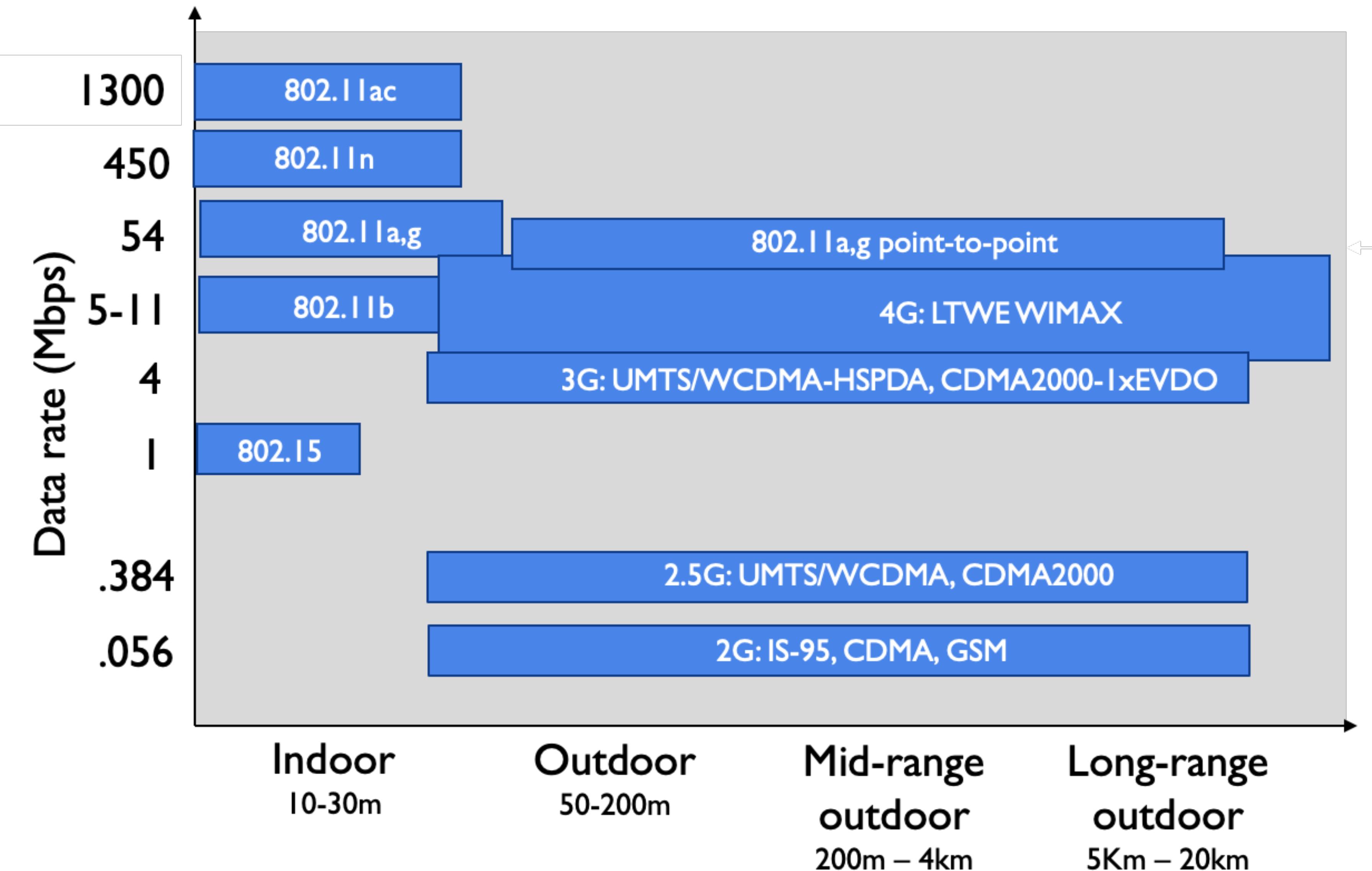
Common Wireless Standards

- 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, >1 Gbps (4x4 MIMO) (wide channels)

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, >1 Gbps (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?

What makes wireless different?

- Consider specific frequency band / wireless technology
 - What makes it different from wired technologies?

What makes wireless different?

- Consider specific frequency band / wireless technology
 - What makes it different from wired technologies?
- A lot really...

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

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

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

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

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

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

Path Loss / Path Attenuation

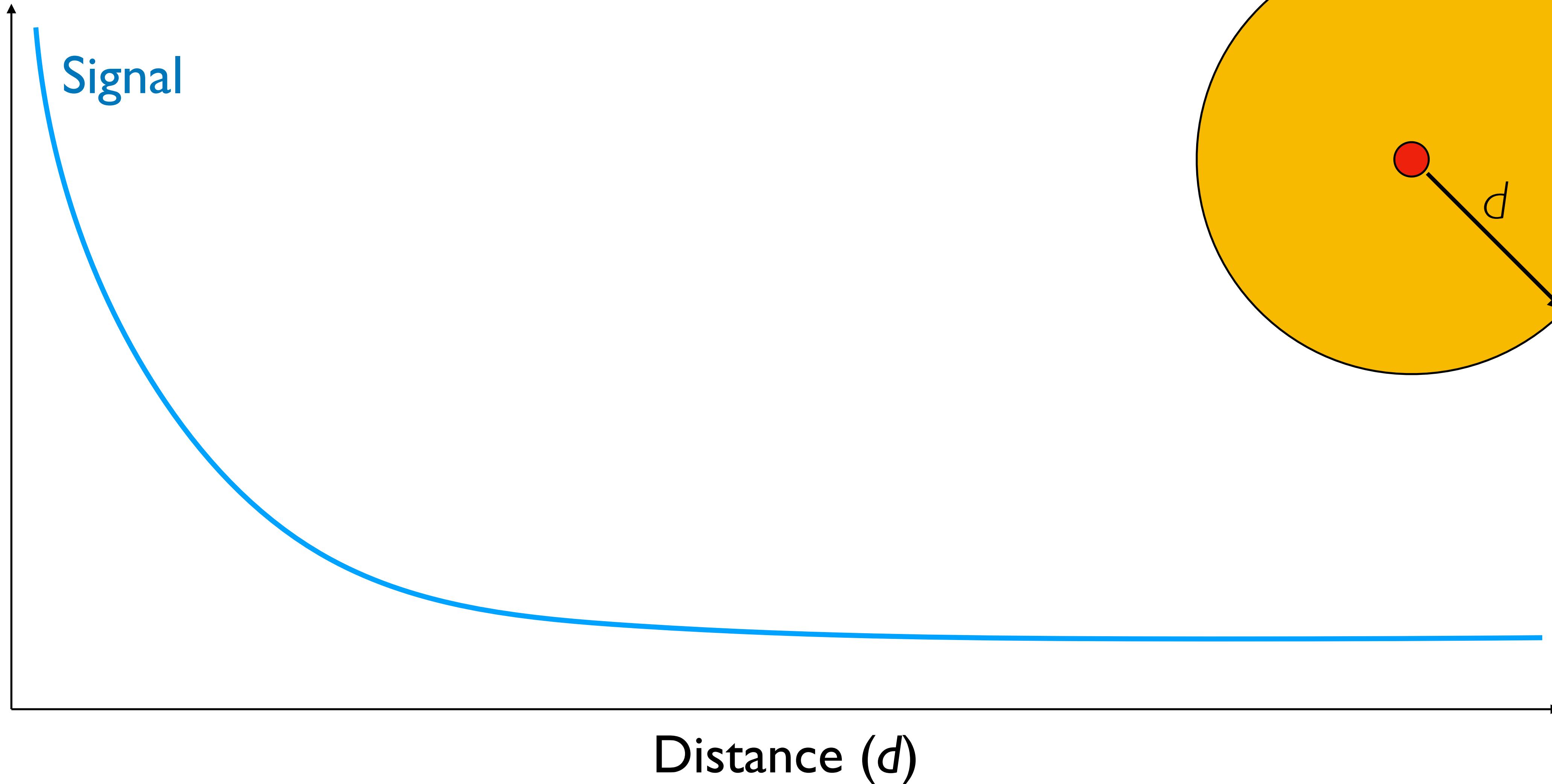
- Free space path loss: Reduction in power density of an EM wave as it propagates through space
 - d = distance
 - c is the speed of light
 - f is frequency
 - λ is wavelength
- Other reasons (not include in FSPL)
 - Reflection, Diffraction, Absorption
 - Terrain counters (urban, rural, vegetation)
 - Humidity
 - ...

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

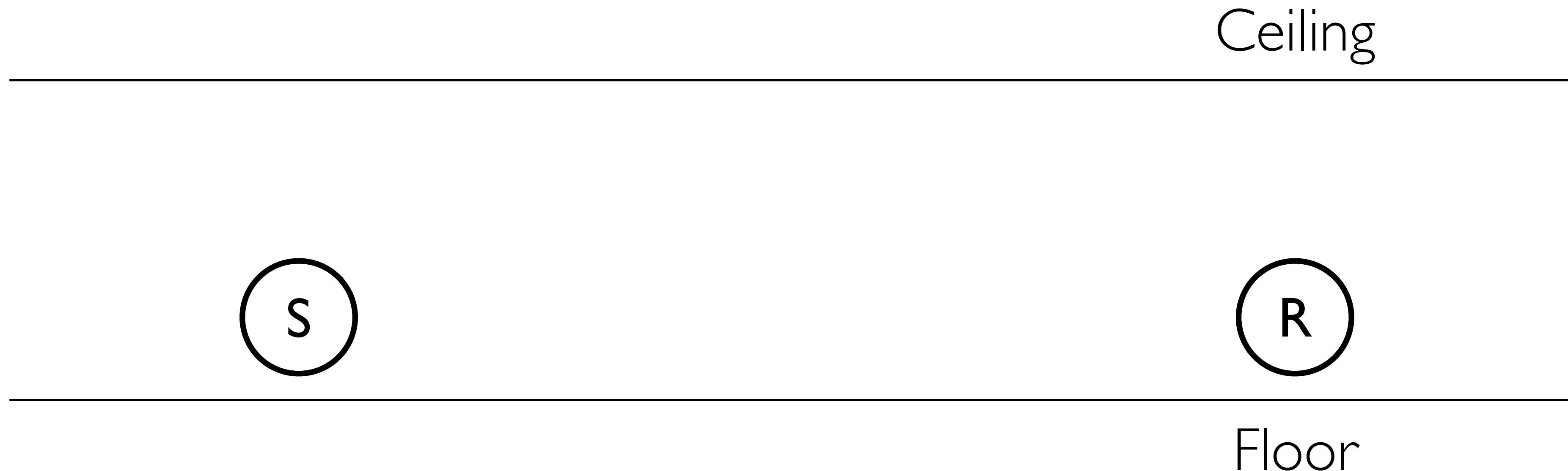
Path Attenuation



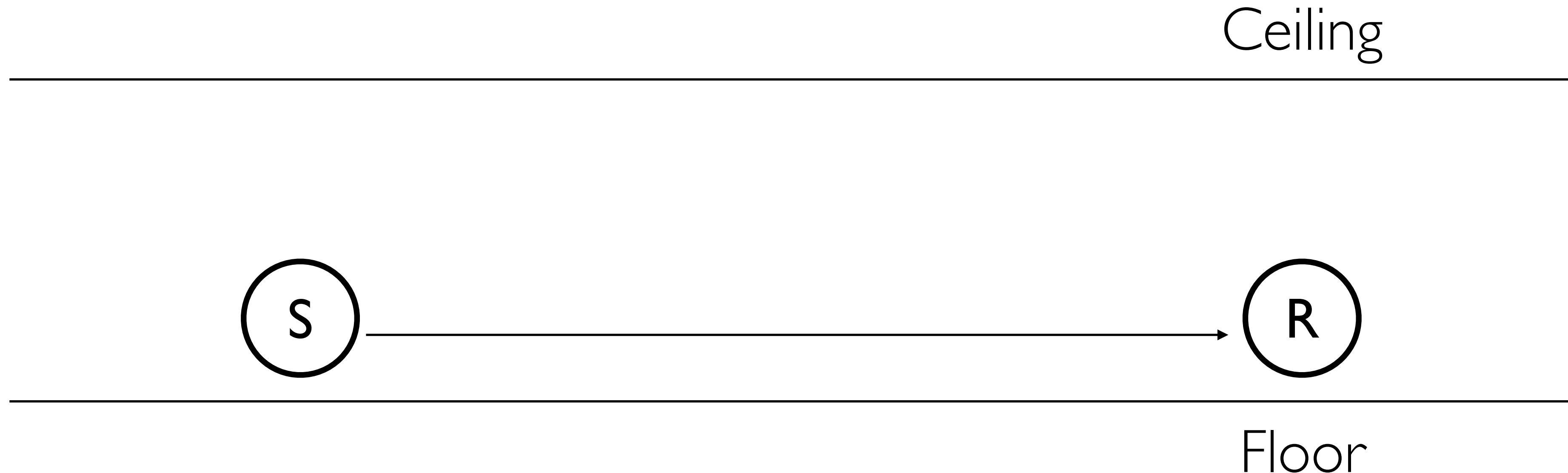
Path Attenuation



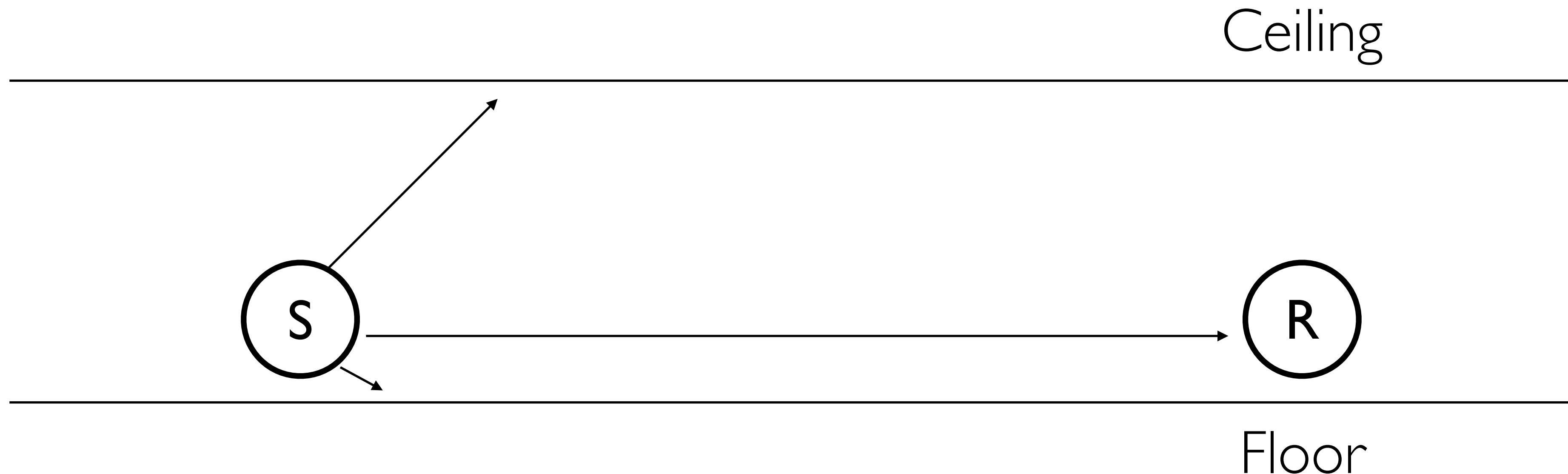
Multipath Effects



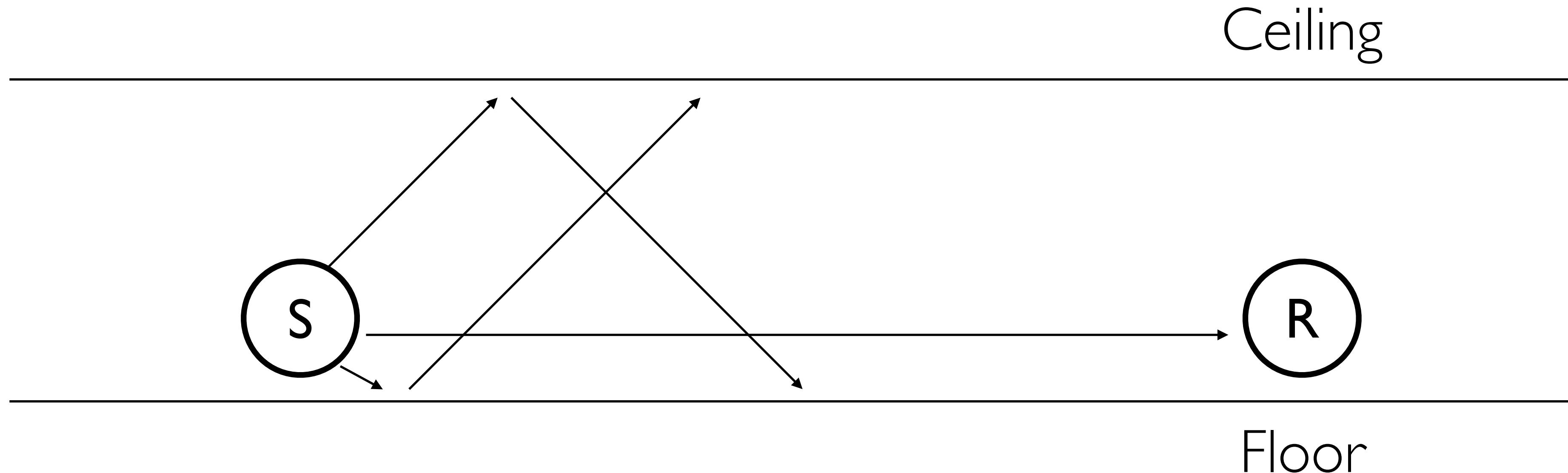
Multipath Effects



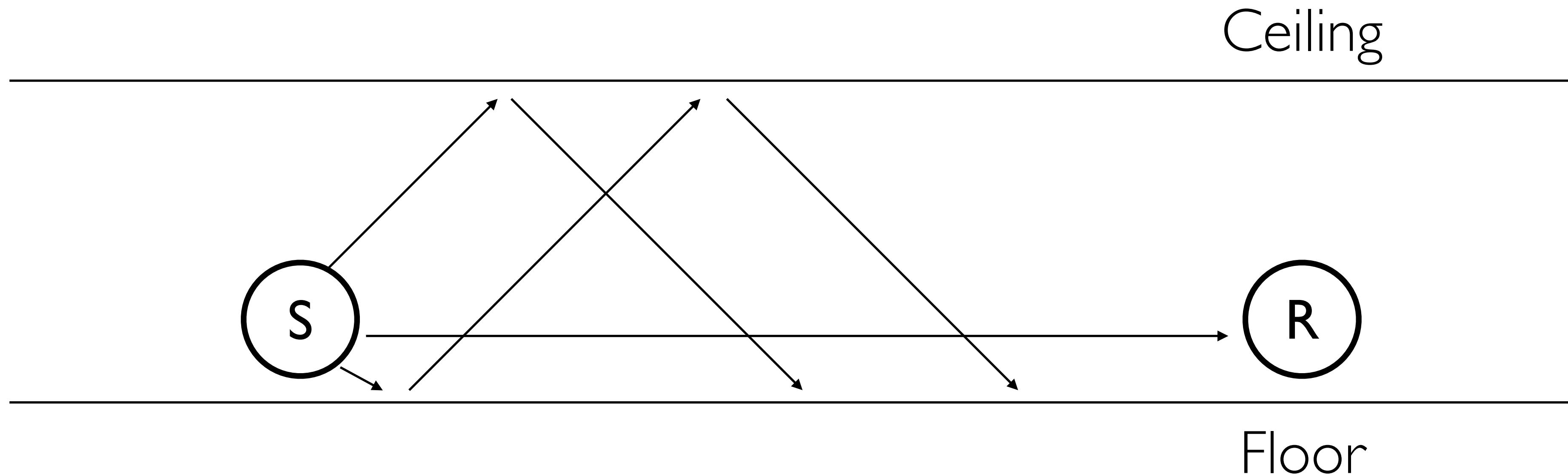
Multipath Effects



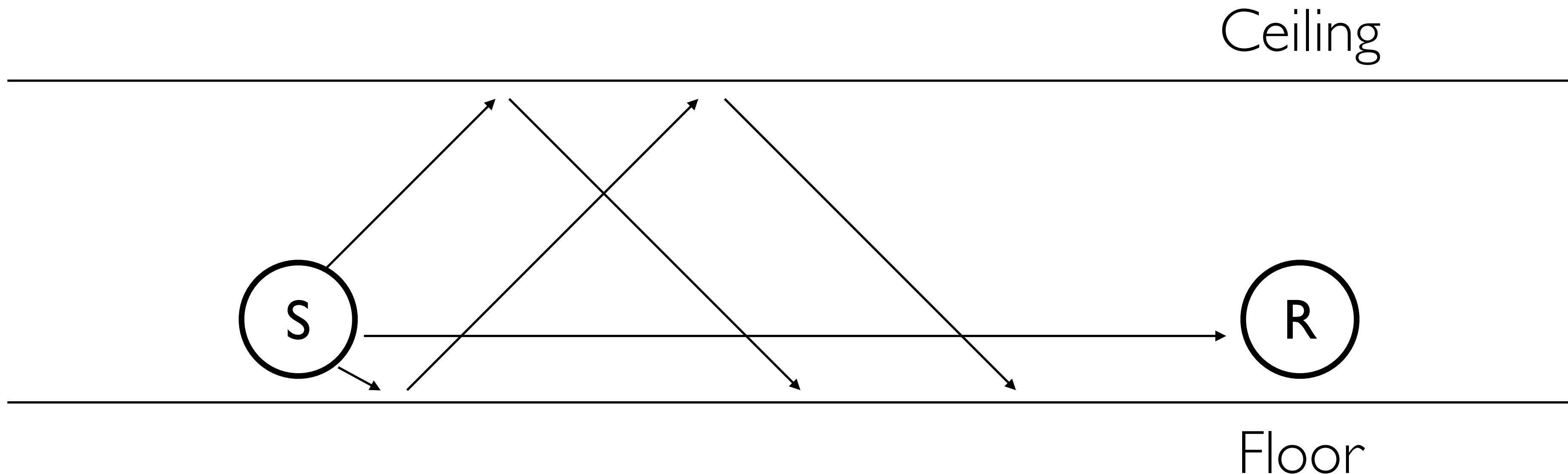
Multipath Effects



Multipath Effects

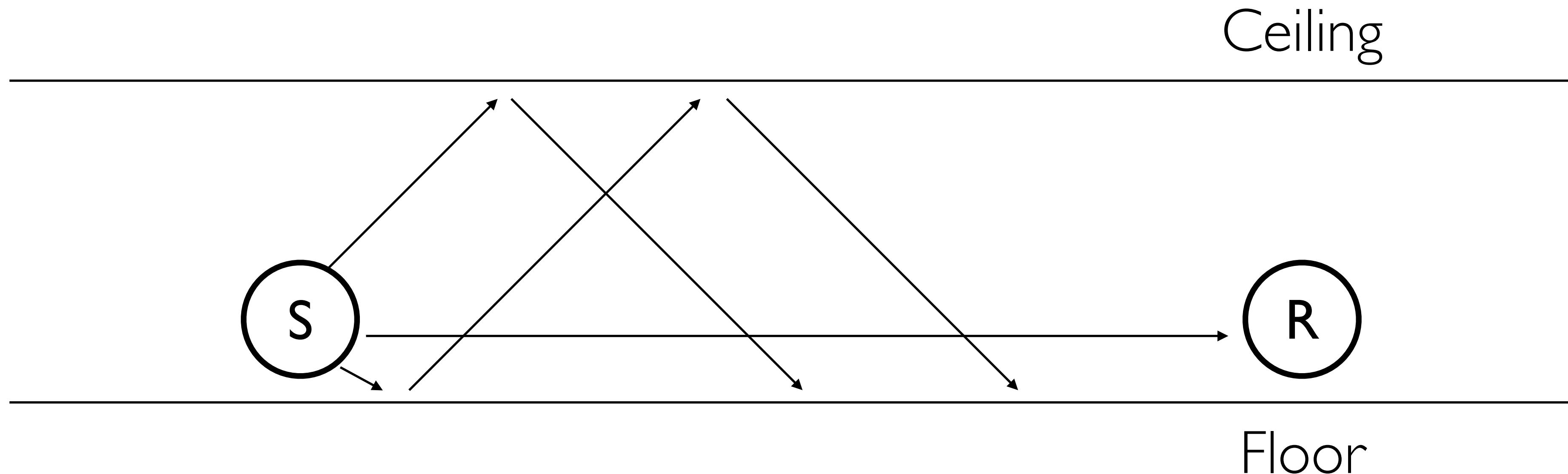


Multipath Effects



- Signals bounce off surface and interfere with one another

Multipath Effects



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

Interference

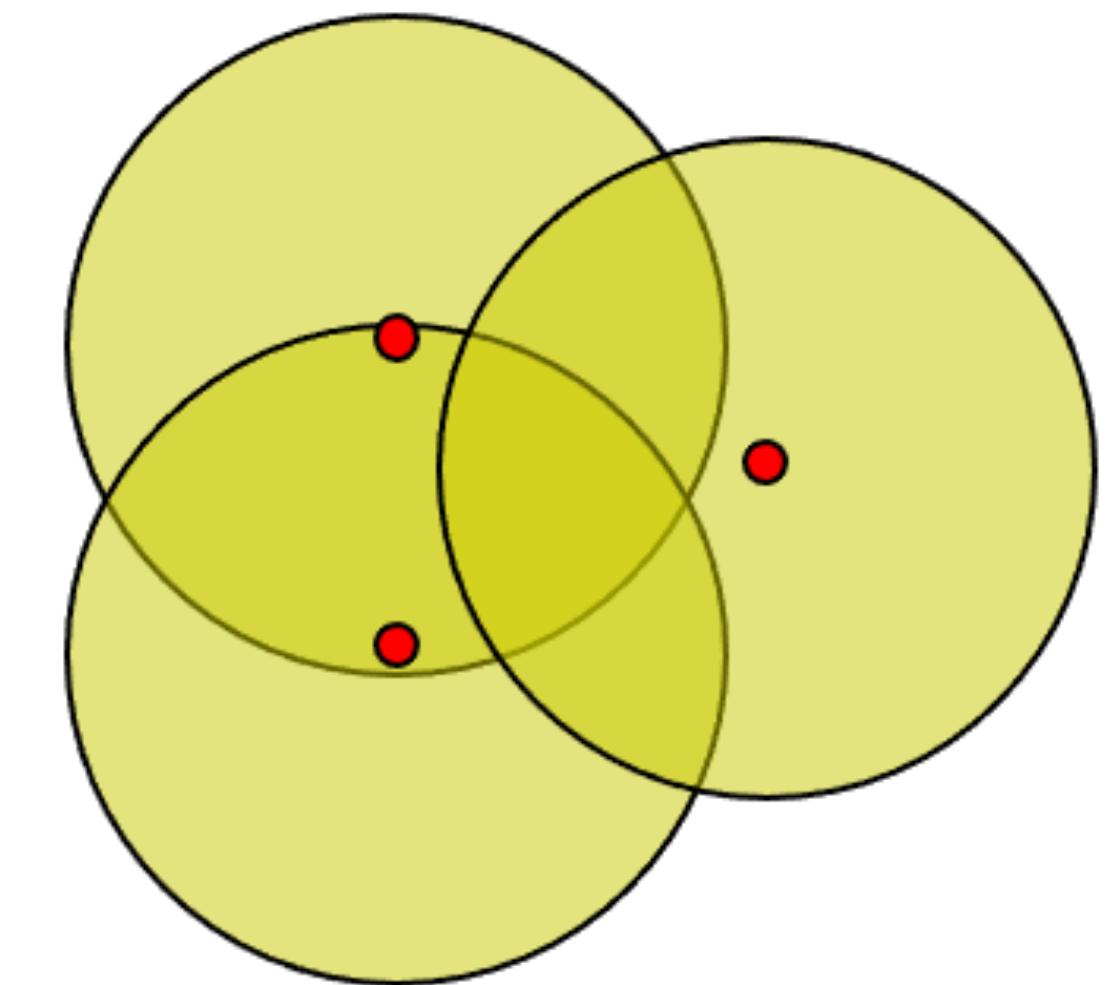
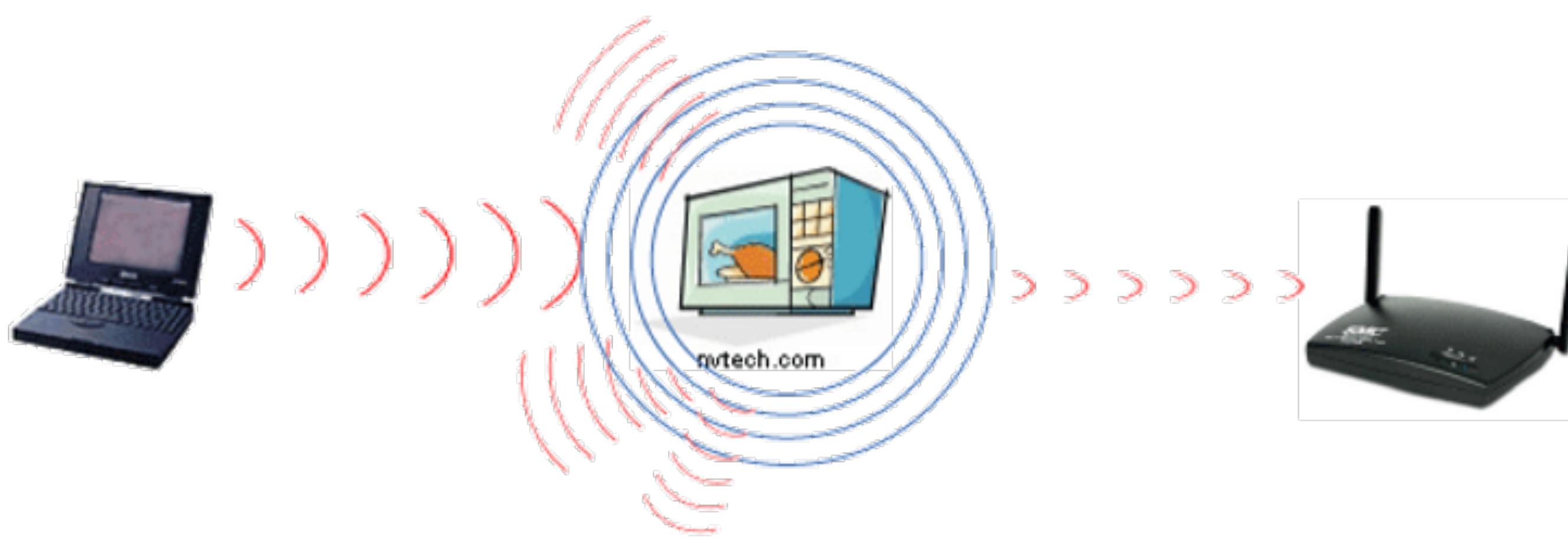
Interference

- External Interference
 - E.g., microwave oven blocks your signal

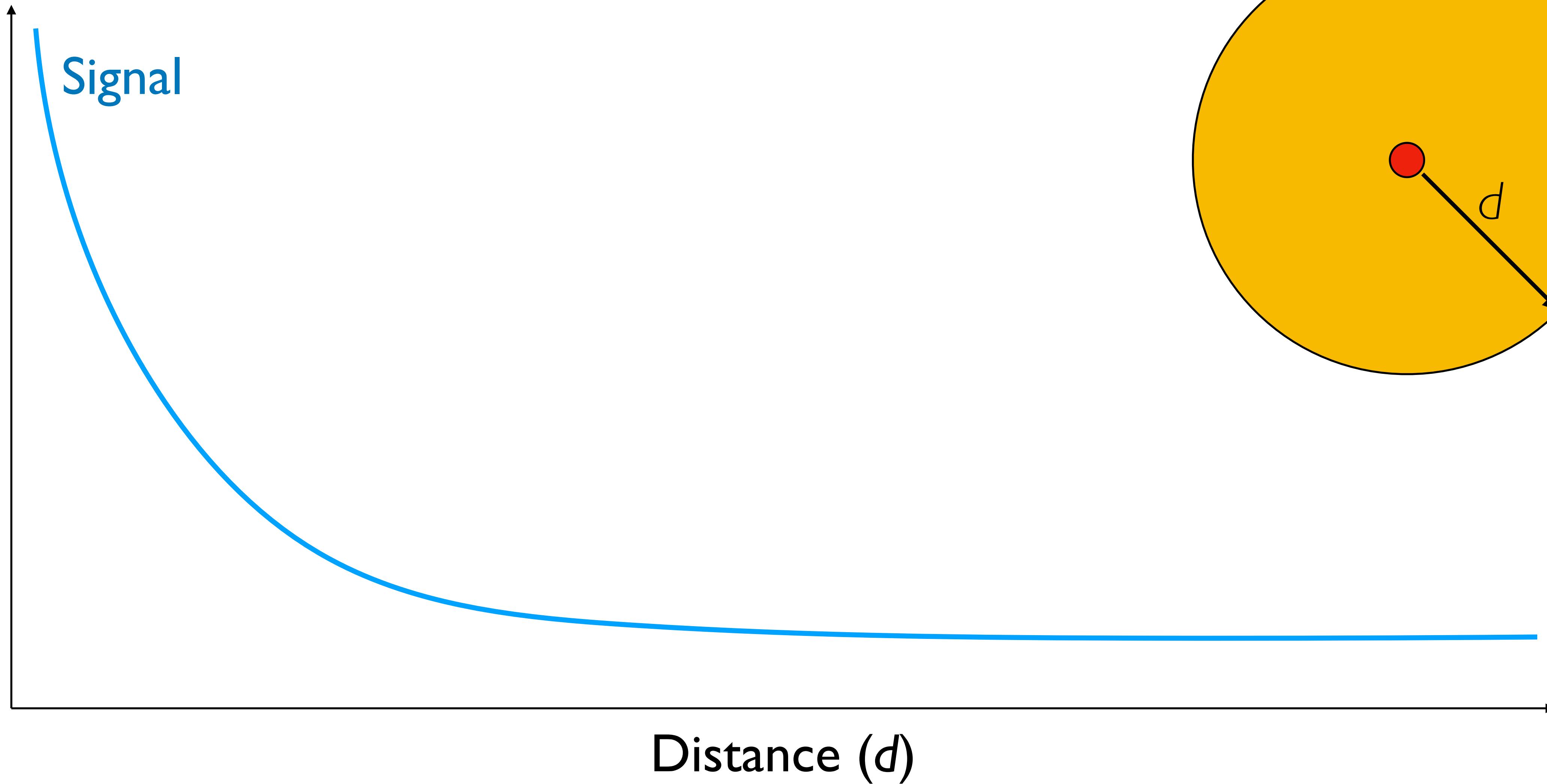


Interference

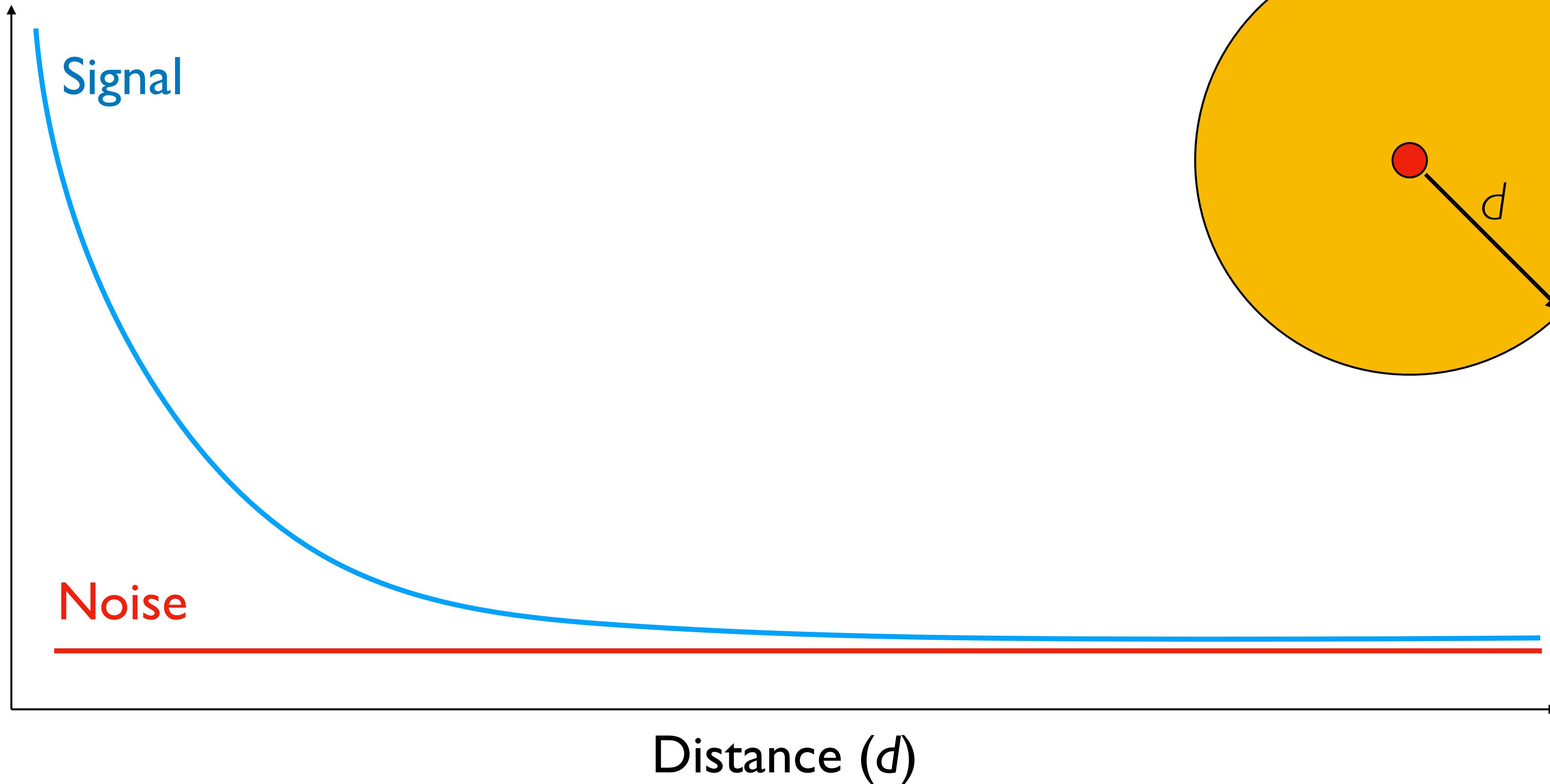
- **External Interference**
 - E.g., microwave oven blocks your signal
- **Internal Interference**
 - Nodes of the same network within range of each other collide with one another's transmissions



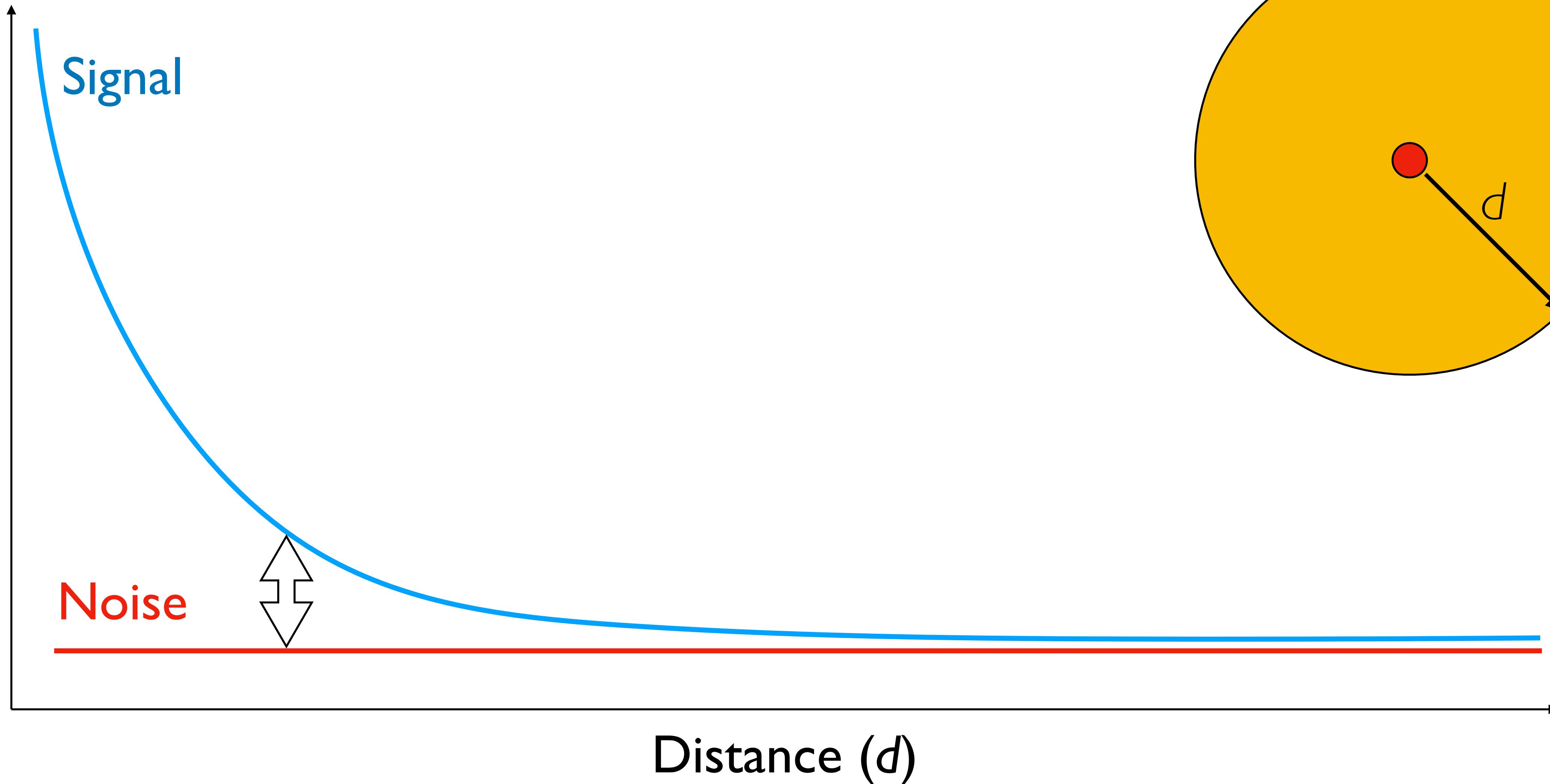
Signal and noise



Signal and noise



Signal and noise



Signal-to-noise Ratio (SNR)

Signal-to-noise Ratio (SNR)

- Relative ratio of:

Signal-to-noise Ratio (SNR)

- **Relative ratio of:**
 - Strength of received signal, and,

Signal-to-noise Ratio (SNR)

- **Relative ratio of:**
 - Strength of received signal, and,
 - Background noise

Signal-to-noise Ratio (SNR)

- Relative ratio of:
 - Strength of received signal, and,
 - Background noise
- Larger SNR = easier to extract signal from 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)

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?

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?

Signal-to-noise Ratio (SNR)

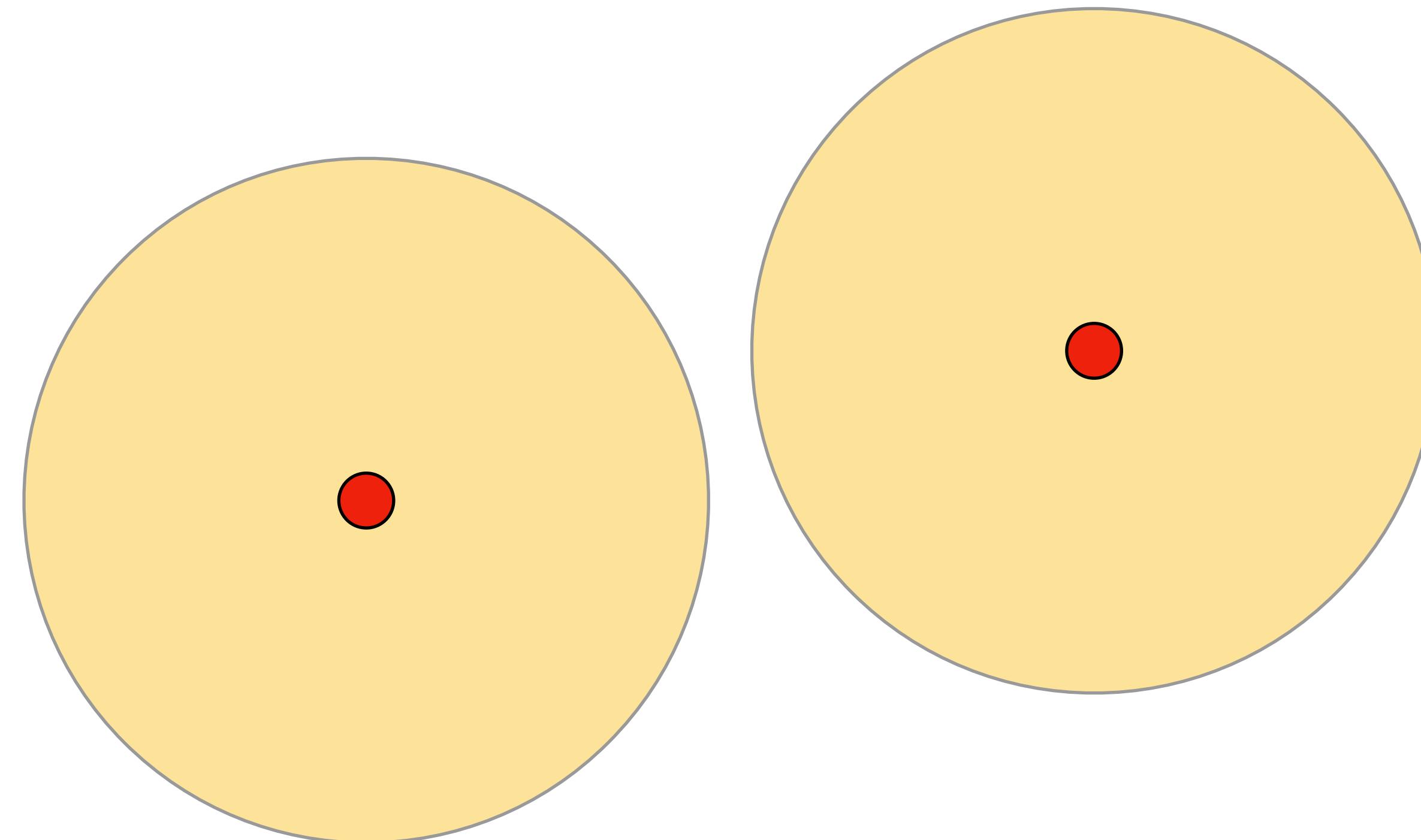
- Why isn't increasing SNR always a good idea?
 - Increased signal strength requires more power

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

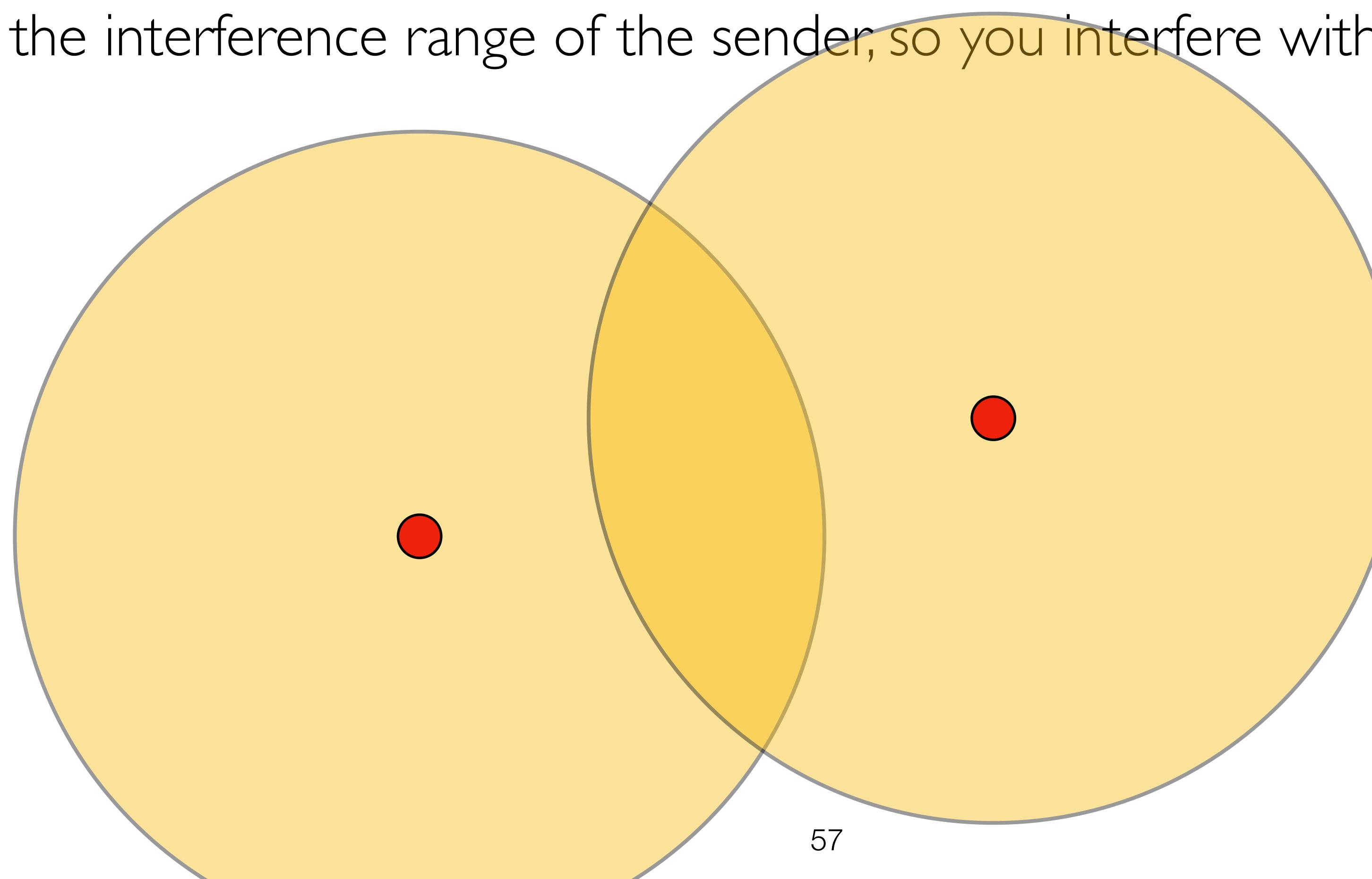
Signal-to-noise Ratio (SNR)

- 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



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



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?

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?