CSE127, Computer Security

Network Security I

UC San Diego

Housekeeping

General course things to know

- PA3 is due tonight, godspeed!
- Midterm is **Thursday,** during class hours, location **Center Hall 109**
    - Class topics will go through web security (last Thursday) and include PA3 material
    - One sheet of paper front and back is allowed as a “cheatsheet”
    - You **must** bring photo ID to the exam
    - Please try to come a little early (7:50am would be good) so you have the full 80 minutes!
- PA4 will be released *after* midterm
    - CTF-style assignment, very little instruction, it’s fun

Word on the midterm…

Structure + preparation

- MCQ / True or False… "select all that apply" (~40 points)
    - -1 for any incorrect response, so to disincentivize guessing (but you can't go lower than 0 for any one question)
    - Based near entirely on lecture material
- Short answer questions; also based on lecture material (~10 points)
- Two longer PA style questions (~40 points)
    - PA1 + PA2 grouped into one (since PA1 is kind of a precursor to PA2)
- Best way to study: **go over the slides, go over the PAs.**

Previously on CSE127…

Applications, Systems, and the Web

- We’ve talked a lot about software, systems, and one instance of a popular system (the web)…
  - Common security issues: mixing code and data
  - Privilege, isolation, how systems do this (or fail to do this) today…
  - Web tracking, privacy, how we’re sort of being watched all the time…
- Now, we’re going to shift gears and plug all of these systems together via the *network*
  - What could go wrong?

Today's lecture — The Internet

Learning Objectives

- Understand how the Internet works (at a 10,000 foot view), the players involved, and the general duct tape tape that keeps the pipes working
- Learn about the OSI network model, understand the hierarchy of abstraction that runs the modern Internet, and TCP
- Talk about the basic security guarantees of the network (hint: there are not many) and what we've done so far to "fix" things

But first, dystopia!

https://www.youtube.com/watch?v=OheUzrXsKrY

The Internet

Understanding the basics

- What do you know about the Internet?

Understanding the basics

- What is the Internet?
    - “A global computer network consisting of interconnected networks using standard communication protocols.”
- My version: **a network of networks that lets computers talk to one another.**

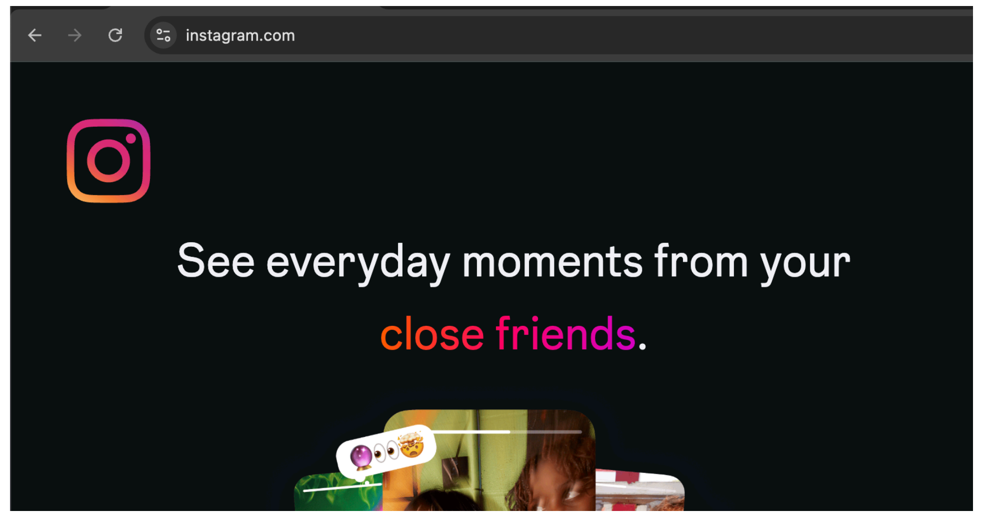
Understanding the basics

- What is the Internet?
  - “A global computer network consisting of interconnected networks using standard communication protocols.”
- My version: **a network of networks that lets computers talk to one another.**
- My other version: **magic**

What is a network of networks?

- Let's say my computer wants to talk to instagram.com. How does it do that?

my laptop



instagram.com

See everyday moments from your close friends.

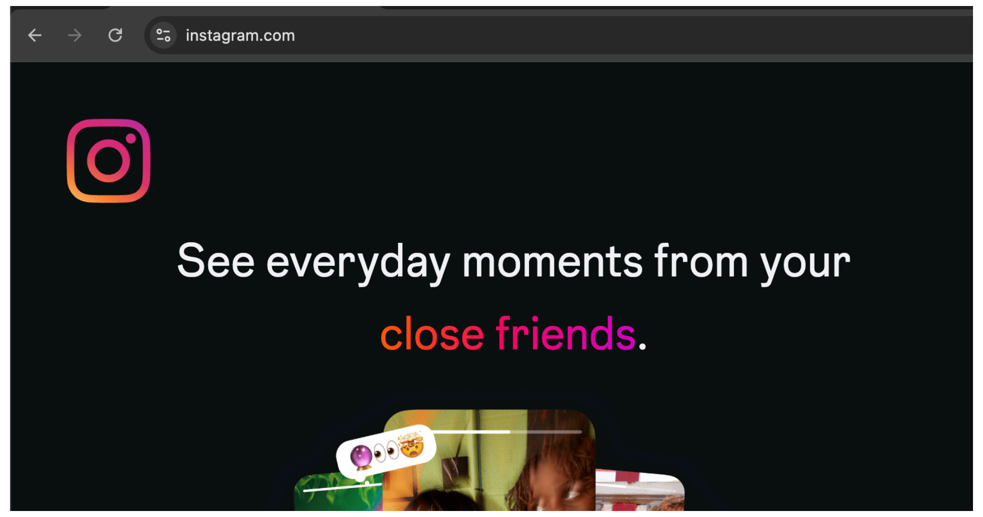
instagram.com

What is a network of networks?

- Let's say my computer wants to talk to instagram.com. How does it do that?

my laptop

router, via WiFi



instagram.com

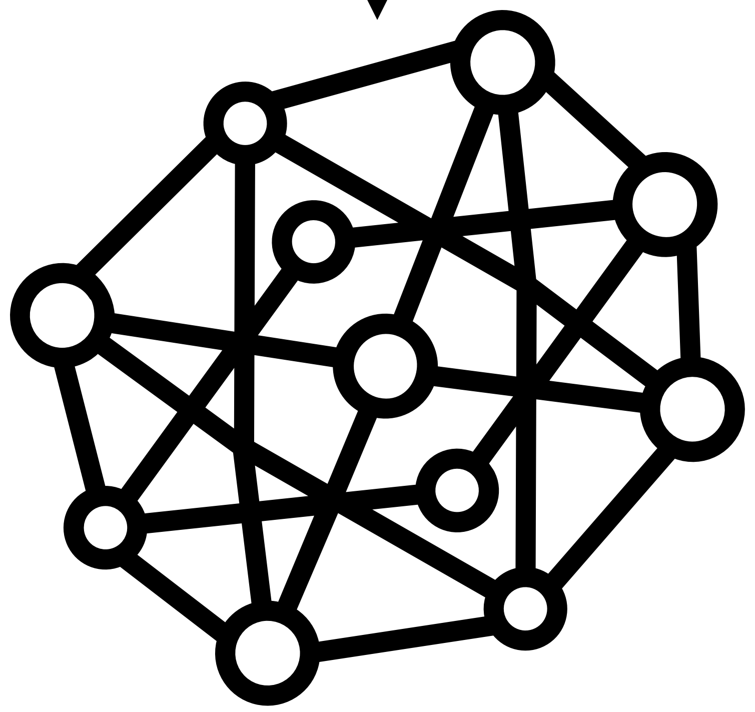
See everyday moments from your close friends.

instagram.com

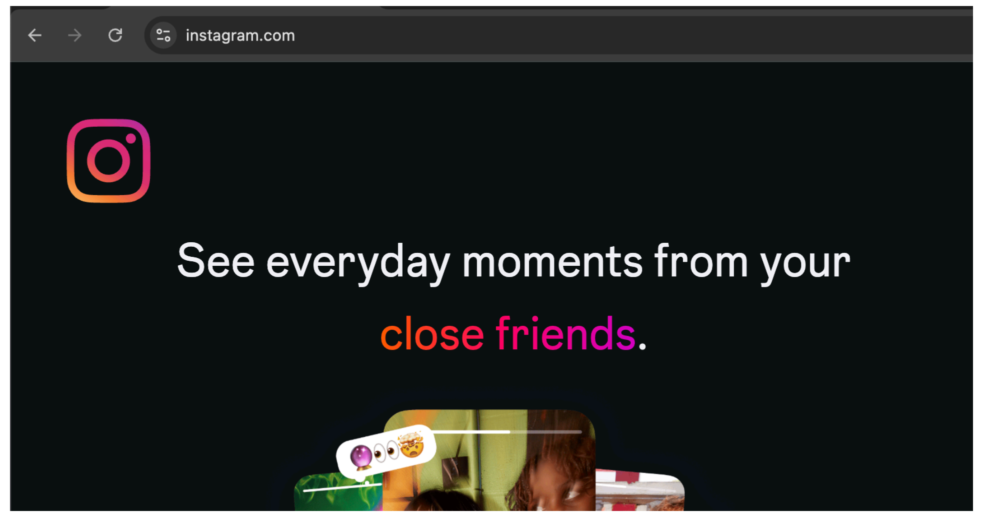
What is a network of networks?

- Let's say my computer wants to talk to instagram.com. How does it do that?

my laptop



UCSD network, via wires



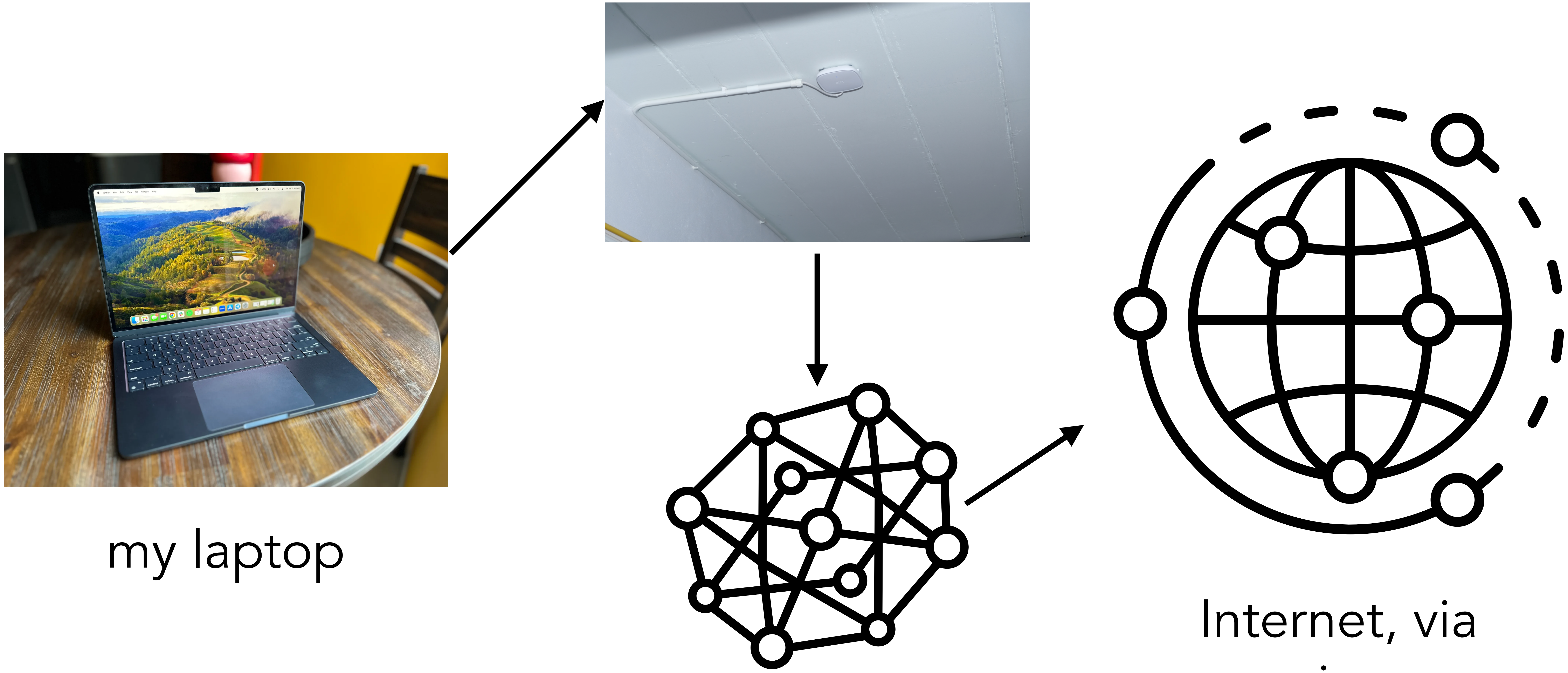
instagram.com

See everyday moments from your close friends.

instagram.com

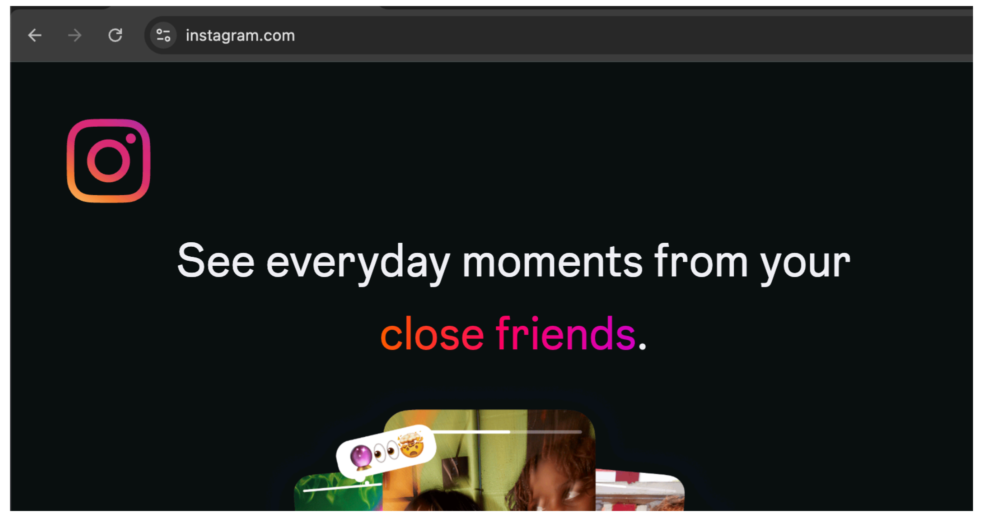
What is a network of networks?

- Let’s say my computer wants to talk to instagram.com. How does it do that?



my laptop

Internet, via wires



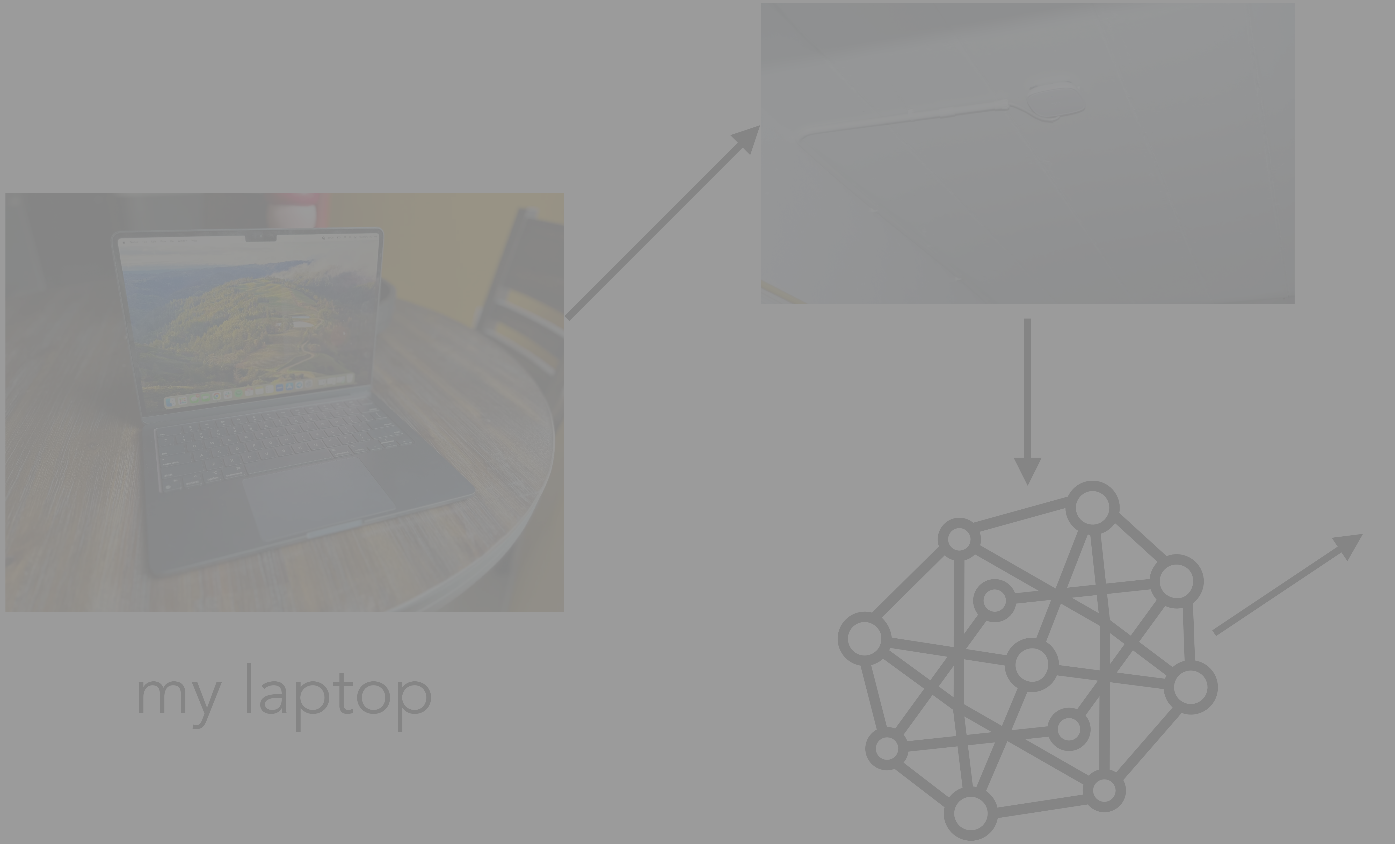
See everyday moments from your close friends.

instagram.com

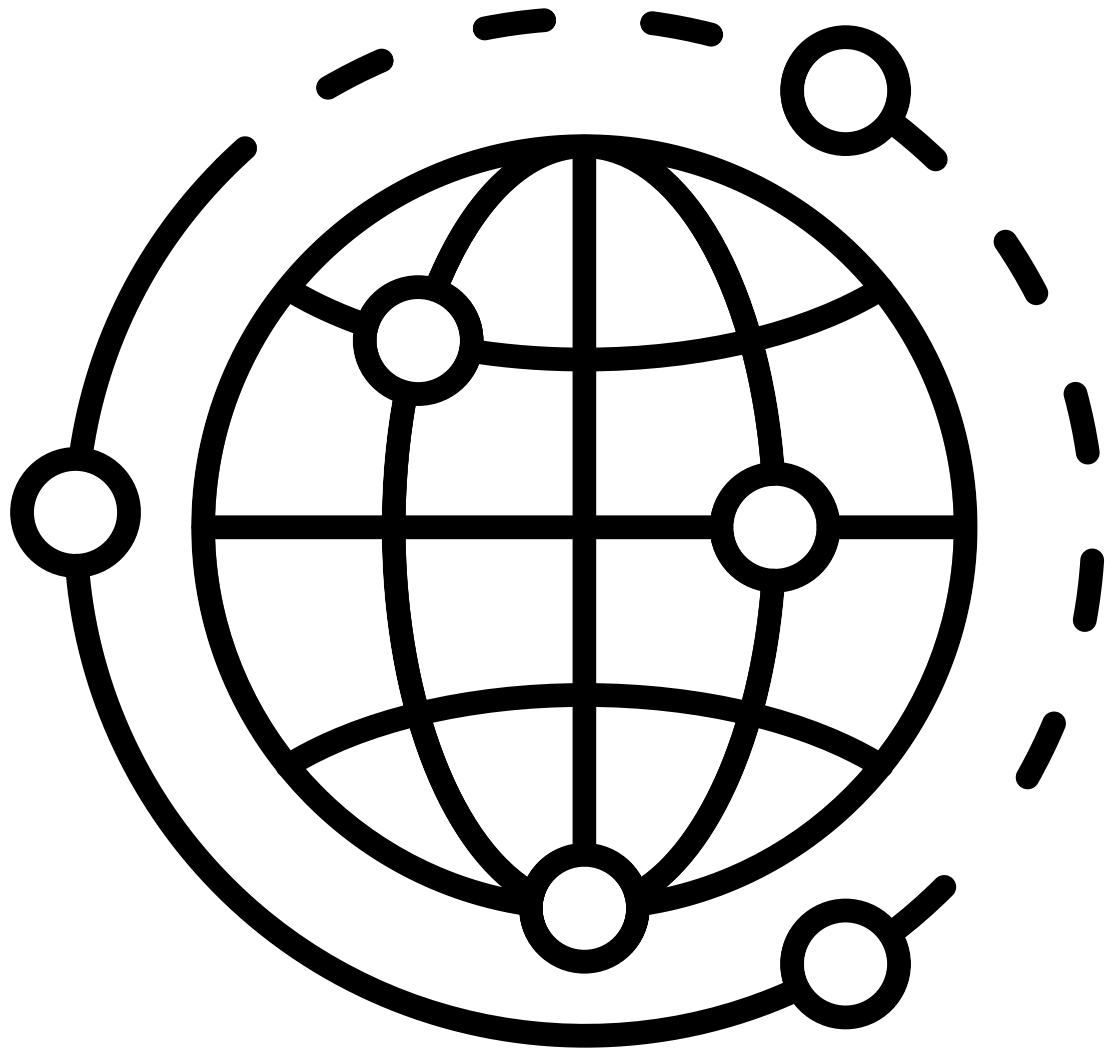
Internet Service Provider: An entity which enables computers access to the Internet

W [illegible] etworks?

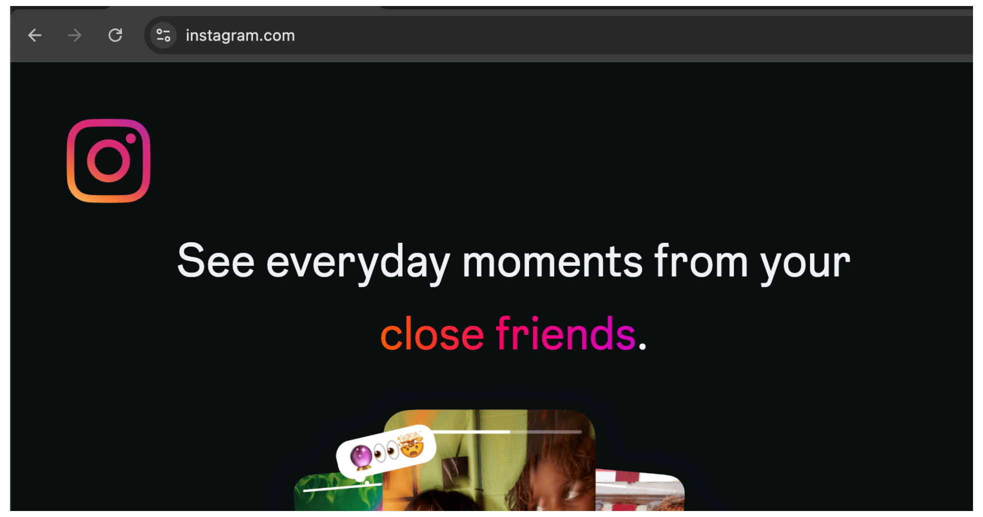
- [illegible] to instagram.com. How does it do that?



my laptop



Internet, via wires



instagram.com

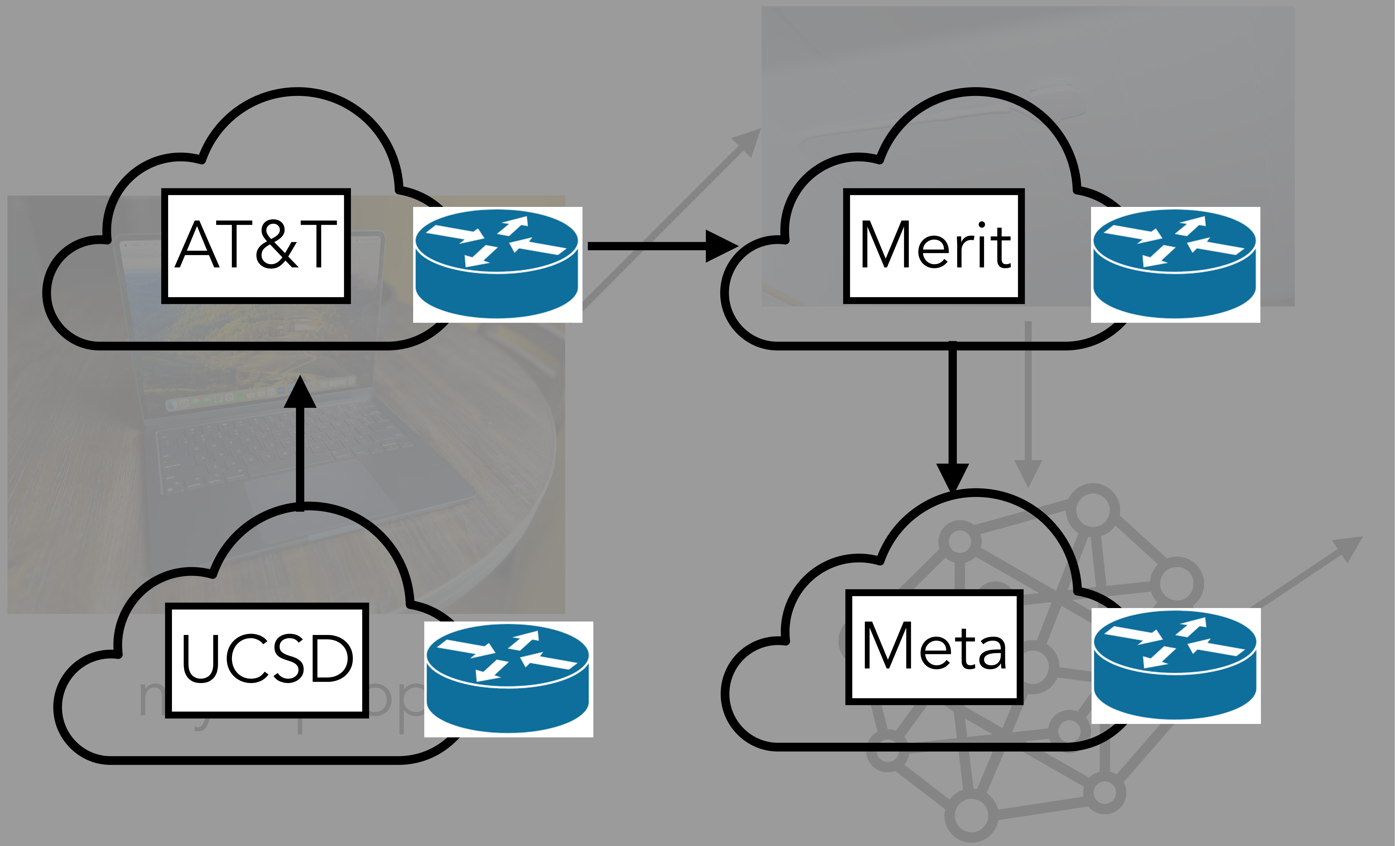
See everyday moments from your close friends.

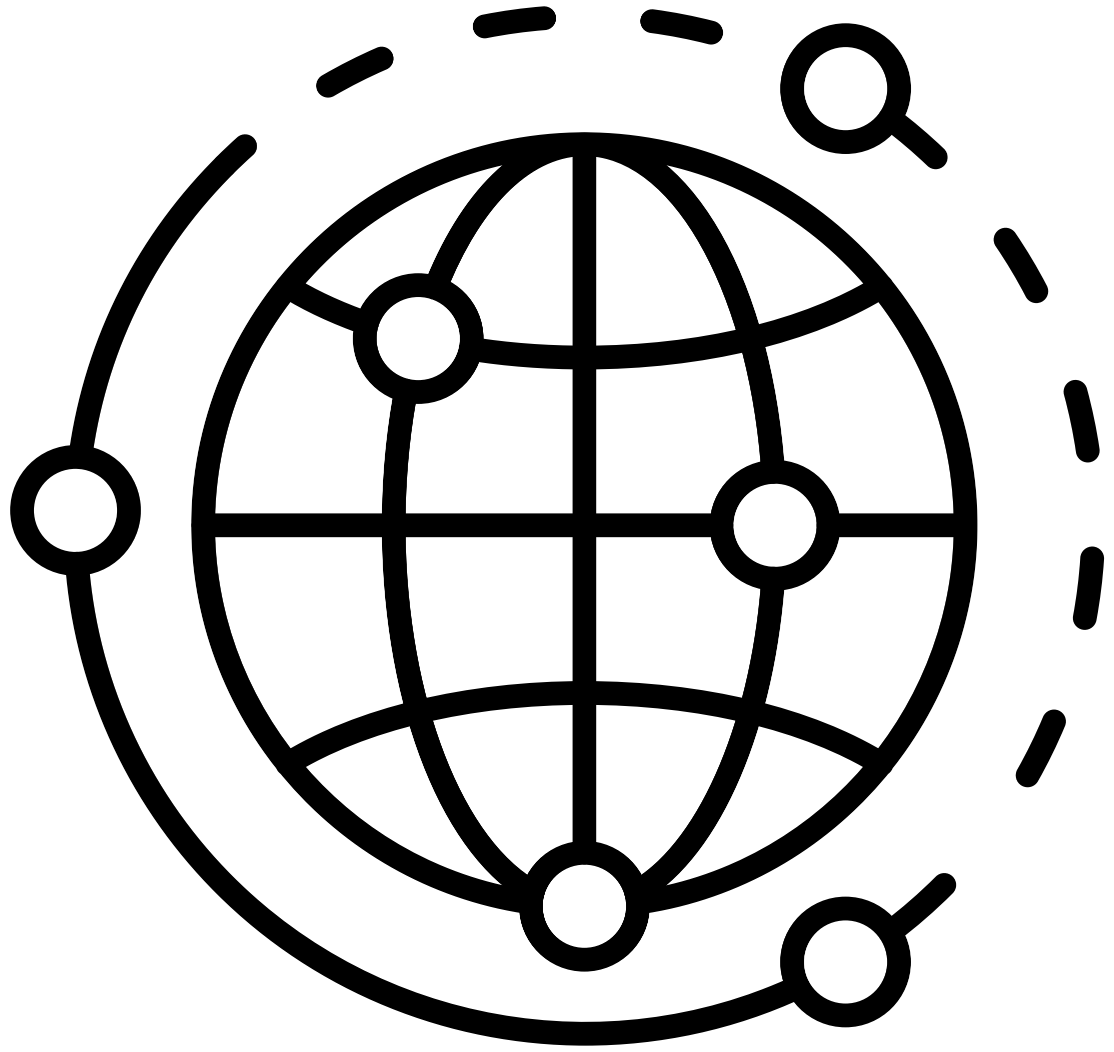
instagram.com

Internet Service Provider: An entity which enables computers access to the Internet

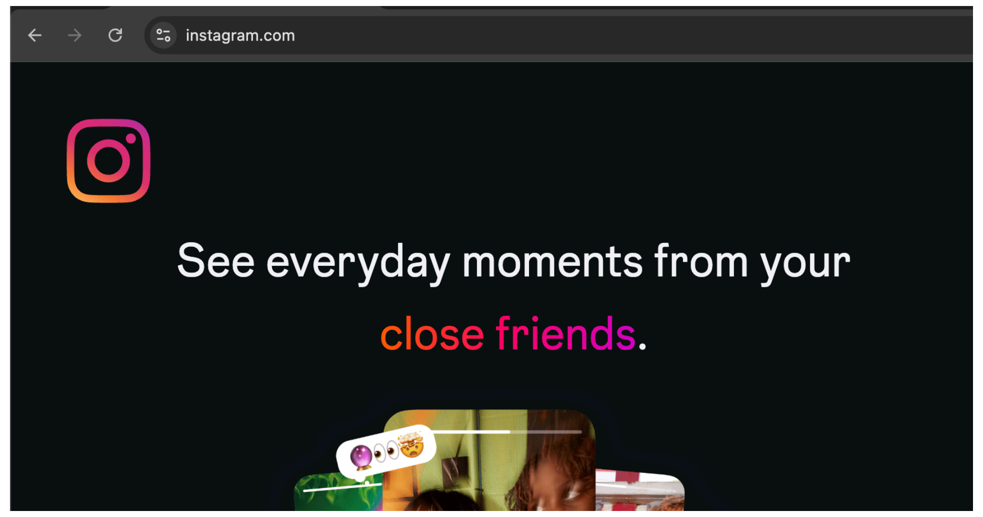
etworks?

to instagram.com. How does it do that?





Internet, via wires



See everyday moments from your close friends.

instagram.com

What is a network of networks?

- Let's say my computer wants to talk to instagram.com. How does it do that?

my laptop

Internet, via wires

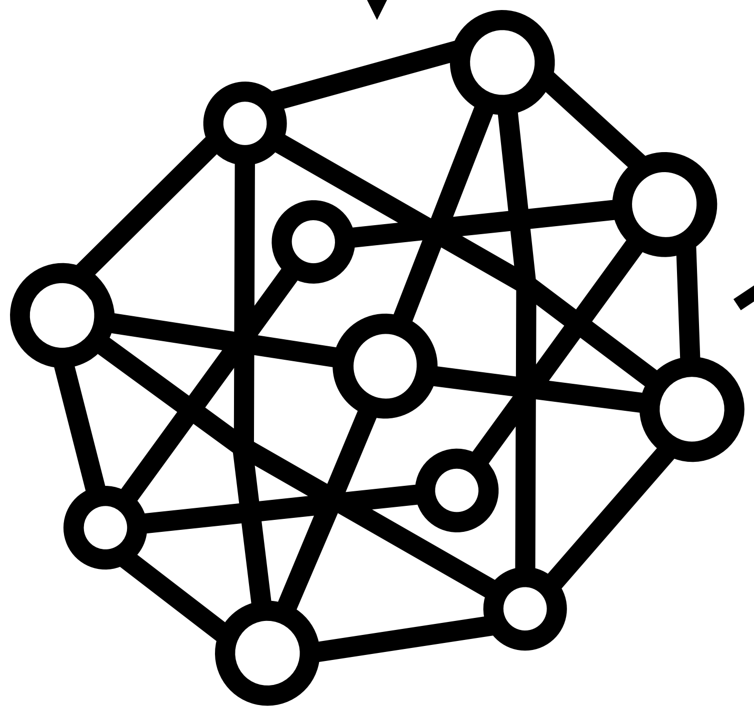
instagram.com

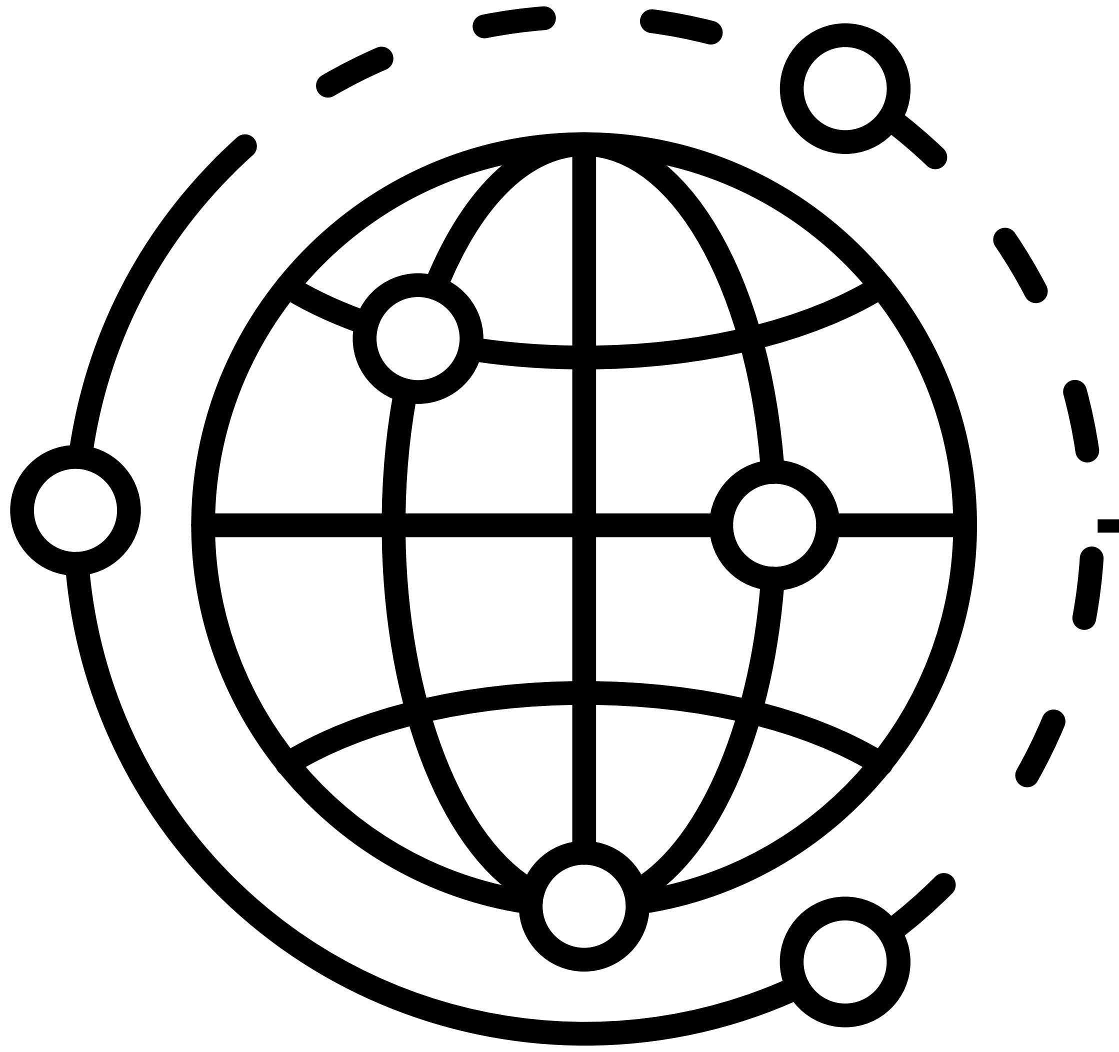
instagram.com
See everyday moments from your
close friends.

What is a network of networks?

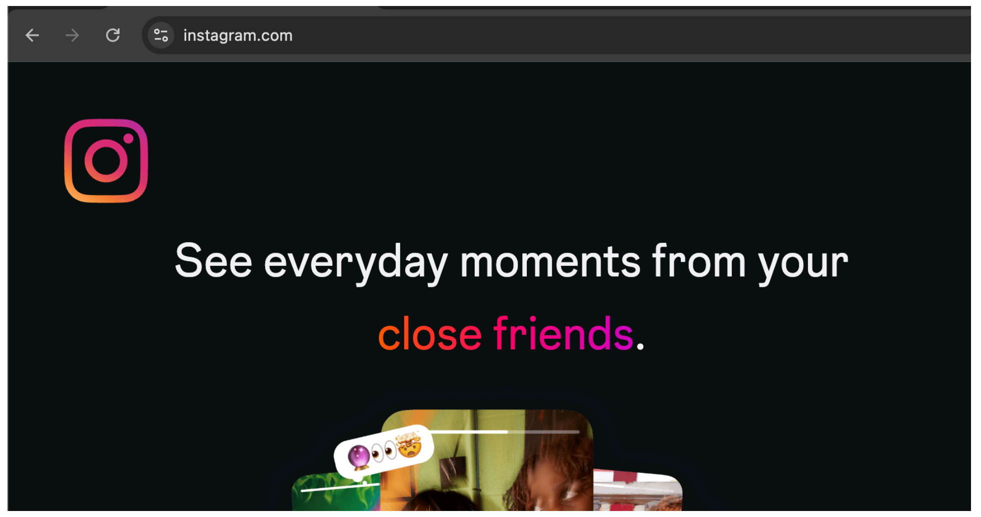
- Let's say my computer wants to talk to instagram.com. How does it do that?

my laptop





Internet, via wires



See everyday moments from your close friends.

instagram.com

This all happens in < 100ms for every request.

How it starts: a packet

- Communication on the Internet is constructed of discrete, *self-addressed* chunks of data: **packets**
  - Networking uses a lot of terms of packets in different contexts… e.g., packet, frame, segment, datagram, cell… sorry
- In contrast to *circuits*, which is the old-school way of communicating between two end points
  - E.g., landline phone lines are called “circuit switching”
- Circuit switching guarantees a connection, but means lines become your limited resource
  - Packet switching is more flexible, but can introduce variable delays / failures

From packets, we have… protocols

- With a group, answer the following questions:
  - What is a network *layer*?
  - What is a network protocol?
  - What is an IP address?
  - What is a port?

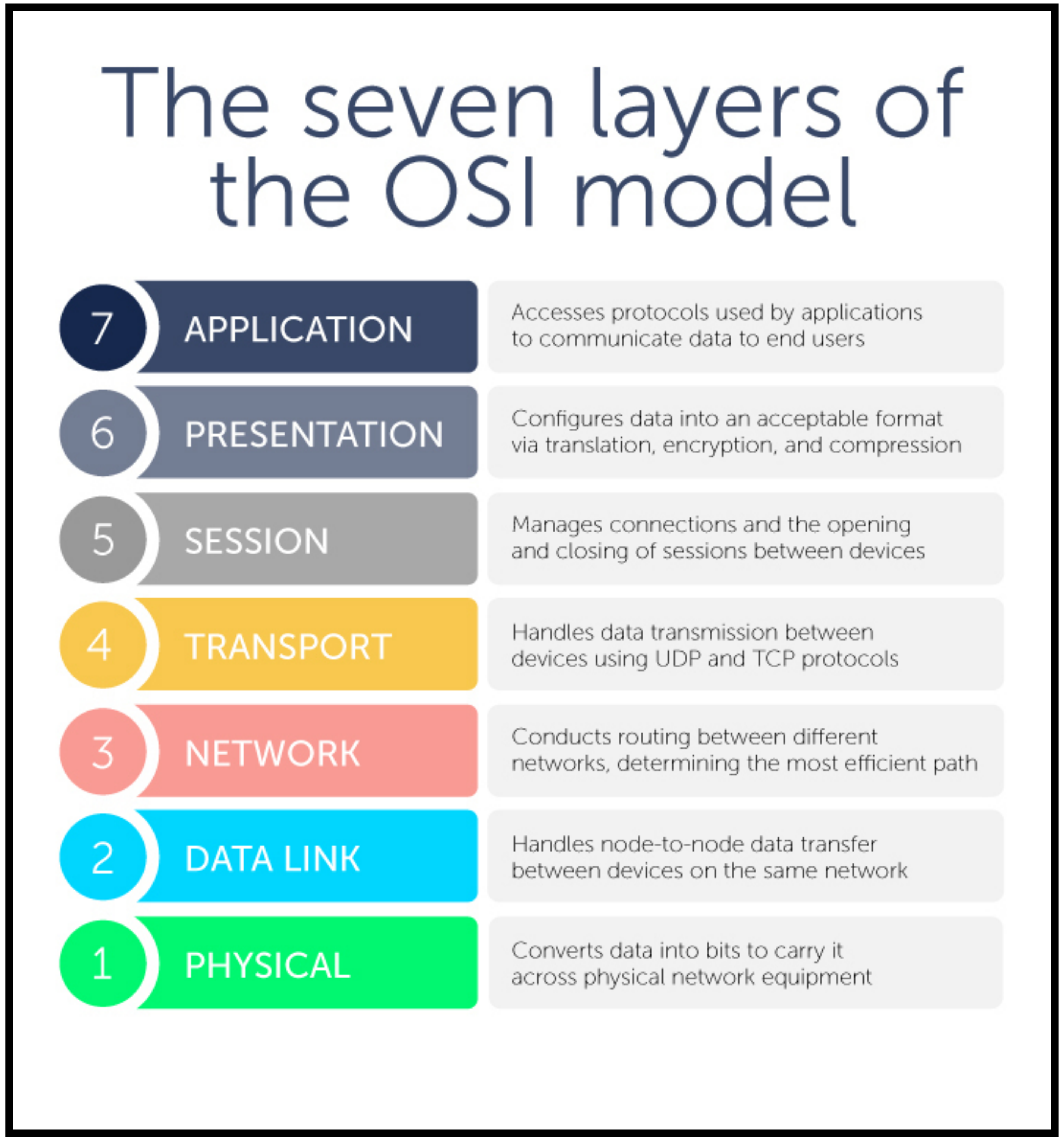
From packets, we have… protocols

- With a group, answer the following questions:
  - **What is a network *layer*?**
  - What is a network protocol?
  - What is an IP address?
  - What is a port?

Network Layers

- The Internet is organized into different *layers* that operate at different levels of abstraction and have different functions

- One version: **Open Systems Interconnection (OSI)** model
  - On the bottom end, you have physical layer; converting data into bits and carrying across physical medium
  - One the top end, you have application layer; designed to produce data to laypeople
  - And everything in between….



The seven layers of the OSI model

7	APPLICATION	Accesses protocols used by applications to communicate data to end users
6	PRESENTATION	Configures data into an acceptable format via translation, encryption, and compression
5	SESSION	Manages connections and the opening and closing of sessions between devices
4	TRANSPORT	Handles data transmission between devices using UDP and TCP protocols
3	NETWORK	Conducts routing between different networks, determining the most efficient path
2	DATA LINK	Handles node-to-node data transfer between devices on the same network
1	PHYSICAL	Converts data into bits to carry it across physical network equipment

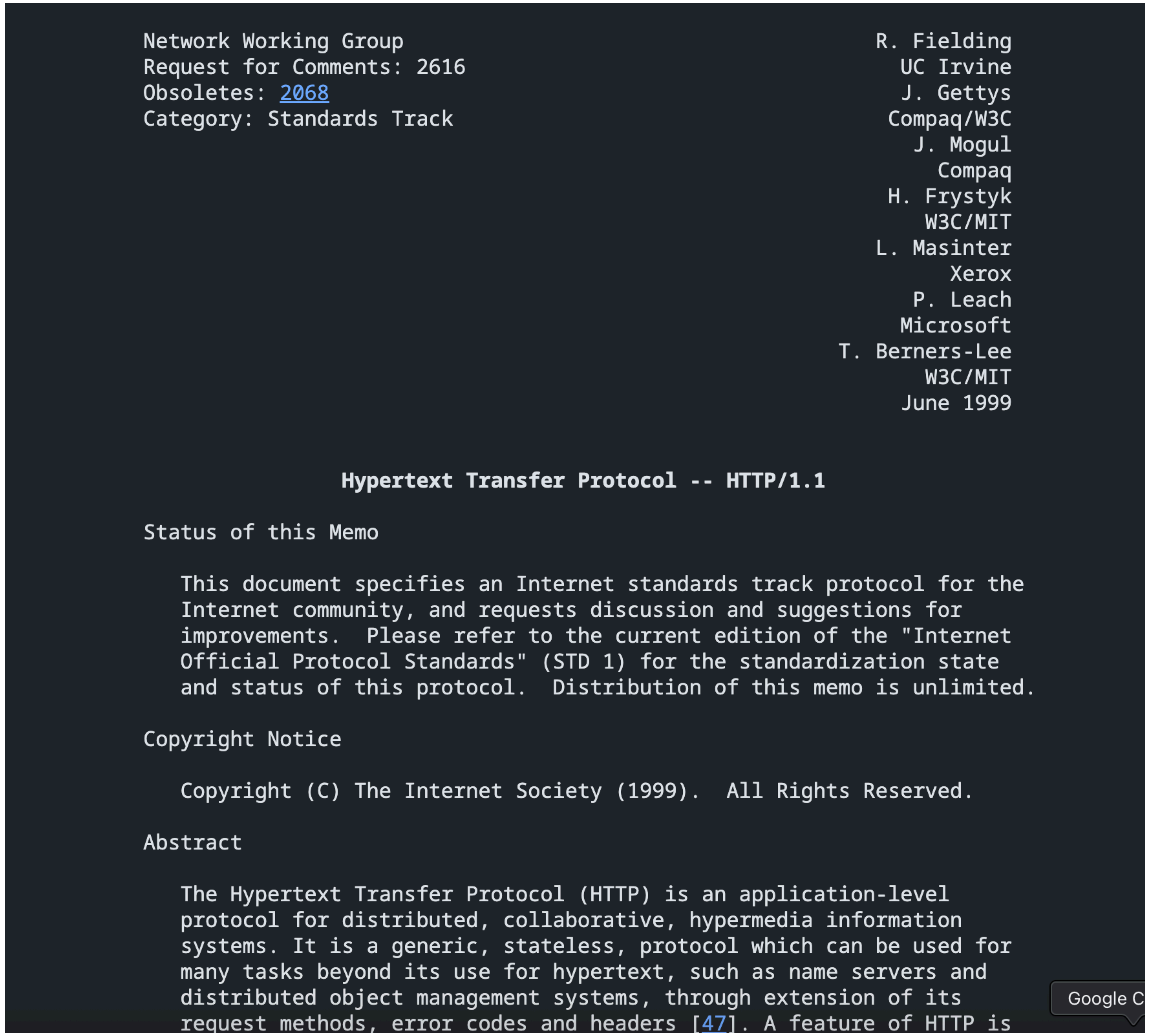
https://bluecatnetworks.com/glossary/what-is-the-osi-model/

From packets, we have… protocols

- With a group, answer the following questions:
  - What is a network *layer*?
  - **What is a network protocol?**
  - What is an IP address?
  - What is a port?

Network Protocols

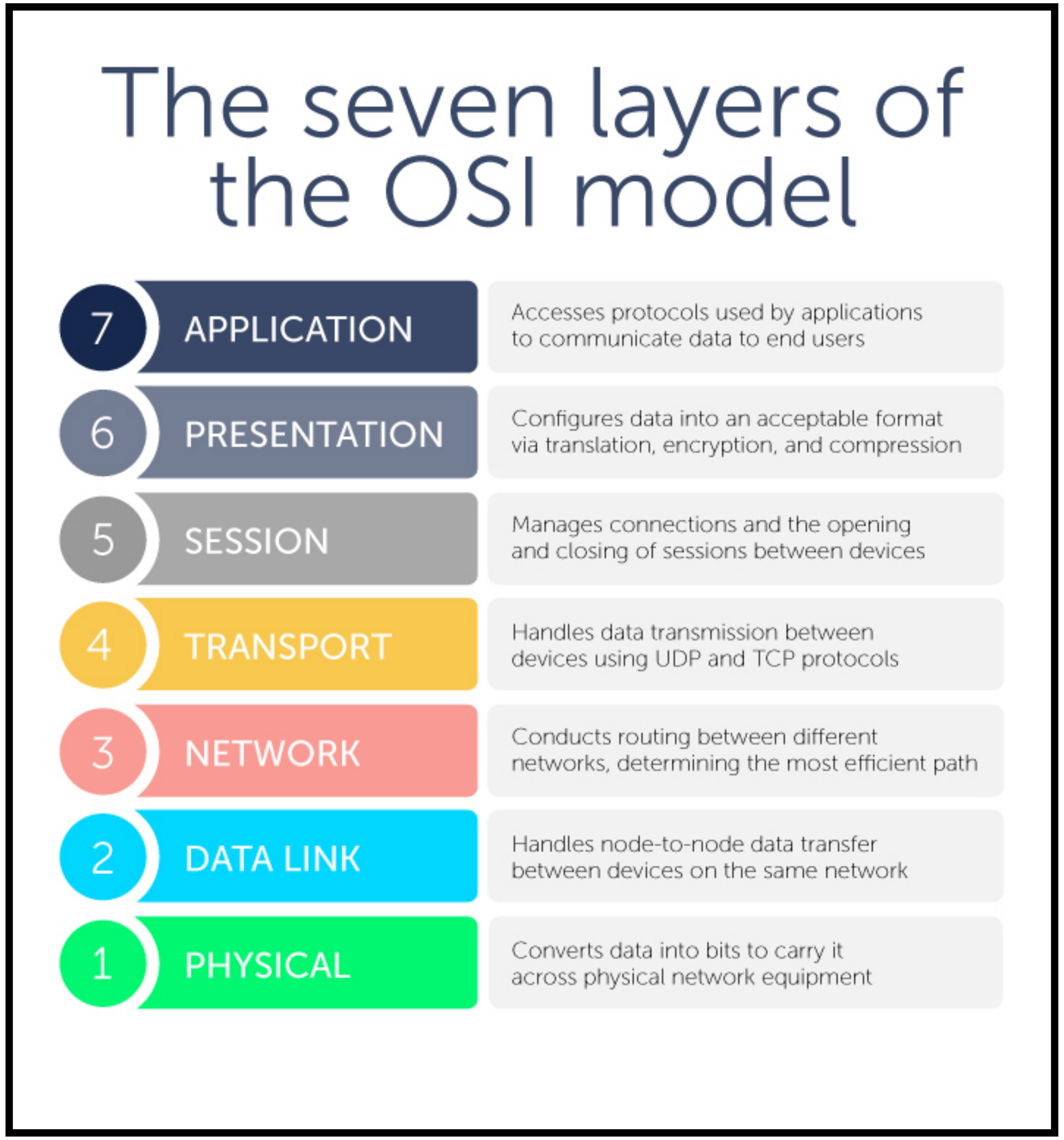
- How machines actually *speak* data to one another
- A spec that tells you exactly how to send, receive, and parse bytes of data you receive on the wire from an underlying network
    - Usually encoded in documents called RFCs (request for comments) by the Internet Engineering Task Force (IETF)
- ***Hundreds of thousands*** of protocols exist…. not all are used (e.g., Telnet)



```
Network Working Group                                      R. Fielding
Request for Comments: 2616                                  UC Irvine
Obsoletes: 2068                                              J. Gettys
Category: Standards Track                                  Compaq/W3C
                                                              J. Mogul
                                                                Compaq
                                                            H. Frystyk
                                                               W3C/MIT
                                                           L. Masinter
                                                                 Xerox
                                                              P. Leach
                                                             Microsoft
                                                        T. Berners-Lee
                                                               W3C/MIT
                                                             June 1999


                Hypertext Transfer Protocol -- HTTP/1.1

Status of this Memo

   This document specifies an Internet standards track protocol for the
   Internet community, and requests discussion and suggestions for
   improvements.  Please refer to the current edition of the "Internet
   Official Protocol Standards" (STD 1) for the standardization state
   and status of this protocol.  Distribution of this memo is unlimited.

Copyright Notice

   Copyright (C) The Internet Society (1999).  All Rights Reserved.

Abstract

   The Hypertext Transfer Protocol (HTTP) is an application-level
   protocol for distributed, collaborative, hypermedia information
   systems. It is a generic, stateless, protocol which can be used for
   many tasks beyond its use for hypertext, such as name servers and
   distributed object management systems, through extension of its
   request methods, error codes and headers [47]. A feature of HTTP is
```

Some examples protocols by the layers…

- L7 — HTTP, FTP, SSH
- L4 — UDP, TCP, QUIC
  - User Datagram Protocol: single packet transmission with no reliability
  - Transmission Control Protocol: connection establishment, reliable transmission, flow-control
- L3 — Internet Protocol (IP), provides fragmentation, reassembly, and *routing*
- L2 — Ethernet; communication between wires



The seven layers of the OSI model

7	APPLICATION	Accesses protocols used by applications to communicate data to end users
6	PRESENTATION	Configures data into an acceptable format via translation, encryption, and compression
5	SESSION	Manages connections and the opening and closing of sessions between devices
4	TRANSPORT	Handles data transmission between devices using UDP and TCP protocols
3	NETWORK	Conducts routing between different networks, determining the most efficient path
2	DATA LINK	Handles node-to-node data transfer between devices on the same network
1	PHYSICAL	Converts data into bits to carry it across physical network equipment

https://bluecatnetworks.com/glossary/what-is-the-osi-model/

From packets, we have… protocols

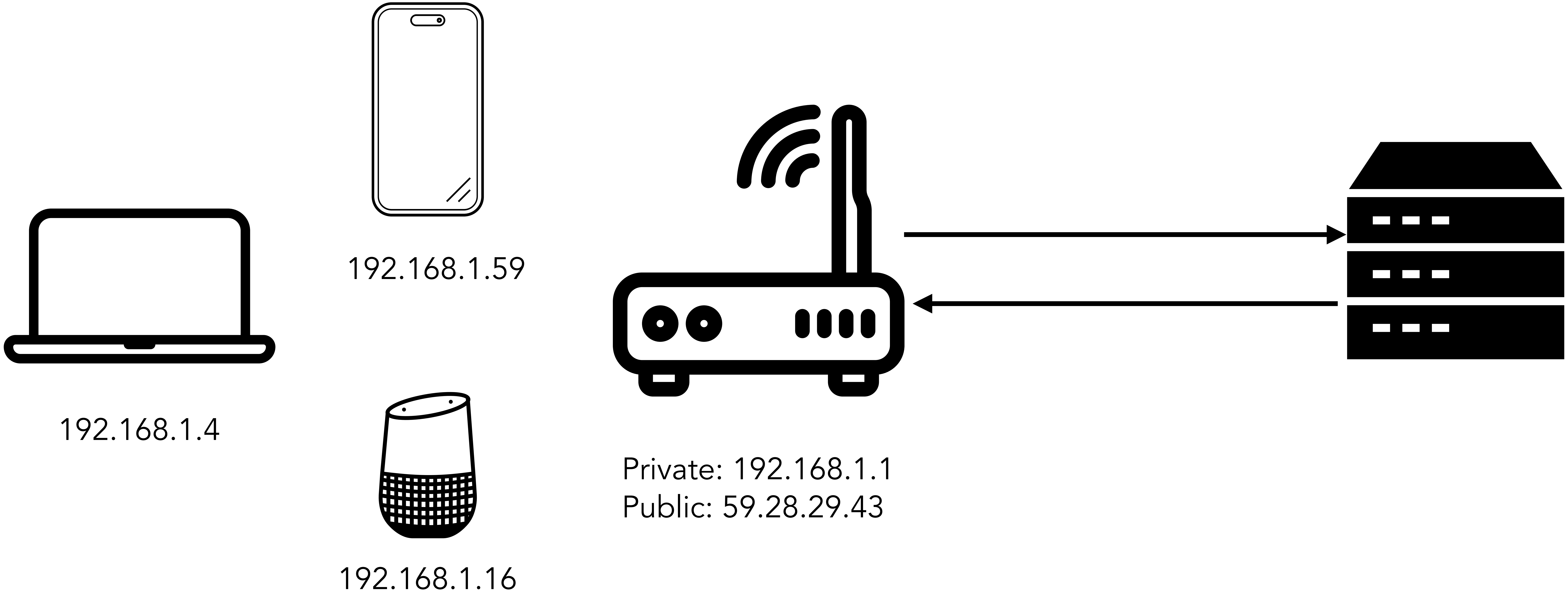
- With a group, answer the following questions:
    - What is a network *layer*?
    - What is a network protocol?
    - **What is an IP address?**
    - What is a port?

IP addresses

- The most common unit of addressing on the modern Internet
  - IPv4 — 32 bits (~4.3 billion unique IP addresses)
  - IPv6 — 128 bits (340 undecillion unique IP addresses)
- Essentially, an IP address is just a number; you've probably seen them all the time
  - e.g., 192.168.1.1, 127.0.0.1, 10.0.0.1…
- Two network spaces
  - Public IP space — fixed number of IPs, and every IP is unique
  - Private IP space — infinite number of IPs through network-address translation

Network Address Translation

- Your home router will do what's called **network address translation:** handle packet flow from an infinite number of devices inside your private network to the public world



What does an IP address do for you?

- Routing
  - Your machine knows the location of the **local** router (gateway)
  - IP gateway knows routes to other networks to send packets (and packets usually take many hops to get where they're going)
- Error reporting
  - Send Internet Control Message Protocol (ICMP) packet back to source if there was a problem
- Fragmentation, reassembly, dropping packets
  - Sometimes packets are too big; they get split up; IP takes care of all this under the hood (take CSE 123 if you like this stuff)

IP addresss under the hood

IPv4 header format

Offset	Octet	0								1								2								3							
Octet	Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
0	0	*Version (4)*				*IHL*				*DSCP*						*ECN*		*Total Length*															
4	32	*Identification*																*Flags*			*Fragment Offset*												
8	64	*Time to Live*								*Protocol*								*Header Checksum*															
12	96	*Source address*																															
16	128	*Destination address*																															
20	160	*(Options) (if IHL > 5)*																															
⋮	⋮																																
56	448																																

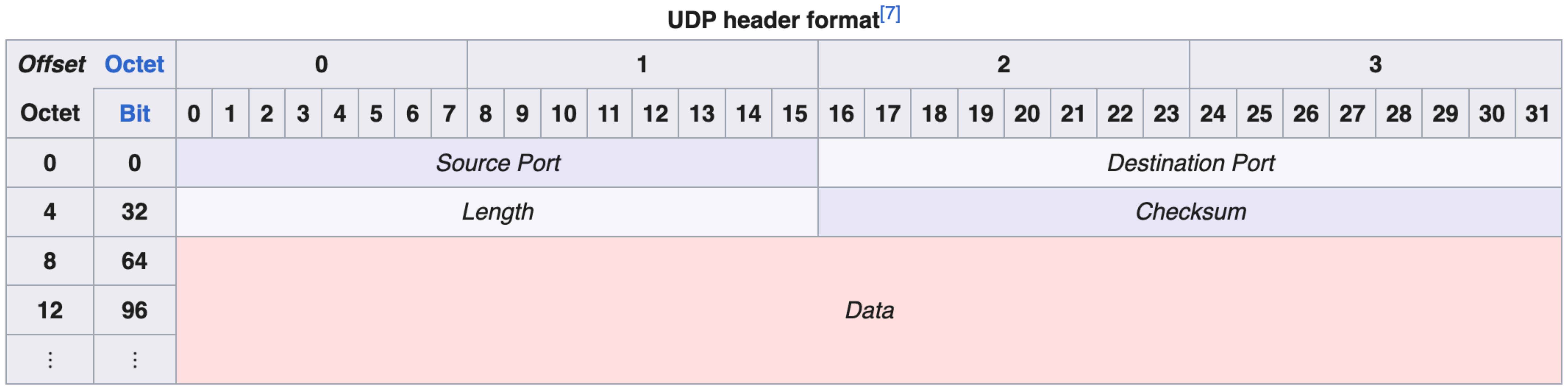
https://en.wikipedia.org/wiki/IPv4#Header

From packets, we have… protocols

- With a group, answer the following questions:
  - What is a network *layer*?
  - What is a network protocol?
  - What is an IP address?
  - **What is a port?**

Ports

- Most systems usually come with just one network interface card (NIC) — speaks packets in hardware
  - But so many applications need to use the NIC, so…
- Network protocols run on *ports;* an abstraction provided by operating systems to separate network traffic from one another
- Ports are used for inbound **and** outbound connections, exposed at L4
  - UDP: IP + ports

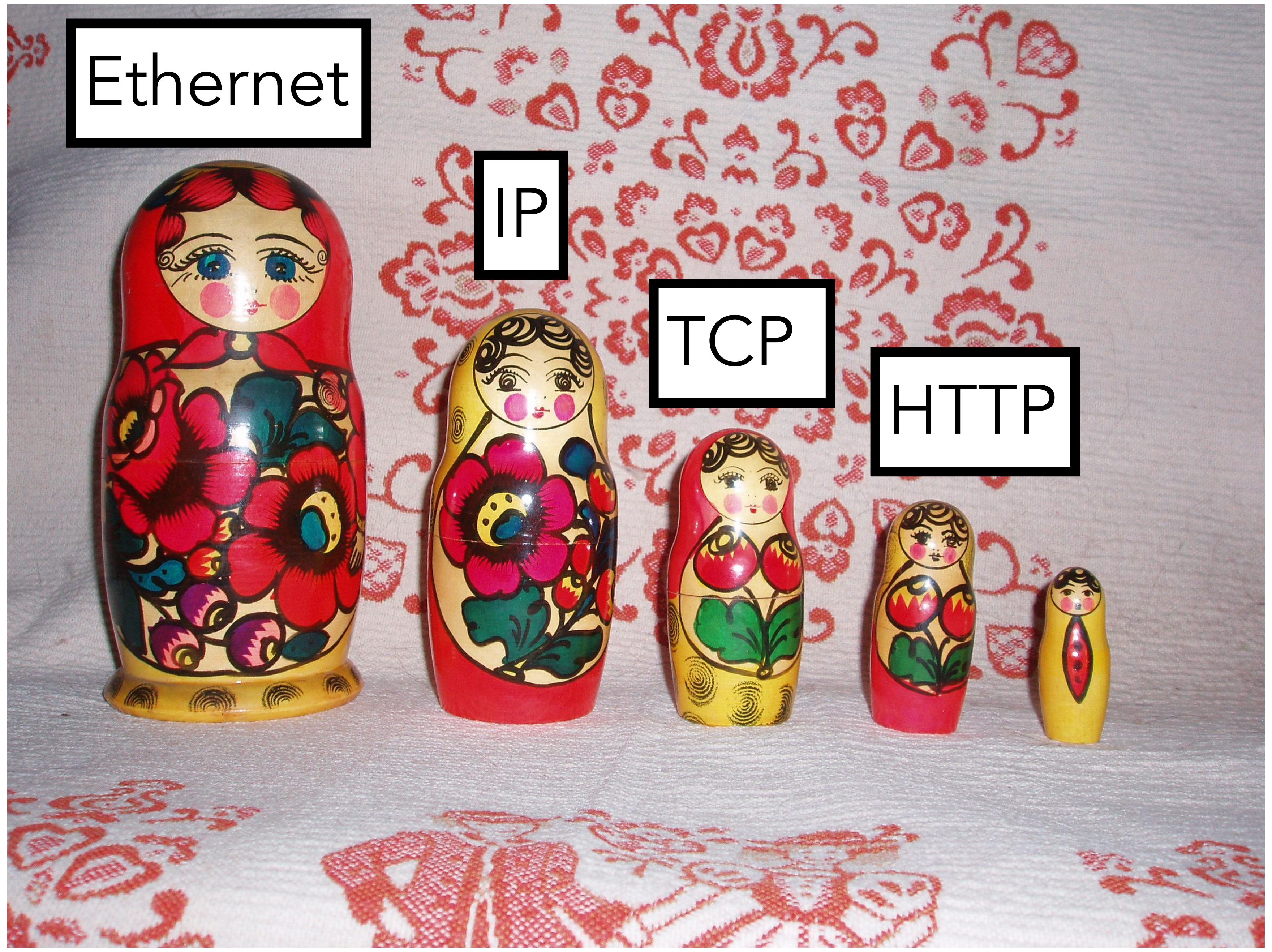


UDP header format[7]

Offset	Octet	0								1								2								3							
Octet	Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
0	0	*Source Port*																*Destination Port*															
4	32	*Length*																*Checksum*															
8	64	*Data*																															
12	96																																
⋮	⋮																																

Data gets *encapsulated* by lower layers

- When sending packets, higher layer packets get put inside lower level packets for efficient communication
  - E.g., UDP packet will be *encapsulated* by an IP packet which is *encapsulated* by a link-layer packet which is sent over the wire
- Error correction will happen at each layer
- This way, sending ethernet bytes isn't affected by what's inside (could be holding any lower level protocol)



Network encapsulation

Examples of protocols and default ports

- Protocols are usually operated at a *default port* for communication
  - HTTP — port 80
  - HTTPS — port 443
  - Simple Mail Transfer Protocol (SMTP) — port 25
  - File Transfer Protocol — port 21
  - Secure SHell (SSH) — port 22
  - Telnet — port 23
- But you can run any networked service at any port, really! E.g., VMs

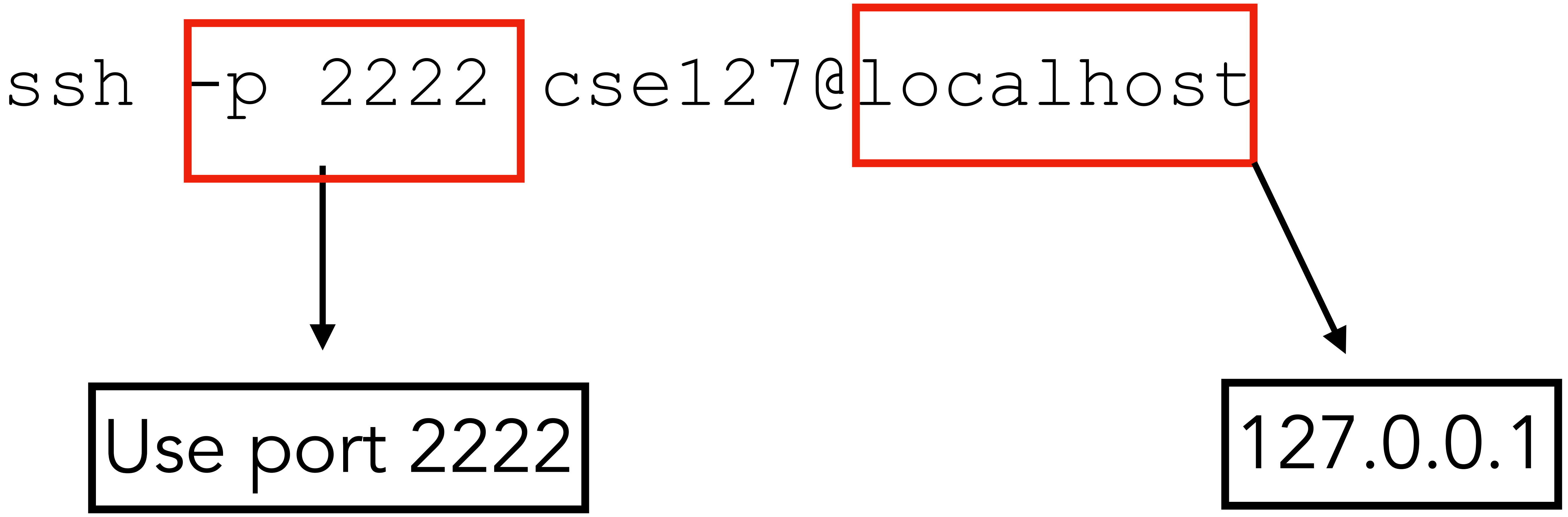
SSH explained

- When you SSH into a machine, you are speaking a *network protocol* that connects *one IP address* to *another IP address*

```
ssh -p 2222 cse127@localhost
```

SSH explained

- When you SSH into a machine, you are speaking a *network protocol* that connects *one IP address* to *another IP address*



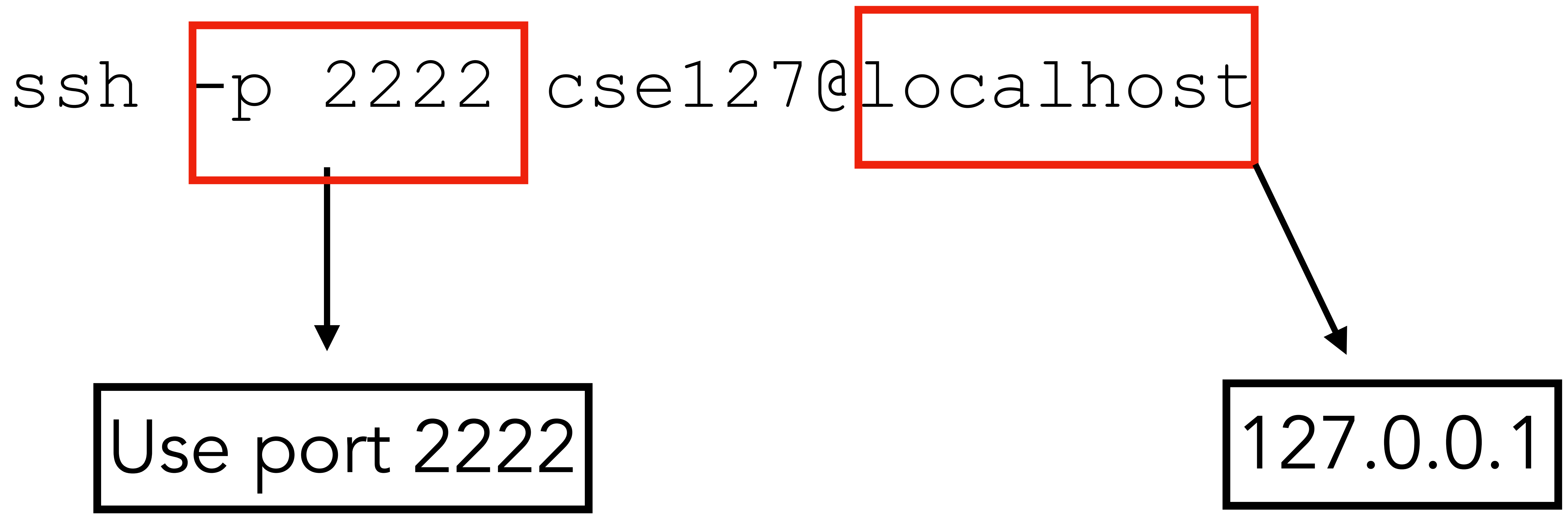
```
ssh -p 2222 cse127@localhost
```

Use port 2222

127.0.0.1

SSH explained

- When you SSH into a machine, you are speaking a *network protocol* that connects *one IP address* to *another IP address*



```
ssh -p 2222 cse127@localhost
```

Use port 2222

127.0.0.1

Network services can run on the same machine; your own computer has **localhost** which you can always connect to by default!

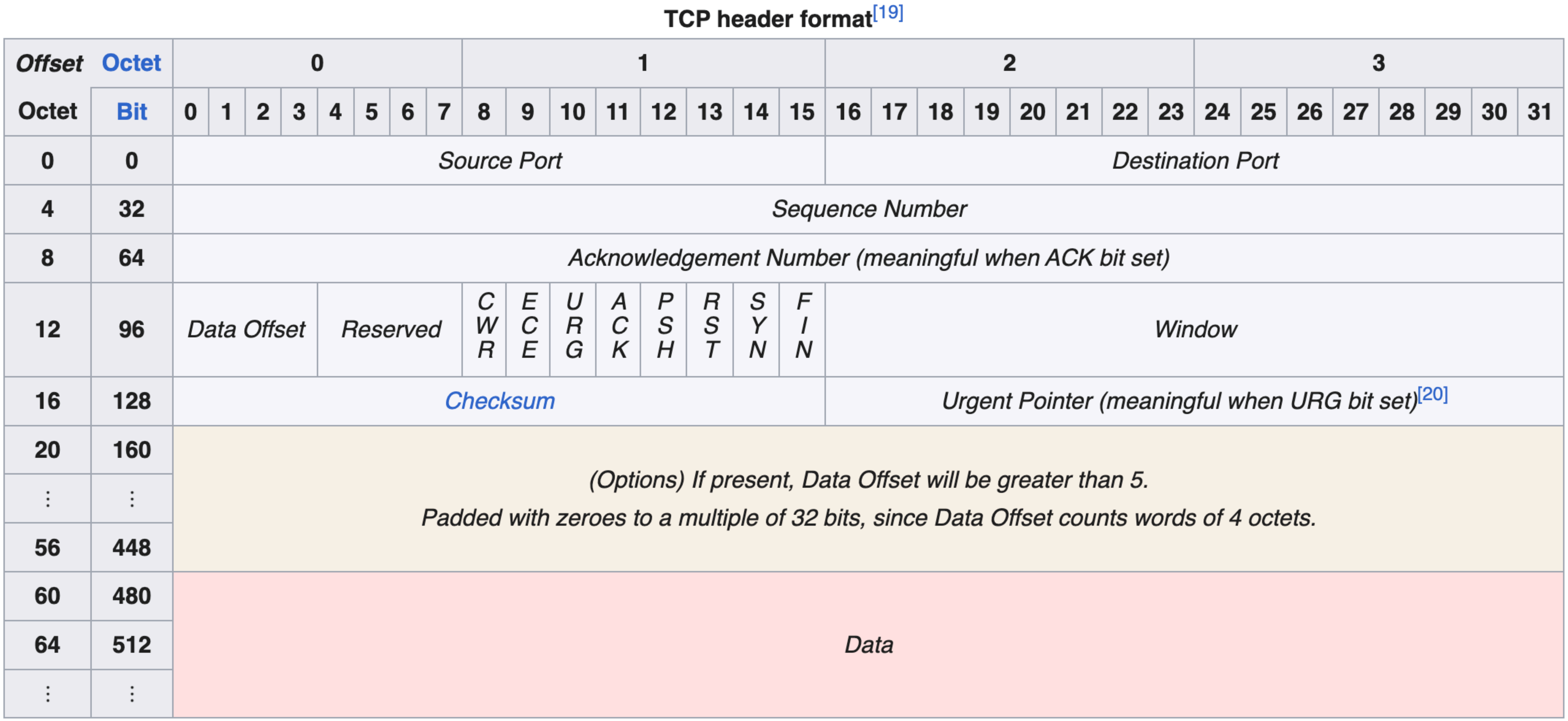
TCP runs most things

- Most application network protocols you’re used to using run on top of TCP
  - E.g., web protocols (HTTP/HTTPS), SMTP, SSH, FTP, etc.
- The reason for this is that TCP provides reliable, ordered delivery of bytes
  - Establishes a stateful, bi-directional session between two IP:port endpoints

TCP runs most things

- Most application network protocols you’re used to using run on top of TCP
    - E.g., web protocols (HTTP/HTTPS), SMTP, SSH, FTP, etc.
- The reason for this is that TCP provides reliable, ordered delivery of bytes
    - Establishes a stateful, bi-directional session between two IP:port endpoints
- This is in contrast to UDP (and IP), which offer **no reliability guarantees at all**
    - AKA: packets can vanish at anytime
- Each side of a TCP connection maintains…
    - Sequence number: seq base number + count of bytes sent
    - Acknowledgement number: acknowledgement base + count of bytes received

TCP Header

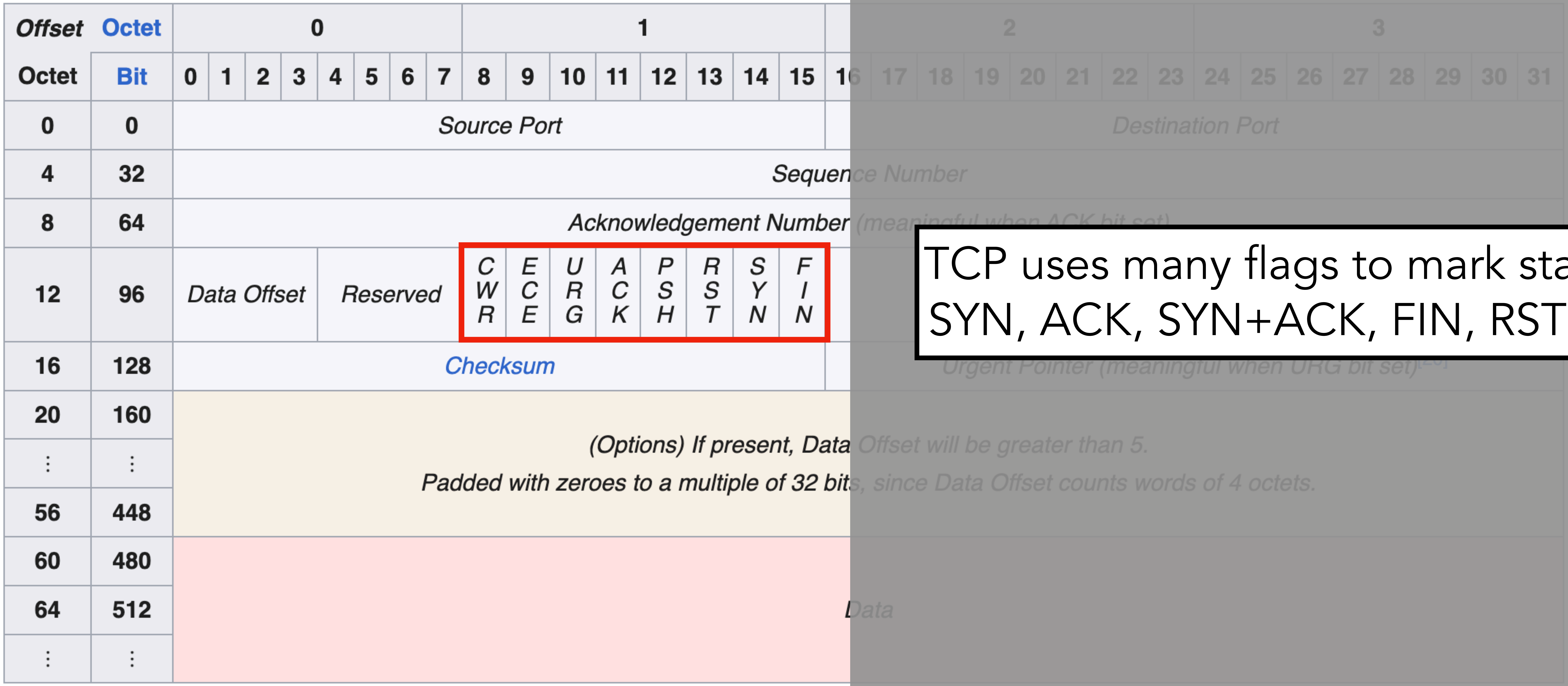


TCP header format[19]

Offset	Octet	0								1								2								3							
Octet	Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
0	0	*Source Port*																*Destination Port*															
4	32	*Sequence Number*																															
8	64	*Acknowledgement Number (meaningful when ACK bit set)*																															
12	96	*Data Offset*				*Reserved*				*CWR*	*ECE*	*URG*	*ACK*	*PSH*	*RST*	*SYN*	*FIN*	*Window*															
16	128	*Checksum*																*Urgent Pointer (meaningful when URG bit set)*[20]															
20	160	*(Options) If present, Data Offset will be greater than 5.* *Padded with zeroes to a multiple of 32 bits, since Data Offset counts words of 4 octets.*																															
⋮	⋮																																
56	448																																
60	480	*Data*																															
64	512																																
⋮	⋮																																

TCP Header

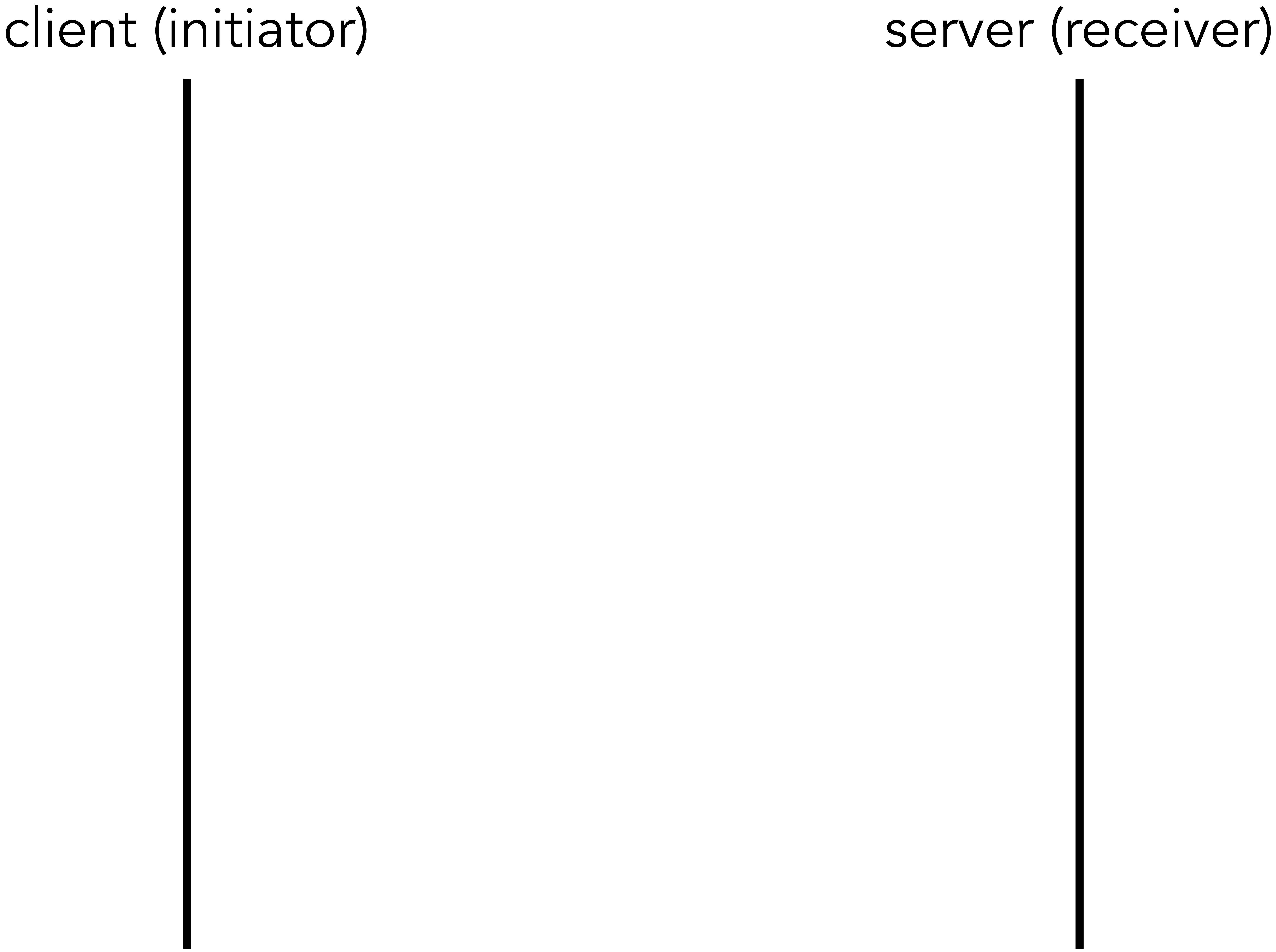
TCP header format[19]



Offset	Octet	0								1								2								3							
Octet	Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
0	0	Source Port																Destination Port															
4	32	Sequence Number																															
8	64	Acknowledgement Number (meaningful when ACK bit set)																															
12	96	Data Offset				Reserved				CWR	ECE	URG	ACK	PSH	RST	SYN	FIN																
16	128	Checksum																Urgent Pointer (meaningful when URG bit set)[20]															
20	160	(Options) If present, Data Offset will be greater than 5. Padded with zeroes to a multiple of 32 bits, since Data Offset counts words of 4 octets.																															
⋮	⋮																																
56	448																																
60	480	Data																															
64	512																																
⋮	⋮																																

TCP uses many flags to mark state:
SYN, ACK, SYN+ACK, FIN, RST….

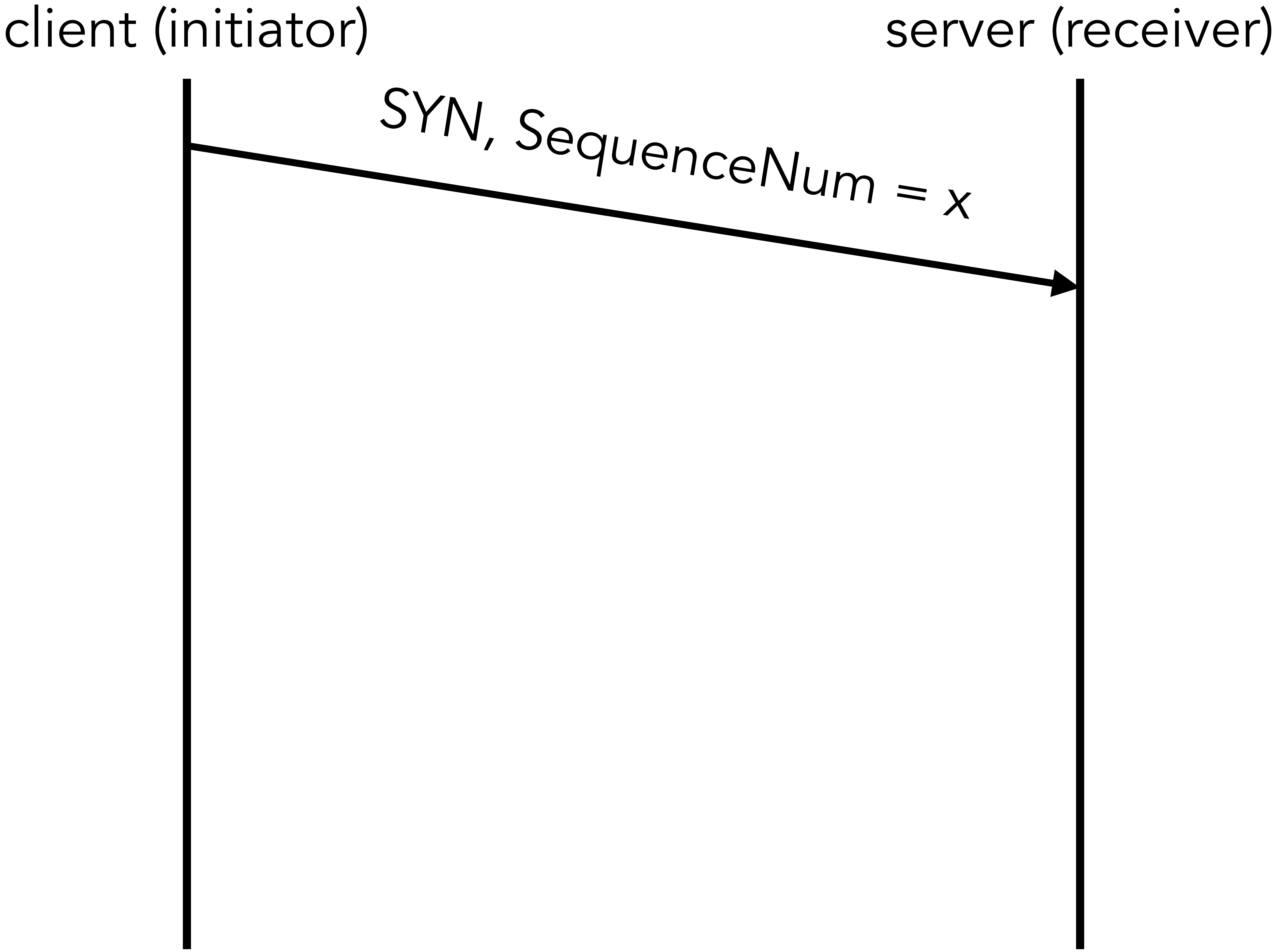
The TCP Handshake



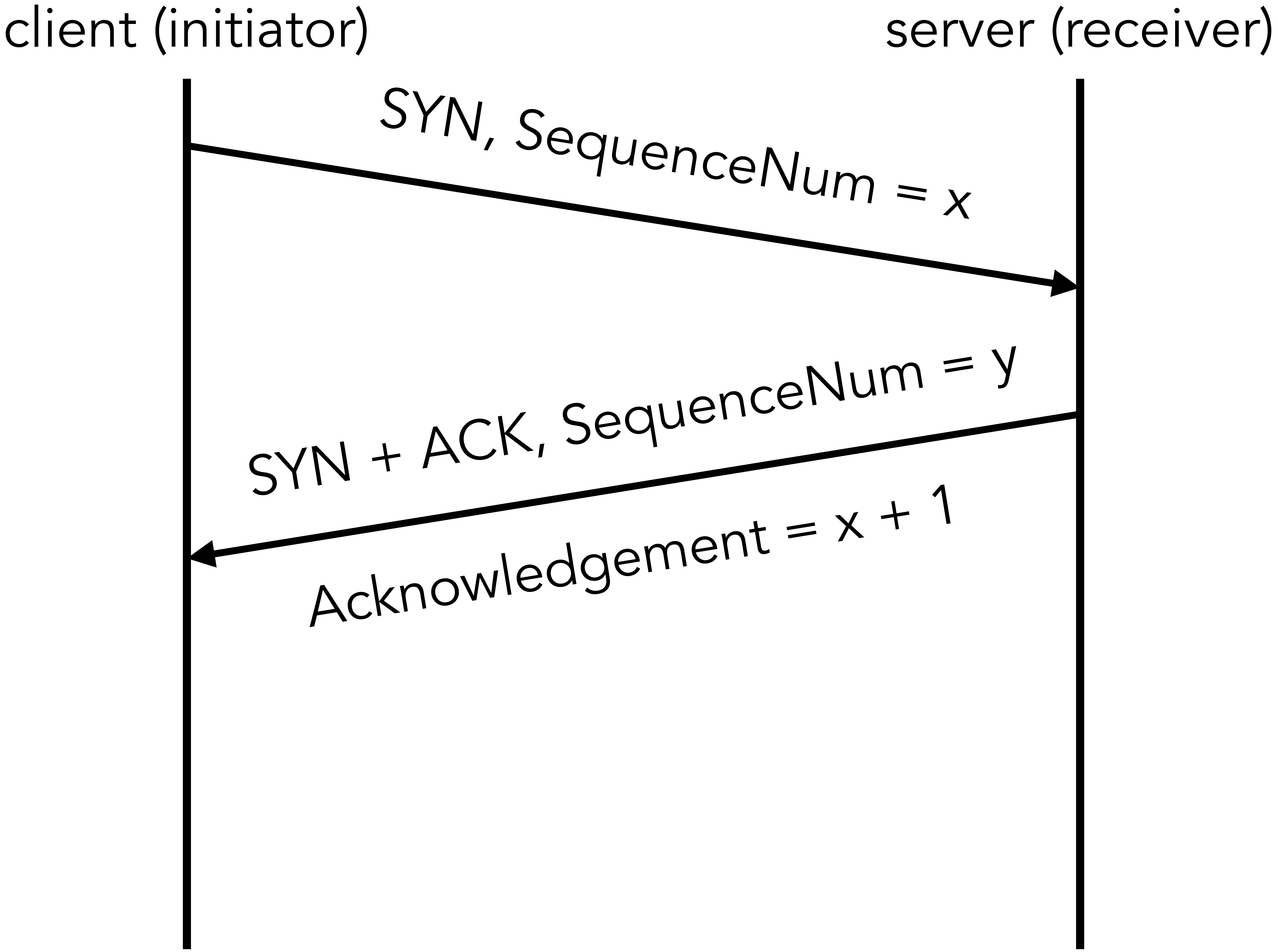
client (initiator)

server (receiver)

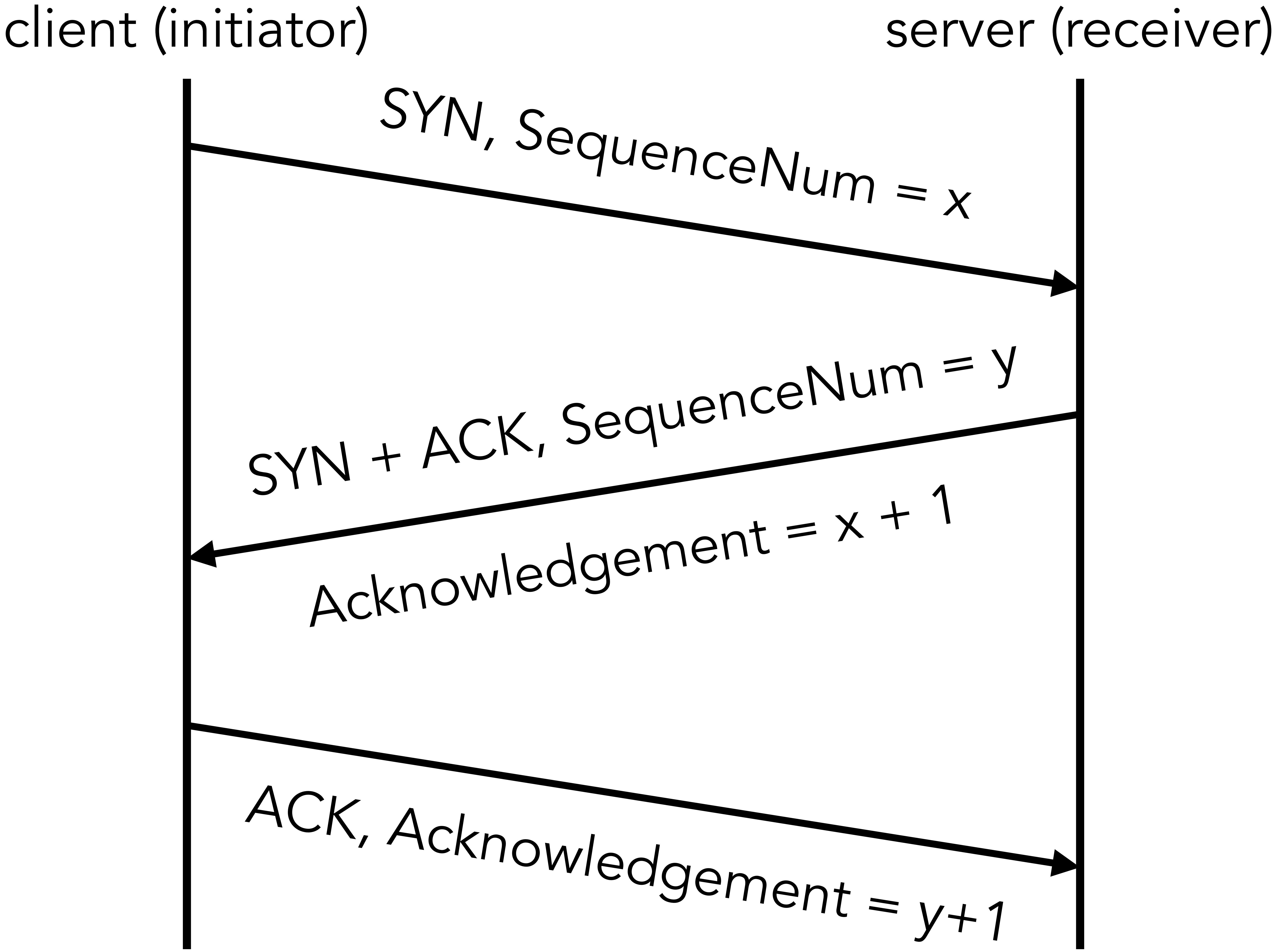
The TCP Handshake



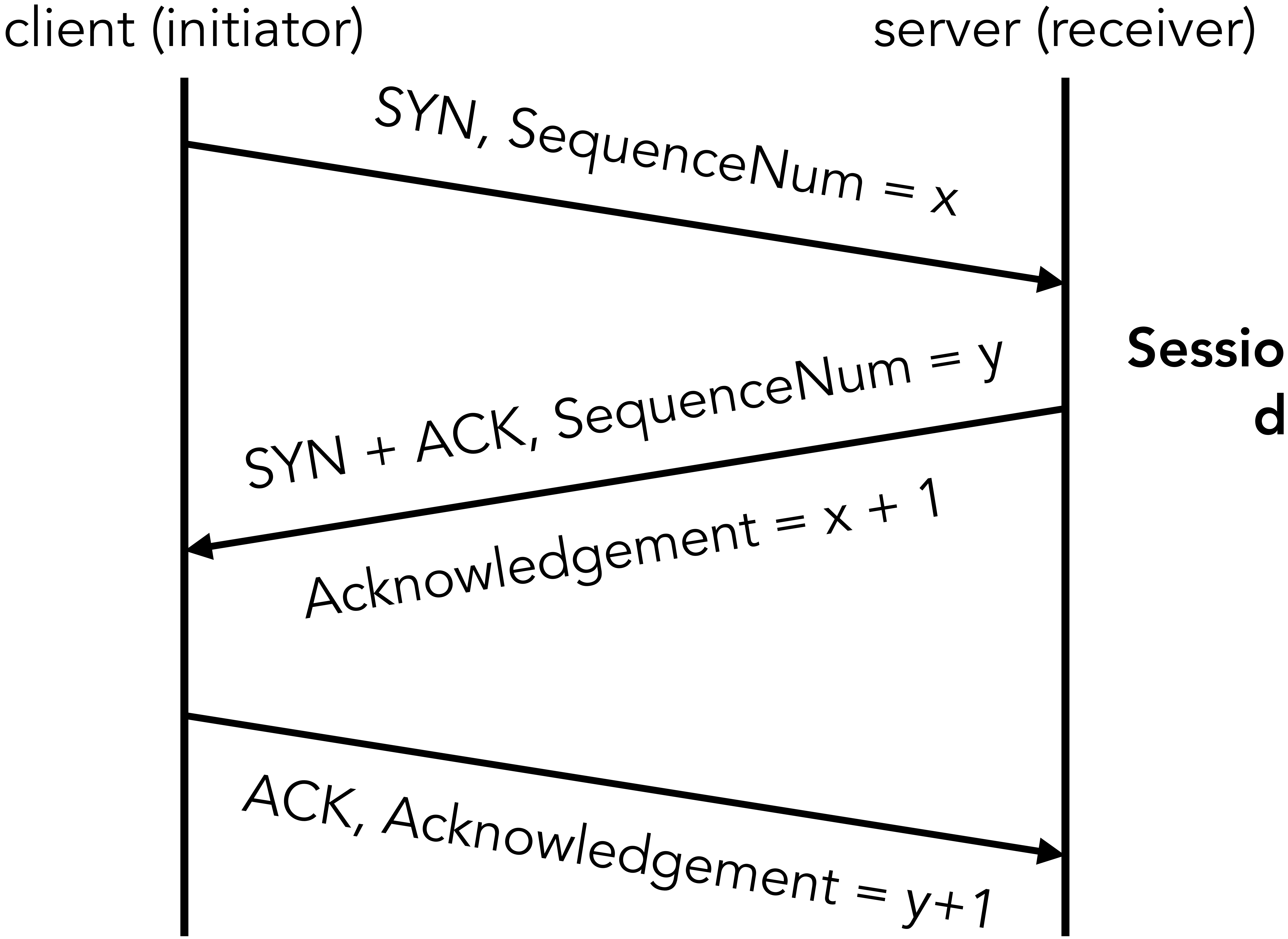
The TCP Handshake



The TCP Handshake



The TCP Handshake



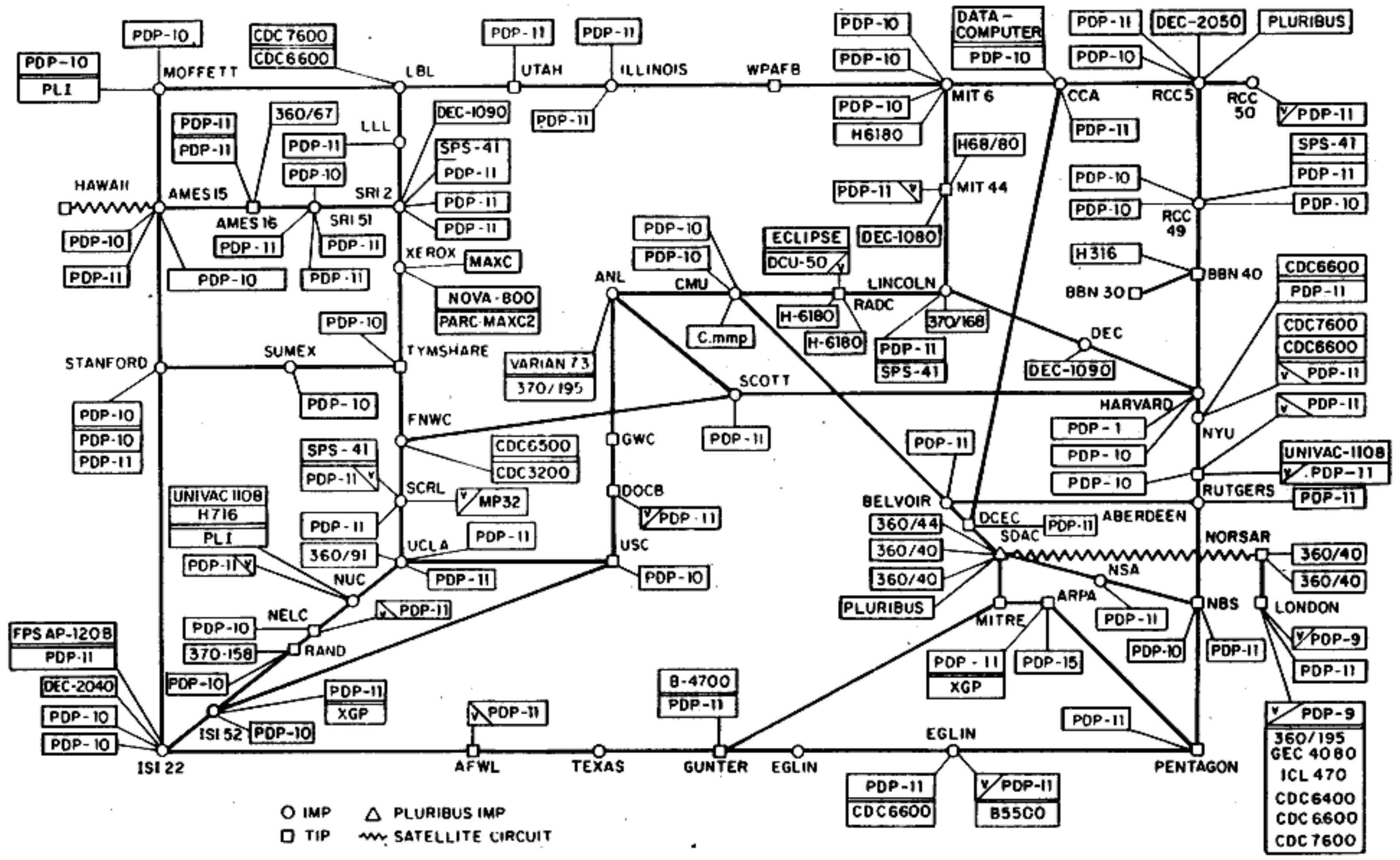
Session established, data time!

Internet Security

TCP / IP Security (1970s)

- Original TCP/IP design: Trusted network and hosts
  - Administered entirely by mutually trusted parties
- End-to-end principle
  - All business logic lives at the nodes, network is dumb simple and maintains no state
- Robustness principle
  - *Be conservative in what you do, be liberal in receiving* – aka, try to accept any datagram you can, but don't spew garbage

ARPANET LOGICAL MAP, MARCH 1977



(PLEASE NOTE THAT WHILE THIS MAP SHOWS THE HOST POPULATION OF THE NETWORK ACCORDING TO THE BEST INFORMATION OBTAINABLE, NO CLAIM CAN BE MADE FOR ITS ACCURACY)

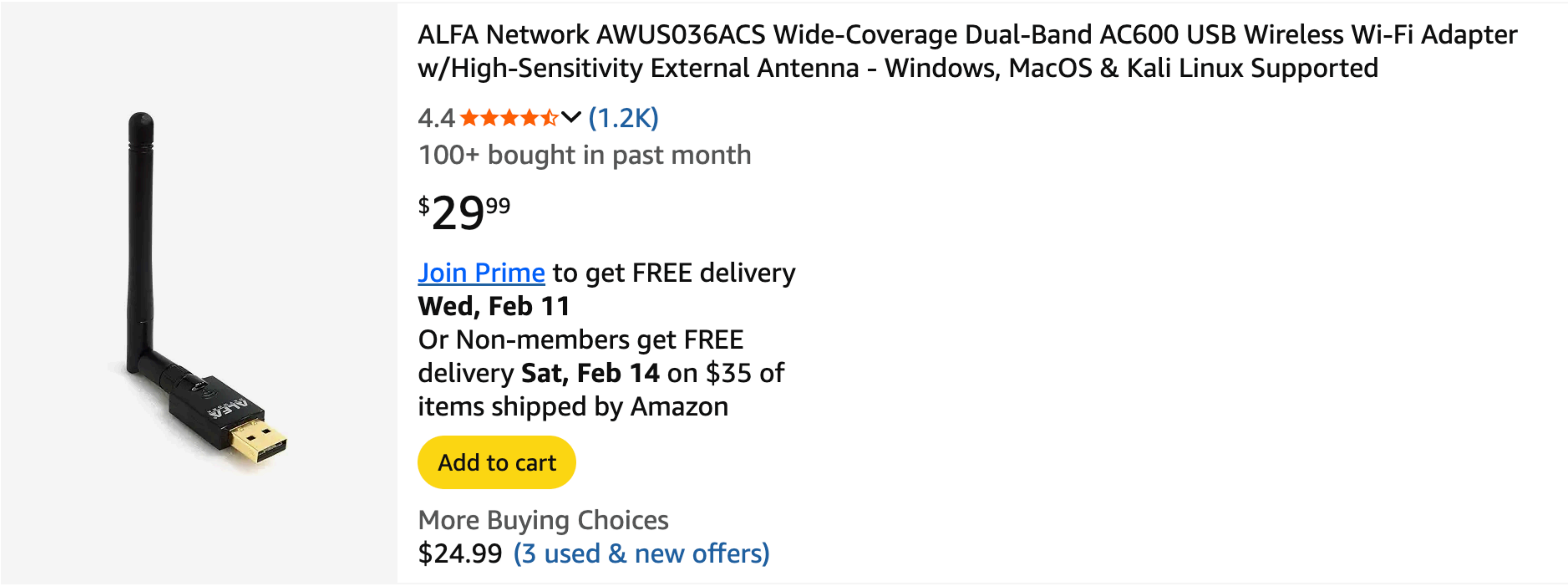
NAMES SHOWN ARE IMP NAMES, NOT (NECESSARILY) HOST NAMES

TCP / IP Security (1980s)

- We’ve made a mistake
    - As the Internet exploded in usage and popularity, fundamental trust assumption went out the window
    - “When describing such attacks, our basic assumption is the attacker has more or less complete control over some machine connected to the Internet… indeed the attacker **may even be a rogue system administrator.”** – Steve Bellovin 1989
- 1980s threat model
    - Can’t trust the hosts
    - But **network** was still deemed to be trusted and worked on by a small number of people…

TCP / IP Security (today)

- We can’t trust the network either
  - Network equipment might be compromised
  - Untrusted network operators (ISP could be malicious)
  - *Anyone* can access the physical channel of wireless networks (e.g., WiFi sniffing…) — easily sniffed



ALFA Network AWUS036ACS Wide-Coverage Dual-Band AC600 USB Wireless Wi-Fi Adapter w/High-Sensitivity External Antenna - Windows, MacOS & Kali Linux Supported

4.4 ★★★★☆ (1.2K)

100+ bought in past month

$29.99

Join Prime to get FREE delivery **Wed, Feb 11**

Or Non-members get FREE delivery **Sat, Feb 14** on $35 of items shipped by Amazon

More Buying Choices

$24.99 (3 used & new offers)

Network Attacker Models

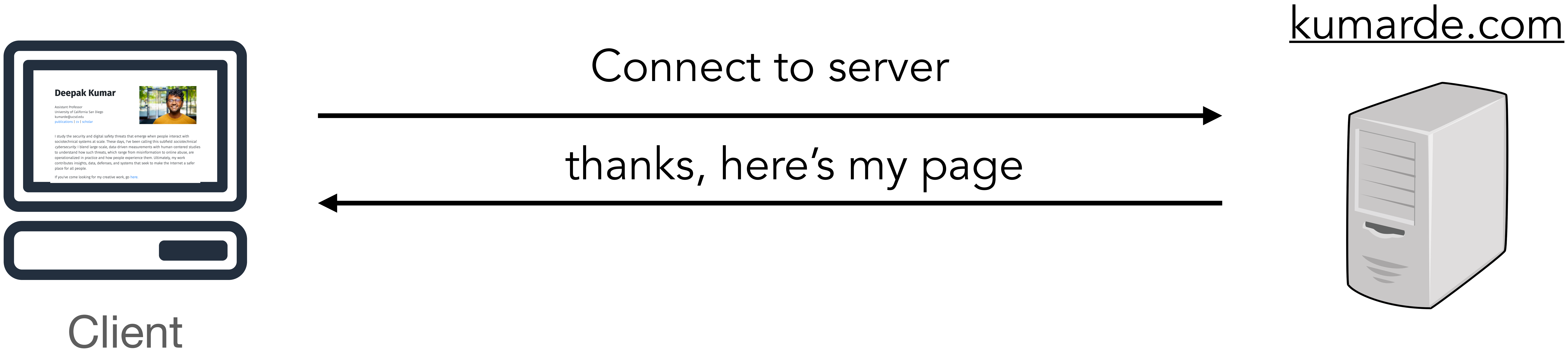
- Man-in-the-middle attacker
  - An entity *between* the two communicating machines can tamper with, manipulate, or read traffic
- Passive eavesdropper
  - Attacker has a passive tap or recorded traces (could easily be your ISP)
- Off-path attacker
  - Attacker can inject traffic into the network (anyone w/ access to the network)

Man in the Middle (MiTM)

kumarde.com

Client

Man in the Middle (MiTM)



Deepak Kumar

Assistant Professor
University of California San Diego
kumarde@ucsd.edu
publications | cv | scholar

I study the security and digital safety threats that emerge when people interact with sociotechnical systems at scale. These days, I've been calling this subfield *sociotechnical cybersecurity.* I blend large-scale, data-driven measurements with human-centered studies to understand how such threats, which range from misinformation to online abuse, are operationalized in practice and how people experience them. Ultimately, my work contributes insights, data, defenses, and systems that seek to make the Internet a safer place for all people.

If you've come looking for my creative work, go here.

Client

Connect to server

thanks, here’s my page

kumarde.com

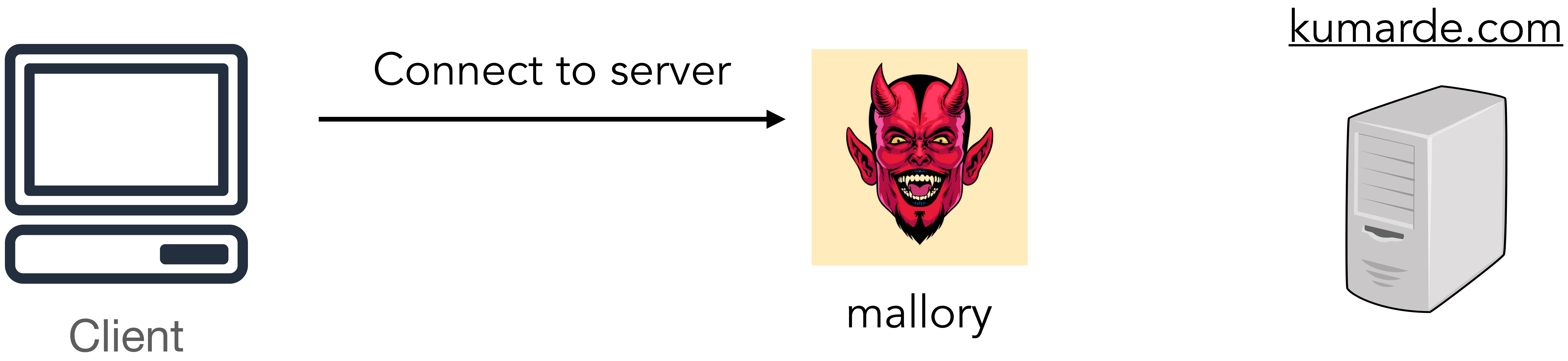
Man in the Middle (MiTM)

kumarde.com

mallory

Client

Man in the Middle (MiTM)



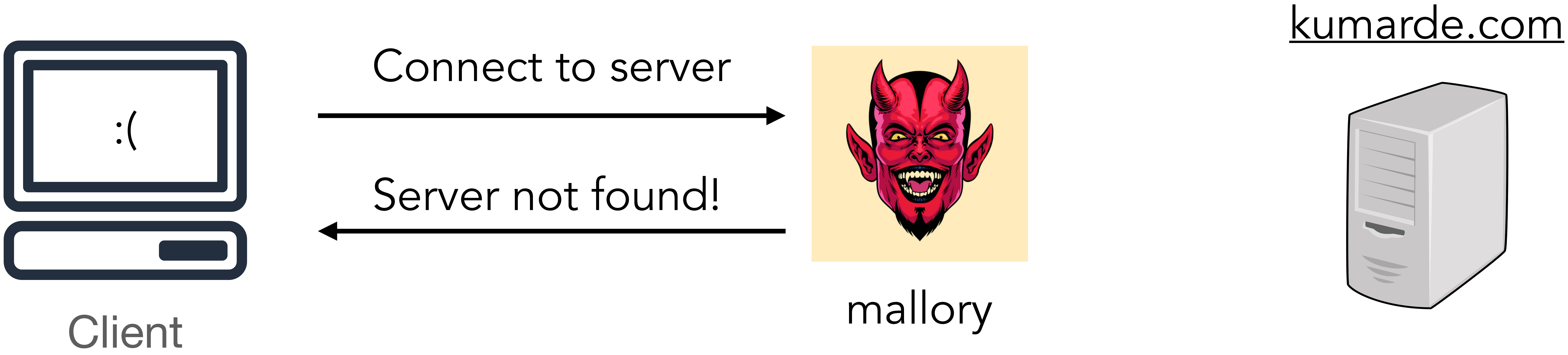
Connect to server

Client

mallory

kumarde.com

Man in the Middle (MiTM)



:(

Client

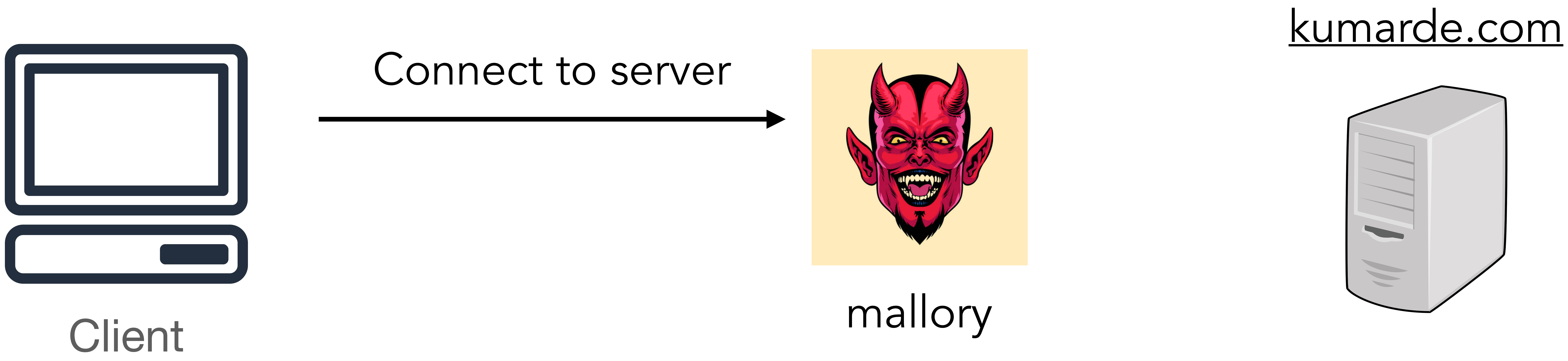
Connect to server

Server not found!

mallory

kumarde.com

Man in the Middle (MiTM)



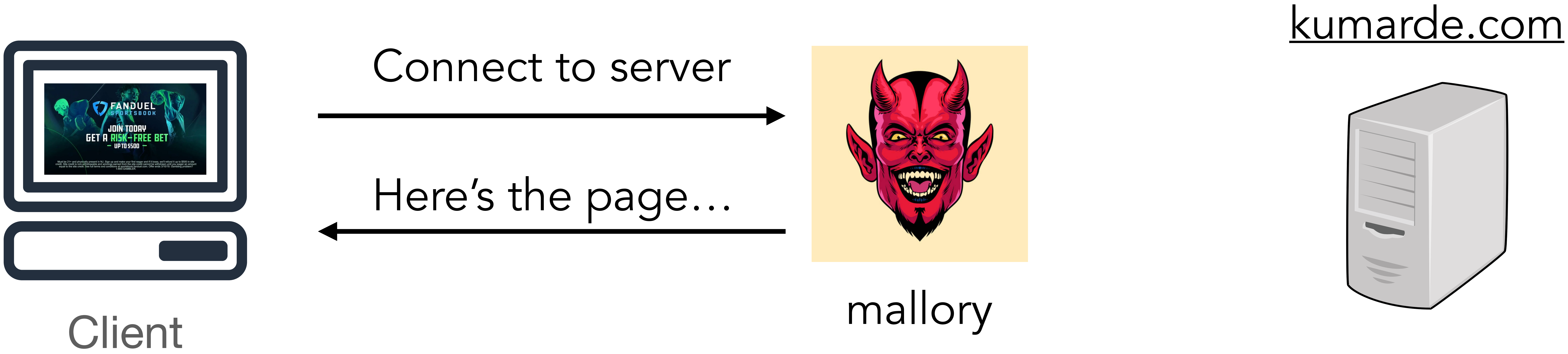
Client

Connect to server

mallory

kumarde.com

Man in the Middle (MiTM)



kumarde.com

Connect to server

Here's the page…

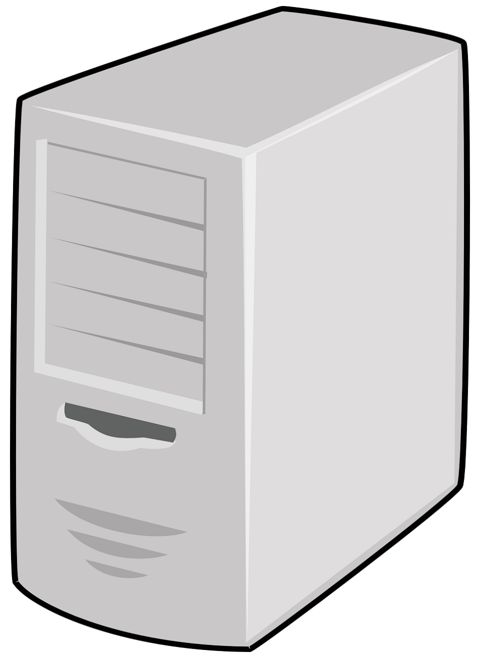
Client

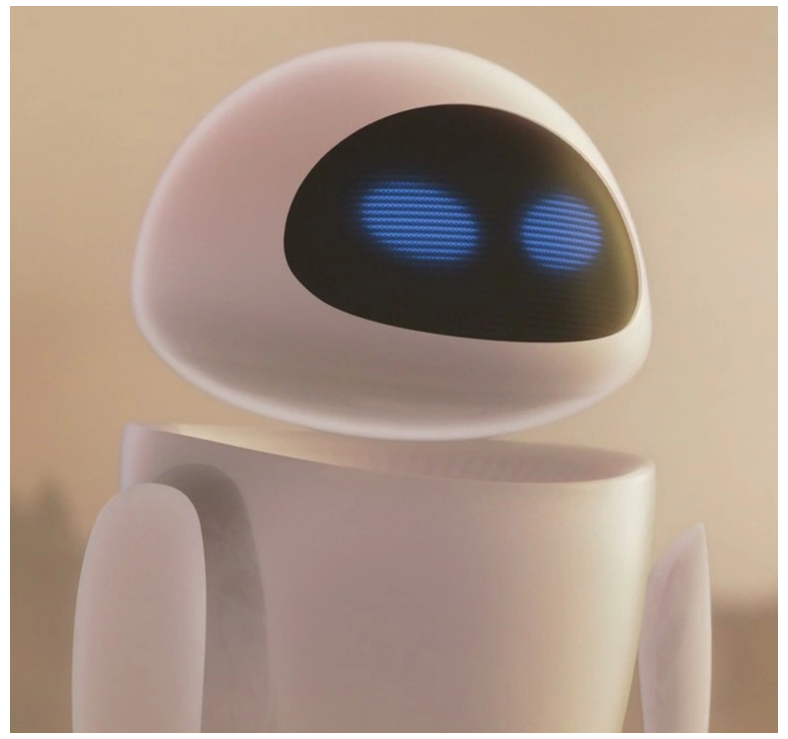
mallory

Passive Eavesdropper

Client

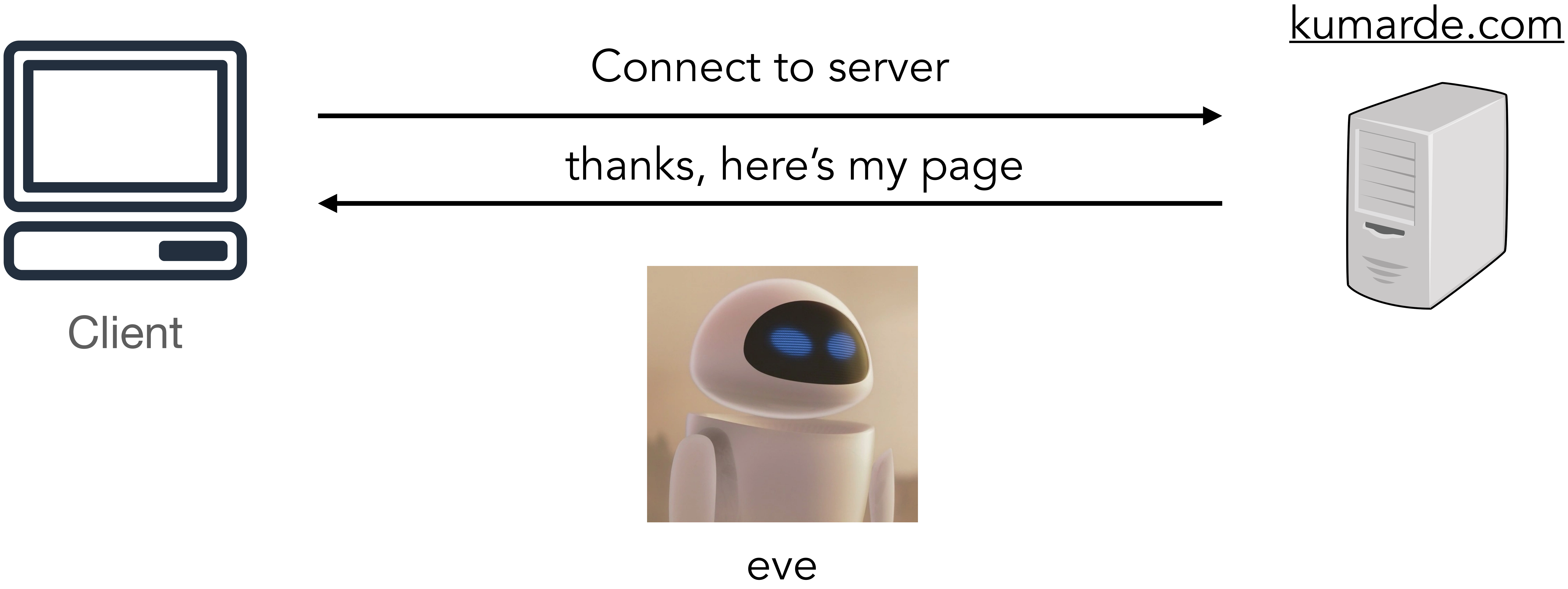
kumarde.com





eve

Passive Eavesdropper



kumarde.com

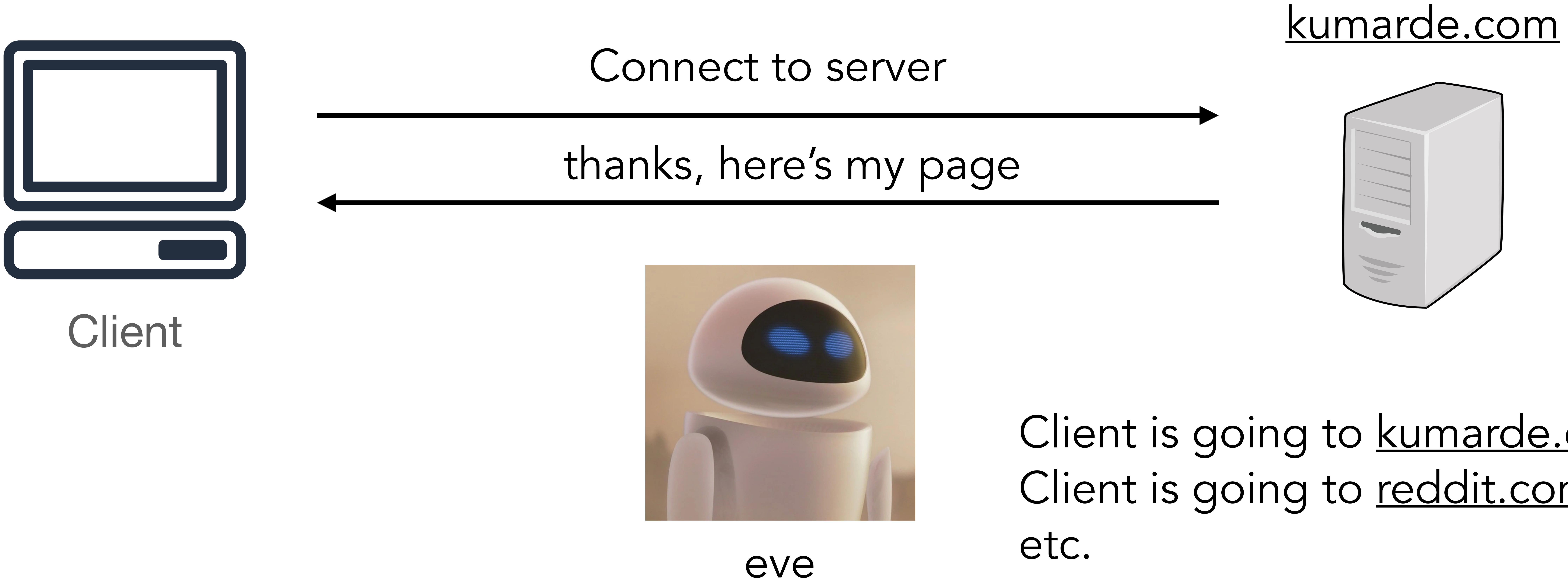
Connect to server

thanks, here’s my page

Client

eve

Passive Eavesdropper



Client is going to kumarde.com
Client is going to reddit.com
etc.

The Internet was built with no security in mind

- **No confidentiality**
  - Who can see the packets you send?

The Internet was built with no security in mind

- **No confidentiality**
  - Who can see the packets you send?
    - Network (ISP, routers, any hop along the way)
    - Anyone who can sniff WiFi
- **No authentication**
  - Attacker with direct access to network can spoof source IP address… doesn’t matter
    - UDP especially vulnerable! We’ll see some consequences of this next week…

Even the link layer is unprotected

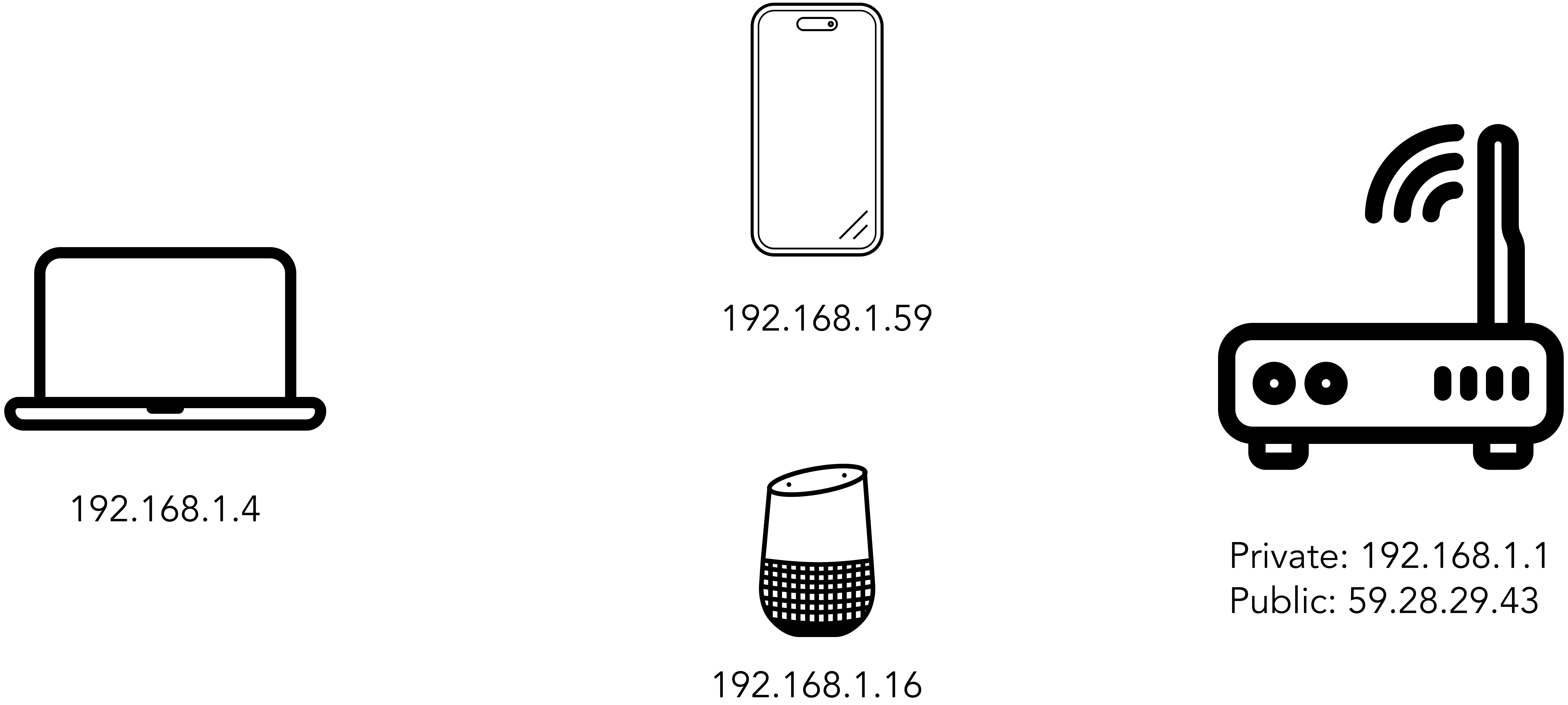
- Physical channel is often shared by multiple hosts on the network (e.g., coffee shop WiFi)
- Link layer controls access to the physical medium
  - Known as the Media Access Control (MAC) layer
- How do you make sure that each host only gets the frames addressed to it?

Even the link layer is unprotected

- Physical channel is often shared by multiple hosts on the network (e.g., coffee shop WiFi)
- Link layer controls access to the physical medium
  - Known as the Media Access Control (MAC) layer
- How do you make sure that each host only gets the frames addressed to it?
  - ….honor system, duh
  - NIC (which has a MAC address assigned) usually filters
- But almost all NICs support “promiscuous” mode — all frames are picked up

Going back to home networks

- How do devices know how to send packets to other devices?
  - Through a protocol called ARP – address routing protocol



192.168.1.59

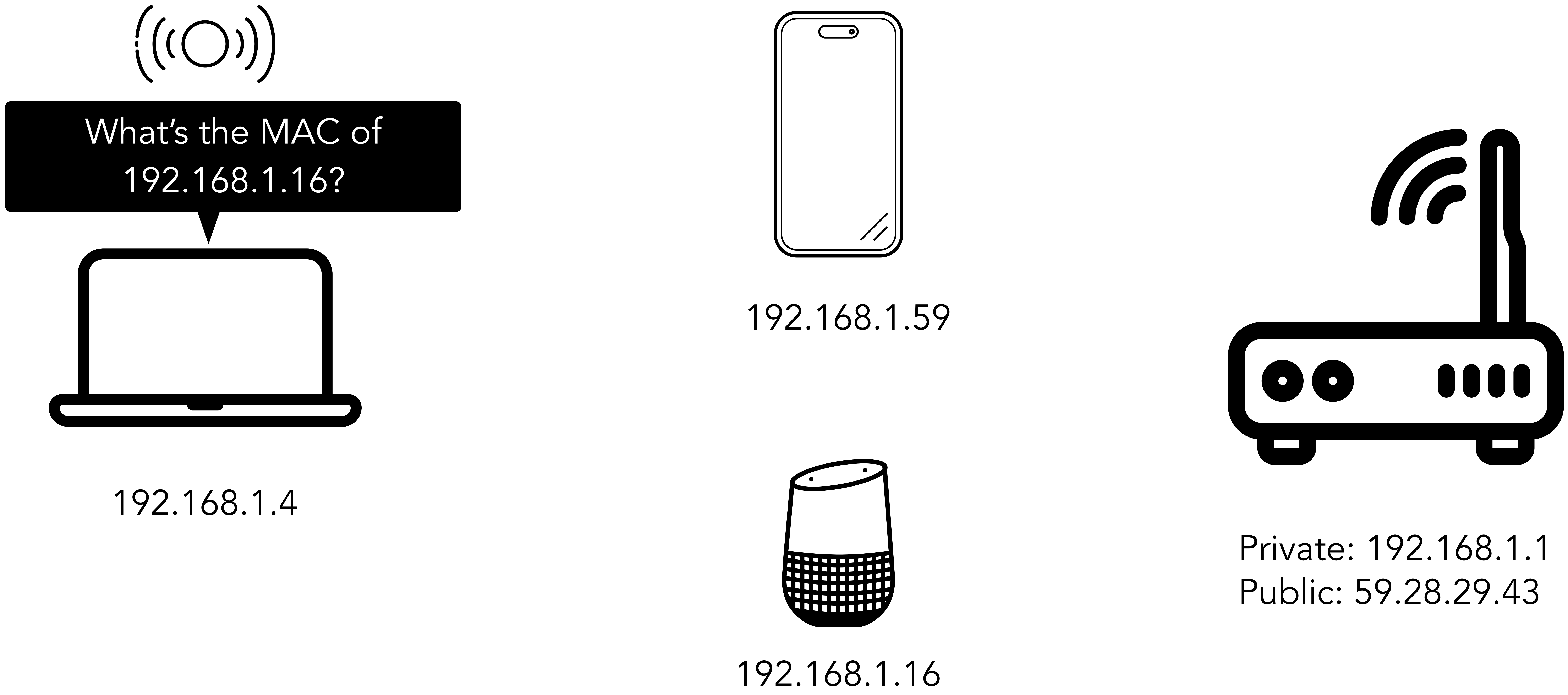
192.168.1.4

Private: 192.168.1.1
Public: 59.28.29.43

192.168.1.16

Going back to home networks

- How do devices know how to send packets to other devices?
  - Through a protocol called ARP – address routing protocol



What's the MAC of 192.168.1.16?

192.168.1.4

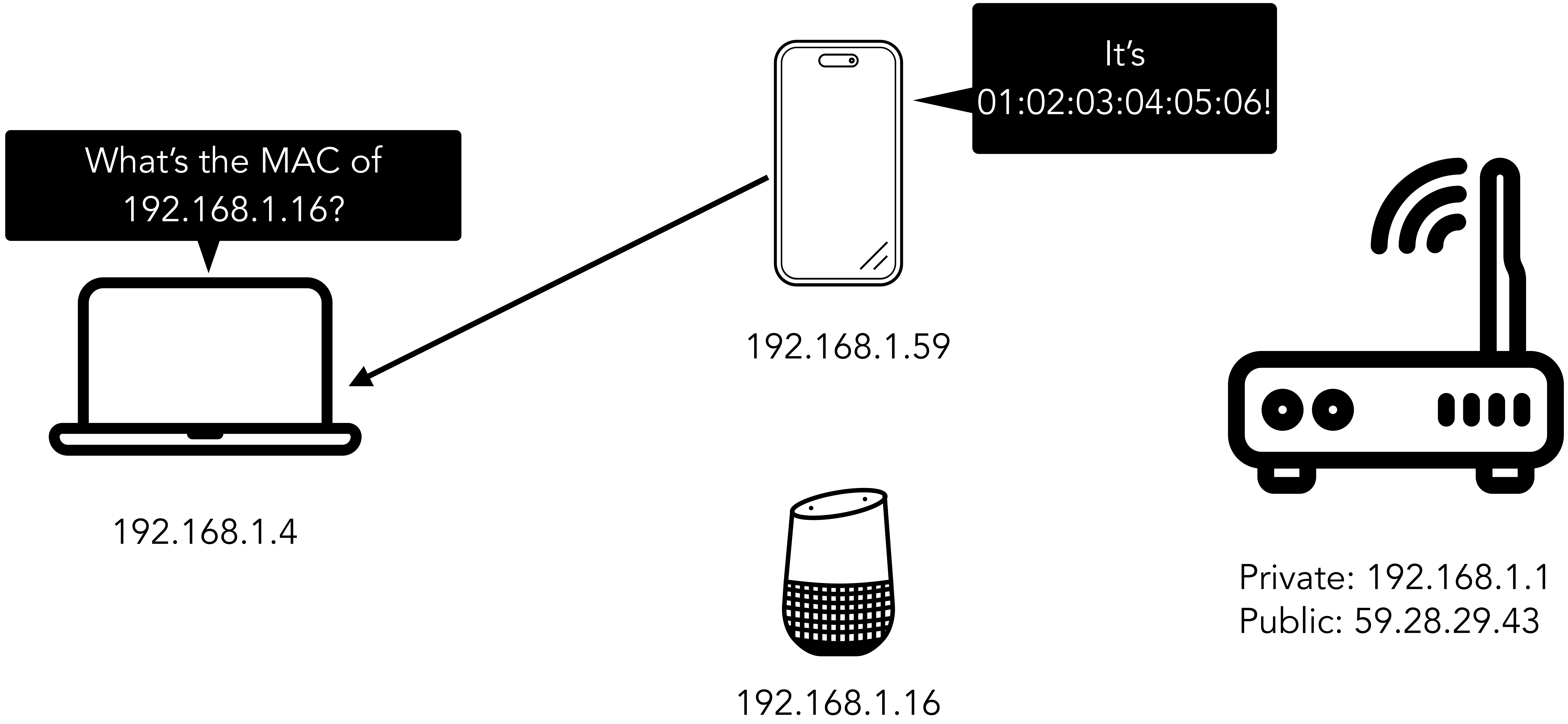
192.168.1.59

192.168.1.16

Private: 192.168.1.1
Public: 59.28.29.43

Going back to home networks

- How do devices know how to send packets to other devices?
  - Through a protocol called ARP – address routing protocol



It’s 01:02:03:04:05:06!

What’s the MAC of 192.168.1.16?

192.168.1.59

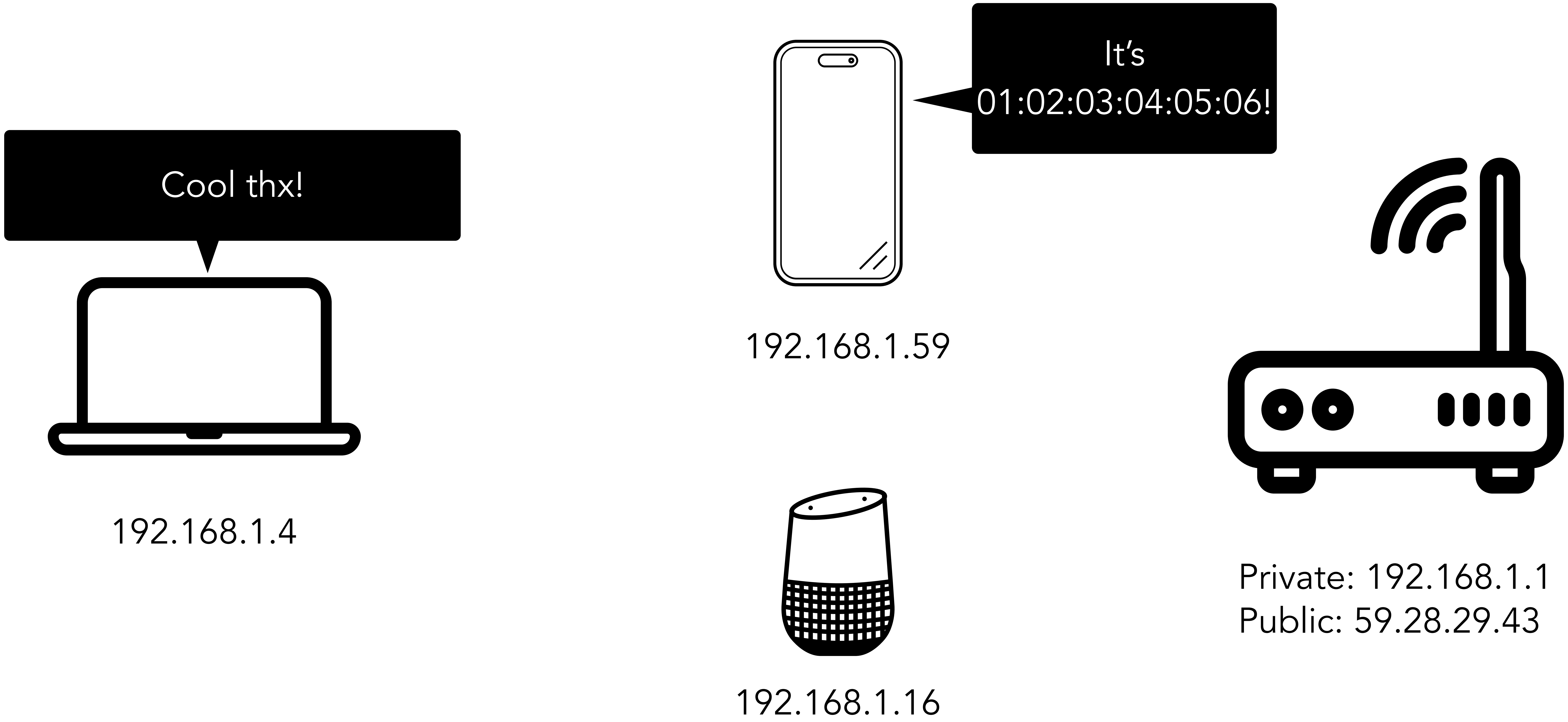
192.168.1.4

Private: 192.168.1.1
Public: 59.28.29.43

192.168.1.16

Going back to home networks

- How do devices know how to send packets to other devices?
  - Through a protocol called ARP – address routing protocol



It’s 01:02:03:04:05:06!

Cool thx!

192.168.1.59

192.168.1.4

192.168.1.16

Private: 192.168.1.1
Public: 59.28.29.43

Since this is unprotected, it can be spoofed

- ARP requests are broadcast, and anyone can send an ARP reply
- Attacker on a local network can **impersonate any other host**
  - E.g., Who has the MAC address for IP address X?
  - Um, me? Send traffic to me! (ARP proxying)
- Attack is known as *ARP Spoofing*
- Most local networks remain vulnerable (the entire home network is considered a **trust boundary**)
- ARP spoofing is a classic example of a MiTM attack, but also…

Addressing is hard

- The same “ARP Spoofing” style attack repeats at almost every protocol layer
  - A curse of the abstraction
- Source needs to send something to the destination
  - How to know which address corresponds to the identifier?
    - Domain to IP address failures (next time)
    - IP to ethernet (ARP)
    - IP routing in general…

Addressing is hard

- The same “ARP Spoofing” style attack repeats at almost every protocol layer
  - A curse of the abstraction
- Source needs to send something to the destination
  - How to know which address corresponds to the identifier?
    - Domain to IP address failures (next time)
    - IP to ethernet (ARP)
    - **IP routing in general…**

Network routing in one slide

- Say I want to send a packet to 1.1.1.1…
  - Step 1: Is host on local network?
    - If yes, send directly
    - If no, send via default gateway (router)
  - Step 2: Create IP packer
  - Step 3: Create and send link-layer (e.g., Ethernet) frame
  - Step 4: Gateway picks next router in path and forwards the IP packet
    - How does the router know where to send the packet?

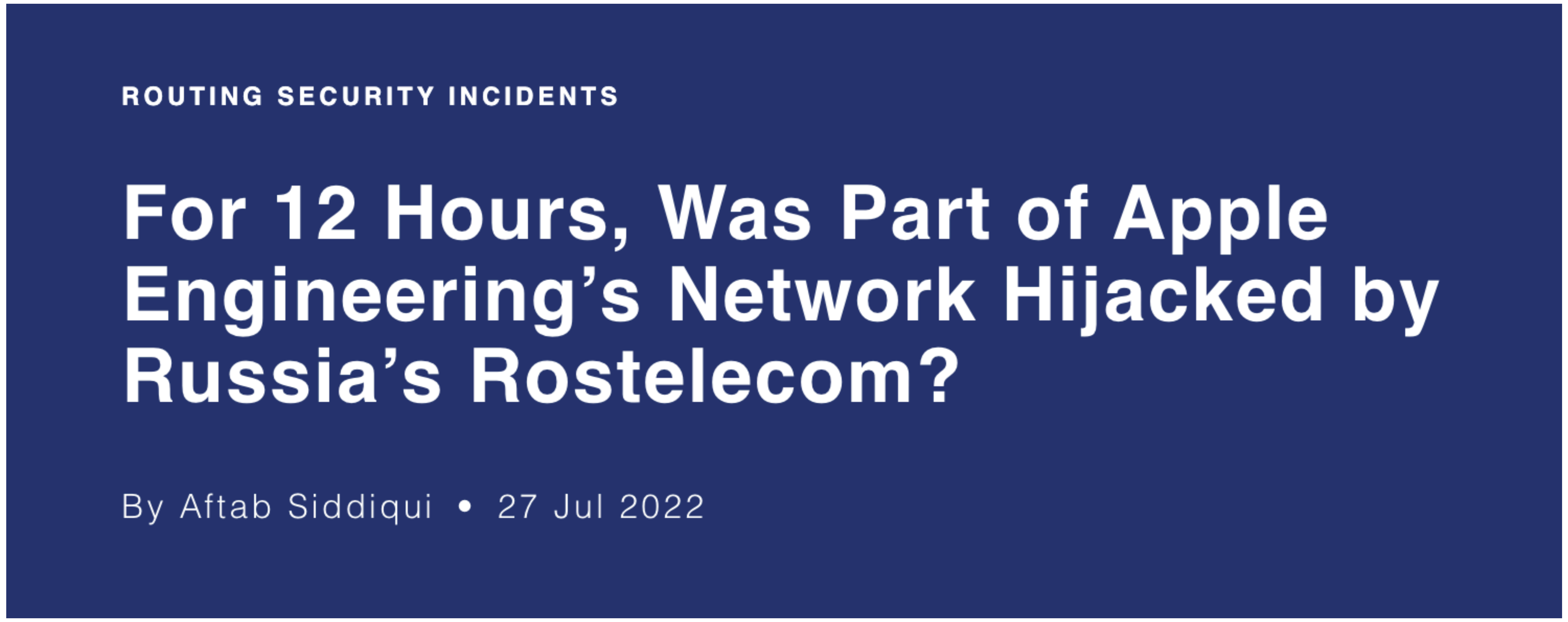
Border Gateway Protocol (BGP)

- BGP is used to manage IP routing information between networks
  - E.g., how UCSD knows to pass to AT&T who knows to pass to Merit… etc.
- Each BGP node maintains connections to a set of trusted neighbors (with weak authentication)
- Neighbors share routing information
  - E.g., I can reach Meta directly, I can reach Meta via Merit, etc.
- **No authorization here**
  - Bad BGP nodes may provide incorrect routing information that redirects IP traffic

BGP Hijacking

KLAYSWAP|KLAYSWAP-BGP-HIJACK

KlaySwap crypto users lose funds after BGP hijack



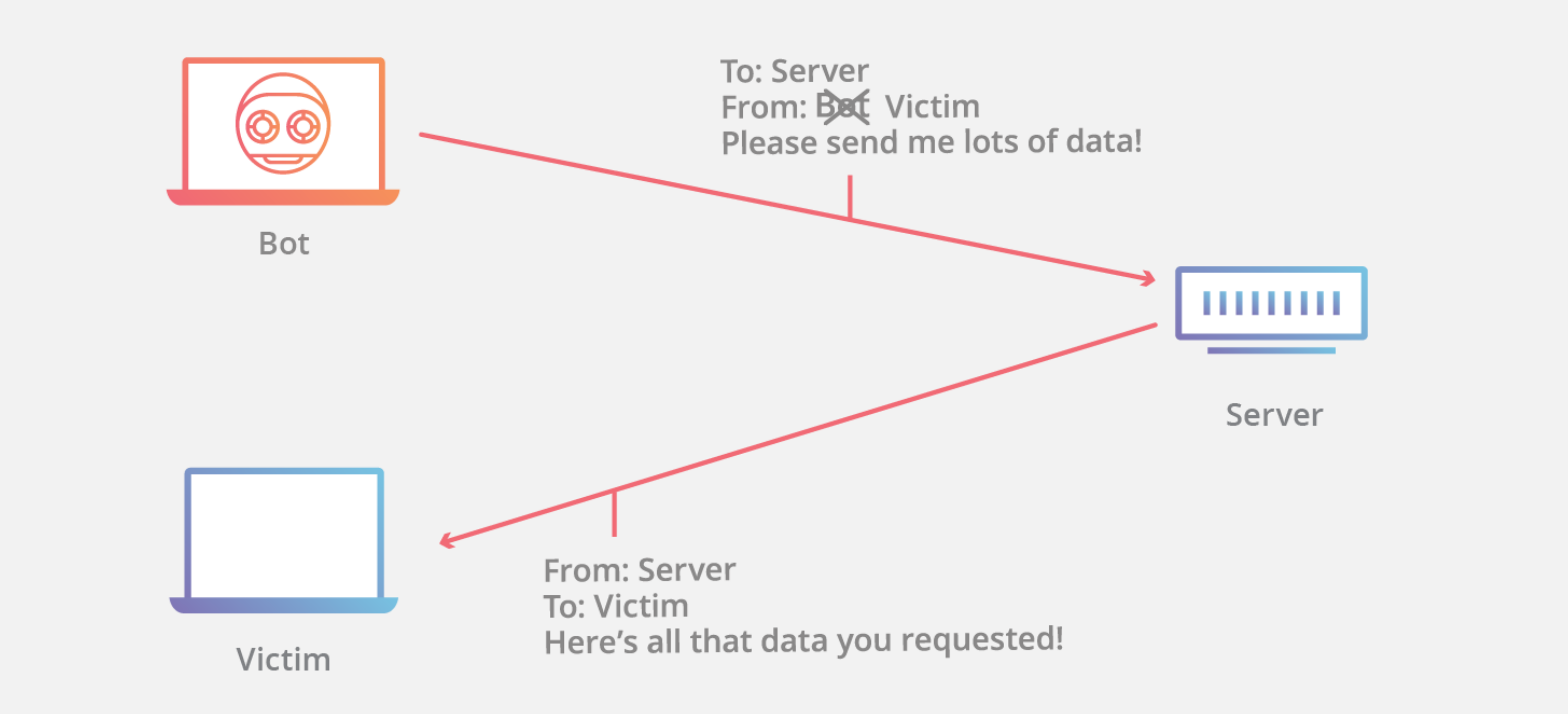
ROUTING SECURITY INCIDENTS

For 12 Hours, Was Part of Apple Engineering's Network Hijacked by Russia's Rostelecom?

By Aftab Siddiqui • 27 Jul 2022

IP Spoofing!

- Even if routing is correct, Mallory can still spoof Alice’s IP address
  - Mallory can send IP packets claiming to be from Alice
  - Mallory may not be able to receive IP packets addressed to Alice, but maybe that’s fine… basic DDoS attacks are enabled this way! More next time.



https://www.cloudflare.com/learning/ddos/glossary/ip-spoofing/

In sum…

- Networks were not designed with security in mind, and many assumptions in networks still rely on old trust models
    - E.g., no protection inside local networks, no confidentiality built in to network transport
- Much of security is **stapled on later**, and sometimes it remains unstapled (e.g., BGP hijacks are still very much a thing today)!
    - My read: it’s kind of a miracle nothing explodes
- We will continue to discuss ways we can exploit and defend networks after the midterm

Next time

- Midterm! Good luck studying!