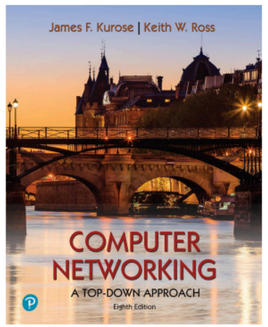
CS 241: Systems Programming
Lecture 26. Networking

Fall 2025

Prof. Stephen Checkoway



Slides adapted from the slides that accompany this book

Computer Networking: A Top-Down Approach
8th edition
Jim Kurose, Keith Ross
Pearson, 2020

Why learn about networks?

Networks are everywhere!

- Every time you connect to an online service, you are sending and receiving data through multiple computer networks.
- Most large-scale online services are applications that use computer networks to communicate and collectively processes millions of requests per second.

Our world is reliant on computer networks for its day-to-day operation

Our dependence on computer networks is only going to grow over time!

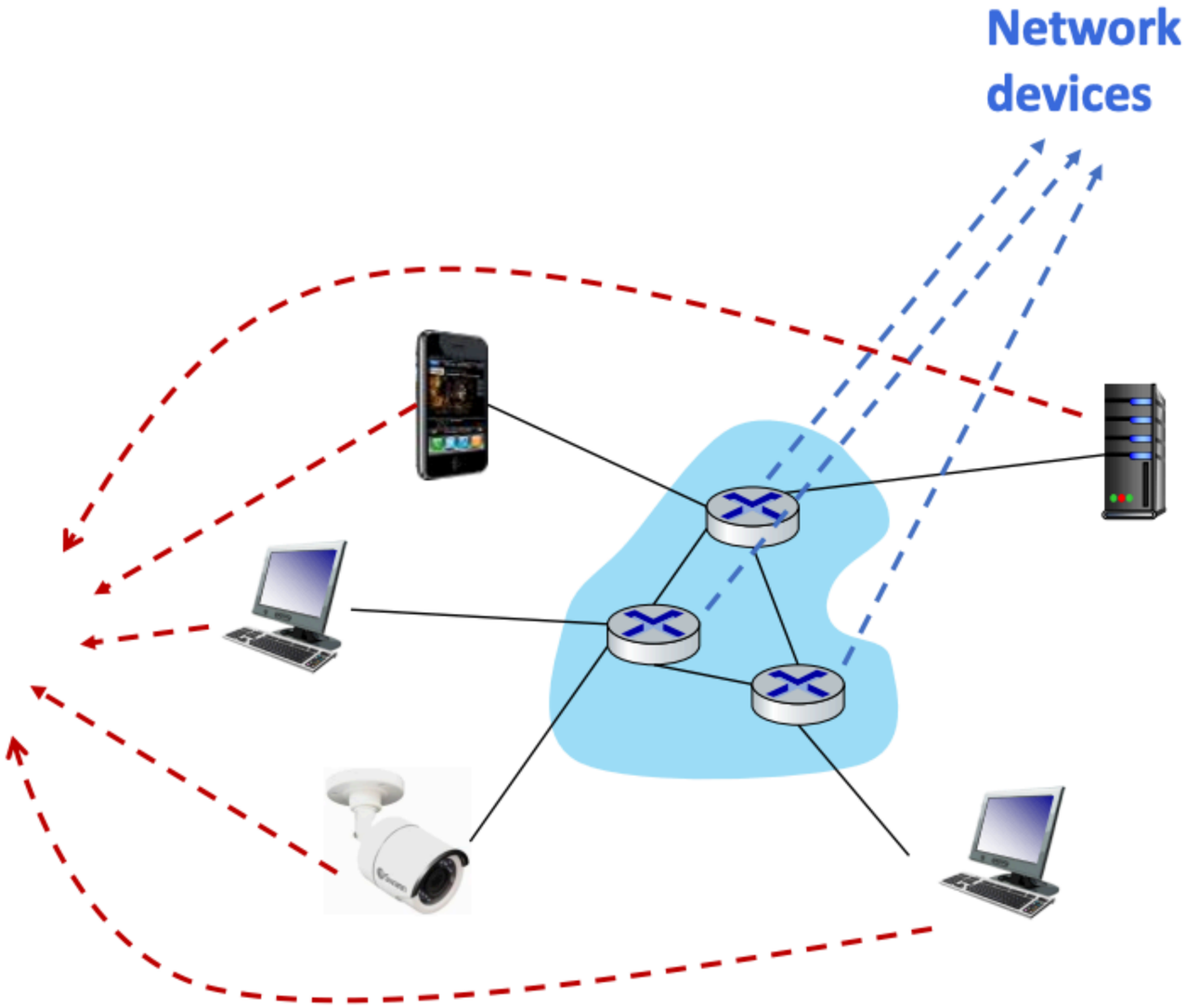
What is a network?

Networks are systems!

- A system that enables computers to exchange information

The information exchange is governed by *protocols*

Data is transmitted in small chunks called *packets*



Protocols

The information exchange is governed by *protocols*

- Protocols are standardized rules that tell us what messages should look like and how to respond to messages

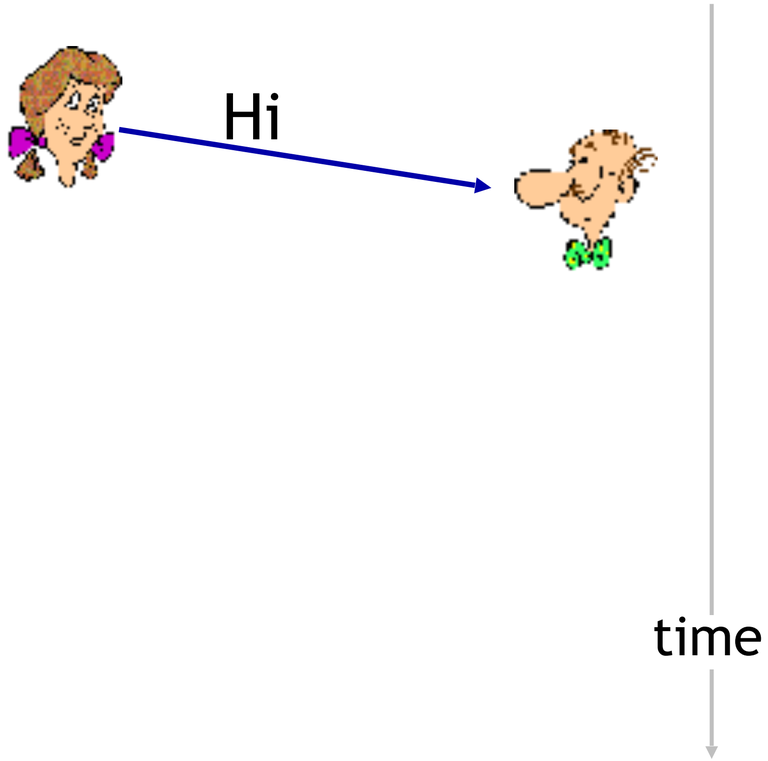
Imagine trying to send mail without standardized addresses!

Protocols are necessary to allow us to communicate without having to establish rules first

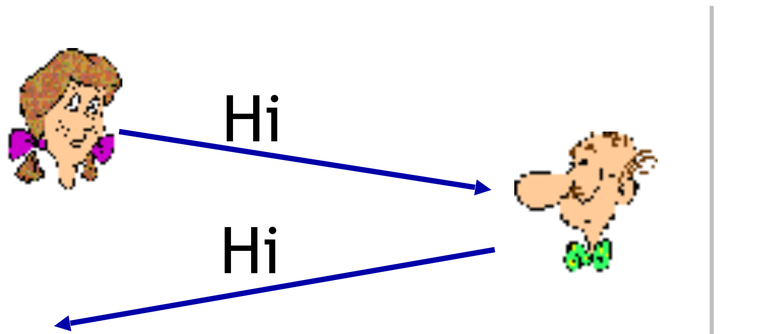
Human Protocols

time

Human Protocols

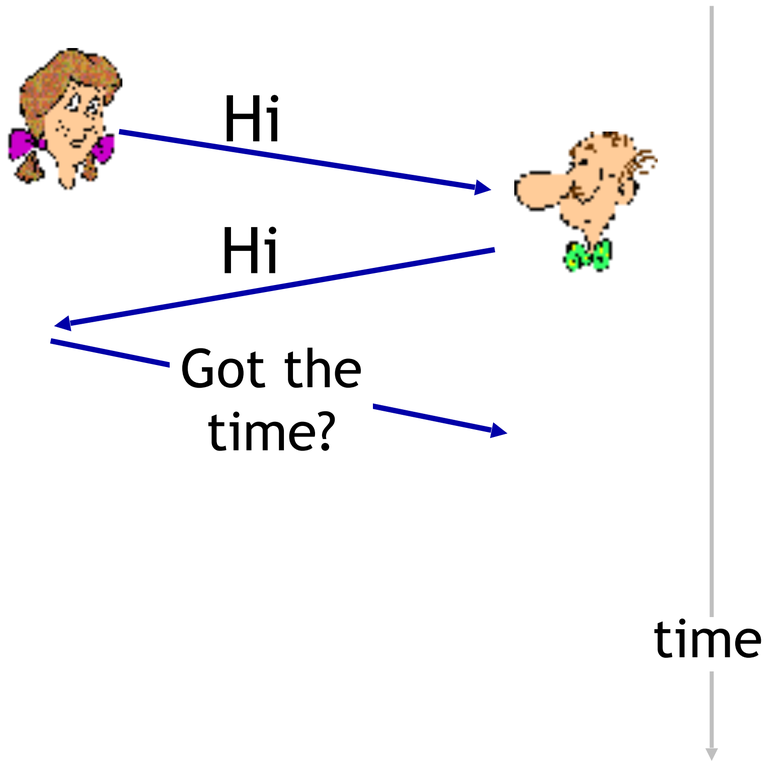


Human Protocols

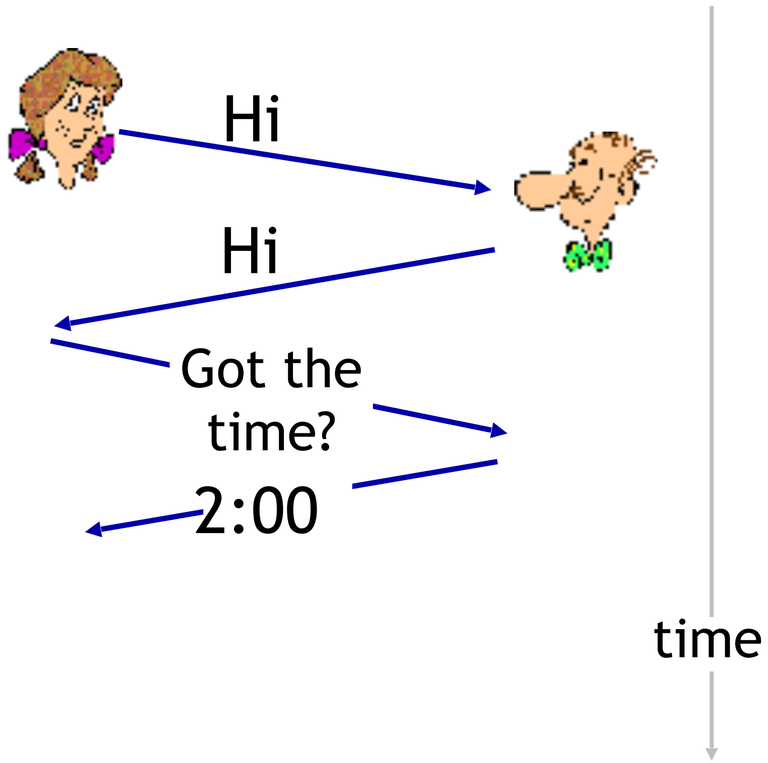


time

Human Protocols



Human Protocols



Protocol Message

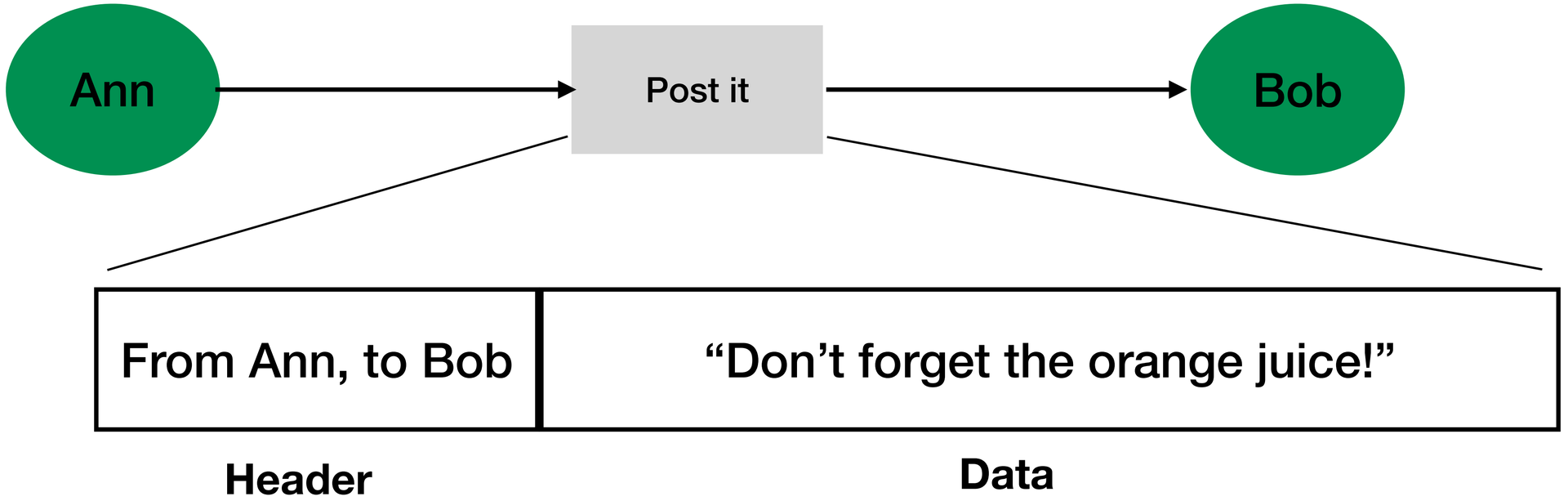
Message: contains header and data

Data: what receiver wants to know

Header: information to support protocol

- Source and destination address (more on this later!)
- State of protocol operation
- Error control (to check integrity of received data)

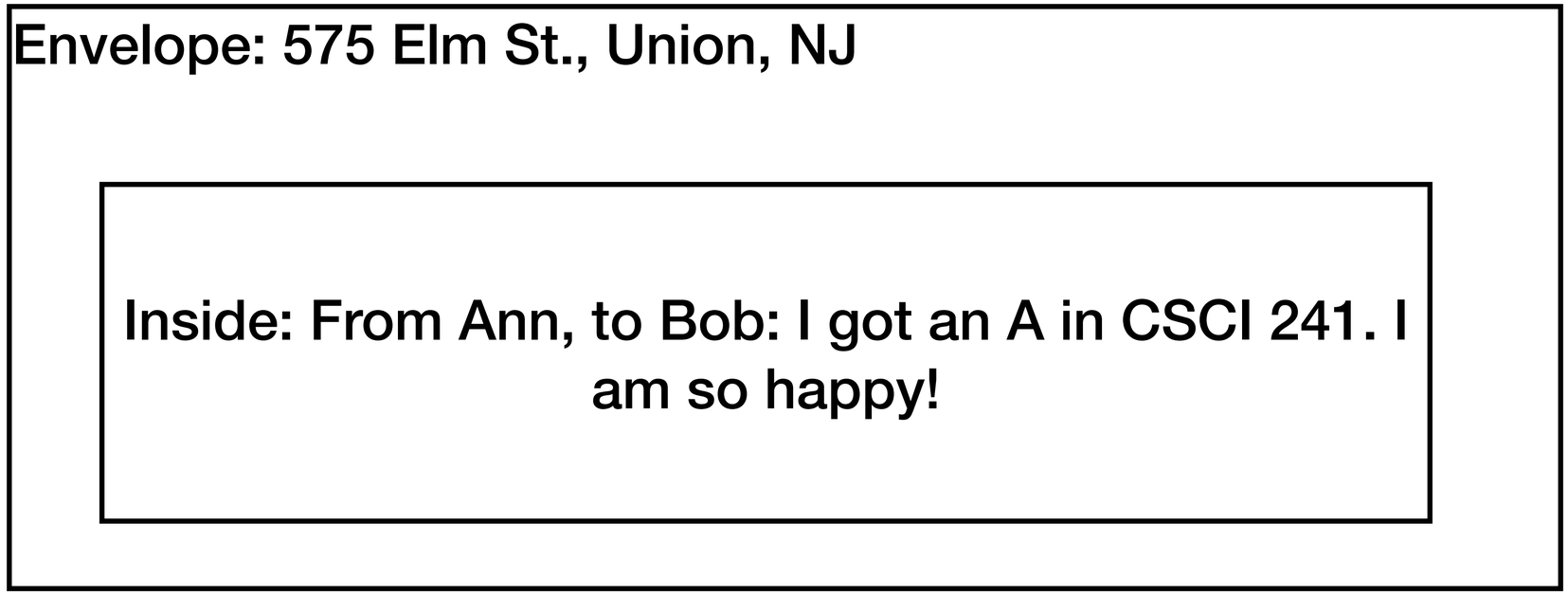
Example: Ann sends message to Bob



Protocol:

- Message format: (from, to), message content
- Transfer procedure: post on refrigerator

Suppose Ann is mailing the post-it to Bob. What's the header now?

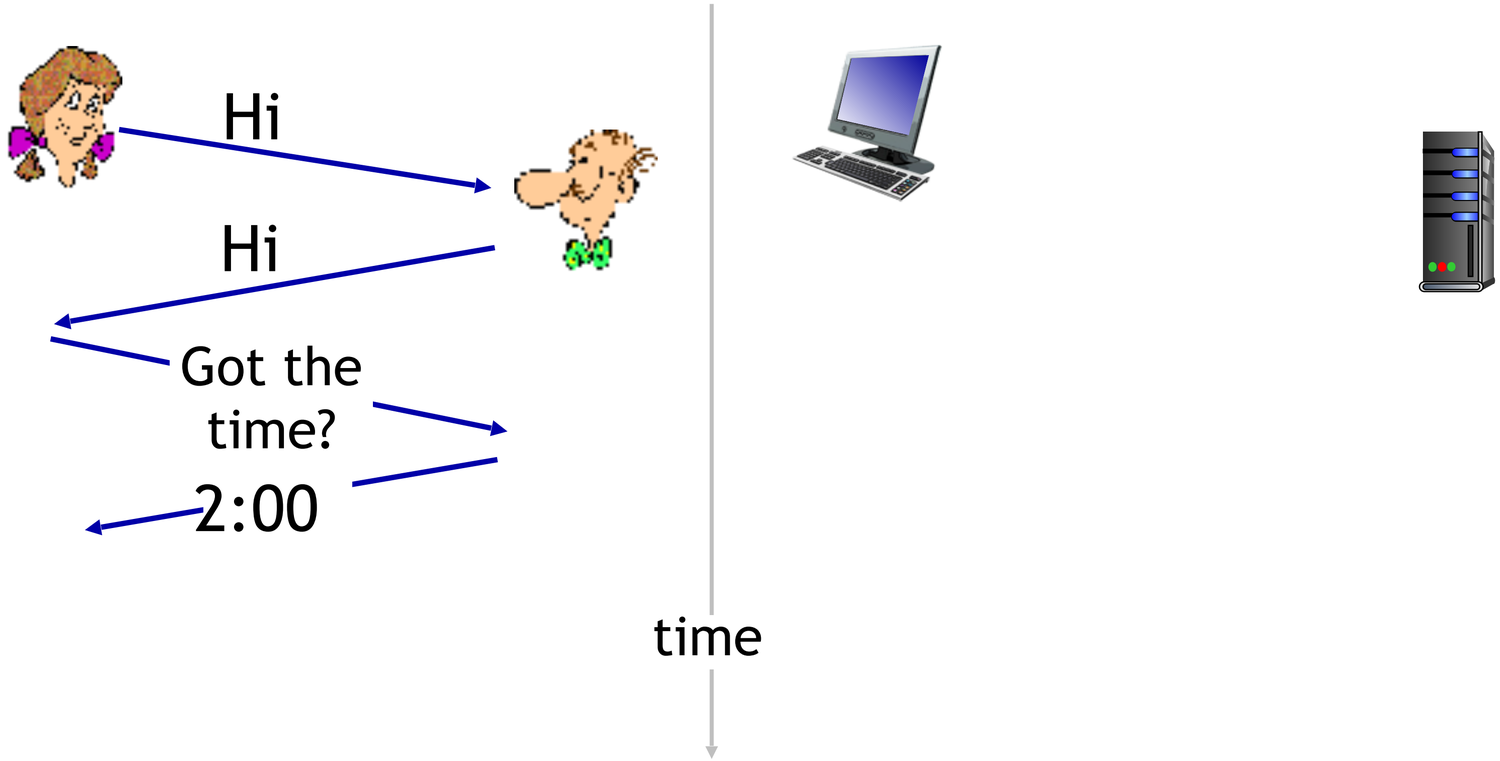


A. The address on the envelope

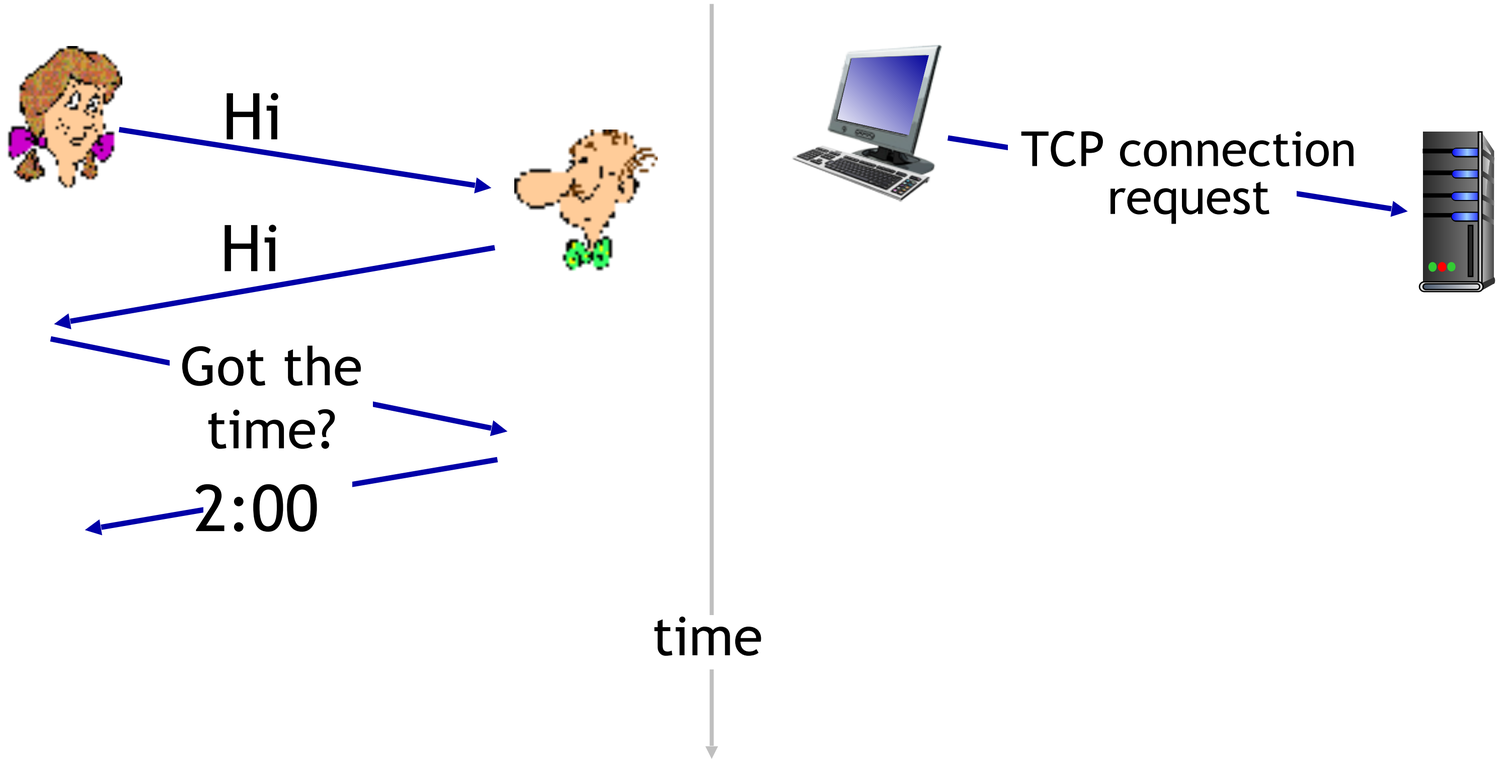
B. The "from Ann, to Bob"

C. Something else

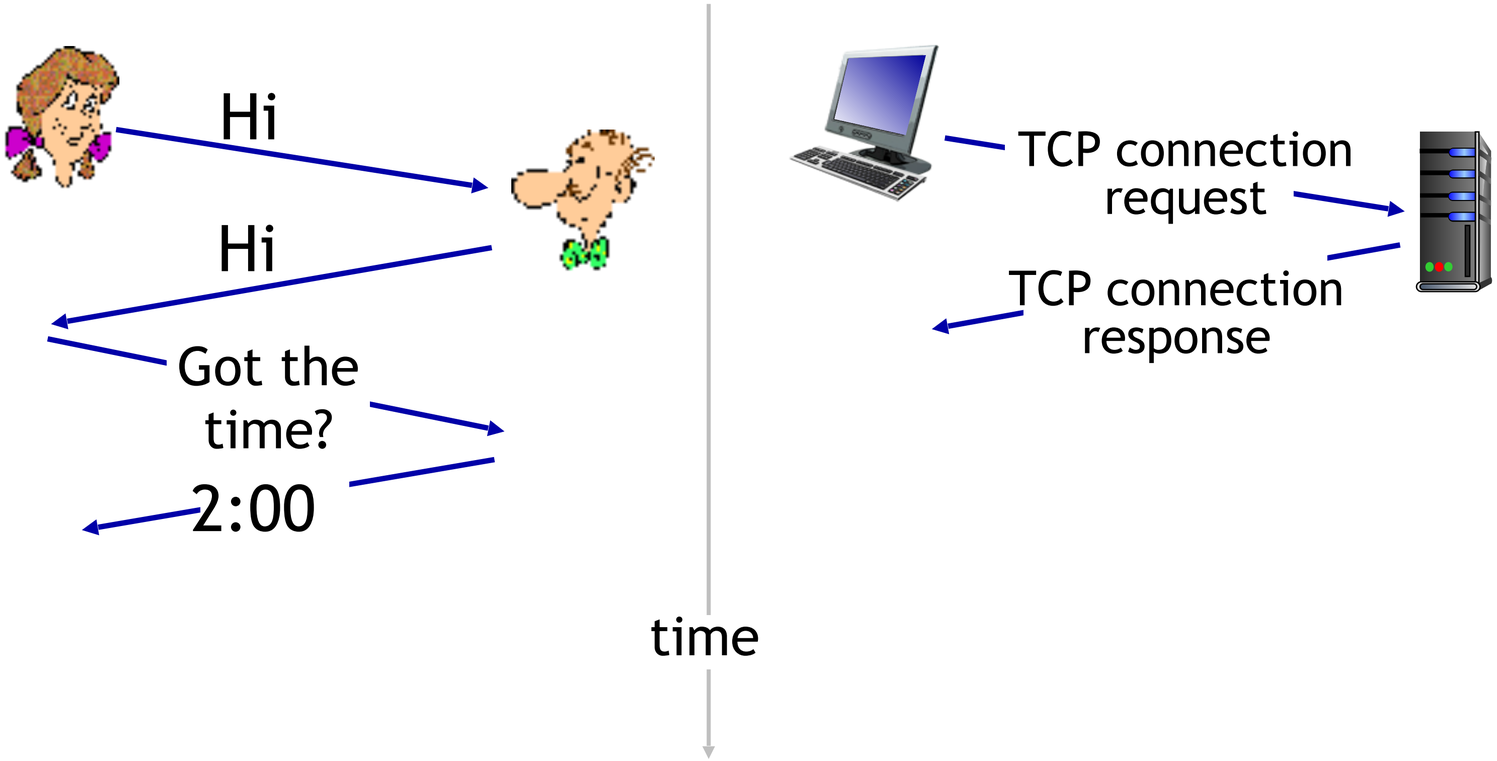
Network Protocols



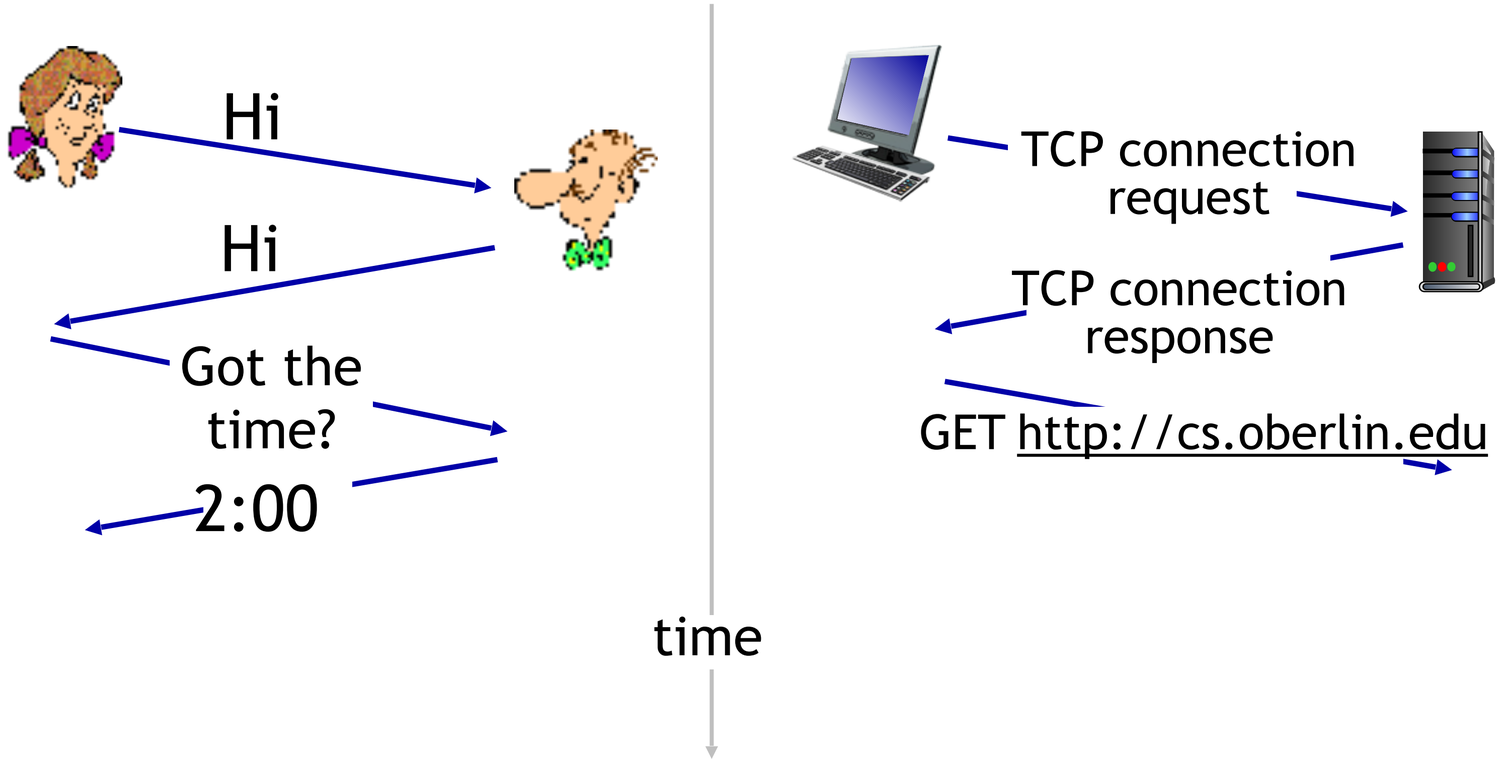
Network Protocols



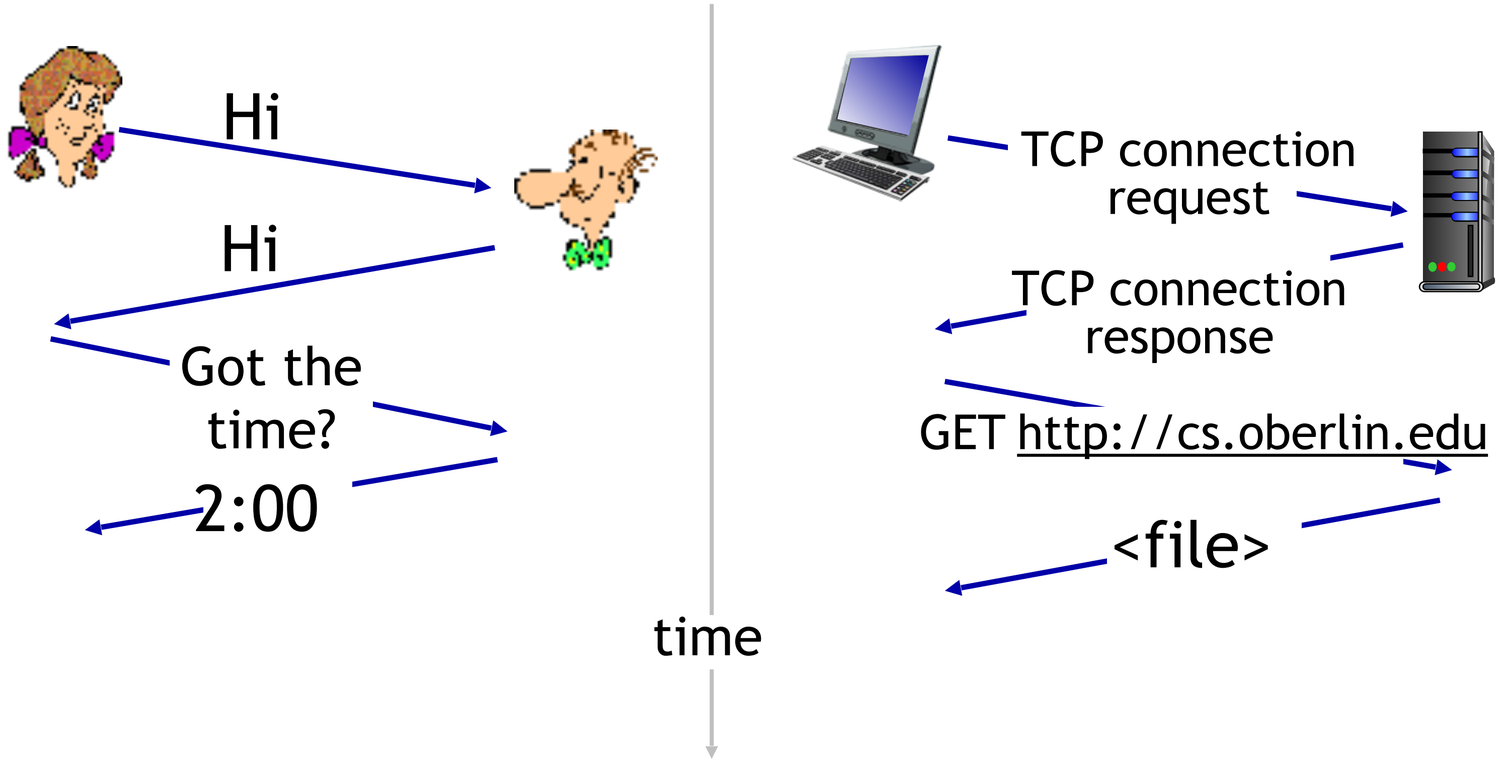
Network Protocols



Network Protocols



Network Protocols

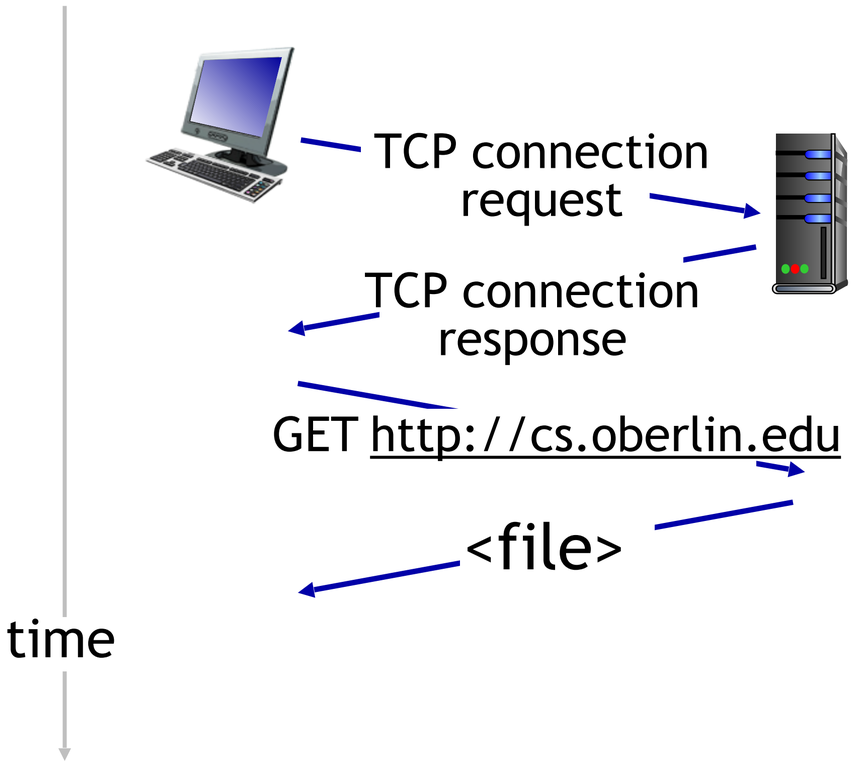


Network Protocols

Network protocols are between computers (devices) instead of humans

A protocol defines:

- the ***format*** and ***order*** of messages sent/received between network entities
- the ***actions*** taken upon message receipt



Protocol Examples

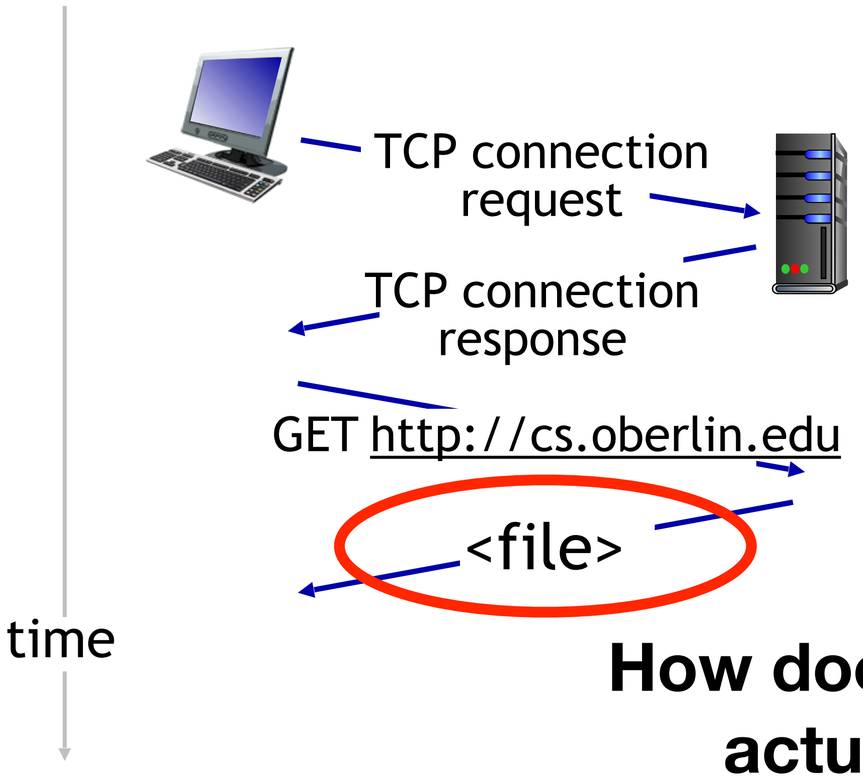
HTTP	**Hypertext Transfer Protocol**; communication between web servers and web browsers
DNS	**Domain Name System**; translates IP addresses to URLs
SSH	**Secure Shell**; secure connections between clients and servers
DHCP	**Dynamic Host Configuration Protocol**; assigns IP addresses to devices when they connect to a network
SMTP	**Simple Mail Transfer Protocol**; email!

Network Protocols

Network protocols are between computers (devices) instead of humans

A protocol defines:

- the ***format*** and ***order*** of messages sent/received between network entities
- the ***actions*** taken upon message receipt



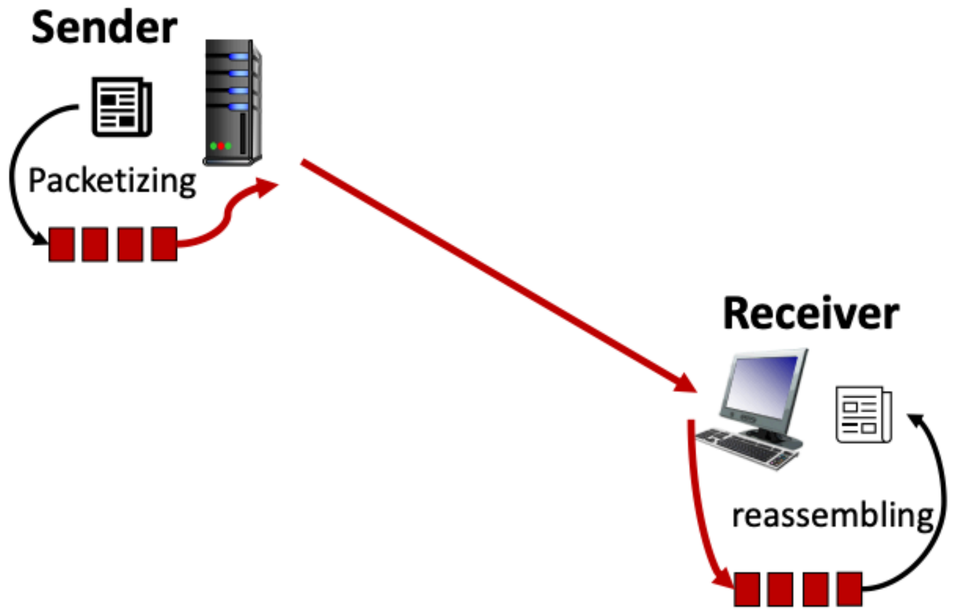
How does the data actually get transmitted?

Packets: Data Transmission Units

Sender breaks data into smaller chunks

These are known as ***packets***, and they have a standard format (defined by the protocol)

Receiver receives data chunks and reassembles the data



What are the benefits of transmitting data in smaller chunks?

Think of what would happen if you tried to send all of the data at once. What happens if the data gets lost before it gets to the receiver?

Think of what happens in your OS if a process had to completely finish executing before the next one could start. Can you apply that to packet transmission?

A. Select A

Want to see what protocols you're using? Try Wireshark!

Source	Destination	Protocol	Length	Info
192.168.1.184	132.162.201.24	TCP	66	49829 → 22 [ACK] Seq=1 Ack=1 Win=131712 Len=0 TSval=558373617 TSecr=1694129657
192.168.1.184	132.162.201.24	SSHv2	87	Client: Protocol (SSH-2.0-OpenSSH_8.1)
132.162.201.24	192.168.1.184	SSHv2	112	Server: Protocol (SSH-2.0-OpenSSH_7.6p1 Ubuntu-4ubuntu0.7+esm3)
192.168.1.184	132.162.201.24	TCP	66	49829 → 22 [ACK] Seq=22 Ack=47 Win=131712 Len=0 TSval=558373657 TSecr=1694129697
132.162.201.24	192.168.1.184	TCP	66	22 → 49829 [ACK] Seq=47 Ack=22 Win=65280 Len=0 TSval=1694129700 TSecr=558373617
192.168.1.184	132.162.201.24	SSHv2	1458	Client: Key Exchange Init
132.162.201.24	192.168.1.184	SSHv2	1178	Server: Key Exchange Init
192.168.1.184	132.162.201.24	TCP	66	49829 → 22 [ACK] Seq=1414 Ack=1159 Win=130560 Len=0 TSval=558373687 TSecr=1694129727
132.162.201.24	192.168.1.184	TCP	66	22 → 49829 [ACK] Seq=1159 Ack=1414 Win=64128 Len=0 TSval=1694129765 TSecr=558373660
192.168.1.184	132.162.201.24	SSHv2	114	Client: Diffie-Hellman Key Exchange Init
132.162.201.24	192.168.1.184	SSHv2	518	Server: Diffie-Hellman Key Exchange Reply, New Keys, Encrypted packet (len=172)
192.168.1.184	132.162.201.24	TCP	66	49829 → 22 [ACK] Seq=1462 Ack=1611 Win=130560 Len=0 TSval=558373763 TSecr=1694129808
192.168.1.184	132.162.201.24	SSHv2	82	Client: New Keys
132.162.201.24	192.168.1.184	TCP	66	22 → 49829 [ACK] Seq=1611 Ack=1478 Win=64128 Len=0 TSval=1694129887 TSecr=558373767
192.168.1.184	132.162.201.24	SSHv2	110	Client: Encrypted packet (len=44)
132.162.201.24	192.168.1.184	TCP	66	22 → 49829 [ACK] Seq=1611 Ack=1522 Win=64128 Len=0 TSval=1694129924 TSecr=558373841
132.162.201.24	192.168.1.184	SSHv2	110	Server: Encrypted packet (len=44)
192.168.1.184	132.162.201.24	TCP	66	49829 → 22 [ACK] Seq=1522 Ack=1655 Win=131008 Len=0 TSval=558373877 TSecr=1694129924
192.168.1.184	132.162.201.24	SSHv2	134	Client: Encrypted packet (len=68)
132.162.201.24	192.168.1.184	SSHv2	150	Server: Encrypted packet (len=84)
192.168.1.184	132.162.201.24	TCP	66	49829 → 22 [ACK] Seq=1590 Ack=1739 Win=130944 Len=0 TSval=558373914 TSecr=1694129961
192.168.1.184	132.162.201.24	SSHv2	438	Client: Encrypted packet (len=372)

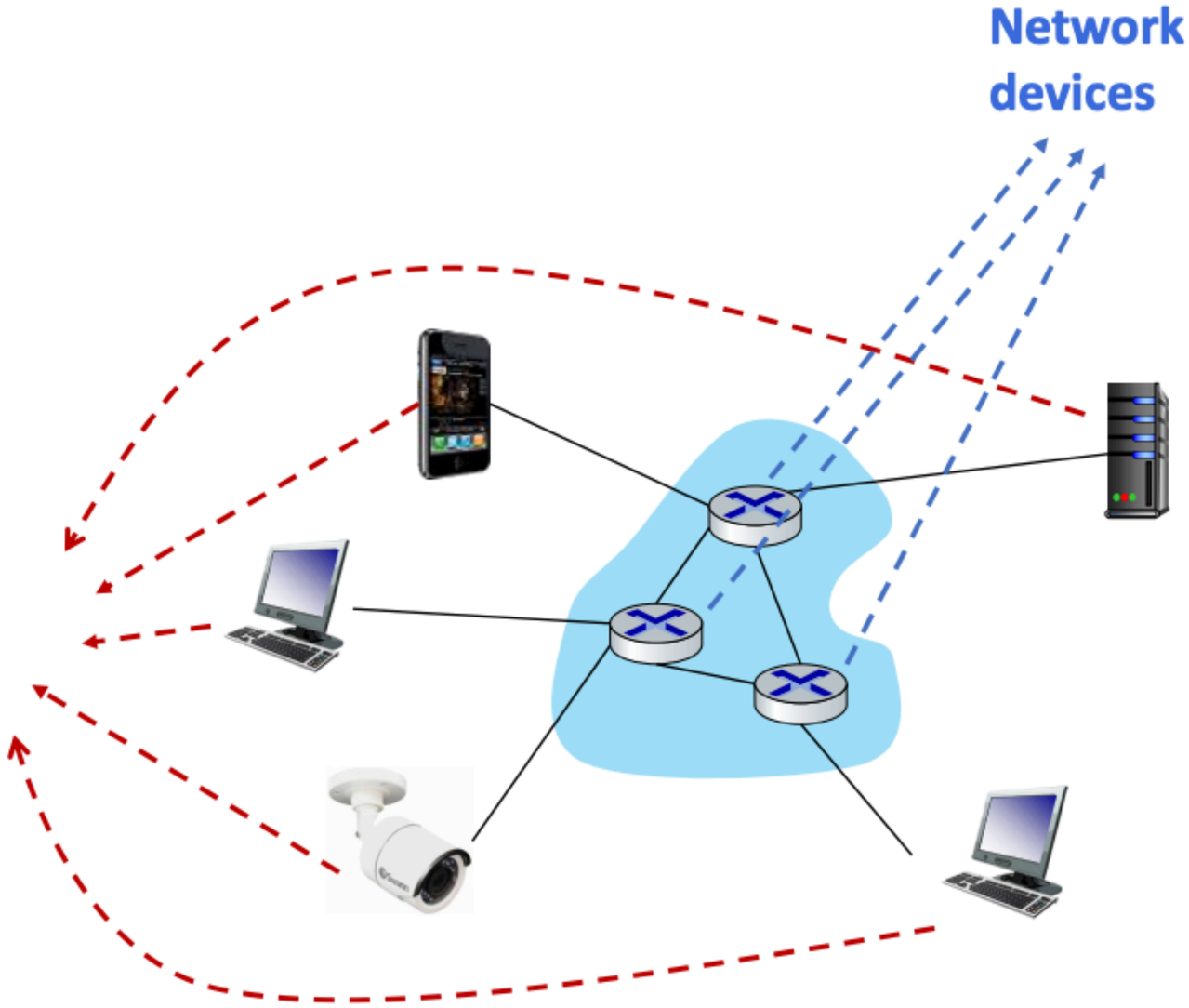
Seems simple, right?

Networks are systems!

- A system that enables computers to exchange information

The information exchange is governed by *protocols*

Data is transmitted in small chunks called *packets*



Seems simple, right?

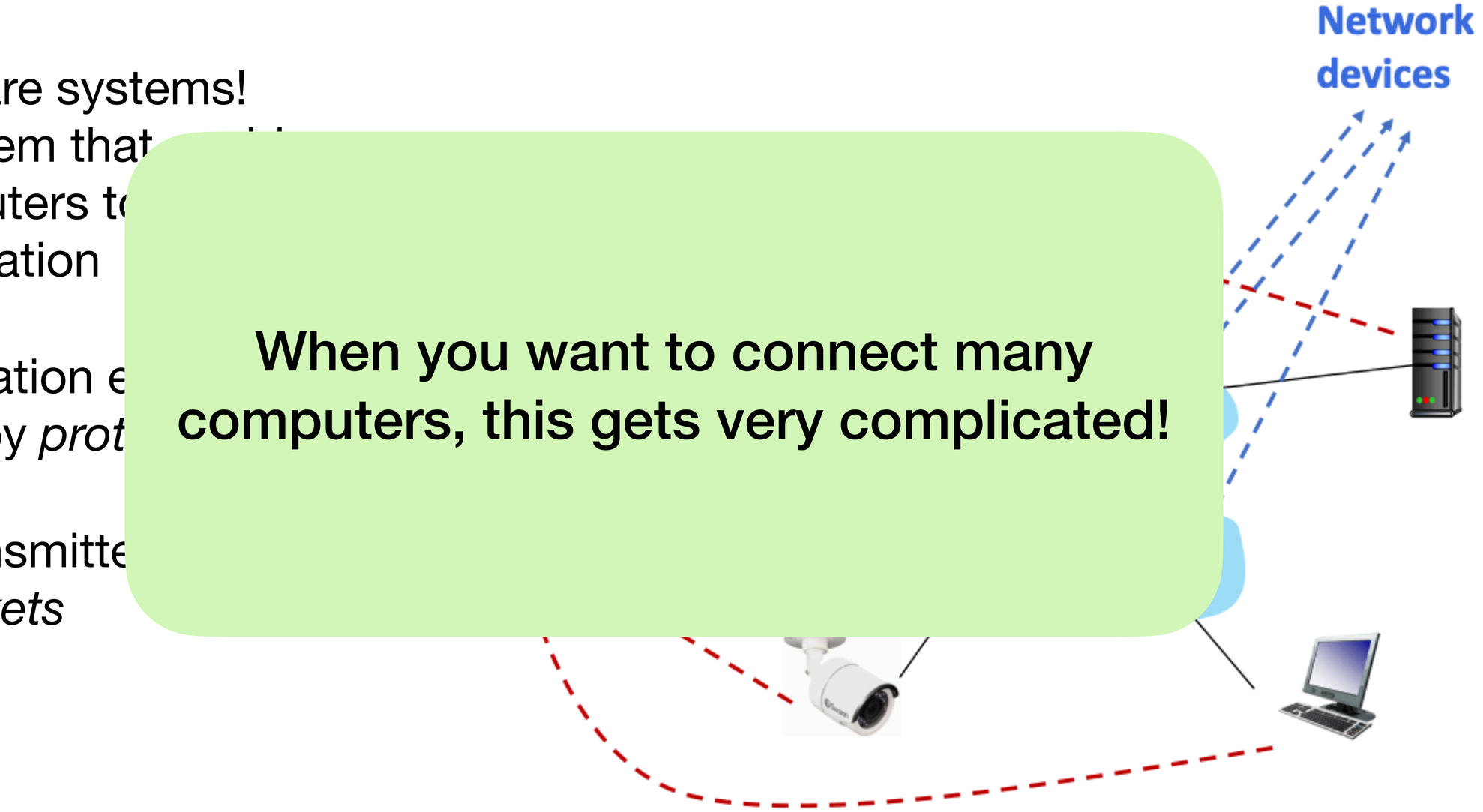
Networks are systems!

- A system that
  computers t
  information

The information e
governed by *prot*

Data is transmitte
called *packets*

When you want to connect many computers, this gets very complicated!

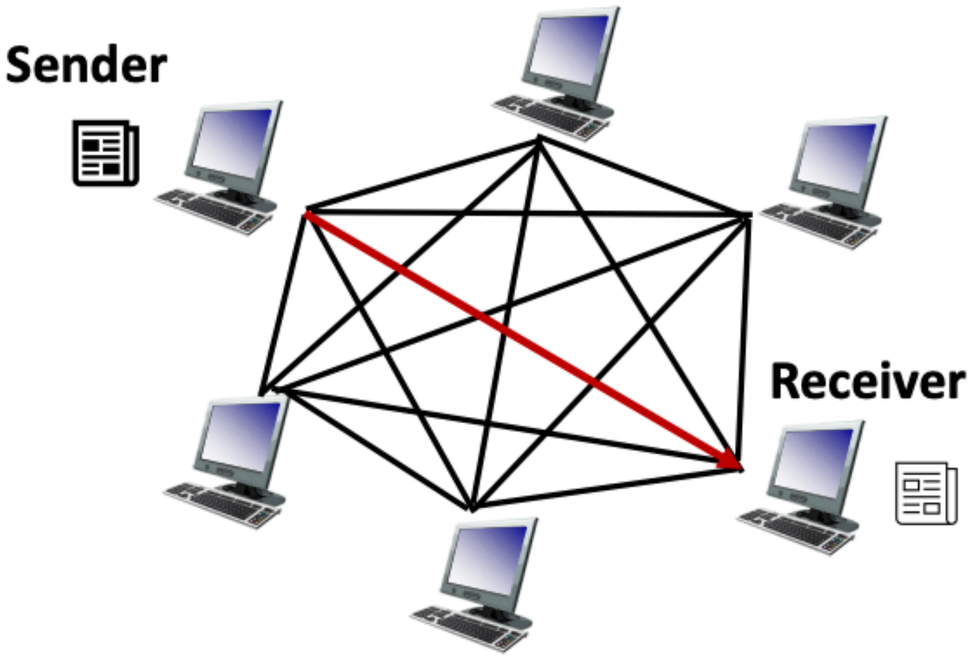


Connecting many computers

If we have *N* computers that want to communicate, we would need *N(N-1)/2* dedicated links to connect each pair

If *N* is small, this would work!

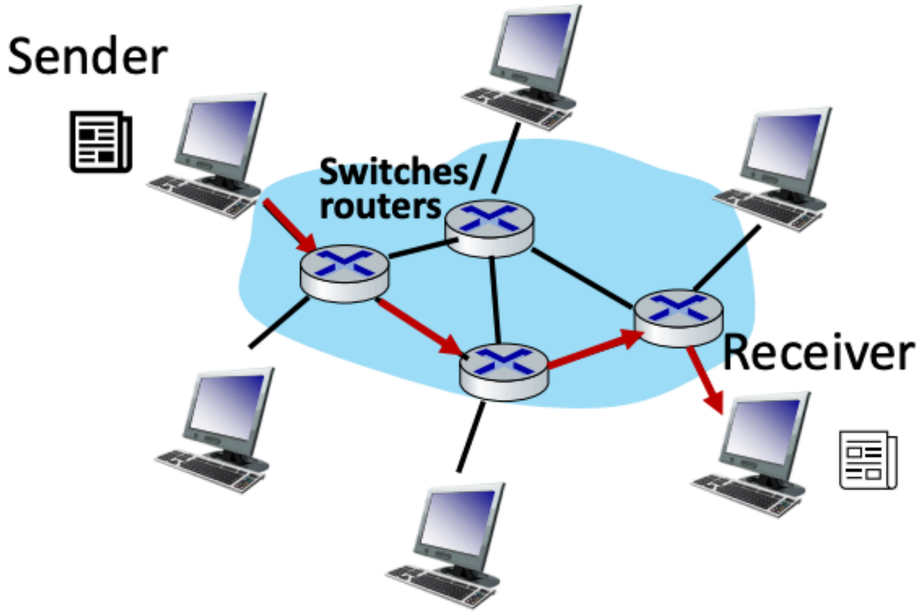
But we have millions of devices communicating at any given of time



Connecting many computers

Solution: we use a collection of **shared** network devices and links

Switches/routers relay data so it gets to the correct destination



What are the pros and cons of having shared network devices and links?

If scale wasn’t an issue, why might you choose to have dedicated links instead?

A. Select A

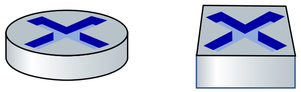
The components of the Internet

Billions of connected computing *devices*:

- *hosts* = *end systems*
- running *network apps* at Internet's "edge"

Communication links

- fiber, copper, radio, satellite
- transmission rate: *bandwidth*



Packet switches: forward packets (chunks of data)

- *routers*, *switches*

Networks

- collection of devices, routers, links: managed by an organization

“Fun” Internet-Connected Devices

Amazon Echo

IP picture frame

Pacemaker & Monitor

Tweet-a-watt:
monitor energy use

bikes

cars

Web-enabled toaster +
weather forecaster

Security Camera

Internet
refrigerator

Slingbox: remote
control cable TV

AR devices

scooters

Internet phones

Gaming devices

sensorized,
bed
mattress

diapers

More Internet Components

Internet: "network of networks"

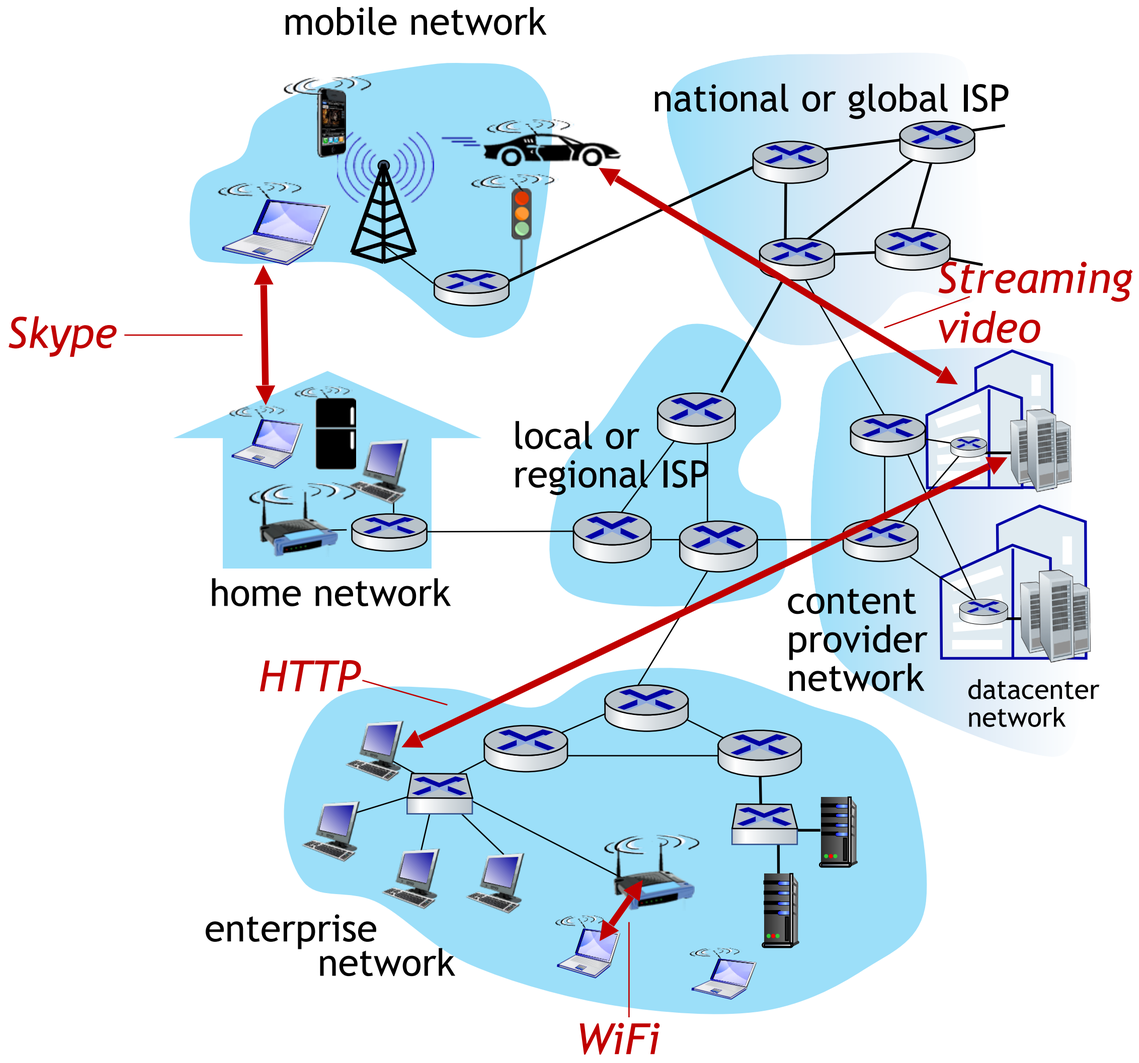
- Interconnected Internet Service Providers (ISPs)

protocols are *everywhere*

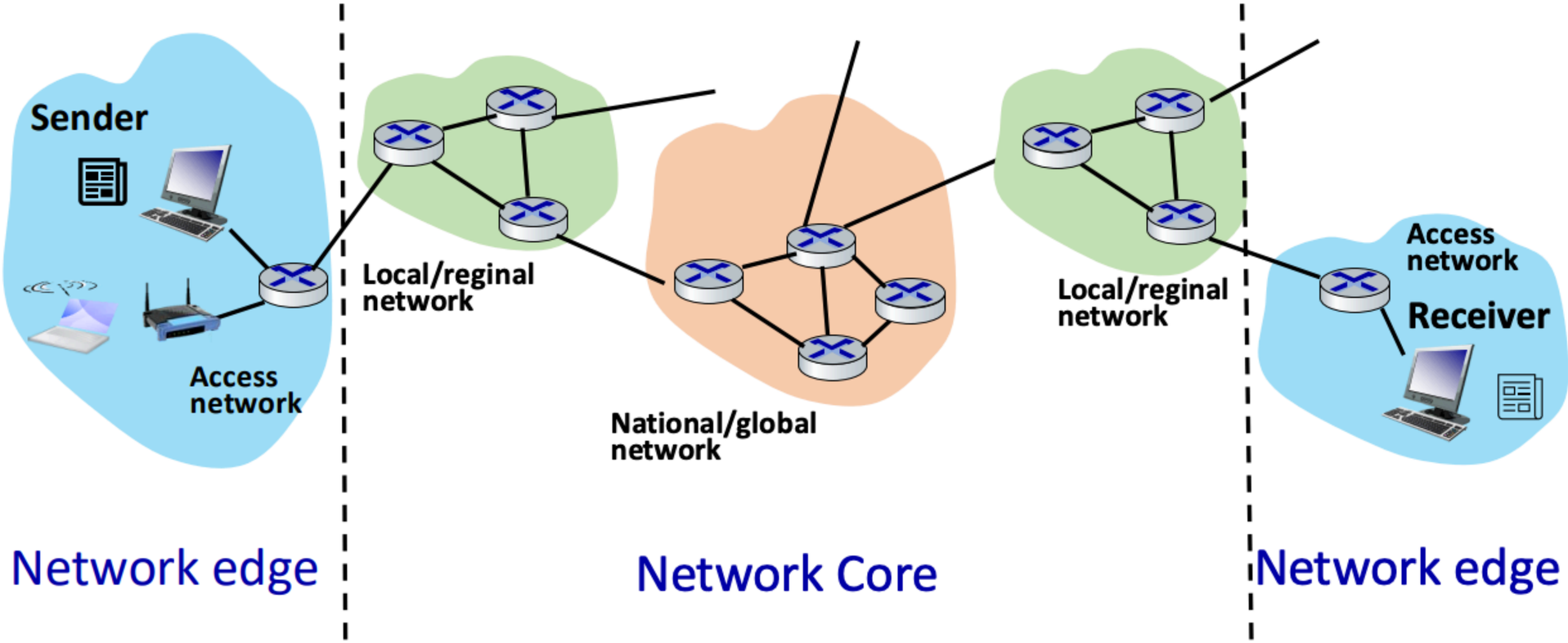
- control sending, receiving of messages
- e.g., HTTP (Web), streaming video, Skype, WiFi

Internet standards define protocols

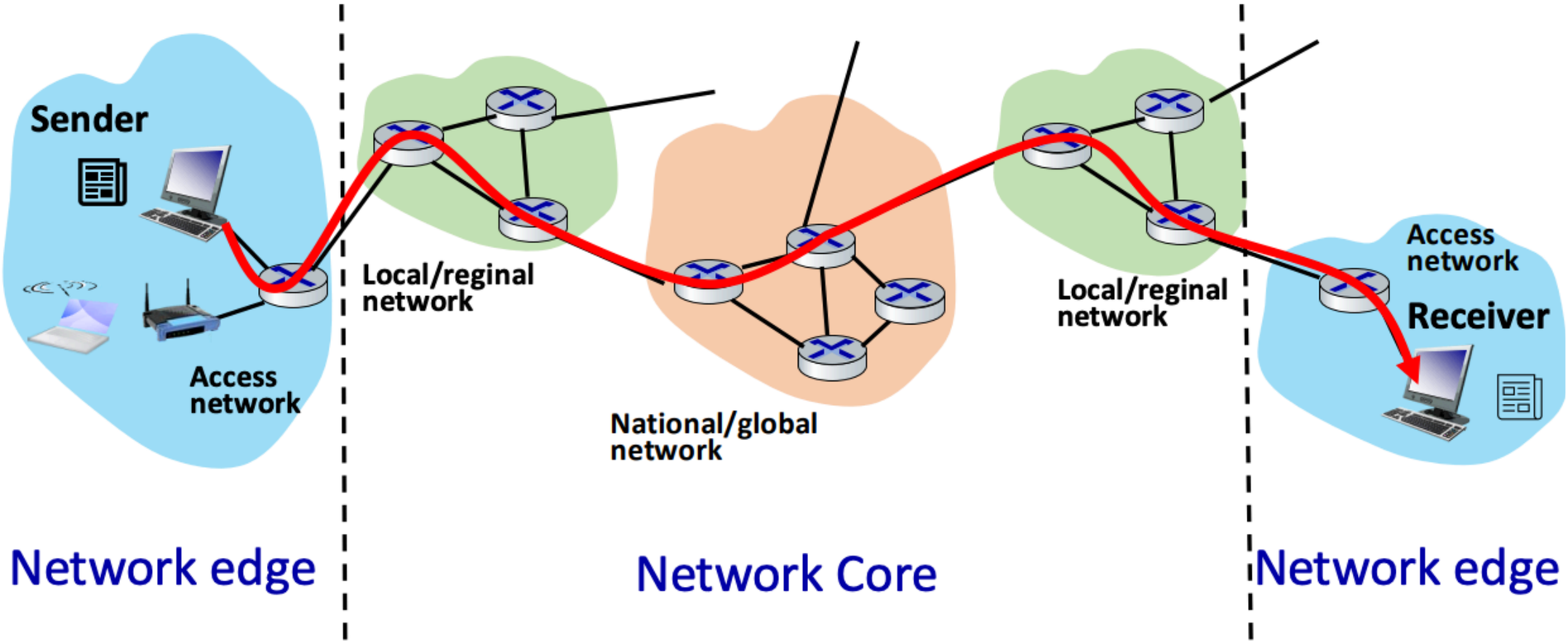
- RFC: Request for Comments
- IETF: Internet Engineering Task Force



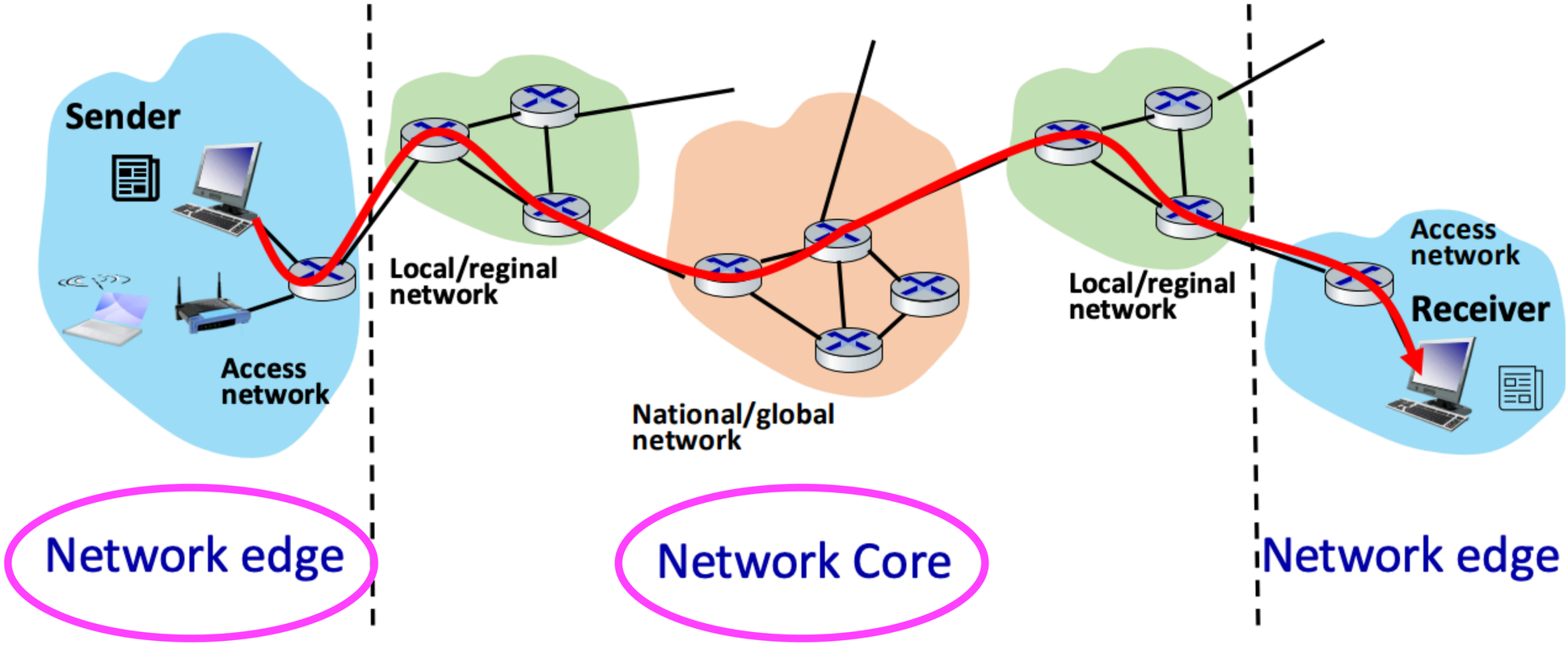
A packet passes through many networks



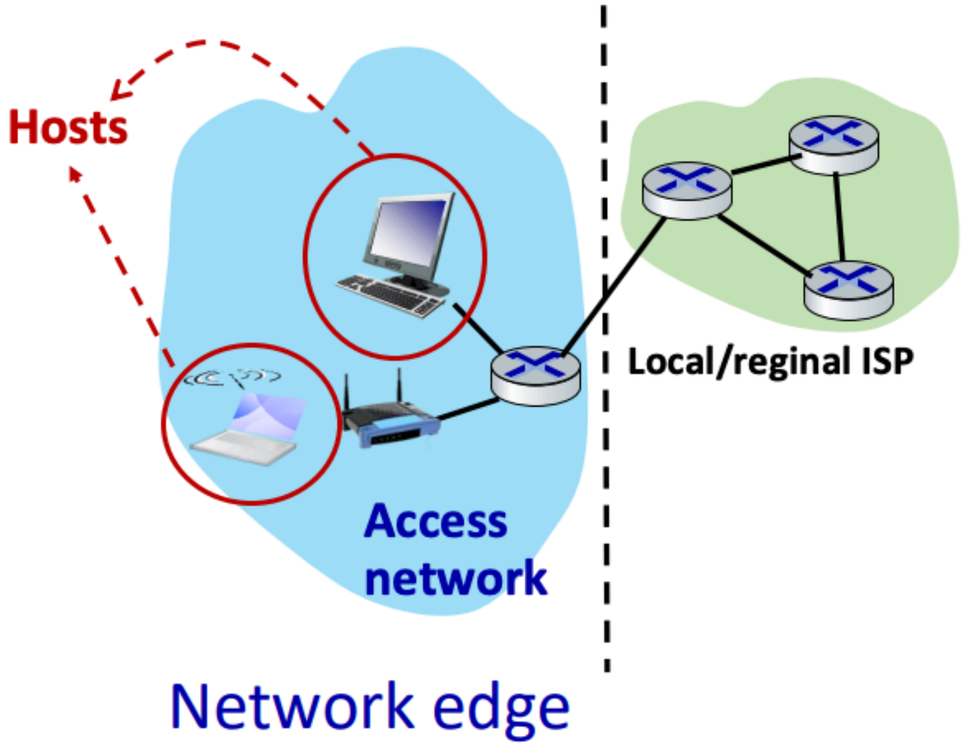
A packet passes through many networks



Types of Networks



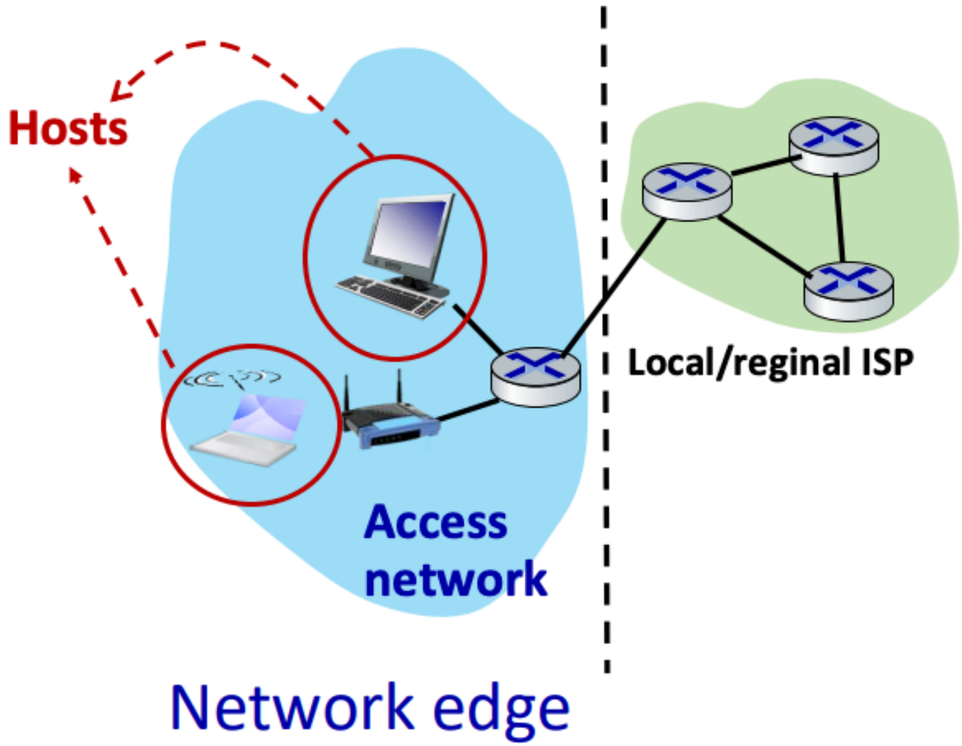
The Network Edge



The Network Edge

Hosts are computing devices connected to the network (e.g., phones, laptops, etc.)

- They run applications that communicate with other hosts



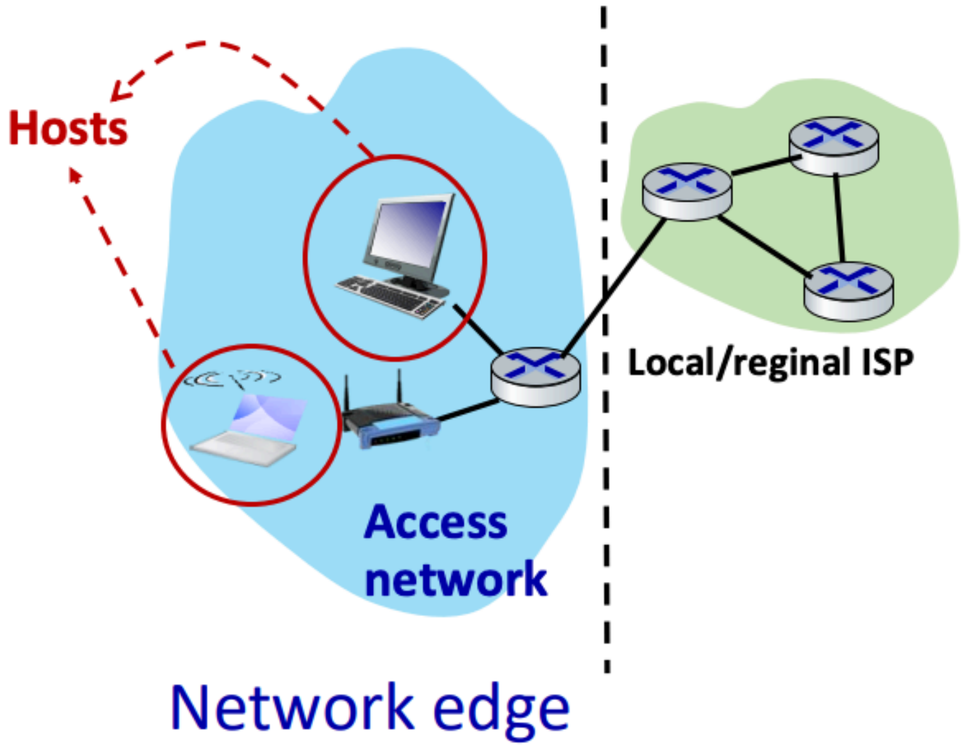
The Network Edge

Hosts are computing devices connected to the network (e.g., phones, laptops, etc.)

- They run applications that communicate with other hosts

Access networks connect hosts to the Internet

- Can be wired or wireless
- They contain an edge router, which is the entry point to the Internet



Wireless Access Network

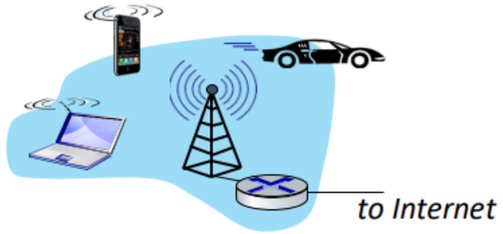
Shared wireless access network connects hosts to router via a base station (access point)

Wireless local area networks
(WLANs)

- typically within or around building (~30 m)

Wide-area cellular access networks

- provided by mobile, cellular network operator (10's km)
- 4G/5G cellular networks



Data Center Access Network

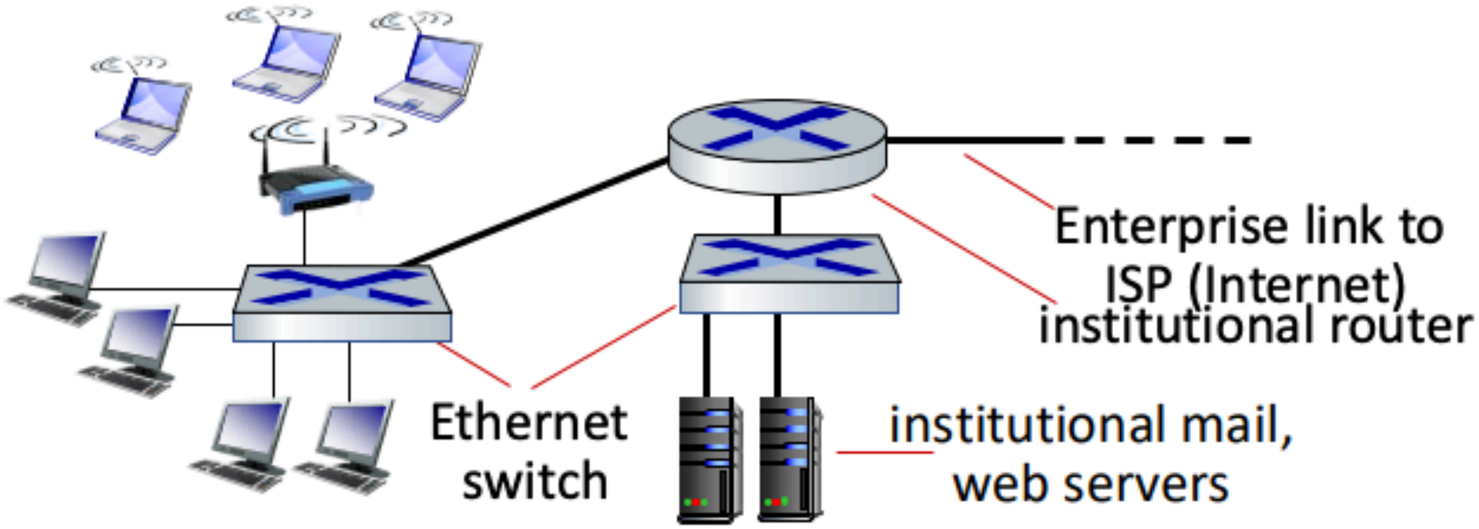
Connect hundreds to thousands of servers together, and to the Internet

Courtesy: Massachusetts Green High Performance Computing Center (mghpcc.org)

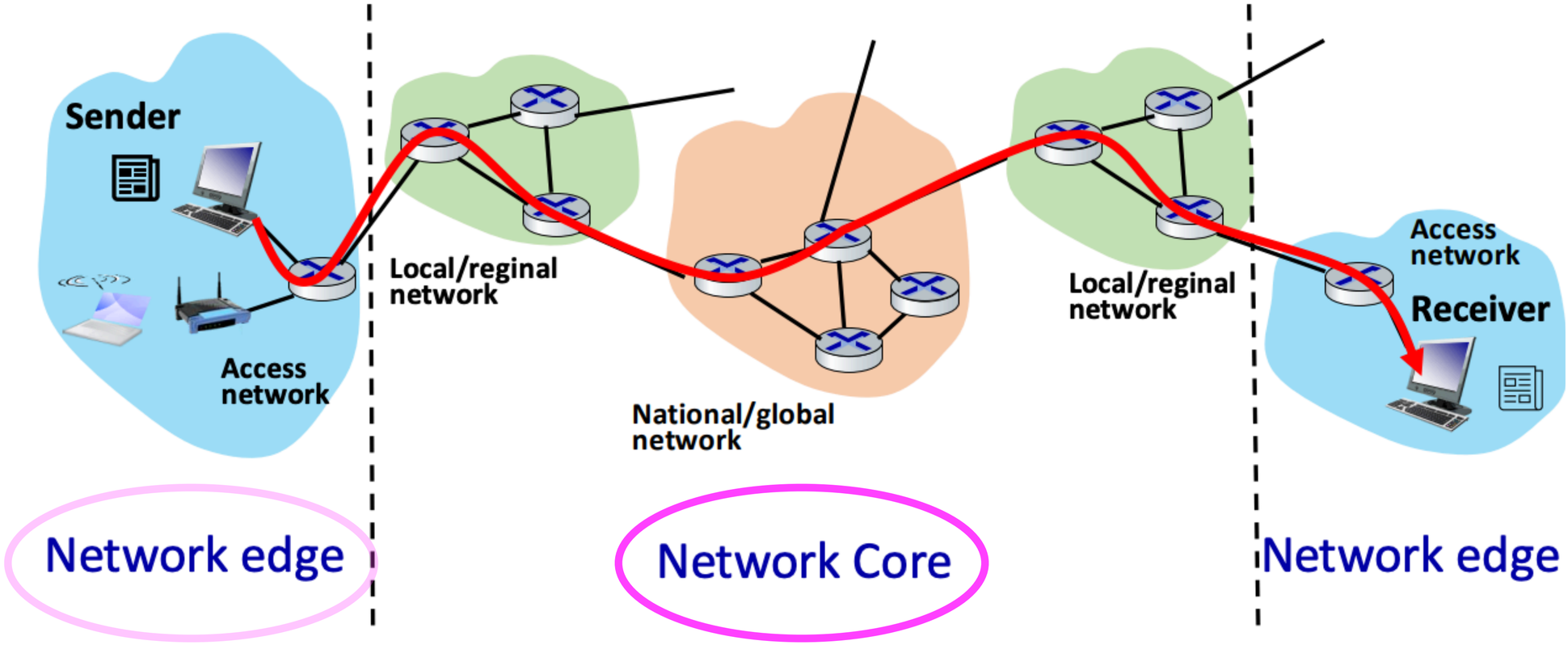
Enterprise Access Network

Operated by companies, universities, etc.

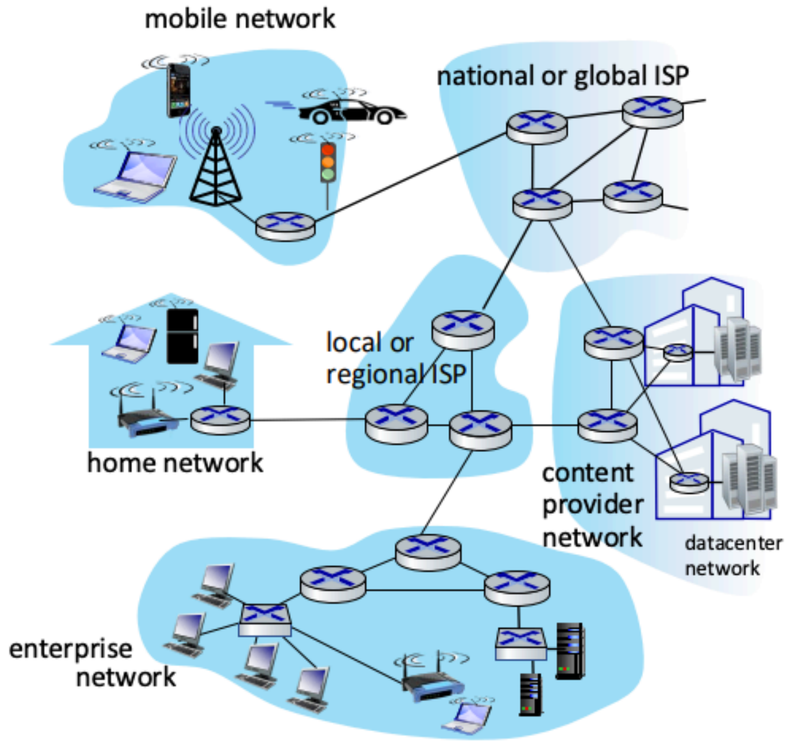
Mix of wired and wireless technologies



Types of Networks



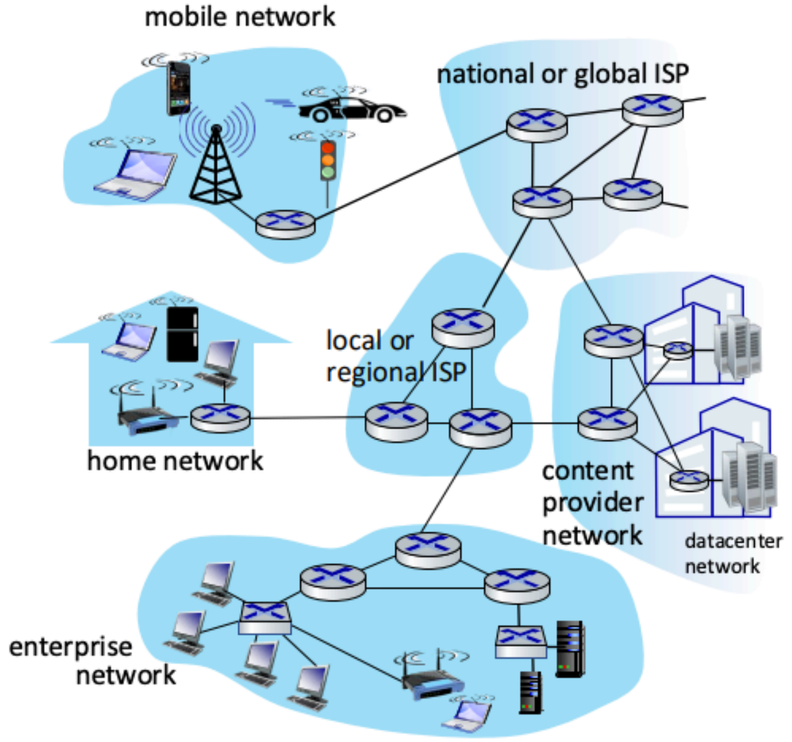
The Network Core



The Network Core

Responsible for connecting access networks together

- Otherwise hosts from different access networks couldn't communicate!



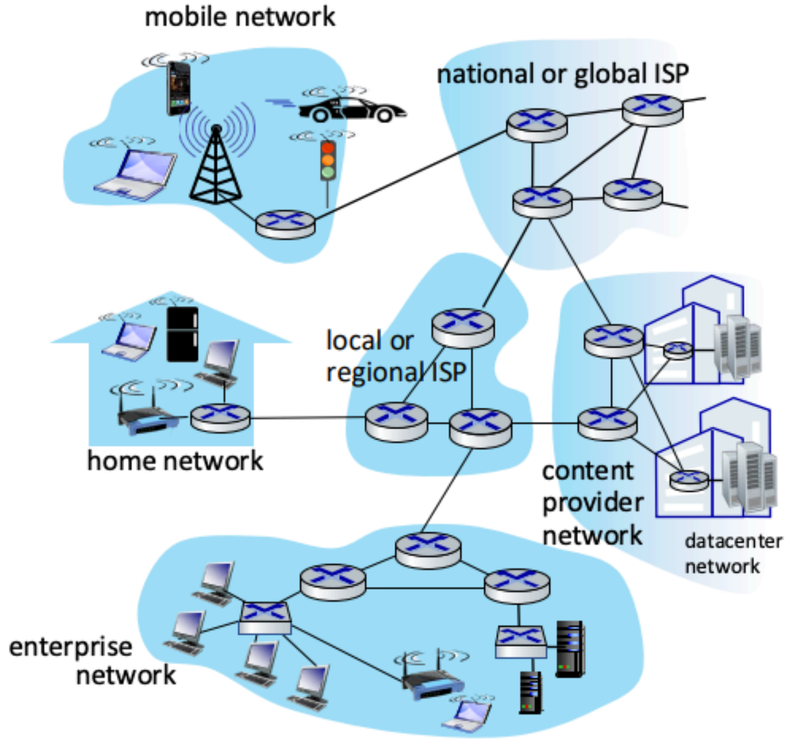
The Network Core

Responsible for connecting access networks together

- Otherwise hosts from different access networks couldn't communicate!

The network core is a complex network of networks

- Its evolution is driven by economics and national policies



What are some examples of applications that use the Internet?

A. Select A

Network Application Examples

- Social networking
- Web
- Text messaging
- Email
- Multi-player network games
- Streaming video/music
- Video conferencing
- Internet searching
- Remote logins
- …

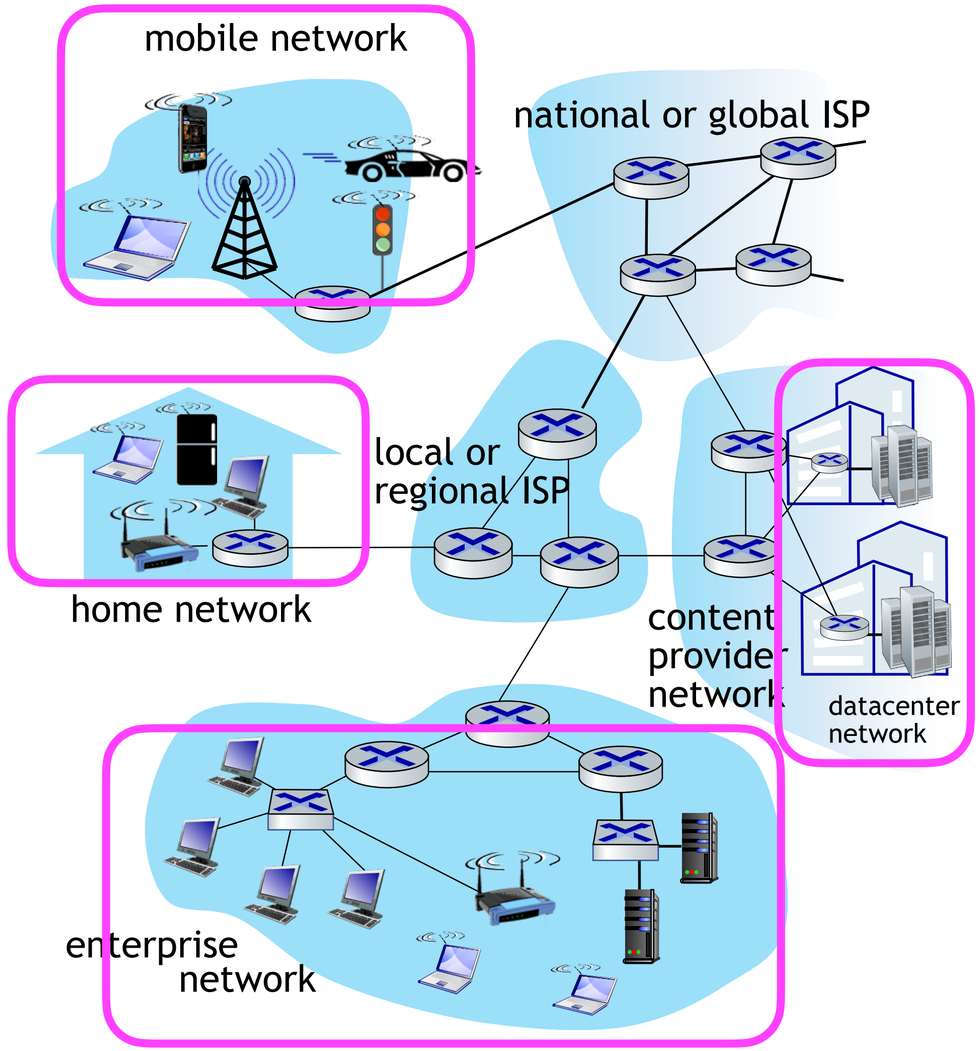
Writing a network app

Our programs should:

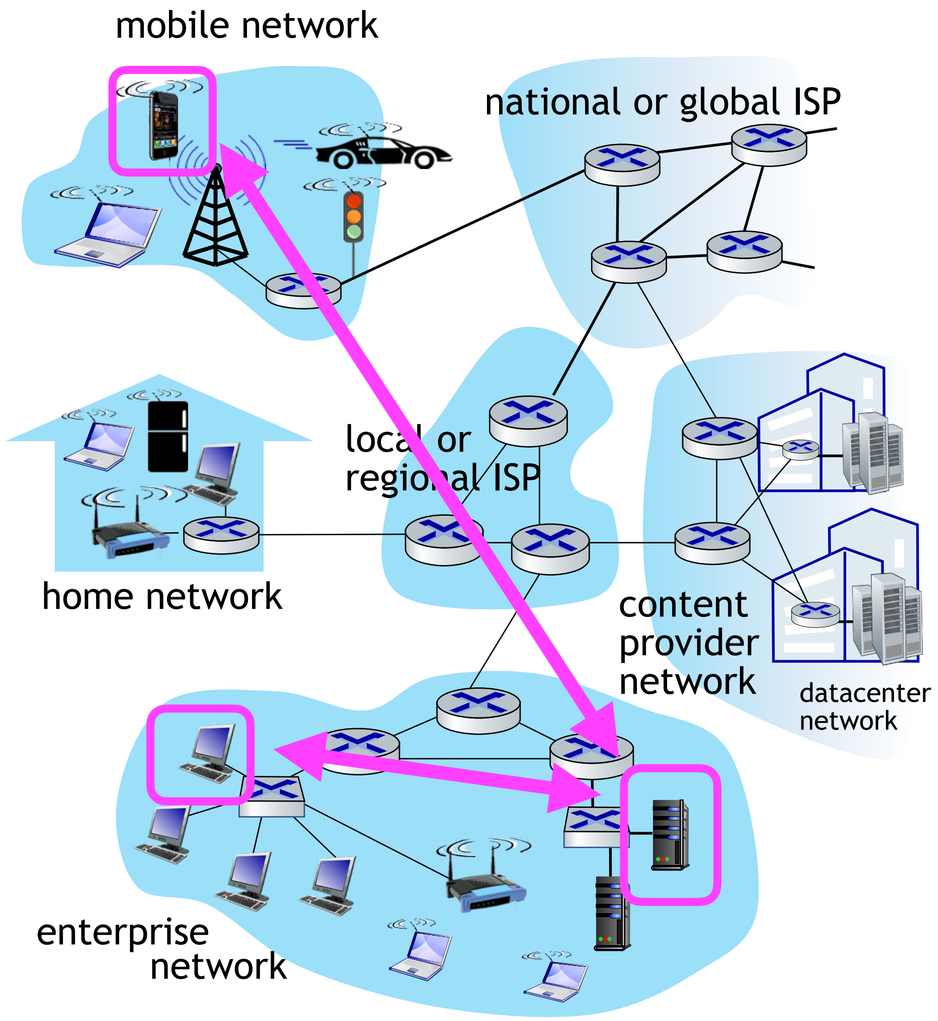
- Run on (different) hosts
- Communicate over a network

We do not need to write programs for the network core

- Network core devices do not run user applications
- Applications on host allow for more rapid development and deployment



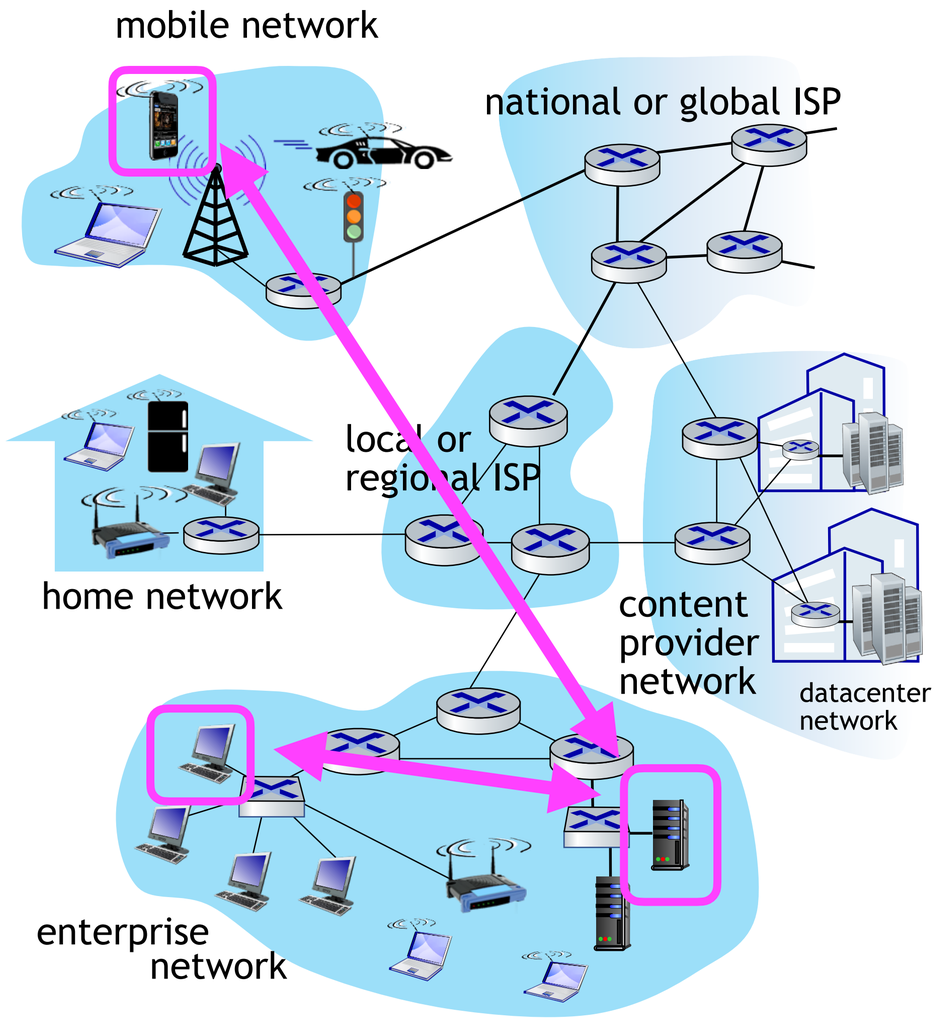
Client-Server Paradigm



Client-Server Paradigm

Server:

- Always-on host
- Permanent network address
- Often in data centers



Client-Server Paradigm

Server:

- Always-on host
- Permanent network address
- Often in data centers

Client:

- Contact/communicate with server
- May be intermittently connected
- May have dynamic address
- Do ***not*** communicate directly with each other