

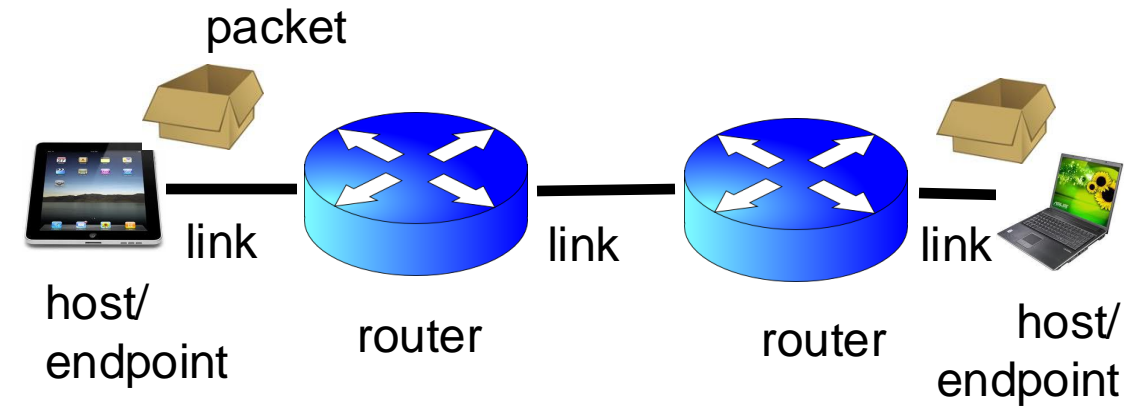
# Switching, Layering & Measurement

Lecture 2

<http://www.cs.rutgers.edu/~sn624/352-F24>

Srinivas Narayana

# Review of definitions



- **Endpoint or Host:** Machine running user application
- **Packet:** a unit of data transmission (ex: 1500 bytes)
- **Link:** a physical communication channel between two or more machines
- **Router:** A machine that processes packets moving them from one link to another towards a destination
- **Network:** Collection of interconnected machines
- **Address:** a unique name given to a machine (more later)

# Today's lecture

- Dive a bit deeper into how Internet communication works
  - Links: how does communication work physically?
  - Routers: how do they move data between links?
  - Endpoints: how is networking organized at endpoints?
- Understand how to measure the Internet

How do machines talk?

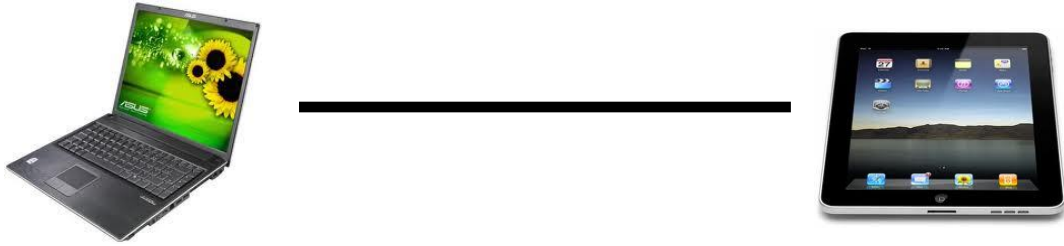
# How do machines communicate?

- With 1s and 0s
  - Digital computers deal with 1s and 0s
- How do we transmit 1s and 0s in a network?



Encoding and  
Decoding problem

# Physical transmission on a single link

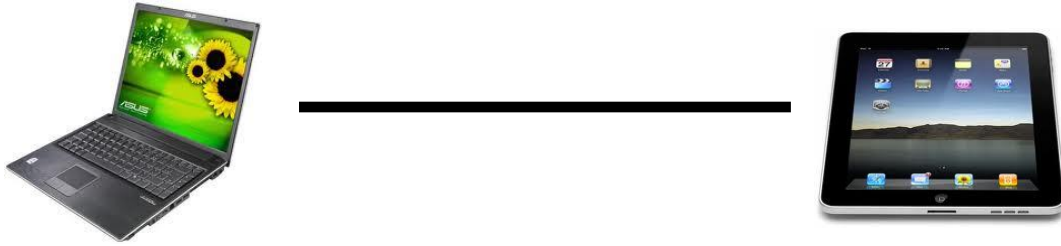


Physical signaling (light, AC voltages, etc.) are often **analog**

Convert bits to signals through **modulation** of the physical characteristics of signals: **encoding**

Convert signals back to digital by **decoding** physical signals

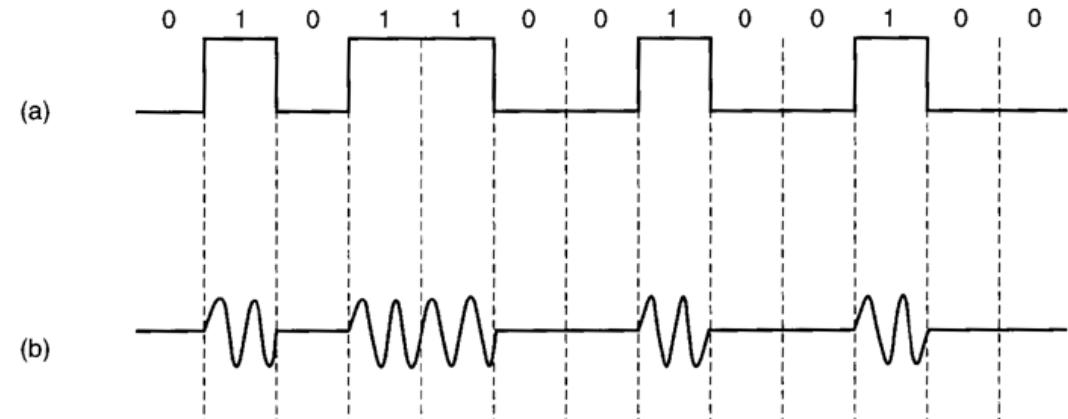
# Physical transmission on a single link



Physical signaling (light, AC voltages, etc.) are often **analog**

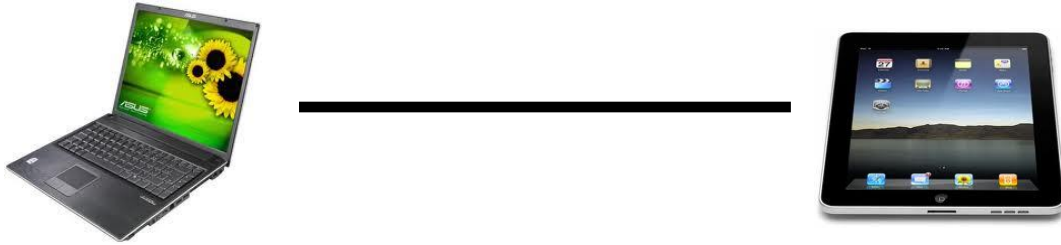
Convert bits to signals through **modulation** of the physical characteristics of signals: **encoding**

Convert signals back to digital by **decoding** physical signals



**Fig. 2-18.** (a) A binary signal. (b) Amplitude modulation. (c) Frequency modulation. (d) Phase modulation.

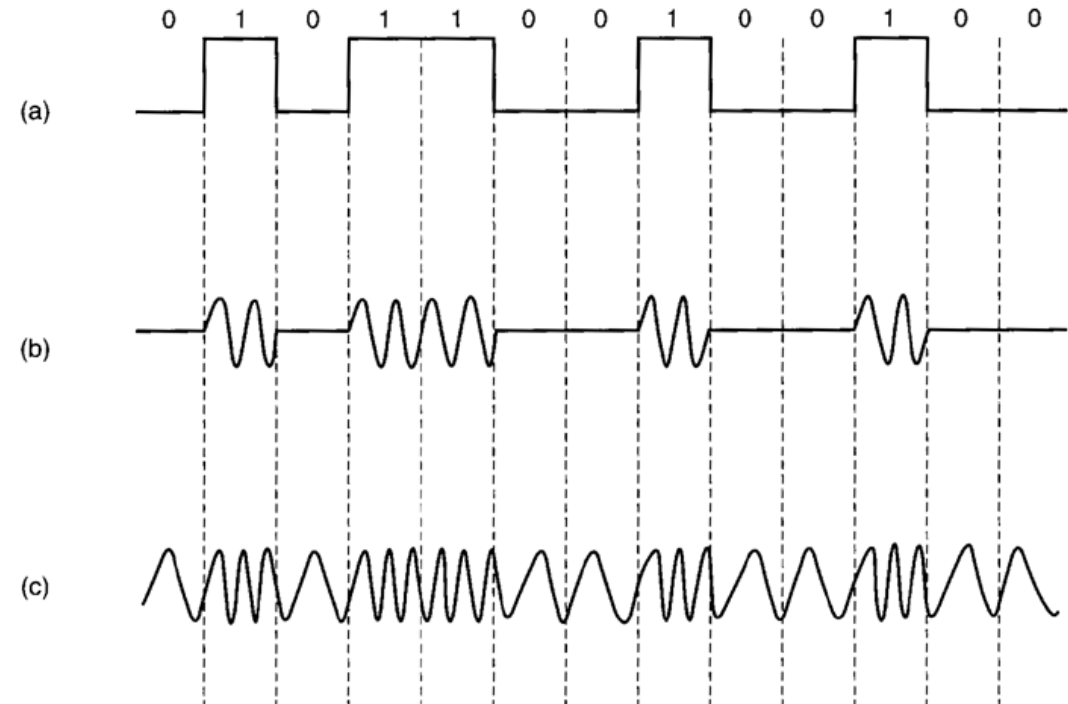
# Physical transmission on a single link



Physical signaling (light, AC voltages, etc.) are often **analog**

Convert bits to signals through **modulation** of the physical characteristics of signals: **encoding**

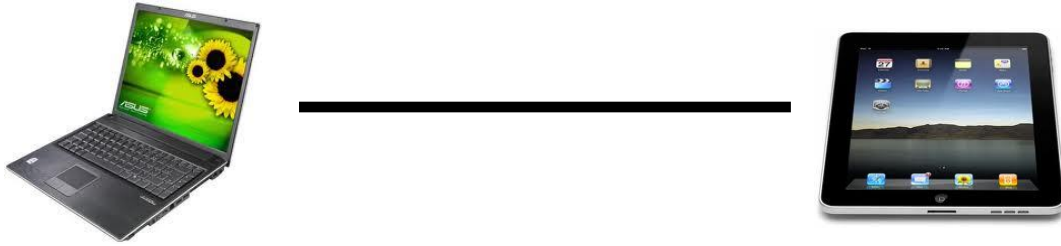
Convert signals back to digital by **decoding** physical signals



**Fig. 2-18.** (a) A binary signal. (b) Amplitude modulation. (c) Frequency modulation. (d) Phase modulation.



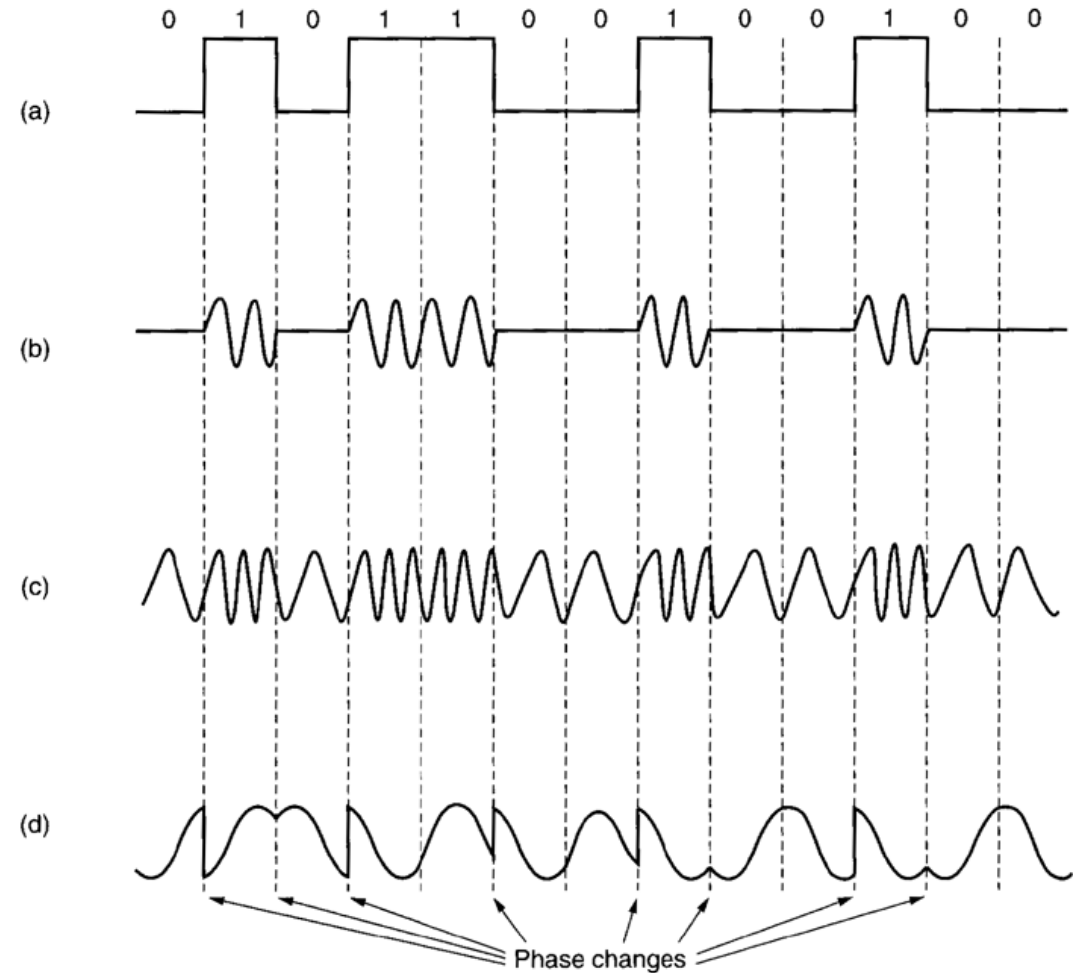
# Physical transmission on a single link



Physical signaling (light, AC voltages, etc.) are often **analog**

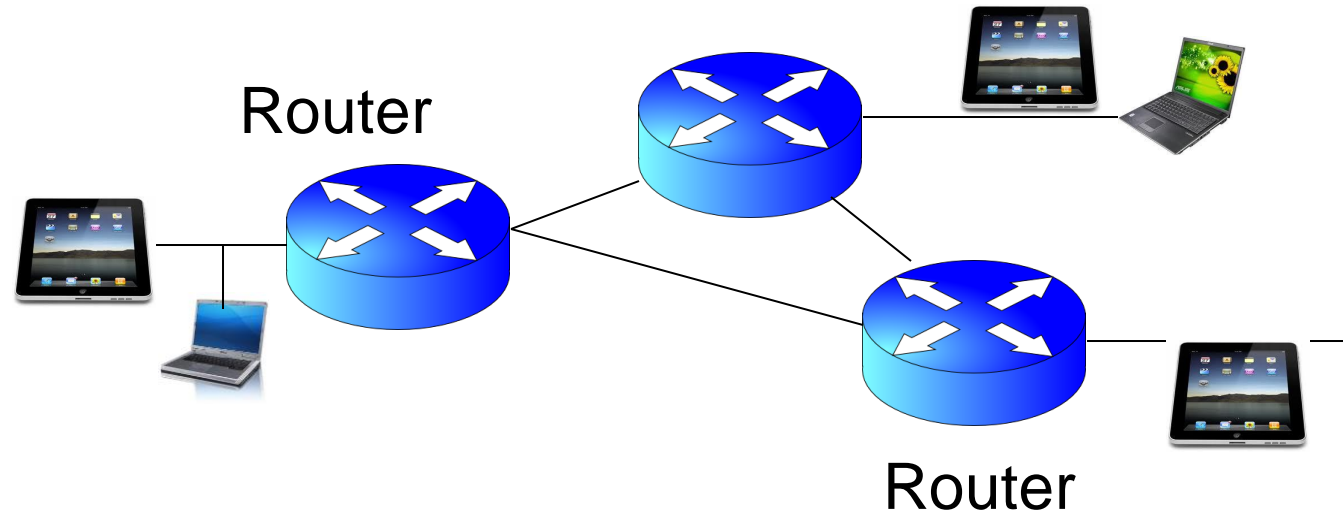
Convert bits to signals through **modulation** of the physical characteristics of signals: **encoding**

Convert signals back to digital by **decoding** physical signals



**Fig. 2-18.** (a) A binary signal. (b) Amplitude modulation. (c) Frequency modulation. (d) Phase modulation.

# Routers and Multi-link networks



- Routers need a way to move data across links
- We use the term **switching** to denote physically moving data from one link to another
- There are different possibilities to switch data across links

# Switching methods

Host applications transfer data containing many **messages**.  
A message could be a single web page, chat message

(1) Circuit Switching

(2) Message Switching

(3) Packet Switching

# Circuit switching

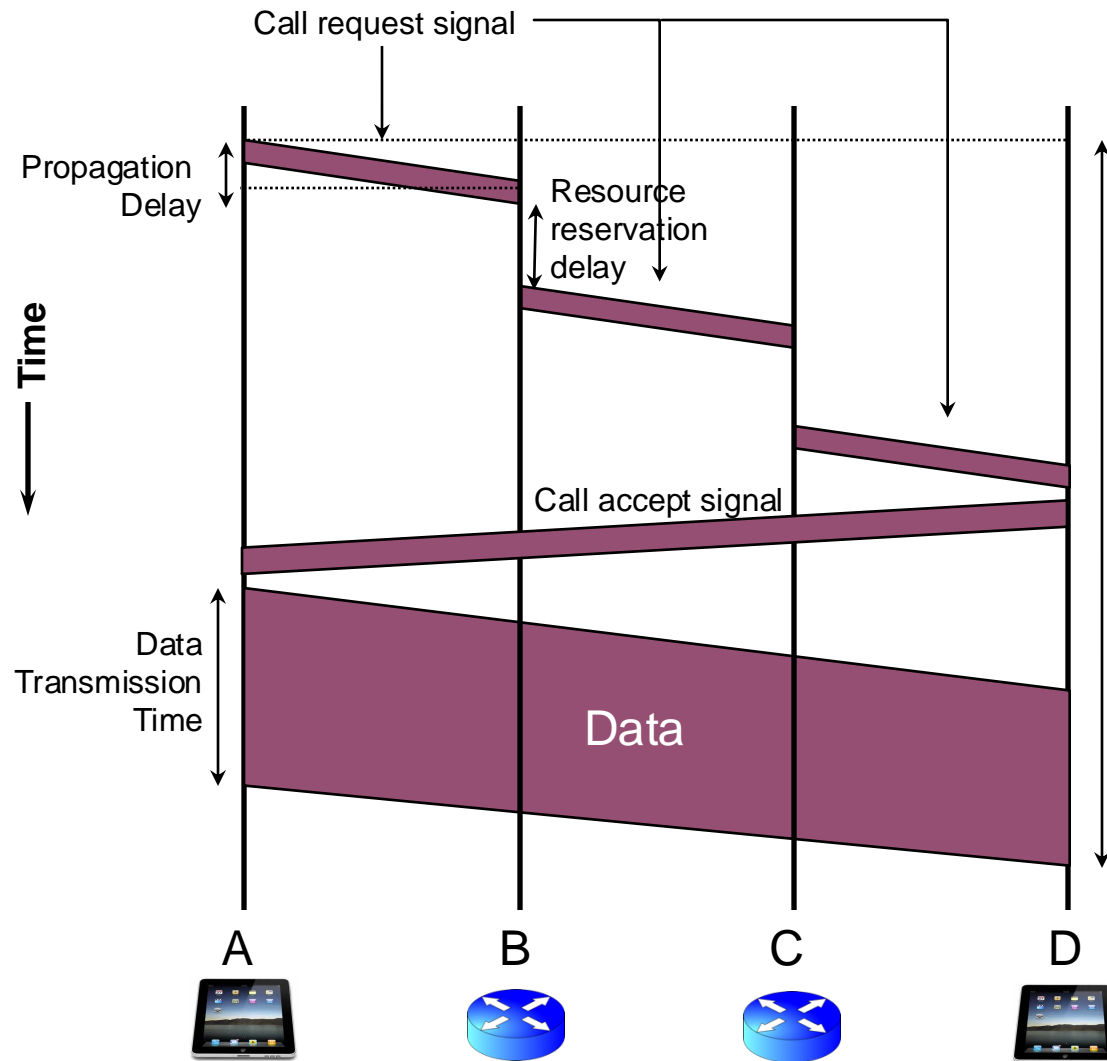
- Provides service by setting up the **full path of connected links** from the origin to the destination
- Example: Telephone network



Dedicated path with  
dedicated resources



# Circuit switching



1. **Setup:** Control message sets up a path from origin to destination
2. Accept signal informs source that data transmission may proceed
3. **Data transmission** begins
4. Entire path remains allocated to the transmission (whether used or not)
5. When transmission is complete, source releases the circuit

# Message switching



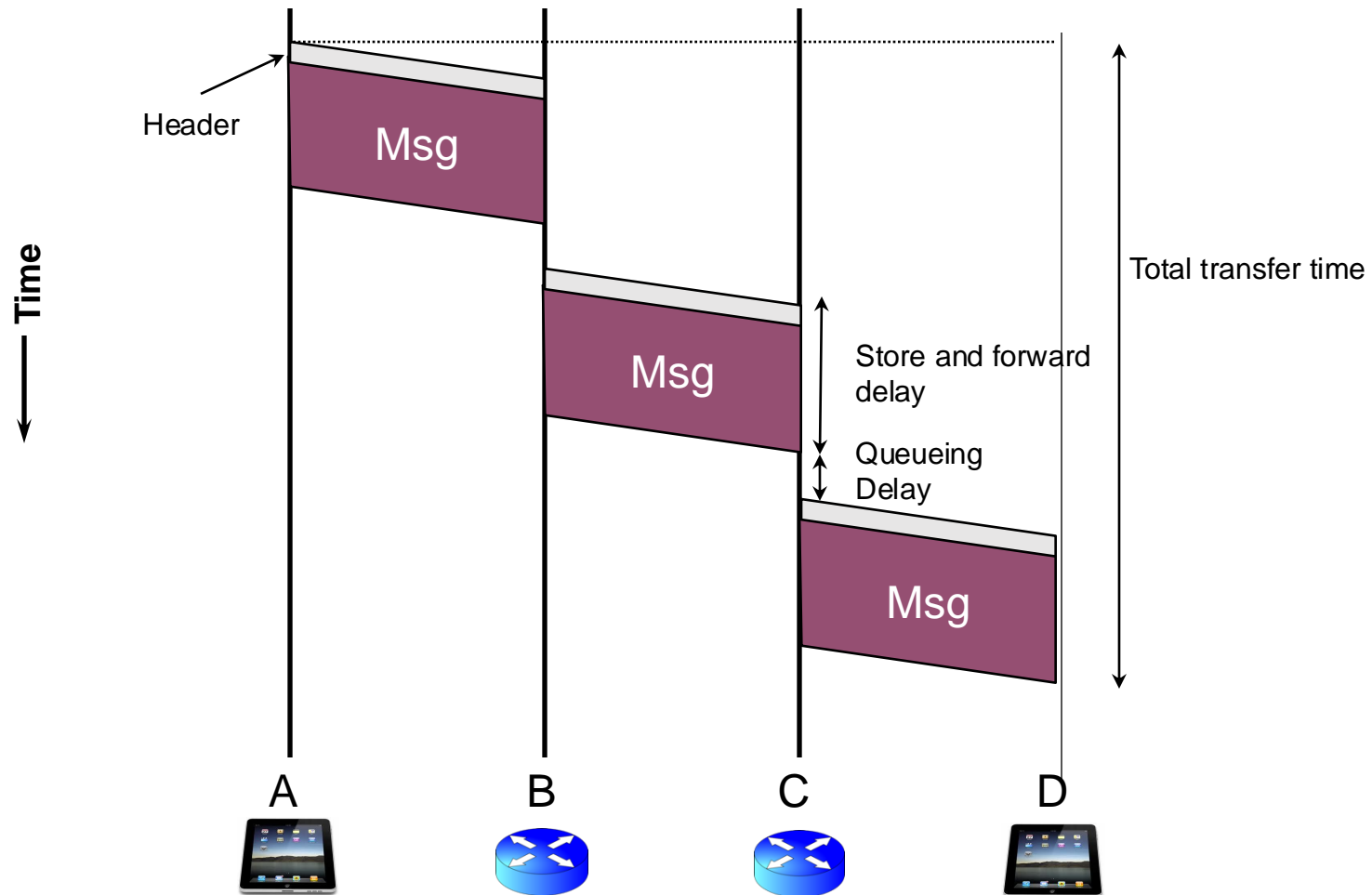
- Each message is addressed to a destination
- **Header:** metadata that denotes how to process a message
  - Typically includes a destination **address**
- The message “hops” from node to node through a network **while allocating only one link at a time**
- (Compare to circuit switching, where all links are reserved at the same time, regardless of use.)
- Analogy: Postal service



# Message switching

- When the entire message is received at a router, the next step and link in its journey are selected (**routing**)
- If this selected link is busy, the message waits in a **queue** until the link becomes free
- **Store and forward** switching
  - Router waits for all bits of a message to arrive on incoming link before sending the first bit of the message on the outgoing link
  - Alternative: **cut-through** switching sends bits as they arrive

# Message Switching

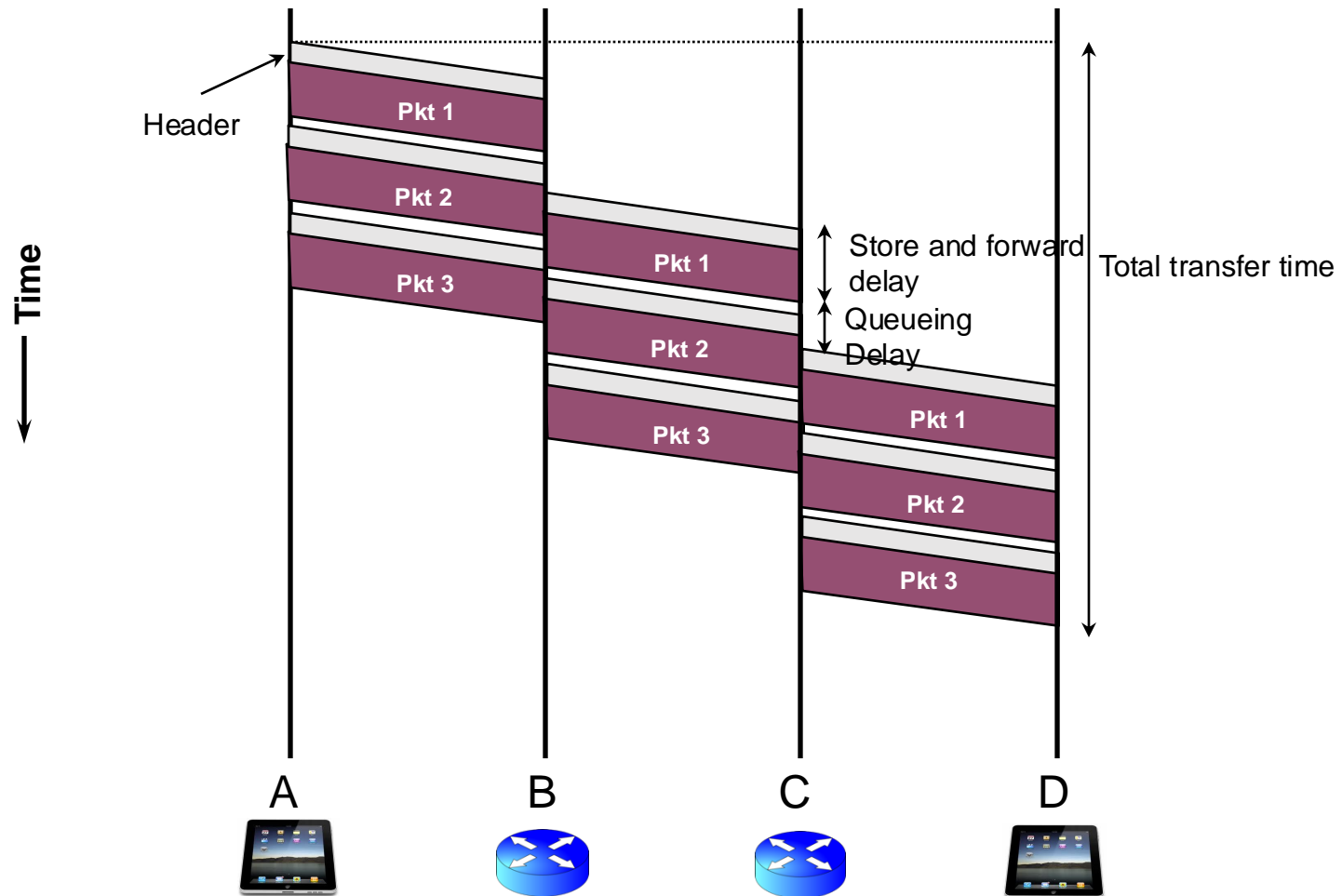




# Packet switching

- Messages are split into smaller pieces called **packets**
  - Packets have a maximum length
  - Packets are numbered and addressed
  - Packets are sent through the network one at a time
- **Pipelining**: different parts of a message concurrently transmitted over different links
  - Provides higher utilization of link resources

# Packet switching



The Internet uses store-and-forward packet switching.

# Comparisons across switching tech

- Circuit switching incurs an initial delay in setting up the resources along the path
  - Packet (and message) switching can start transmitting data right away
- Packet switching doesn't reserve resources for the conversation
  - While circuit switching does. Needs **admission control**
  - Packet switching makes resource reservation decisions per packet
- Fewer or no guarantees → easier to build
  - Telephone networks are more reliable but harder to build

# Comparisons across switching tech

## (1) **Total Delay to transfer a message**

Short Bursty Messages:

Packet < Circuit

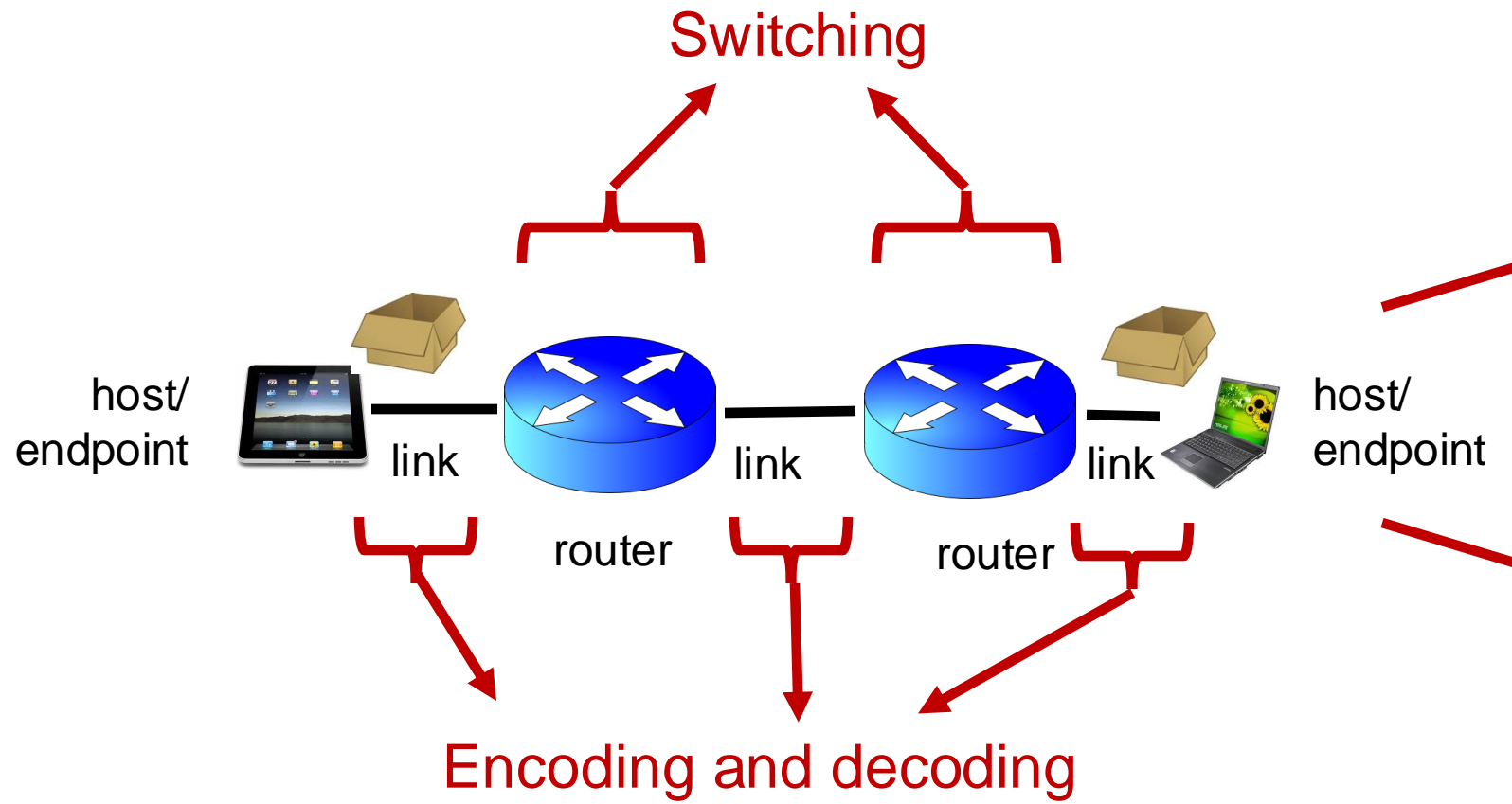
Long Continuous Messages:

Circuit < Packet

## (2) Header overhead (wastage, or what % of bits on the wire is metadata?)

If typical messages are larger than typical packets:

Packet > Message



Many layers...

# Layering and Protocols

# Solving the problems of communication

- Communication over the Internet is a complex problem
  - Application (e.g., web)
  - Guarantees (e.g., reliability)
  - Data movement across the Internet (routing)
  - Link concerns (encoding/decoding, medium access control)
- We solve complex problems by breaking them into simpler ones!
- **Layering** simplifies understanding, testing, and maintaining
- It is easy to improve or replace solutions at one layer without affecting others



# Software/hardware organization at hosts



Communication functions broken up and “stacked”

Each layer depends on the one below it.

Each layer supports the one above it.

The interfaces between layers are well-defined and standardized.

Internet software and hardware  
are arranged in **layers**.

Layering provides **modularity**

Each layer: well-defined **function**  
& **interfaces** to layers above & below it.

Functionality is implemented in **protocols**.



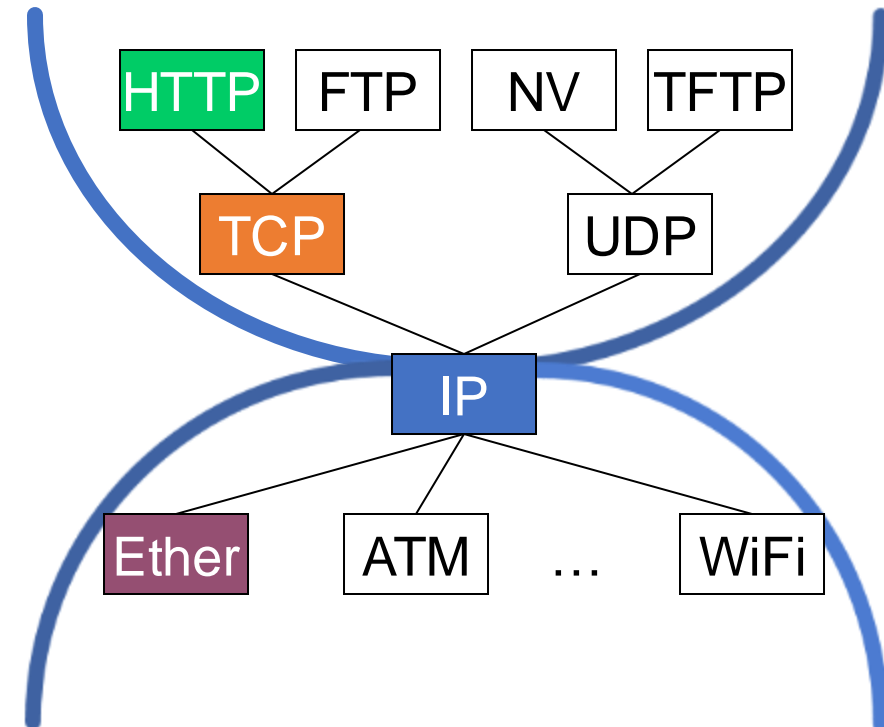
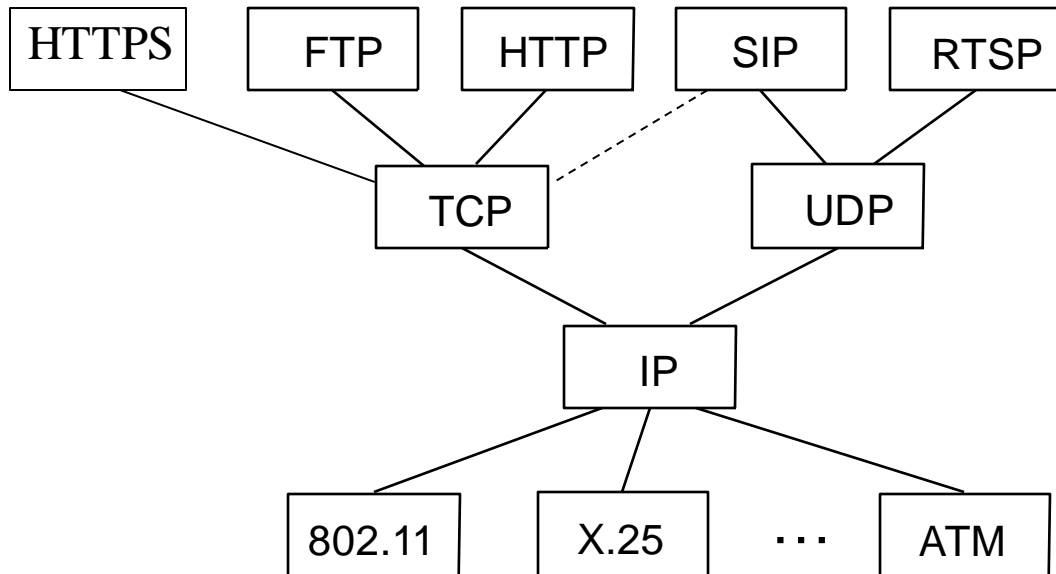
# Protocols: The “rules” of networking

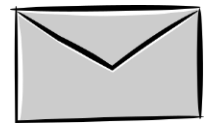
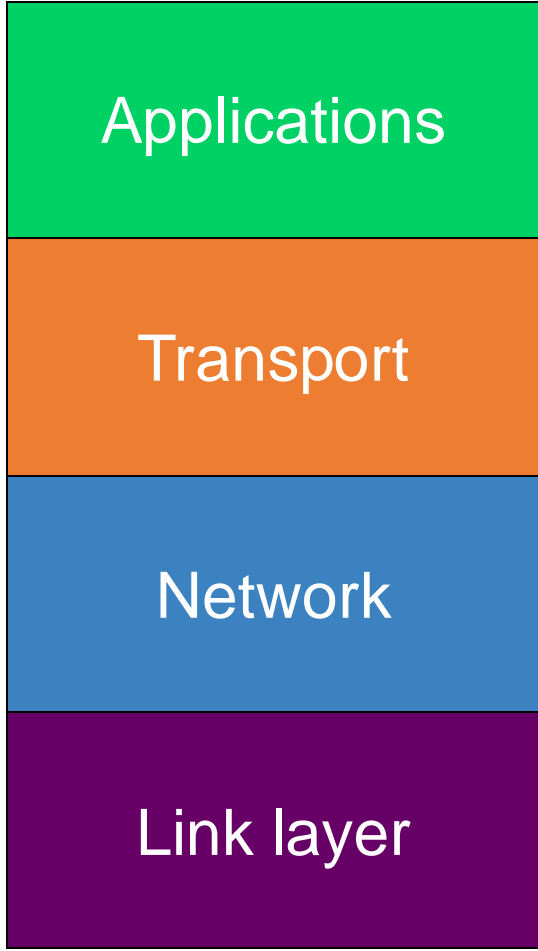
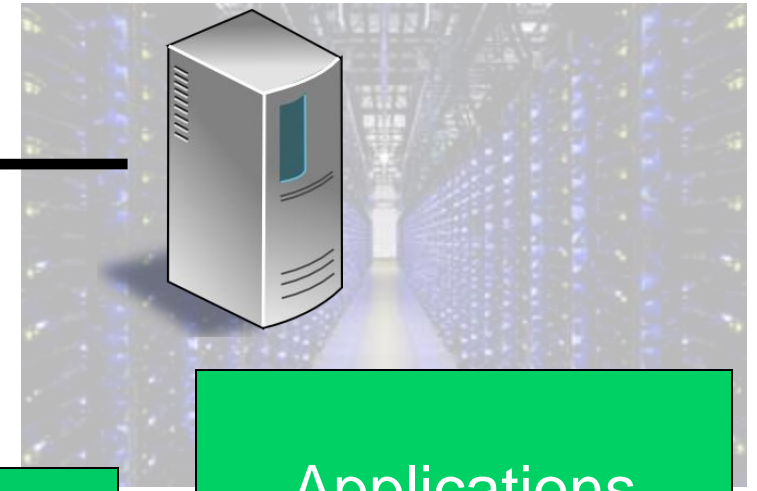
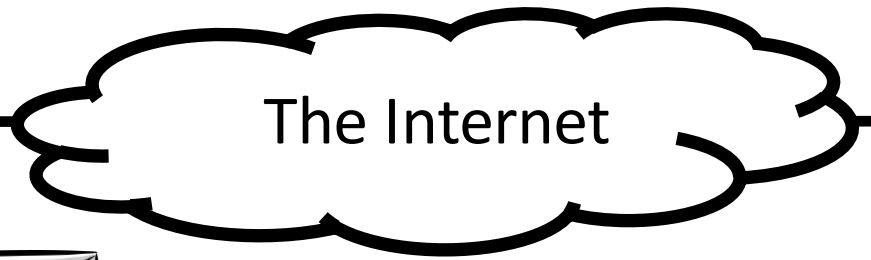
- Protocols consist of two things
- **Message format**
  - structure of messages exchanged with an endpoint
- **Actions**
  - operations upon receiving, or not receiving, messages
- Example of a Zoom conversation:
  - Message format: English words and sentences
  - Actions: when a word is heard, say “yes”; when nothing is heard for more than 3 seconds, say “can you hear me?”

# The protocols of the Internet

- Standardized by the Internet Engineering Task Force (IETF)
  - through documents called **RFCs** (“Request For Comments”)

- **Layering of protocols**

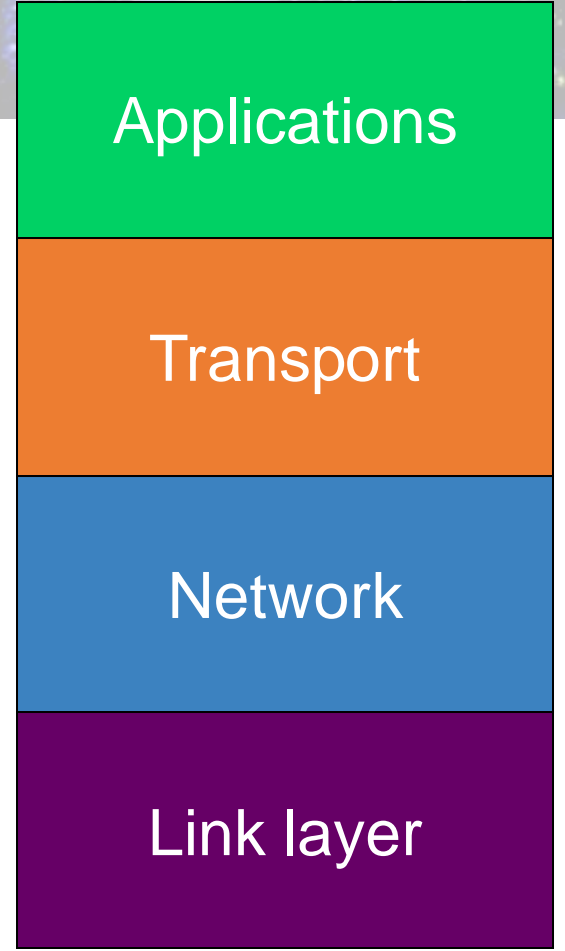
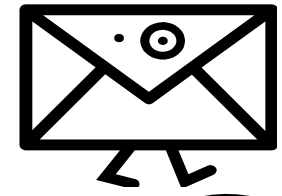
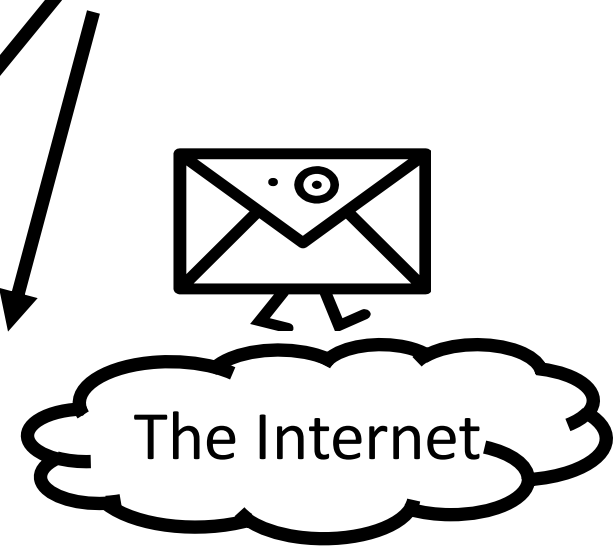


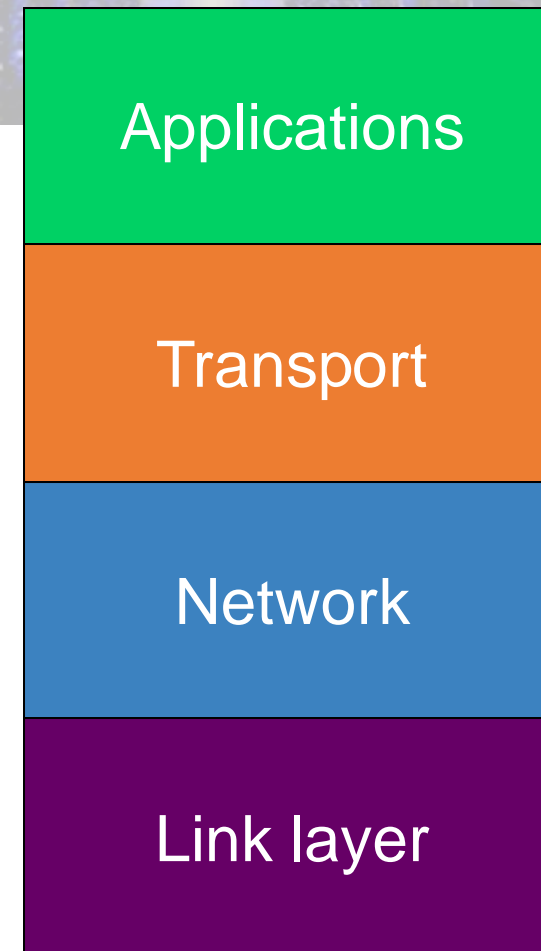
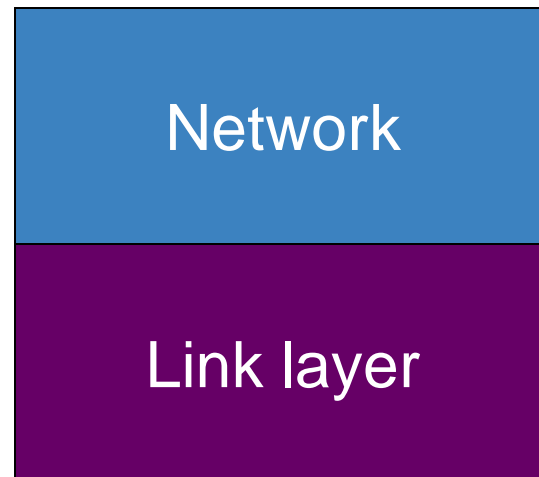
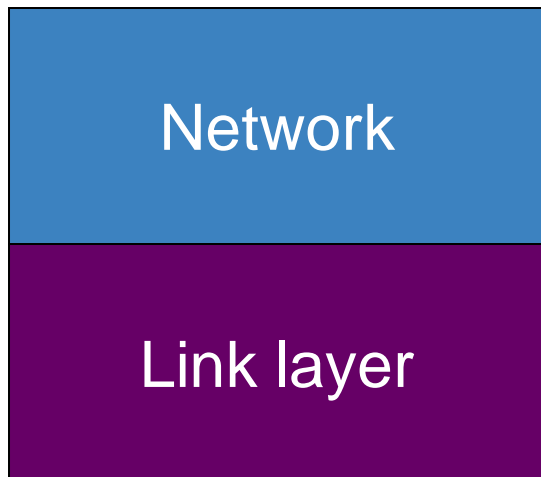
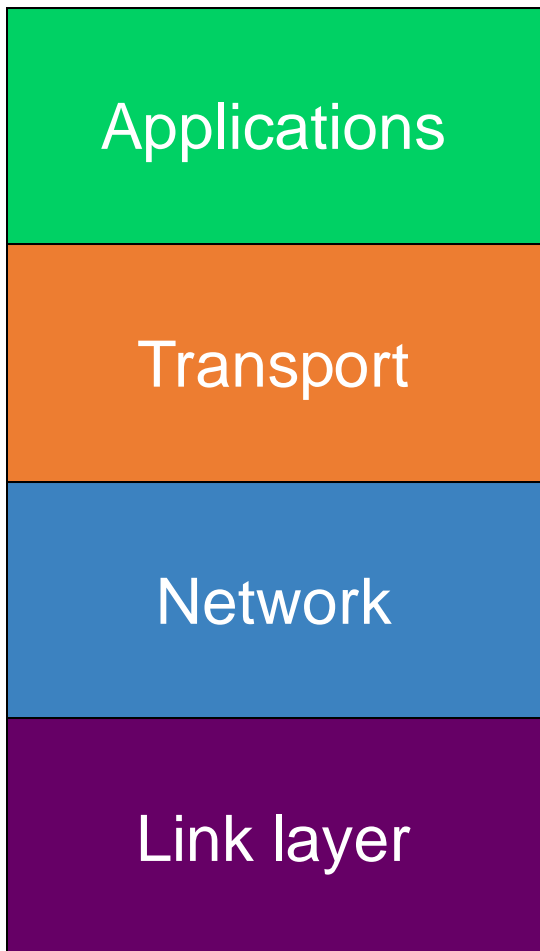
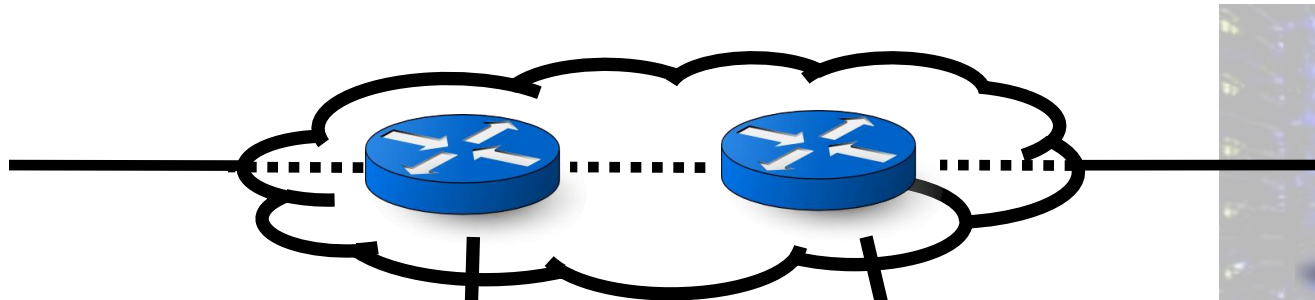


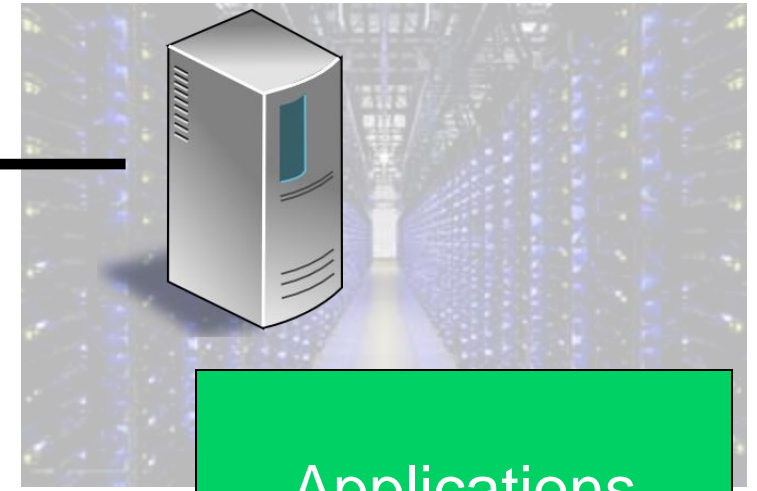
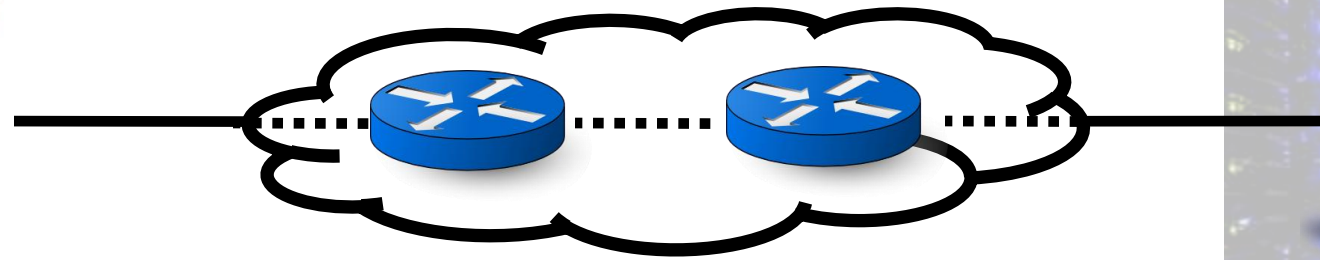
Packet starts as an app message



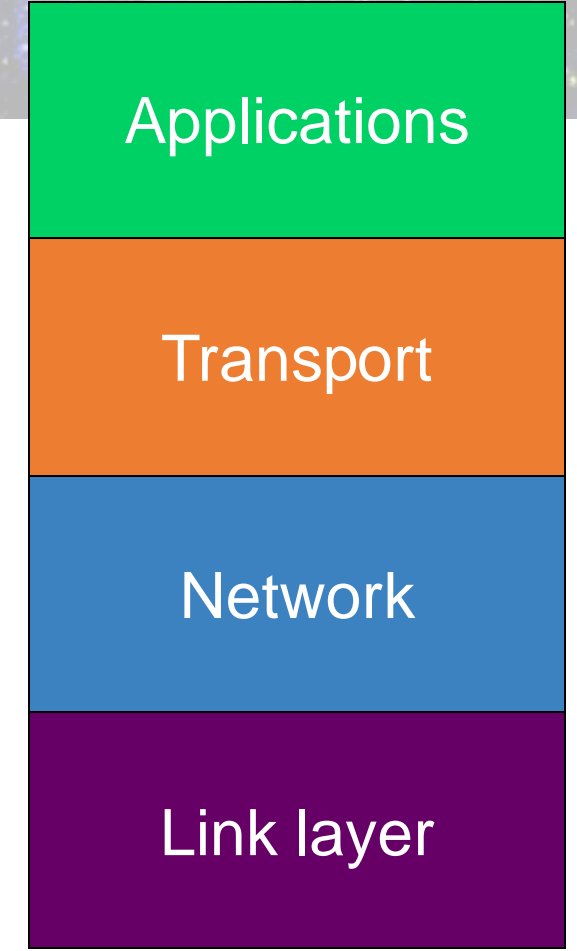
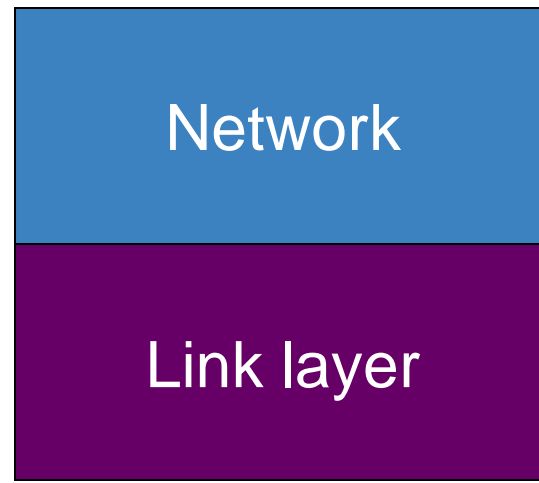
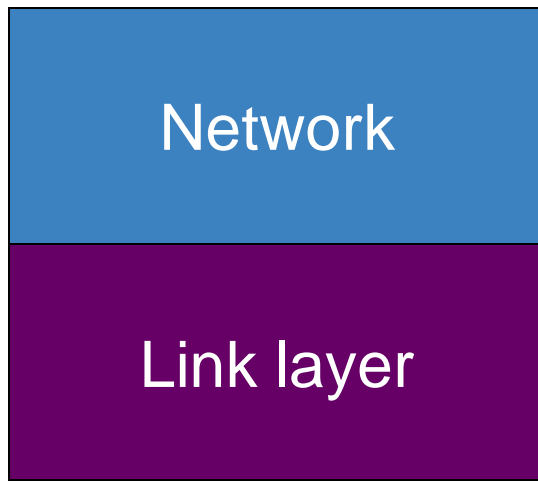
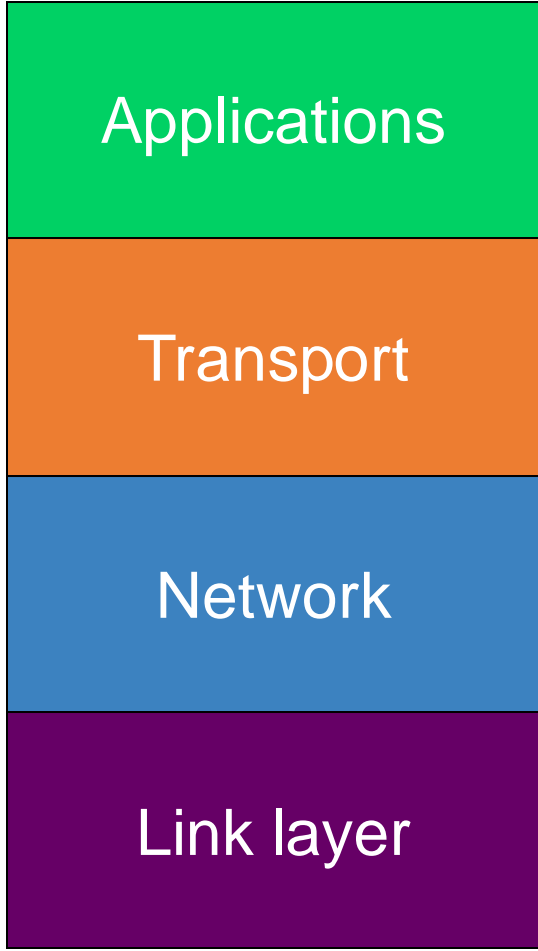
Packet takes on headers at each layer







Routers have network and link layers too!

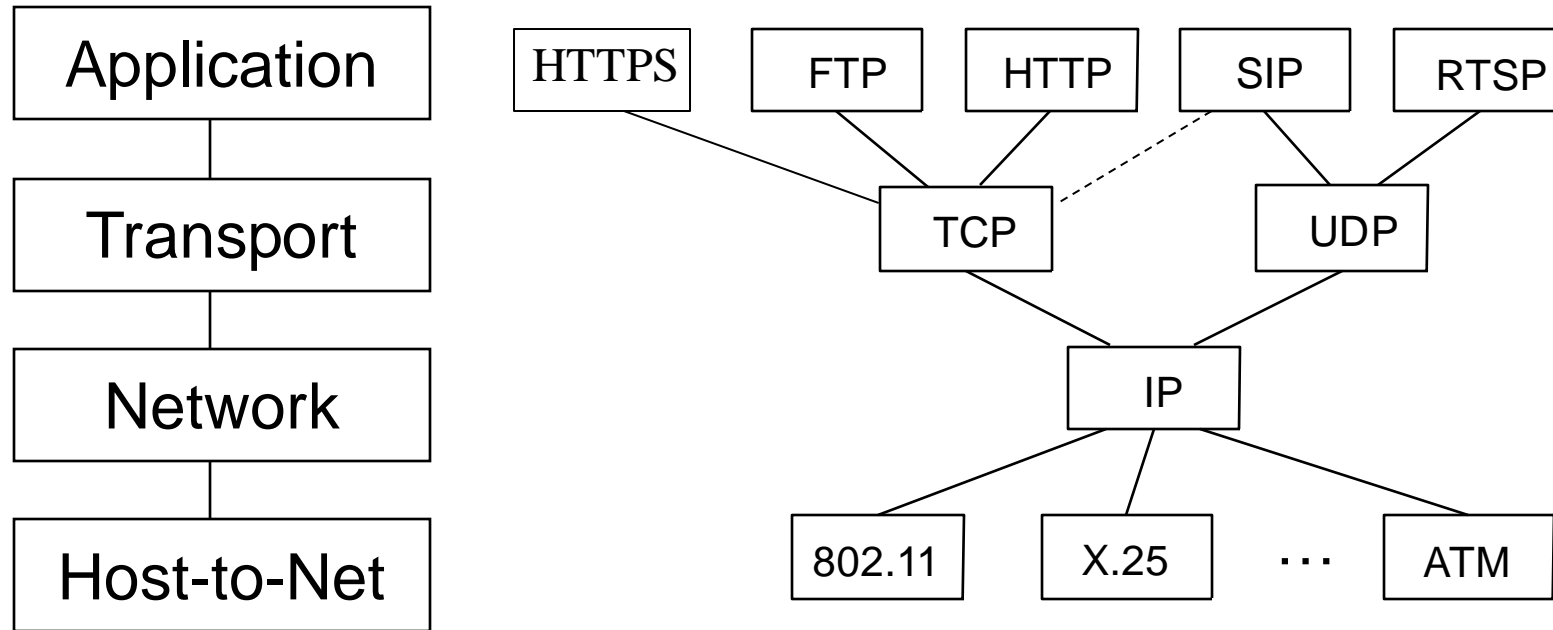


# Layering

- Communication over the Internet is a complex problem.
- Layering simplifies understanding, testing, maintaining
- Easy to improve or replace protocol at one layer without affecting others



# This course has layers



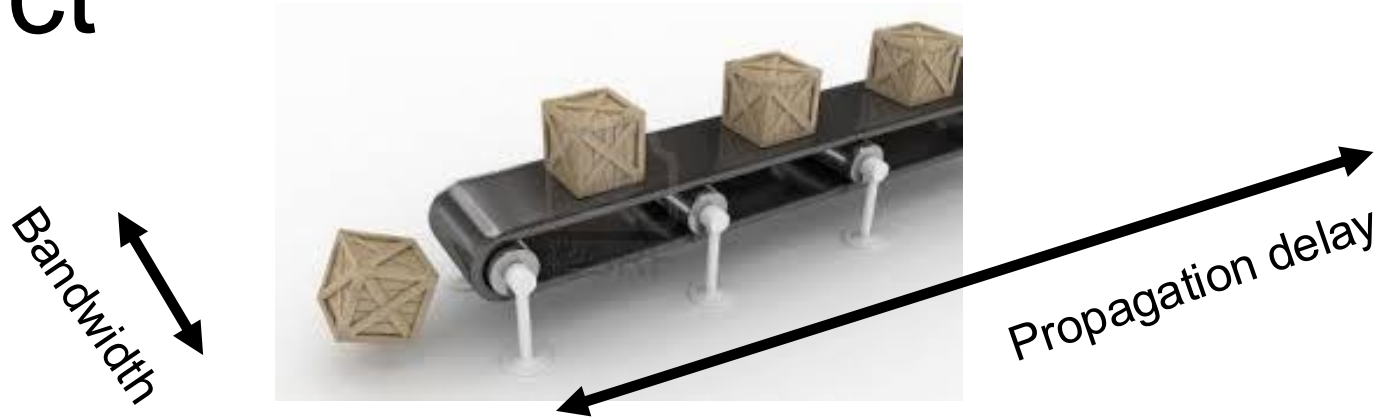
# Measuring the Internet

Speed, by any other name

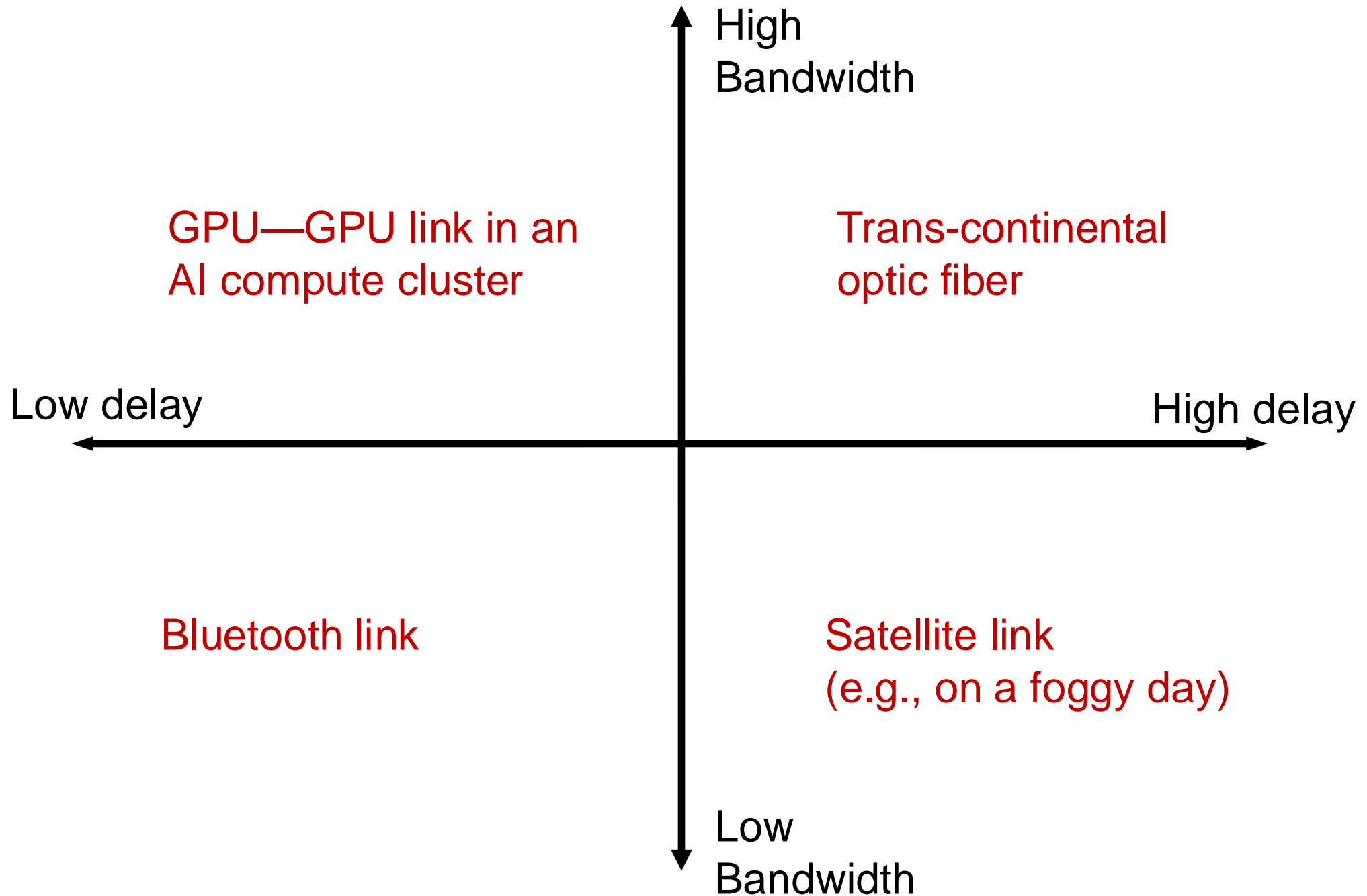
# What exactly do we mean by speed?

- A packet consists of many bits, including header and data
  - **Packet size**: length of the packet (bits or bytes) incl. header and data
- **Bandwidth**: For a single link, amount of data it can transmit per unit time (bits/second or Bytes/second or packets/second)
- **Total packet delay**: time from the first bit@sender to the last bit@receiver

# Bandwidth and delay are related but distinct



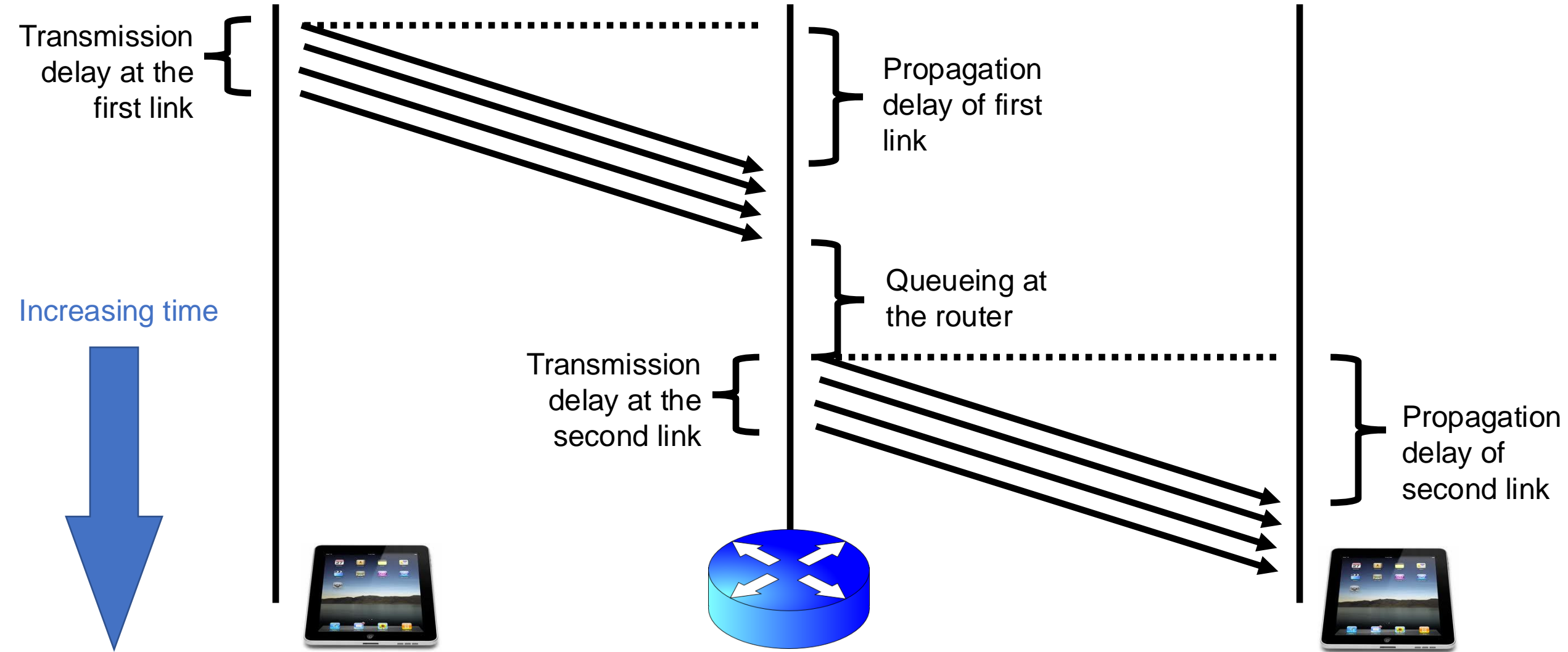
- Total packet delay = time for a box to travel the length of the belt
- Bandwidth = the number of boxes put on the belt per minute (“rate”)



# Total Packet Delay has a few pieces

- **Propagation delay:** Time needed to move one bit across (second)
  - Imposed by the communication medium; depends on the link “length”
- **Transmission delay:** Time from first bit@sender to last bit@sender
  - Determined by link bandwidth and packet size
  - Packet size / link bandwidth
- **Queueing delay:** Time that a packet waits for transmission
  - Determined by contention for the link
- **Total packet delay** = propagation delay + queueing delay + transmission delay for a single packet

# Visualizing the components of delay



# Bandwidth and delay demo

- Throughput (related to bandwidth)
  - `iperf -s #` at the destination
  - `iperf -c <destination> #` at the source,
  - e.g., `iperf -c localhost`
- (total) delay
  - `ping <destination>`
  - e.g., `ping google.com`
- (you can try it!)