Networks and Protocols

Introduction

Introduction

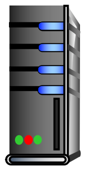
- How can machines communicate?
  - When they are directly connected

  - When they are not directly connected

- But, we talk about how many machines?

Introduction

- A simple Task : Send information from one computer to another
  - Endpoints called hosts
    - Could be computer, iPhone, laptop, etc.

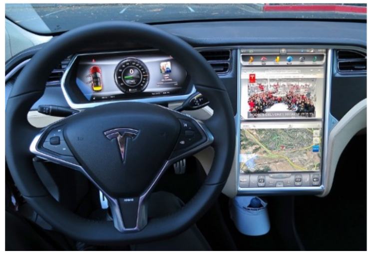


Desktop

Cellular phone

Gaming devices

Traffic lights

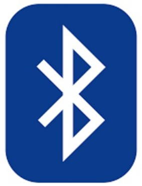


cars

Security Camera

Internet refrigerator

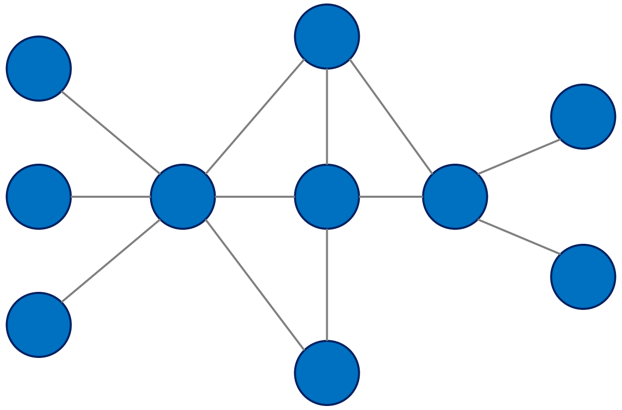
  - Communication links
    - We don’t care what the physical technology is: Ethernet, wireless, cellular, etc.



- Protocols: control sending, receiving of messages

Introduction

- Two machines ⇨ link
- (But, what if we talk about many machines?)
- Multiple machines ⇨ network
  - A system of “links” that interconnect “nodes” in order to move “information” between nodes



- Multiple networks ⇨ Inter-network ⇨ Internet!
- Because, you know Internet, we will focus primarily on the Internet

How The internet Works

The internet: Service view vs component view

- **Service view**

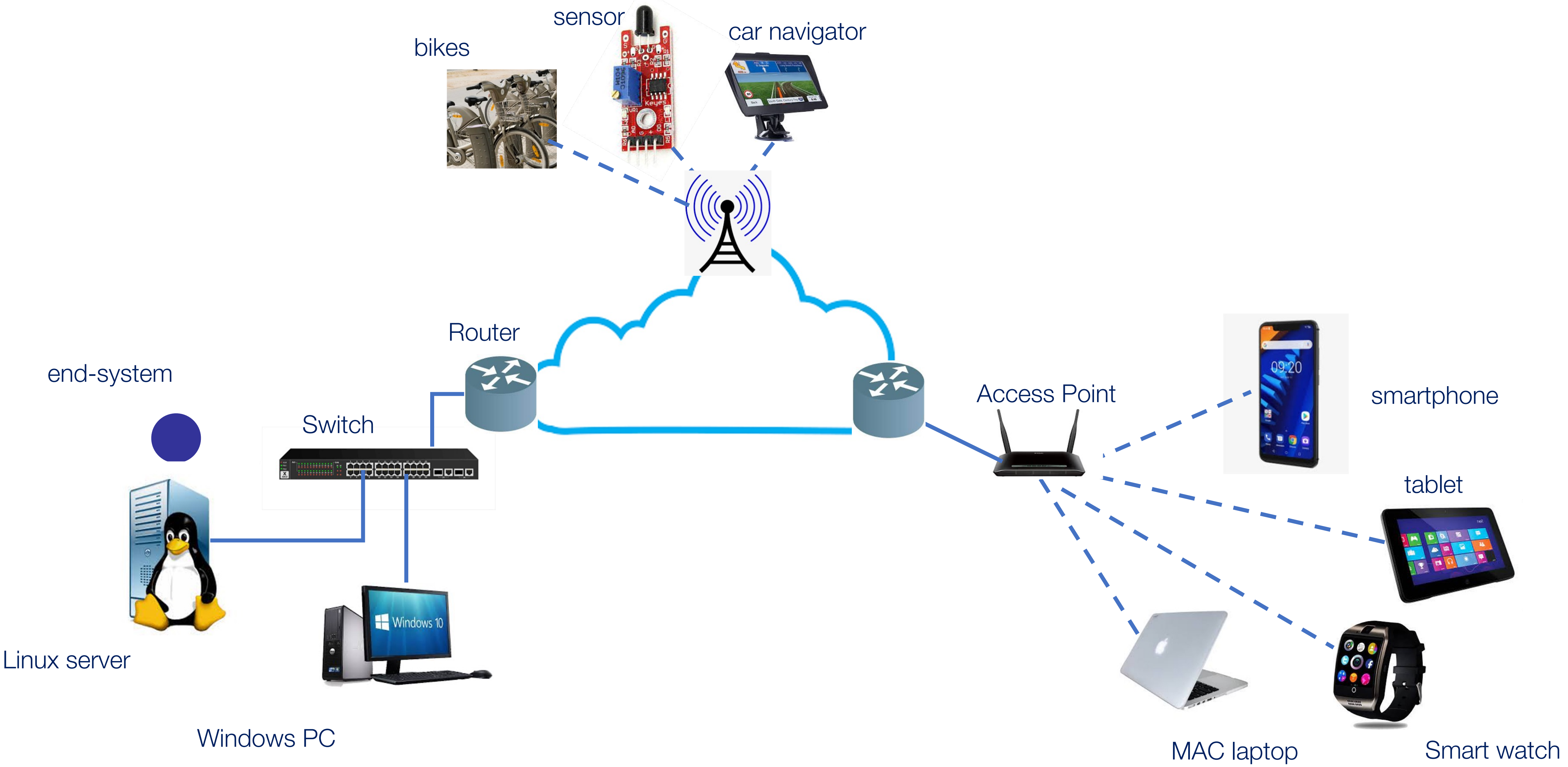
- Communication *infrastructure* enables distributed apps:
  - Web, VoIP, email, games, e-commerce, file sharing
- Communication services provided to apps:
  - reliable data delivery
  - QoS data delivery
  - “best effort” data delivery

- **Component view**

- Protocols control sending, receiving of messages
  - e.g., TCP, IP, HTTP, Ethernet
- Internet: “network of networks”
  - loosely hierarchical
- Internet standards
  - RFC: Request For Comments
  - IETF: Internet Eng. Task Force

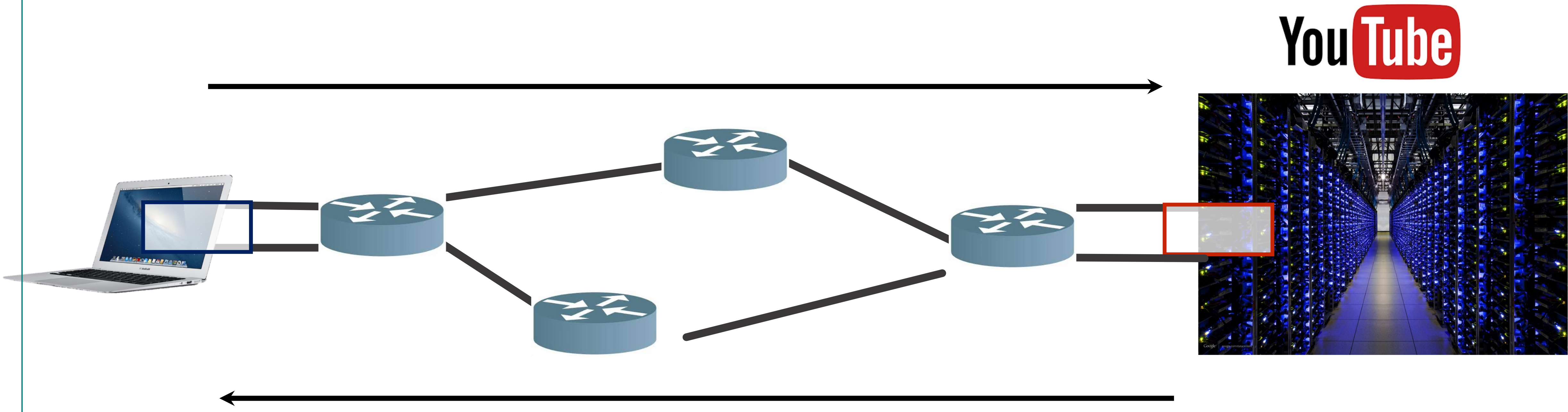
The internet : component view

- The Internet consists of many end-systems managed by many parties



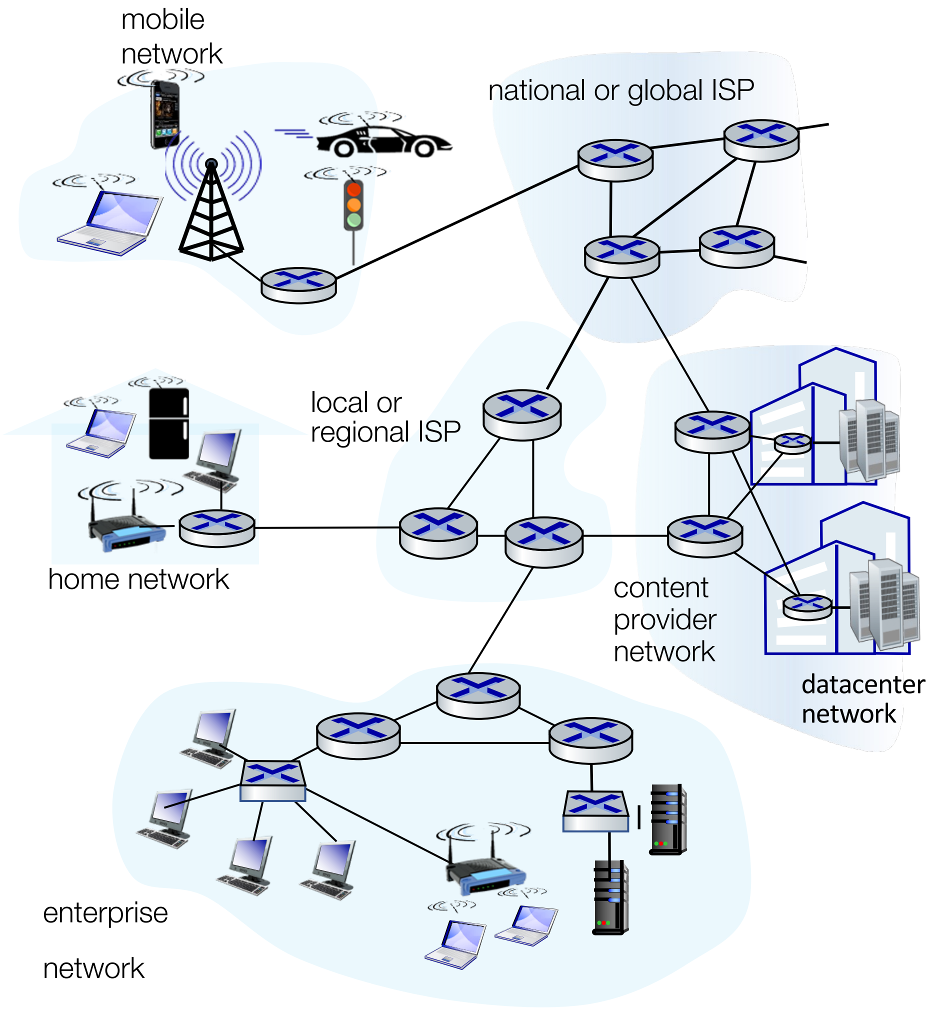
How The internet Works

- ROUGHLY, what happens when I click on a Web page e.g. on YouTube?



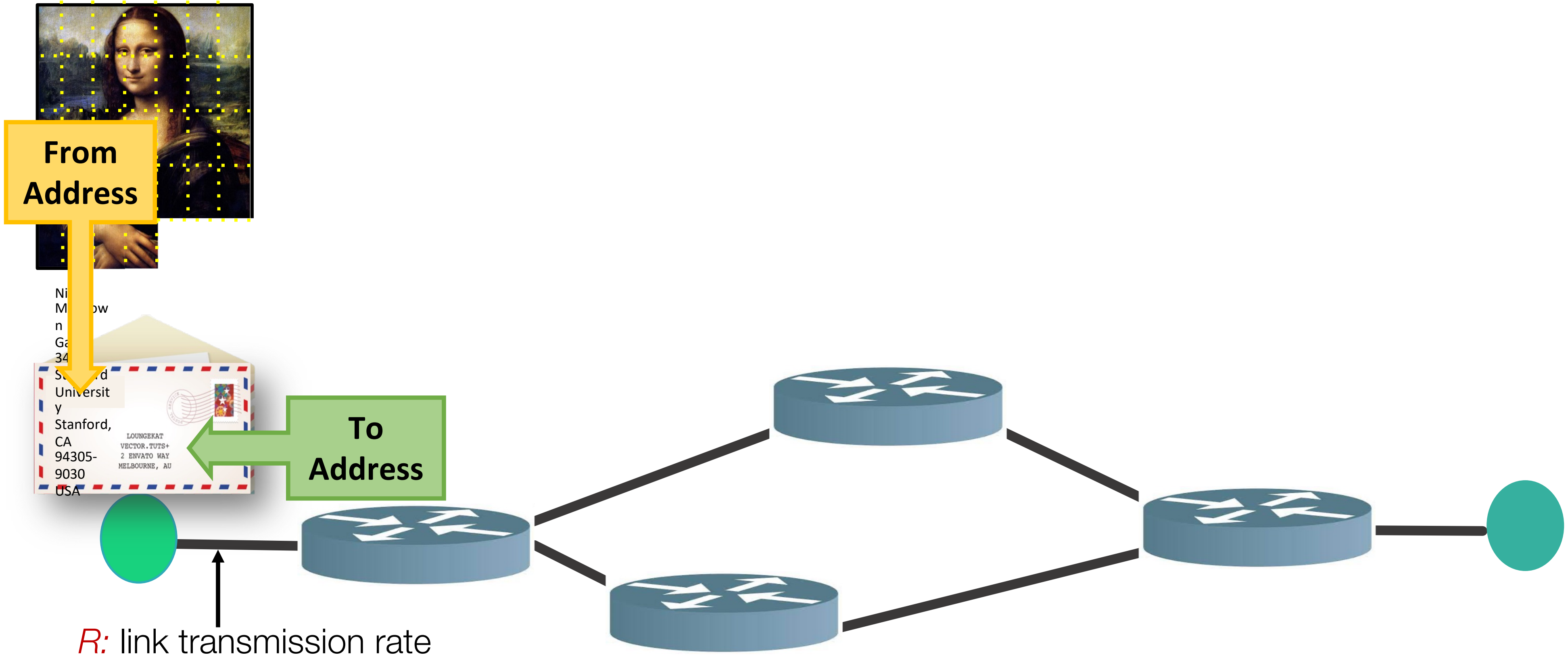
The internet: Terminology

- Network edge:
  - hosts: clients and servers
  - servers often in data centers
- Access networks, physical media:
  - wired, wireless communication links
- Network core:
  - interconnected routers
  - network of networks



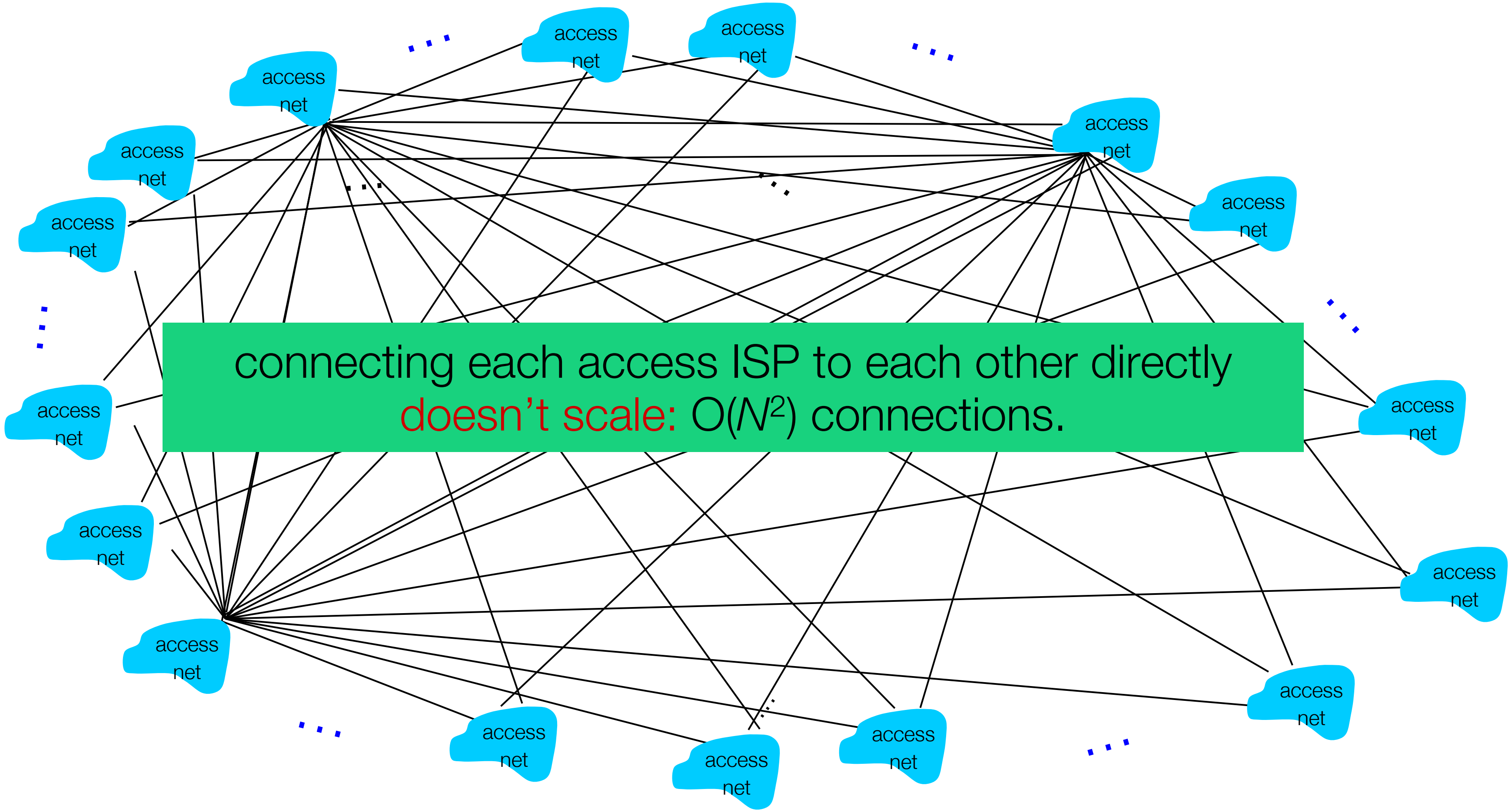
Host: sends packets of data

- host sending function:
  - takes application message
  - breaks into smaller chunks, known as packets, of length L bits
  - transmits packet into access network at transmission rate R
    - link transmission rate, aka link capacity, aka link bandwidth



Network core: connecting everyone

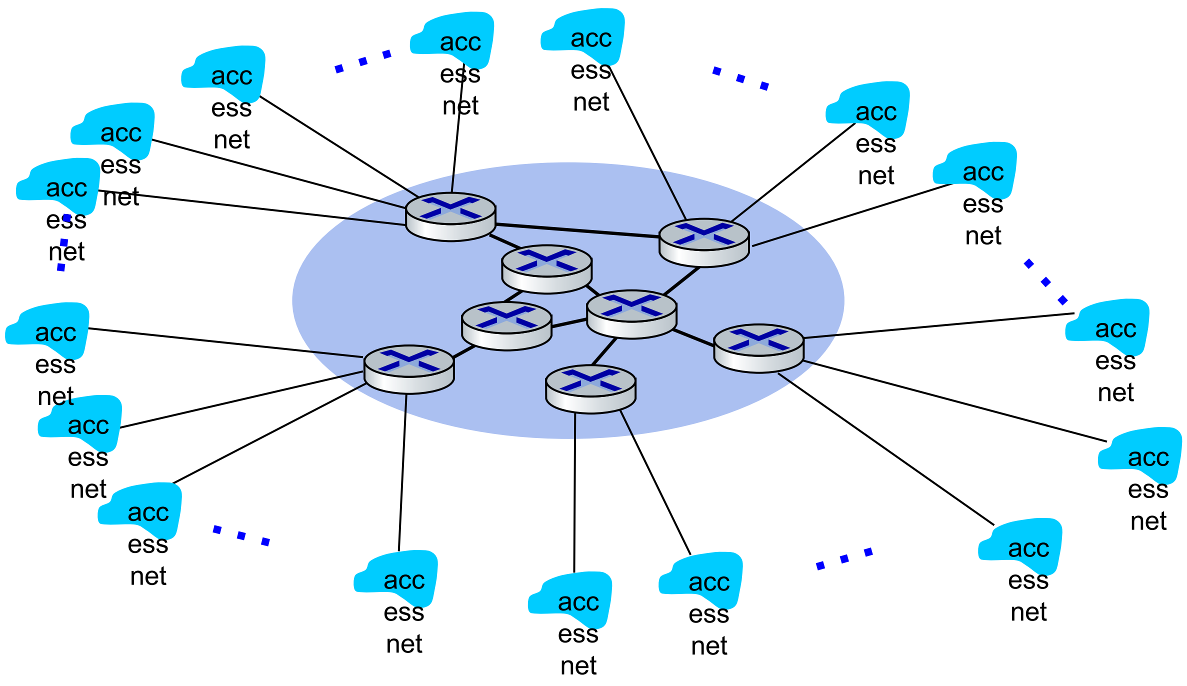
- Any two end-hosts (anywhere!) can talk to each other via the Internet
- How to connect millions of millions of nets?



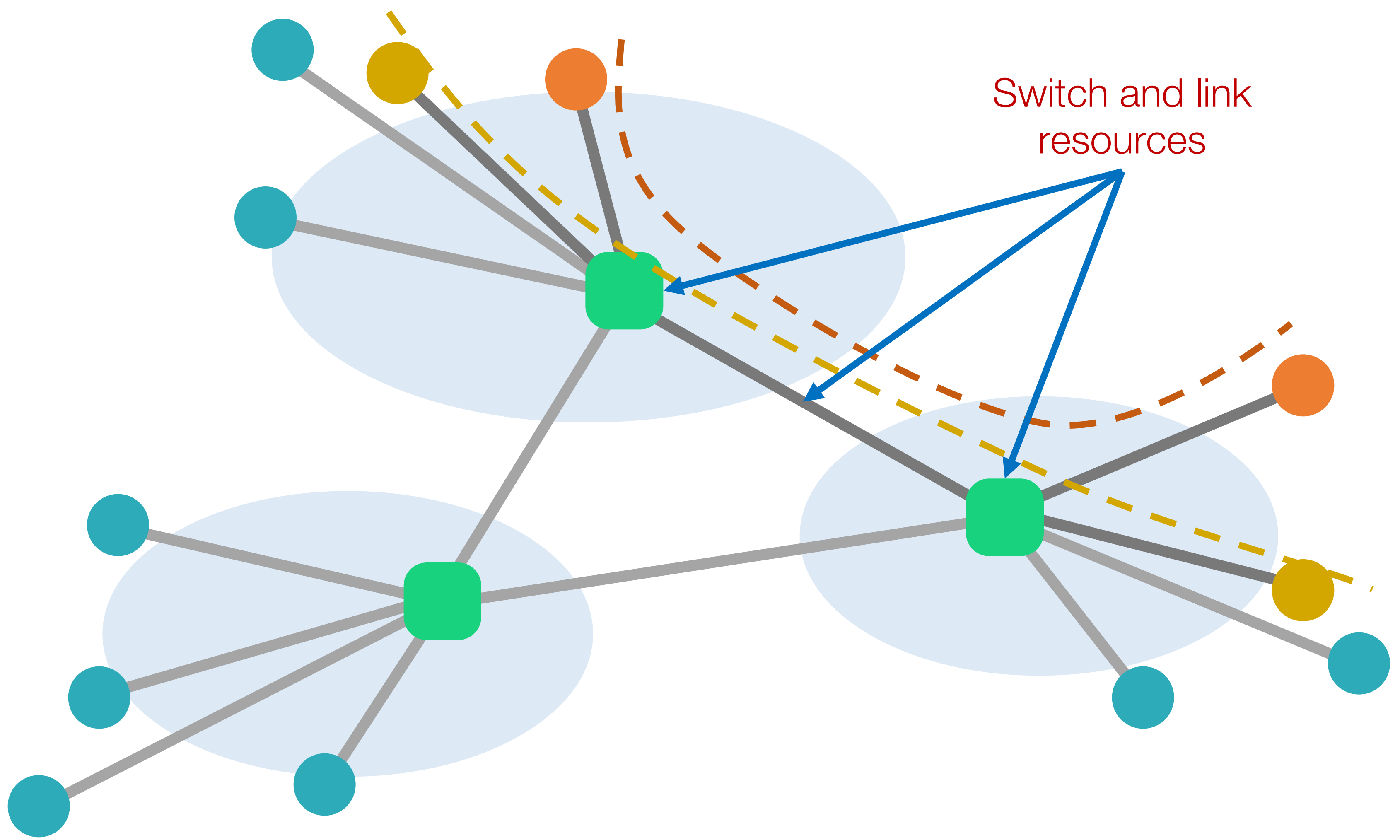
connecting each access ISP to each other directly doesn’t scale: $O(N^2)$ connections.

Switched networks

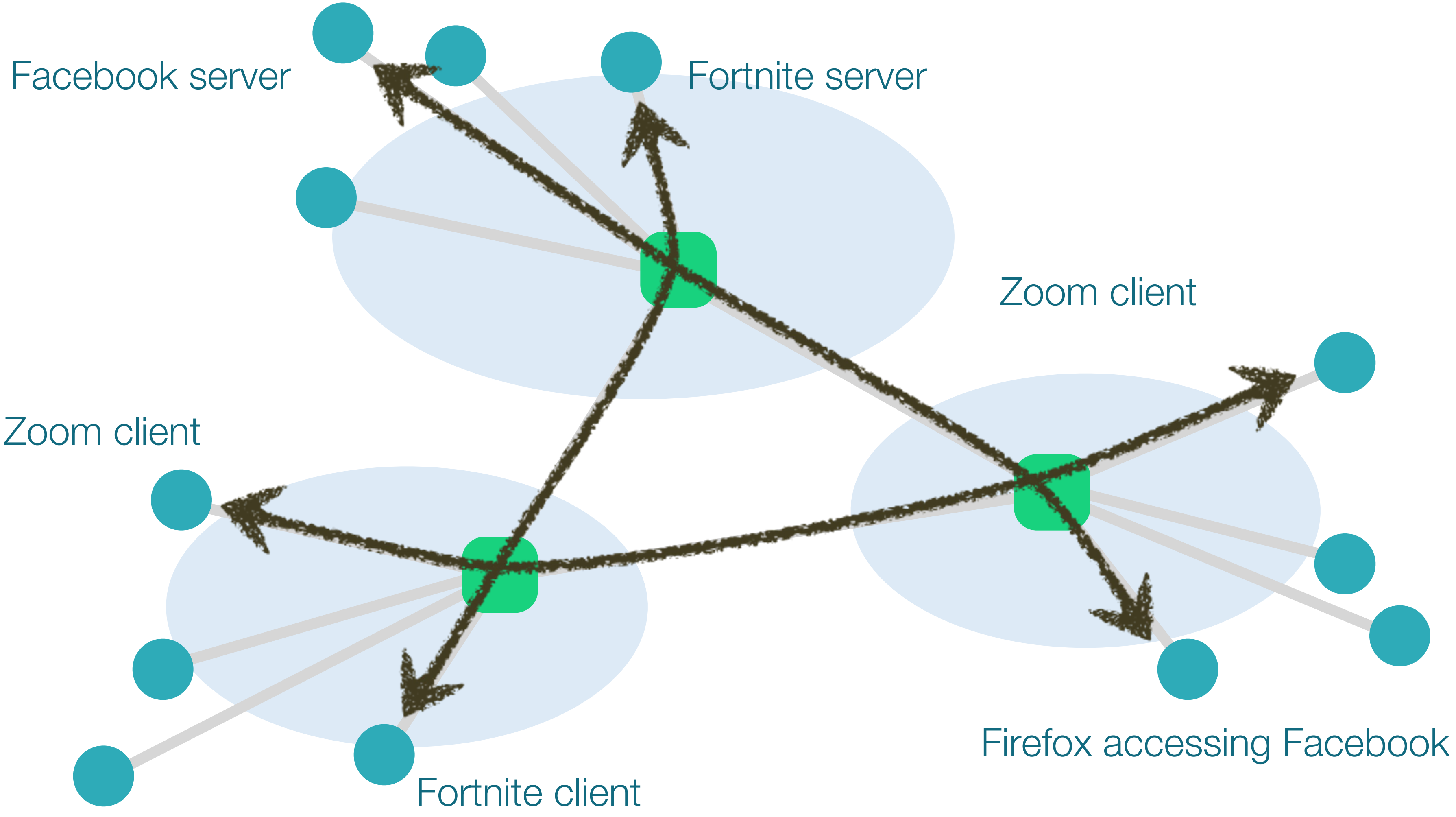
- Solution: Switched networks
  - Instead of directly connecting end-systems and networks
    - Use switches to connect them
    - Allows us to scale!



When do we need to share the network?



Shared among many services



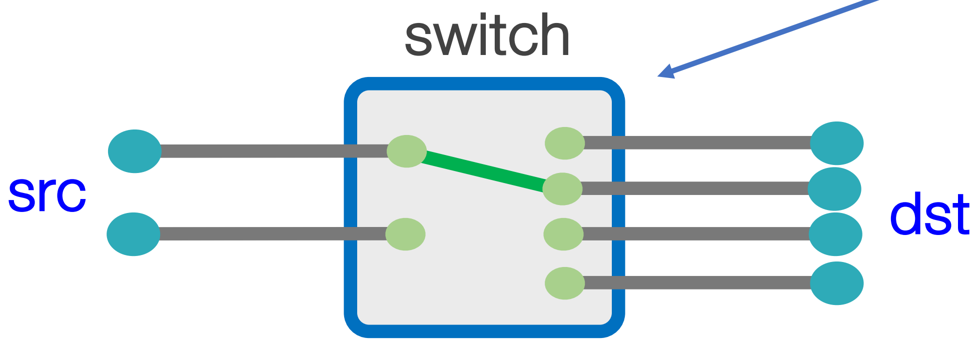
Two ways to share switched networks

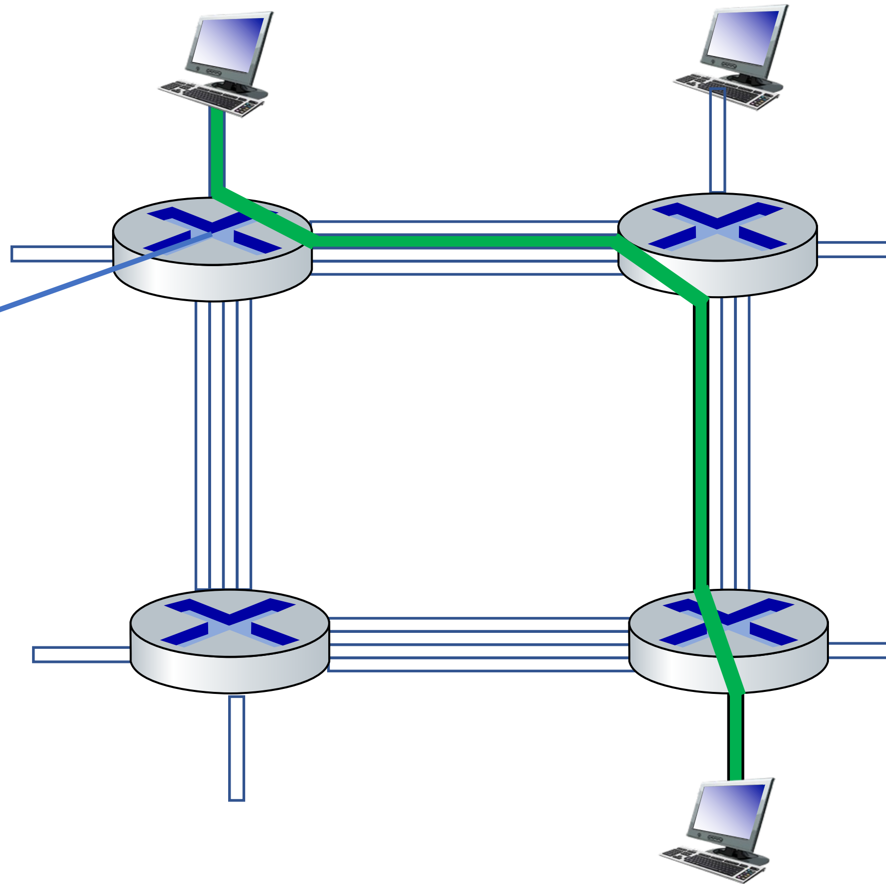
- Two ways to share switched networks
  - Circuit switching
    - Resource reserved per connection
    - Admission control: per connection
  - Packet switching via statistical multiplexing
    - Packets treated independently, on-demand
    - Admission control: per packet

Circuit switching

End-end resources allocated to, reserved for “call” between source and destination

- In diagram, each link has four circuits.
  - call gets 2nd circuit in top link and 1st circuit in right link.
- Reservation establishes a “circuit” within a switch
- Dedicated resources: no sharing
  - circuit-like (guaranteed) performance
- circuit segment idle if not used by call (no sharing)
- commonly used in traditional telephone networks



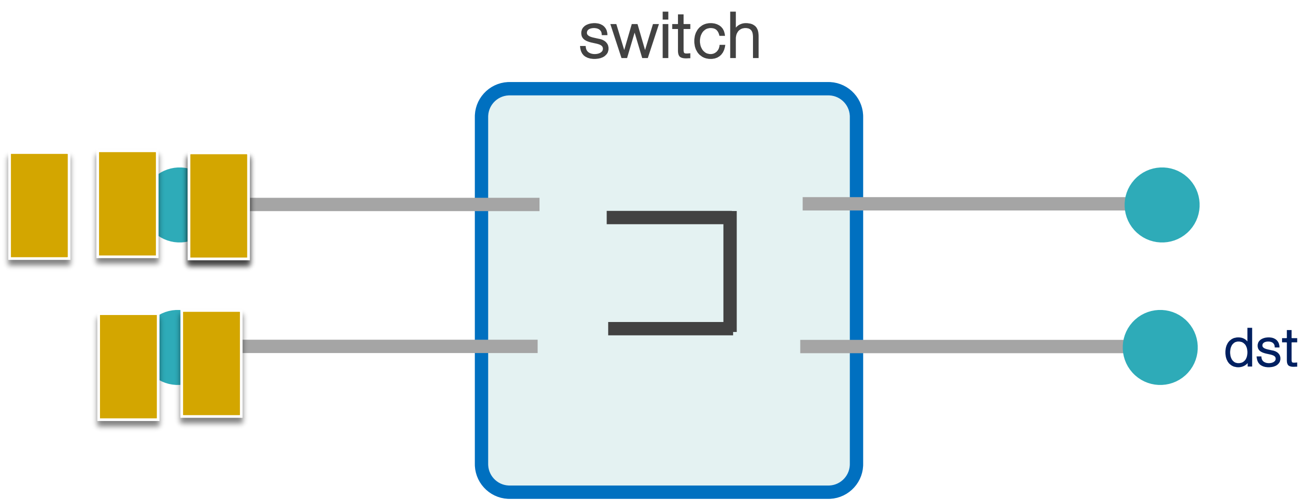


Circuit switching

- Pros
  - Predictable performance
  - Simple/fast switching (once circuit established)
- Cons
  - Complexity of circuit setup/teardown
  - Circuit setup adds delay
  - Switch fails its circuit(s) fails
  - Extremely inefficient when traffic is bursty!
    - Think: Are you always clicking?

Packet switching: Store and forward

- Each packet contains destination (dst)
- Each packet treated independently
- With buffers to absolve transient overloads



Packet switching

- Pros
  - Efficient use of network resources
  - Simpler to implement
  - Robust: can “route around trouble”
- Cons
  - Unpredictable performance
  - Requires buffer management and congestion control

Statistical multiplexing

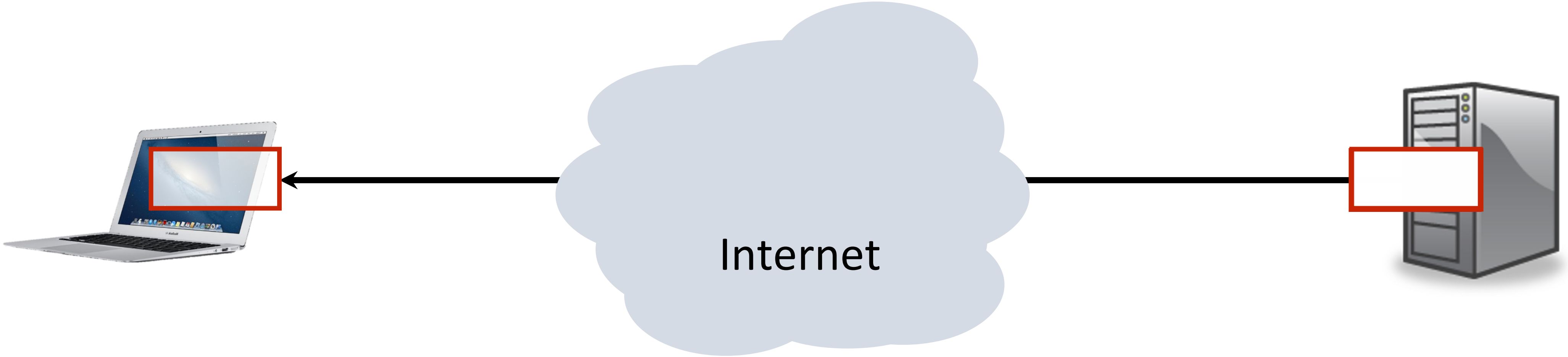
- Allowing more demands than the network can handle
  - Hoping that not all demands are required at the same time
  - Results in unpredictability
  - Works well except for the extreme cases

Performance metrics

Loss, delay, throughput

Performance metrics

- How does one evaluate the performance of Internet communication?
- Consider a source end-system that is sending data to a destination end-system over the Internet.



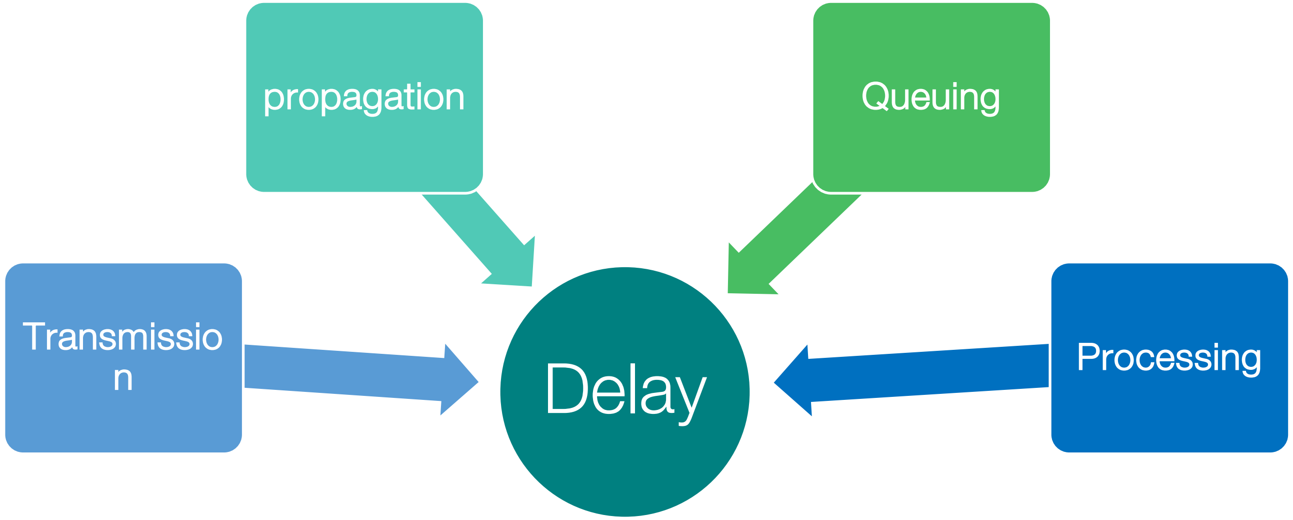
- What are the simple performance metrics to quantify the quality of this communication?

Performance metrics

- What are the simple performance metrics to quantify the quality of this communication?
- We choose three metrics
  1. Delay
     - How long does it take to send a packet from its source to destination?
  2. Loss
     - What fraction of the packets sent to a destination are dropped?
  3. Throughput
     - At what rate is the destination receiving data from the source

Performance metrics: Delay

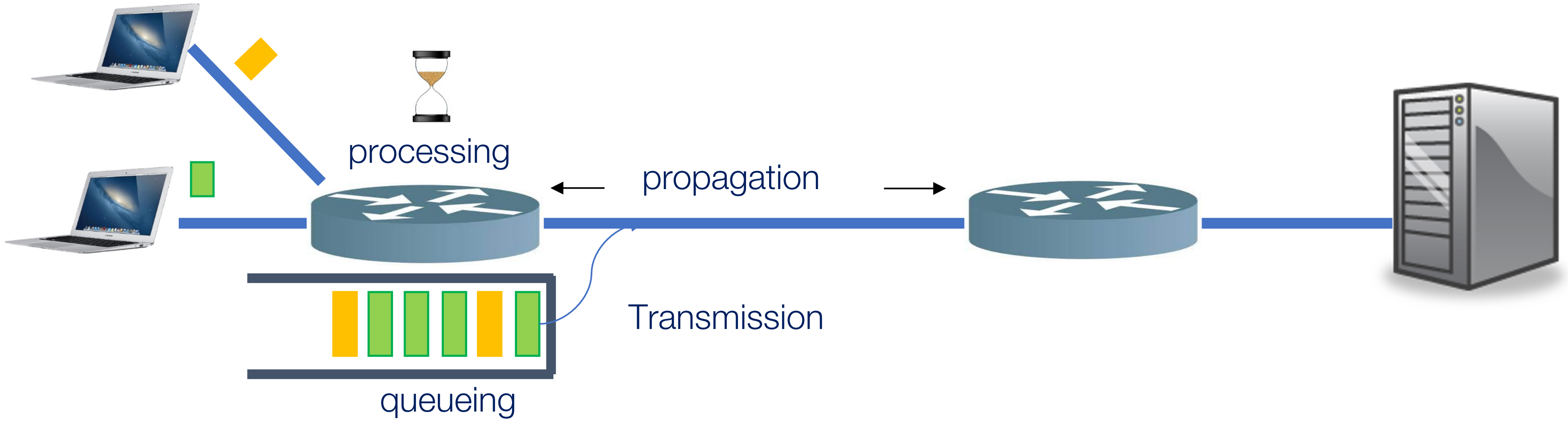
- How do packet delay and loss occur?
- They are four sources for packet Delay



$$\text{Delay} = d_{\text{processing}} + d_{\text{queuing}} + d_{\text{transmission}} + d_{\text{propagtion}}$$

Performance metrics: Delay

- They are four sources for packet Delay



Processing delay

- check bit errors
- determine output link
- typically < microsecs

Queueing delay

- time waiting at output link for transmission
- depends on congestion level of router

Transmission delay:

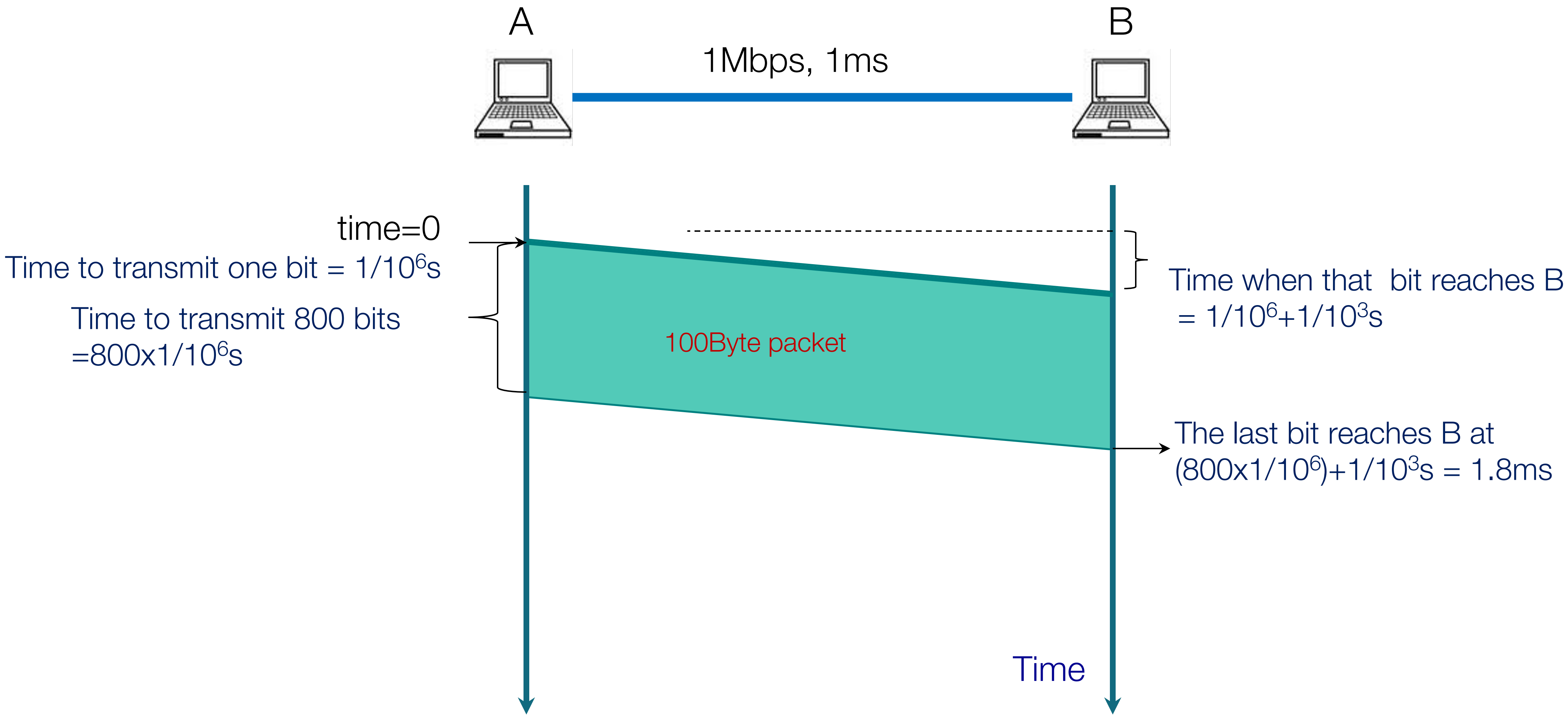
- *L*: packet length (bits)
- *R*: link *transmission rate (bps)*
- $d_{trans} = L/R$

Propagation delay:

- *d*: length of physical link
- *s*: propagation speed (~2x10^8 m/sec)
- $d_{prop} = d/s$

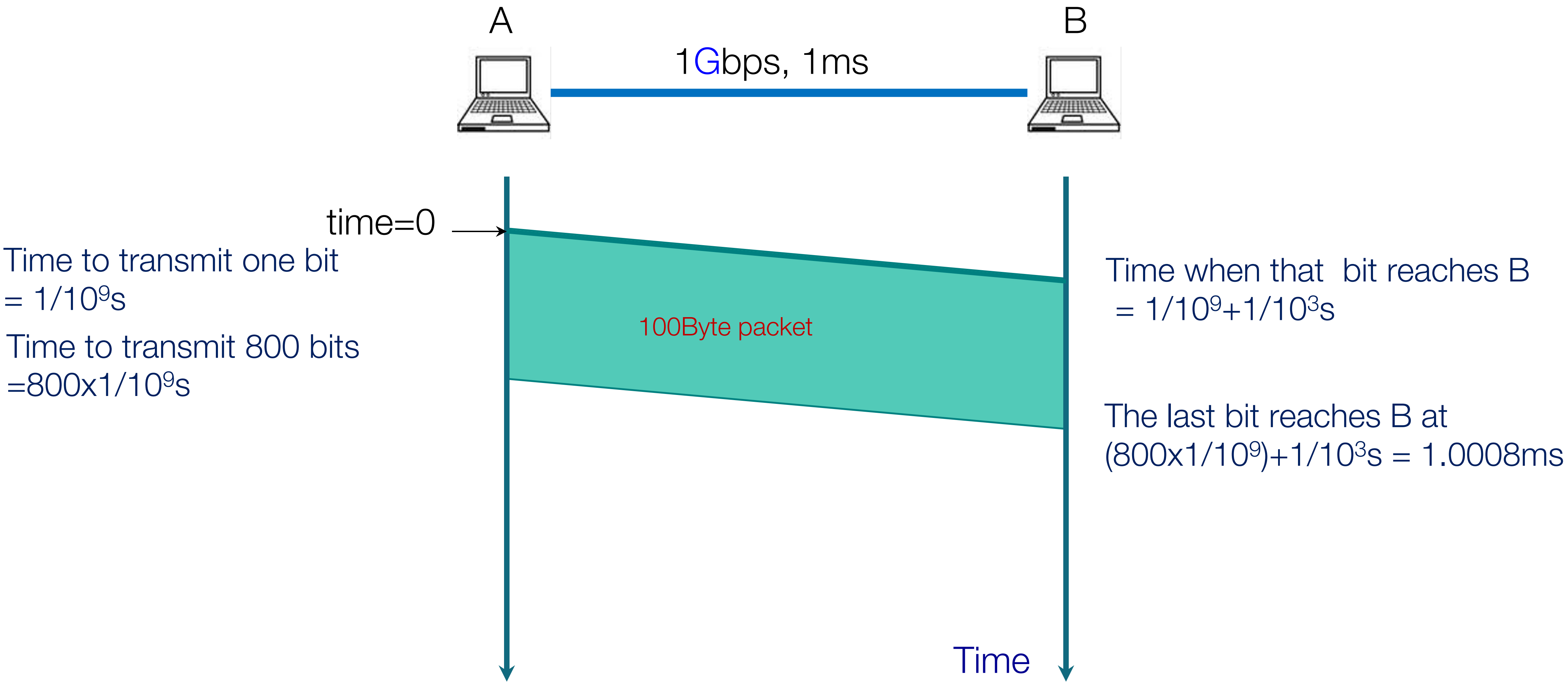
Transmission delay & propagation delay : Example

- Packet delay sending a 100-byte packet



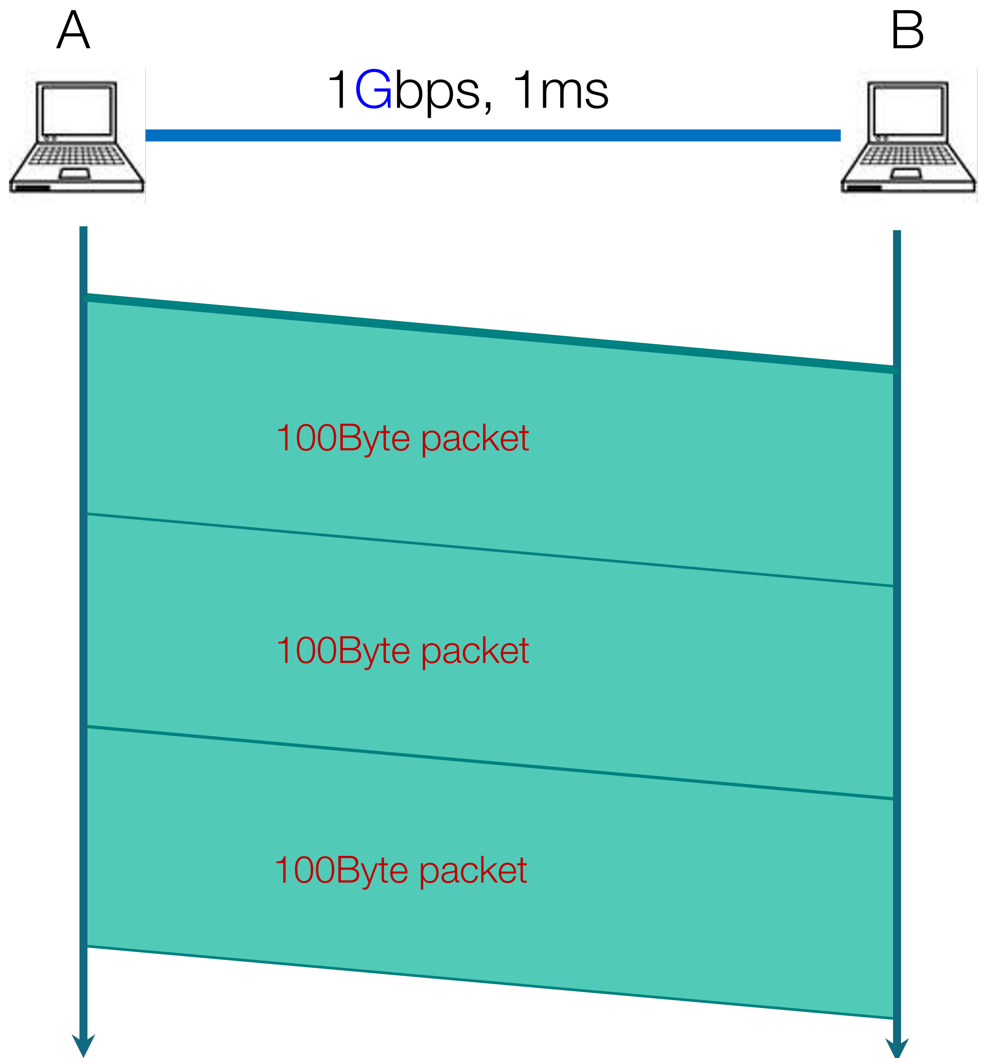
Transmission delay & propagation delay : Example

- Packet delay sending a 100-byte packet



Transmission delay & propagation delay : Example

- Packet delay sending a large file using 100-byte packets



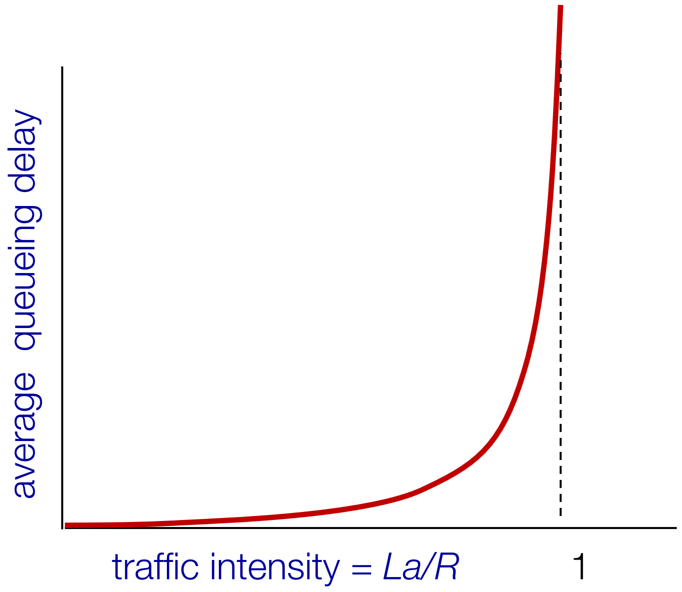
Packet queueing delay

- How long does a packet have to sit in a buffer before it is processed?
- Depends on traffic pattern
  - Arrival rate at the queue
  - Nature of arriving traffic (bursty or not?)
  - Transmission rate of outgoing link
- Characterized with statistical measures
  - Average queuing delay
  - Variance of queuing delay
  - Probability delay exceeds a threshold value

Packet queueing delay

- a: average packet arrival rate (pkt/s)
- L: packet length (bits)
- R: link bandwidth (bits/s)

Traffic intensity: $\frac{L \cdot a}{R} = \frac{\text{arrival rate of bits}}{\text{service rate of bits}}$



measure of the average occupancy of a server or resource during a specified period of time,

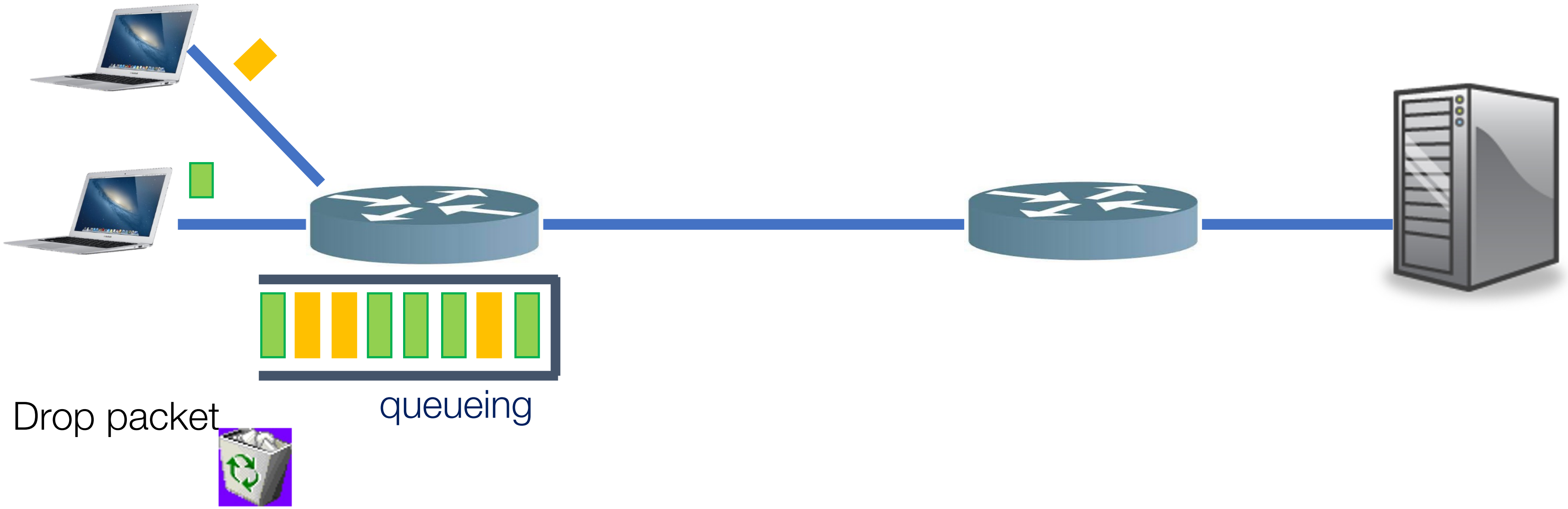
- La/R ~ 0: avg. queueing delay small
- La/R < 1: The router can handle more average traffic
- La/R > 1: The rate at which bits arrive exceeds the rate bits can be transmitted and queuing delay will grow without bound

Processing delay

- How long does the switch take to process a packet?
    - Generally negligible

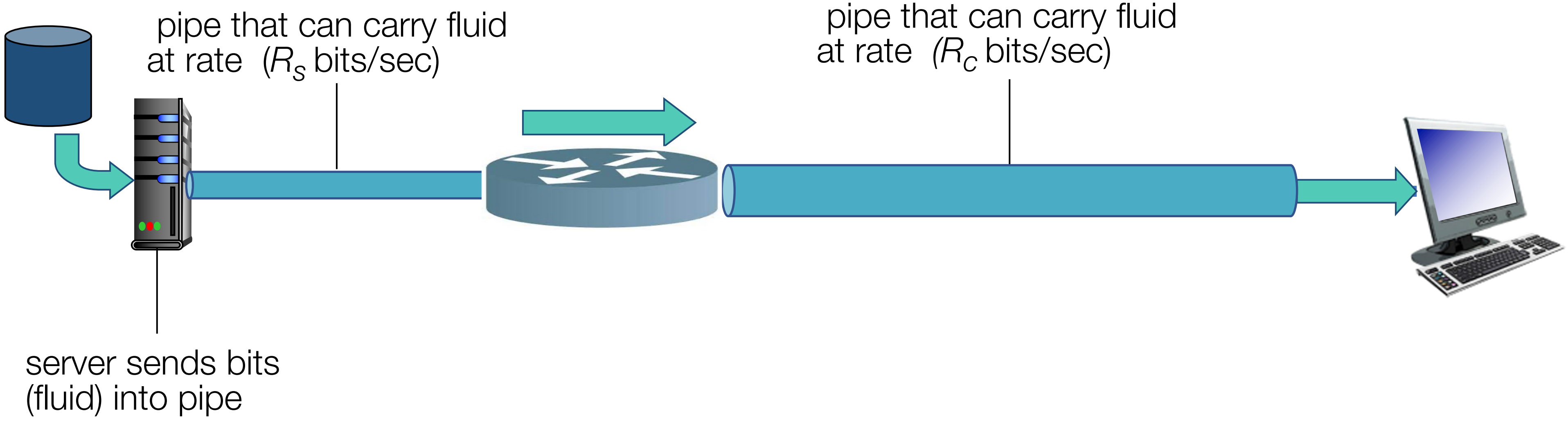
Packet loss

- What fraction of the packets sent to a destination are dropped?
- How do packet loss occur?
  - Queue (aka buffer) preceding link in buffer has finite capacity
  - Packet arriving to full queue dropped (aka lost)
- Lost packet may be retransmitted by previous node, by source, or not at all



Throughput

- Throughput: rate (bits/time unit) at which bits are being sent from sender to receiver
  - Instantaneous: rate at given point in time
  - Average: rate over longer period of time



Round Trip Time (RTT)

- Time for a packet to go from a source to a destination and to come back
- Why do we care?
  - Measuring delay is hard from one end
- RTT/2 equals *average* end-to-end delay
  - Why not exact?

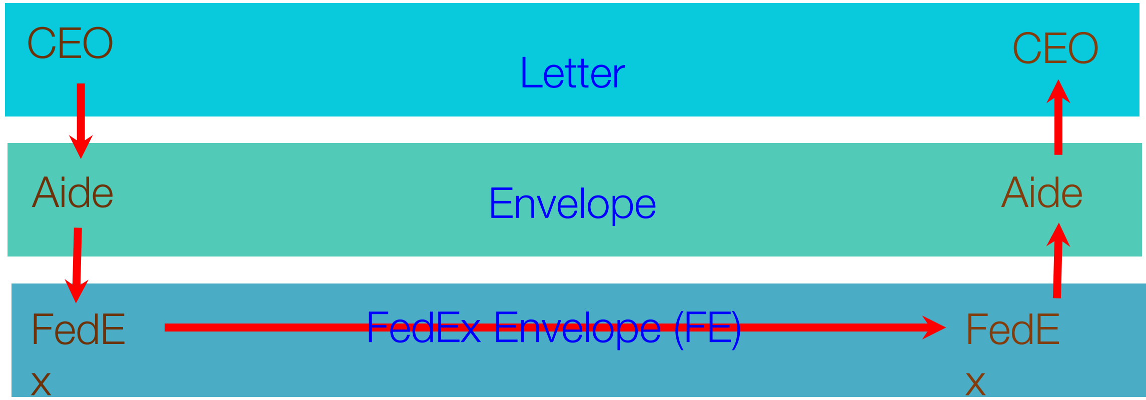
Protocol layers, service models

Protocol “layers” and reference models

- Networks are complex, with many “pieces”:
  - hosts
  - routers
  - links of various media
  - applications
  - protocols
  - hardware, software
- Question: is there any hope of organizing structure of network?
  - and/or our discussion of networks?

Inspiration

- CEO A writes letter to CEO B
  - Folds letter and hands it to administrative aide
- Aide:
  - Puts letter in envelope with CEO B's full name
  - Takes to FedEx
- FedEx Office
  - Puts letter in larger envelope
  - Puts name and address on FedEx envelope
  - Puts package on FedEx delivery truck
- FedEx delivers to other company



The path of the letter

- Layers: each layer implements a service
  - “Peers” in same layer understand each other
  - relying on services provided by layer below

CEO	Semantic Content	CEO
Aide	Identity	Aide
FedE x	Location	FedE x

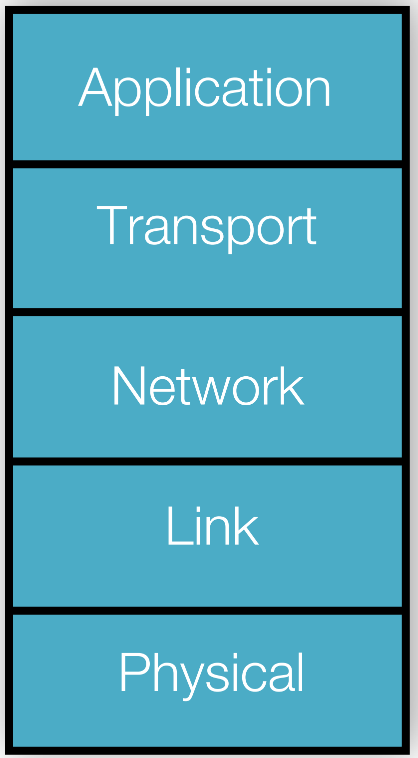
- Lowest level has most packaging

Why layering?

- Approach to designing/discussing complex systems:
  - explicit structure allows identification, relationship of system’s pieces
    - layered *reference model* for discussion
  - modularization eases maintenance, updating of system
    - change in layer's service *implementation*: transparent to rest of system
- Method:
  - Decompose the problem into tasks
  - Organize these tasks
  - Assign tasks to entities (who does what)

Layered Internet protocol stack

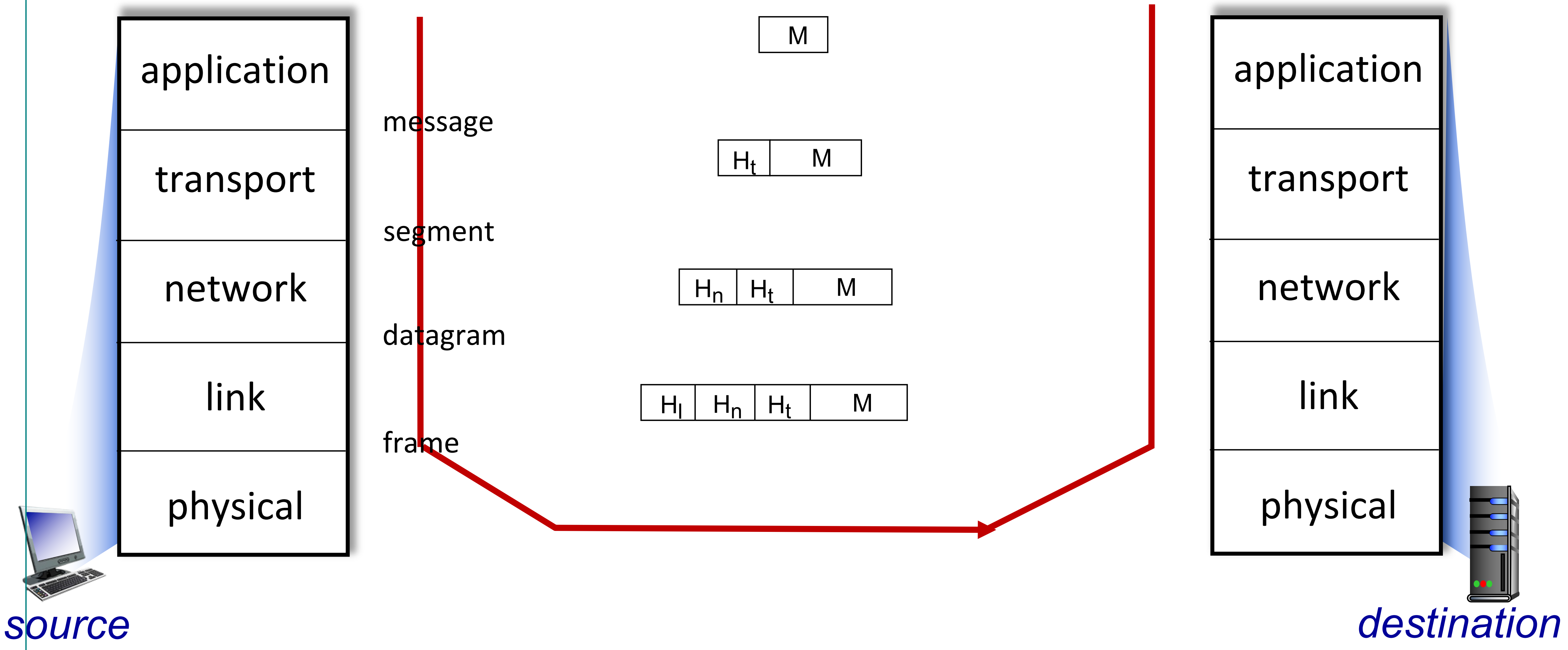
- Application: supporting network applications
  - HTTP, IMAP, SMTP, DNS
- Transport: process-process data transfer
  - TCP, UDP
- Network: routing of datagrams from source to destination
  - IP, routing protocols
- Link: data transfer between neighboring network elements
  - Ethernet, 802.11 (WiFi), PPP
- Physical: bits “on the wire”



Layered Internet protocol stack

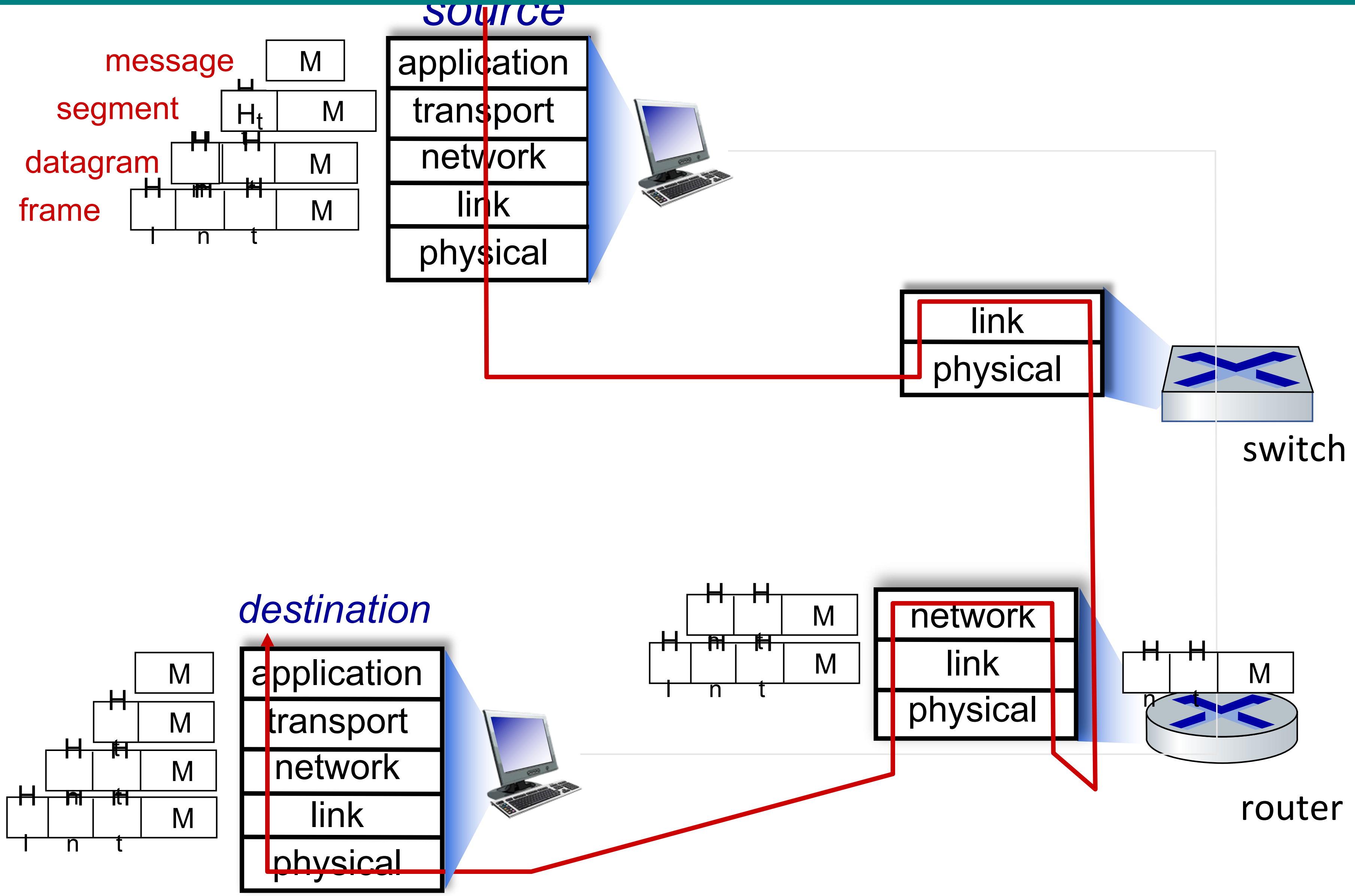
- What gets implemented at the end systems?
  - Bits arrive on wire, must make it up to application
  - Therefore, all layers must exist at host!
- What gets implemented at routers and switches?
  - Switches only need to support physical and link layers
  - Routers support physical, link and network layers

Services, Layering and Encapsulation

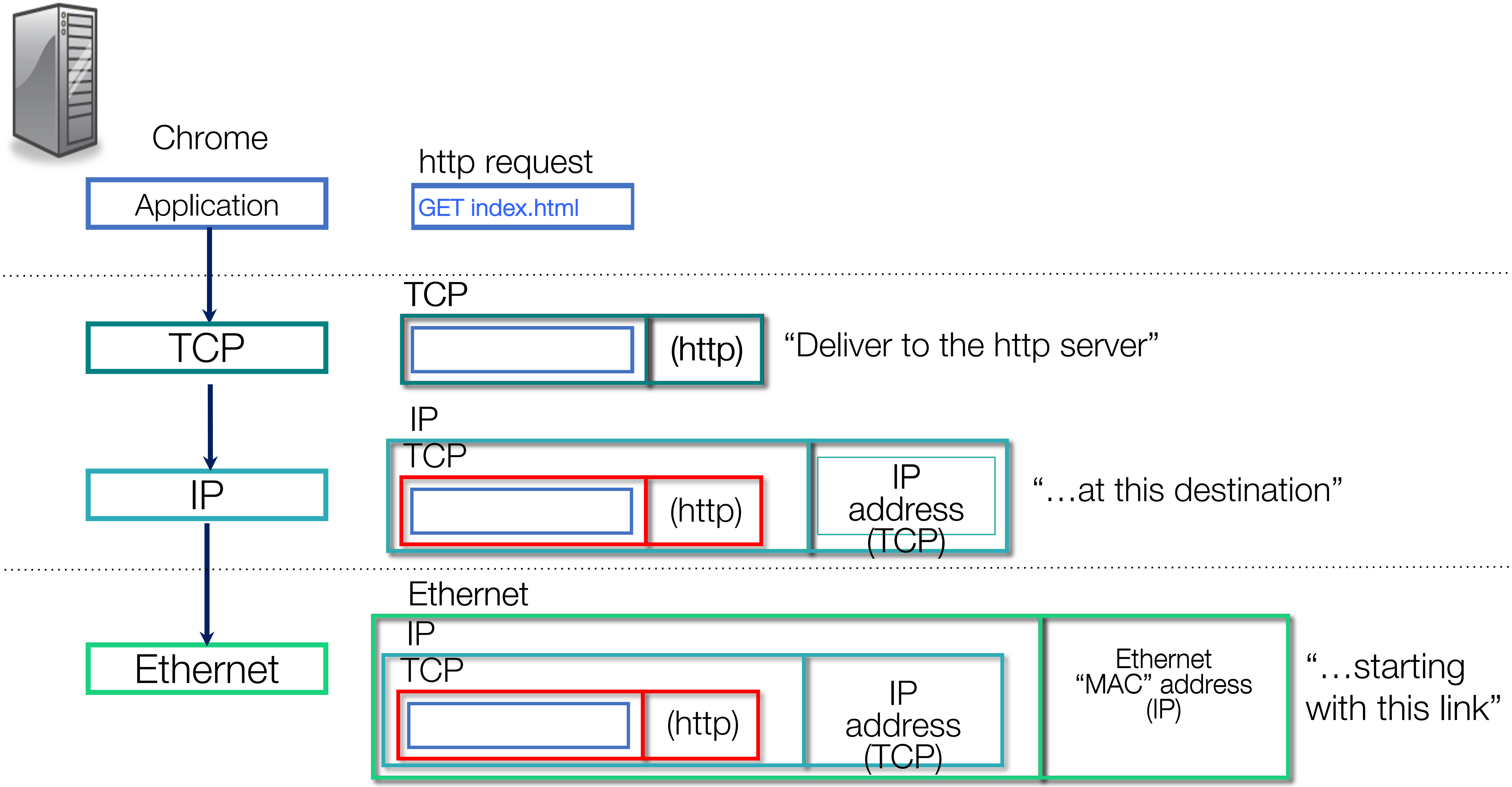


ensia

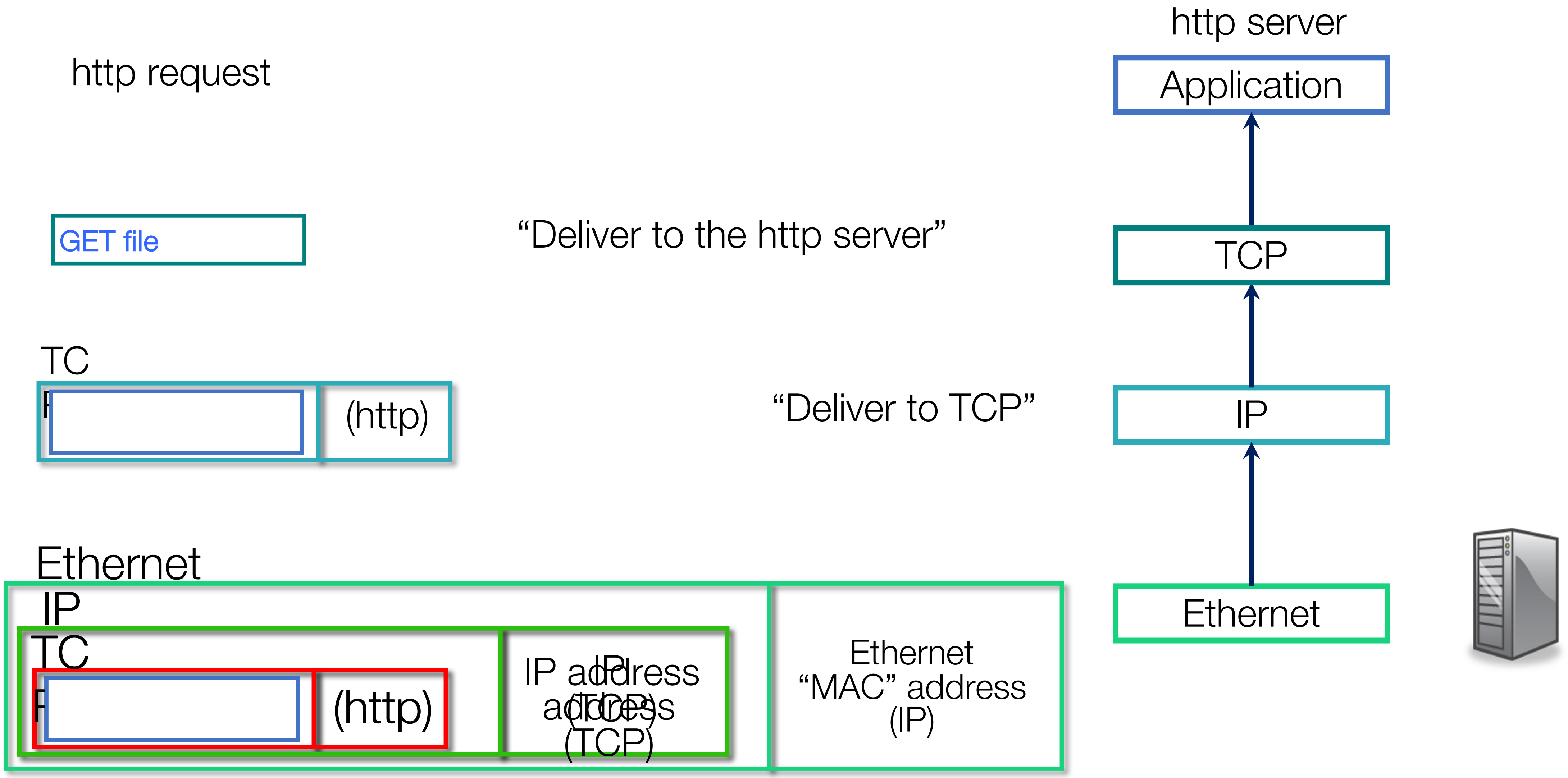
Encapsulation: an end-end view



Encapsulation: Example



Encapsulation: Example



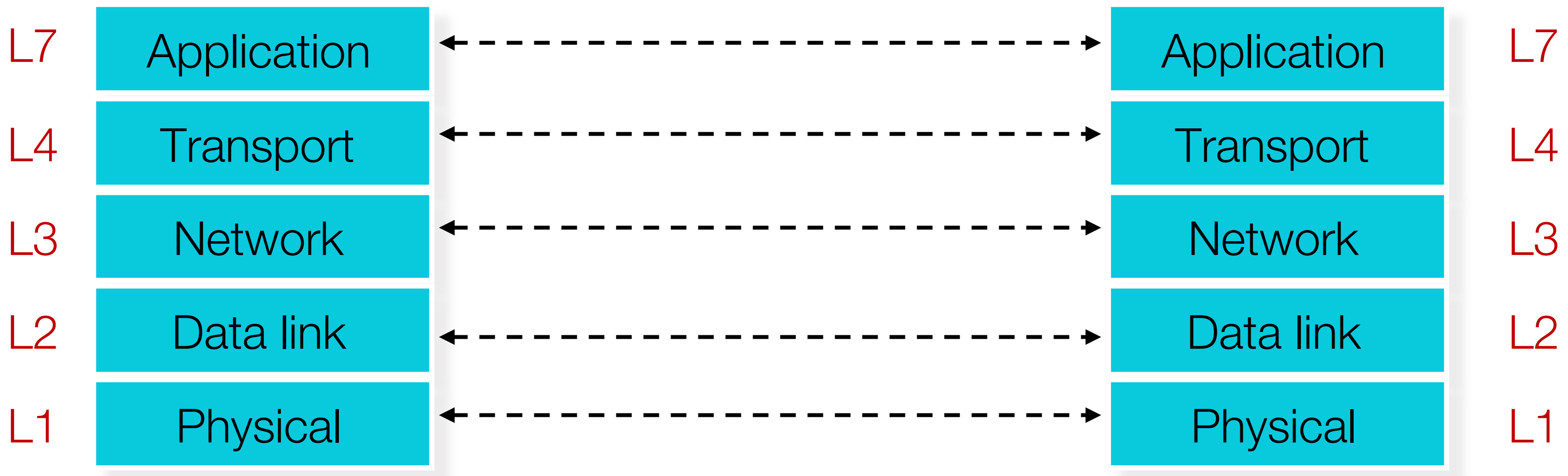
OSI layers

- OSI stands for Open Systems Interconnection model
  - Developed by the ISO
- Session and presentation layers are often implemented as part of the application layer

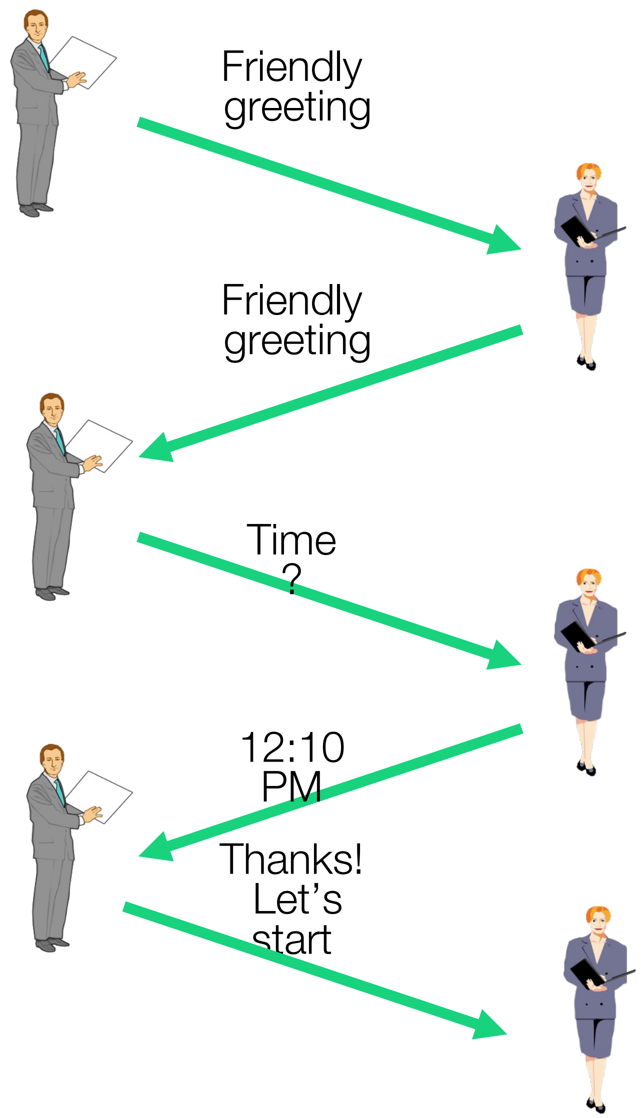
L7	Application
L6	Presentation
L5	Session
L4	Transport
L3	Network
L2	Data link
L1	Physical

Layers and protocols

- Communication between peer layers on different systems is defined by protocols



What is a Protocol?



- Protocol: An agreement between parties (in the same layer) on how to communicate, it defines
  - syntax of communication
    - Header □ instructions on how to process payload
    - Each protocol defines the format of its headers
      - e.g., "the first 32 bits carry the destination address"
  - And semantics
    - "First a hello, then a request…"
    - We will study many protocols later in the semester
    - Protocols exist at many levels, hardware, and software
    - Defined by standards bodies like IETF, IEEE, ITU.

Credit

Kurose, JHU, CMU, Stanford