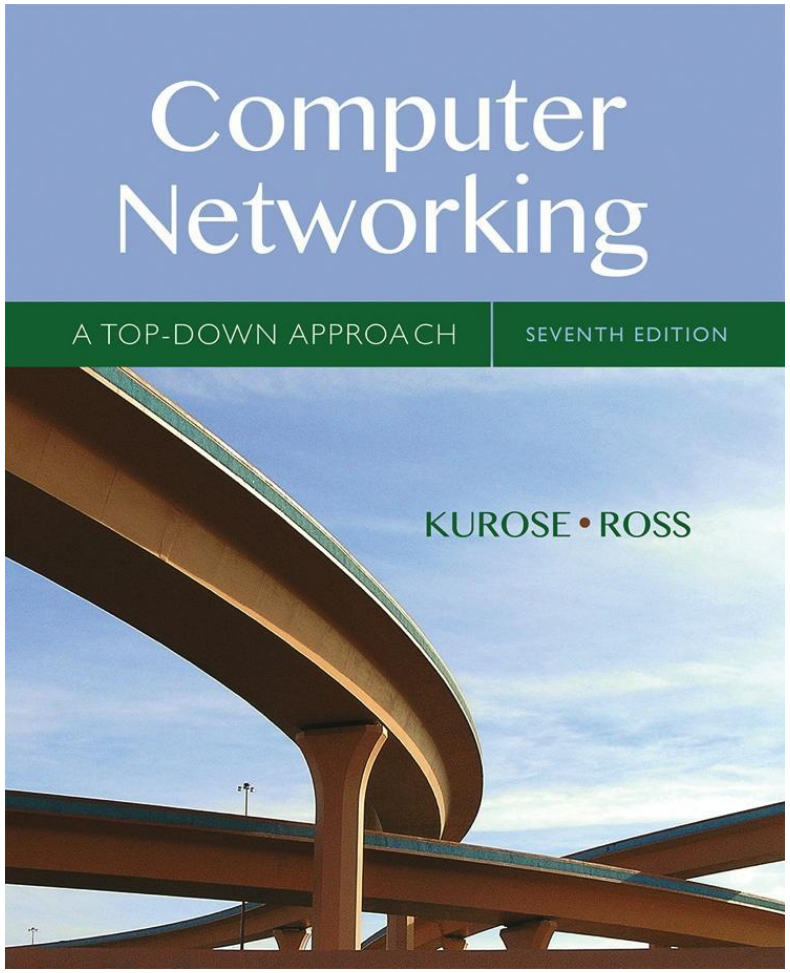
COMPUTER NETWORKS

Computer Networks and the Internet

Department of Computer Science and Engineering

Text Book



Computer Networking: A Top-Down Approach
Jim Kurose, Keith Ross
Pearson, 2017
7th Edition

COMPUTER NETWORKS

Computer Networks and the Internet

Compiled by KUNDHAVAI K R

Department of Computer Science and Engineering

Unit – 1 Computer Networks and the Internet

1.1 Introduction to Computer Networks

1.2 What is the Internet?

- A nuts-and-bolts and Services description, Protocol

1.3 Network edge

- End systems, Access networks, Physical media

1.4 Network core

- Packet switching, Circuit switching, Network structure

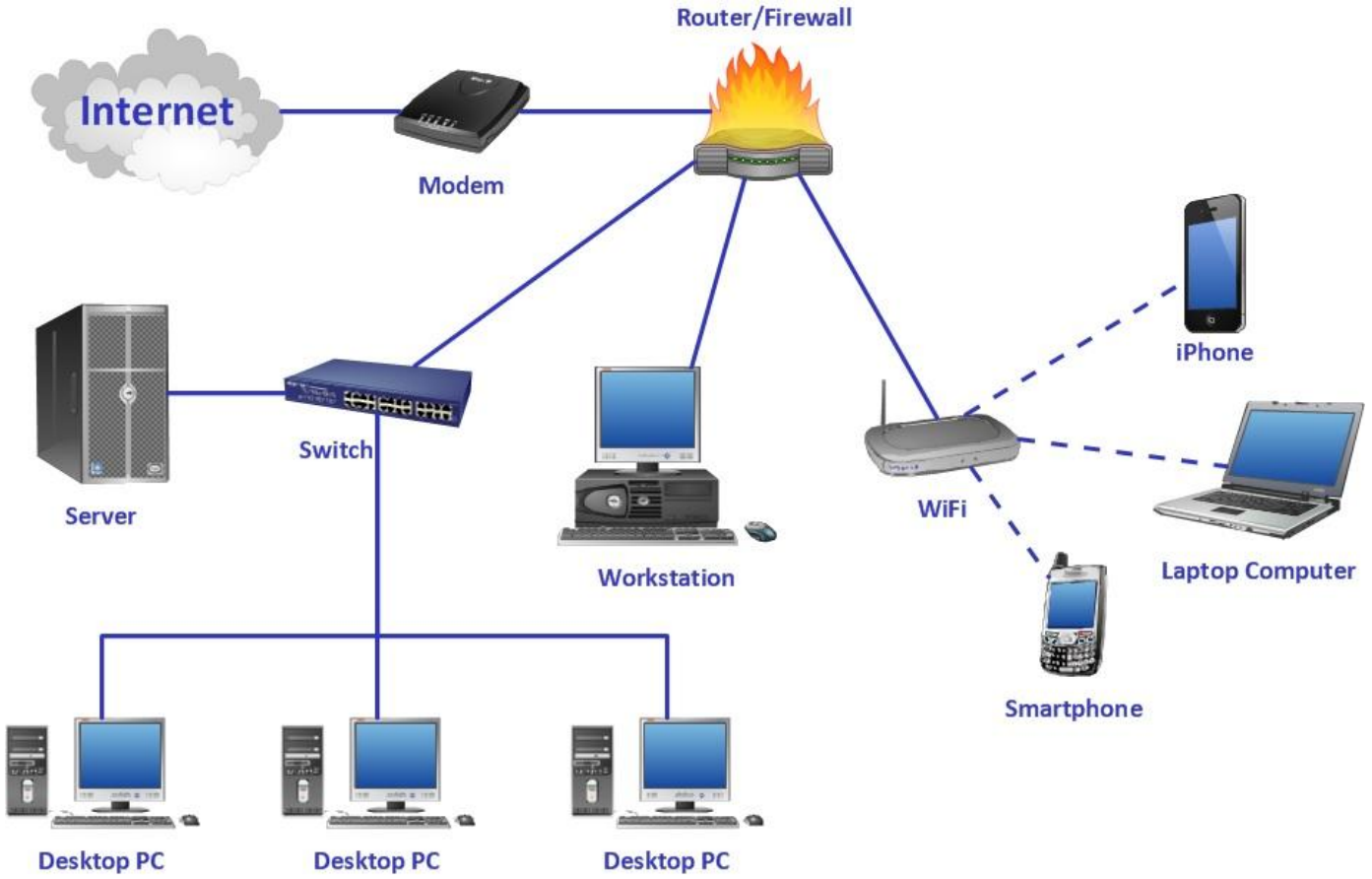
1.5 Delay, Loss & Throughput in networks

1.6 Protocol layers, Service models

- OSI model and TCP/IP protocol suite

COMPUTER NETWORKS

Introduction to Computer Networks



- Two or more devices connected together.
- Communicate with each other, share data or resources

What is the Internet?

- A massive network of networks.
- A computer network that interconnects billions of computing devices throughout the world.
- Traditional devices – PCs, Workstations, Servers – web pages, emails, etc.
- Internet “things” – laptops, PDAs, TVs, gaming consoles, home security systems, home appliances, watches, cars, traffic control systems, etc.,
- There are a couple of ways to answer the question – WHAT IS THE INTERNET?
  - First, we can describe the nuts and bolts of the Internet, that is, the basic hardware and software components that make up the Internet.
  - Second, we can describe the Internet in terms of a networking infrastructure that provides services to distributed applications.

COMPUTER NETWORKS

The Internet: A "Nuts and Bolts" View

Billions of connected computing *devices*:

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

Packet switches: forward packets (chunks of data)

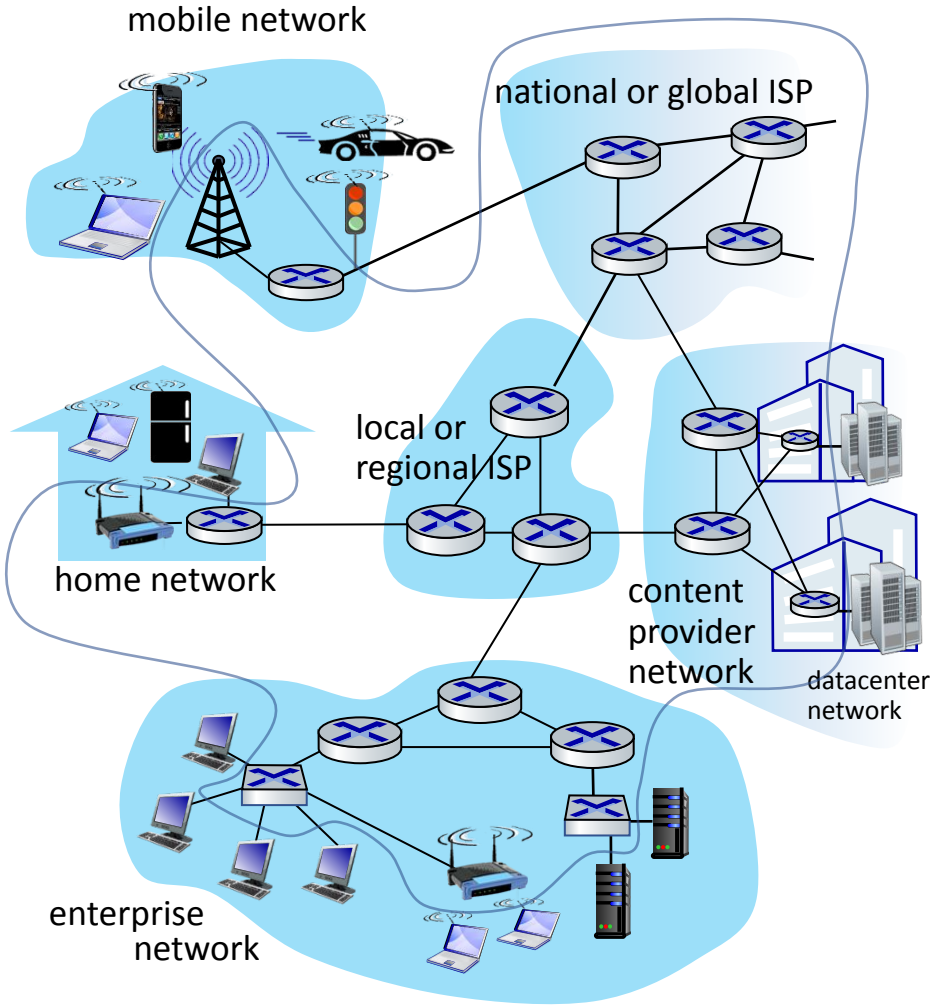
- *routers, switches*

Communication links

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

Networks

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



COMPUTER NETWORKS

"Fun" Inter-connected Devices

Amazon Echo

Internet refrigerator

IP picture frame

Tweet-a-watt: monitor energy use

Web-enabled toaster + weather forecaster

Security Camera

Slingbox: remote control cable TV

AR *devices*

Internet phones

sensorized, bed mattress

Fitbit

Others?

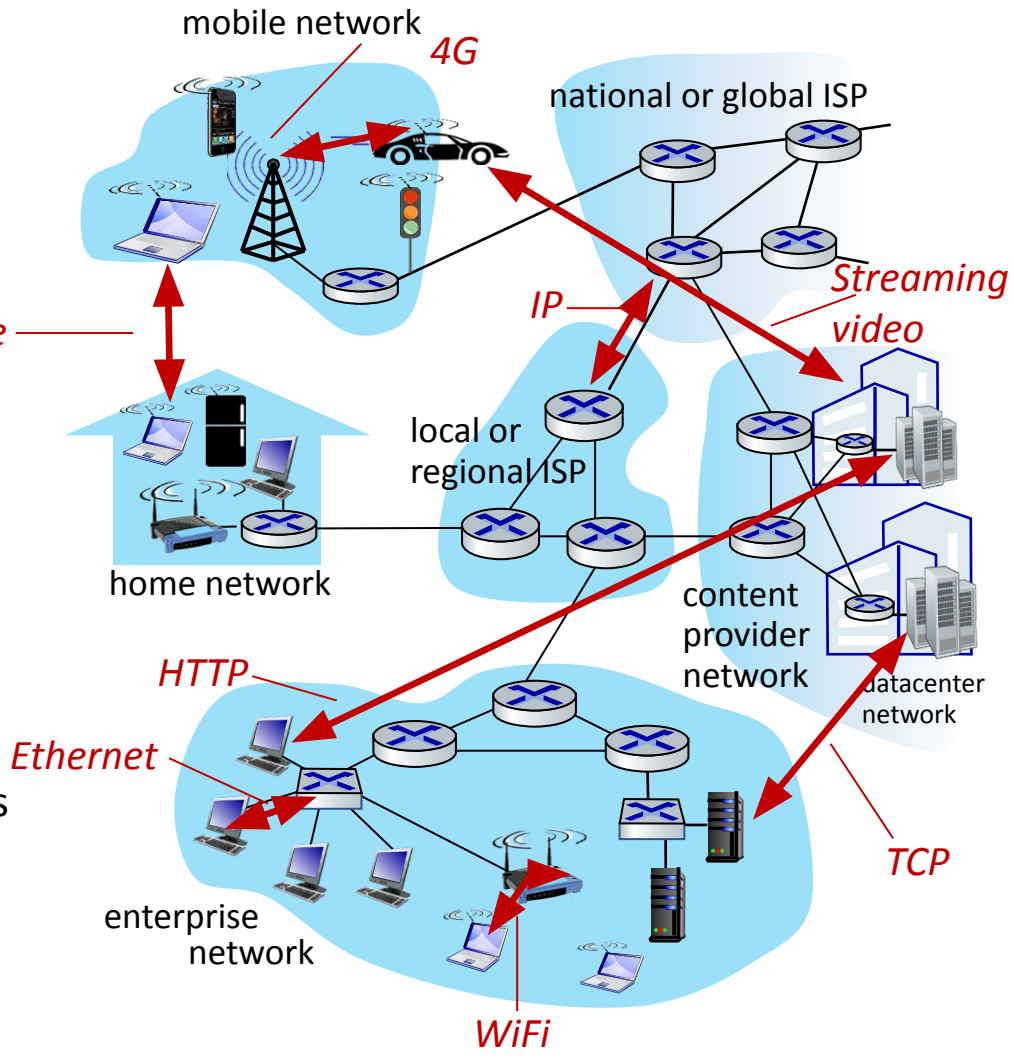
There will be 41 Billion IoT devices by 2027*

* https://www.businessinsider.com/internet-of-things-report?IR=T

COMPUTER NETWORKS

The Internet: A "Nuts and Bolts" View

- *Internet:* "network of networks"
  - Interconnected ISPs
  - Each ISP is in itself a network of packet switches and communication links
- *Protocols* are *everywhere*
  - control sending, receiving of messages
  - e.g., HTTP (Web), streaming video, Skype, TCP, IP, WiFi, 4G, Ethernet

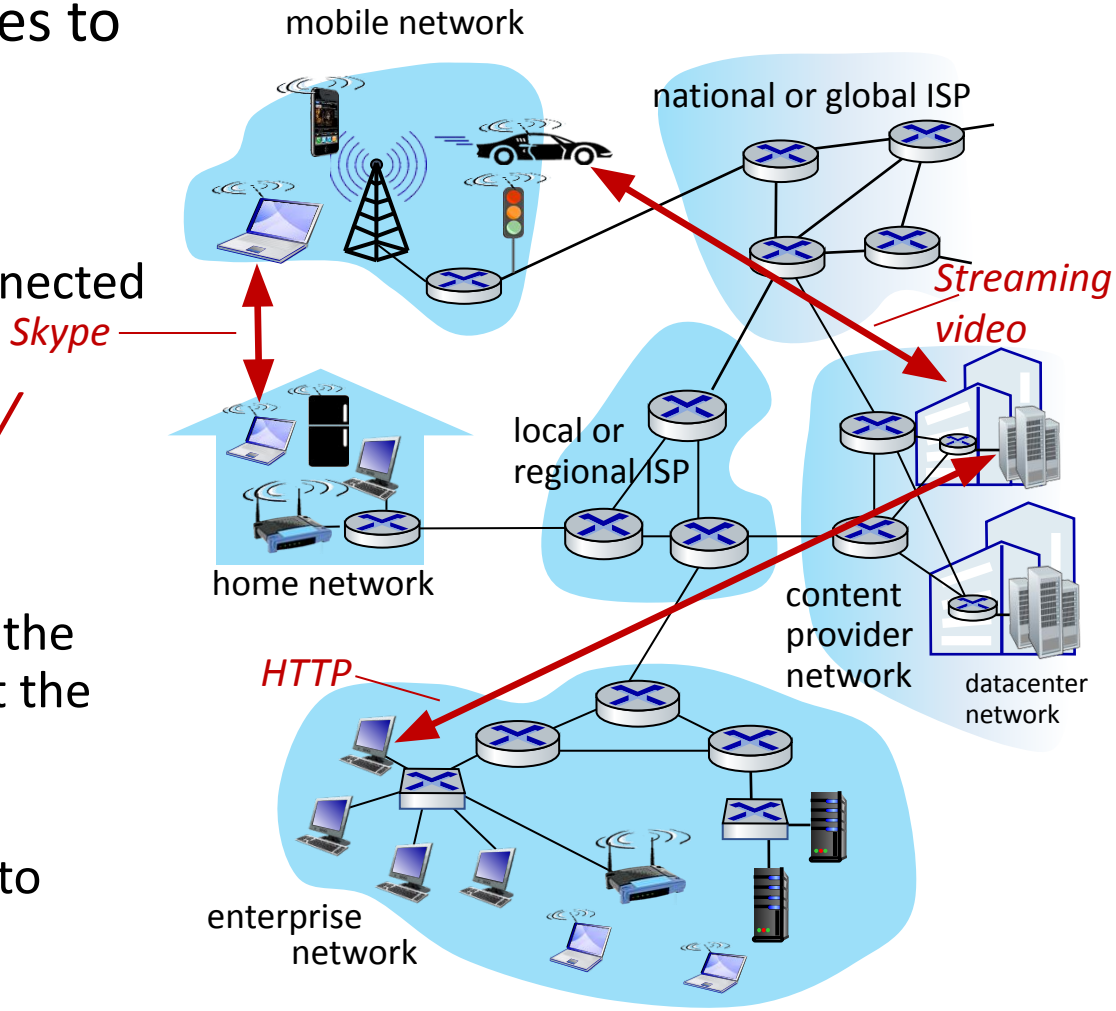


- *Internet standards*
- It's important that everyone agree on what each and every protocol does, so that people can create systems and products that interoperate. This is where standards come into play
  - RFC: Request for Comments
  - IETF: Internet Engineering Task Force

COMPUTER NETWORKS

The Internet: A "Service" View

- *Infrastructure* that provides services to applications:
  - Web, streaming video, multimedia teleconferencing, email, games, e-commerce, social media, inter-connected appliances, …
- provides *programming interface / socket interface* to distributed applications:
  - socket interface is a set of rules that the sending program must follow so that the Internet can deliver the data to the destination program.
  - provides service options, analogous to postal service



COMPUTER NETWORKS

What is a Protocol?

Human protocols:

- "what's the time?"
- "I have a question"
- introductions

… specific messages sent

… specific actions taken when message received, or other events

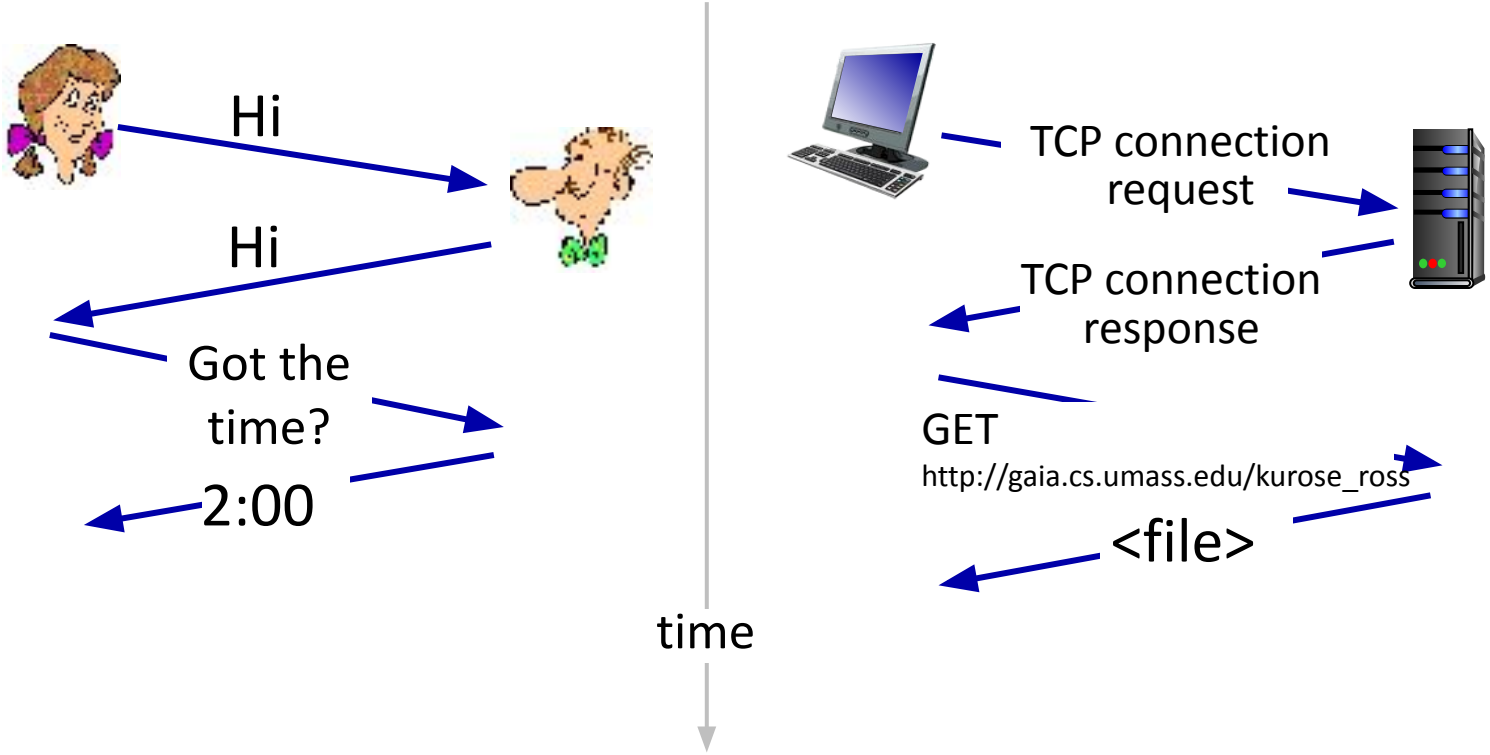
Network protocols:

- computers (devices) rather than humans
- all communication activity in Internet governed by protocols

What's a Protocol?

Consider what you do when you want to ask someone for the time of day.

A human protocol and a computer network protocol:



Network Protocol

- All activity in the Internet that involves two or more communicating remote entities is governed by a protocol.
- For example, hardware-implemented protocols in two physically connected computers
  - **control the flow of bits** on the “wire” between the two network interface cards;
  - congestion-control protocols in end systems **control the rate at which packets are transmitted** between sender and receiver;
  - protocols in routers **determine a packet’s path** from source to destination.
- Protocols are running everywhere in the Internet.

What's a Protocol – In a nutshell, definition

- If people run different protocols (for example, if one person has manners but the other does not, or if one understands the concept of time and the other does not) the protocols do not interoperate and no useful work can be accomplished.
- The same is true in networking—**it takes two (or more) communicating entities running the same protocol in order to accomplish a task.**

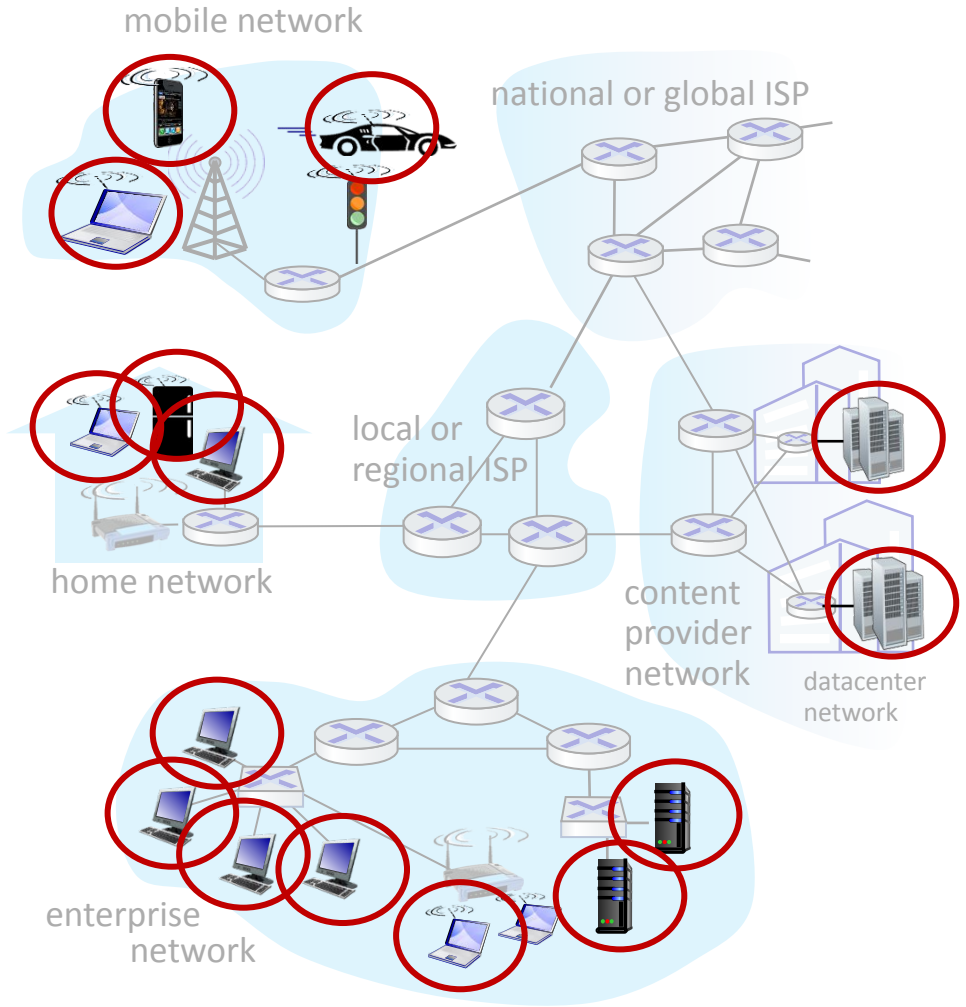
Protocols define the format, order of messages sent and received among network entities, and actions taken on msg transmission, receipt.

COMPUTER NETWORKS

Network Edge: A closer look at network structure

Network edge:

- Hosts: clients & servers
- Servers in data centers



DATA CENTERS:
Example:
Google has 50-100 data centers, including about 15 large centers, each with more than 100,000 servers.

COMPUTER NETWORKS

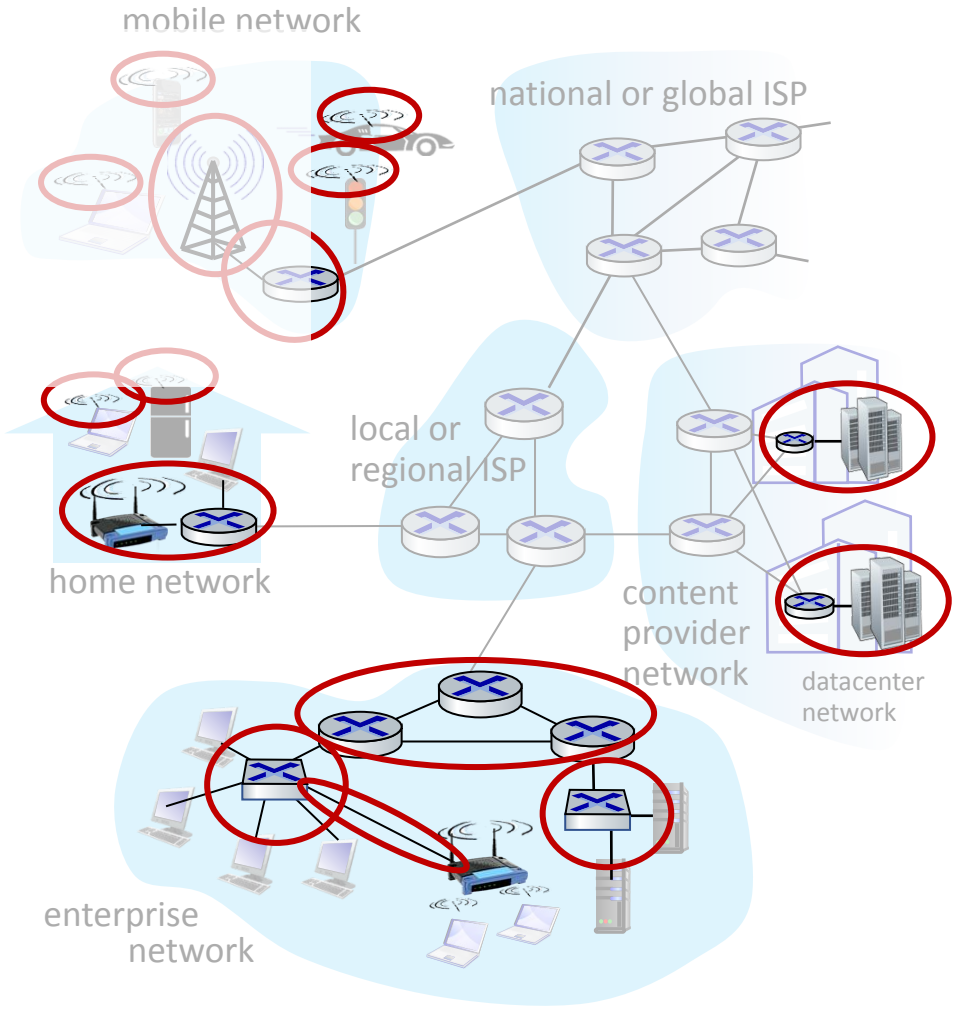
Network Edge: A closer look at network structure

Network edge:

- Hosts: clients & servers
- Servers in data centers

Access networks, physical media:

- wired, wireless communication links



COMPUTER NETWORKS

Network Edge: A closer look at network structure

Network edge:

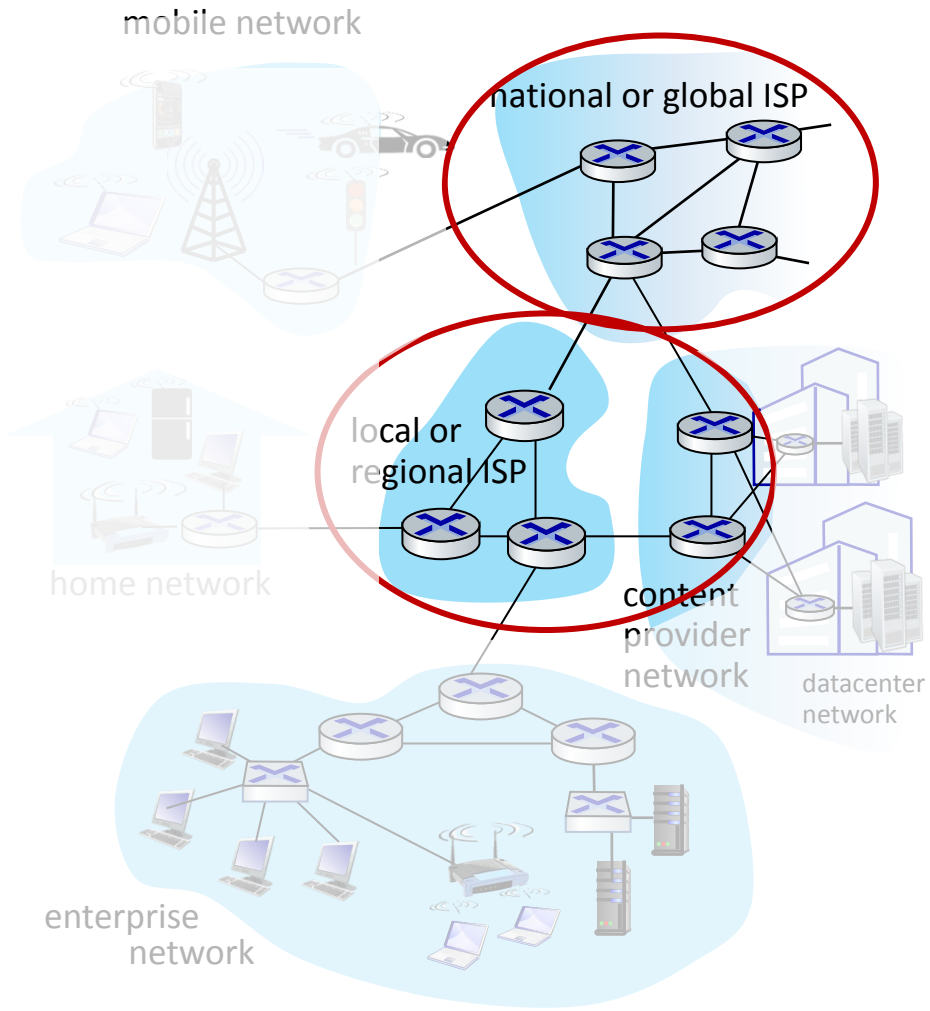
- Hosts: clients and servers
- Servers in data centers

Access networks, physical media:

- wired, wireless communication links

Network core:

- interconnected routers
- network of networks



COMPUTER NETWORKS

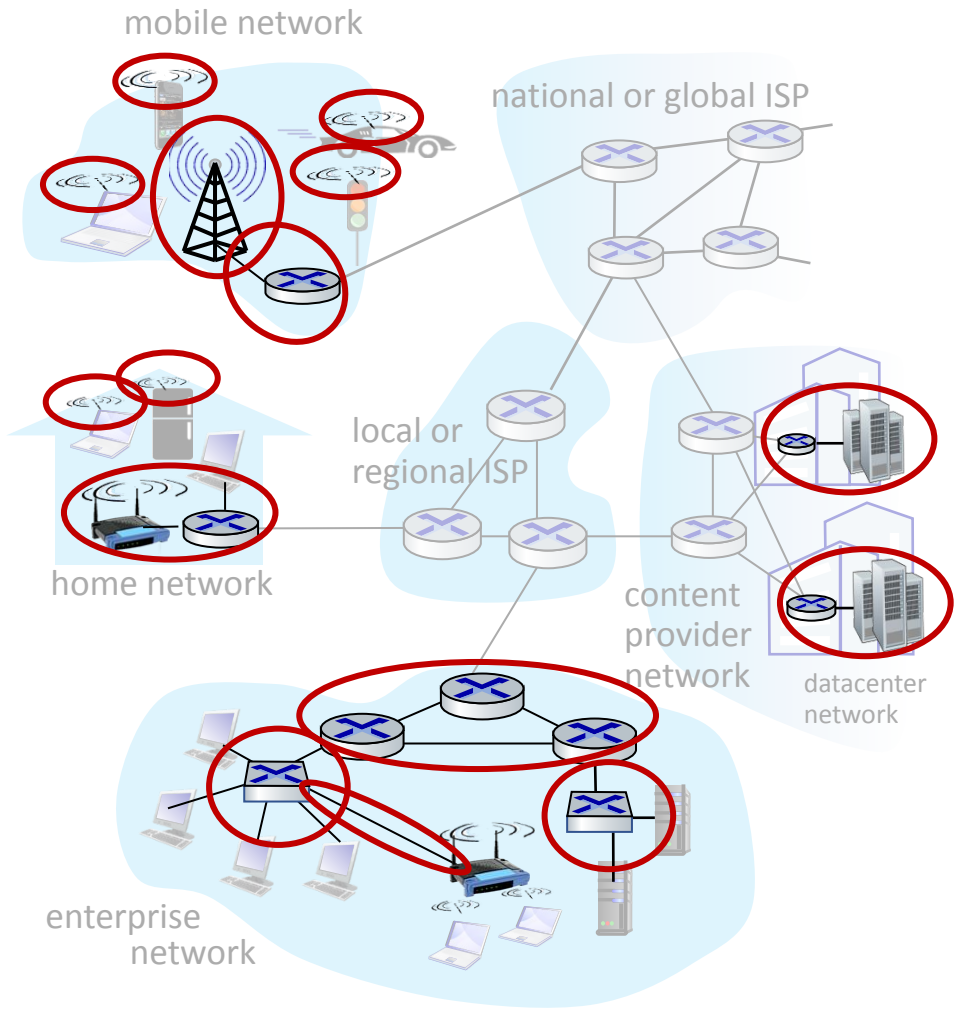
Network Edge: Access networks and Physical media

Q: How to connect end systems to edge router?

- Residential access networks
- Institutional access networks (school, company)
- Mobile access networks (WiFi, 4G/5G)

What to look for:

- Transmission rate (bits per second) of access network?
- Shared or dedicated access among users?



Access network—the network that physically connects an end system to the **first router (also known as the "edge router")** on a path from the end system to any other distant end system

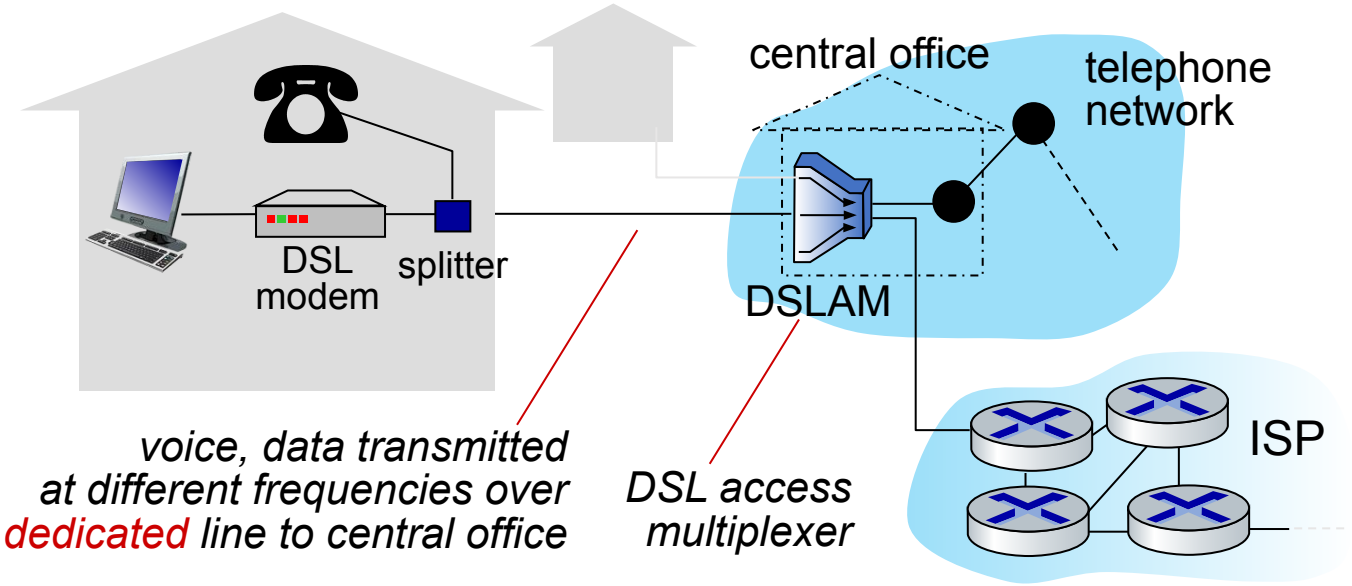
Access Networks – Residential / Home

- Today, the two most prevalent types of broadband residential access are
  - **Digital subscriber line (DSL) and**
  - **Cable.**

DSL vs CABLE

- Cable modem speeds are upto two times faster compared to DSL modem.
- Speeds in cable modem fluctuates depending on the number of subscribers on the network.but this is not the case in DSL.
- DSL boasts a bit better security compared to cable modem.

Network Edge: Access Networks - Digital Subscriber Line (DSL)



- 24-52 Mbps – downstream transmission rate
- 3.5-16 Mbps – upstream transmission rate
- Asymmetric access

DSL standards

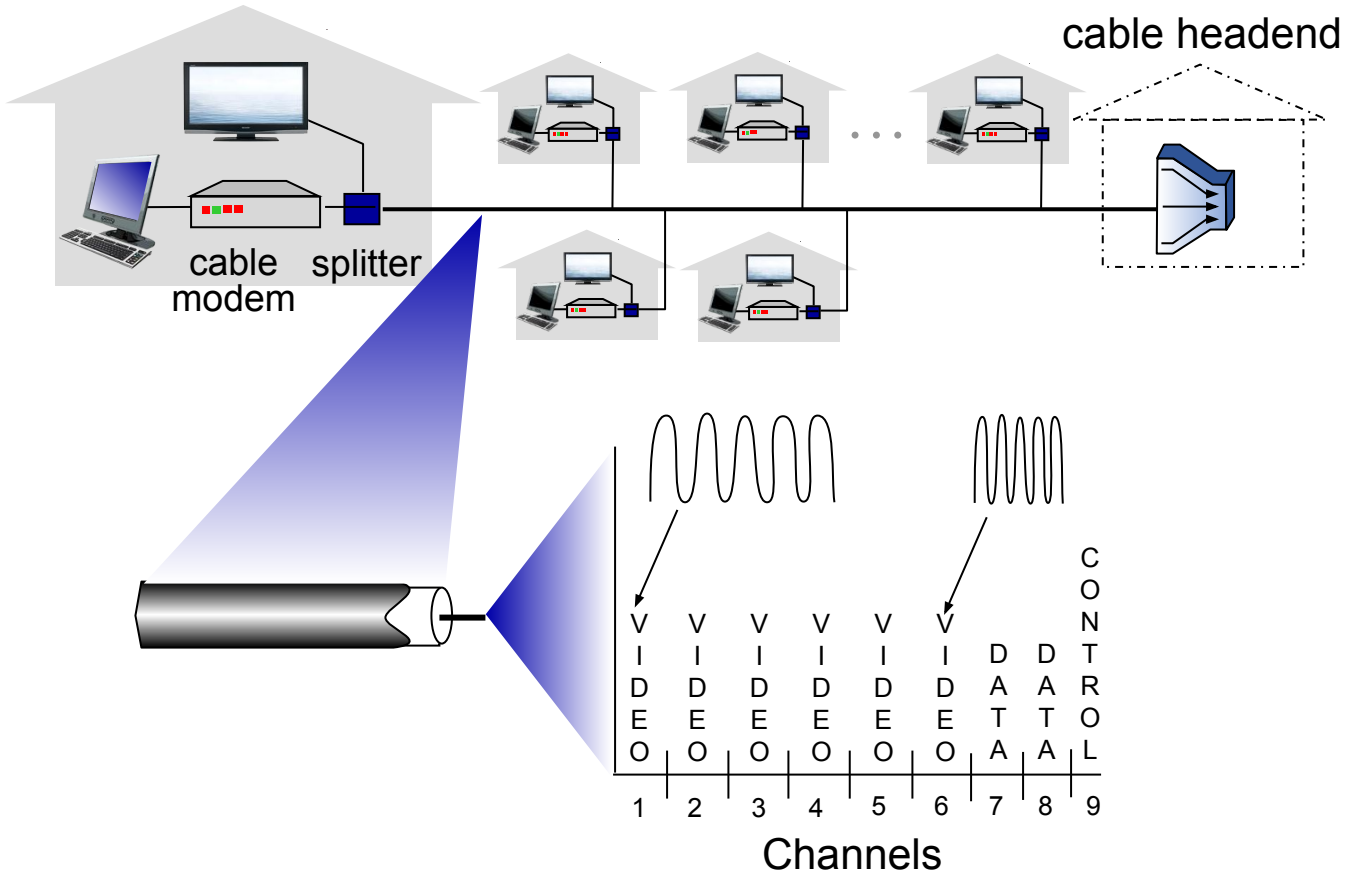
- use *existing* telephone line to central office DSLAM
  - **data** over DSL phone line goes to Internet
  - **voice** over DSL phone line goes to telephone net
- A high-speed downstream channel, in the 50 kHz to 1 MHz band
- A medium-speed upstream channel, in the 4 kHz to 50 kHz band
- An ordinary two-way telephone channel, in the 0 to 4 kHz band

data and telephone signals are encoded at different frequencies

Access Networks – Cable bases access

- Cable Internet access makes use of the cable television company's existing cable television infrastructure.
- A residence obtains cable Internet access from the same company that provides its cable television.
- Cable Internet access requires special modems, **called cable modems**

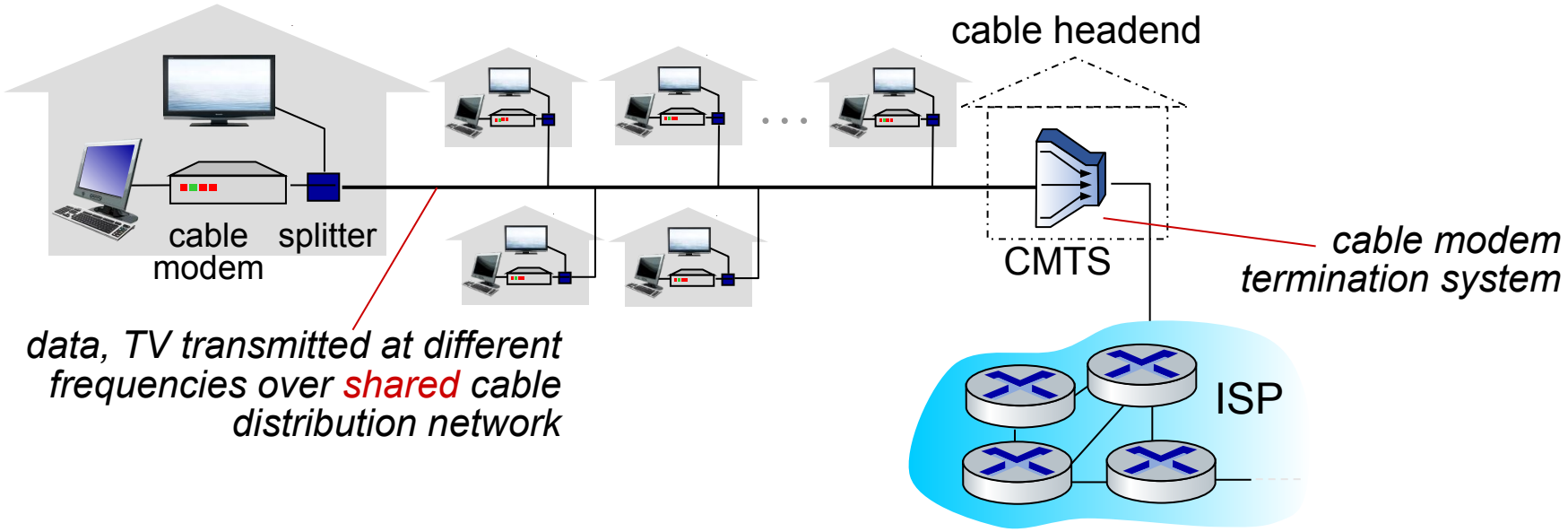
Network Edge: Access Networks: Cable-based access



Frequency division multiplexing (FDM): different channels transmitted in different frequency bands

COMPUTER NETWORKS

Network Edge: Access Networks: Cable-based access



Cable Internet access requires special modems, called cable modems. The cable modem is typically an external device and connects to the **home PC through an Ethernet port.**

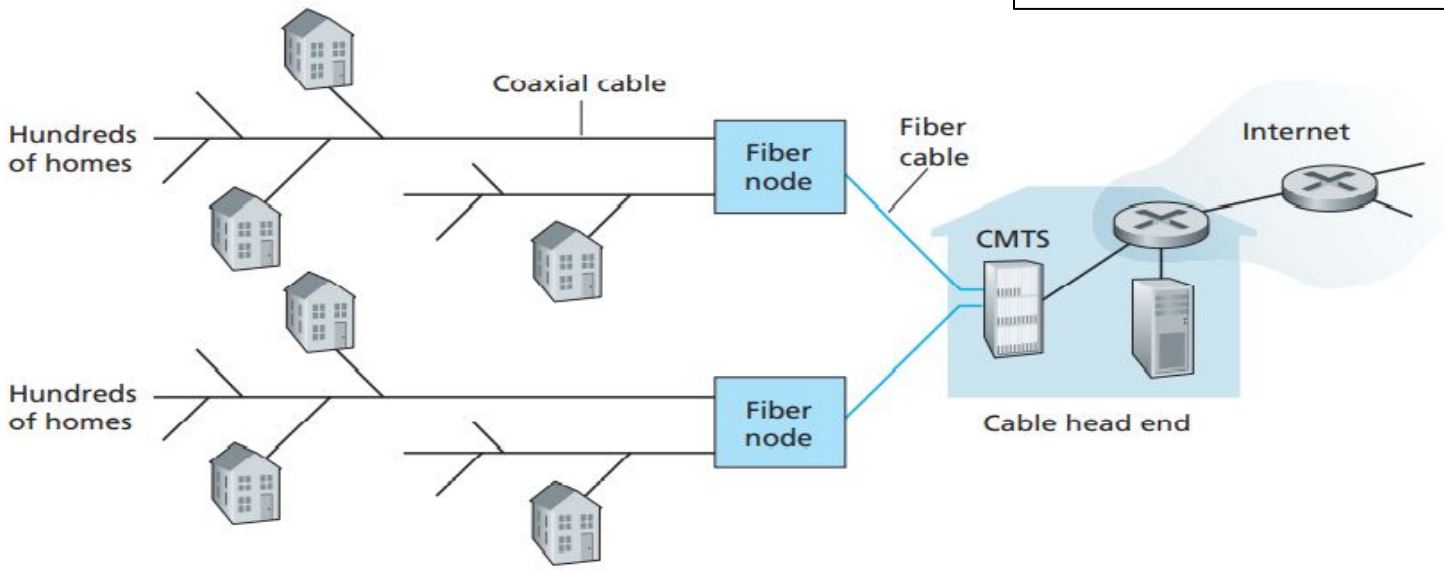
Access Network: Cable modem with HFC

- HFC: hybrid fiber coax
  - Asymmetric:

    up to 40 Mbps – 1.2 Gbps downstream transmission rate,

    30-100 Mbps upstream transmission rate

Symmetrical connections, a connection with equal download and upload speeds.
For example, with a 500/500 Mbps fiber internet connection you get 500 Mbps of download AND 500 Mbps of upload speeds.
An asymmetrical connection, however, does NOT have equal download/upload speeds.



Access Network: FTTH – new technology

Apart from DSL and Cable modem, an up-and-coming technology that provides even higher speeds is **fiber to the home (FTTH) .**

FTTH concept is simple—provide **an optical fiber path** from the CO directly to the home.

FTTH can potentially provide Internet access rates in the gigabits per second range.

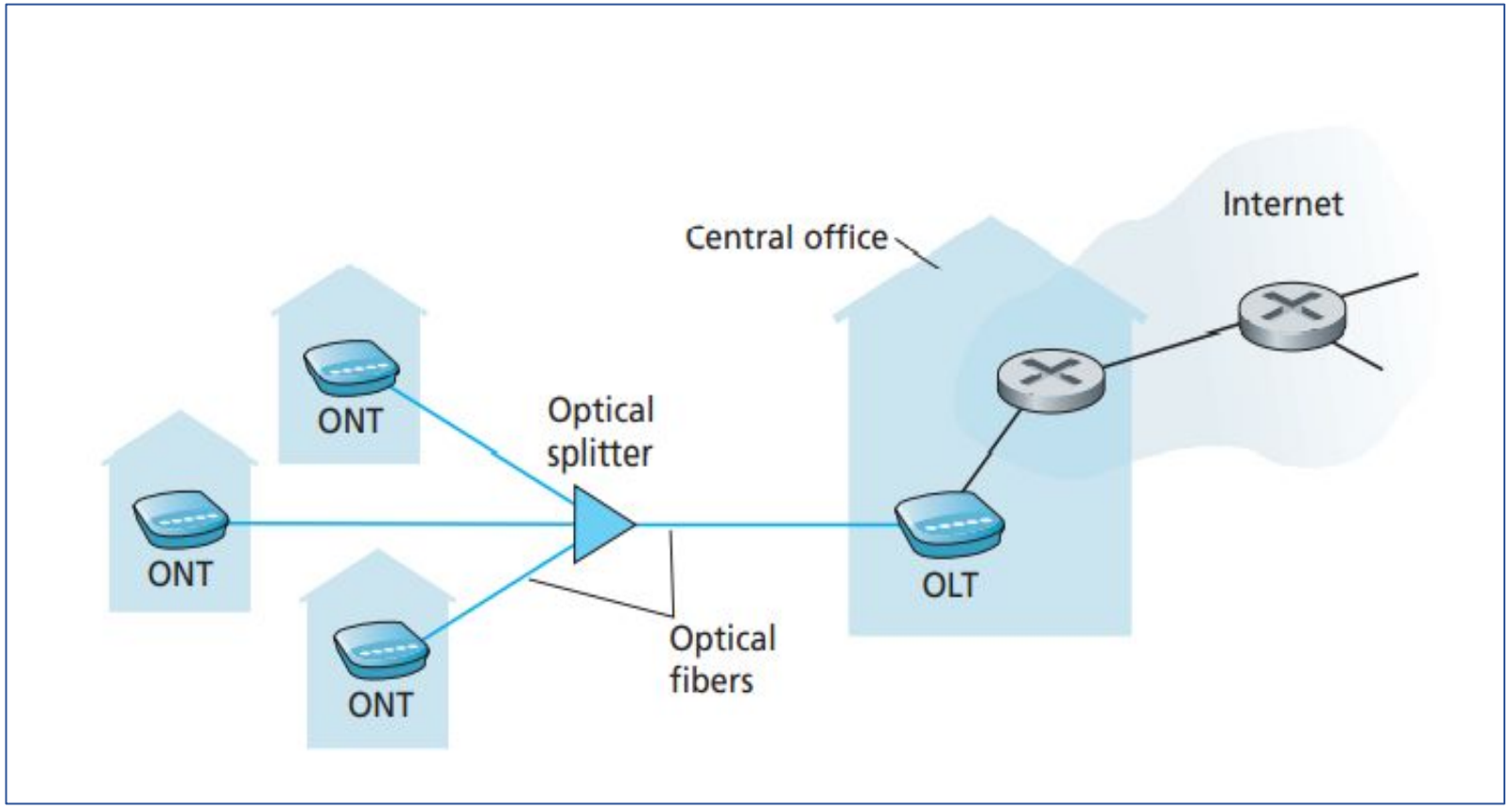
Each fiber leaving the central office is actually shared by many homes; it is not until the fiber gets relatively close to the homes that it is split into individual customer-specific fibers.

There are two competing optical-distribution network architectures that perform this splitting: **active optical networks** (AONs) and **passive optical networks** (PONs).

AON is switched Ethernet

Access Network: FTTH – new technology

FTTH using PON's architecture



In locations where DSL, cable, and FTTH are not available, a **satellite link** can be used to connect a residence.
StarBand and HughesNet are two such satellite access providers.

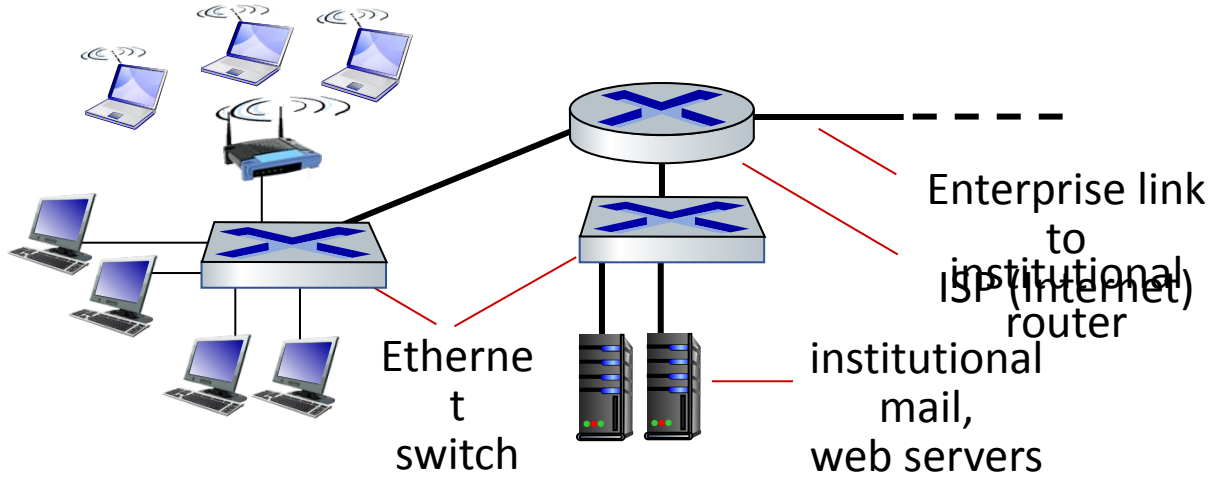
Dial-up access over traditional phone lines is based on the same model as DSL
Dial-up access is incredibly slow at 56 kbps.

Access Network: 5G fixed Wireless

- In addition to DSL, Cable, and FTTH, 5G fixed wireless is beginning to be deployed.
- 5G fixed wireless not only promises high-speed residential access, but will do so without installing costly and failure-prone cabling from the telco's CO to the home.
- With 5G fixed wireless, using beam-forming technology, data is sent wirelessly from a provider's base station to the a modem in the home.

Beamforming is a type of radio frequency (RF) management in which a wireless signal is directed toward a specific receiving device.

Network Edge: Access Networks – Enterprise networks



- **Ethernet** is by far the most prevalent LAN technology in corporate, university, and home networks.
- Ethernet users use **twisted-pair copper wire** to connect to an Ethernet switch.

- companies, universities, schools etc.
- mix of **wired, wireless link technologies**, connecting a mix of switches and routers (we'll cover differences shortly)
  - **Ethernet - wired access:** users typically have 100 Mbps or 1 Gbps access to the Ethernet switch, whereas **servers may have 1 Gbps or even 10 Gbps access.**

Network Edge: Access Networks – Enterprise networks

In a wireless LAN setting, wireless users transmit/receive packets to/from an **access point** that is connected into the enterprise's network

A wireless LAN user must typically be within a few tens of meters of the access point. Wireless LAN access based on **IEEE 802.11 technology**, more informally known as **WiFi.**

Ethernet and WiFi access networks were initially deployed in enterprise (corporate, university) settings, they have recently become relatively common components of **home networks** too.

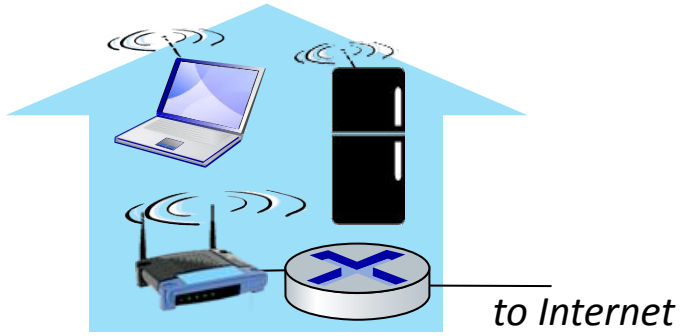
Network Edge: Wireless Access Networks

Shared *wireless* access network connects end system to router

- via base station aka “access point”

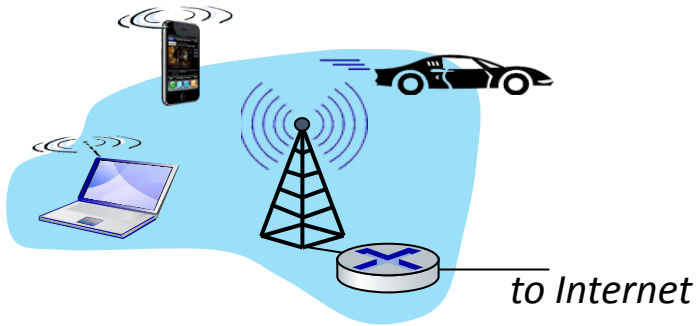
Wireless local area networks (WLANs)

- typically within or around building (~100 ft)
- 802.11b/g/n (WiFi): 11, 54, 450 Mbps transmission rate

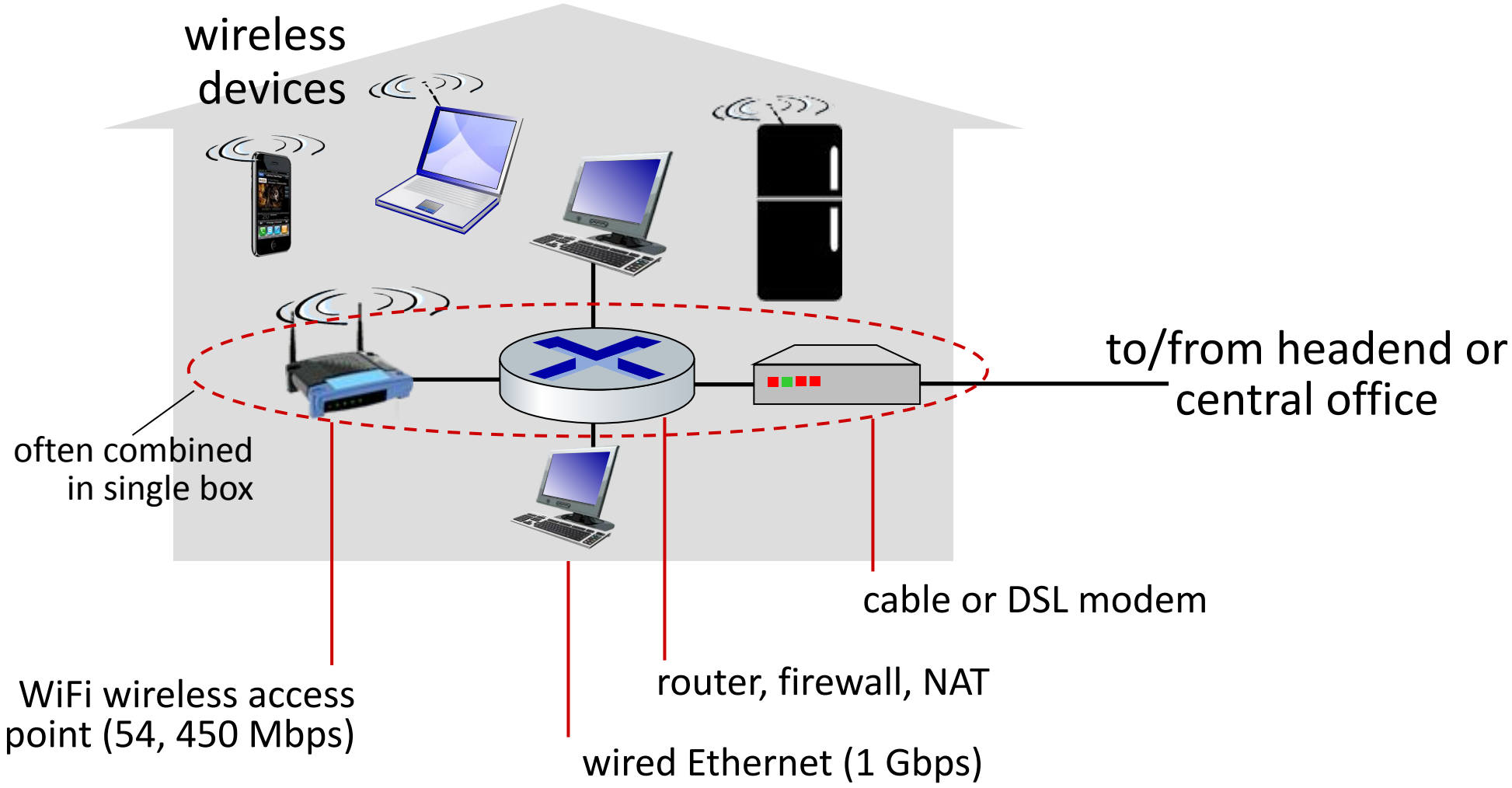


Wide-area cellular access networks

- provided by mobile, cellular network operator (10’s km)
- 10’s Mbps
- 4G cellular networks and 5G cellular networks



Network Edge: Access Networks – Home access



PHYSICAL MEDIA

COMPUTER NETWORKS

Physical medium: Introduction

- **Bit,** when traveling from source to destination, passes through a series of transmitter-receiver pairs.
- For each transmitter receiver pair, the bit is sent by propagating **electromagnetic waves or optical pulses** across a physical medium.

- **The physical medium can take many shapes** and forms and does not have to be of the same type for each transmitter-receiver pair along the path.
- Examples of physical media include
  - **Twisted-pair copper wire**
  - **Coaxial cable**
  - **Multimode fiber-optic cable**
  - **Terrestrial radio spectrum**
  - **Satellite radio spectrum**

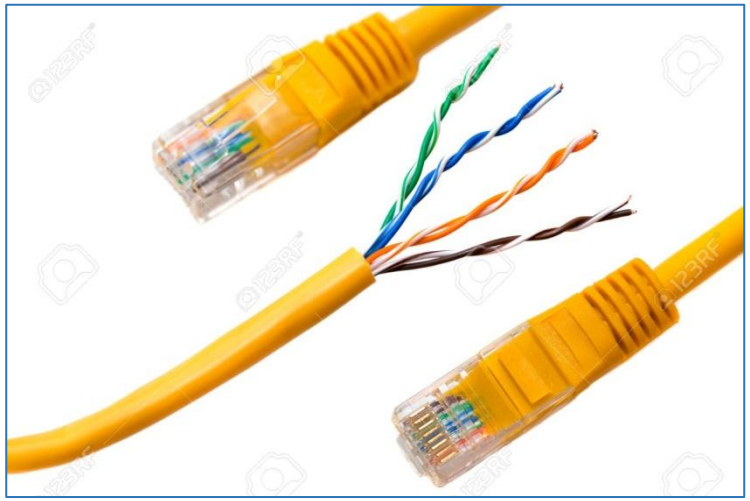
Network Edge: Physical media

- Physical media fall into two categories:
  - **Guided media**
  - **Unguided media.**
- With guided media, the waves are guided along a **solid medium**, such as a fiber-optic cable, a twisted-pair copper wire, or a coaxial cable.
- With unguided media, the **waves propagate in the atmosphere** and in outer space, such as in a wireless LAN or a digital satellite channel.

COMPUTER NETWORKS

Physical Media – Twisted Pair Copper Wire

- Twisted pair consists of **two insulated copper wires**, each about **1 mm thick**, arranged in a regular spiral pattern.
- A wire pair constitutes a single communication link.
- **Unshielded twisted pair (UTP)** is commonly used for computer networks within a building, that is, for LANs.
- Data rates for LANs - range from **10 Mbps to 10 Gbps.**



Physical Media – Twisted Pair Copper Wire

The data rates that can be achieved **depend on the thickness** of the wire and the **distance** between transmitter and receiver.

Reason for Twisting

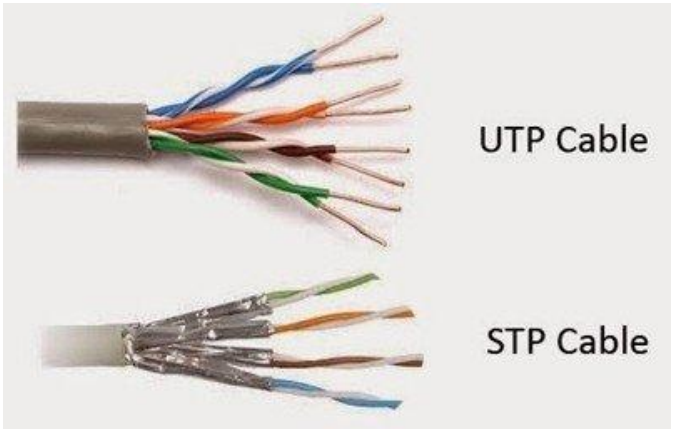
External waves(noise, interferences, and cross talks) cancel out due to the different twists.

Applications of Twisted-Pair Cables

- In telephone lines
- In DSL lines
- In LANs

Types

- **Unshielded Twisted Pair (UTP):** These generally comprise of wires and insulators.
- **Shielded Twisted Pair (STP):** They have a braided wired mesh that encases each pair of insulated wires.



Physical Media – Twisted Pair Copper Wire

Categories of Twisted–Pair Cables

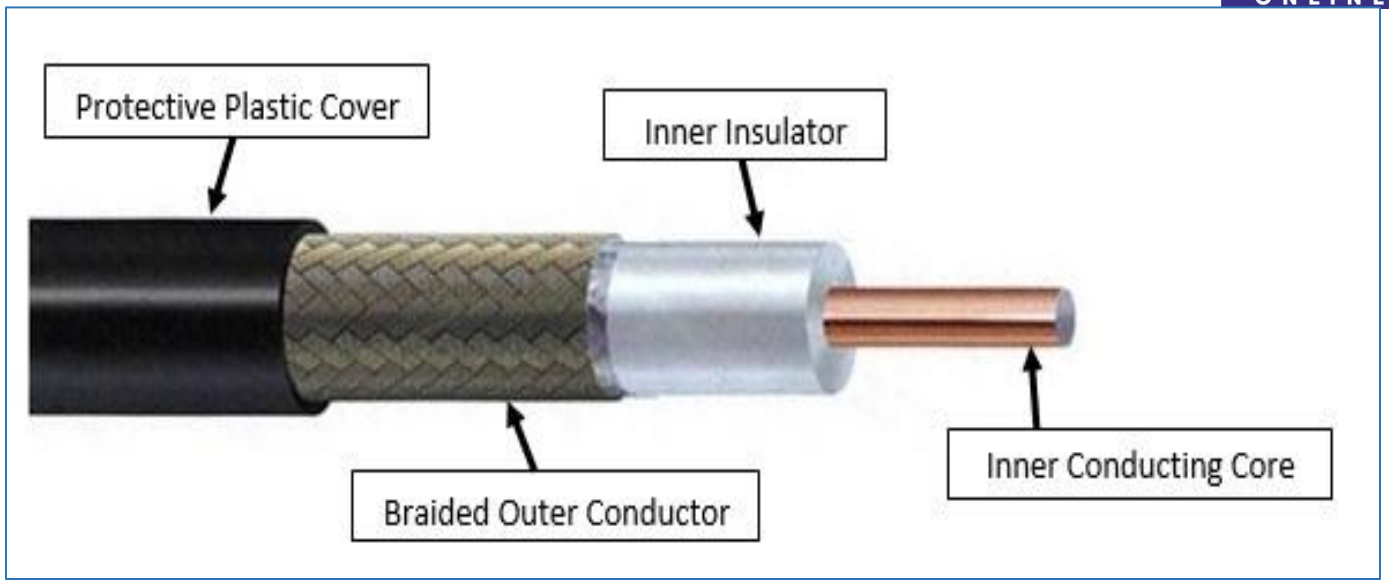
- Category 1 – UTP used in telephone lines with data rate < 0.1 Mbps
- Category 2 – UTP used in transmission lines with a data rate of 2 Mbps
- Category 3 – UTP used in LANs with a data rate of 10 Mbps
- Category 4 – UTP used in Token Ring networks with a data rate of 20 Mbps
- Category 5 – UTP used in LANs with a data rate of 100 Mbps
- Category 6 – UTP used in LANs with a data rate of 200 Mbps
- Category 7 – STP used in LANs with a data rate of 10 Mbps

COMPUTER NETWORKS

Physical media – Coaxial Cable

Coaxial cable:

- Two copper conductors
- Concentric rather than parallel
- bidirectional
- broadband:
  - multiple frequency channels on cable
  - 100’s Mbps per channel



Physical media – Coaxial Cable

- Coaxial cable comes in two varieties:
  - **Baseband coaxial cable**
  - **Broadband coaxial cable**.

Baseband coaxial cable, also called **50-ohm cable,** is about a centimeter thick, lightweight, and easy to bend.

Broadband coaxial cable, also called 75-ohm cable, is quite a bit thicker, heavier, and stiffer than the baseband variety.

Physical media – Coaxial Cable

Categories of Coaxial Cables

Coaxial cables are categorized into three types as per radio government (RG) ratings –

- RG – 59: Has impedance of 75W and used in cable TV
- RG – 58: Has impedance of 50W and used in thin Ethernet
- RG – 11: Has impedance of 50W and used in thick Ethernet

Applications of Coaxial Cables

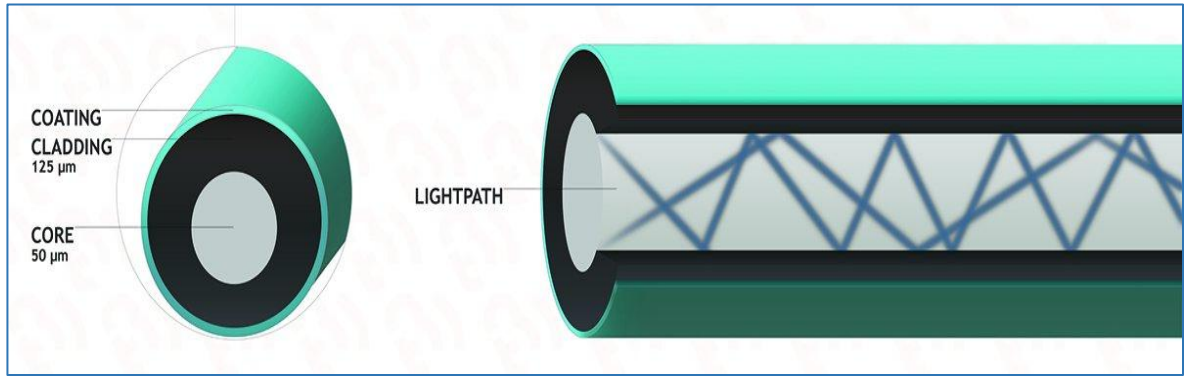
- In analog telephone networks: A single coaxial network can carry about 10,000 voice signals.
- In digital telephone networks: A coax has a data rate of 600 Mbps.
- In cable TV networks
- In traditional Ethernet LANs
- In MANs

COMPUTER NETWORKS

Physical media – Fiber optic cable

- Thin, flexible medium that conducts **pulses of light**, with each pulse representing a bit.
- A single optical fiber - **tremendous bit rates**, up to tens or even hundreds of gigabits per second.
- **Immune** to electromagnetic interference.
- Mostly for **overseas links.**
- **High cost.**
- The Optical Carrier (OC) standard link speeds range from **51.8 Mbps to 39.8 Gbps.**

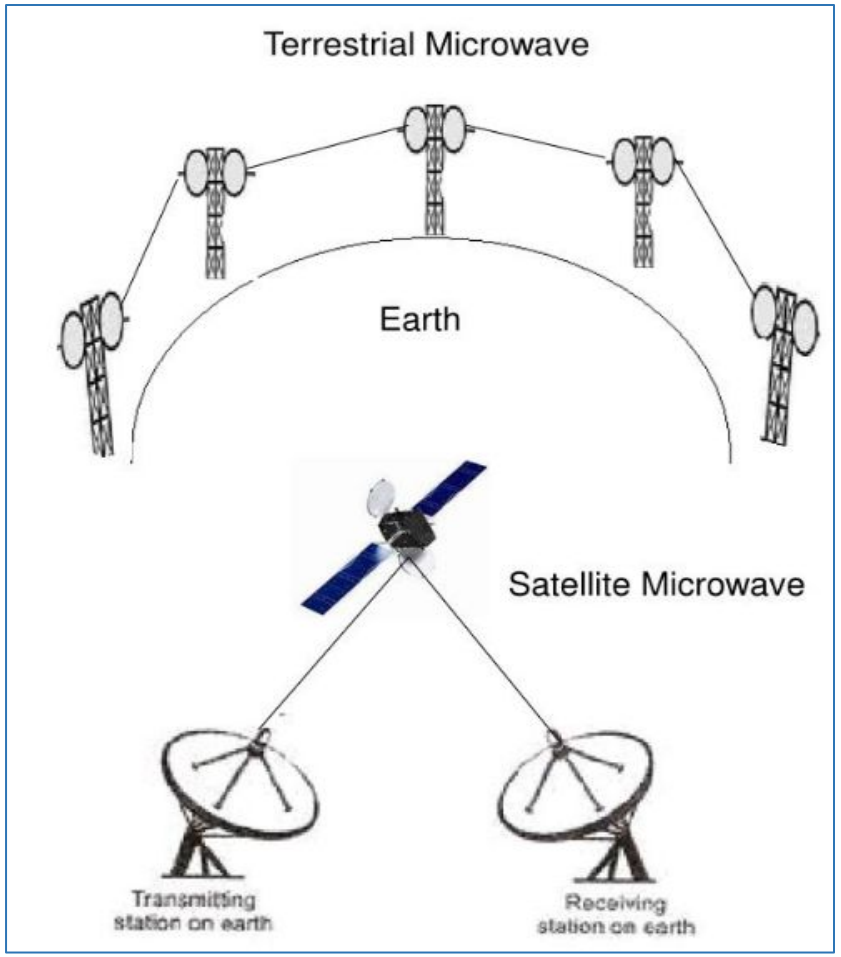
Standards in use today include OC-1, OC-3, OC-12, OC-24, OC-48, OC-96, OC-192, OC-768



Physical media – Terrestrial Radio Channel

Wireless radio

- signal carried in electromagnetic spectrum
- no physical “wire”
- broadcast and “half-duplex” (sender to receiver)
- propagation environment effects:
  - reflection
  - obstruction by objects
  - interference



COMPUTER NETWORKS

Physical media – Terrestrial Radio Channel

- Terrestrial radio channels can be broadly classified into three groups:
    - short distance (e.g., with one or two meters);
    - local areas- ten to a few hundred meters
    - wide area - tens of kilometers.

Radio link types:

- terrestrial microwave
    - up to 45 Mbps channels
- Wireless LAN (WiFi)
    - Up to 100's Mbps
- wide-area (e.g., cellular)
    - 4G cellular: ~ 10's Mbps

COMPUTER NETWORKS

Physical Media: Satellite radio channels

Satellite links two or more Earth-based microwave transmitter/ receivers, known as ground stations.

- **Two types of satellites** are used in communications:
  - **Geostationary satellites.**
  - **Low-earth orbiting** (LEO) satellites

- **Geostationary satellites** permanently remain above the same spot on Earth.
- This stationary presence is achieved by placing the satellite in orbit at 36,000 kilometers above Earth's surface.
- Signal propagation delay - 280 milliseconds.

- **LEO satellites** are placed much closer to Earth and do not remain permanently above one spot on Earth.
- They rotate around Earth (just as the Moon does) and may communicate with each other, as well as with ground stations.
- To provide continuous coverage to an area, many satellites need to be placed in orbit.

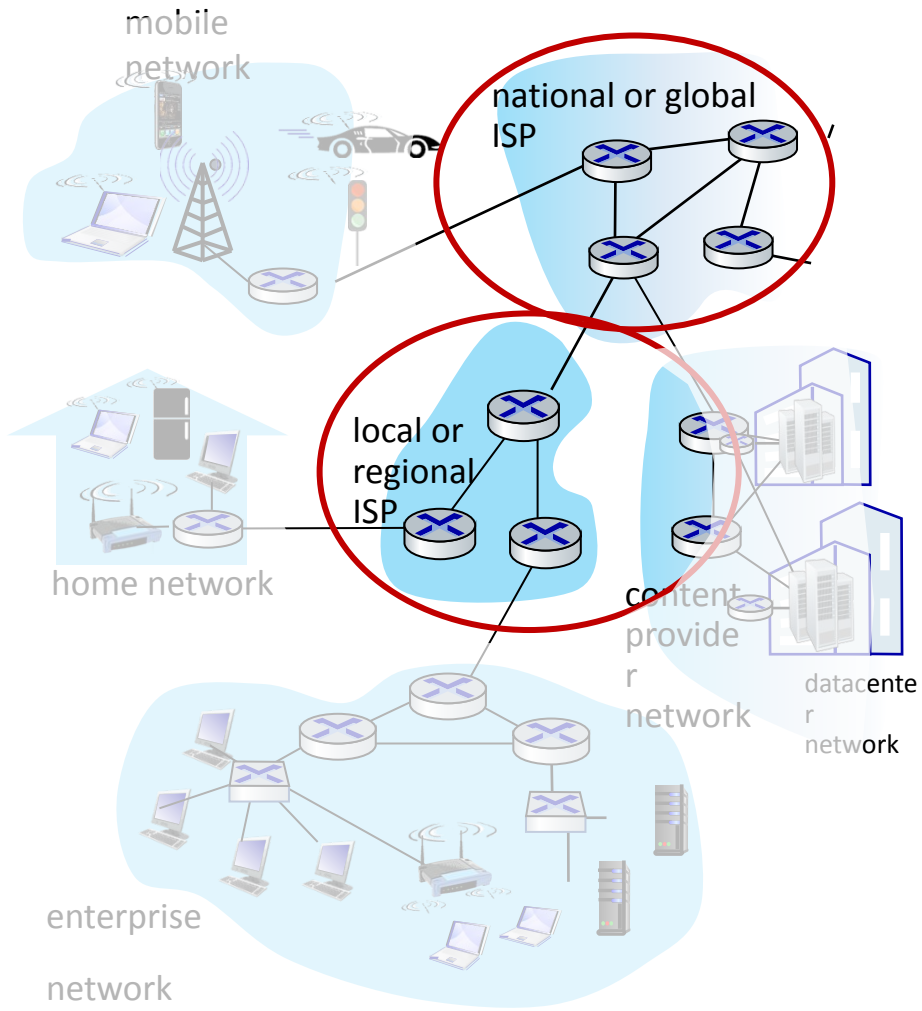
Propagation Delay = Distance / Propagation Speed

NETWORK CORE

COMPUTER NETWORKS

Network Core: Packet Switching

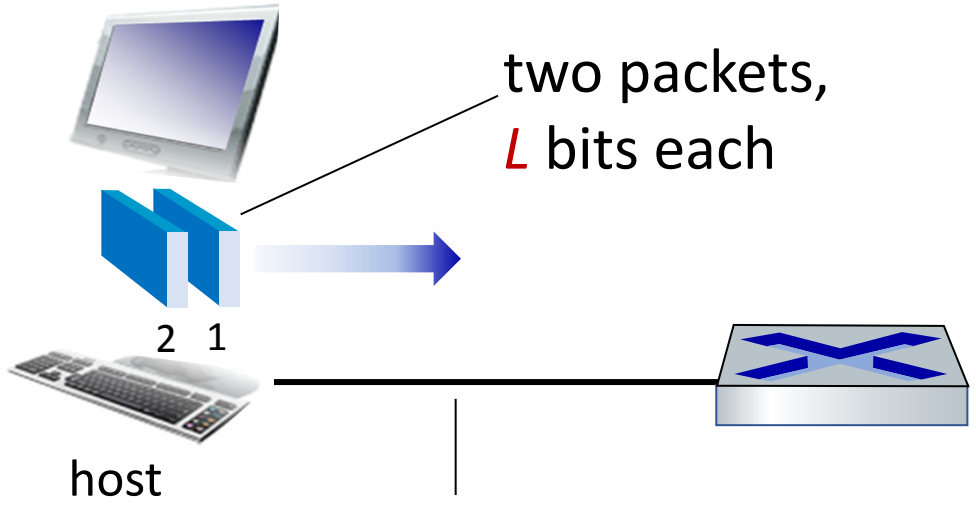
- mesh of interconnected routers
- packet-switching: hosts break application-layer messages into *packets*
  - forward packets from one router to the next, across links on path from source to destination
  - each packet transmitted at full link capacity



Hosts: Send packets of data

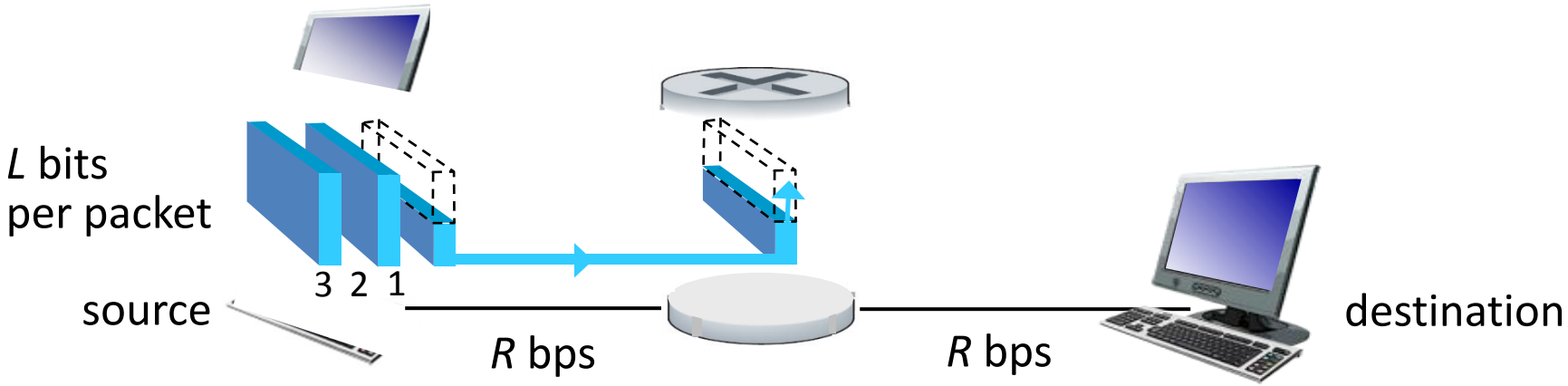
Host sending function:

- 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*



$$\text{packet transmission delay} = \text{time needed to transmit } L\text{-bit packet into link} = \frac{L \text{ (bits)}}{R \text{ (bits/sec)}}$$

Network Core: Packet Switching: store-and-forward

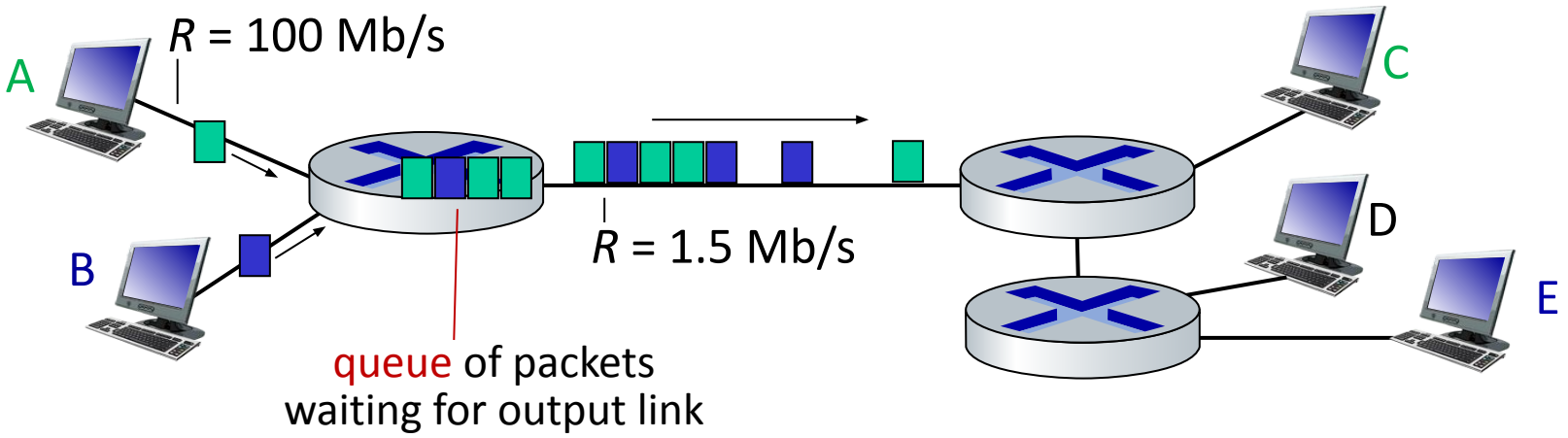


- Transmission delay: takes L/R seconds to transmit (push out) L-bit packet into link at R bps
- *Store and forward:* *entire* packet must arrive at router before it can be transmitted on next link
- *End-end delay:* $2L/R$ (above), assuming zero propagation delay (more on delay shortly)

One-hop numerical example:

- L = 10 Kbits
- R = 100 Mbps
- one-hop transmission delay = 0.1 msec

Network Core: Packet Switching: queuing delay, loss



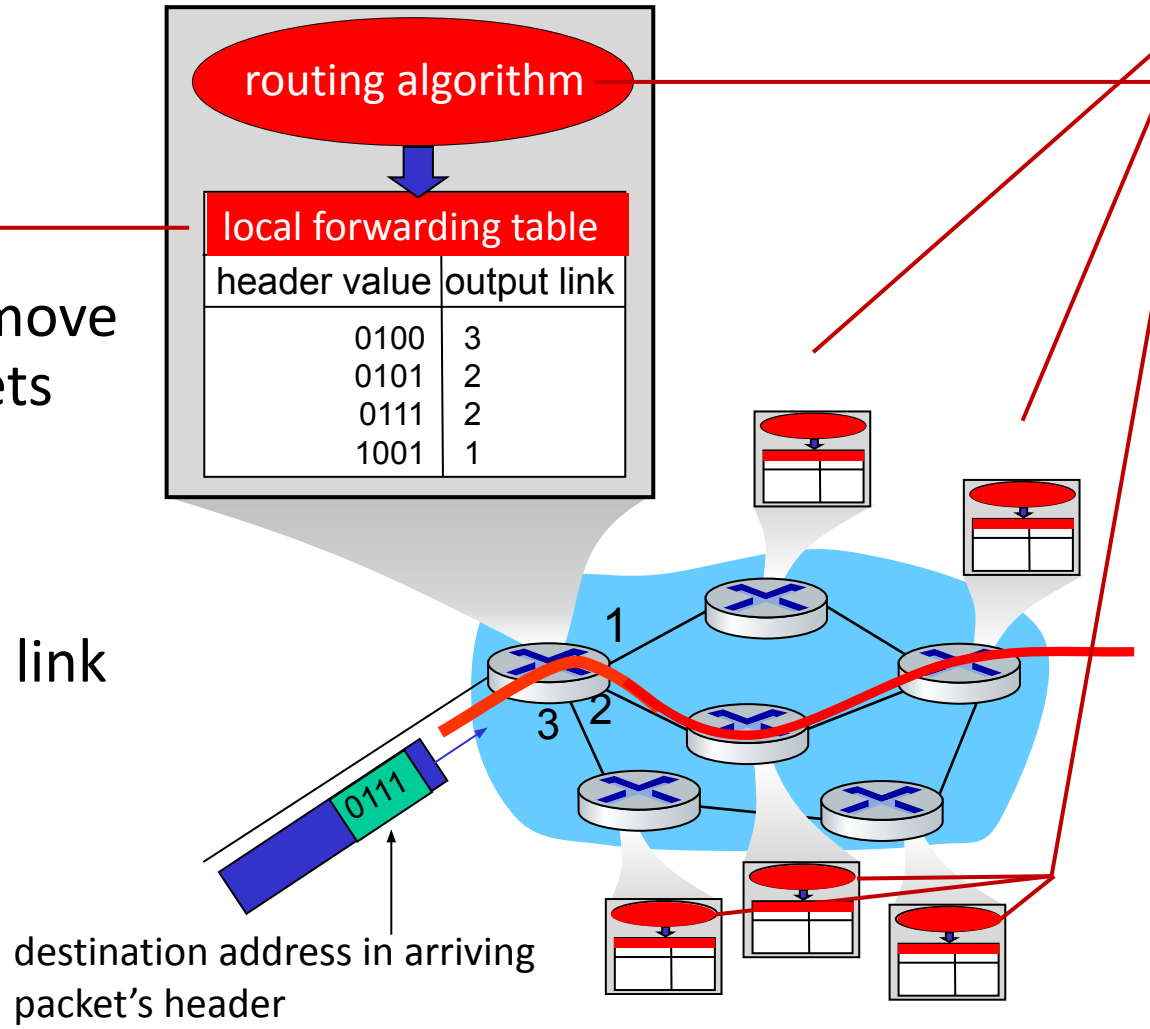
Packet queuing and loss: if arrival rate (in bps) to link exceeds transmission rate (bps) of link for a period of time:

- packets will queue, waiting to be transmitted on output link
- packets can be dropped (lost) if memory (buffer) in router fills up

Network Core: Two Key Network Core Functions

Forwarding:

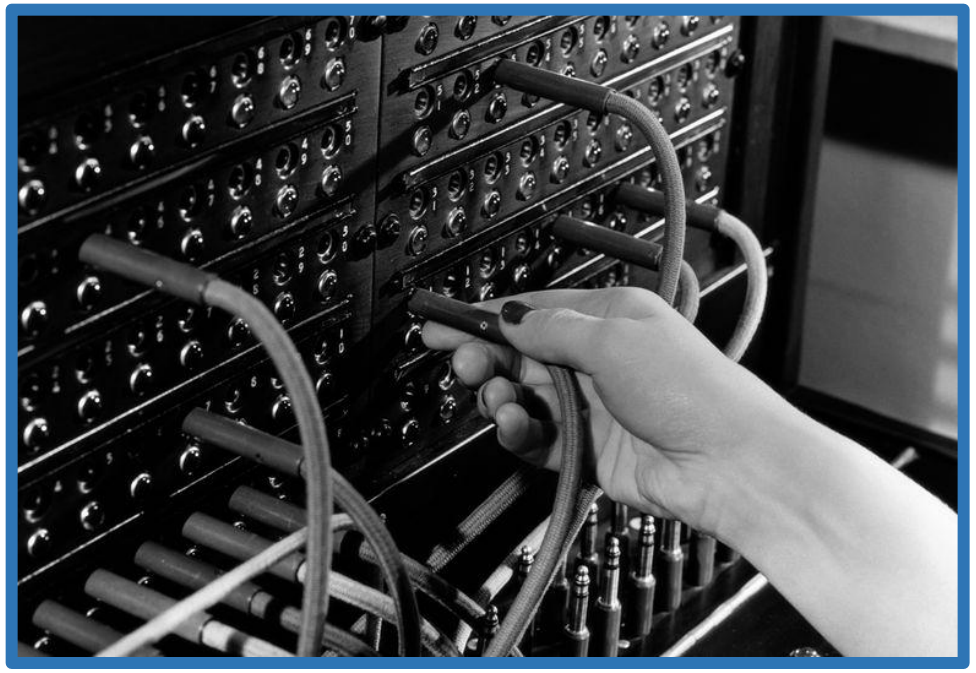
- *local* action: move arriving packets from router's input link to appropriate router output link

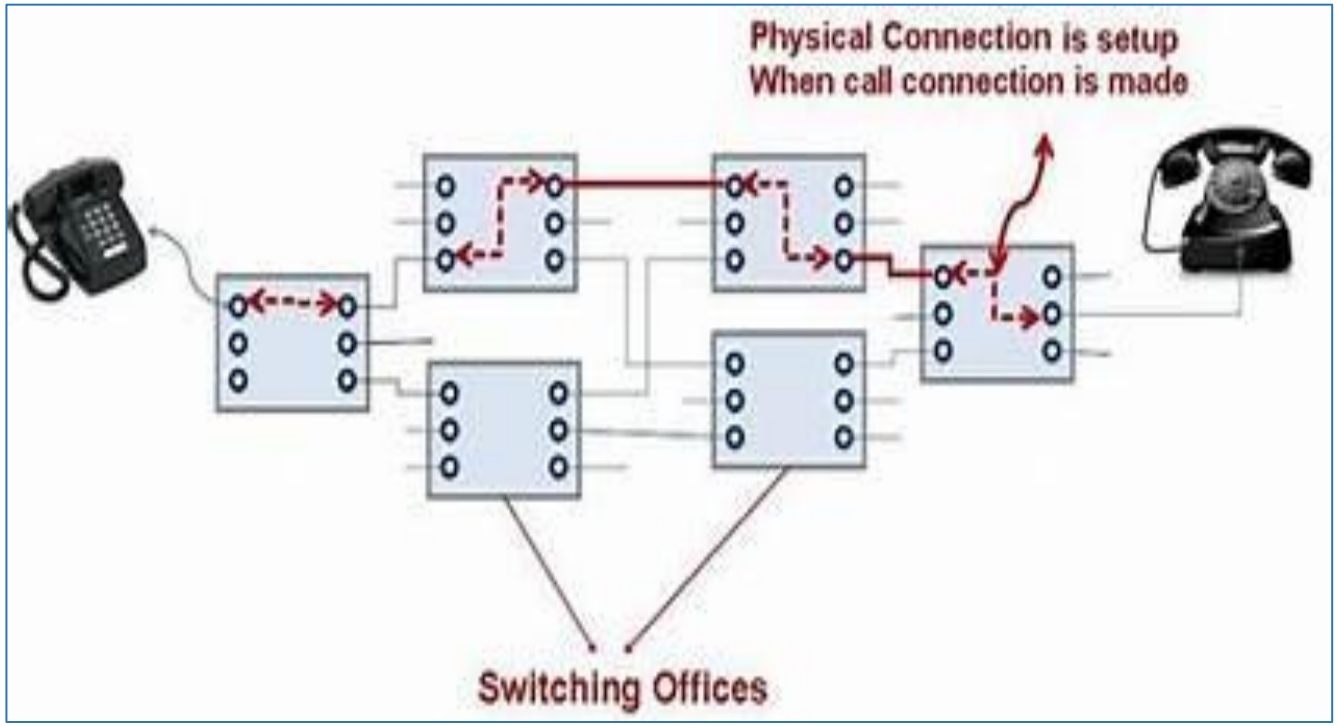


Routing:

- *global* action: determine source-destination paths taken by packets
- routing algorithms

CIRCUIT SWITCHING NETWORKS

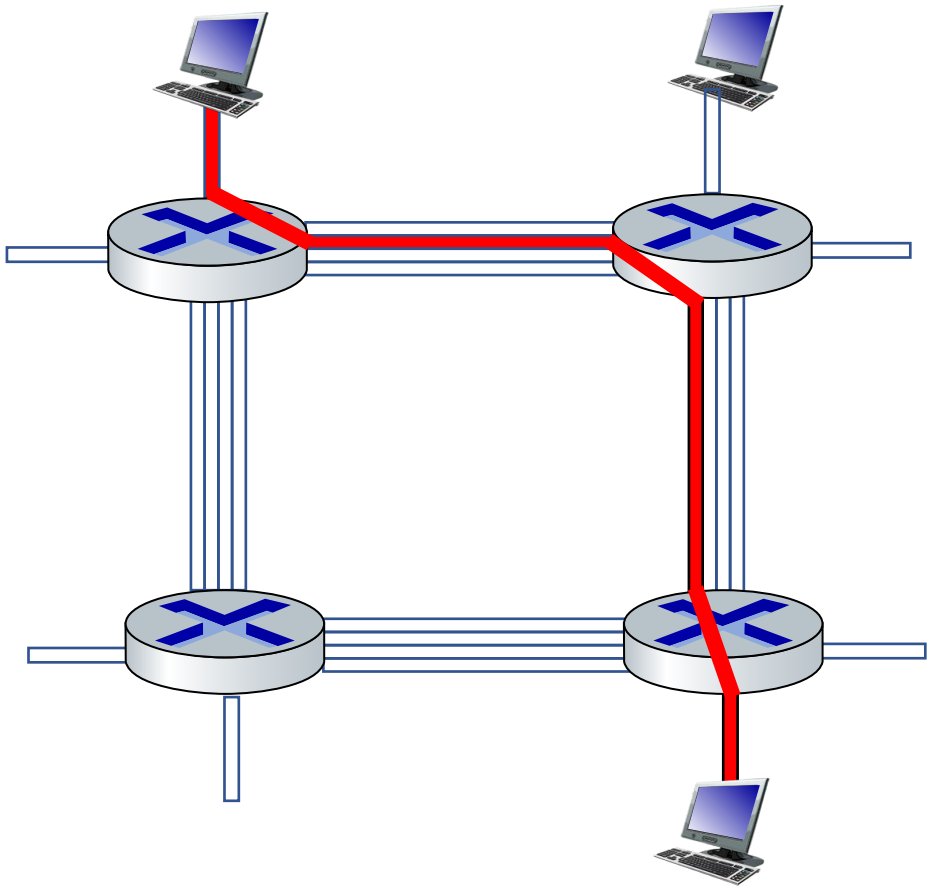




Network Core: Circuit Switching

end-end resources allocated to, reserved for "call" between source and destination (eg: telephone)

- in diagram, each link has four circuits.
  - call gets 2nd circuit in top link and 1st circuit in right link.
- dedicated resources: no sharing
  - circuit-like (guaranteed) performance
- circuit segment idle if not used by call (no sharing)
- commonly used in traditional telephone networks



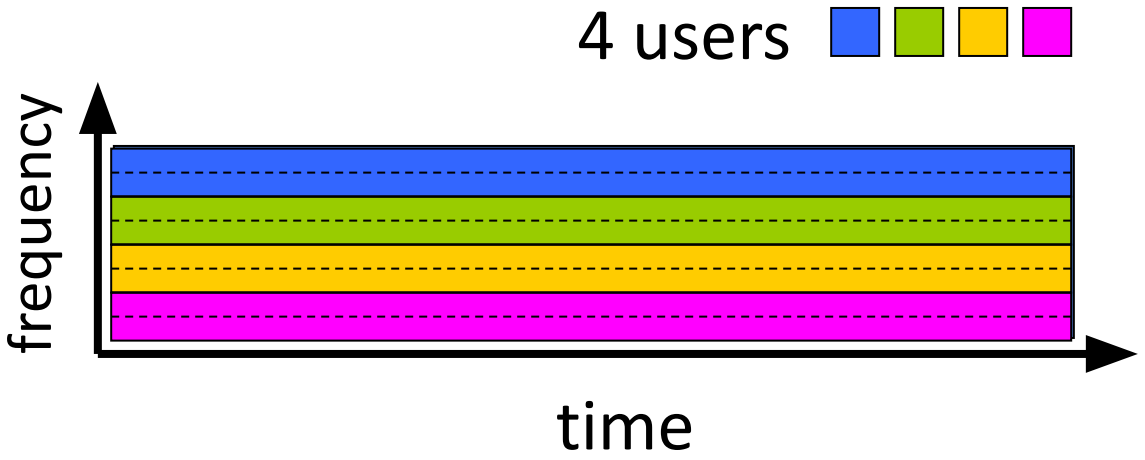
Multiplexing in Circuit Switched Networks: FDM & TDM

- **A circuit** in a link is implemented with either
  - **Frequency-division multiplexing (FDM) or**
  - **Time-division multiplexing (TDM)**

Multiplexing in Circuit Switched Networks: FDM & TDM

Frequency Division Multiplexing (FDM)

- The frequency spectrum of a link is divided up among the connections established across the link.
- optical, electromagnetic frequencies divided into (narrow) frequency bands
- each call allocated its own band, can transmit at max rate of that narrow band.

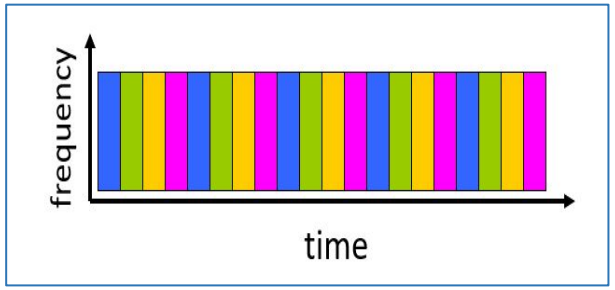


- In **telephone networks,** this frequency band typically has a **width of 4 kHz.**
- **FM radio stations** also use FDM to share the frequency spectrum between **88 MHz and 108 MHz,** with each station being allocated a specific frequency band.

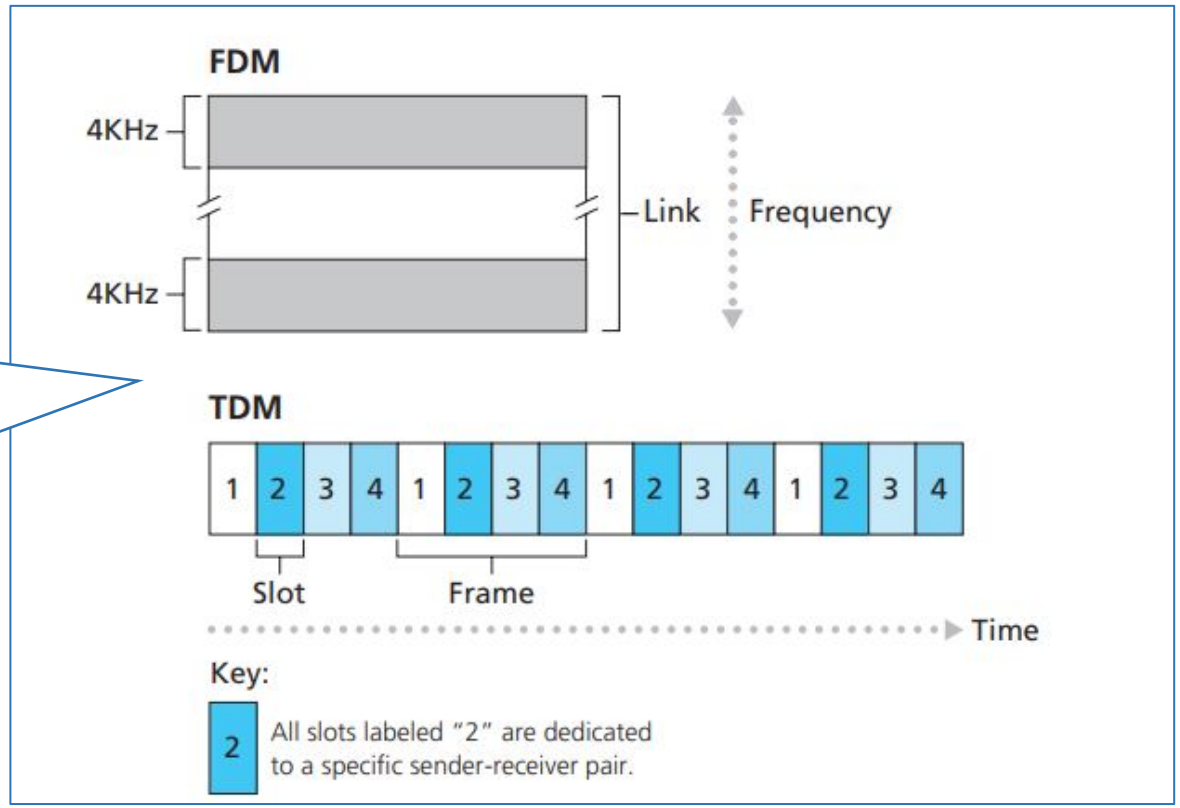
Multiplexing in Circuit Switched Networks: FDM & TDM

Time Division Multiplexing (TDM)

- time divided into frames -> slots
- each call allocated periodic slot(s), can transmit at maximum rate of (wider) frequency band, but only during its time slot(s)



With FDM, each circuit continuously gets a fraction of the bandwidth.
With TDM, each circuit gets all of the bandwidth periodically during brief intervals of time (that is, during slots), each circuit slots



Network Core: Packet Switching vs Circuit Switching

- Connectionless/ Connection oriented
- Designed for data
- Flexible
- Out of order, assembled at the destination - datagrams
- Store & Forward
- Network layer
- Bandwidth is saved (dynamic)
- Transmission of data – Source, routers
- Transmission delay

- Connection oriented
- Designed for voice
- Inflexible
- Message received in same order
- FDM & TDM
- Physical layer
- Bandwidth is wasted (fixed)
- Transmission of data – source
- Call setup delay

Suppose a link transmits data @ 1Mbps:

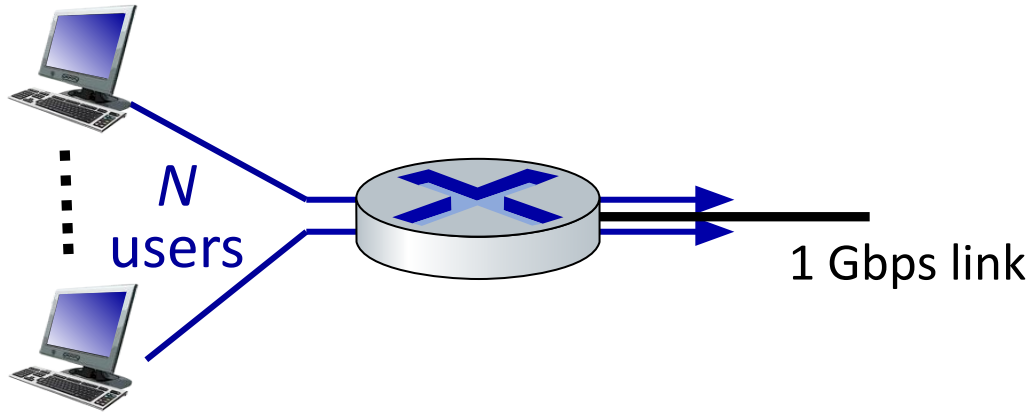
Consider a scenario where every user remains active for 10% of the time and transmits data @100kbps and the rest 90% time the users remain inactive(do not transmit data).

Network Core: Packet Switching vs Circuit Switching

packet switching allows more users to use network!

Example:

- 1 Gb/s link
- each user:
  - 100 Mb/s when "active"
  - active 10% of time



- *circuit-switching:* 10 users
- *packet switching:* with 35 users,
- probability > 10 active users at same time is less than .0004 *
- 10 or few active users, probability 0.9996

Q: how did we get value 0.0004?

Q: what happens if > 35 users ?

* Check out the online interactive exercises for more examples: http://gaia.cs.umass.edu/kurose_ross/interactive

Network Core: Packet Switching vs Circuit Switching

Circuit Switching:

- For circuit switching, the number of users that can be supported simultaneously on a link is determined by the link capacity divided by the individual user capacity.
- Link Capacity = 1 Mbps
- Individual User Capacity = 100 kbps
- Number of users = Link Capacity / Individual User Capacity

$$\text{Number of users} = \frac{1\ \text{Mbps}}{100\ \text{kbps}} = \frac{1{,}000{,}000\ \text{bits per second}}{100{,}000\ \text{bits per second}} = 10\ \text{users}$$

Therefore, in a circuit-switched scenario, only 10 users can be supported simultaneously due to the dedicated allocation of 100 kbps for each user at all times.

* Check out the online interactive exercises for more examples: http://gaia.cs.umass.edu/kurose_ross/interactive

COMPUTER NETWORKS

Network Core: Packet Switching vs Circuit Switching

Packet Switching:

- With packet switching, the probability of a specific user being active is 0.1 (10% active).
- The probability of having 11 or more simultaneously active users out of 35 users can be calculated using a binomial probability formula.

$$P(X \geq 11) = 1 - P(X \leq 10)$$

Where:

- $P(X \geq 11)$ = Probability of having 11 or more active users simultaneously.
- $P(X \leq 10)$ = Probability of having 10 or fewer active users simultaneously.

Given:

- Total Users = 35
- Probability of a specific user being active = 0.1 (10%)

Using the binomial probability formula:

$$P(X \geq 11) = 1 - P(X \leq 10) = 1 - \sum_{k=0}^{10} \binom{35}{k} \times (0.1)^k \times (0.9)^{35-k}$$

This calculation shows that the probability ↓ aving 11 or more simultaneously active users is approximately 0.0004. Hence, the probability of having 10 or fewer active users is 0.9996 (1 - 0.0004), which is very high.

Therefore, based on this calculation, in the majority of cases (with a probability of 0.9996), the aggregate arrival rate of data remains below the link's output rate of 1 Mbps in a packet-switched network, allowing for efficient data transmission without delays, similar to circuit switching. When more than 10 users are active simultaneously (with a very low probability of 0.0004), an output queue might form temporarily until the input rate falls back below the output rate.

* Check out the online interactive exercises for more examples: http://gaia.cs.umass.edu/kurose_ross/interactive

Network Core: Packet Switching vs Circuit Switching

Is packet switching a “slam dunk winner”?

- great for “bursty” data – sometimes has data to send, but at other times not
  - resource sharing
  - simpler, no call setup
- excessive congestion possible: packet delay and loss due to buffer overflow
  - protocols needed for reliable data transfer, congestion control
- *Q:* How to provide circuit-like behavior?
  - bandwidth guarantees traditionally used for audio/video applications

Q: human analogies of reserved resources (circuit switching) versus on-demand allocation (packet switching)?

Packet Switching vs Circuit Switching – Numerical Example

- How long does it take to send a file of 640,000 bits (1 byte = 8 bits) from host A to host B over a circuit-switched network?
  - All links are 1.536 Mbps
  - Each link uses TDM with 24 slots/sec
  - 500 msec to establish end-to-end circuit

Let's work it out!

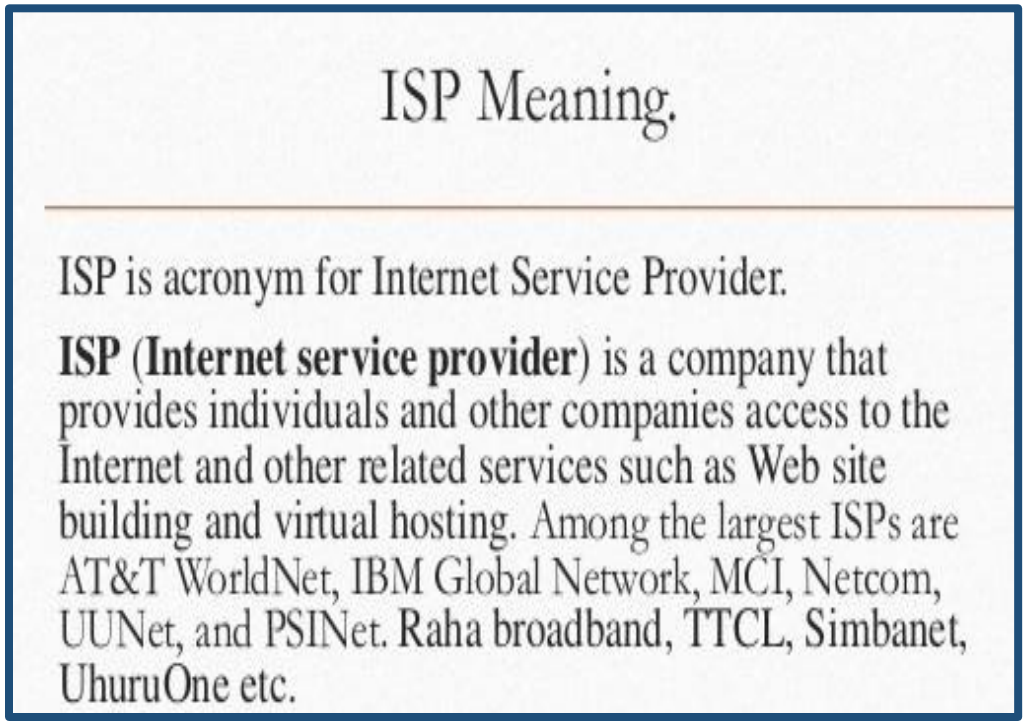
Solution:

- Each circuit has a transmission rate of (1.536 Mbps)/24 = 64 kbps
- It takes (640,000 bits)/(64 kbps) = 10 seconds to transmit the file
- To this 10 seconds we add the circuit establishment time, giving 10.5 seconds to send the file

NETWORK OF NETWORKS

Internet Structure: a "network of networks"

- End systems connect to Internet via access ISPs (Internet Service Providers)
  - Residential, company and university ISPs
- Access ISPs in turn must be interconnected.
  - So that any two hosts can send packets to each other
- Resulting network of networks is very complex
  - Evolution was driven by economics and national policies
- Let's take a stepwise approach to describe current Internet structure



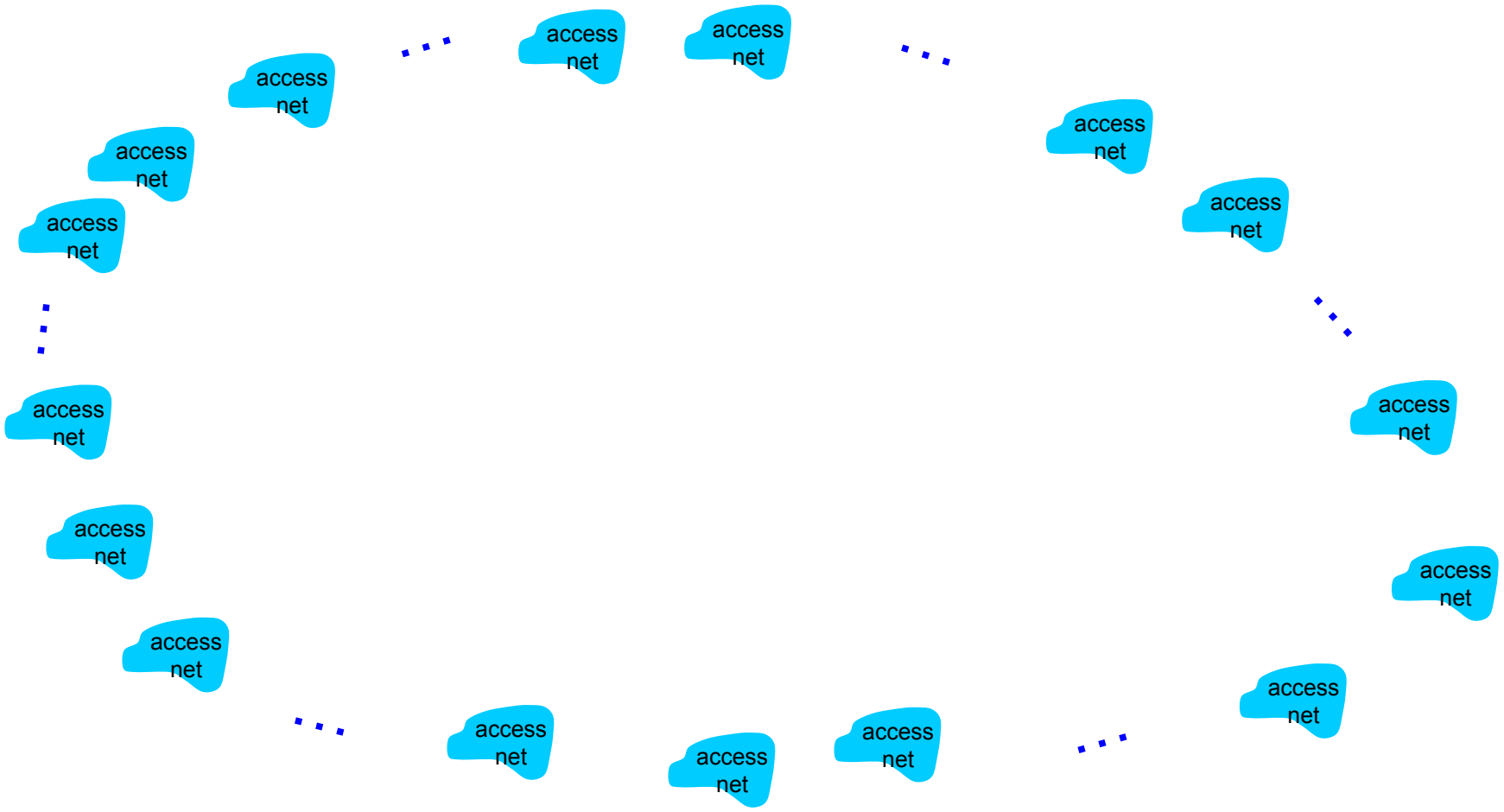
ISP Meaning.

ISP is acronym for Internet Service Provider.

ISP (Internet service provider) is a company that provides individuals and other companies access to the Internet and other related services such as Web site building and virtual hosting. Among the largest ISPs are AT&T WorldNet, IBM Global Network, MCI, Netcom, UUNet, and PSINet. Raha broadband, TTCL, Simbanet, UhuruOne etc.

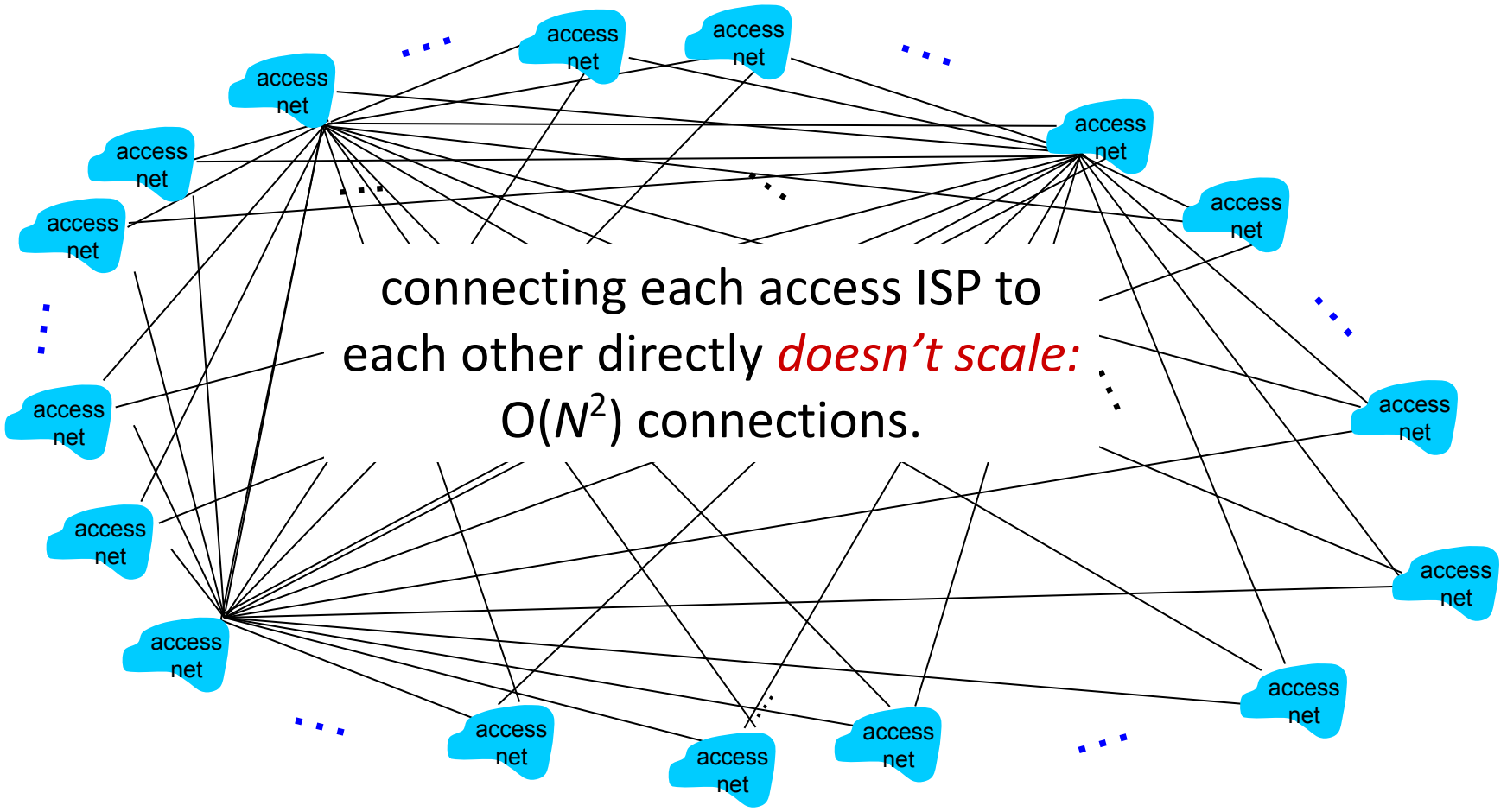
Internet Structure: a “network of networks”

Question: given *millions* of access ISPs, how to connect them together?



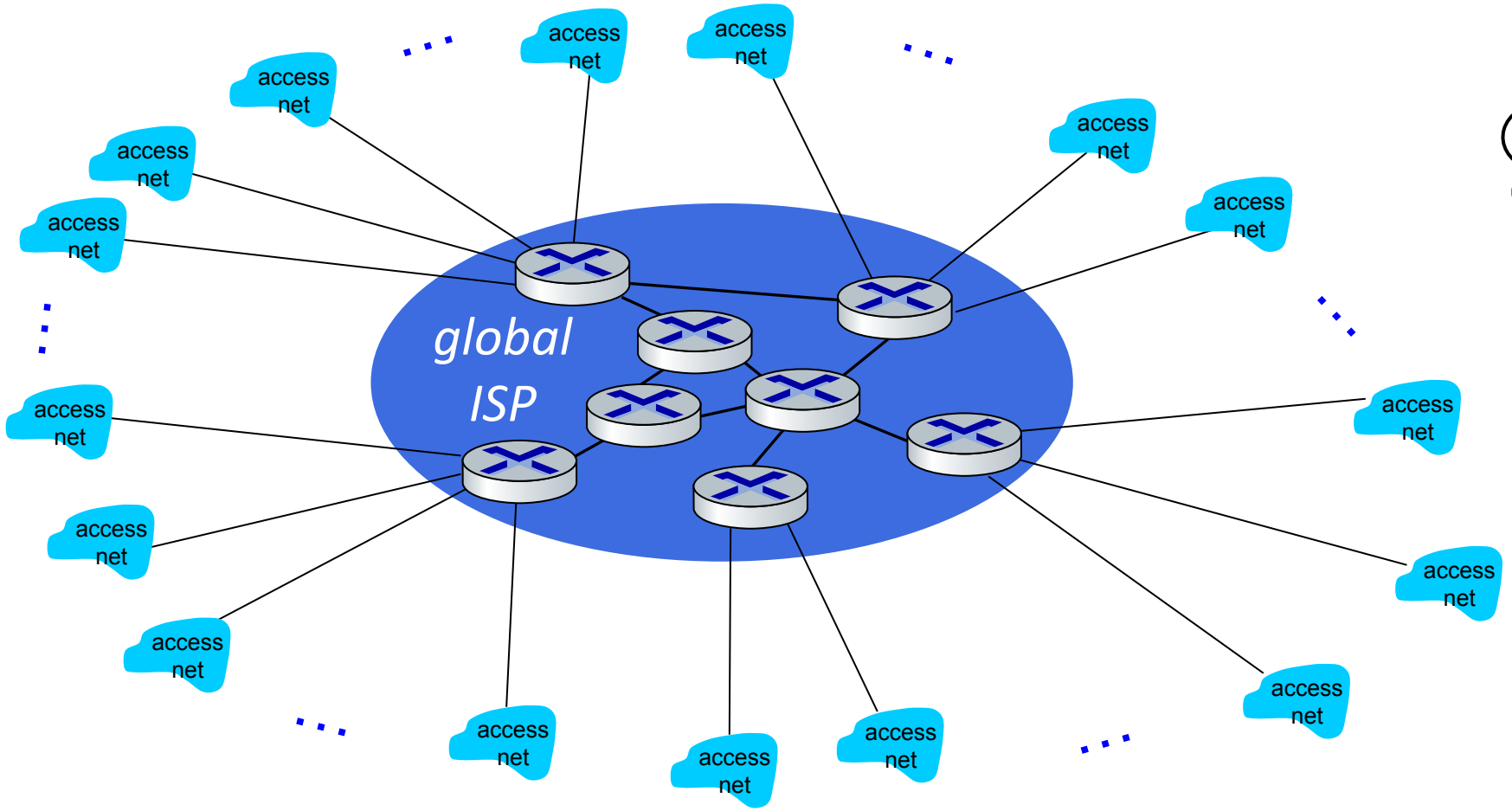
Internet Structure: a "network of networks"

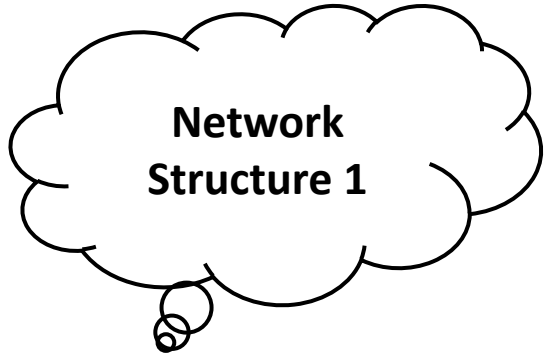
Question: given *millions* of access ISPs, how to connect them together?



Internet Structure: a "network of networks"

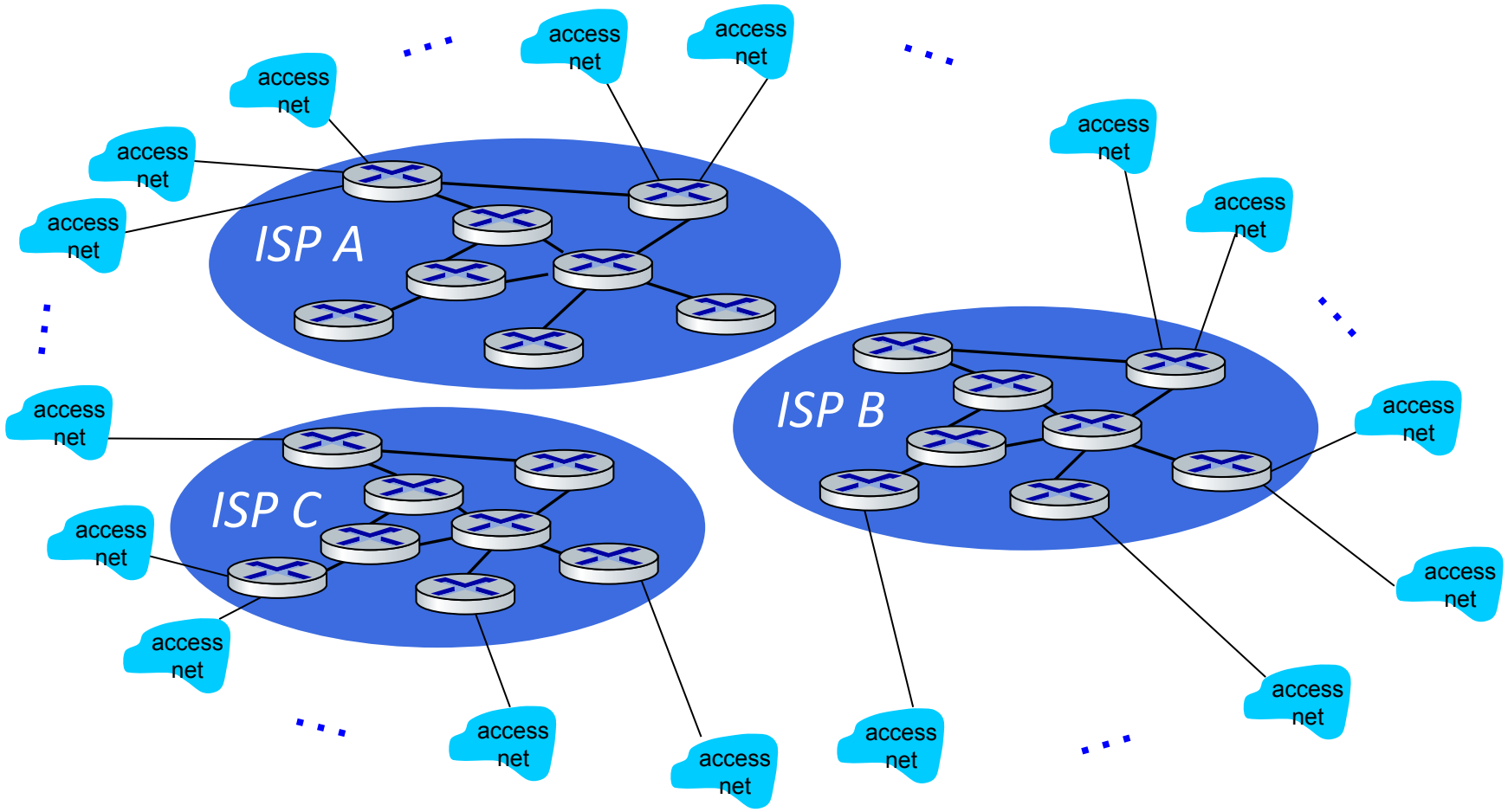
Option: connect each access ISP to one global transit ISP?
Customer and provider ISPs have economic agreement.





Internet Structure: a "network of networks"

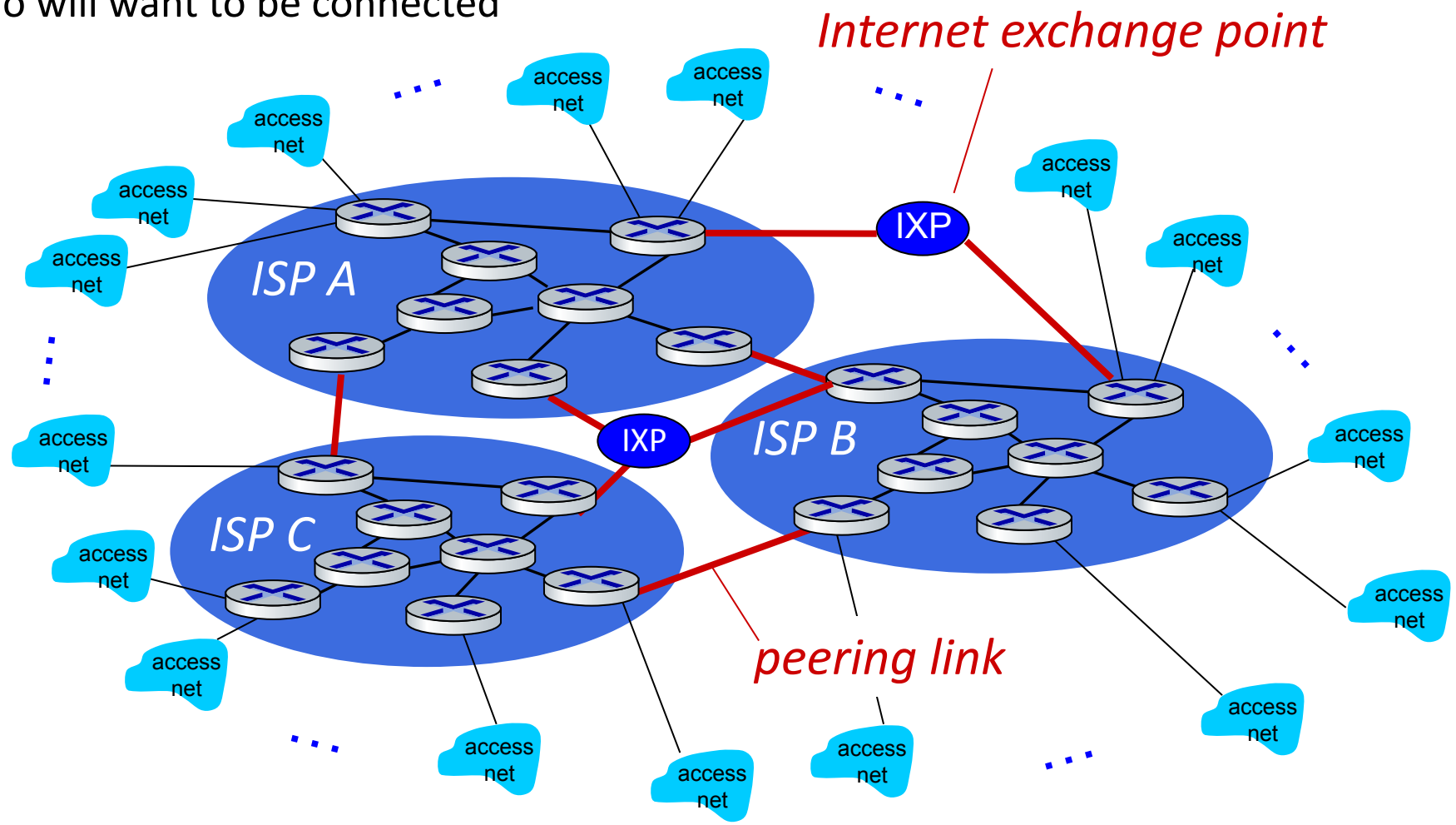
But if one global ISP is viable business, there will be competitors ….



Network Structure 2

Internet Structure: a "network of networks"

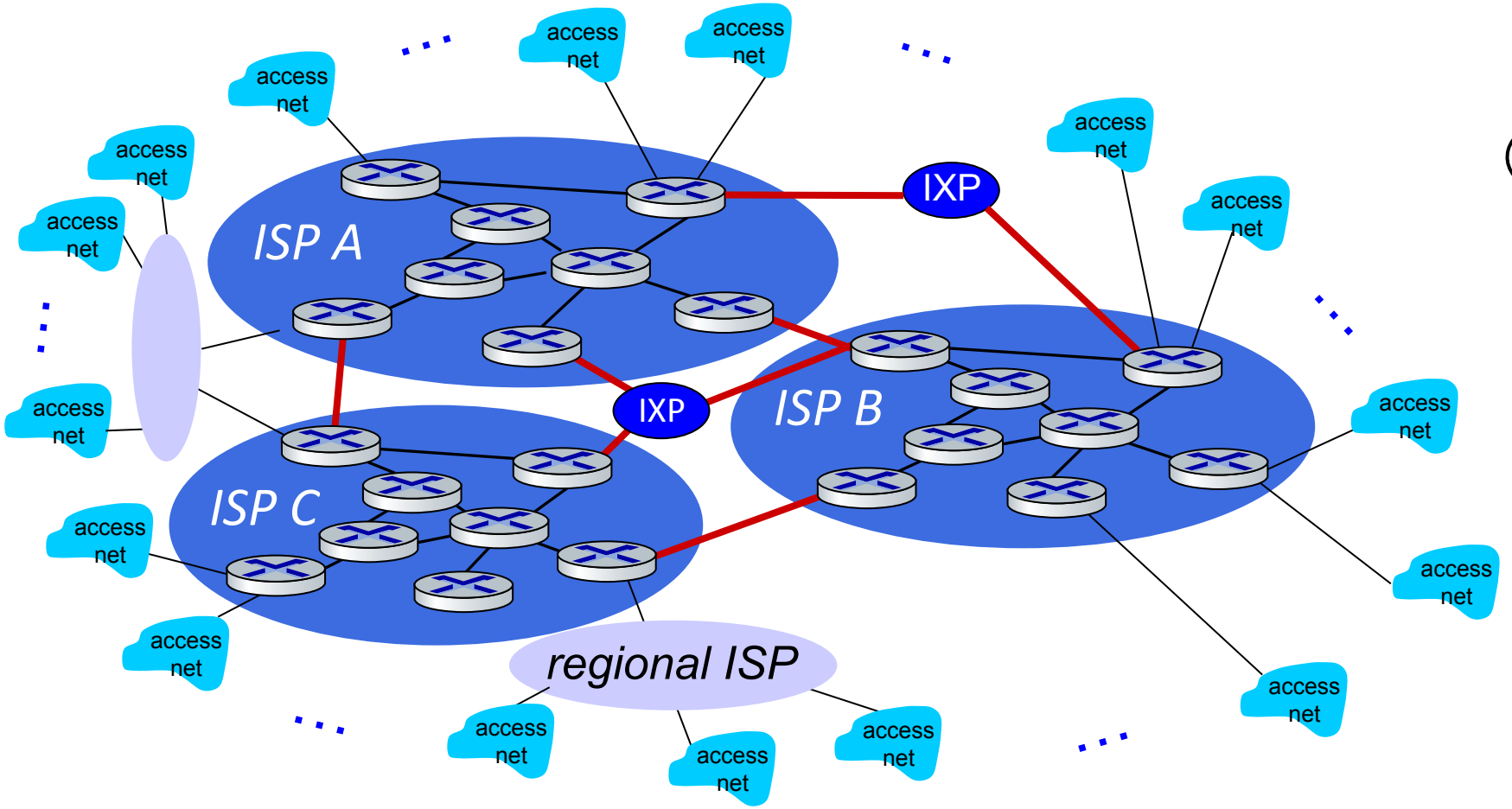
But if one global ISP is viable business, there will be competitors ….
who will want to be connected



Network Structure 3

Internet Structure: a “network of networks”

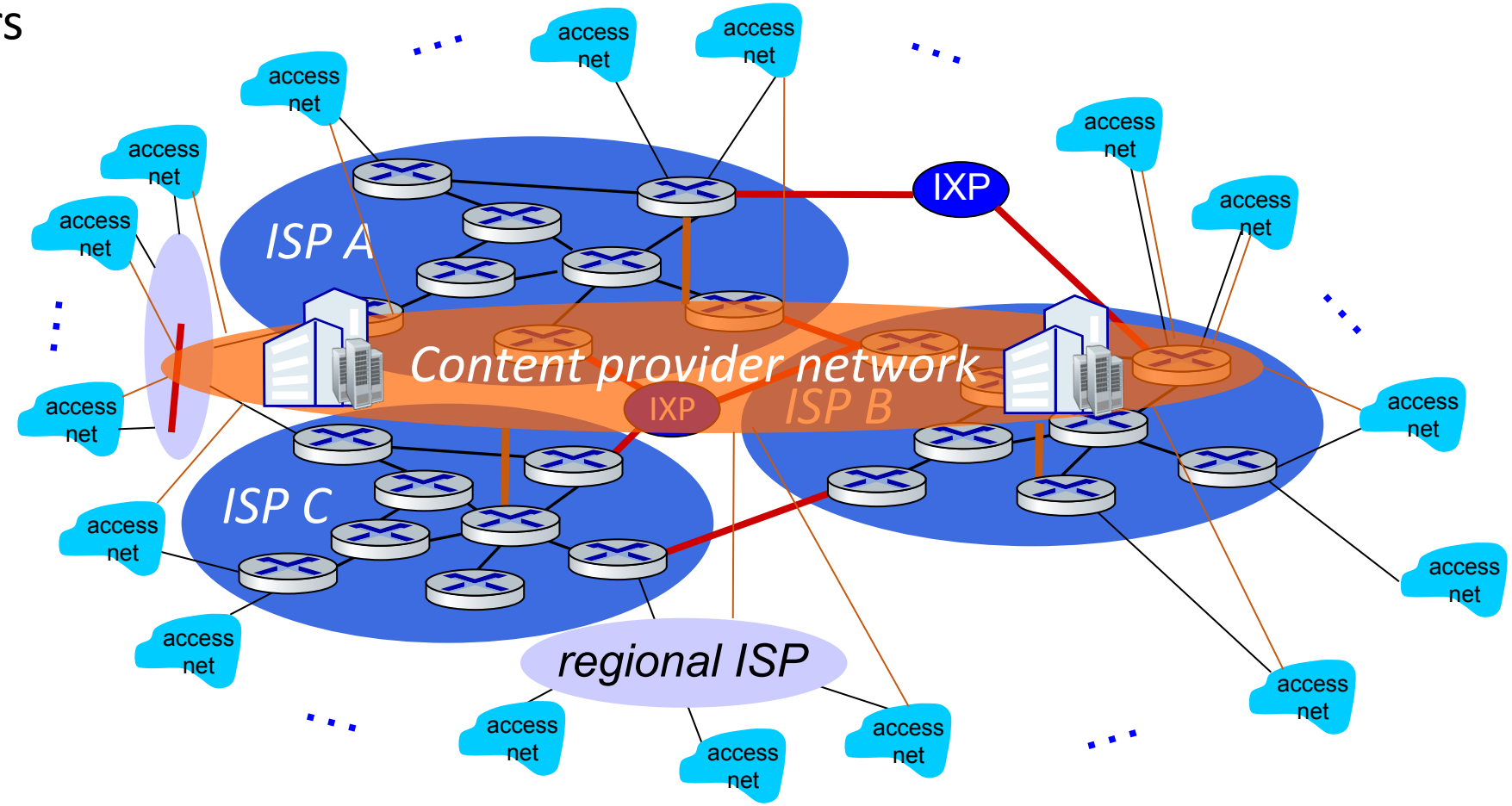
… and regional networks may arise to connect access nets to ISPs



Network Structure 4

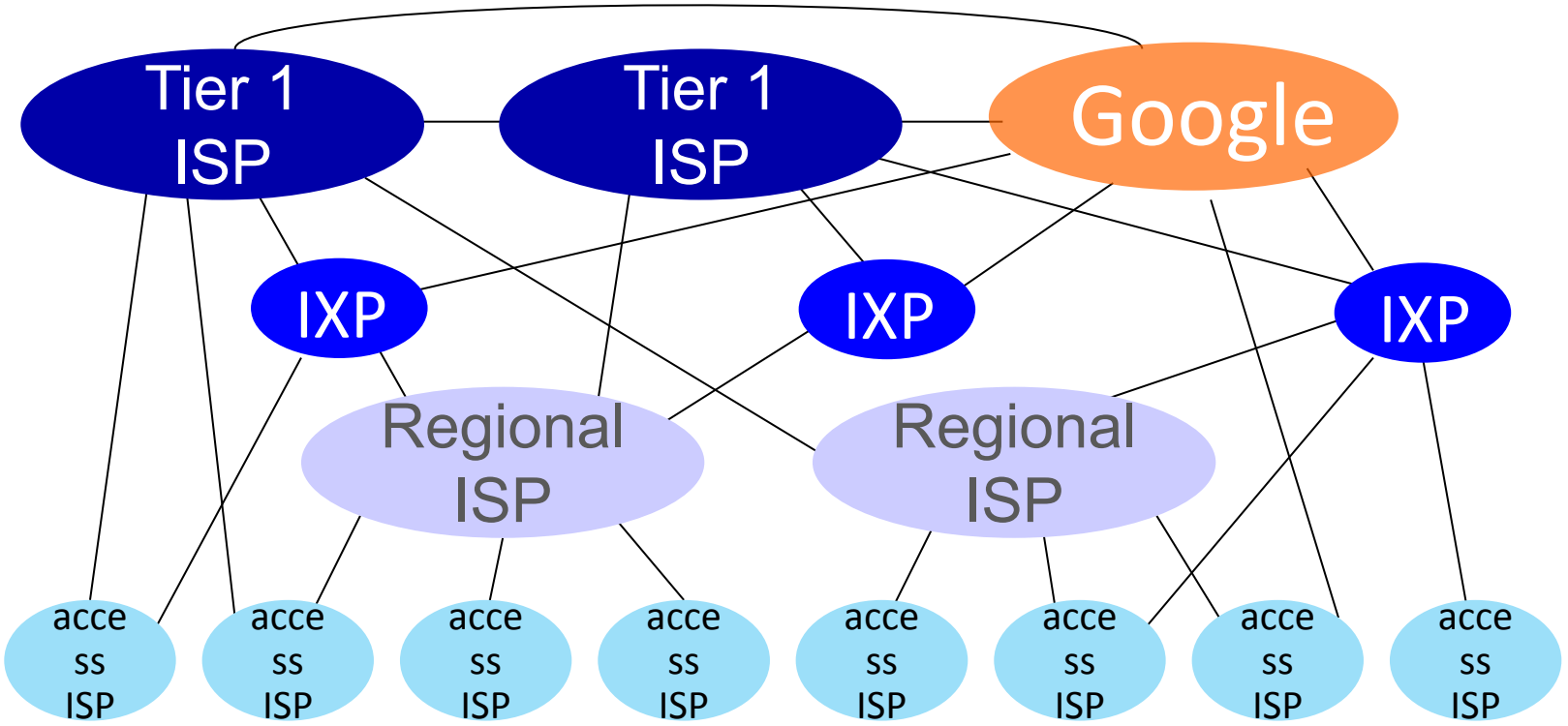
Internet Structure: a "network of networks"

… and content provider networks (e.g., Google, Microsoft, Akamai) may run their own network, to bring services, content close to end users



Network Structure 5

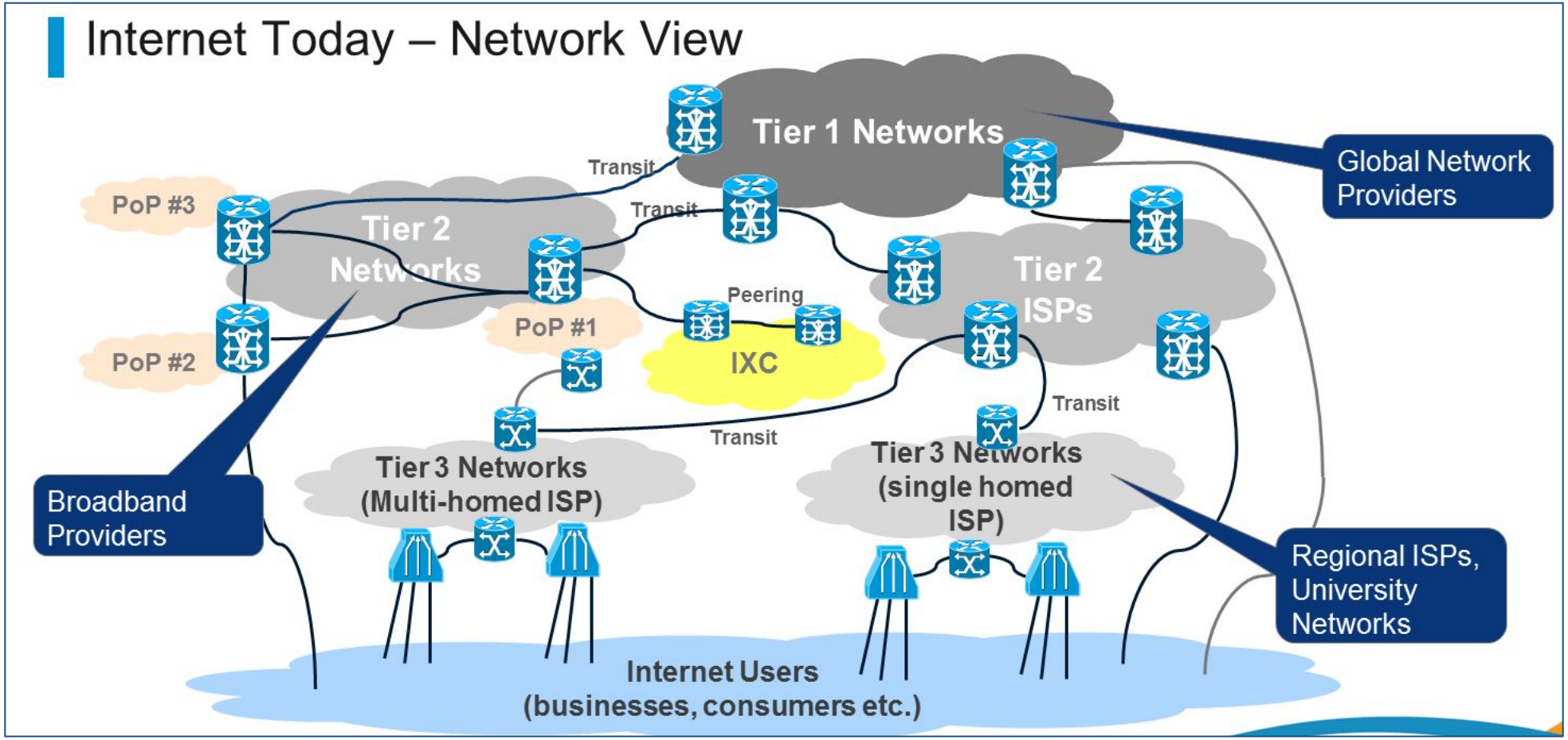
Internet Structure: a "network of networks"



At "center": small # of well-connected large networks

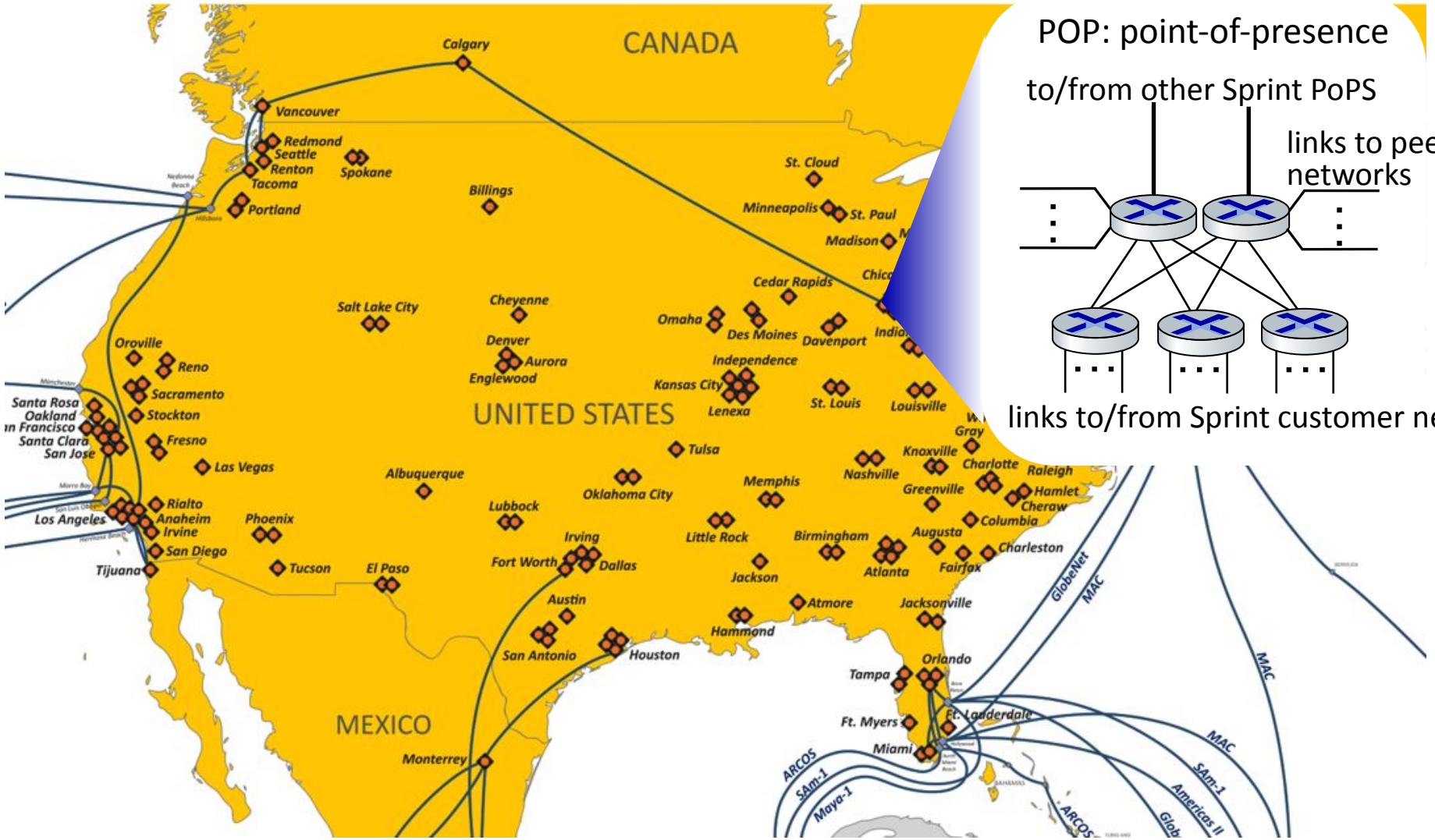
- "tier-1" commercial ISPs (e.g., Level 3, Sprint, AT&T, NTT), national & international coverage
- content provider networks (e.g., Google, Facebook): private network that connects its data centers to Internet, often bypassing tier-1, regional ISPs

Internet Structure: a “network of networks”



COMPUTER NETWORKS

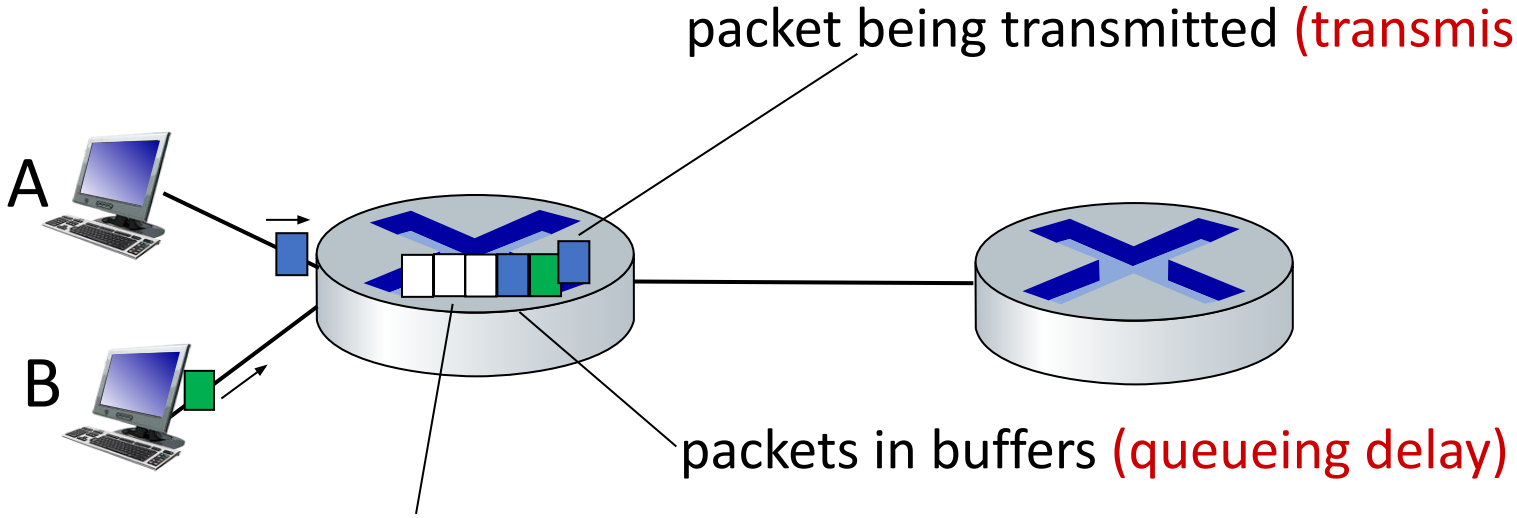
Network Core: Tier 1 ISP Network Map: Sprint 2019



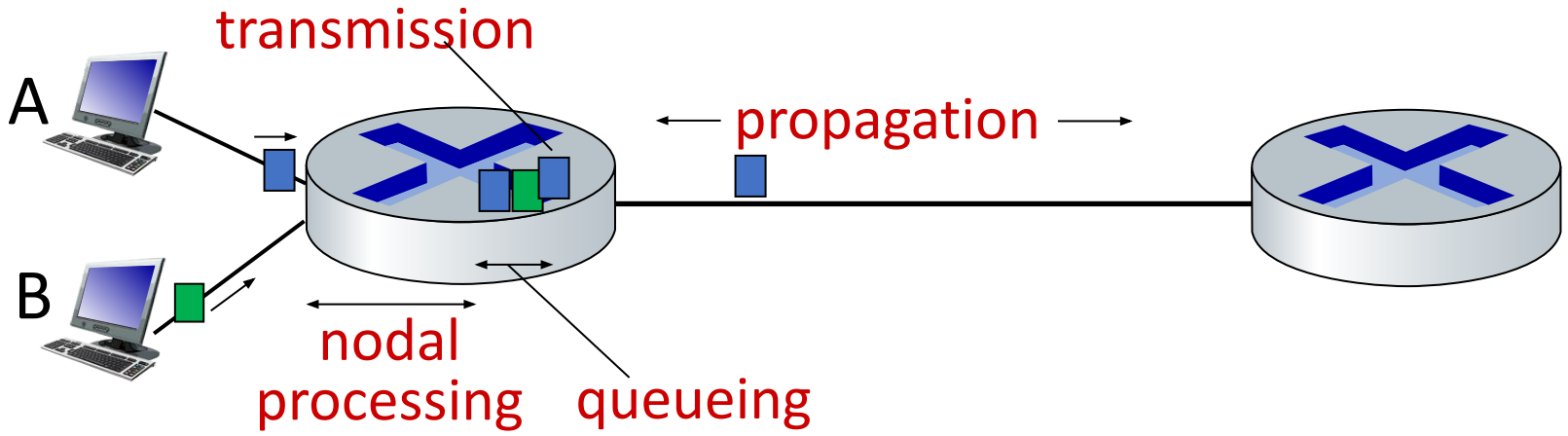
How do packet loss and delay occurs?

packets *queue* in router buffers

- packets queue, wait for turn
- arrival rate to link (temporarily) exceeds output link capacity: packet loss



Performance: Packet Delay – 4 Sources



$$d_{nodal} = d_{proc} + d_{queue} + d_{trans} + d_{prop}$$

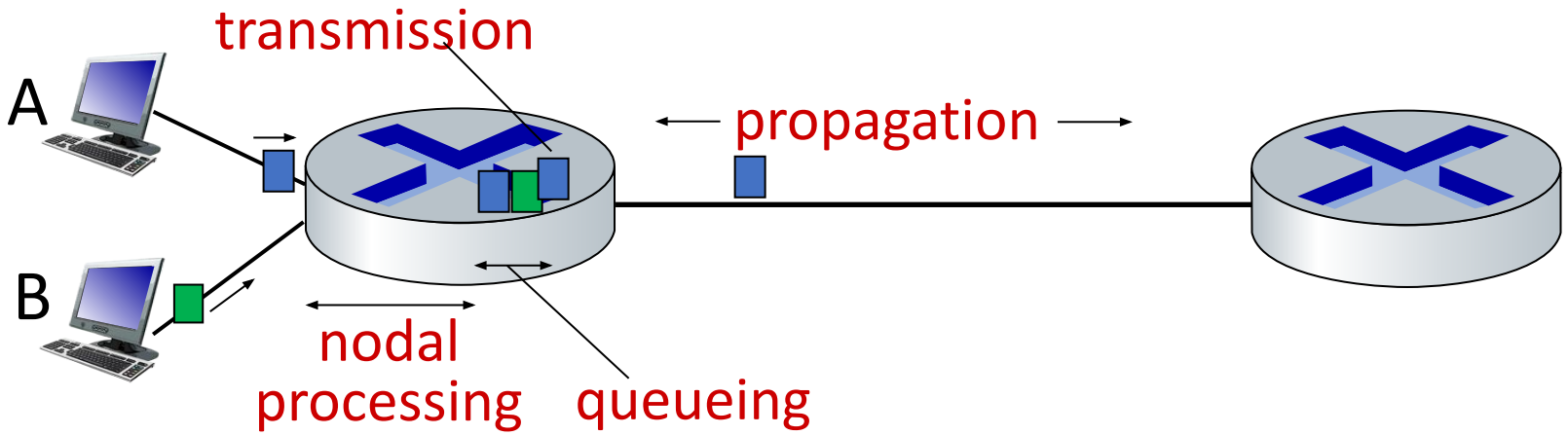
d_{proc}: nodal processing

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

d_{queue}: queueing delay

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

Performance: Packet Delay – 4 Sources



* Check out the online interactive exercises:
http://gaia.cs.umass.edu/kurose_ross

$$d_{nodal} = d_{proc} + d_{queue} + d_{trans} + d_{prop}$$

d_{trans}: transmission delay:

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

d_{prop}: propagation delay:

- d: length of physical link
- s: propagation speed (~2×10^8 m/sec)
- $d_{prop} = d/s$

d_{trans} and d_{prop} *very* different

Transmission Delay vs Propagation Delay

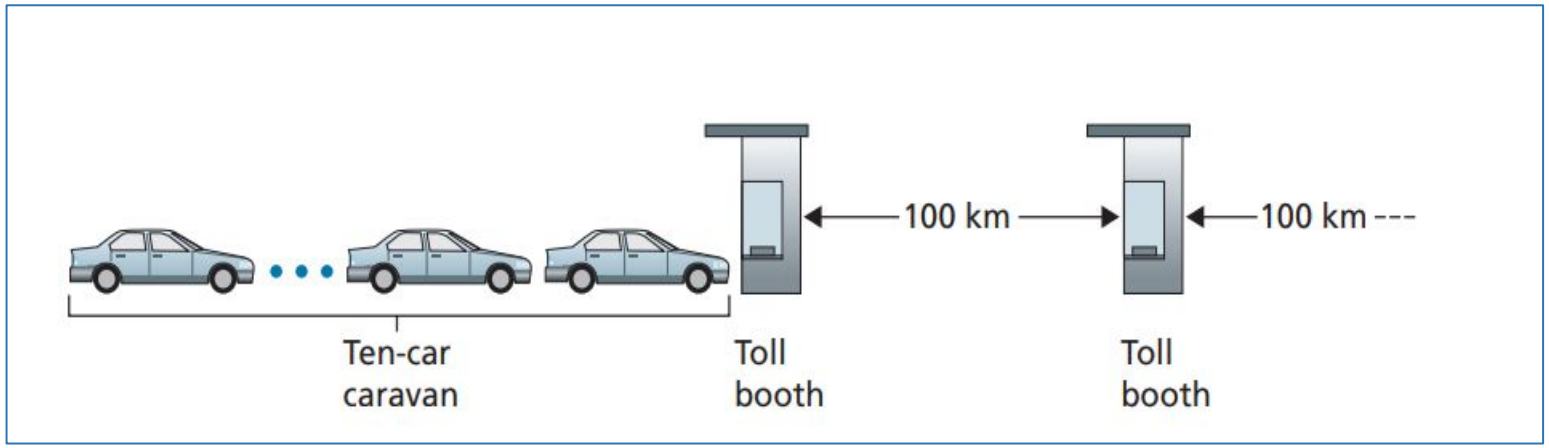
Transmission Delay	Propagation Delay
Time required for the router to push out the packet.	Time it takes a bit to propagate from one router to the next.
A function of the **packet's length** and the **transmission rate** of the link.	A function of the **distance** between the two routers.
$d_{trans} = L/R$	$d_{prop} = d/s$
Nothing to do with the distance between the two routers.	Nothing to do with the packet's length or the transmission rate of the link.

https://wps.pearsoned.com/ecs_kurose_compnetw_6/216/55463/14198702.cw/index.html

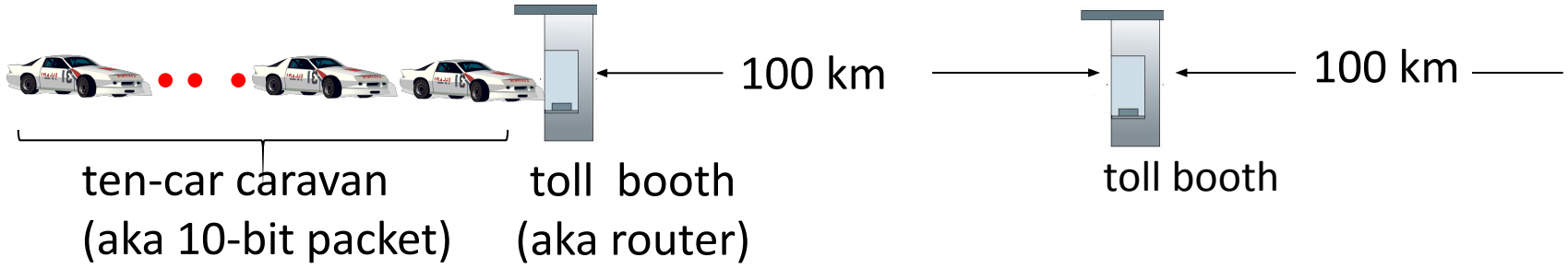
COMPUTER NETWORKS

Caravan Analogy – Transmission Delay & Propagation Delay

- Consider a highway that has a tollbooth every 100 kilometers, as shown in Figure. You can think of the highway segments between tollbooths as links and the tollbooths as routers.
- Suppose that cars travel (that is, propagate) on the highway at a rate of 100 km/hour (that is, when a car leaves a tollbooth, it instantaneously accelerates to 100 km/hour and maintains that speed between tollbooths).
- Suppose next that 10 cars, traveling together as a caravan, follow each other in a fixed order.
- You can think of each car as a bit and the caravan as a packet.



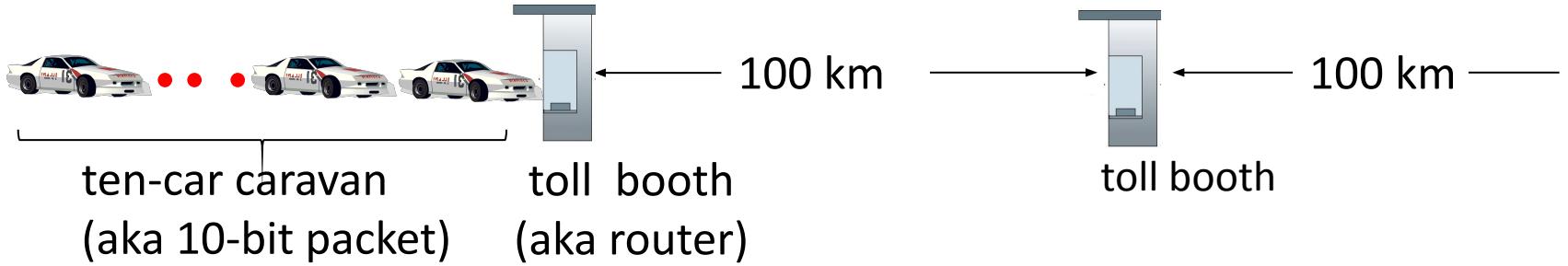
Performance: Delay – Caravan Analogy



- cars "propagate" at 100 km/hr
- toll booth takes 12 sec to service car (bit transmission time)
- car ~ bit; caravan ~ packet
- *Q:* How long until caravan is lined up before 2nd toll booth?

- time to "push" entire caravan through toll booth onto highway = 12*10 = 120 sec **(trans delay)**
- time for last car to propagate from 1st to 2nd toll both: 100km/(100km/hr) = 1 hr **(propagation delay = distance/speed)**
- *A:* 62 minutes

Performance: Delay – Caravan Analogy (more)



- suppose cars now "propagate" at 1000 km/hr
- and suppose toll booth now takes one min to service a car
- *Q:* Will cars arrive to 2nd booth before all cars serviced at first booth?

A: Yes! after 7 min, first car arrives at second booth; three cars still at first booth

COMPUTER NETWORKS

Animation Link

Understand the difference between Transmission and Propagation Delay

https://computerscience.unicam.it/marcantoni/reti/applet/TransmissionVsPropagationDelay/traProp.html

Queuing delay:

https://computerscience.unicam.it/marcantoni/reti/applet/QueuingAndLossInteractive/1.html

Queuing Delay & Packet Loss

Performance: Packet Queueing Delay revisited

Unlike other delays (d_{proc}, d_{trans}, d_{prop}), d_{queue} is interesting.

- Can vary from packet to packet.
- Characterize d_{queue} -> average, variance, probability that it exceeds some specified value.

When is the queuing delay large and when is it insignificant?

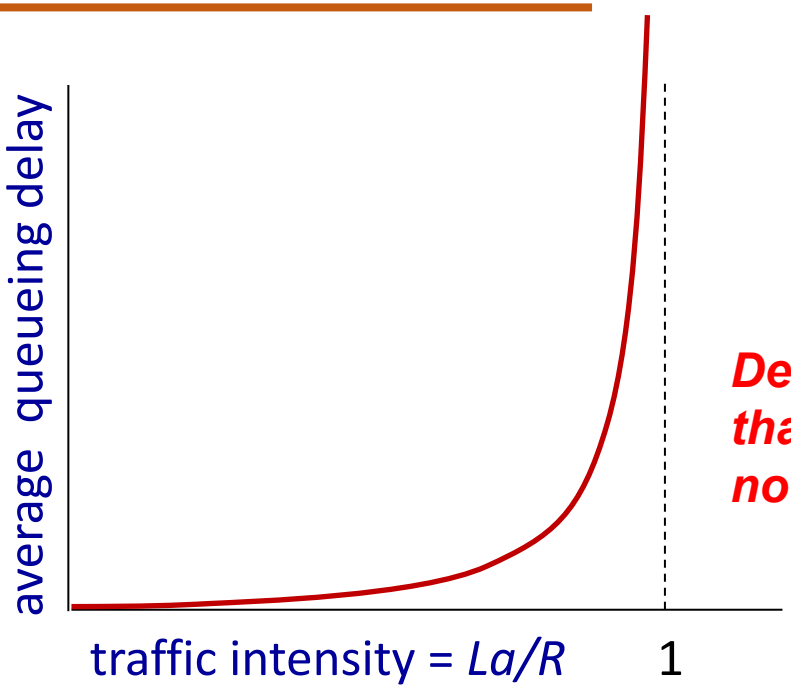
- Rate at which traffic arrives at the queue,
- Transmission rate of the link,
- Nature of the arriving traffic – periodically or in bursts

Performance: Packet Queueing Delay revisited

- *R:* link bandwidth (bps)
- *L:* packet length (bits)
- *a:* average packet arrival rate (pps)
- La: avg. rate at which bits arrive at the queue

- La/R > 1: more "work" arriving is more than can be serviced - average delay infinite!
- La/R <= 1: nature of arriving traffic
- *La/R* ~ 0: avg. queueing delay small

La/R > 1: Average rate at which bits arrive at the queue exceeds the rate at which the bits can be transmitted from the queue.



Design your system so that the traffic intensity is no greater than 1.

La/R ~ 0

La/R -> 1

Traceroute – visit: traceroute.org

- Traceroute is a **simple program** that can run in any Internet host.
- Suppose there are N - 1 routers between the source and the destination. Then the source will send N special packets into the network, with each packet addressed to the ultimate destination.
- These N special packets are marked 1 through N, with the first packet marked 1 and the last packet marked N.
- When the nth router receives the nth packet marked n, the router does not forward the packet toward its destination, but instead sends a message back to the source.
- When the destination host receives the Nth packet, it too returns a message back to the source.
- The source records the time that elapses between when it sends a packet and when it receives the corresponding return message; it also records the name and address of the router (or the destination host) that returns the message.

COMPUTER NETWORKS

Traceroute – visit: traceroute.org

traceroute: gaia.cs.umass.edu to www.eurecom.fr

3 delay measurements from gaia.cs.umass.edu to cs-gw.cs.umass.edu

```
1  cs-gw (128.119.240.254)  1 ms  1 ms  2 ms
2  border1-rt-fa5-1-0.gw.umass.edu (128.119.3.145)  1 ms  1 ms  2 ms
3  cht-vbns.gw.umass.edu (128.119.3.130)  6 ms 5 ms 5 ms
4  jn1-at1-0-0-19.wor.vbns.net (204.147.132.129)  16 ms 11 ms 13 ms
5  jn1-so7-0-0-0.wae.vbns.net (204.147.136.136)  21 ms 18 ms 18 ms
6  abilene-vbns.abilene.ucaid.edu (198.32.11.9)  22 ms  18 ms  22 ms
7  nycm-wash.abilene.ucaid.edu (198.32.8.46)  22 ms  22 ms  22 ms
8  62.40.103.253 (62.40.103.253)  104 ms 109 ms 106 ms
9  de2-1.de1.de.geant.net (62.40.96.129)  109 ms 102 ms 104 ms
10  de.fr1.fr.geant.net (62.40.96.50)  113 ms 121 ms 114 ms
11  renater-gw.fr1.fr.geant.net (62.40.103.54)  112 ms  114 ms  112 ms
12  nio-n2.cssi.renater.fr (193.51.206.13)  111 ms  114 ms  116 ms
13  nice.cssi.renater.fr (195.220.98.102)  123 ms  125 ms  124 ms
14  r3t2-nice.cssi.renater.fr (195.220.98.110)  126 ms  126 ms  124 ms
15  eurecom-valbonne.r3t2.ft.net (193.48.50.54)  135 ms  128 ms  133 ms
16  194.214.211.25 (194.214.211.25)  126 ms  128 ms  126 ms
17  * * *
18  * * *
19  fantasia.eurecom.fr (193.55.113.142)  132 ms  128 ms  136 ms
```

3 delay measurements to border1-rt-fa5-1-0.gw.umass.edu

trans-oceanic link

looks like delays *decrease*!

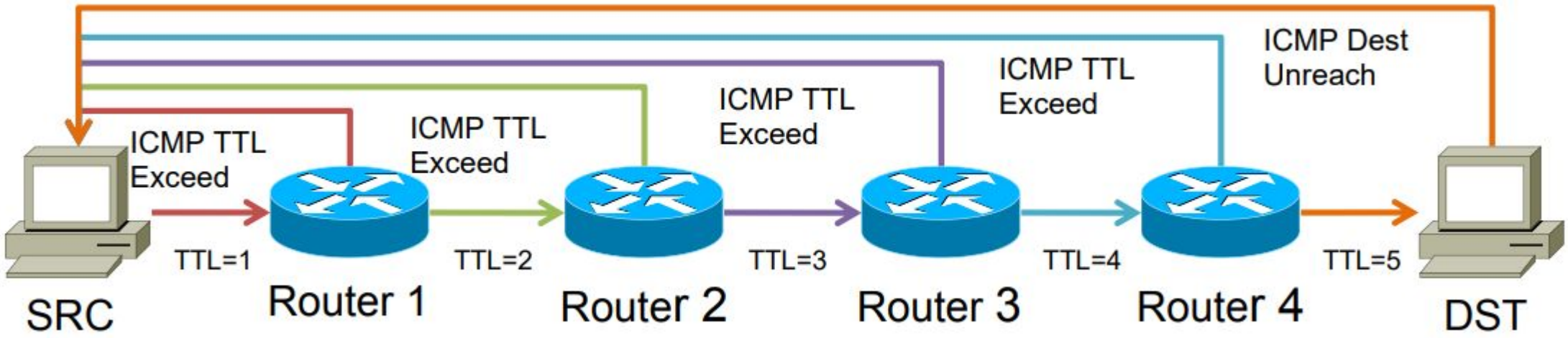
* means no response (probe lost, router not replying)

In the given traceroute output:

- The first column represents the hop number.
- The second column shows the hostname of the router or the IP address.
- The next three columns display the round-trip times in milliseconds for three consecutive packets.

* Do some traceroutes from exotic countries at www.traceroute.org

How Traceroute works?

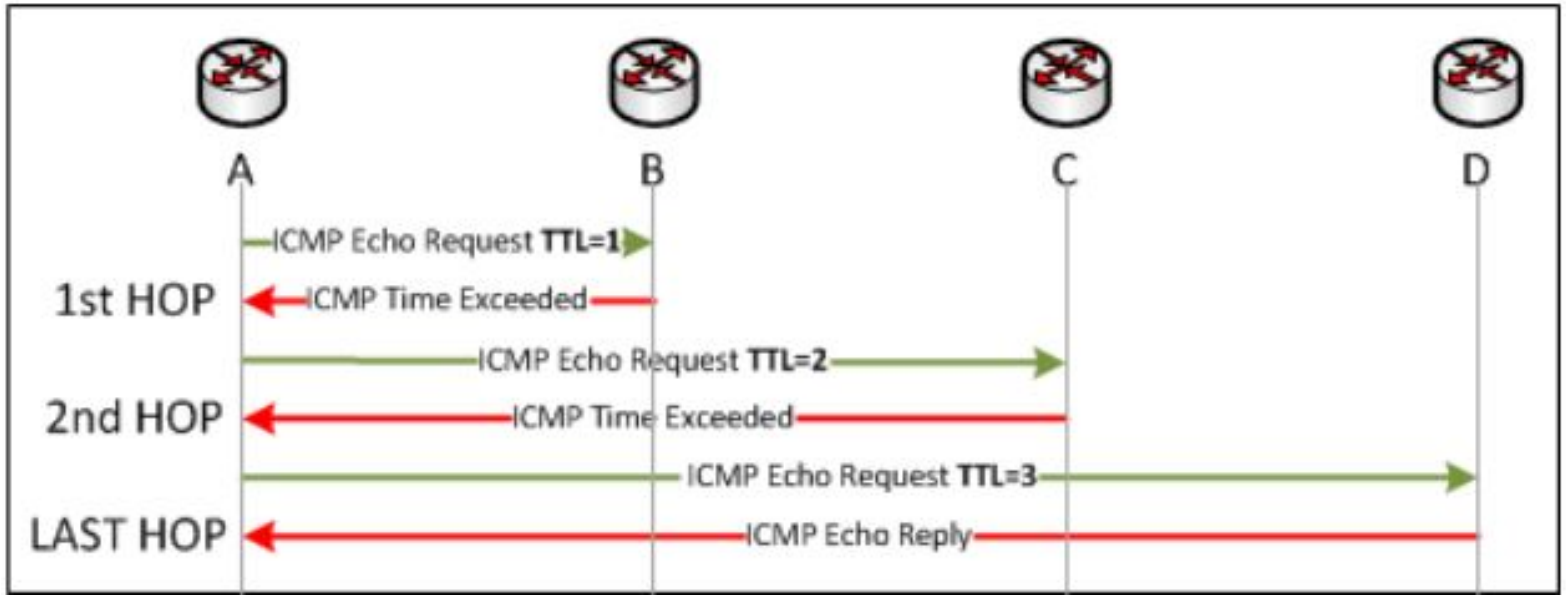


Refer RFC 1393, **Traceroute Using an IP Option**

Don't Trust Traceroute (Completely)

COMPUTER NETWORKS

How Traceroute works?




```
Router#traceroute 10.12.0.2
Type escape sequence to abort.
Tracing the route to 10.12.0.2

  1    10.10.0.2        3 msec    5 msec    5 msec
  2    10.11.0.2        8 msec    3 msec    3 msec
  3    10.12.0.2        5 msec    5 msec    10 msec
```

COMPUTER NETWORKS

How Traceroute works?

```
C:\Users\KUNDHAVI>ping google.com

Pinging google.com [142.250.70.78] with 32 bytes of data:
Reply from 142.250.70.78: bytes=32 time=30ms TTL=119
Reply from 142.250.70.78: bytes=32 time=31ms TTL=119
Reply from 142.250.70.78: bytes=32 time=31ms TTL=119
Reply from 142.250.70.78: bytes=32 time=62ms TTL=119

Ping statistics for 142.250.70.78:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 30ms, Maximum = 62ms, Average = 38ms
```

- Ping
- Traceroute
- PingPlotter

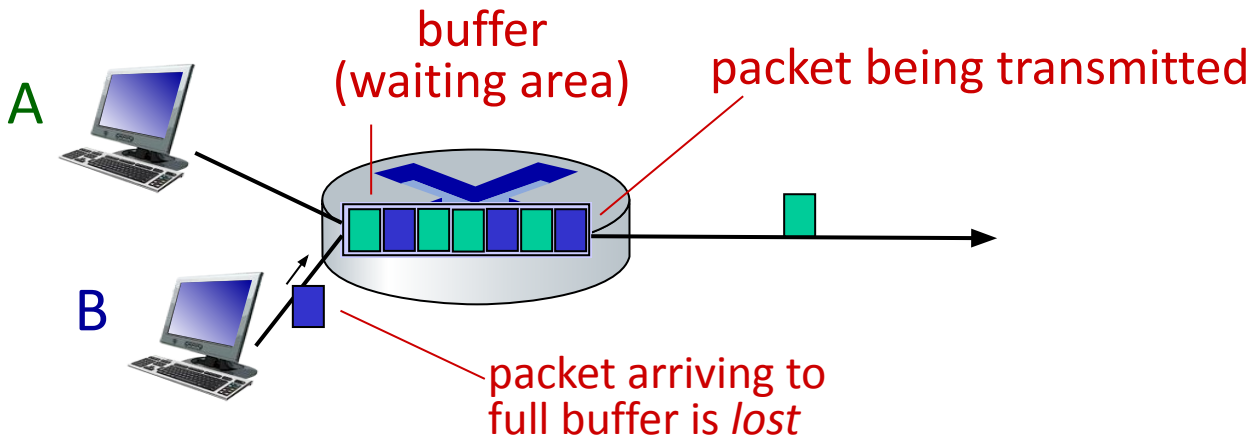
```
C:\Users\KUNDHAVI>tracert google.com

Tracing route to google.com [142.250.70.78]
over a maximum of 30 hops:

  1     2 ms     3 ms     2 ms  dsldevice.lan [192.168.1.1]
  2     9 ms    33 ms     6 ms  223.178.56.1
  3     8 ms     9 ms    43 ms  nsg-corporate-97.95.187.122.airtel.in [122.187.95.97]
  4     *        *        *     Request timed out.
  5     *       15 ms    14 ms  142.250.169.206
  6    13 ms    15 ms    13 ms  142.251.71.187
  7   176 ms   110 ms    12 ms  74.125.242.155
  8     *       26 ms    43 ms  64.233.174.3
  9     *        *      118 ms  142.250.238.182
 10    30 ms    29 ms    29 ms  108.170.248.193
 11    29 ms    29 ms    28 ms  192.178.86.201
 12    30 ms    31 ms    31 ms  pnbomb-ab-in-f14.1e100.net [142.250.70.78]
```

Performance: Packet loss

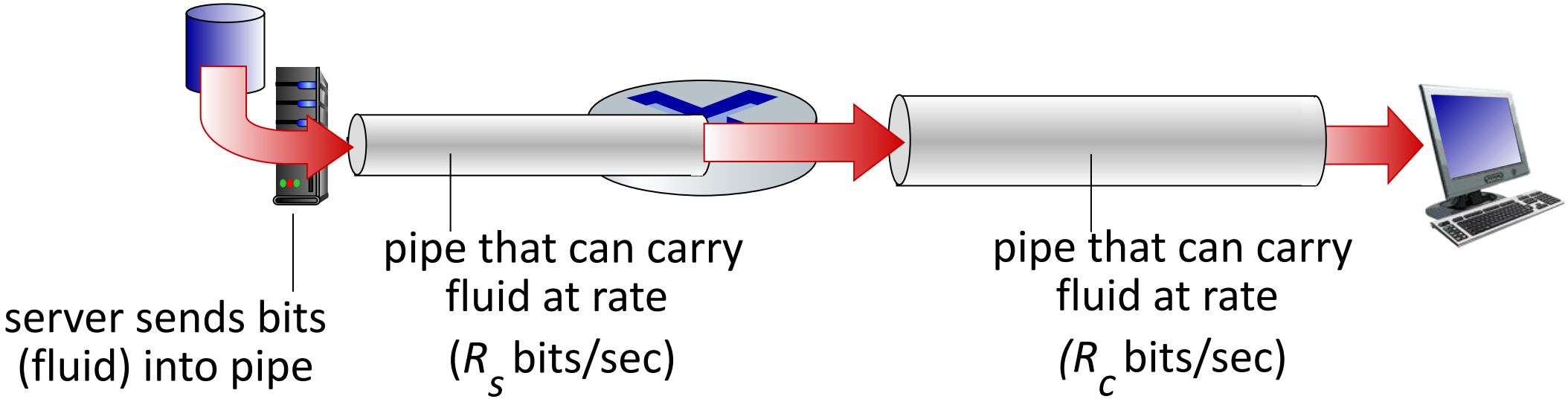
- queue (aka buffer) preceding link in buffer has finite capacity
- packet arriving to full queue dropped (aka lost)
- lost packet may be retransmitted by previous node, by source end system, or not at all



* Check out the Java applet for an interactive animation on queuing and loss

Performance: Throughput

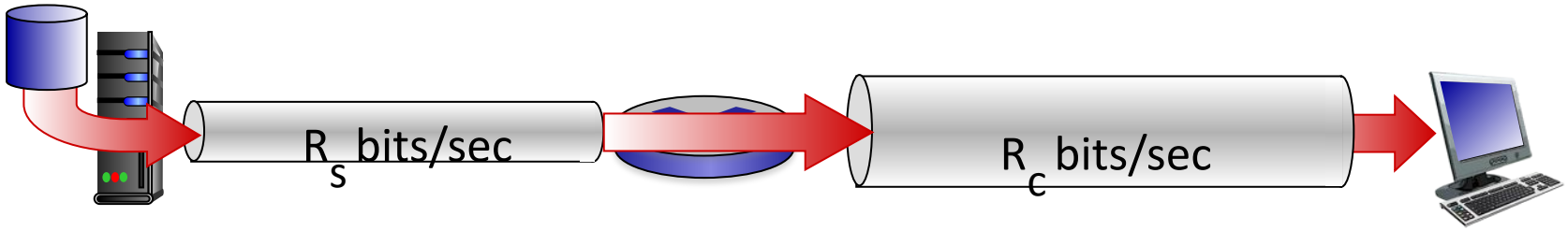
- *throughput:* rate (bits/time unit) at which bits are being sent from sender to receiver
  - *instantaneous:* rate at given point in time
  - *average:* rate over longer period of time



COMPUTER NETWORKS

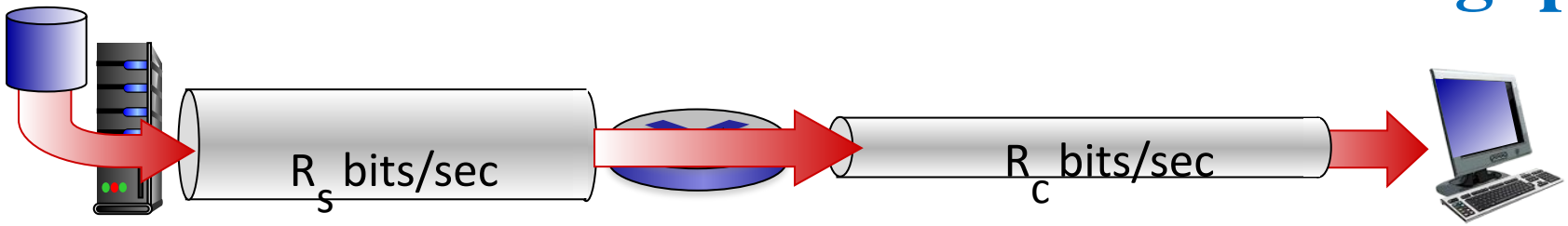
Performance: Throughput (more)

$R_s < R_c$ What is average end-end throughput?



$R_s > R_c$ What is average end-end throughput?

Throughput = min{*Rs*, *Rc*}



bottleneck link

link on end-end path that constrains end-end throughput.

Throughput – Numerical Example

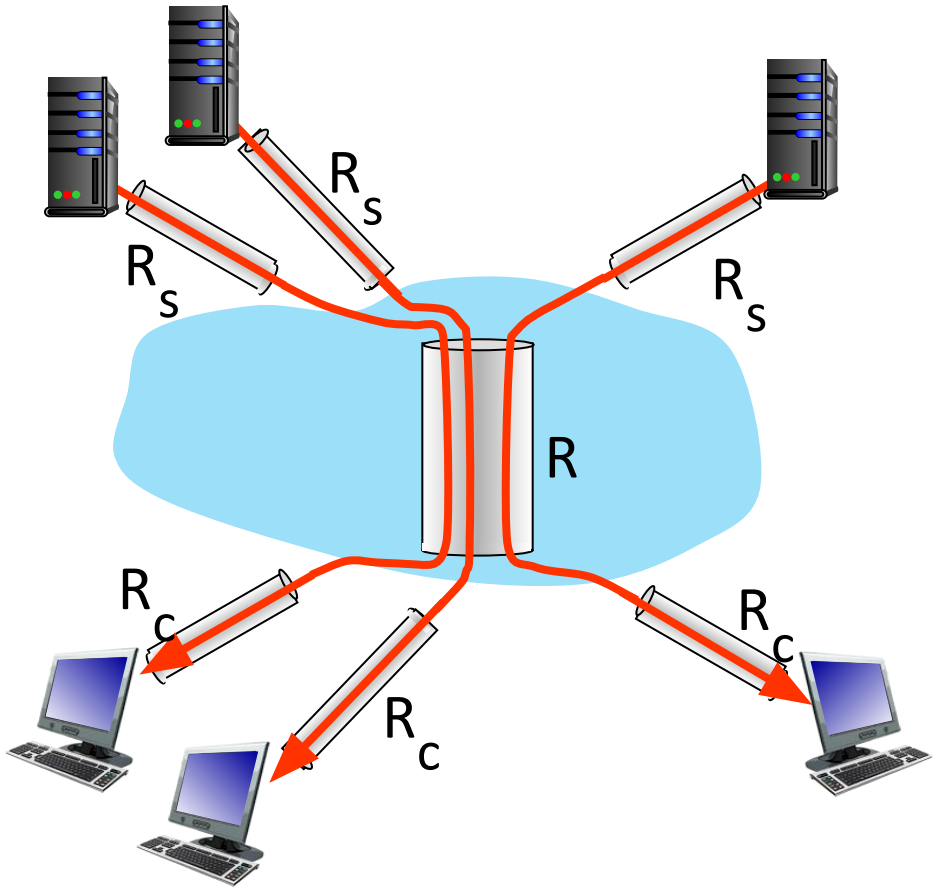
- Suppose you are downloading an MP3 file of F = 32 million bits.
- The server has a transmission rate of R_S = 2 Mbps and you have an access link of R_C = 1 Mbps.
- What is the time needed to transfer the file?

Let's work it out!

Solution:

- 32 seconds!

Performance: Throughput – Network Scenario



10 connections (fairly) share backbone bottleneck link *R* bits/sec

- per-connection end-end throughput: $min(R_c, R_s, R/10)$

Suppose R_s = 2 Mbps,

R_c = 1 Mbps,

R = 5 Mbps

10 clients from 10 servers = 10 downloads

End-to-end throughput for each download is now reduced to 500 kbps.

* Check out the online interactive exercises for more examples: http://gaia.cs.umass.edu/kurose_ross/

Computer Networks and the Internet

Protocol Layers

Compiled by Kundhavai K R

Department of Computer Science and Engineering

COMPUTER NETWORKS

"Protocol Layers" and reference models

Networks are complex, with many "pieces":

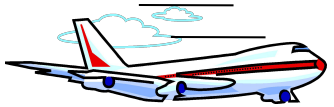
- hosts
- routers
- links of various media
- applications
- protocols
- hardware, software

Question:

is there any hope of *organizing* structure of network?

…. or at least our *discussion* of network arch?

Protocol Layers --- Example: Organization of Air Travel



ticket (purchase)

baggage (check)

gates (load)

runway takeoff

airplane routing

ticket (complain)

baggage (claim)

gates (unload)

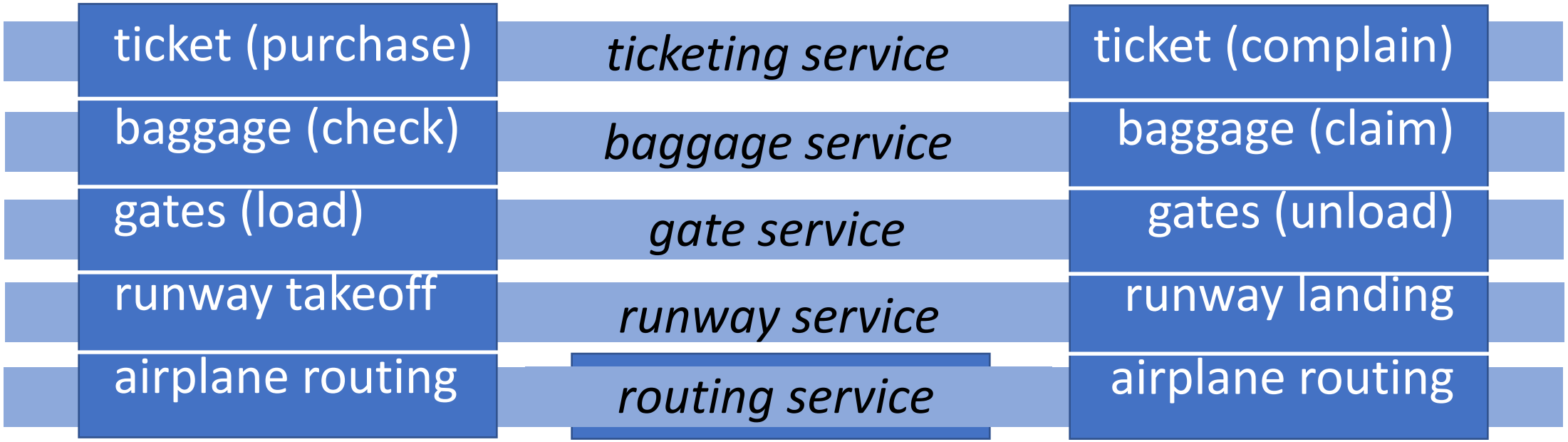
runway landing

airplane routing

airplane routing

airline travel: a series of steps, involving many services

Protocol Layers -- Layering of Airline functionality



layers: each layer implements a service

- via its own internal-layer actions
- relying on services provided by layer below

Protocol Layers -- Why layering?

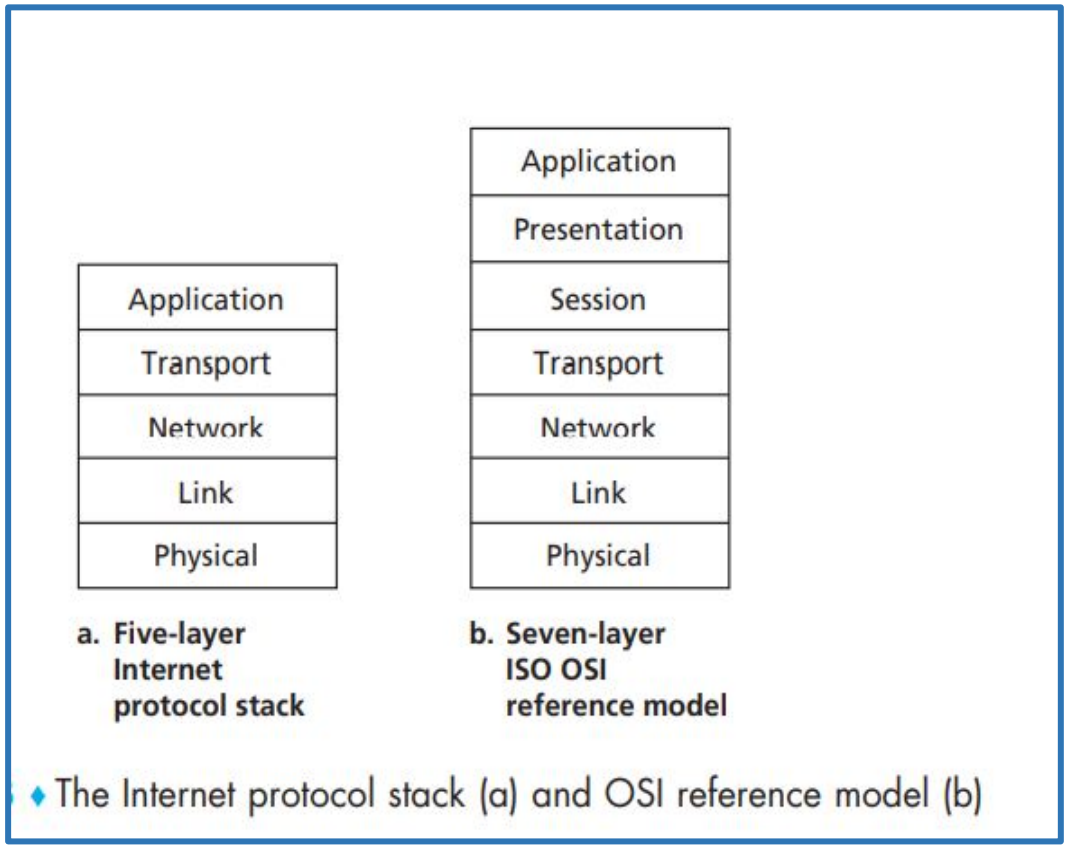
- A **layered architecture** allows us to discuss a well-defined, specific part of a large and complex system.
- This **simplification** itself is of considerable value by **providing modularity**, making it much easier to change the implementation of the service provided by the layer.
- As long as the layer provides the same service to the layer above it, and uses the same services from the layer below it.

- modularization eases maintenance, updating of system
  - **change in layer's service *implementation***: transparent to rest of system
  - e.g., change in gate procedure doesn't affect rest of system

A protocol layer can be implemented in software, in hardware, or in a combination of the two.

Protocol Layers

Top-down approach



The Internet protocol stack (a) and OSI reference model (b)

The protocols of the various layers are called the protocol stack. The Internet protocol stack consists of five layers: the physical, link, network, transport, and application layers.

Open Systems Interconnection (OSI) model – introduced in late 1970s by ISO.

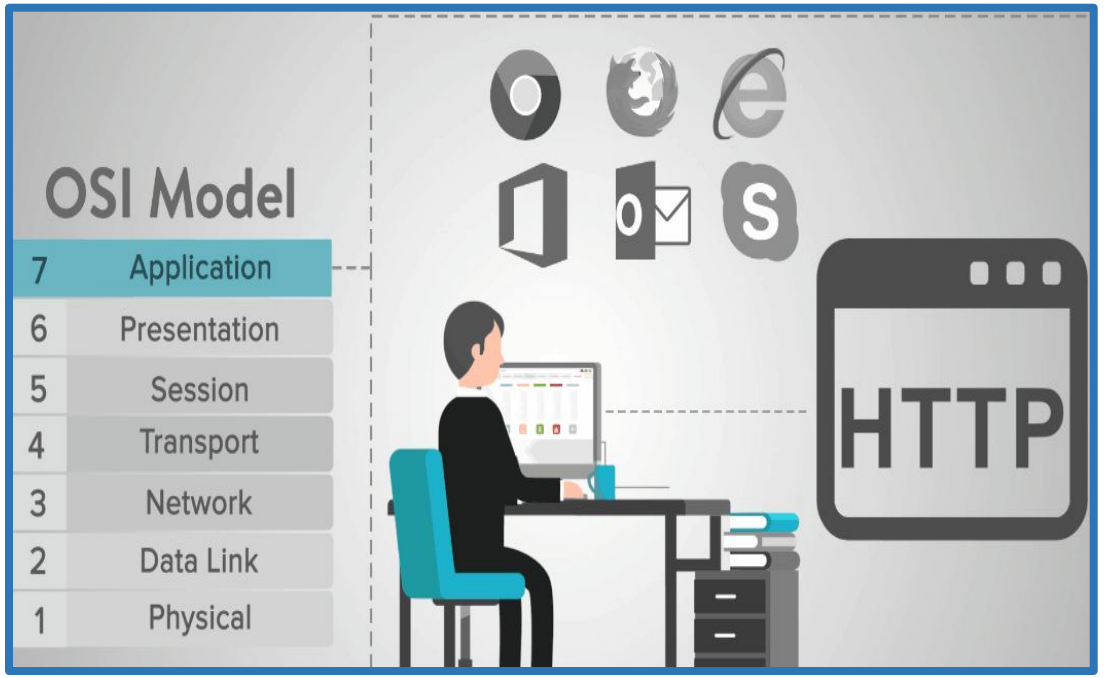
REMEMBER THIS PHRASE

Please **D**o **N**ot **T**ouch **S**teve's **P**et **A**lligator

Protocol Layers – Application Layer

- The application layer is where **network applications** and their application-layer protocols reside.
  - **HTTP** protocol (which provides for Web document request and transfer)
  - **SMTP** (which provides for the transfer of e-mail messages)
  - **FTP** (which provides for the transfer of files between two end systems)
  - **DNS** (translates domain names to IP address)

An application-layer protocol is distributed over multiple end systems, with the application in **one end system** using the protocol **to exchange packets** of information with the application in **another end system**.
Packet of information at the application layer is known as a **MESSAGE**.



Protocol Layers – Transport Layer

- Transport layer **transports** application-layer messages between application endpoints.
  - Two transport protocols, **TCP and UDP,** either of which can transport application-layer messages.
    - **TCP** provides a **connection-oriented service** to its applications.
      - This service **includes guaranteed delivery** of application layer messages to the destination and **flow control** (that is, sender/receiver speed matching).
      - **Segmentation & reassembly, sockets, flow and error control.**
      - TCP also breaks long messages into shorter segments and provides a **congestion-control mechanism**, so that a source throttles its transmission rate when the network is congested.
    - **UDP** protocol provides a **connectionless service** to its applications.
      - This provides **no reliability, no flow control, and no congestion control.**
- Transport-layer packet is known as **SEGMENT.**

Protocol Layers – Network Layer

- Transport-layer protocol in a source host passes a segment and a destination address to the network layer. (postal service).
- The network layer then provides the service of delivering the segment to the transport layer in the destination host. (**Addressing**)
- Network layer includes the **IP protocol** -has fields in the datagram as well as how the end systems and routers act on these fields.
- Many **routing protocols** present in this layer helps to determine the routes that datagrams take between sources and destinations.
- Simply referred to as the **IP layer.**
- Network-layer packet is known as **DATAGRAM**.

Protocol Layers – Link Layer

- The **network layer relies** on the services of the **link layer** for moving a datagram from one node to another.
- The services provided by the link layer depend on the **specific link-layer protocol** that is employed over the link.
  - Some link-layer protocols provide - **reliable delivery, from transmitting node, over one link, to receiving node.**
- This reliable delivery service is **different** from the reliable delivery service of TCP, which provides reliable delivery from **one end system to another**.
- Link layer protocols include **Ethernet, WiFi, and the cable access network's DOCSIS protocol**.
- A datagram may be handled by different link-layer protocols at different links along its route.(many nodes).
- The frame header contains the **MAC address.**
- Ex: A datagram may be handled by Ethernet on one link and by PPP on the next link. The network layer will receive a different service from each of the different link-layer protocols.
- Link layer packet is known as **FRAMES**.

COMPUTER NETWORKS

Protocol Layers – MAC ADRRESS VS IP ADDRESS

Aspect	MAC Address	IP Address
Type of Address	Physical address assigned to network interface	Logical address assigned to devices on a network
Uniqueness	Globally unique per network interface	Uniqueness is network specific and can change
Layer	Operates at the data link layer (Layer 2)	Operates at the network layer (Layer 3)
Routing	Not used for routing packets	Used for routing packets within a network
Protocol	Independent of the network layer protocol	Depends on network layer protocol (e.g., IP)
Format	Expressed as a 12-digit hexadecimal number	Expressed as a series of four decimal numbers

Aspect	MAC Address	IP Address
Assignment	Assigned by the manufacturer of the network device	Assigned by network administrators or DHCP servers
Changeability	Permanent and usually unchangeable	Can be static or dynamic; may change with network
Visibility	Mainly used within a local network segment	Can be used for global, inter-network communication
Size	Fixed length (48 bits, 6 bytes)	IPv4 is 32 bits (4 bytes), IPv6 is 128 bits (16 bytes)

MAC address is the one used on frames at the link layer, i.e. the frame header contains source and destination MAC Addresses.
How can a node know the MAC address of the node to which it is going to send the frame? It only knows its IP.
For this purpose we have **ARP Protocol** used by **link layer** to solve this problem.

Protocol Layers – Physical Layer

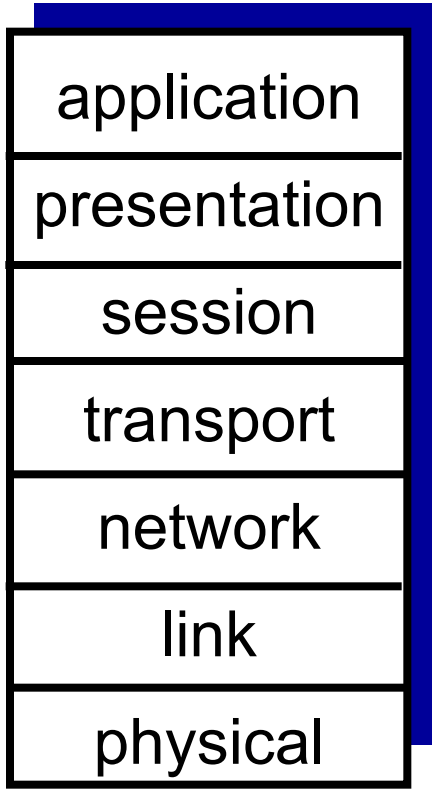
- The job of the physical layer is to move the individual bits within the frame from one node to the next.
- The protocols in this layer are again link dependent and further depend on the actual transmission medium of the link (for example, twisted-pair copper wire, single-mode fiber optics).
- In each case, a bit is moved across the link in a different way.
- Physical layer packet is known as **BITS.**

OSI Layer in a nutshell

- *application:* supporting network applications (access to network resources)
  - IMAP, SMTP, HTTP,DNS
- *transport:* process-process data transfer (segmentation & reassembly, sockets, connection, flow and error control)
  - TCP, UDP
- *network:* routing of datagrams from source to destination (addressing, routing)
  - IP, routing protocols
- *link:* data transfer between neighboring network elements (framing, addressing, flow & error control)
  - Ethernet, 802.11 (WiFi), PPP
- *physical:* bits “on the wire”

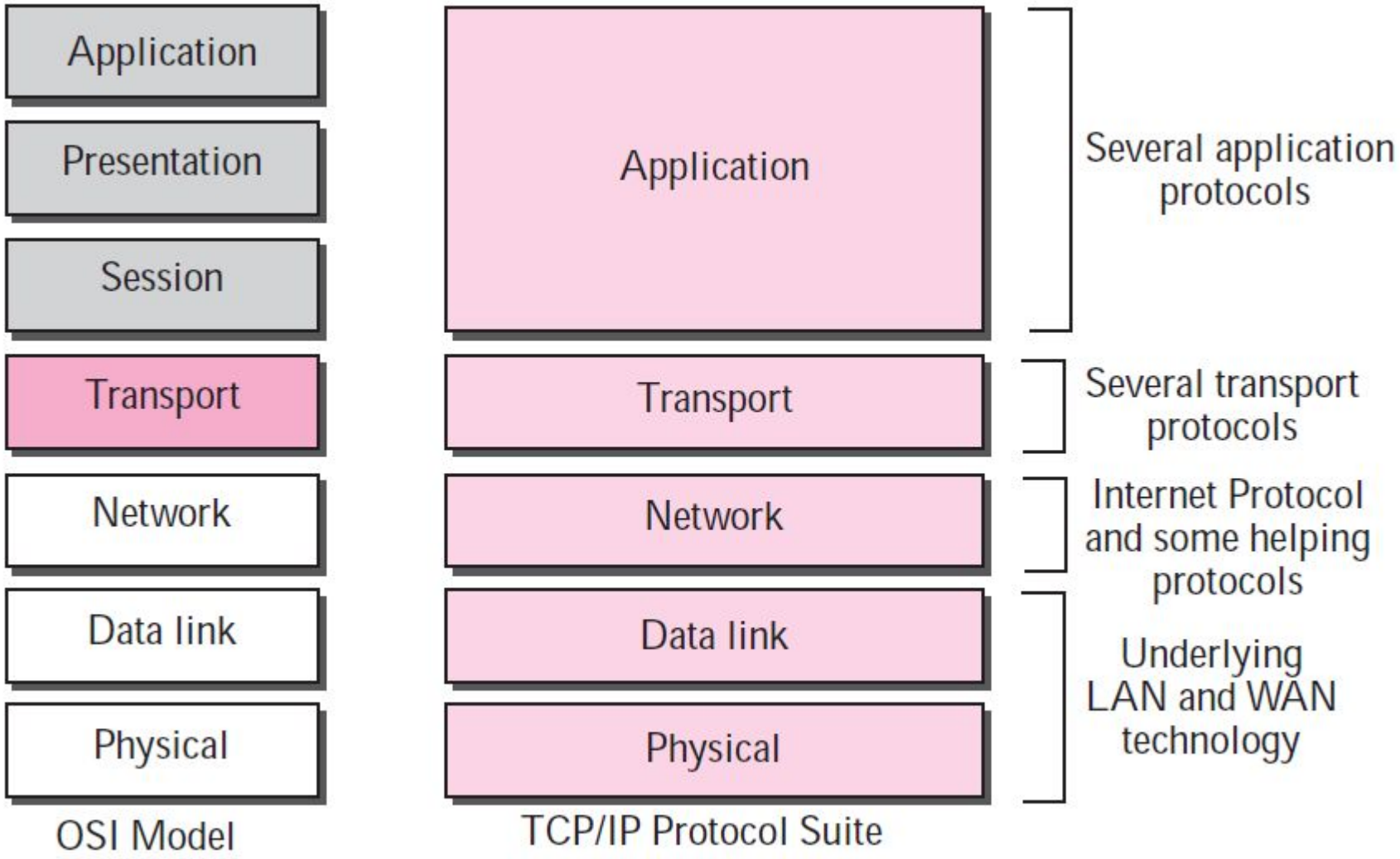
OSI reference model – Remaining layers

- *Presentation:* allow applications to interpret meaning of data, (e.g., encryption, compression, machine-specific conventions)
- *Session:* synchronization, checkpointing, recovery of data exchange
- Internet stack "missing" these layers!
  - these services, *if needed, the application developer* must to build that functionality into the application layer.



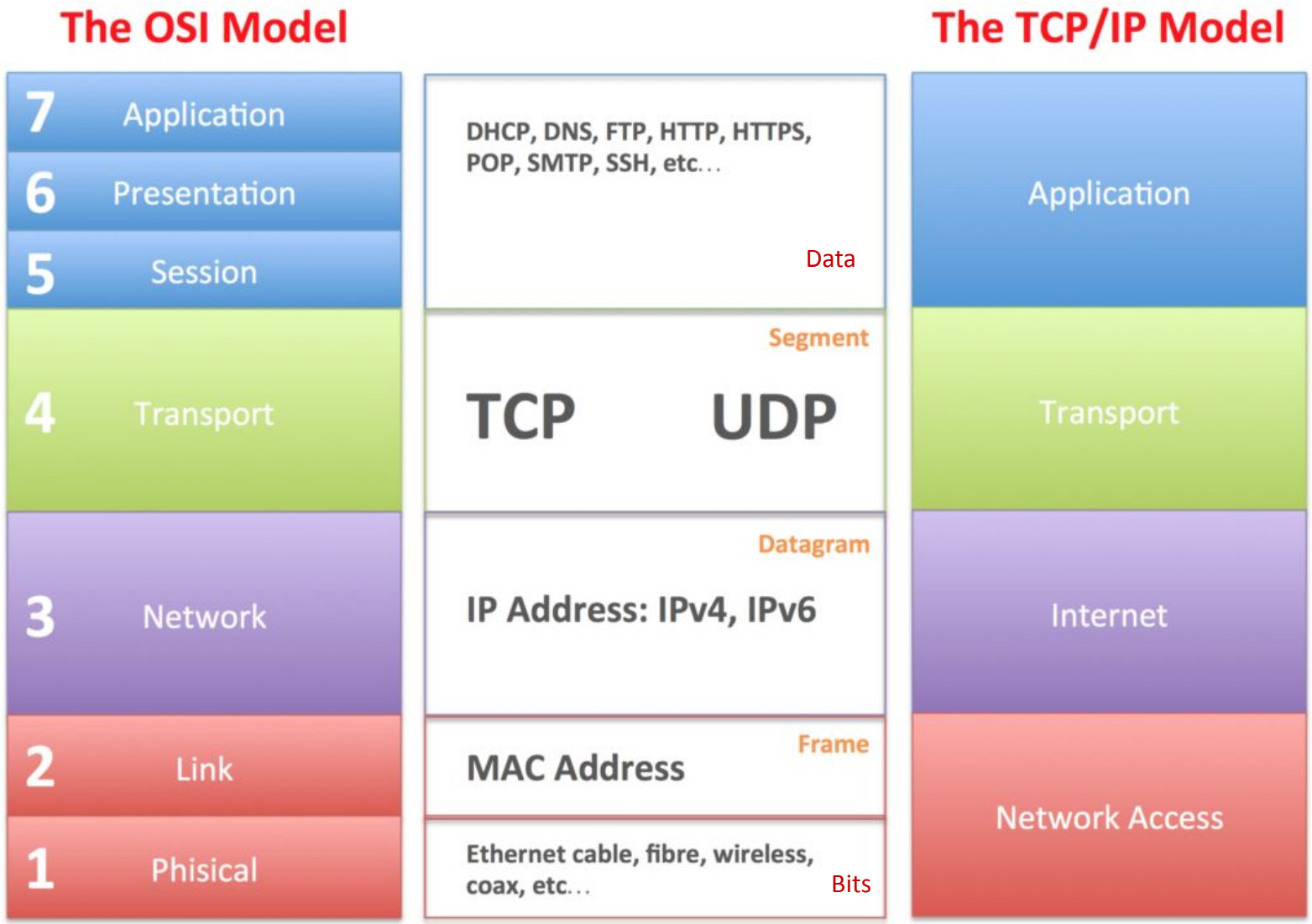
COMPUTER NETWORKS

TCP/IP vs OSI reference model



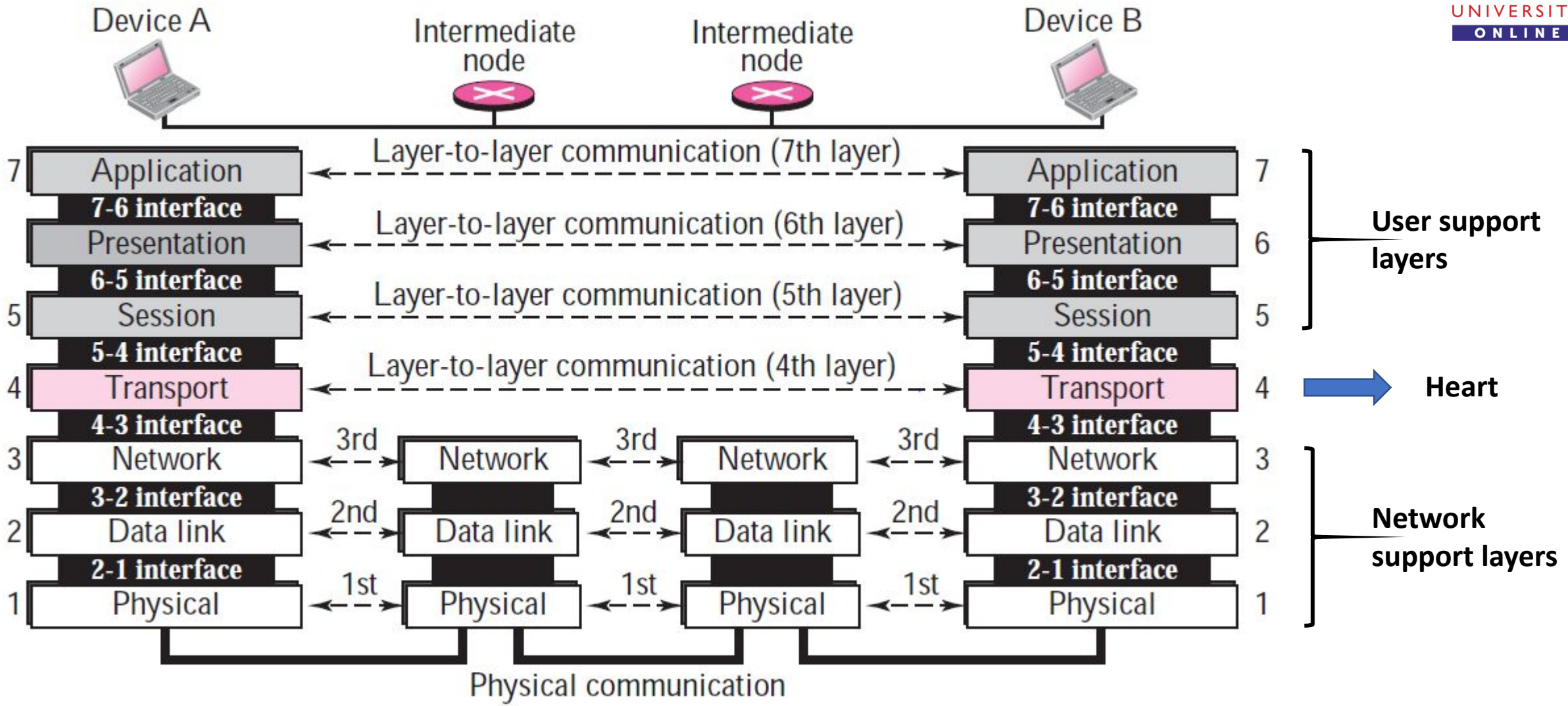
COMPUTER NETWORKS

OSI vs TCP/IP Protocol Suite

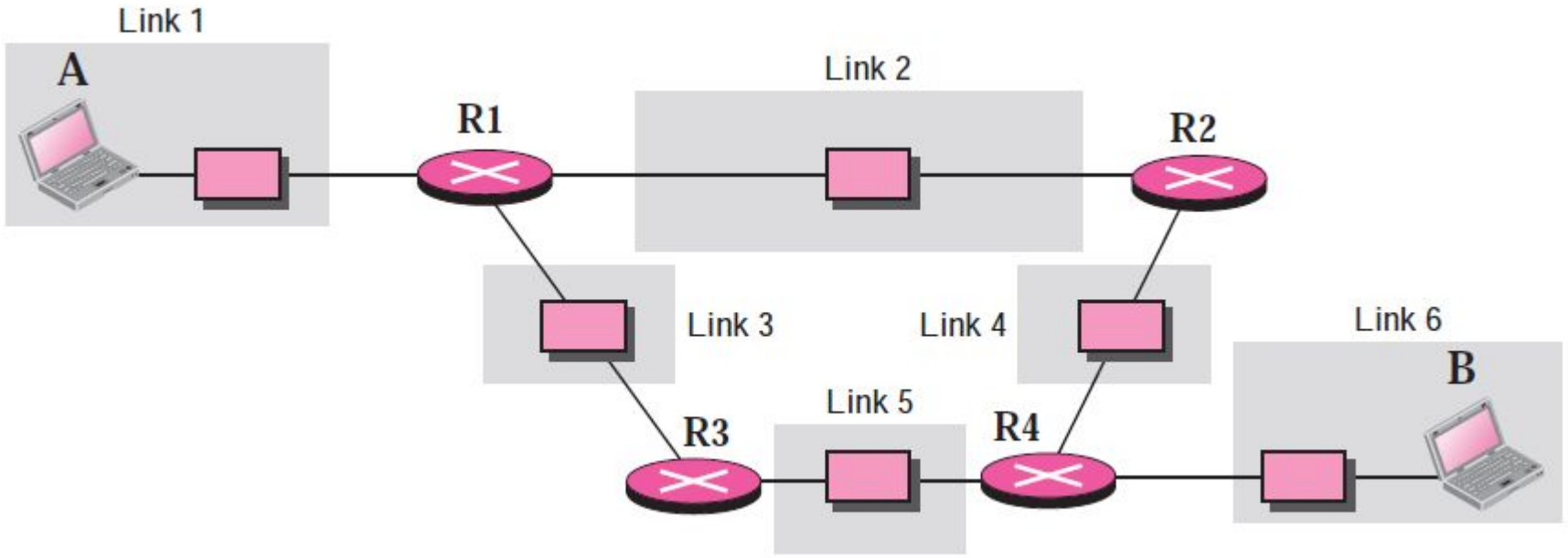


COMPUTER NETWORKS

OSI reference model (more)

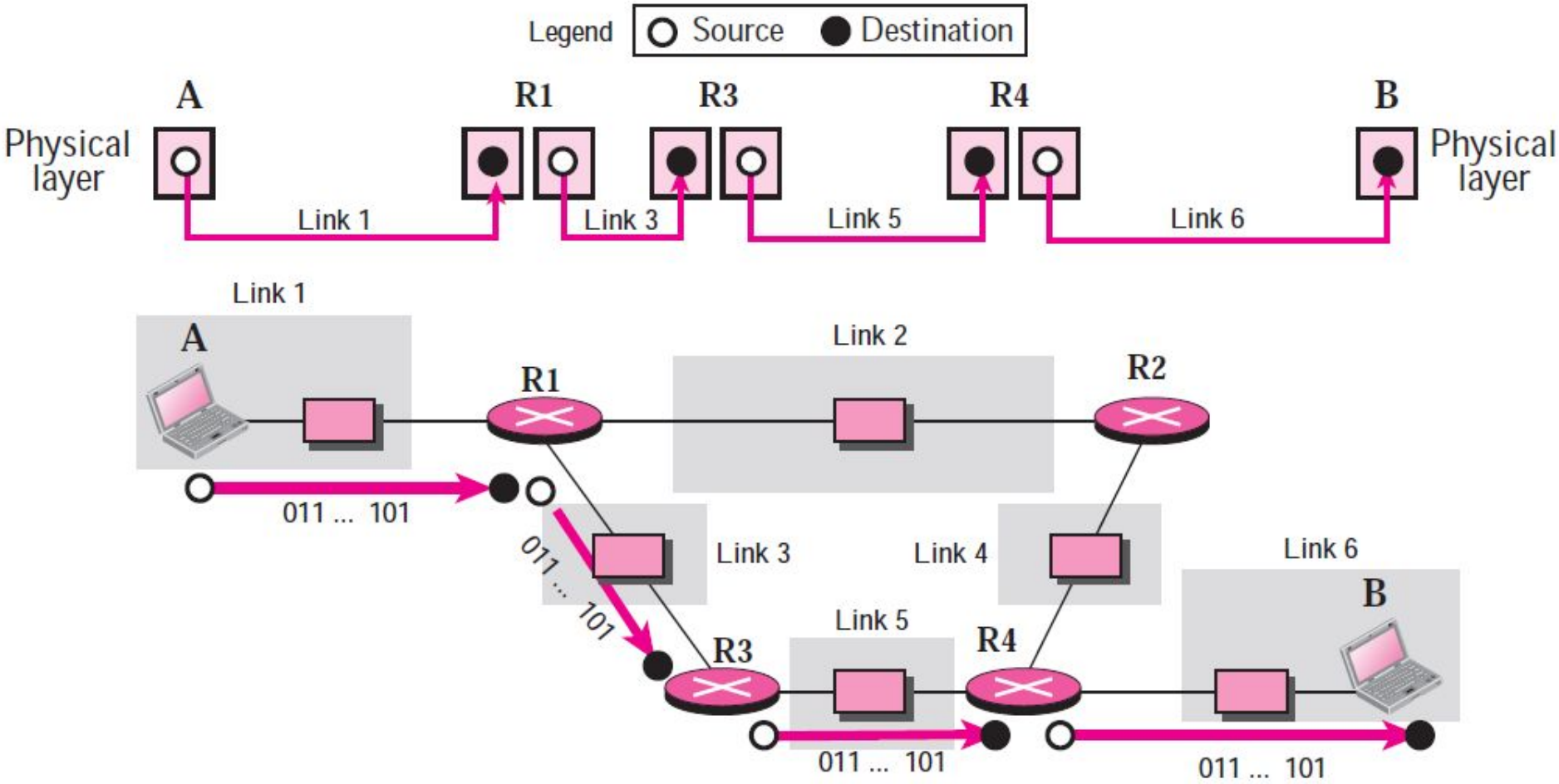


Layers in the TCP/IP Protocol Suite (more)



A private internet

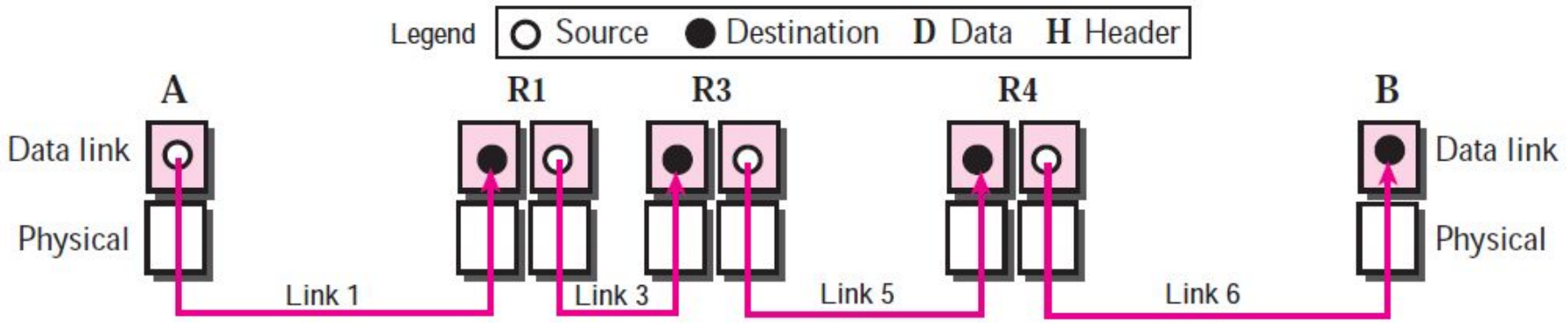
Layers in the TCP/IP Protocol Suite (more)

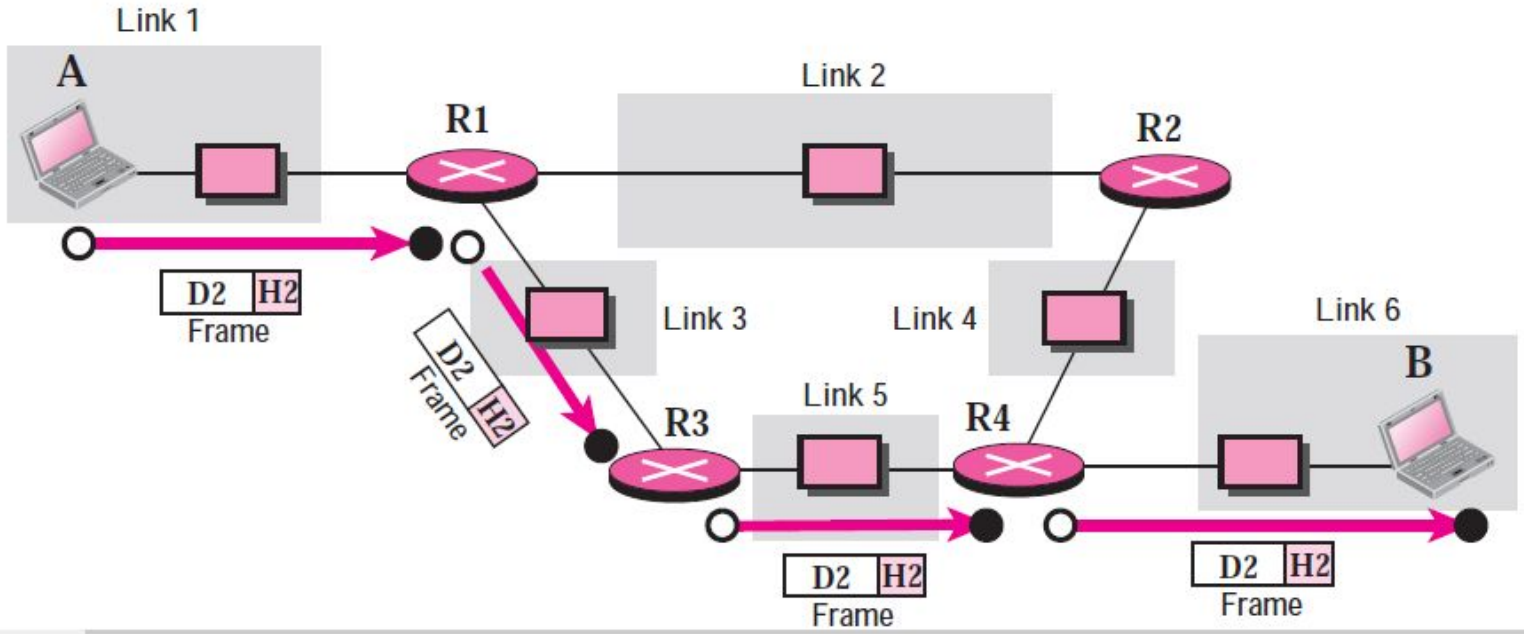


Communication at the physical layer

Unit of Communication – bit

Layers in the TCP/IP Protocol Suite (more)



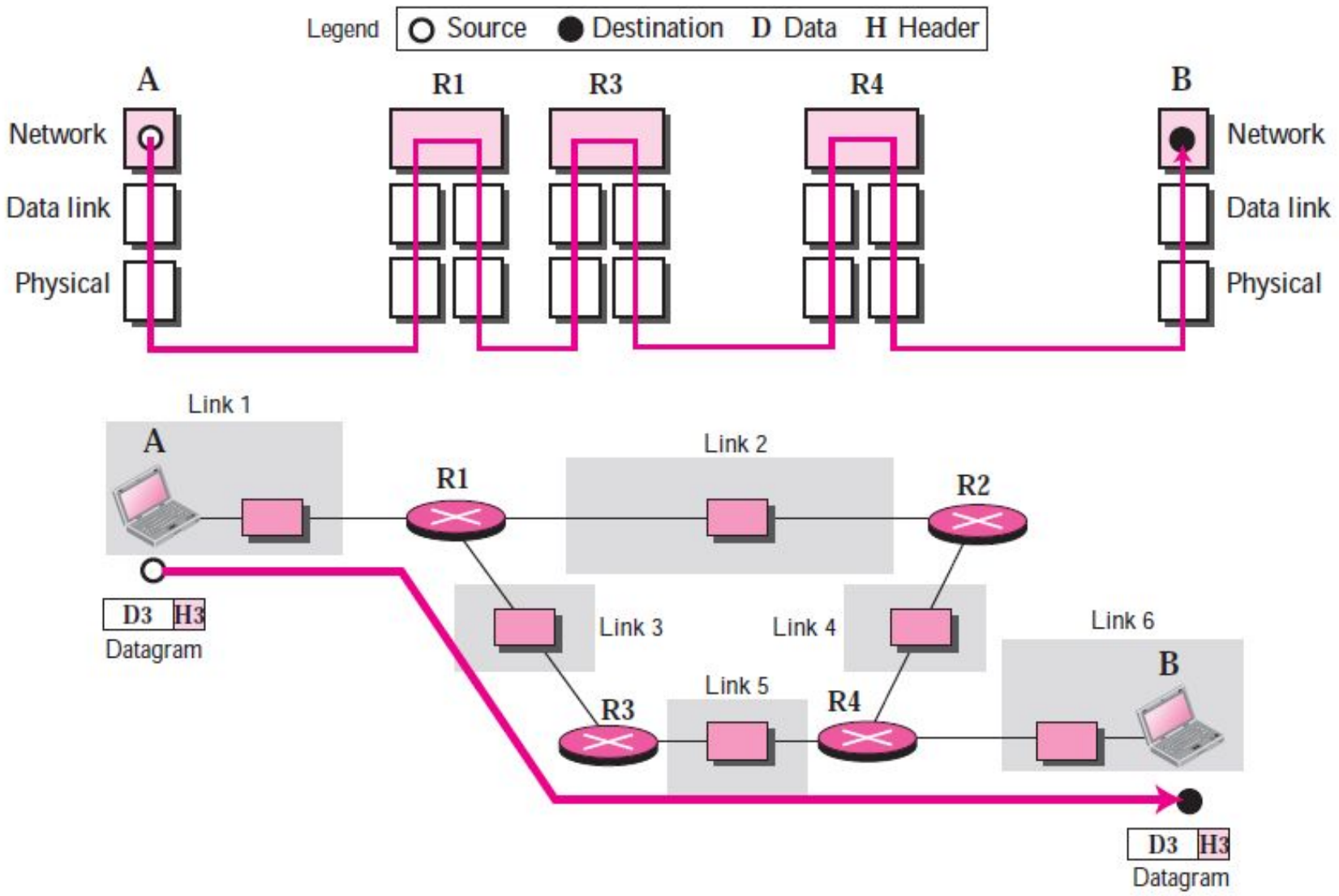


Communication at the data link layer

Unit of Communication – frame

COMPUTER NETWORKS

Layers in the TCP/IP Protocol Suite (more)

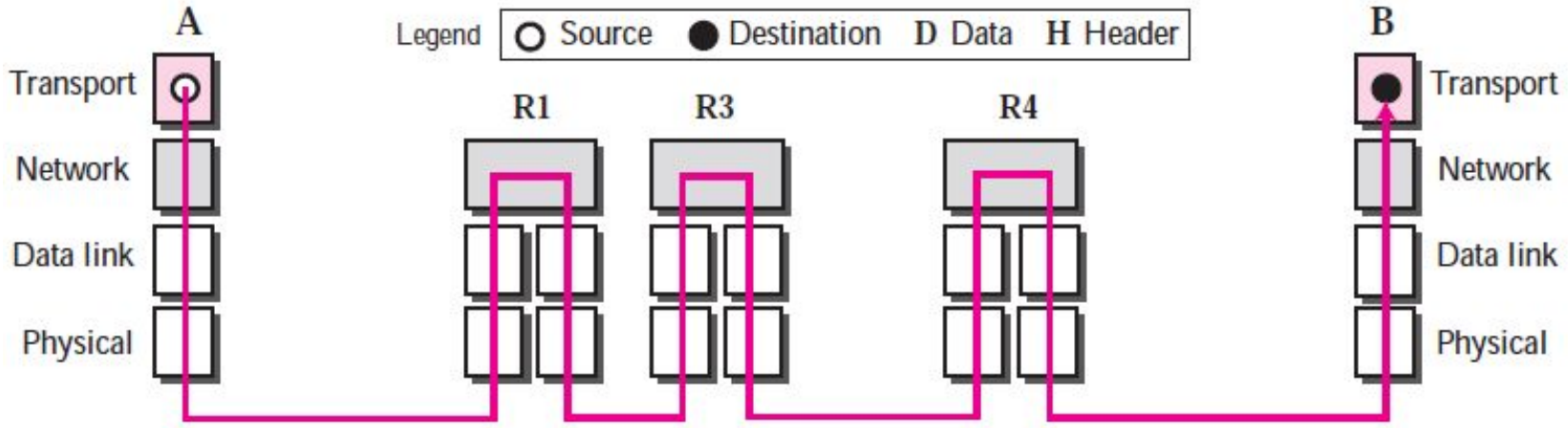


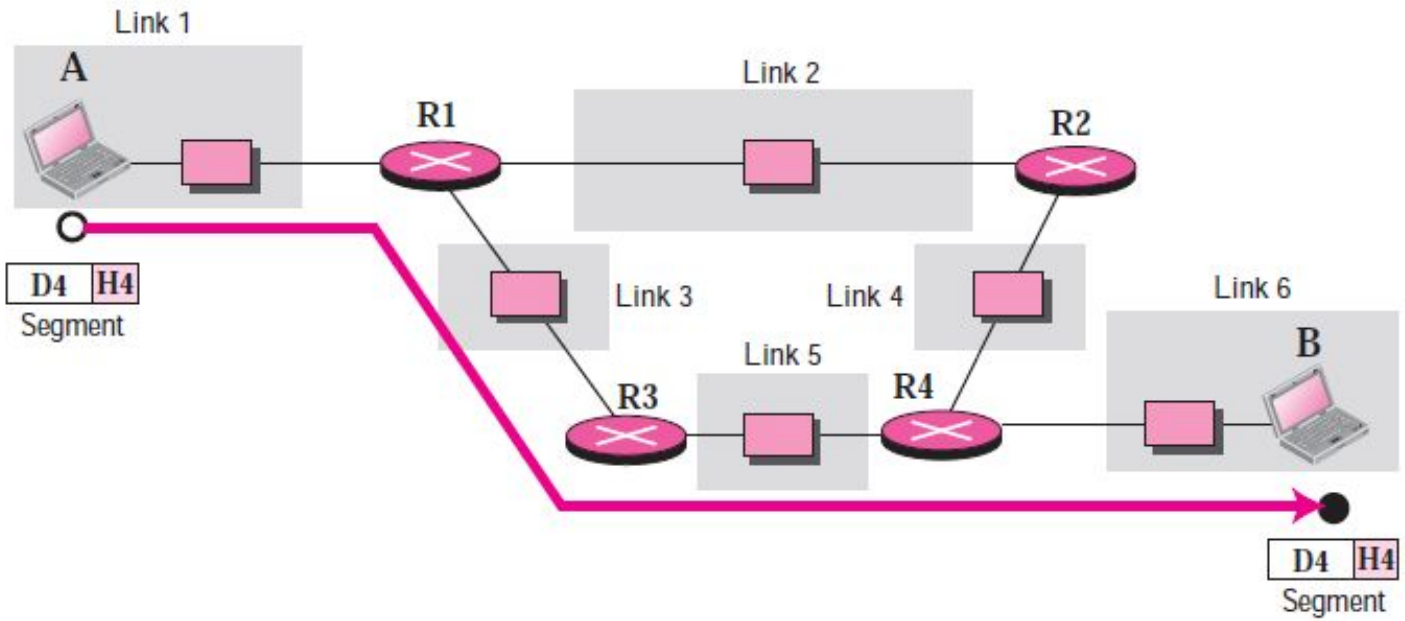
Communication at the network layer

Unit of Communication – datagram / packet

COMPUTER NETWORKS

Layers in the TCP/IP Protocol Suite (more)



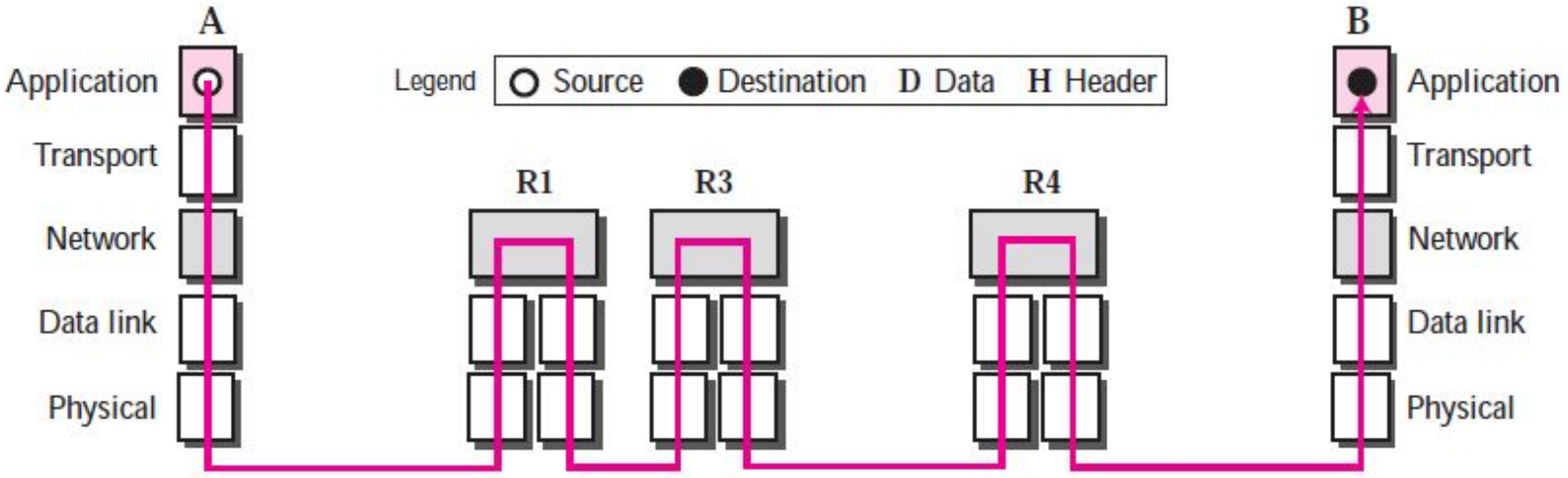


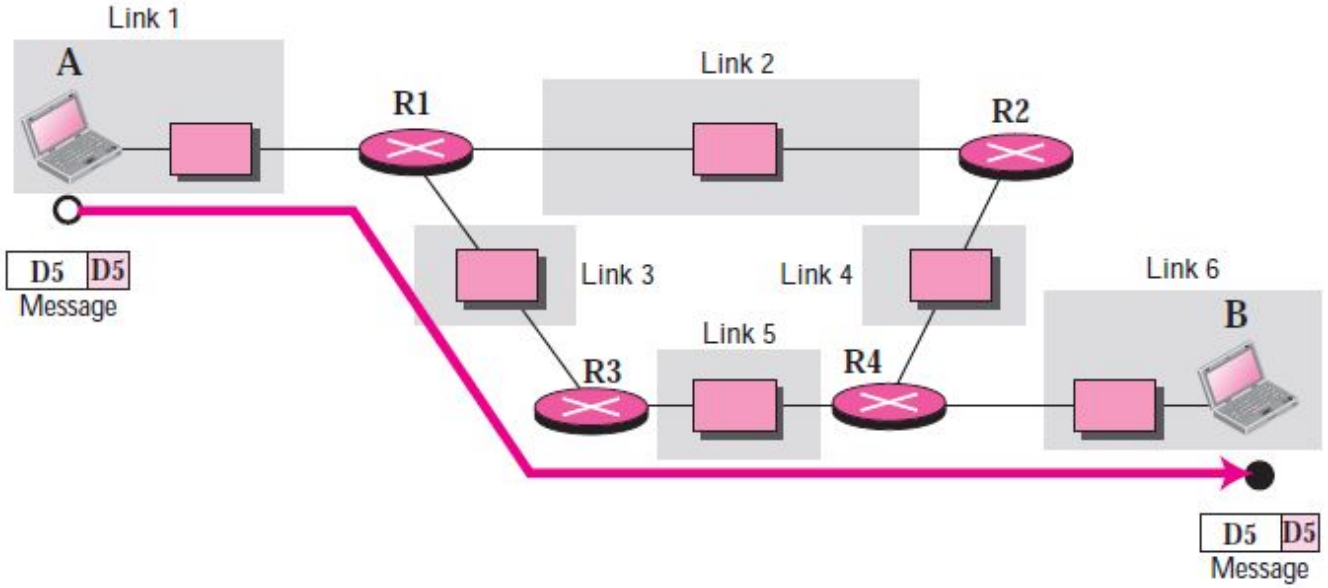
Communication at the transport layer

Unit of Communication – segment

COMPUTER NETWORKS

Layers in the TCP/IP Protocol Suite (more)

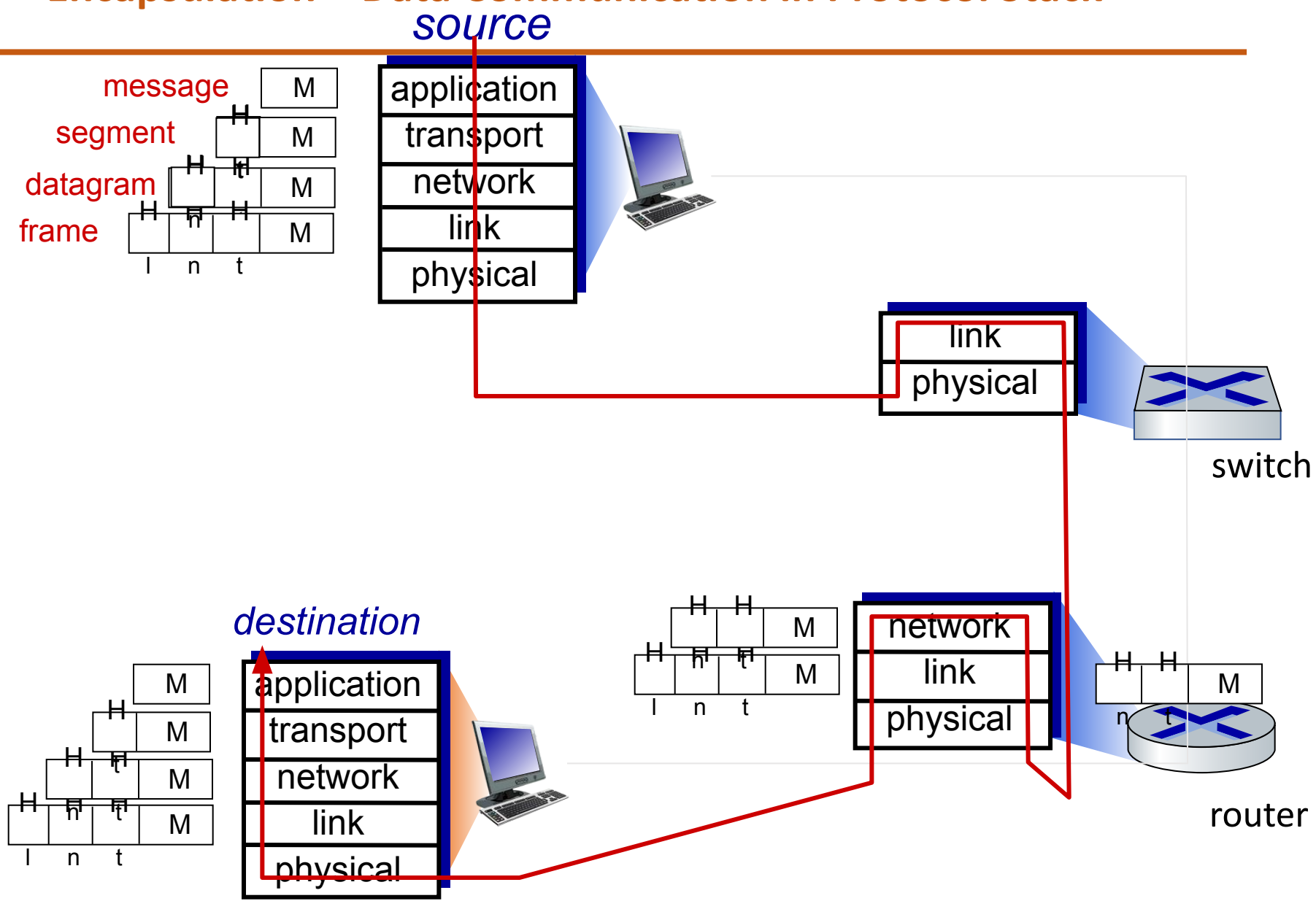




Communication at the application layer

Unit of Communication – message

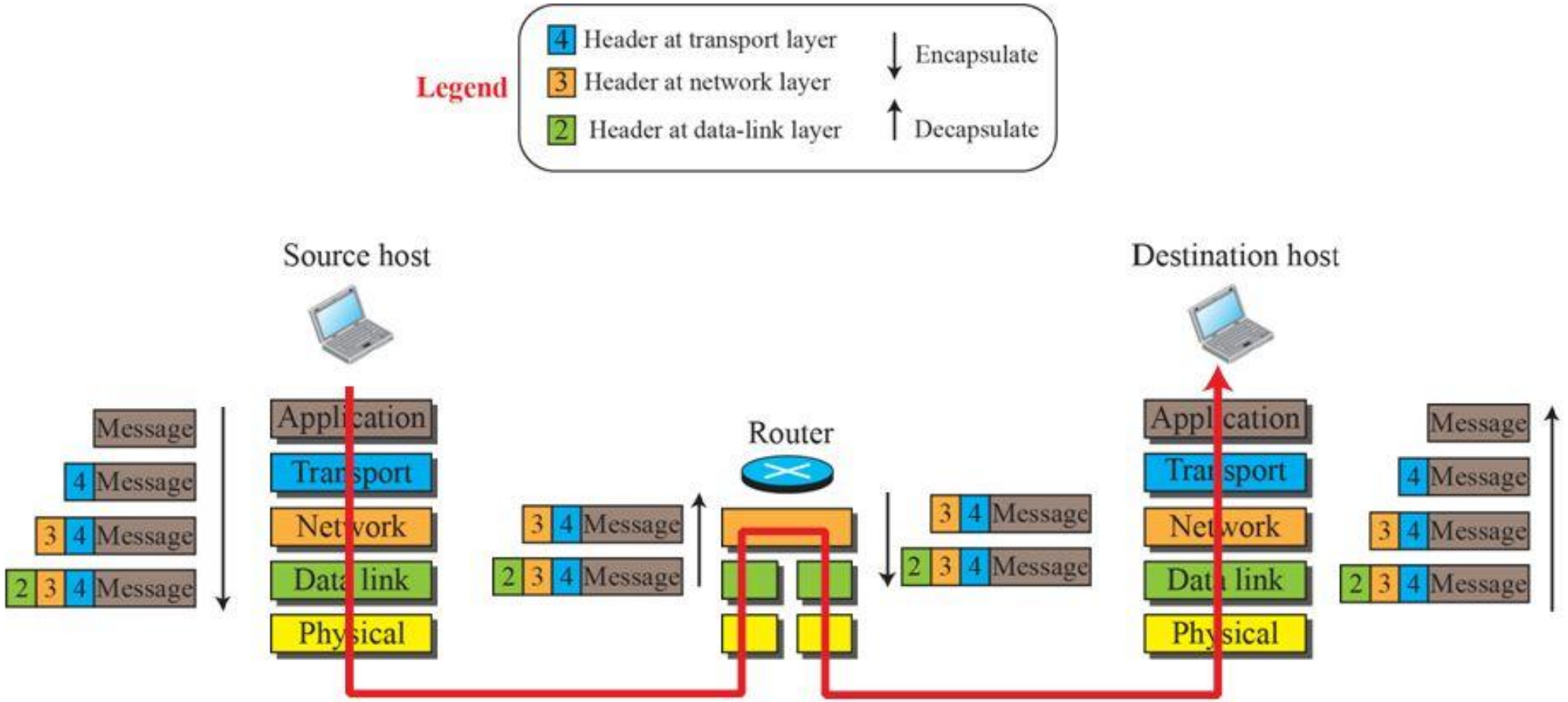
Encapsulation – Data Communication in Protocol Stack



Hosts, routers, and link-layer switches; each contains a different set of layers, reflecting their differences in functionality

COMPUTER NETWORKS

Encapsulation & Decapsulation



Encapsulation & Decapsulation

What is Encapsulation?

Encapsulation marks where a packet, or unit of data, begins and ends.

The beginning part of a packet is called the *header*, and the end of a packet is called the *trailer*.

The data between the header and trailer is sometimes referred to as the *payload*.

The packet header carries information in its first few bytes to mark where the packet begins and to identify the type of information it carries.

As a packet travels from its source to a destination, different layers in a computing system contribute to the packet header.

The header info varies depending on which protocol is used, as each protocol has a defined format.

What is decapsulation?

Decapsulation is the process of removing the header and trailer information from a packet, as it moves toward its destination.

The destination device receives the data in its original form.

Computer Networks and the Internet

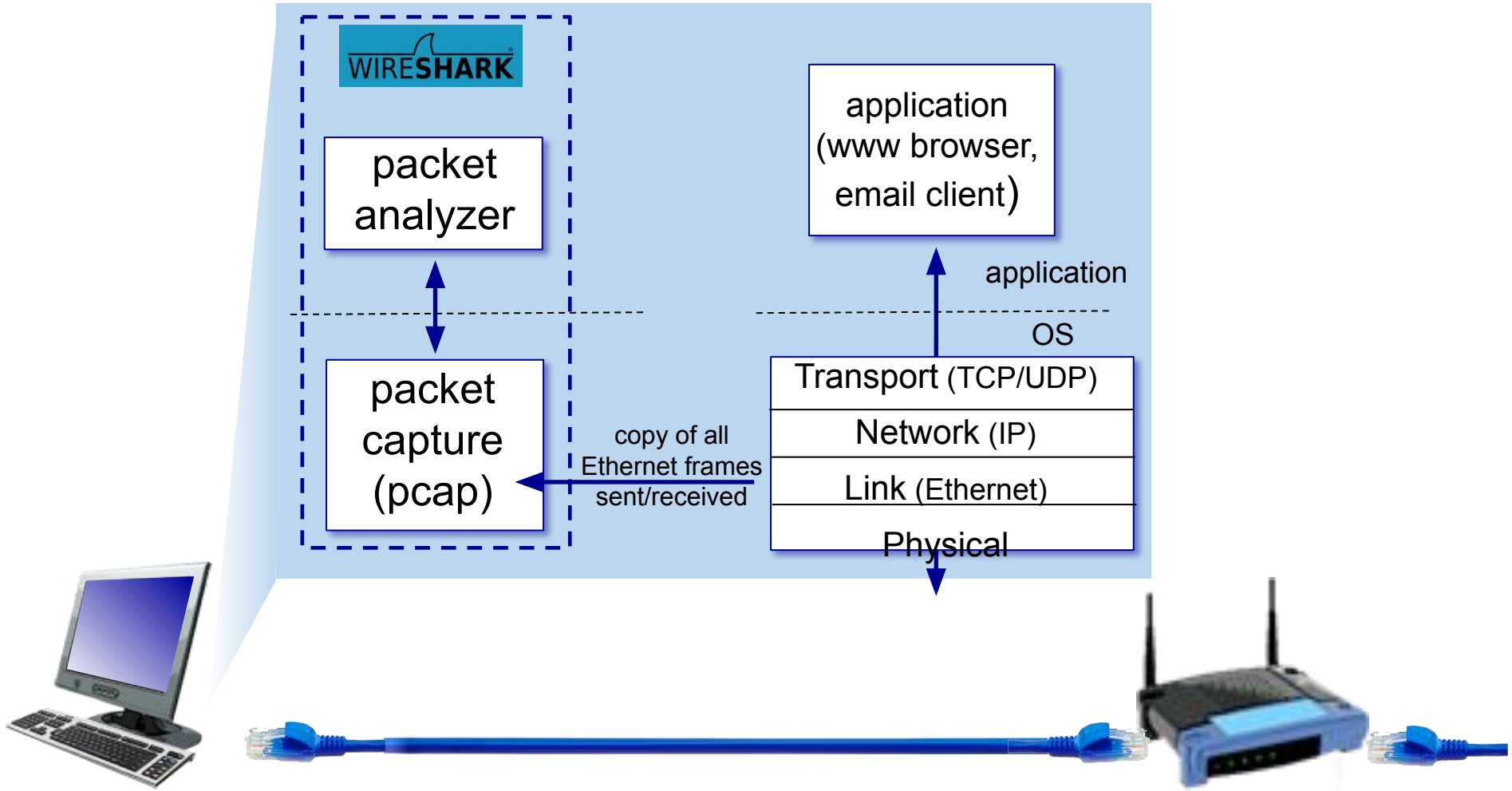
Interactive Exercises

Department of Computer Science and Engineering

For the interactive exercises follow the PDF that is uploaded onto the drive.

Also follow the below link to find similar practice questions.
https://gaia.cs.umass.edu/kurose_ross/interactive/

Wireshark


WIRESHARK
packet analyzer
packet capture (pcap)
copy of all Ethernet frames sent/received
application (www browser, email client)
application
OS
Transport (TCP/UDP)
Network (IP)
Link (Ethernet)
Physical

COMPUTER NETWORKS

Application Layer

Compiled by Kundhavai K R
Department of Computer Science and Engineering

Unit – 2 Application Layer

2.1 Principles of Network Applications

2.2 Web, HTTP and HTTPS

Unit – 2 Application Layer

Application Layer: Overview

- Numerous useful and entertaining applications have indeed been created.
- These applications have been the driving force behind the Internet's success, motivating people in homes, schools, governments, and businesses to make the Internet an integral part of their daily activities.

COMPUTER NETWORKS

Some Network Apps

- **Social networking** - Facebook, Instagram, Twitter, and
- **Web - WWW**
- **Text messaging –** WeChat, WhatsApp
- **E–mail –** Outlook, Gmail, Yahoo mail
- **Multi–user network online games** - World of Warcraft.
- **Streaming stored video** (YouTube, Hulu, Netflix)
- **P2P file sharing**
- **Traffic monitoring apps -** Waze

- **voice over IP** and **real–time video conferencing** - Skype, Facetime, and Google Hangouts
- **Internet search**
- **Remote login**
- And many more…

COMPUTER NETWORKS

Application Layer

Our goals of this unit:

- Conceptual *and* implementation aspects of application–layer protocols
  - **transport–layer service models**
  - **client–server paradigm**
  - **peer–to–peer paradigm**

- Learn about protocols by examining popular application–layer protocols
  - **HTTP**
  - **SMTP, IMAP**
  - **DNS**
- Programming network applications
  - **socket API** – in python

COMPUTER NETWORKS

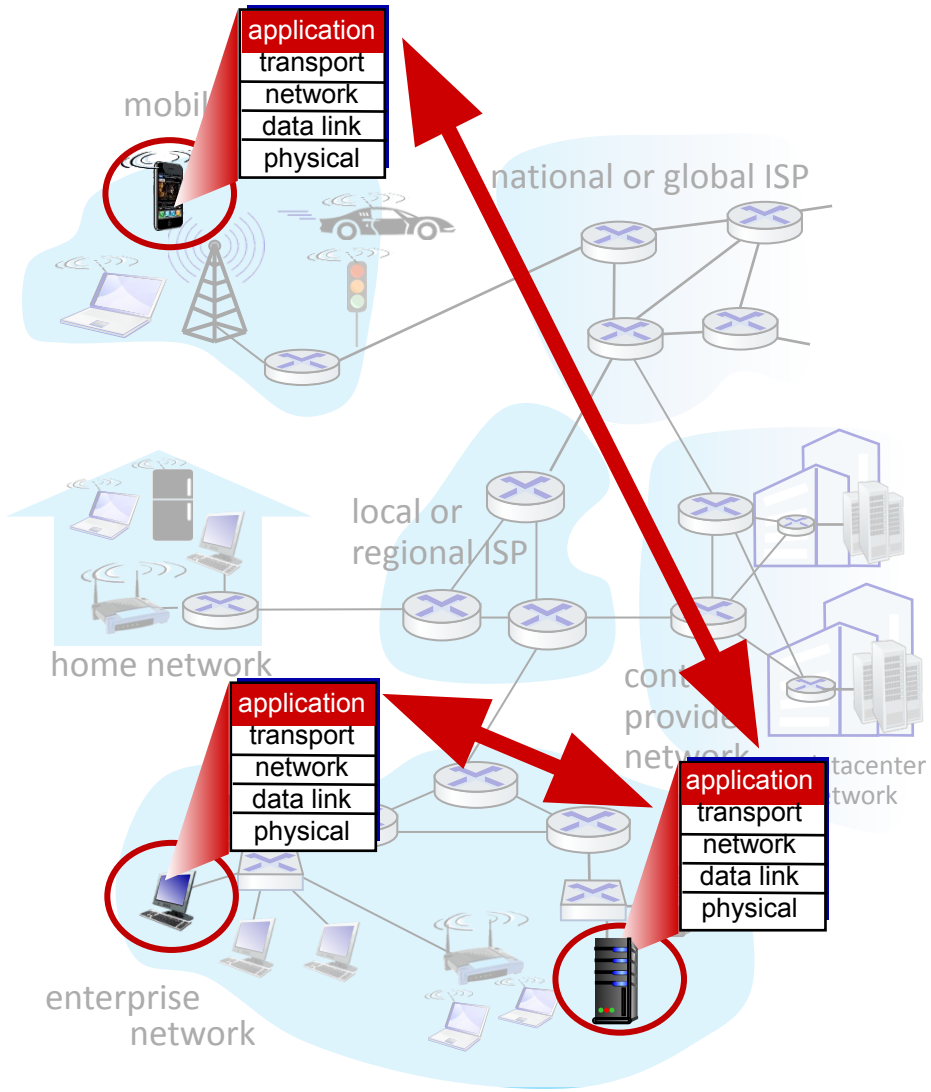
Creating a Network App

write programs that:

- run on (different) end systems
- communicate over network
- e.g., web server software communicates with browser software

no need to write software for network–core devices

- network–core devices do not run user applications
- applications on end systems allows for rapid app development, propagation



Network Application Architectures :

- The application architecture - is designed by the application developer and dictates how the application is structured over the various end systems.
- Two architectural paradigms used in modern network applications:
    - **The client-server architecture**
    - **The peer-to-peer (P2P) architecture**

COMPUTER NETWORKS

Client–Server Paradigm

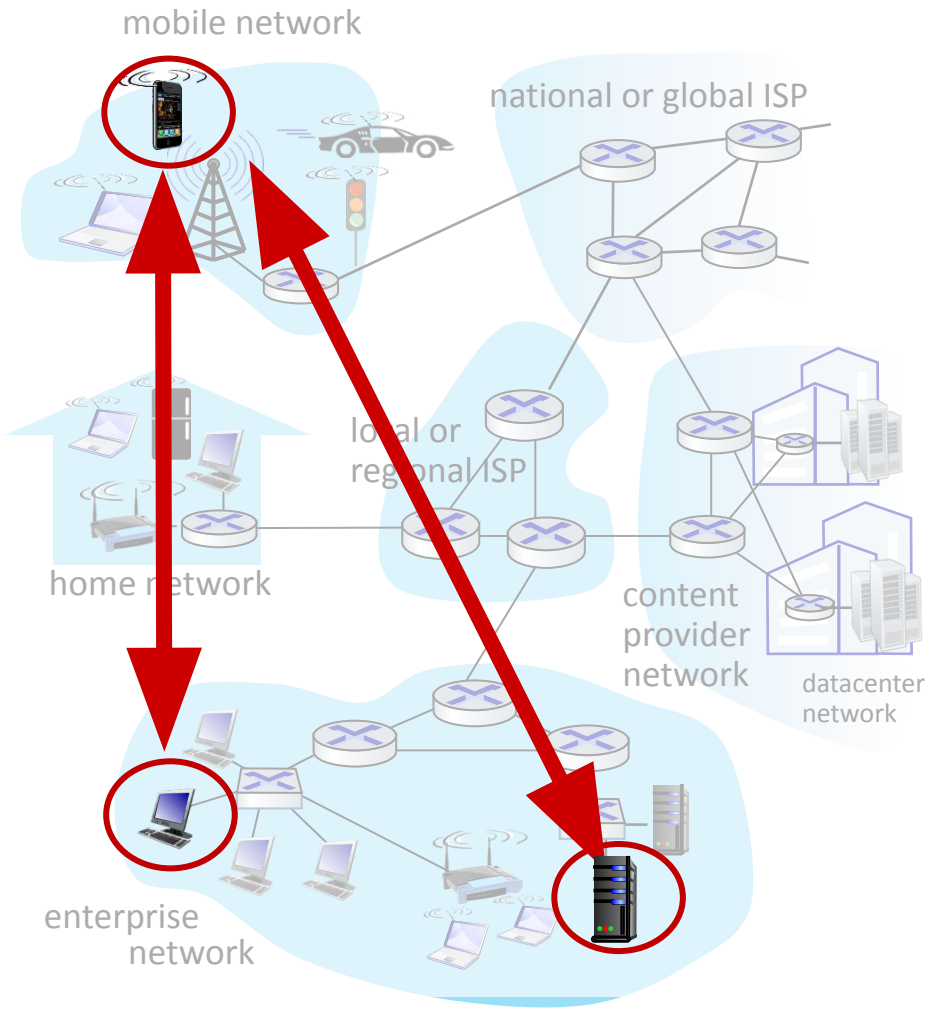
Server:

- always switched–on host
- permanent IP address
- often in data centers, for scaling

Clients:

- contact, communicate with server
- may be intermittently connected
- may have dynamic IP addresses
- do not communicate directly with each other - Web application, two browsers do not directly communicate

- Examples: HTTP, Web, FTP, Telnet, and e-mail.



COMPUTER NETWORKS

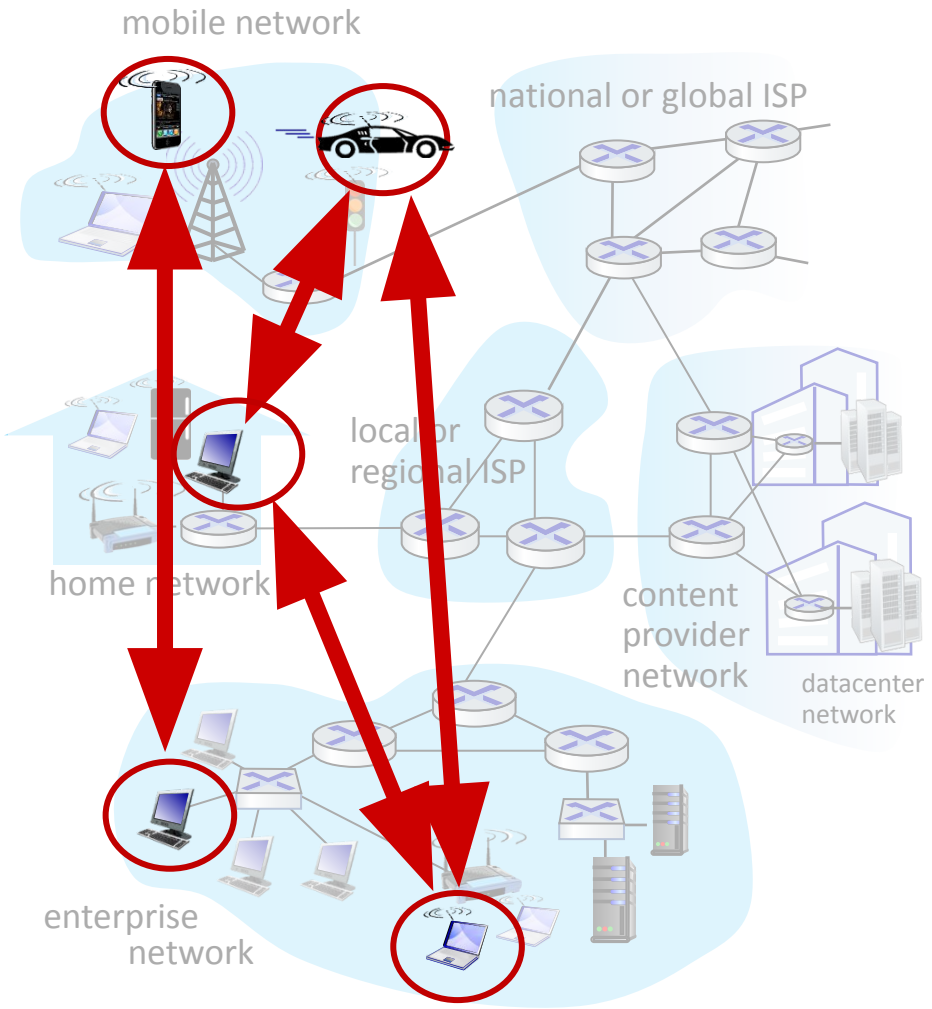
Client–Server Paradigm

- In a client-server application, a single-server host is incapable of keeping up with all the requests from clients.
- A data center, housing a large number of hosts, is often used to create a powerful virtual server.
  - **The most popular Internet services—such as search engines (e.g., Google, Bing, Baidu)**
  - **Internet commerce (e.g., Amazon, eBay, Alibaba), Web-based e-mail (e.g., Gmail and Yahoo Mail)**
  - **Social networking (e.g., Facebook, Instagram, Twitter, and WeChat)—employ one or more data centers.**
- Example: Google has 30 to 50 data centers distributed around the world, which collectively handle search, YouTube, Gmail, and other services.
- **A data center** can have hundreds of thousands of servers, which must be powered and maintained.
- The **service providers must pay recurring interconnection and bandwidth costs** for sending data from their data centers.

COMPUTER NETWORKS

Peer–to–Peer Architecture

- *not* always dependent–on server
- arbitrary end systems directly communicate
- peers request service from other peers, provide service in return to other peers
  - *self scalability* – **new peers bring new service capacity, as well as new service demands**
- peers are intermittently connected and change IP addresses
  - complex management
- example: P2P file sharing.

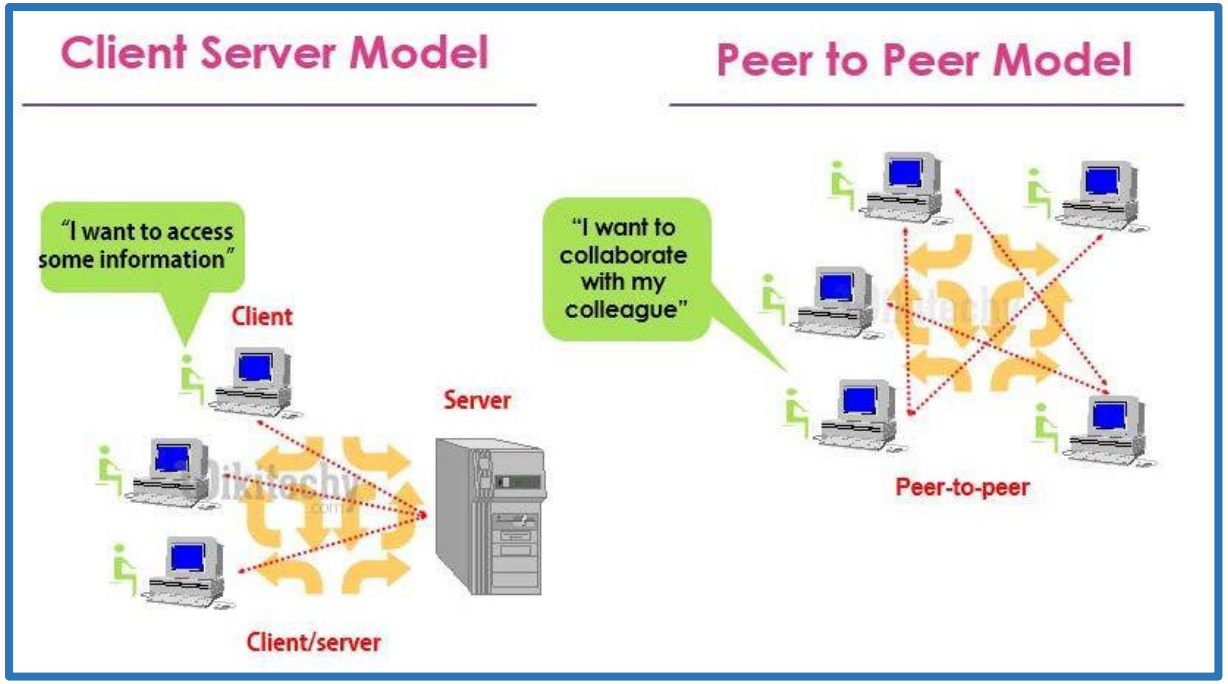


COMPUTER NETWORKS

Peer–to–Peer Architecture

- Many of today's most popular and traffic-intensive applications are based on P2P architectures.
  - These applications include file sharing (e.g., **BitTorrent)**

Disadv: P2P applications face challenges of **security, performance, and reliability** due to their highly decentralized structure.



COMPUTER NETWORKS

Peer–to–Peer Architecture vs Client Server Architecture

Aspect	Peer-to-Peer (P2P) Architecture	Client-Server Architecture
Network Organization	Decentralized	Centralized
Resource Sharing	Distributed sharing of resources among peers	Centralized resource hosting on servers
Centralization	No single point of control	Server-centric model
Control and Security	Peers have more control over shared resources	Centralized control and security measures
Scalability and Performance	Scalability achieved through adding peers	Centralized management for consistent performance

COMPUTER NETWORKS

Processes Communicating : Basic understanding

process: program running within a host

- within same host, two processes communicate using inter–process communication (defined by OS)
- processes in different hosts communicate by exchanging messages

clients, servers

client process: process that initiates communication

server process: process that waits to be contacted

- **For example:**
- In the Web application a **client browser process** exchanges messages with a **Web server process.**
- In a P2P file-sharing system, a file is transferred from a process in one peer to a process in another peer.

- With the Web**, a browser is a client process** and a Web **server is a server process.**
- With P2P file sharing, the peer that is downloading the file is labeled as the client, and the peer that is uploading the file is labeled as the server.

Processes Communicating: Formal Definition

- *In the context of a communication session between a pair of processes, the process that initiates the communication (that is, initially contacts the other process at the beginning of the session) is labeled as the **client**.*
- *The process that waits to be contacted to begin the session is the **server**.*

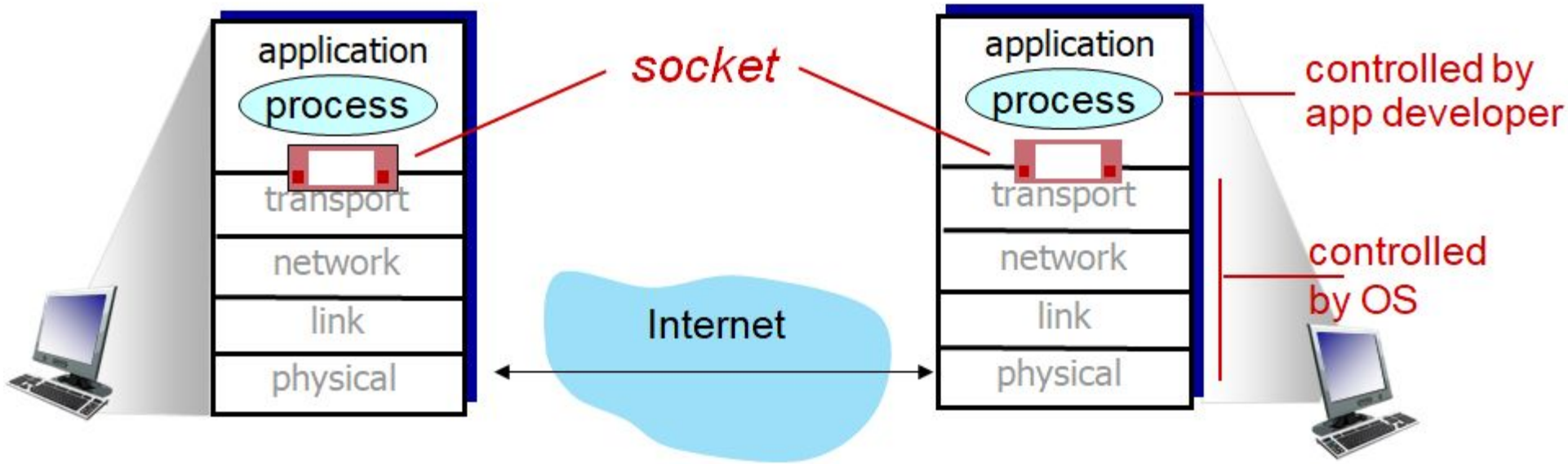
SOCKETS

Sockets

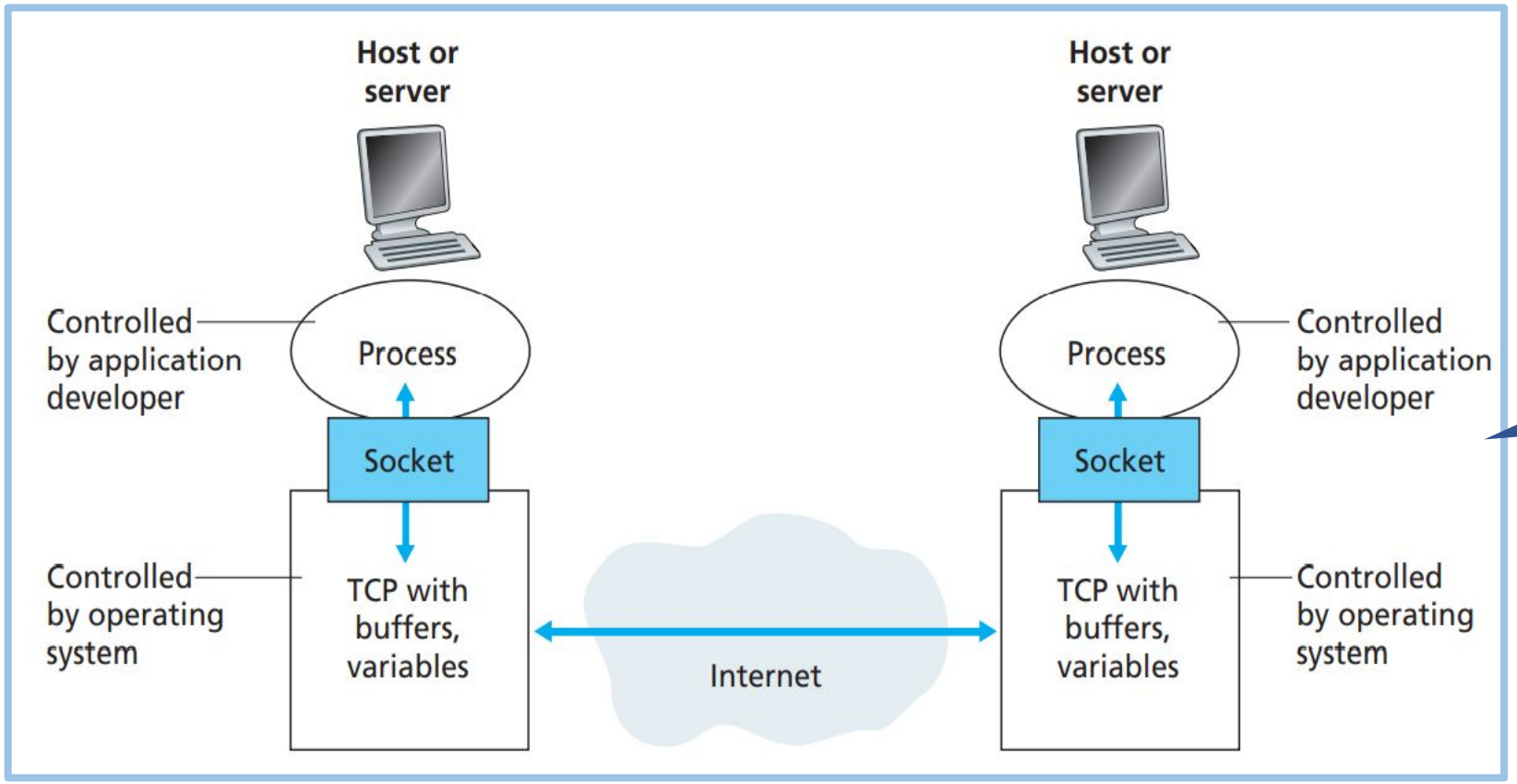
The Interface Between the Process and the Computer Network:

- A process sends messages into, and receives messages from, the network through a **software** interface called a **socket.**
- Socket analogous to door
  - ✔ sending process pushes the message out door
  - ✔ sending process relies on transport infrastructure on other side of door to deliver message to socket at receiving process
  - ✔ **Two sockets involved**: one on each side

Sockets


application
process
transport
network
link
physical
socket
Internet
application
process
transport
network
link
physical
controlled by
app developer
controlled
by OS

Sockets



Application processes, sockets, and underlying transport protocol

Sockets - Summary

- **Socket** is the **interface** between the **application layer** and the **transport layer** within a host.
- Aka **- Application Programming Interface (API).**
- The application developer has control of everything on the application-layer side of the socket but has little control of the transport-layer side of the socket.
- The control that the application developer has on the transport-layer side is
  - The choice of transport protocol and
  - The ability to fix a few transport-layer parameters such as maximum buffer and maximum segment sizes

PES UNIVERSITY ONLINE

ADDRESSING PROCESSES

COMPUTER NETWORKS

Addressing Processes

In order for a process running on one host to send packets to a process running on another host, the receiving process needs to have an address.

To identify the receiving process, two pieces of information need to be specified:
(1) the address of the host and
(2) an identifier that specifies the receiving process in the destination host.

- In the Internet, the host is identified by its ***IP address***.
- An IP address is a 32–bit quantity that can be thought of as uniquely identifying the host.
- In addition to knowing the address of the host to which a message is destined, the sending process must also identify the receiving process (more specifically, the receiving socket) running in the host.
- This information is needed because in general a host could be running many network applications.

Addressing Processes

- A destination ***port number*** serves this purpose.
- Popular applications have been assigned specific port numbers.
- For example, a Web server is identified by port number 80.
- A mail server process (using the SMTP protocol) is identified by port number 25.
- A list of well–known port numbers for all Internet standard protocols can be found at **www.iana.org**.

SOCKET = IP ADDRESS + PORT NUMBER

COMPUTER NETWORKS

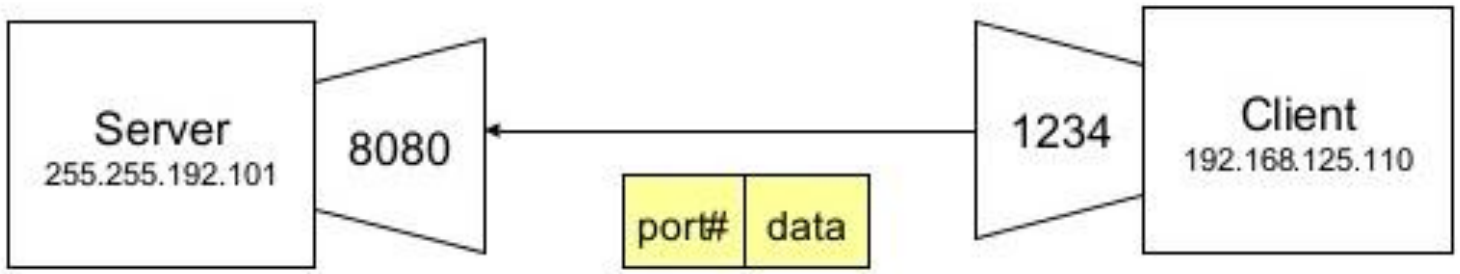
Addressing Processes

- to receive messages, process must have *identifier*
- host device has unique 32–bit IP address
- *Q:* does IP address of host on which process runs suffice for identifying the process?
- *A:* no, *many* processes can be running on same host

- *identifier* includes both IP address and port numbers associated with process on host
- example port numbers:
  - HTTP server: 80
  - mail server: 25
- to send HTTP message to gaia.cs.umass.edu web server:
  - IP address: 128.119.245.12
  - port number: 80

Addressing Processes

Server and Client have network end points called sockets. Sockets are bound to a specific port.



Socket = IP+ Port

Server Socket = 255.255.192.101:8080

Client Socket = 192.168.125.110:1234

Default Port numbers - FEW

Protocol	Port
File Transfer (FTP)	21
Secure FTP / SSH FTP (SFTP)	22
FTP Secure (FTPS)	989 (data), 990 (command)
Trivial File Transfer Protocol (TFTP)	69
Telnet	23
Hypertext Transfer Protocol (HTTP)	80
Hypertext Transfer Protocol Secure (HTTPS)	443
Secure Copy (SCP)	22
Secure Shell (SSH)	22
Simple Mail Transport Protocol (SMTP)	25
Simple Network Management Protocol (SNMP)	160, 161, 162
NetBIOS	137 (name service), 138 (datagram), 139 (session)

COMPUTER NETWORKS

IP Address VS Port Number

IP Address	Port Number
Used to identify a host	used to identify an application/services on your system
An IP address is the address of the layer-3 IP protocol.	A port number is a layer-4 address used by some layer-4 protocols e.g. TCP and UDP
The IP address for IPv4 is of 32 bits (4 bytes) size and for ipv6 is 128 bits (16 bytes)	The Port number is 16 bits and assigned by the Network operating system when the application process creates the sockets
IP address is provided by network administrator or admin user of system	Port no. for application is decided by the Kernel of the OS. This port no. is called port address.
To find IP address, issue the command "ipconfig" under CLI and press "Enter." Your IP address should be listed under "Ethernet adapter/LAN"	To find port number used for application, Type "netstat -a" and press "Enter." A list of all your active TCP/IP connections will populate showing port number used by source and destination hosts.
IP address works to send datagram traffic across network from source machine to destination machine	After IP delivers the packet to destination , with the help of the port numbers OS directs the data to the correct application
E.g. – 192.168.0.1, 172.16.0.1 are some of IP address examples.	E.g. – Port number 80 for http traffic , 67 and 68 for DHCP traffic etc.

Sockets VS Port Number

SOCKET	PORT
An internal endpoint for sending and receiving data within a node on a computer network	A numerical value that is assigned to an application in an endpoint of communication
Works as the interface to send and receive data through a specific port	Helps to identify a specific application or a process

TRANSPORT LAYER SERVICES AVAILABLE TO APPS

Transport Layer Services for applications

- The application at the sending side pushes messages through the socket.
- At the other side of the socket, the transport-layer protocol has the responsibility of getting the messages to the socket of the receiving process.
- When you develop an application, you must choose one of the available transport-layer protocols.
- **How do you make this choice?**
    - **Look at the services provided by the available transport-layer protocols, and then pick the protocol with the services that best match your application's needs**.
- **Services that a transport-layer protocol can offer to applications invoking it are:**
    - **Reliable data transfer**
    - **Throughput**
    - **Timing**
    - **Security.**

COMPUTER NETWORKS

Services by transport layer to the applications

Reliable data transfer:

- some apps (e.g., file transfer, web transactions) require 100% reliable data transfer – can't afford to loose packets
- acceptable for loss-tolerant applications like multimedia applications such as conversational audio/video can tolerate some loss
- Transport layer provides
  - **Reliable and Unreliable**
  - **Based on the needs the developer can pick one.**

Throughput:

- some apps (e.g., multimedia) require minimum amount of throughput to be "effective" (**bandwidth–sensitive apps**)
- other apps ("**elastic apps** - Electronic mail, file transfer, and Web transfers ") make use of whatever throughput they get

Transport Layer Services for applications

Timing

- Some apps (e.g., Internet telephony, interactive games) require low delay to be "effective"
- An example guarantee might be that every bit that the sender pumps into the socket arrives at the receiver's socket no more than 100 msec later. Such a service would be appealing to interactive real-time applications, such as Internet telephony, virtual environments, teleconferencing, and multiplayer games, all of which require tight timing constraints on data delivery in order to be effective.

Security

- Encryption - Confidentiality between the two processes
- Data integrity
- End-point authentication

Transport service requirements of some applications

Application	Data Loss	Throughput	Time-Sensitive
File transfer/download	No loss	Elastic	No
E-mail	No loss	Elastic	No
Web documents	No loss	Elastic (few kbps)	No
Internet telephony/ Video conferencing	Loss-tolerant	Audio: few kbps–1 Mbps Video: 10 kbps–5 Mbps	Yes: 100s of msec
Streaming stored audio/video	Loss-tolerant	Same as above	Yes: few seconds
Interactive games	Loss-tolerant	Few kbps–10 kbps	Yes: 100s of msec
Smartphone messaging	No loss	Elastic	Yes and no

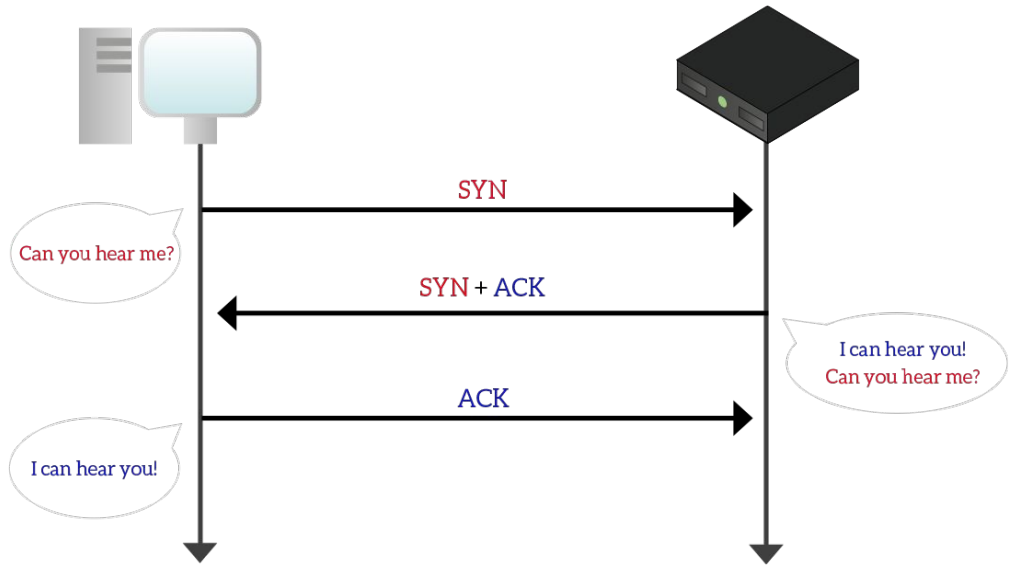
TRANSPORT SERVICES PROVIDED BY THE INTERNET

Internet Transport Protocol Services

TCP service:

- *reliable transport* between sending and receiving process
- *flow control:* sender won't overwhelm receiver
- *congestion control:* throttle sender when network overloaded
- *Connection–oriented:* setup required between client and server processes – FULL DUPLEX
- Data sent **without error** and in **proper order**.
- Provides **TCP – SSL** (Secure socket layer) for security purpose.

TCP - Three way handshake

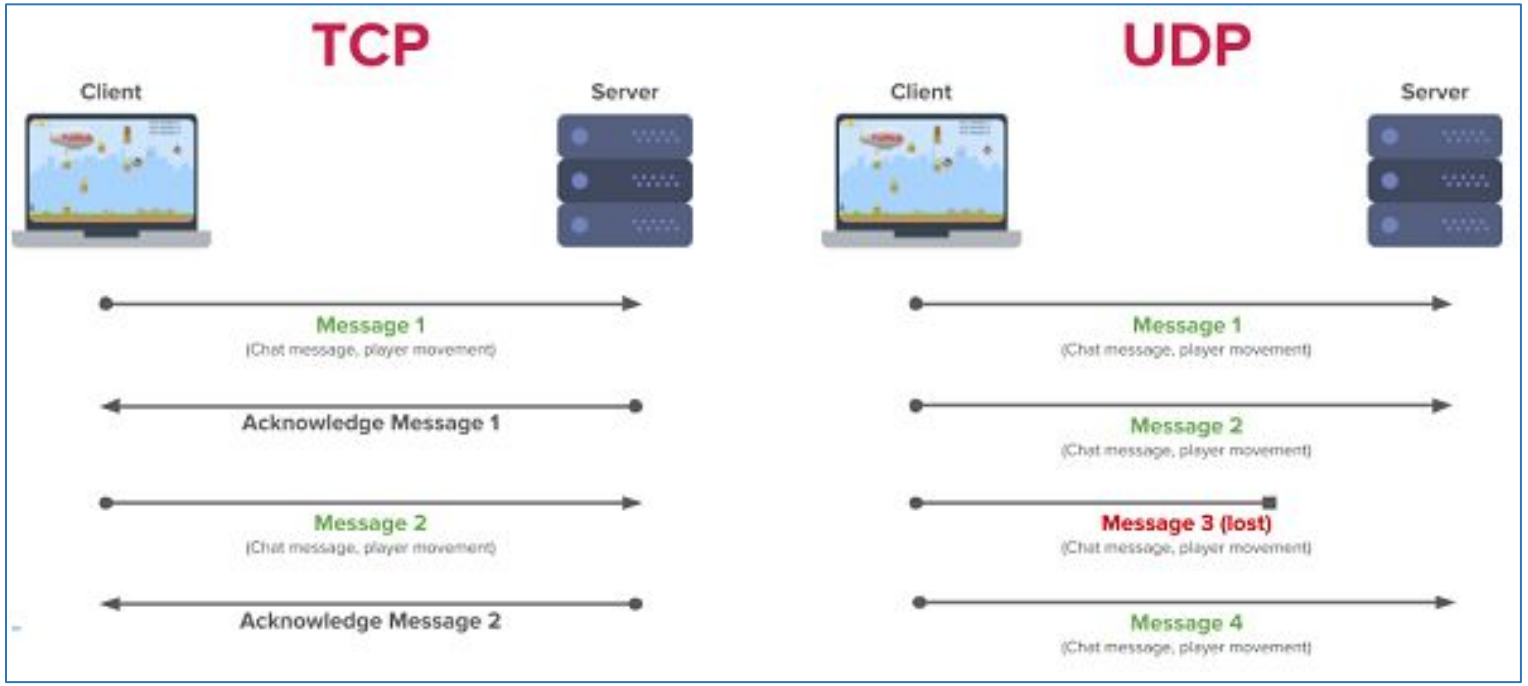


COMPUTER NETWORKS

Internet Transport Protocol Services

UDP service:

- *unreliable data transfer* between sending and receiving process
- UDP is Connectionless.
- *does not provide:* reliability, flow control, congestion control, timing, throughput guarantee, security, or connection setup.



	Reliable	Best-Effort
Connection Type	Connection-oriented	Connectionless
Protocol	TCP	UDP
Sequencing	Yes	No
Uses	▪ E-mail ▪ File sharing ▪ Downloading	▪ Voice streaming ▪ Video streaming

Internet Transport Protocol Services

Application	Application-Layer Protocol	Underlying Transport Protocol
Electronic mail	SMTP [RFC 5321]	TCP
Remote terminal access	Telnet [RFC 854]	TCP
Web	HTTP 1.1 [RFC 7230]	TCP
File transfer	FTP [RFC 959]	TCP
Streaming multimedia	HTTP (e.g., YouTube), DASH	TCP
Internet telephony	SIP [RFC 3261], RTP [RFC 3550], or proprietary (e.g., Skype)	UDP or TCP

An Application–layer Protocol defines:

- **The types of messages exchanged**
  - Example - request messages and response messages
- **The syntax of the various message types**
  - Such as - fields in the message and how the fields are defined.
- **The semantics of the fields**
  - The meaning of the information in the fields
- **Rules for determining when and how a process sends messages** and responds to messages.

- Some application-layer protocols are specified in RFCs and are therefore in the **public domain.**
  - Example – **HTTP, FTP, Telnet, DNS**
- Many other application-layer protocols are proprietary and **intentionally not available in the public domain.**
  - Example, **Skype** uses proprietary application-layer protocols.

The Web and HTTP

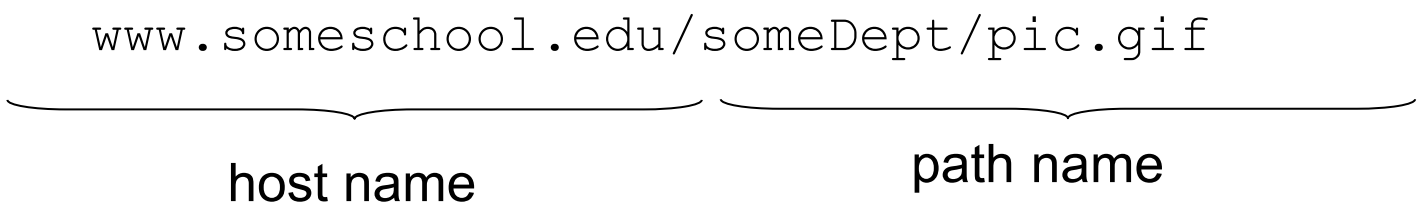
Web, HTTP and HTTPS

- Web operates on **demand**.
- Users receive **what they want, when they want it.**
- Easy for any individual to make information available over the Web—everyone can become a **publisher at extremely low cost.**
- In addition to being available on demand, the Web has many other wonderful features:
  - **Hyperlinks and search engines –** helps us navigate through ocean of information.
  - **Photos and videos**
  - **Forms**
  - **JavaScript**
  - Web and its protocols serve **as a platform for YouTube, Web-based e-mail (such as Gmail)**
  - **Instagram**
  - **Google Maps**
  - **Many more…..**

Web, HTTP and HTTPS

A quick review…

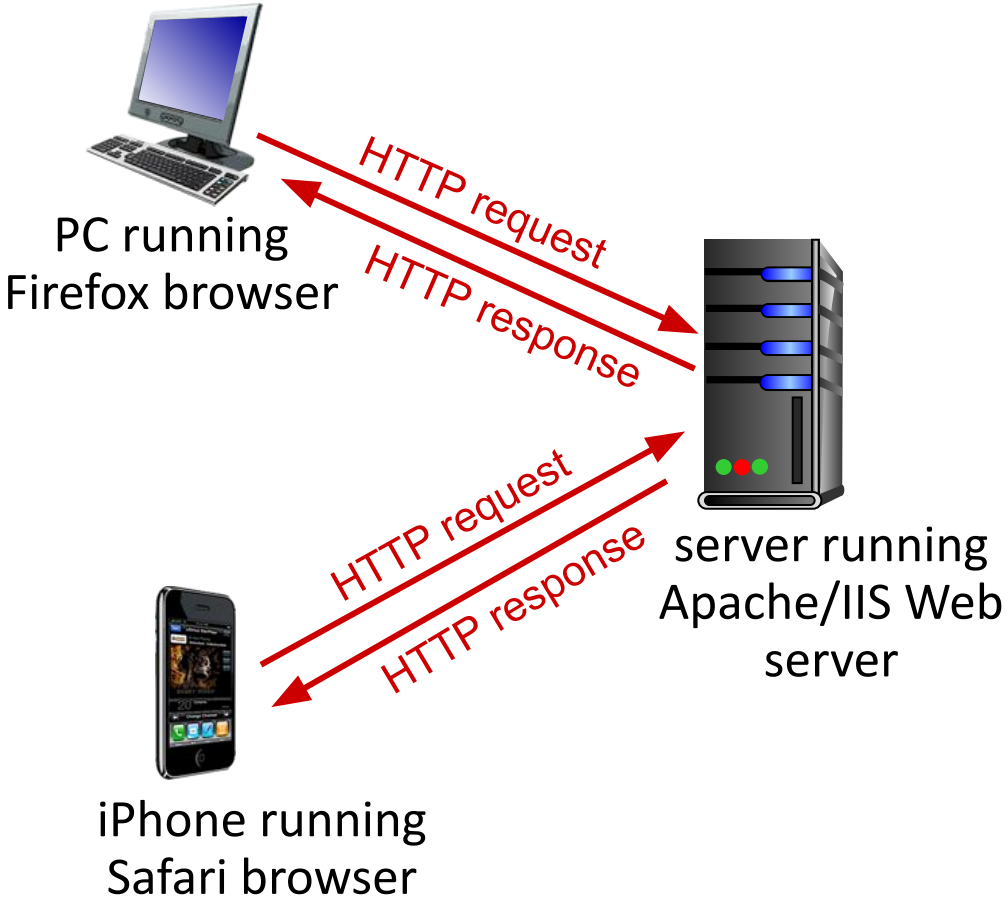
- web page consists of *objects,* each of which can be stored on different Web servers
- object can be HTML file, JPEG image, Javascrpit file, CSS stylesheet, audio file, video clip
- web page consists of *base HTML–file* which includes *several referenced objects, each* addressable by a *URL,*
- URL has two components:
  - the hostname of the server that houses the object
  - the object's path name.
- Example : If a Web page contains HTML text and 5 JPEG images, then the Web page has 6 objects: the base HTML file plus the 5 images.



HTTP Overview

HTTP: Hypertext Transfer Protocol

- Web's application layer protocol is at the heart of the Web.
- HTTP is implemented in two programs.
  - ***Client program:*** browser that requests, receives, (using HTTP protocol) and "displays" Web objects
  - ***Server program:*** Web server sends (using HTTP protocol) objects in response to requests
- They talk to each other by exchanging **HTTP messages.**



Defined in RFC 1945; RFC 2616

HTTP Overview (more)

HTTP uses TCP:

- client initiates TCP connection (creates socket) to server, port 80
- server accepts TCP connection from client
- HTTP messages (application–layer protocol messages) exchanged between browser (HTTP client) and Web server (HTTP server)
- TCP connection closed

HTTP is "stateless"

- server maintains *no* information about past client requests

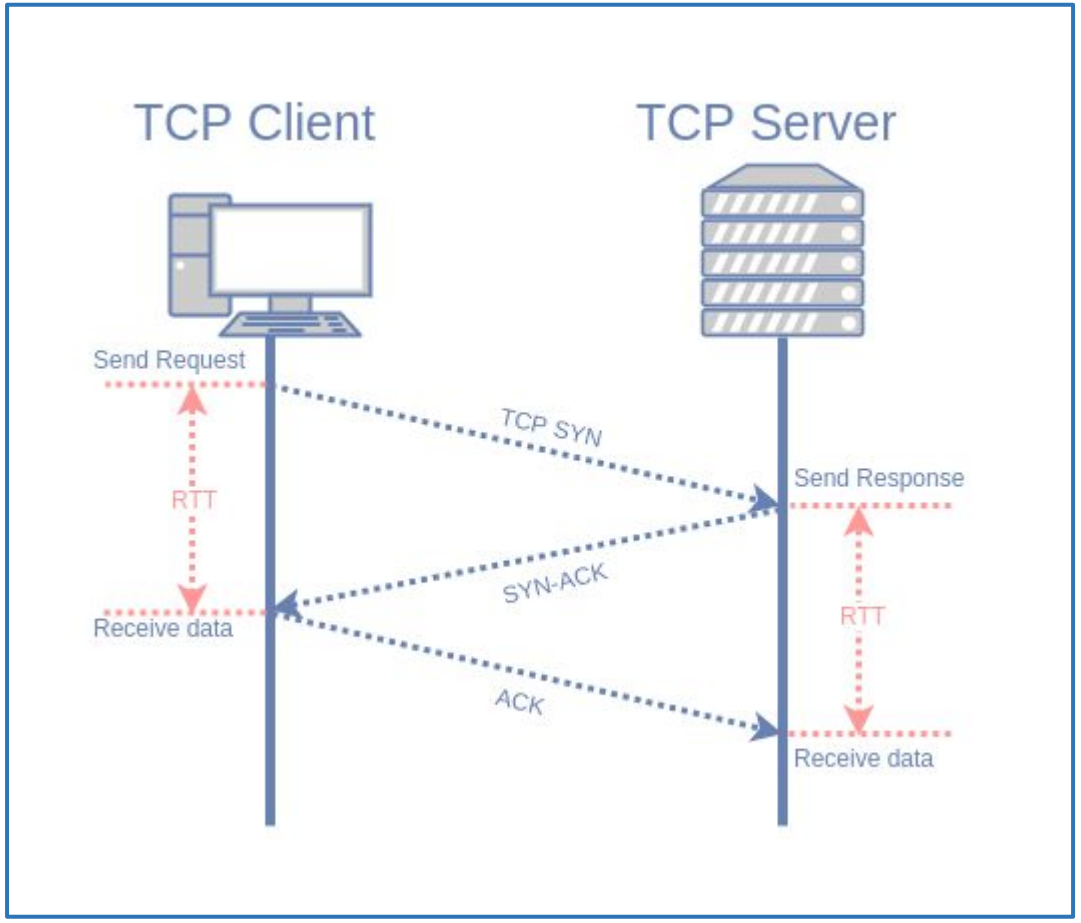
aside

protocols that maintain "state" are complex!

- past history (state) must be maintained
- if server/client crashes, their views of "state" may be inconsistent, must be reconciled

COMPUTER NETWORKS

HTTP Connections: Types, What is RTT



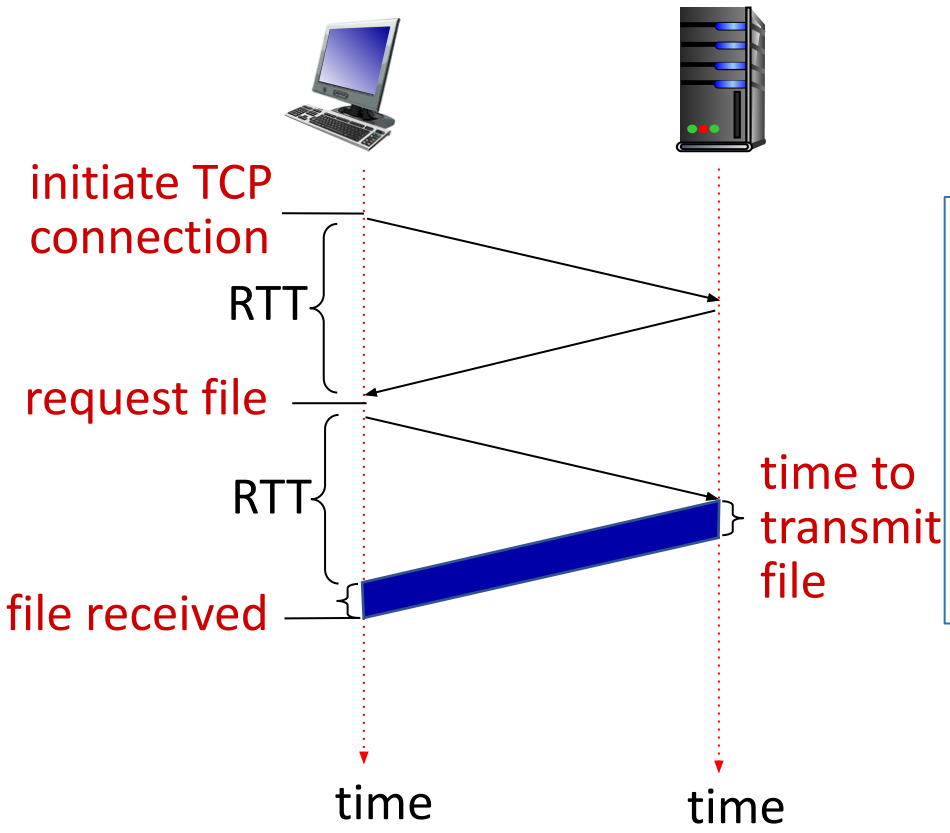
Non-persistent and **persistent** are the two types of HTTP connections used to connect the client with the webserver.

Non–persistent HTTP: response time

RTT (definition): time for a small packet to travel from client to server and back

HTTP response time (per object):

- one RTT to initiate TCP connection
- one RTT for HTTP request and first few bytes of HTTP response to return
- objject/file transmission time



NOTE:
The RTT includes packet-propagation delays, packet queuing delays in intermediate routers and switches, and packet-processing delays.

Non–persistent HTTP response time = 2RTT+ file transmission time

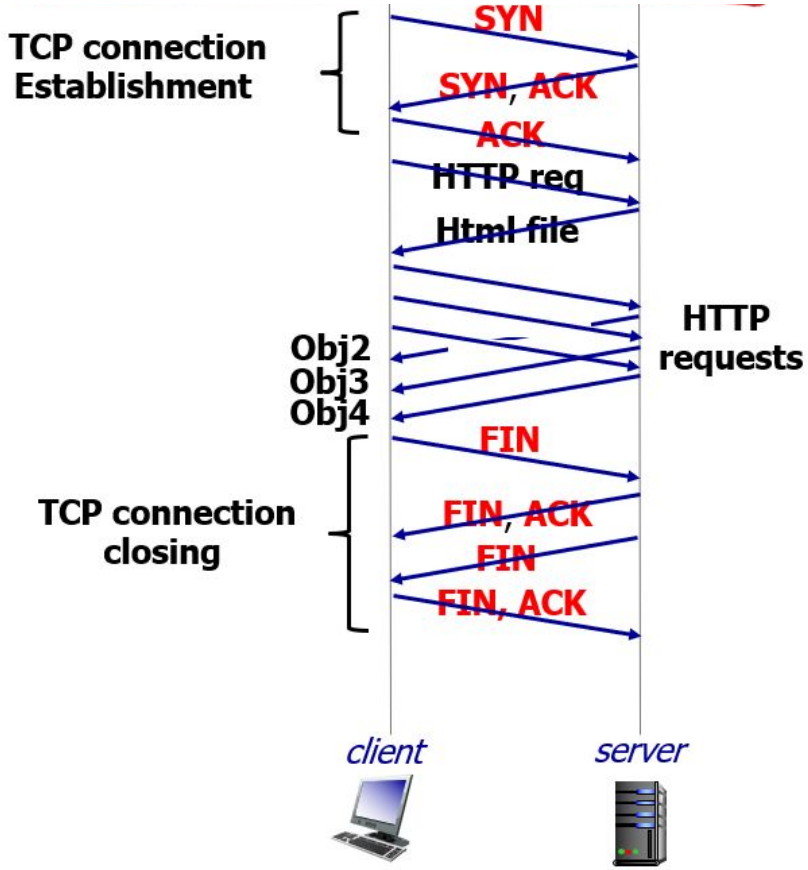
Persistent HTTP (HTTP 1.1)

Non-persistent HTTP issues:

- requires 2 RTTs per object
- OS overhead for *each* TCP connection (TCP buffer and variables)
- browsers often open multiple parallel TCP connections to fetch referenced objects in parallel

Persistent HTTP (HTTP1.1):

- server leaves connection open after sending response
- subsequent HTTP messages between same client/server sent over open connection
- client sends requests as soon as it encounters a referenced object
- as little as one RTT for all the referenced objects (cutting response time in half)



Persistent HTTP connection

- HTTP 1.1 persistent connections, the server leaves **the TCP connection open after sending a response.**
- Subsequent requests and responses between the same client and server can be sent over the **same connection.**

HTTP Connections: Summary

Non-persistent

- The non-persistent connection takes the connection time of **2RTT + *file transmission time*.**
- It takes the first RTT (round-trip time) to establish the connection between the server and the client.
- The second RTT is taken to request and return the object.
- This case stands for a single object transmission.

Persistent

- A persistent connection takes **2 RTT** for the connection and then transfers as many objects, as wanted, over this single connection.

RTT stands for **the round-trip time** taken for an object request and then its retrieval. In other words, it is the time taken to request the object from the client to the server and then retrieve it from the server back to the client.

Non–persistent HTTP: example

User enters URL: **`www.someSchool.edu/someDepartment/home.index`**
(base HTML file containing text, references to 10 jpeg images)

1a. HTTP client initiates TCP connection to HTTP server (process) at www.someSchool.edu on port 80

1b. HTTP server at host www.someSchool.edu waiting for TCP connection at port 80 "accepts" connection, notifying client

2. HTTP client sends HTTP *request message* (containing URL) into TCP connection socket. Message indicates that client wants object someDepartment/home.index

3. HTTP server receives request message, forms *response message* containing requested object, and sends message into its socket

time

COMPUTER NETWORKS

Non–persistent HTTP: example (more)

User enters URL: **`www.someSchool.edu/someDepartment/home.index`**
(containing text, references to 10 jpeg images)

4. HTTP server closes TCP connection.

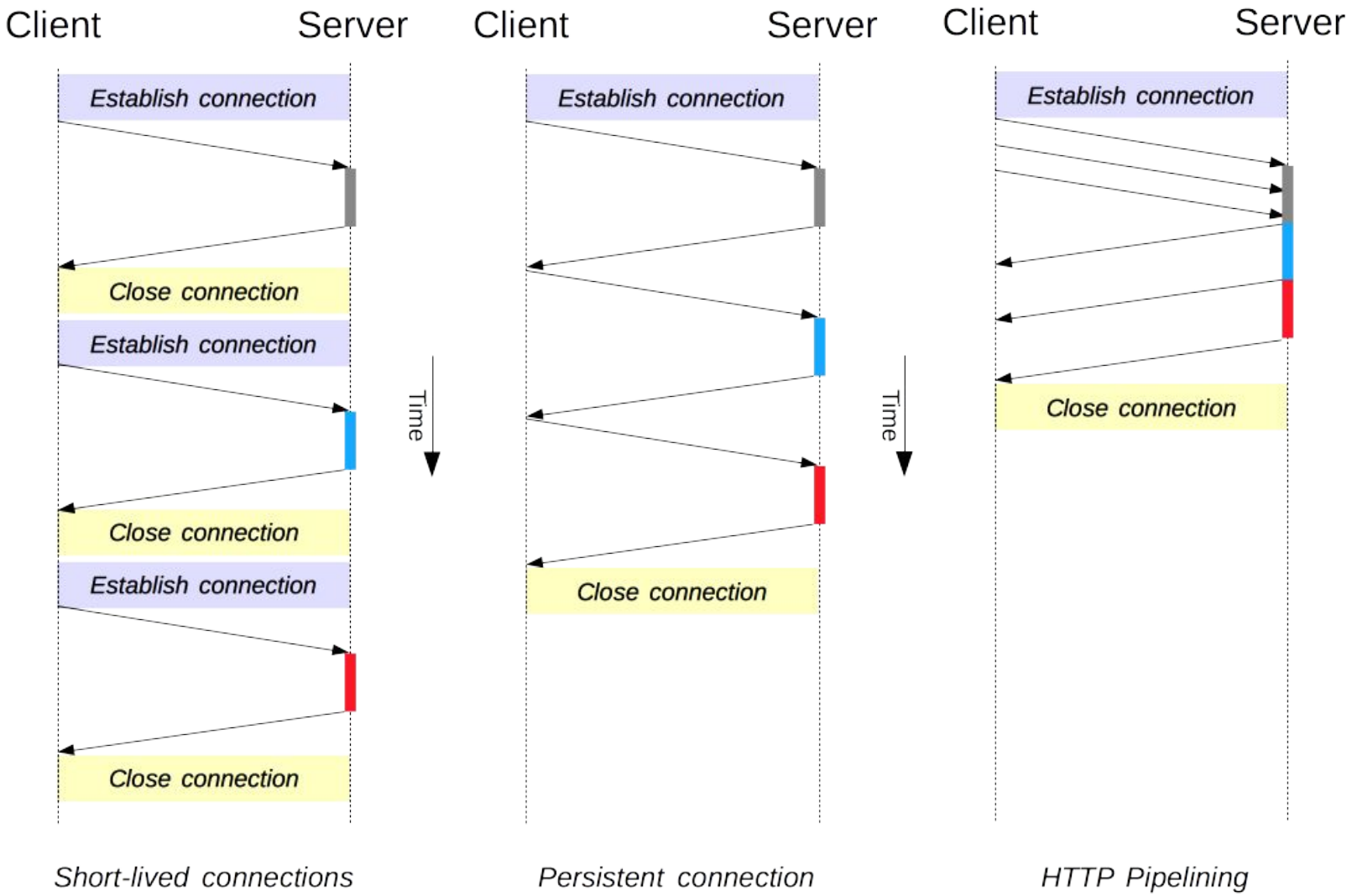
5. HTTP client receives response message containing html file, displays html. Parsing html file, finds 10 referenced jpeg objects

6. Steps 1-5 repeated for each of 10 jpeg objects

time

COMPUTER NETWORKS

Connection Management in HTTP/1.x

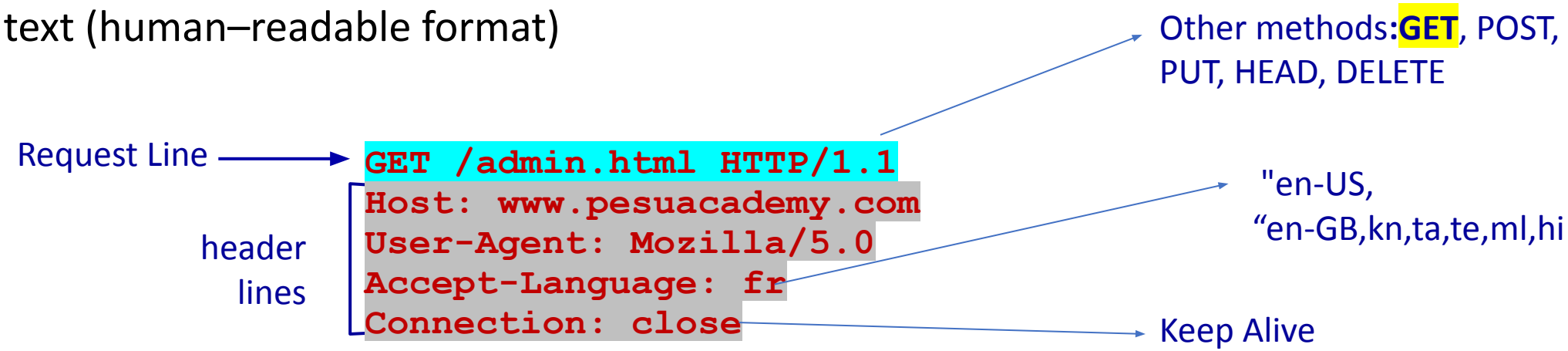


The default mode of HTTP uses persistent connections with pipelining.

HTTP MESSAGE FORMAT

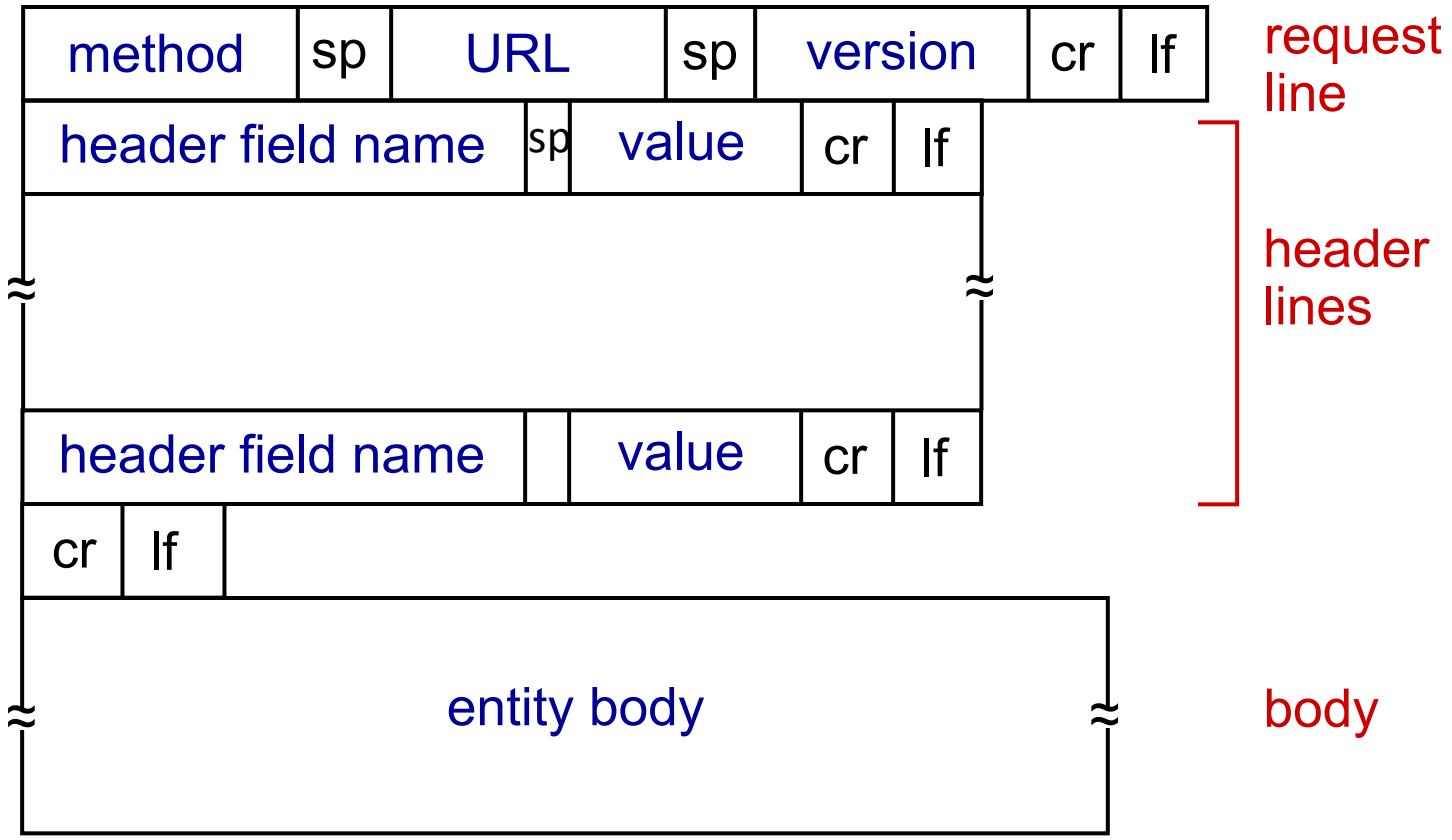
HTTP Request Message

- two types of HTTP messages: *request*, *response*
- HTTP request message:
  - ASCII text (human–readable format)



* Check out the online interactive exercises for more examples: ttp://gaia.cs.umass.edu/kurose_ross/interactive/

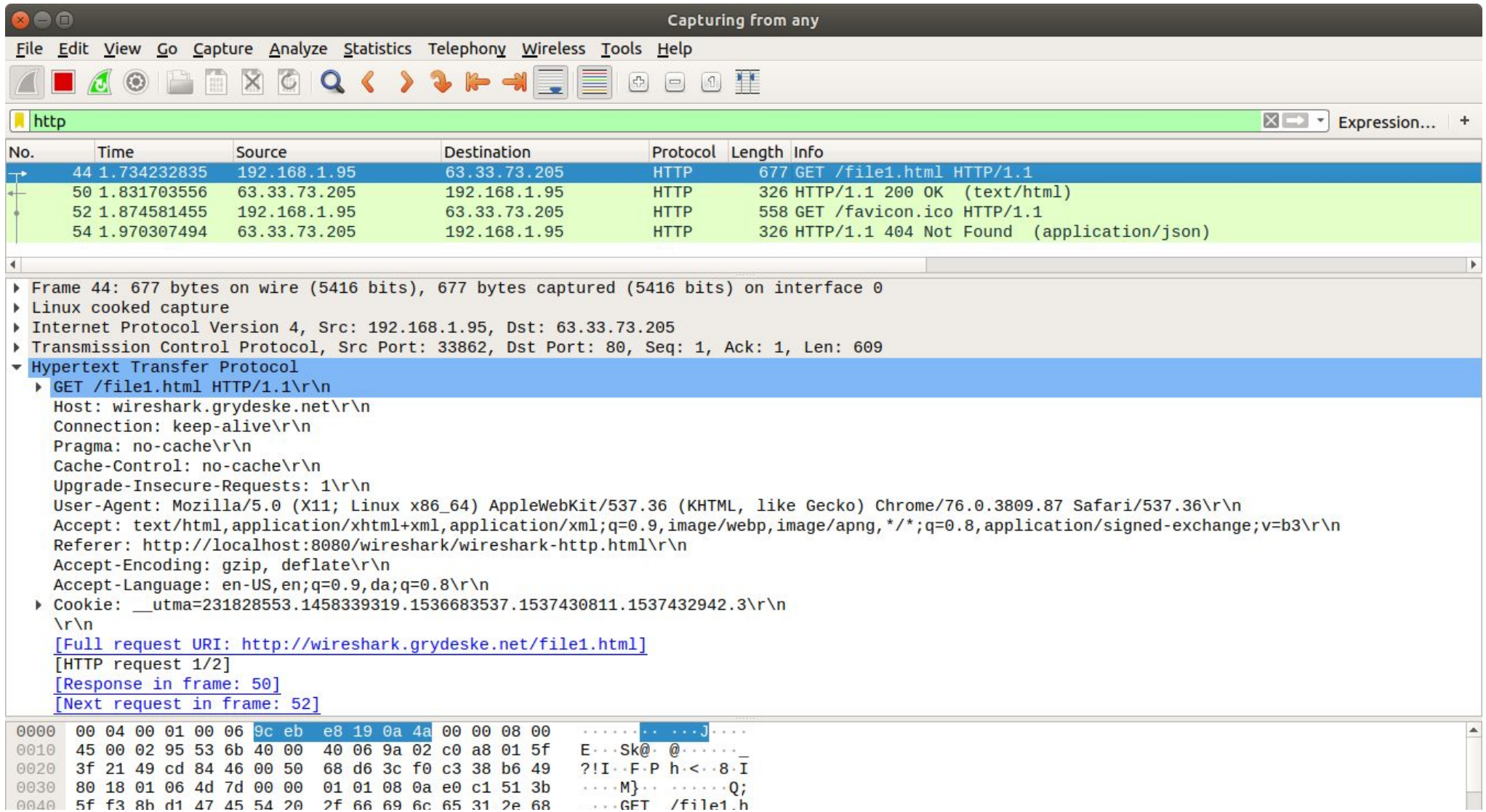
HTTP Request Message: General Format



HTTP specifications [RFC 1945; RFC 2616; RFC 7540] -] **include the definitions of the HTTP message formats**

COMPUTER NETWORKS

HTTP Request Message – Wireshark Capture



Other HTTP Request Messages

POST method:

- web page often includes **form input**
- user input sent from client to server in **entity body** of HTTP POST request message

GET method (for sending data to server):

- include user data in URL field of HTTP GET request message (following a '?'):

HEAD method:

- Similar to a GET method.
- Often used for debugging

PUT method:

- It allows a user to upload an object to a specific path (directory) on a specific Web server.

DELETE method

allows a user, or an application, to delete an object on a Web server.

https://www.geeksforgeeks.org/explore?page=1&sortBy=submissions

COMPUTER NETWORKS

Get and Post Differences

GET Method	POST Method
Used to retrieve information from the server.	Used to send data to the server to create/update a resource.
Appends data to the URL, visible to all.	Includes data in the request body, not displayed in the URL.
Limited by the URL length; less data can be sent.	No limitations on data size; suitable for large amounts of data.
Can be cached.	Not cached by default.
Less secure as data is exposed in the URL.	More secure; data is concealed within the request body.
Ideal for searching and retrieving data.	Ideal for transactions and updating data.

COMPUTER NETWORKS

HTTP Response Message

status line (protocol status code status phrase)

header lines

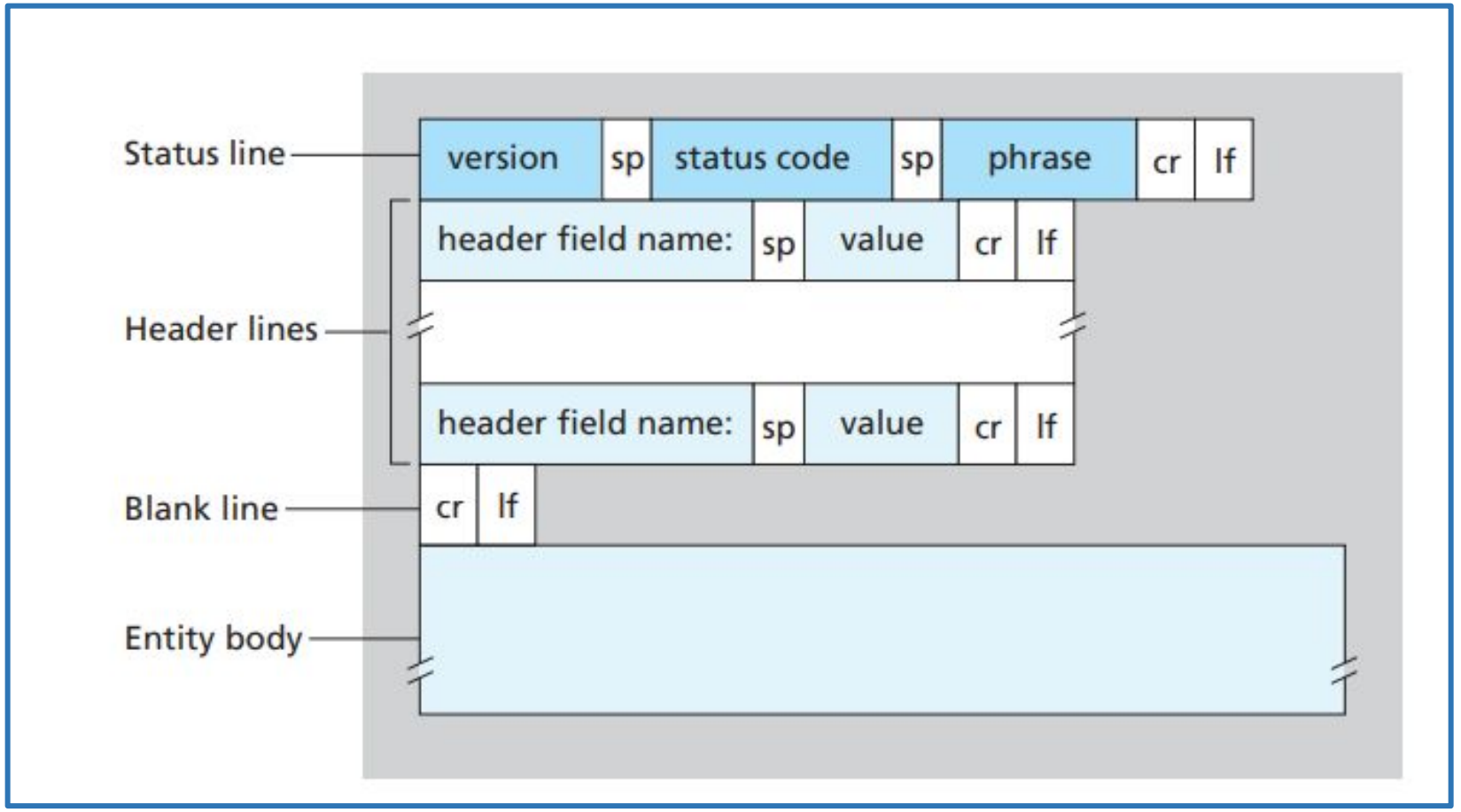
Data e.g: requested HTML file

```
HTTP/1.1 200 OK\r\n
Connection: Keep-Alive\r\n
Date: Fri,18 Aug 2023 20:09:20 GMT\r\n
Server: Apache/2.0.52 (CentOS)\r\n
Last-Modified: Thu,30 Mar 2023 17:00:02 GMT\r\n
Content-Length: 2652\r\n (bytes)
Content-Type: text/html\r\n
\r\n
data data data data data ...
```

* Check out the online interactive exercises for more examples: http://gaia.cs.umass.edu/kurose_ross/interactive/

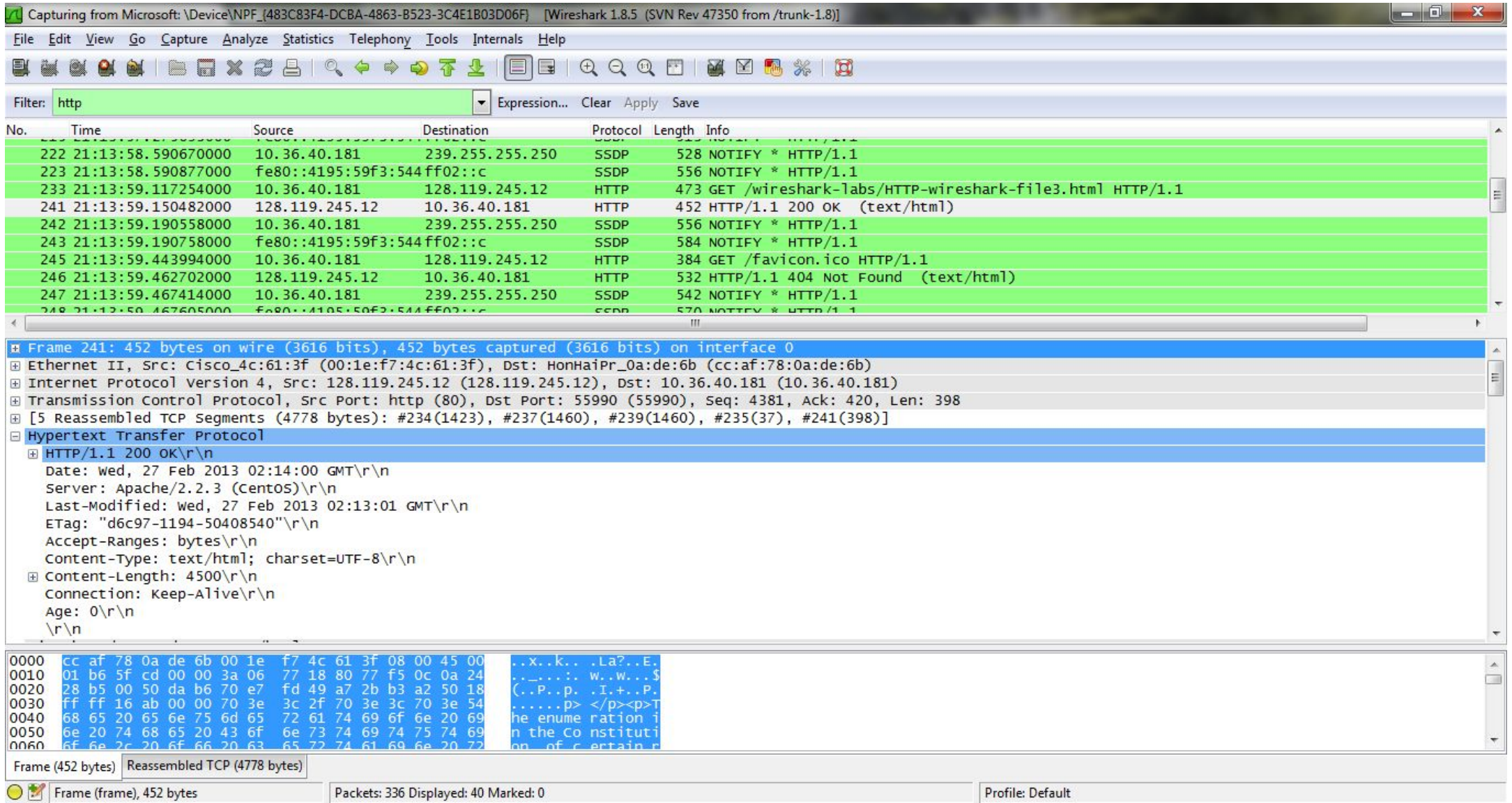
COMPUTER NETWORKS

HTTP Response Format



COMPUTER NETWORKS

HTTP Response Message – Wireshark Capture



HTTP Response Status Codes

- status code appears in 1st line in server–to–client response message.
- some sample codes:

200 OK
- request succeeded, requested object later in this message

301 Moved Permanently
- requested object moved, new location specified later in this message (in Location: field)

400 Bad Request
- request msg not understood by server

404 Not Found
- requested document not found on this server

505 HTTP Version Not Supported
- Used http version not supported

HTTP vs HTTPS

COMPUTER NETWORKS

HTTP vs HTTPS

What is HTTP?
http://www.seopressor.com/blog/

- ***HTTP*** stands for ***Hypertext Transfer Protocol***.
- At it's most basic, it allows for the communication between different systems.
- It's most commonly used to ***transfer data from a web server to a browser*** in order to allow users to view web pages.
- It's the protocol that was used for basically all early websites.

COMPUTER NETWORKS

HTTP vs HTTPS (more)

What is HTTPS?

https://www.unionbankonline.co.in/

- ***HTTPS*** stands for ***Hypertext Transfer Protocol Secure***.
- The problem with the regular HTTP protocol is that the information that flows from server to browser is not encrypted, which means it can be *easily stolen*.
- HTTPS protocols remedy this by using an ***SSL*** (***Secure Sockets Layer***) certificate, which helps ***create a secure encrypted connection*** between the server and the browser, thereby protecting potentially sensitive information from being stolen as it is transferred between the server and the browser.

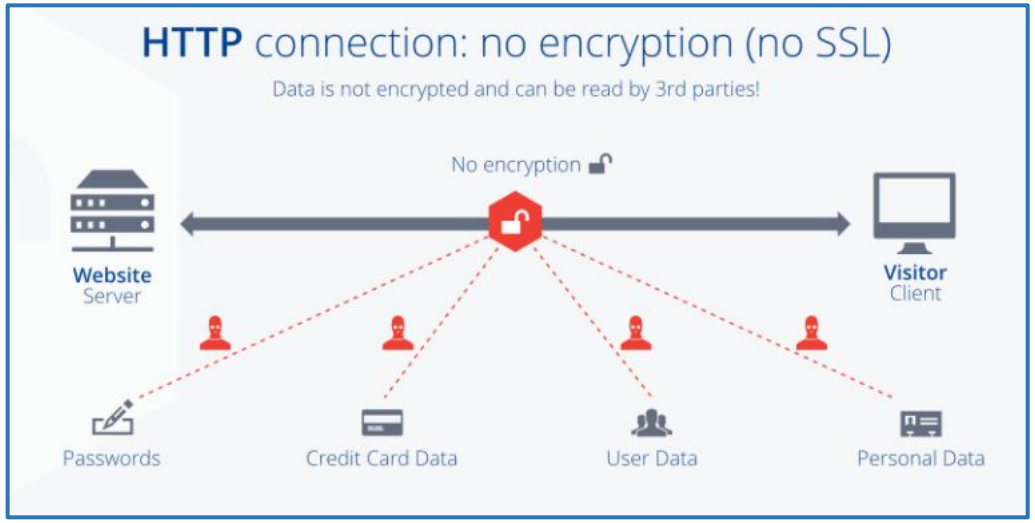
HTTP vs HTTPS (more)

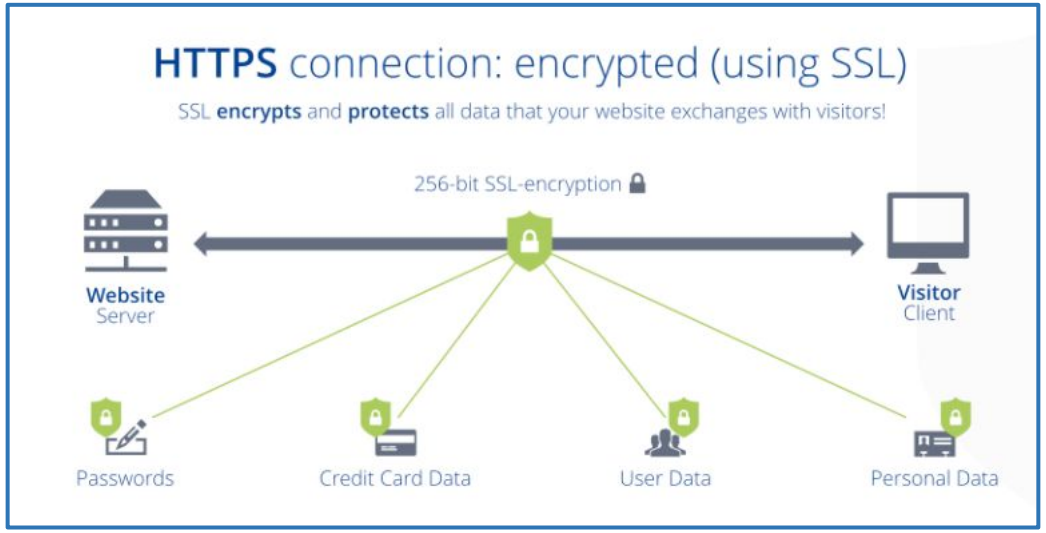
What is the main difference between HTTP and HTTPS?

COMPUTER NETWORKS

HTTP vs HTTPS (more)

- The most important difference between the two protocols is the **SSL certificate**.
- In fact, HTTPS is basically an HTTP protocol with additional security.
- However, this additional security can be extremely important, especially for websites that take sensitive data from its users, such as *credit card information and passwords*.

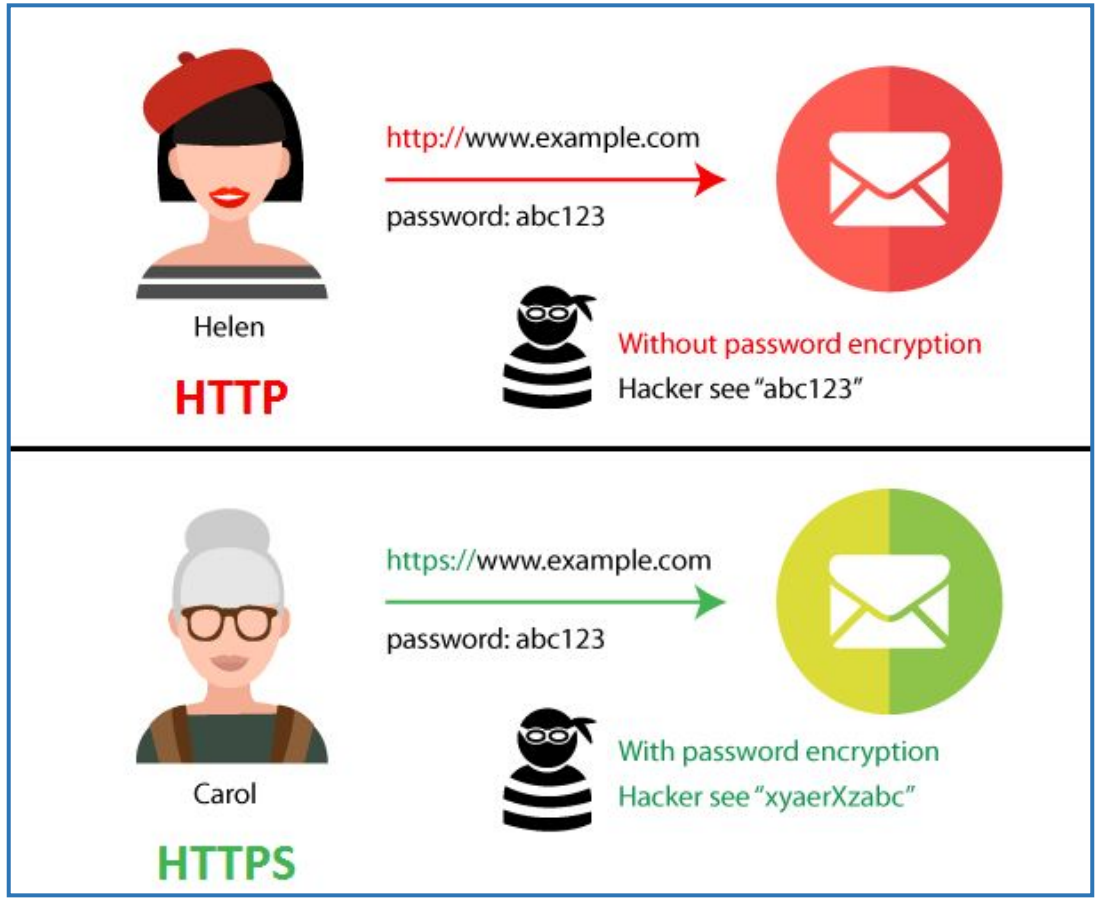




COMPUTER NETWORKS

HTTP vs HTTPS (more)

- **How do HTTPS works?**
- The SSL certificate encrypts the information that users supply to the site, which basically **translates the data into a code**.
- Even if someone manages to steal the data being communicated between the sender and the recipient, they would not be able to understand it due to this encryption.



COMPUTER NETWORKS

HTTP vs HTTPS (more)

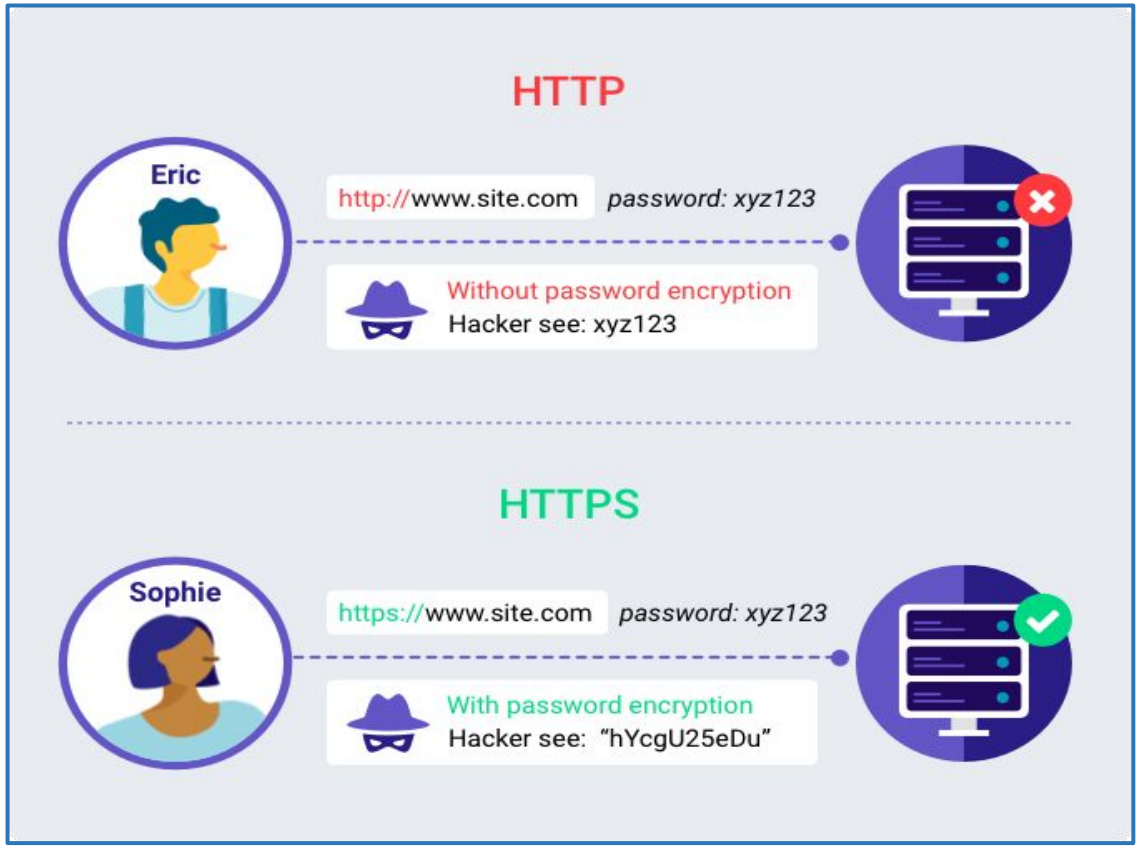
- But in addition to adding that *extra layer of security*, **HTTPS is also secured via TLS (Transport Layer Security) protocol**.
- TLS helps provide **data integrity**, which helps *prevent the transfer of data* from being modified or corrupted, and **authentication**, which proves to your users that they are communicating with the intended website.

SSL	TLS
Standard security protocol for establishing an encrypted link between a web server and a browser	Protocol that provides communication security between client/server applications that communicate with each other over the interne
Introduced in the year 1994 by Netscape Communications	Introduced in 1999 by Internet Engineering Task Force (IETF)
Stands for Secure Socket Layer	Stands for Transport Layer Security
Not as secure as TSL	More secure
Comparatively less complex	A complex protocol

COMPUTER NETWORKS

HTTP vs HTTPS (more)

- Users can identify whether a site uses HTTPS protocol by the **web address**.
- The very first part of the web address *(before the "www")* indicates whether the site uses HTTP or HTTPS protocols.



COMPUTER NETWORKS

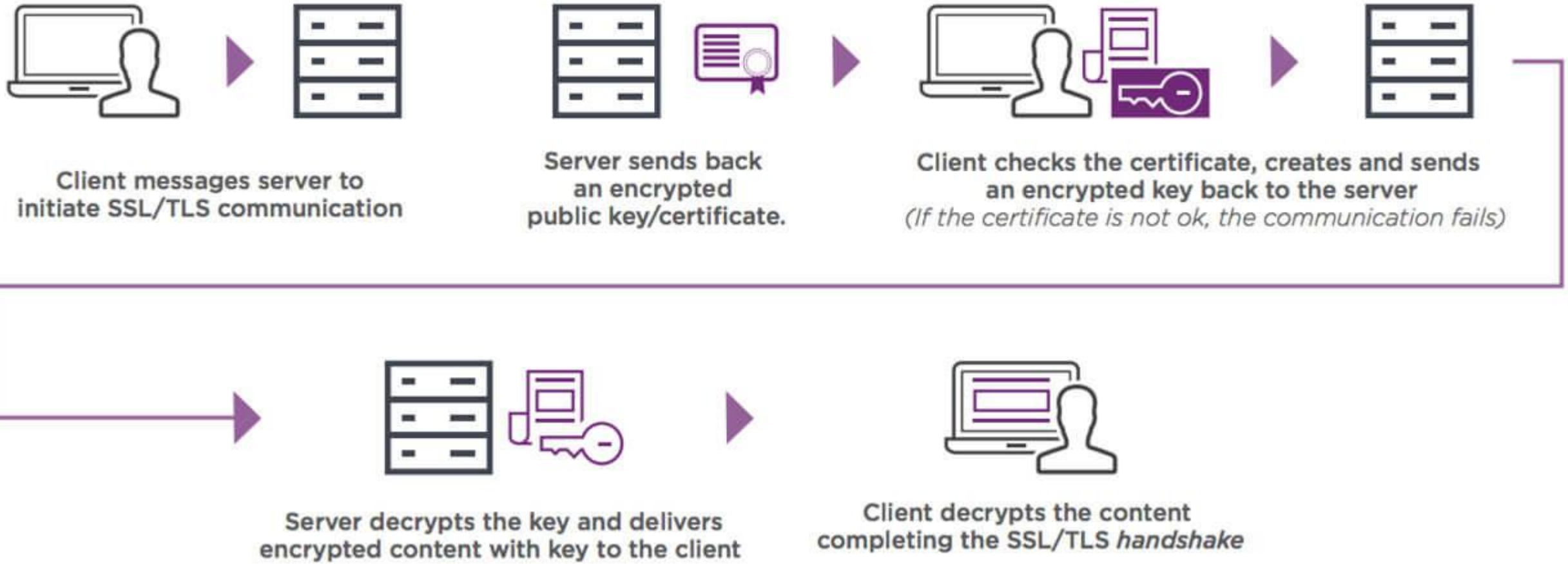
HTTP vs HTTPS (more)

- ✔ HTTP operates at application layer, while HTTPS operates at transport layer.
- ✔ No SSL certificates are required for HTTP, with HTTPS it is required that you have an SSL certificate and it is signed by a CA.
- ✔ HTTP doesn't require domain validation, where as HTTPS requires at least domain validation and certain certificates even require legal document validation.

HTTP	HTTPS
System for transmitting and receiving information over Internet	HTTPS need arose to address exchange of confidential information over Insecure Internet.
HTTP is appropriate when non-sensitive information needs to be exchanged.	Transfer of encrypted information
Used in Blogs, entertainment and articles.	For financial and other confidential information , HTTPS will be the right choice
Used for data transfer with header from client	HTTP within SSL/TLS
Sir Timothy John invented the protocol	Netscape corporation invention the protocol
Uses Port number 80	Uses port number 443
SSL Certificates not used for communication	Uses SSL Certificates for communication

COMPUTER NETWORKS

How does SSL works? (more)



COMPUTER NETWORKS

How does SSL works?

Client Hello

Server communicates with the client using SSL.
This includes the SSL version number, cipher settings, and session-specific data.

Server Hello

The server responds with a “server hello” message.
This includes the server’s SSL version number, cipher settings, session-specific data, an SSL certificate with a public key, and other information that the client needs to communicate with the server over SSL.

Authentication

The client verifies the server’s SSL certificate from the CA (Certificate Authority) and authenticates the server.
If the authentication fails, then the client refuses the SSL connection and throws an exception. If the authentication succeeds, then proceed to the next step.

Decryption

The client creates a session key, encrypts it with the server’s public key and sends it to the server.
If the server has requested client authentication (mostly in server to server communication), then the client sends their own certificate to the server.

Encryption with Session Key

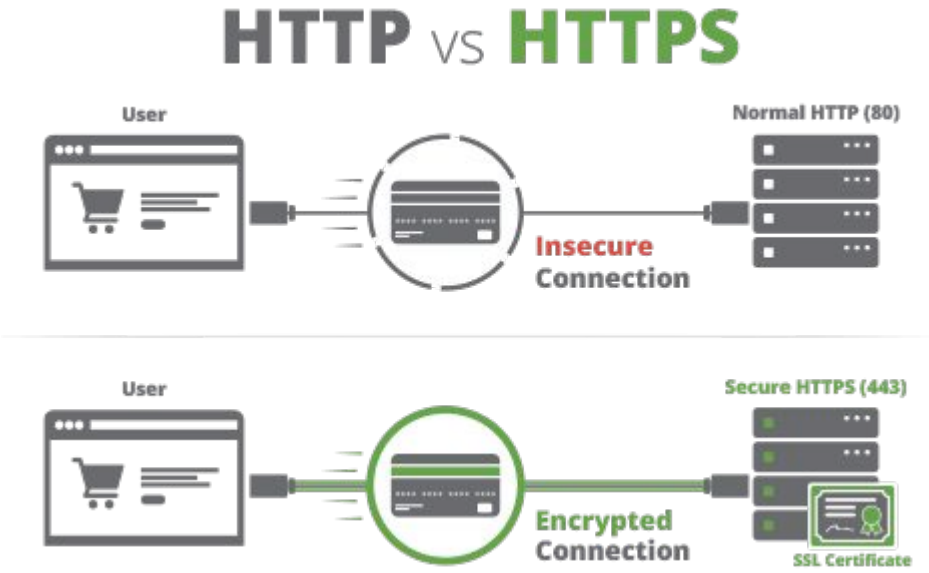
The server decrypts the session key with its private key and sends the acknowledgement to the client encrypted with the session key.

Benefits of HTTPS over HTTP using SSL Certificates

- Stronger Google ranking.
- Improved security.
- Increased customer confidence / safer experience.
- Build customer trust and improve conversions.

COMPUTER NETWORKS

HTTP vs HTTPS



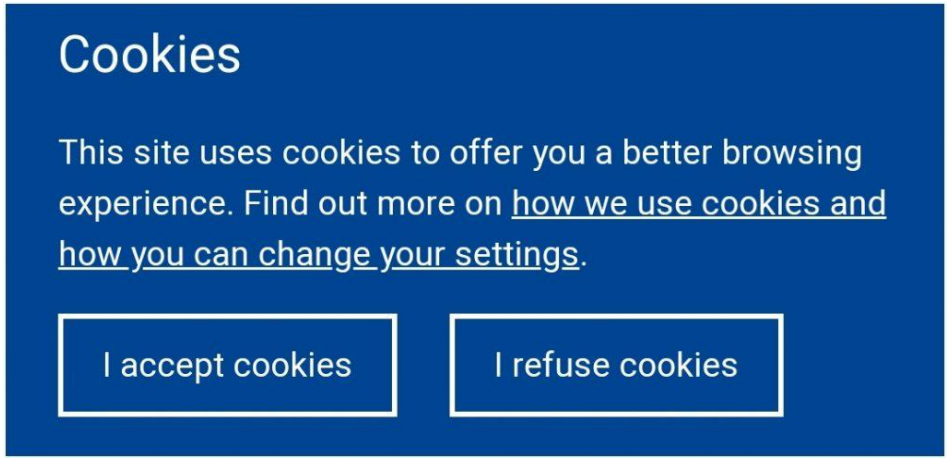
- HTTPS is HTTP with encryption – All communications between browser and server are encrypted (bi-directional).
- ‘S’ refers ‘Secure’ or HTTP over Secure Socket Layer.
- Uses TLS (SSL) to encrypt normal HTTP requests and responses.
- Attackers can't read the data crossing the wire and you know you are talking to the server you think you are talking too.

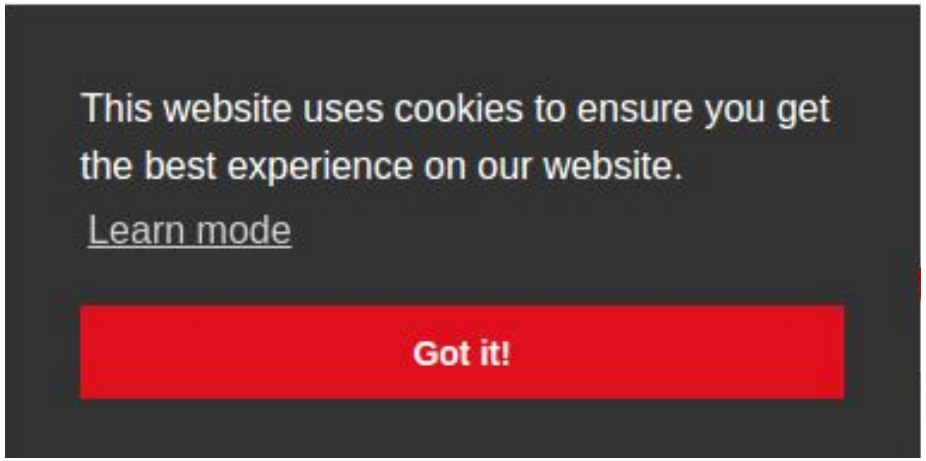
COOKIES

COMPUTER NETWORKS

Cookies

- Website/HTTP/Internet cookies
- Piece of data from a specific website
- Stored on a user's computer
- Allows sites to keep track of users
- Eg: language selection



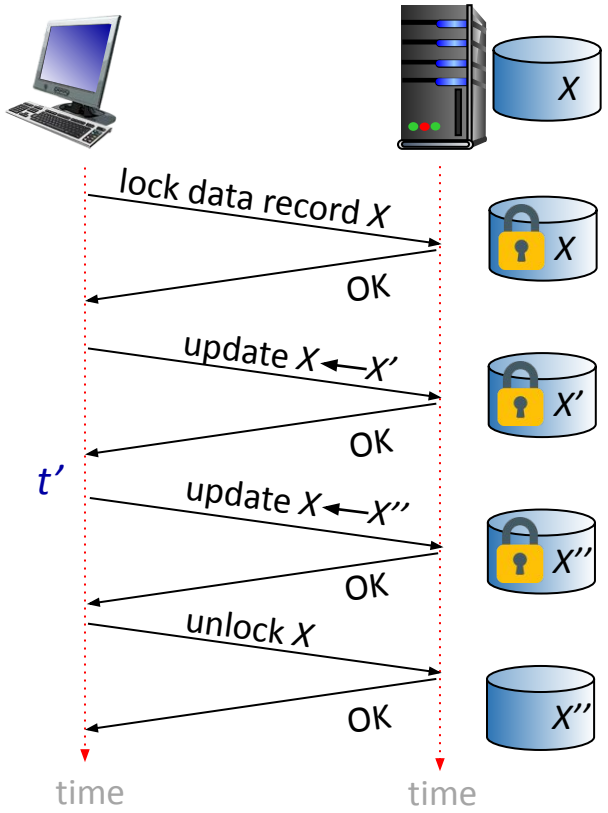


Maintaining user/server state: cookies

Recall: HTTP GET/response interaction is *stateless*

- no need for client/server to track "state" of multi–step exchange
- all HTTP requests are independent of each other
- no need for client/server to "recover" from a partially-completed-but-never-completely-completed transaction

a stateful protocol: client makes two changes to X, or none at all



Q: what happens if network connection or client crashes at *t'* ?

Maintaining user/server state: cookies (more)

Web sites and client browser use *cookies* to maintain some state between transactions

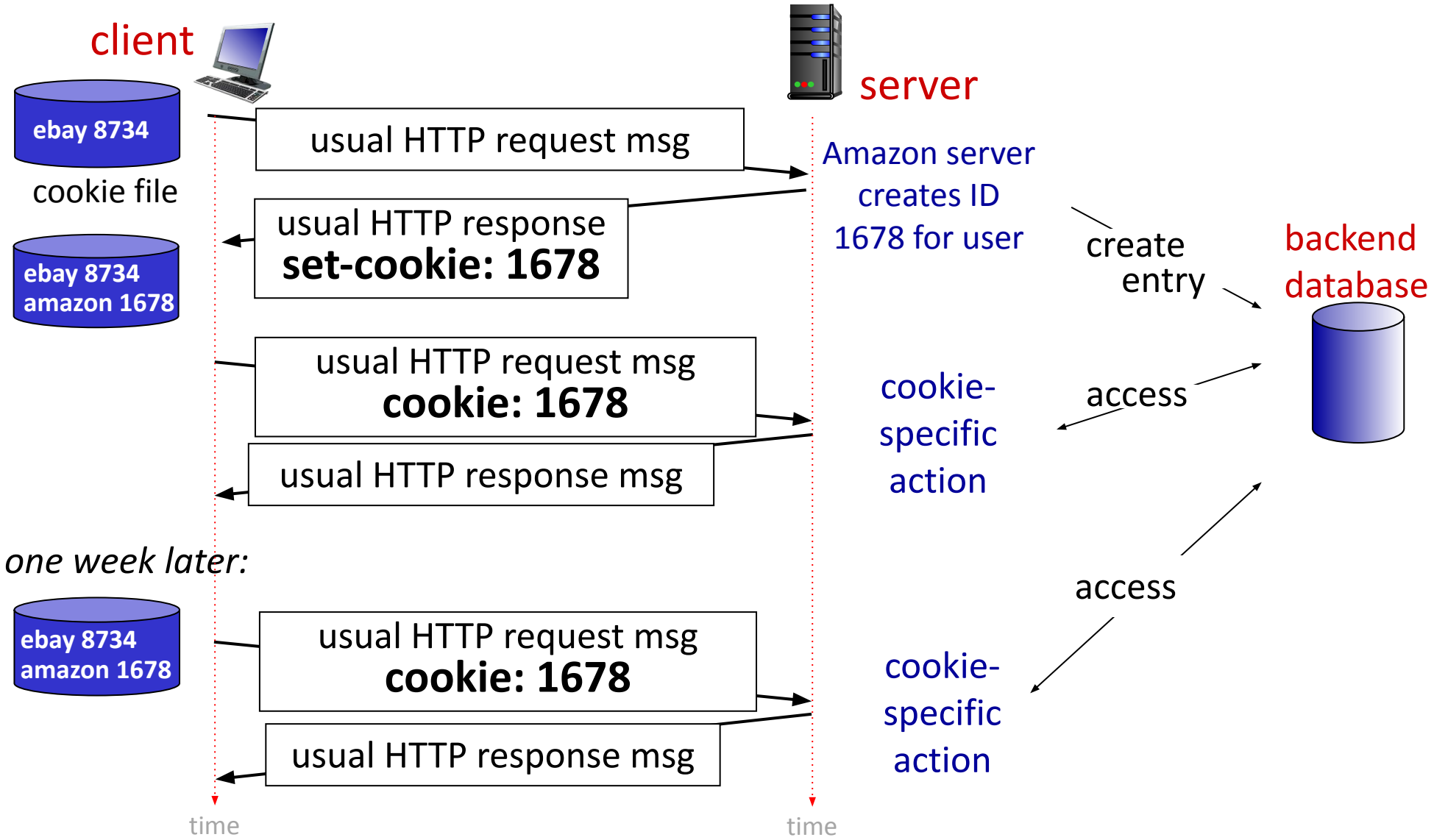
Four components:

- ✔ **cookie header line of HTTP *response* message**
- ✔ **cookie header line in next HTTP *request* message**
- ✔ **cookie file kept on user's host, managed by user's browser**
- ✔ **Back–end database at Web site**

Example:

- Susan uses browser on laptop, visits specific e–commerce site for first time
- when initial HTTP requests arrives at site, site creates:
  - unique ID (aka "cookie")
  - entry in backend database for ID
- subsequent HTTP requests from Susan to this site will contain cookie ID value, allowing site to "identify" Susan

Maintaining user/server state: cookies (more)



HTTP Cookies: Comments

What cookies can be used for:

- track user's browsing history
- remembering login details
- track visitor count
- shopping carts
- recommendations
- save coupon codes for you

cookies and privacy:

- cookies permit sites to *learn* a lot about you on their site.
- third party persistent cookies (tracking cookies) allow common identity (cookie value) to be tracked across multiple web sites

WEB CACHES

Web Caches

- A Web cache—also called a **proxy server**—is a network entity that satisfies HTTP requests on the behalf of an origin Web server.
- The Web cache has its **own disk storage** and keeps copies of **recently requested objects** in this storage.

COMPUTER NETWORKS

Web Caches - EXAMPLE

Teachers and students of PES UNIVERSITY needed fast Internet to research and communicate effectively. But they were challenged by the expensive and limited bandwidth for the university's 2000+ connected devices. At 10 Lakh per month for an 100Mbps Internet connection, the university ruled out a bandwidth upgrade.

Instead they opted for a web caching solution.

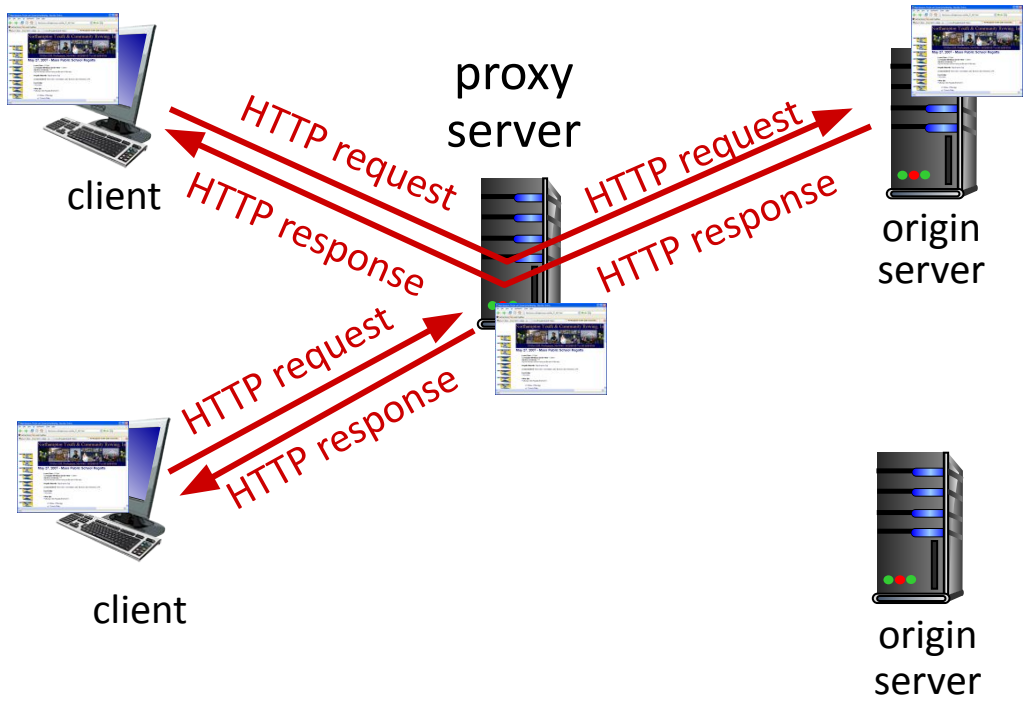
- The **web cache appliance** was used in conjunction with a **Cisco router**. (real time)
- The router redirects the web traffic while the device **keeps a copy of all videos, images, and other static content.**
- Caching the content enabled the university to have **significant improvements in web performance without spending more on bandwidth.**
- This was especially **useful** during back-to-back group lessons when **students had to access the same content.**

COMPUTER NETWORKS

Web Caches (Proxy Servers)

Goal: **satisfy client request without involving origin server**

- user configures browser to point to a *Web cache*
- browser sends all HTTP requests to cache
  - *if* object in cache: cache returns object to client
  - *else* cache requests object from origin server, caches received object, then returns object to client



Web Caches (Proxy Servers)

- **Web cache acts as both client and server at the same time.**
  - When it receives requests from and sends responses to a browser, it is a server.
  - When it sends requests to and receives responses from an origin server, it is a client.
- **Typically cache is installed by ISP (university, company, residential ISP).**
  - For example, a university might install a cache on its campus network and configure all of the campus browsers to point to the cache.

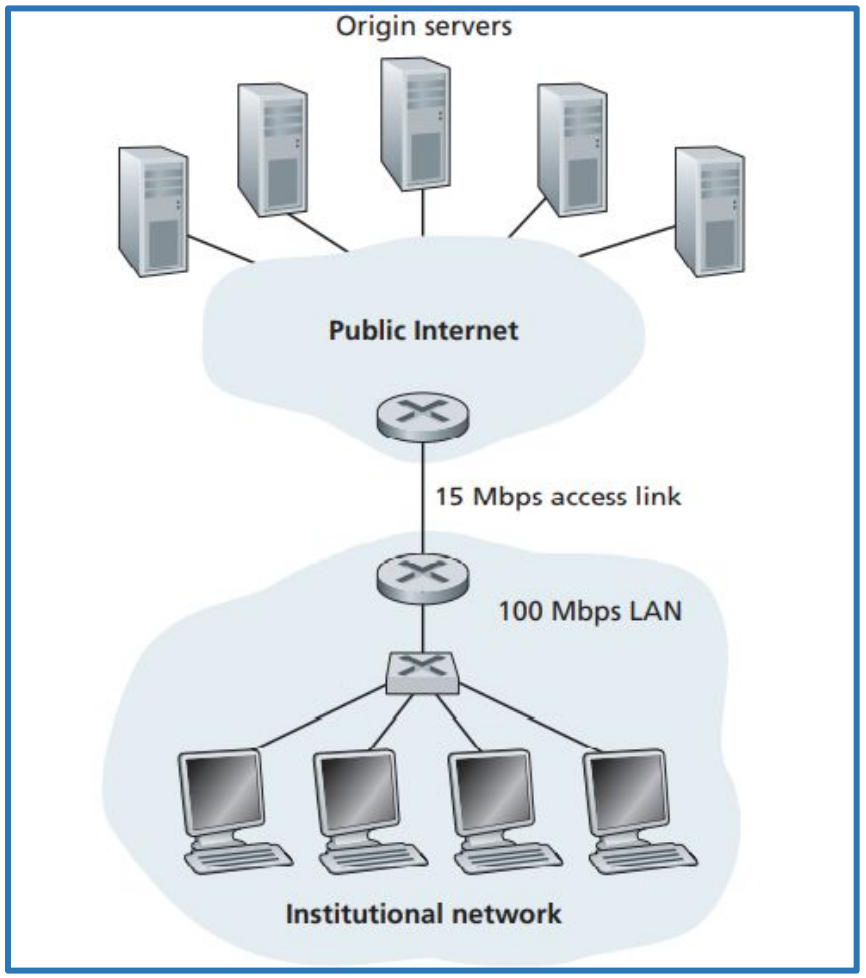
Why Web caching?

- **reduce response time** for client request (speed)
  - cache is closer to client
- **reduce traffic** on an institution's access link (saves bandwidth)
  - does not have to upgrade bandwidth as quickly, thereby reducing costs.

COMPUTER NETWORKS

Web Caches : Example

Consider a Scenario:



Scenario:

- access link rate: **15 Mbps**
- Average RTT from institutional router to server: **2 sec**
- Web object size: **1Mbits**
- Average request rate from browsers to origin servers: **15req/sec.**

Performance:

- LAN utilization= ?
- access link utilization = ?
- end–to–end delay = **Internet delay + access link delay + LAN delay**

  = 2 sec + ? minutes + ? secs

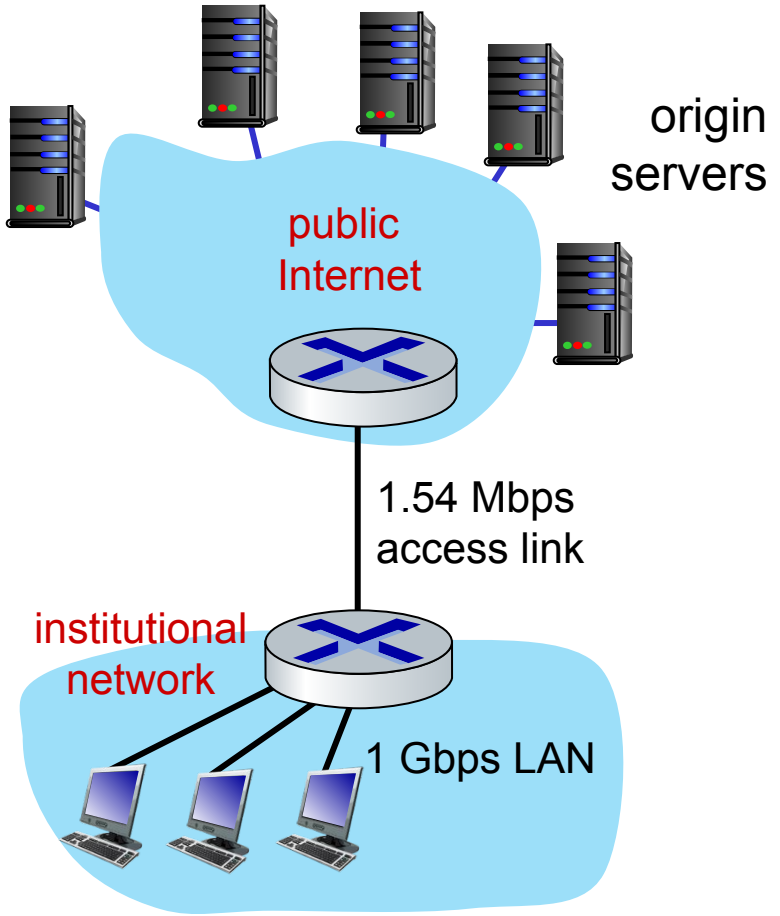
Caching example

(15 req/sec) * (100 Kbits/req)/(1.54 Mbps) = 0.974

(15 req/sec) * (100 Kbits/req)/(1 Gbps) = 0.0015

Scenario:

- access link rate: 1.54 Mbps
- RTT from institutional router to server: 2 sec
- Web object size: 100K bits
- Average request rate from browsers to origin servers: 15req/sec
  - average data rate to browsers: 1.50 Mbps



Performance:

- LAN utilization: .0015
- access link utilization = .97
- end–to–end delay = Internet delay + access link delay + LAN delay

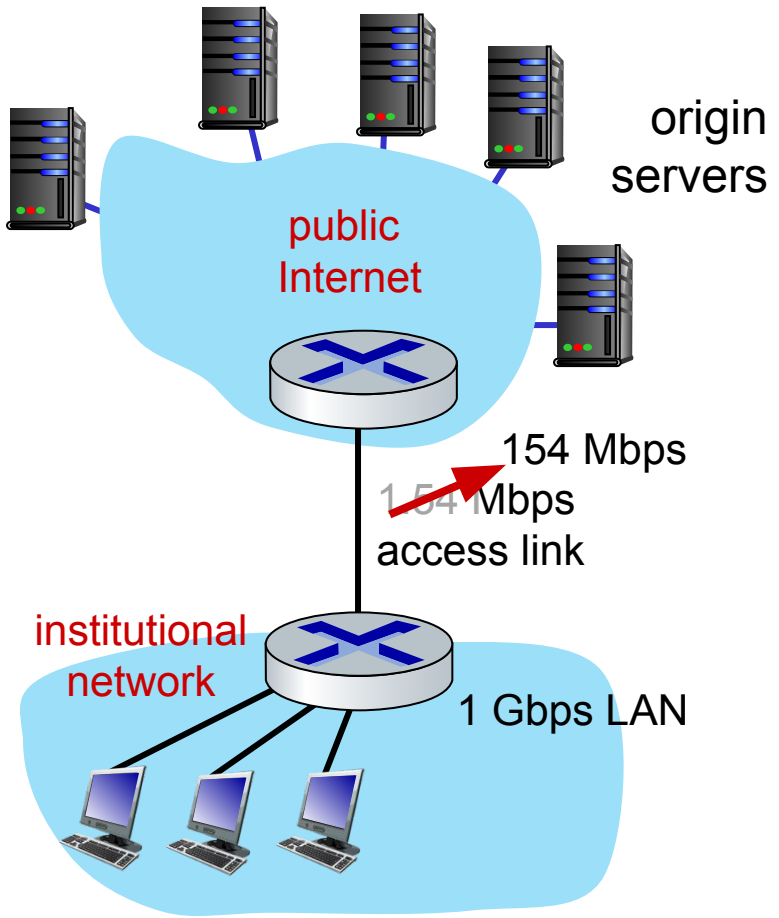
  = 2 sec + minutes + usecs

problem: large delays at high utilization!

Caching example: buy a faster access link

Scenario:

- access link rate: 1.54 Mbps → 154 Mbps
- RTT from institutional router to server: 2 sec
- Web object size: 100K bits
- Avg request rate from browsers to origin servers: 15/sec
  - avg data rate to browsers: 1.50 Mbps



Performance:

- LAN utilization: .0015
- access link utilization = .97 → .0097
- end-end delay = Internet delay + access link delay + LAN delay
  = 2 sec + minutes + usecs → msecs

Cost: faster access link (expensive!)

Caching example: install a web cache

Scenario:

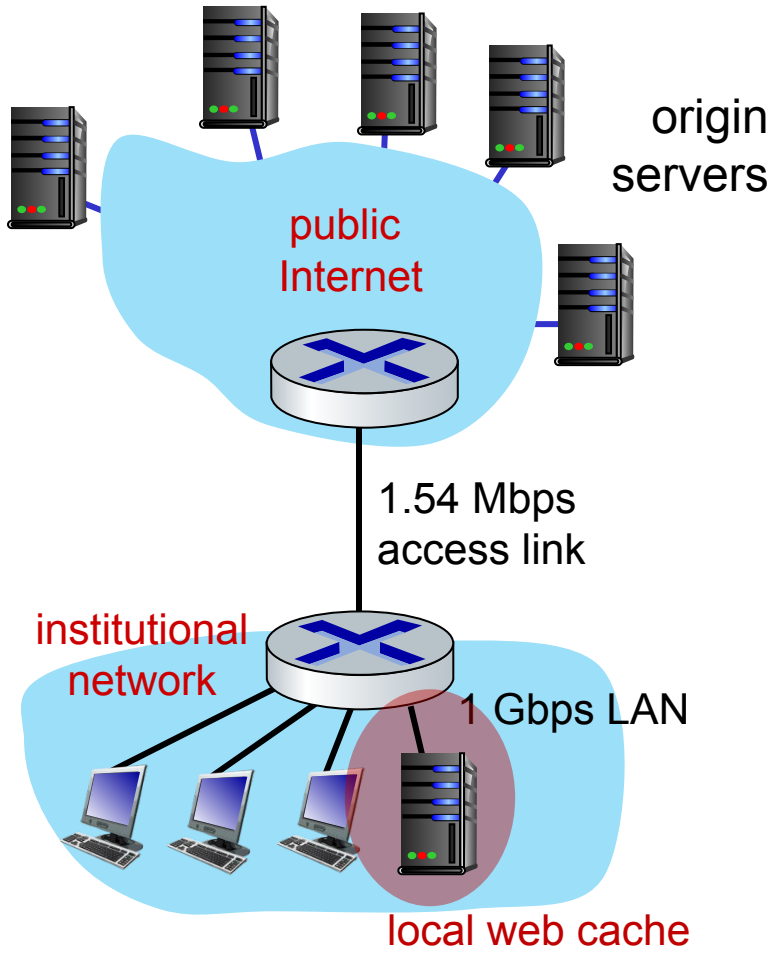
- access link rate: 1.54 Mbps
- RTT from institutional router to server: 2 sec
- Web object size: 100K bits
- Avg request rate from browsers to origin servers: 15/sec
  - avg data rate to browsers: 1.50 Mbps

Performance:

- LAN utilization: .?
- access link utilization = ?
- average end-end delay = ?

How to compute link utilization, delay?

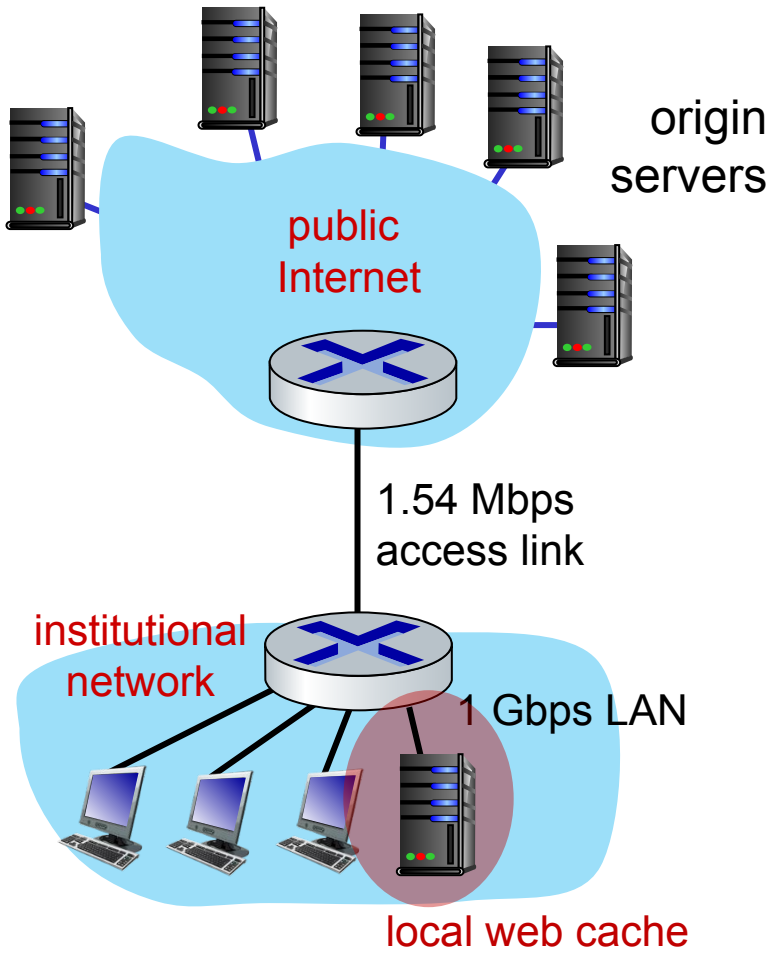
Cost: web cache (cheap!)



Caching example: install a web cache

Calculating access link utilization, end-end delay with cache:

- suppose cache hit rate is 0.4: 40% requests satisfied at cache, 60% requests satisfied at origin
- access link: 60% of requests use access link
- data rate to browsers over access link
  = 0.6 * 1.50 Mbps = .9 Mbps
- utilization = 0.9/1.54 = .58
- average end–to–end delay
  = 0.6 * (delay from origin servers)
  + 0.4 * (delay when satisfied at cache)
  = 0.6 (2.01) + 0.4 (~msecs) = ~ 1.2 secs



lower average end–to–end delay than with 154 Mbps link (and cheaper too!)

Conditional GET

- A new problem—**the copy of an object residing in the cache may be stale.**
- In other words, the object housed in the Web server may have been modified since the copy was cached at the client.
- Fortunately, HTTP has a mechanism that allows a cache to verify that its objects are up to date. This mechanism is called the **conditional GET.**
- An HTTP request message is a so-called conditional GET message if
  - **the request message uses the GET method and**
  - **the request message includes an If-Modified-Since: header line.**

COMPUTER NETWORKS

Conditional Get - Example

- On the behalf of a requesting browser, a proxy cache sends a request message to a Web server:

GET /fruit/kiwi.gif HTTP/1.1

Host: www.exotiquecuisine.com

- Second, the Web server sends a response message with the requested object to the cache:

HTTP/1.1 200 OK

Date: Sat, 3 Oct 2015 15:39:29

Server: Apache/1.3.0 (Unix)

Last-Modified: Wed, 9 Sep 2015 09:23:24

Content-Type: image/gif

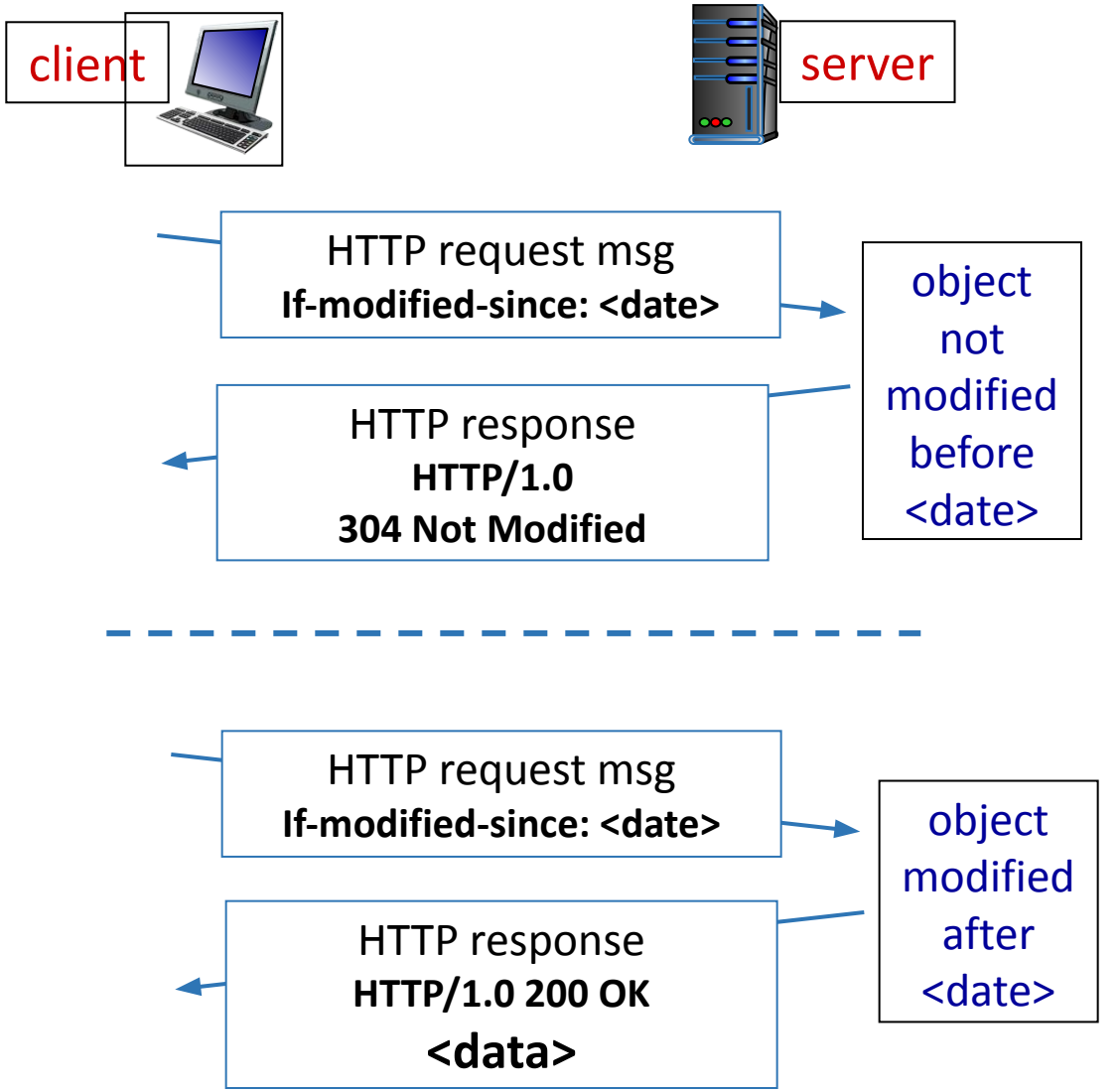
(data data data data data ...)

- The cache forwards the object to the requesting browser but also caches the object locally.
- The cache also stores the last-modified date along with the object.

Conditional Get - Example

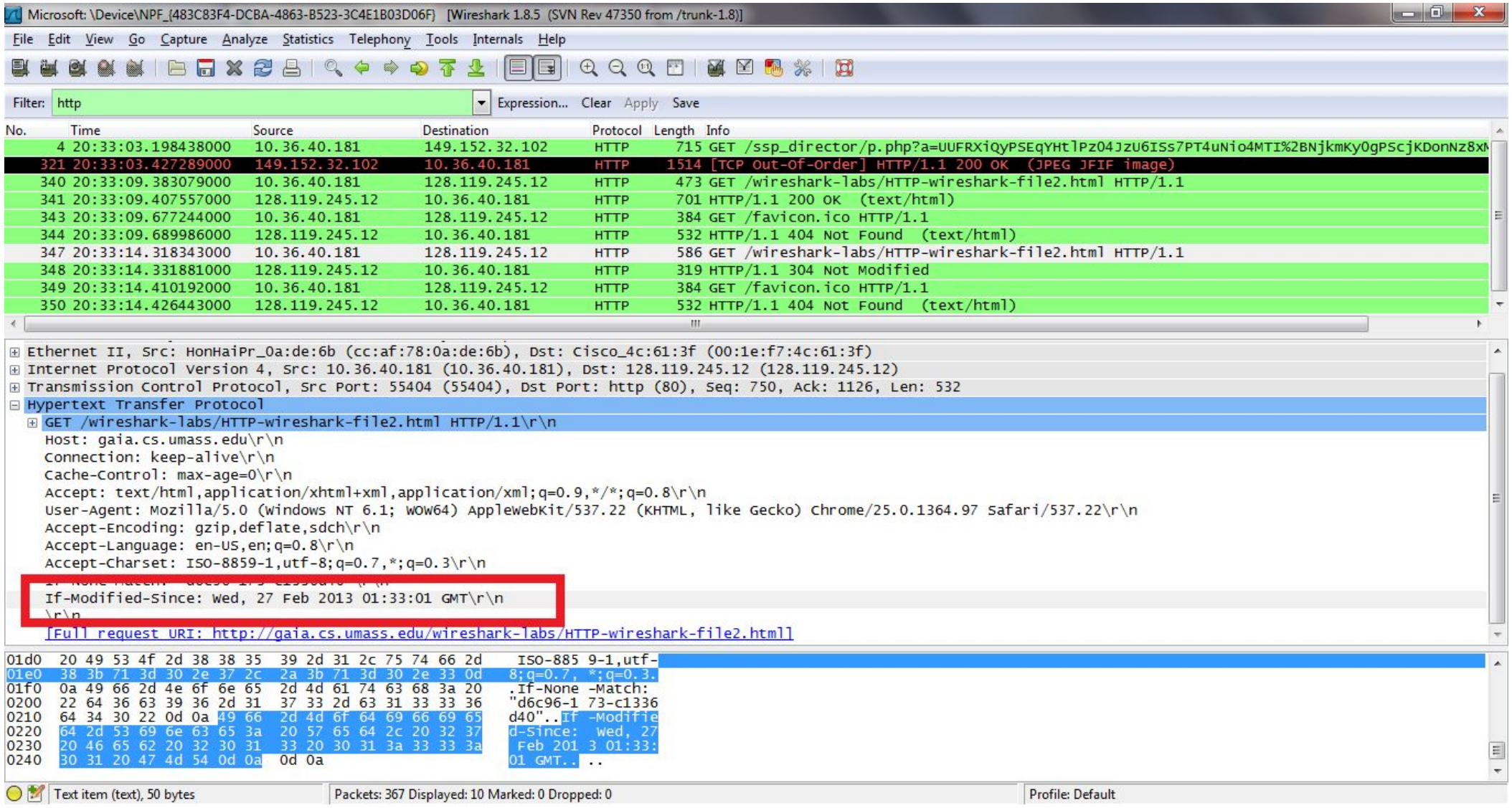
- one week later, another browser requests the **same object via the cache**, and the object is still in the cache.
- Since this object may have been modified at the Web server in the past week, the cache performs an **up-to-date check by issuing a conditional GET**.
- Specifically, the cache sends:
  - **GET /fruit/kiwi.gif HTTP/1.1**
  - **Host: www.exotiquecuisine.com**
  - **If-modified-since: Wed, 9 Sep 2015 09:23:24**
- Note that the value of the **If-modified-since**: header line is exactly **equa**l to the value of the **Last-Modified: header line that was sent by the server one week ago.**
- This conditional GET is telling the server to **send the object only if the object has been modified** since the specified date.

Conditional Get – Example



COMPUTER NETWORKS

Conditional Get (more)



THANK YOU

TEAM NETWORKS

Department of Computer Science and Engineering