

Networks & Communications

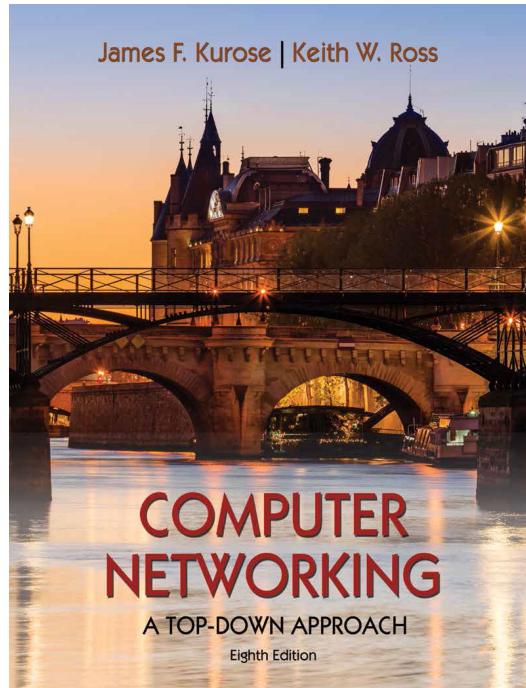
MS216
Semester 1
Session 2021-22

The Internet is ...

- ❖ A federated system ...
- ❖ of enormous scale ...
- ❖ with tremendous dynamic range and diversity ...
- ❖ that is asynchronous in operation ...
- ❖ failure prone ...
- ❖ and constantly evolving

Text Book

Class textbook:
Computer Networking: A Top-Down Approach (8th ed.)
J.F. Kurose, K.W. Ross
Pearson, 2020



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*



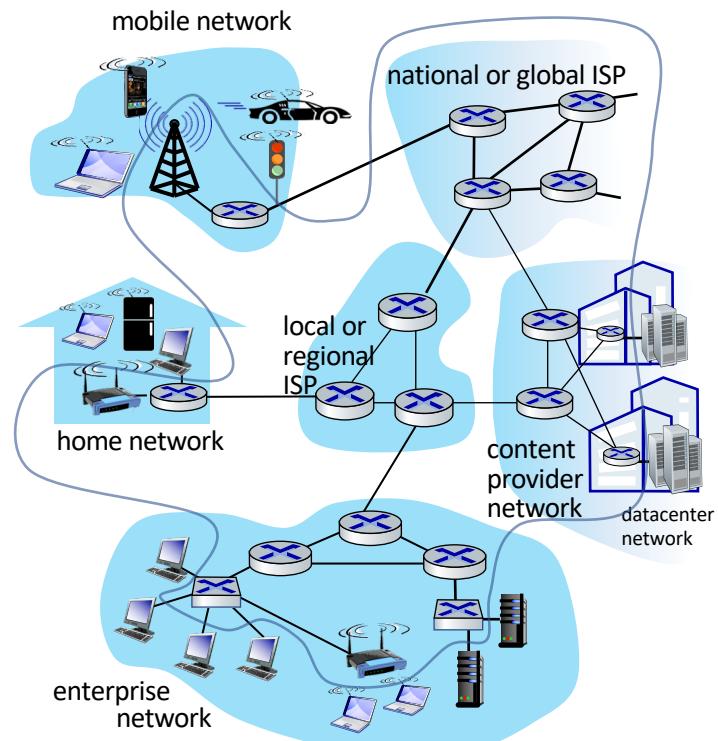
Communication links

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



Networks

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



“Fun” Internet-connected devices



Amazon Echo



Internet refrigerator



Security Camera



Internet phones



IP picture frame



Slingbox: remote control cable TV



Pacemaker & Monitor



Web-enabled toaster + weather forecaster



Gaming devices



sensorized, bed mattress



Fitbit



Tweet-a-watt:
monitor energy use



bikes



cars

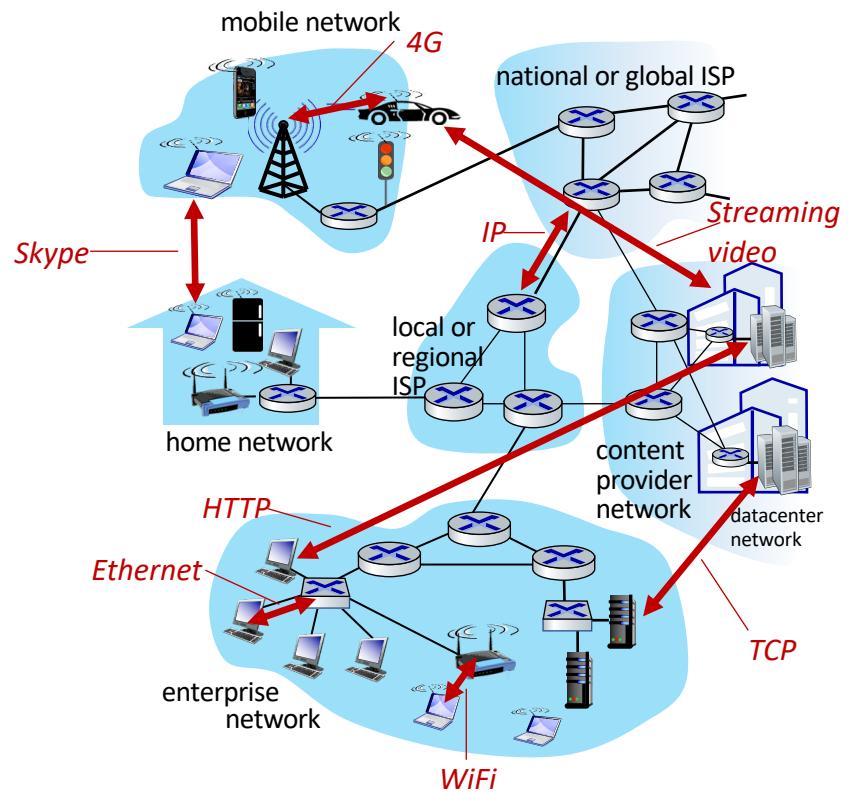


scooters

Others?

The Internet: a “nuts and bolts” view

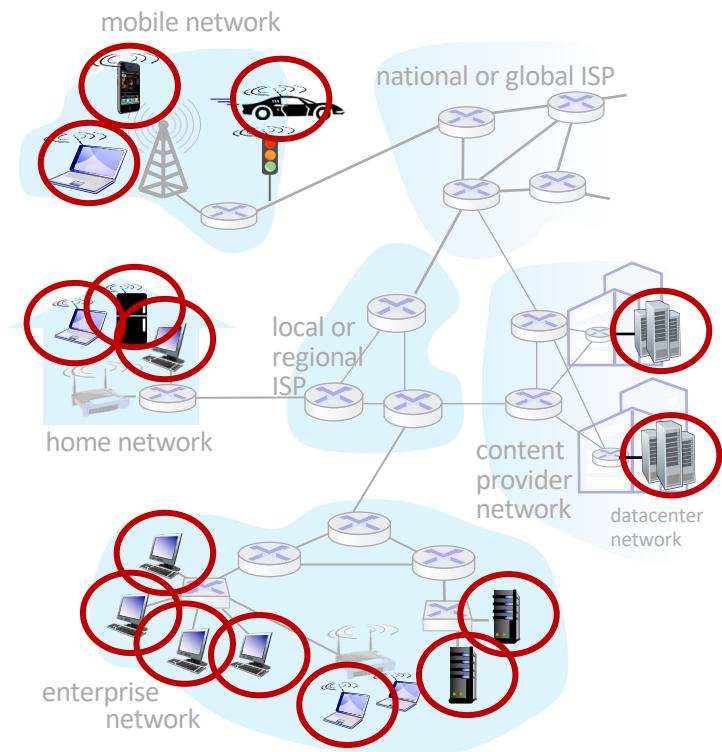
- *Internet: “network of networks”*
 - Interconnected ISPs
- *protocols are everywhere*
 - control sending, receiving of messages
 - e.g., HTTP (Web), streaming video, Skype, TCP, IP, WiFi, 4G, Ethernet
- *Internet standards*
 - RFC: Request for Comments
 - IETF: Internet Engineering Task Force



A closer look at Internet structure

Network edge:

- hosts: clients and servers
- servers often in data centers



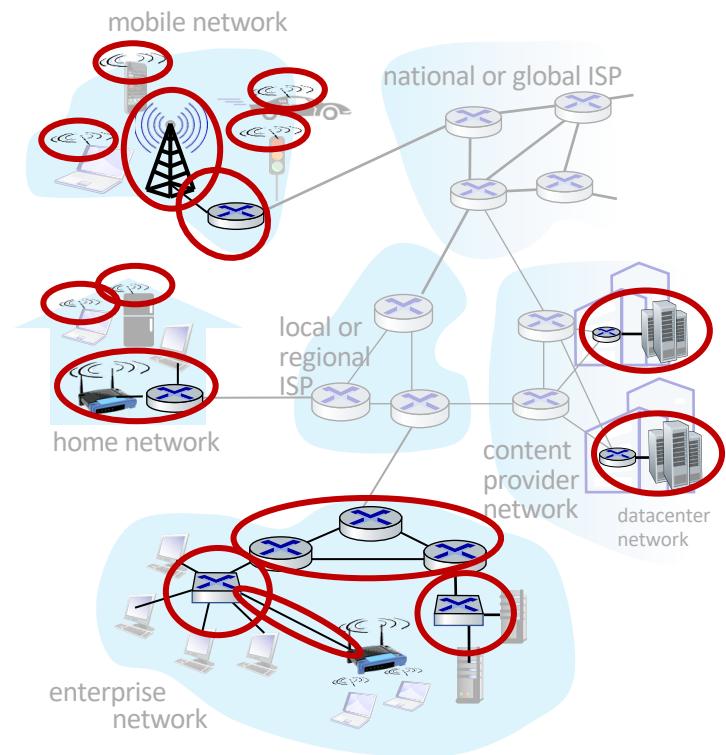
A closer look at Internet structure

Network edge:

- hosts: clients and servers
- servers often in data centers

Access networks, physical media:

- wired, wireless communication links



A closer look at Internet structure

Network edge:

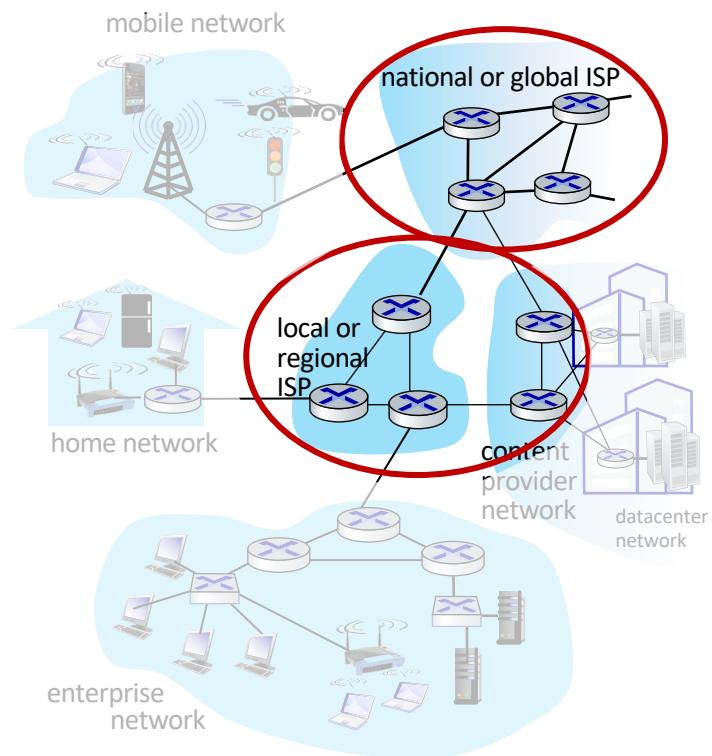
- hosts: clients and servers
- servers often in data centers

Access networks, physical media:

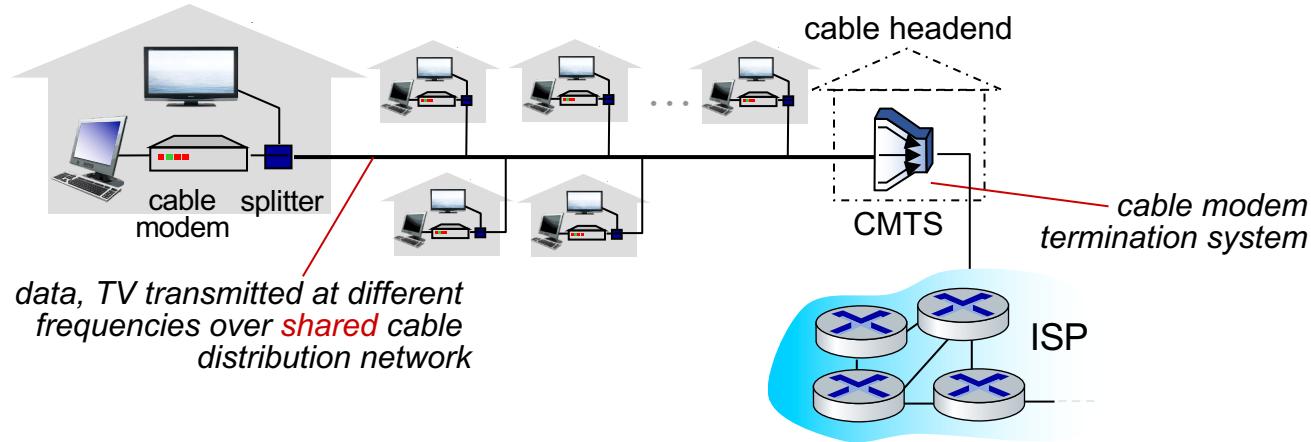
- wired, wireless communication links

Network core:

- interconnected routers
- network of networks

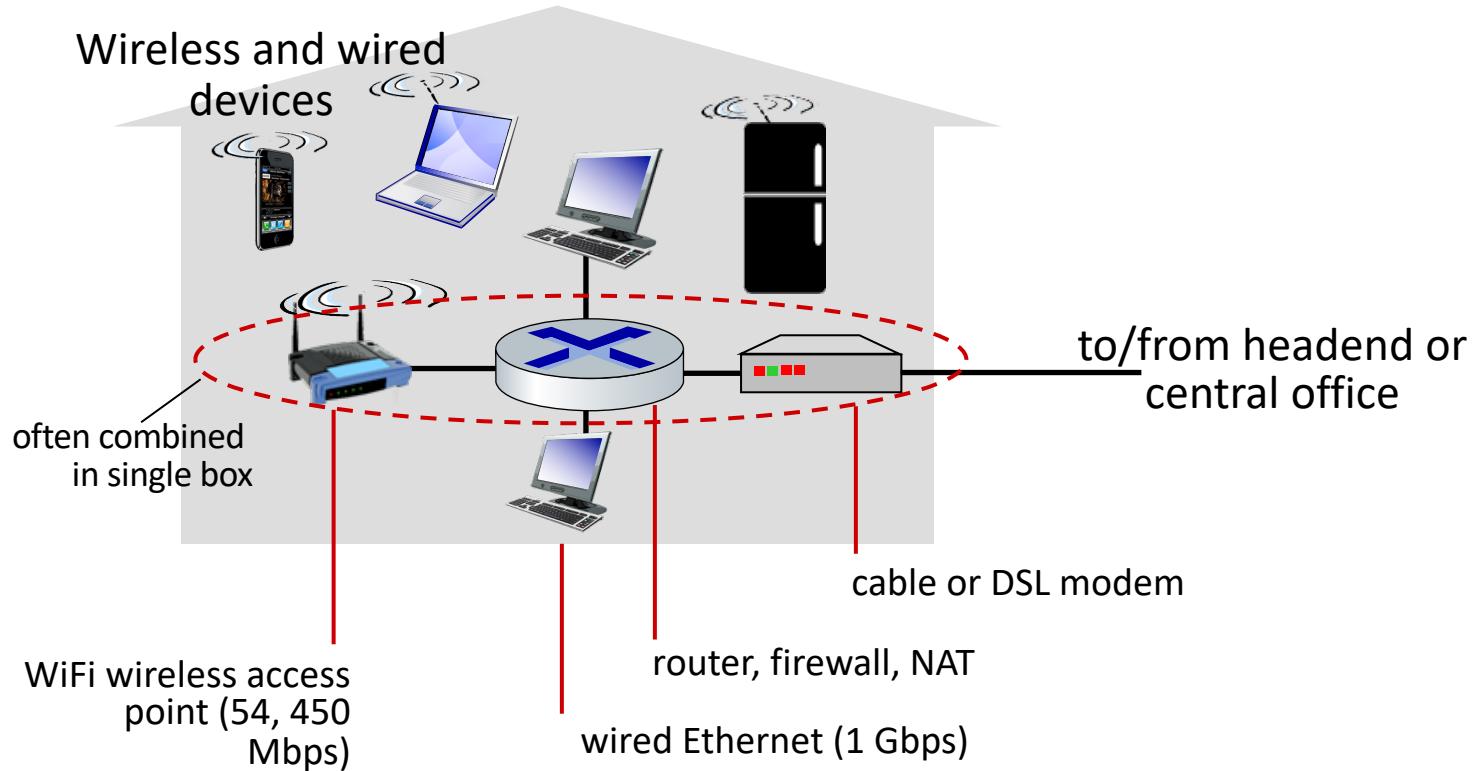


Access networks: cable-based access

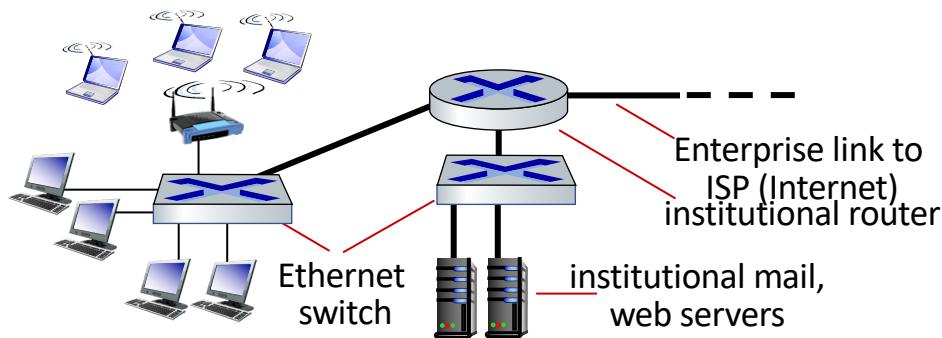


- HFC: hybrid fiber coax
 - asymmetric: up to 40 Mbps – 1.2 Gbs downstream transmission rate, 30-100 Mbps upstream transmission rate
- **network** of cable, fiber attaches homes to ISP router
 - homes **share access network** to cable headend

Access networks: home networks



Access networks: NUIG network

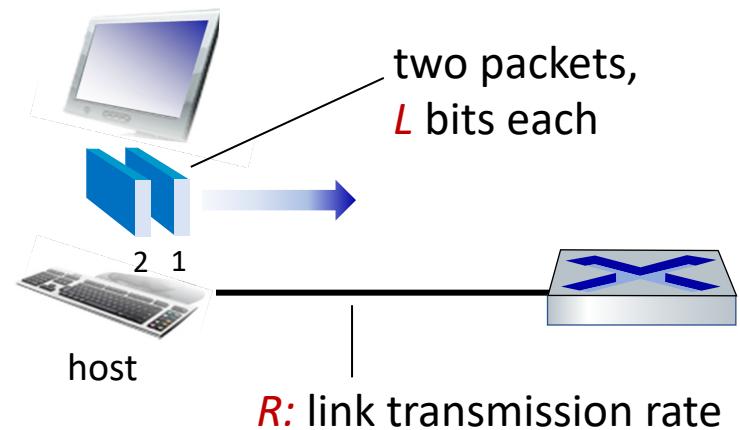


- companies, universities, etc.
- mix of wired, wireless link technologies, connecting a mix of switches and routers (we'll cover differences shortly)
 - Ethernet: wired access at 100Mbps, 1Gbps, 10Gbps
 - WiFi: wireless access points at 11, 54, 450 Mbps

Host: sends *packets* of data

host sending function:

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



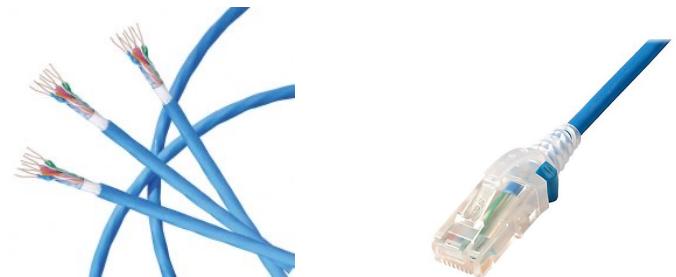
$$\text{packet transmission delay} = \frac{\text{time needed to transmit } L\text{-bit packet into link}}{R \text{ (bits/sec)}} = \frac{L \text{ (bits)}}{R \text{ (bits/sec)}}$$

Links: physical media

- **bit:** propagates between transmitter/receiver pairs
- **physical link:** what lies between transmitter & receiver
- **guided media:**
 - signals propagate in solid media: copper, fiber, coax
- **unguided media:**
 - signals propagate freely, e.g., radio

Twisted pair (TP)

- two insulated copper wires
 - Category 5: 100 Mbps, 1 Gbps Ethernet
 - Category 6: 10Gbps Ethernet



Links: physical media

Coaxial cable:

- two concentric copper conductors
- bidirectional
- broadband:
 - multiple frequency channels on cable
 - 100's Mbps per channel



Fiber optic cable:

- glass fiber carrying light pulses, each pulse a bit
- high-speed operation:
 - high-speed point-to-point transmission (10's-100's Gbps)
- low error rate:
 - repeaters spaced far apart
 - immune to electromagnetic noise



Links: physical media

Wireless radio

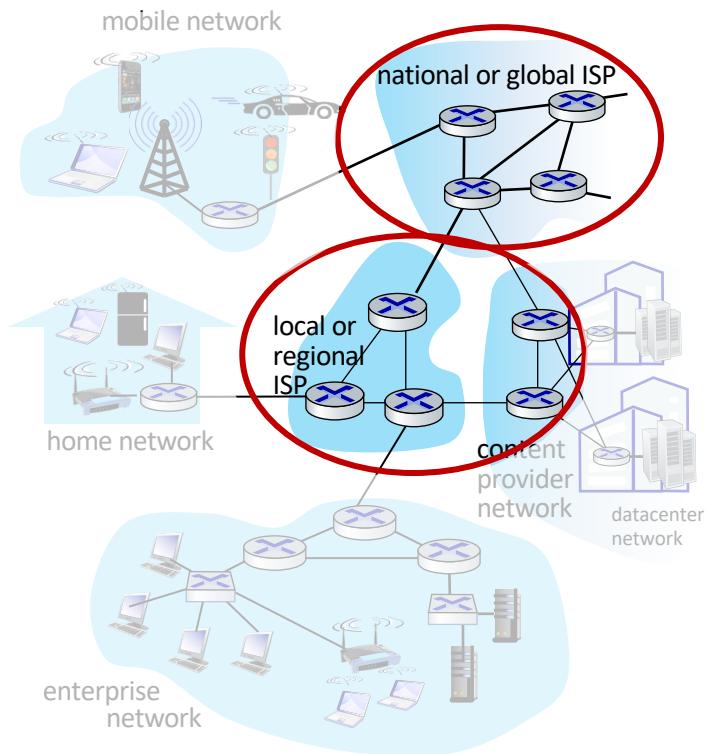
- signal carried in various “bands” in electromagnetic spectrum
- no physical “wire”
- broadcast, “half-duplex” (sender to receiver)
- propagation environment effects:
 - reflection
 - obstruction by objects
 - Interference/noise

Radio link types:

- **Wireless LAN (WiFi)**
 - 10-100's Mbps; 10's of meters
- **wide-area** (e.g., 4G cellular)
 - 10's Mbps over ~10 Km
- **Bluetooth:** cable replacement
 - short distances, limited rates
- **terrestrial microwave**
 - point-to-point; 45 Mbps channels
- **satellite**
 - up to 45 Mbps per channel
 - 270 msec end-end delay

The network core

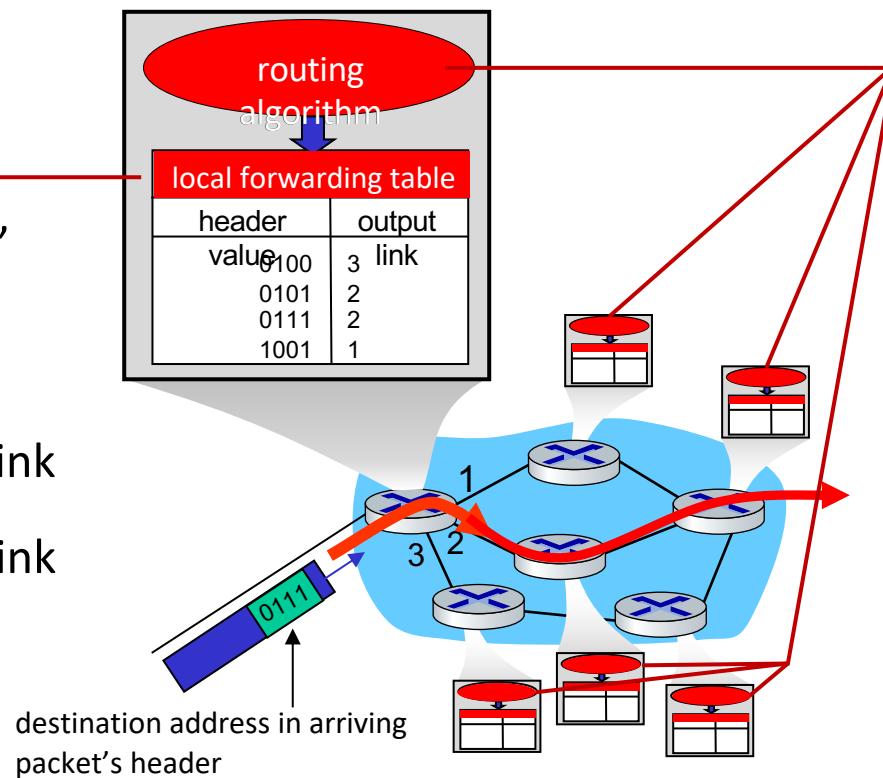
- mesh of interconnected routers
- **packet-switching**: hosts break application-layer messages into *packets*
 - network **forwards** packets from one router to the next, across links on path from **source to destination**



Two key network-core functions

Forwarding:

- aka “switching”
- *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

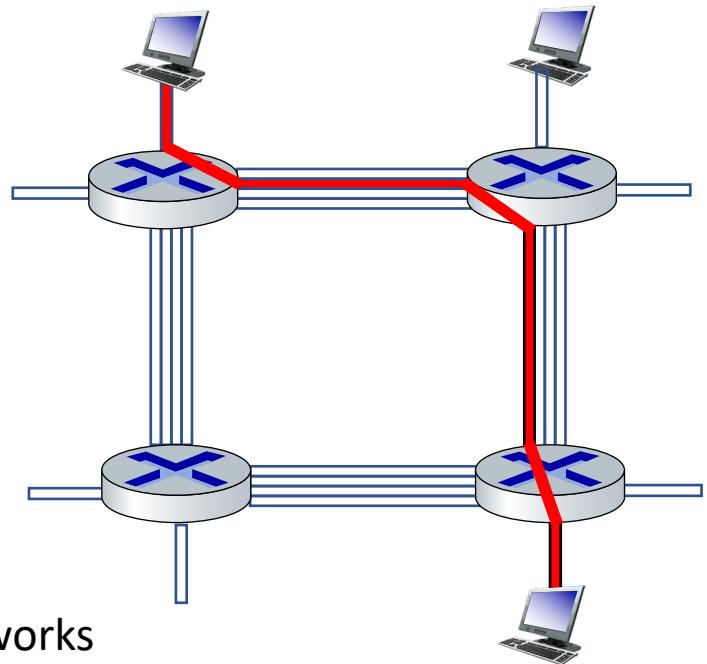




Alternative to packet switching: circuit switching

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

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



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

Attendance

Check-In is currently running .

Students can check in until check in period closes or is ended.

6 8 9 6

End Check In

Layered Internet protocol stack

Protocol “layers” and reference models

Networks are complex,
with many “pieces”:

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

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

- and/or our *discussion*
of networks?

Example: organization of air travel



end-to-end transfer of person plus baggage

ticket (purchase)
baggage (check)
gates (load)
runway takeoff
airplane routing

ticket (complain)
baggage (claim)
gates (unload)
runway landing
airplane routing

airplane routing

How would you *define/discuss* the system of airline travel?

- a series of steps, involving many services

Example: organization of air travel



layers: each layer implements a service

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

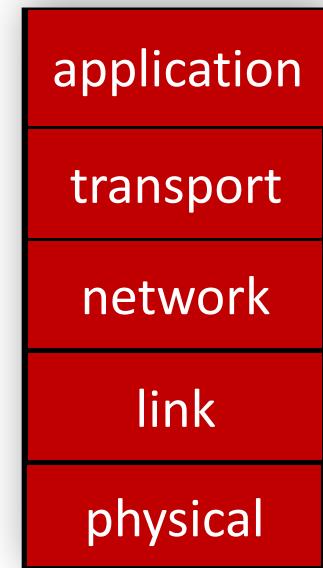
Why layering?

Approach to designing/discussing complex systems:

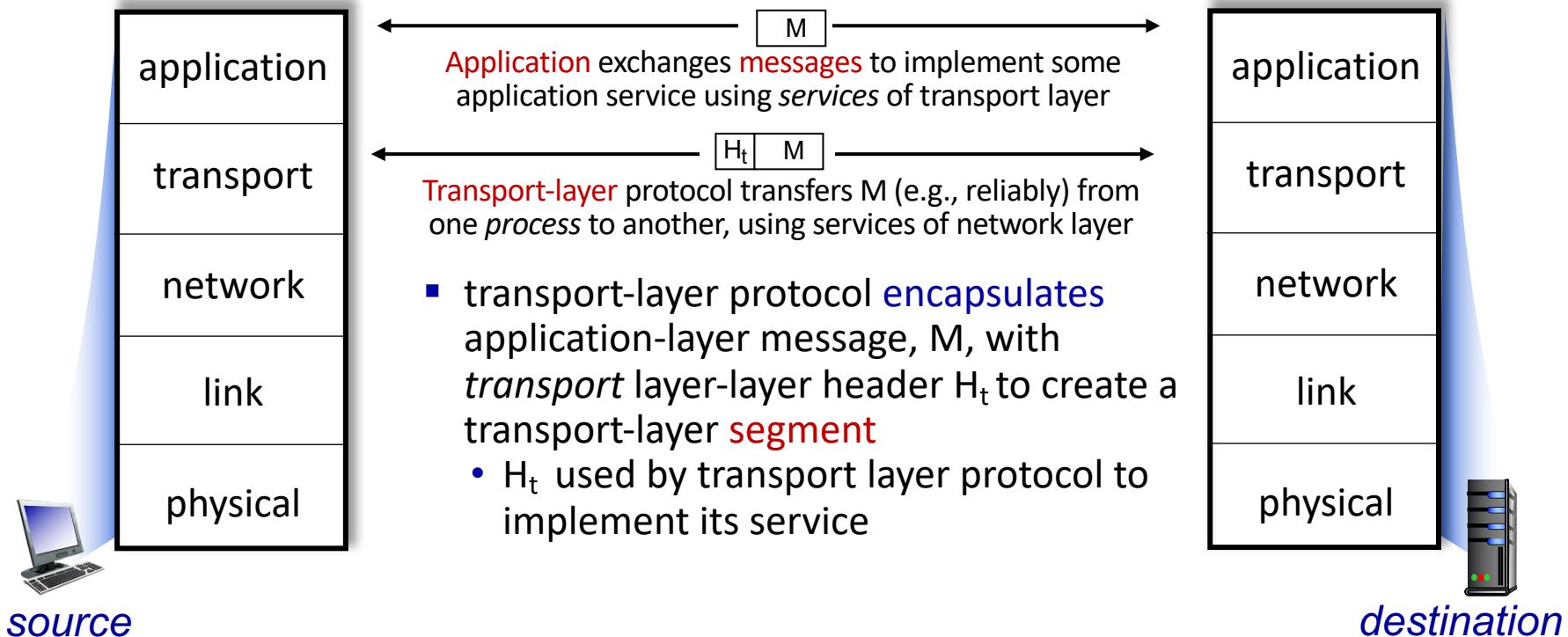
- explicit structure allows identification, relationship of system's pieces
 - layered *reference model* for discussion
- modularization eases maintenance, updating of system
 - change in layer's service *implementation*: transparent to rest of system
 - e.g., change in gate procedure doesn't affect rest of system

Layered Internet protocol stack

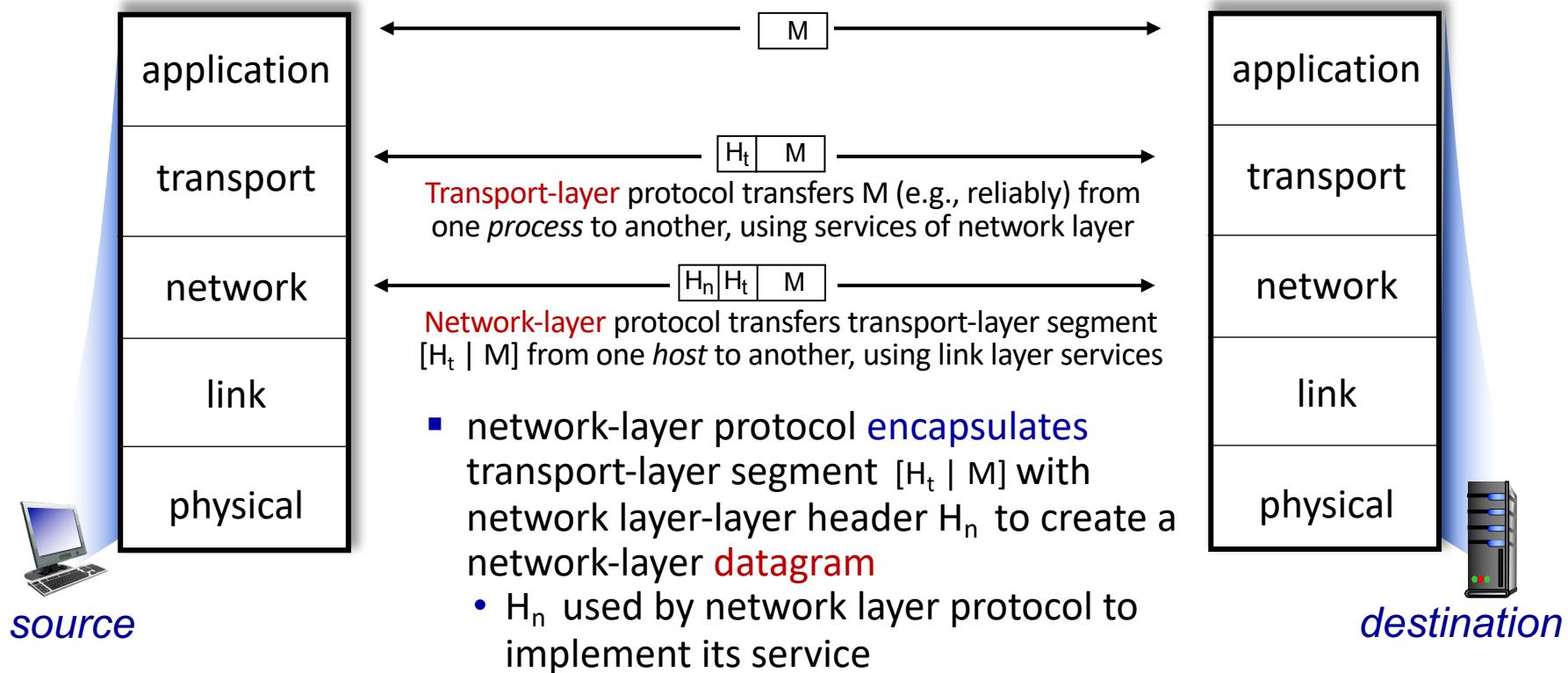
- ***application***: supporting network applications
 - HTTP, IMAP, SMTP, DNS
- ***transport***: process-process data transfer
 - TCP, UDP
- ***network***: routing of datagrams from source to destination
 - IP, routing protocols
- ***link***: data transfer between neighboring network elements
 - Ethernet, 802.11 (WiFi), PPP
- ***physical***: bits “on the wire”



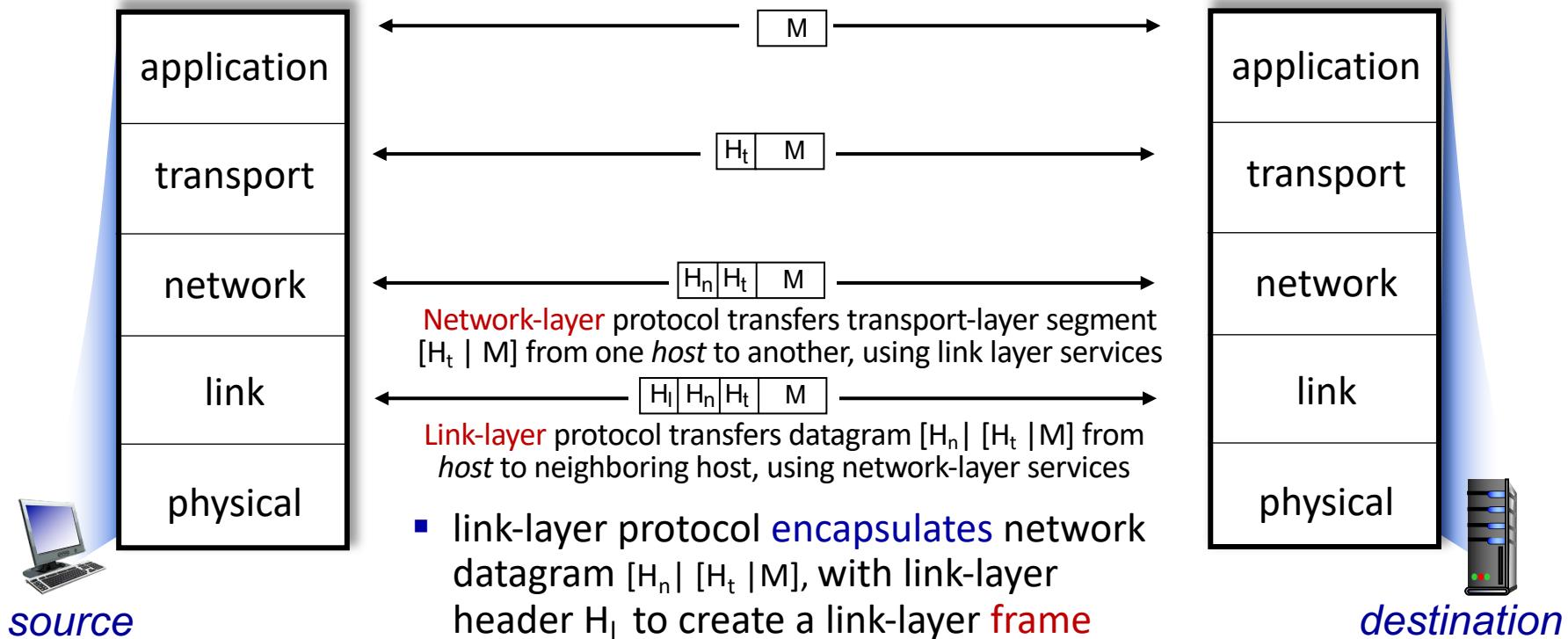
Services, Layering and Encapsulation



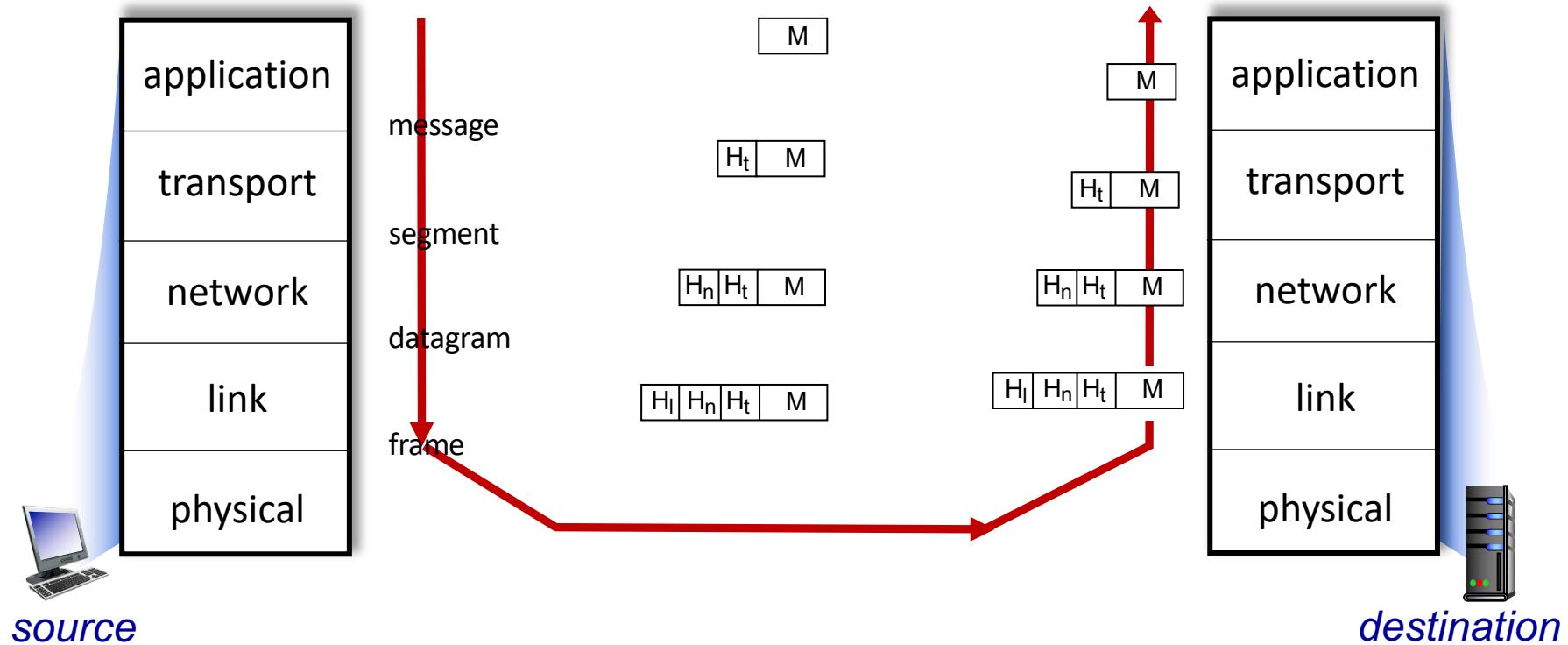
Services, Layering and Encapsulation



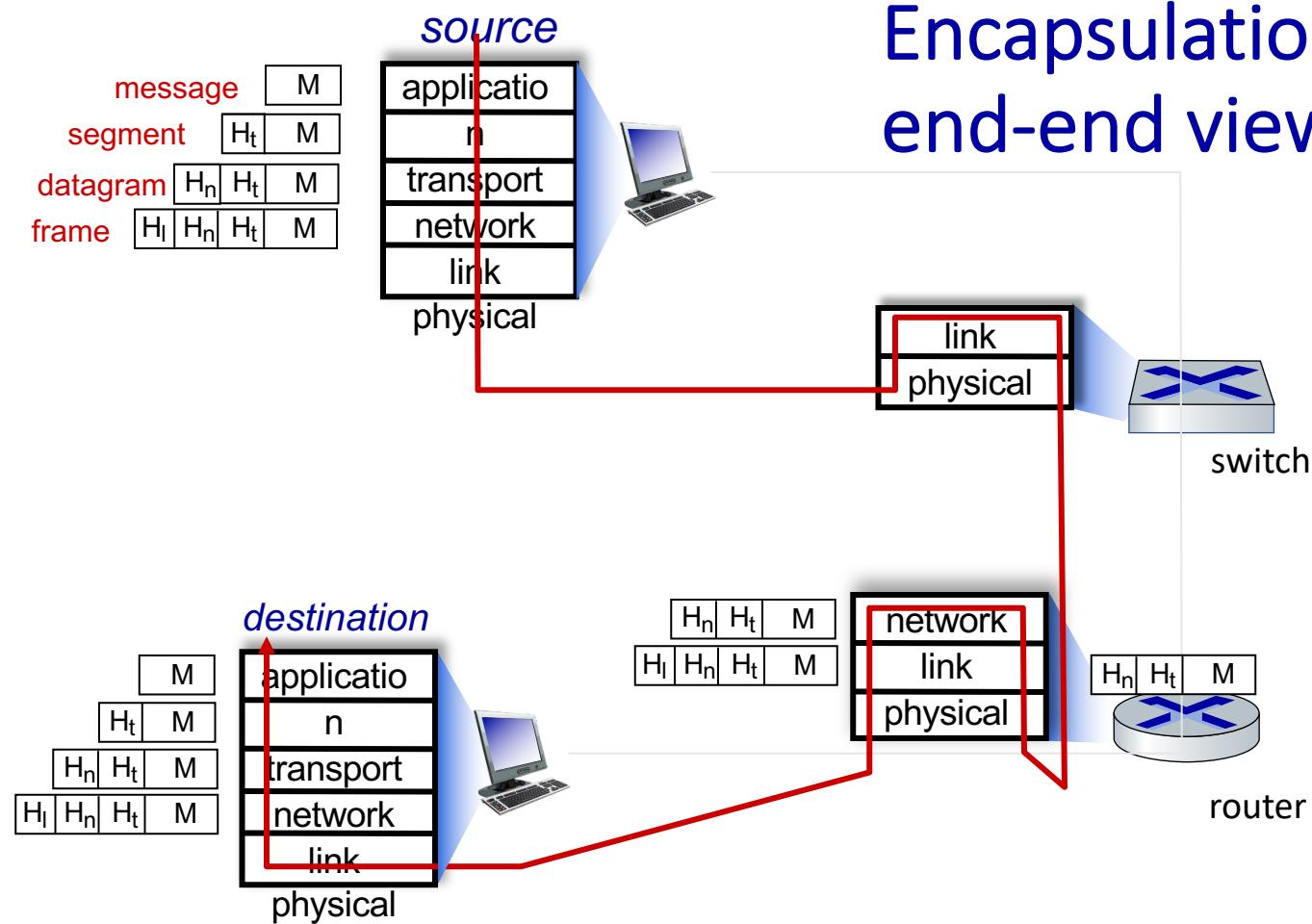
Services, Layering and Encapsulation



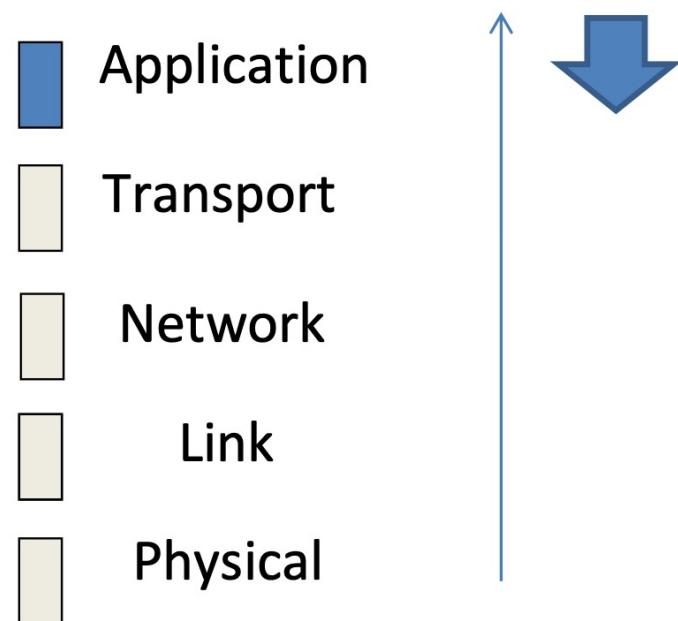
Services, Layering and Encapsulation



Encapsulation: an end-end view

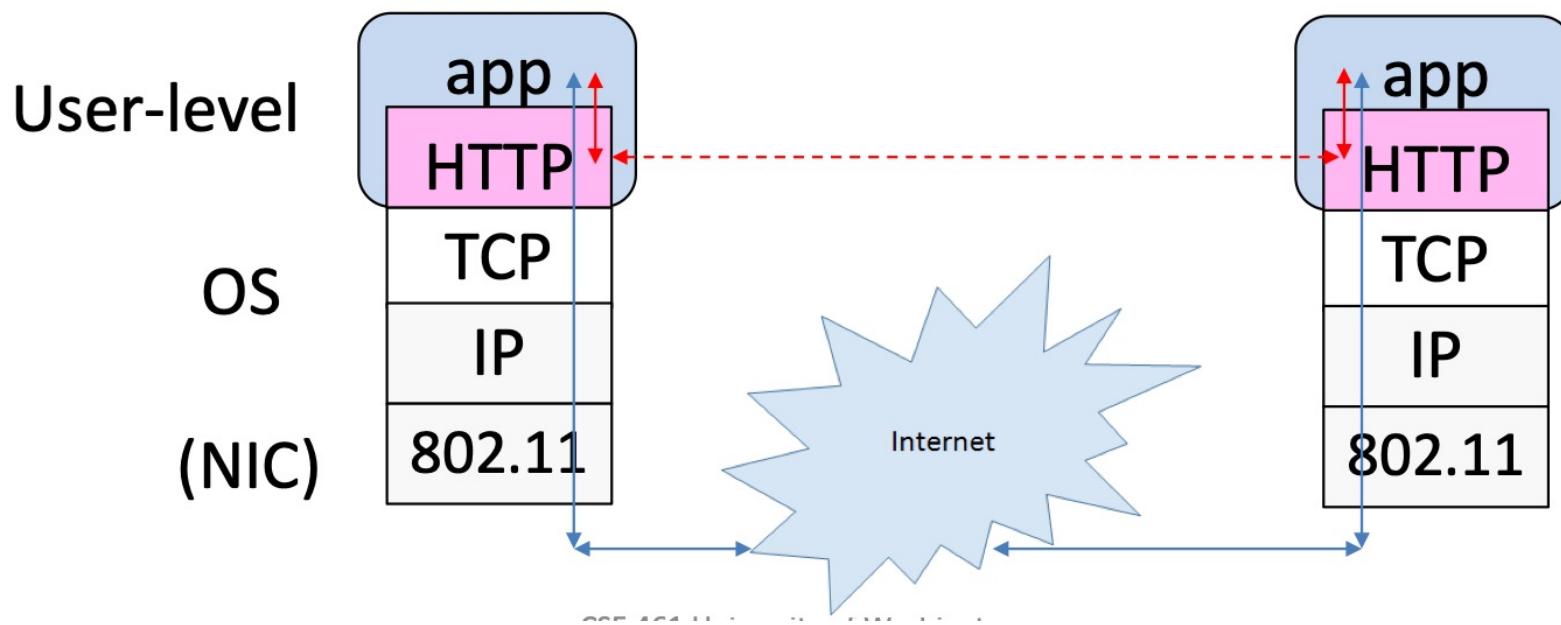


Application Layer



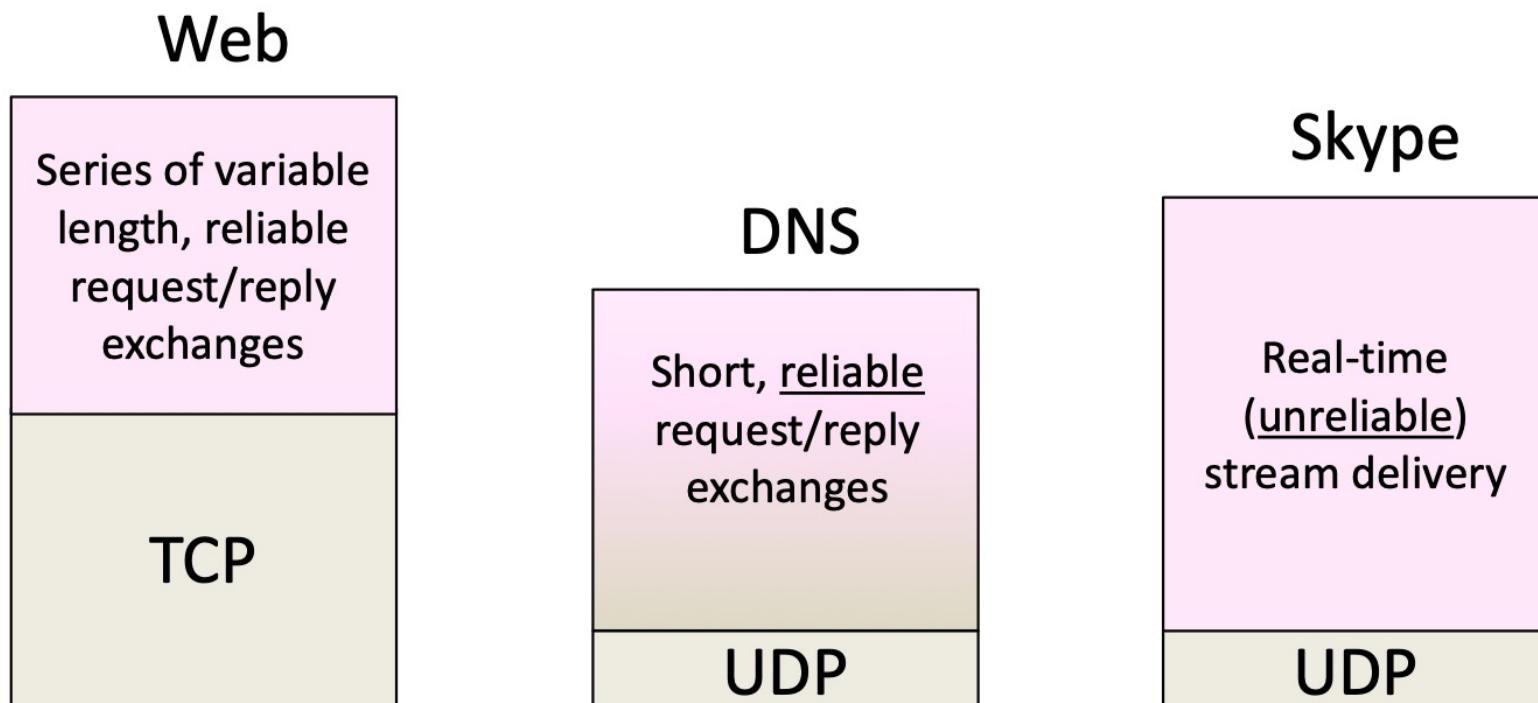
Implementation

- Application layer protocols are often part of “the app”
 - Libraries running in untrusted space



Application Communication Needs

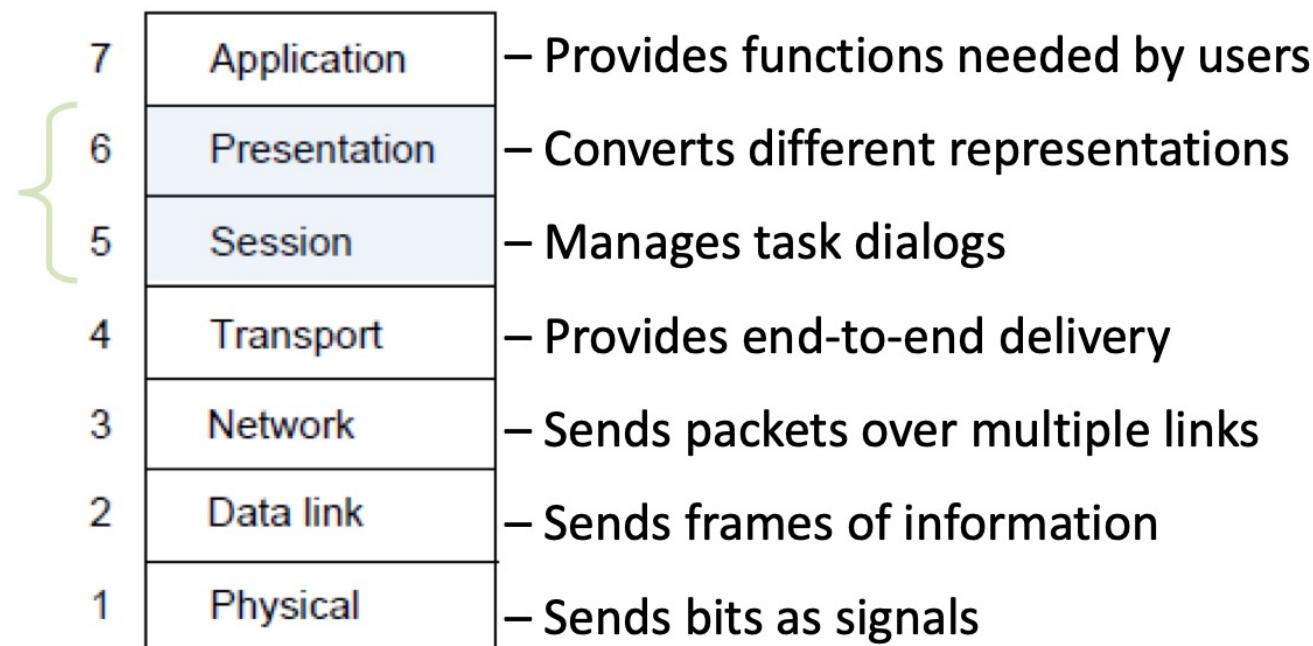
- Vary widely; must build on Transport services



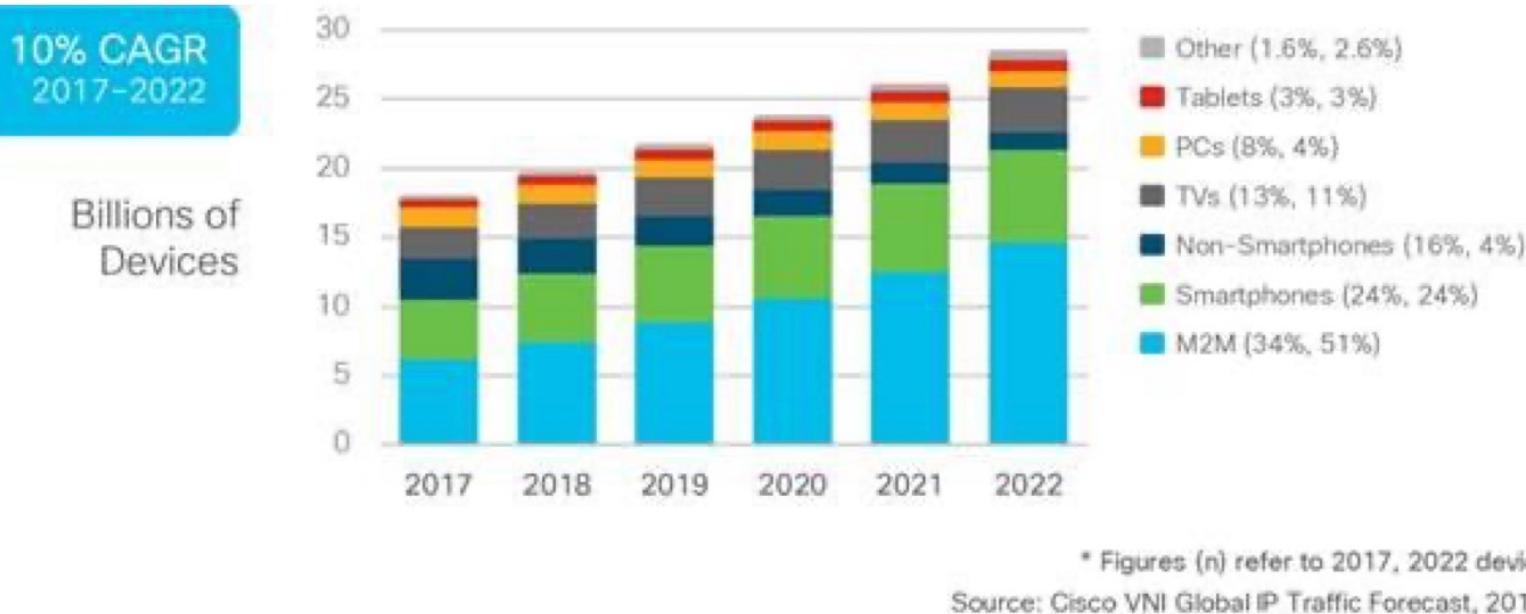
OSI Session/Presentation Layers

- Two relevant concepts ...

Considered part of the application, not strictly layered!

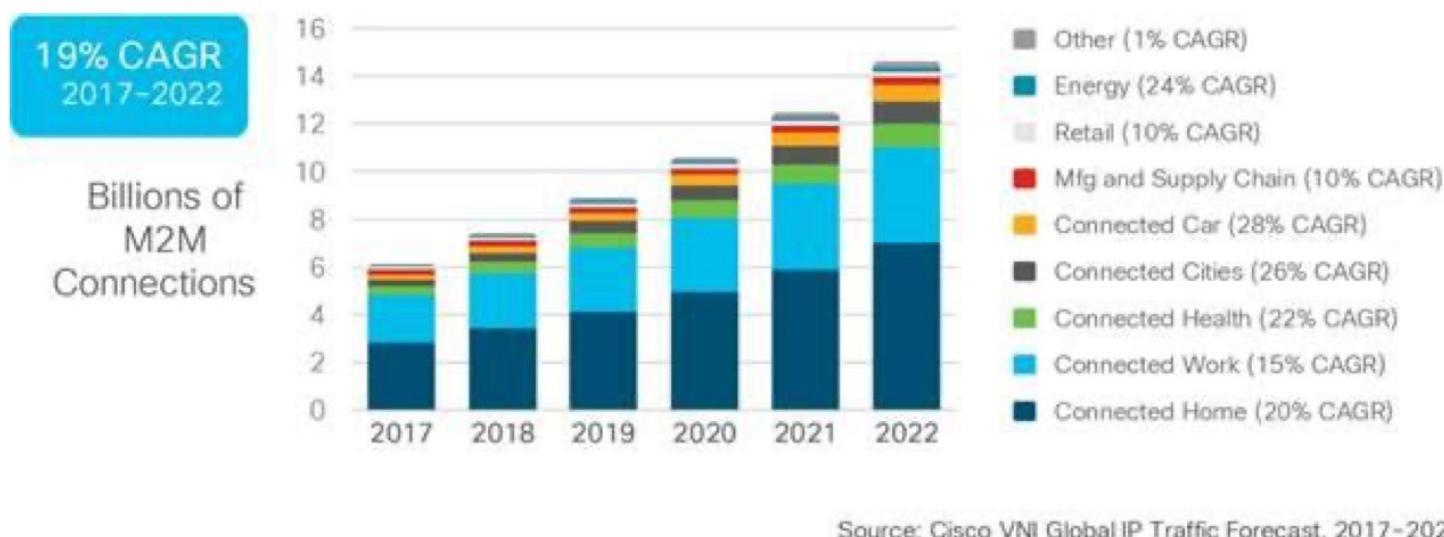


Which Devices Matter?

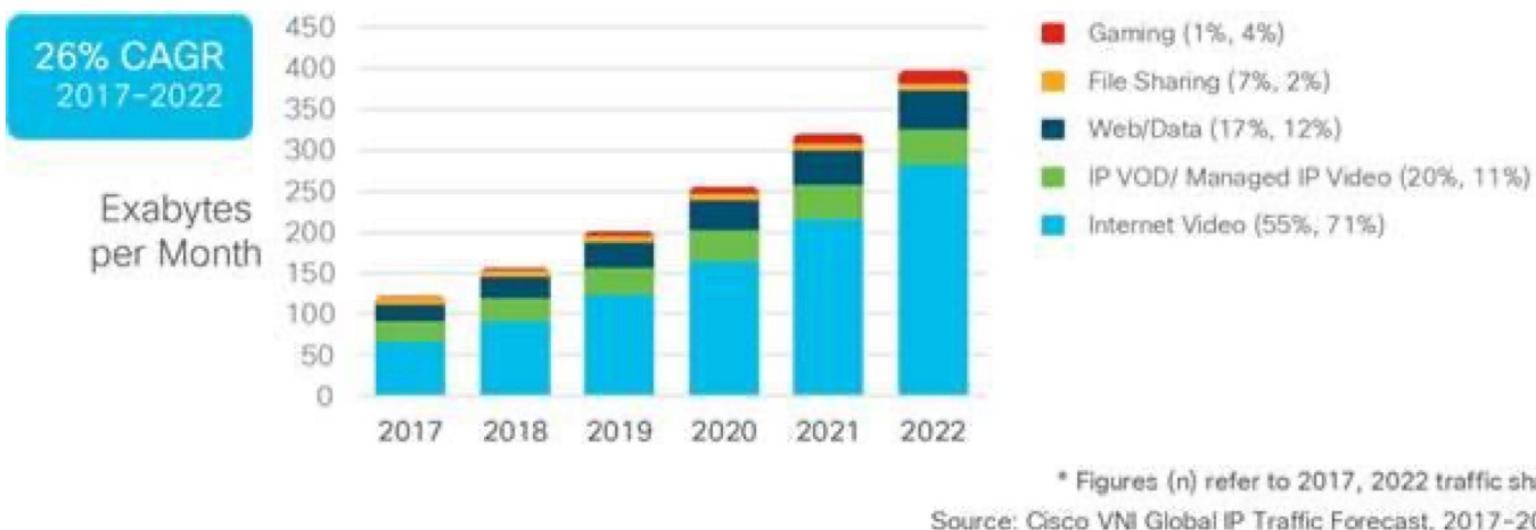


Graph of connected devices

M2M Connections By Industry



Application Data



Application layer: overview

Our goals:

- 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 and infrastructure
 - HTTP
 - SMTP, IMAP
 - DNS
 - video streaming systems, CDNs
- programming network applications
 - socket API

Some network apps

- social networking
- Web
- text messaging
- e-mail
- multi-user network games
- streaming stored video
(YouTube, Hulu, Netflix)
- P2P file sharing
- voice over IP (e.g., Skype)
- real-time video conferencing (e.g., Zoom)
- Internet search
- remote login
- ...

Q: your favorites?

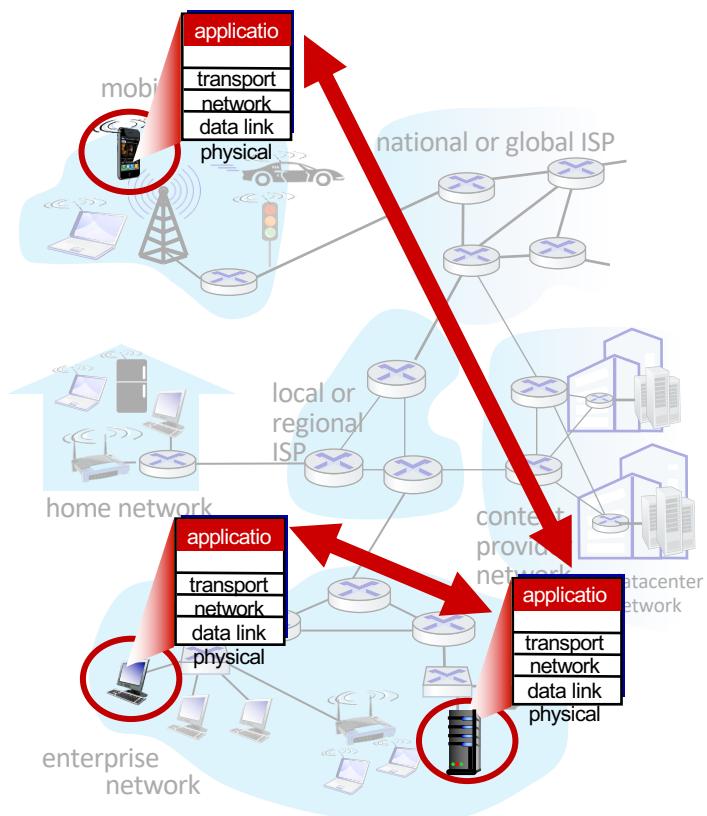
Creating a network app

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



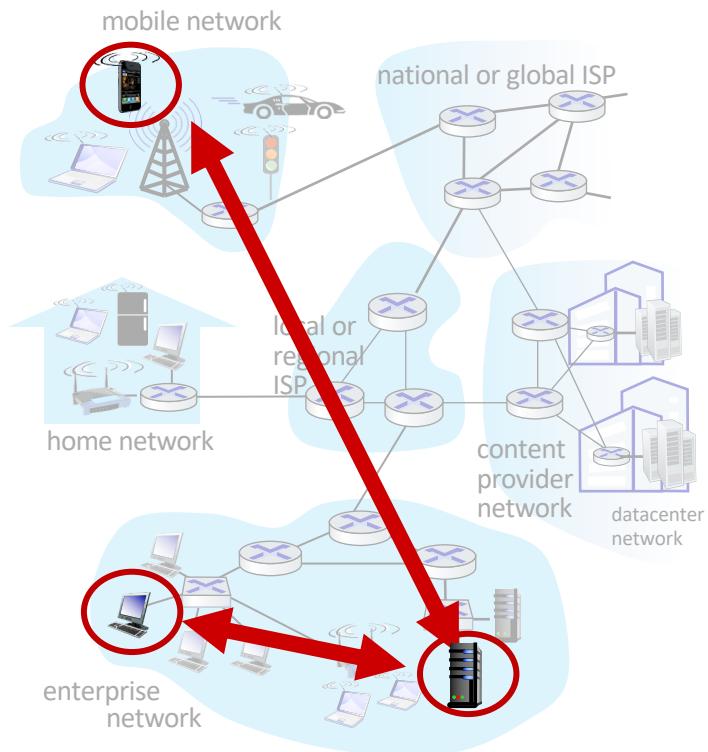
Client-server paradigm

server:

- always-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
- examples: HTTP, IMAP, FTP



Peer-peer architecture

- no always-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

