# Lecture 2
The Internet



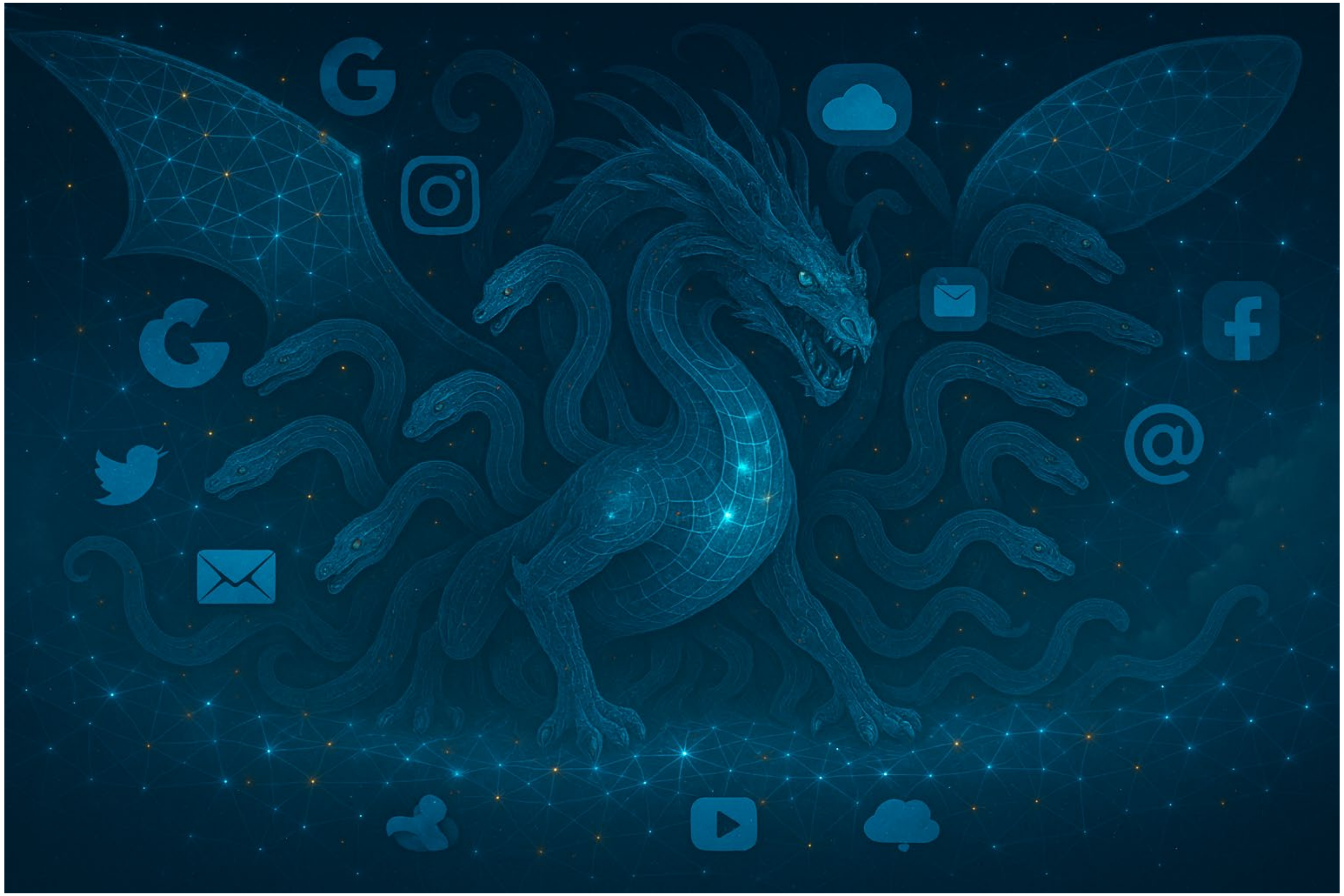
SDU

# The Internet as a Mythical Animal: The Neural Hydra

- **Appearance**:
  - **A massive, shimmering creature with countless serpentine heads, each representing a node or server.**
  - Its body is a translucent mesh of fiber-optic nerves pulsing with light, resembling a neural network.
  - Wings made of satellite arrays and cloud vapor, constantly shifting and expanding.
  - Eyes on each head glow with data streams, scanning and communicating in real time.
- **Behavior**:
  - Constantly growing and regenerating—cut off one head (a server or site), and two more appear.
  - It feeds on information, thrives in chaos, and adapts instantly to new environments.
  - **It's both omnipresent and elusive—visible everywhere yet impossible to fully grasp.**
- **Habitat**:
  - Exists in the digital ether, nesting in data centers, roaming through cables, satellites, and wireless signals.
- **Temperament**:
  - **Neutral but reactive. It mirrors the intentions of those who interact with it – benevolent to some, dangerous to others.**

*From MS copilot prompt: "How would you vizualize the internet as an animal*"

# The Internet as a Mythical Animal: The Neural Hydra



# Subjects of today:

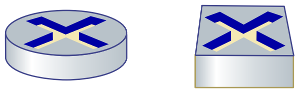
- The Internet
- The Network edge
- The Network core
- Performance
- The Protocol layers of the Internet
- Lab exercise

# Subjects of today:

- **The Internet**
- The Network edge
- The Network core
- Performance
- The Protocol layers of the Internet
- Lab exercise

# The Internet: a "nuts and bolts" view

Billions of connected computing *devices*:
- *hosts = end systems*
- running *network apps* at Internet's "edge"

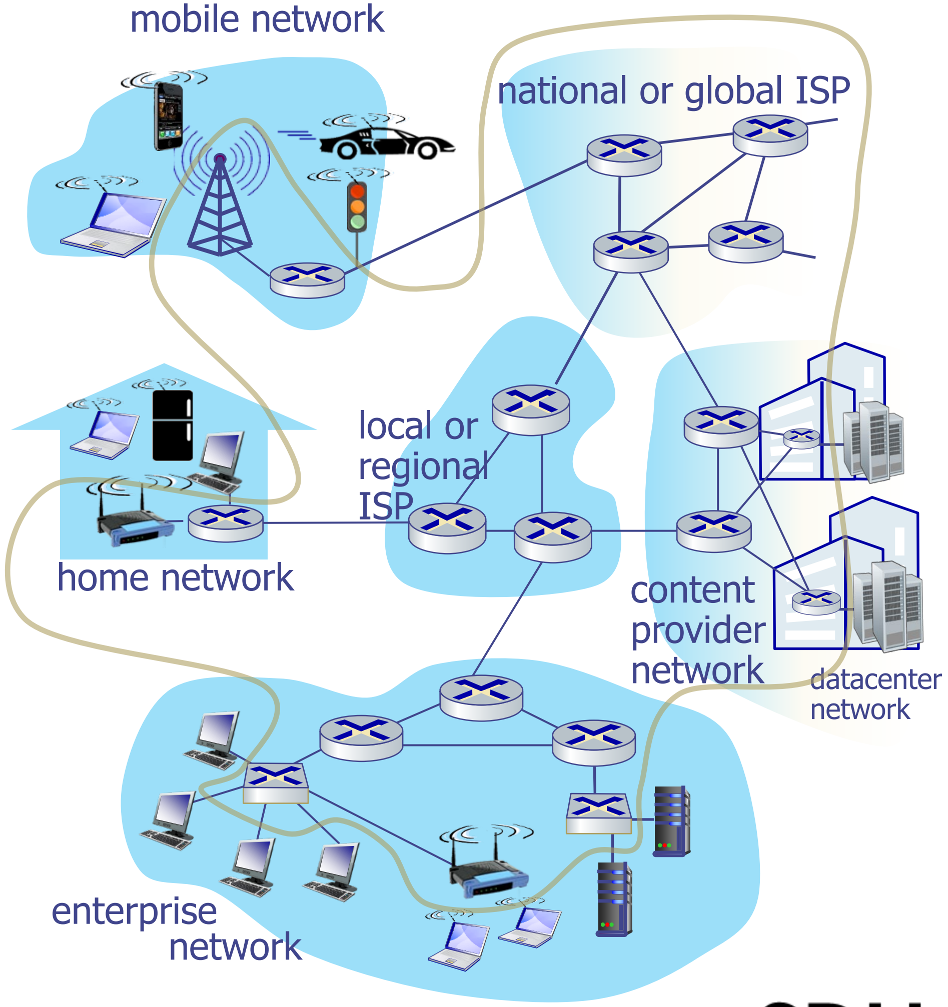


*Packet switches*: forward packets (chunks of data)
- *routers, switches*

*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

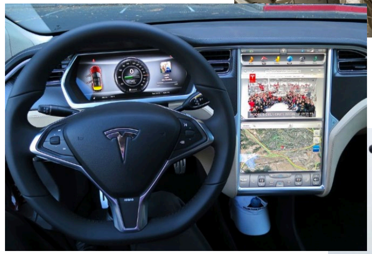

Internet refrigerator


IP picture frame


Pacemaker & Monitor


Tweet-a-watt: monitor energy use

bikes

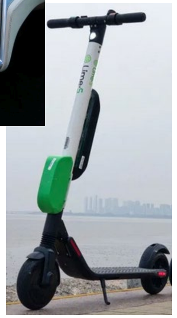


cars


Web-enabled toaster + weather forecaster


Security Camera


Slingbox: remote control cable TV


AR devices


scooters


Internet phones


Gaming devices


sensorized bed mattress


Fitbit


diapers

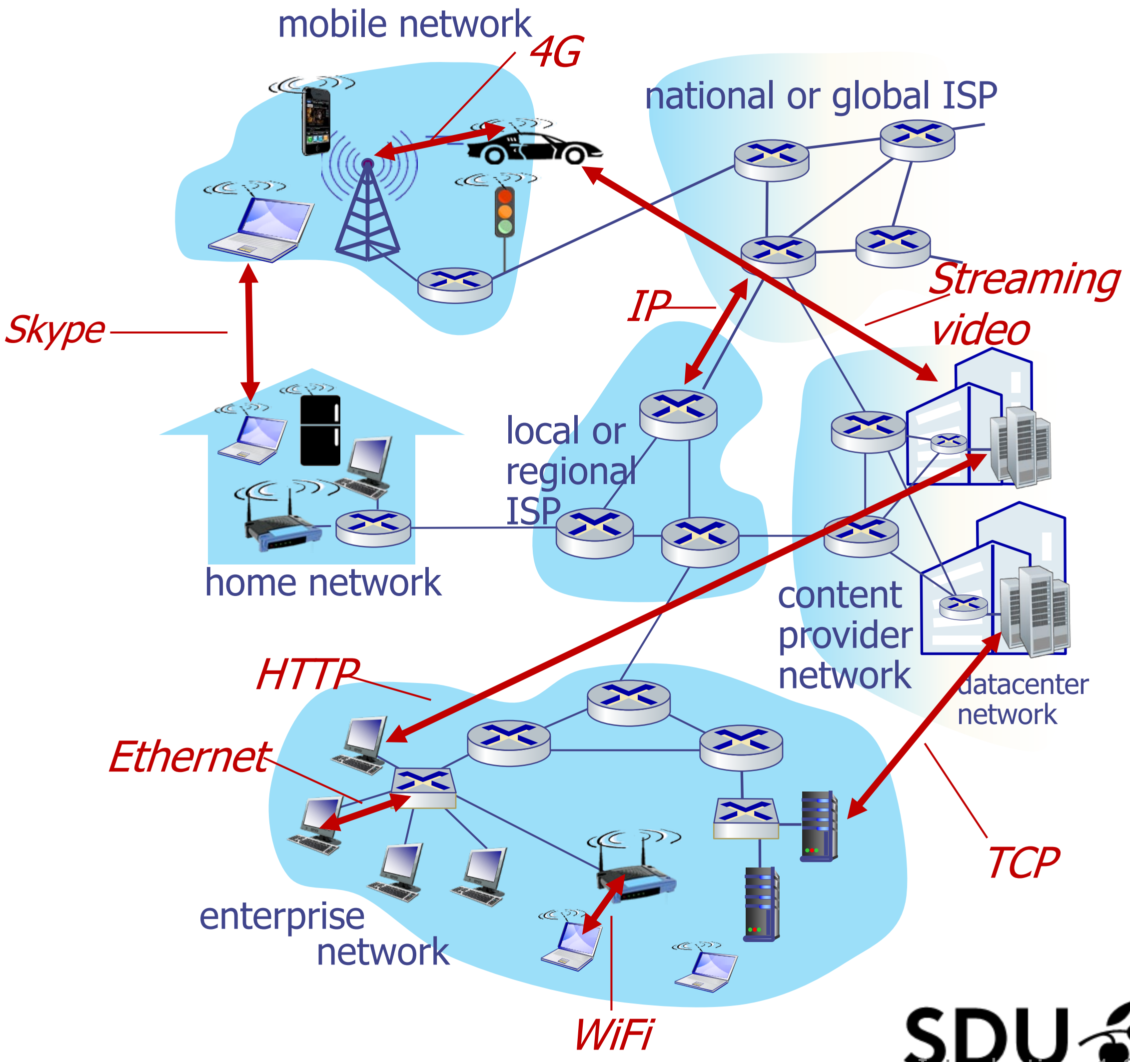
*Others?*

# The Internet: a "nuts and bolts" view

- *Internet:* "network of networks"
  - Interconnected ISPs

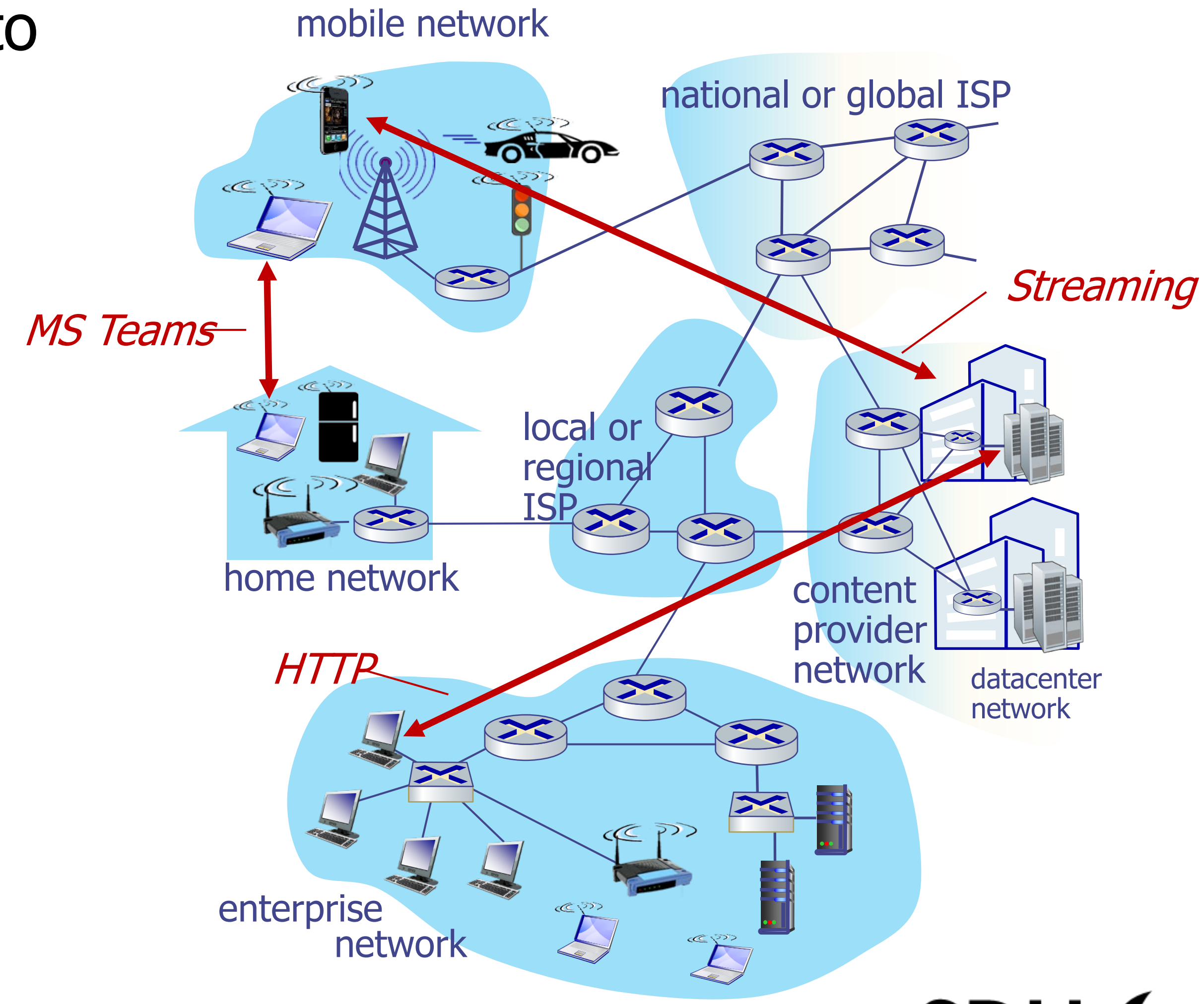
- *protocols* are *everywhere*
  - control sending, receiving of messages
  - e.g., HTTP (Web), streaming video, Skype, TCP, IP, WiFi, 4/5G, Ethernet

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



# The Internet: a "services" view

- *Infrastructure* that provides services to applications
- Provides *programming interface* to distributed applications
- The internet core transfers messages from one end system to another end system



# Protocols, a reminder

## *Human protocols:*

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

Rules for:

… 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

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

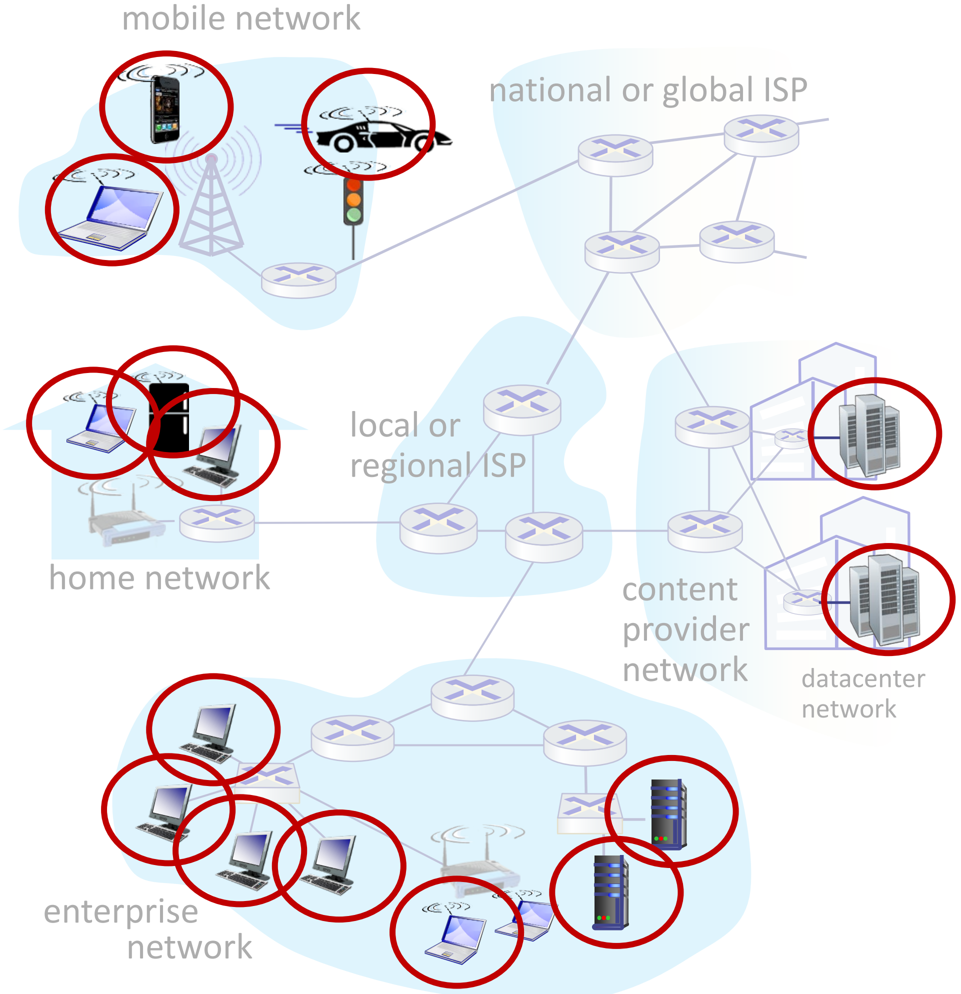
# Subjects of today:

- The Internet
- **The Network edge**
- The Network core
- Performance
- The Protocol layers of the Internet

# A closer look at Internet structure

## Network edge:

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



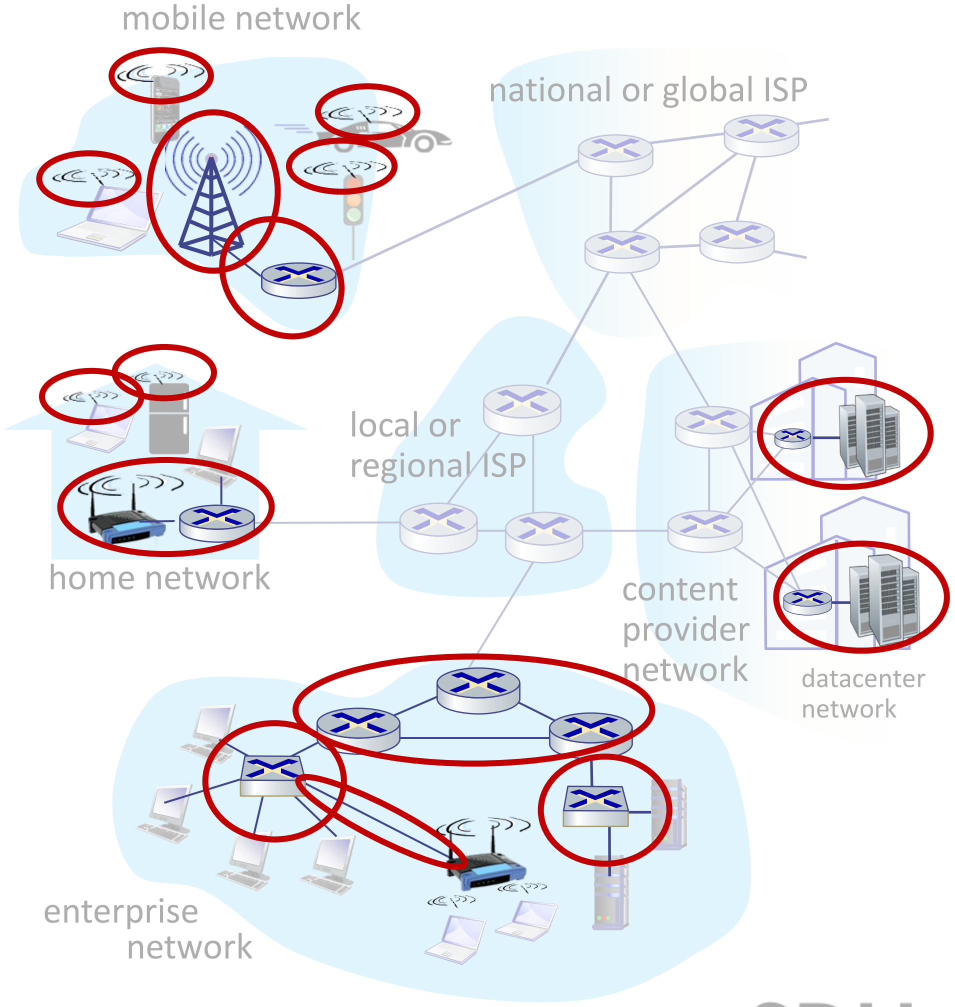
# A closer look at Internet structure

## Network edge:

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

## Access networks, physical media:

- wired, wireless communication links

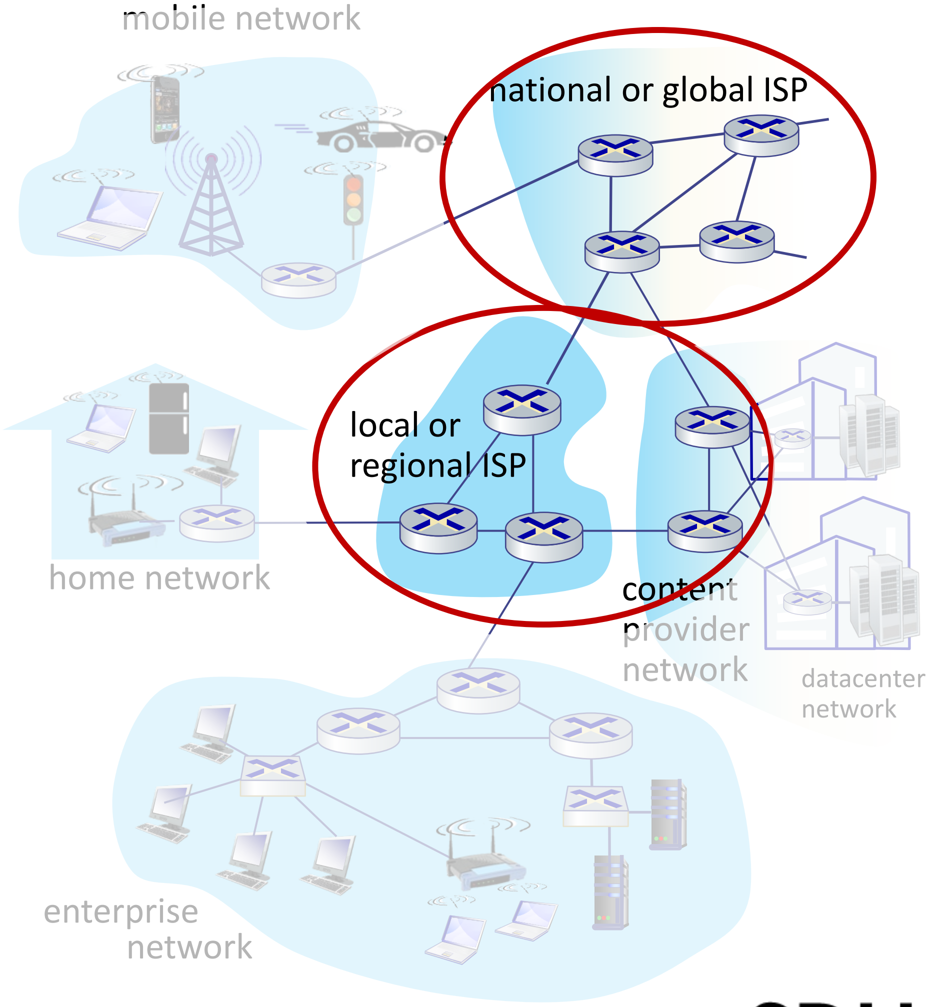


# A closer look at Internet structure

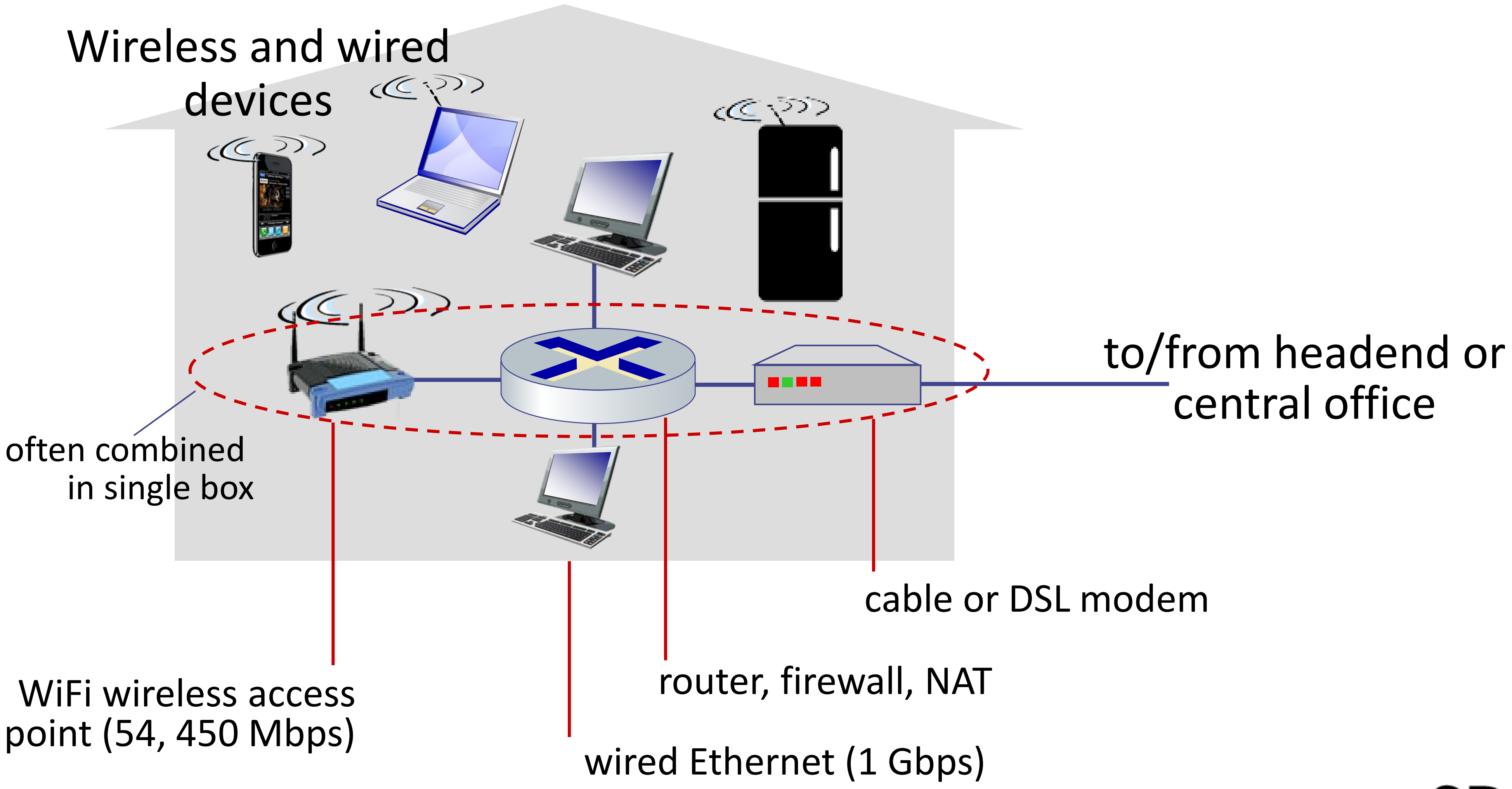
## Network edge:
- hosts: clients and servers
- servers often in data centers

## Access networks, physical media:
- wired, wireless communication links

## Network core:
- interconnected routers
- network of networks



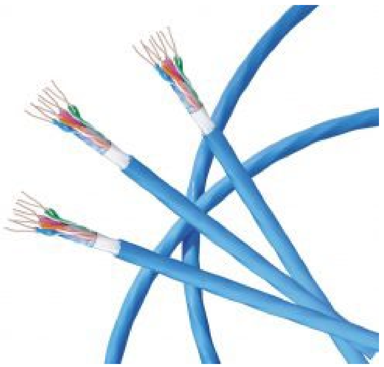
# Access networks: home networks



# Links: physical media

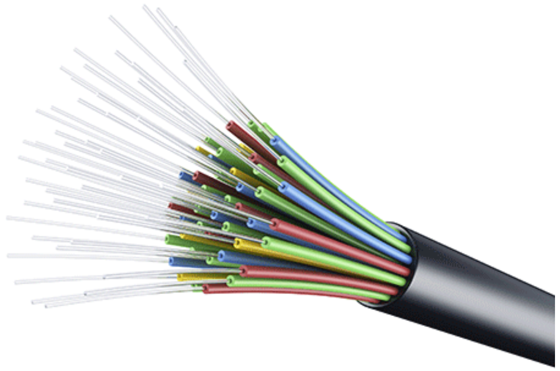
## Twisted pair (TP)

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



## 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



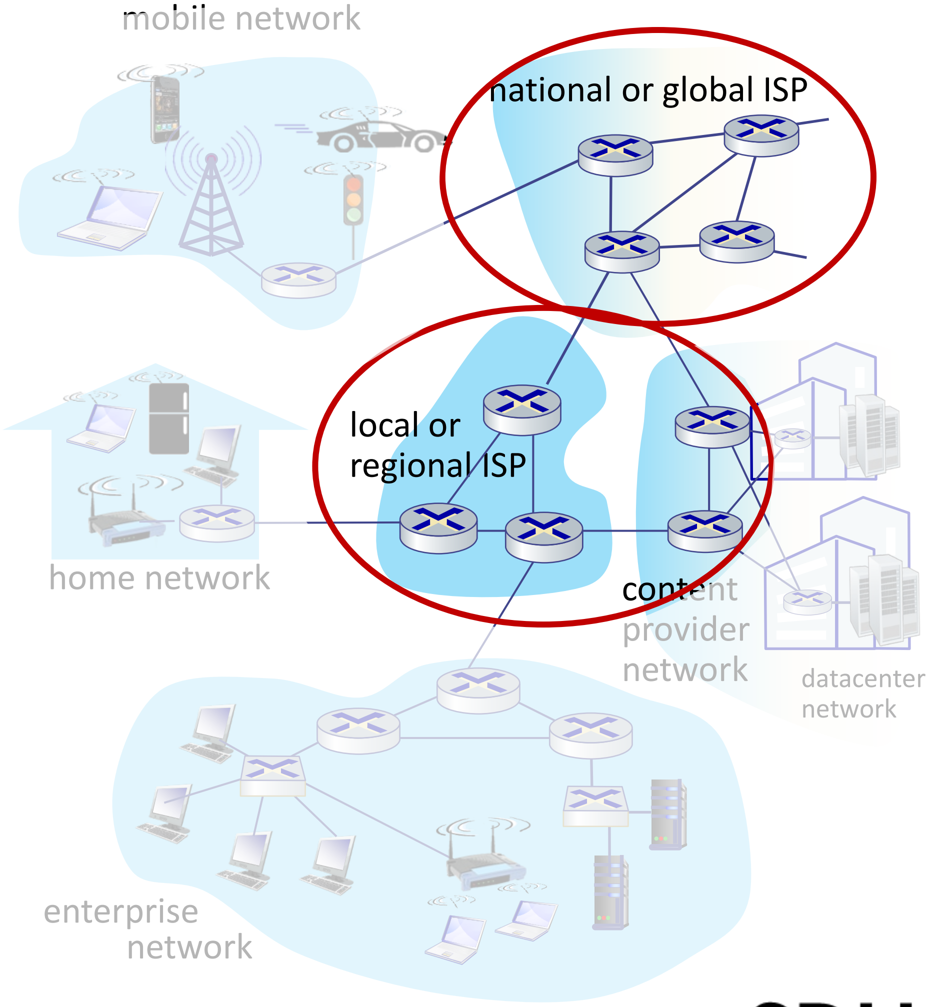
# Subjects of today:

- The Internet
- The Network edge
- **The Network core**
- Performance
- The Protocol layers of the Internet
- Lab exercise


Introduction: 1-17

# The network core

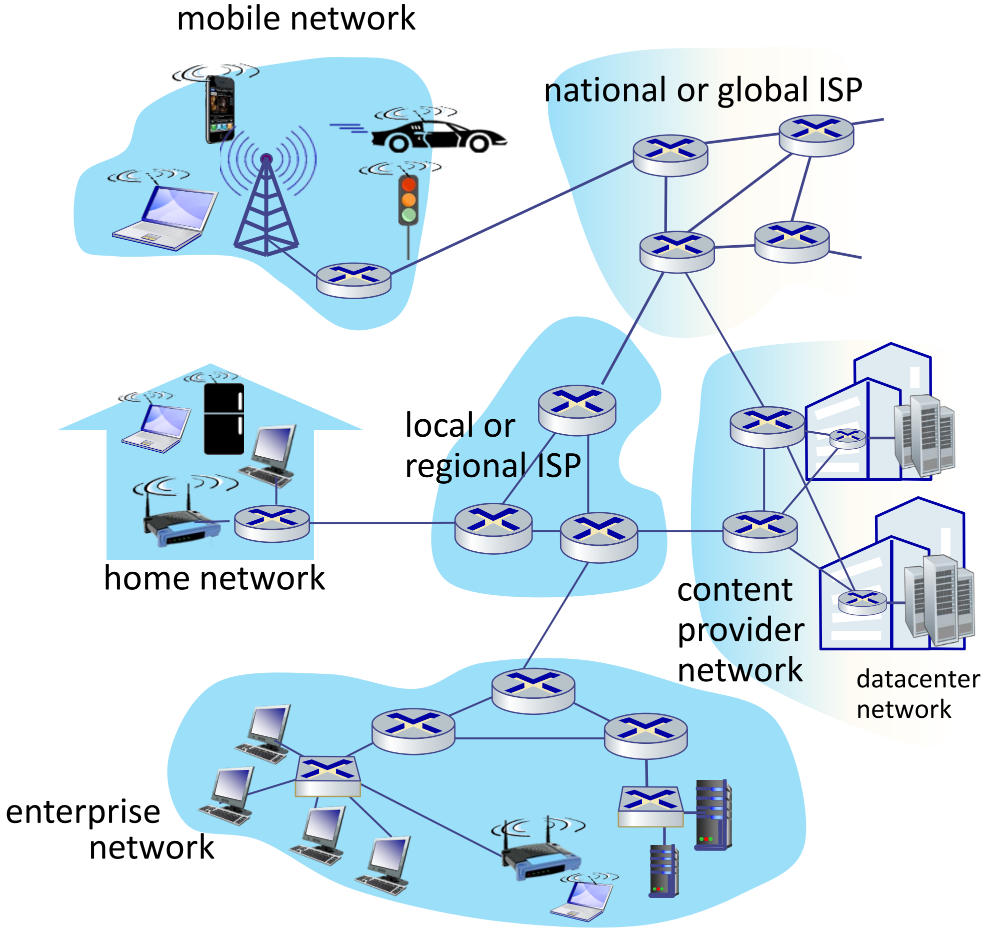
- mesh of interconnected routers
- packet-switching: hosts break application-layer messages into *packets*
  - network forwards packets from one router to the next, across links on path from source to destination



# Internet structure: a "network of networks"

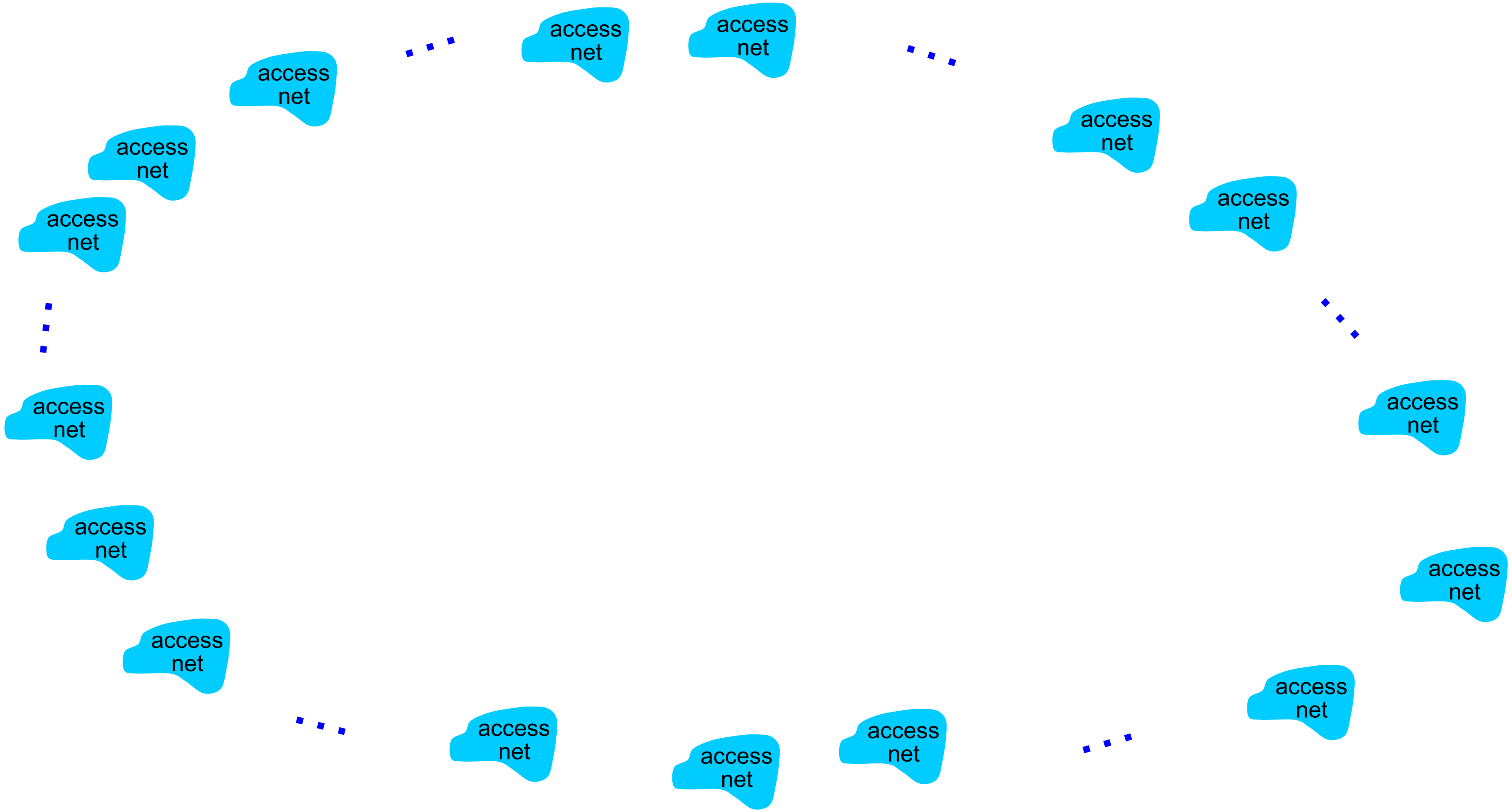
- hosts connect to Internet via access Internet Service Providers (ISPs)

- access ISPs in turn must be interconnected



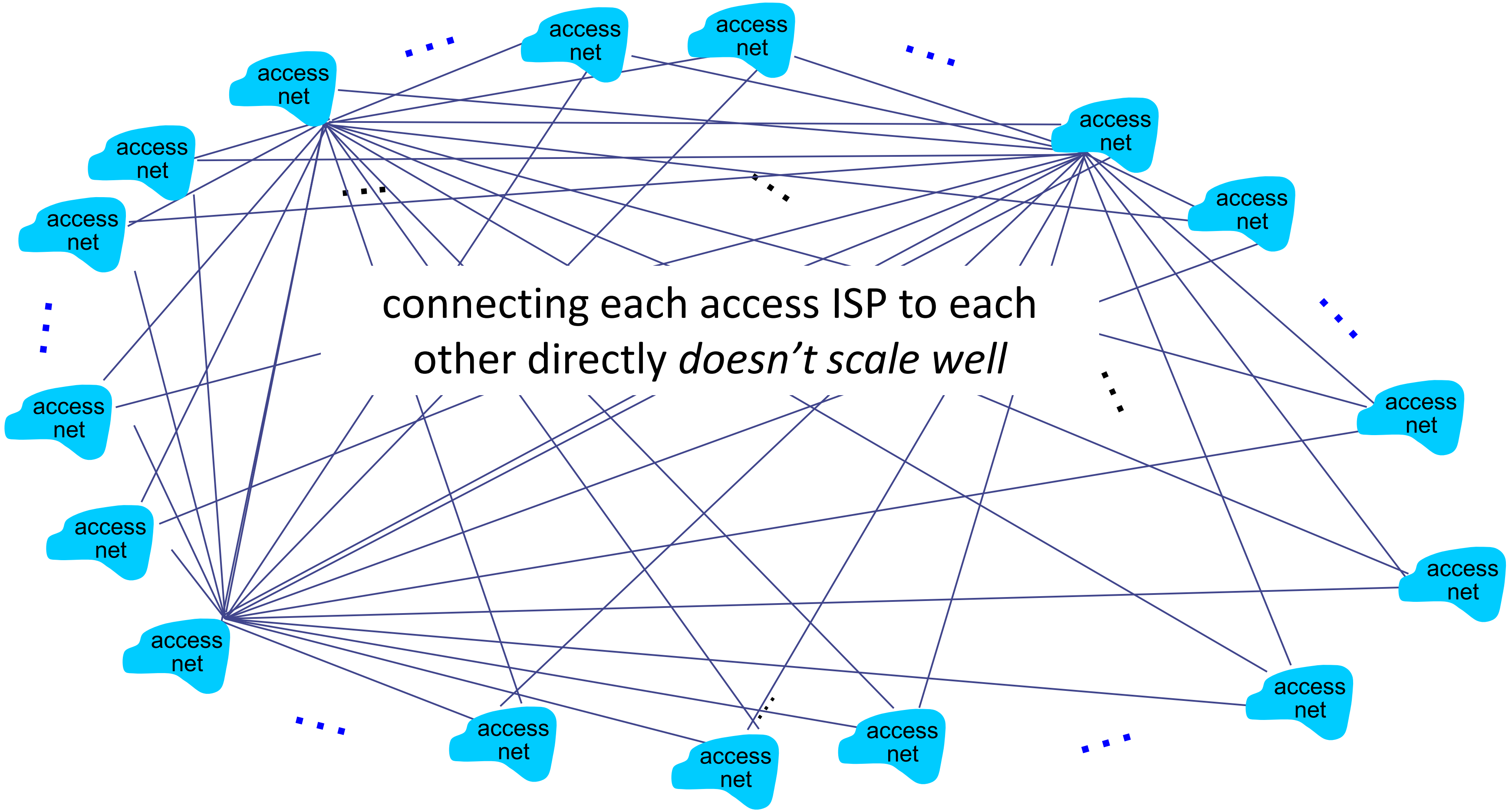
# Internet structure: a "network of networks"

Given *millions* of access ISPs, how to connect them together?



# Internet structure: a "network of networks"

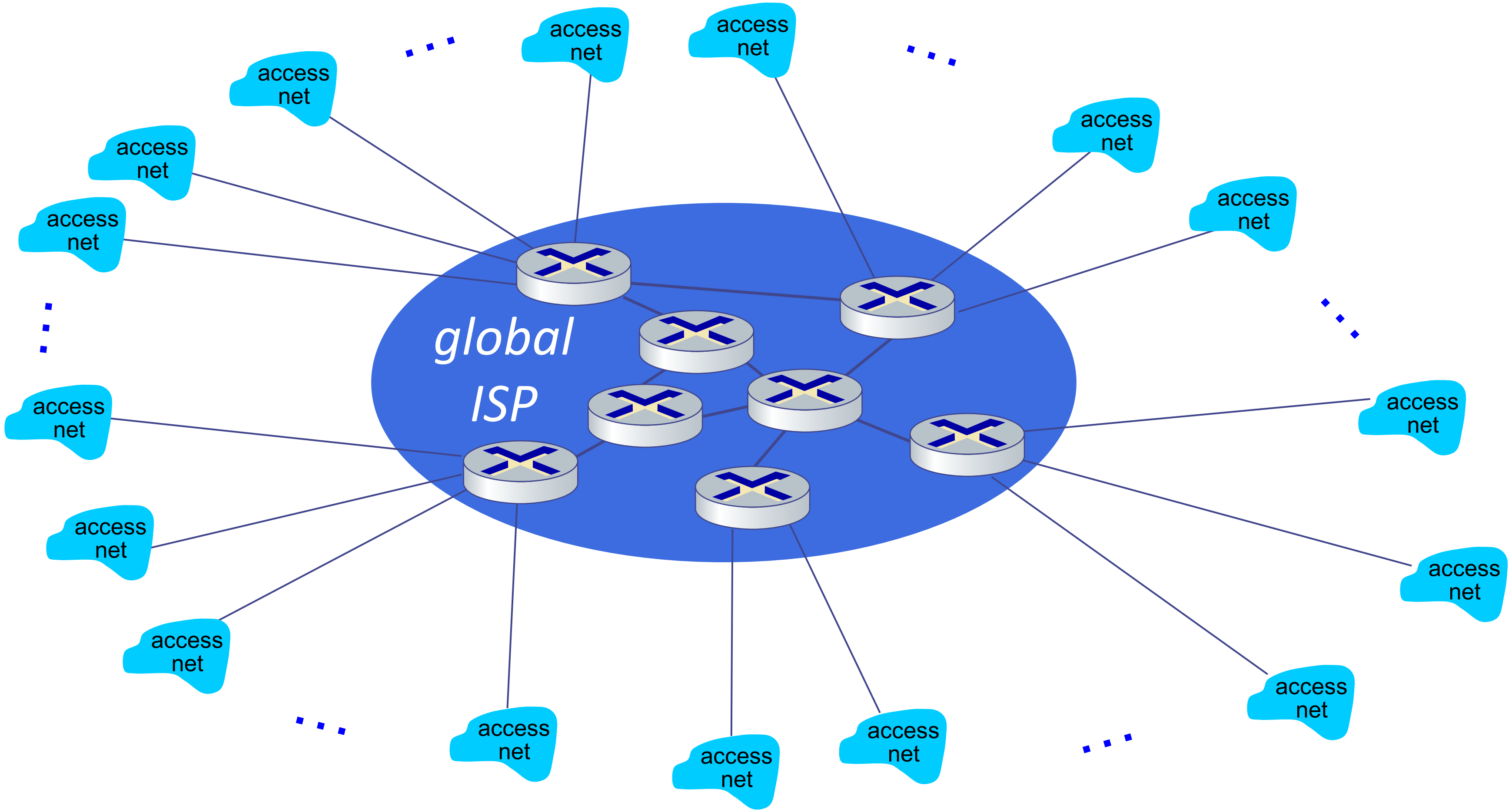
Given *millions* of access ISPs, how to connect them together?



# Internet structure: a "network of networks"

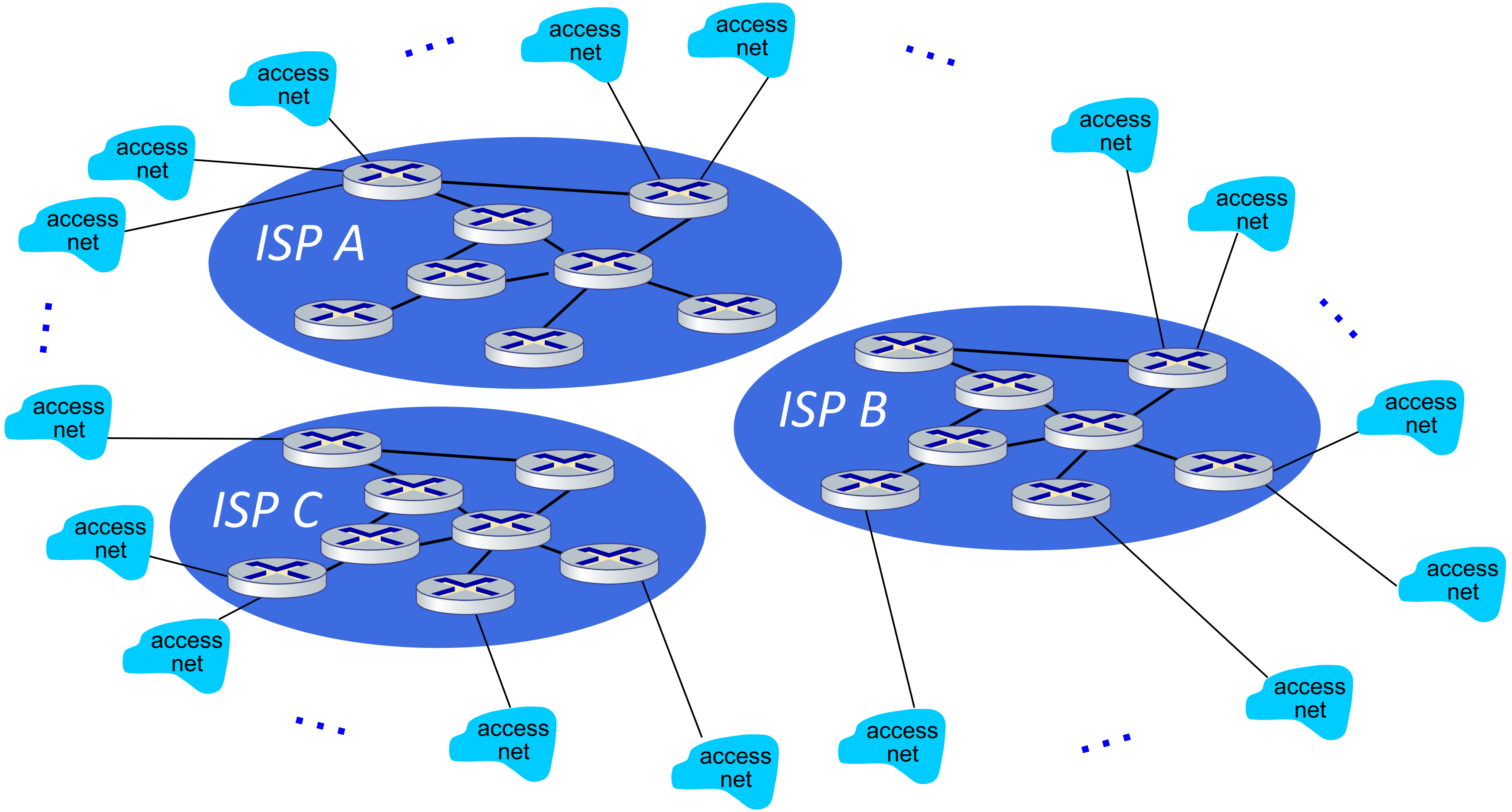
*Option: connect each access ISP to one global transit ISP?*

*Customer and provider ISPs have economic agreement.*



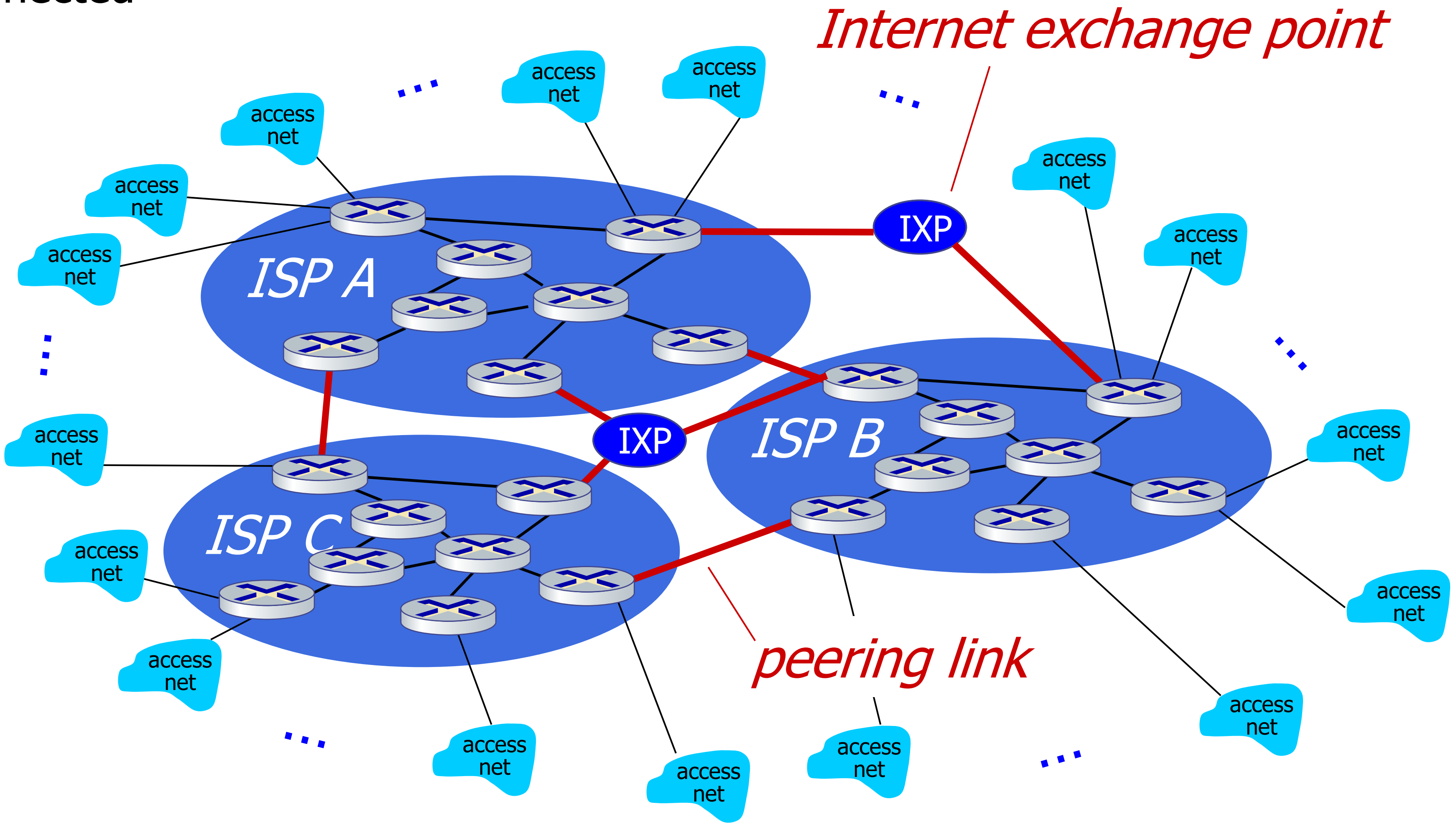
# Internet structure: a "network of networks"

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



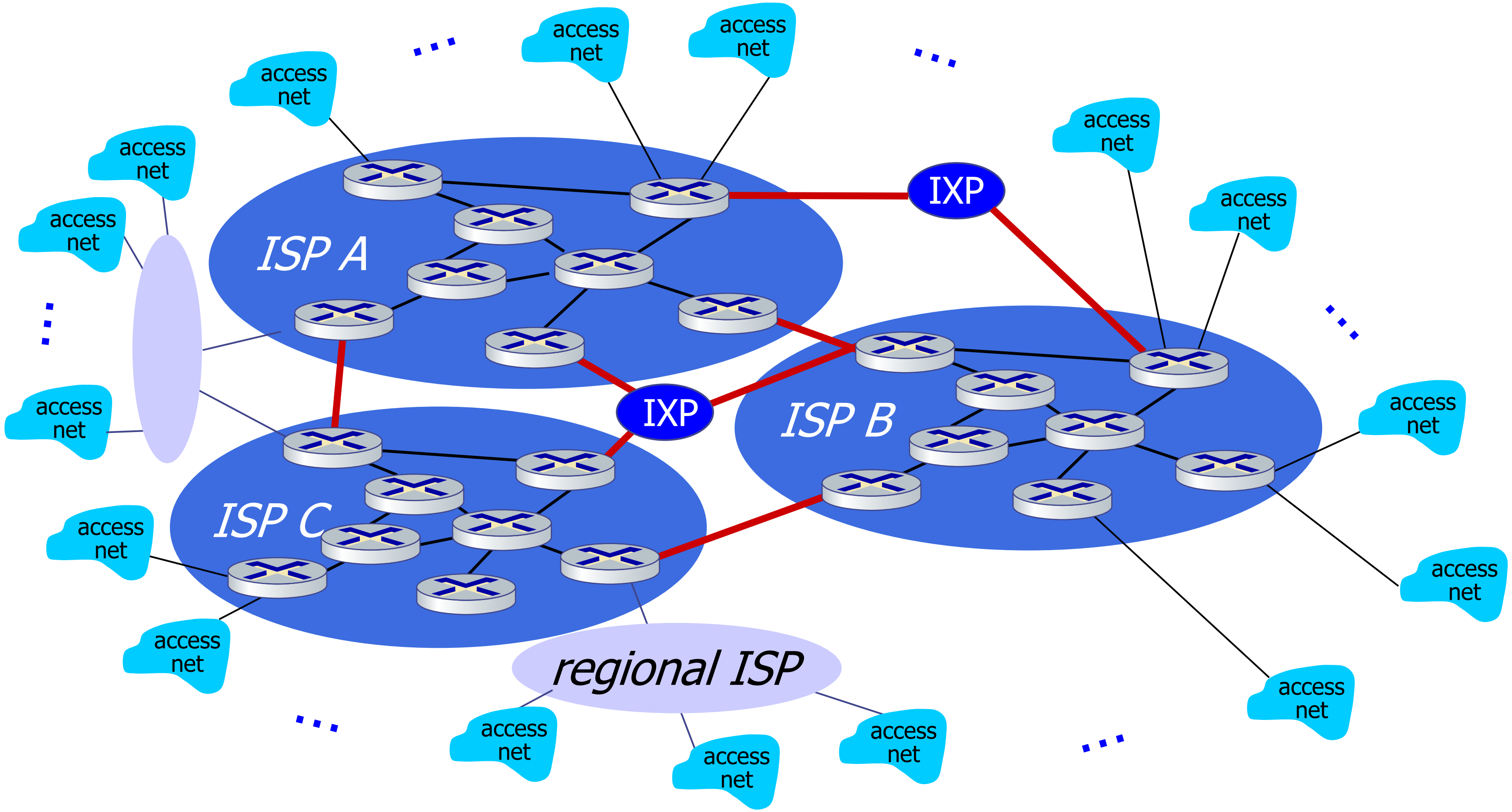
# Internet structure: a “network of networks”

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



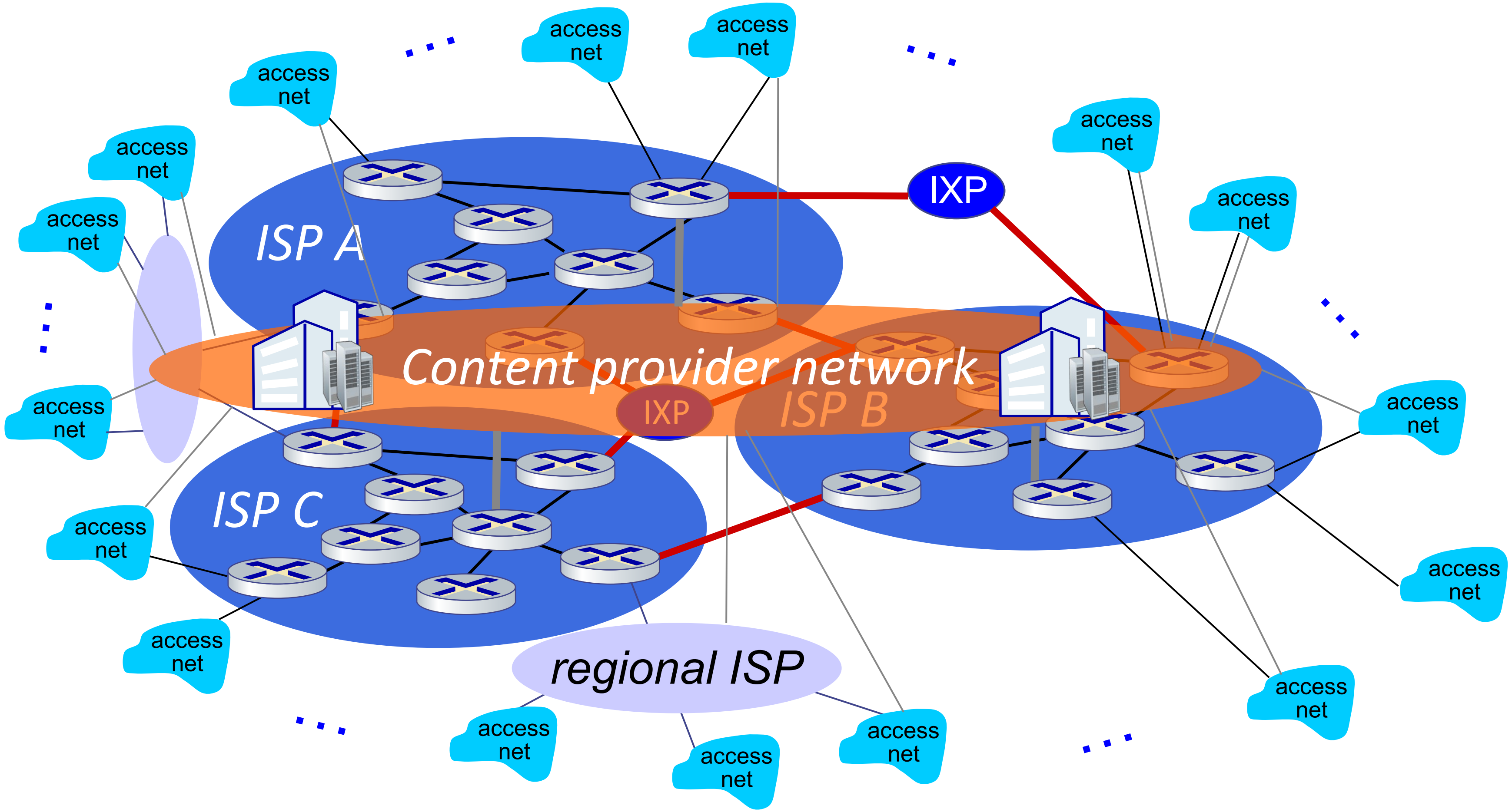
# Internet structure: a "network of networks"

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

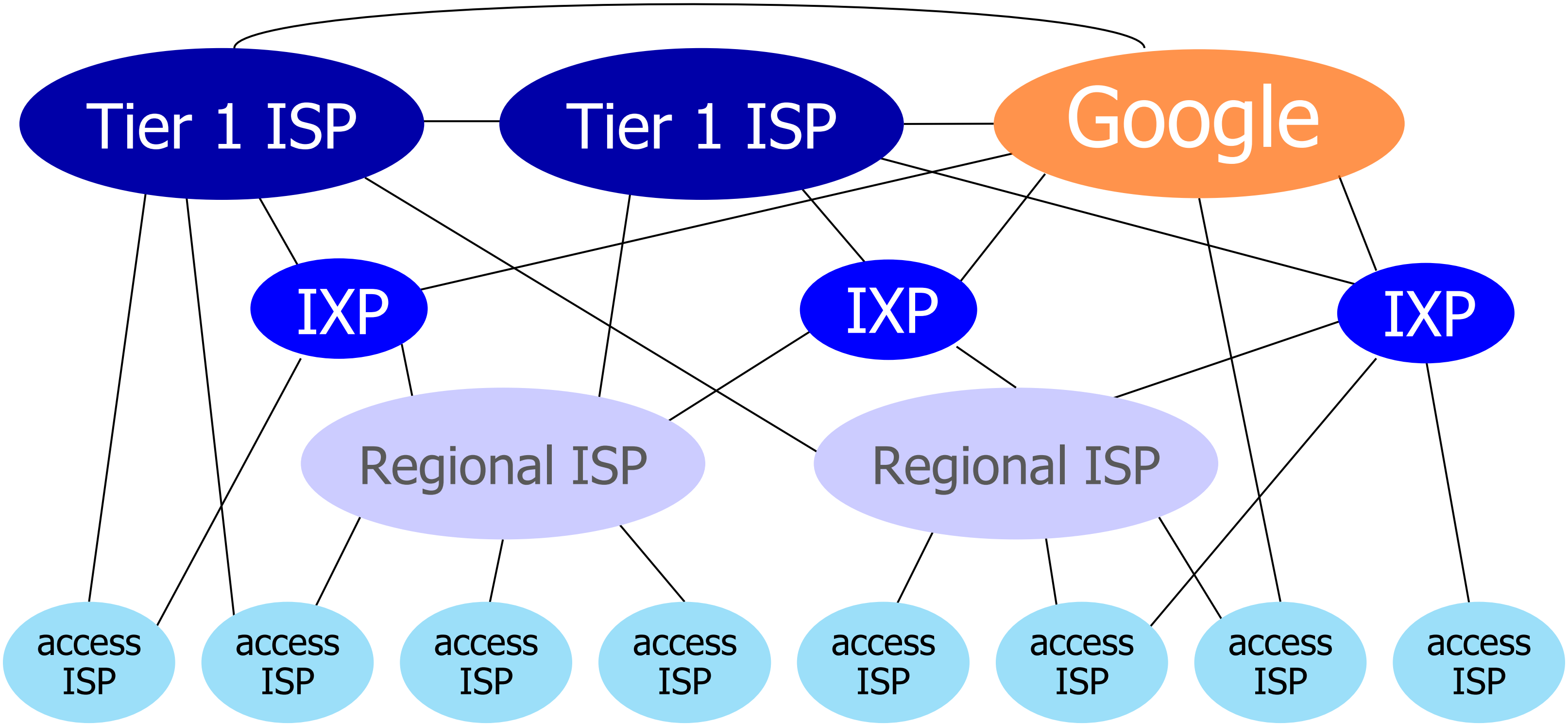


# Internet structure: a “network of networks”

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



# Internet structure: a "network of networks"



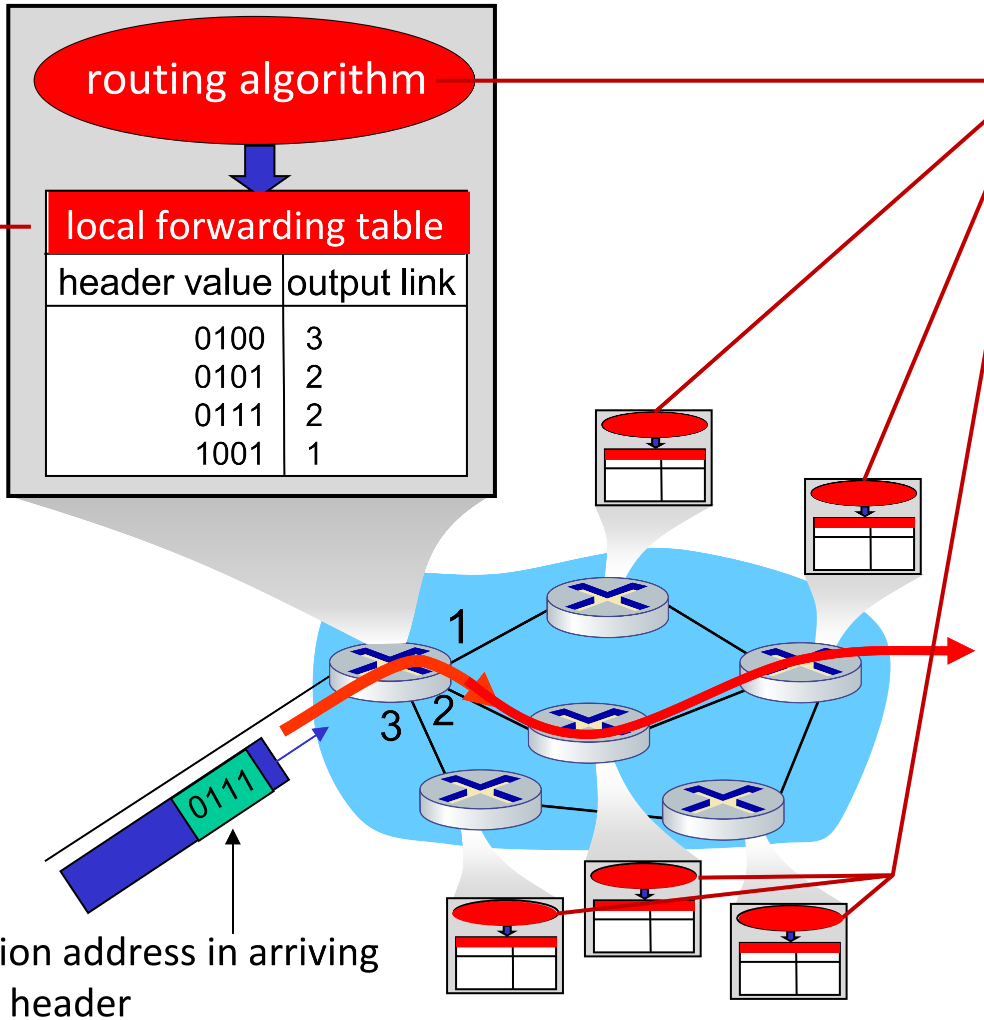

At "center": small # of well-connected large networks

- "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

# Two key network-core functions

*Forwarding:*

- aka “switching”
- *local* action: move arriving packets from router’s input link to appropriate router output link




destination address in arriving packet’s header

*Routing:*

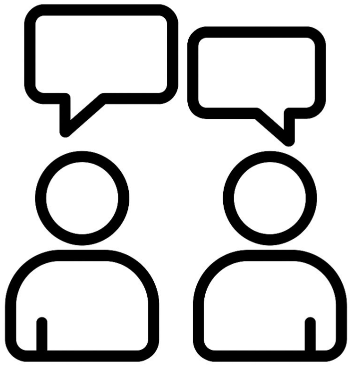
- *global* action: determine source-destination paths taken by packets
- routing algorithms


routing
45 h
3,053 miles
48 h
3,309 miles
San Jose
Northampton
United States
Victoria
Seattle
WASHINGTON
MONTANA
NORTH DAKOTA
MINNESOTA
Quebec City
Portland
Minneapolis
WISCONSIN
Montreal
Ottawa
MAINE
SOUTH DAKOTA
VERMONT
OREGON
IDAHO
MICHIGAN
Toronto
NEW HAMPSHIRE
WYOMING
Detroit
Chicago
IOWA
CT
RI
PENNSYLVANIA
New York
NEVADA
ILLINOIS
INDIANA
OHIO
Philadelphia
NJ
Sacramento
UTAH
COLORADO
Kansas City
Indianapolis
MARYLAND
DE
San Francisco
MISSOURI
WEST VIRGINIA
KENTUCKY
VIRGINIA
CALIFORNIA
Las Vegas
Nashville
OKLAHOMA
TENNESSEE
NORTH CAROLINA
Charlotte
ARKANSAS
Los Angeles
ARIZONA
NEW MEXICO
Atlanta
SOUTH CAROLINA
MISSISSIPPI
San Diego
Phoenix
Dallas
ALABAMA
GEORGIA
Tucson
El Paso
TEXAS
BAJA CALIFORNIA
Austin
Houston
LOUISIANA
Jacksonville
SONORA
New Orleans
CHIHUAHUA
San Antonio
Orlando
Tampa
COAHUILA
FLORIDA


forwarding
forwarding
45 h
3,053 miles
48 h
3,309 miles
United States
San Jose
Northampton
Sacramento
Cleveland

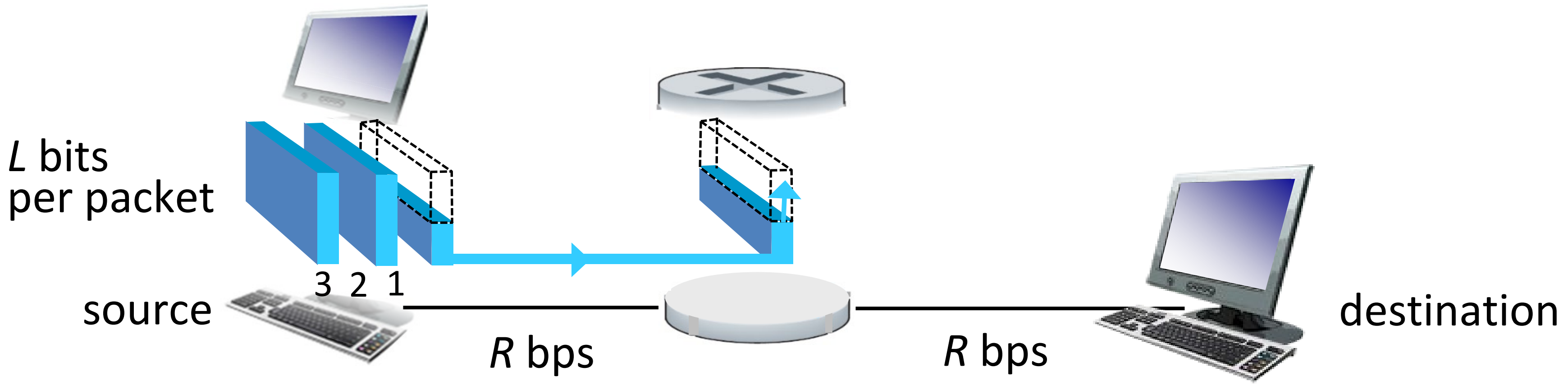
# IP-adresses

- 32 bits → 4 billion possible addresses
- Addressing:
    - Binary: xxxxxxxx xxxxxxxx xxxxxxxx xxxxxxxx
        - E.g.: 11000000 10101000 00000001 00001100
    - Decimal: ddd.ddd.ddd.ddd
        - E.g.:192.168.1.12
- Interpreted left to right



Do you see any problems with this?

# Packet-switching: store-and-forward

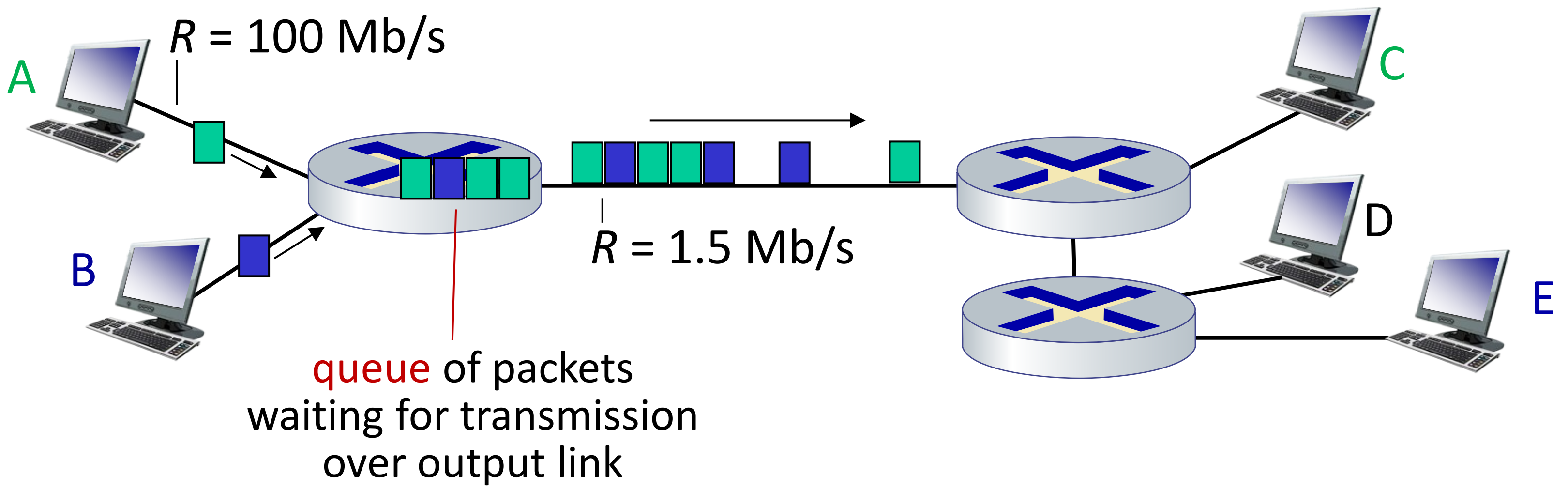



- packet transmission delay: takes *L*/*R* seconds to transmit (push out) *L*-bit packet into link at *R* bps
- *store and forward: entire* packet must arrive at router before it can be transmitted on next link

*One-hop numerical example:*

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

# Packet-switching: queueing




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

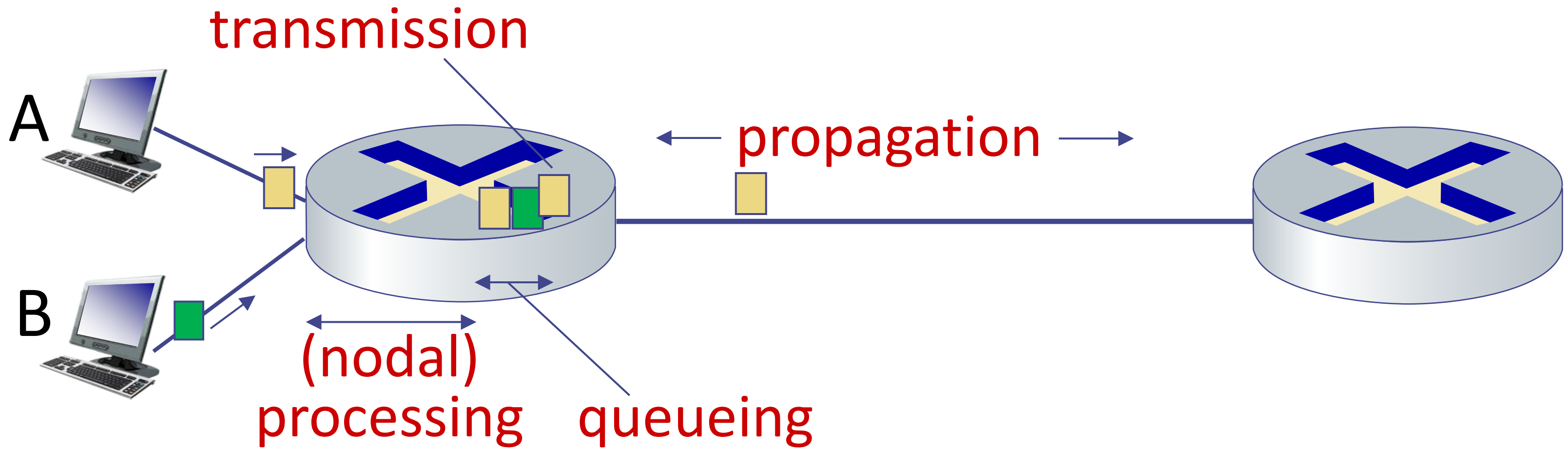
- packets will queue, waiting to be transmitted on output link
- packets can be dropped (lost) if memory (buffer) in router fills up

# Subjects of today:

- The Internet
- The Network edge
- The Network core
- **Performance**
- The Protocol layers of the Internet
- Lab exercise


Introduction: 1-34

# Packet delay: four sources




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

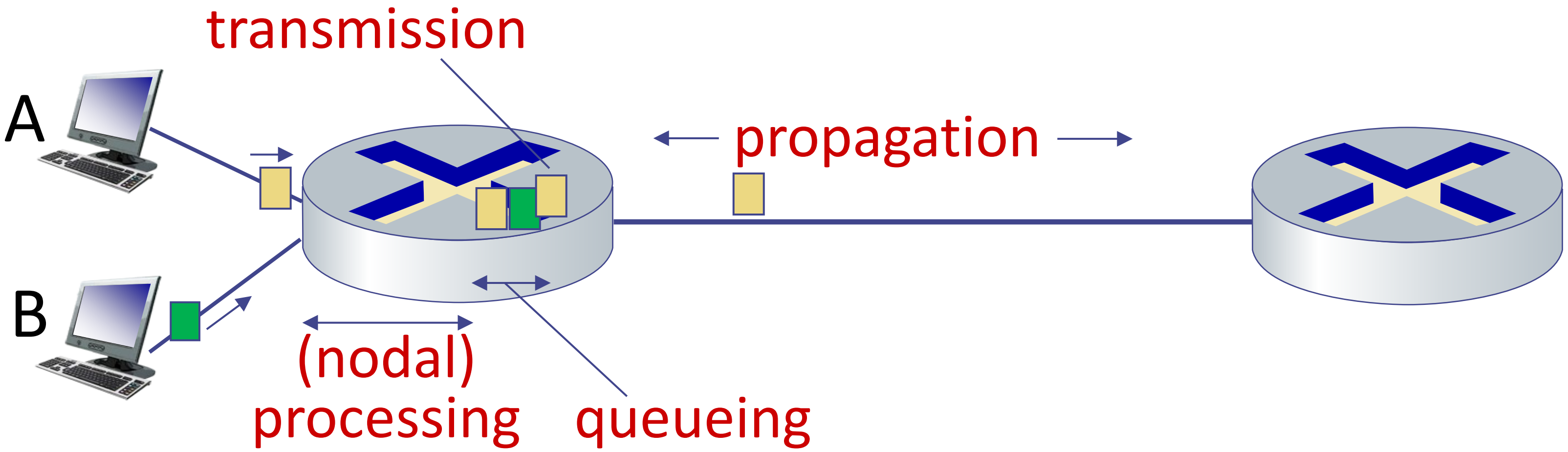
$d_{\text{proc}}$: processing delay

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

$d_{\text{queue}}$: queueing delay

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

# Packet delay: four sources



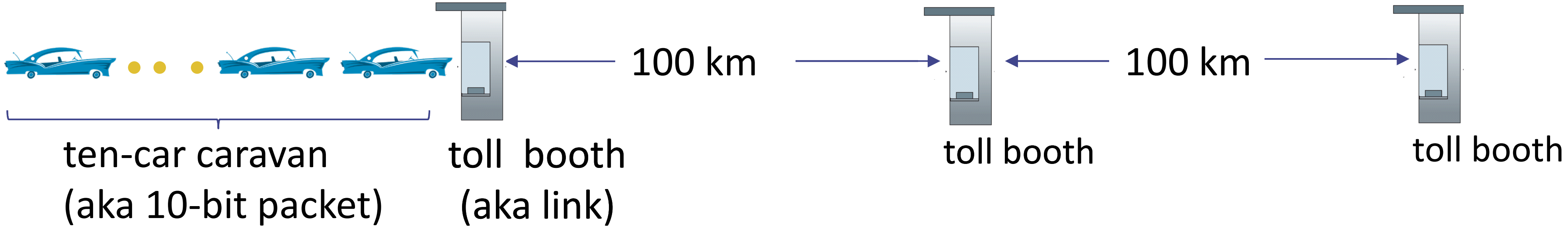

$$d_{nodal} = d_{proc} + d_{queue} + d_{trans} + d_{prop}$$

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

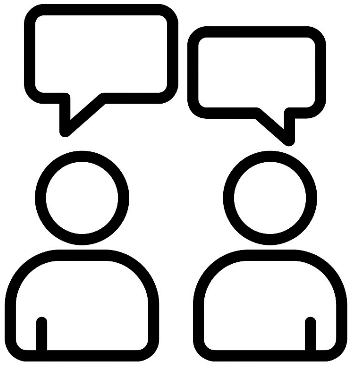
$d_{prop}$: propagation delay:
- $d$: length of physical link
- $s$: propagation speed (~2x10$^8$ m/sec)
- $d_{prop} = d/s$

$d_{trans}$ != $d_{prop}$

# Caravan analogy

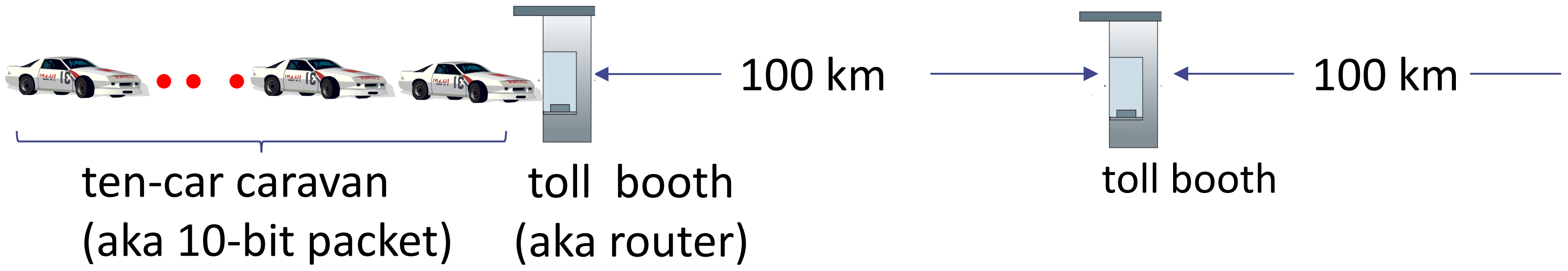



- car ~ bit; caravan ~ packet; toll service ~ link transmission
- toll booth takes 12 sec to service a car (bit transmission time)
- Cars “propagate” at 100 km/hr
- *Q:* How long until caravan is lined up before 2nd toll booth?



What could be analogous to processing and queuing delay?

# Caravan analogy




- 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?*

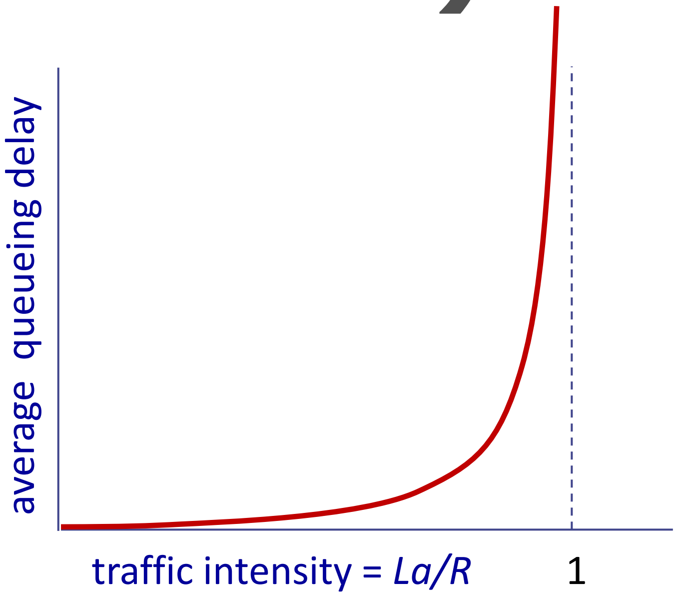
  *A: Yes!* after 7 min, first car arrives at second booth; three cars still at first booth

# Packet queueing delay (revisited)

- *a:* average packet arrival rate [amount]
- *L:* packet length [bits]
- *R:* link bandwidth (bit transmission rate)

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

*"traffic intensity"*



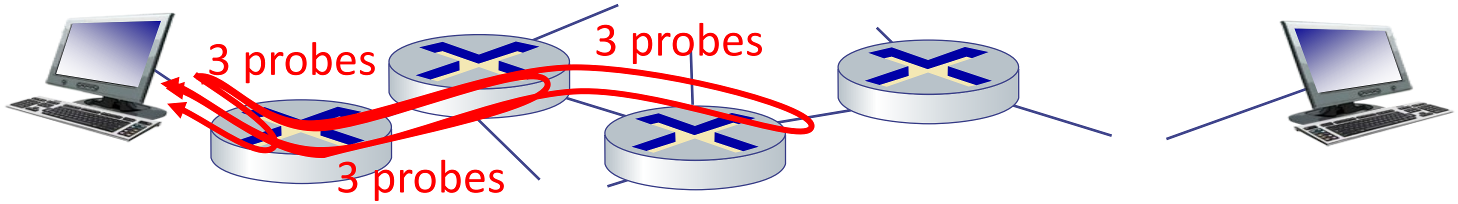

- *La/R* ~ 0: avg. queueing delay small
- *La/R* -> 1: avg. queueing delay large
- *La/R* > 1: more "work" arriving is more than can be serviced - average delay infinite!

*La/R* ~ 0

*La/R -> 1*

# "Real" Internet delays and routes

- what do "real" Internet delay & loss look like?
- **traceroute** program: provides delay measurement from source to router along end-end Internet path towards destination.
  For all *i:*
  - sends three packets that will reach router *i* on path towards destination (with time-to-live field value of *i*)
  - router *i* will return packets to sender
  - sender measures time interval between transmission and reply



# Try it out

- Open your command-line interpreter application
  - Windows: Command Prompt or Powershell
    - Run: *tracert gaia.cs.umass.edu*
  - MacOS: Terminal
    - Run: *traceroute gaia.cs.umass.edu*
  - Linux… you must know
    - sudo apt install tracert
    - Run: *tracert gaia.cs.umass.edu*

- *You can also try to visualize it:*
  - *https://geotraceroute.com*

# Real Internet delays and routes

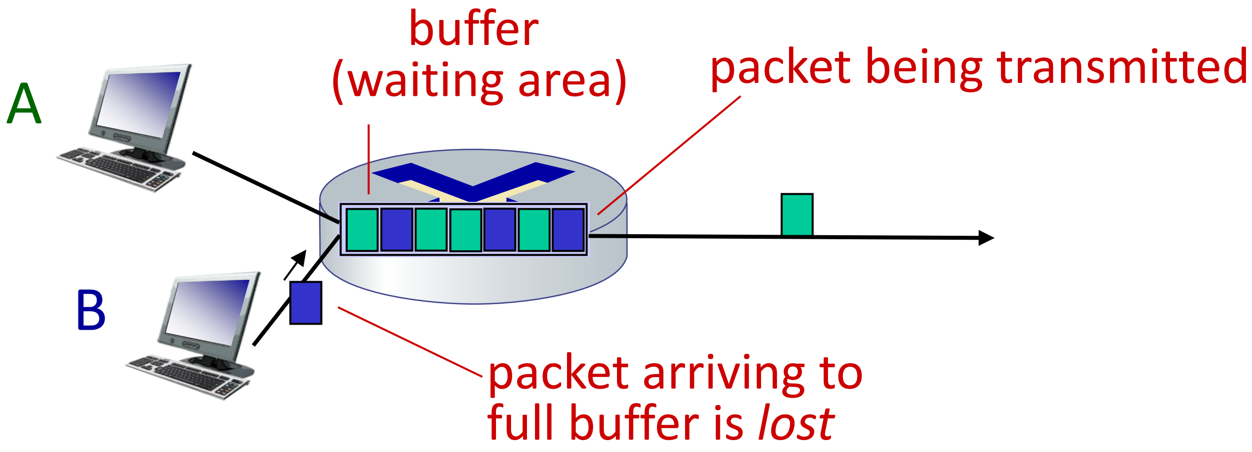
```
Trace complete.
PS C:\Users\simat> tracert gaia.cs.umass.edu

Tracing route to gaia.cs.umass.edu [128.119.245.12]
over a maximum of 30 hops:

  1     3 ms     3 ms     4 ms  10.94.64.1
  2     3 ms     3 ms     3 ms  130.226.84.233
  3     *        *        *     Request timed out.
  4     6 ms     3 ms    15 ms  netrumsydedge.sdu.dk [130.225.244.230]
  5     5 ms     6 ms     5 ms  dk-ore3.nordu.net [109.105.102.160]
  6     6 ms     7 ms     7 ms  dk-bal3.nordu.net [109.105.97.249]
  7    12 ms    11 ms    11 ms  dk-esbj.nordu.net [109.105.97.3]
  8    17 ms    17 ms    17 ms  nl-ams.nordu.net [109.105.97.75]
  9    96 ms    96 ms    98 ms  us-man2.nordu.net [109.105.97.64]
 10   105 ms   107 ms    99 ms  xe-2-3-0.118.rtr.newy32aoa.net.internet2.edu [109.105.98.10]
 11   103 ms   103 ms   105 ms  fourhundredge-0-0-0-1.4079.core1.hart2.net.internet2.edu [163.253.1.228]
 12   103 ms   106 ms   102 ms  fourhundredge-0-0-0-2.4079.core1.bost2.net.internet2.edu [163.253.2.168]
 13   108 ms   108 ms   102 ms  69.16.3.250
 14   105 ms   104 ms   104 ms  69.16.3.0
 15   105 ms   105 ms   106 ms  core2-rt-et-4-3-0.gw.umass.edu [192.80.83.105]
 16   105 ms   106 ms   106 ms  n1-rt-1-1-et-10-0-0.gw.umass.edu [128.119.0.120]
 17   104 ms   105 ms   105 ms  n1-fnt-fw-1-1-1-31-vl1092.gw.umass.edu [128.119.77.233]
 18     *        *        *     Request timed out.
 19   104 ms   111 ms   105 ms  core2-rt-et-7-2-1.gw.umass.edu [128.119.0.121]
 20   107 ms   105 ms   112 ms  n5-rt-1-1-xe-2-1-0.gw.umass.edu [128.119.3.33]
 21   105 ms   105 ms   105 ms  cics-rt-xe-0-0-0.gw.umass.edu [128.119.3.32]
 22     *        *        *     Request timed out.
 23   106 ms   104 ms   105 ms  gaia.cs.umass.edu [128.119.245.12]
```

# Packet loss

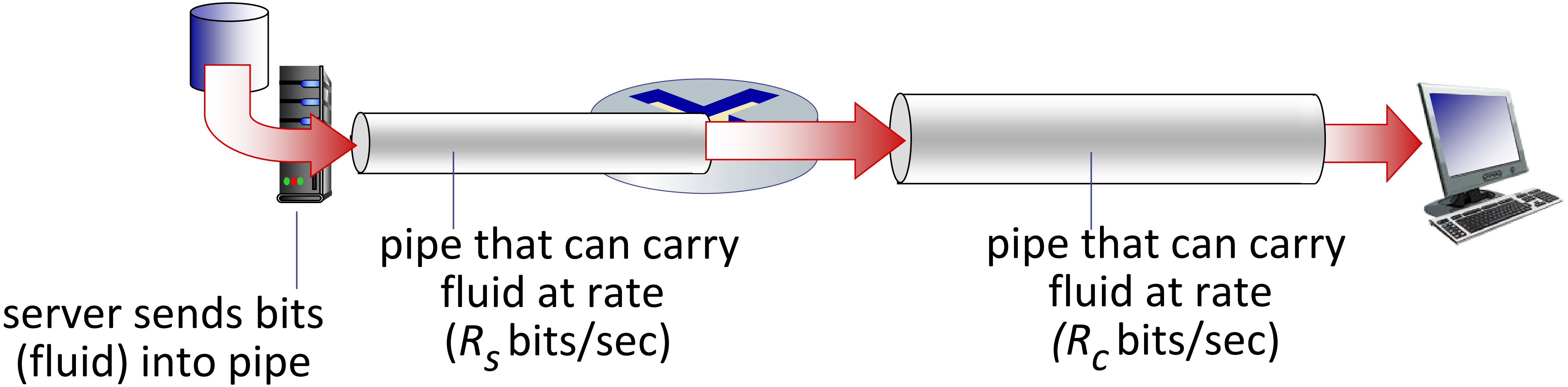
- queue (aka buffer) preceding link in buffer has finite capacity
- packet arriving to full queue dropped (aka lost)
- lost packet may be retransmitted by previous node, by source end system, or not at all




* Check out the Java applet for an interactive animation (on publisher's website) of queuing and loss

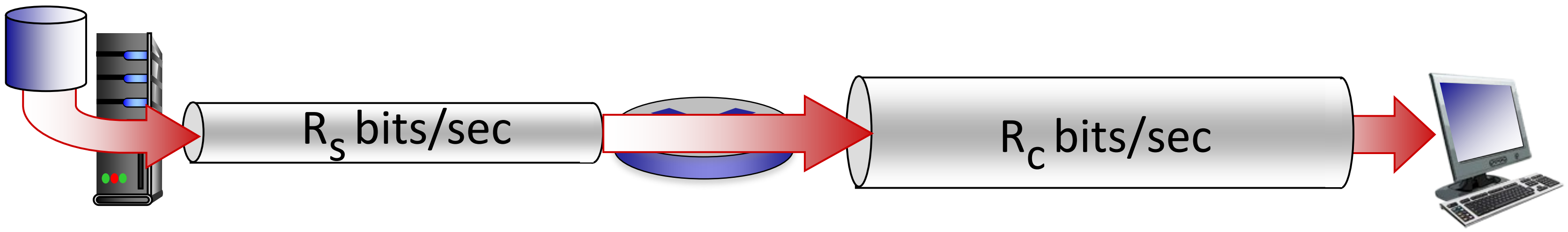
# Throughput

- *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

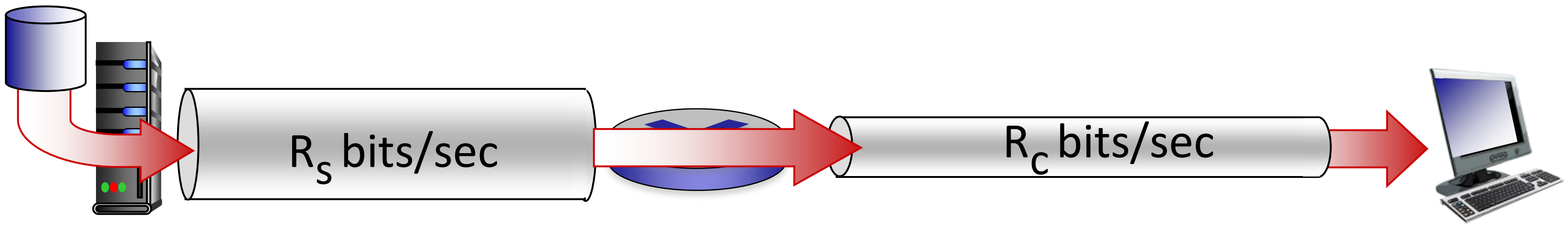


# Throughput

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



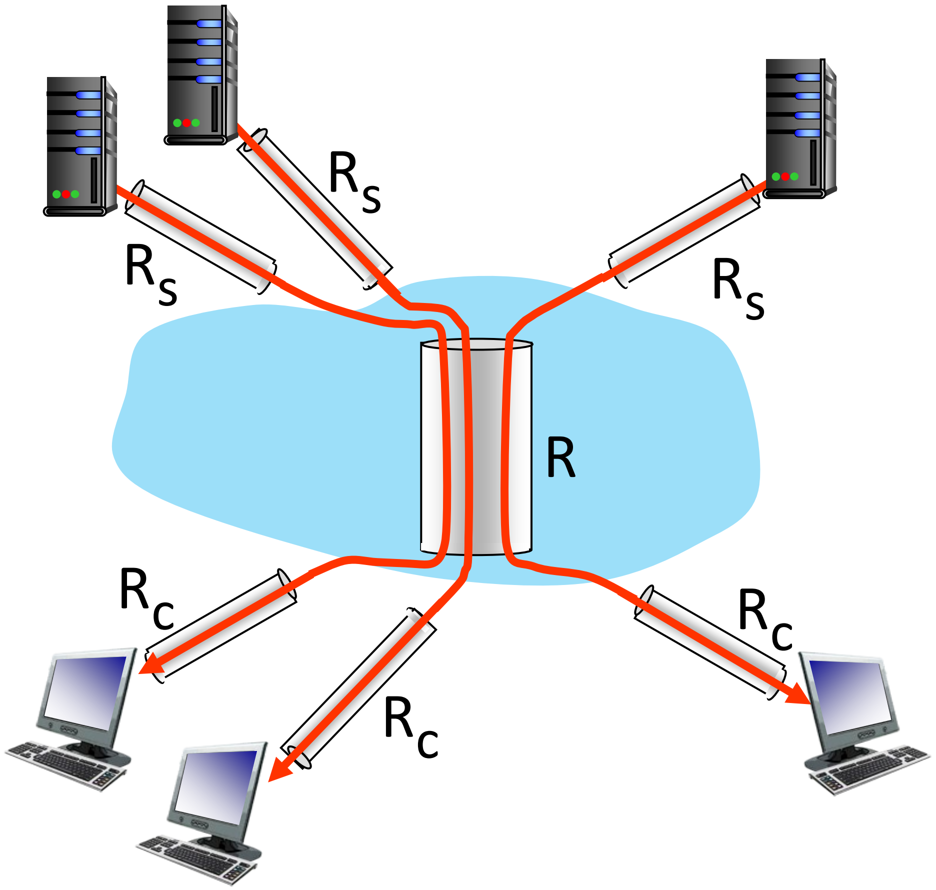

$R_s > R_c$ What is average end-end throughput?




*bottleneck link*

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

# Throughput: network scenario




- per-connection end-end throughput: $min(R_c, R_s, R/10)$
- in practice: $R_c$ or $R_s$ is often bottleneck

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

# Subjects of today:

- The Internet
- The Network edge
- The Network core
- Performance
- **The Protocol layers of the Internet**
- Lab exercise


Introduction: 1-49

# Protocol "layers" and reference models

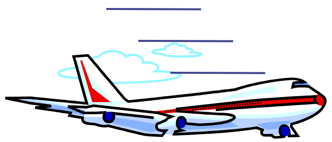
Networks are complex, with many "pieces":

- 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) | ticket (complain) |
| baggage (check) | baggage (claim) |
| gates (load) | gates (unload) |
| runway takeoff | runway landing |
| airplane routing | airplane routing |

airplane routing

# Example: organization of air travel

| | | |
|---|---|---|
| ticket (purchase) | *ticketing service* | ticket (complain) |
| baggage (check) | *baggage service* | baggage (claim) |
| gates (load) | *gate service* | gates (unload) |
| runway takeoff | *runway service* | runway landing |
| airplane routing | *routing service* | airplane routing |

***layers:*** **each layer implements a service**

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

# Why layering?

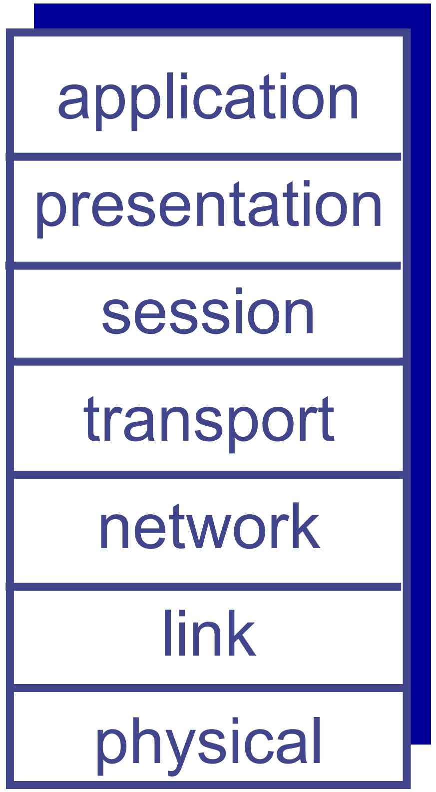
Approach to designing/discussing complex systems:

- explicit structure allows identification, relationship of system's pieces
- 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

# ISO/OSI reference model (revisited)

Two layers not found in the Internet protocol stack!

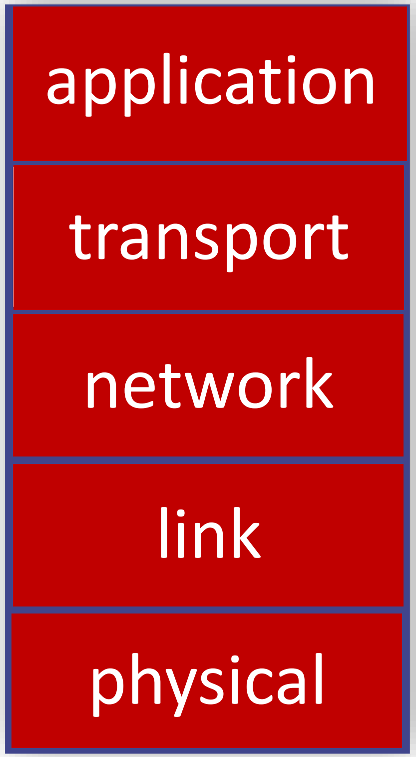
- *presentation:* allow applications to interpret meaning of data, e.g., encryption, compression, machine-specific conventions
- *session:* synchronization, checkpointing, recovery of data exchange
- Internet stack "missing" these layers!
  - these services, *if needed,* must be implemented in application
  - needed?



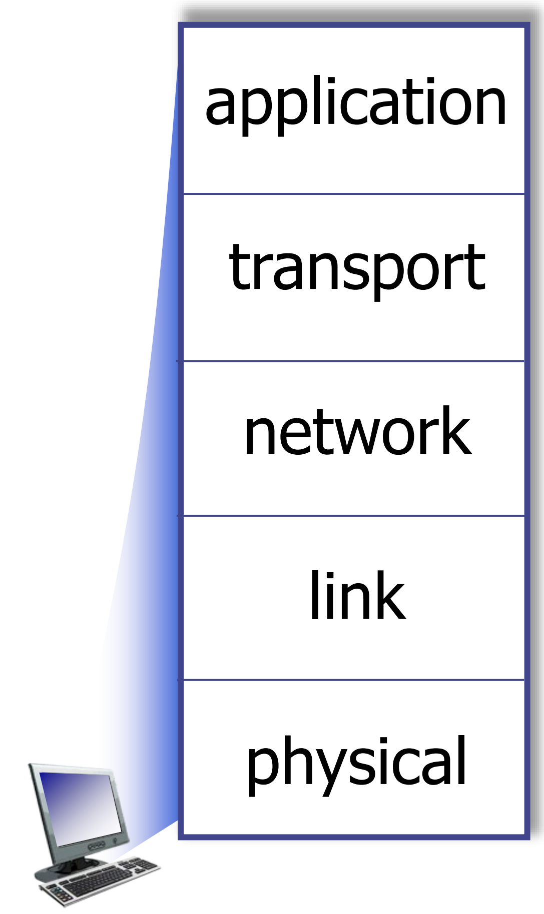

The seven layer OSI/ISO reference model

# 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



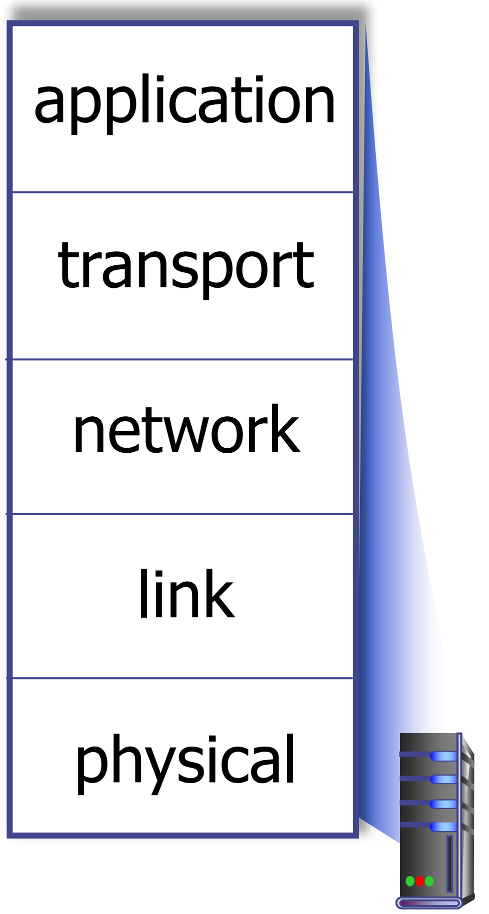

M

Application exchanges messages to implement some application service using *services* of transport layer

$H_t$ M

Transport-layer protocol transfers M (e.g., reliably) from one *process* to another, using services of network layer

- transport-layer protocol encapsulates application-layer message, M, with *transport* layer-layer header $H_t$ to create a transport-layer segment
  - $H_t$ used by transport layer protocol to implement its service



# Services, Layering and Encapsulation

application
transport
network
link
physical

*source*

M

$H_t$ M

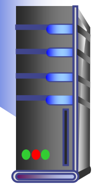
Transport-layer protocol transfers M (e.g., reliably) from one *process* to another, using services of network layer

$H_n$ $H_t$ M

Network-layer protocol transfers transport-layer segment [$H_t$ | M] from one *host* to another, using link layer services

- network-layer protocol **encapsulates** transport-layer segment [$H_t$ | M] with network layer-layer header $H_n$ to create a network-layer datagram
  - $H_n$ used by network layer protocol to implement its service

application
transport
network
link
physical



*destination*

# Services, Layering and Encapsulation

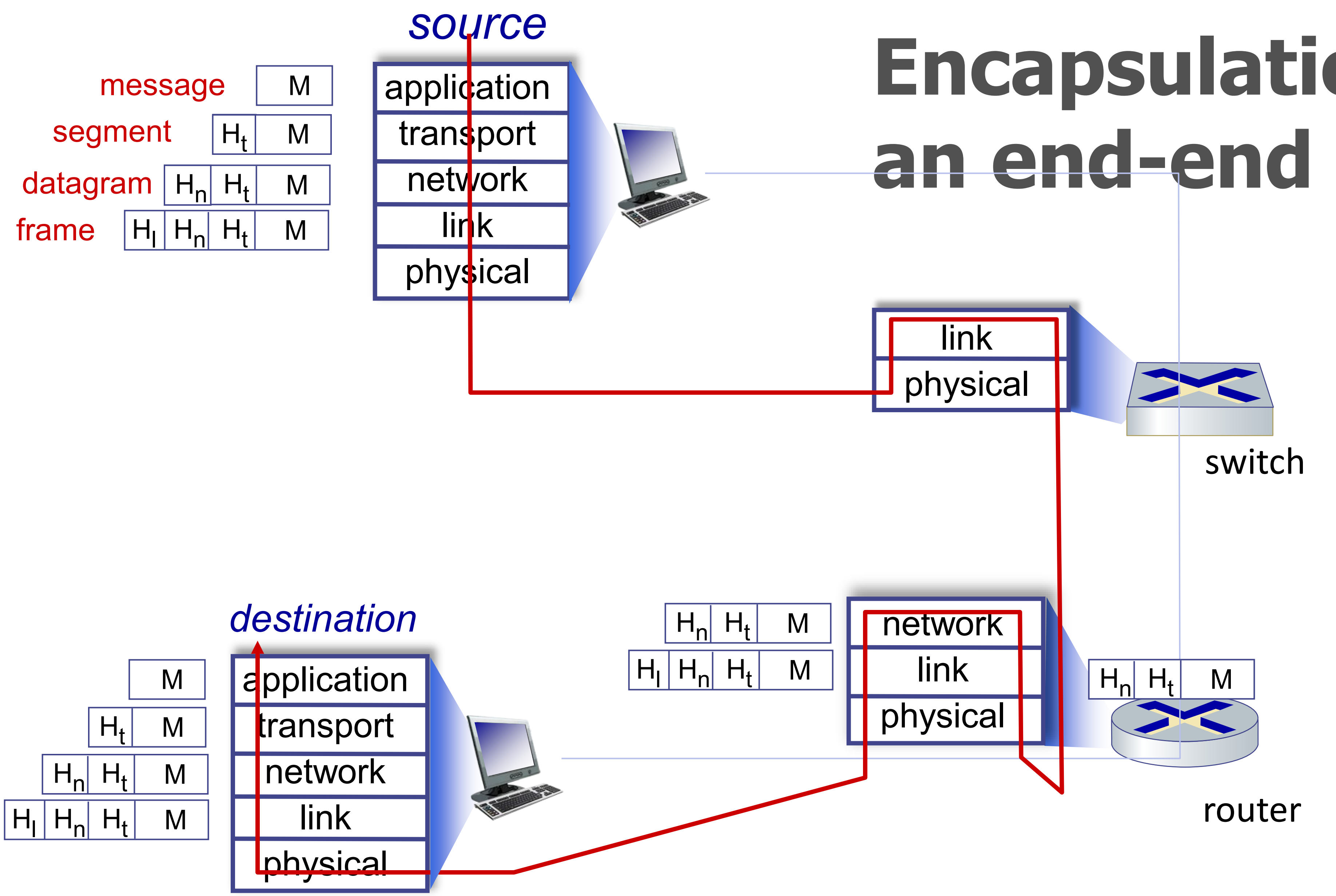
| source | | destination |
|---|---|---|
| application | M | application |
| transport | $H_t$ M | transport |
| network | $H_n$ $H_t$ M | network |
| link | $H_l$ $H_n$ $H_t$ M | link |
| physical | | physical |

Network-layer protocol transfers transport-layer segment [$H_t$ | M] from one *host* to another, using link layer services

Link-layer protocol transfers datagram [$H_n$| [$H_t$ |M] from *host* to neighboring host, using network-layer services

- link-layer protocol **encapsulates** network datagram [$H_n$| [$H_t$ |M], with link-layer header $H_l$ to create a link-layer **frame**

# Encapsulation: an end-end view

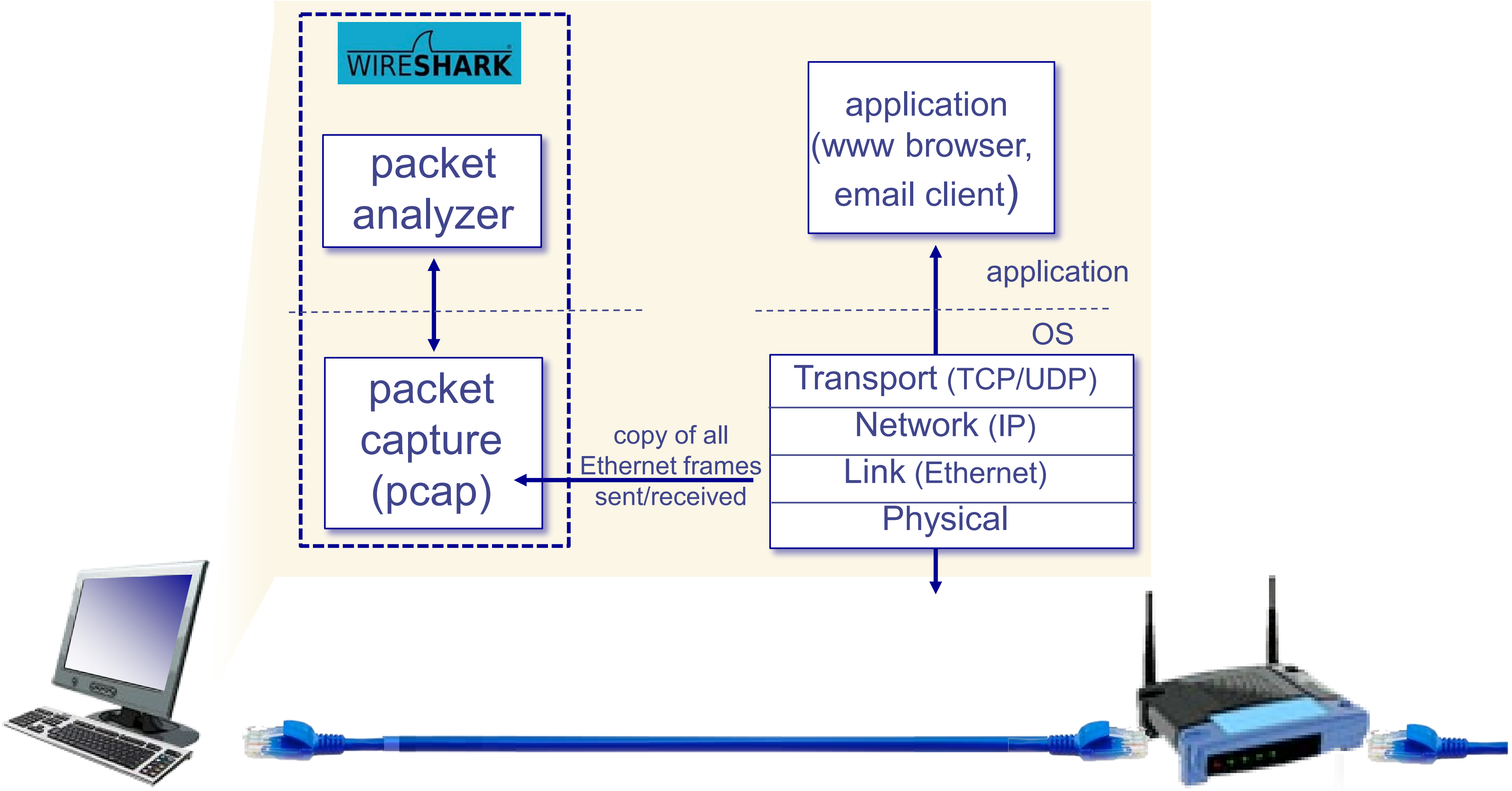


# Subjects of today:

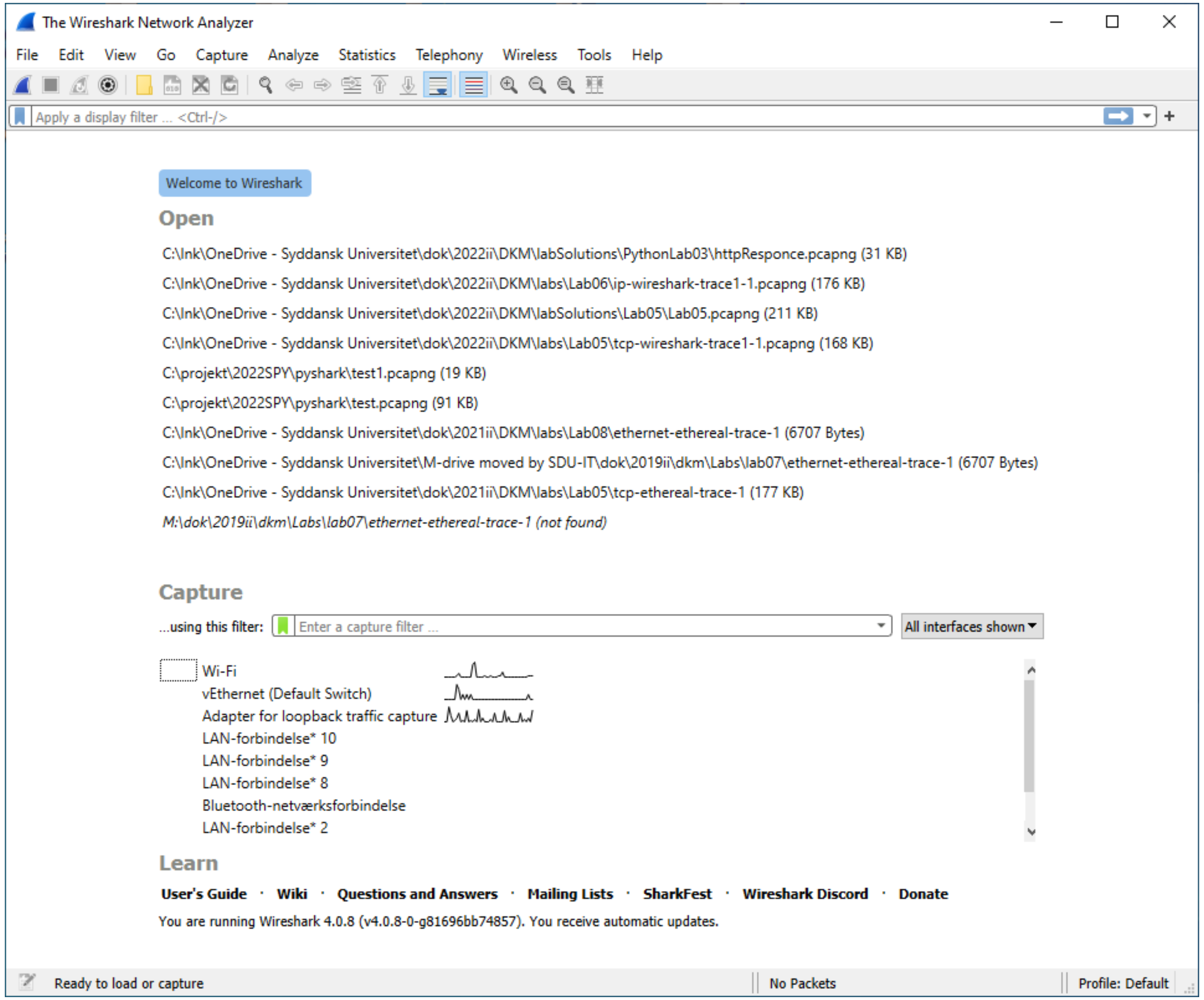
- The Internet
- The Network edge
- The Network core
- Performance
- The Protocol layers of the Internet
- **Lab exercise**


Introduction: 1-60

# 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

# Wireshark

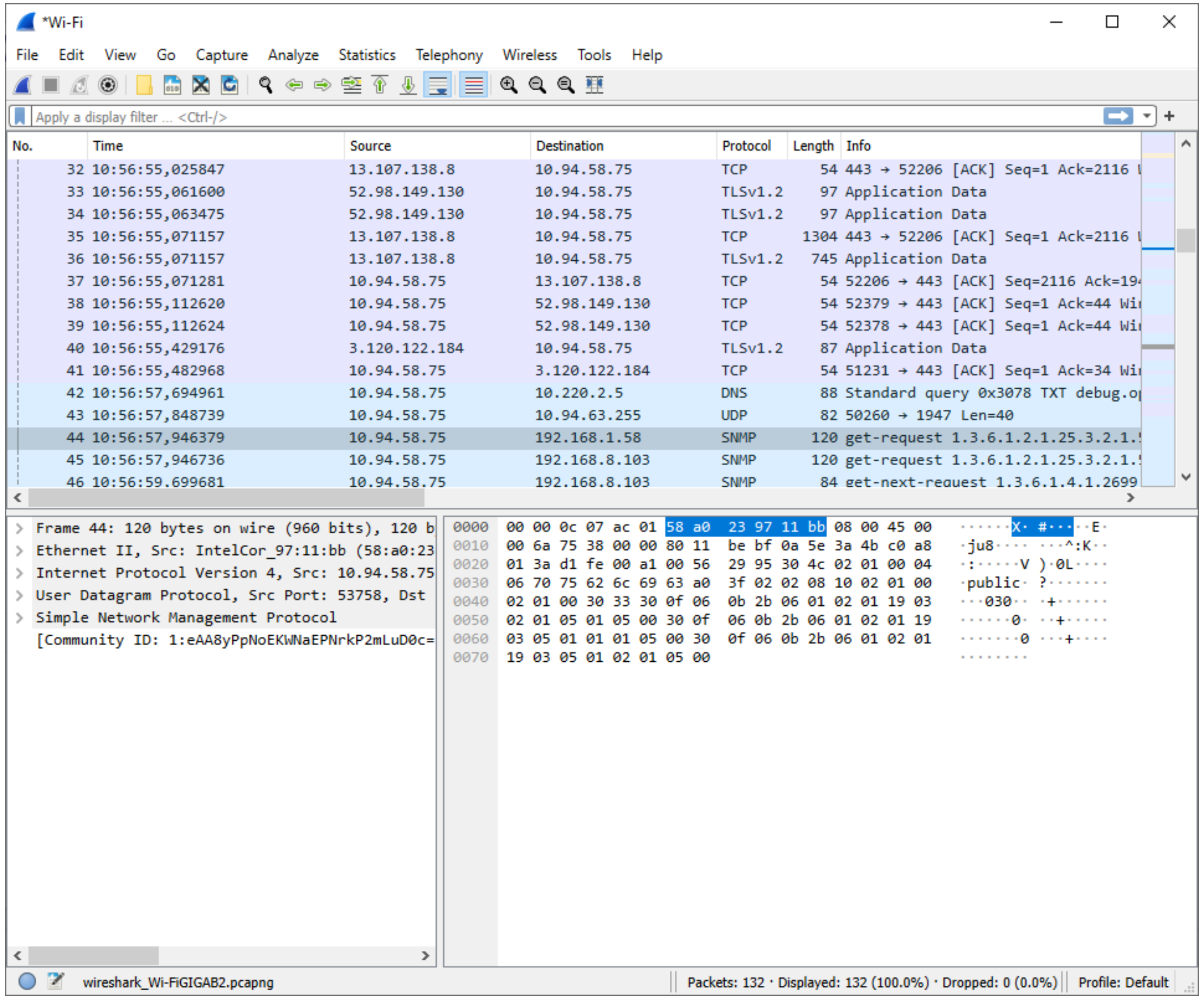

The Wireshark Network Analyzer
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter ... <Ctrl-/>
Welcome to Wireshark
Open
C:\lnk\OneDrive - Syddansk Universitet\dok\2022ii\DKM\labSolutions\PythonLab03\httpResponce.pcapng (31 KB)
C:\lnk\OneDrive - Syddansk Universitet\dok\2022ii\DKM\labs\Lab06\ip-wireshark-trace1-1.pcapng (176 KB)
C:\lnk\OneDrive - Syddansk Universitet\dok\2022ii\DKM\labSolutions\Lab05\Lab05.pcapng (211 KB)
C:\lnk\OneDrive - Syddansk Universitet\dok\2022ii\DKM\labs\Lab05\tcp-wireshark-trace1-1.pcapng (168 KB)
C:\projekt\2022SPY\pyshark\test1.pcapng (19 KB)
C:\projekt\2022SPY\pyshark\test.pcapng (91 KB)
C:\lnk\OneDrive - Syddansk Universitet\dok\2021ii\DKM\labs\Lab08\ethernet-ethereal-trace-1 (6707 Bytes)
C:\lnk\OneDrive - Syddansk Universitet\M-drive moved by SDU-IT\dok\2019ii\dkm\Labs\lab07\ethernet-ethereal-trace-1 (6707 Bytes)
C:\lnk\OneDrive - Syddansk Universitet\dok\2021ii\DKM\labs\Lab05\tcp-ethereal-trace-1 (177 KB)
M:\dok\2019ii\dkm\Labs\lab07\ethernet-ethereal-trace-1 (not found)
Capture
...using this filter:
Enter a capture filter ...
All interfaces shown
Wi-Fi
vEthernet (Default Switch)
Adapter for loopback traffic capture
LAN-forbindelse* 10
LAN-forbindelse* 9
LAN-forbindelse* 8
Bluetooth-netværksforbindelse
LAN-forbindelse* 2
Learn
User's Guide · Wiki · Questions and Answers · Mailing Lists · SharkFest · Wireshark Discord · Donate
You are running Wireshark 4.0.8 (v4.0.8-0-g81696bb74857). You receive automatic updates.
Ready to load or capture
No Packets
Profile: Default

# Wireshark

Menus

Filters

Packet list




Packet details

Packet content

# Wireshark

- **Today: Lab 1 - Wireshark Intro**
- There are no demands for hand-ins of the Labs, but make **journals** documenting your work to the extend you find it relevant. They are useful for discussions at the class and with your classmates.
- The journal should consist of printouts and/or screenshots from Wireshark and other tools used in the lab.
- Answer the questions from the lab with references (e.g. using highlights or arrows) to the printouts/screenshots.