Introduction to Networking

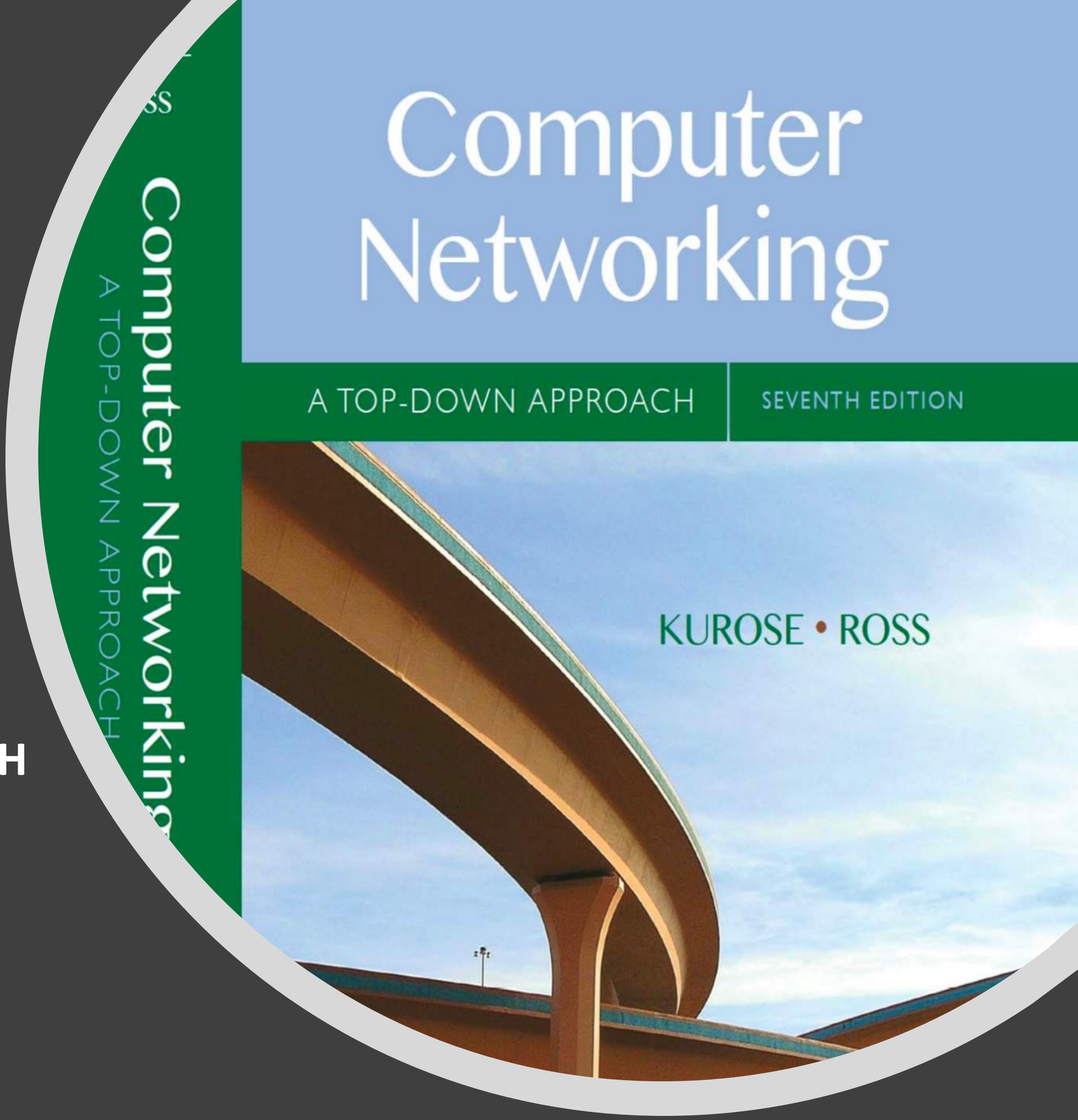
CAN201 – Lecture 1

Module Leader: Dr. Fei Cheng & Dr. Gordon Boateng

Basic Information

- **Teachers:**
  - Dr. Fei Cheng, fei.cheng@xjtlu.edu.cn
  - Dr. Gordon Boateng, gordon.boateng@xjtlu.edu.cn

- **Textbook:**
  - **Computer Networking: A TOP-DOWN APPROACH**
    - Seventh Edition
    - Author: Jim Kurose, Keith Ross



TAs’ emails

Donghui Li	donghui.li25@student.xjtlu.edu.cn
Haotian Yin	haotian.yin23@student.xjtlu.edu.cn
Ziying Yang	ziying.yang24@student.xjtlu.edu.cn
Yang Yu	yang.yu25@student.xjtlu.edu.cn
Jia Liu	jia.liu21@student.xjtlu.edu.cn
Ziyanrna Wang	ziyanran.wang25@student.xjtlu.edu.cn
Yifan Zhan	yifan.zhan22@student.xjtlu.edu.cn
Wei Shao	wei.shao25@student.xjtlu.edu.cn

“Protocol” for CAN201 related email

[协议]

EMAIL TITLE SHOULD START WITH “[CAN201]”

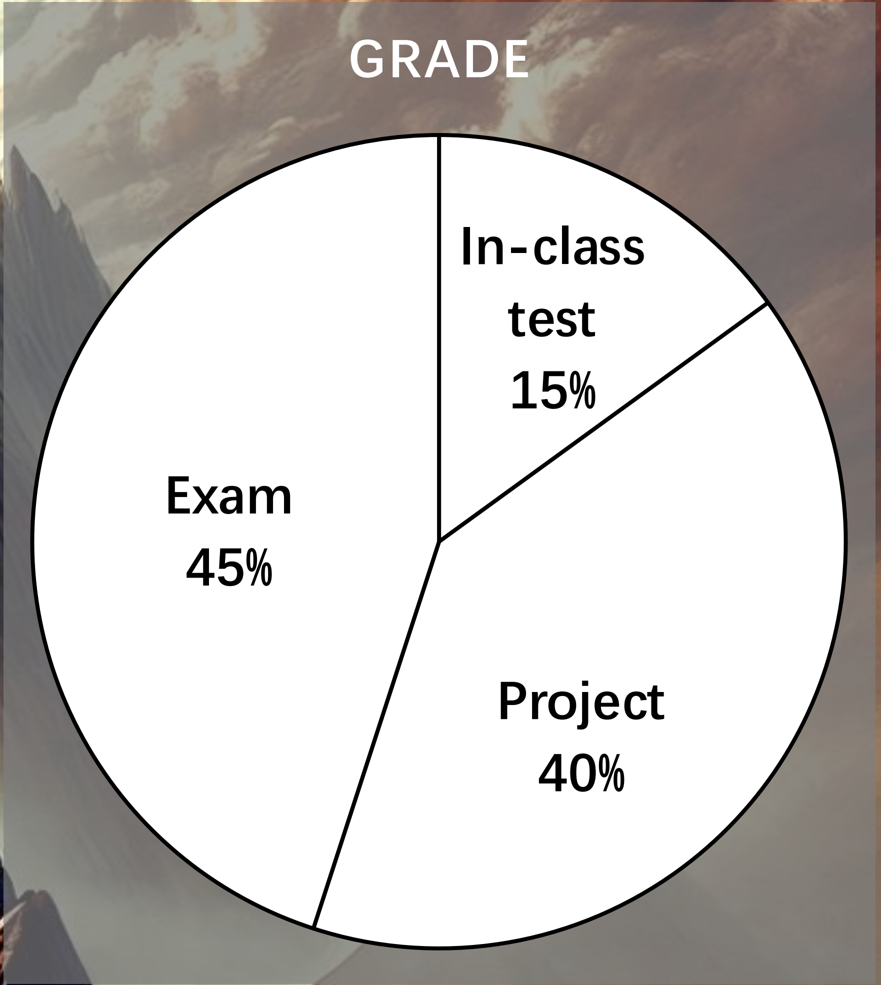
WAIT 2 DAYS

IF I DON’T REPLY, SEND ME AGAIN…

Or call 81889089 -> forward to my mobile

Assessments

- In class test: 15%
- Project: 40% (Two Parts/Group work)
- Final Exam: 45%



Grouping

- Free grouping.
- 4-5 students in one group.
- Try to find you group members!
- I will release a website next week for you to register the group!

In Class Test

In-Class Test (15% of the module mark)

Assessment Type: *CW*
Learning outcomes assessed: *AB*
Duration: *N/A*
Resit opportunity: *S*

Assessment Task	Learning Outcomes	Weighting	Release Date	Due Date
IN-CLASS TEST	AB	15%	20/Nov/2025	21/Nov/2025 20:00
Artificial Intelligence Permissions	Not allowed.			
Brief Description of the Assessment Task				
Fill-in-the-blank questions, testing basic knowledge.				

Project

Assessment Task	Learning Outcomes	Weighting	Release Date	Due Date
CW Part 1	ALL	20%	07/Oct/2025	17/Nov/2025 23:59
Artificial Intelligence Permissions	Not allowed.			
Brief Description of the Assessment Task				
File uploading and downloading should be one of the m ost im portant network-based applications in our daily life. This part of the networking pro ject aim s to use Python Socket program m ing to im plem ent a client-side application for file uploading and downloading based on a given protocol. The exam iner will define and release the protocol description and the server-side application on Learning m all. However, the released server-side code m ight have som e syntax bugs. The student should firstly fix all the bugs and run the server-side code. Then, you should im plem ent the client-side application using Python and test your code using the server-side application.				

Assessment Task	Learning Outcomes	Weighting	Release Date	Due Date
CW Part 2	ALL	20%	04/Nov/2025	12/Dec/2025 22:00
Artificial Intelligence Permissions	Not allowed.			
Brief Description of the Assessment Task				
This part of the networking project aim s to use Mininet to create a sim ple SDN network topology and em ulate a traffic control function through u sing the SDN flow entry. Assum ing that the client side only knows the service running on server 1 and com m unicates with server 1 (without know ing the existence of the service on server 2). However, the SDN controller can m anipulate (forward/redirect) the traffic without the awareness of the client. The student needs to develop the SDN controller application to create effective flow entry to m ake the incom ing network traffic forwar d or redirected.				

Assignment Policies

Programming projects are ~ 20% of your grade

Assignments will be done in Python 3 and other plugins/modules.

Late submission will follow the university late submission policy [1].

Academic dishonesty will not be tolerated.

[1] Item 42-43, Part F, Code of Practice on Assessment of XJTLU.

Academic Dishonesty Policy

- **Academic dishonesty is anything that involves you not doing the assignment yourself, including:**
  - Copying from your “friend”
  - Copying from an internet or other source
  - Paying someone to complete the assignment
- **Submitting a course assignment that is not your work falls under the University major plagiarism or collusion policy. The penalties for this are:**
  - Automatic failure of the course, with NO RESIT
  - A permanent mark on your transcript
  - A letter to your parents notifying them of your actions
- **The use of Generative AI for content generation is not permitted on all assessed coursework in this module.**

Lecture Syllabus

Week 1

Lecture 1 - the theoretical basis for networking:
protocol, layer and model…

Week 2

Lecture 2 - the application layer:
principles, web and HTTP…

Week 2

Lecture 3 - the application layer:
Email, DNS, P2P, multimedia protocols…

Week 4

Lecture 4 - the transport layer:
multiplexing, UDP and reliable data transfer…

Week 5

Lecture 5 - the transport layer:
TCP and congestion control…

Lecture Syllabus

Week 6 — Lecture 6 - the network layer: router, internet, IPv4 and IPv6 …

Week 7: no class…

Week 8 — Lecture 7 - the network layer: routing algorithm…

Week 9 — Lecture 8 - the network layer: control plane …

Week 10 — Lecture 9 - The link layer 1

Lecture Syllabus

Week 11 — Lecture 10 - The link layer 2, Network Security 1

Week 12 — Lecture 11 - Network Security 2

Week 13 — Lecture 12 - Network Security 3 and review

Happy ending (Hopefully)

Lab Sessions

Lab time
SC464

Thu. or Fri.

Labs will be split between **Python** and **networking** experiments

Attendance to labs is required

Labs & Projects Plan

Week Number and/or Date	Lab	Topic/Theme/Title	Lecturer
Week 1	Lab 1	Linux OS, Virtual Machine and Mininet	Fei & Gordon
Week 2	Lab 2	Wireshark and Tcpdump	Fei Cheng
Week 3	Lab 3	Python1: IDE and Interpreter	Fei Cheng
Week 4	Lab 4	Python2: Basic syntax & Data structure	Fei Cheng
Week 5	Lab 5	Python3: Network programming (Socket)	Fei Cheng
Week 6	Lab 6	Python4: Modules,Pypi,OOP,Parallel Computing	Fei Cheng
Week 8	Lab 7	Building network topology using Mininet	Gordon
Week 9	Lab 8	SDN controller application with Mininet	Gordon
Week 10	Lab 9	DoS attack on SDN	Gordon
Week 11	Lab 10	TCP connection analysis	Gordon
Week 12	Lab 11	Scanner and Firewall	Gordon
Week 13	Lab 12	Intrusion Detection System	Gordon

New Lab room Arrangement

- We will use the Linux Computer Lab

Ubuntu

Computer Lab Disclaimer
计算机实验室免责声明

Welcome to the SC464 Computer Lab. To ensure a productive and secure environment for all users, please be aware of the following guidelines:

欢迎来到 SC464 计算机实验室。为了确保所有用户有一个高效和安全的环境，请注意以下指南：

1. **Backup Your Work**: Please save and backup your homework or task files on your own device before you shut down or leave the computer. All files stored on these computers will be **permanently removed** upon reboot.

   备份您的工作：在关闭或离开计算机之前，请将您的作业或任务文件保存在您自己的设备上并备份。这些计算机上的所有文件将在重启后被永久删除。

2. **Security and Conduct**: Do not engage in any malicious activities or violate the security policies while using these computers. Any individual found responsible for such actions will be held accountable for their behaviour and subject to disciplinary actions.

   安全和行为：在使用这些计算机时，不要从事任何恶意活动或违反安全政策。任何被发现负责此类行为的个人将对其行为负责，并将面临纪律处分。

3. **Respect Shared Resources**: Please be considerate of other users. Do not install unauthorized software, alter system settings, or use excessive bandwidth.

   尊重共享资源：请考虑其他用户的需求。不要安装未经授权的软件，修改系统设置或使用过多的带宽。

4. **Privacy Notice**: Be aware that this is a shared environment. Avoid storing sensitive personal information on these computers as it may not be secure.

   隐私注意：请注意，这是一个共享环境。出于安全考量，避免在这些计算机上存储敏感的个人信息。

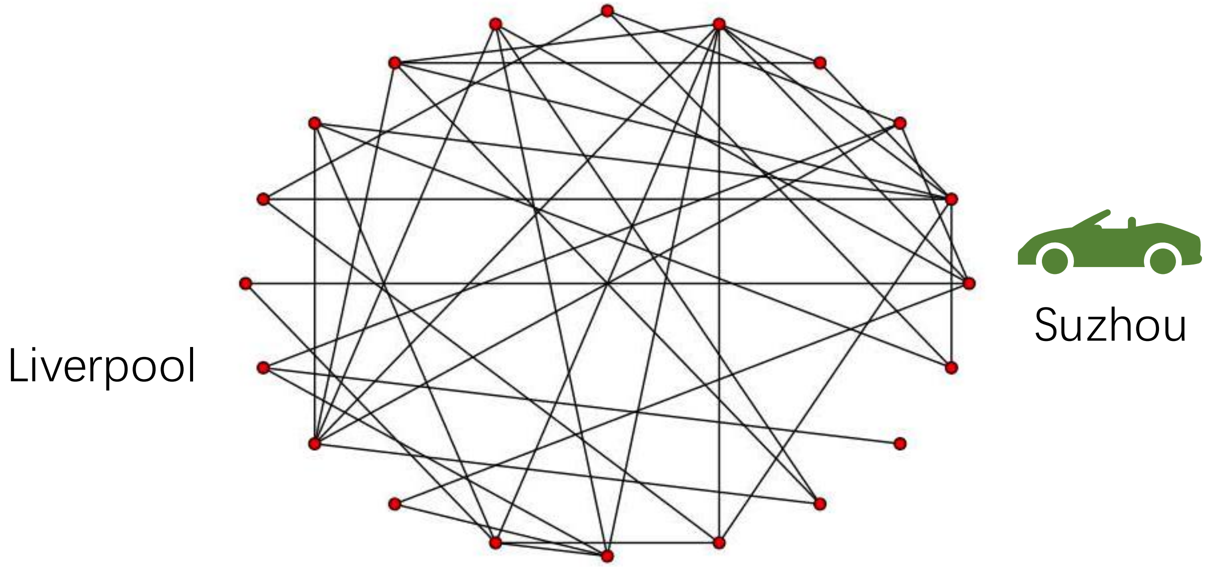
5. **Report Issues**: If you encounter any technical problems or notice any suspicious activity, please report it to the lab administrator immediately.

   报告问题：如果您遇到任何技术问题或注意到任何可疑活动，请立即向实验室管理员报告。

Lecture 1 - Introduction

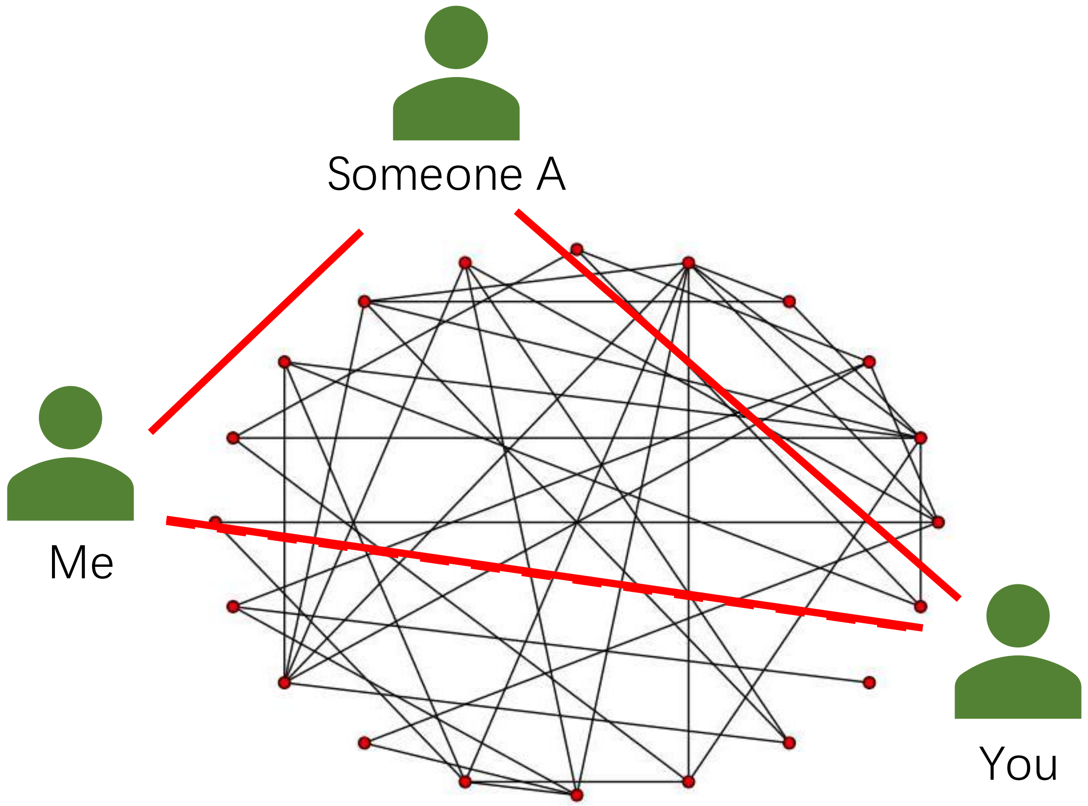
1. What is the network/internet?
2. How does the network work?
    - Network edge
    - Network core
3. How to evaluate the performance?
    - loss, delay, throughput
4. Service models
    - TCP/IP

Network



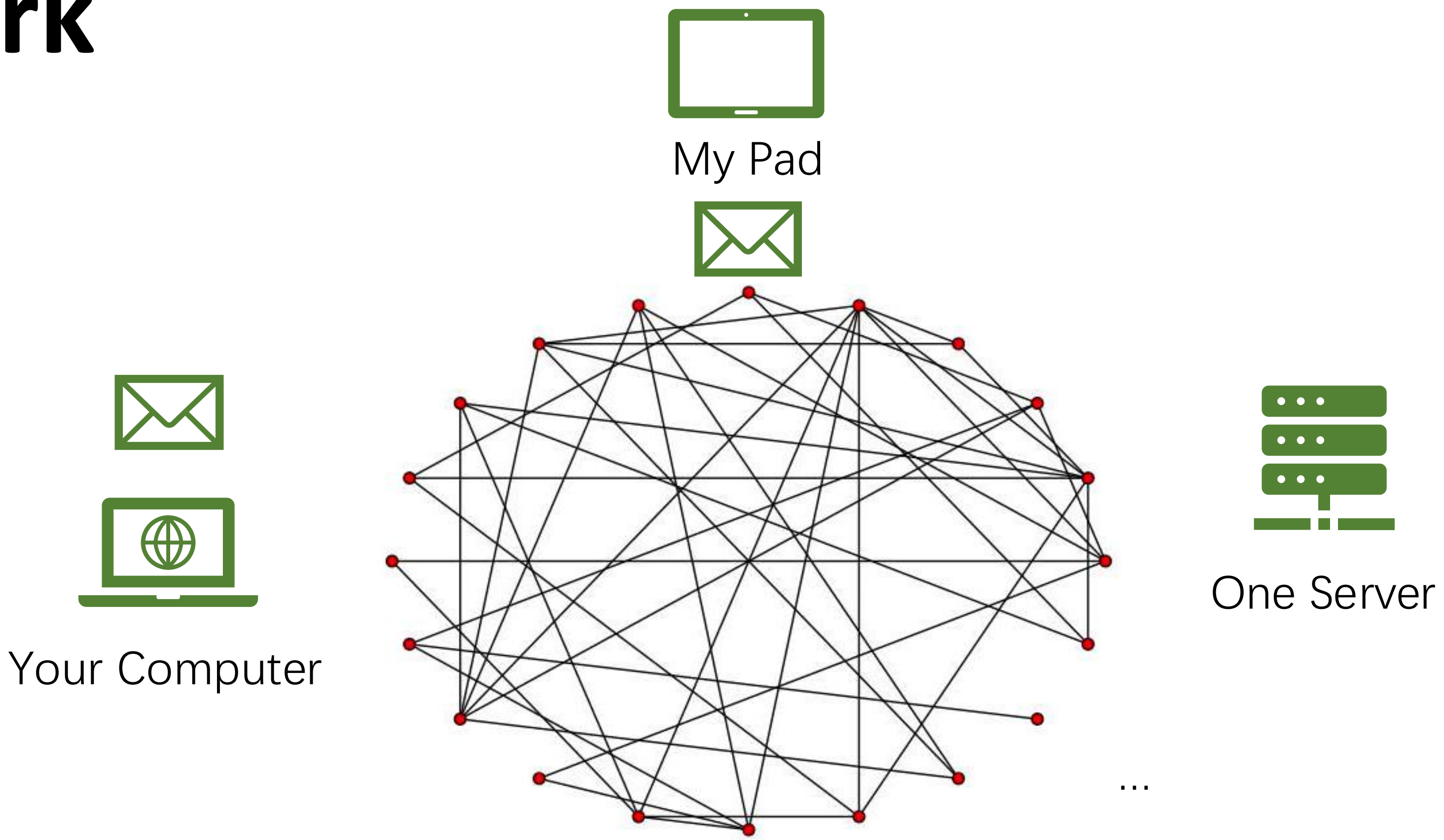
Road Network

Network



Social Network

Network



Computer Network

Devices

Others?

Services

....

INTERNET

Largest

- Project
- Users
- Devices

almost **Every**

- Everywhere
- Everyday
- Everything

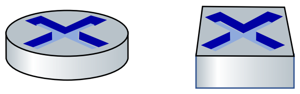

INTERNET

Lecture 1 - Introduction

1. What is the network/internet?
2. How does the network work?
   - Network edge
   - Network core
3. How to evaluate the performance?
   - loss, delay, throughput
4. Service models
   - TCP/IP

The whole picture of the network / internet

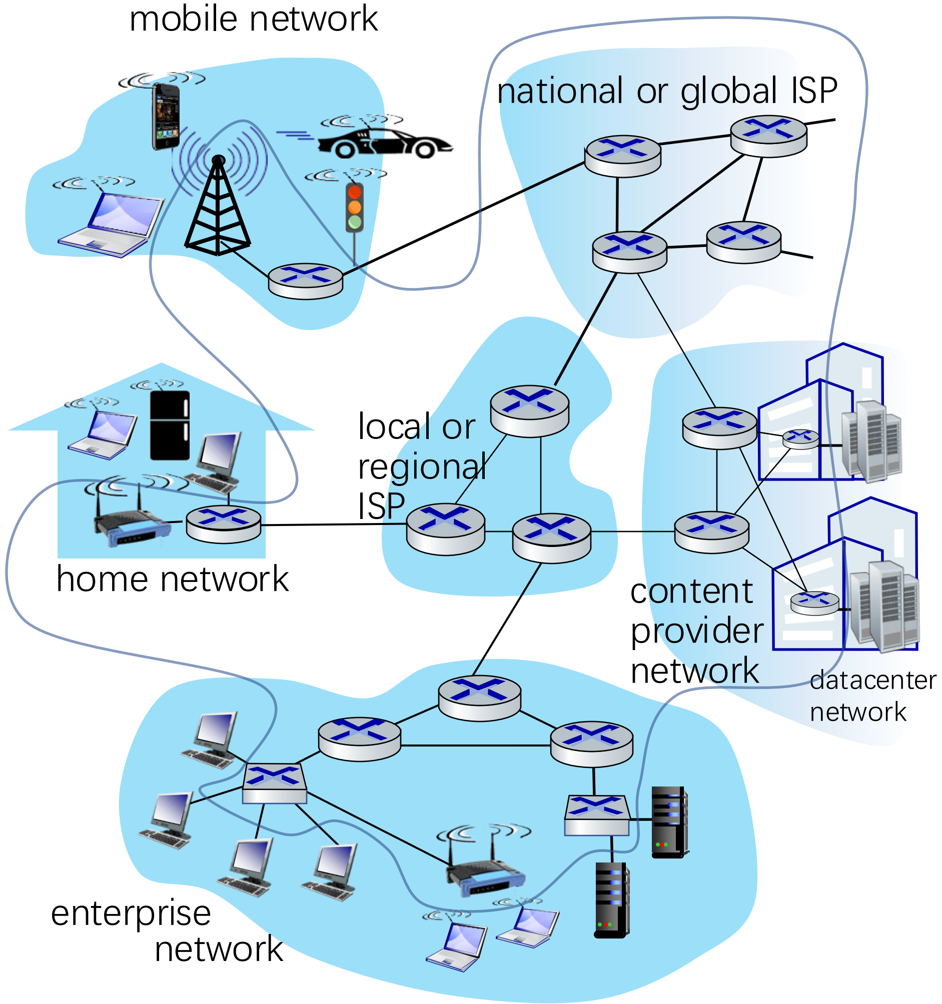
Billions of connected computing *devices*:
- *hosts = end systems*
- running *network apps* at Internet's "edge"



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

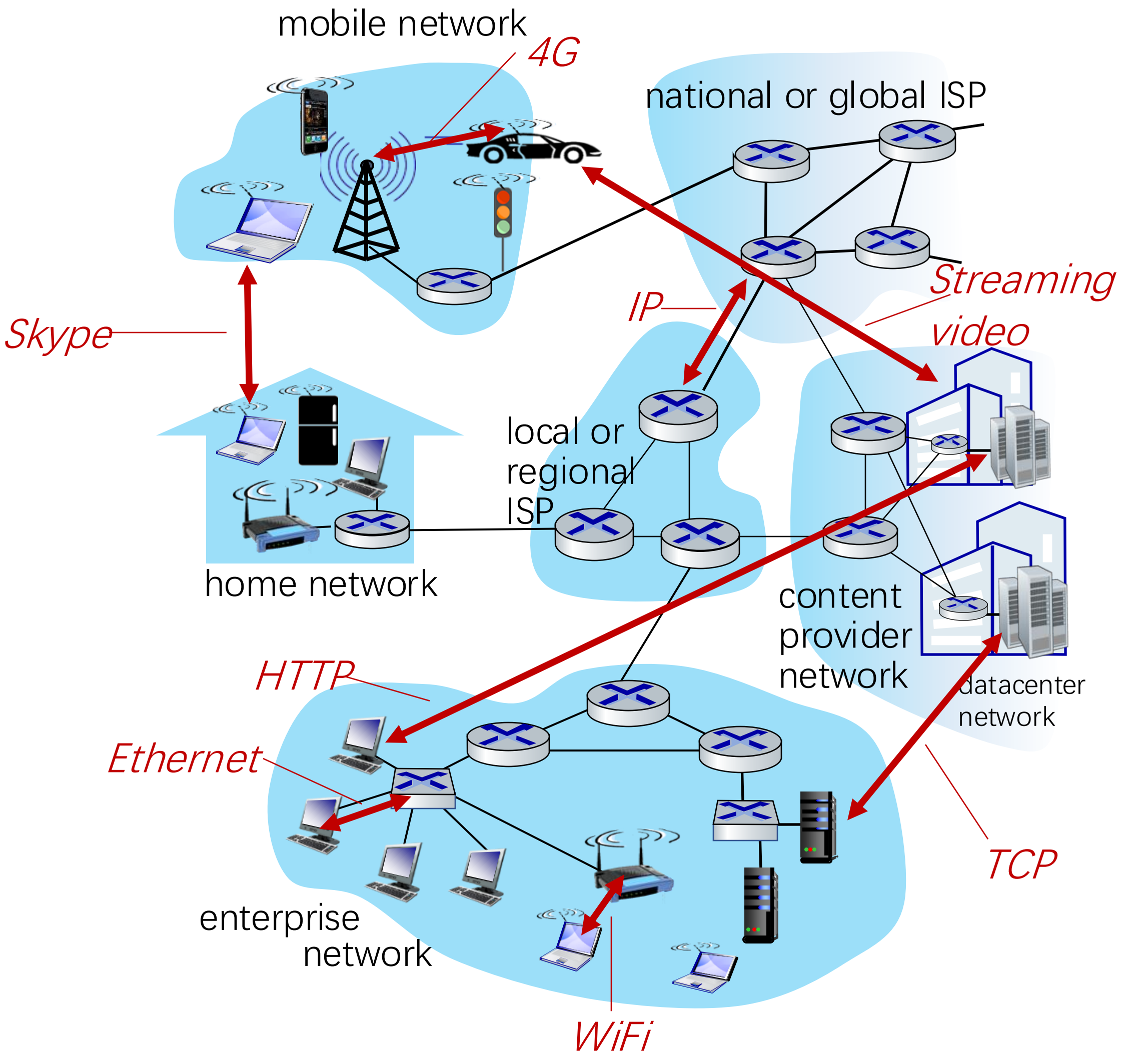
Communication links
- fiber, copper, radio, satellite
- transmission rate: *bandwidth*

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



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, Zoom, TCP, IP, WiFi, 4/5G, Ethernet
- *Internet standards*
  - RFC: Request for Comments
  - IETF: Internet Engineering Task Force



Protocols (协议)

Human protocols:

- “what’ s the time?”
- “I have a question”

Rules for:

… specific messages sent

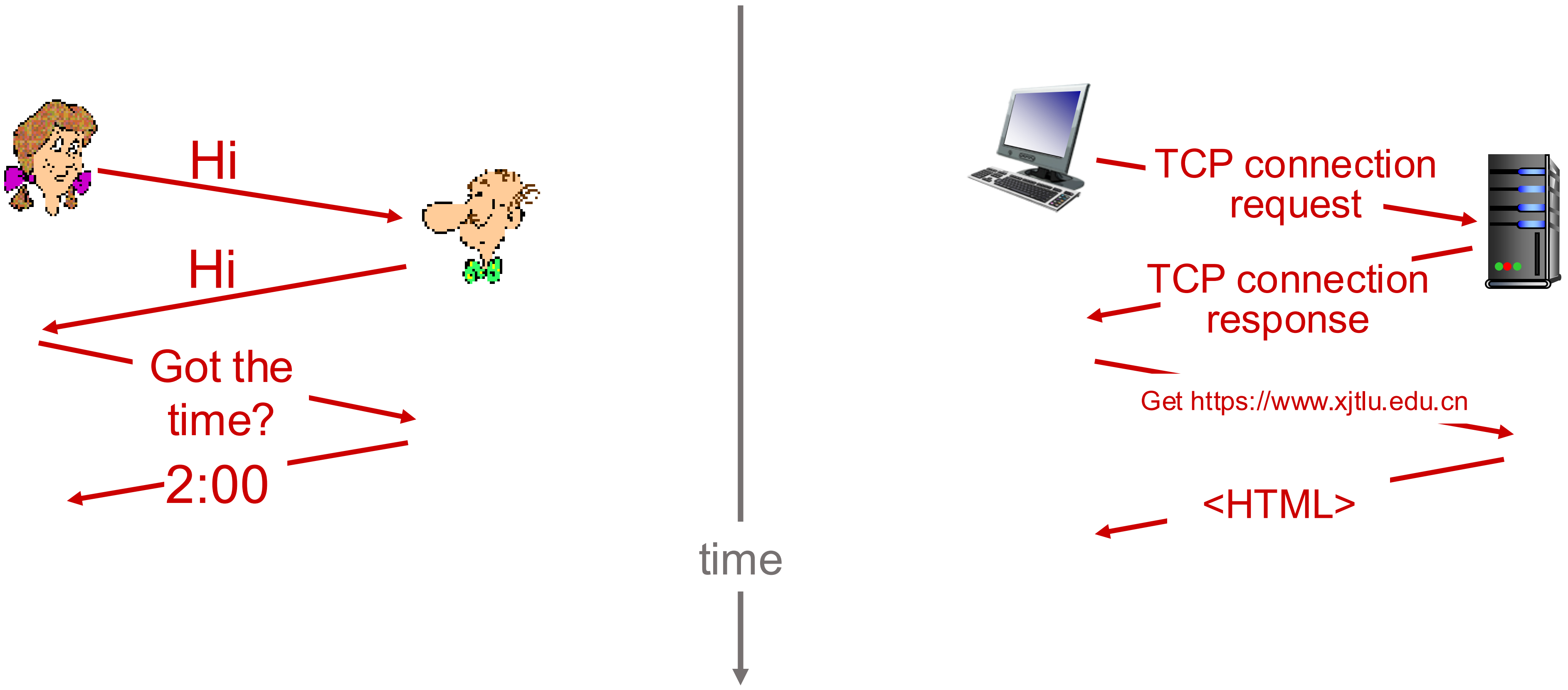
… specific actions taken
when message received,
or other events

Network protocols:

- Computers (devices) rather than humans
- All communication activity in Internet governed by protocols

A set of rules for how data is transmitted across the network

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

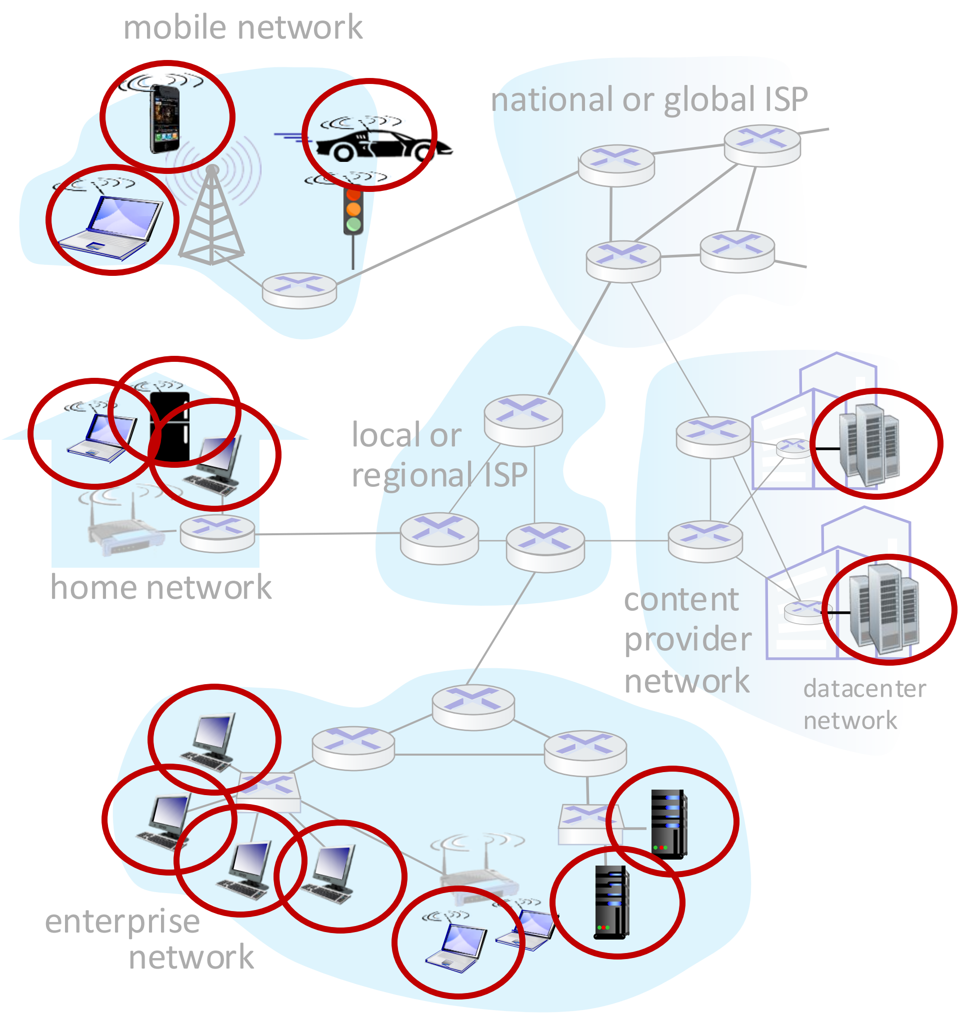


Hi
Hi
Got the time?
2:00
time
TCP connection request
TCP connection response
Get https://www.xjtlu.edu.cn
<HTML>

A closer look at Internet structure

Network edge:

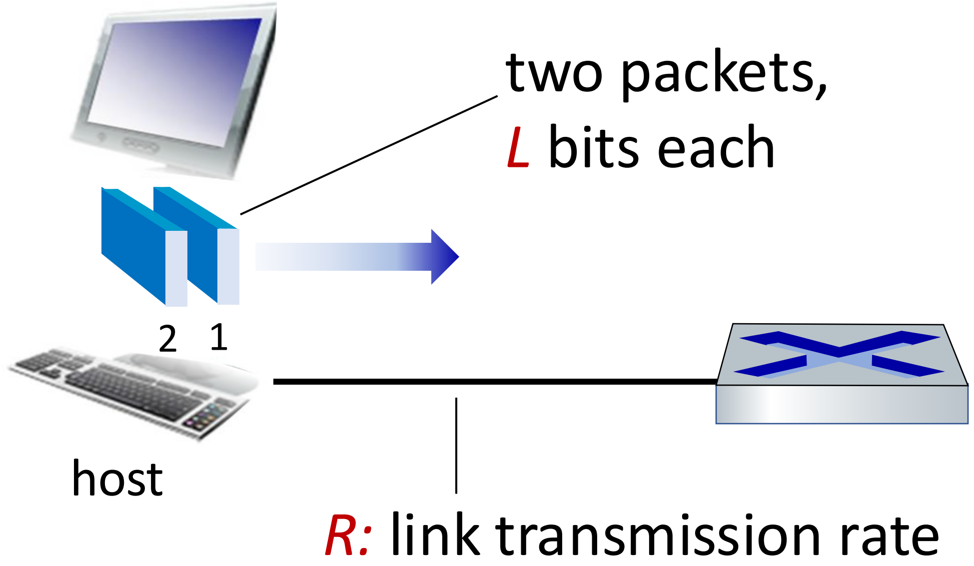
- hosts: clients and servers
- servers often in data centers



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*



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

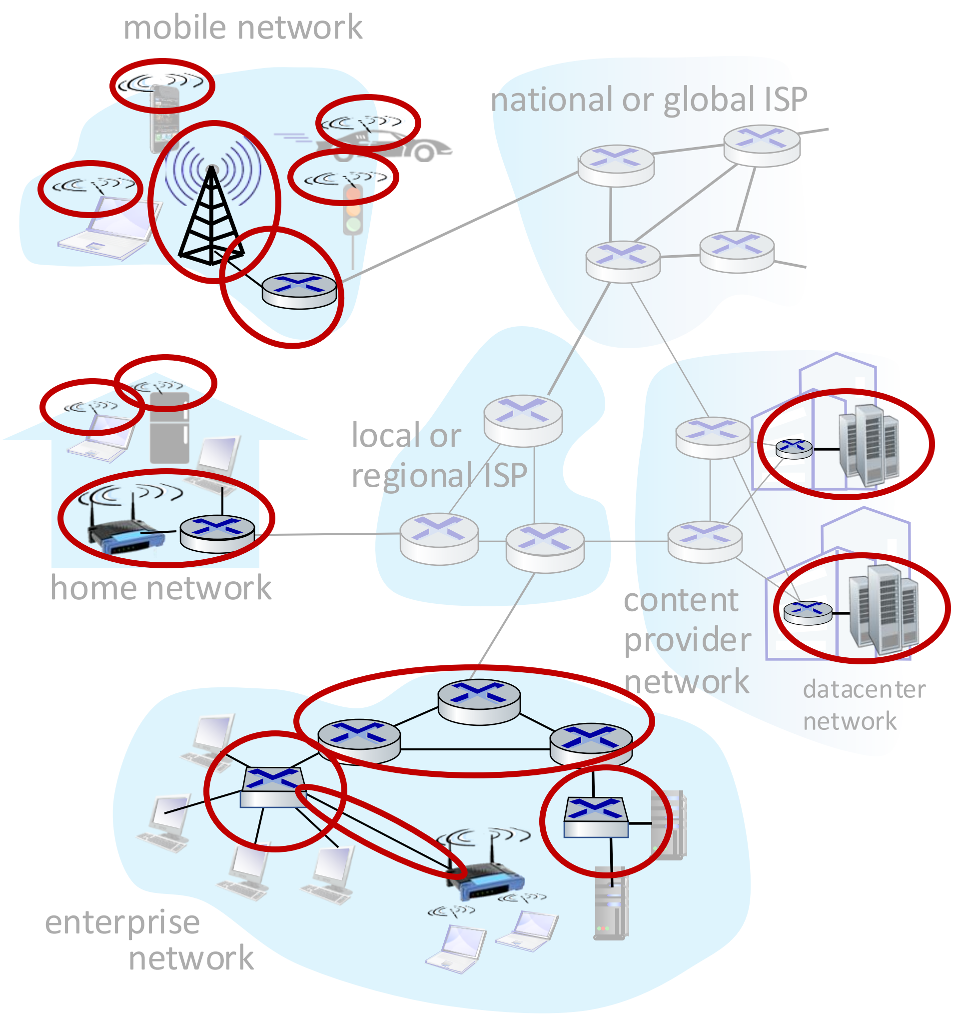
A closer look at Internet structure

Network edge:

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

Access networks, physical media:

- wired, wireless communication links



Physical media

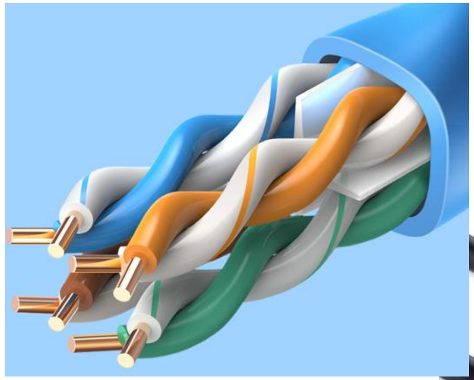
- **Guided media:**
  - signals propagate in solid media: copper, fiber, coaxial cable, glass

- **Unguided media:**
  - signals propagate freely, e.g., radio (electromagnetic waves)

Physical media: twisted pair, coax, fiber

Twisted pair (TP) 双绞线

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



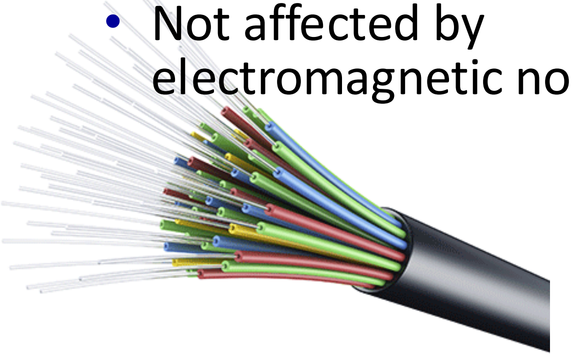
RJ-45
Web socket

Coaxial cable: 同轴电缆

- two concentric copper conductors
- bidirectional
- broadband:
  - multiple channels on cable
  - HFC

Fiber optic cable: 光纤

- glass fiber carrying light pulses, one pulse = a bit
- high-speed operation:
  - high-speed point-to-point transmission (e.g., 10-100 Gbps transmission rate)
- low error rate:
  - repeaters spaced far apart
  - Not affected by electromagnetic noise



Physical media: radio

- Signal carried in electromagnetic spectrum
- No physical "wire"
- Bidirectional
- Propagation environment effects:
  - Reflection
  - Obstruction by objects
  - Interference!

Radio link types:

- Wireless LAN (e.g., WiFi)
  - 54 Mbps – 9.6Gbps
- Wide-area (e.g., cellular)
  - 4G cellular: ~ 100 Mbps
  - 5G cellular: ~ 1Gbps
- Satellite
  - Kbps to 45Mbps channel (or multiple smaller channels)
  - Starlink ~ 1440Mbps
  - 270 msec end-end delay

Different methods to access the internet

Early methods:

PSTN
DSL
TV Net

Modern methods:

FTTH
Wireless

Dial-up Internet access (PSTN)

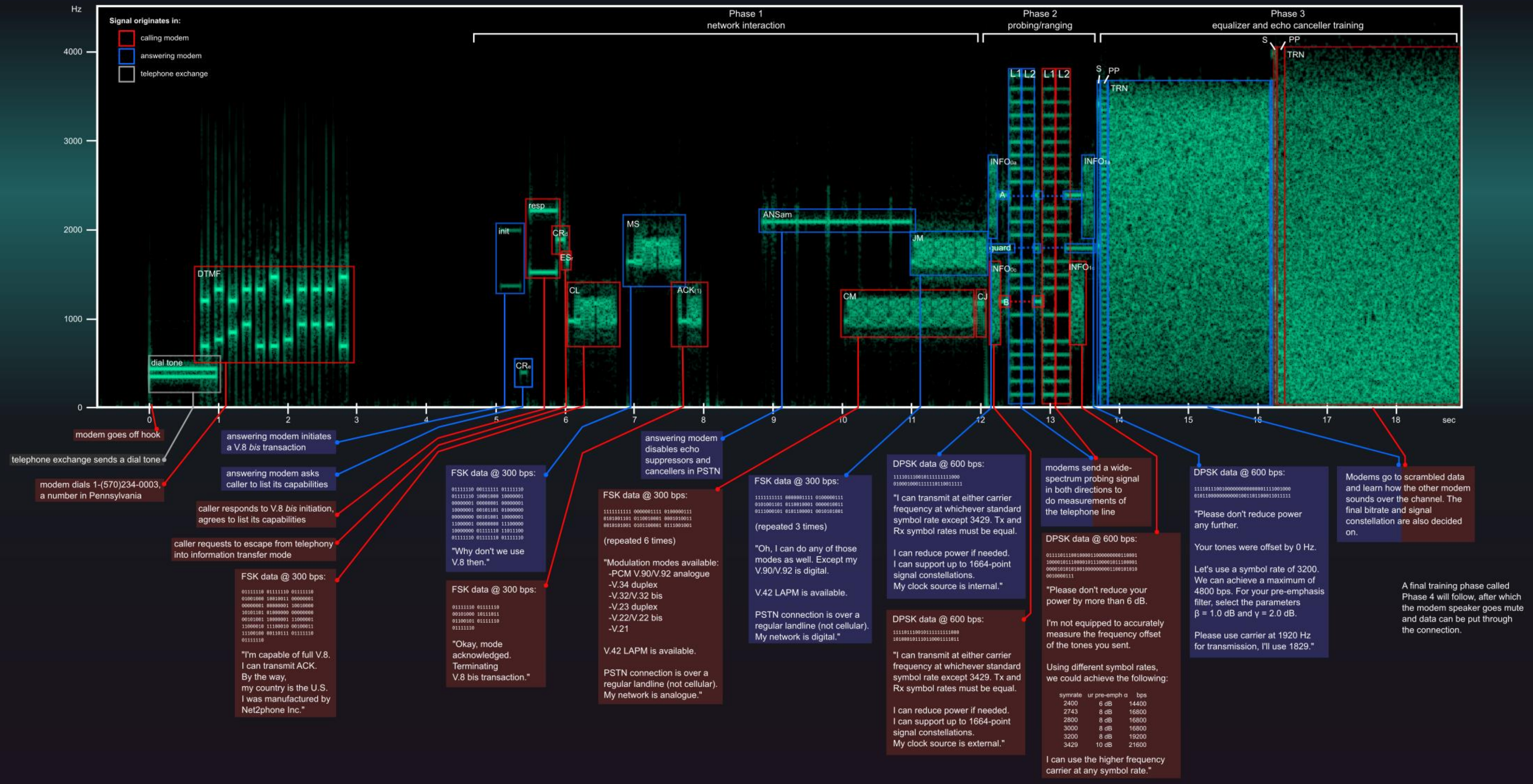
- **Dial-up Internet access** uses the **public switched telephone network (PSTN)** to connect to an Internet service provider (ISP).
- A connection is established by **dialing a telephone number** over a conventional telephone line.
- **Modems** are required to enable communication:
  Modulator and demodulator
  - They **decode audio signals into data** for computers or routers.
  - They **encode digital signals back into audio** for transmission to another modem.

Bandwidth: 56 Kbps ~ 10 min for one mp3 music file

https://en.wikipedia.org/wiki/Dial-up_Internet_access

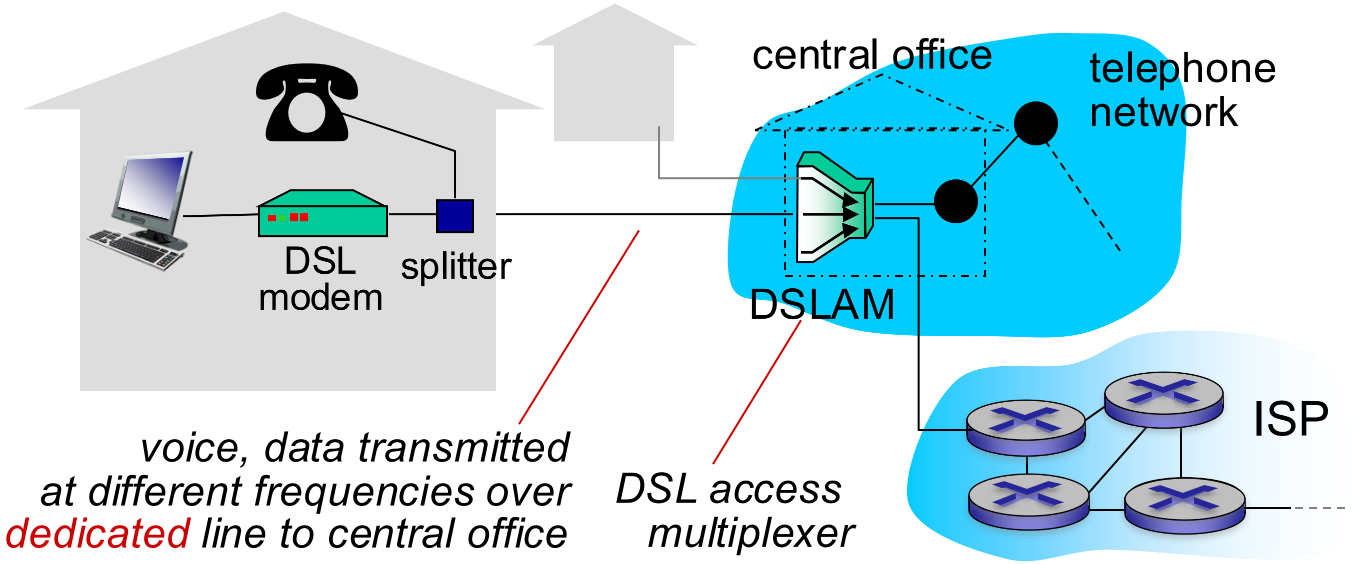
Listen Please

The Sound of the Dialup: an Example Handshake



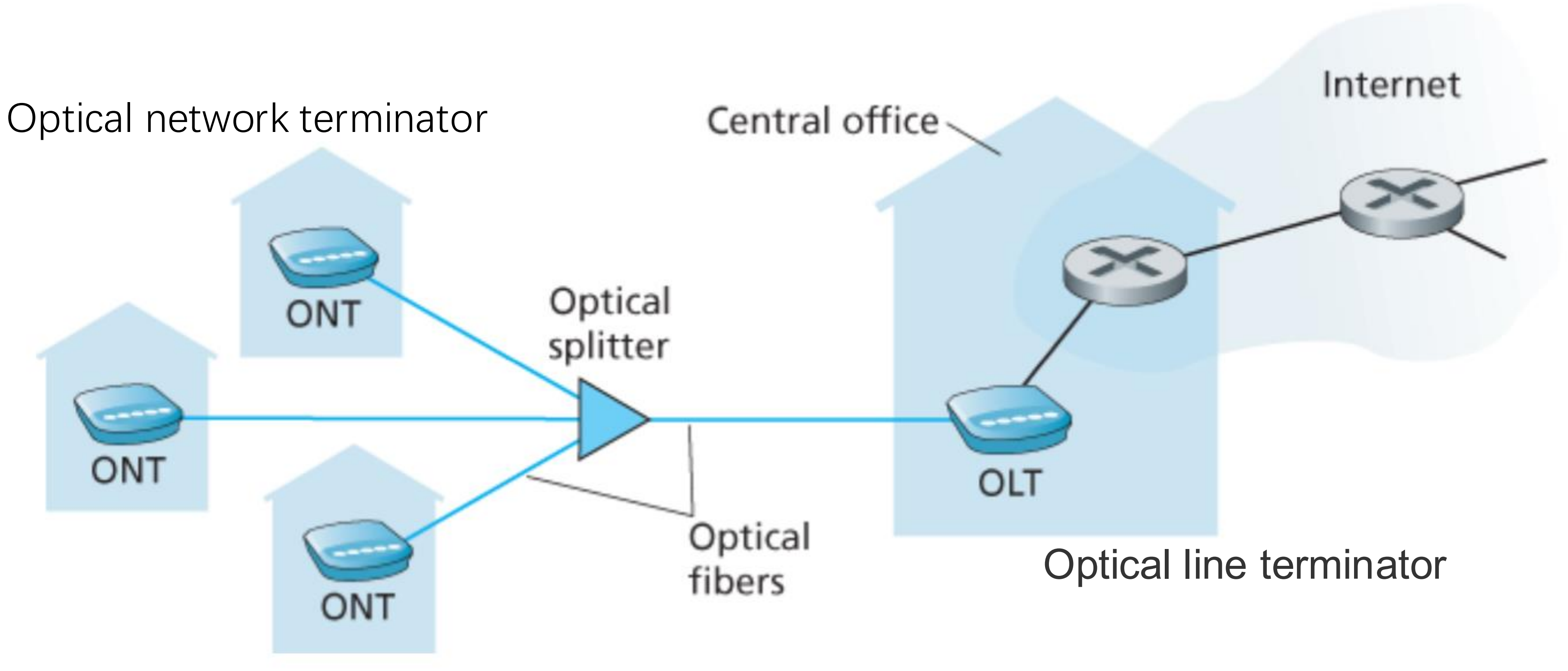
Digital subscriber line (DSL)

- **Telephone line based:**
  - To central office DSL Access Multiplexer (DSLAM)
  - Data over DSL phone line goes to Internet
  - Voice over DSL phone line goes to telephone net



- **Bandwidth**
  - Upstream transmission rate ~< 3 Mbps (typically < 1Mbps)
  - Downstream transmission rate ~< 50 Mbps (typically < 10Mbps)
  - ADSL = Asymmetric Digital Subscriber Line

FTTH: fiber to the home



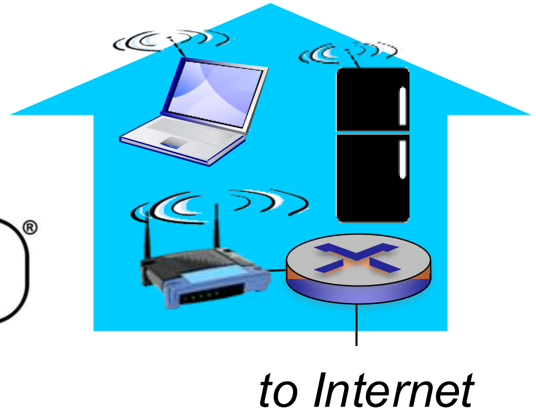
FTTH using the passive optical networks (PONs) distribution architecture

Wireless access networks

- **Base Stations should be used to provide the signal!**

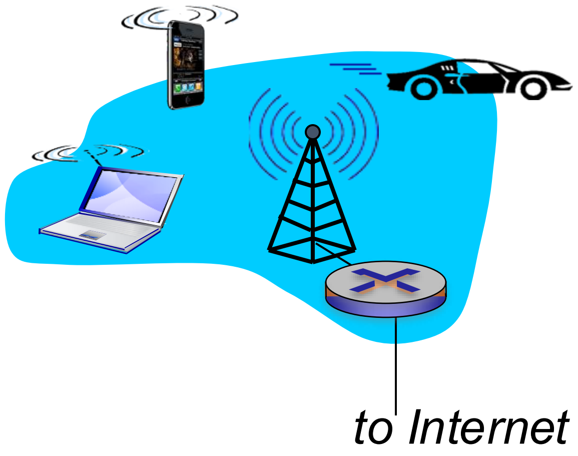
Wireless LANs:

- within building (30 m)
- 802.11b/g/n (WiFi): 11, 54, 450 Mbps transmission rate
- 802.11ax : 9.6 Gbps (MIMO)



Wide-area wireless access

- provided by ISP
- Up to 1Gbps or More
- 3G, 4G-LTE, 5G, Starlink

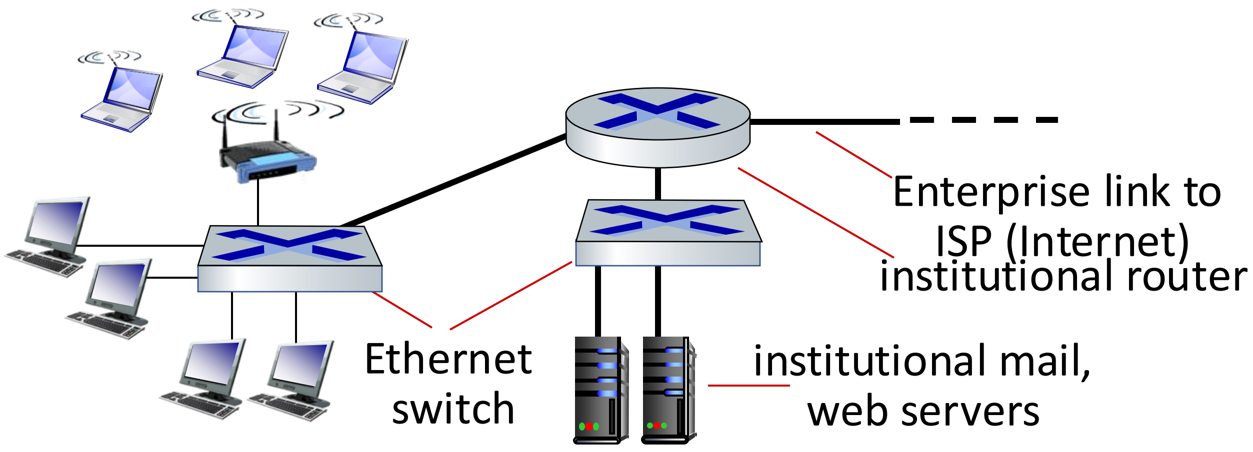


Wireless access networks

STARLINK

- 100 Mbps to 120 Mbps

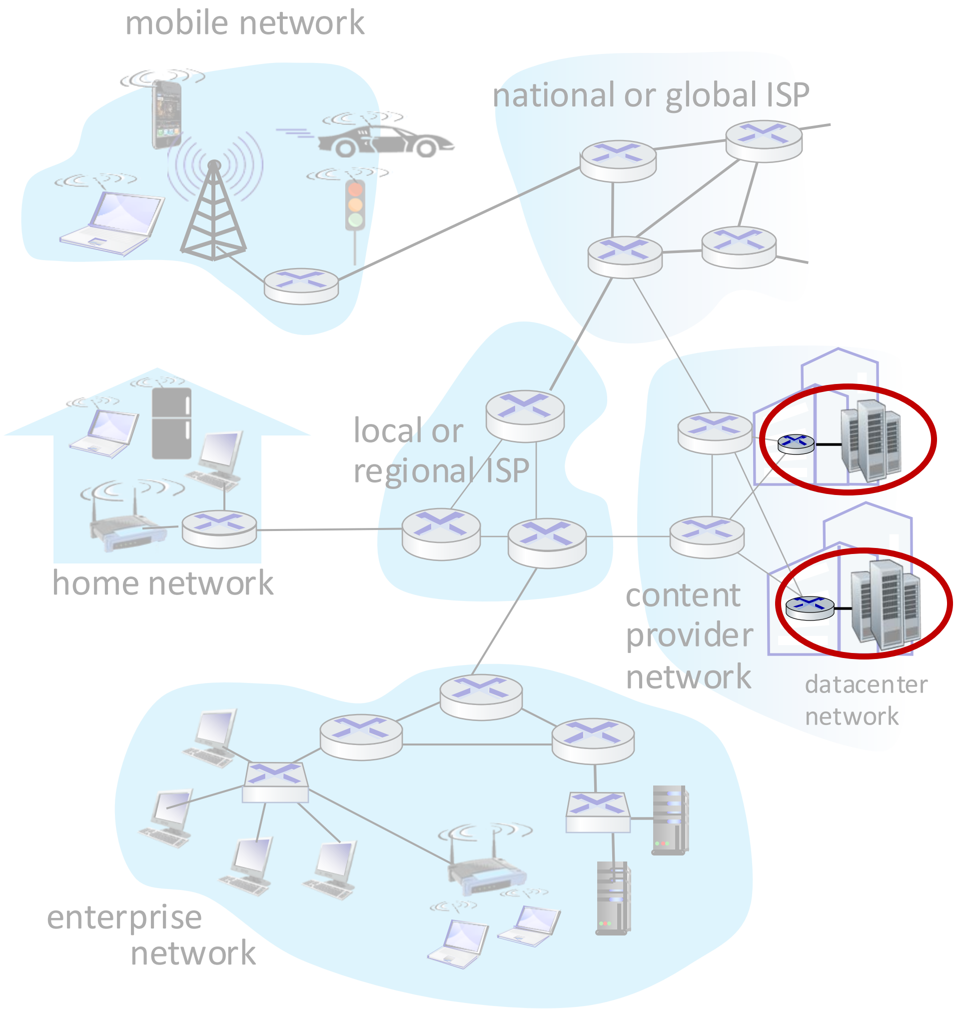
Access networks: enterprise networks



- 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

Access networks: data center networks

- High-bandwidth links connect hundreds to thousands of servers together, and to Internet.



A closer look at Internet structure

Network edge:

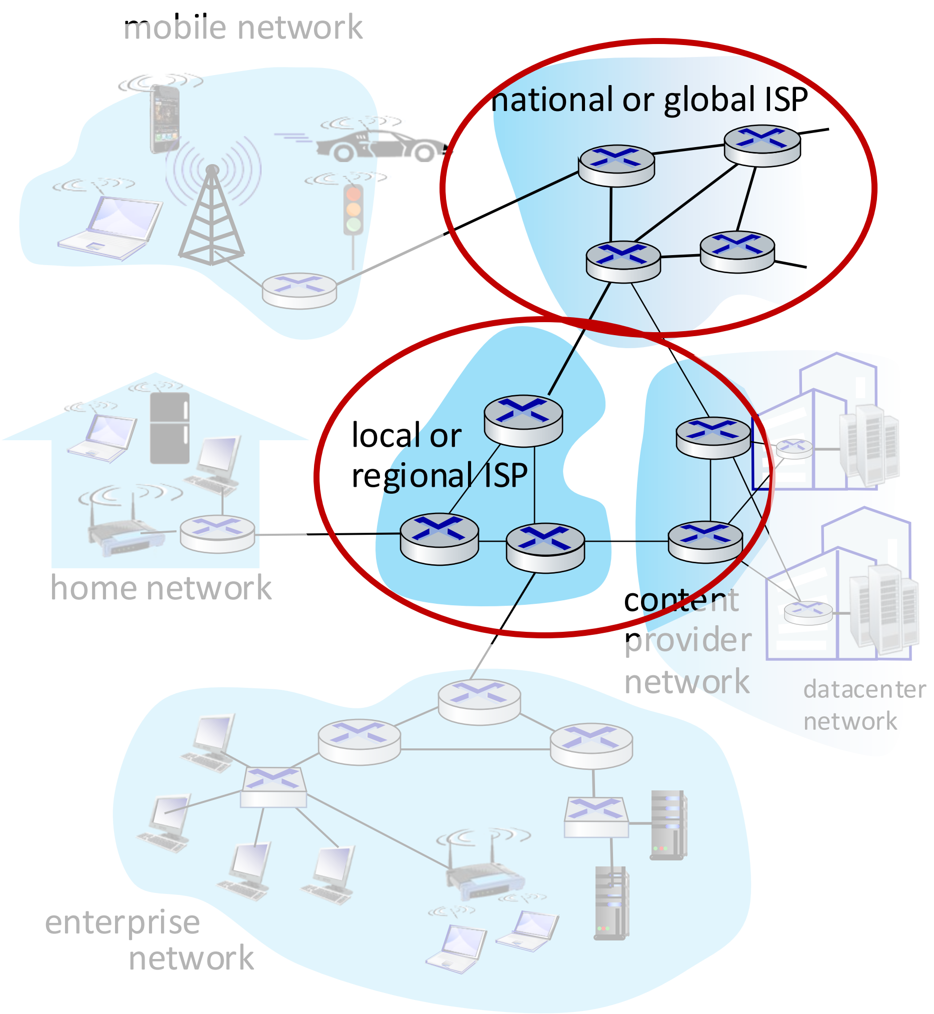
- hosts: clients and servers
- servers often in data centers

Access networks, physical media:

- wired, wireless communication links

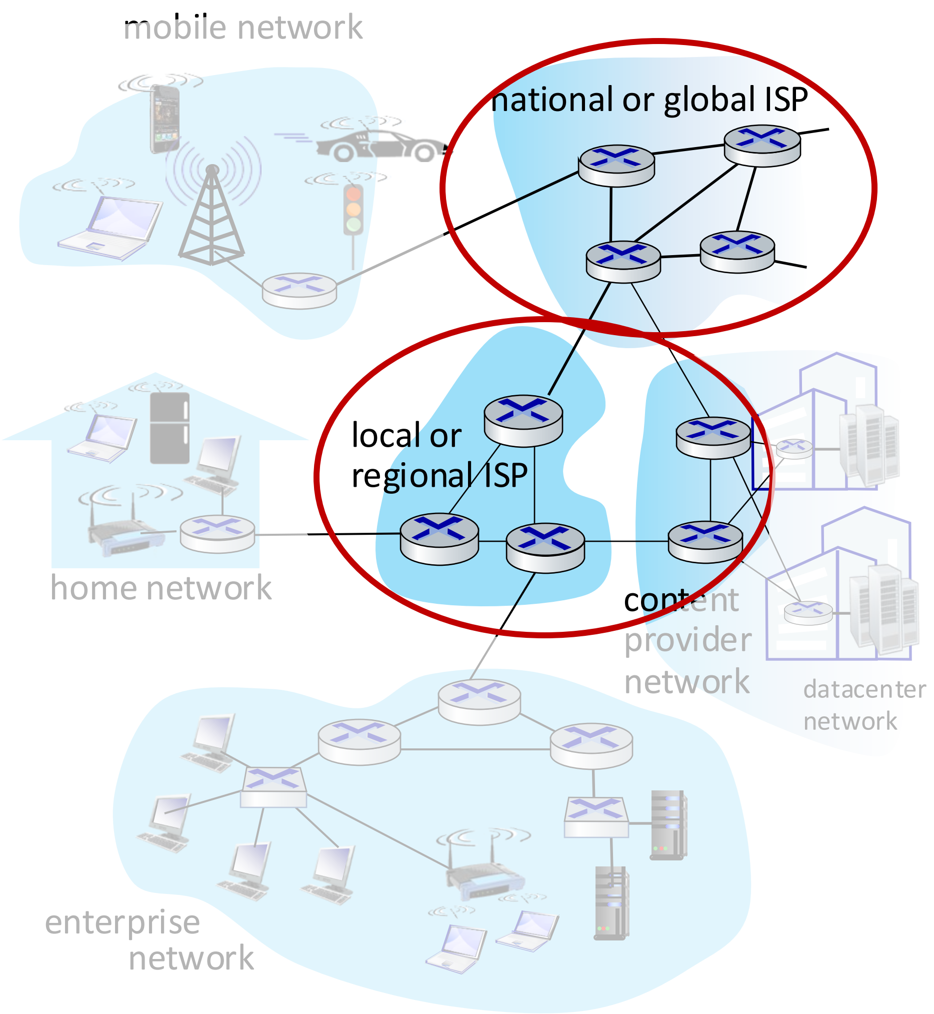
Network core:

- interconnected routers
- network of networks



The network core

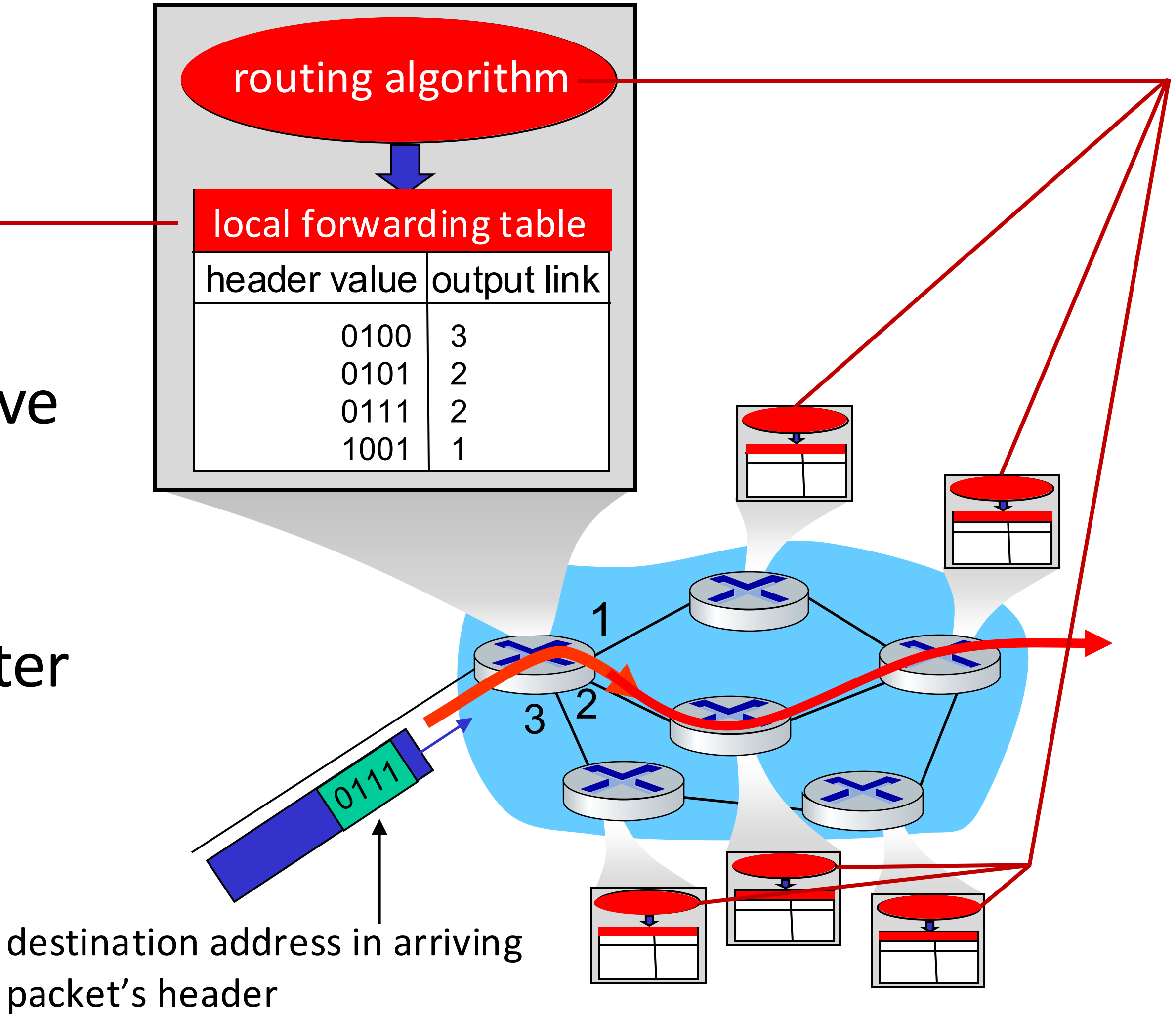
- mesh of interconnected routers
- packet-switching: hosts break application-layer messages into *packets*
  - network forwards packets from one router to the next, across links on path from source to destination



Two key network-core functions

Forwarding:

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

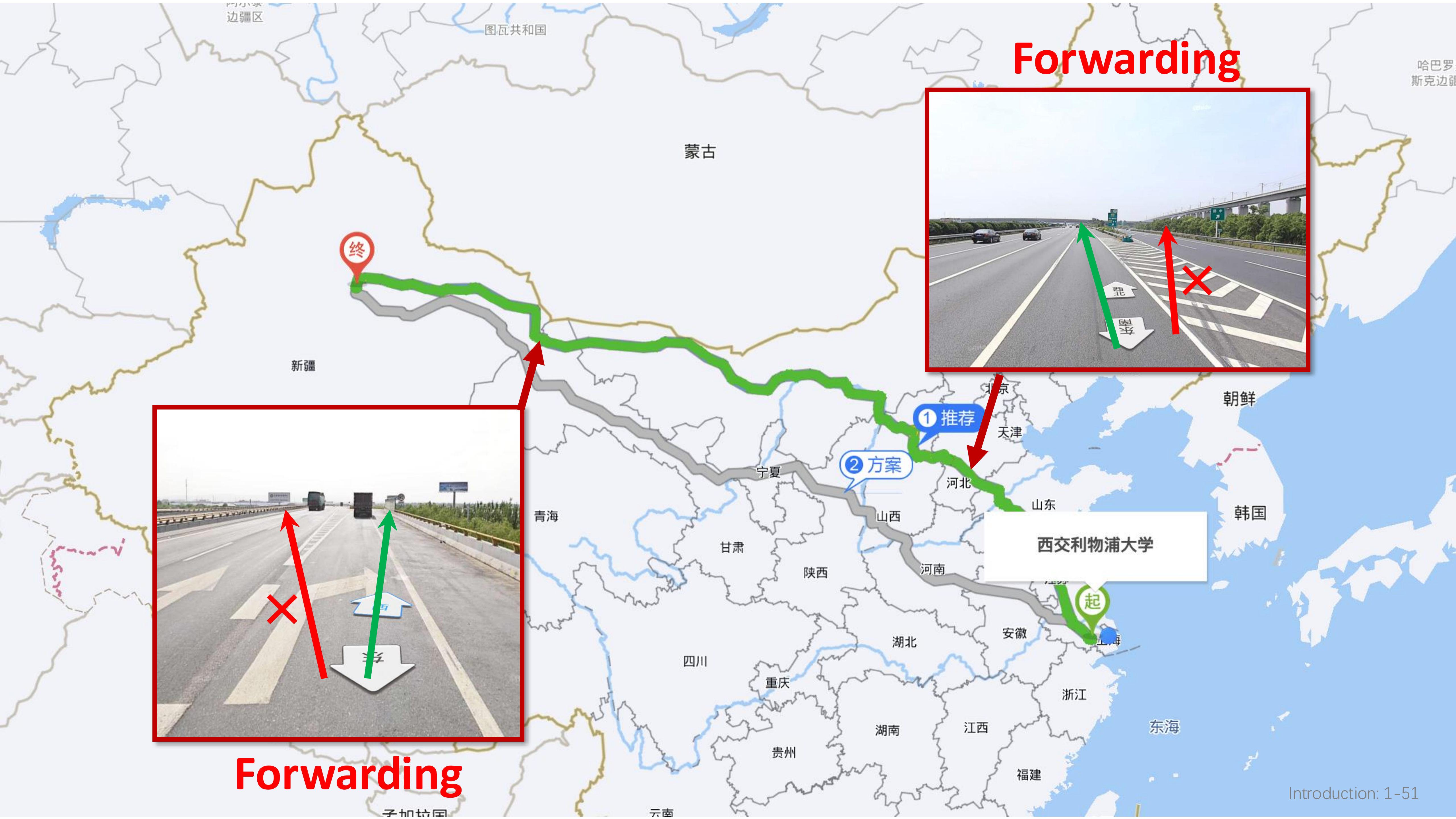


Routing:

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

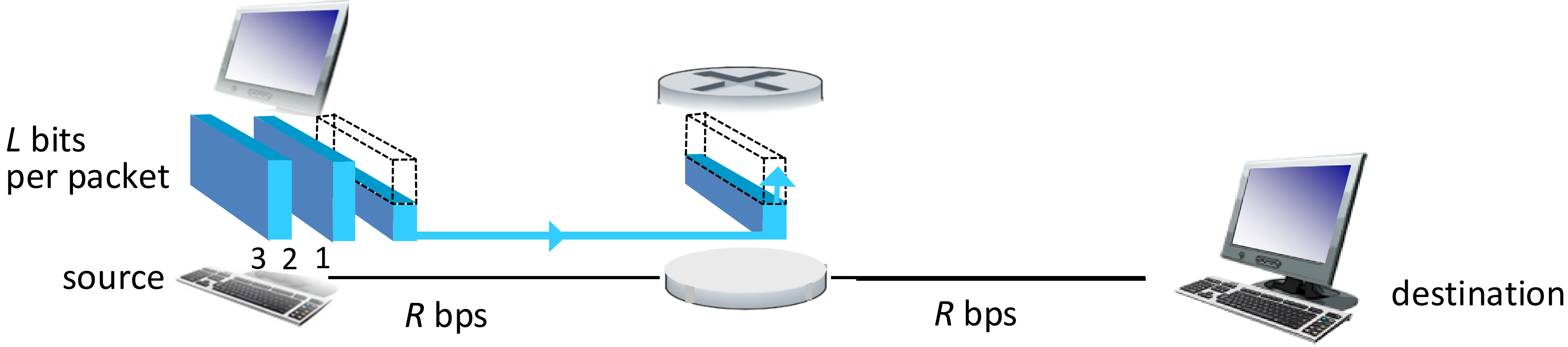
Routing

图瓦共和国
蒙古
黑龙江
内蒙古
吉林
终
新疆
辽宁
北京
朝鲜
1 推荐
天津
2 方案
宁夏
河北
山东
青海
山西
韩国
西交利物浦大学
甘肃
陕西
河南
起
西藏
安徽
湖北
四川
重庆
浙江
湖南
江西
东海
贵州
福建



Forwarding
Forwarding
边疆区
图瓦共和国
蒙古
新疆
终
① 推荐
② 方案
北京
天津
河北
山东
宁夏
山西
青海
甘肃
陕西
河南
西交利物浦大学
起
安徽
湖北
四川
重庆
浙江
湖南
江西
贵州
福建
东海
朝鲜
韩国

Packet Switching: store-and-forward

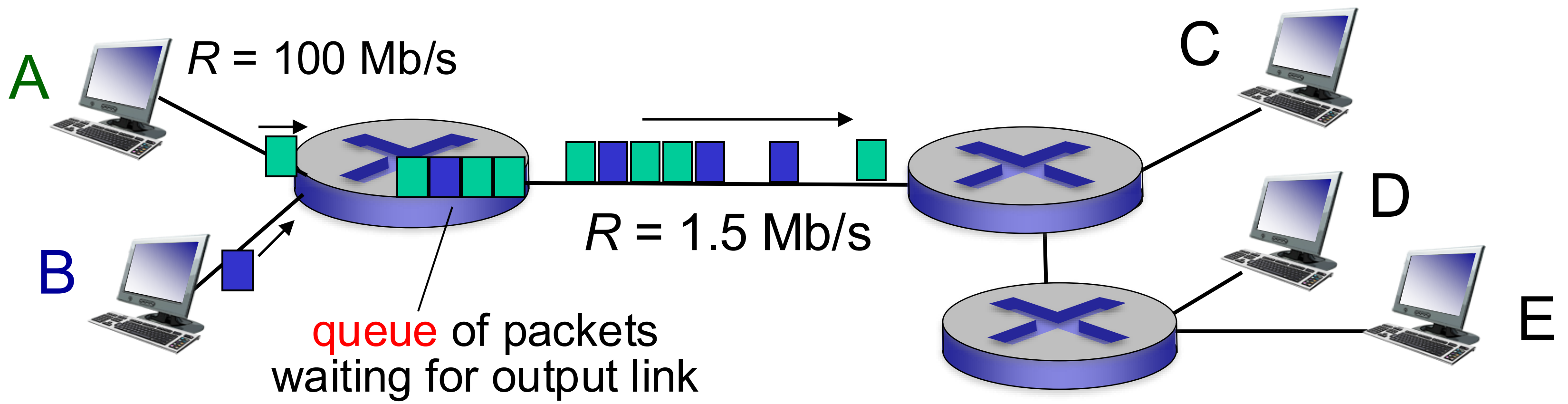


- Takes L/R seconds to transmit (push out) L-bit packet into link at R bps
- *Store and forward:* entire packet must arrive at router before it can be transmitted on next link
- End-end delay = $2L/R$ (assuming zero propagation delay)

One-hop numerical example:

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

Packet Switching: queueing delay, loss



Queuing and loss:

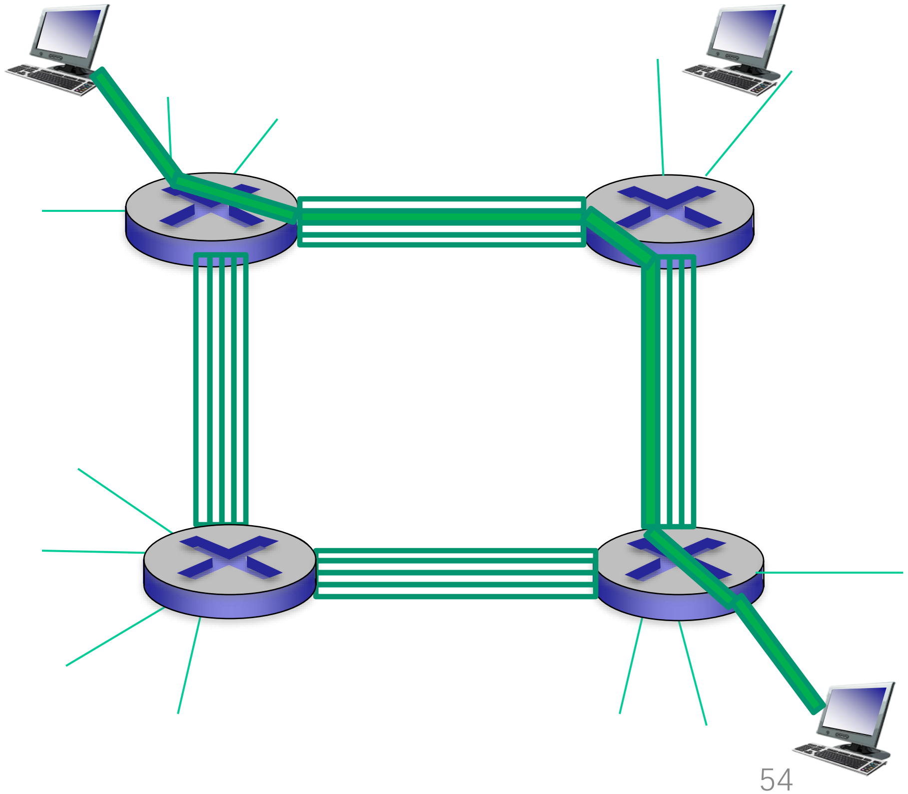
- If arrival rate (in bits) to link exceeds transmission rate of link for a period of time:
  - packets will queue, wait to be transmitted on link
  - packets can be dropped (lost) if memory (buffer) fills up

Alternative: Circuit switching

Operator
接线员

manually circuit switching

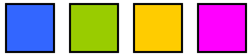
- Dedicated resources: no sharing
  - circuit-like (guaranteed) performance
- Circuit segment is idle if not used by call *(no sharing)*
- Commonly used in traditional telephone networks



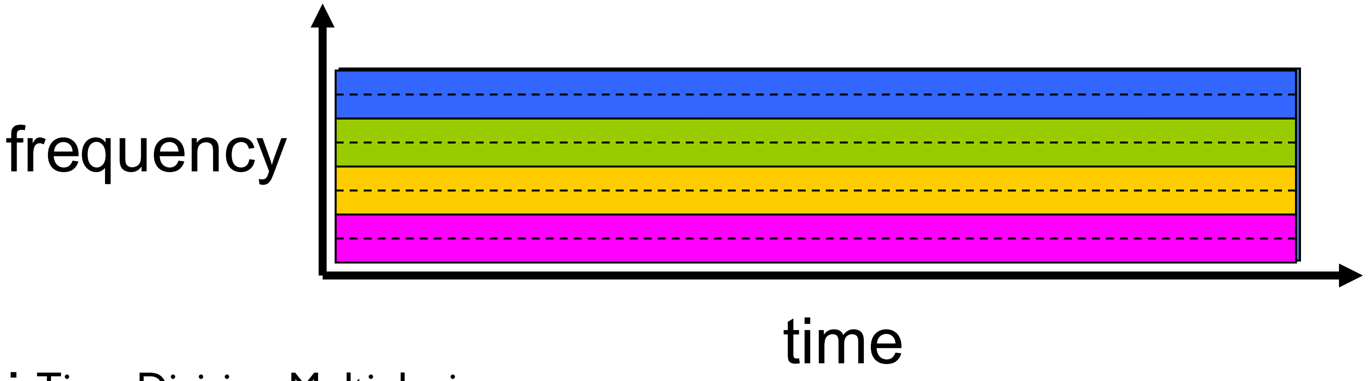
Circuit switching: FDM versus TDM

Example:

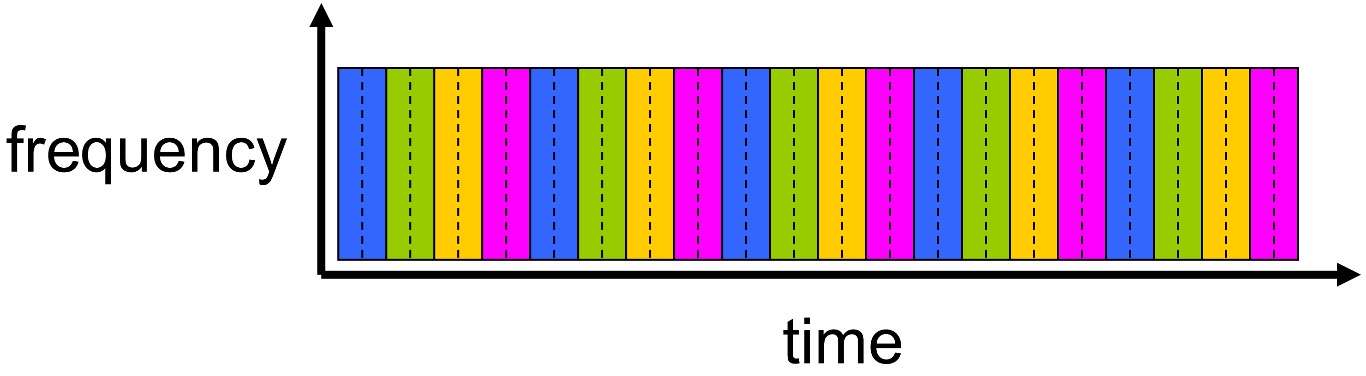
4 users



FDM: Frequency Division Multiplexing



TDM: Time Division Multiplexing

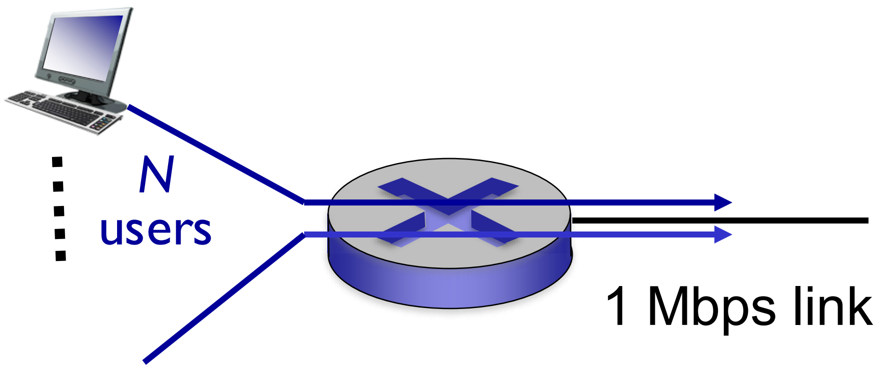


Packet switching VS circuit switching

Packet switching allows more users to use network!

Example:

- 1 Mb/s link
- each user:
  - 100 kb/s
  - aka, active 10% of time



- *Circuit-switching:*
  - 10 users
- *Packet switching:*
  - with 35 users, probability > 10 active at same time is less than 0.0004

$$\sum_{i=11}^{35} C_{35}^{i} p^{i} (1-p)^{35-i}$$

Q: how did we get value 0.0004?

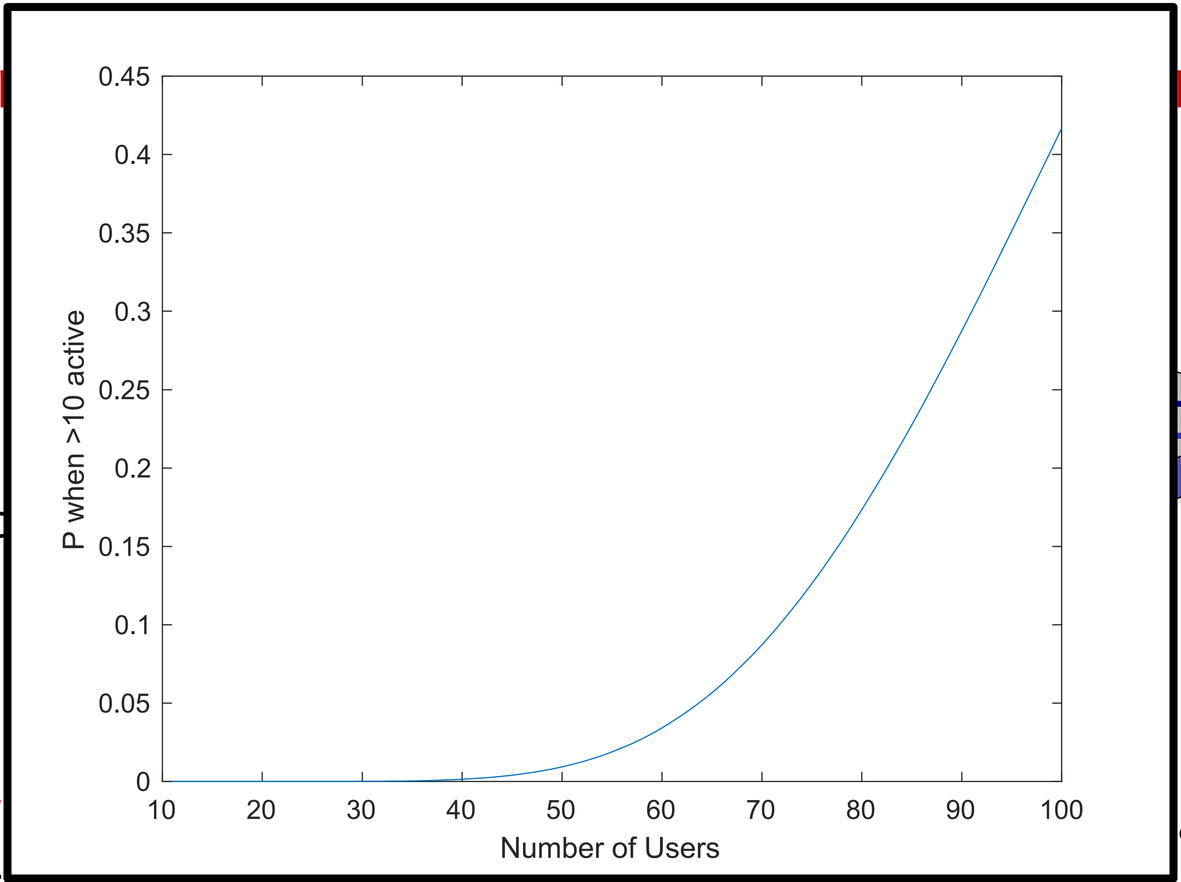
Q: what happens if > 35 users ?

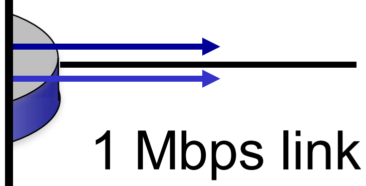
Packet switching VS circuit switching

Packet switchi… …rk!

Example:

- 1 Mb/s link
- each user:
  - 100 kb/s
  - aka, active 10% o…





$$\sum_{i=11}^{35} C_{35}^{\mathrm{i}} p^{i}(1-p)^{35-i}$$

- *Circuit-switching*
  - 10 users
- *Packet switching*
  - with 35 users, probability > 10 active at same time is less than 0.0004

…alue 0.0004?

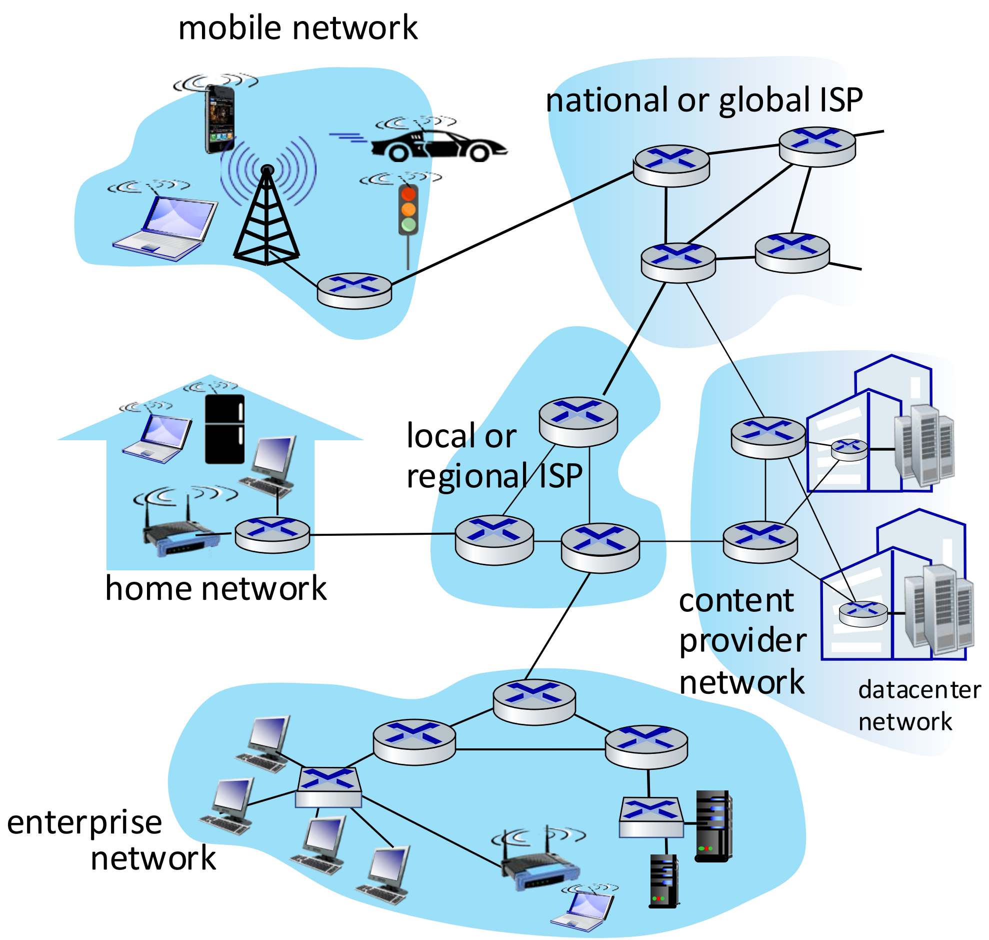
Q: what happens if > 35 users ?

Packet switching VS circuit switching

- **Is packet switching (PS) a "winner?"**
  - PS advantages:
    - resource sharing
    - simpler, no call setup
  - PS drawbacks:
    - excessive congestion possible: delay and loss
      - protocols needed for reliable data transfer, congestion control
  - How to provide circuit-like behavior PS?
    - Bandwidth guarantees

Internet structure: a "network of networks"

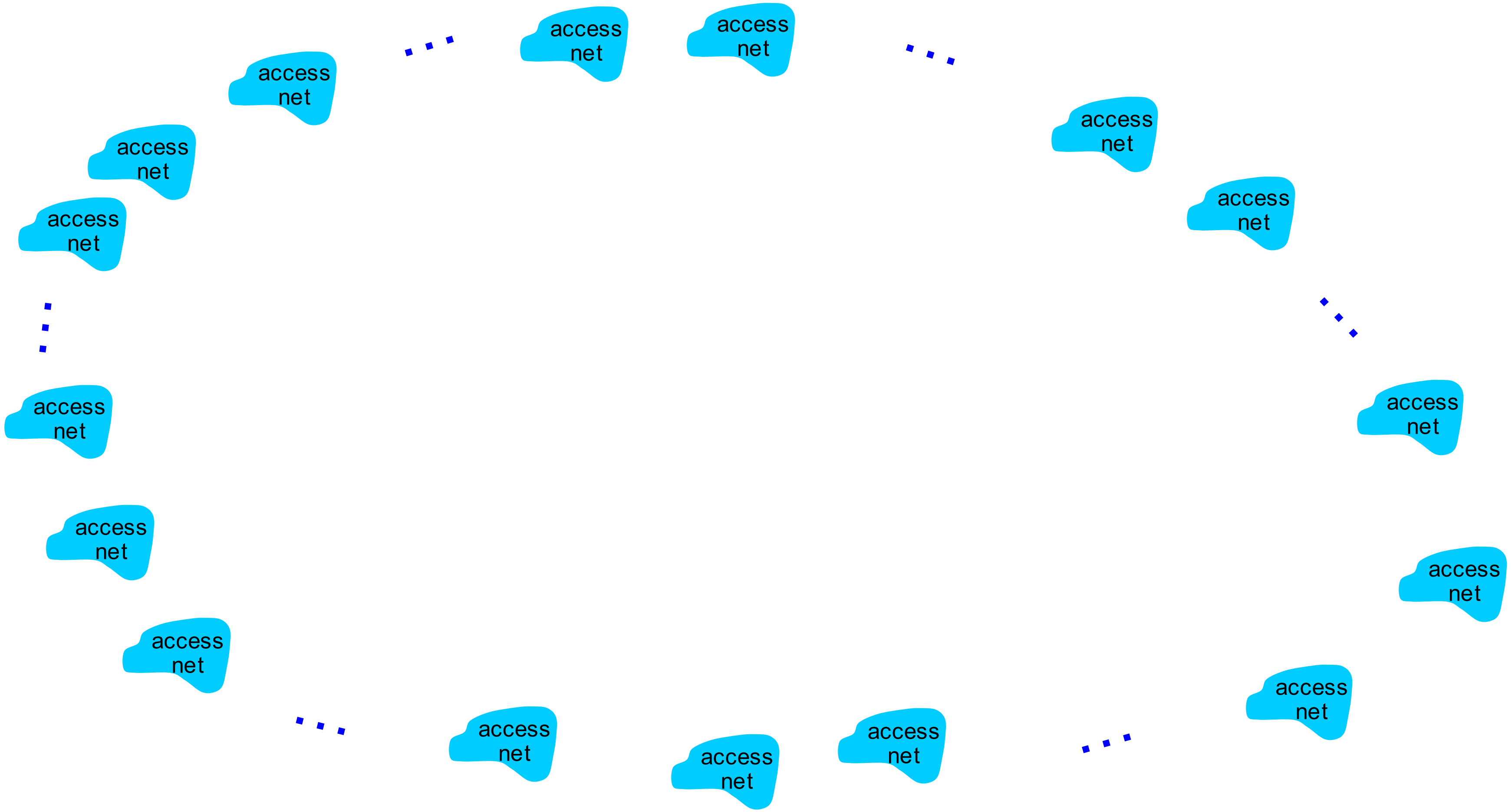
- Hosts connect to Internet via access Internet Service Providers (ISPs)
- Access ISPs in turn must be interconnected
    - So that *any* two hosts *(anywhere!)* can send packets to each other
- Resulting network of networks is very complex
    - Evolution driven by economics, national policies



Let's take a stepwise approach to describe current Internet structure

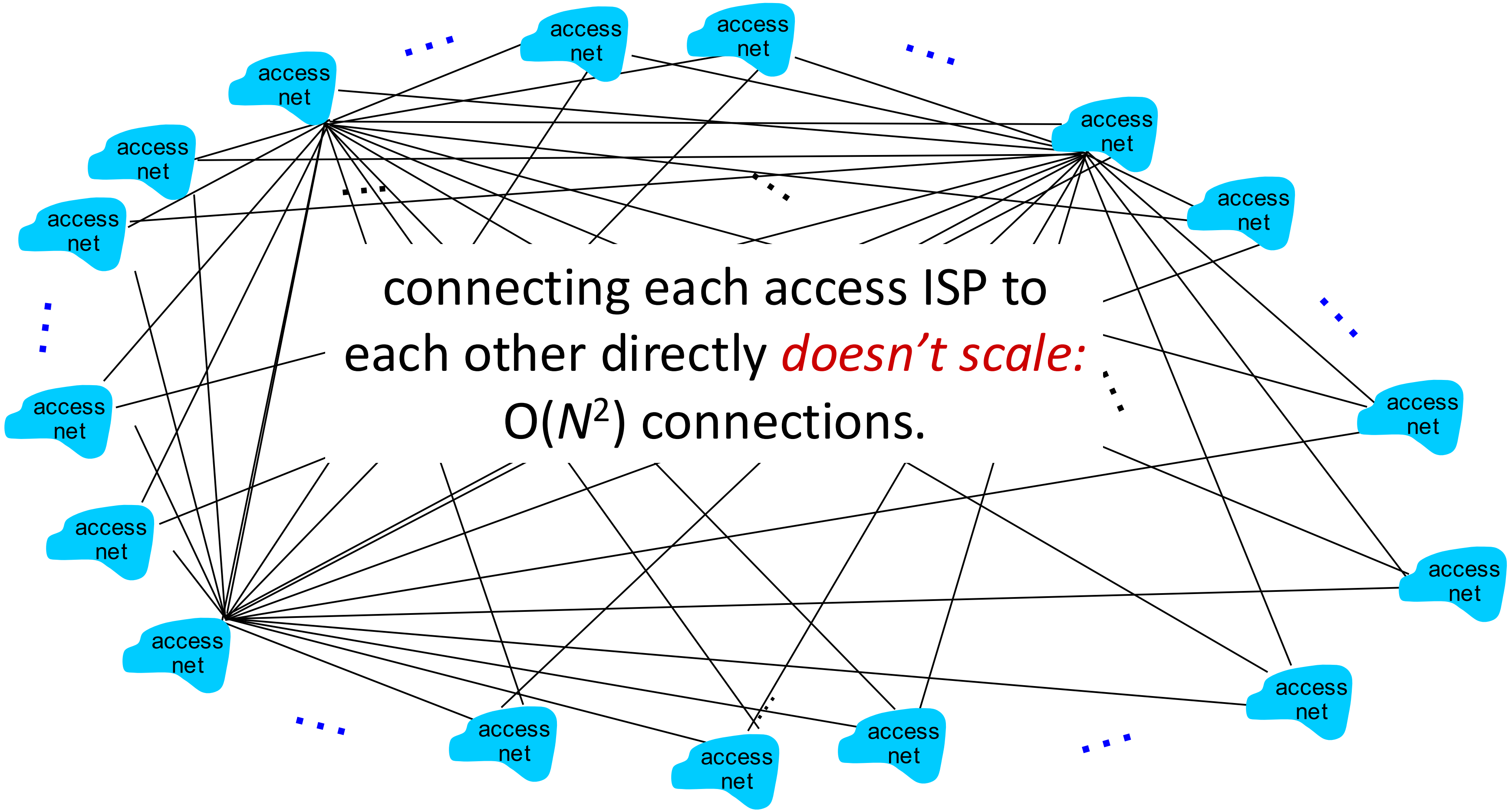
Internet structure: a "network of networks"

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



Internet structure: a "network of networks"

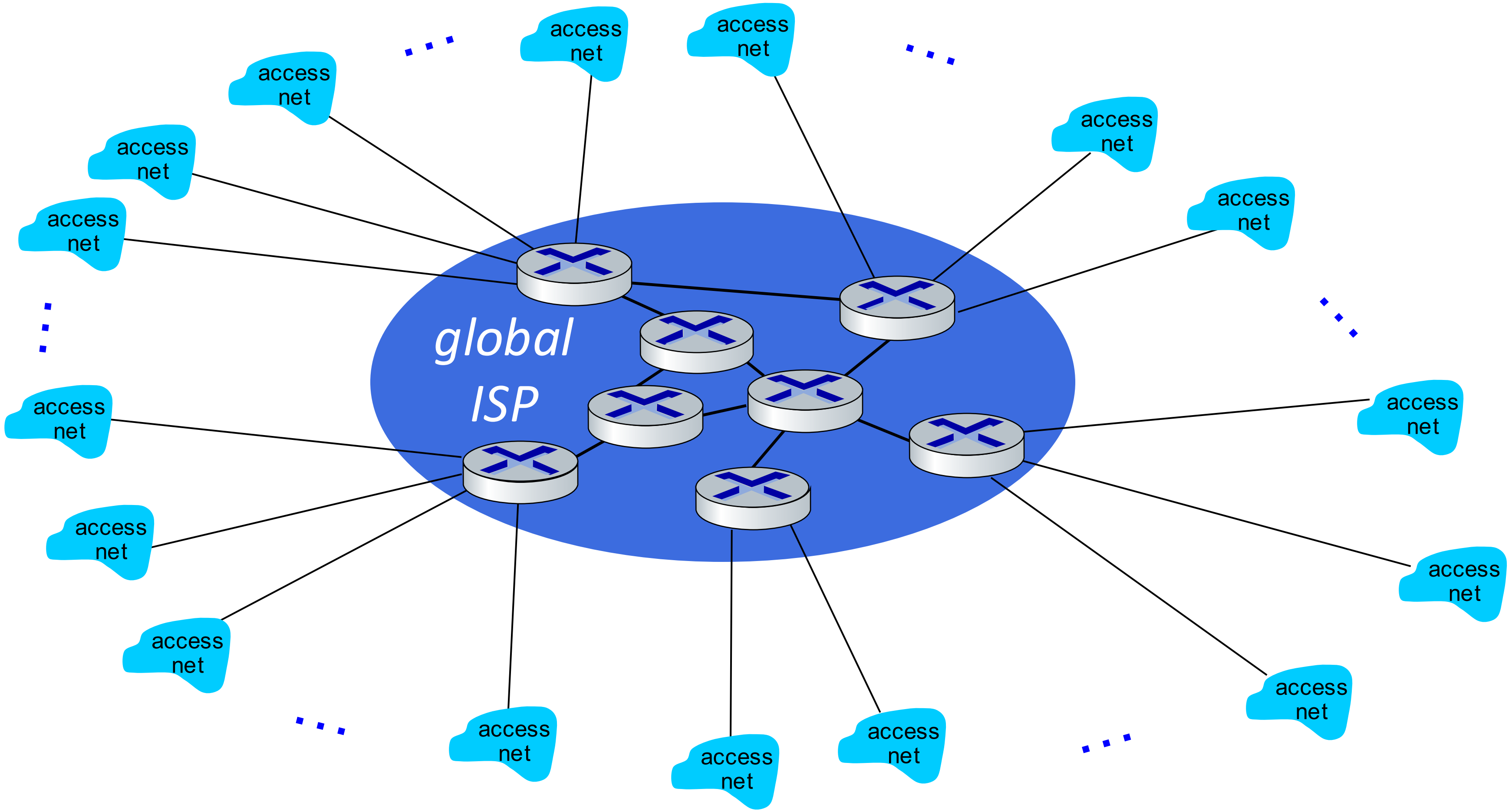
Question: given *millions* of access ISPs, how to connect them together?



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

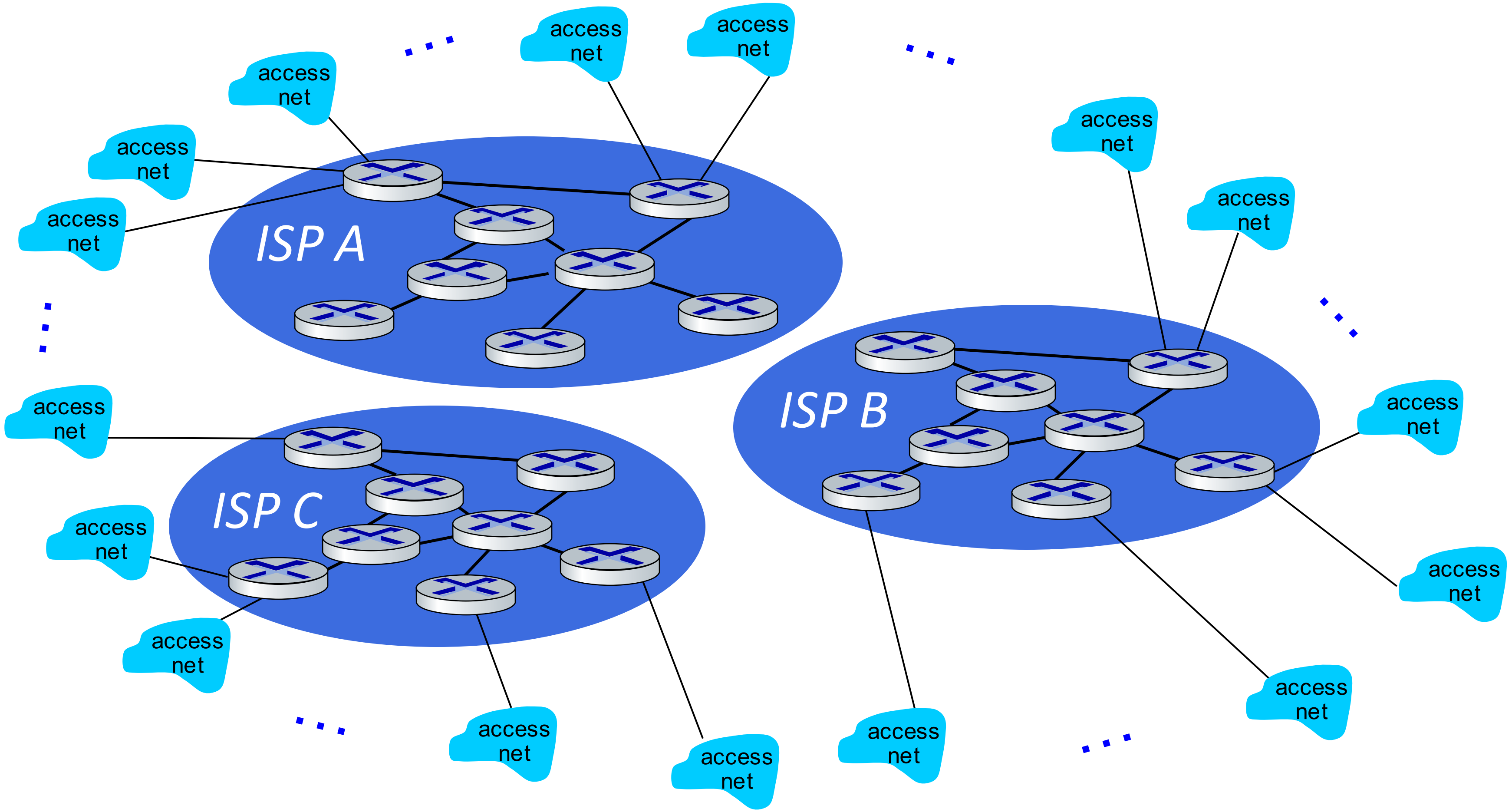
Internet structure: a "network of networks"

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



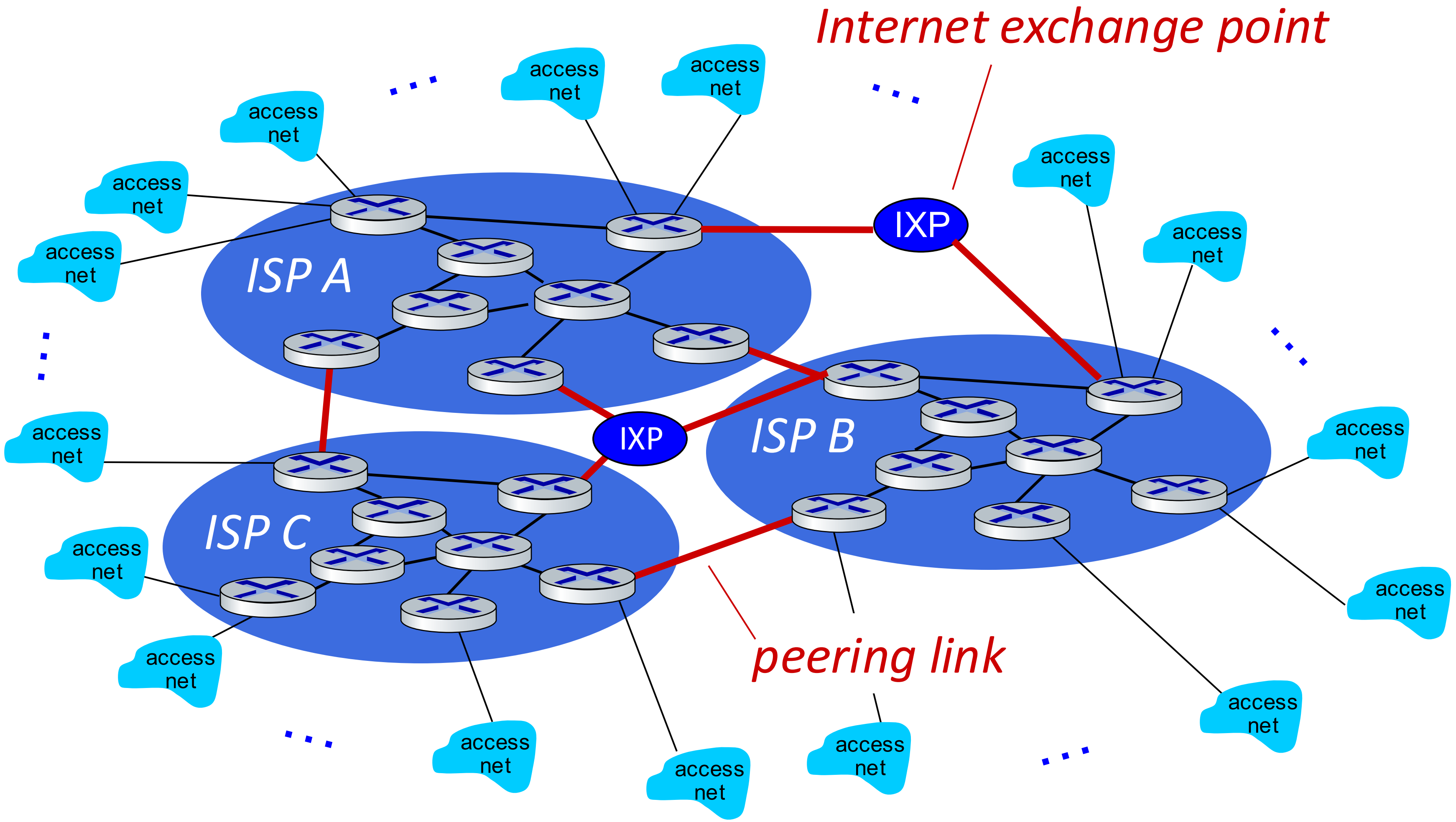
Internet structure: a “network of networks”

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



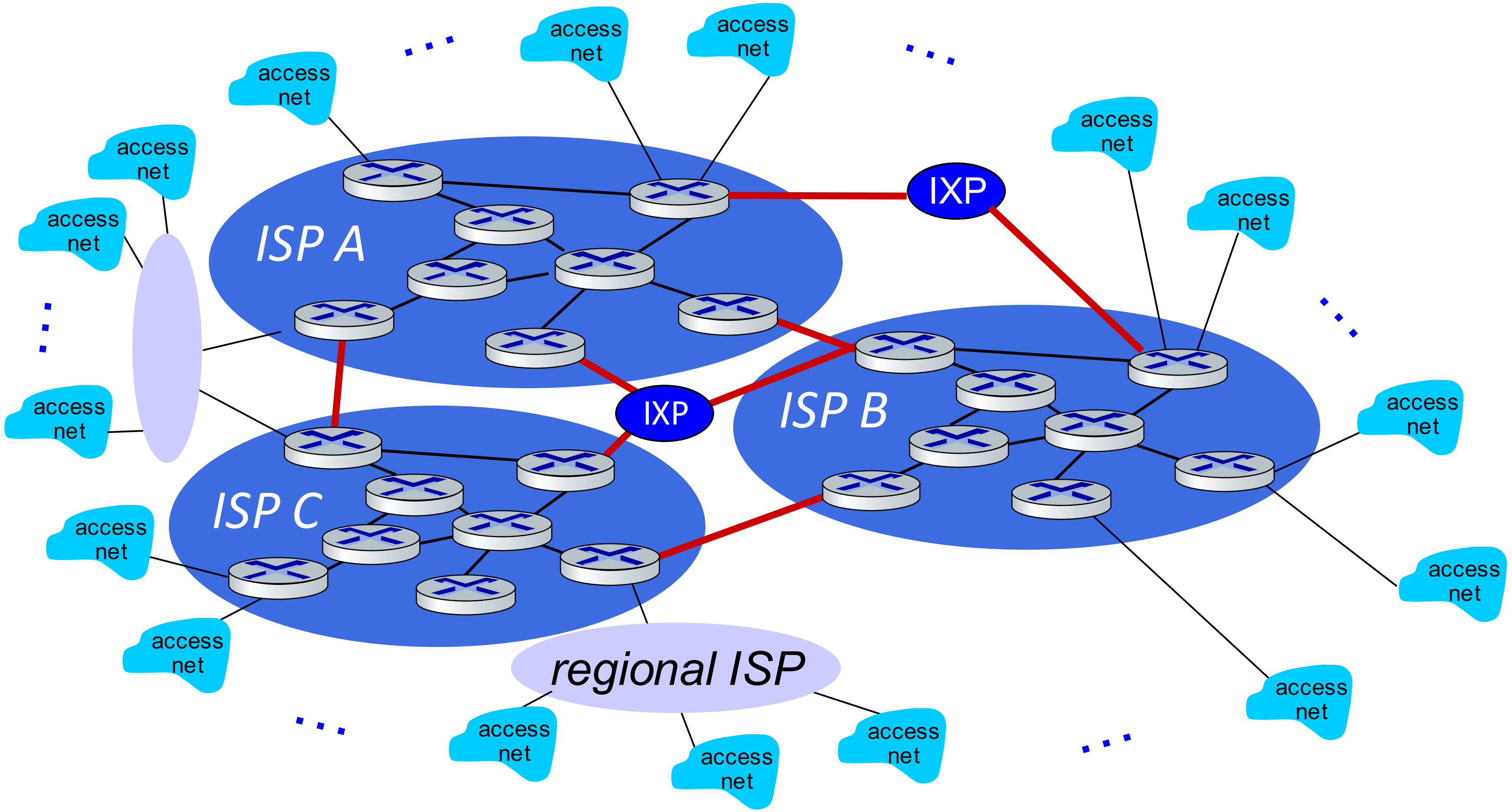
Internet structure: a "network of networks"

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



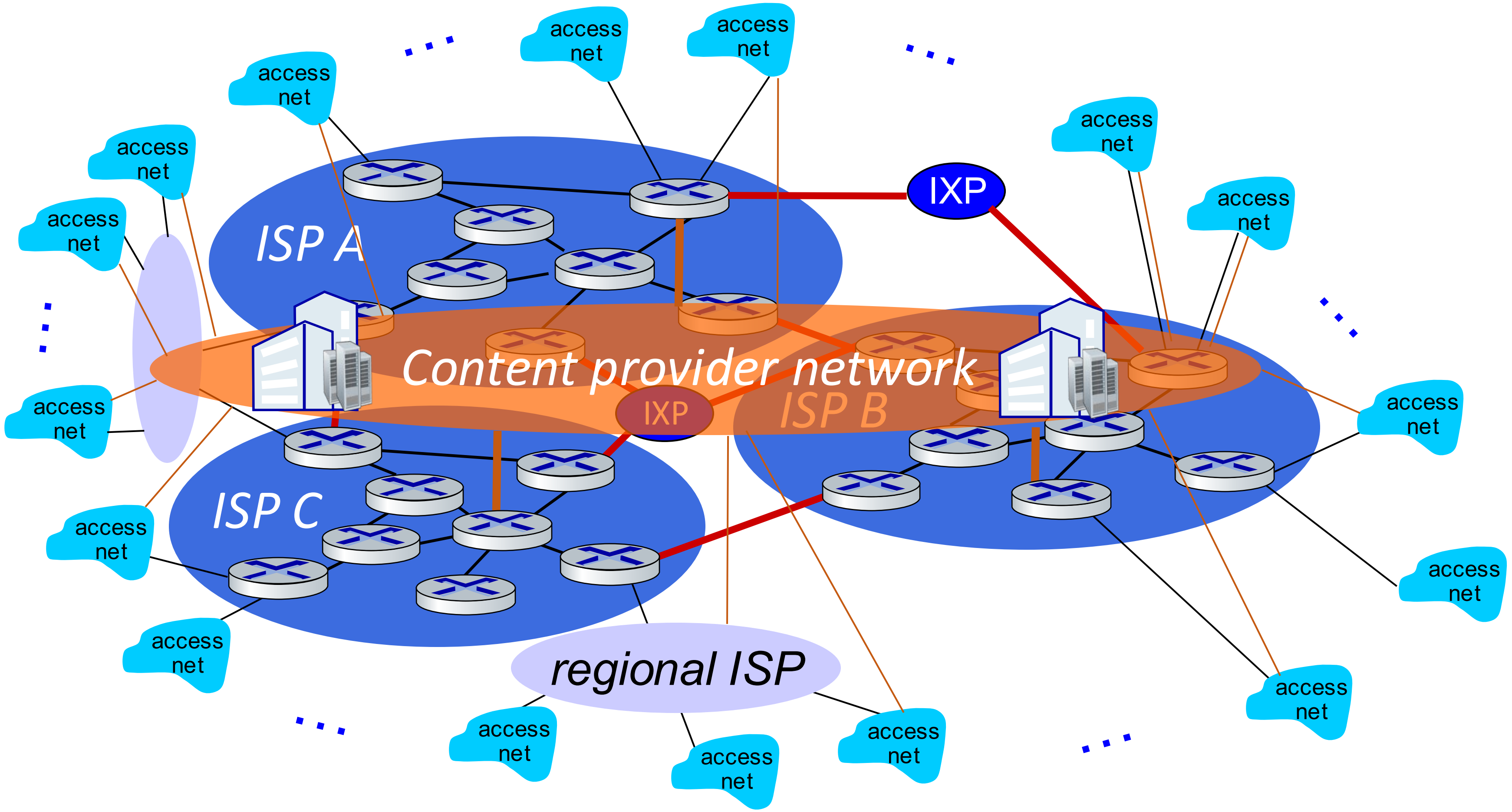
Internet structure: a "network of networks"

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

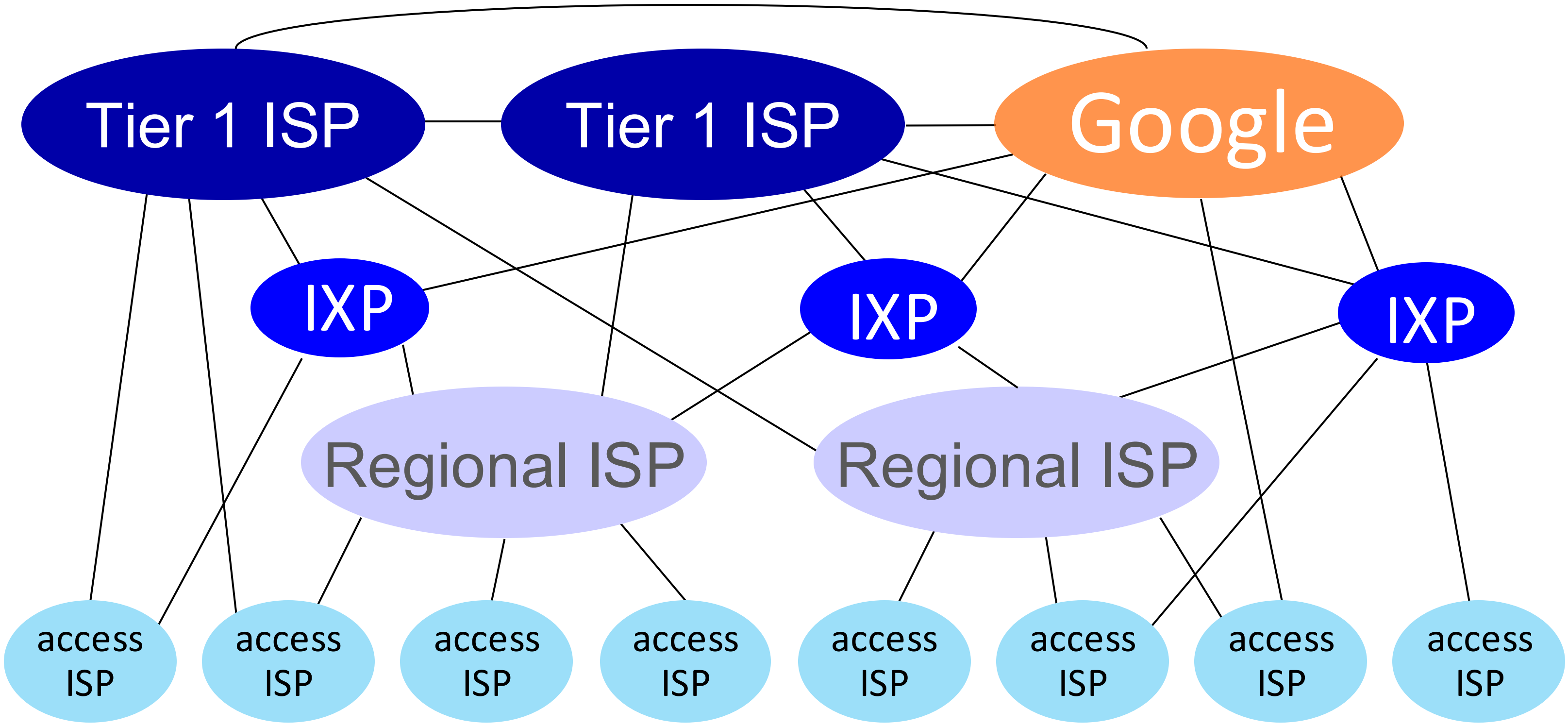


Internet structure: a "network of networks"

… 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"



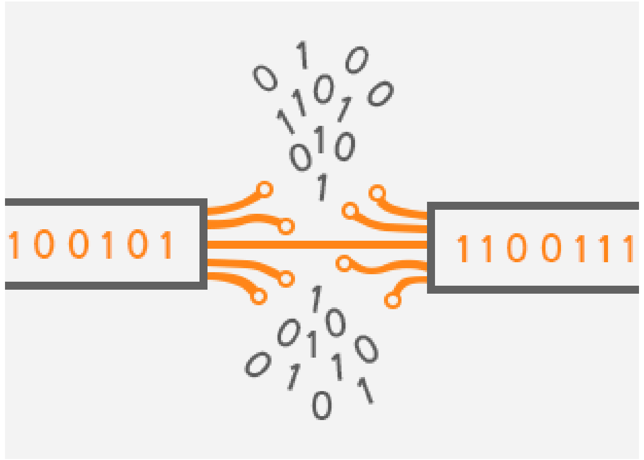
At "center": small # of well-connected large networks

- "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

Lecture 1 - Introduction

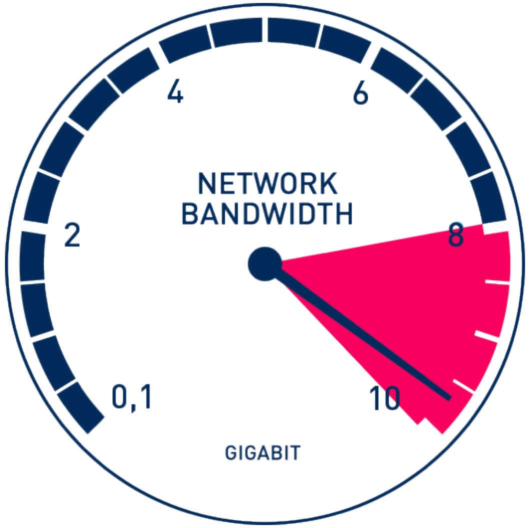
1. What is the network/internet?
2. How does the network work?
   - Network edge
   - Network core
3. How to evaluate the performance?
   - loss, delay, throughput
4. Service models
   - TCP/IP

Performance



Package Loss

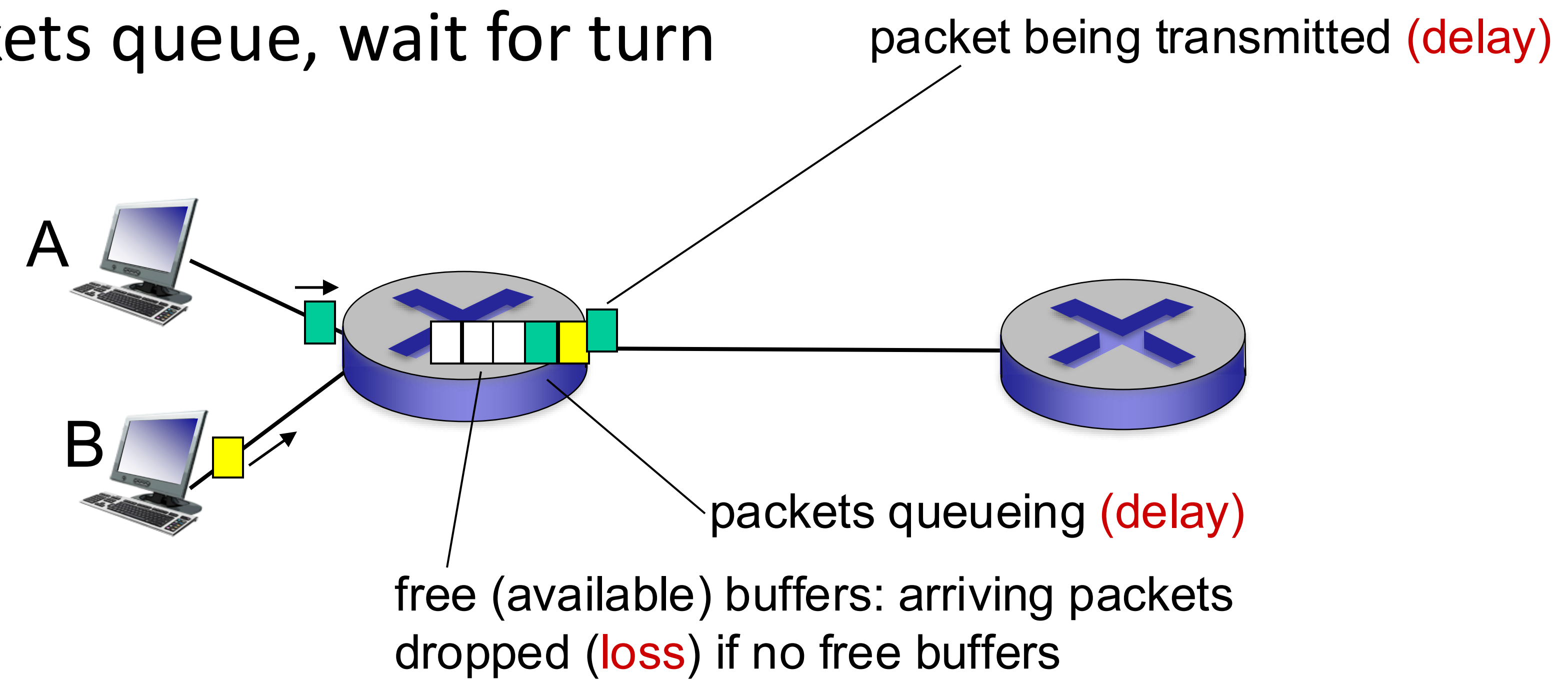
Delay



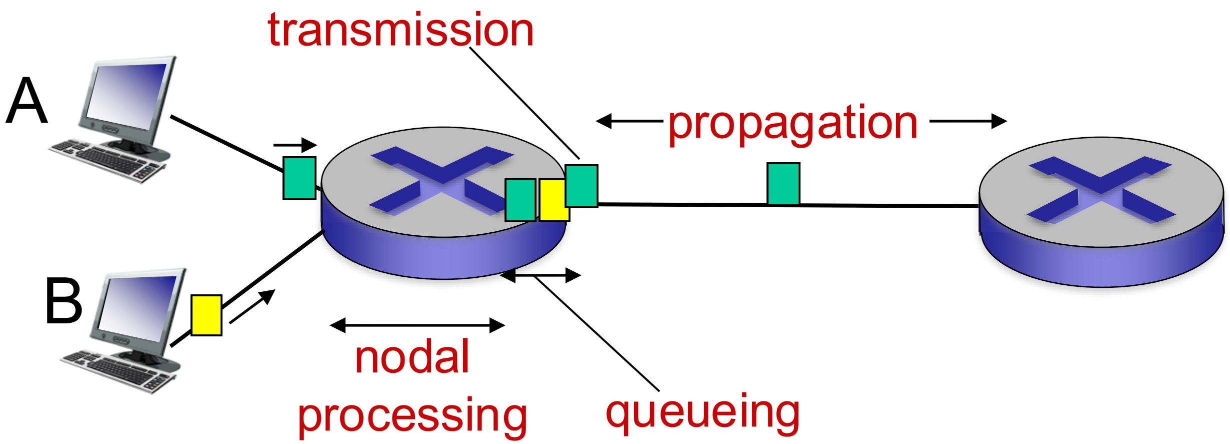
Bandwidth

How do loss and delay occur?

- **Packages queue in router buffers**
  - packet arrival rate to link (temporarily) exceeds output link capacity
  - then, packets queue, wait for turn



Four sources of packet delay

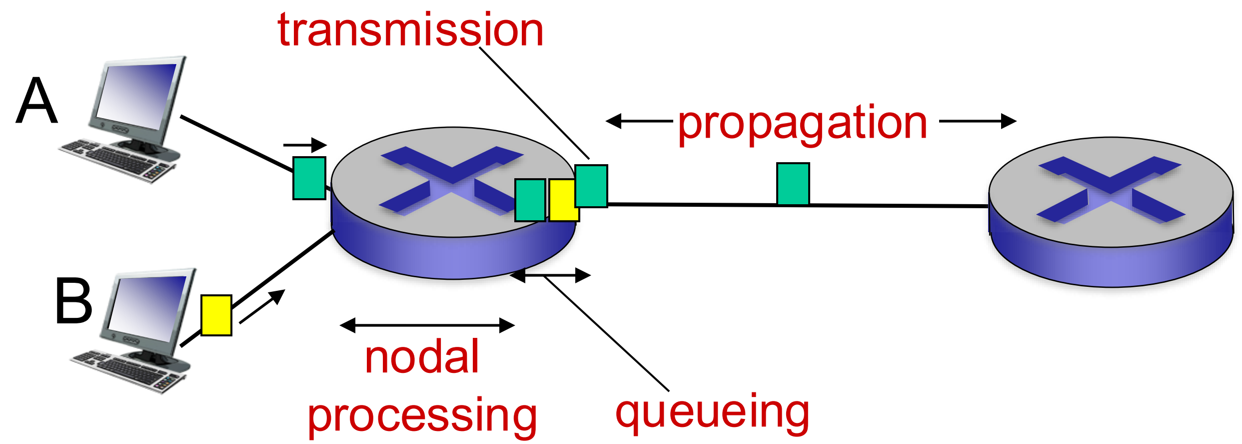


$$d_{nodal} = d_{proc} + d_{queue} + d_{trans} + d_{prop}$$

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

Four sources of packet delay



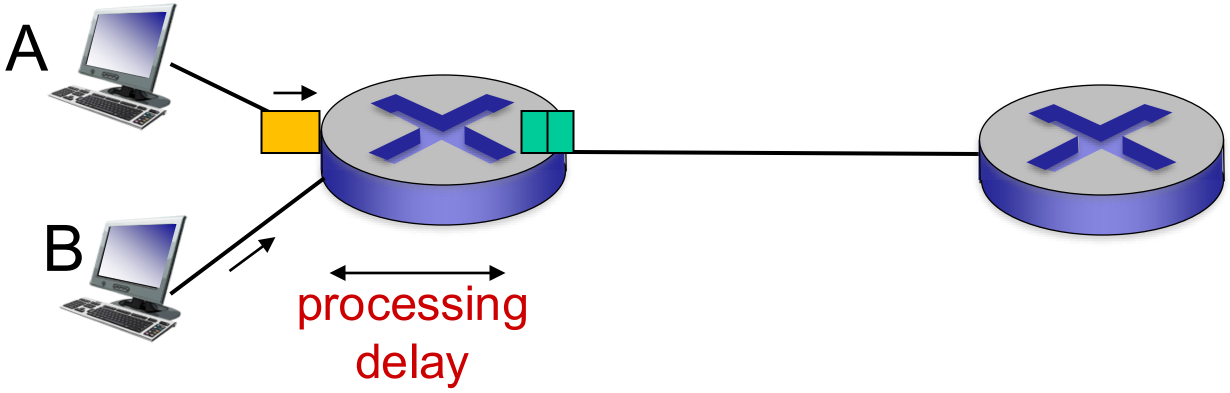
$$d_{nodal} = d_{proc} + d_{queue} + d_{trans} + d_{prop}$$

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

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

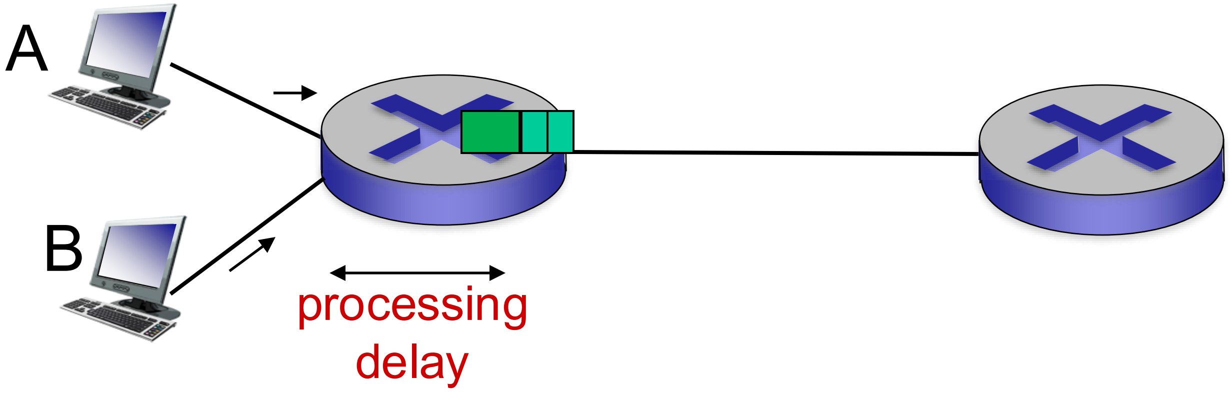
d_{prop}: propagation delay:
- d: length of physical link
- s: propagation speed (~2.9×10^8 m/sec)
- $d_{prop} = d/s$

Four sources of packet delay



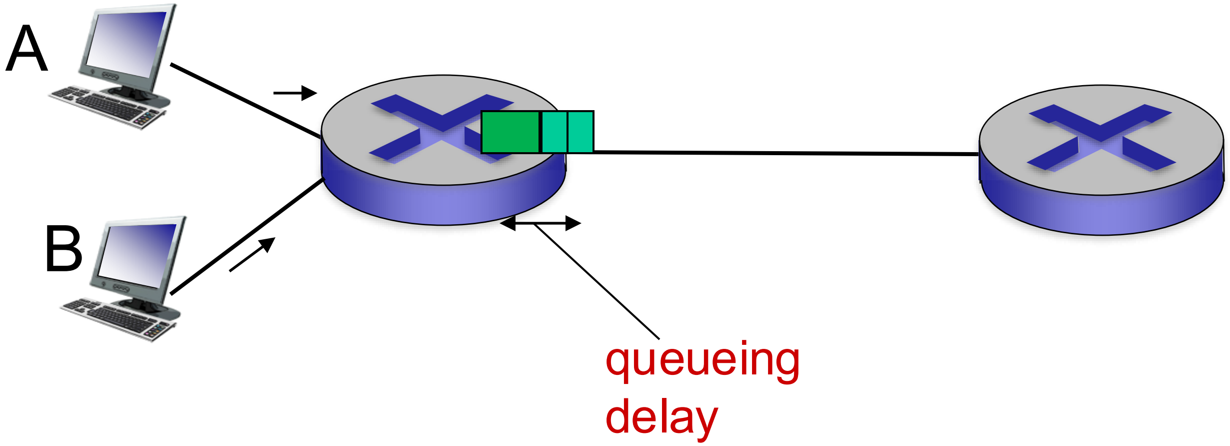
$$d_{nodal} = d_{proc} + d_{queue} + d_{trans} + d_{prop}$$

Four sources of packet delay



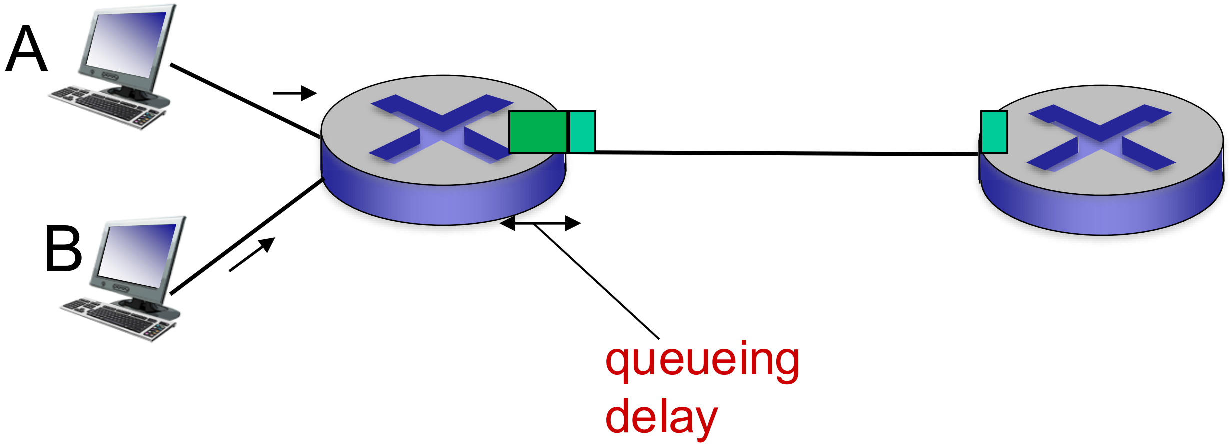
$$d_{nodal} = d_{proc} + d_{queue} + d_{trans} + d_{prop}$$

Four sources of packet delay



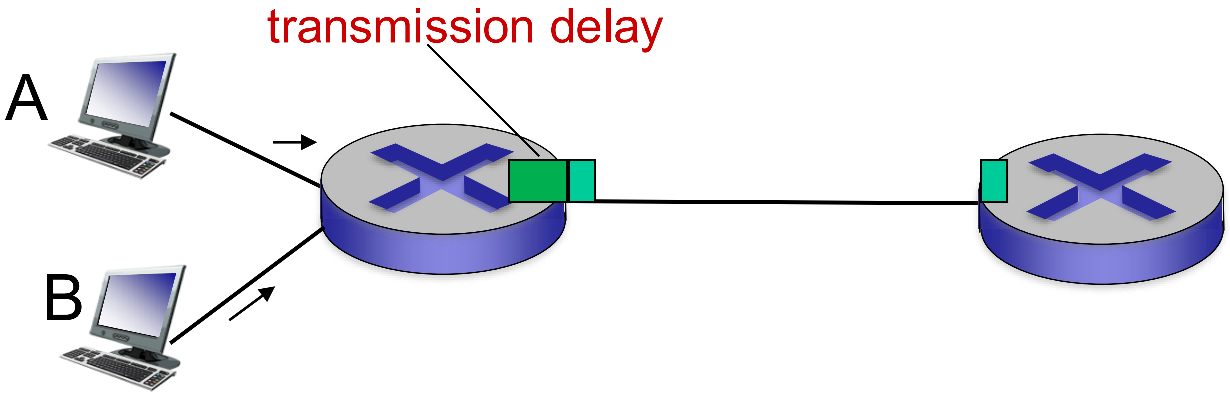
$$\boxed{d_{nodal} = d_{proc} + d_{queue} + d_{trans} + d_{prop}}$$

Four sources of packet delay



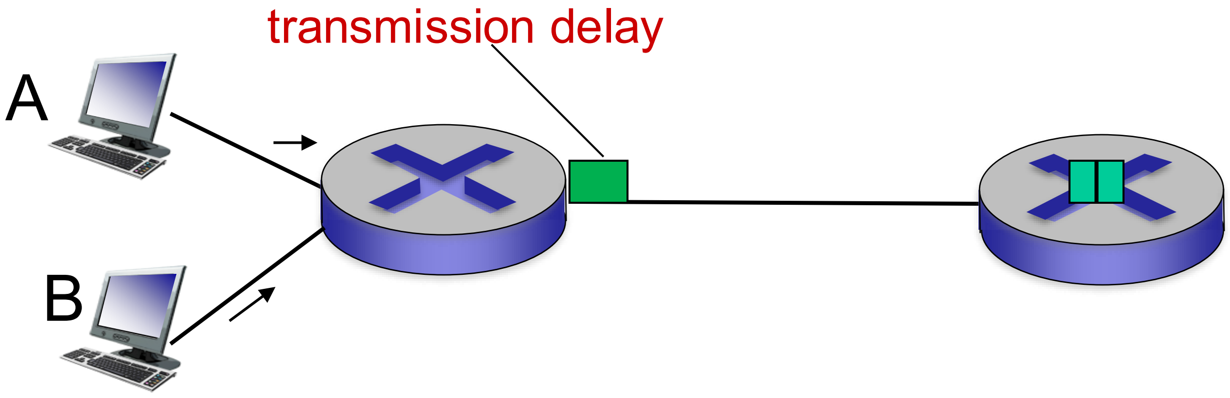
$$d_{nodal} = d_{proc} + d_{queue} + d_{trans} + d_{prop}$$

Four sources of packet delay



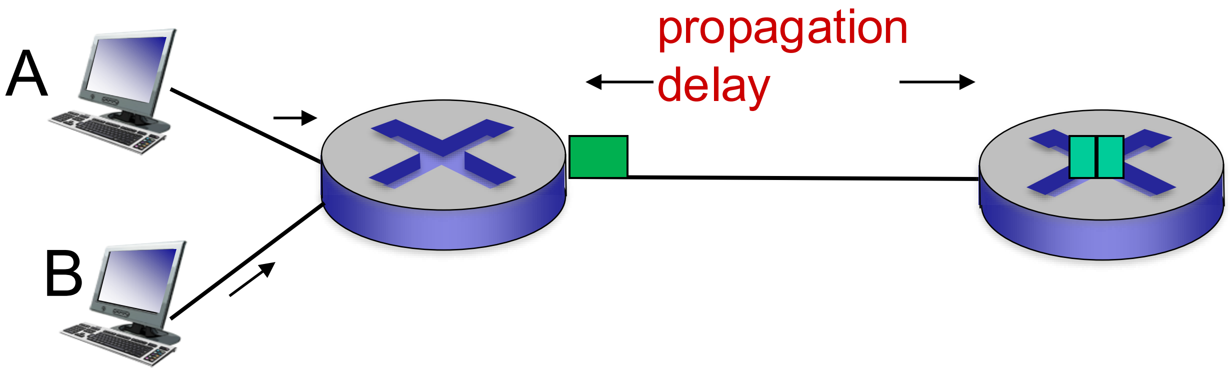
$$d_{nodal} = d_{proc} + d_{queue} + d_{trans} + d_{prop}$$

Four sources of packet delay



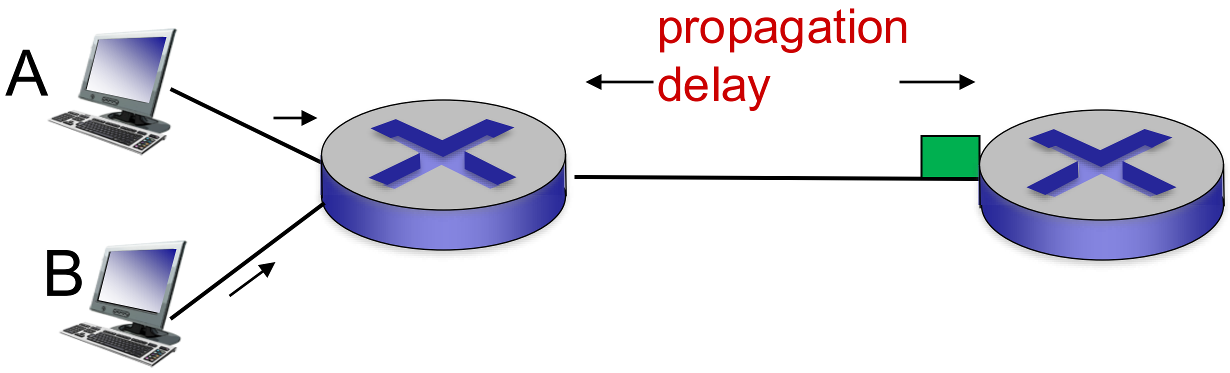
$$d_{nodal} = d_{proc} + d_{queue} + d_{trans} + d_{prop}$$

Four sources of packet delay



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

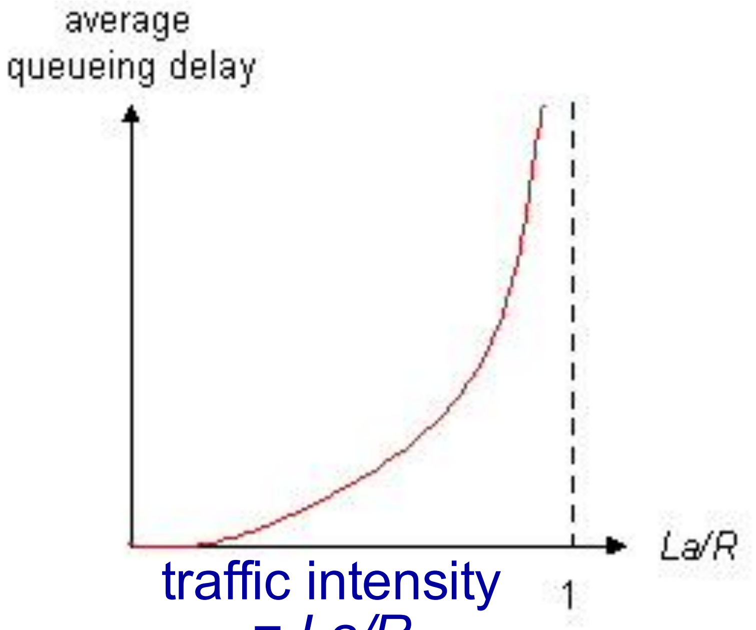
Four sources of packet delay



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

Queueing delay

- *R:* link bandwidth (bps)
- *L:* packet length (bits)
- a: average packet arrival rate



traffic intensity
= *La/R*

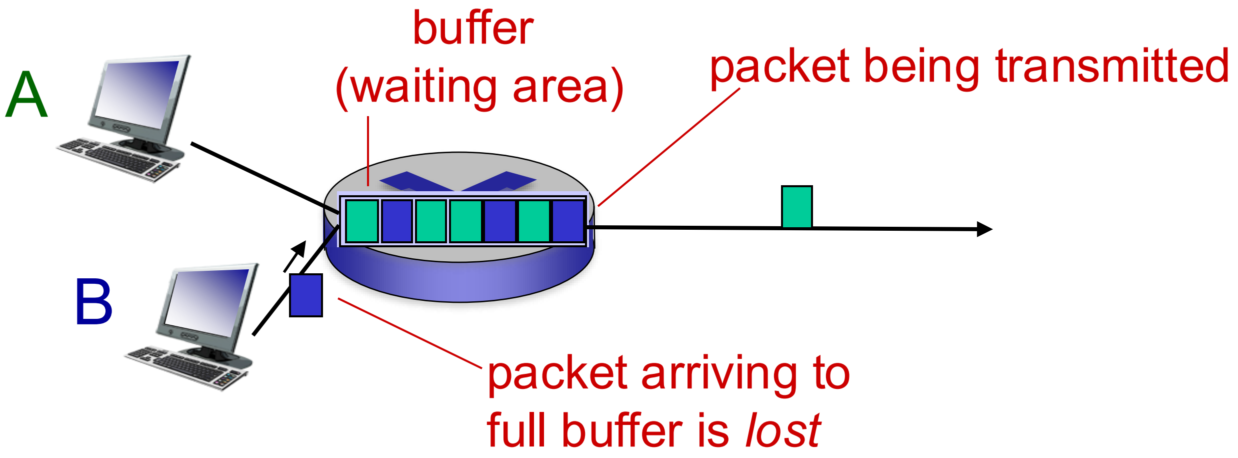
- *La/R* ~ 0: avg. queueing delay small
- *La/R* -> 1: avg. queueing delay large
- *La/R* > 1: more "work" arriving than can be serviced, average delay infinite!

La/R ~ 0

La/R -> 1

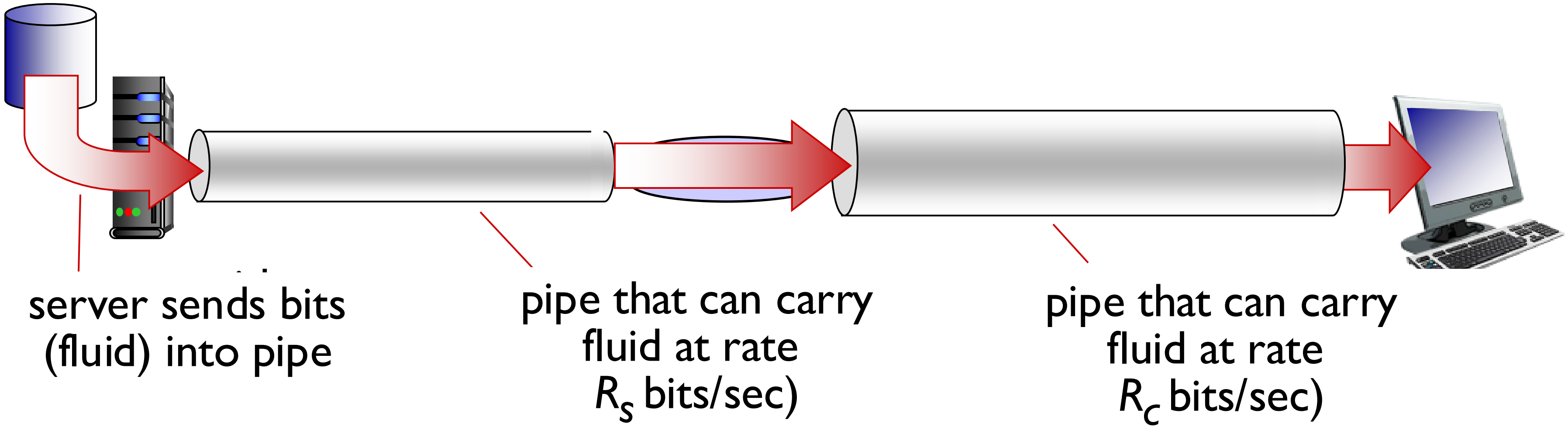
Packet loss

- 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



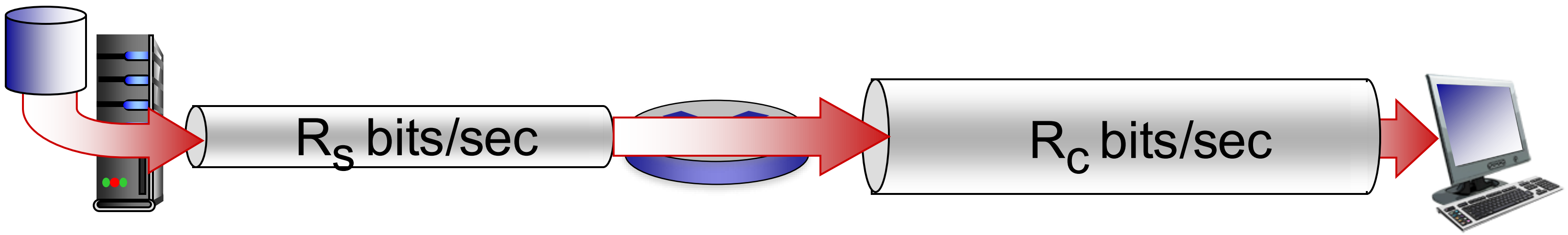
Throughput

- ***Throughput:*** **rate (bits/time unit) at which bits transferred between sender/receiver**
  - *instantaneous:* rate at given point in time
  - *average:* rate over longer period of time

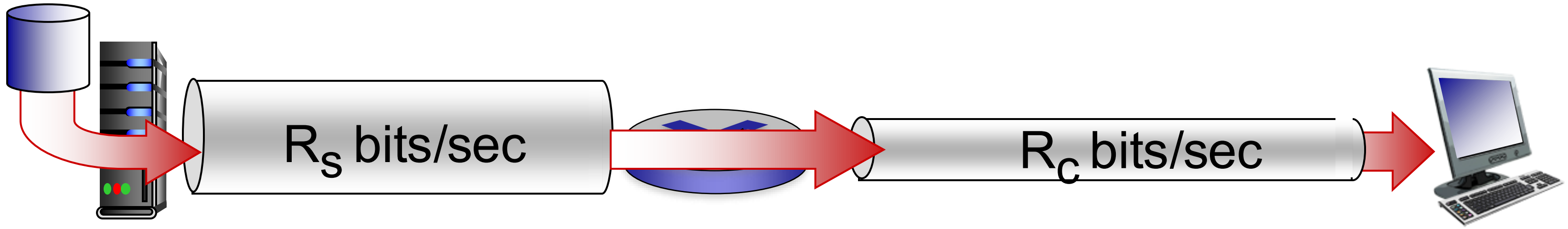


Throughput

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



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

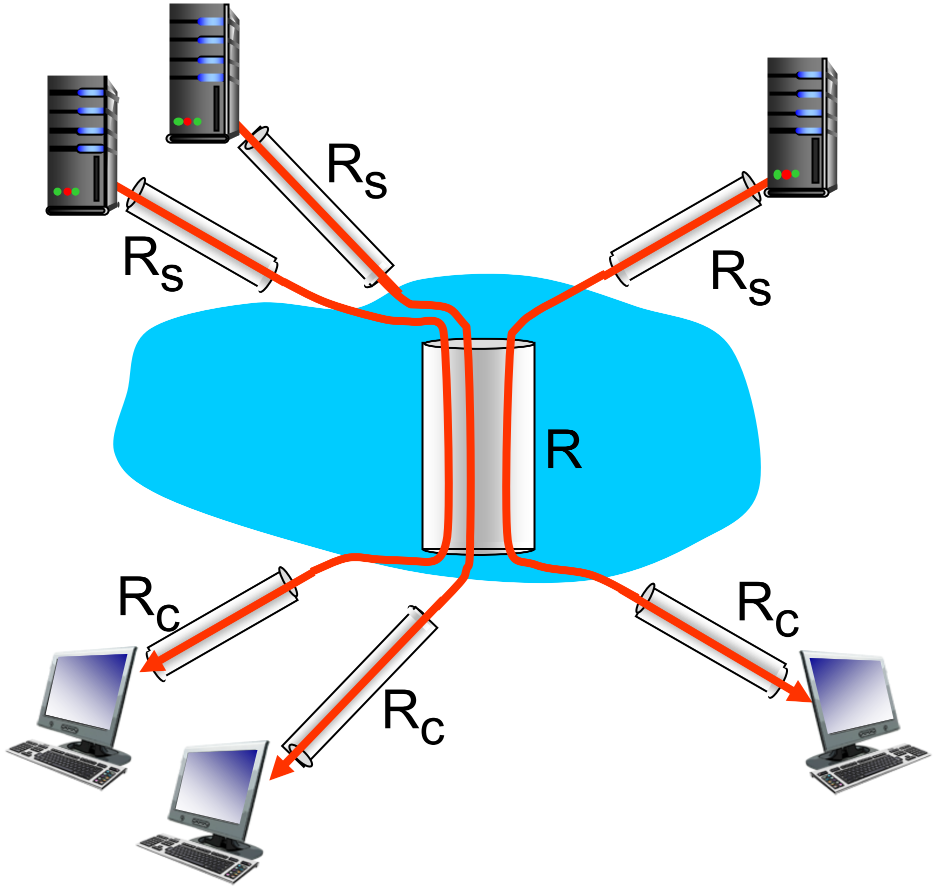


bottleneck link

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

Throughput: Internet scenario

- Per-connection end-end throughput: $min\{R_c, R_s, R/10\}$
- In practice: R_c or R_s is often bottleneck



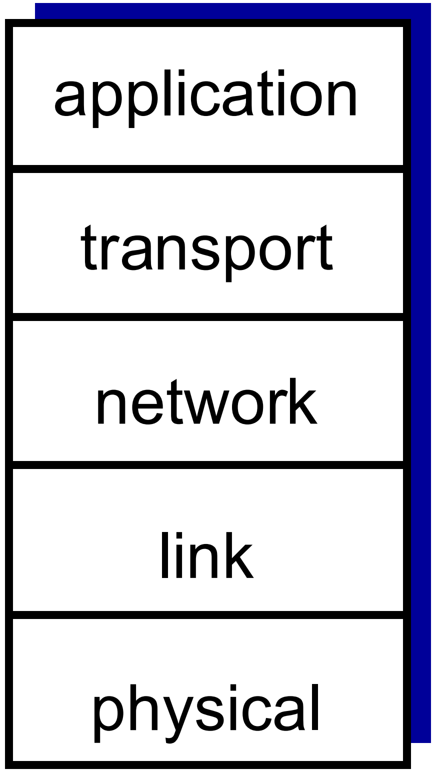
10 connections (fairly) share backbone bottleneck link *R* bits/sec

Lecture 1 - Introduction

1. What is the network/internet?
2. How does the network work?
   - Network edge
   - Network core
3. How to evaluate the performance?
   - loss, delay, throughput
4. **Service models**
   - TCP/IP

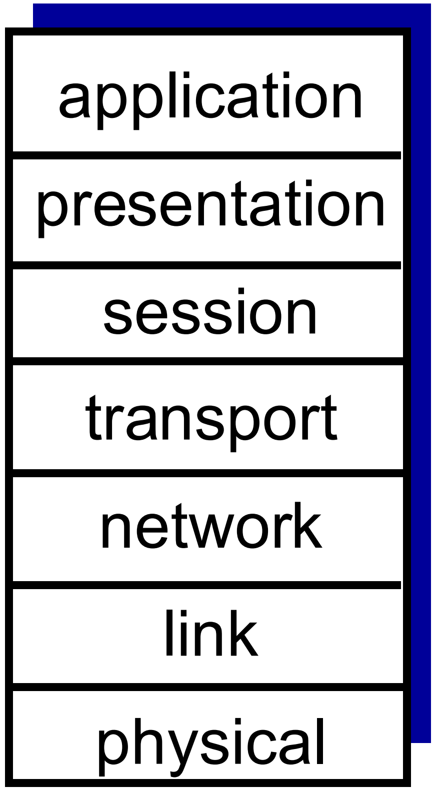
Internet protocol stack

- ***application:*** **supporting network applications**
  - FTP, SMTP, HTTP, 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"**

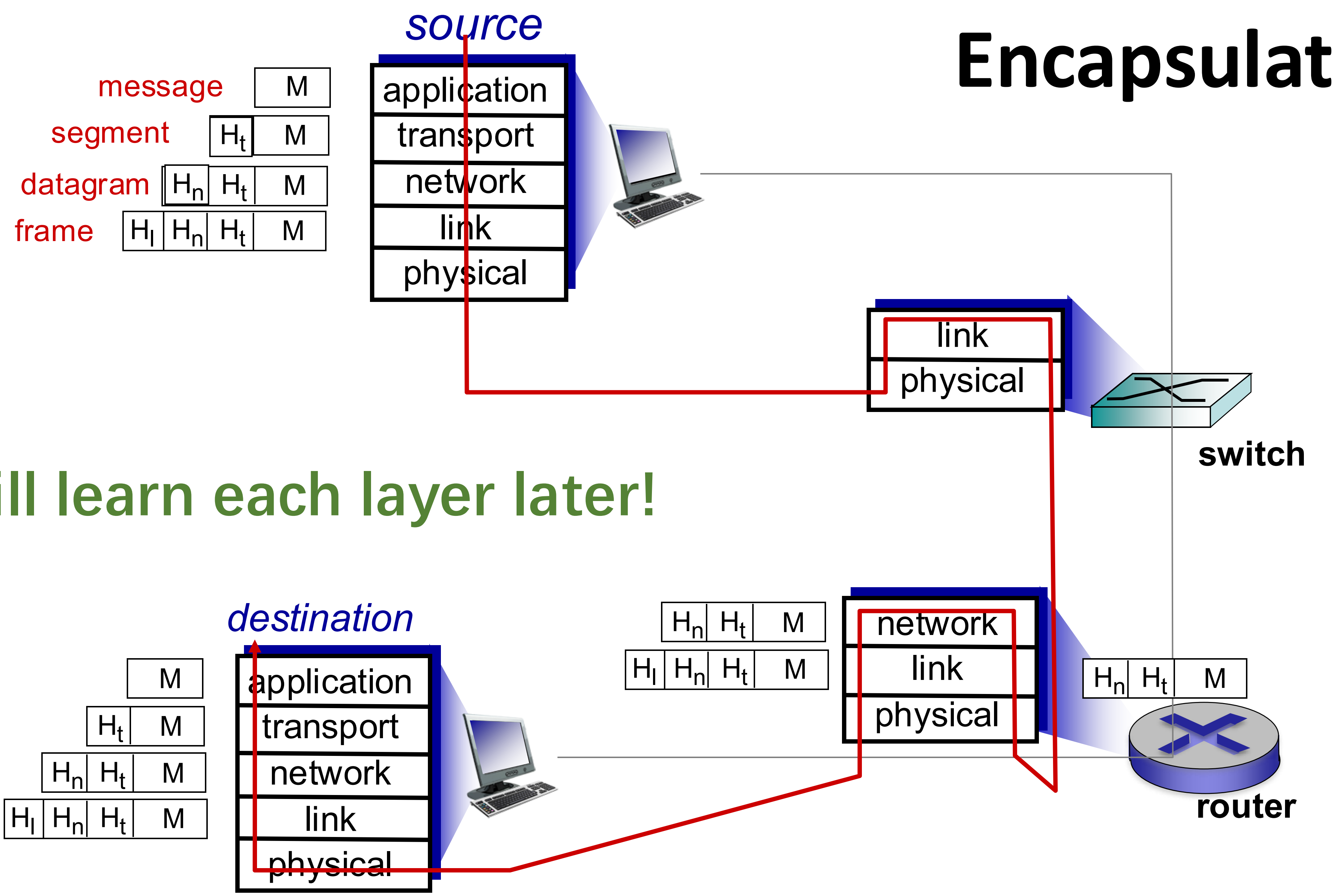


ISO/OSI reference model

- **ISO/OSI = ISO/**Open System Interconnection
- ***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



Encapsulation



We will learn each layer later!

Why layering?

- Divide complex systems to simple components
- Easy for maintenance
- Flexible for updating

- **Thanks**