计算机网络原理与编程

主讲：张召

主讲老师简介

- **张召** 长期从事区块链系统和分布式数据库方面的教学和研究。
- 邮件：zhzhang@dase.ecnu.edu.cn
- 电话、微信：18502118359
- 课程钉钉群，作业提交，讲义分发等

主要学习内容

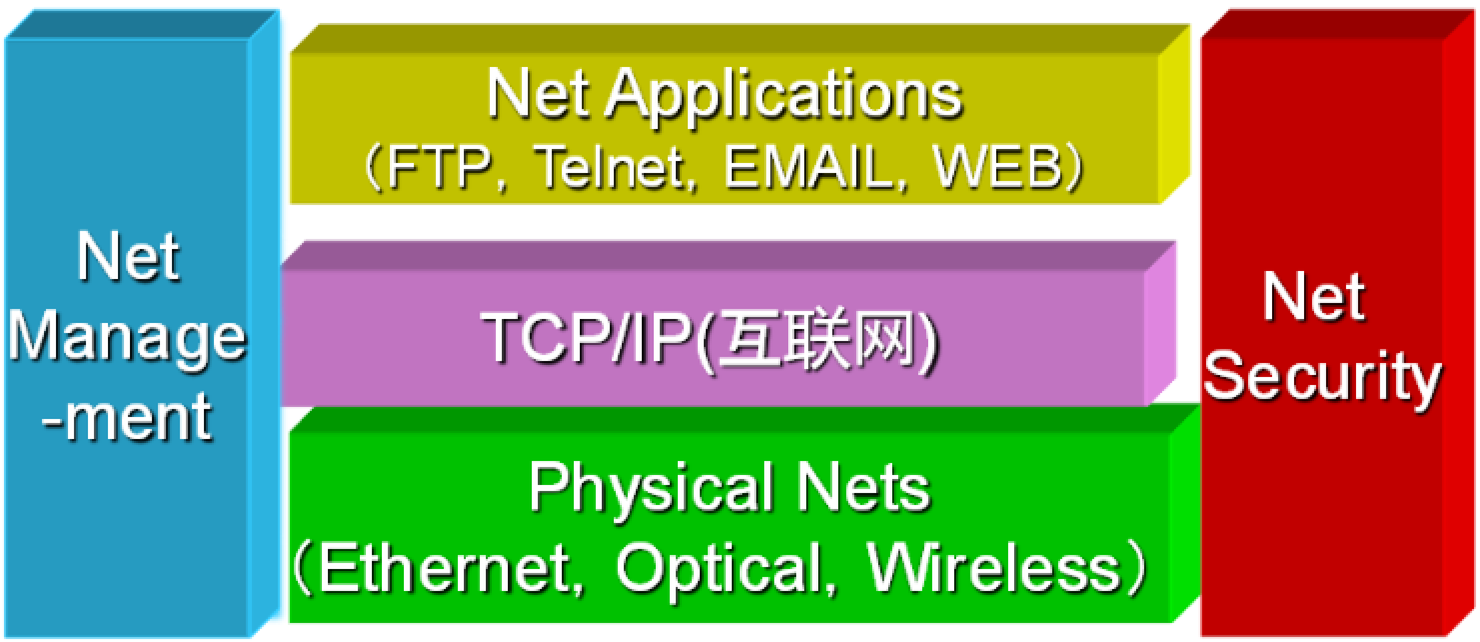
- 面向Internet网络，学习其工作原理和运行机制。
  协议，HTTP，DNS，TCP/UDP，.....
  WireShark协议分析软件
- 网络编程
  以JAVA语言为依托，学习网络编程
  为后续的分布式系统开发打基础

学习方式

- 理论学习结合上机实践
- 上机实践：网络协议分析（Wireshark）和编程（Java网络编程）并重

什么是计算机网络

- 利用通信信道，将地理上分布的多个计算机系统连接起来，在相应网络协议的支持下，实现信息传送、资源共享和网络计算功能的系统。



计算机网络的功能

- 信息传递

  传送文本、图像、视频、声音

- 共享资源

  信息资源、计算资源、存储资源

- 网络计算

分布在不同地点的多台计算机共同完成一定的任务

课程目标

- 我们专业和网络工程，通讯工程等专业学习网络的切入点有什么不同？
- 学习支撑互联网的核心关键技术及其工作原理
- 为将来从事分布式系统相关的学习和工作打下基础。

教材和参考资料

教材：

- James F. Kurose, Keith W. Ross(陈鸣 译)，计算机网络自顶向下方法（原书第7版），机械工业出版社，2018。

参考书：

- Andrew S. Tanenbaum（严伟、潘爱民译），计算机网络（第五版），清华大学出版社，2012

思考问题

- 如果让你来为某个学校建设一个校园网，你需要做哪些事情，包含几个步骤？

- 什么是协议（protocol）？

  通信计算机双方必须共同遵从的一组约定。如怎么样建立连接、怎么样互相识别等。只有遵守这个约定，计算机之间才能相互通信交流。

课程主要内容

- 第一章 计算机网络和因特网

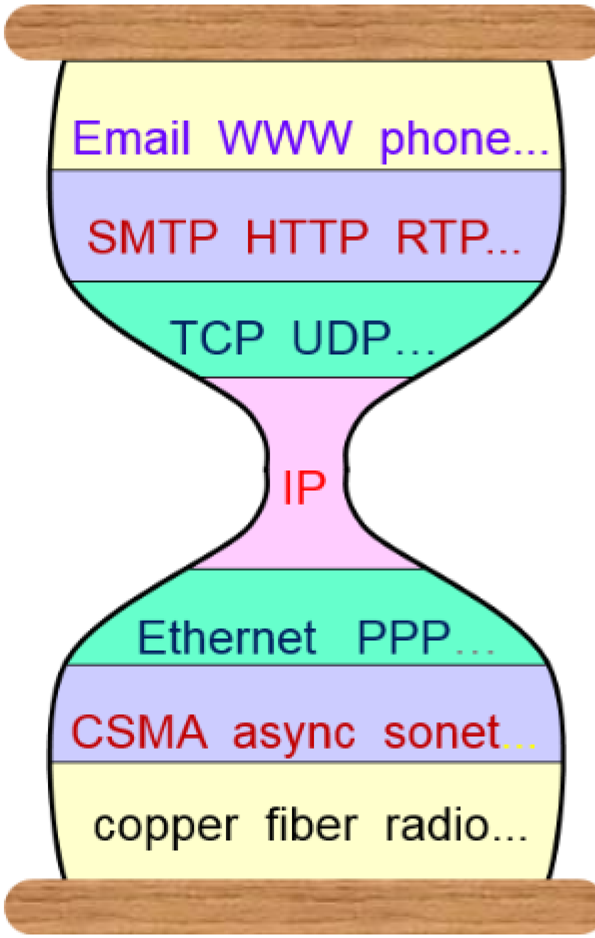
  网络核心、边缘、协议的体系结构

- 第二章 应用层

  Http、电子邮件（SMTP，POP3）、DNS等协议

- 第三章 运输层

  主机上的进程之间提供了可靠、有效的报文传送服务（TCP，UDP）



课程主要内容

- 第四章 网络层：数据平面

  路由器工作原理、IPV4、IPV6编址

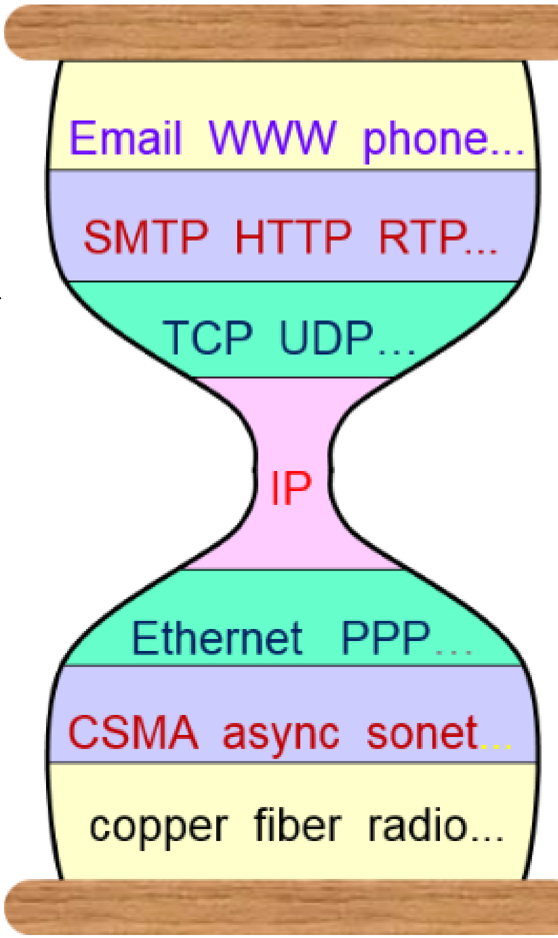
- 第五章 网络层：控制平面

  路由选择算法，SDN控制平面

- 第六章 链路层和局域网

  差错检测和纠正、多访问链路

IP层



高级专题

- 第七章 无线网络和移动网络
- 第八章   计算机网络中的安全
- 第九章 多媒体网络

对同学们的要求

- 认真上课
- 勤于思考
- 勇于实践
- 乐于交流

考核方式

- 平时成绩40%~50%

  课堂表现、作业和上机实践
- 期末闭卷考试 60%~50%

Chapter 1: introduction

our goal:

- get "feel" and terminology（术语）
- more depth, detail *later* in course
- approach:
  - use Internet as example

overview:

- what's the Internet?
- what's a protocol（协议）?
- network edge（边缘）; hosts（主机）, access net, physical media
- network core: packet/circuit（分组/电路） switching, Internet structure
- performance: loss（丢包）, delay（时延）, throughput（吞吐）
- security
- protocol layers, service models
- history

Chapter 1: roadmap

1.1 what *is* the Internet（国际互联网）?

1.2 network edge（网络边缘）

- end systems, access networks（接入网）, links

1.3 network core（网络核心）

- packet switching（分组交换）, circuit switching（电路交换）, network structure

1.4 delay（时延）, loss（丢包）, throughput（吞吐） in networks

1.5 protocol layers, service models

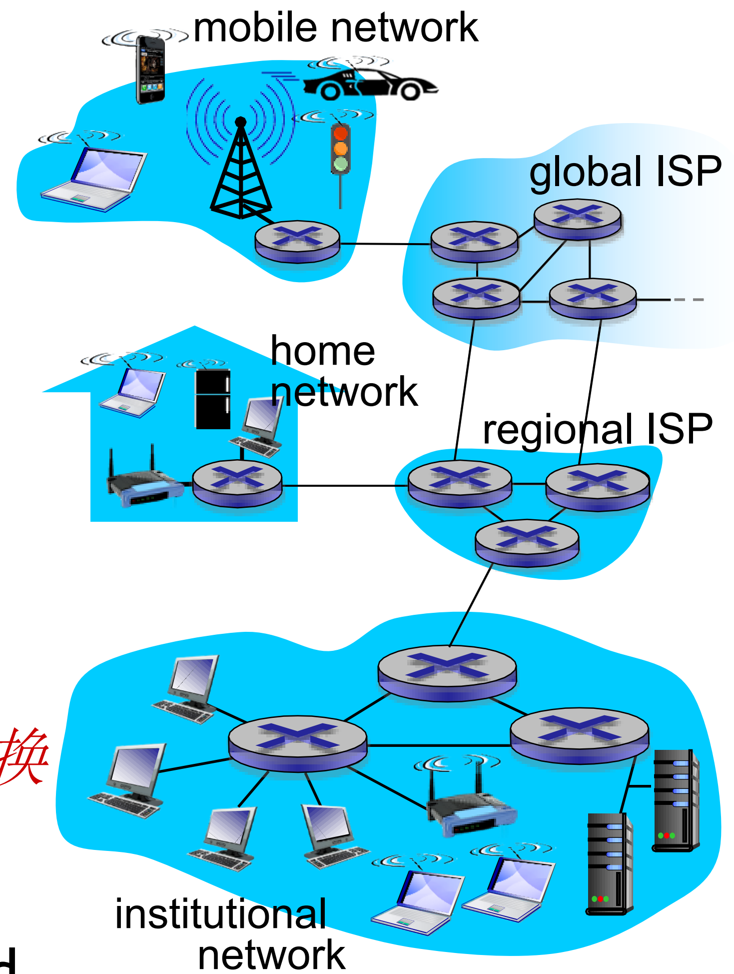
1.6 networks under attack（网络攻击）: security

1.7 history

What's the Internet: "nuts and bolts" view

具体构成的角度看

- billions of connected computing devices:
  - *hosts（主机）* = *end systems（端系统）*
  - running *network apps*
- *通信链路*
  - 光纤,同轴电缆，无线电等
  - 传输速率bit/s: *bandwidth(带宽)*
- *packet switches（分组交换机）:* forward packets (chunks of data)
  - *routers（路由器）* and *switches（交换机）*



"Fun" Internet-connected devices

IP picture frame
http://www.ceiva.com/

Web-enabled toaster +
weather forecaster

Tweet-a-watt:
monitor energy use

Internet
refrigerator

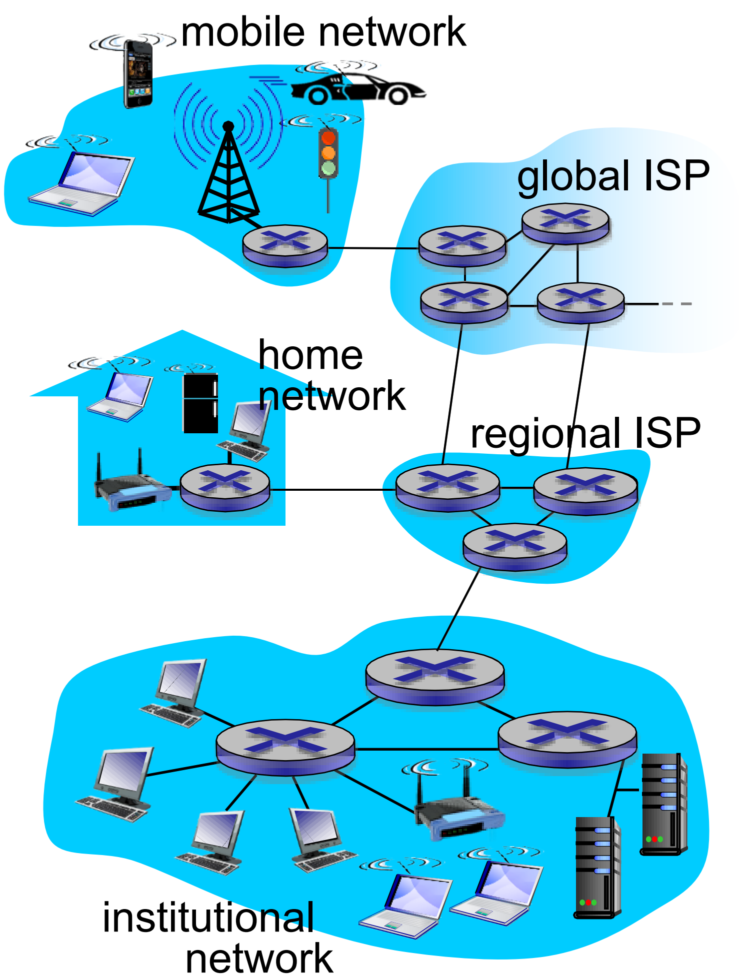
Slingbox: watch,
control cable TV remotely

sensorized,
bed
mattress

Internet phones

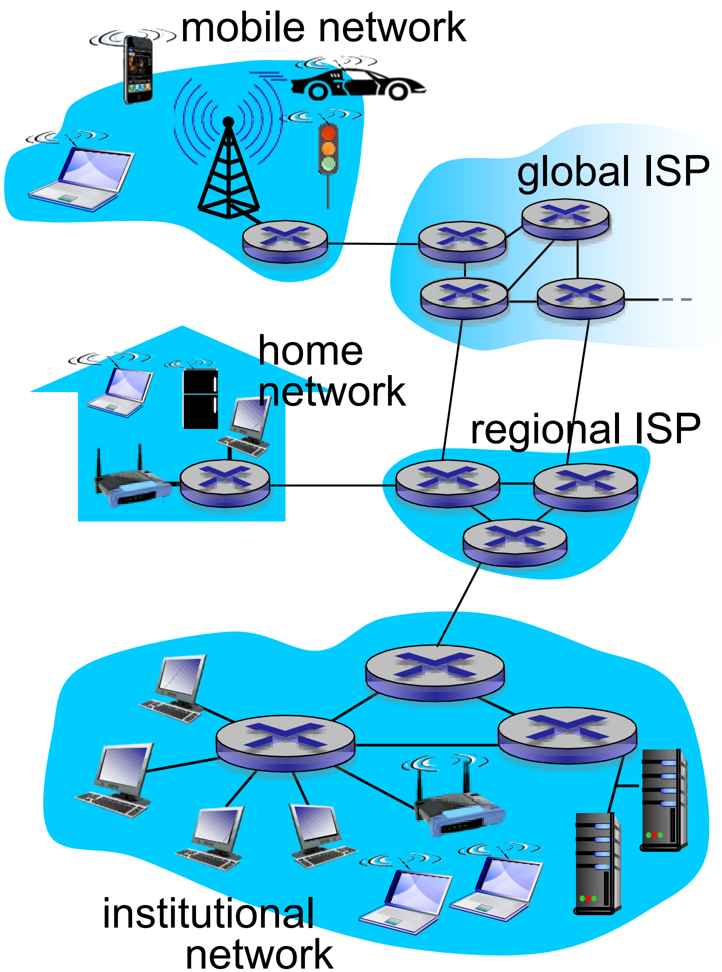
What's the Internet: "nuts and bolts" view

- *Internet:* "network of networks"
  - Interconnected ISPs
- *protocols* control sending, receiving of messages
  - e.g., TCP, IP, HTTP, Skype, 802.11
- *Internet standards*
  - RFC: Request for comments RFC始于1969年，由当时UCLA的Stephen D. Crocker用来记录有关ARPANET开发的非正式文档，最终演变为用来记录互联网规范、协议、过程等的标准文件。
  - IETF: Internet Engineering Task Force（互联网工程应用组），始于1986年的IETF是推动Internet标准规范制定的最主要的组织。



What's the Internet: a service view

- *infrastructure （基础设施） that provides services to applications:*
  - Web, VoIP, email, games, e-commerce, social nets, …
- *provides programming interface to apps*
  - hooks that allow sending and receiving app programs to "connect" to Internet
  - provides service options, analogous to postal service



What's a protocol?（什么是协议？）

human protocols:

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

… specific messages sent

… specific actions taken when messages received, or other events

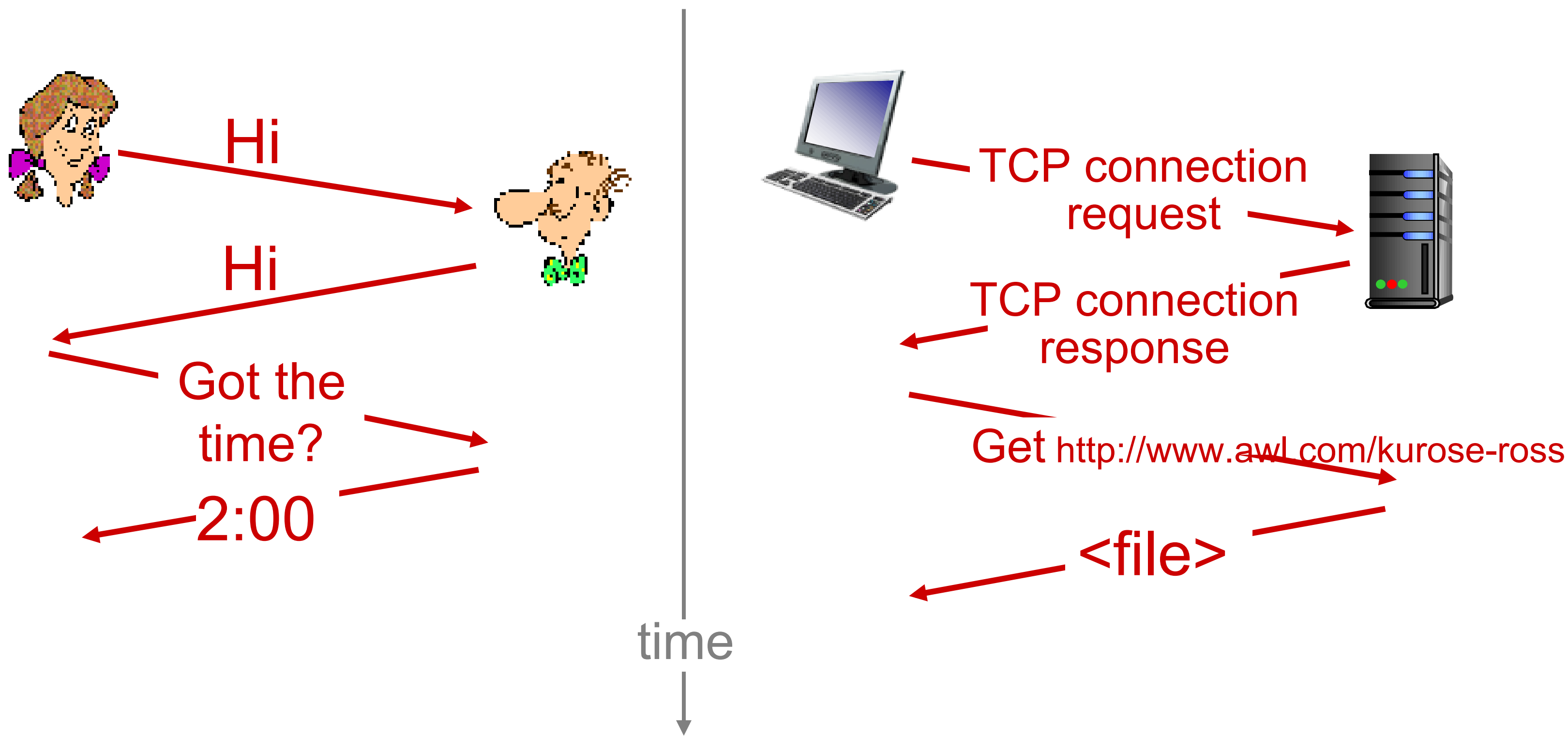
network protocols:

- machines rather than humans
- all communication activity in Internet governed by protocols

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

What's a protocol?

a human protocol and a computer network protocol:



Q: other human protocols?

Chapter 1: roadmap

1.1 what *is* the Internet?

1.2 network edge

- end systems, access networks, links

1.3 network core

- packet switching, circuit switching, network structure

1.4 delay, loss, throughput in networks

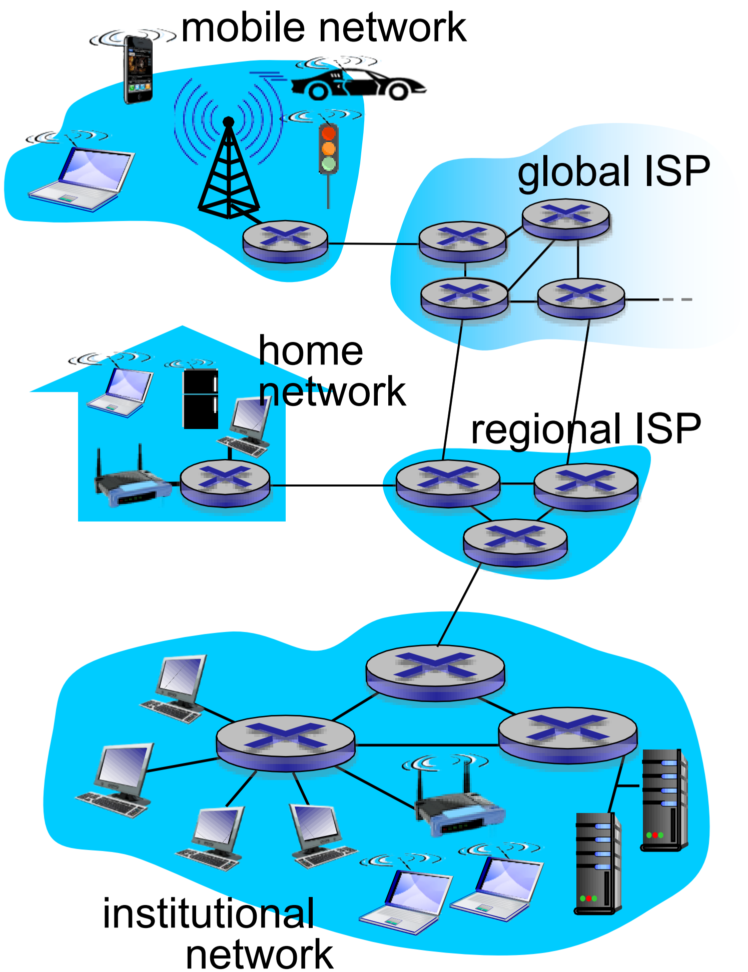
1.5 protocol layers, service models

1.6 networks under attack: security

1.7 history

A closer look at network structure:

- *network edge （网络边缘）：*
  - hosts: clients and servers
  - servers often in data centers
- *access networks, physical media （物理媒介）：* wired, wireless communication links
- *network core（网络核心）：*
  - interconnected routers
  - network of networks



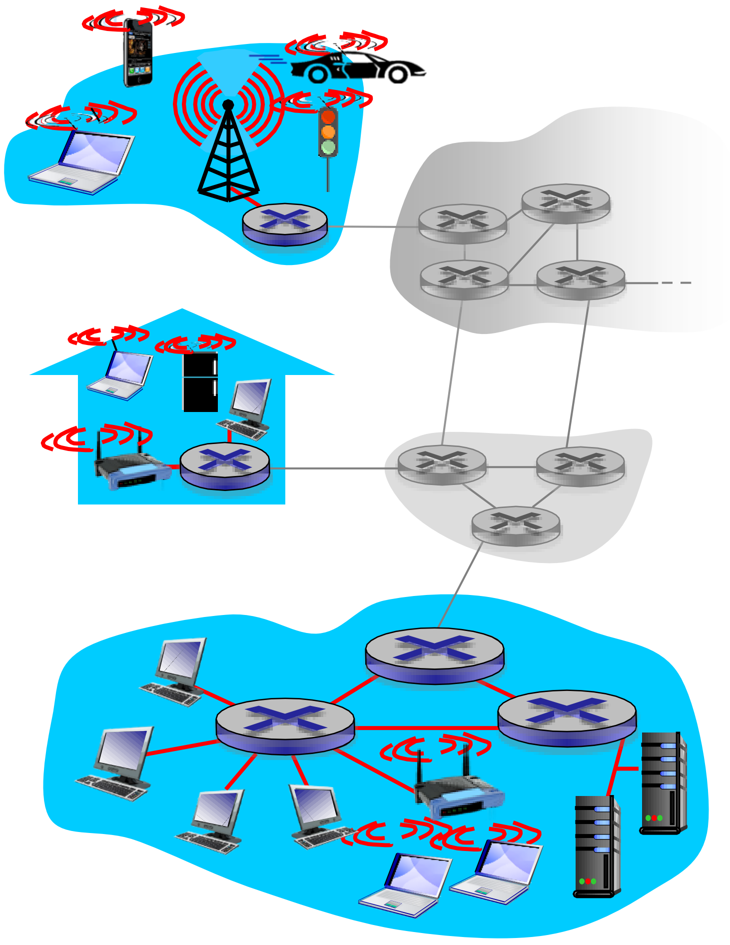
Access networks and physical media

Q: How to connect end systems to edge router?

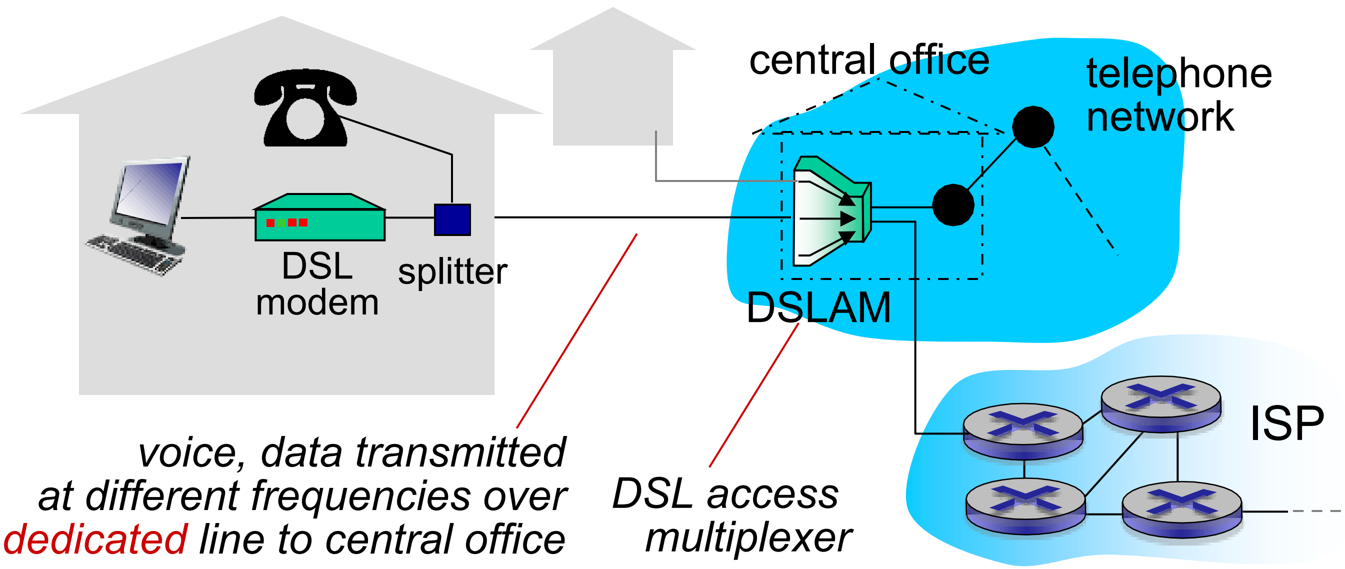
- residential access nets
- institutional access networks (school, company)
- mobile access networks

keep in mind:

- bandwidth (bits per second) of access network?
- shared or dedicated?

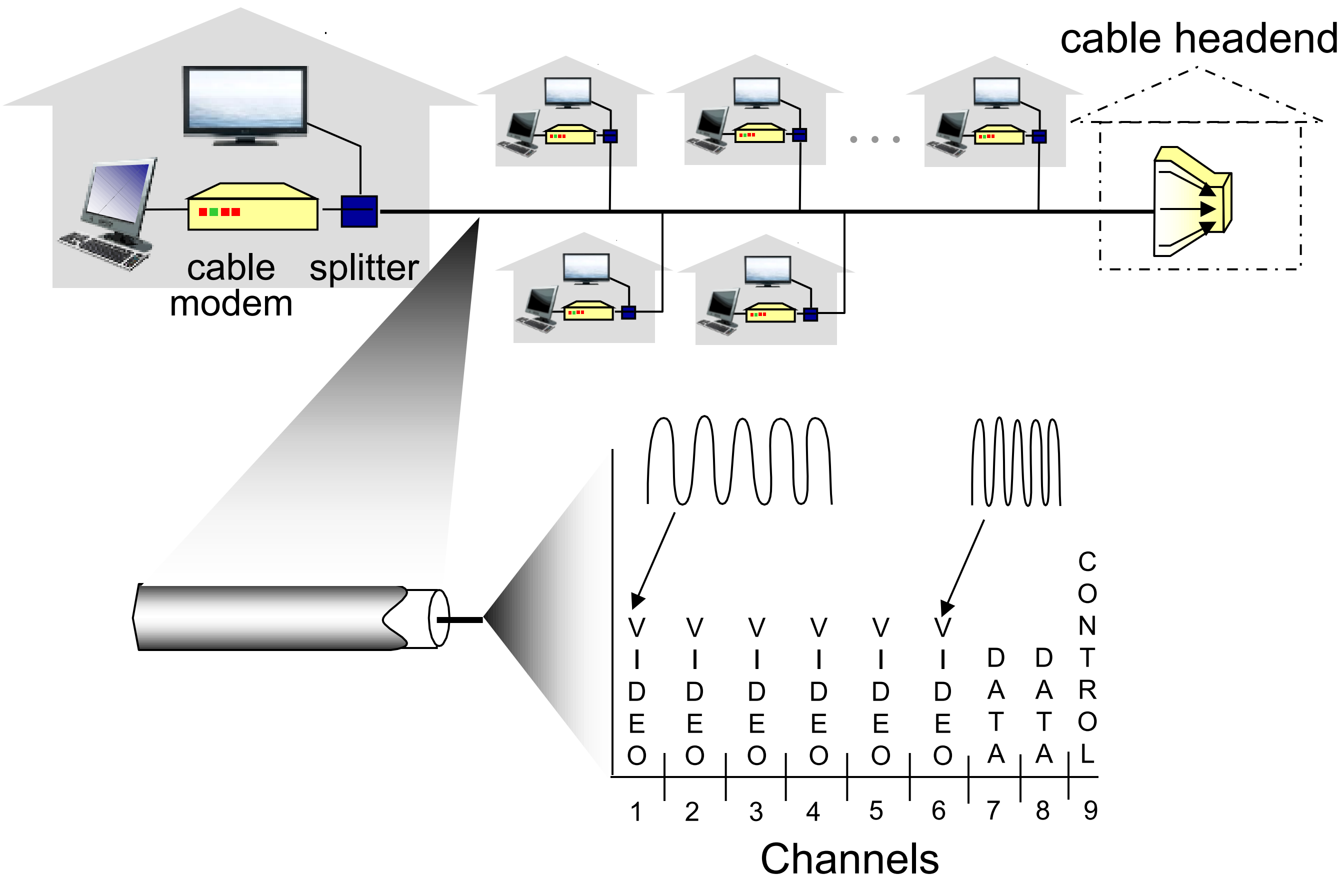


Access network: digital subscriber line (DSL)（数字用户线）



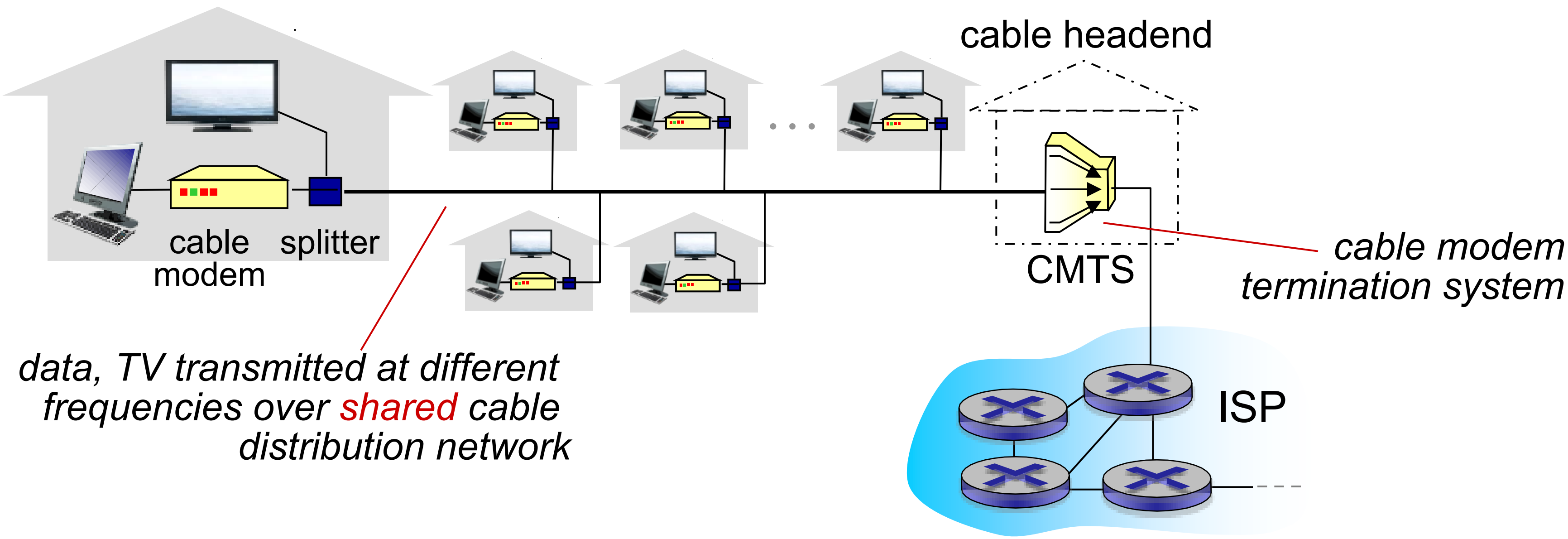
- use *existing* telephone line to central office DSLAM（数字用户线接入复用器）
  - data over DSL phone line goes to Internet
  - voice over DSL phone line goes to telephone net
- < 2.5 Mbps upstream transmission rate (typically < 1 Mbps)
- < 24 Mbps downstream transmission rate (typically < 10 Mbps)

Access network: cable(电缆)network



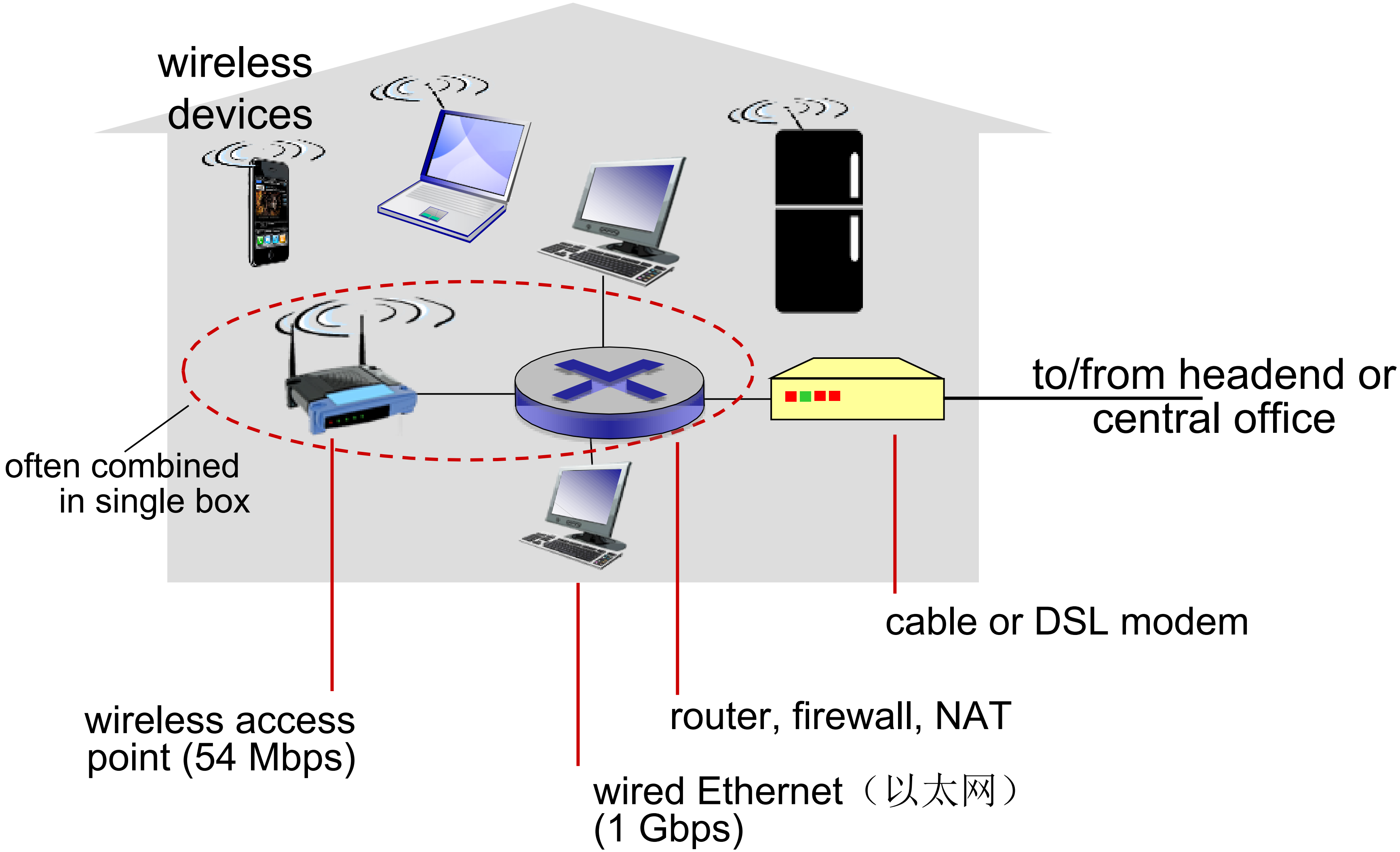
frequency division multiplexing(频分复用): different channels transmitted in different frequency bands（频段）

Access network: cable(电缆) network

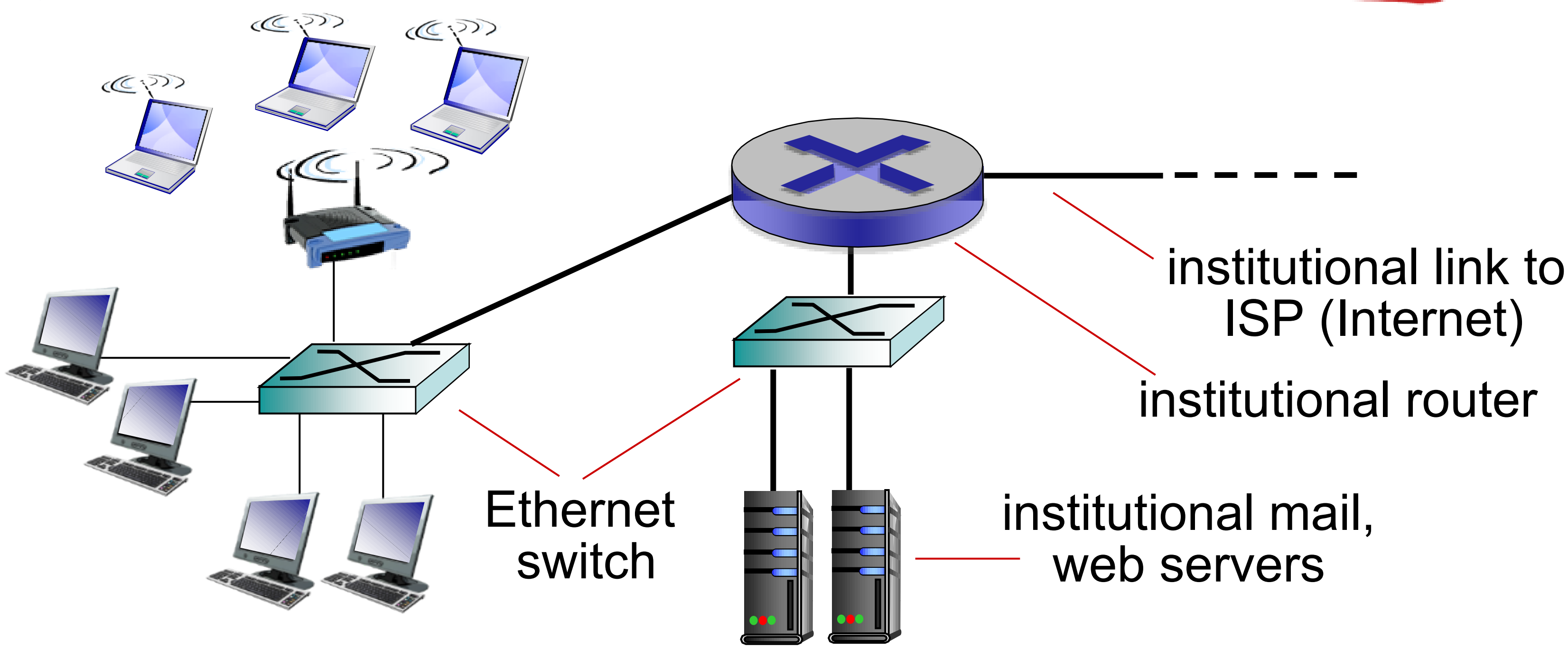


- HFC: hybrid fiber coax（混合光纤同轴）
  - asymmetric: up to 30Mbps downstream transmission rate, 2 Mbps upstream transmission rate
- network of cable, fiber attaches homes to ISP router
  - homes *share access network* to cable headend （电缆头端）
  - unlike DSL, which has dedicated access to central office

Access network: home network



Enterprise access networks (Ethernet) 以太网



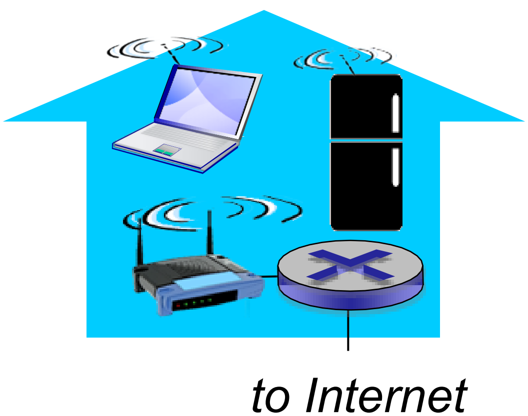
- typically used in companies, universities, etc.
- 10 Mbps, 100Mbps, 1Gbps, 10Gbps transmission rates
- today, end systems typically connect into Ethernet switch

Wireless access networks

- shared *wireless* access network connects end system to router
  - via base station aka "access point"

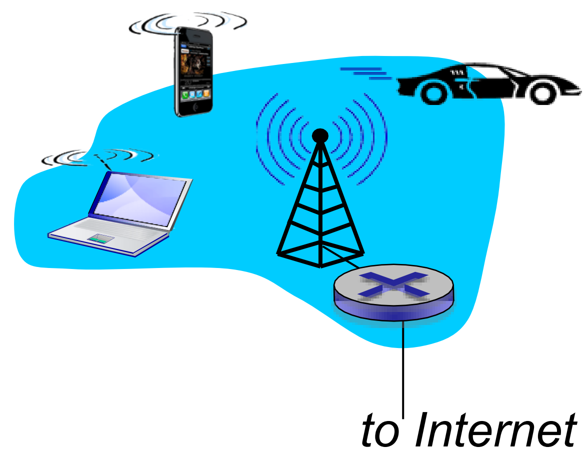
wireless LANs(局域网):

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



wide-area wireless access（广域网）

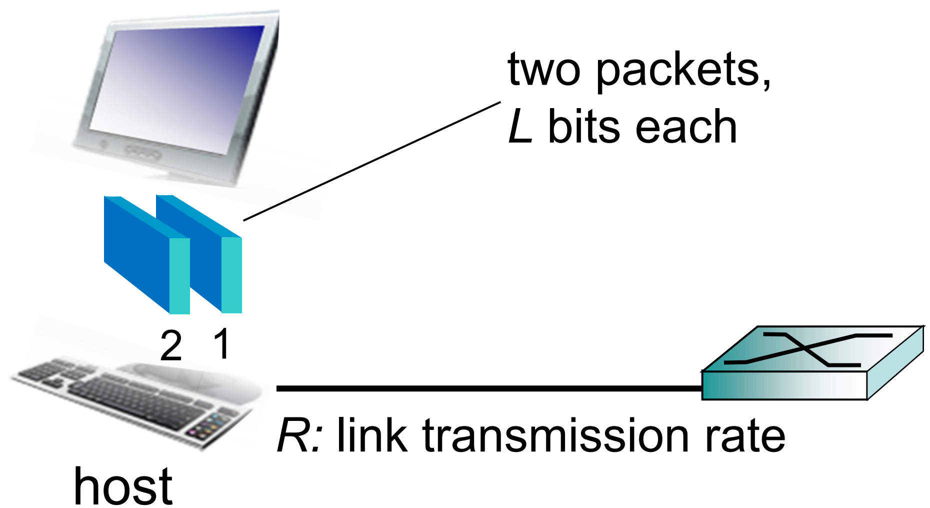
- provided by telco (cellular) operator, 10's km
- between 1 and 10 Mbps
- 3G, 4G



Host: sends *packets*（分组） 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*（传输率） *R*
  - link transmission rate, aka link *capacity, aka link bandwidth*（带宽）



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

Physical media（物理媒介）

- bit: propagates between transmitter/receiver pairs
- physical link（物理链路）: what lies between transmitter & receiver
- guided media（有向介质，有物理限制）:
  - signals propagate in solid media: copper, fiber, coax
- unguided media（无向介质，无实体限制）:
  - signals propagate freely, e.g., radio

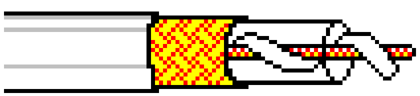
twisted pair (TP)（双绞线）

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

Physical media: coax, fiber

coaxial cable（同轴电缆）:

- two concentric copper conductors
- bidirectional（双工）
- broadband:
  - multiple channels on cable



fiber optic cable（光缆）:

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

Physical media: radio（无线电波）

- signal carried in electromagnetic spectrum
- no physical "wire"
- bidirectional
- propagation environment effects:
  - reflection
  - obstruction by objects
  - interference

radio link types:

- terrestrial microwave（地面微波）
  - e.g. up to 45 Mbps channels
- LAN (e.g., WiFi)
  - 54 Mbps
- wide-area (e.g., cellular)
  - 4G cellular: ~ 10 Mbps
- satellite（卫星）
  - Kbps to 45Mbps channel (or multiple smaller channels)
  - 270 msec end-end delay
  - geosynchronous versus low altitude

课后作业

- 画出目前你正在使用的网路接入方式，并解释。
- 你正在给老师发送一封电子邮件，能说明邮件如何一步步被传送到了老师这里吗？

Chapter 1: roadmap

1.1 what *is* the Internet?

1.2 network edge

- end systems, access networks, links

1.3 network core

- packet switching（分组交换）, circuit switching（电路交换）, network structure

1.4 delay, loss, throughput in networks

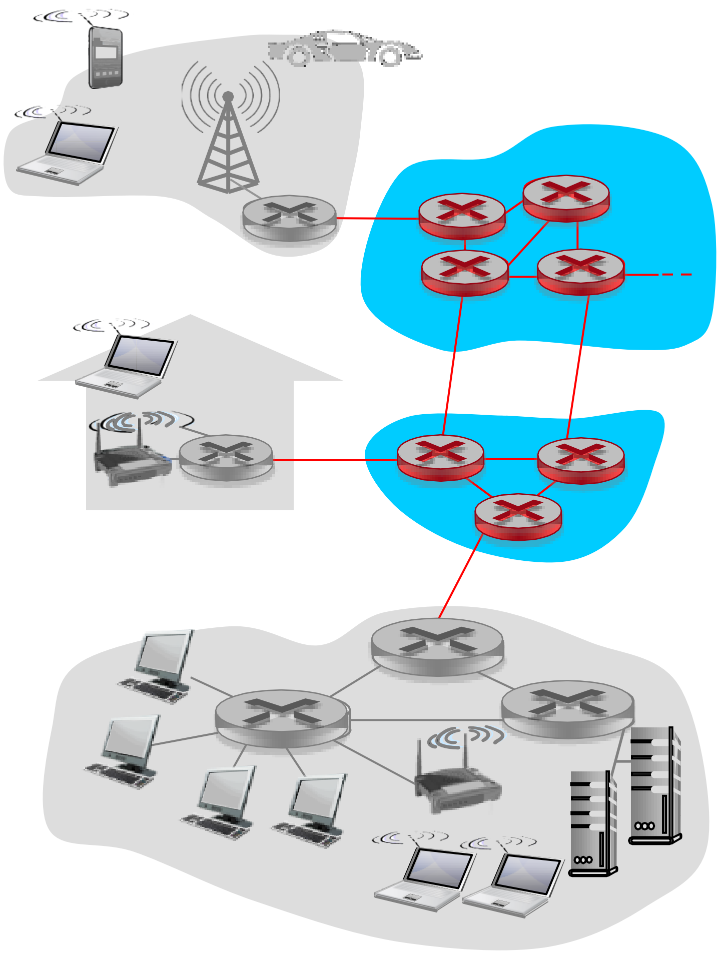
1.5 protocol layers, service models

1.6 networks under attack: security

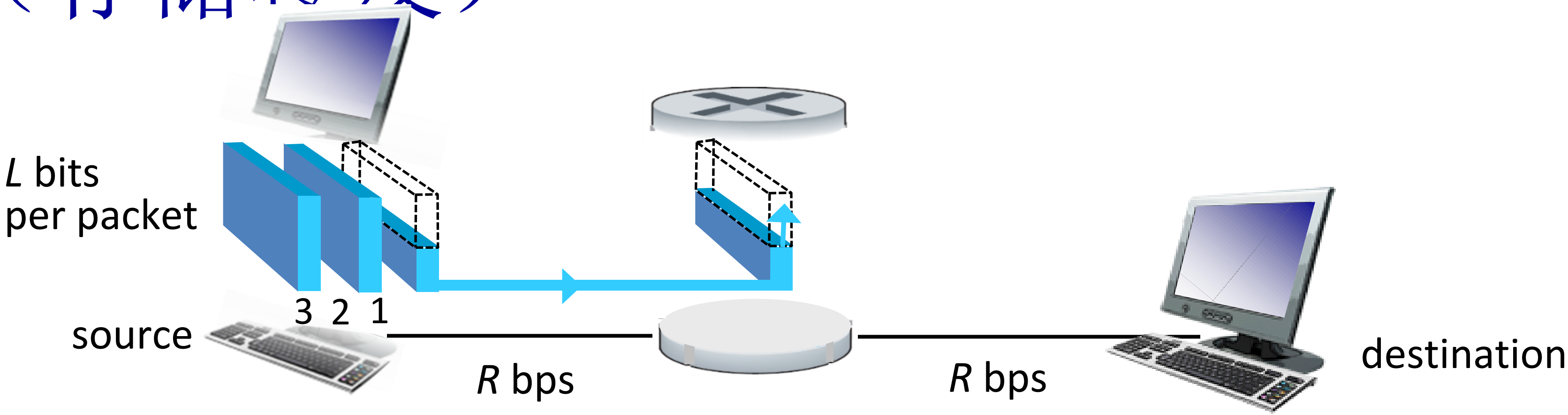
1.7 history

The network core

- mesh of interconnected routers
- packet-switching（分组交换）: 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



Packet-switching: store-and-forward（存储转发）



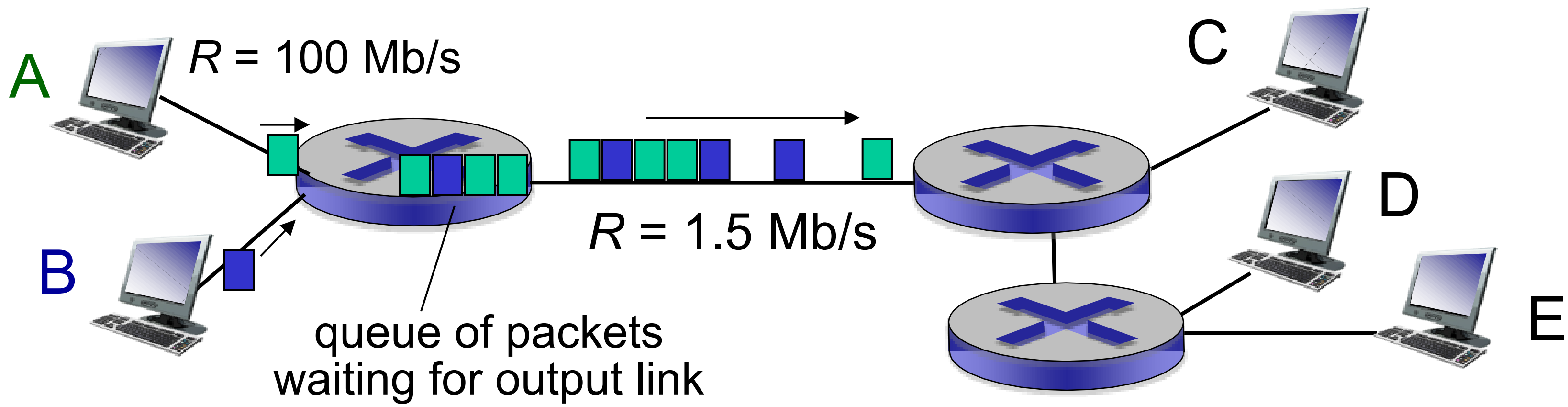
- takes *L*/*R* seconds to transmit (push out) *L*-bit packet into link at *R* bps
- *store and forward:* entire packet must arrive at router before it can be transmitted on next link
- end-end delay = 2*L*/*R* (assuming zero propagation delay)

more on delay shortly …

one-hop numerical example:

- *L* = 7.5 Mbits
- *R* = 1.5 Mbps
- one-hop transmission delay = 5 sec

Packet Switching: queueing delay, loss



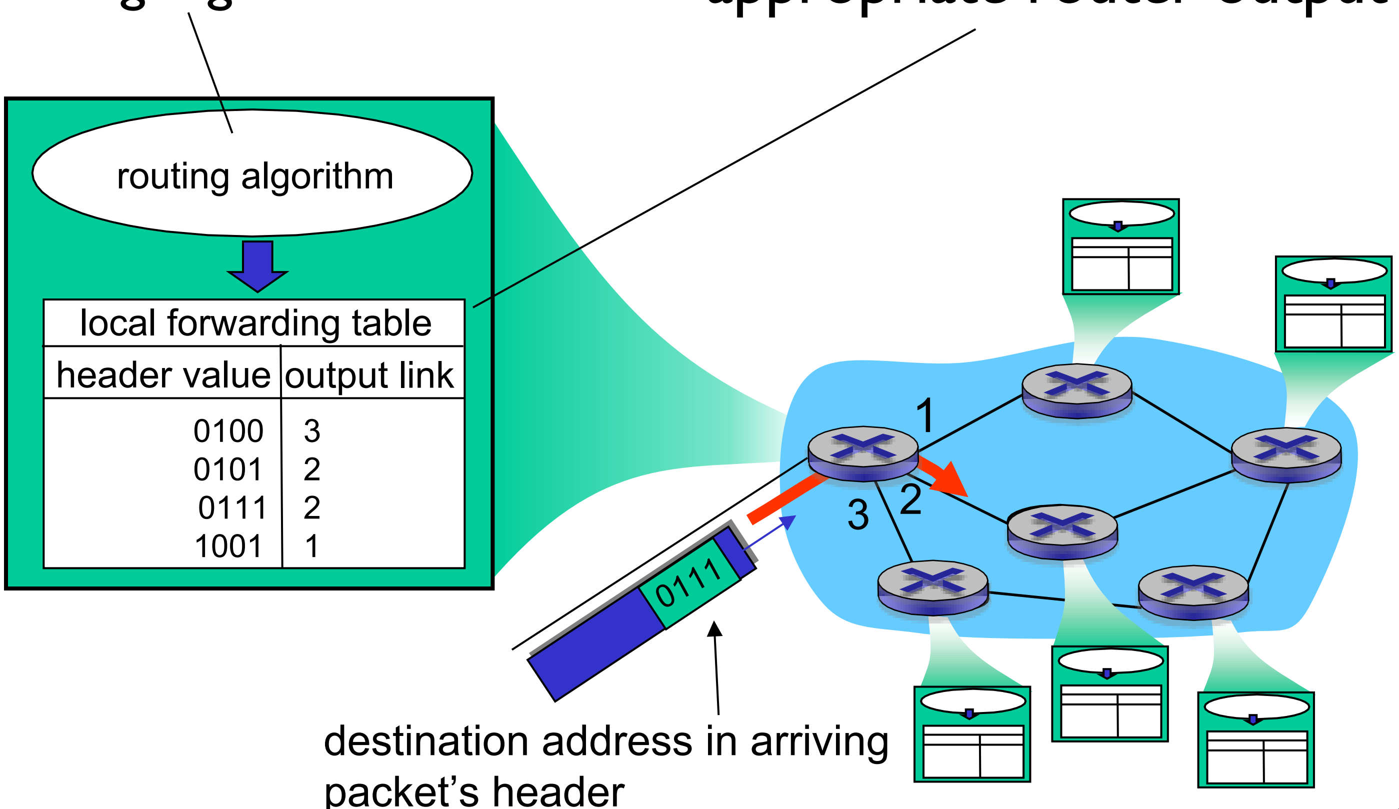
queuing and loss:

- 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

Two key network-core functions

routing（路由）: determines source-destination route taken by packets
- *routing algorithms*

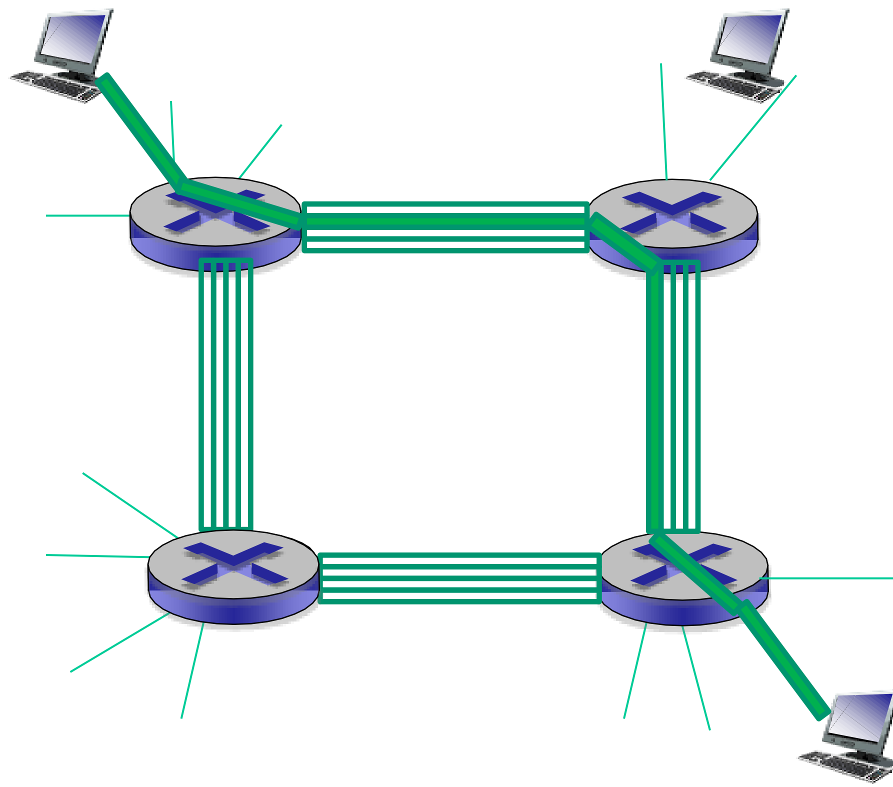
forwarding（转发）: move packets from router's input to appropriate router output



Alternative core: circuit switching （电路交换）

end-end resources allocated to, reserved for "call" between source & dest:

- 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



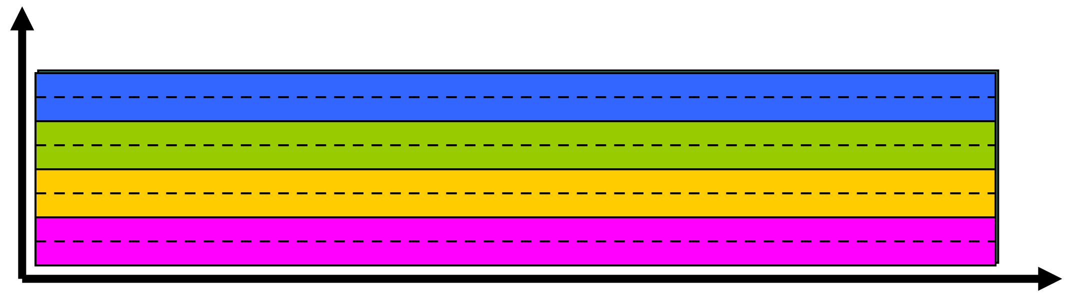
Circuit switching: FDM versus TDM

Example:

4 users

FDM（频分复用）

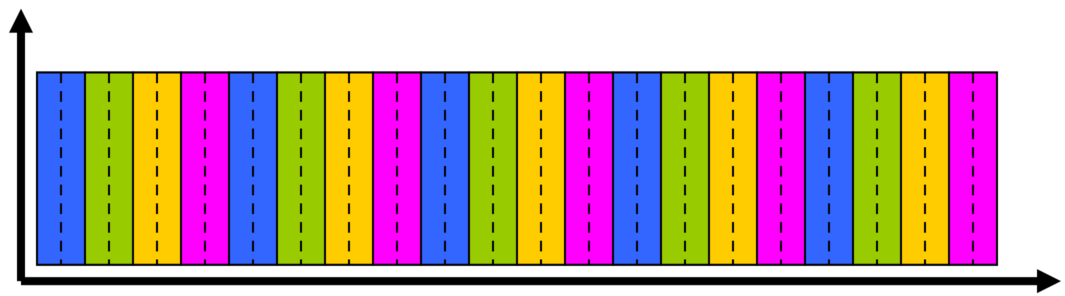
frequency



time

TDM（时分复用）

frequency



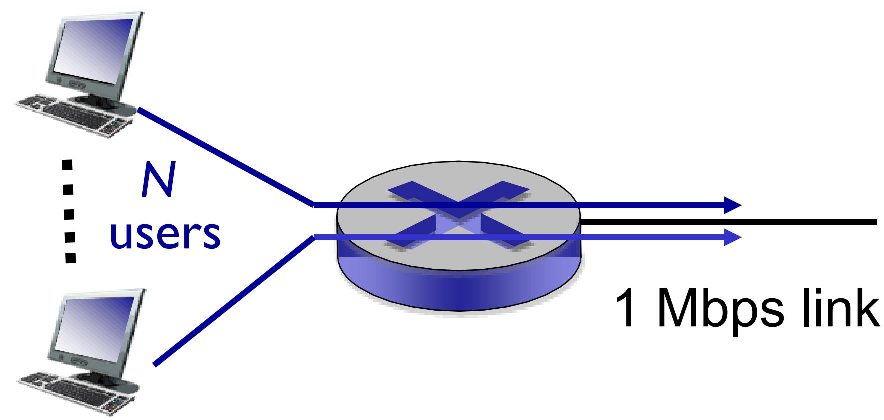
time

Packet switching versus circuit switching

packet switching allows more users to use network!

example:

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



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

使用二项式分布计算

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/

Packet switching versus circuit switching

is packet switching a "slam dunk winner?"

- great for bursty data
  - resource sharing
  - simpler, no call setup
- excessive congestion（拥塞） possible: packet delay and loss
  - protocols needed for reliable data transfer, congestion control
- *Q:* How to provide circuit-like behavior?
  - bandwidth guarantees needed for audio/video apps
  - still an unsolved problem (chapter 7)

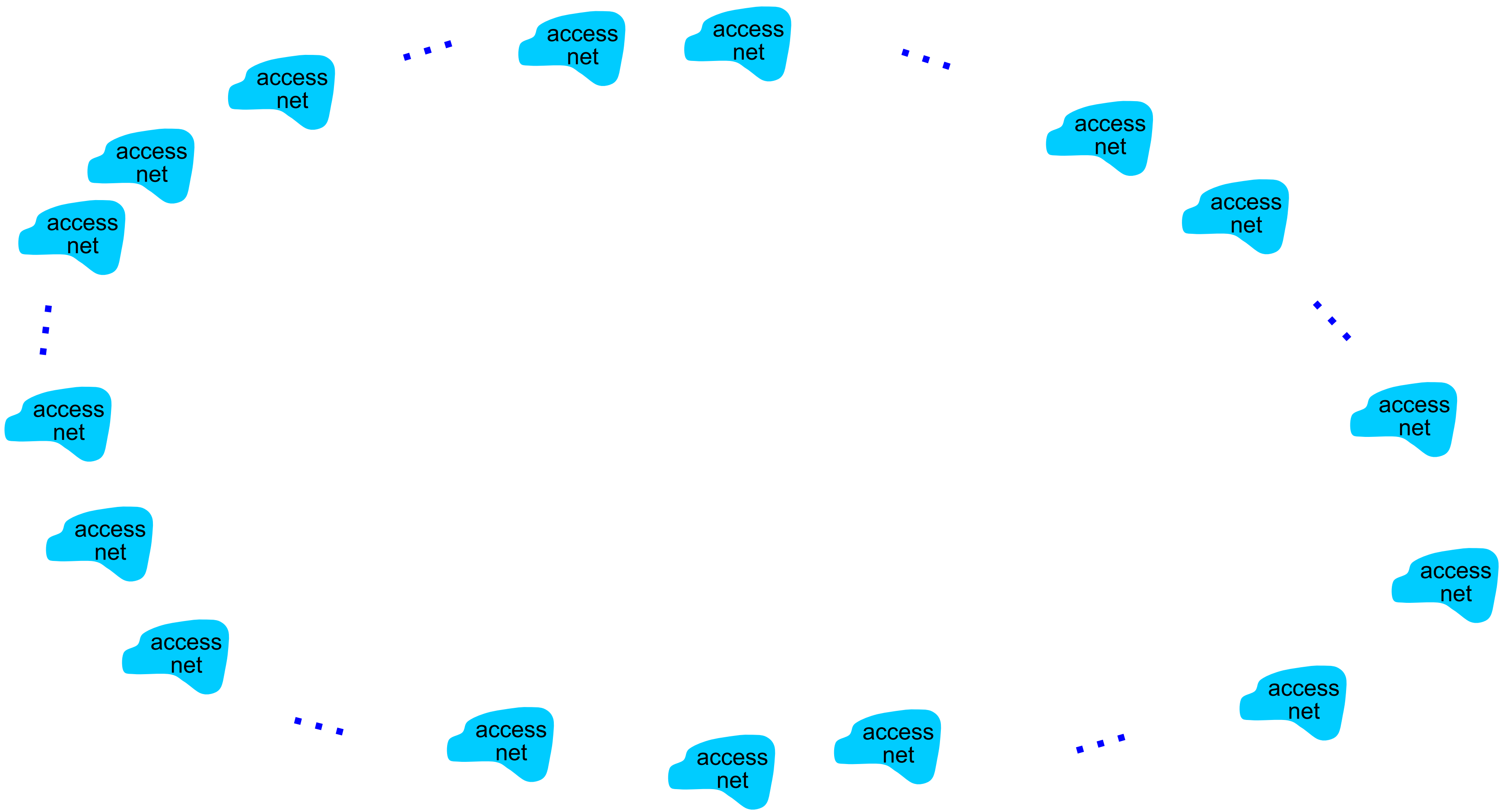
Q: human analogies of reserved resources (circuit switching) versus on-demand allocation (packet-switching)?

Internet structure: network of networks

- End systems connect to Internet via access ISPs (Internet Service Providers，Internet提供商)
  - 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

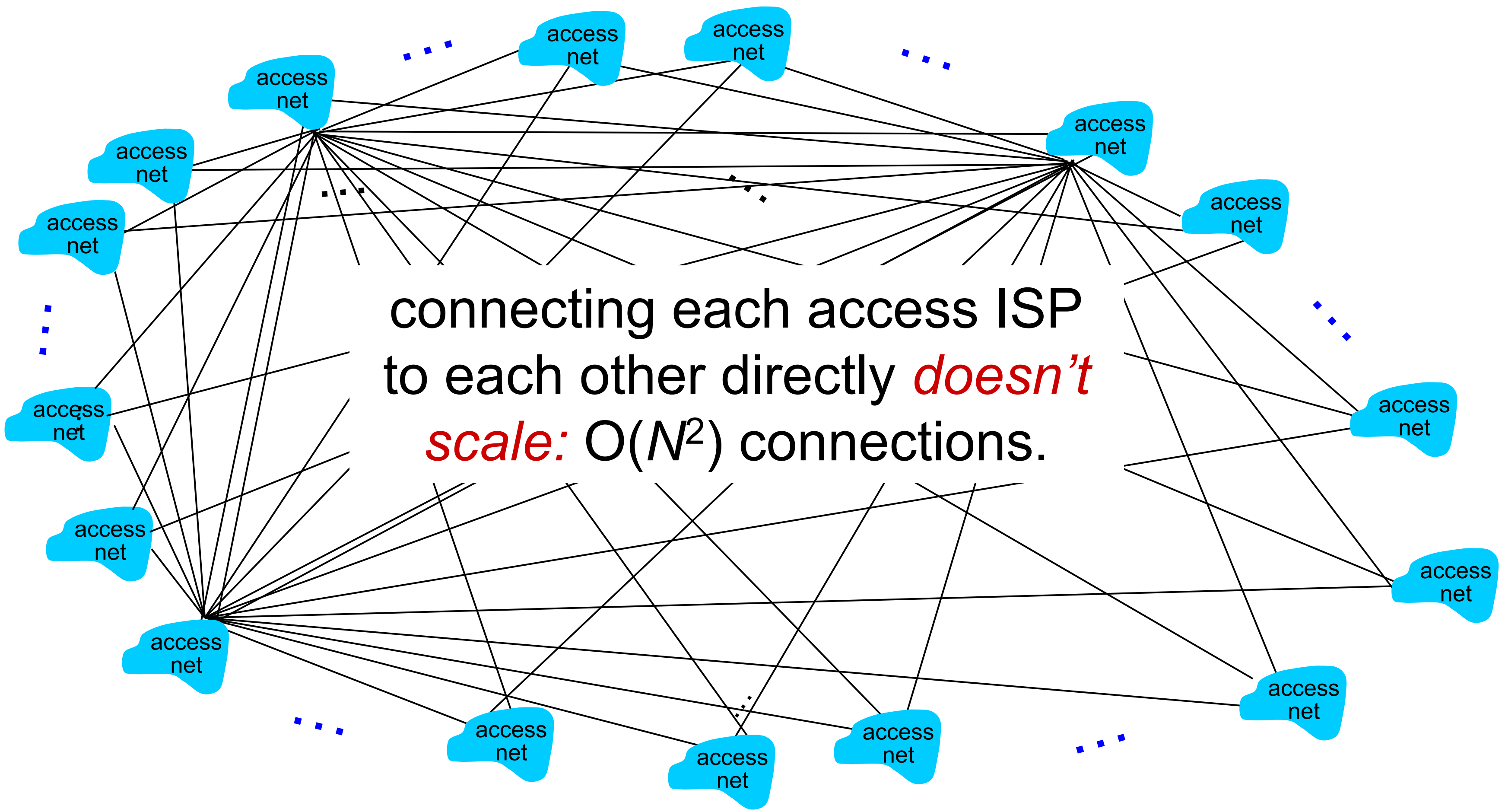
Internet structure: network of networks

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



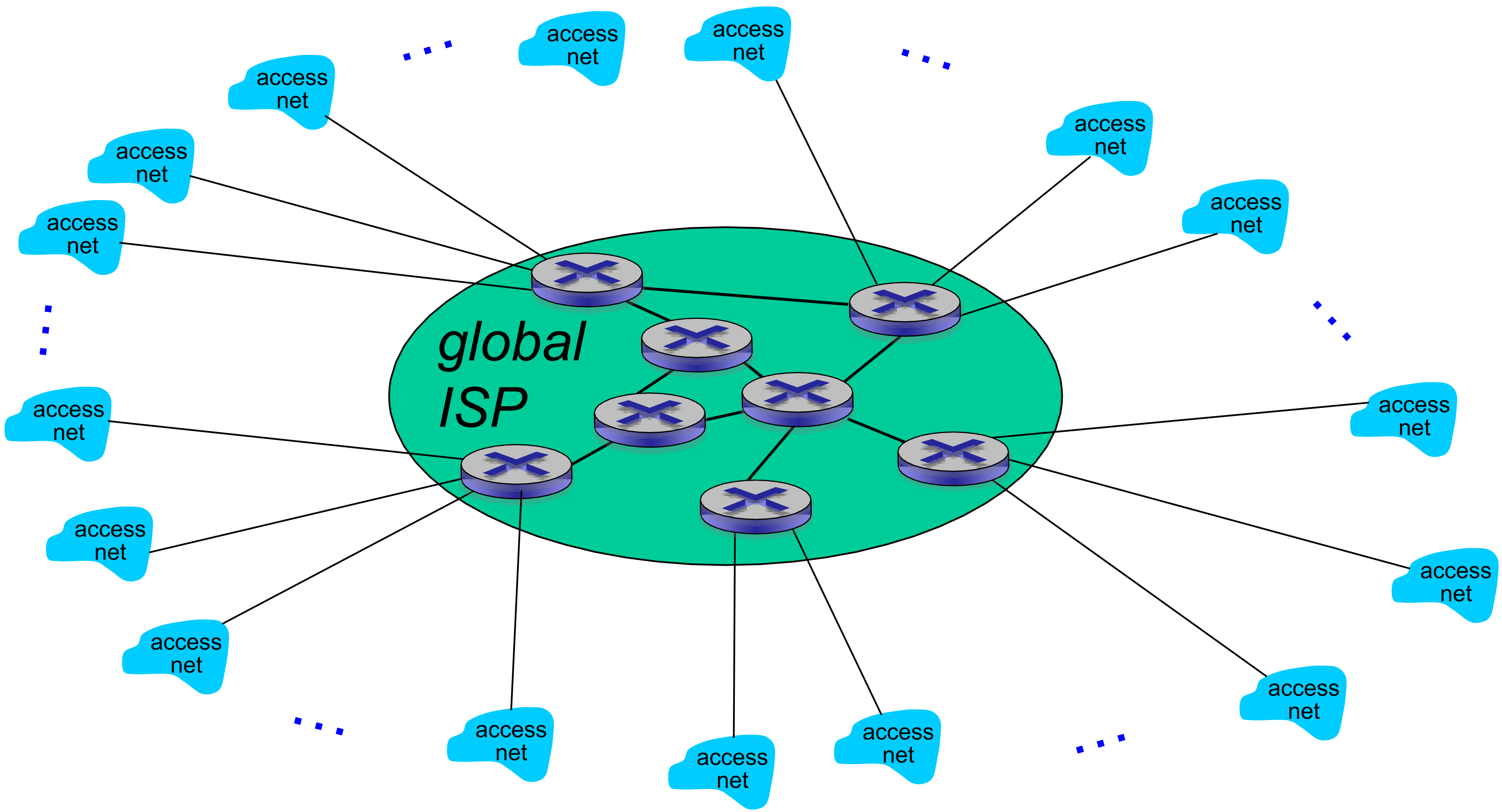
Internet structure: network of networks

Option: connect each access ISP to every other access ISP?



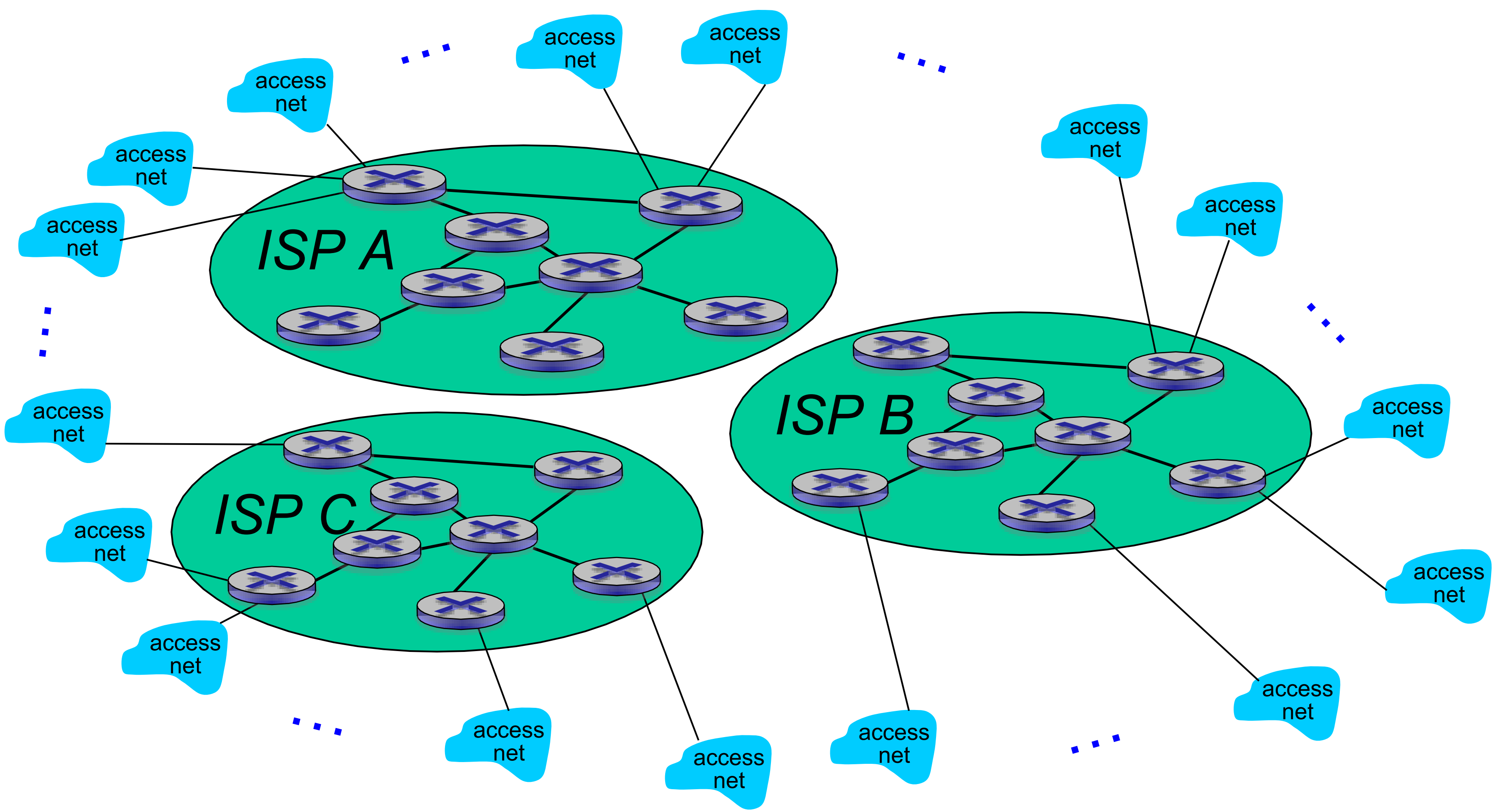
Internet structure: network of networks

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



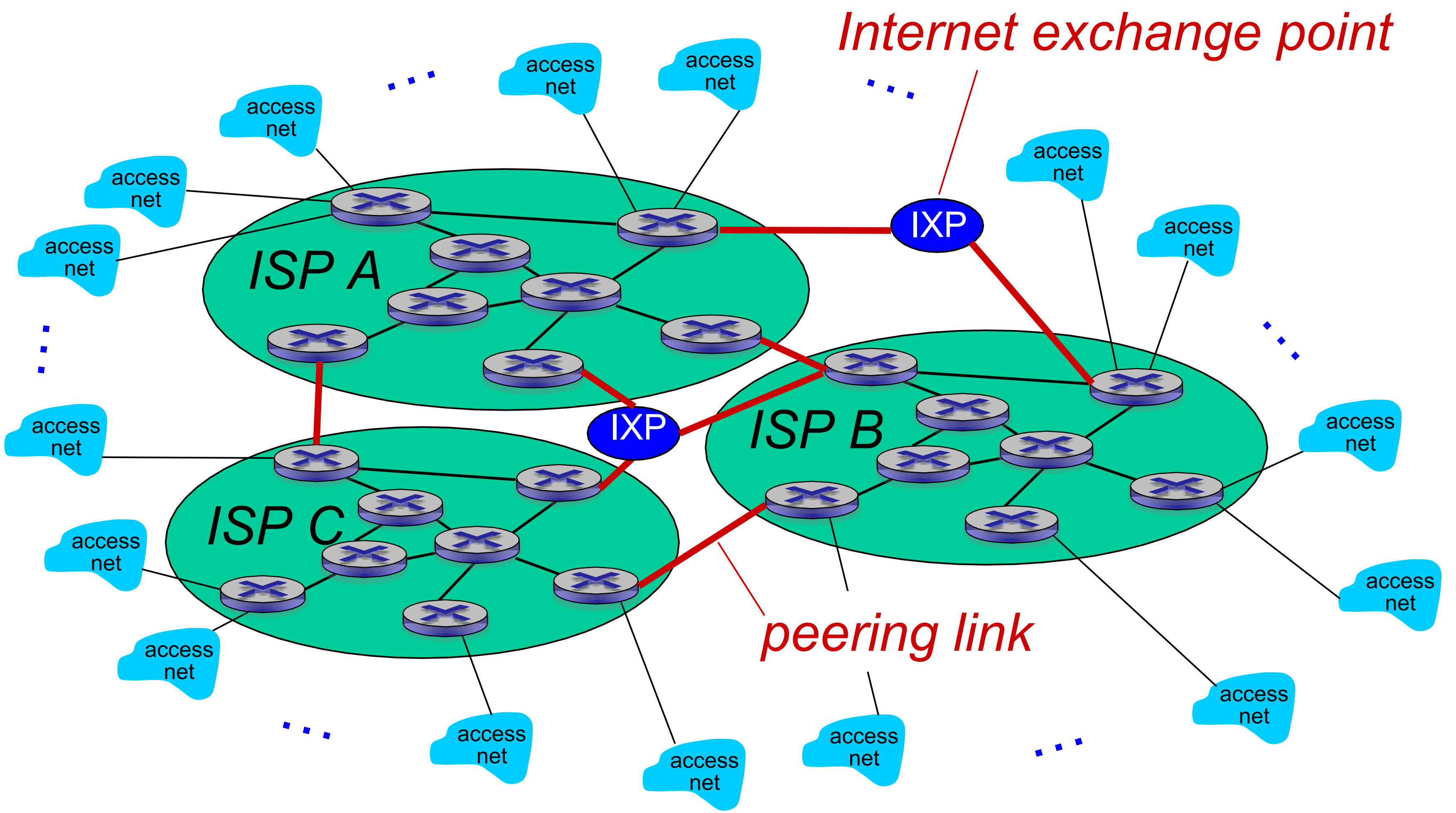
Internet structure: network of networks

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



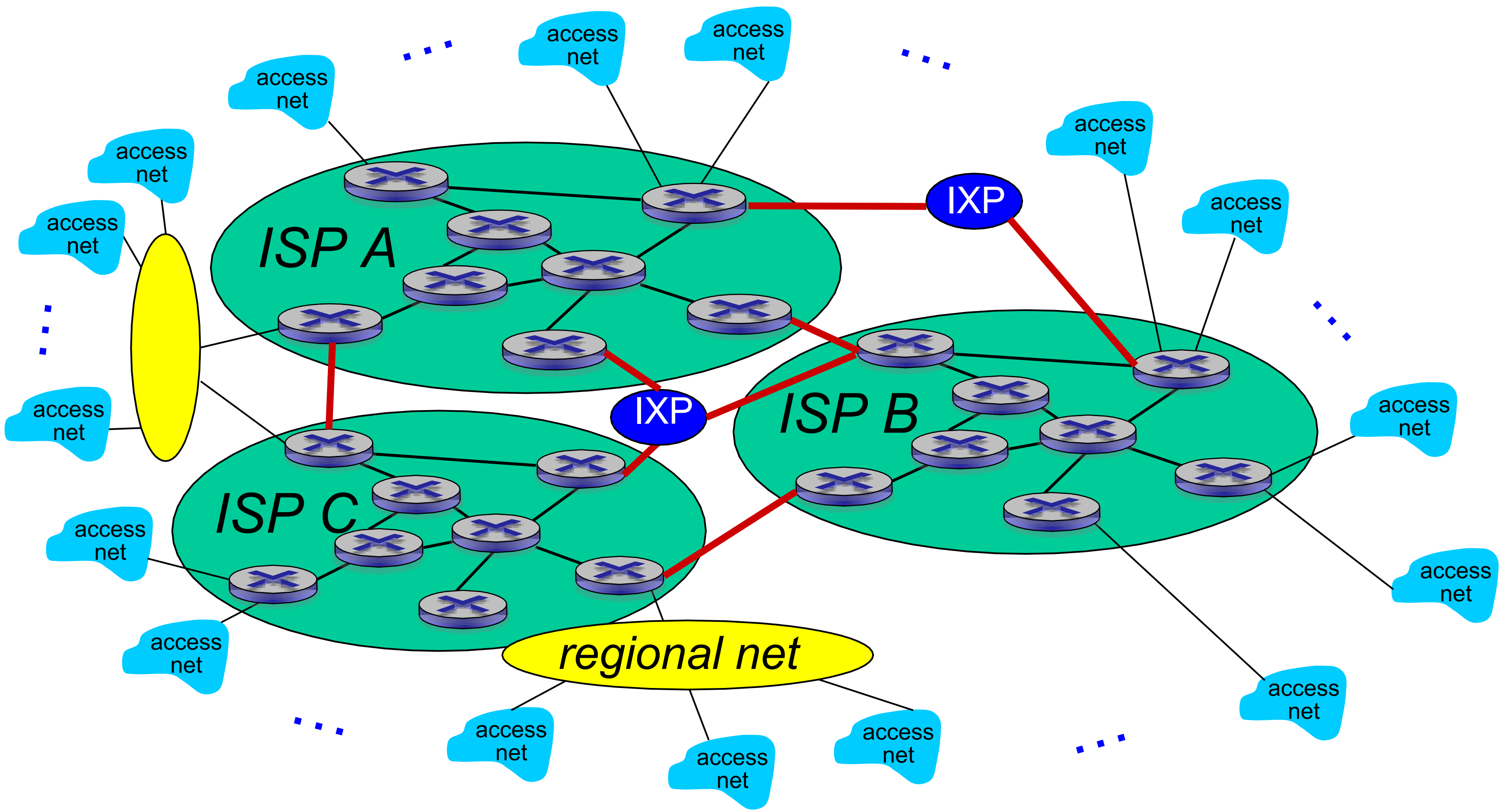
Internet structure: network of networks

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



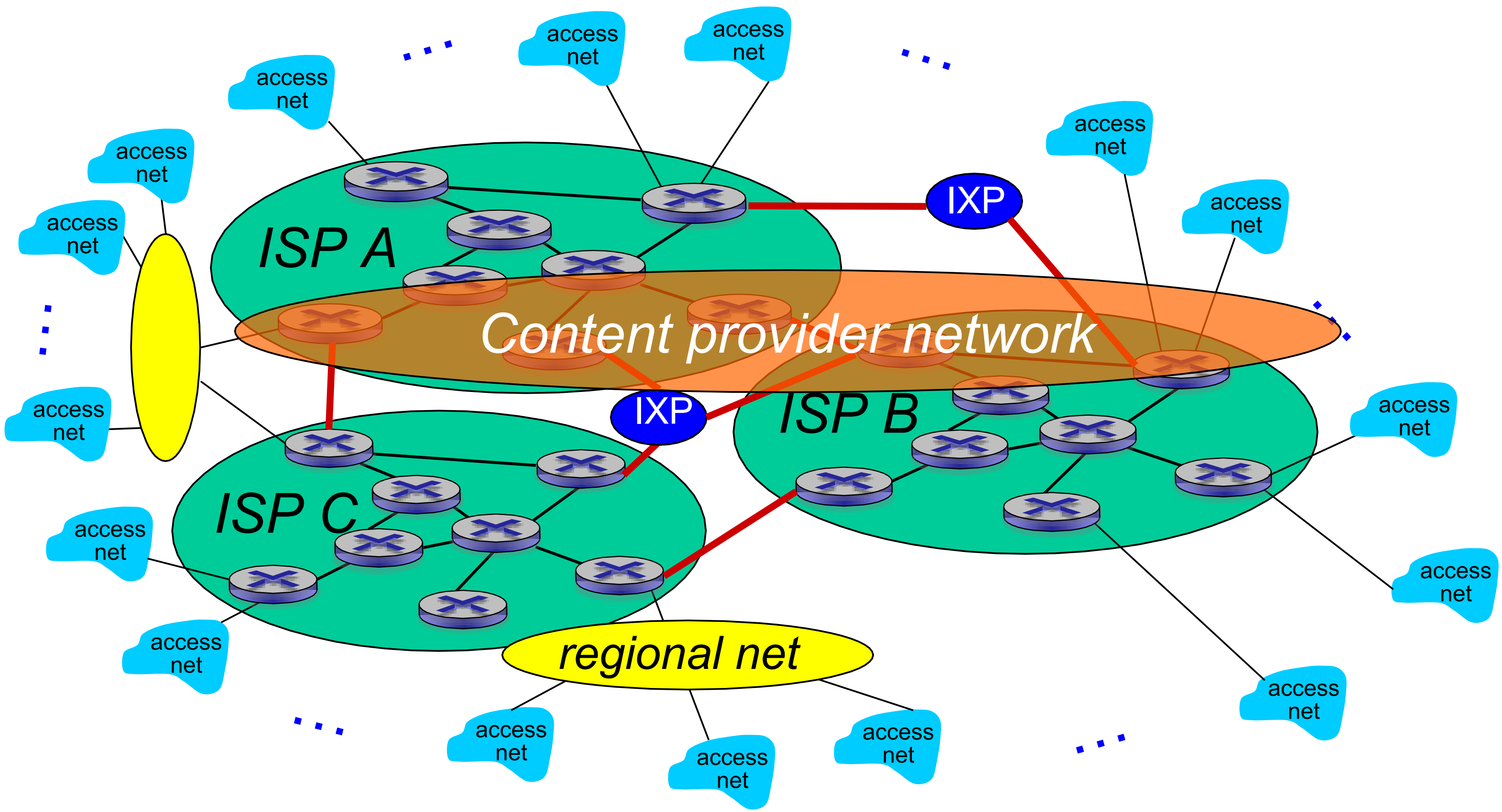
Internet structure: network of networks

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

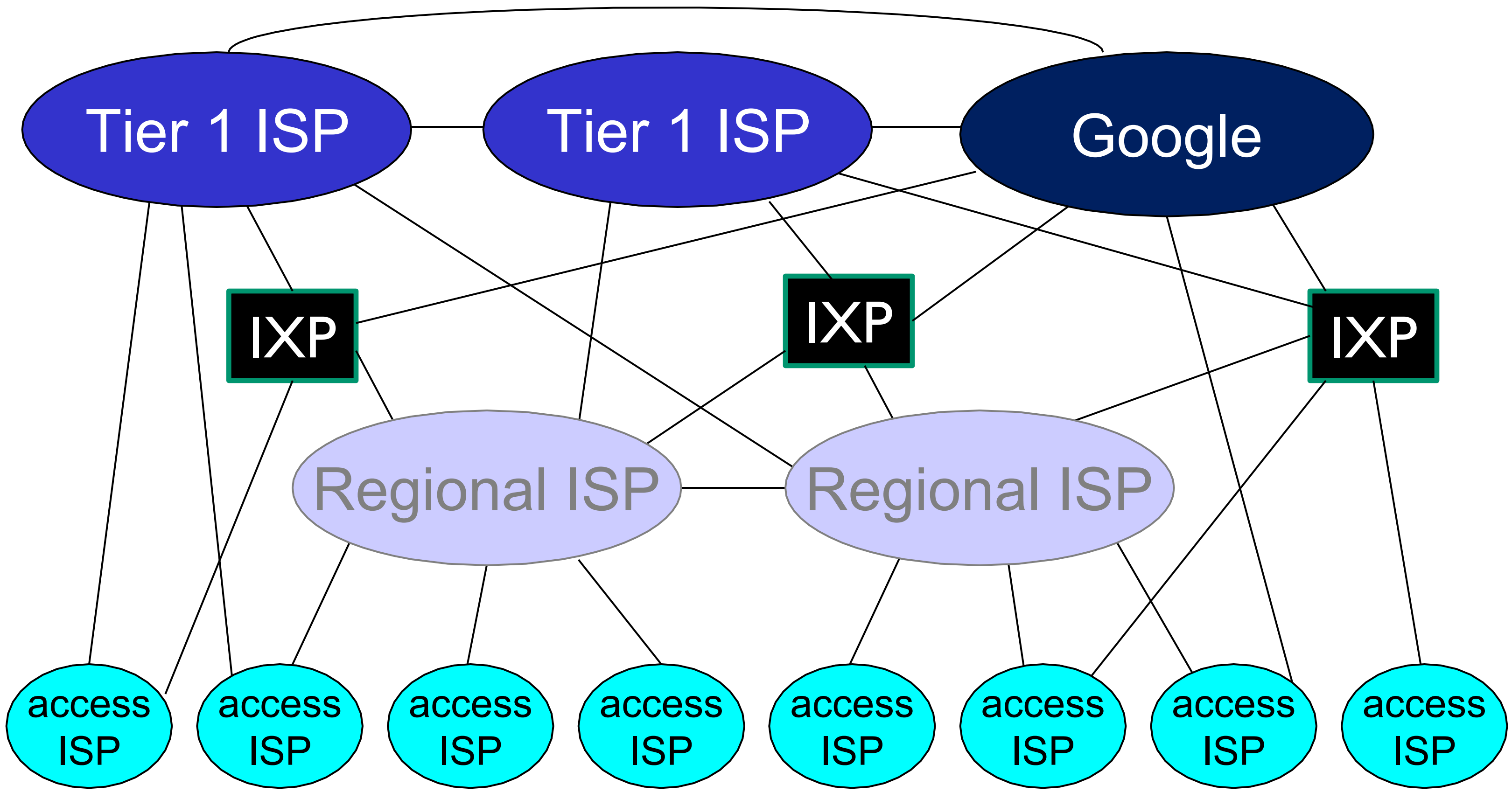


Internet structure: network of networks

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



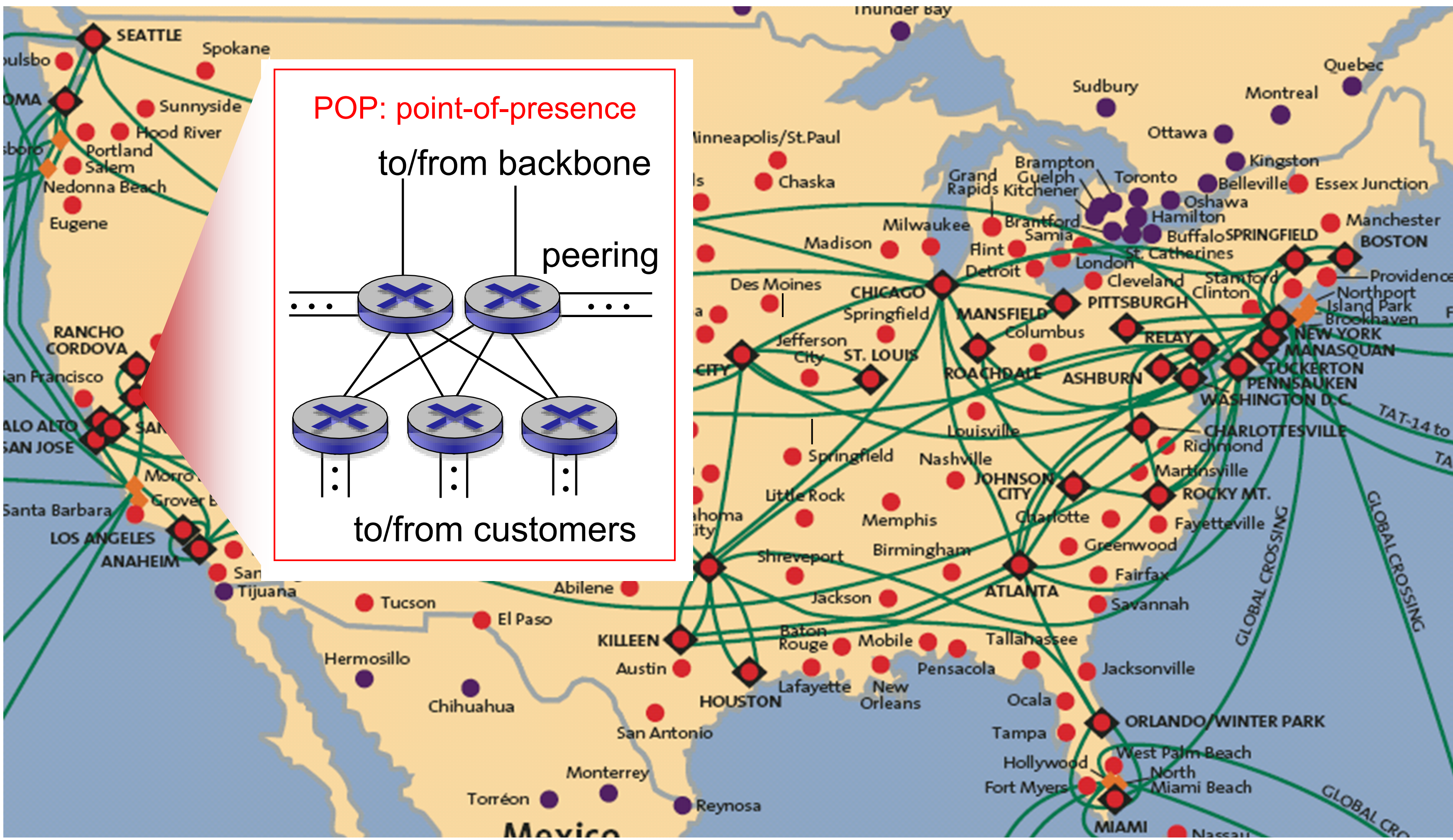
Internet structure: network of networks



- 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 network (e.g., Google): private network that connects it data centers to Internet, often bypassing tier-1, regional ISPs

Tier-1 ISP: e.g., Sprint

POP：入网点



Chapter 1: roadmap

1.1 what *is* the Internet?

1.2 network edge
- end systems, access networks, links

1.3 network core
- packet switching, circuit switching, network structure

1.4 delay, loss, throughput in networks

1.5 protocol layers, service models

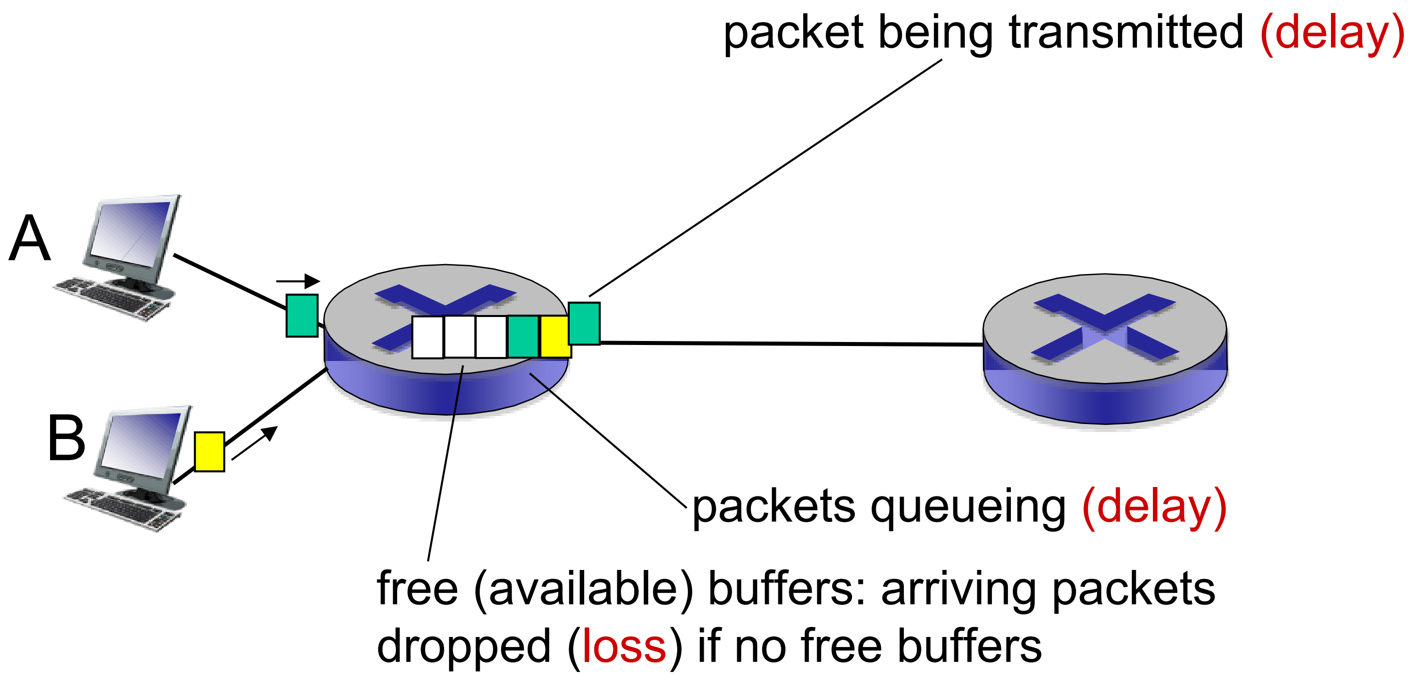
1.6 networks under attack: security

1.7 history

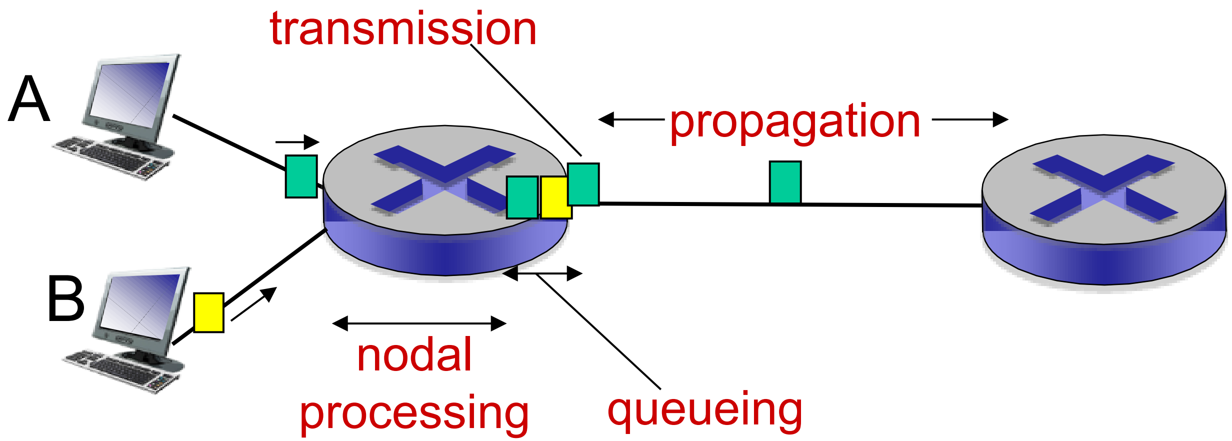
How do loss（丢包） and delay occur?

packets *queue* in router buffers

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



Four sources of packet delay



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

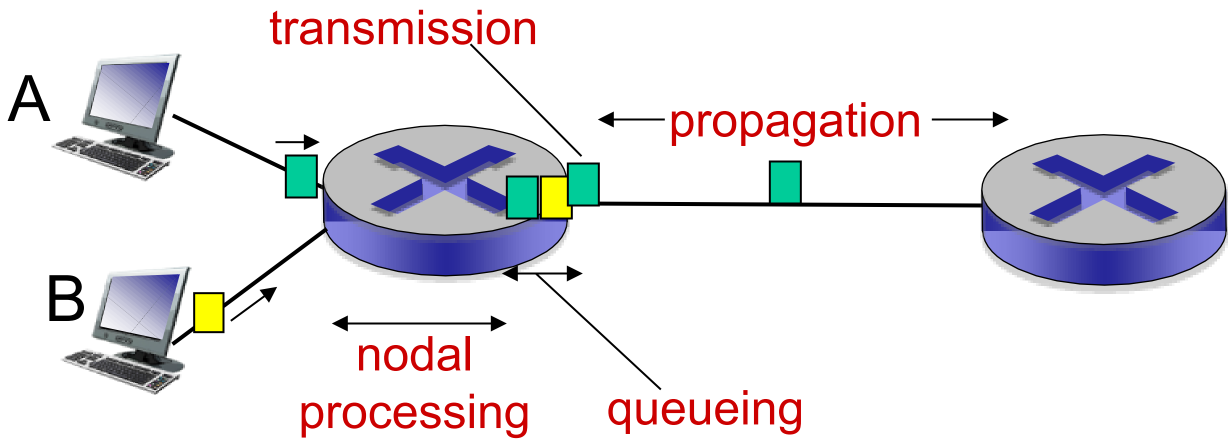
d_{proc}: nodal processing

- 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



传输时延

传播时延

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

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

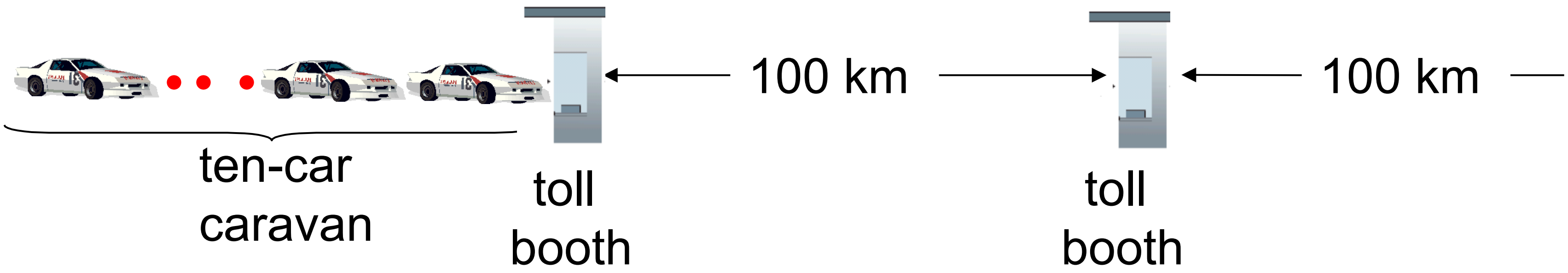
d_{trans} and d_{prop} *very* different

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

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

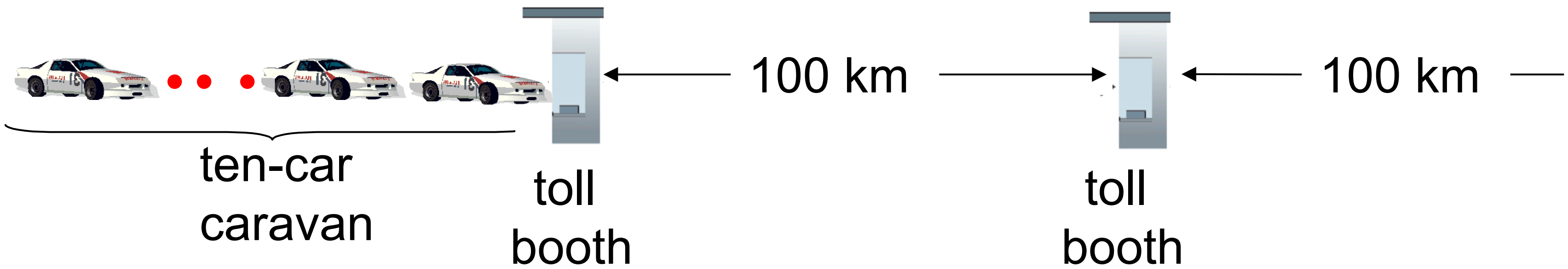
* Check out the Java applet for an interactive animation on trans vs. prop delay

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
- time for last car to propagate from 1st to 2nd toll both: 100km/(100km/hr)= 1 hr
- *A*: 62 minutes

Caravan analogy (more)

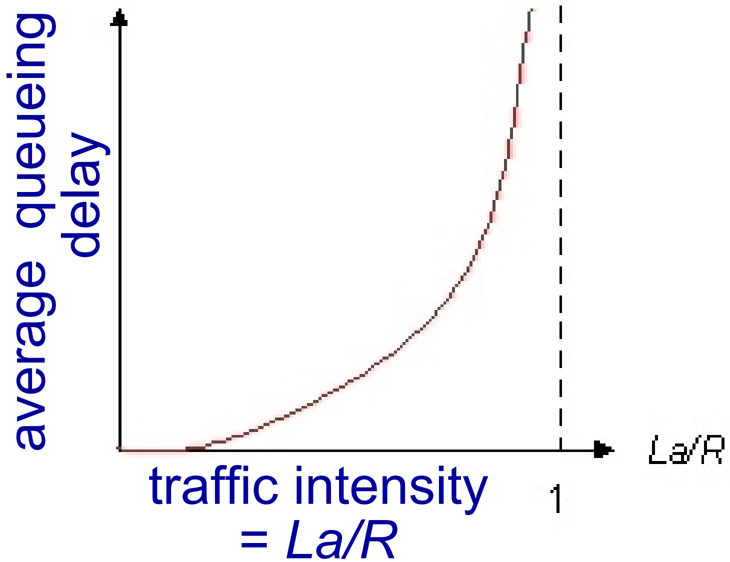


- suppose cars now "propagate" at 1000 km/hr
- and suppose toll booth（收费站） now takes one min to service a car
- *Q:* Will cars arrive to 2nd booth before all cars serviced at first booth?
  - *A: Yes!* after 7 min, first car arrives at second booth; three cars still at first booth

Queueing delay (revisited)

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

traffic intensity *La/R*
流量强度



- *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!

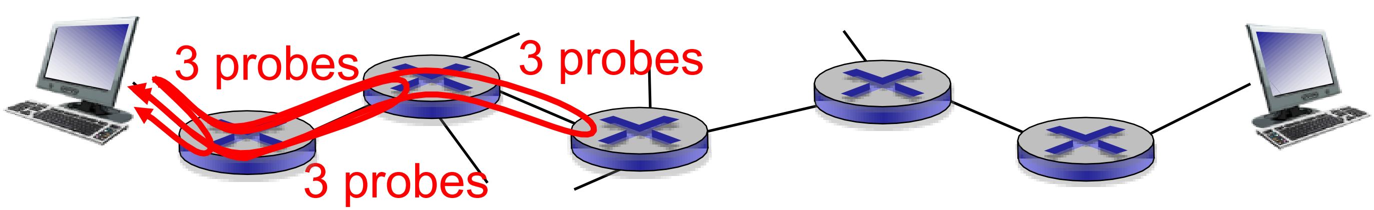
La/R ~ 0

La/R -> 1

* Check online interactive animation on queuing and loss

"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
  - router *i* will return packets to sender
  - sender times interval between transmission and reply.



“Real” Internet delays, routes

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

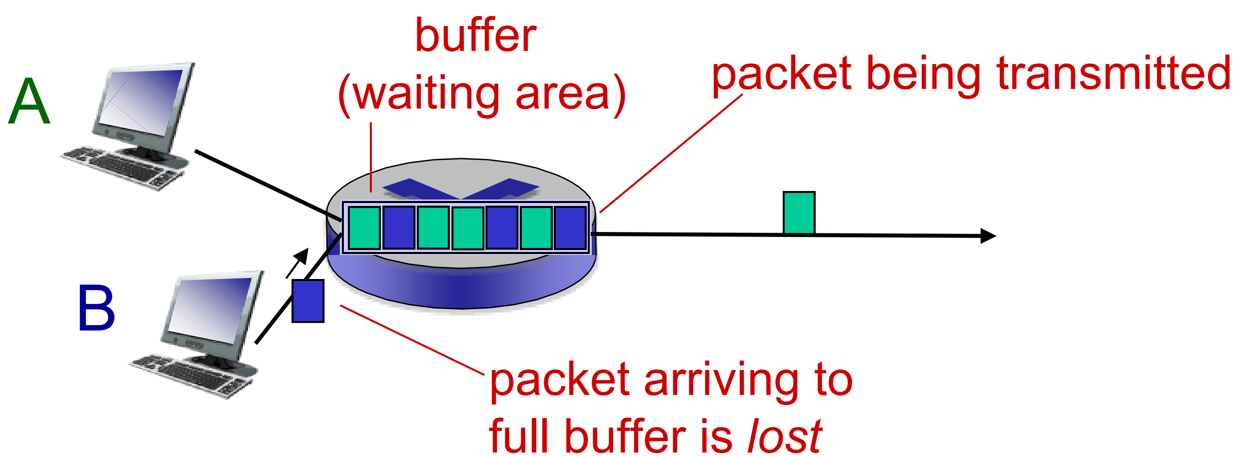
trans-oceanic link

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

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

Packet loss

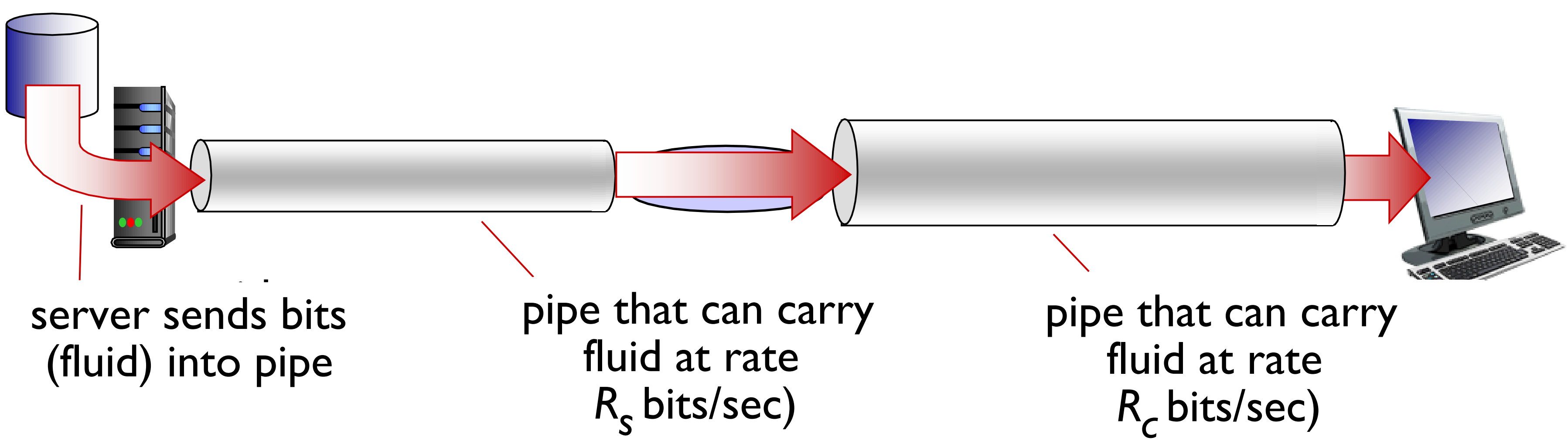
- queue (aka buffer) preceding link in buffer has finite capacity
- packet arriving to full queue dropped (aka lost)
- lost packet may be retransmitted by previous node, by source end system, or not at all



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

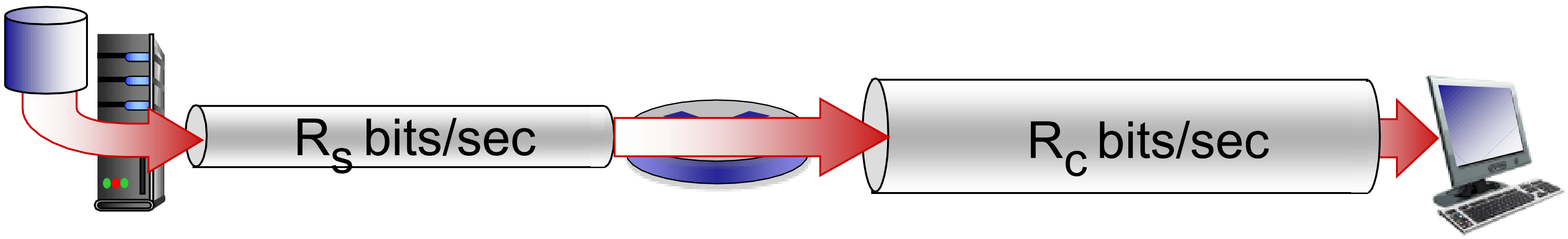
Throughput（吞吐）

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

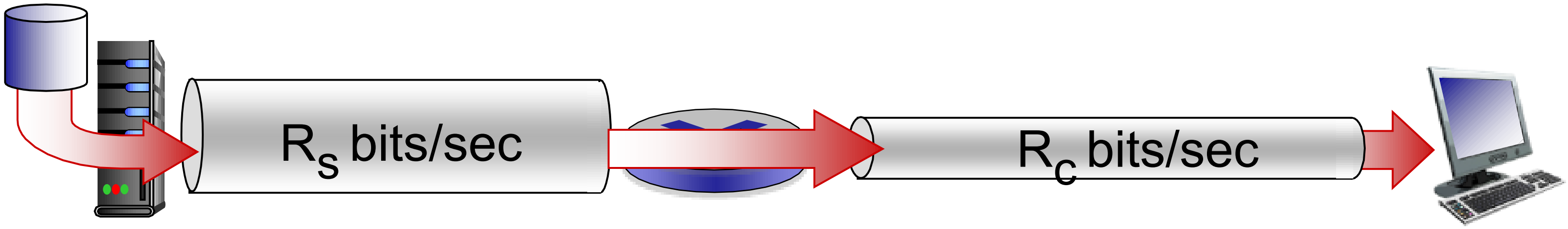


Throughput (more)

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



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

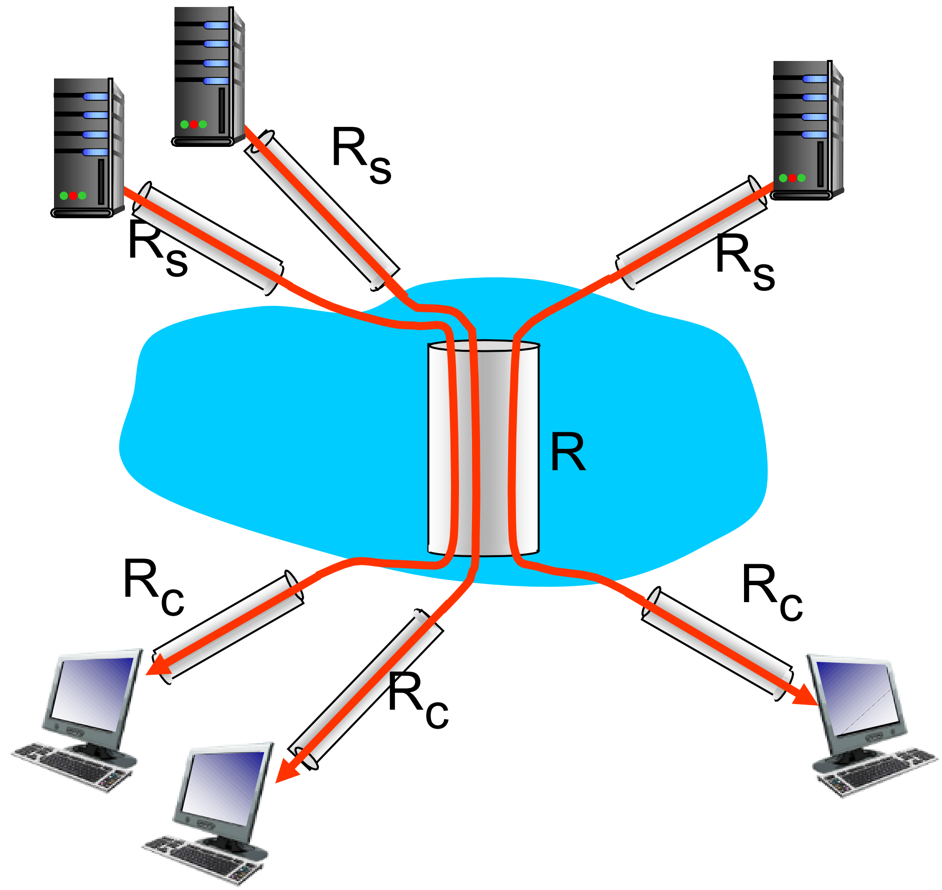


bottleneck link

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

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

课堂作业

- 华东师范大学数据学院和新疆石河子大学的计算机科学与技术学院对口帮扶。
- 我们也鼓励同年级的班级的同学们之间也建立良好的互助关系。
- 假设每个月同学们会互相邮寄一些书籍或者其他学习用品。
- 请说明从华师大数据学院的二年级A班同学张大鹏到新疆石河子大学二年级A班李小明的包裹邮寄和传递过程

Chapter 1: roadmap

1.1 what *is* the Internet?

1.2 network edge

- end systems, access networks, links

1.3 network core

- packet switching, circuit switching, network structure

1.4 delay, loss, throughput in networks

1.5 protocol layers, service models

1.6 networks under attack: security

1.7 history

Protocol "layers"

Networks are complex,
with many "pieces":

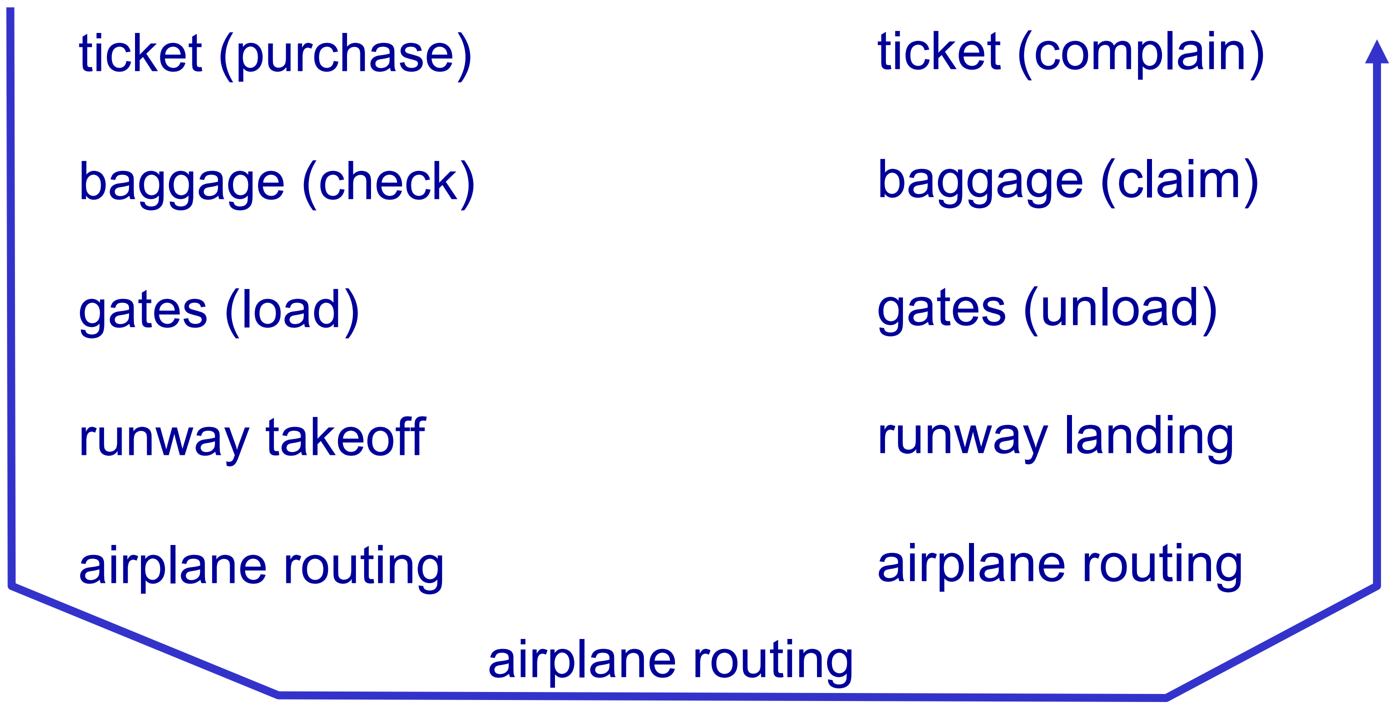
- hosts
- routers
- links of various media
- applications
- protocols
- hardware, software

Question:

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

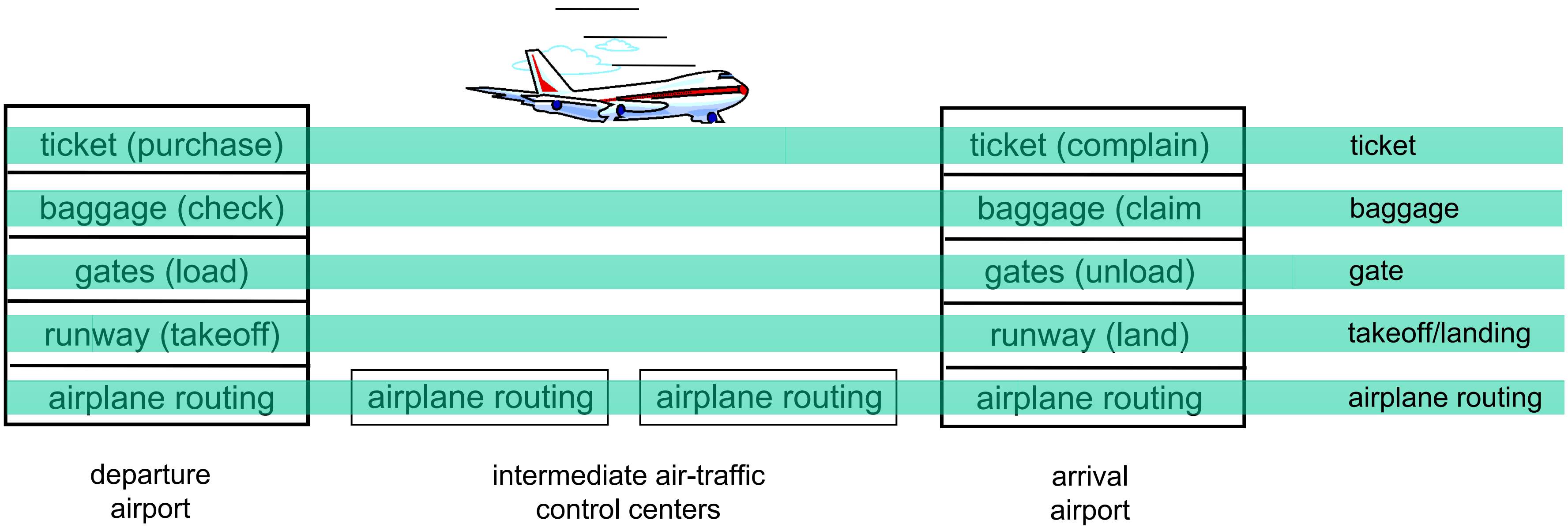
…. or at least our discussion of networks?

Organization of air travel



- a series of steps
- 票务（购买），行李（拖运），登机口（登机），从跑道起飞，按路线飞行，按路线飞行，从跑道着陆，登机口（离机），行李（认领），票务（投诉）

Layering of airline functionality



layers: each layer implements a service

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

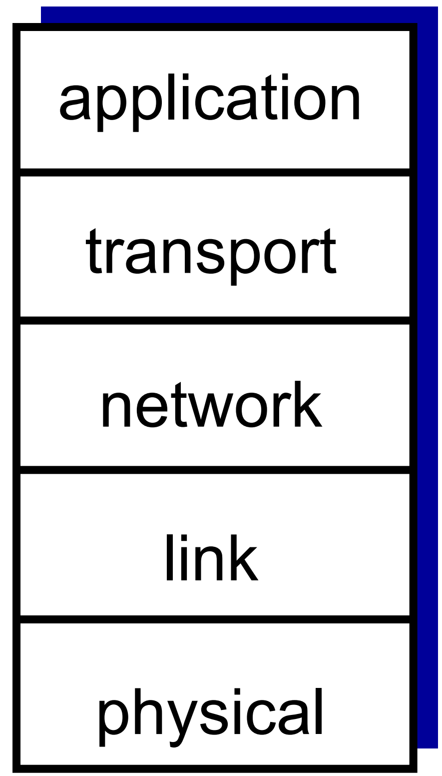
Why layering?

dealing with complex systems:

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

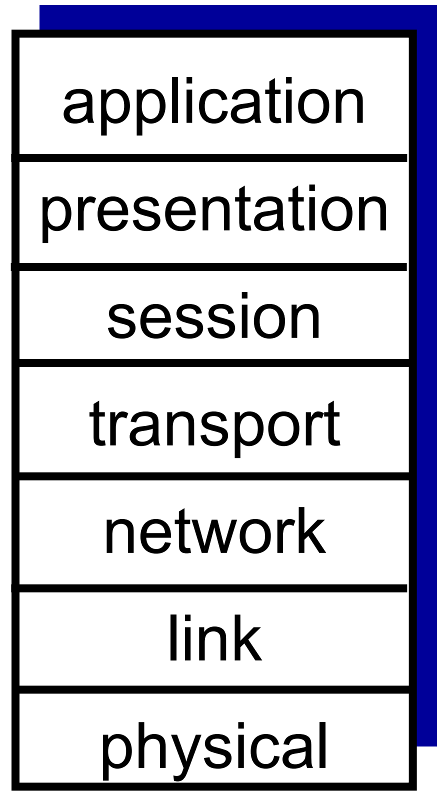
Internet protocol stack（协议栈）

- *application（应用层）*: supporting network applications
  - FTP, SMTP, HTTP 报文
- *transport（传输层）*: process-process data transfer
  - TCP, UDP 报文段
- *network（网络层）*: routing of datagrams from source to destination
  - IP, routing protocols 数据报
- *link（链路层）*: data transfer between neighboring network elements
  - Ethernet, 802.111 (WiFi), PPP
- *physical（物理层）*: bits “on the wire”

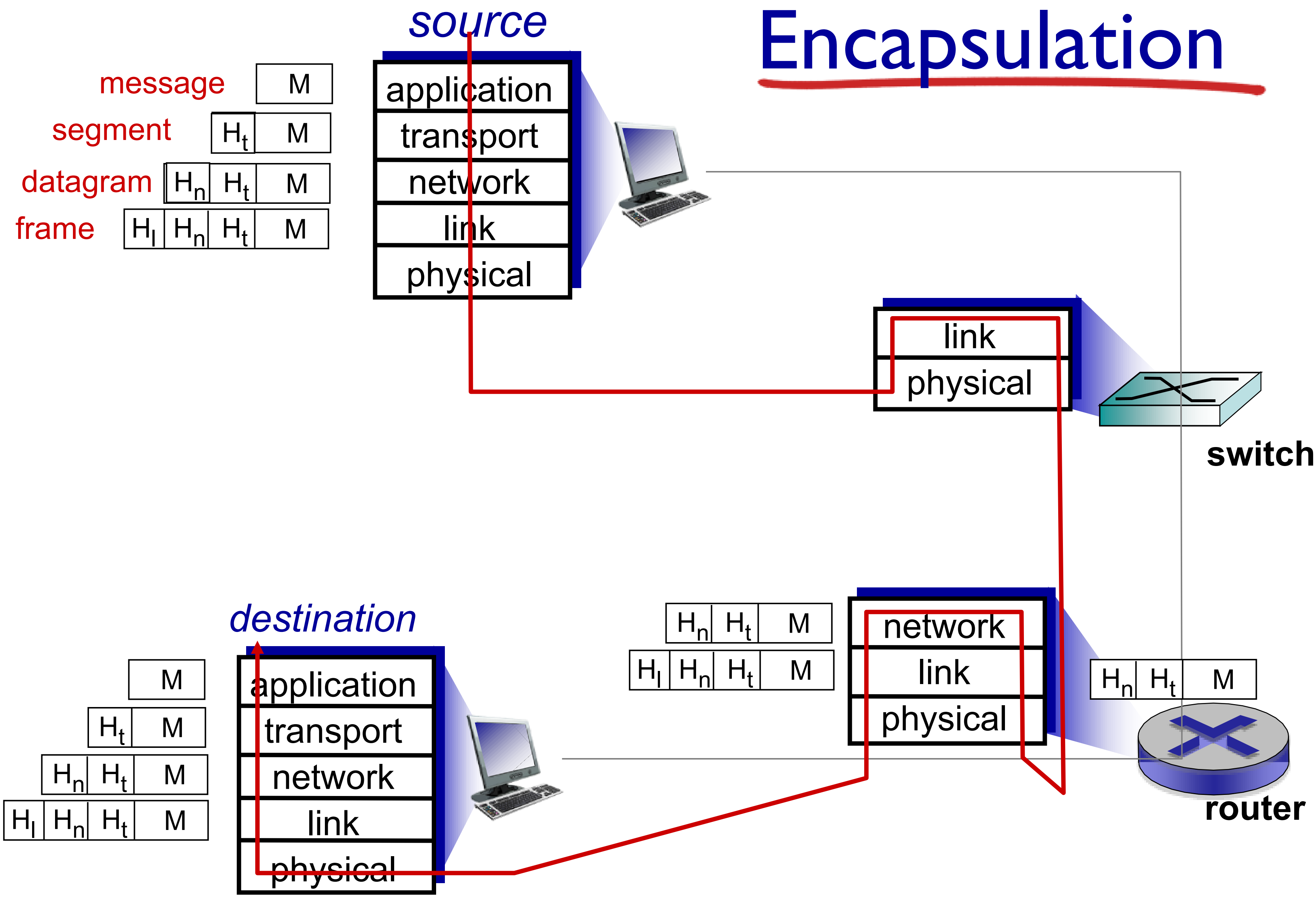


ISO/OSI reference model

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



Encapsulation



Chapter 1: roadmap

1.1 what *is* the Internet?

1.2 network edge

- end systems, access networks, links

1.3 network core

- packet switching, circuit switching, network structure

1.4 delay, loss, throughput in networks

1.5 protocol layers, service models

1.6 networks under attack: security

1.7 history

Network security（网络安全）

- field of network security:
  - how bad guys can attack computer networks
  - how we can defend networks against attacks
  - how to design architectures that are immune to attacks
- Internet not originally designed with (much) security in mind
  - *original vision:* "a group of mutually trusting users attached to a transparent network" ☺
  - Internet protocol designers playing "catch-up"
  - security considerations in all layers!

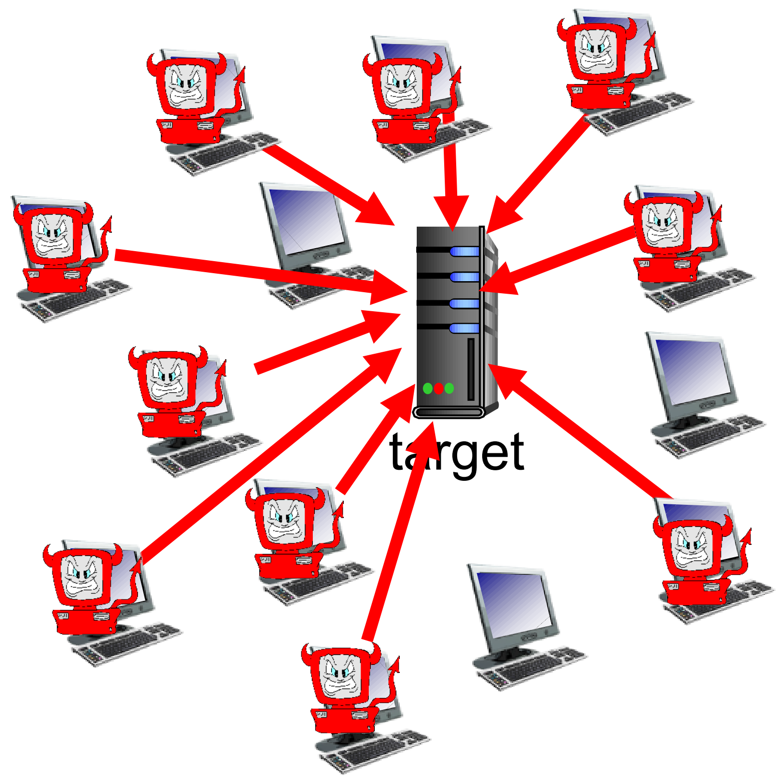
Bad guys: put malware(恶意软件) into hosts via Internet

- malware can get in host from:
  - *virus:* self-replicating infection by receiving/executing object (e.g., e-mail attachment)
  - *worm:* self-replicating infection by passively receiving object that gets itself executed
- spyware malware can record keystrokes, web sites visited, upload info to collection site
- infected host can be enrolled in botnet, used for spam. DDoS attacks

Bad guys: attack server, network infrastructure

Denial of Service (DoS) *拒绝服务攻击*: attackers make resources (server, bandwidth) unavailable to legitimate traffic by overwhelming resource with bogus traffic

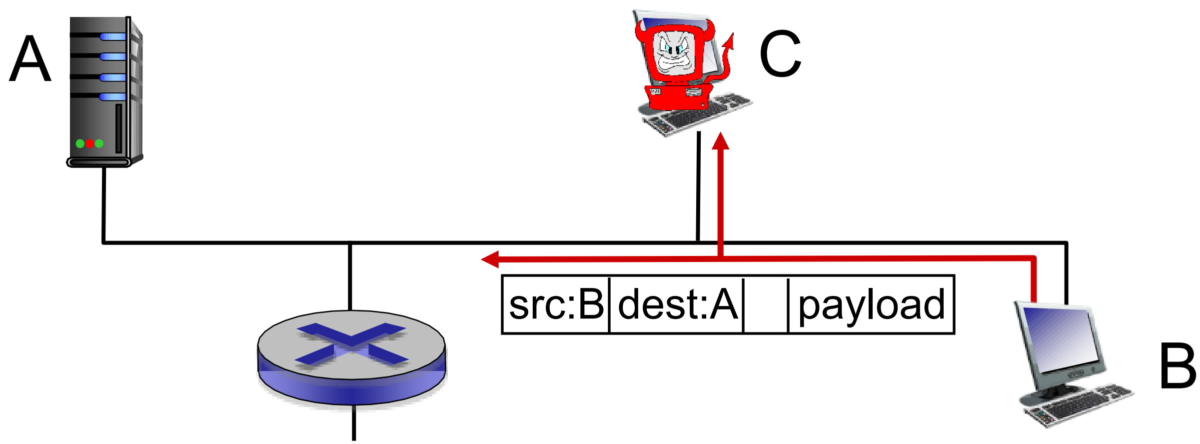
1. select target
2. break into hosts around the network (see botnet 僵尸网络)
3. send packets to target from compromised hosts



Bad guys can sniff packets

packet "sniffing"（分组嗅探）:

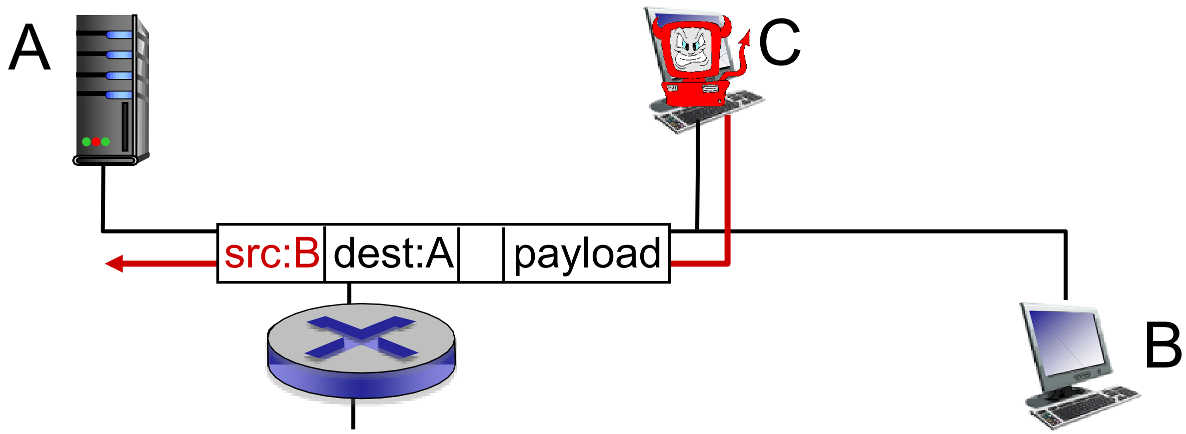
- broadcast media (shared Ethernet, wireless)
- promiscuous network interface reads/records all packets (e.g., including passwords!) passing by



- wireshark software used for end-of-chapter labs is a (free) packet-sniffer

Bad guys can use fake addresses

IP spoofing（IP 哄骗）: send packet with false source address



… *lots more on security (throughout, Chapter 8)*

Chapter 1: roadmap

1.1 what *is* the Internet?

1.2 network edge

- end systems, access networks, links

1.3 network core

- packet switching, circuit switching, network structure

1.4 delay, loss, throughput in networks

1.5 protocol layers, service models

1.6 networks under attack: security

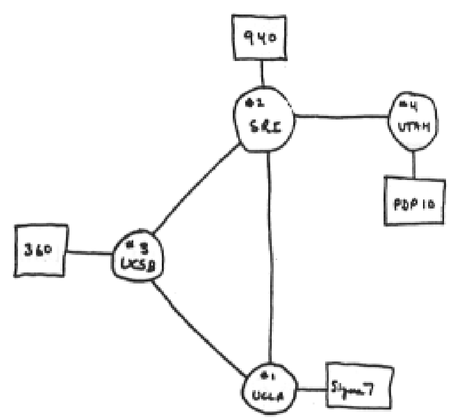
1.7 history

Internet history

1961-1972: Early packet-switching principles

- 1961: Kleinrock - queueing theory shows effectiveness of packet-switching
- 1964: Baran - packet-switching in military nets
- 1967: ARPAnet conceived by Advanced Research Projects Agency
- 1969: first ARPAnet node operational

- 1972:
  - ARPAnet public demo
  - NCP (Network Control Protocol) first host-host protocol
  - first e-mail program
  - ARPAnet has 15 nodes



THE ARPA NETWORK

Internet history

1972-1980: Internetworking, new and proprietary nets

- 1970: ALOHAnet satellite network in Hawaii
- 1974: Cerf and Kahn - architecture for interconnecting networks
- 1976: Ethernet at Xerox PARC
- late70's: proprietary architectures: DECnet, SNA, XNA
- late 70's: switching fixed length packets (ATM precursor)
- 1979: ARPAnet has 200 nodes

Cerf and Kahn's internetworking principles:

- minimalism, autonomy - no internal changes required to interconnect networks
- best effort service model
- stateless routers
- decentralized control

define today's Internet architecture

Internet history

1980-1990: new protocols, a proliferation of networks

- 1983: deployment of TCP/IP
- 1982: smtp e-mail protocol defined
- 1983: DNS defined for name-to-IP-address translation
- 1985: ftp protocol defined
- 1988: TCP congestion control

- new national networks: CSnet, BITnet, NSFnet, Minitel
- 100,000 hosts connected to confederation of networks

Internet history

1990, 2000's: commercialization, the Web, new apps

- early 1990's: ARPAnet decommissioned
- 1991: NSF lifts restrictions on commercial use of NSFnet (decommissioned, 1995)
- early 1990s: Web
  - hypertext [Bush 1945, Nelson 1960's]
  - HTML, HTTP: Berners-Lee
  - 1994: Mosaic, later Netscape
  - late 1990's: commercialization of the Web

late 1990's – 2000's:

- more killer apps: instant messaging, P2P file sharing
- network security to forefront
- est. 50 million host, 100 million+ users
- backbone links running at Gbps

Internet history

2005-present

- ~5B devices attached to Internet (2016)
  - smartphones and tablets
- aggressive deployment of broadband access
- increasing ubiquity of high-speed wireless access
- emergence of online social networks:
  - Facebook: ~ one billion users
- service providers (Google, Microsoft) create their own networks
  - bypass Internet, providing "instantaneous" access to search, video content, email, etc.
- e-commerce, universities, enterprises running their services in "cloud" (e.g., Amazon EC2)

Introduction: summary

covered a "ton" of material!

- Internet overview
- what's a protocol?
- network edge, core, access network
  - packet-switching versus circuit-switching
  - Internet structure
- performance: loss, delay, throughput
- layering, service models
- security
- history

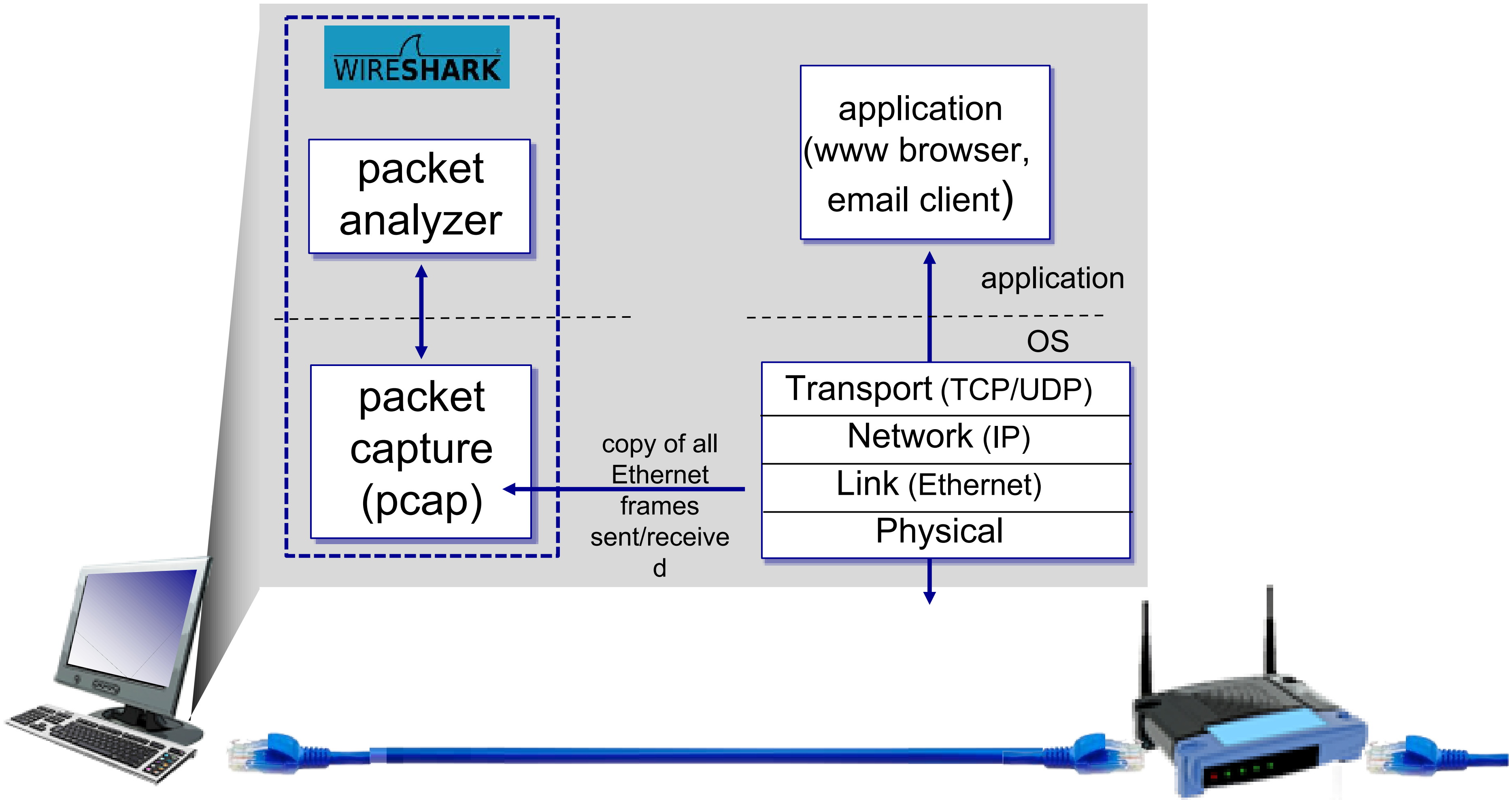
you now have:

- context, overview, "feel" of networking
- more depth, detail *to follow!*

课后习题

- P1，P3，P9，P12, P20，P22
- 在下周五上课之前在钉钉群以作业的方式提交

Chapter 1
Additional Slides


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