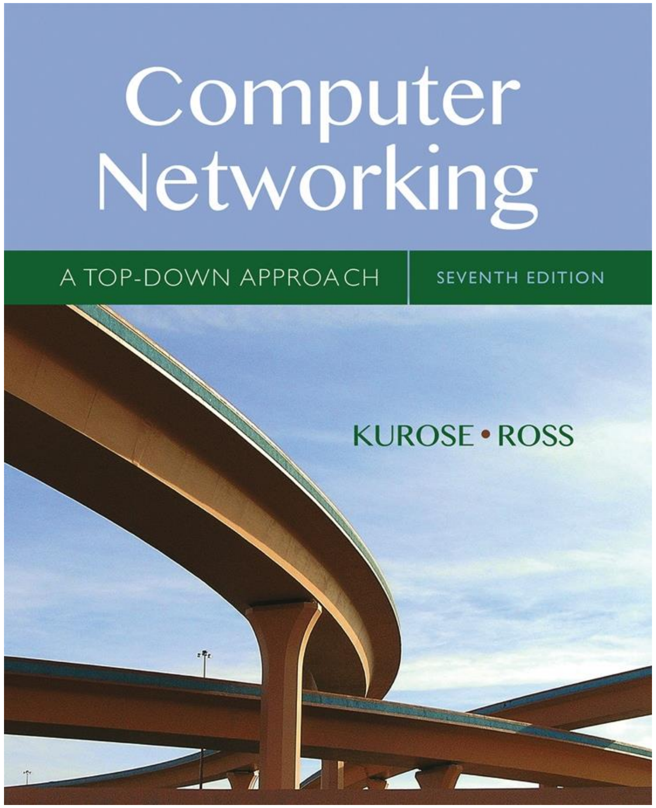
Internet Protocols EBU5403

Michael Chai (michael.chai@qmul.ac.uk)
Richard Clegg (r.clegg@qmul.ac.uk)
Cunhua Pan (c.pan@qmul.ac.uk)

	Week 1	Week 2	Week 3	Week 4
Group 1	Michael	Cunhua	Michael	Cunhua
Group 2	Richard			
Group 3	Michael	Cunhua	Michael	Cunhua

Week 1 Introduction

Many of these slides (and the theme) come from the course text book by Jim Kurose and Keith Ross
The original slides are freely available to download online.



Computer Networking: A Top Down Approach

7th edition
Jim Kurose, Keith Ross
Pearson/Addison Wesley
April 2016

Structure of course

- Week 1
  - Introduction to IP Networks
  - The Transport layer (part 1)
- Week 2
  - The Transport layer (part II)
  - The Network layer (part I)
  - Class test
- Week 3
  - The Network layer (part II)
  - The Data link layer (part I)
  - Router lab tutorial (assessed lab work after this week)
- Week 4
  - The Data link layer (part II)
  - Network management and security
  - Class test

Week 1: IP networks introduction

our goal:

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

overview:

- what's the Internet?
- what's a protocol?
- network edge; hosts, access net, physical media
- network core: packet/circuit switching, Internet structure
- performance: loss, delay, throughput
- protocol layers, service models
- history

Introduction to IP: roadmap

1.1 what *is* the Internet?

1.2 network edge

- end systems, access networks, links

1.3 network core

- network structure

1.4 delay, loss, throughput in networks

1.5 protocol layers, service models

1.6 history

Lecture 1

Lecture 2

Introduction to IP: roadmap

1.1 what *is* the Internet?

1.2 network edge

- end systems, access networks, links

1.3 network core

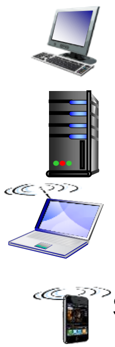
- network structure

1.4 delay, loss, throughput in networks

1.5 protocol layers, service models

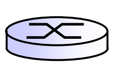
1.6 history

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

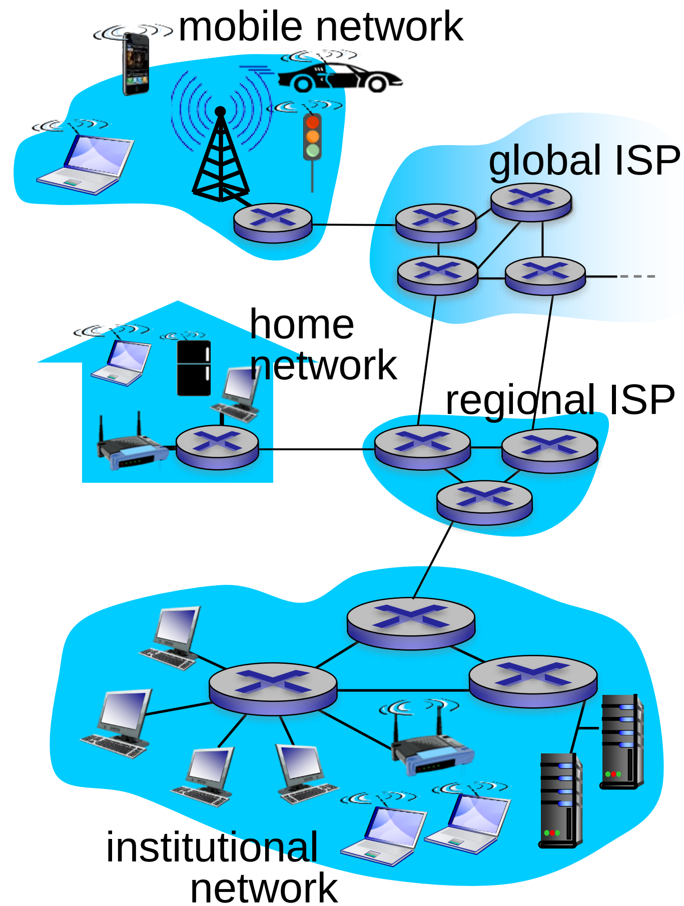


- billions of connected computing devices:
  - *hosts* = *end systems*
  - running *network apps*

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



- *packet switches:* forward packets (chunks of data)
  - *routers* and *switches*



ISP (Internet Service Provider)

"Fun" Internet-connected devices

IP picture frame

Web-enabled toaster + weather forecaster

Tweet-a-watt: monitor energy use

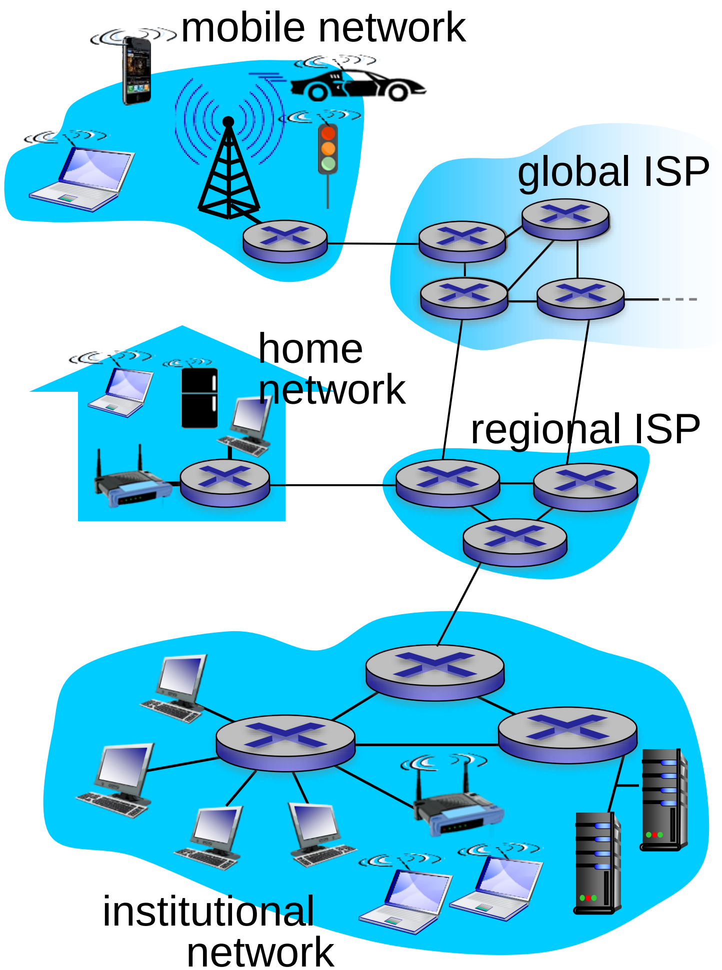
Internet refrigerator

sensorized, bed mattress

Internet phones

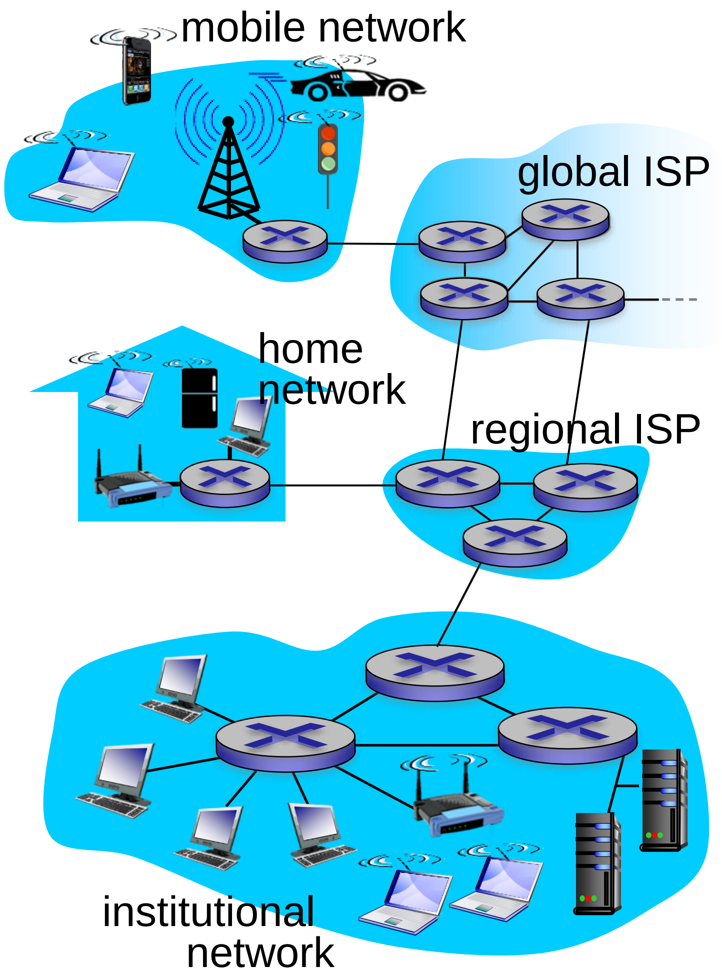
What's the Internet: "nuts and bolts" view

- *Internet:* "network of networks"
  - Interconnected Internet Service Providers (ISPs)
- *protocols* control sending, receiving of messages
  - TCP (Transmission Control Protocol)
  - IP (Internet Protocol)
  - HTTP (HyperText Transfer Protocol)
  - 802.11 (WiFi standard)
  - Many many more
- *Internet standards*
  - RFC: Request for comments
  - IETF: Internet Engineering Task Force



What's the Internet: a service view

- *infrastructure that provides services to applications:*
  - Web, VoIP (Voice over IP), 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?

human protocols:

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

… specific messages sent

… specific actions taken when messages received, or other events

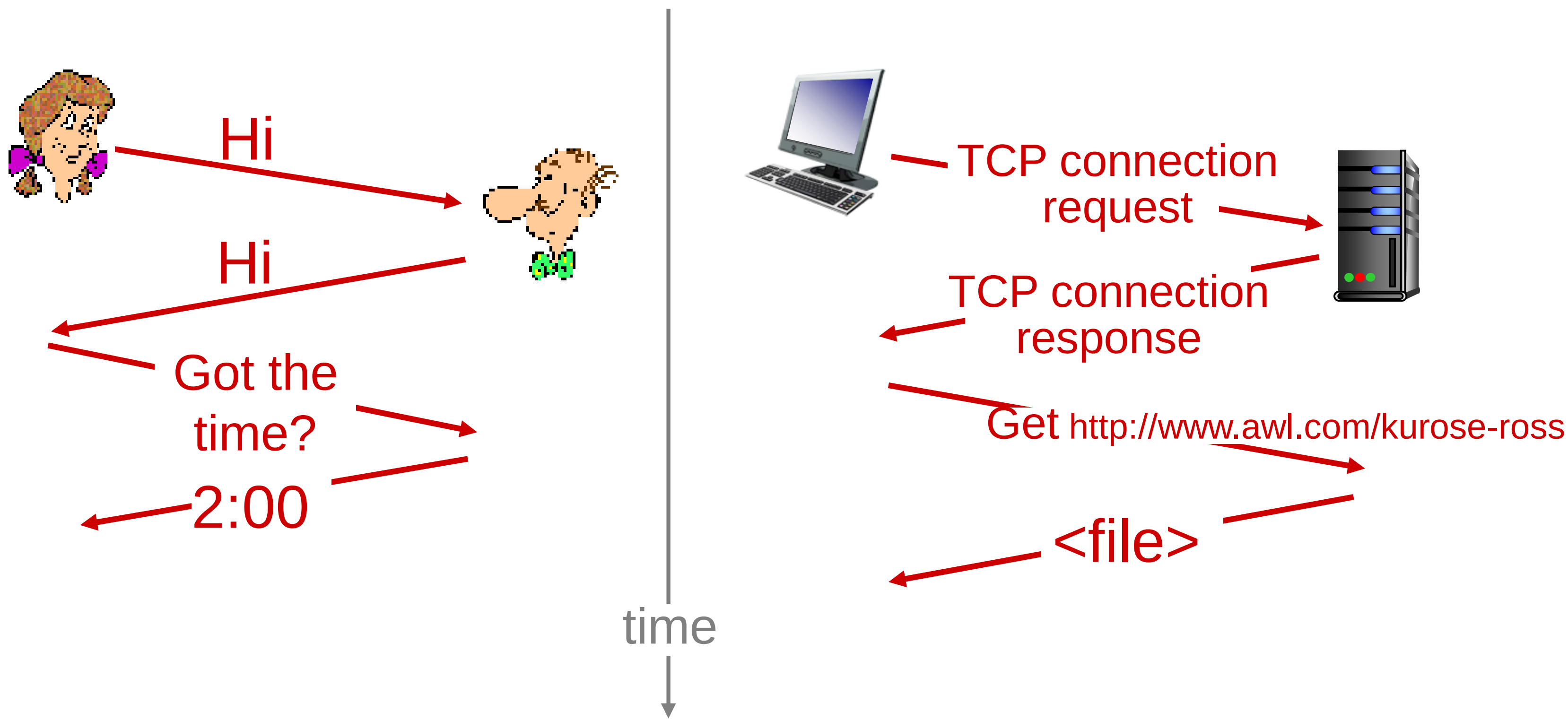
network protocols:

- 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?

Introduction to IP: roadmap

1.1 what *is* the Internet?

1.2 network edge

- end systems, access networks, links

1.3 network core

- network structure

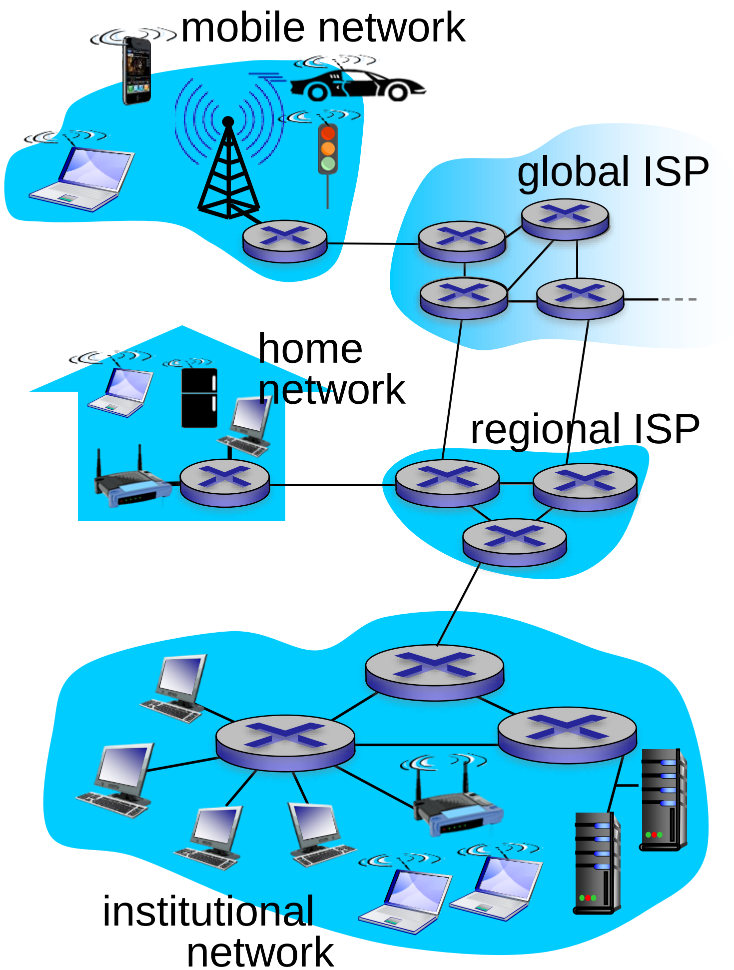
1.4 delay, loss, throughput in networks

1.5 protocol layers, service models

1.6 history

A closer look at network structure:

- *network edge:*
  - hosts: clients and servers
  - servers often in data centers
  - part of the network with the users and computers
- *access networks, physical media:* wired, wireless communication links
  - part of network connecting edge to rest of network
- *network core:*
  - the "middle" of the network
  - interconnected routers
  - network of networks



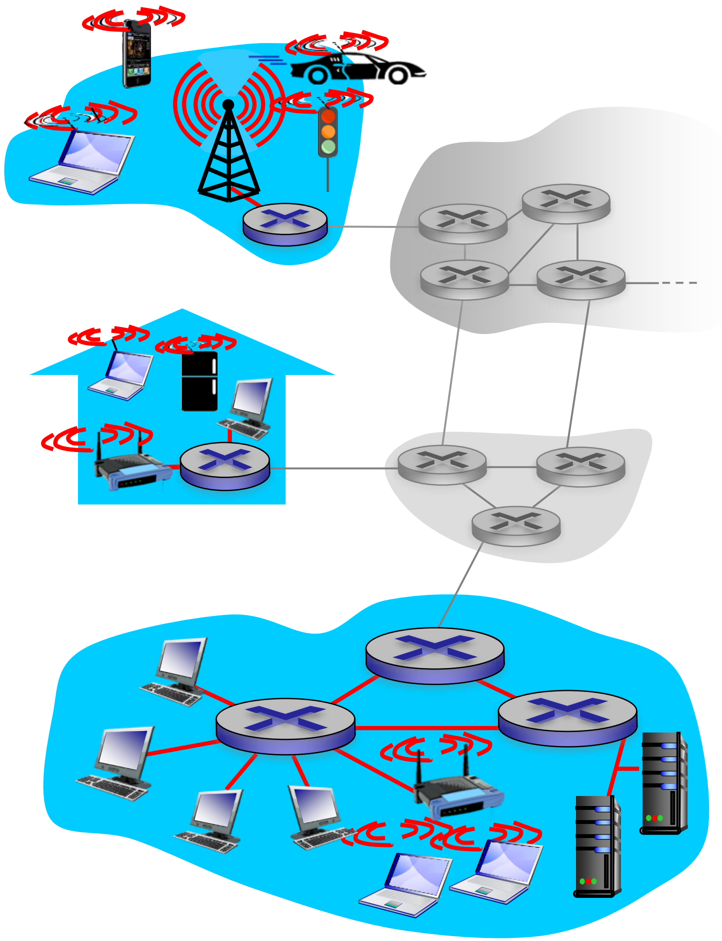
Access networks and physical media

Q: How to connect end systems to edge router?

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

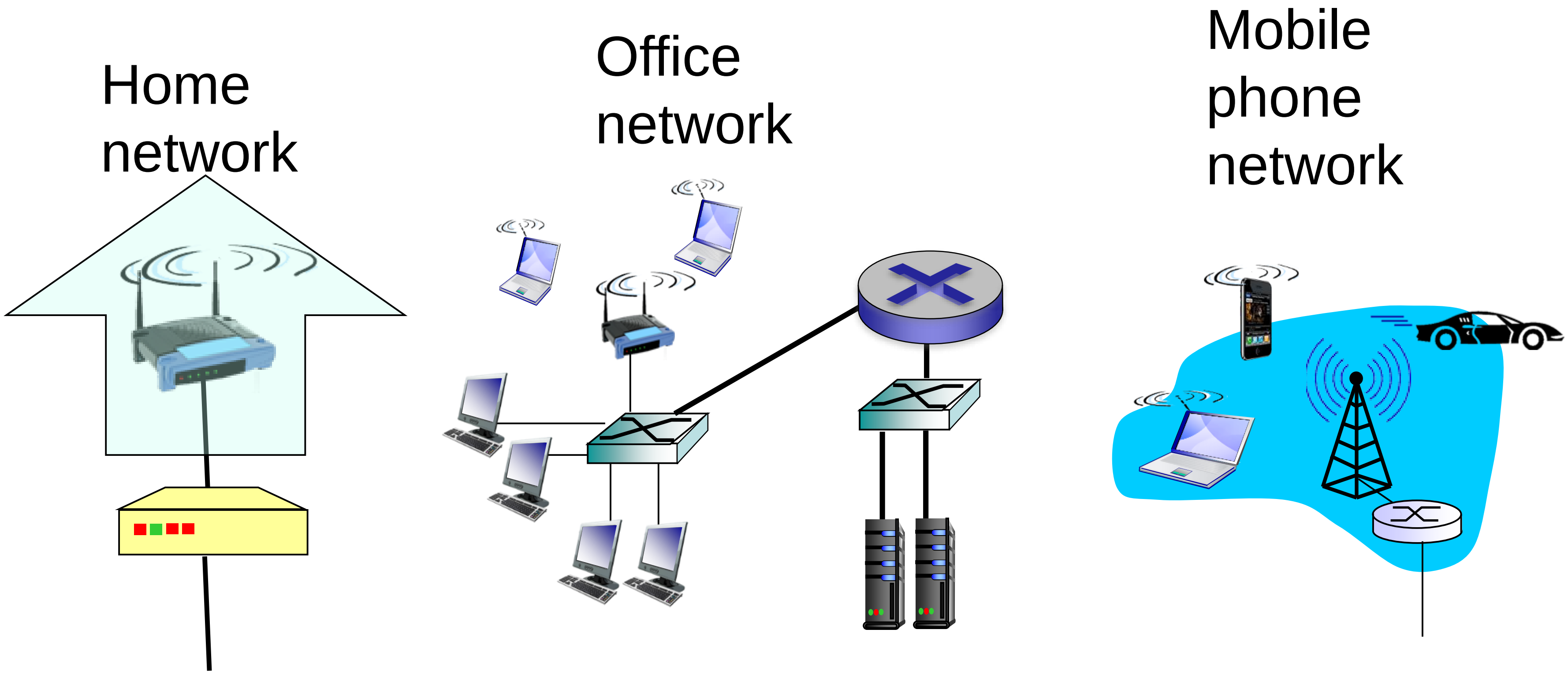
keep in mind:

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



Access networks:

- These networks connect end users to the rest of the internet.



Network basics bits and bytes

- A bit is a "binary digit" – a single 0 or 1.
- A byte is a group of eight bits – can be thought of as a number from 0 to 255 (or -128 to 127) or as two digits of "hexadecimal" (eg A0, FF, 10).
- Amounts of data are usually specified in bytes.
  - 1KB = 1 kilobyte = 1000 bytes = 8000bits
  - 1MB = 1 megabyte = 1 million bytes = 8 million bits
  - 1GB = 1 gigabyte = 1000 million bytes = 8000 million bits
- BUT speeds are usually in bits per second (not bytes)
  - 1b/s (or bps) = 1 bit per second
  - 1Kb/s = 1000 bits per second
  - 1Mb/s = 1 million bits per second
  - 1Gb/s = 1000 million bits per second

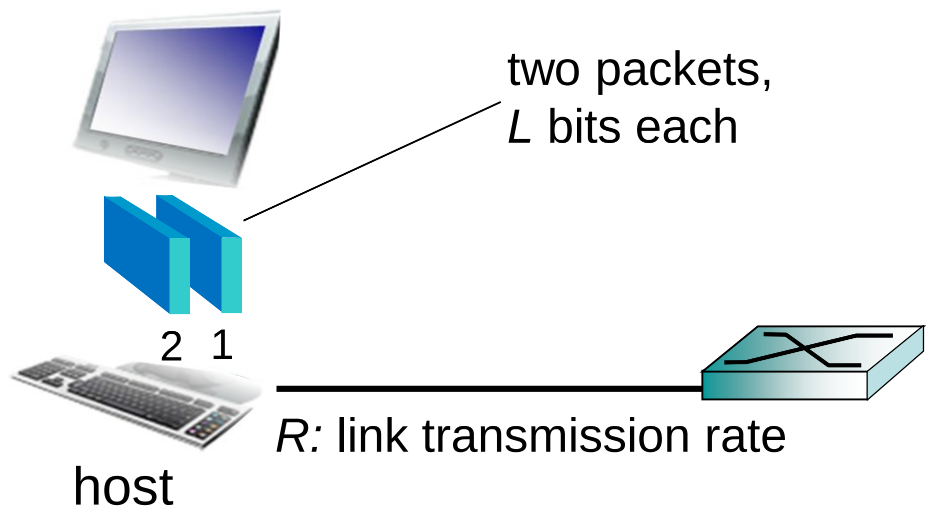
Network basics – packets

- Imagine we want to transmit 10GB of data.
- This may take several hours to send.
- It is useful to split the data down into smaller units (known as packets).
- This "packet" of data can be sent relatively quickly.
- The packet can be, for example, checked for errors.
- The packet could be retransmitted if a problem was detected.

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*



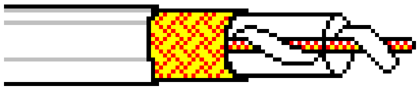
$$\text{packet transmission delay} = \text{time needed to transmit } L\text{-bit packet into link} = \frac{L \text{ (bits)}}{R \text{ (bits/sec)}}$$

Physical media

- Physical media represent the actual hardware that carries the data. Think about physical wires or radio transmissions (wireless).

Ethernet cable

Fibre optic



Coaxial cable

WiFi (802.11)

3/4/5G mobile (mobile phone)

Satellite

Introduction to IP: roadmap

1.1 what *is* the Internet?

1.2 network edge

- end systems, access networks, links

1.3 network core

- network structure

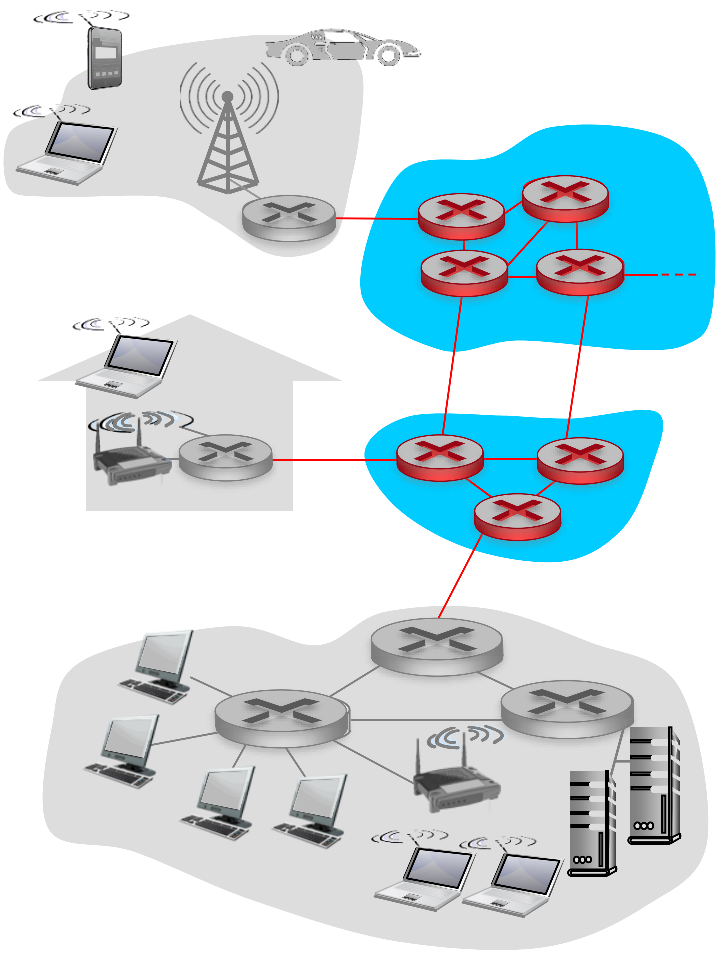
1.4 delay, loss, throughput in networks

1.5 protocol layers, service models

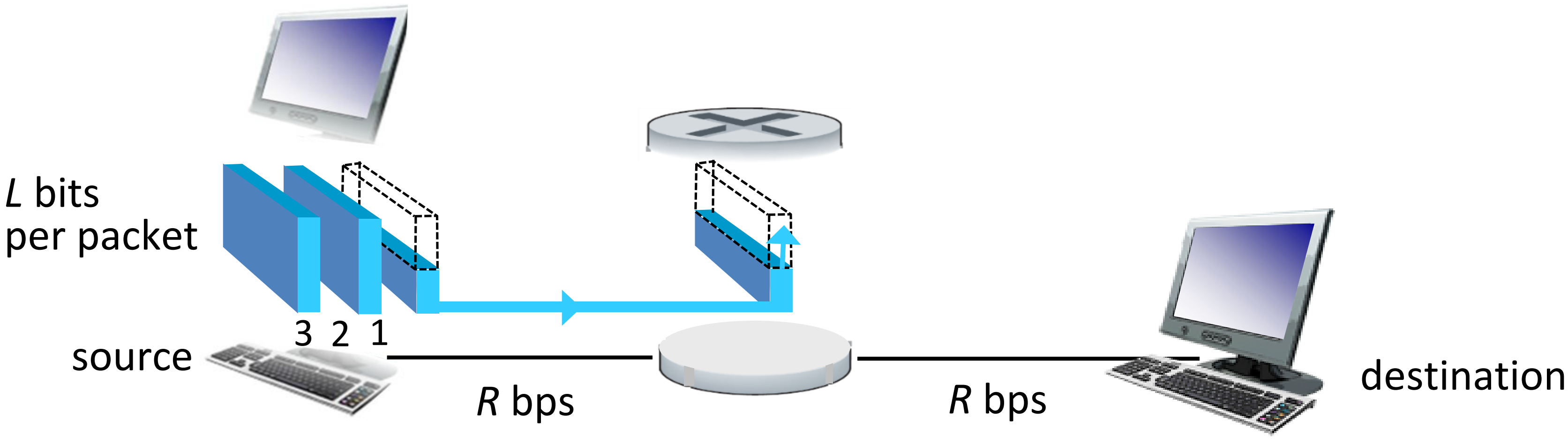
1.6 history

The network core

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



Packet-switching: store-and-forward



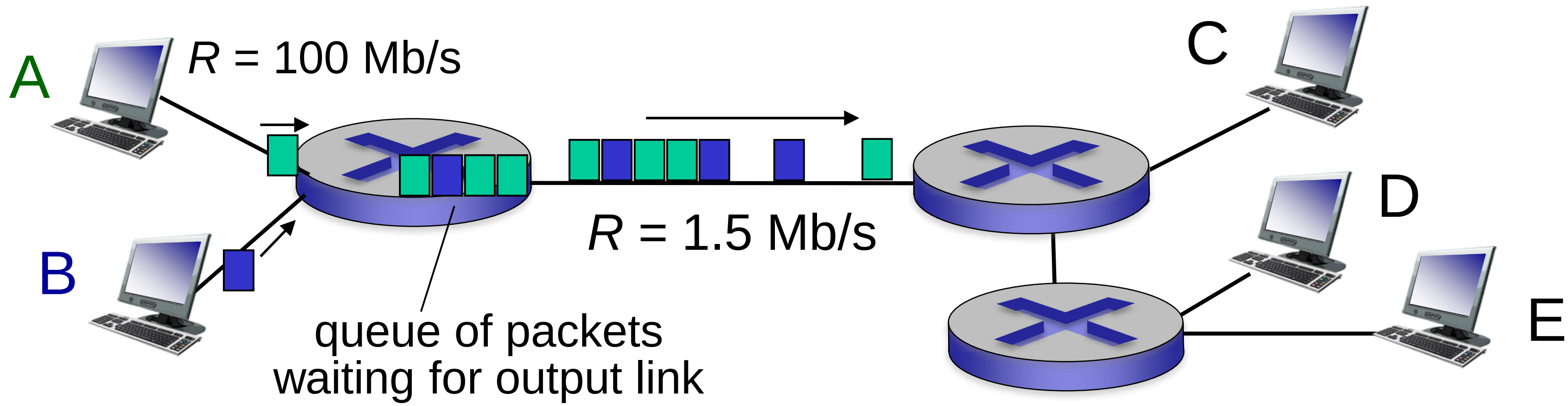
- takes *L*/*R* seconds to transmit (push out) *L*-bit packet into link at *R* bps
- *store and forward:* entire packet must arrive at router before it can be transmitted on next link
- end-end delay = 2*L*/*R* (assuming zero propagation delay)

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



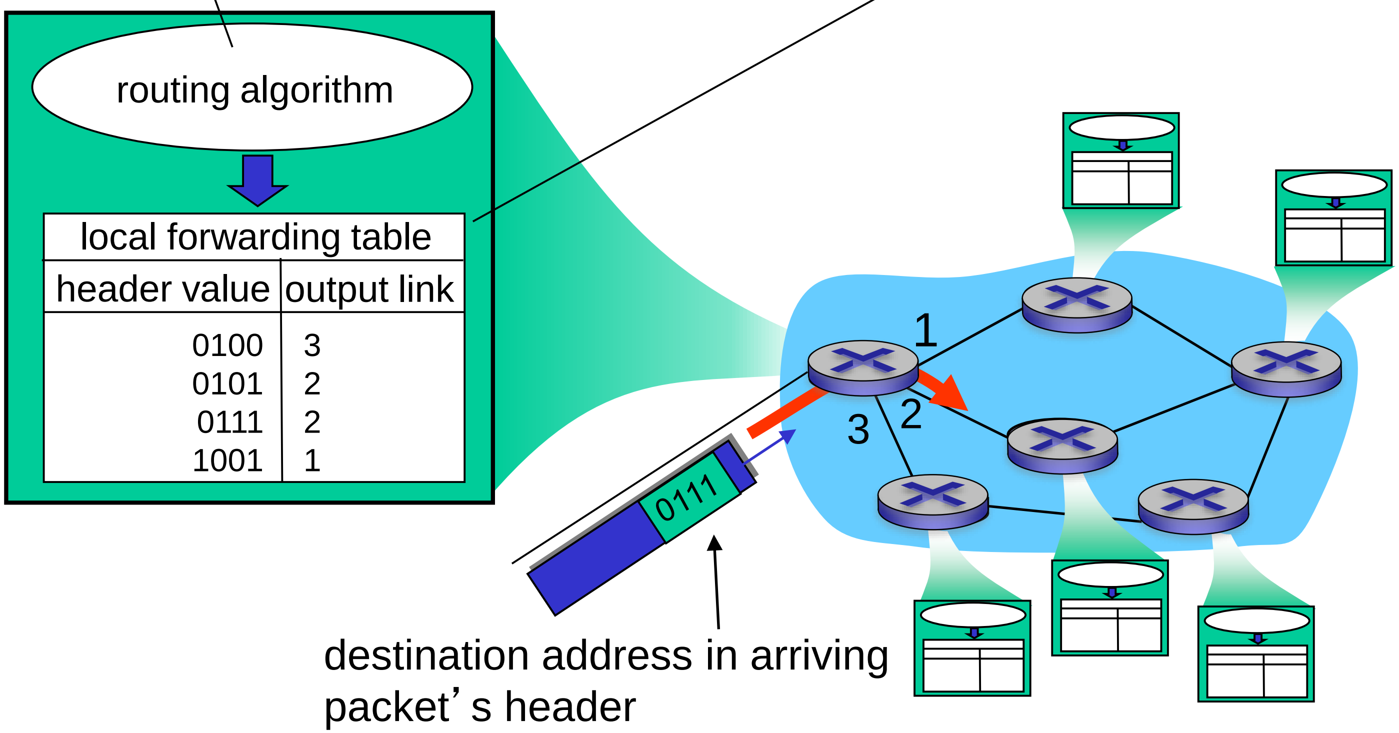
queuing and loss:

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

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

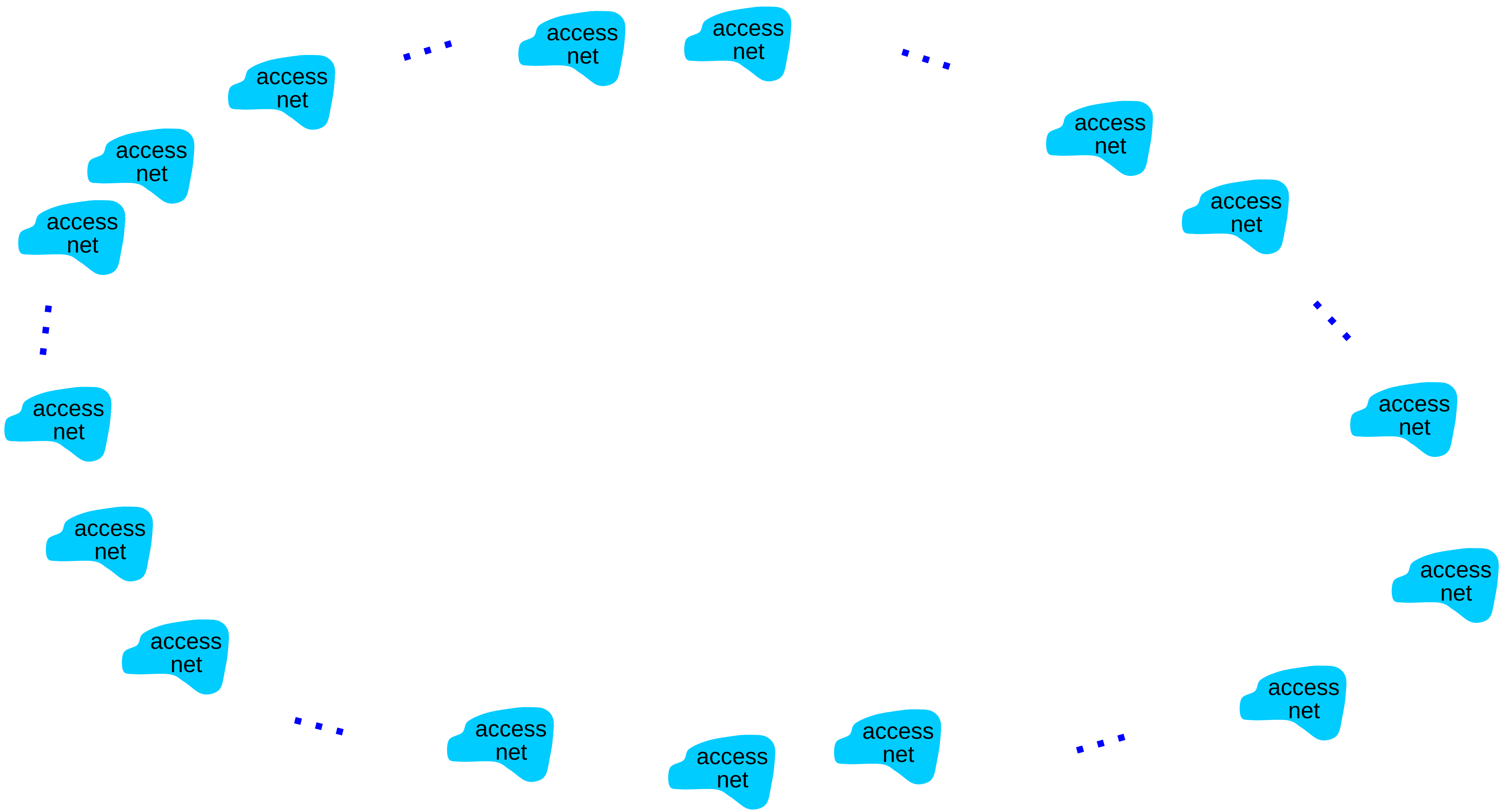


Internet structure: network of networks

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

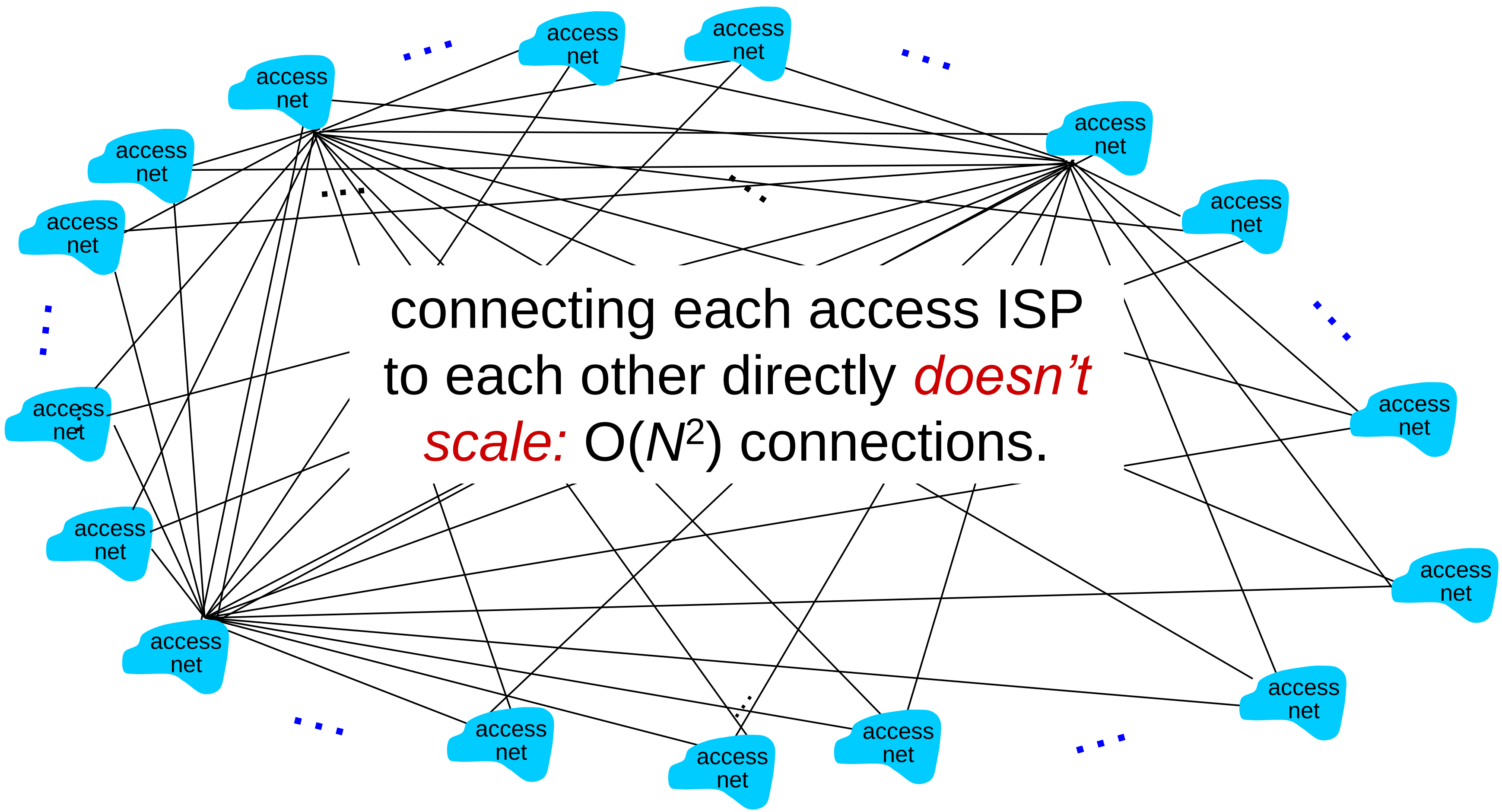
Internet structure: network of networks

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



Internet structure: network of networks

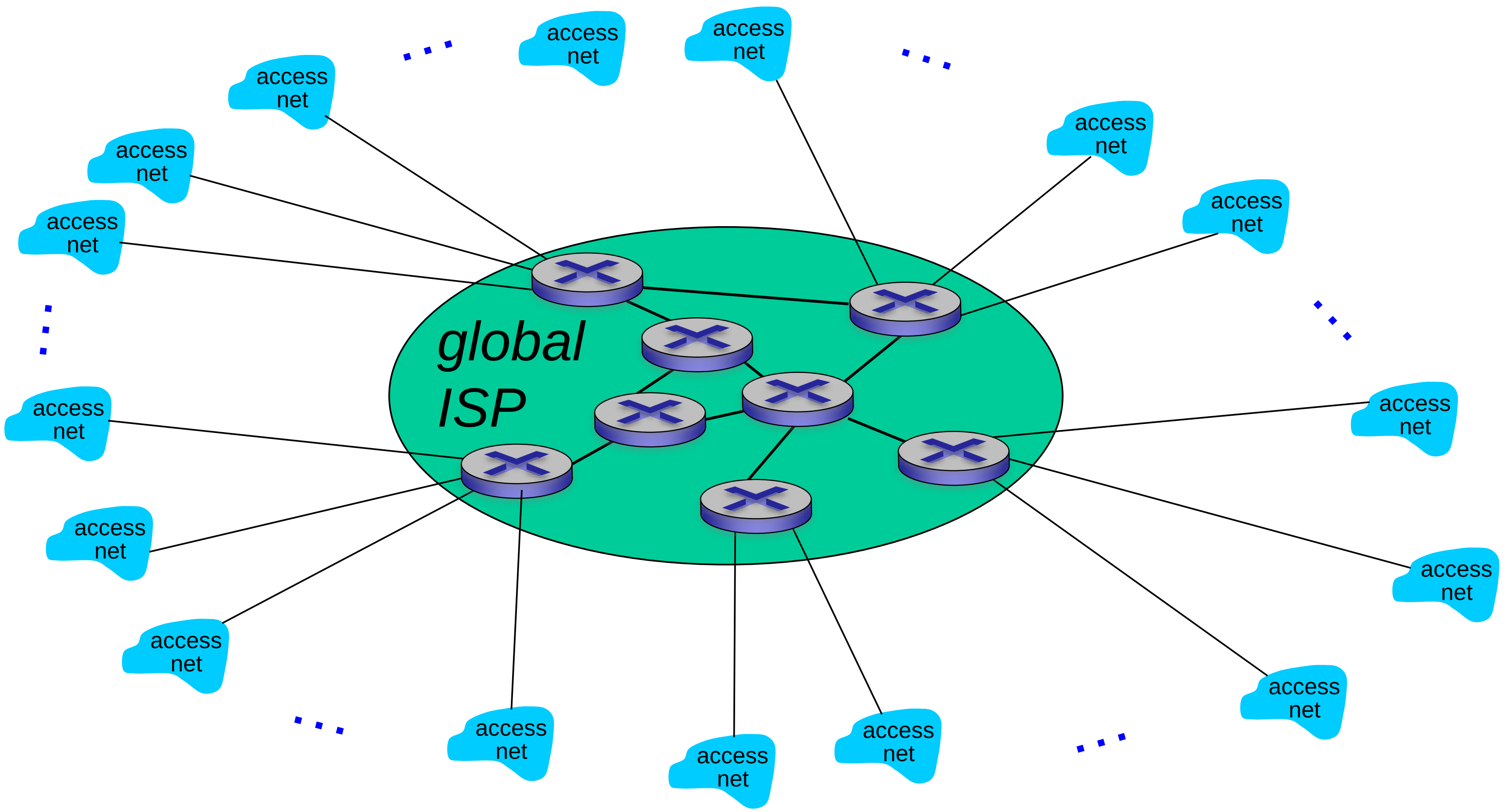
Option: connect each access ISP to every other access ISP?



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

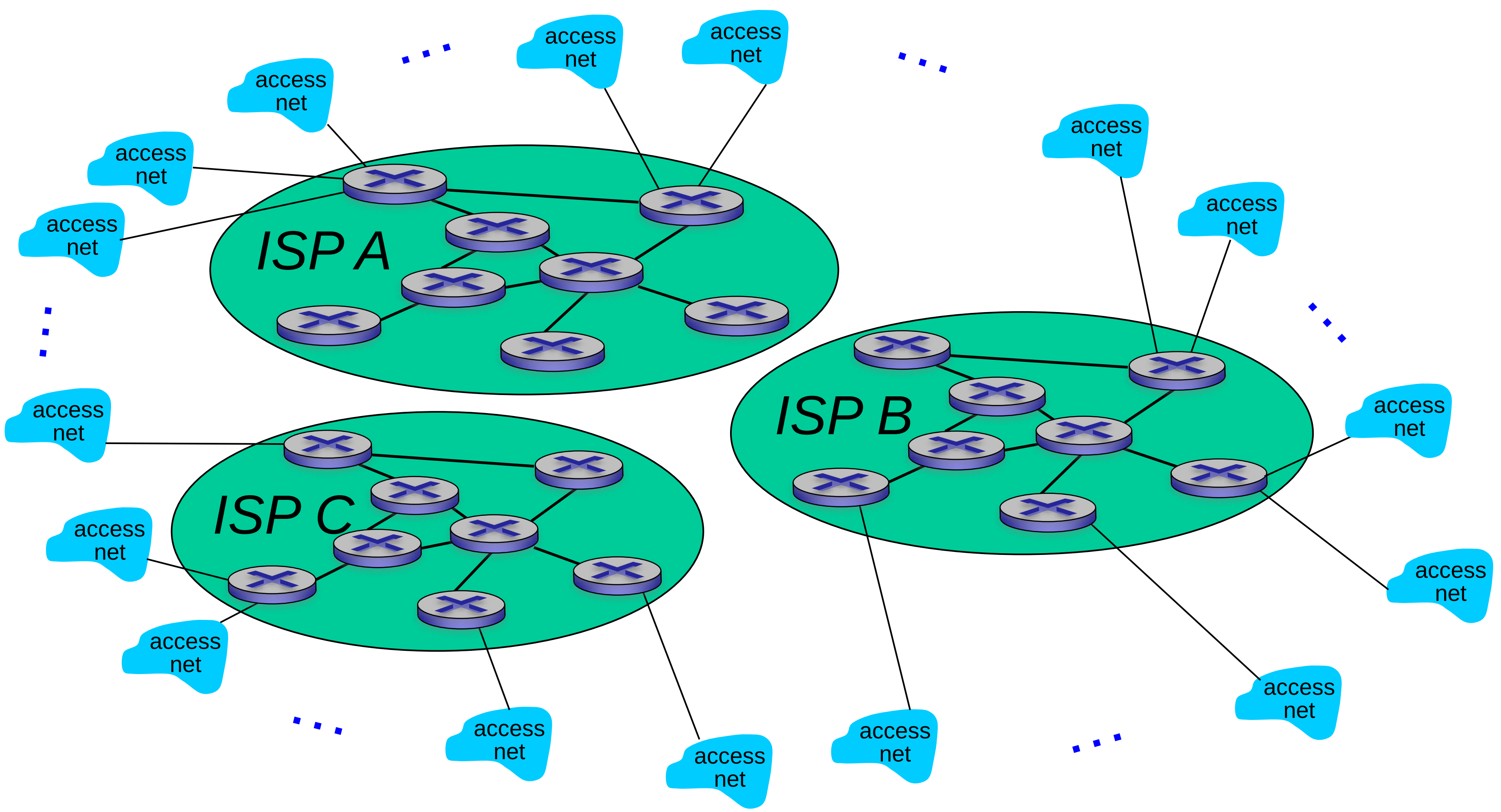
Internet structure: network of networks

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



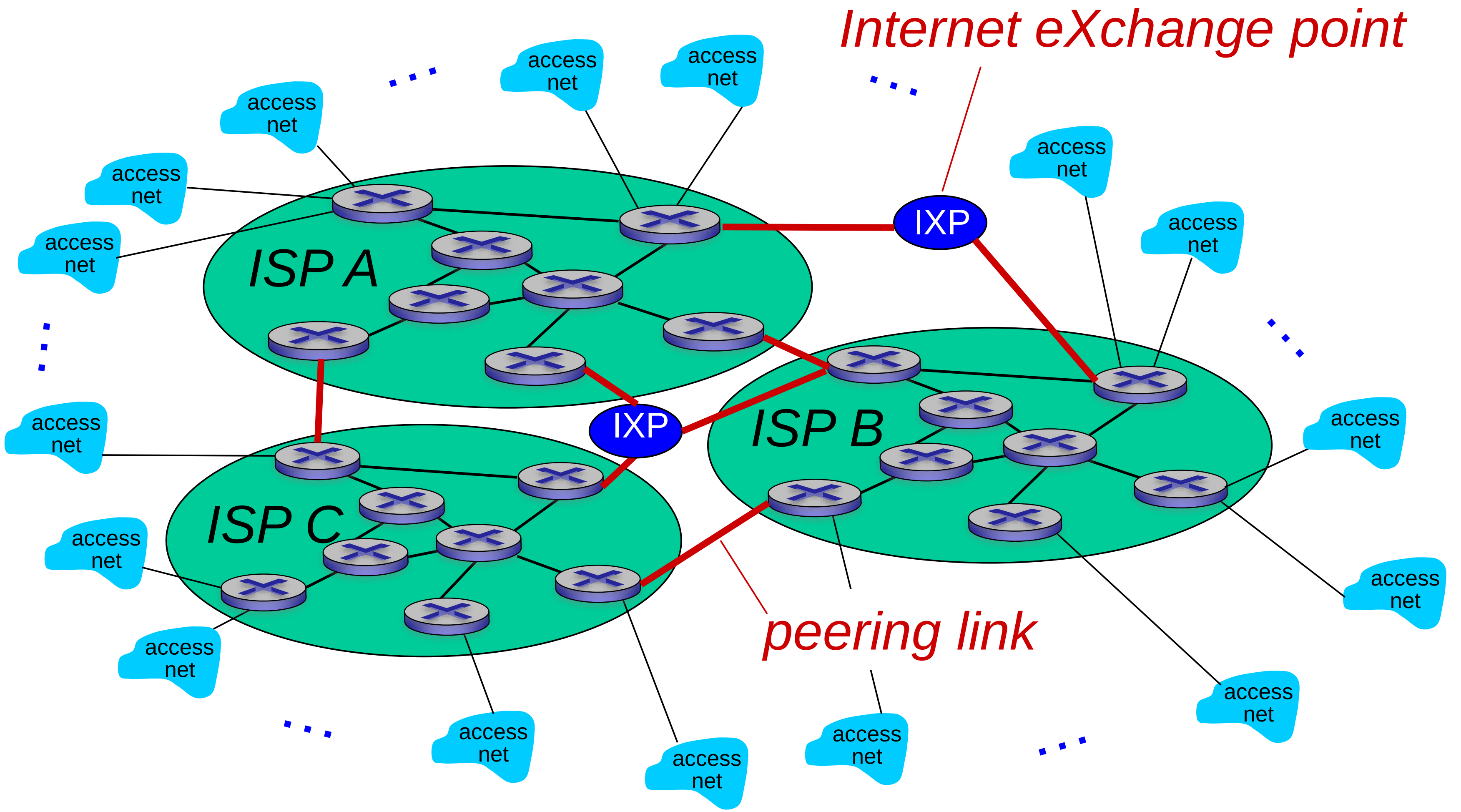
Internet structure: network of networks

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



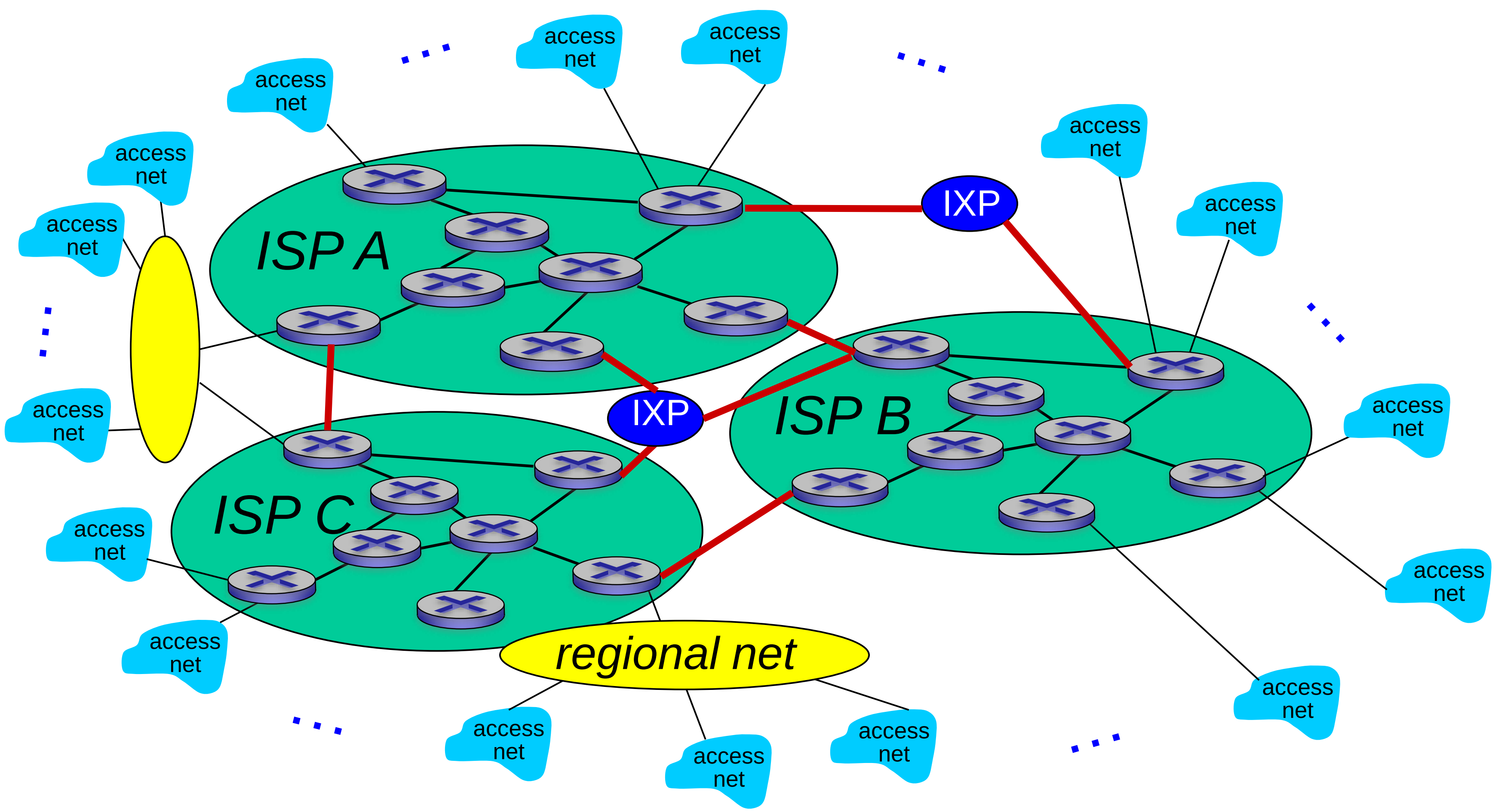
Internet structure: network of networks

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



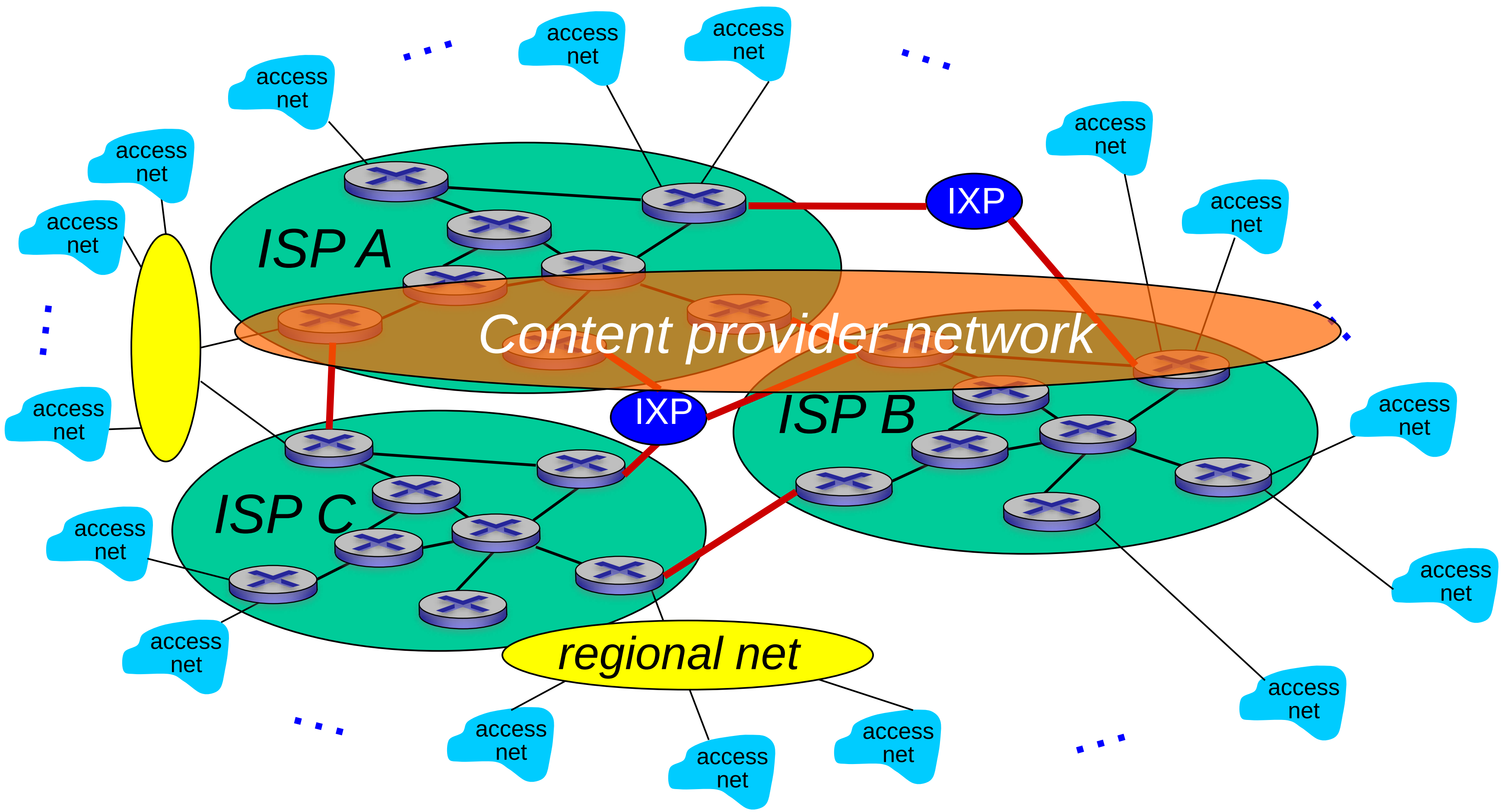
Internet structure: network of networks

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

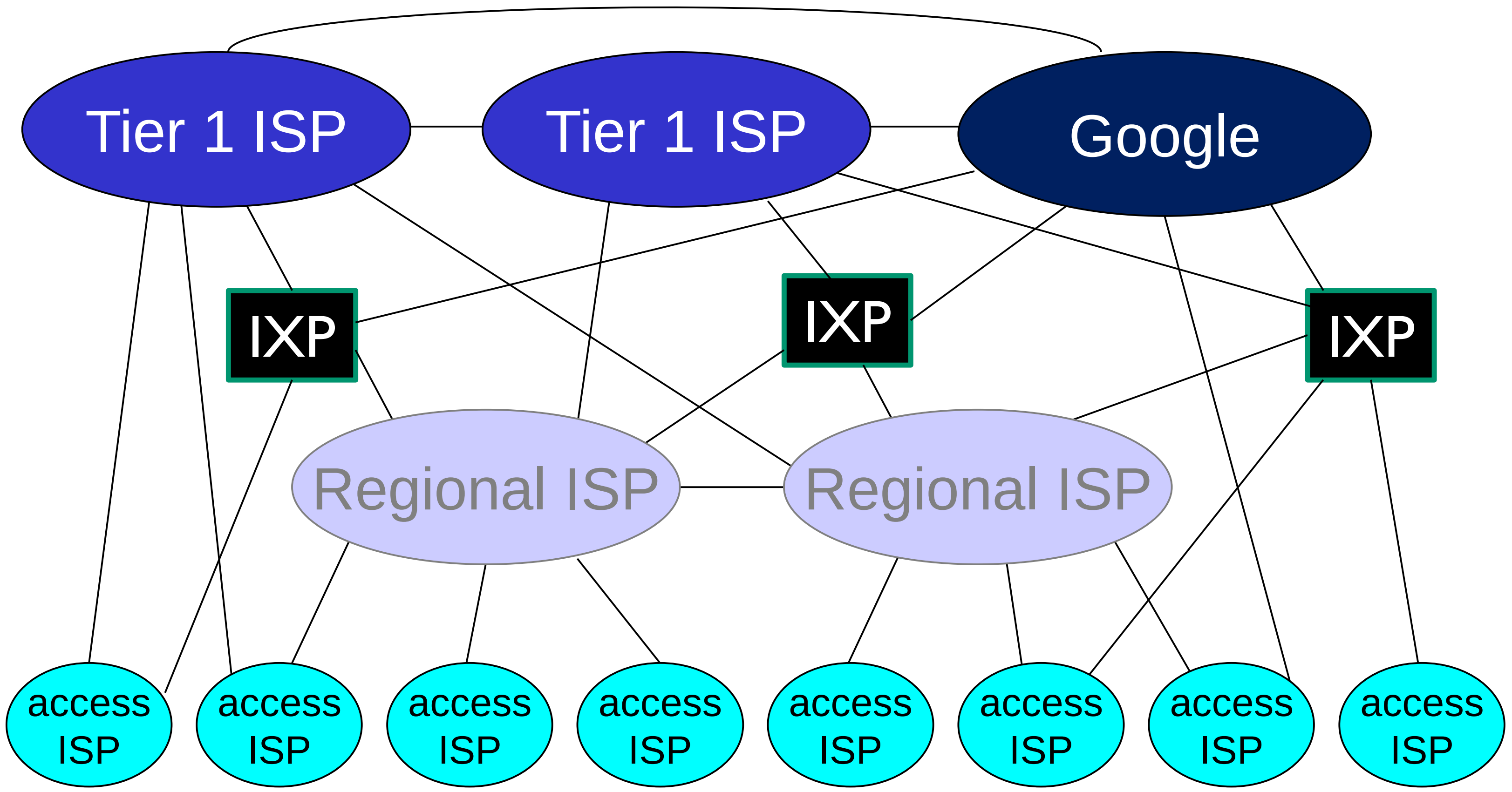


Internet structure: network of networks

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

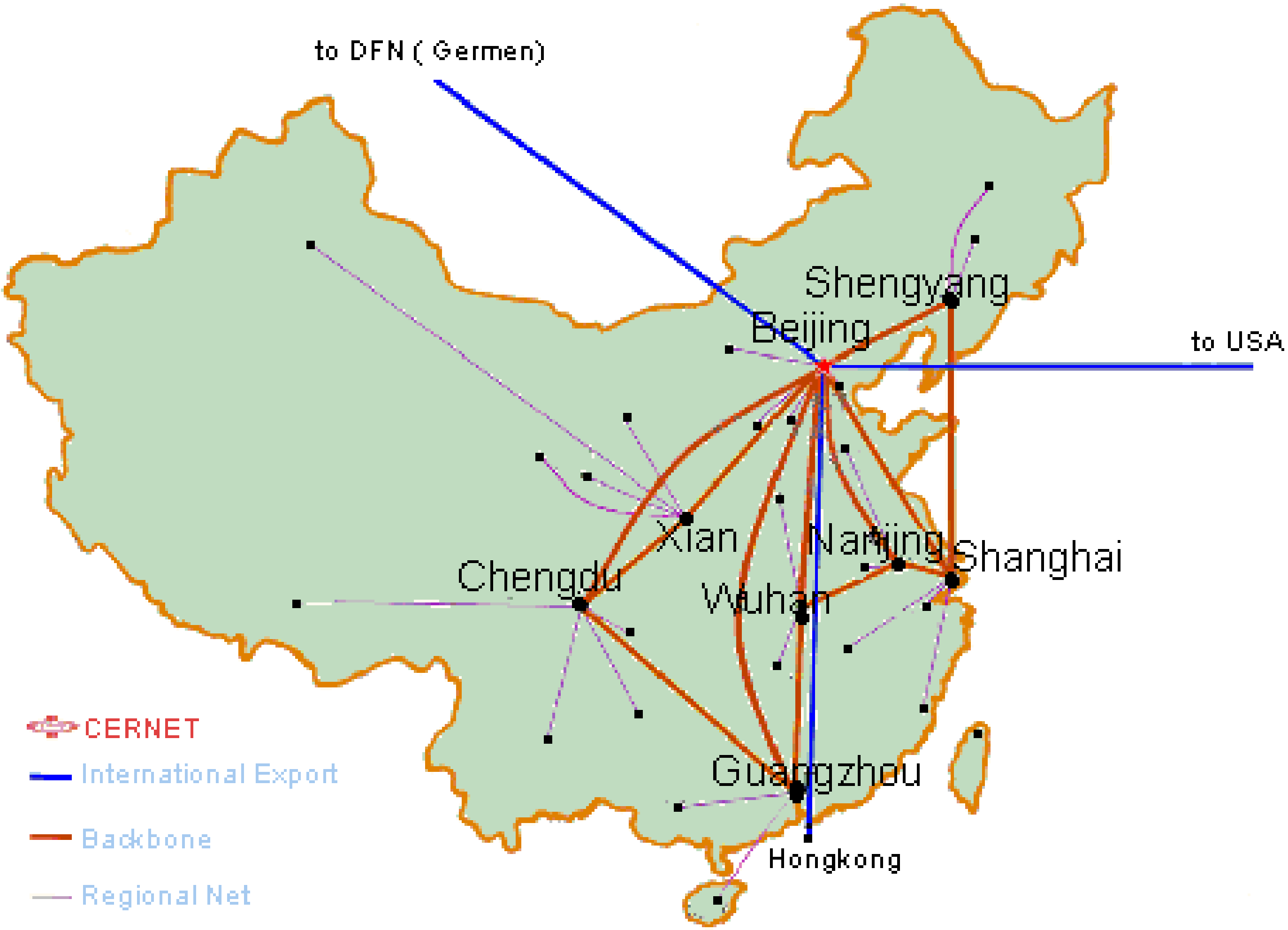


Internet structure: network of networks



- 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

Chinese education internet



What have we learned?

- Internet
  - A network of networks. Inter = in between.
  - Protocols govern how systems connect.
  - Later lectures will explain these protocols in detail.
- Edge systems connected by powerful core
  - Access systems connect homes and offices.
  - Internet service providers carry traffic from access (edge).
  - Tier 1 (high bandwidth) providers at "top of tree".
  - Internet eXchange Points (IXPs) connect nearby ISPs.
  - Content Distribution Networks (CDNs) push content close to users.

Structure of course

- Week 1
  - Introduction to IP Networks
  - The Transport layer (part 1)
- Week 2
  - The Transport layer (part II)
  - The Network layer (part I)
  - Class test
- Week 3
  - The Network layer (part II)
  - The Data link layer (part I)
  - Router lab tutorial (assessed lab work after this week)
- Week 4
  - The Data link layer (part II)
  - Network management and security
  - Class test

Introduction to IP: roadmap

1.1 what *is* the Internet?

1.2 network edge

- end systems, access networks, links

1.3 network core

- network structure

1.4 delay, loss, throughput in networks

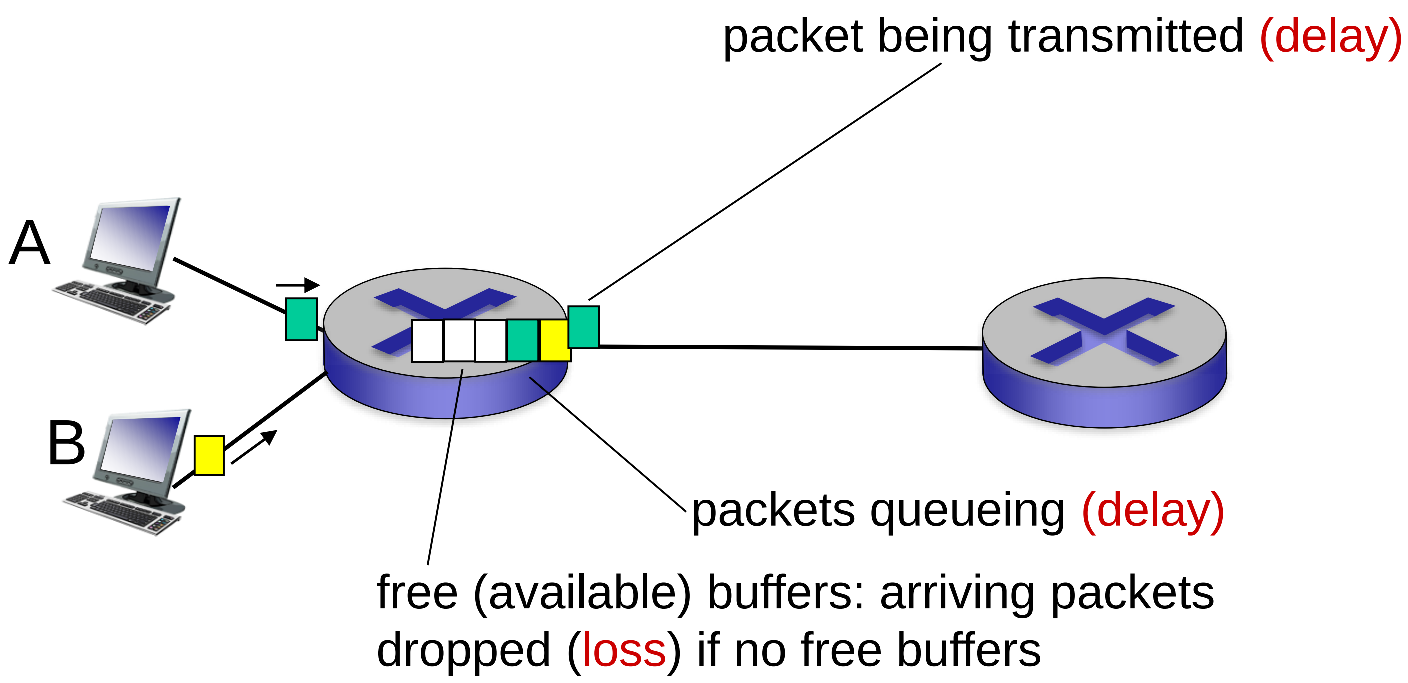
1.5 protocol layers, service models

1.6 history

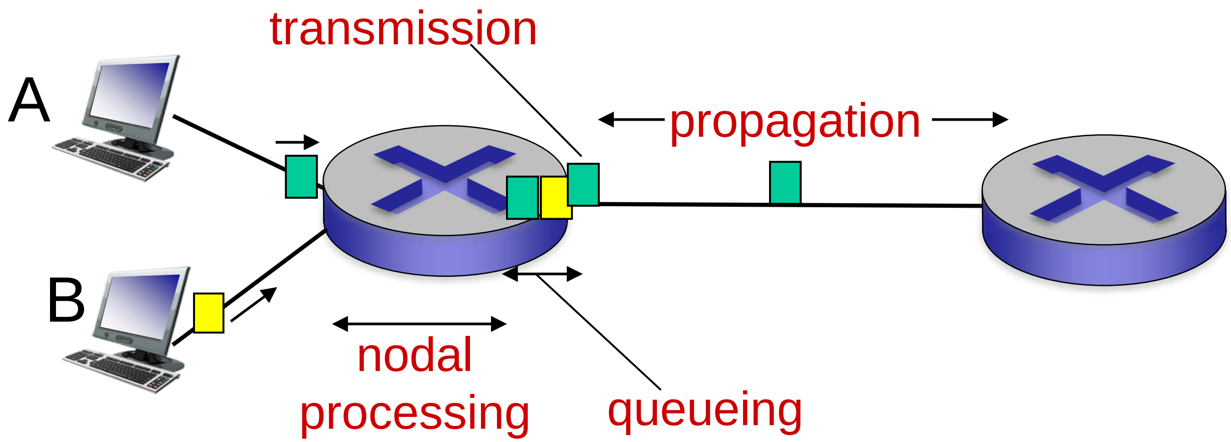
How do loss and delay occur?

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



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

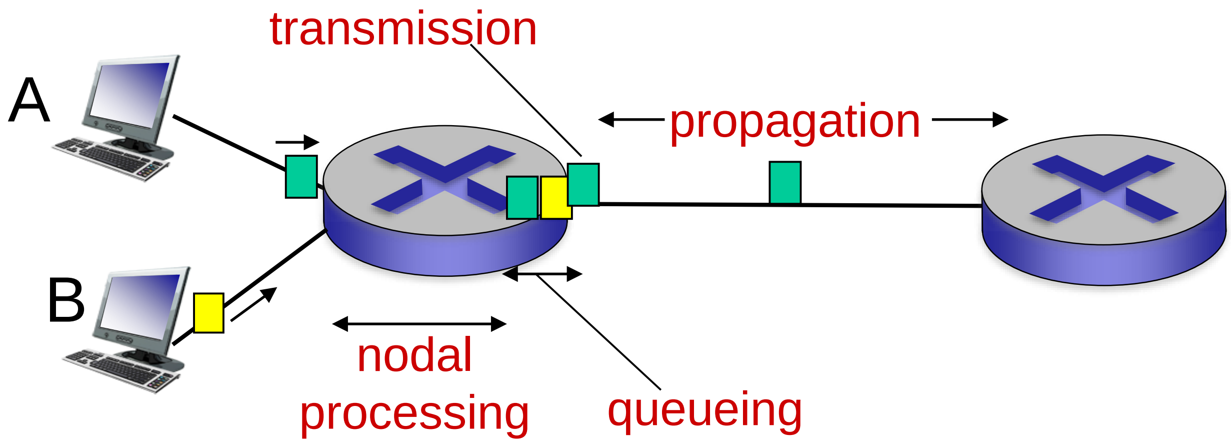
d_{proc}: nodal processing

- 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



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

d_{trans}: transmission delay:

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

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

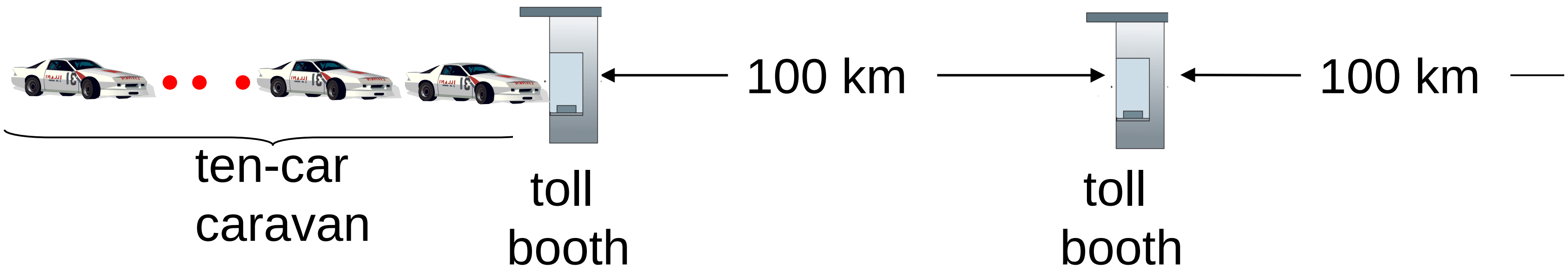
d_{prop}: propagation delay:

- d: length of physical link
- s: propagation speed (~2x10^8 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

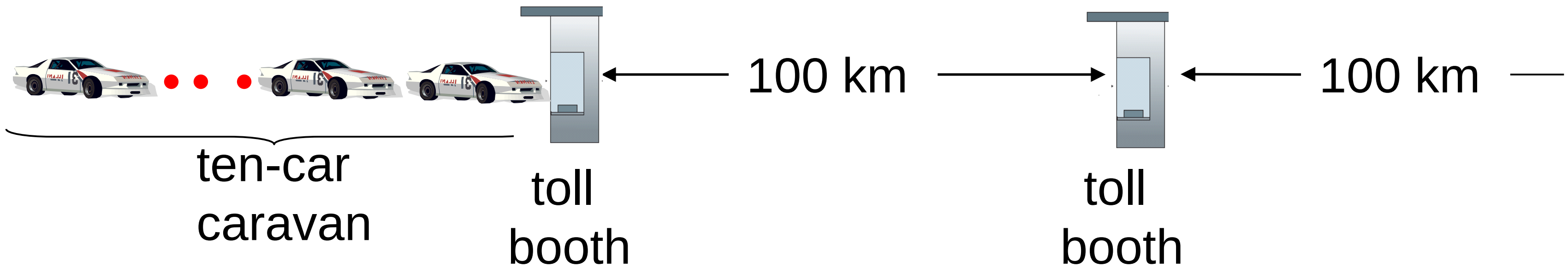
Caravan (group of cars) analogy



- 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 (group of cars) analogy (2)

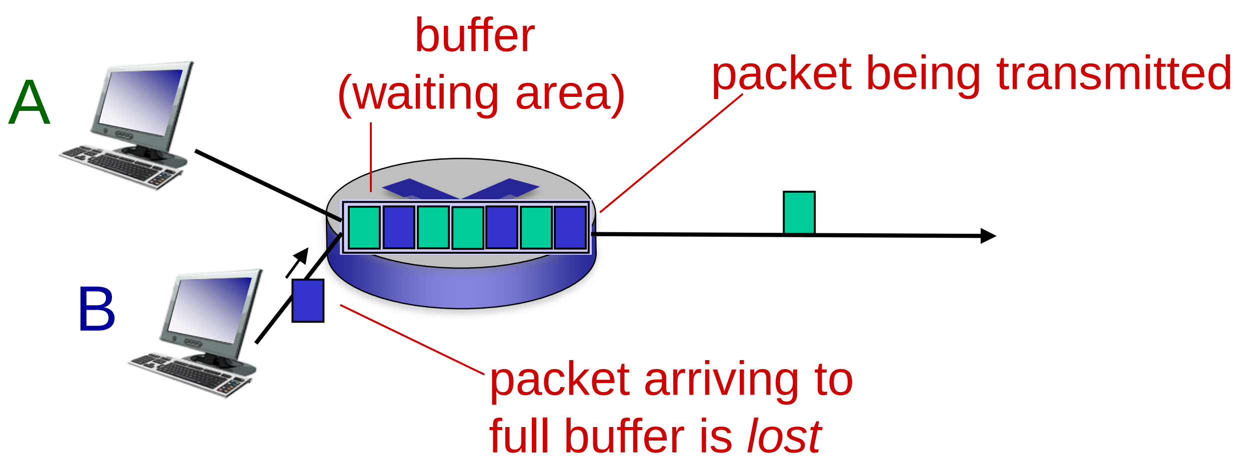


- suppose cars now "propagate" at 1000 km/hr
- and suppose toll booth now takes one min to service a car
- *Q:* Will cars arrive to 2nd booth before all cars serviced at first booth?

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

Packet loss

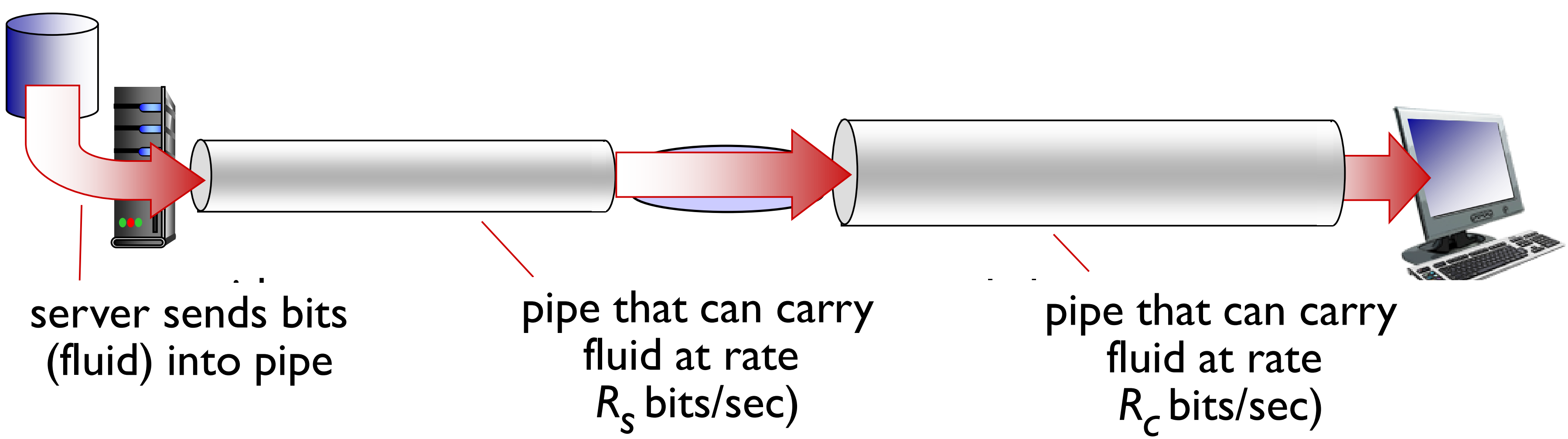
- queue (aka buffer) preceding link in buffer has finite capacity
- packet arriving to full queue dropped (aka lost)
- lost packet may be retransmitted by previous node, by source end system, or not at all



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

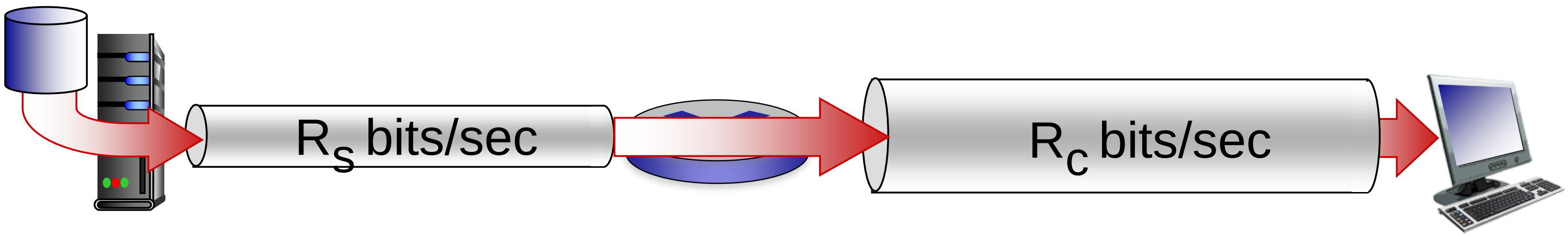
Throughput

- *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

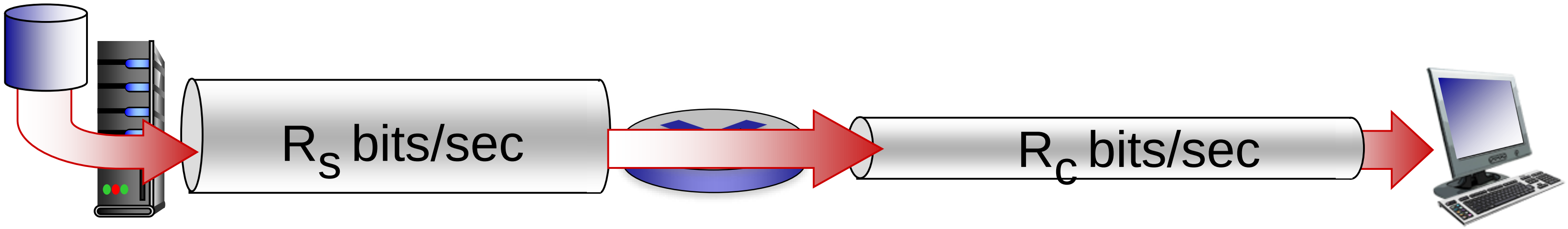


Throughput (more)

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



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

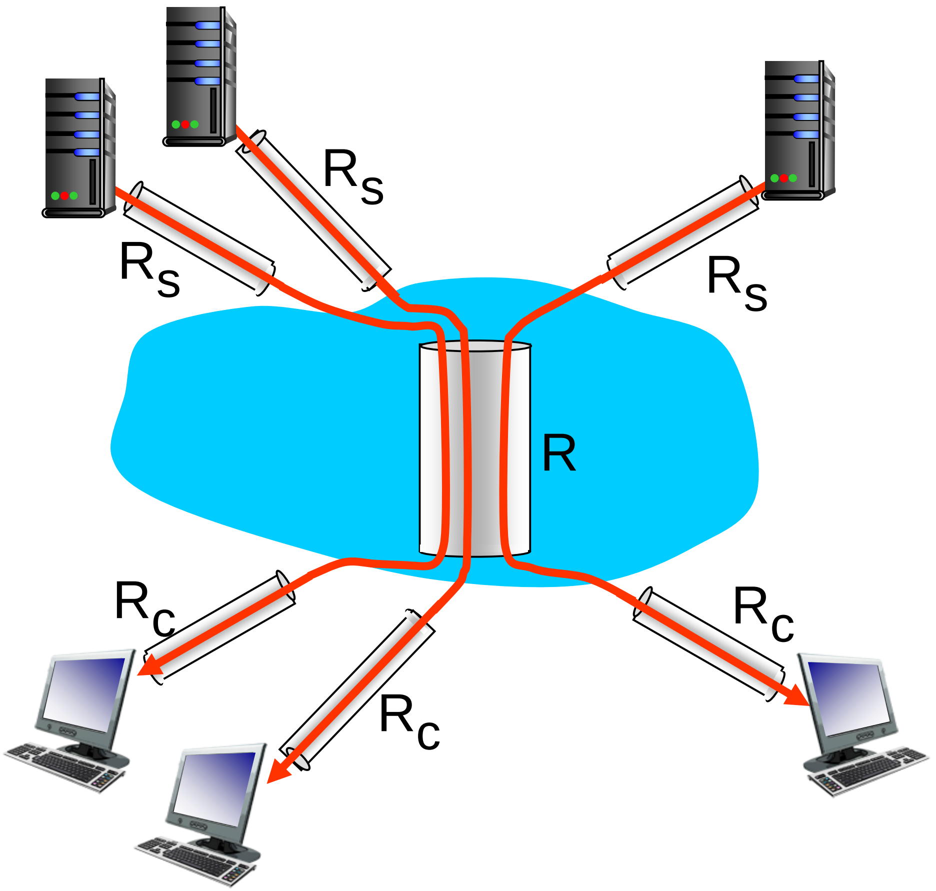


bottleneck link

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/

Introduction to IP: roadmap

1.1 what *is* the Internet?

1.2 network edge

- end systems, access networks, links

1.3 network core

- network structure

1.4 delay, loss, throughput in networks

1.5 protocol layers, service models

1.6 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?

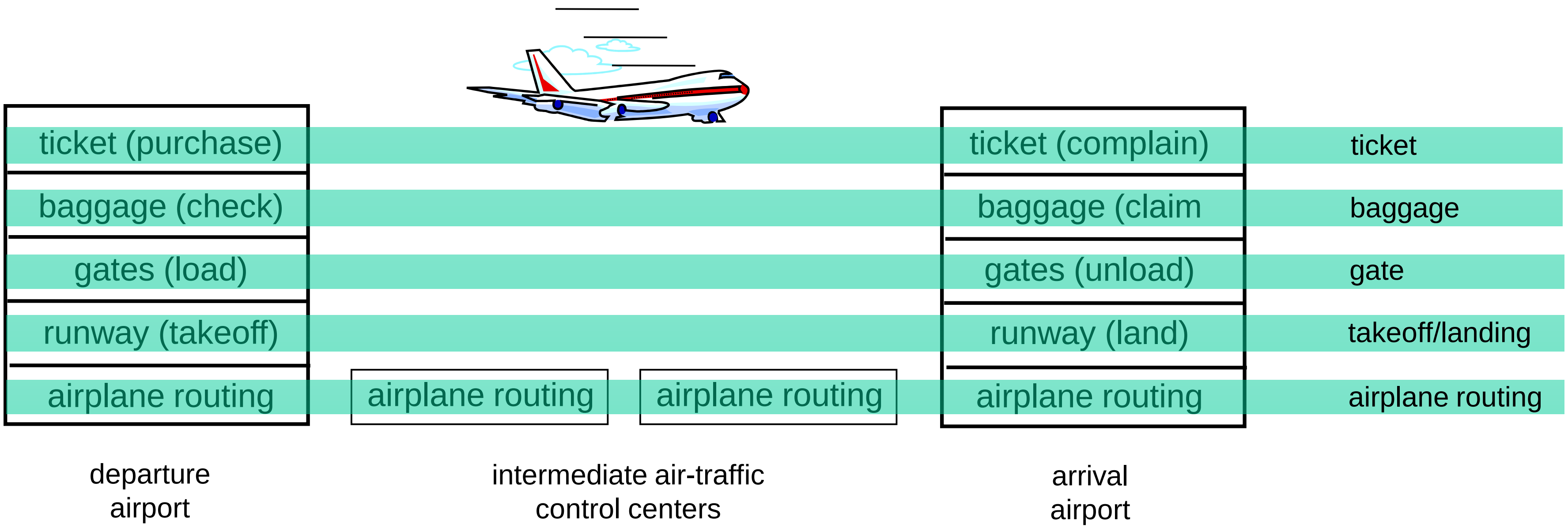
Organization of air travel

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

airplane routing

- a series of steps

Layering of airline functionality



layers: each layer implements a service

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

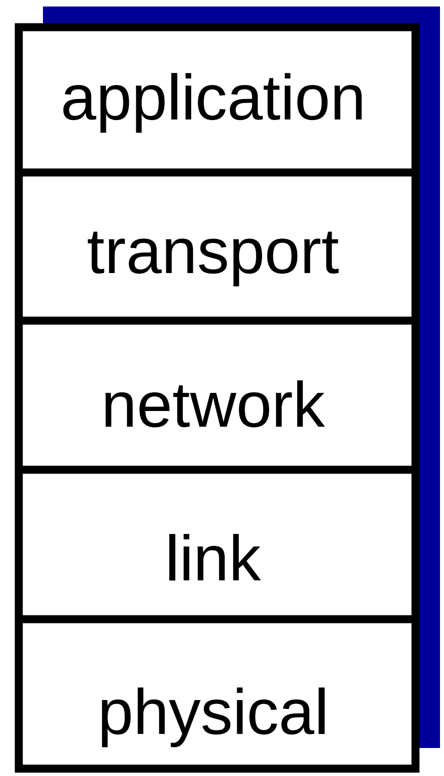
Why layering?

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 to rest of system
  - e.g., change in gate procedure doesn't affect rest of system
- layering considered harmful?

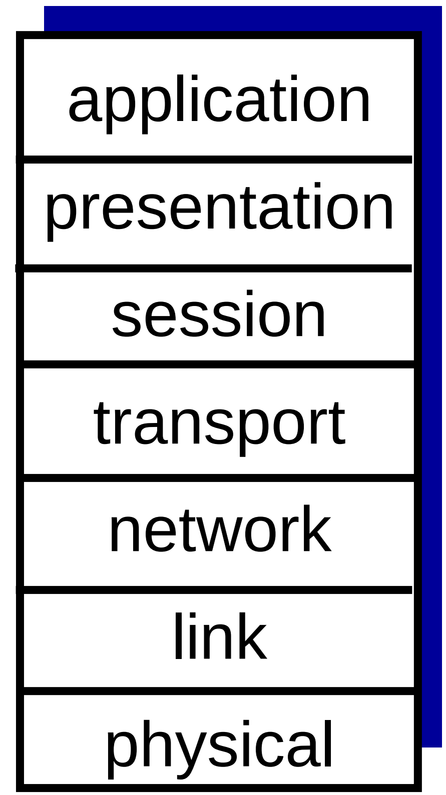
TCP/IP model (internet model)

- *application:* supporting network applications
  - FTP, SMTP, HTTP
- *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, 802.111 (WiFi), PPP
- *physical:* bits "on the wire"



ISO/OSI reference model

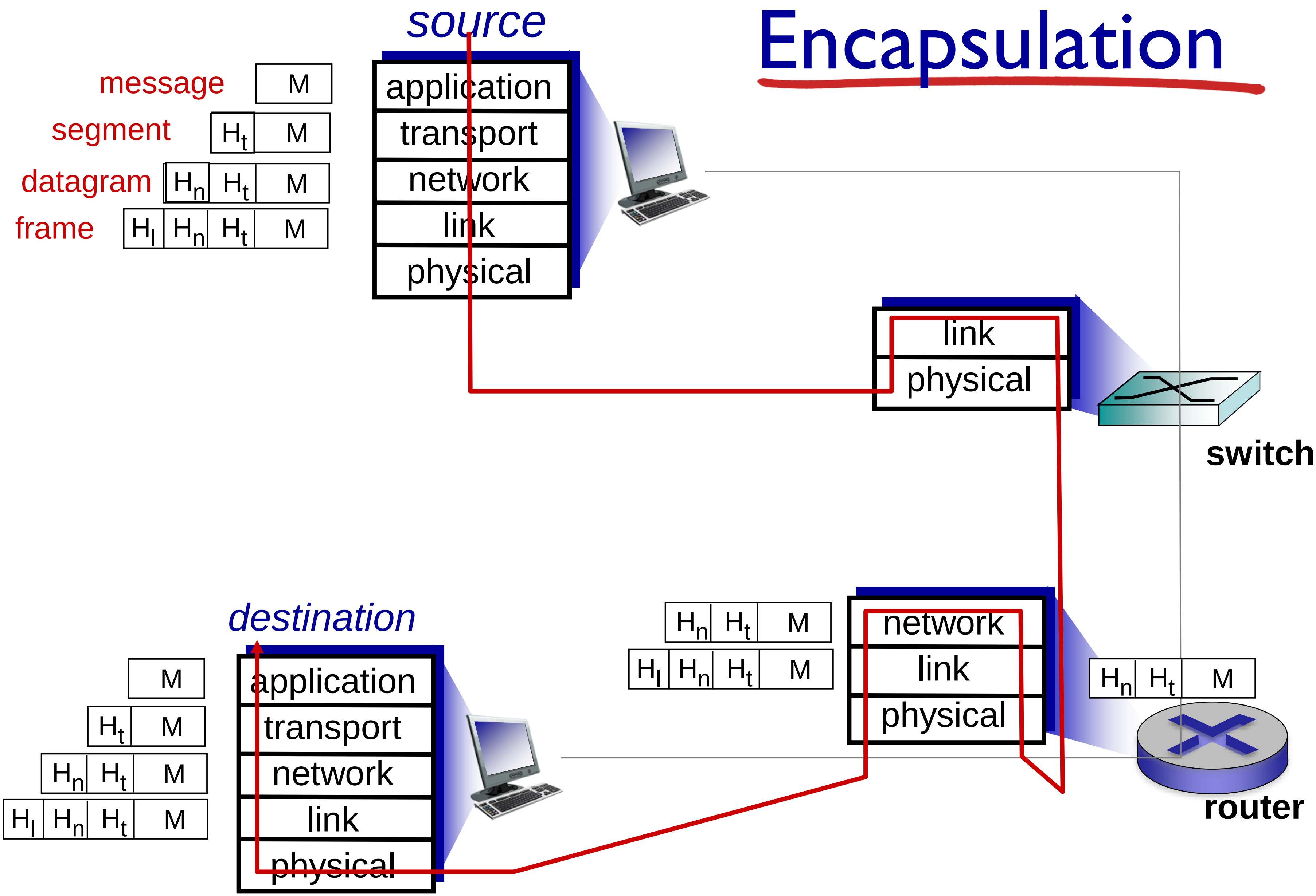
- ISO = International Standards Office
- OSI = Open Systems Interconnection
- *presentation:* allow applications to interpret meaning of data, e.g., encryption, compression, machine-specific conventions
- *session:* synchronization, checkpointing, recovery of data exchange
- Internet stack “missing” these layers!
  - these services, *if needed*, must be implemented in application
  - needed?



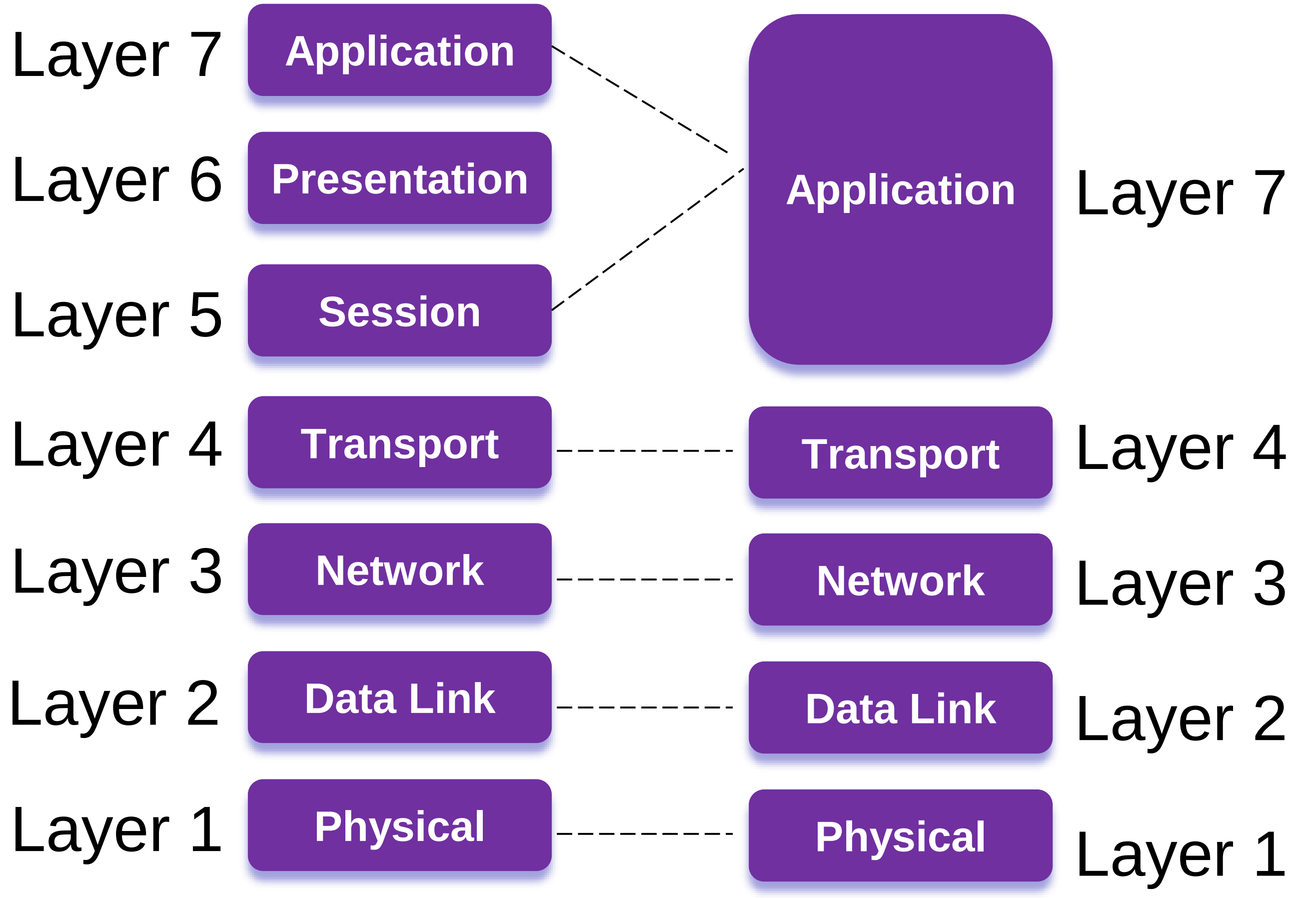
Layering and headers

- Most layers of the TCP/IP model are associated with a particular type of "header" or sometimes a "header and trailer".
- What is a header?
  - Information separate from the data being sent that says things about that data.
  - Which computer is it being sent to?
  - Which program on that computer must receive it?
  - How long is this data?
- For example layer 3 (network) has a "network address" which identifies the host that should receive the data.
- Layer 4 (transport) has a port that identifies which program should receive it.
- At each lower layer a new header is added incorporating the headers underneath. A layer 2 packet has a layer 2 header but includes headers from layer 3 and 4.

Encapsulation



ISO/OSI (left) vs TCP/IP (right)



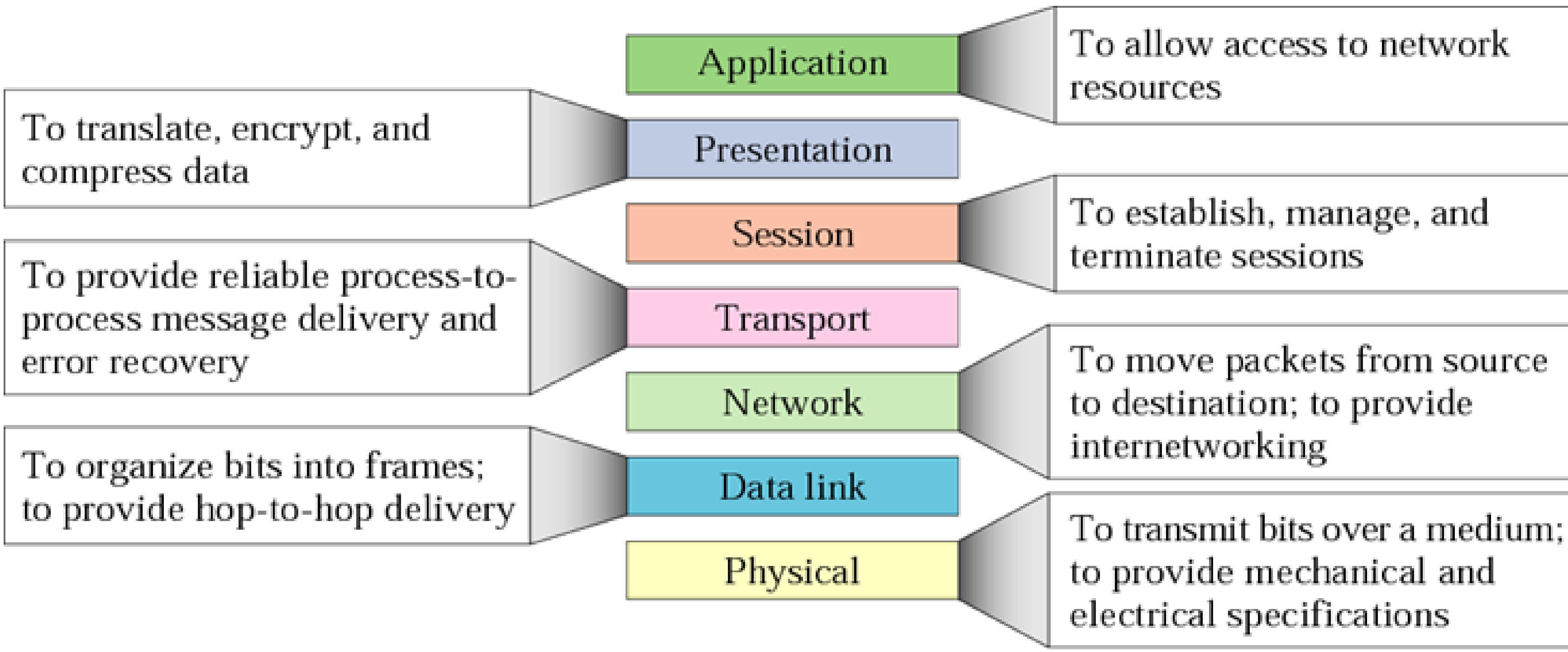
TCP/IP layers

- Layer 7 – Application layer – this is the data for programs you use on your computer
  - HTTP (www) data, SMTP (sent emails), FTP (file transfer) and specific formats for games, torrent etc.
- Layer 6 – Presentation layer
  - Related to character sets and presentation of data (unused in TCP/IP or real Internet)
- Layer 5 – Session layer
  - Related to whole lifetime of connection – is the connection real time (unused in TCP/IP or real Internet)
- Layer 4 – Transport layer – this is for the end-to-end connection between machines.
  - Information related to reliability
  - Information related to which program on machine sent/receives data.

TCP/IP layers

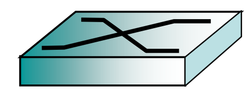
- Layer 3 – Network (Internet) layer – this is to get the data the whole journey from its start computer to its end computer (Internet, between networks)
  - Address of computer on internet (IP address)
  - Checksum to see if data is corrupted
- Layer 2 – Data link layer – this is to get the data to nearby computers (on same local network)
  - Media Access Control (MAC address) specific to individual computer (see later lectures)
- Layer 1 – Physical layer (how bits 1s and 0s are actually transmitted)
  - Think of cables in the ground or radio waves in the air

ISO/OSI (different summary)



Devices for layers

- Router
  - This is a layer 3 device – it reads a layer 3 address and works out which direction a packet should go.
  - It is typically more complex and adaptive than a switch.



- Switch
  - This is a layer 2 device – it reads a layer 2 address and works out which nearby computer should get a message.
  - Typically simpler than a router.
- Repeater
  - This is a layer 1 device – it strengthens or reconstructs a corrupted signal and carries on sending it.

ISO/OSI (left) vs TCP/IP (right)

- Why two models of layers?
- International Standards Office/Open Systems Interconnection model planned by committee.
  - Planning took a long time.
  - Model is idealized.
- Transmission Control Protocol/Internet Protocol built by engineers
  - Built up over time to "get things working".
  - New applications and changes to protocols through experience.
- When ISO/OSI design completed TCP/IP already "too big to change".
  - Would it be better if we had ISO/OSI?
  - Session layer presentation layer useful but don't exist today.
  - Perhaps but we can't get there from where we are.

How to remember the layers

- Please Do Not Throw Pizza + Sausage Away
- **P**lease – **P**hysical
- **D**o – **D**atalink
- **N**ot – **N**etwork
- **T**hrow – **T**ransport
- **S**ausage – **S**ession
- **P**izza – **P**resentation
- **A**way – **A**pplication

Chapter 1: roadmap

1.1 what *is* the Internet?

1.2 network edge

- end systems, access networks, links

1.3 network core

- network structure

1.4 delay, loss, throughput in networks

1.5 protocol layers, service models

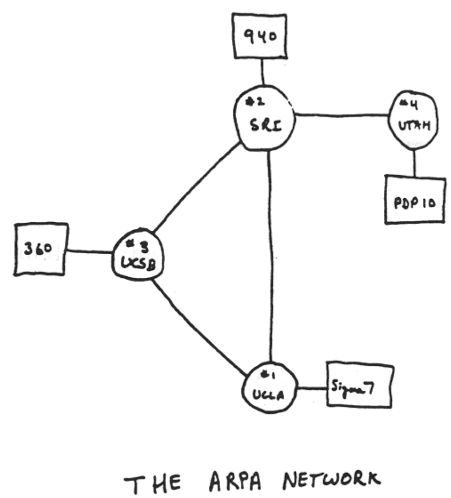
1.6 history

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 (US military)
- 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



Internet history

1972-1980: Internetworking, new and proprietary nets

- 1970: ALOHAnet satellite network in Hawaii
- 1973: UK connects to internet (at University College London)
- 1974: Cerf and Kahn - architecture for interconnecting networks
- 1976: Ethernet at Xerox PARC
- Late 70s: proprietary architectures: DECnet, SNA, XNS
- late 70s: 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 (simple mail transfer protocol) e-mail protocol defined
- 1983: DNS (Domain Name Server) defined for name-to-IP-address translation
- 1985: ftp protocol defined
- 1988: TCP congestion control
- 1989: first Internet connection in China

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

Internet history

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

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

late 1990s – 2000s:

- more killer apps: instant messaging, peer-to-peer (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)

ARPAnet May 1973

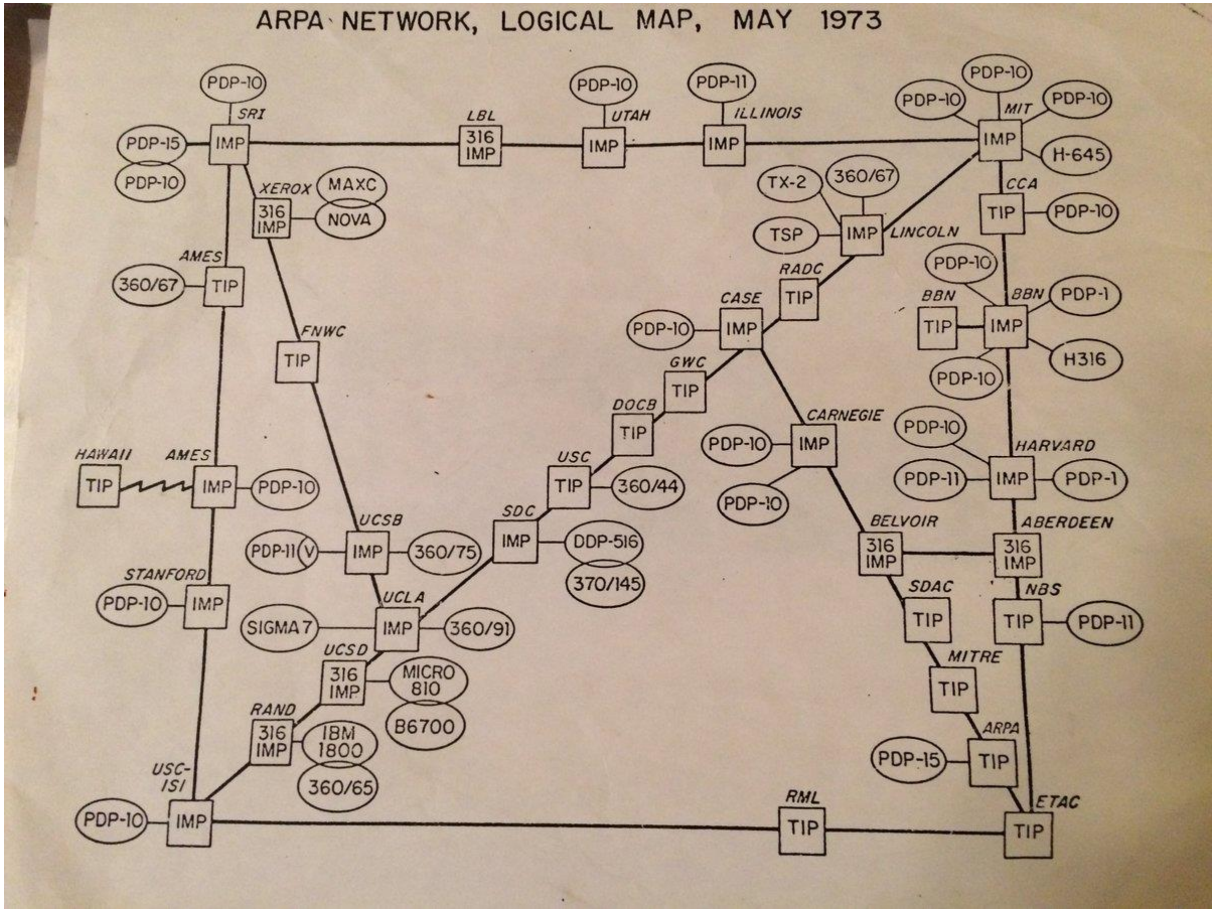
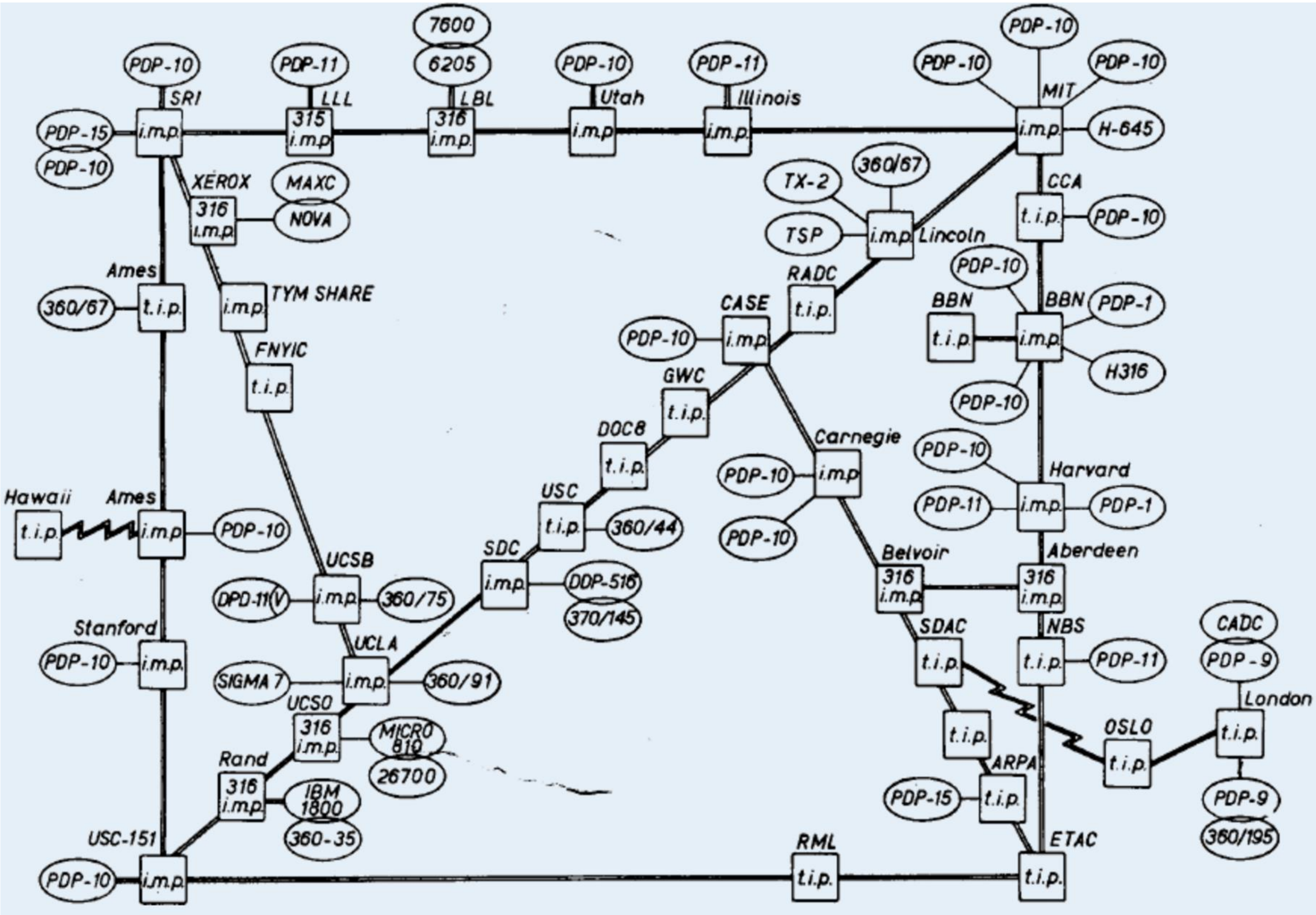
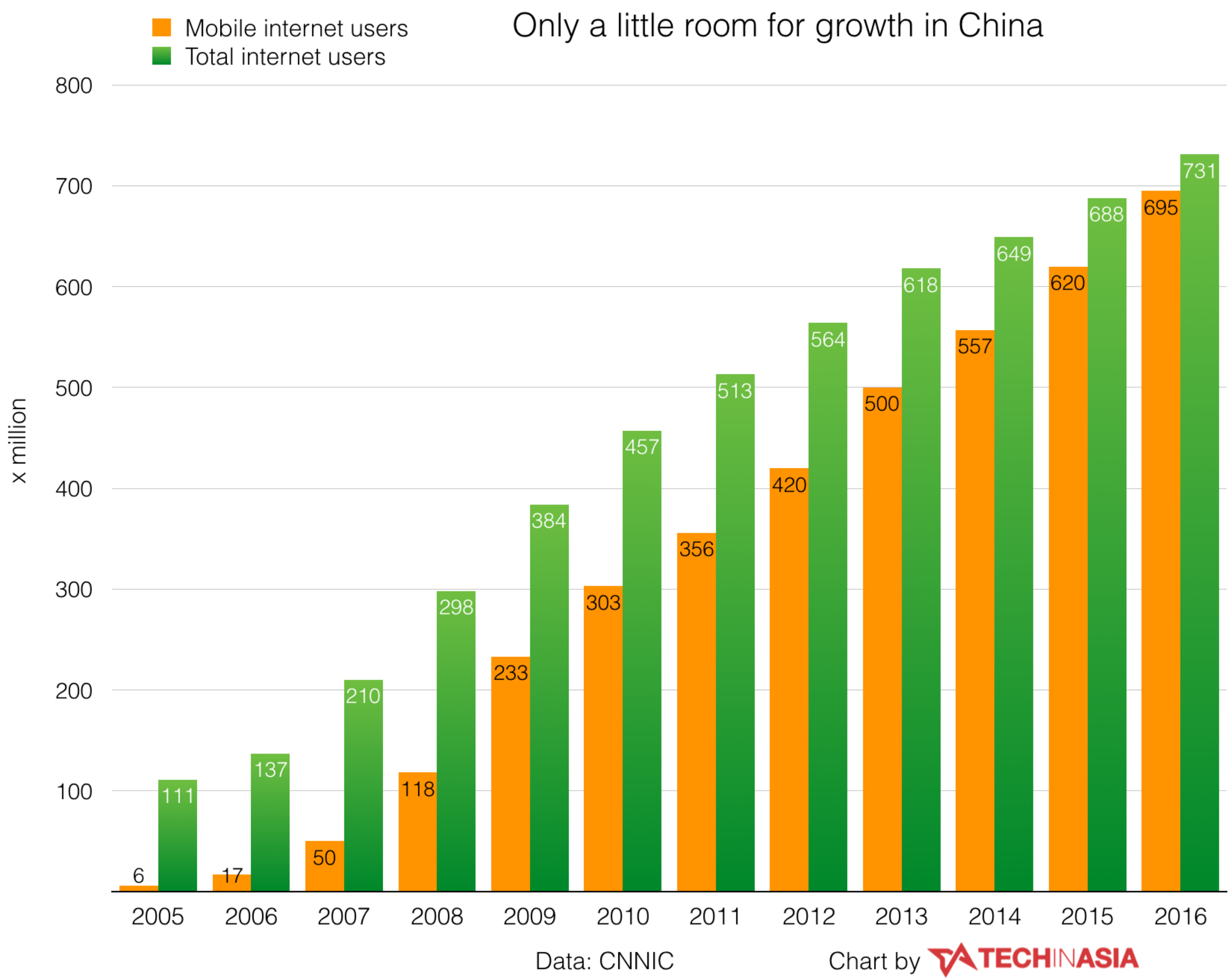


Image courtesy of David and Paul Newbury

ARPAnet November 1973



Internet in China (most users in world)



What have we learned?

- Internet performance:
  - Delay and loss together limit throughput.
  - Loss of packets when network busy (but not a problem – a normal part of network function).
- Protocol layers create Internet:
  - OSI/ISO: Physical, Datalink, Network, Transport, Presentation, Session, Application
  - TCP/IP: No Presentation + Session Layer
- Network history:
  - Small beginnings but now the planet's largest machine.
  - China joined the Internet late but grew to the largest country online.