Computer Networks and Applications

COMP 3331/COMP 9331

Week 1

Introduction to Computer Networks

Reading Guide: Chapter 1, Sections 1.1 - 1.5

Acknowledgment

- Majority of lecture slides are from the author's lecture slide set
  - Enhancements + *additional material*

1. Introduction

Goals:

- get "feel" and terminology
- defer depth and detail to *later* in course
- understand concepts using the Internet as example

1. Introduction: roadmap

1.1 what *is* the Internet?

1.2 network edge

- end systems, access networks, links

1.3 network core

- packet switching, circuit 电路 switching, network structure

1.4 delay, loss, throughput in networks

1.5 protocol layers

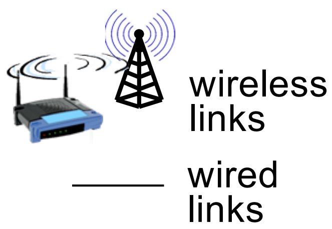
1.6 networks under attack: security

1.7 history

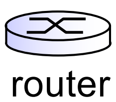
Hobbe's Internet Timeline - http://www.zakon.org/robert/internet/timeline/

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

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

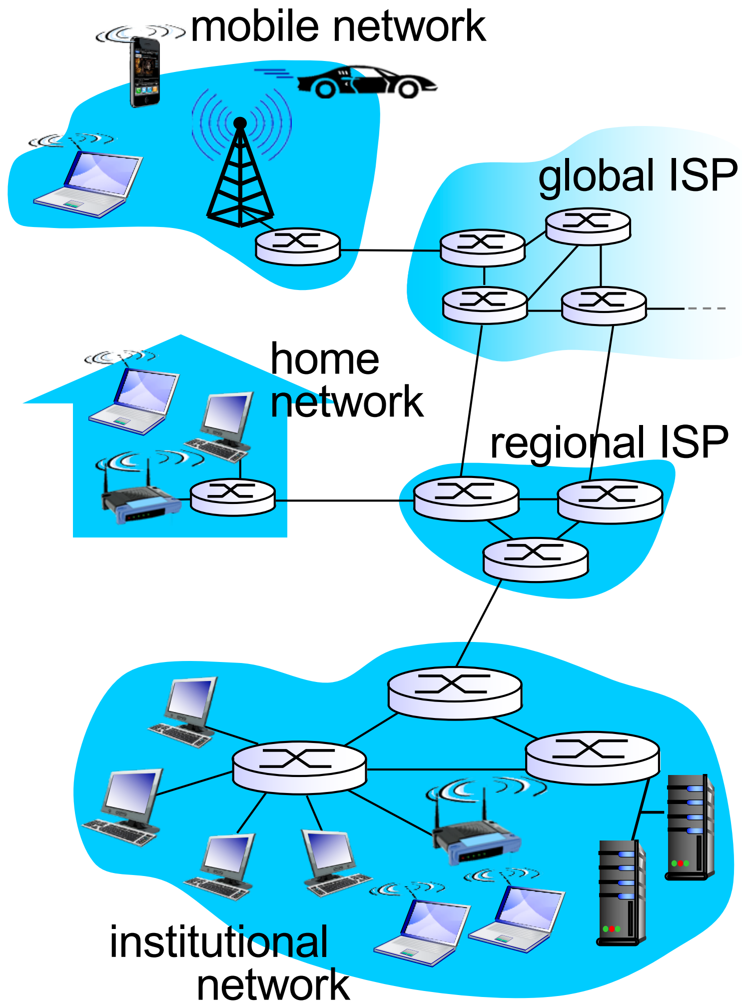


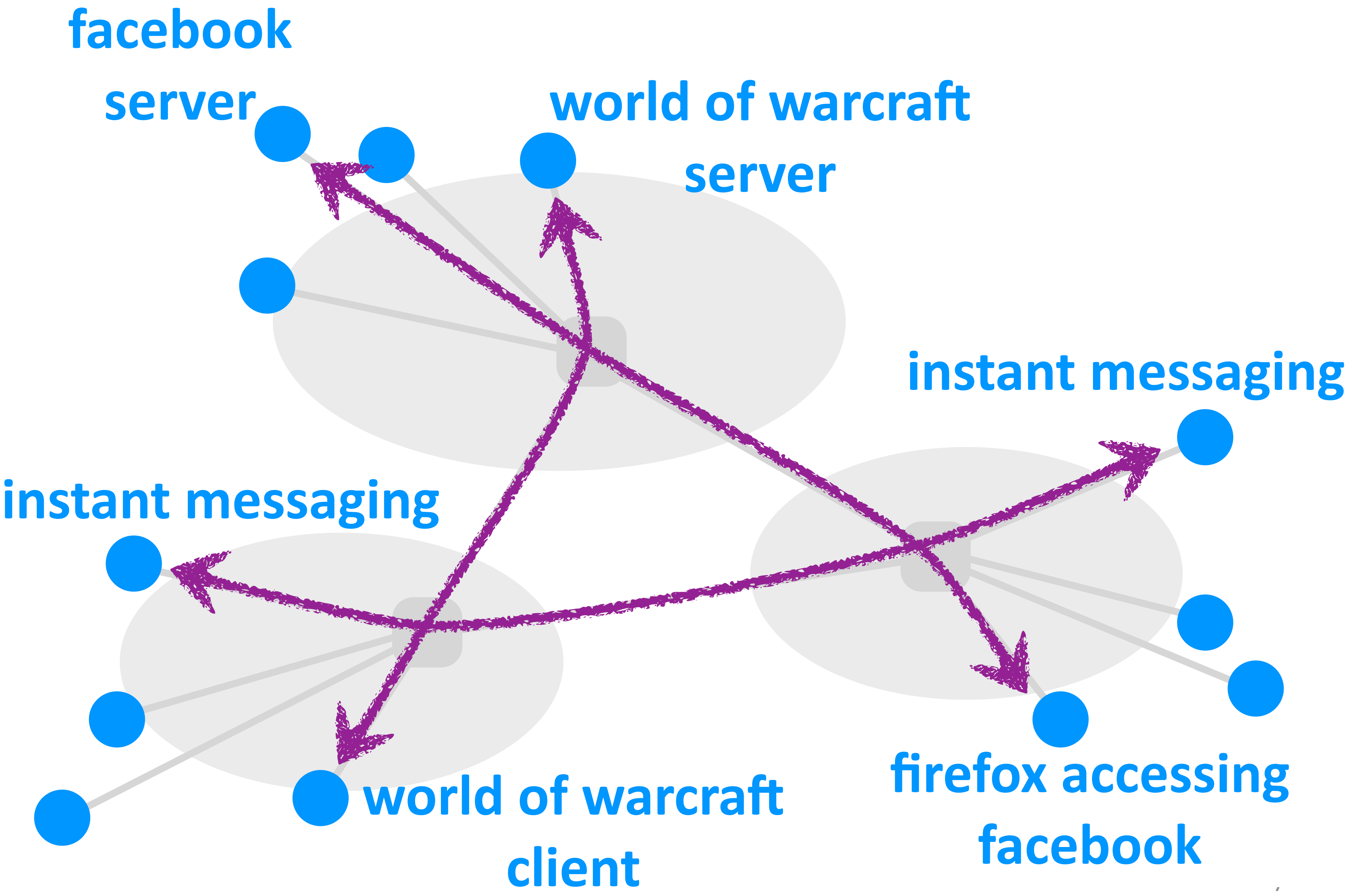
- *communication links*
  - fiber, copper, radio, satellite
  - transmission rate: *bandwidth*



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

路由器




facebook
server
world of warcraft
server
instant messaging
instant messaging
world of warcraft
client
firefox accessing
facebook

“Fun” Internet appliances

Picture frame

Web-enabled toaster + weather forecaster

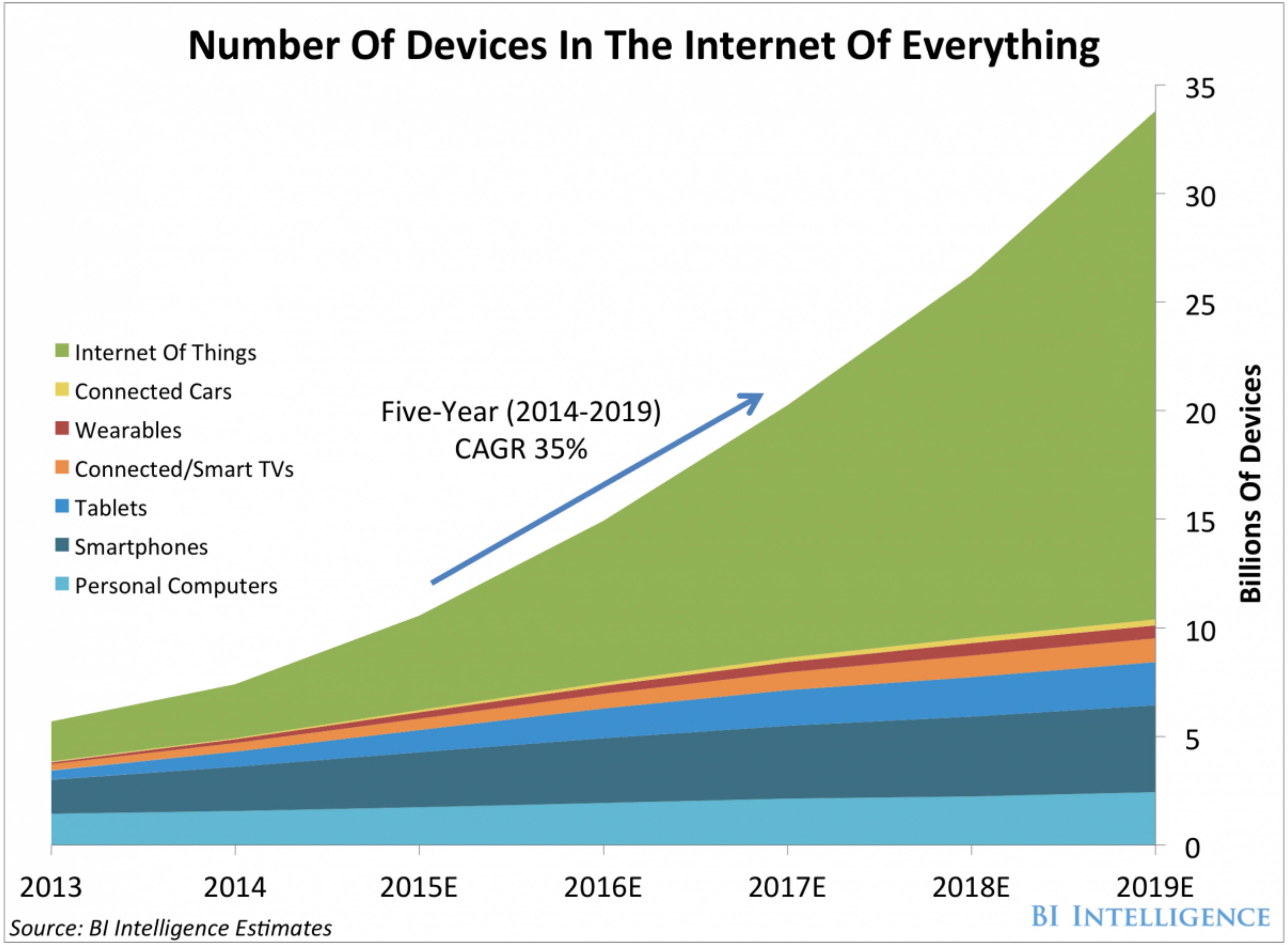
Tweet-a-watt: monitor energy use

Internet refrigerator

Networked TV Set top Boxes

sensorized, bed mattress

Smart Lightbulbs

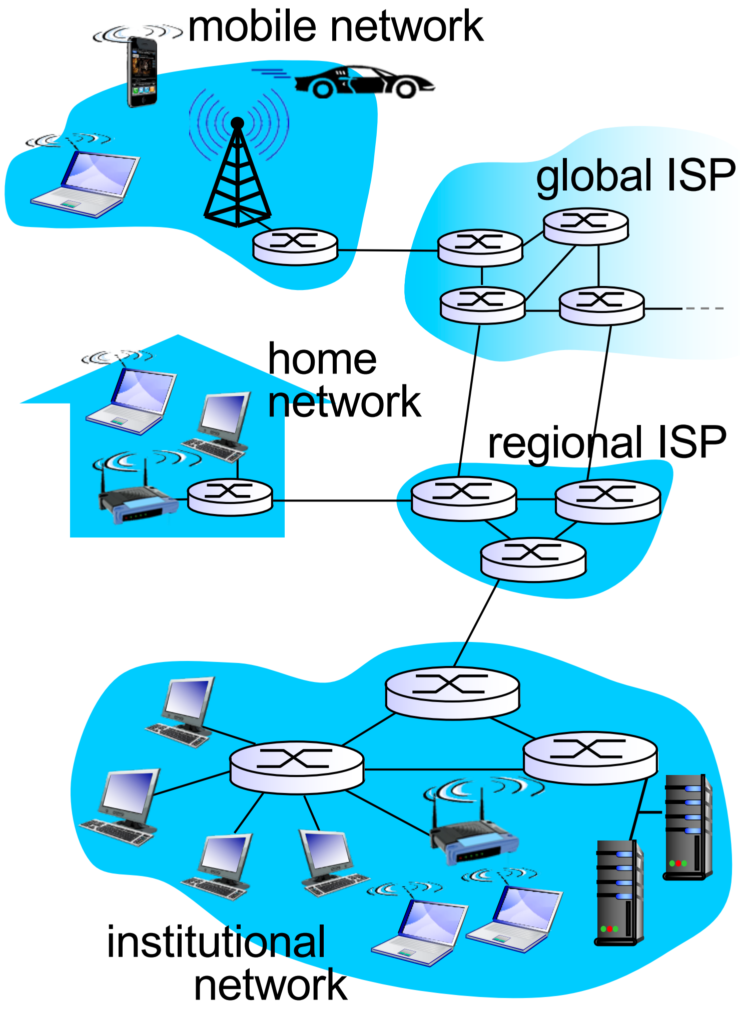


Number Of Devices In The Internet Of Everything
Internet Of Things
Connected Cars
Wearables
Connected/Smart TVs
Tablets
Smartphones
Personal Computers
Five-Year (2014-2019)
CAGR 35%
35
30
25
20
15
10
5
0
Billions Of Devices
2013
2014
2015E
2016E
2017E
2018E
2019E
Source: BI Intelligence Estimates
BI INTELLIGENCE

Smart Building

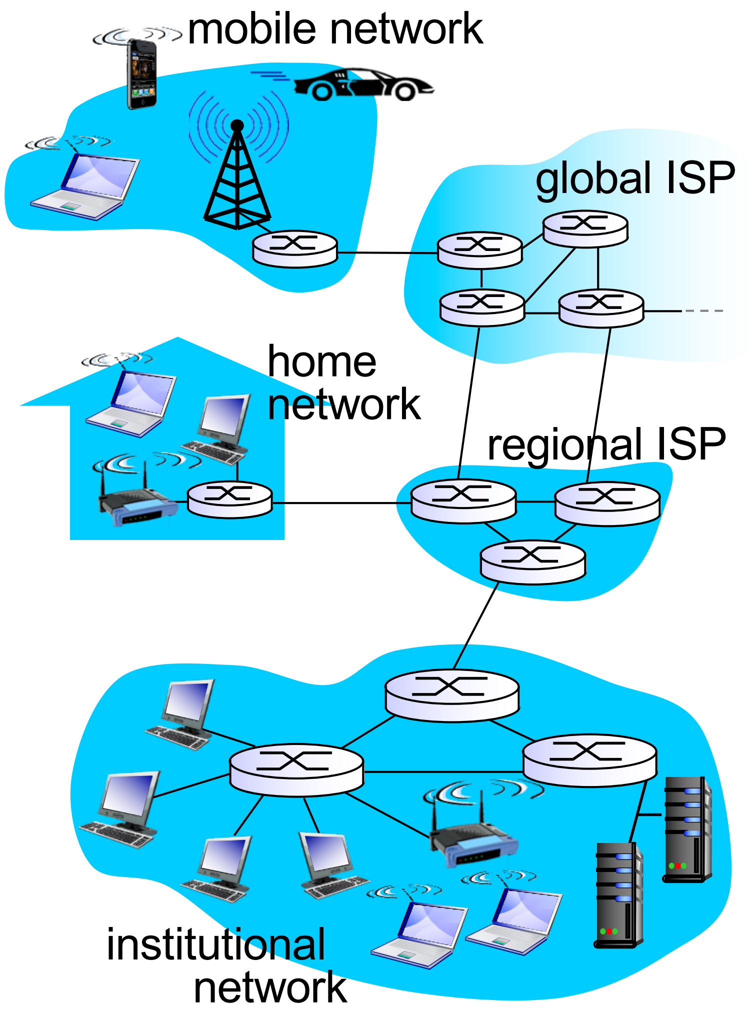
What's the Internet: "nuts and bolts" view

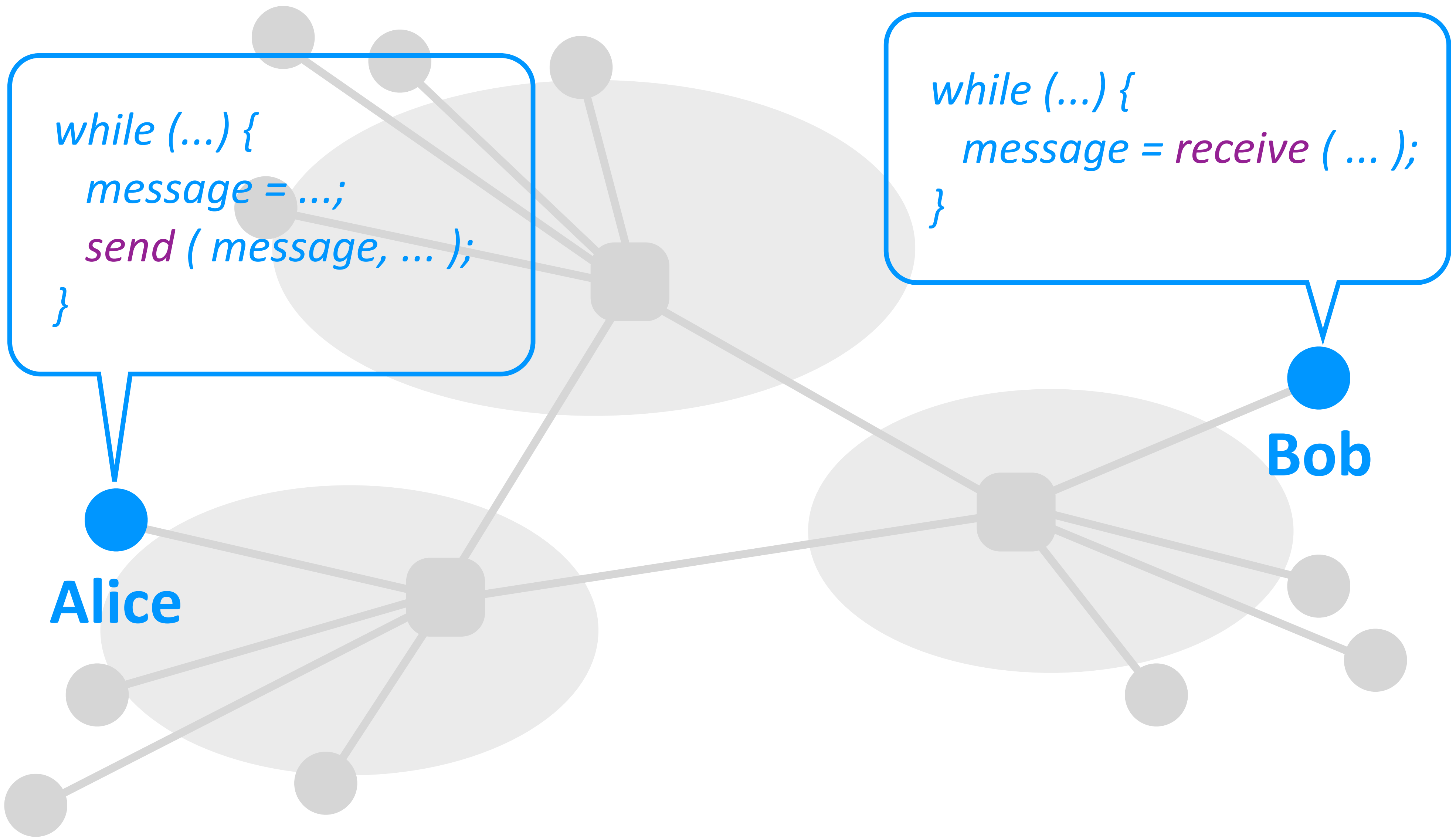
- *Internet:* "network of networks"
  - Interconnected ISPs
- 条约
  protocols control sending, receiving of msgs
  - e.g., TCP, IP, HTTP, Skype, 802.11
- *Internet standards*
  - RFC: Request for comments
  - IETF: Internet Engineering Task Force

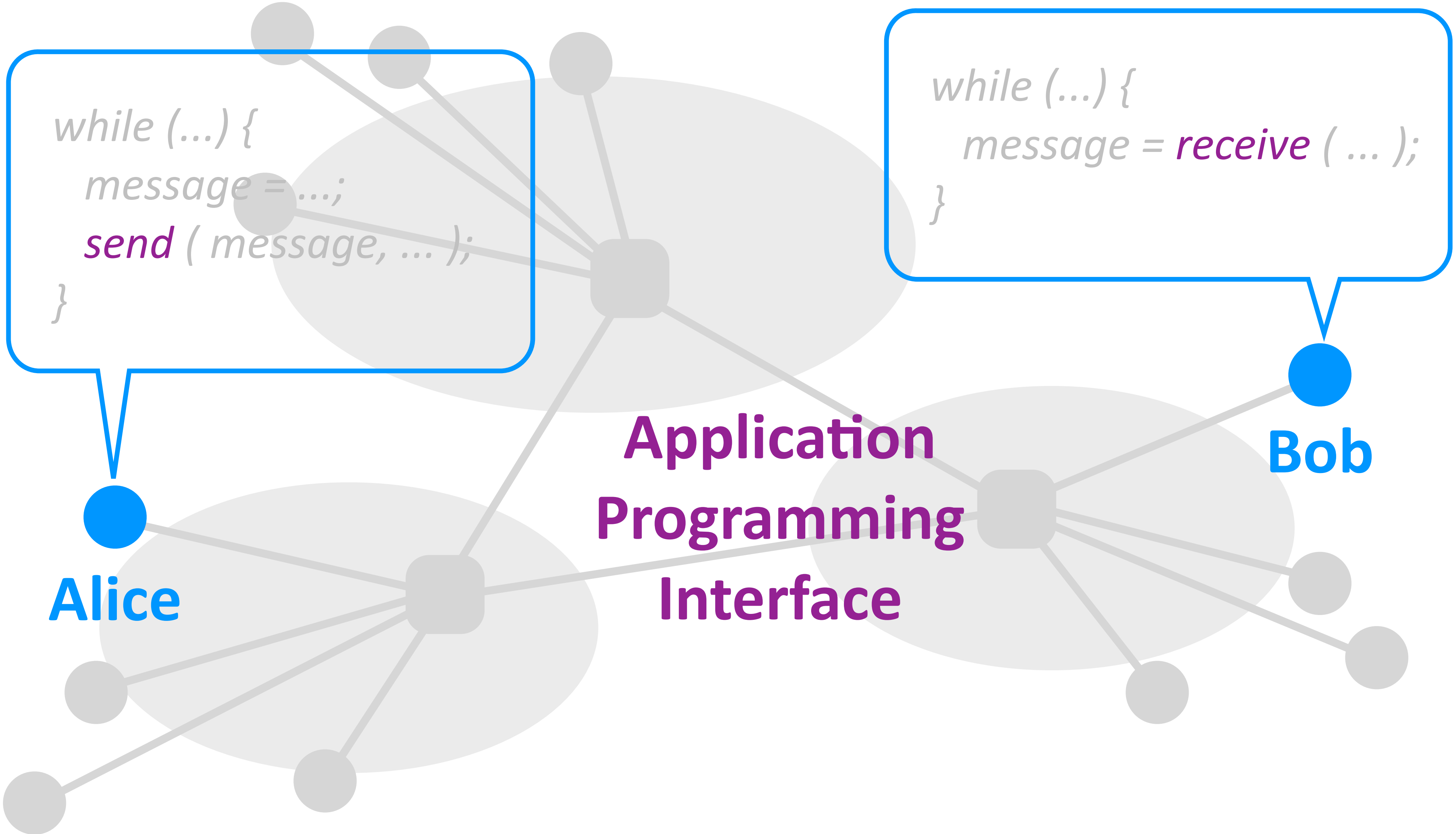


What's the Internet: a service view

- *Infrastructure that provides services to applications:*
  - Web, VoIP, 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




while (...) {
message = ...;
send (message, ...);
}
while (...) {
message = receive (...);
}
Bob
Alice


while (...) {
message = ...;
send (message, ...);
}
while (...) {
message = receive (...);
}
Alice
Bob
Application
Programming
Interface

What's a protocol?

human protocols:

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

… specific msgs sent

… specific actions taken when msgs received, or other events

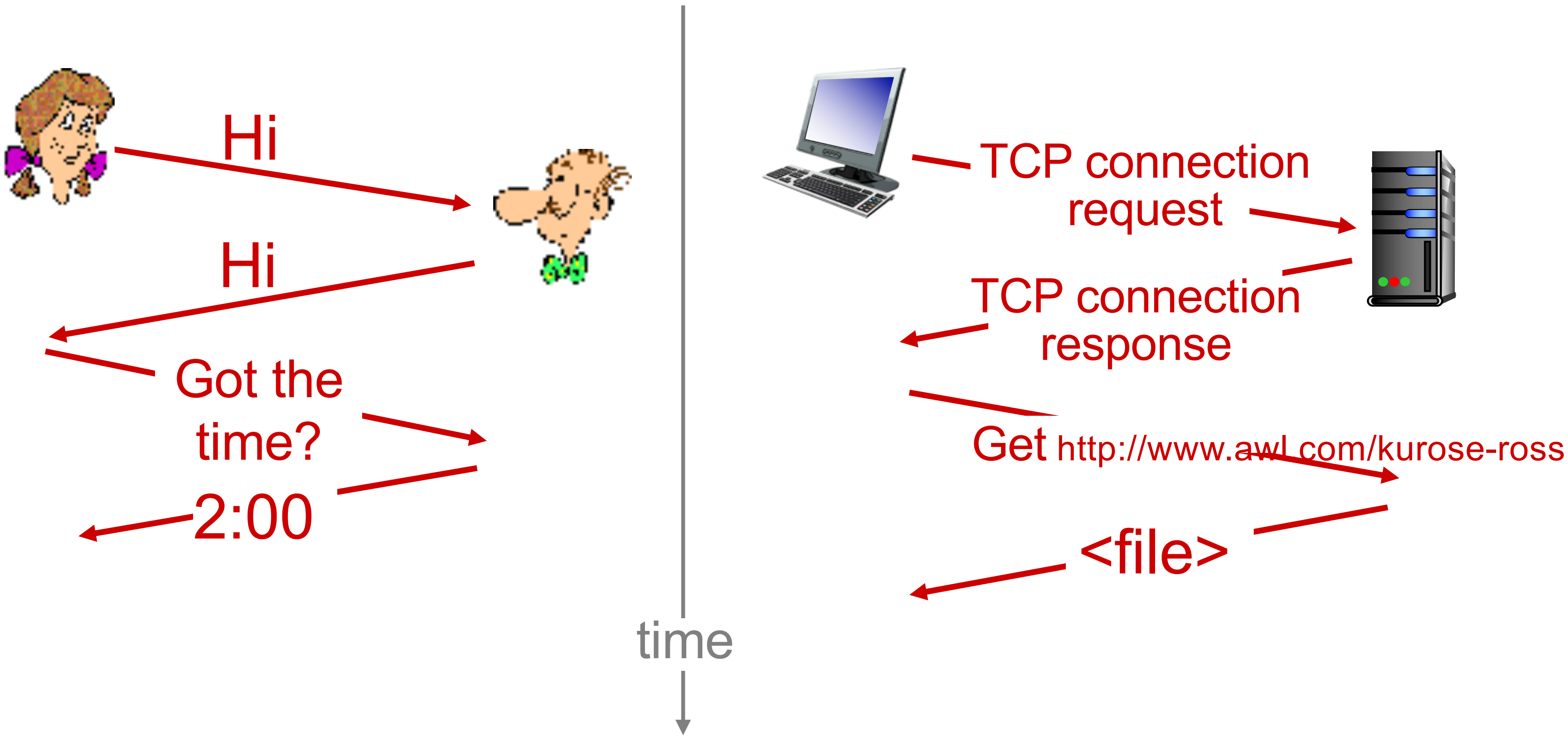
network protocols:

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

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

What's a protocol?

a human protocol and a computer network protocol:



Q: other human protocols?

1. Introduction: 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

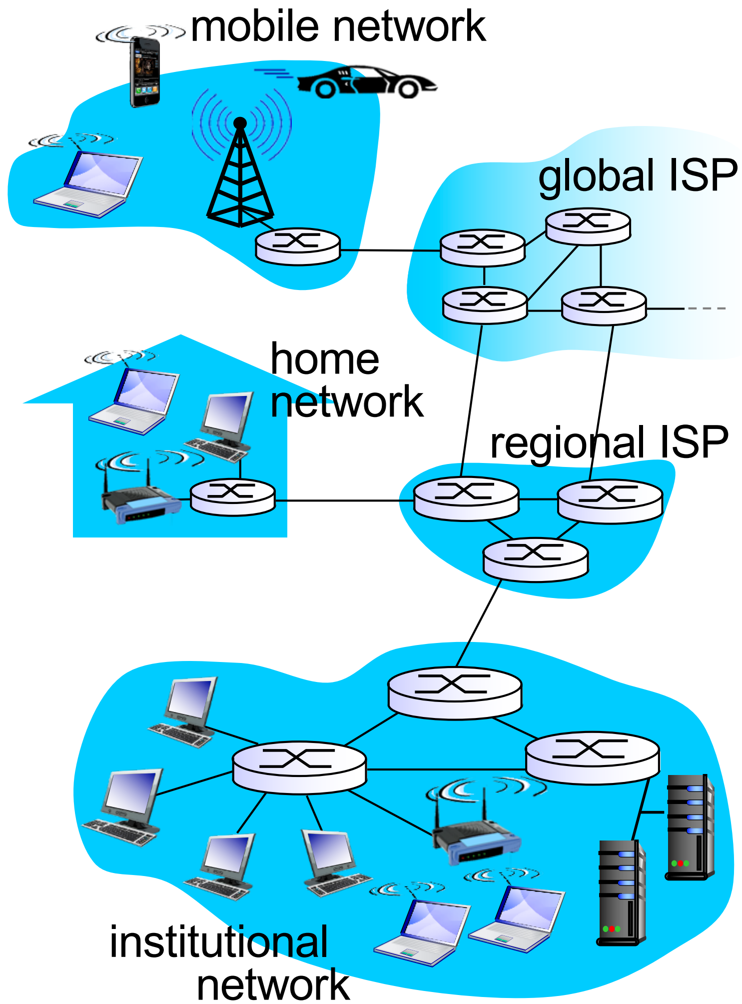
1.5 protocol layers, service models

1.6 networks under attack: security

1.7 history

A closer look at network structure:

- *network edge:*
  - hosts: clients and servers
  - servers often in data centers
- *access networks, physical media:* wired, wireless communication links
- *network core:*
  - interconnected routers
  - network of networks



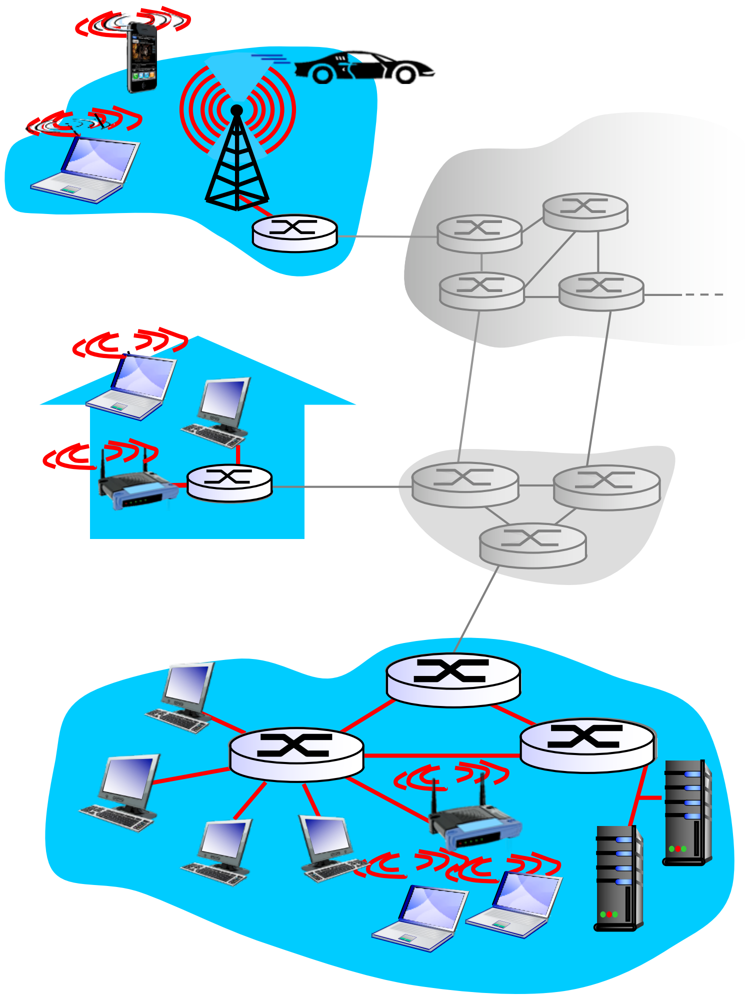
Access networks and physical media

Q: How to connect end systems to edge router?

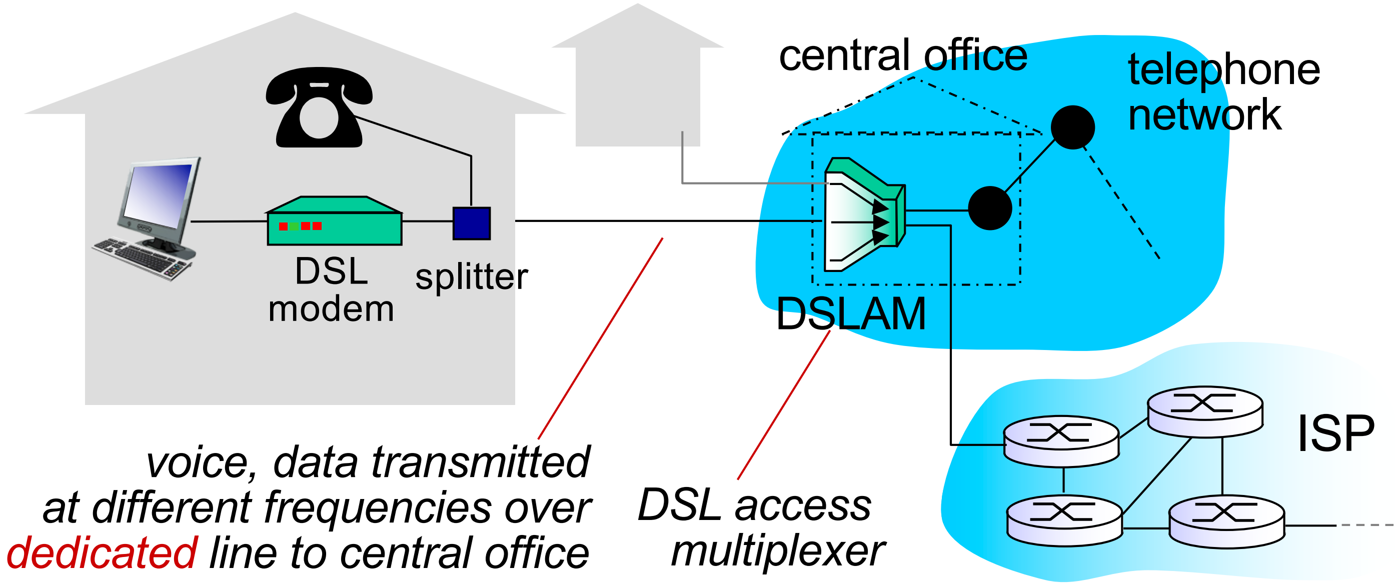
- residential access nets
- institutional access networks (school, company)
- mobile access networks

keep in mind:

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

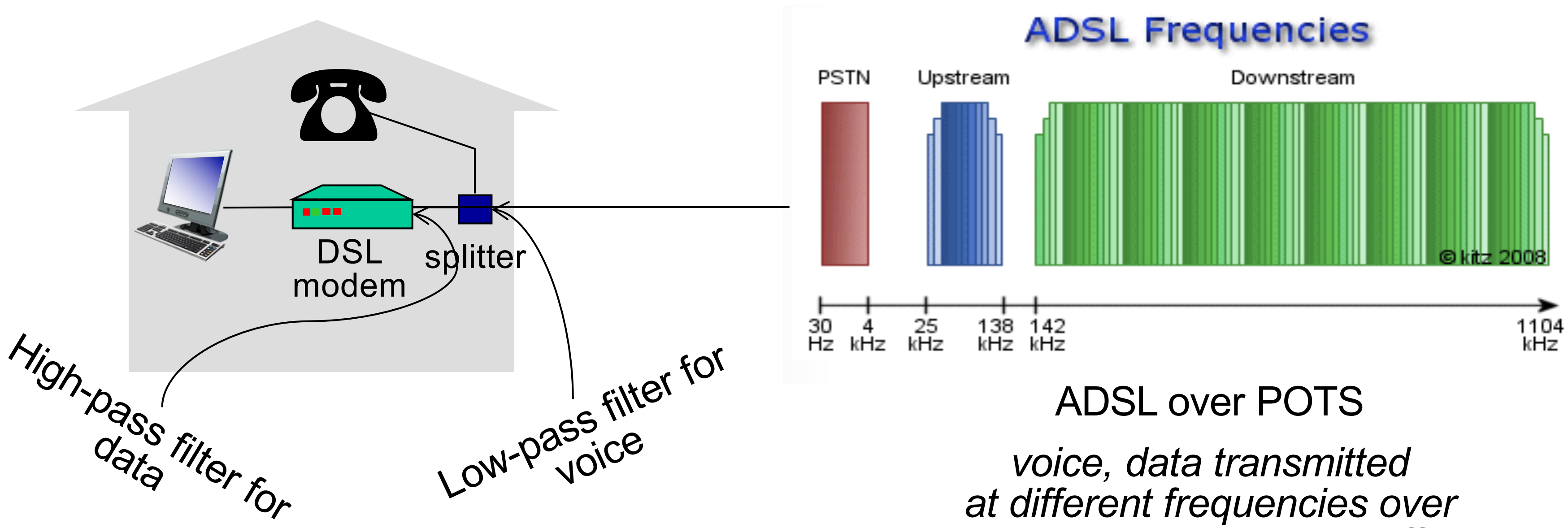


Access net: digital subscriber line (DSL)



- use *existing* telephone line to central office DSLAM
  - data over DSL phone line goes to Internet
  - voice over DSL phone line goes to telephone net

Access net: digital subscriber line (DSL)

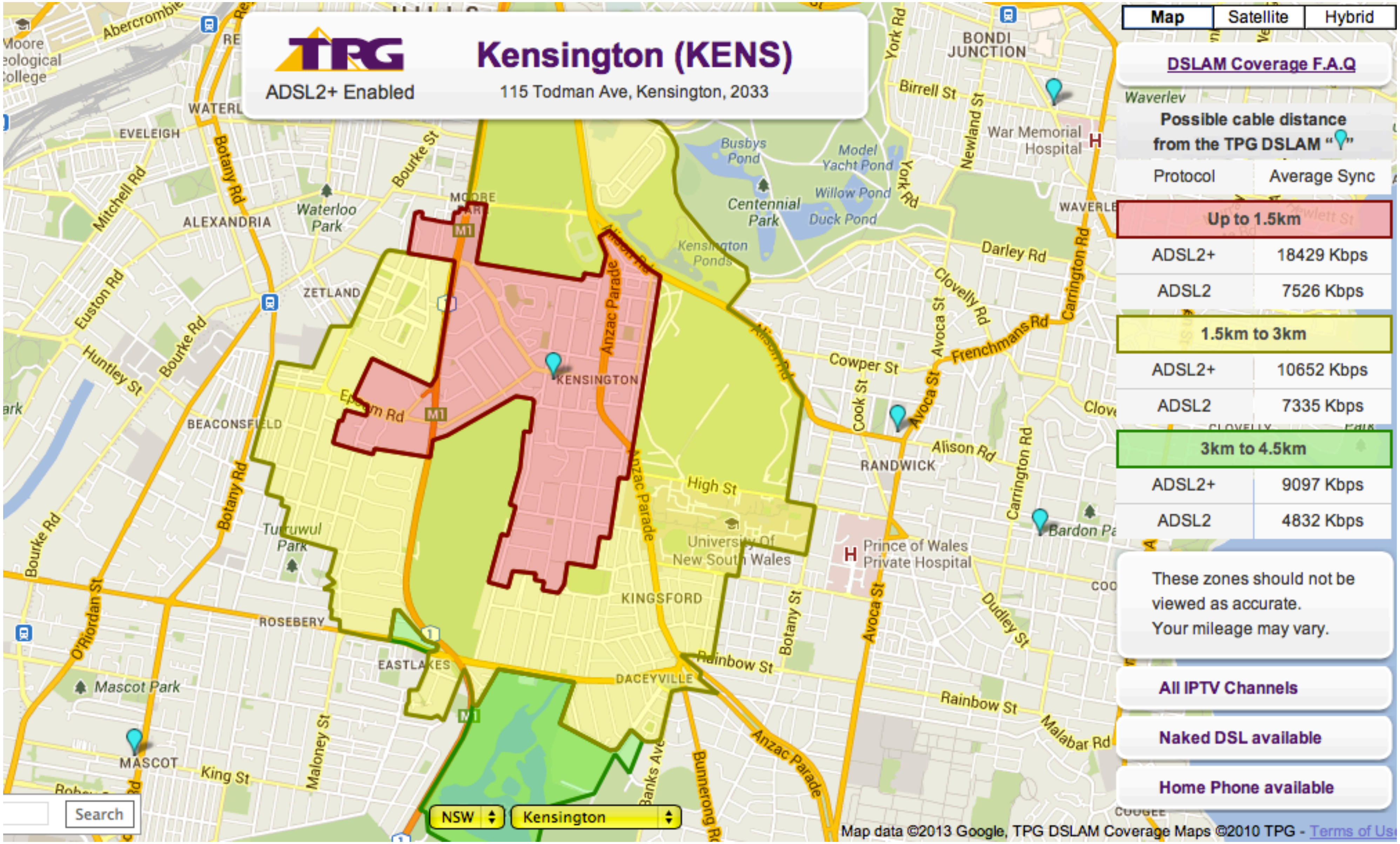


ADSL over POTS

voice, data transmitted at different frequencies over dedicated line to central office

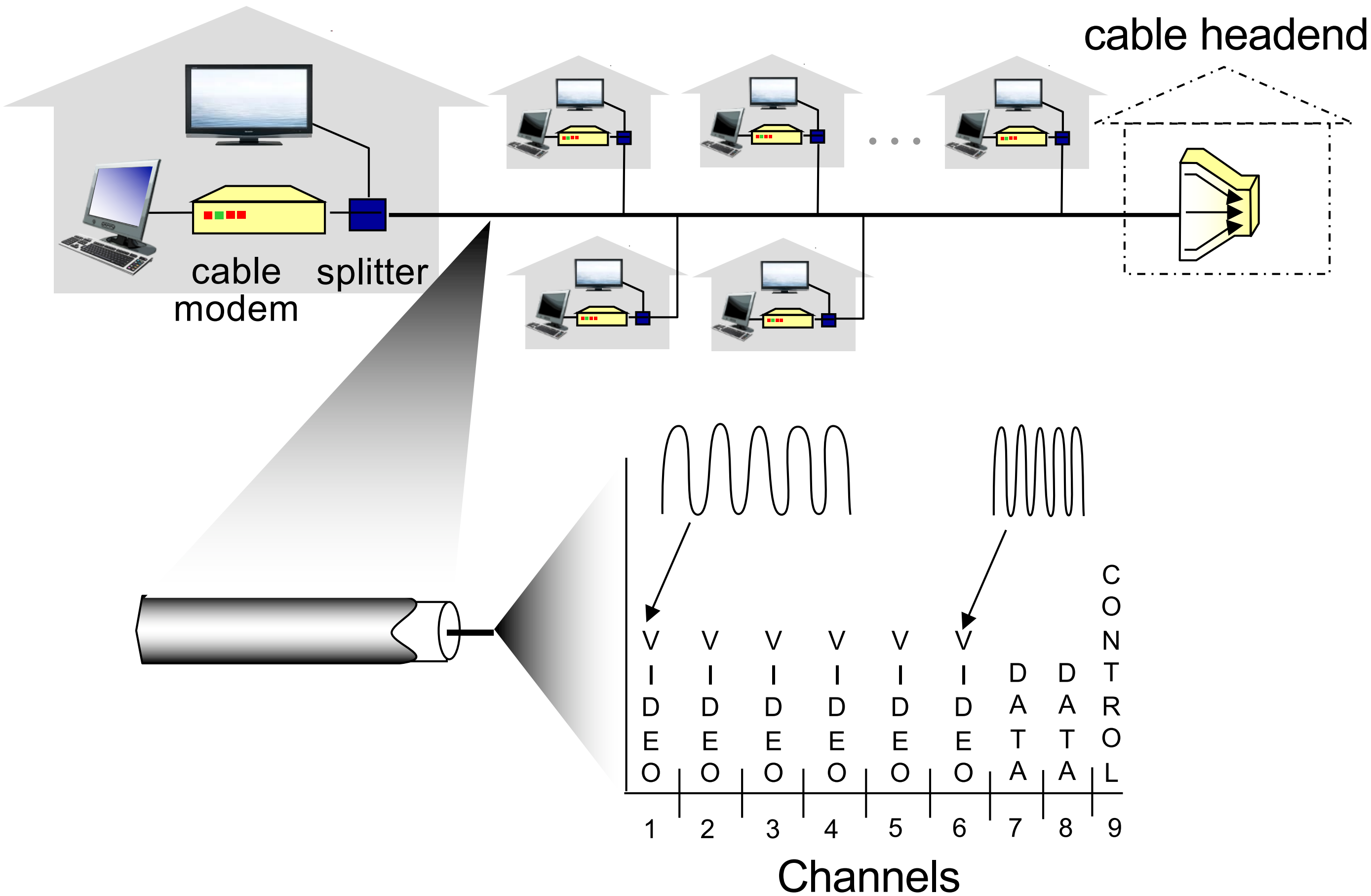
- Different data rates for upload and download (ADSL)
  - < 2.5 Mbps upstream transmission rate (typically < 1 Mbps)
  - < 24 Mbps downstream transmission rate (typically < 10 Mbps)

Access net: digital subscriber line (DSL)



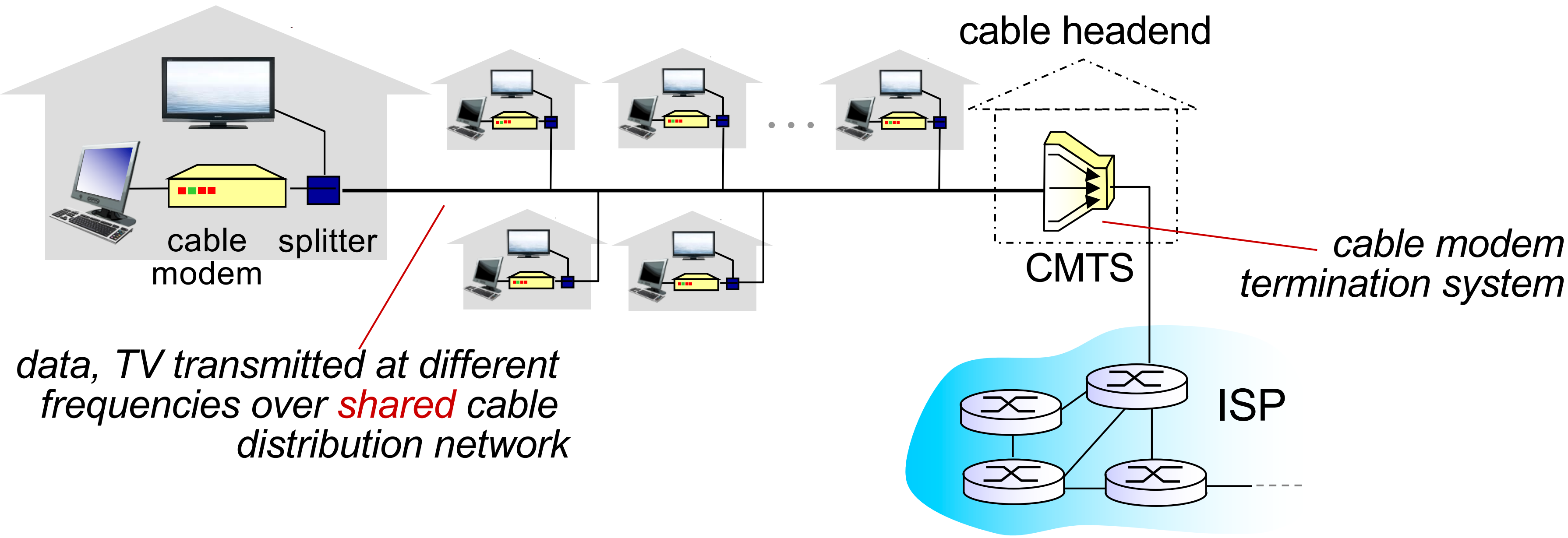
电缆

Access net: cable network



frequency division multiplexing: different channels transmitted in different frequency bands

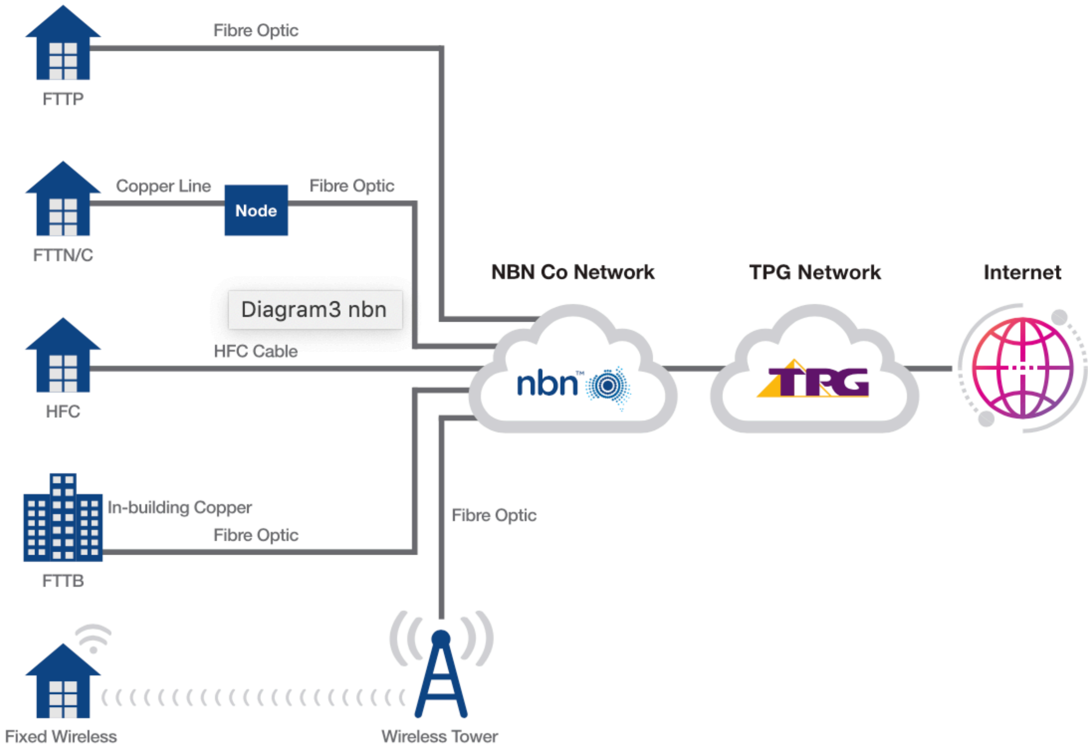
Access net: cable network

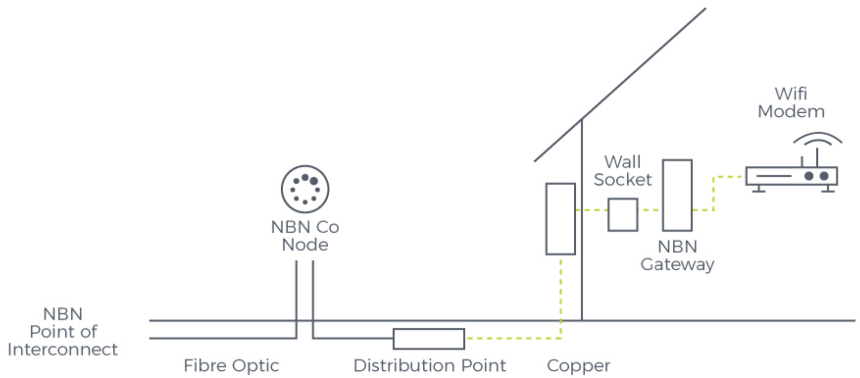


- HFC: hybrid fiber coax
  - asymmetric: up to 30Mbps downstream transmission rate, 2 Mbps upstream transmission rate
- network of cable, fiber attaches homes to ISP router
  - homes *share access network* to cable headend
  - unlike DSL, which has dedicated access to central office

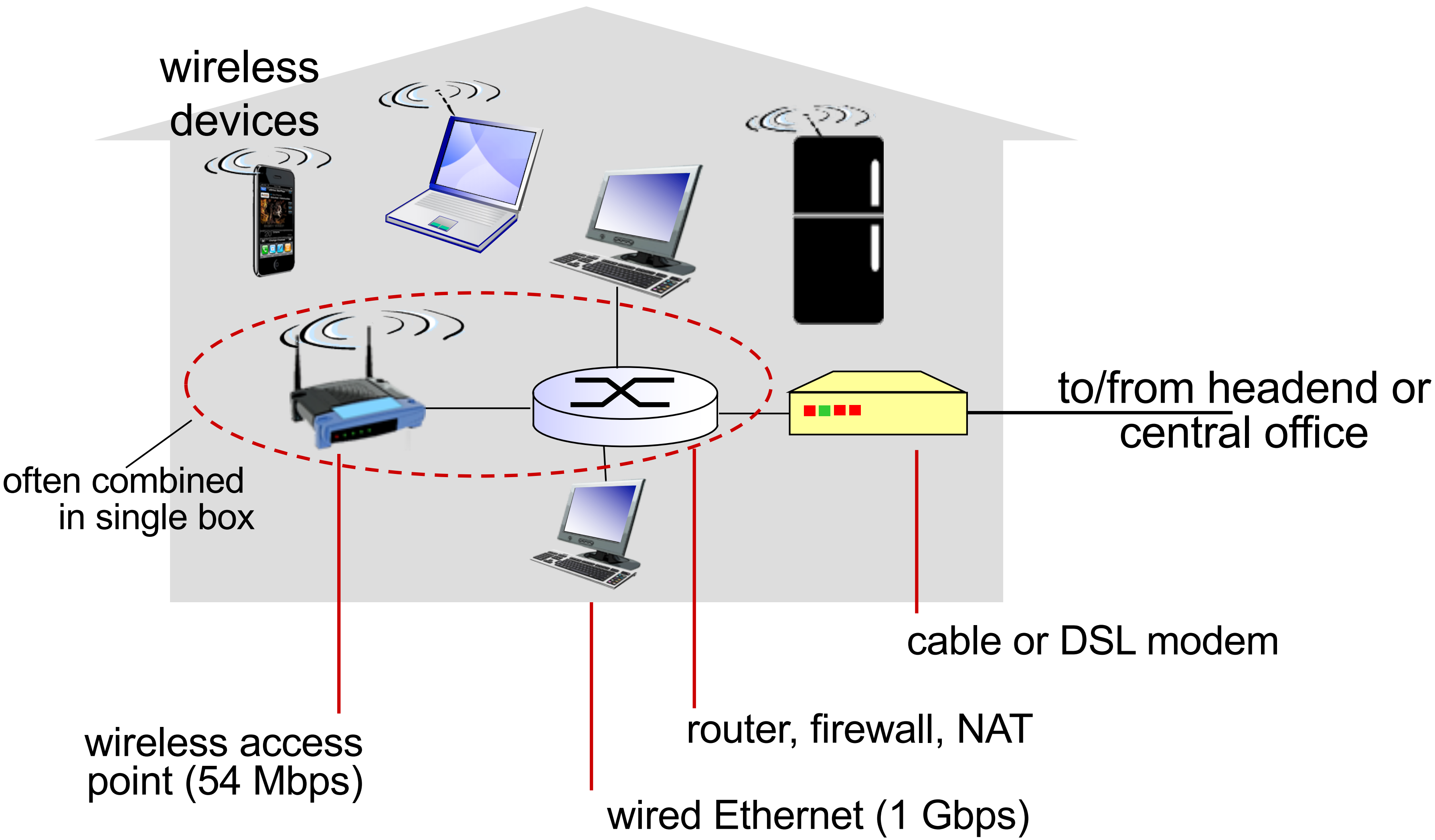
Fiber to the home/premise/curb

- Fully optical fiber path all the way to the home
  - e.g., NBN, Google, Verizon FIOS
  - ~30 Mbps to 1Gbps

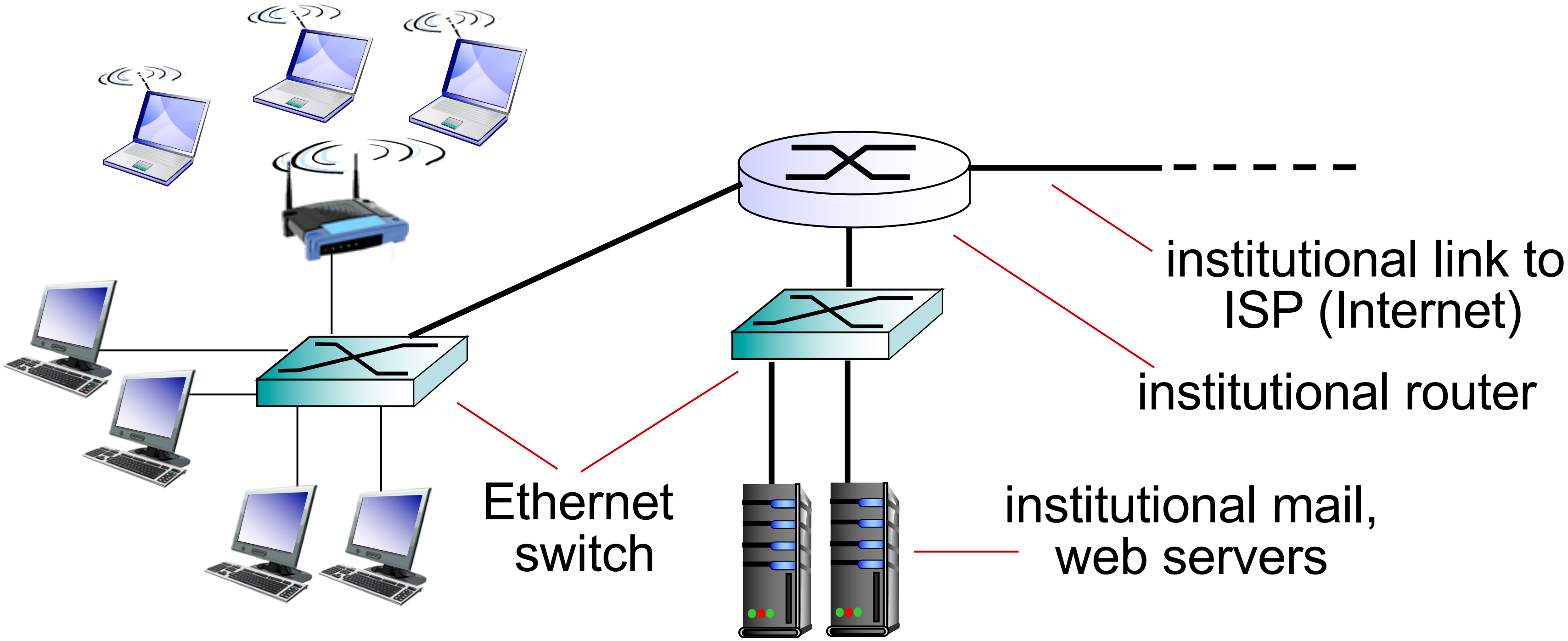




Access net: home network



Enterprise access networks (Ethernet)



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

Wireless access networks

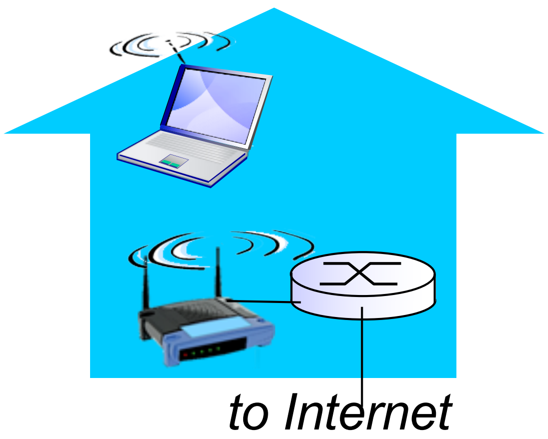
- shared *wireless* access network connects end system to router
  - via base station aka "access point"

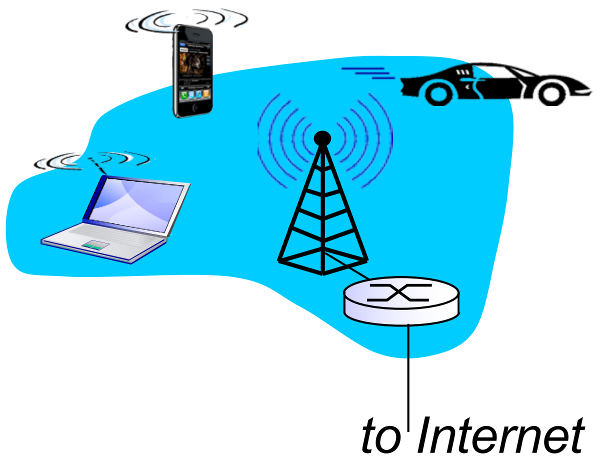
wireless LANs:

- within building (100 ft)
- 802.11b/g/n (WiFi): 11, 54, 300 Mbps transmission rate
- 802.11ac: 1 Gpbs(2.4GHz) + 4.34Gbps (5GHz)
- 802.11ax: WiFi 6

wide-area wireless access

- provided by telco (cellular) operator, 10's km
- between 10 and 100 Mbps
- 4G, 5G

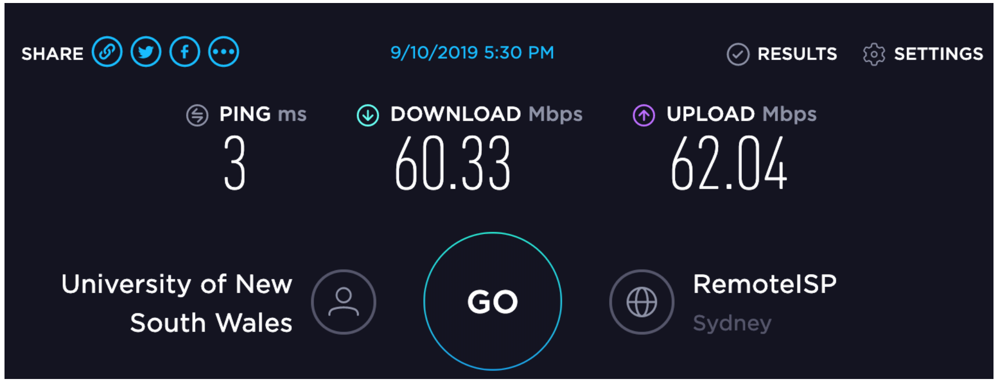




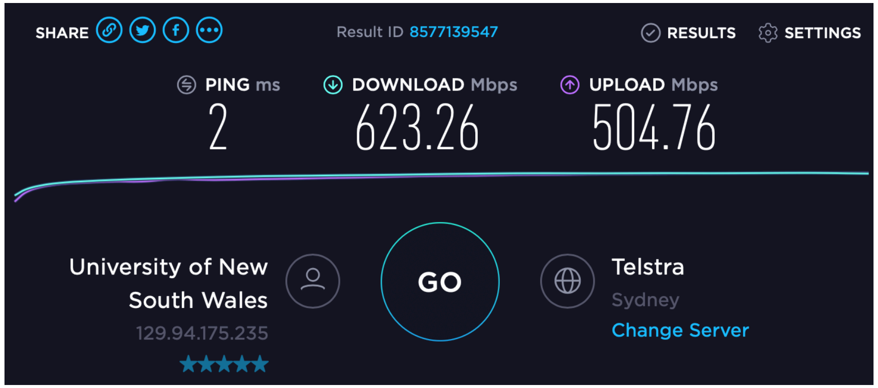
Sample results

Can you explain the differences?

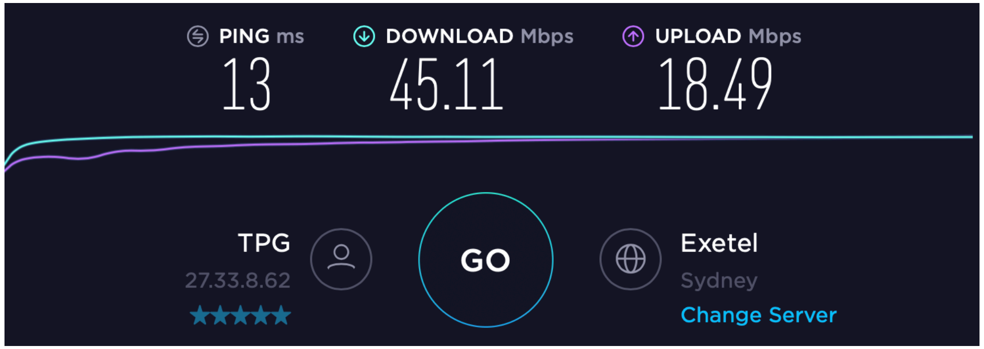
Uniwide



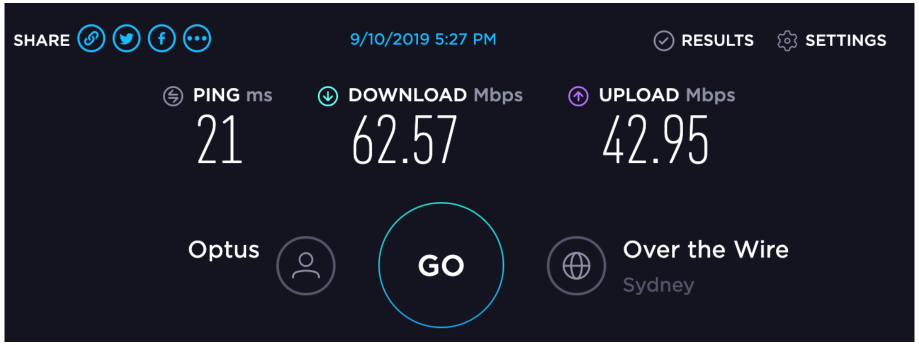
Wired Network @ CSE



FTTC + Cable + WiFi @ my home



4G Network



Physical media

Self Study

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

Physical media: twisted pair, coax, fiber

Self Study

twisted pair (TP)

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

coaxial cable:

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

fiber optic cable:

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

Physical media: radio

Self Study

- signal carried in electromagnetic spectrum, i.e., no physical "wire"
- propagation environment effects:
  - reflection
  - obstruction by objects
  - interference

radio link types:

- terrestrial microwave
  - e.g. up to 45 Mbps channels
- LAN (e.g., WiFi)
  - 11Mbps, 54 Mbps, 450 Mbps, Gbps
- wide-area (e.g., cellular)
  - 4G cellular: ~ 10 Mbps
- satellite
  - Kbps to 45Mbps channel (or multiple smaller channels)
  - 270 msec end-end delay
  - geosynchronous versus low earth-orbiting (LEO)

1. Introduction: 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

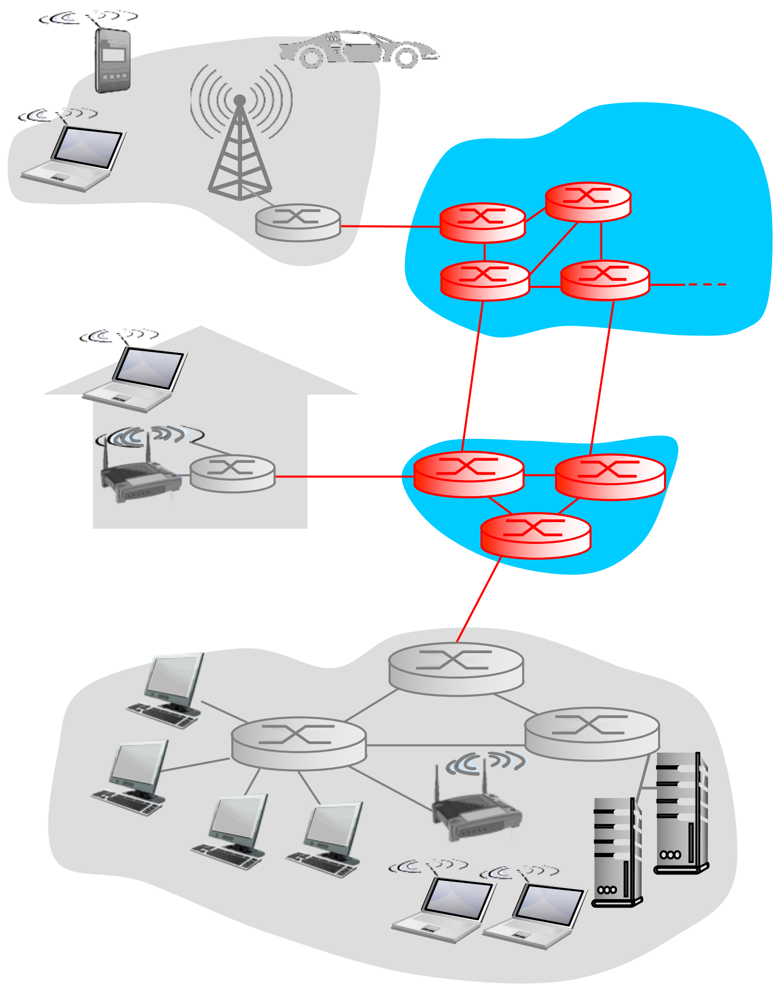
1.6 networks under attack: security

1.7 history

The network core

- mesh of interconnected routers/switches
- Two forms of switched networks:
  - Circuit switching: used in the legacy telephone networks

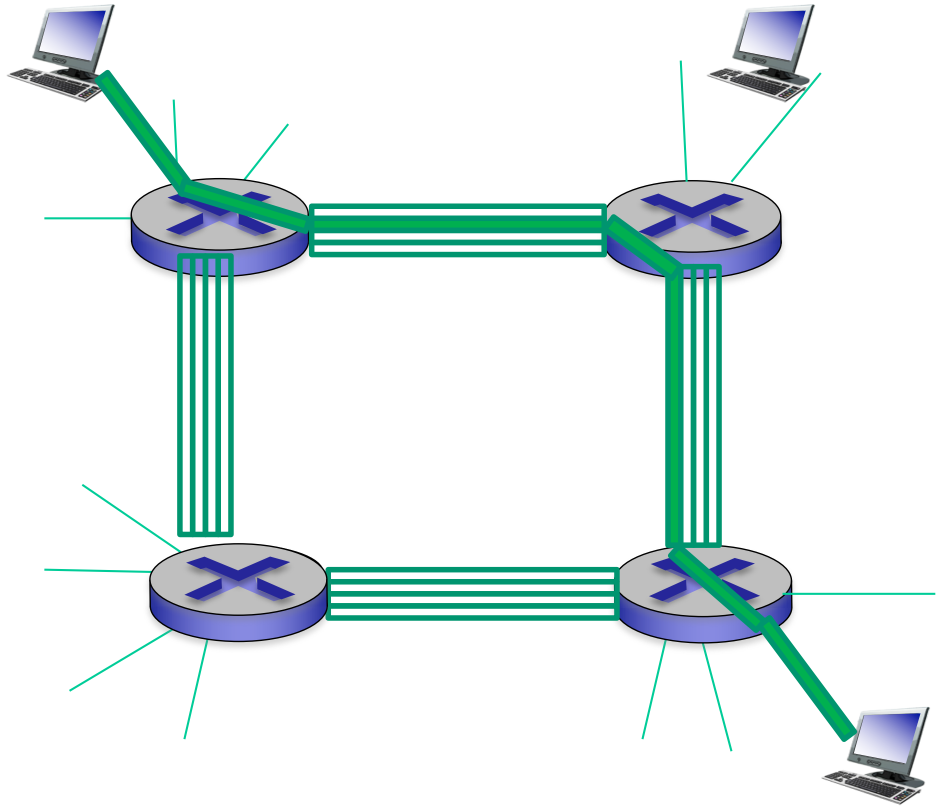
  - Packet switching: used in the Internet



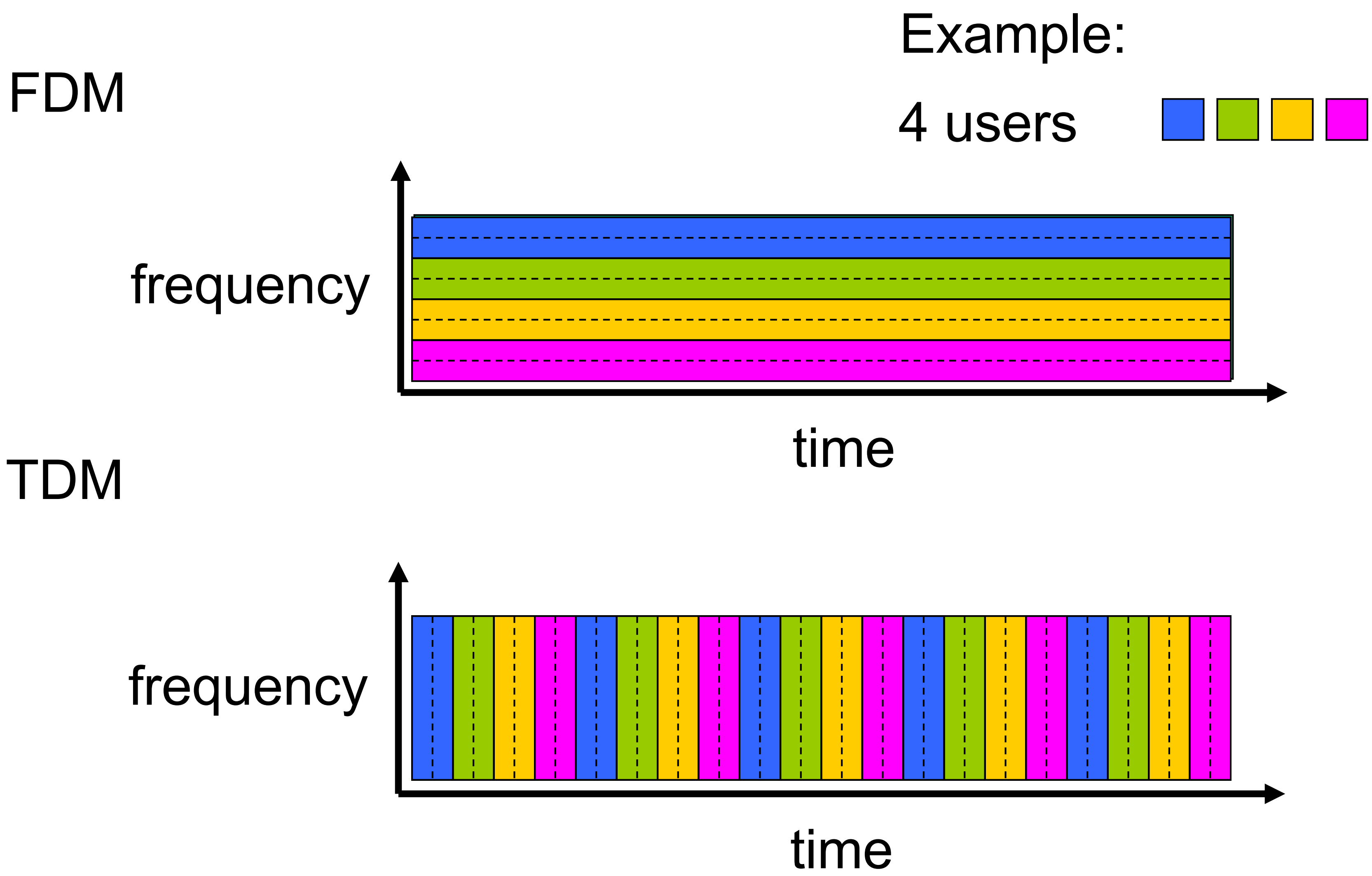
Circuit Switching

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

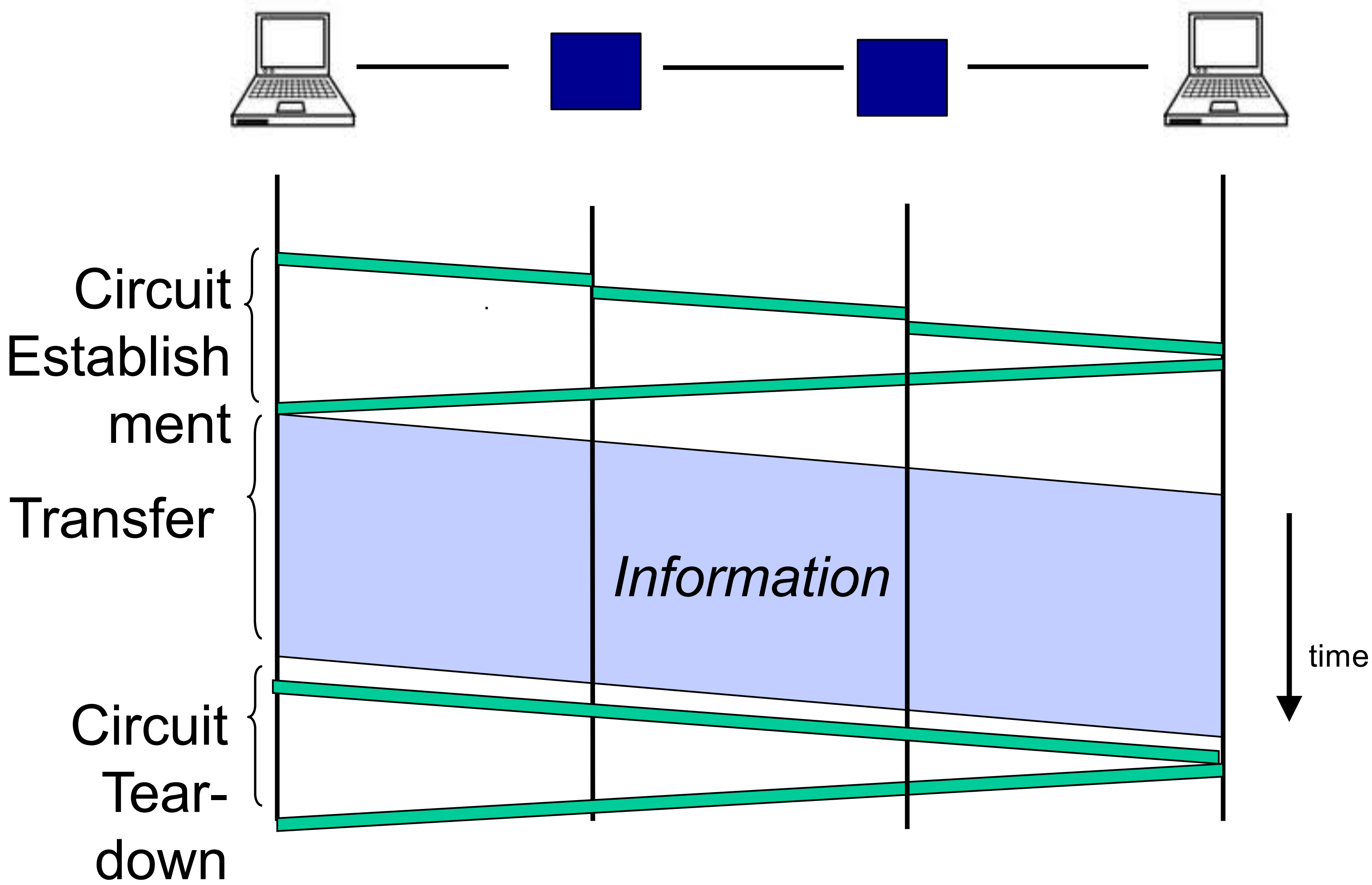
- in diagram, each link has four circuits.
  - call gets 2nd circuit in top link and 1st circuit in right link.
- dedicated resources: no sharing
  - circuit-like (guaranteed) performance
- circuit segment idle if not used by call *(no sharing)*
- commonly used in traditional telephone networks



Circuit switching: FDM versus TDM



Timing in Circuit Switching



可行的

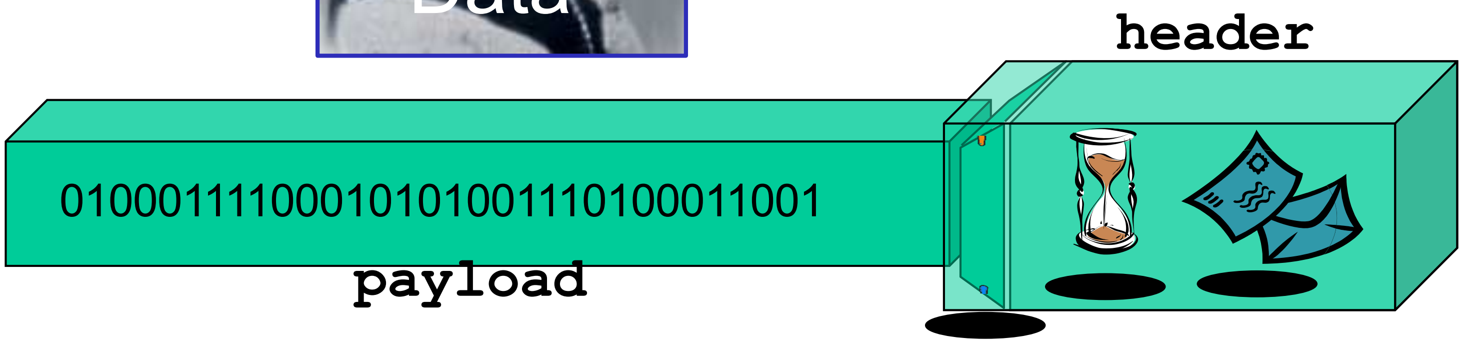
Why circuit switching is not feasible?

- Inefficient
  - Computer communications tends to be very bursty. For example viewing a sequence of web pages
  - Dedicated circuit cannot be used or shared in periods of silence
  - Cannot adopt to network dynamics
- Fixed data rate
  - Computers communicate at very diverse rates. For example viewing a video vs using telnet or web browsing
  - Fixed data rate is not useful
    固定的
- Connection state maintenance
  - Requires per communication state to be maintained that is a considerable overhead
  - Not scalabale

Packet Switching

- Data is sent as chunks of formatted bits (Packets)
- Packets consist of a "header" and "payload"

1. Internet Address
2. Age (TTL)
3. Checksum to protect header



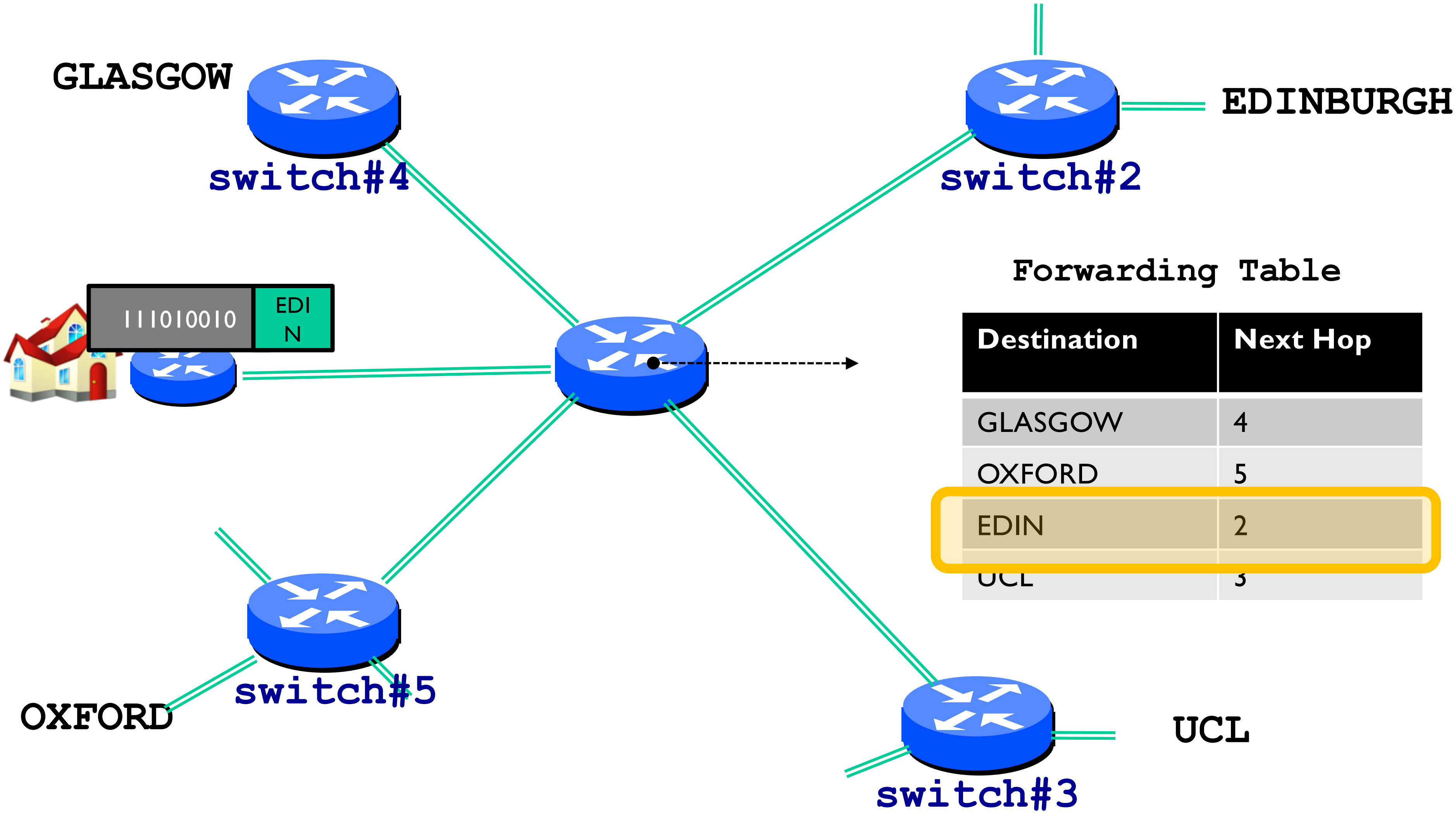
Packet Switching

- Data is sent as chunks of formatted bits (Packets)
- Packets consist of a "header" and "payload"
  - payload is the data being carried
  - header holds instructions to the network for how to handle packet (think of the header as an API)

Packet Switching

- Data is sent as chunks of formatted bits (Packets)
- Packets consist of a “header” and “payload”
- Switches “forward” packets based on their headers

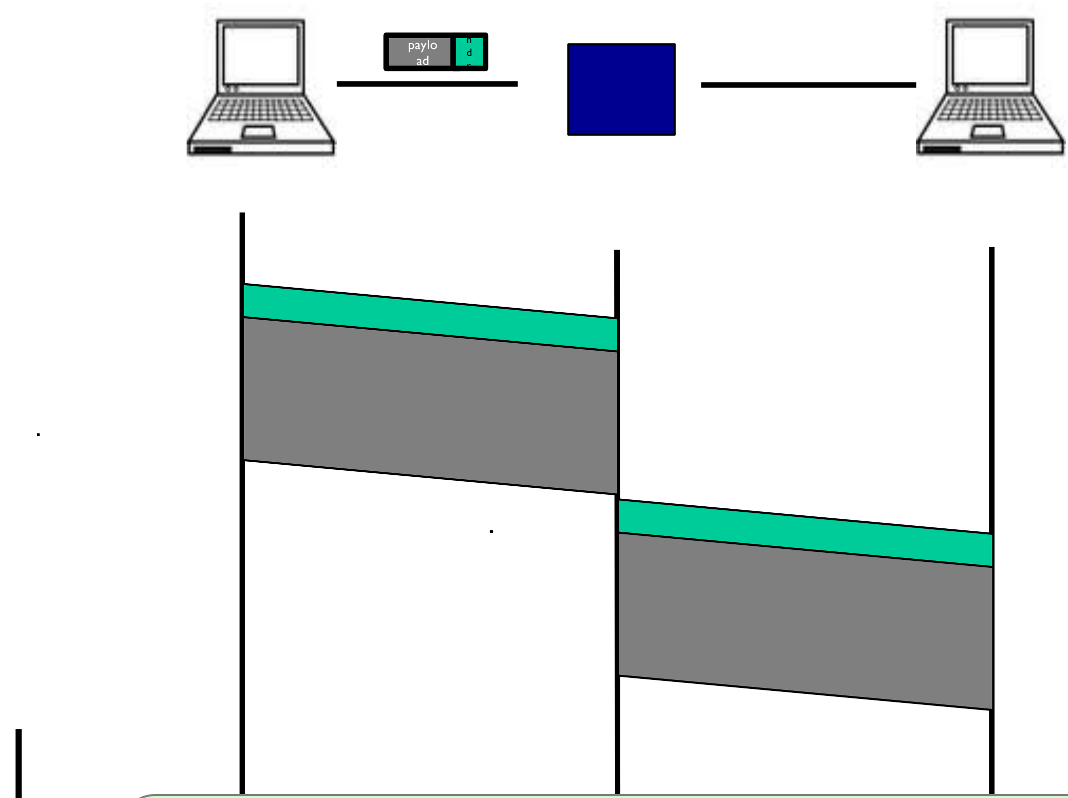
Switches forward packets



Forwarding Table

Destination	Next Hop
GLASGOW	4
OXFORD	5
EDIN	2
UCL	3

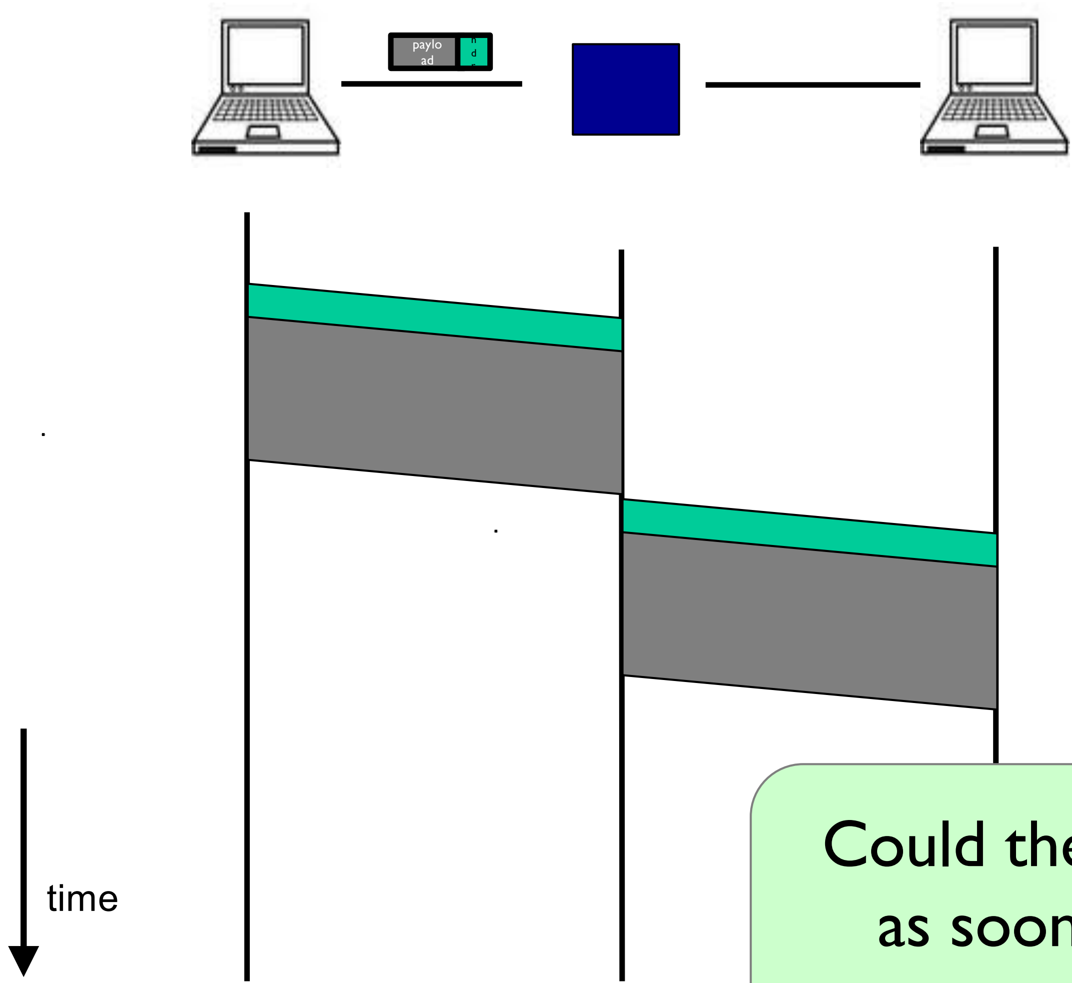
Timing in Packet Switching



What about the time to process the packet at the switch?

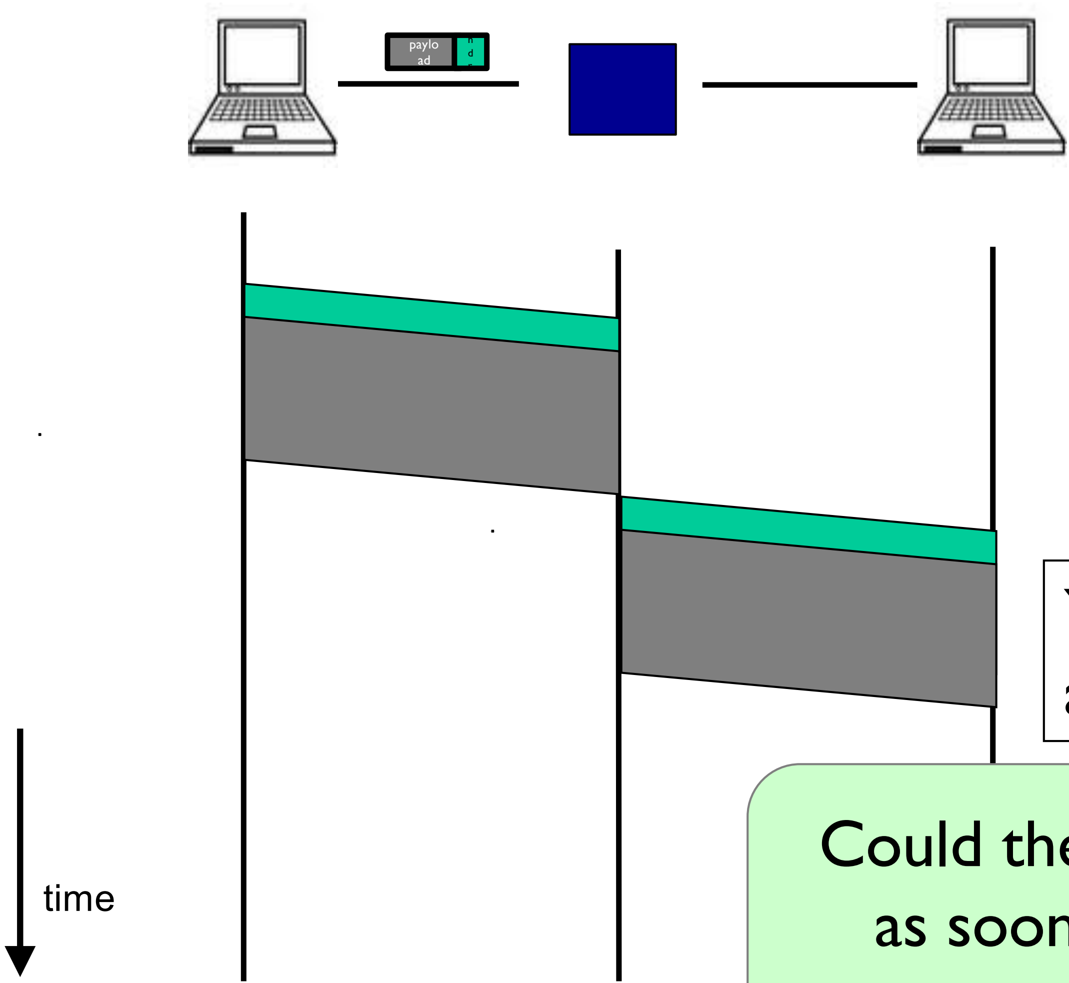
- We'll assume it's relatively negligible (mostly true)

Timing in Packet Switching



Could the switch start transmitting as soon as it has processed the header?

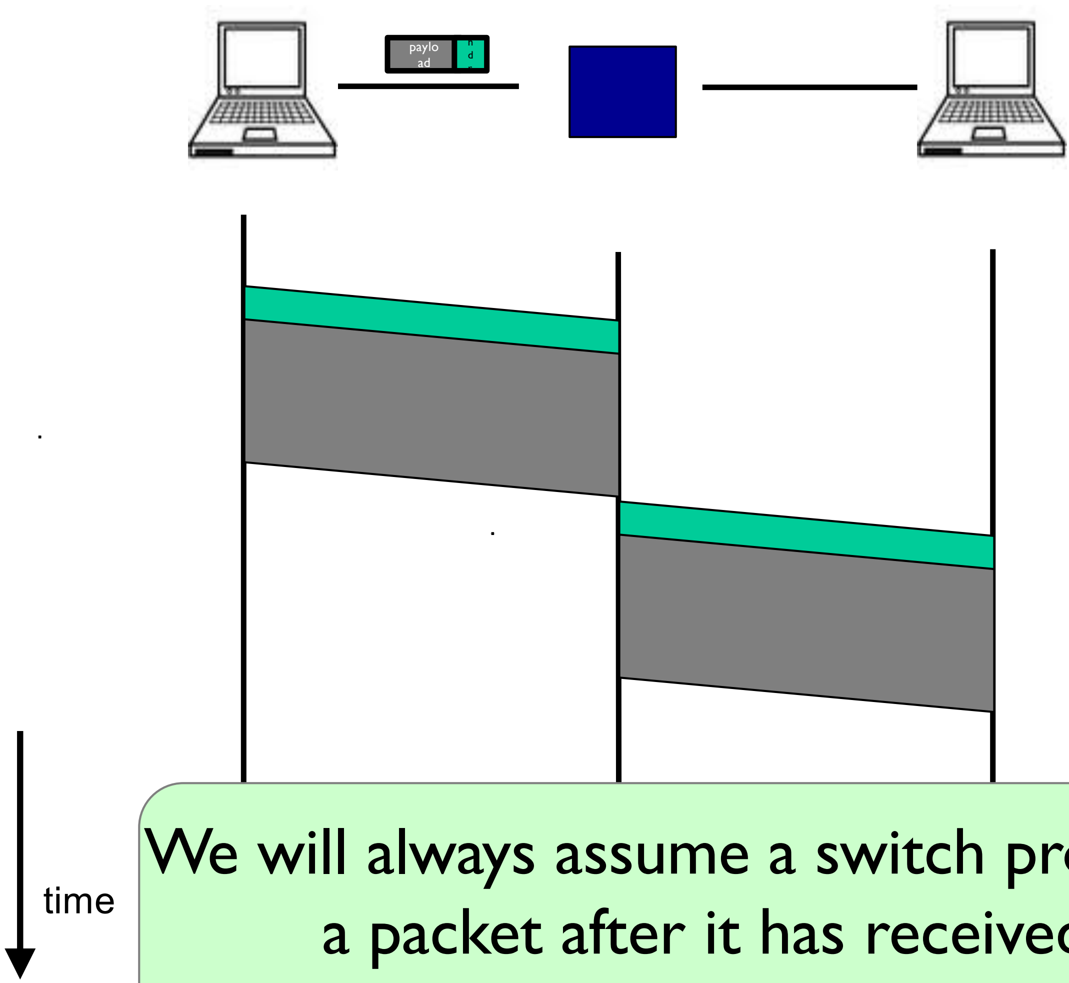
Timing in Packet Switching



Yes! This would be called a "cut through" switch

Could the switch start transmitting as soon as it has processed the header?

Timing in Packet Switching



We will always assume a switch processes/forwards a packet after it has received it entirely.
This is called "store and forward" switching

Packet Switching

- Data is sent as chunks of formatted bits (Packets)
- Packets consist of a “header” and “payload”
- Switches “forward” packets based on their headers

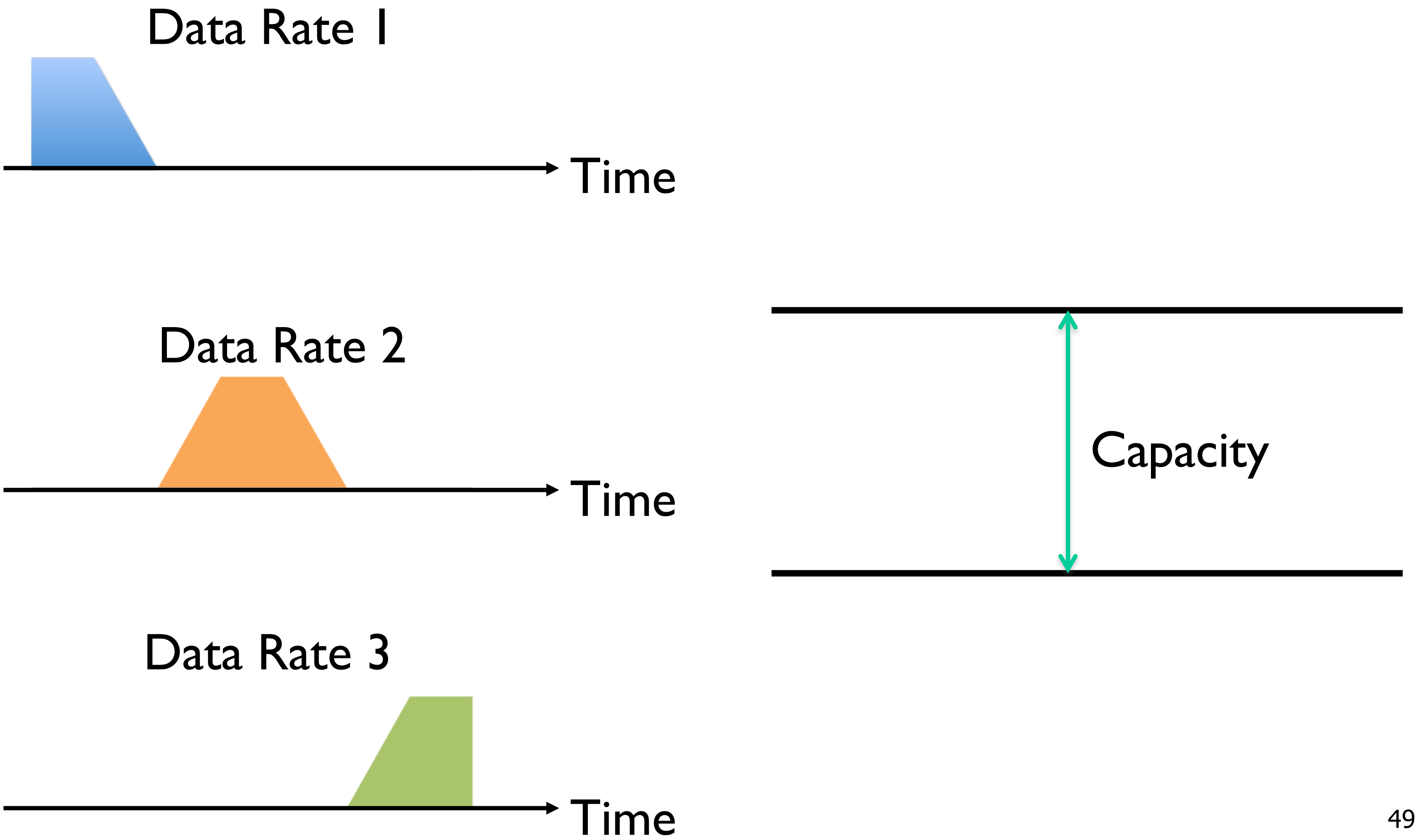
Packet Switching

- Data is sent as chunks of formatted bits (Packets)
- Packets consist of a "header" and "payload"
- Switches "forward" packets based on their headers
- Each packet travels independently
  - no notion of packets belonging to a "circuit"

Packet Switching

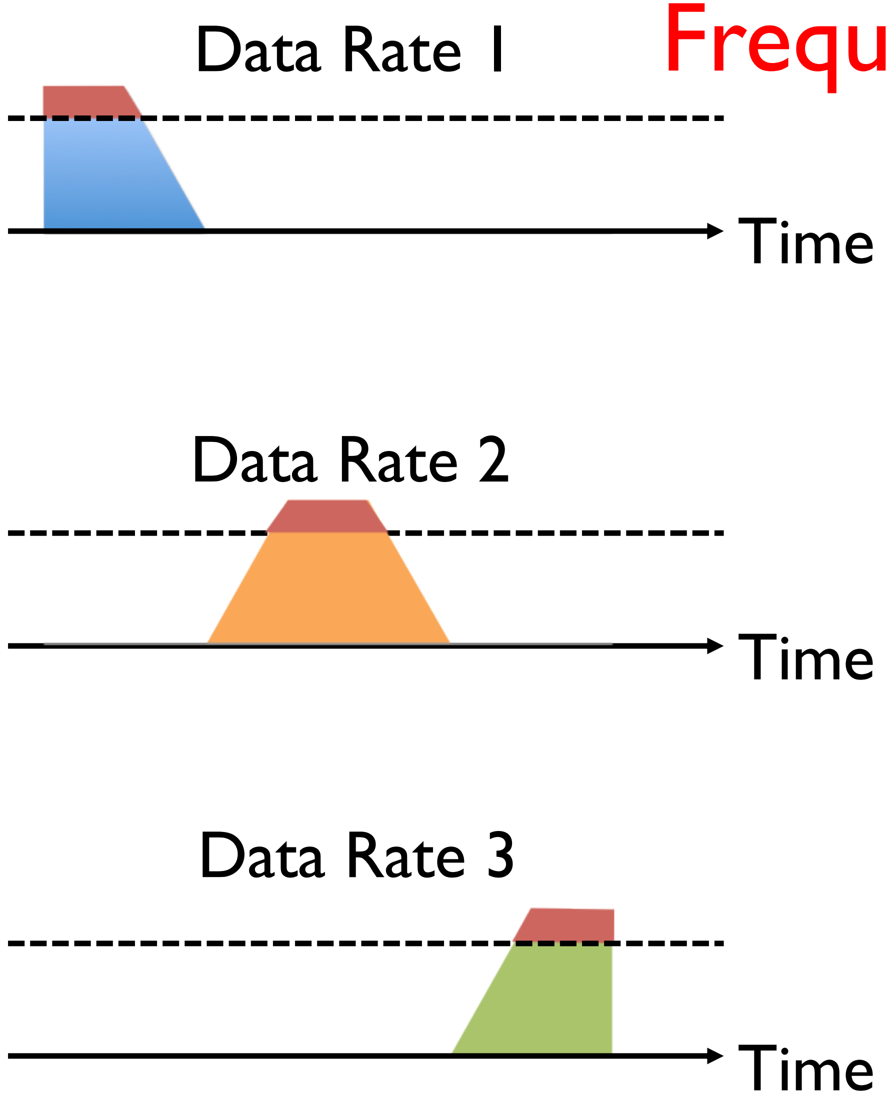
- Data is sent as chunks of formatted bits (Packets)
- Packets consist of a "header" and "payload"
- Switches "forward" packets based on their headers
- Each packet travels independently
- No link resources are reserved in advance. Instead packet switching leverages statistical multiplexing
  多路传输

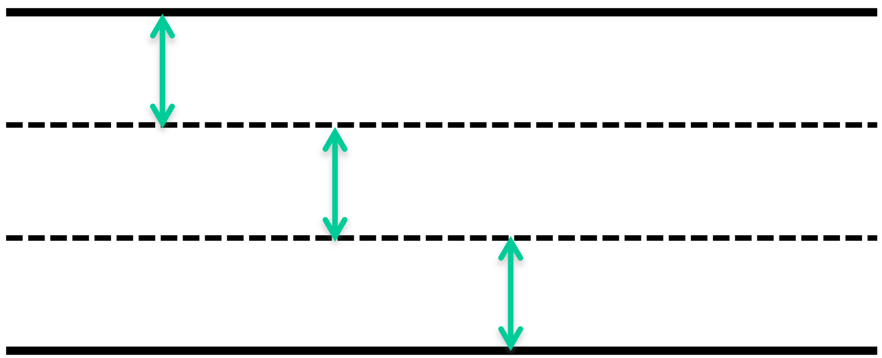
Three Flows with Bursty Traffic



When Each Flow Gets 1/3rd of Capacity

Frequent Overloading





When Flows Share Total Capacity

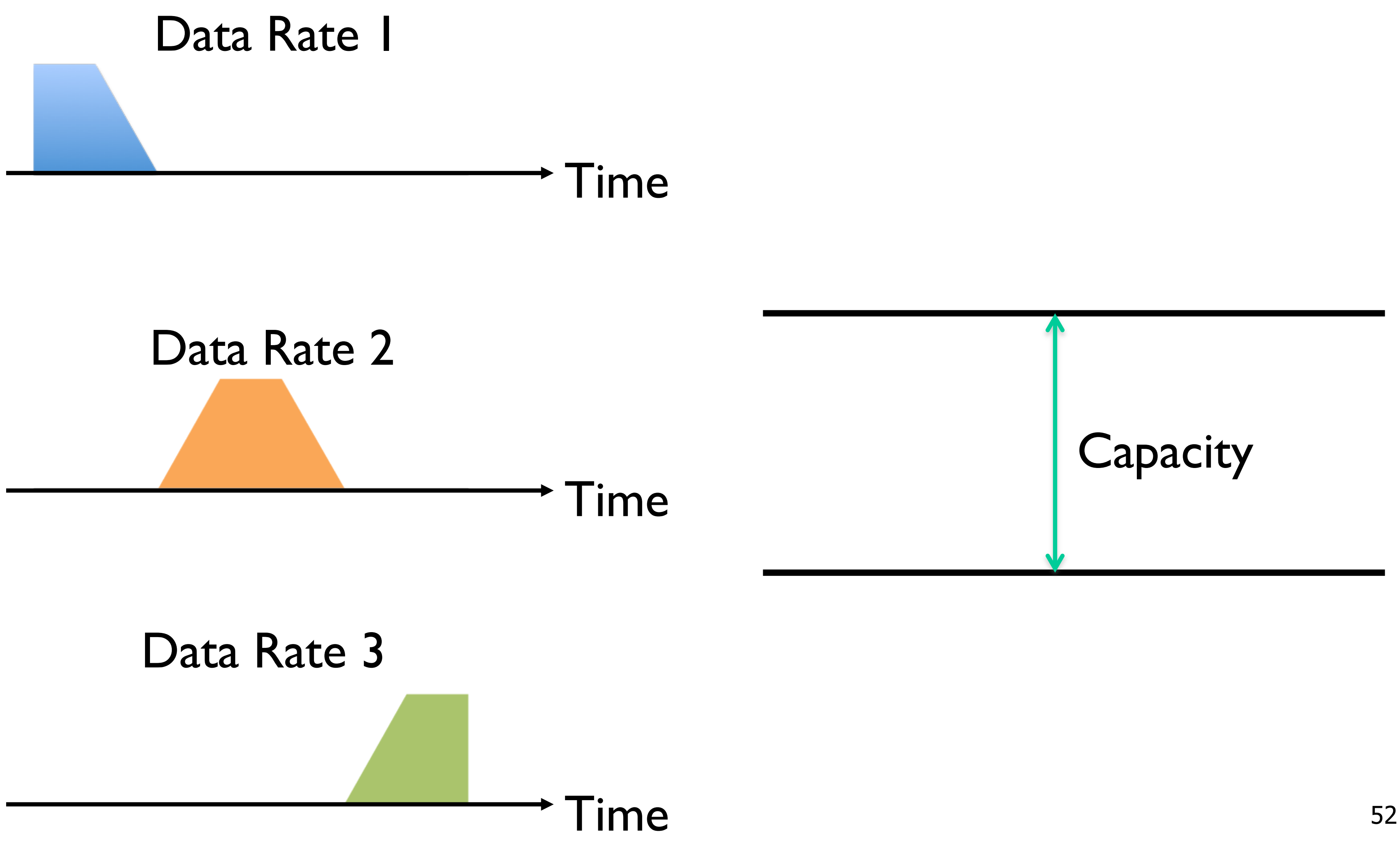
超载

No Overloading

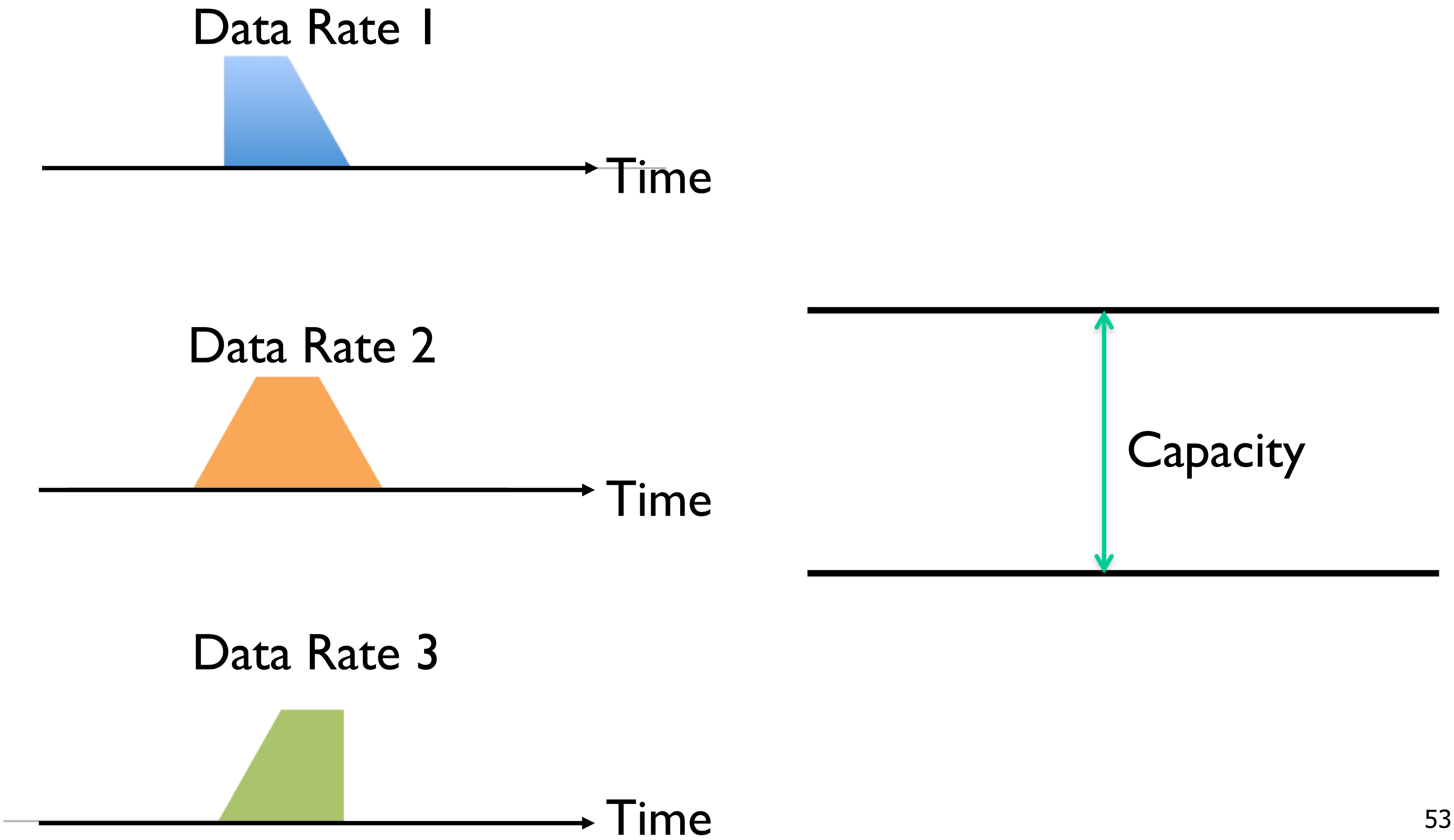
Statistical multiplexing relies on the assumption that not all flows burst at the same time.

Very similar to insurance, and has same failure case

Three Flows with Bursty Traffic

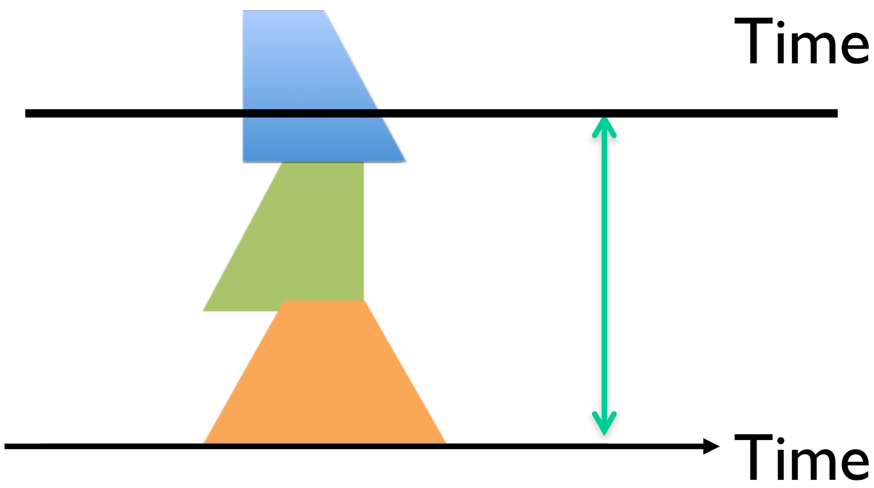


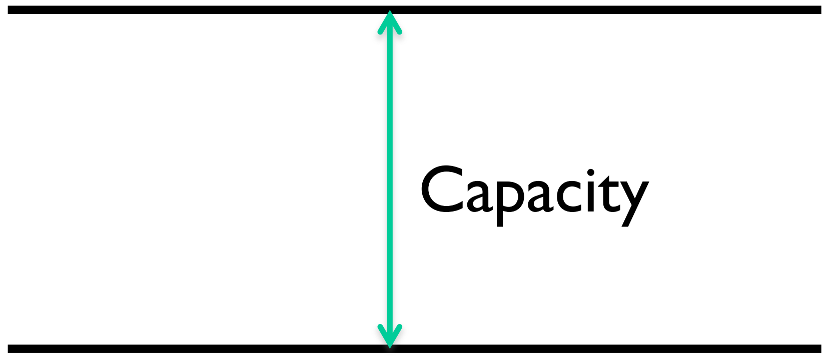
Three Flows with Bursty Traffic



Three Flows with Bursty Traffic

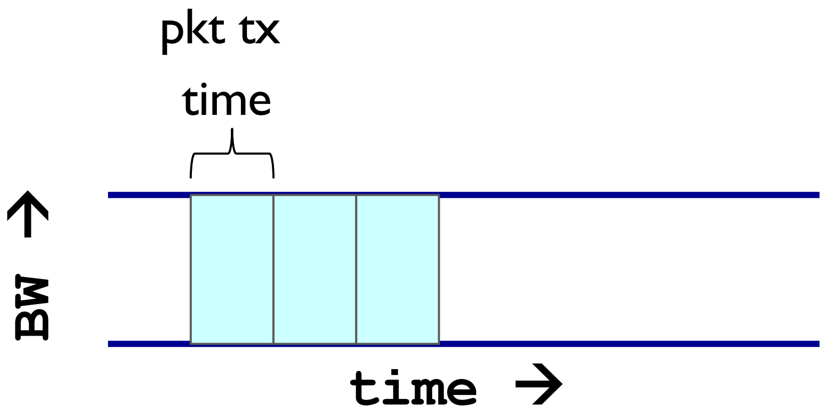
Data Rate 1+2+3 >> Capacity



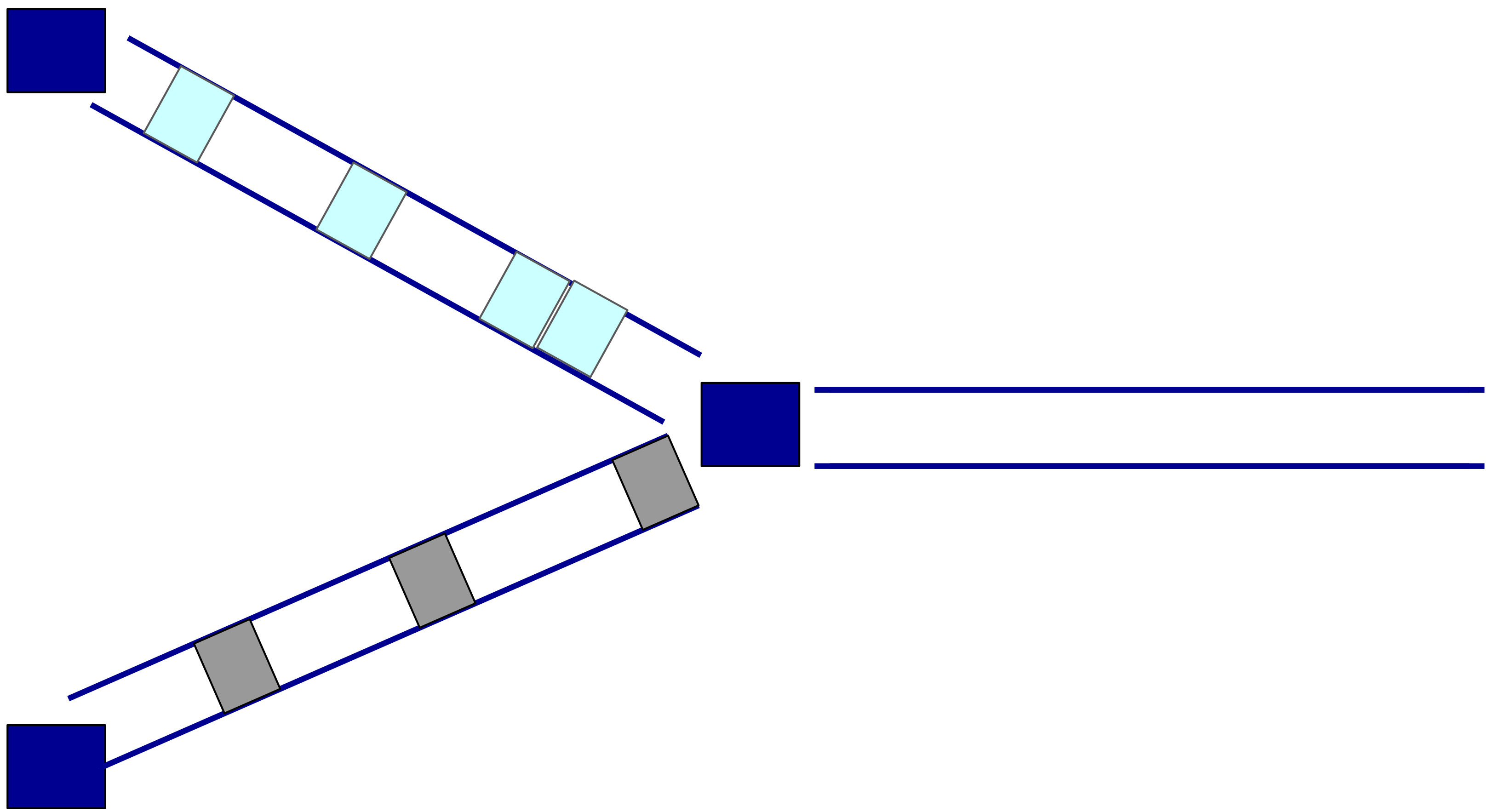


What do we do under overload?

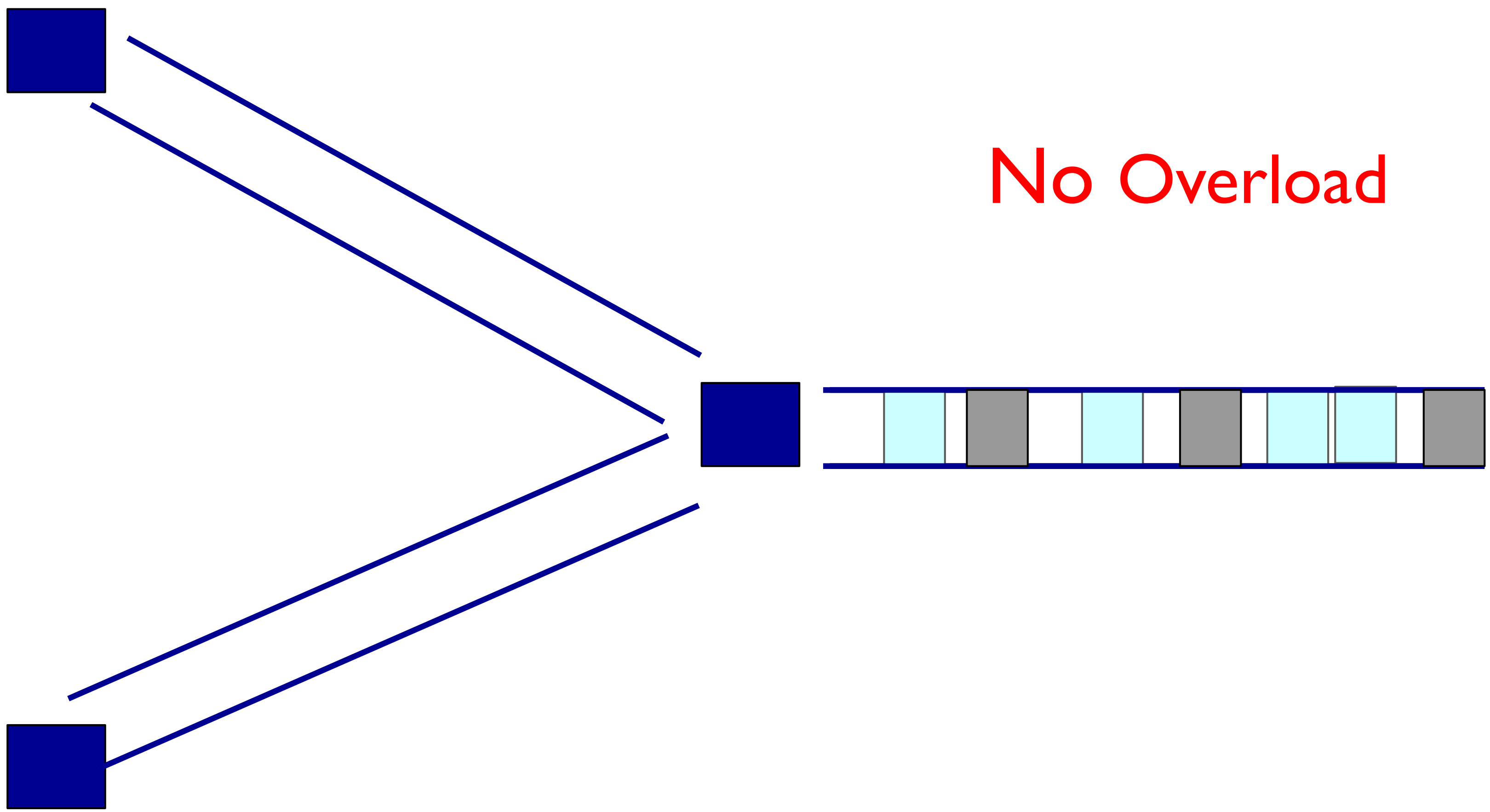
Statistical multiplexing: pipe view



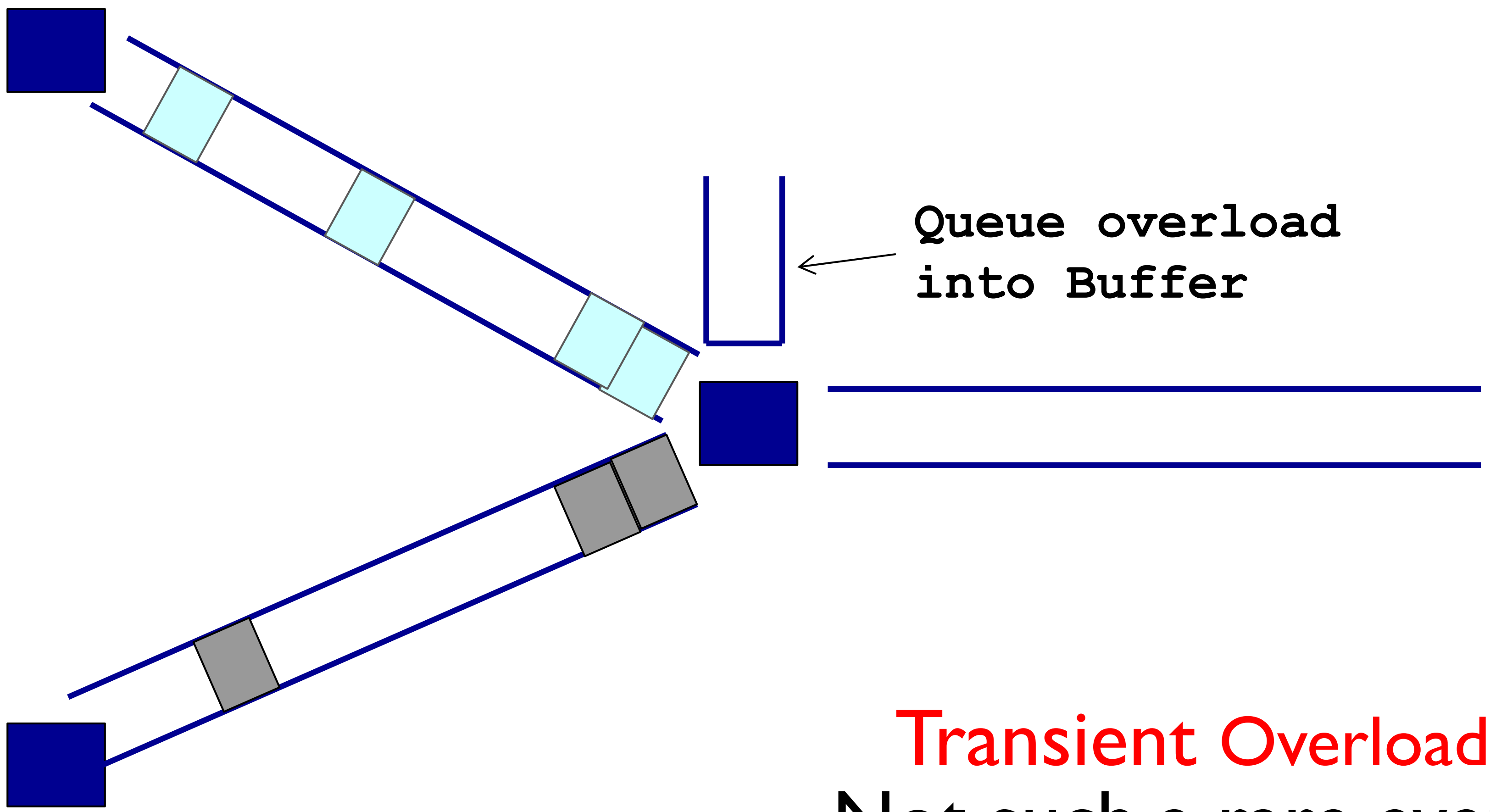
Statistical multiplexing: pipe view



Statistical multiplexing: pipe view

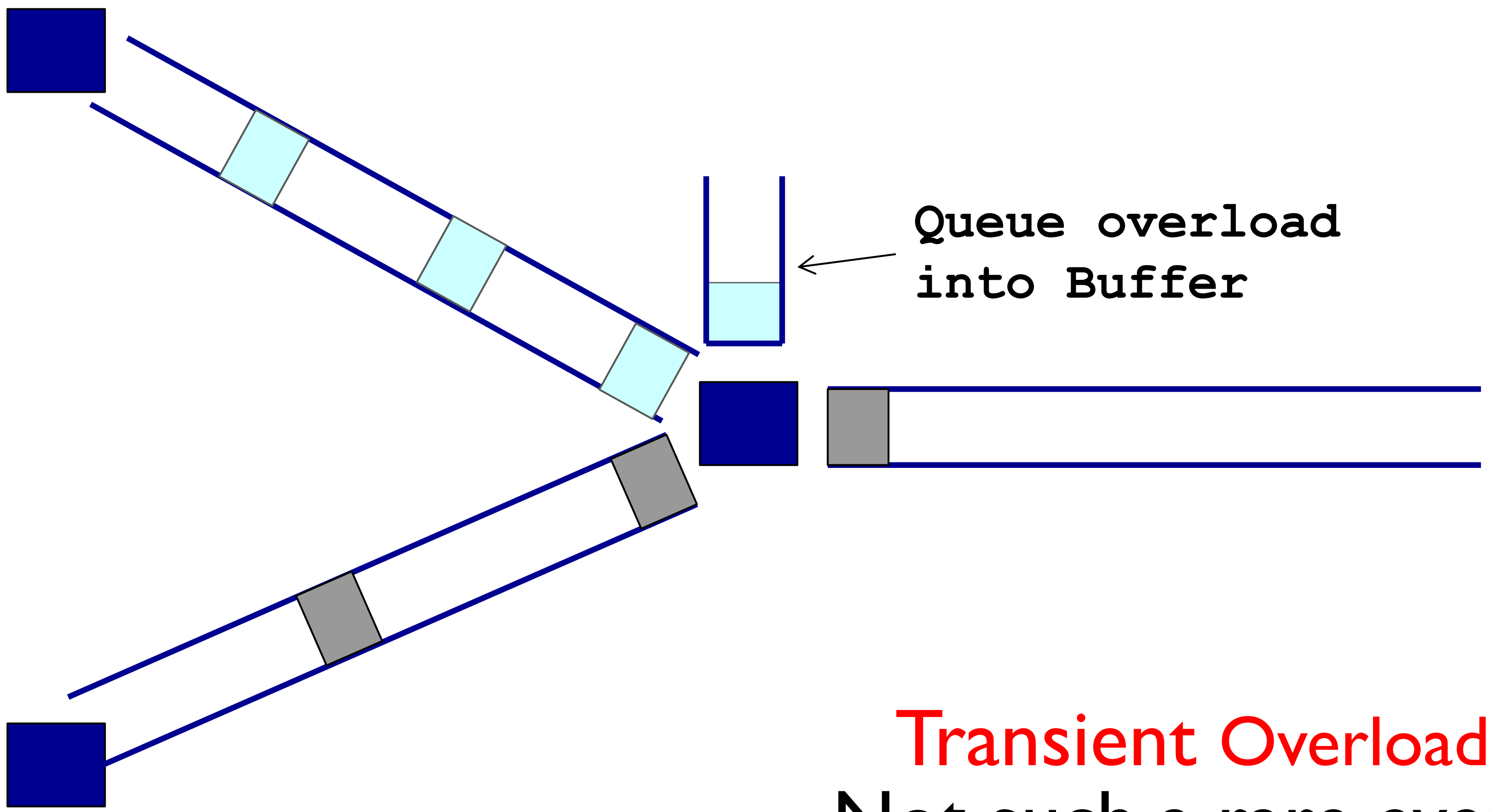


Statistical multiplexing: pipe view



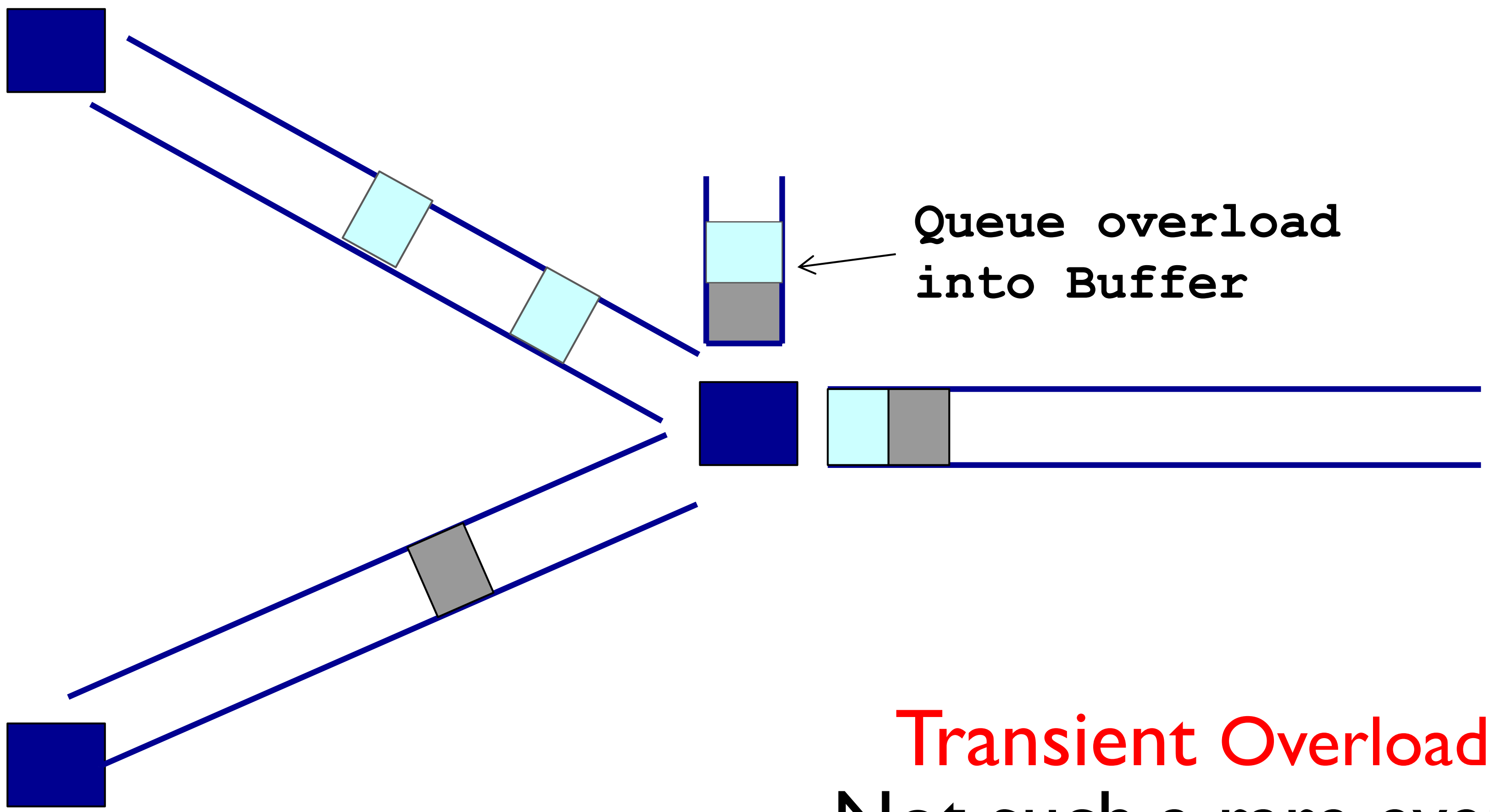
Transient Overload
Not such a rare event

Statistical multiplexing: pipe view



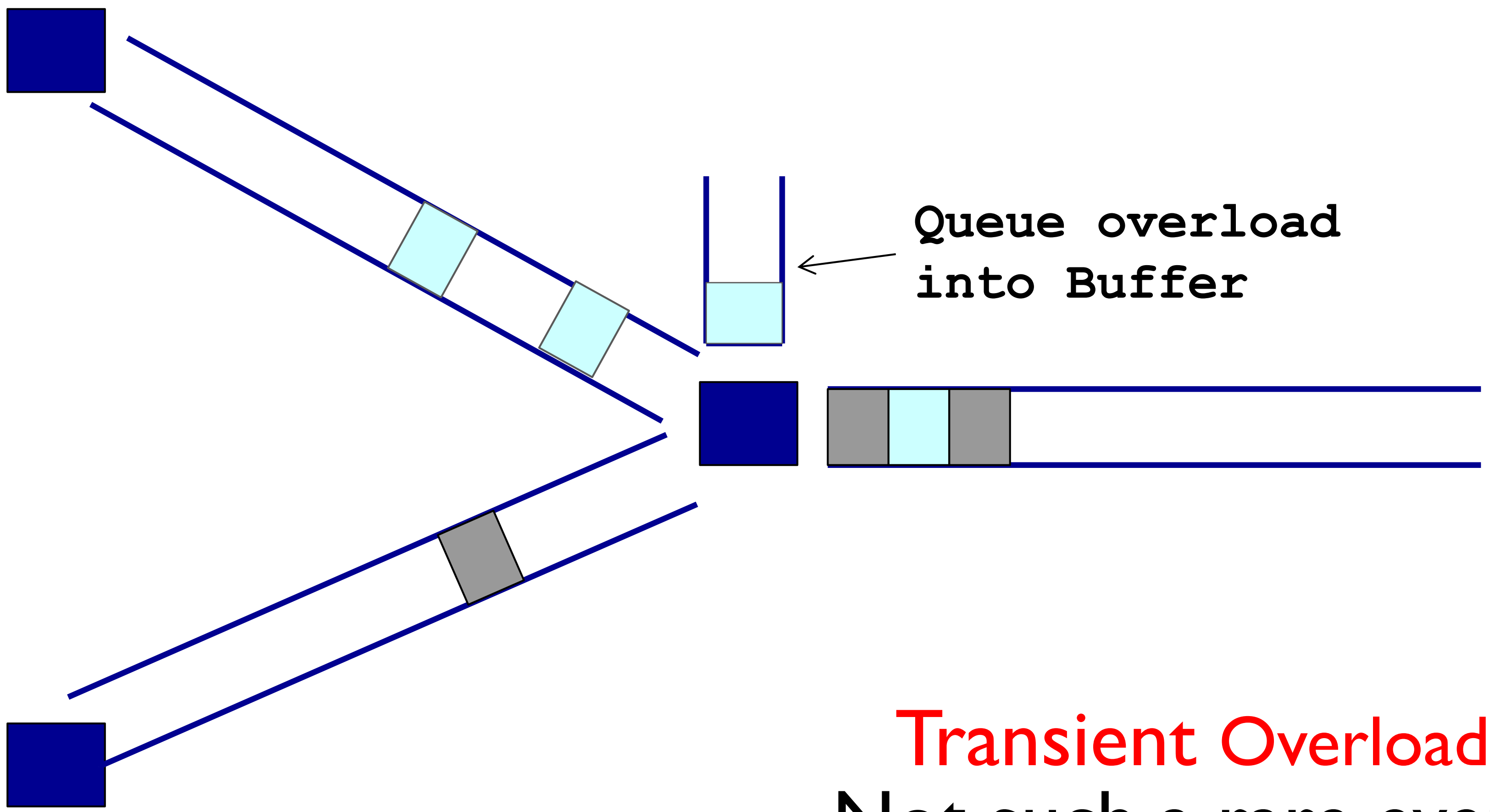
Transient Overload
Not such a rare event

Statistical multiplexing: pipe view



Transient Overload
Not such a rare event

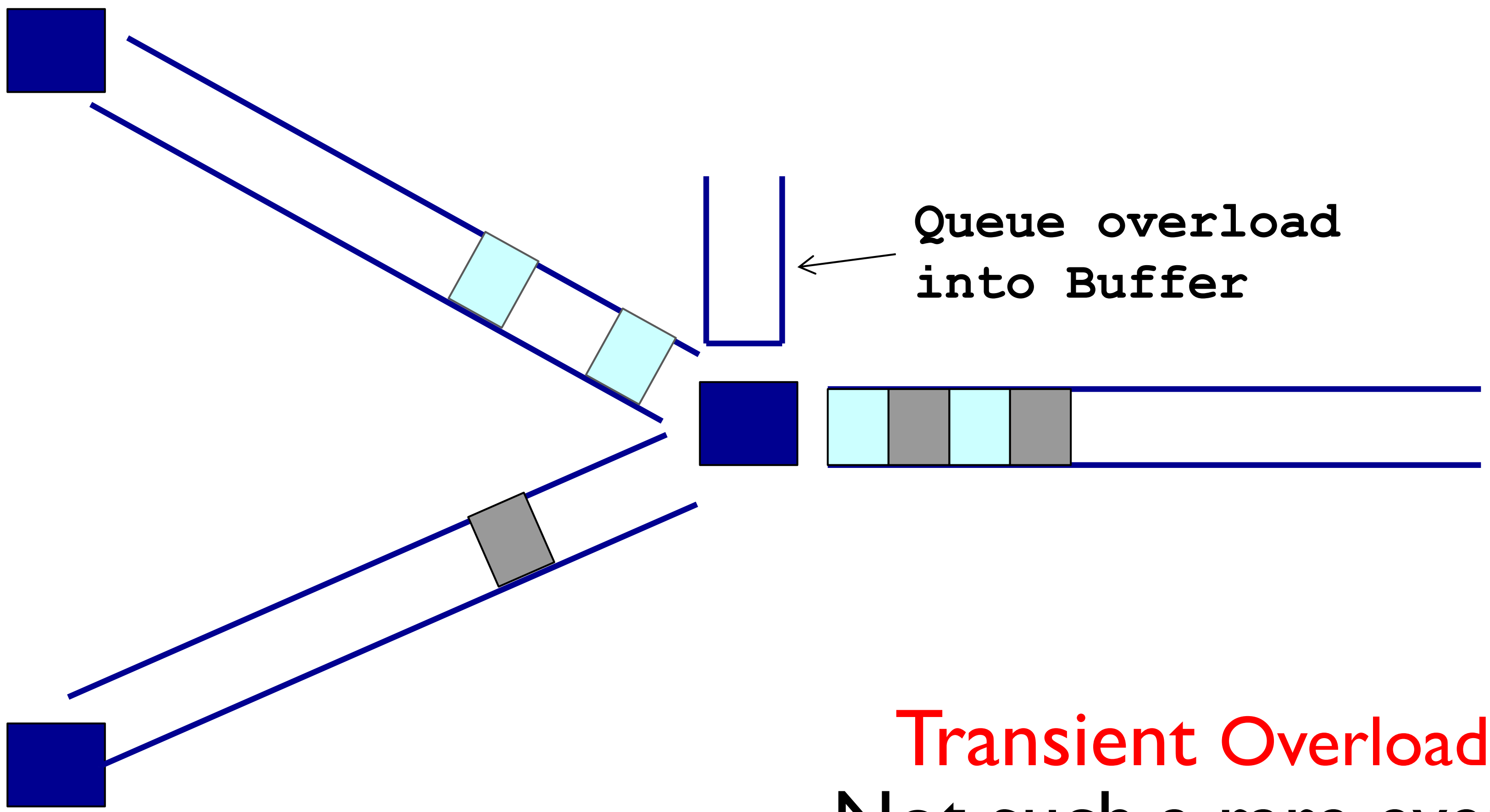
Statistical multiplexing: pipe view



Transient Overload

Not such a rare event

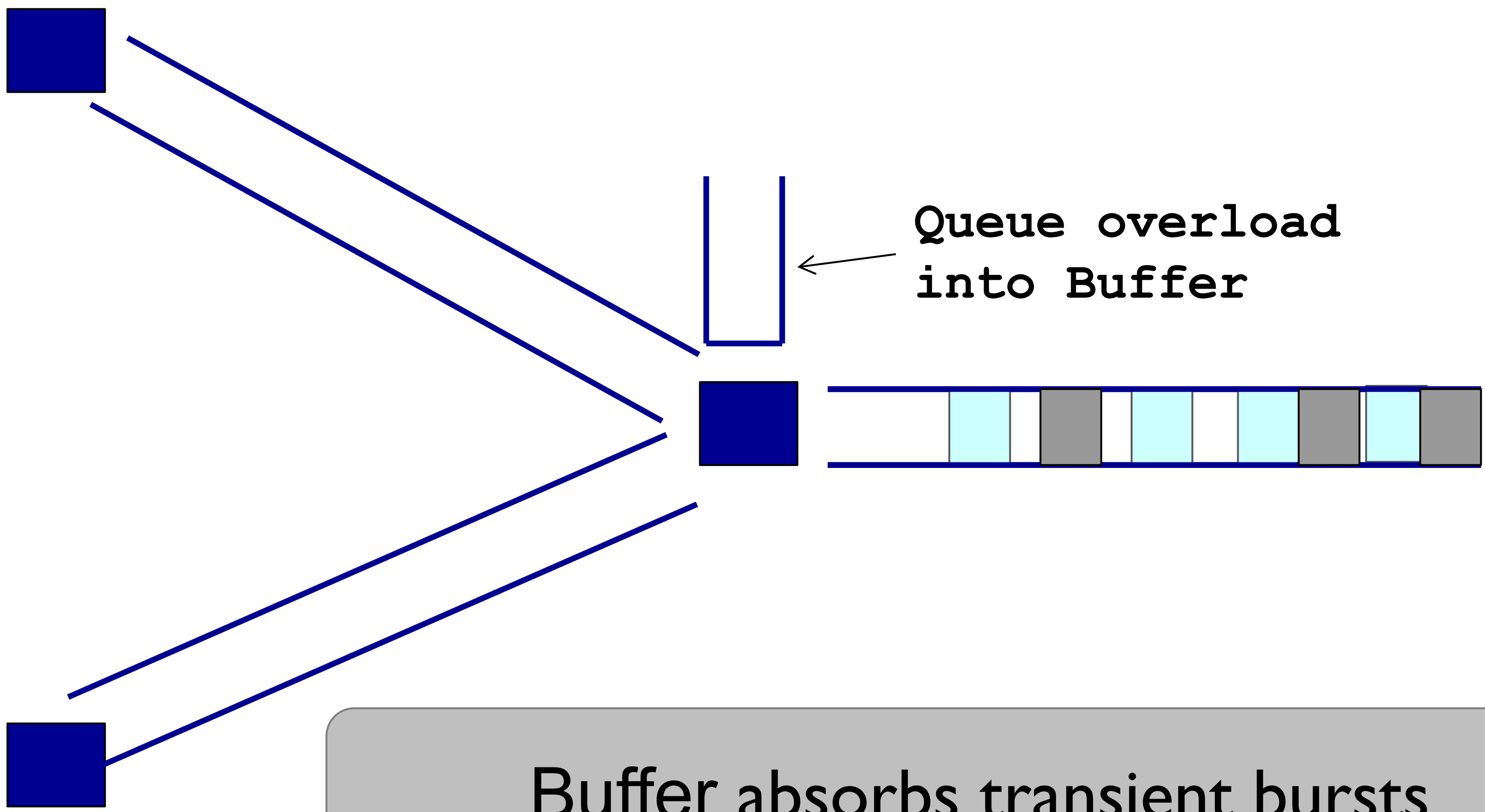
Statistical multiplexing: pipe view



Transient Overload

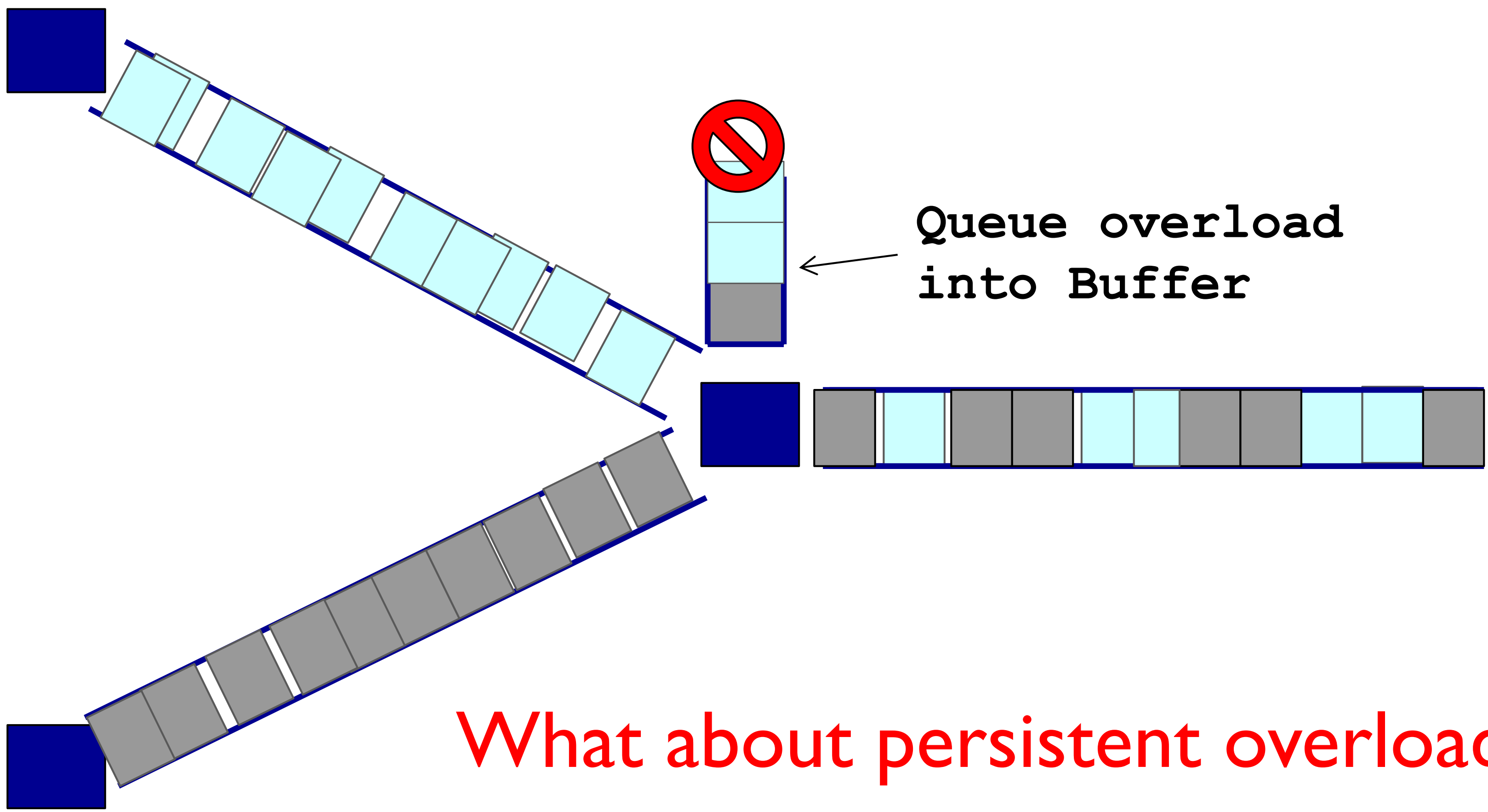
Not such a rare event

Statistical multiplexing: pipe view



Buffer absorbs transient bursts

Statistical multiplexing: pipe view



What about persistent overload?

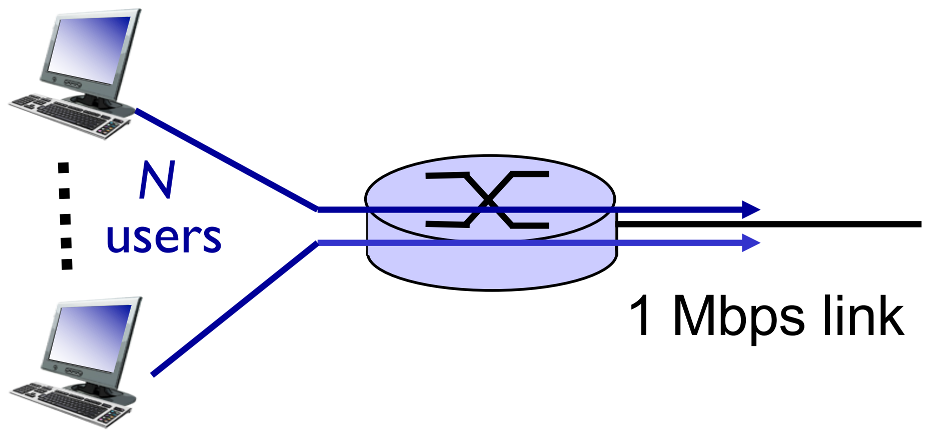
Will eventually drop packets

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 *

Q: how did we get value 0.0004?

Q: what happens if > 35 users say 70?

Hint: Bernoulli Trials and Binomial Distribution

Probability Basics

In general, if the random variable X follows the binomial distribution with parameters $n \in \mathbb{N}$ and $p \in [0,1]$, we write $X \sim B(n, p)$. The probability of getting exactly k successes in n trials is given by the probability mass function:

$$f(k, n, p) = \Pr(k; n, p) = \Pr(X = k) = \binom{n}{k} p^k (1-p)^{n-k}$$

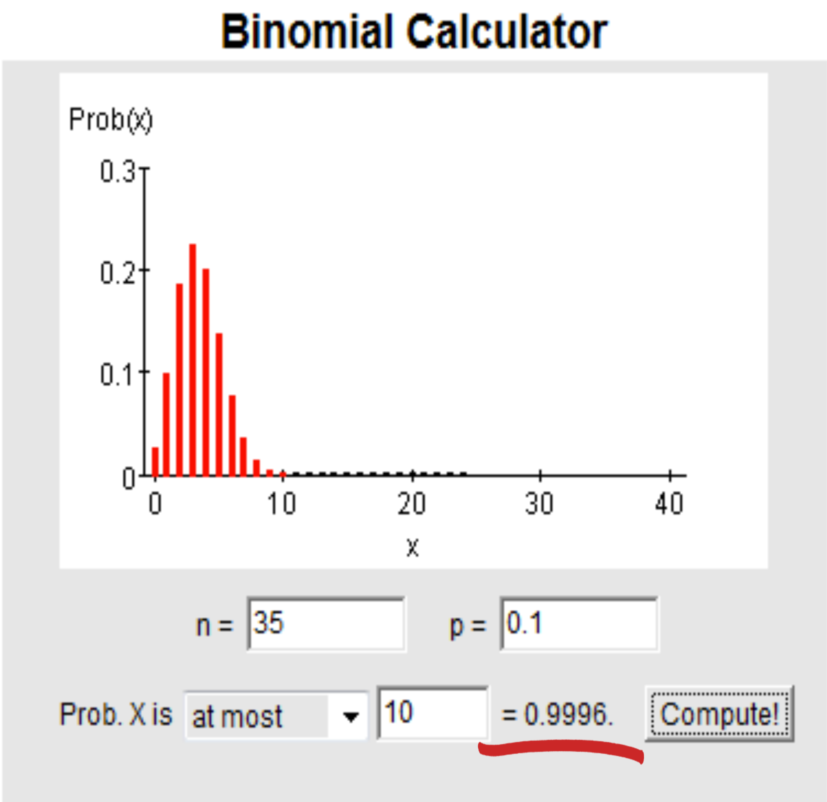
for $k = 0, 1, 2, ..., n$, where

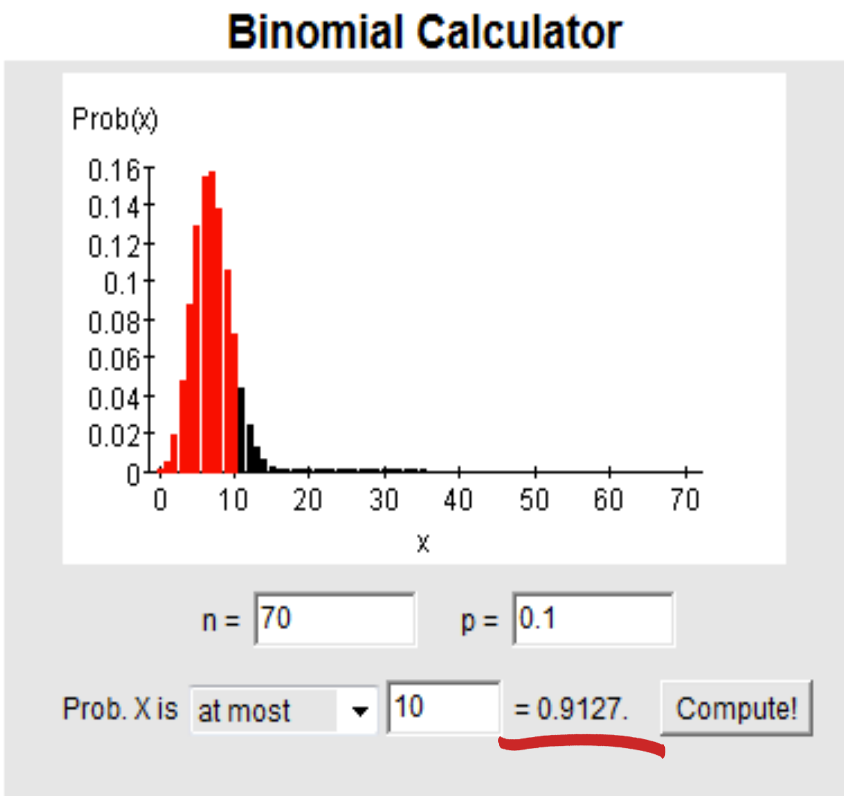
$$\binom{n}{k} = \frac{n!}{k!(n-k)!}$$

The cumulative distribution function can be expressed as:

$$F(k; n, p) = \Pr(X \le k) = \sum_{i=0}^{\lfloor k \rfloor} \binom{n}{i} p^i (1-p)^{n-i}$$

Statistical Multiplexing Gain (SMG)





SMG: 35/10=3.5

SMG: 70/10=7

Packet switching versus circuit switching

is packet switching a "slam dunk winner?"

- great for bursty data
  - resource sharing
  - simpler, no call setup
- excessive congestion 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

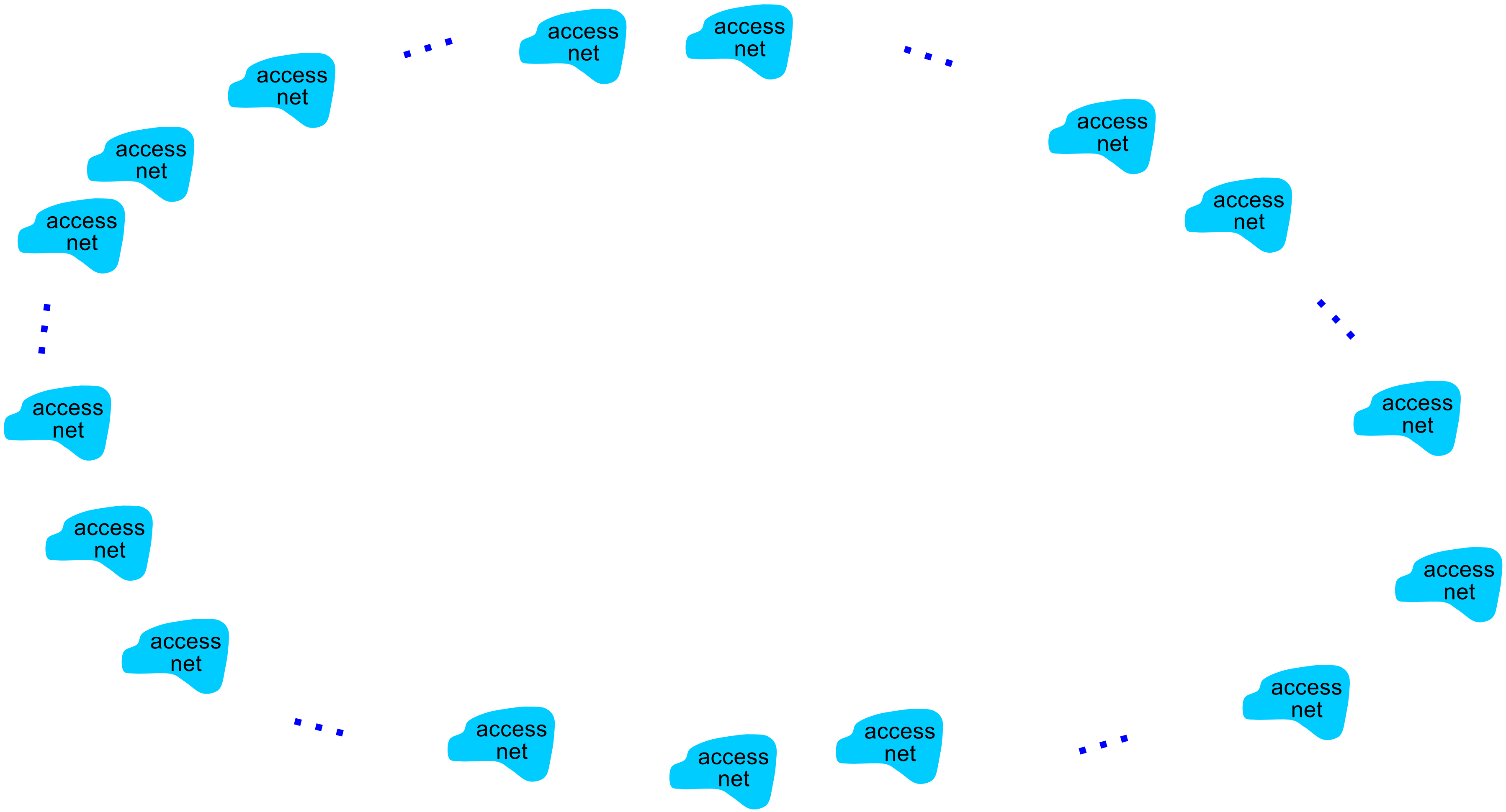
Q: human analogies of reserved resources (circuit switching) versus on-demand allocation (packet-switching)?

Internet structure: network of networks

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

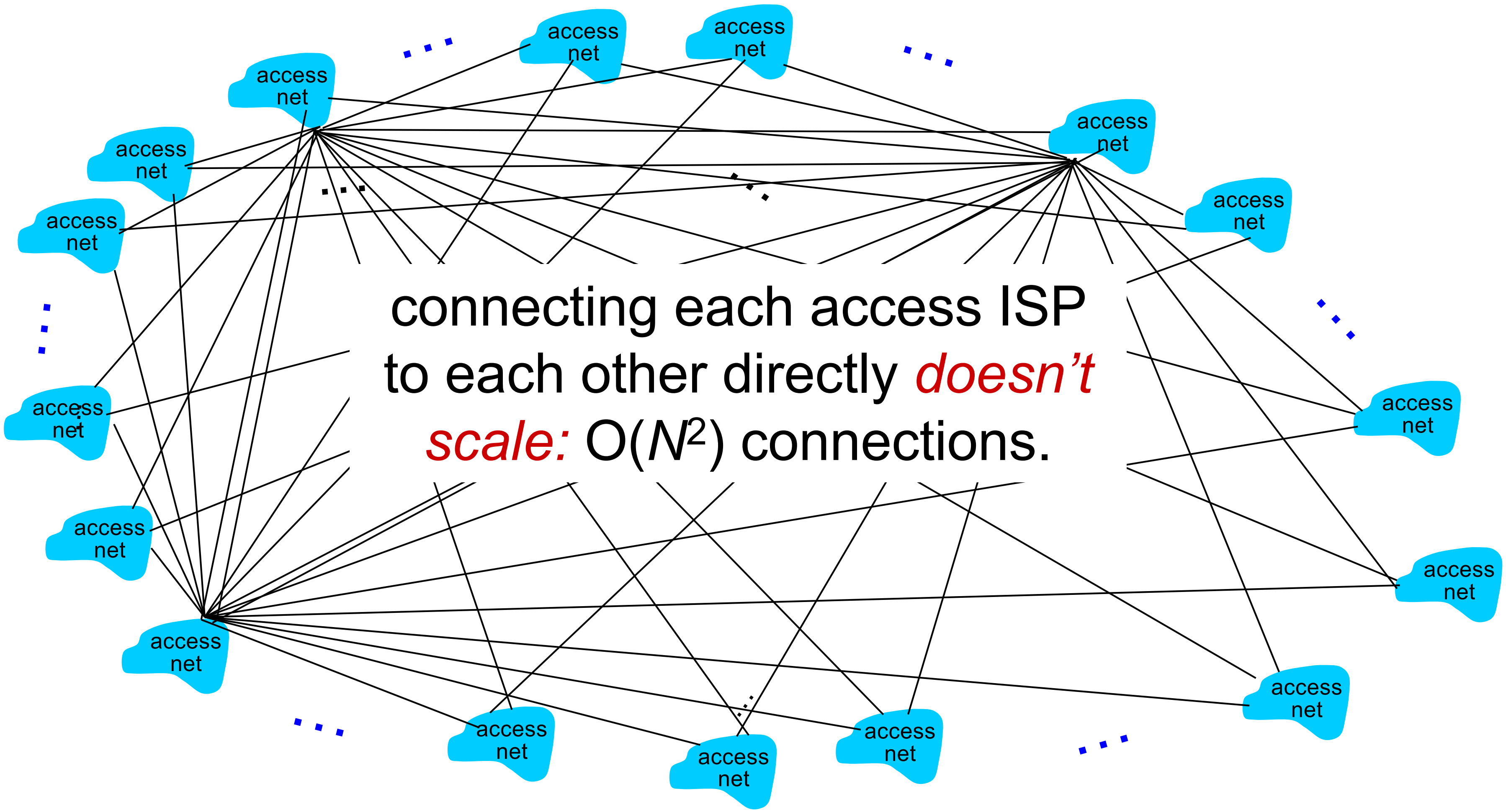
Internet structure: network of networks

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



Internet structure: network of networks

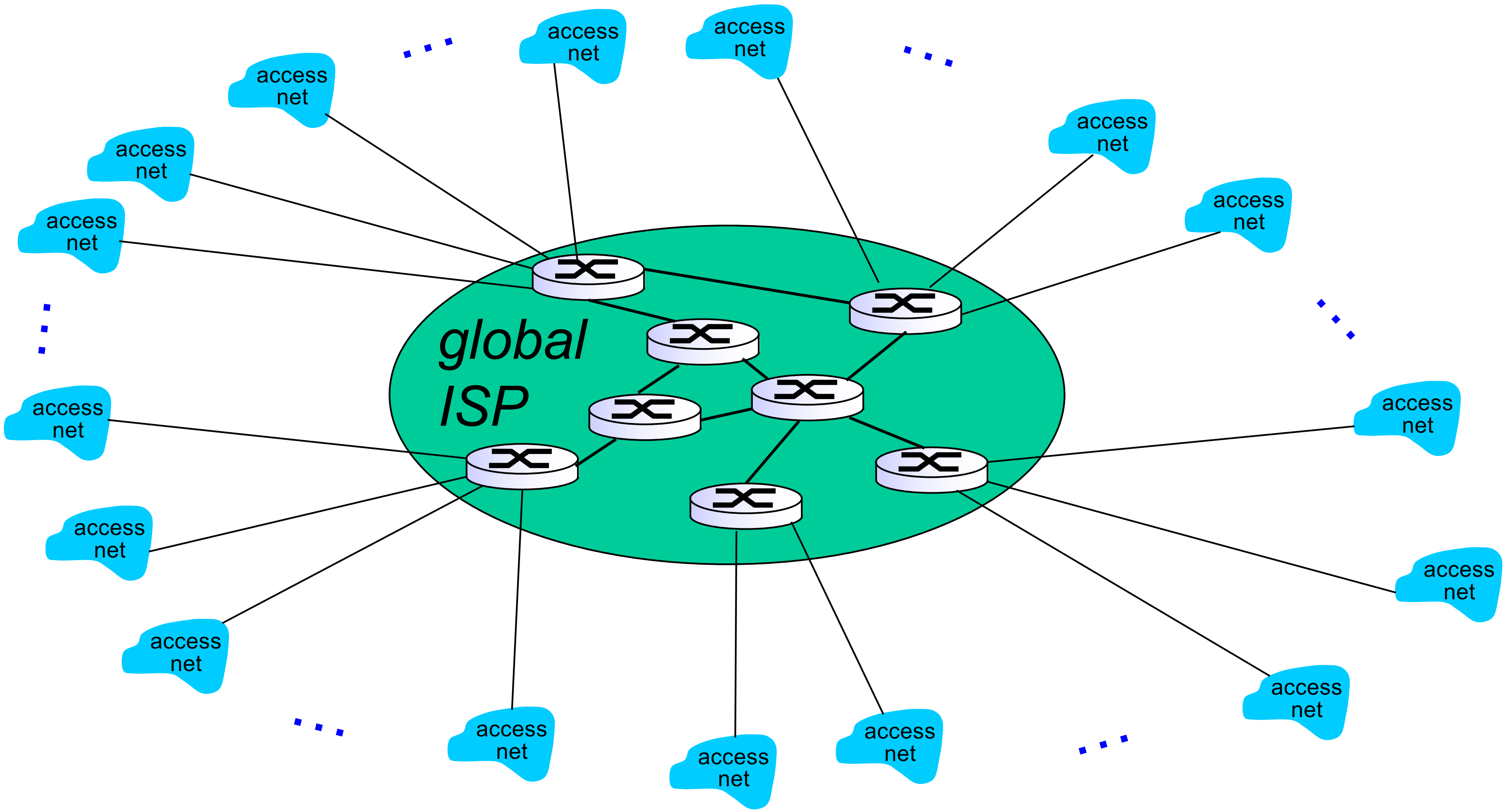
Option: connect each access ISP to every other access ISP?



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

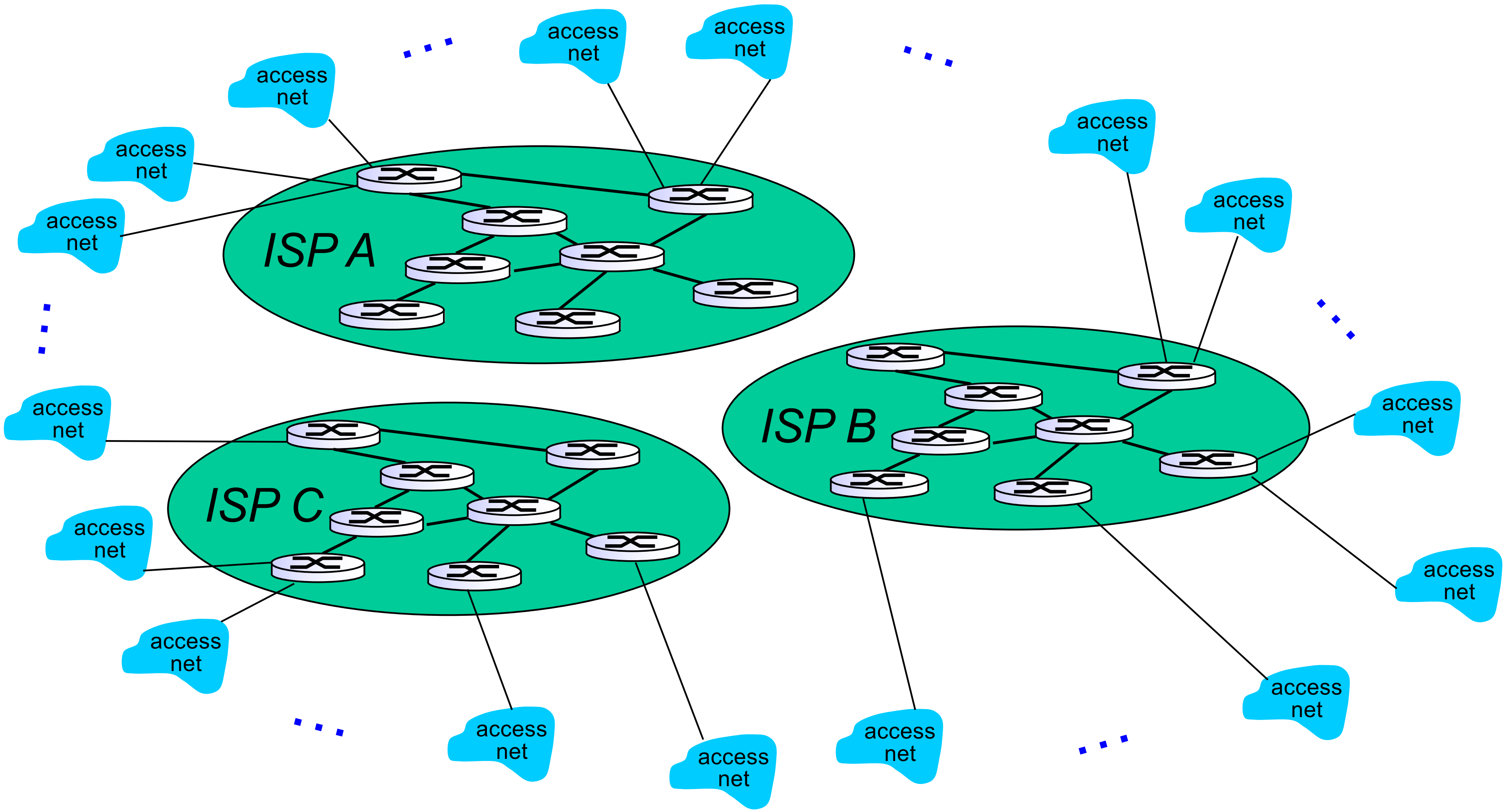
Internet structure: network of networks

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



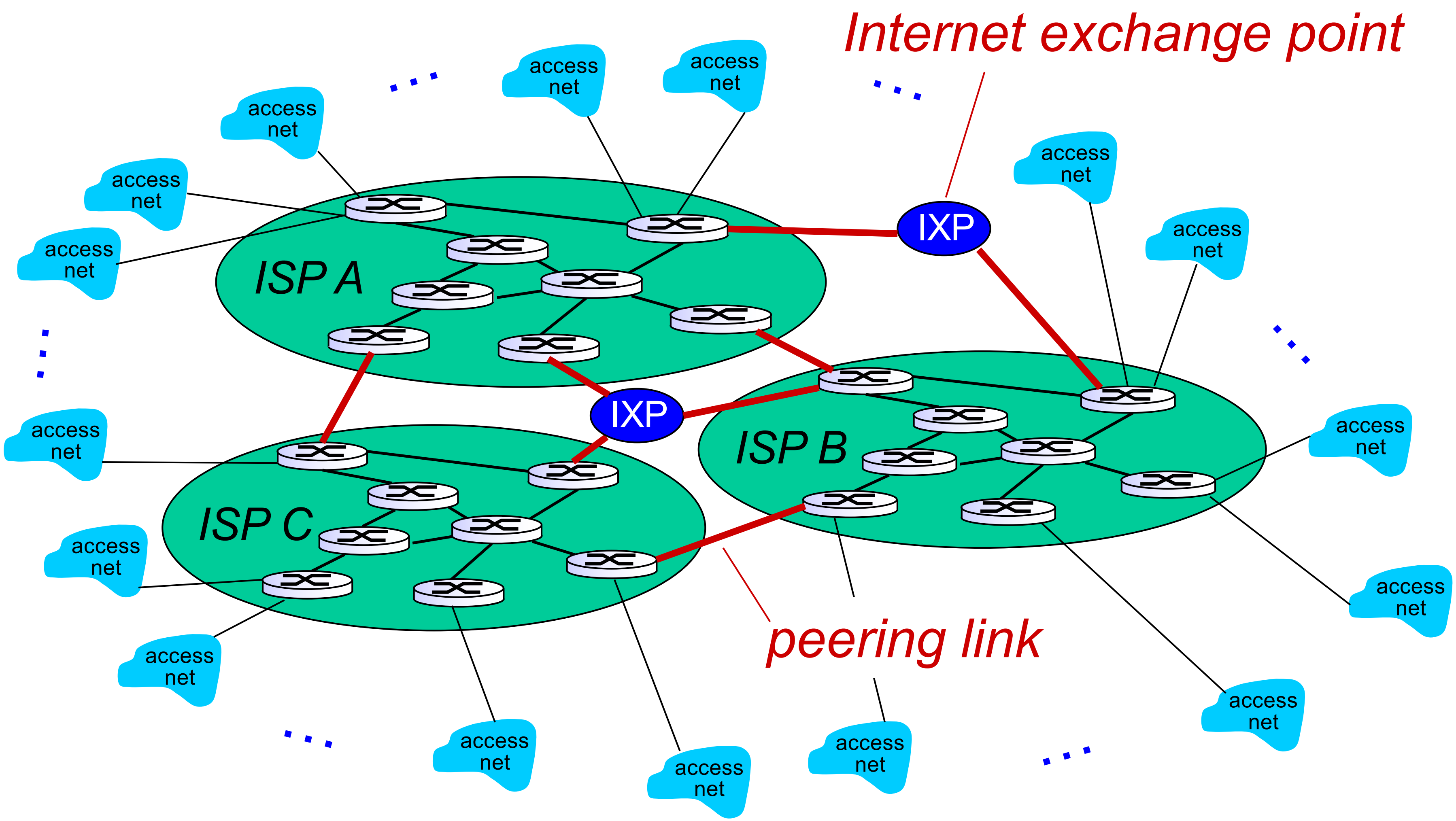
Internet structure: network of networks

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



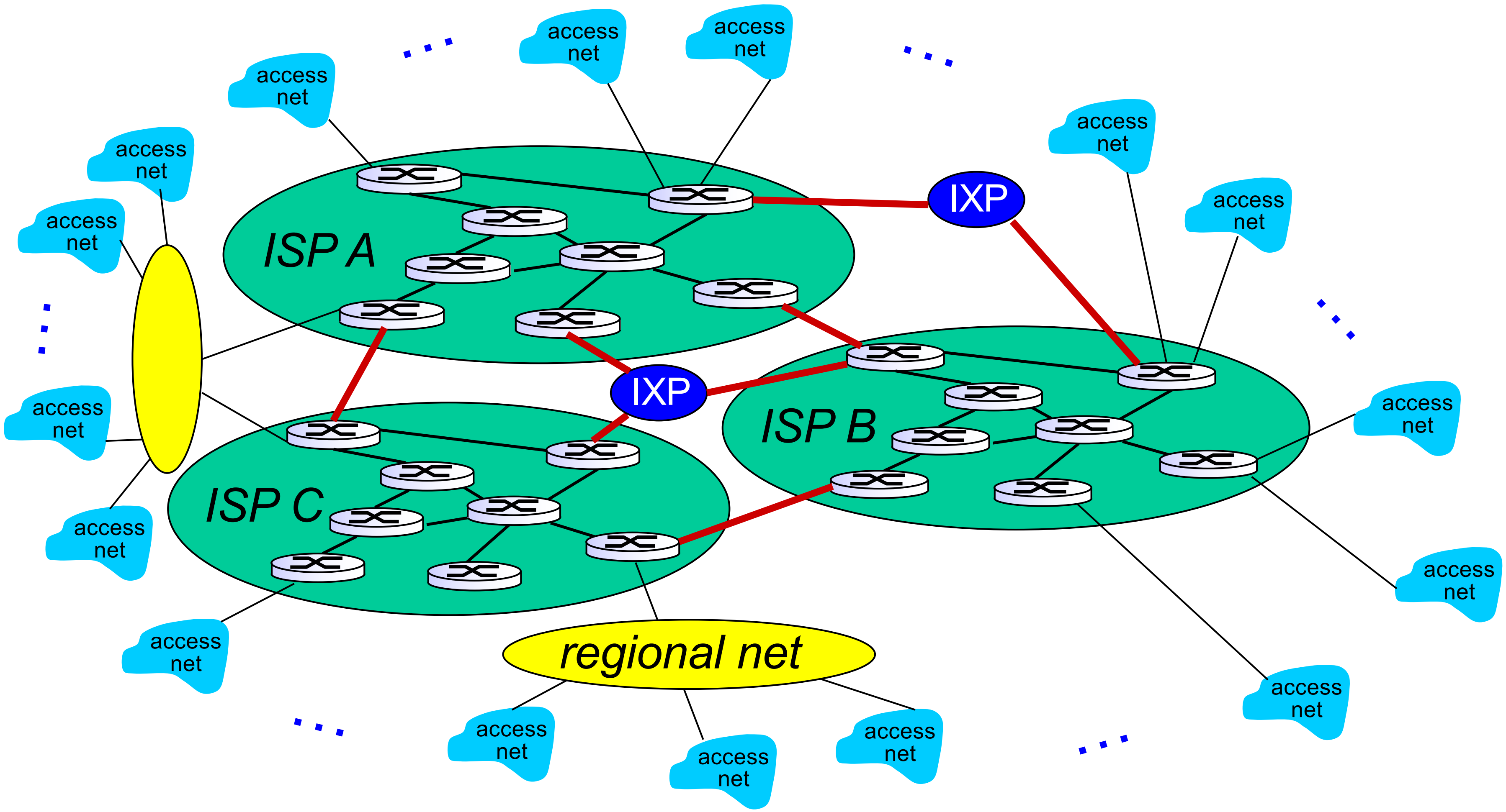
Internet structure: network of networks

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



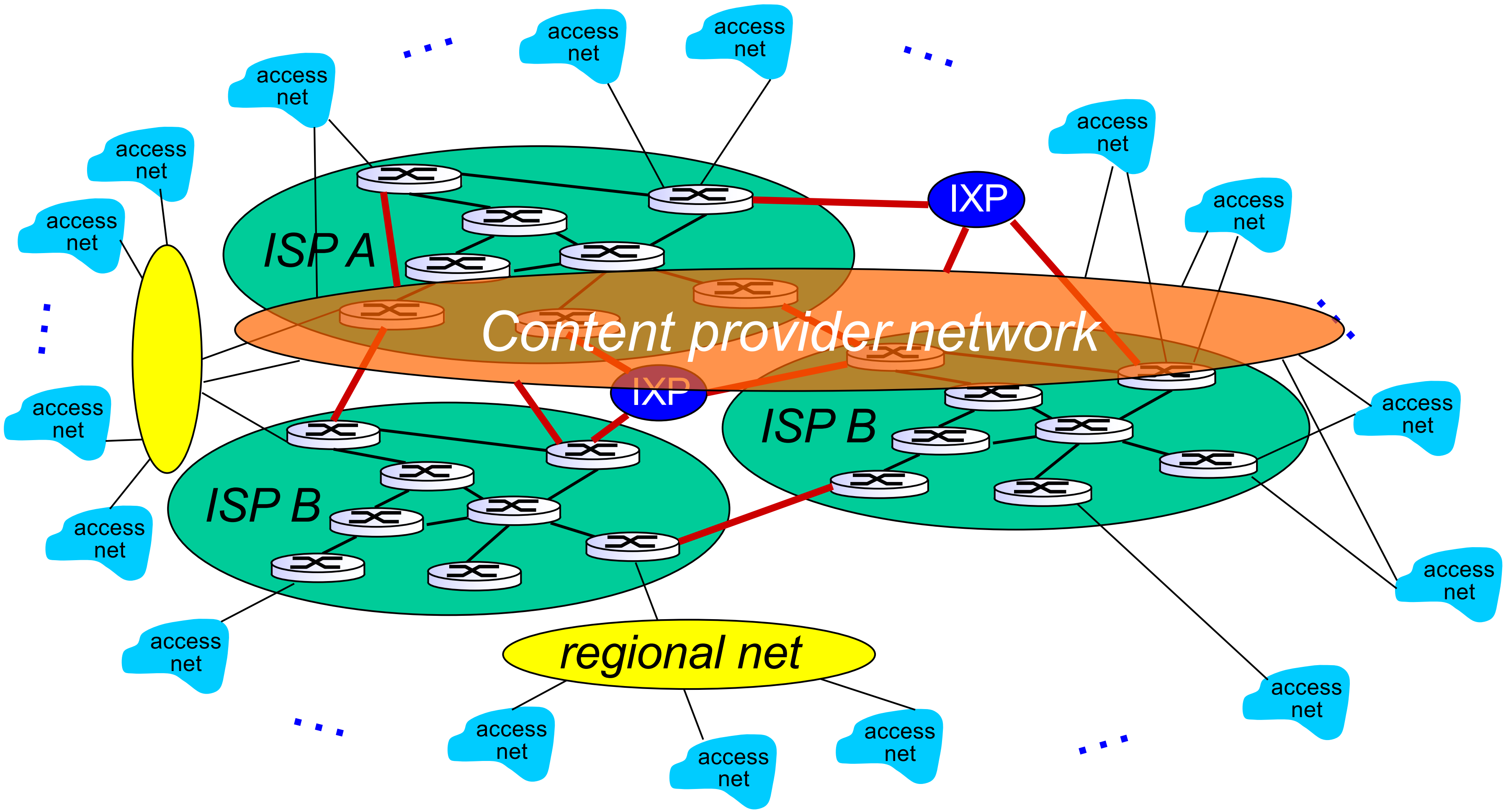
Internet structure: network of networks

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

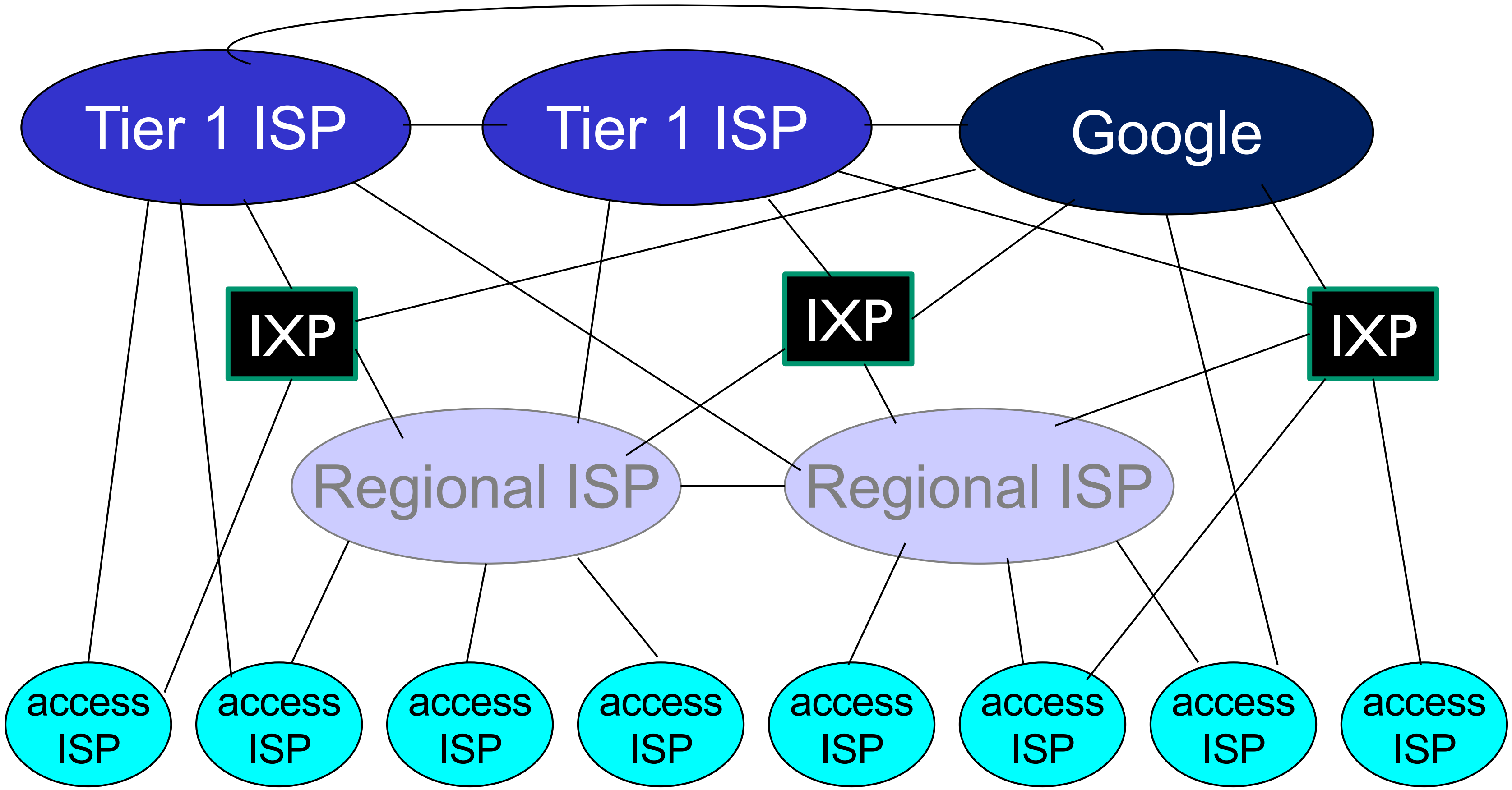


Internet structure: network of networks

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



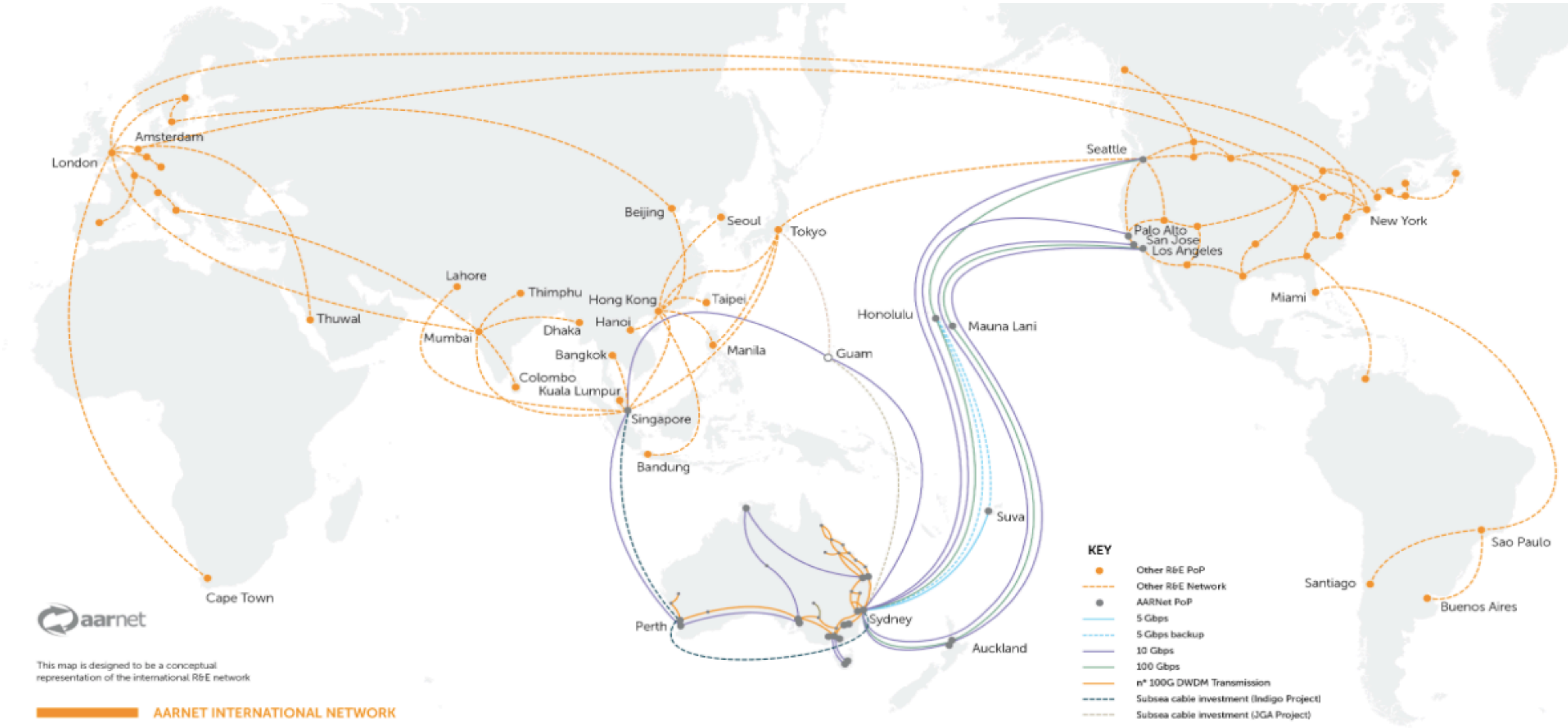
Internet structure: network of networks



- at center: small # of well-connected large networks
  - "tier-1" commercial ISPs (e.g., Level 3, Sprint, AT&T, NTT, Orange, Deutsche Telekom), national & international coverage
  - content provider network (e.g., Google): private network that connects it data centers to Internet, often bypassing tier-1, regional ISPs

AARNET: Australia's Academic and Research Network

- https://www.aarnet.edu.au/
- https://www.submarinecablemap.com



1. Introduction: 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

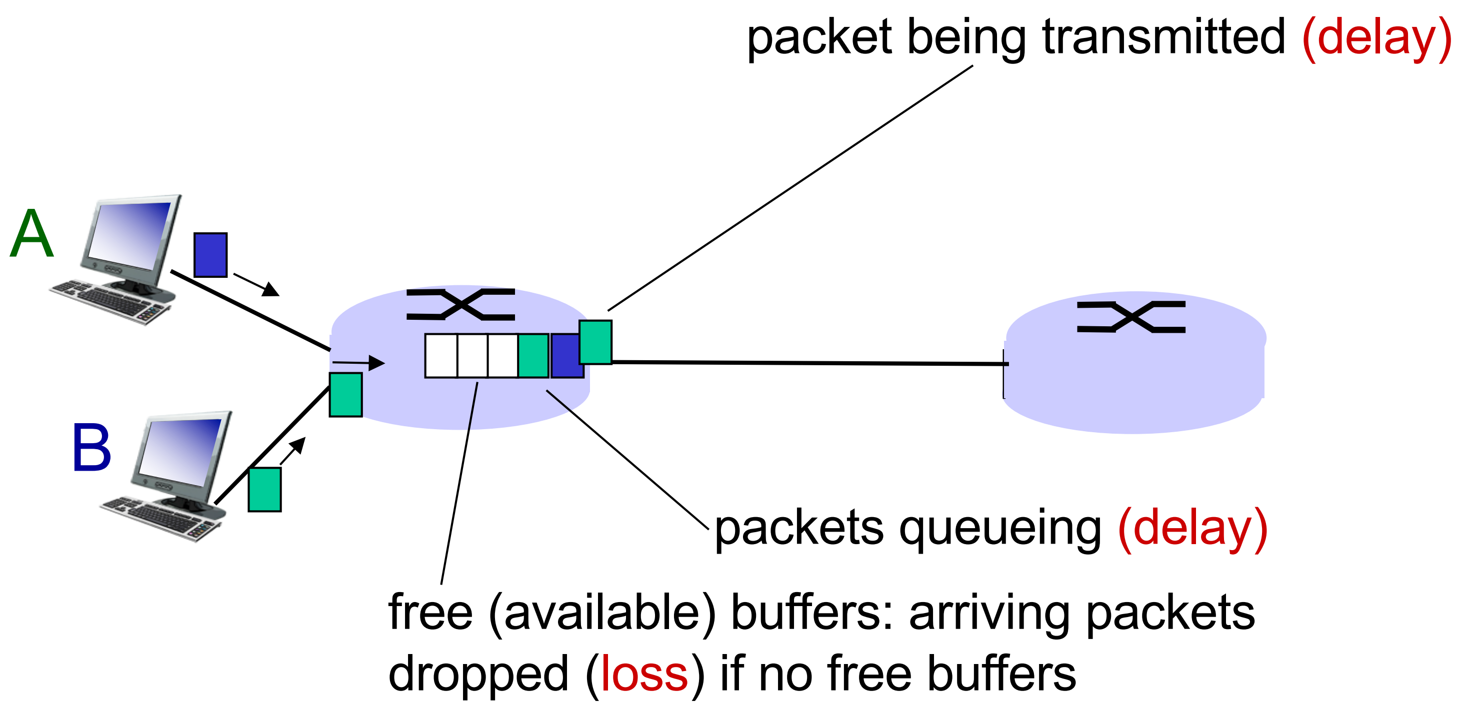
1.6 networks under attack: security

1.7 history

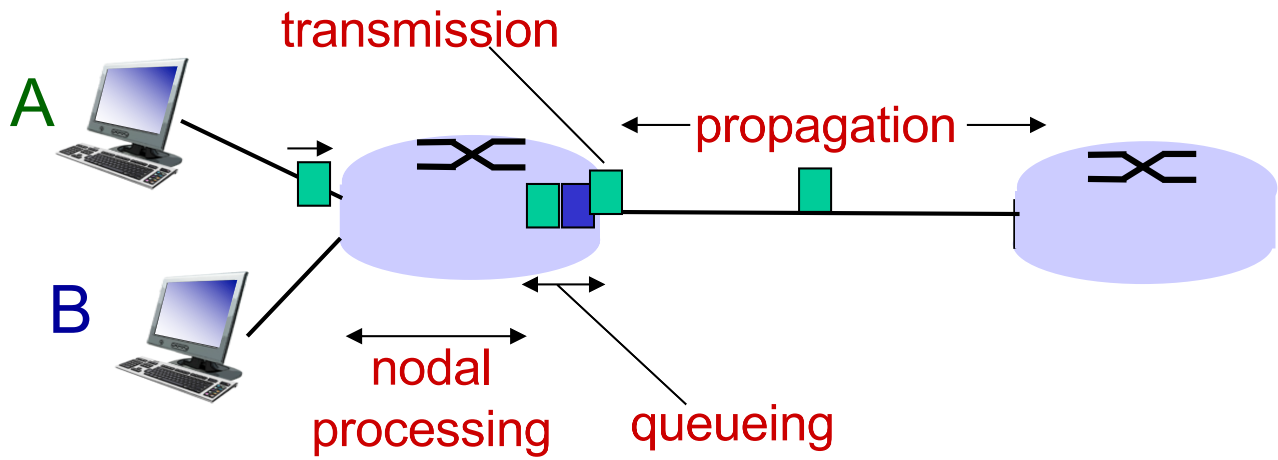
How do loss and delay occur?

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



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

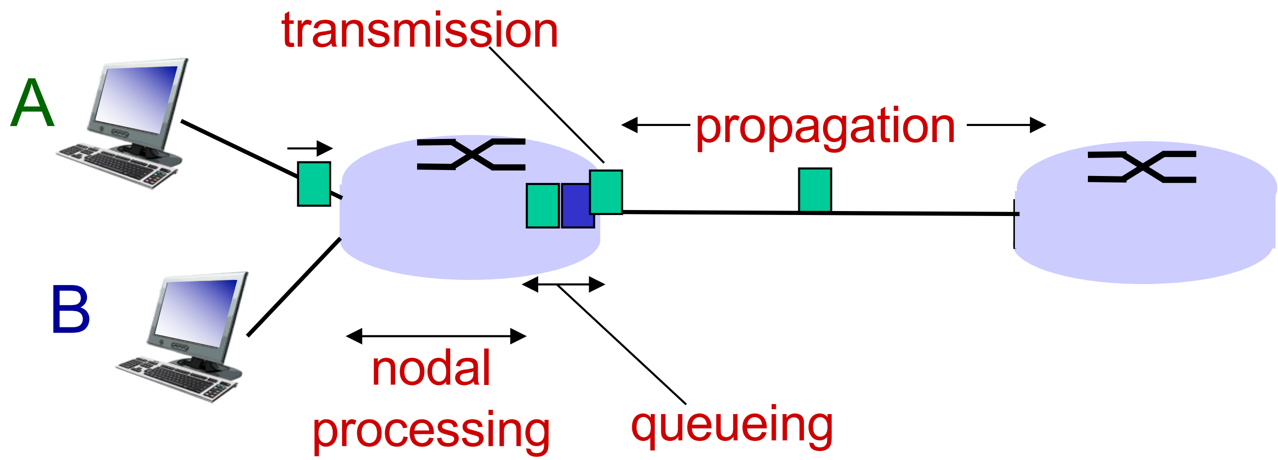
d_{proc}: nodal processing

- 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



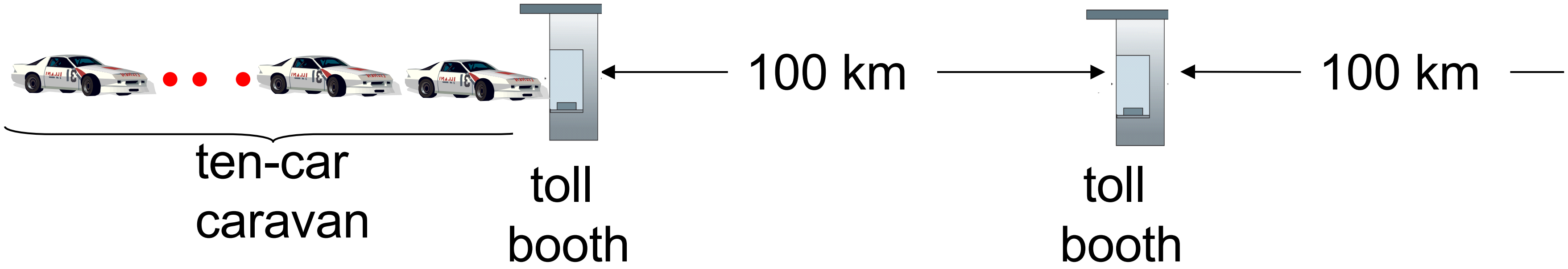
$$d_{nodal} = d_{proc} + d_{queue} + d_{trans} + d_{prop}$$

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 in medium (~2×10^8 m/sec)
- $d_{prop} = d/s$

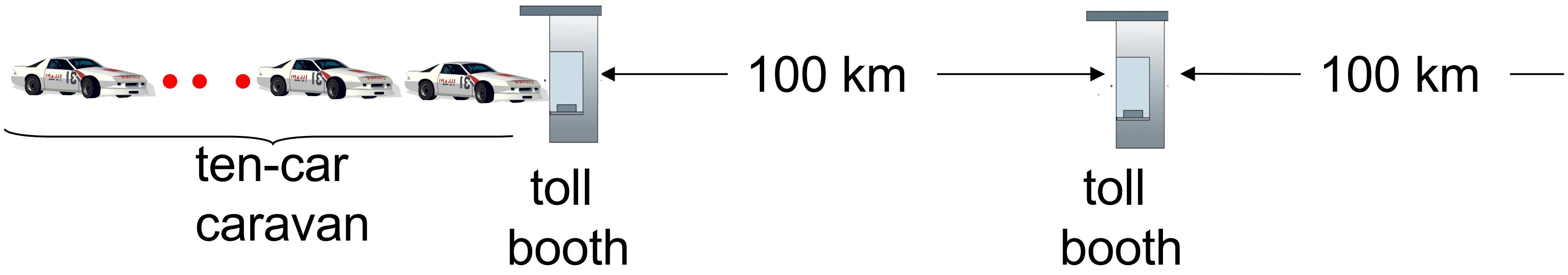
Caravan analogy



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

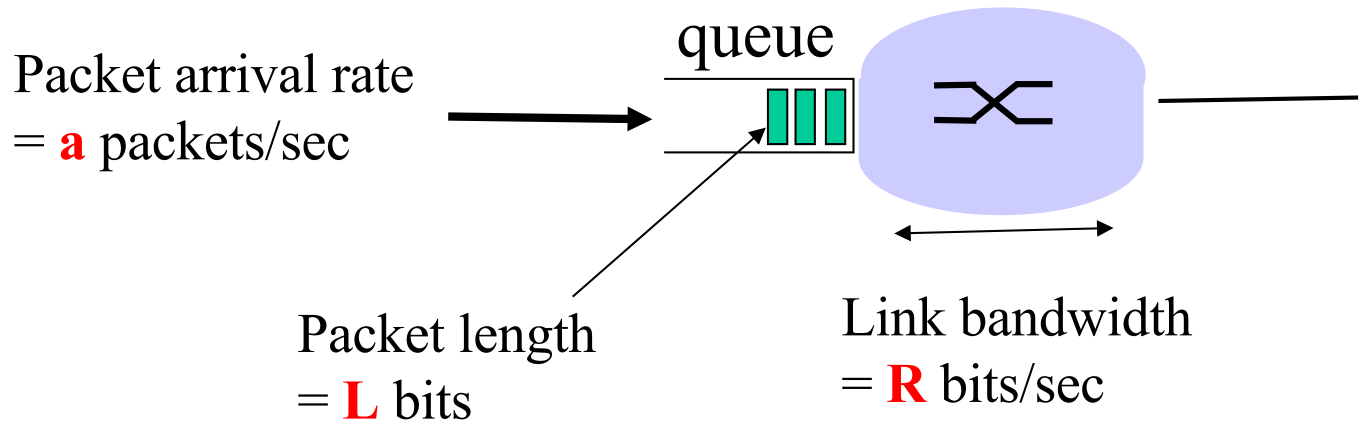
- time to "push" entire caravan through toll booth onto highway = 12*10 = 120 sec
- time for last car to propagate from 1st to 2nd toll both: 100km/(100km/hr)= 1 hr
- *A:* 62 minutes

Caravan analogy (more)



- Suppose cars now "propagate" at 1000 km/hr
- And suppose toll booth 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, 1st car arrives at second booth; three cars still at 1st booth.

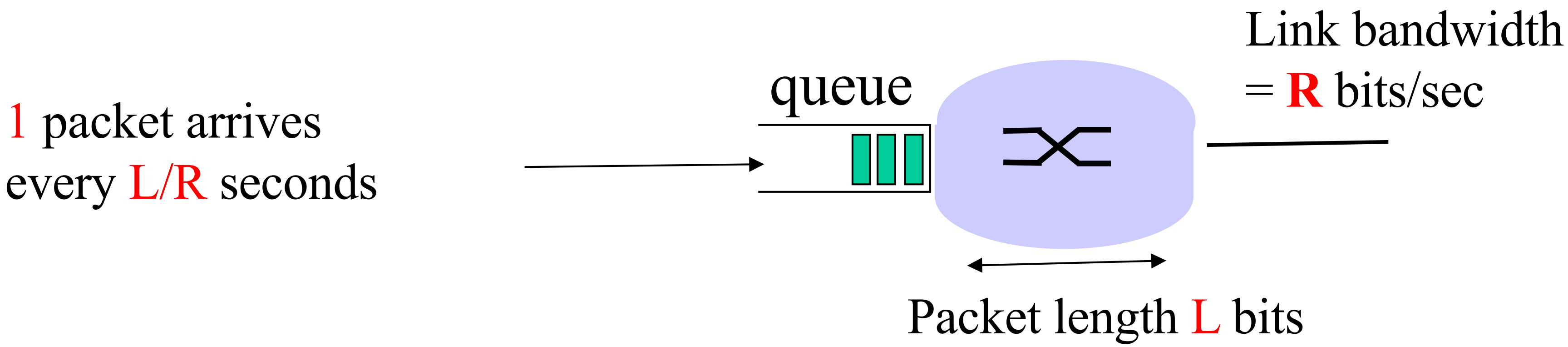
Queueing delay (more insight)



- Every second: aL bits arrive to queue
- Every second: R bits leave the router
- Question: what happens if $aL > R$?
- Answer: queue will fill up, and packets will get dropped!!

aL/R is called traffic intensity

Queueing delay: illustration

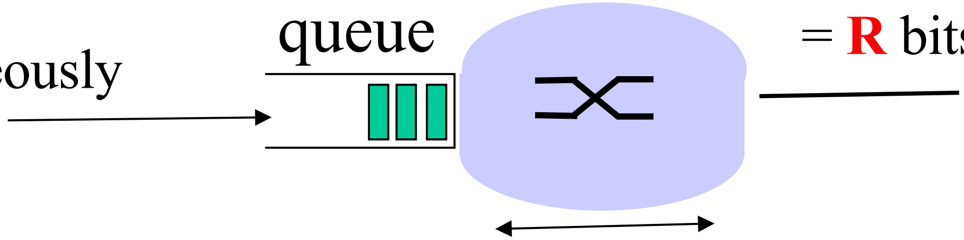


Arrival rate: a = 1/(L/R) = R/L (packet/second)

Traffic intensity = aL/R = (R/L) (L/R) = 1

Average queueing delay = 0
(queue is initially empty)

Queueing delay: illustration



Arrival rate: $a = N/(LN/R) = R/L$ packet/second

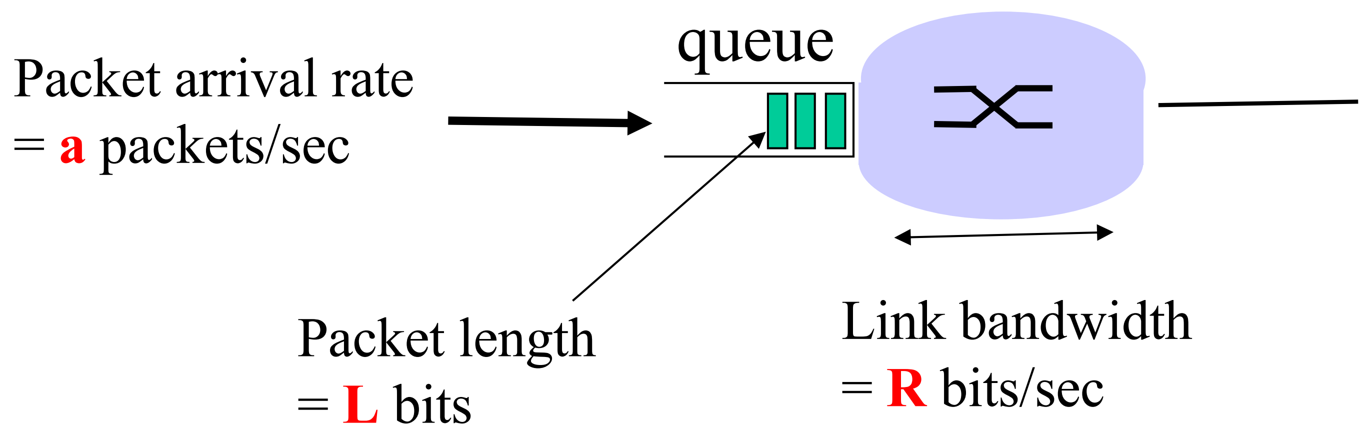
Traffic intensity $= aL/R = (R/L)\,(L/R) = 1$

Average queueing delay (queue is empty at time 0) ?

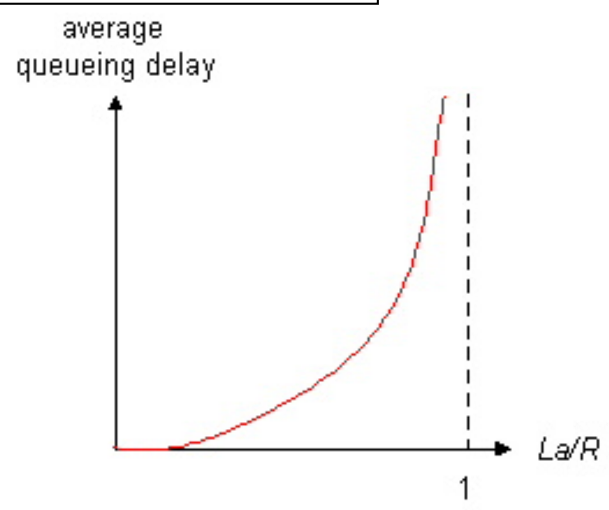
$\{0 + L/R + 2L/R + \ldots + (N-1)L/R\}/N = L/(RN)\{1+2+\ldots+(N-1)\} = L(N-1)/(2R)$

Note: traffic intensity is same as previous scenario, but queueing delay is different

Queueing delay: behaviour



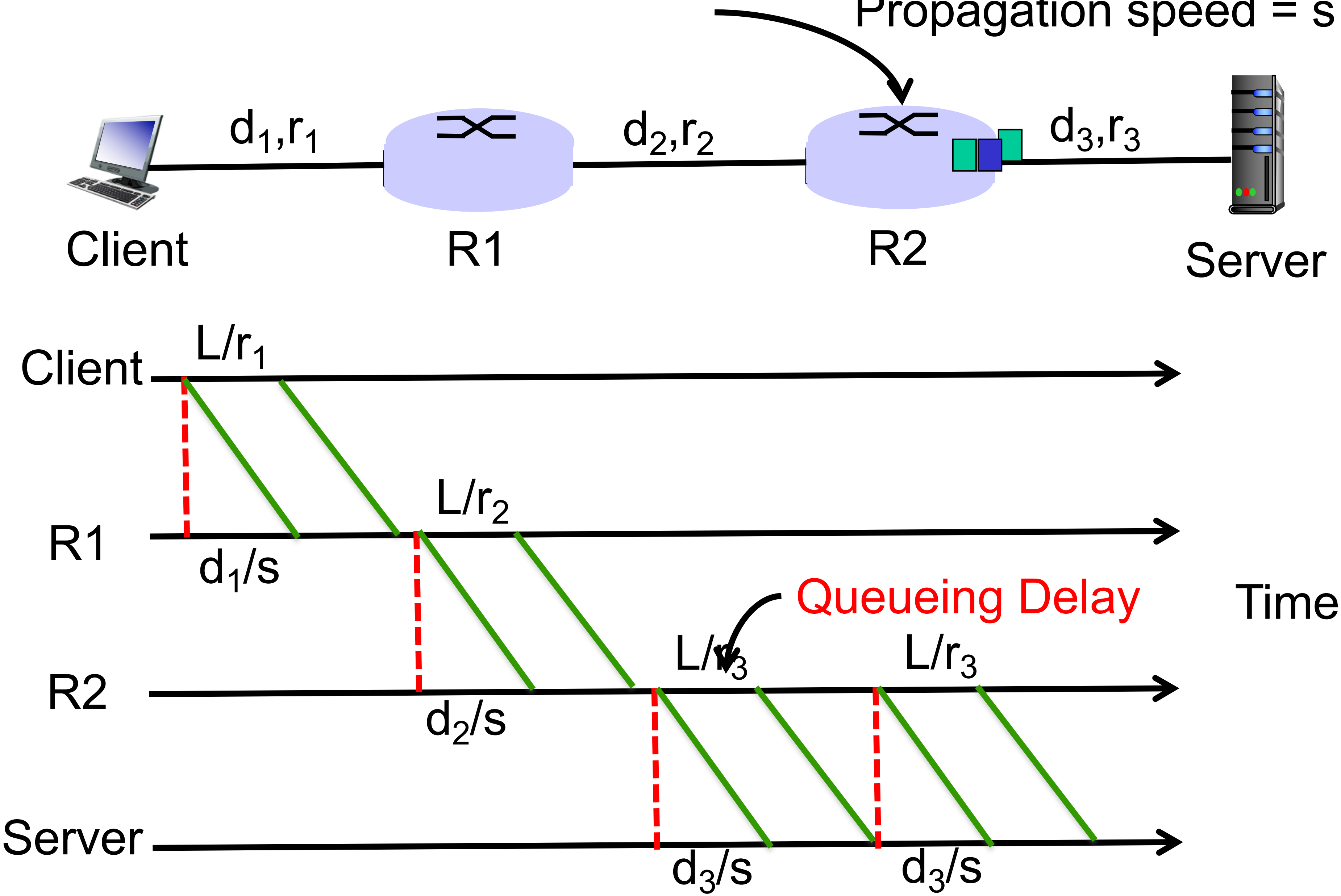
Interactive Java Applet:
http://computerscience.unicam.it/marcantoni/reti/applet/QueuingAndLossInteractive/1.html



- La/R ~ 0: avg. queueing delay small
- La/R -> 1: delays become large
- La/R > 1: more "work" than can be serviced, average delay infinite! (this is when a is random!)

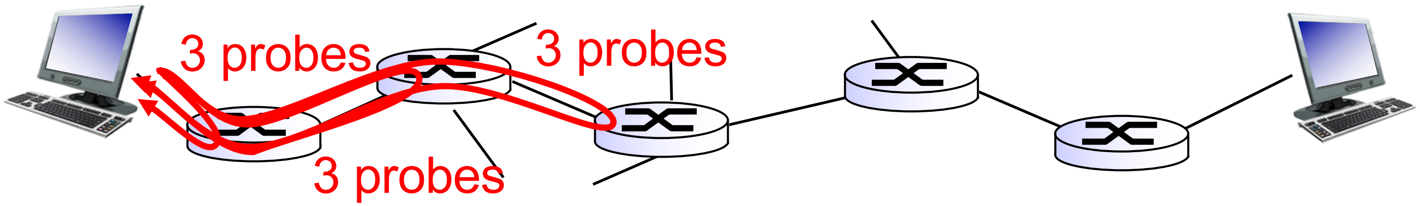
End to End Delay

Packet length = L
Propagation speed = s



"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

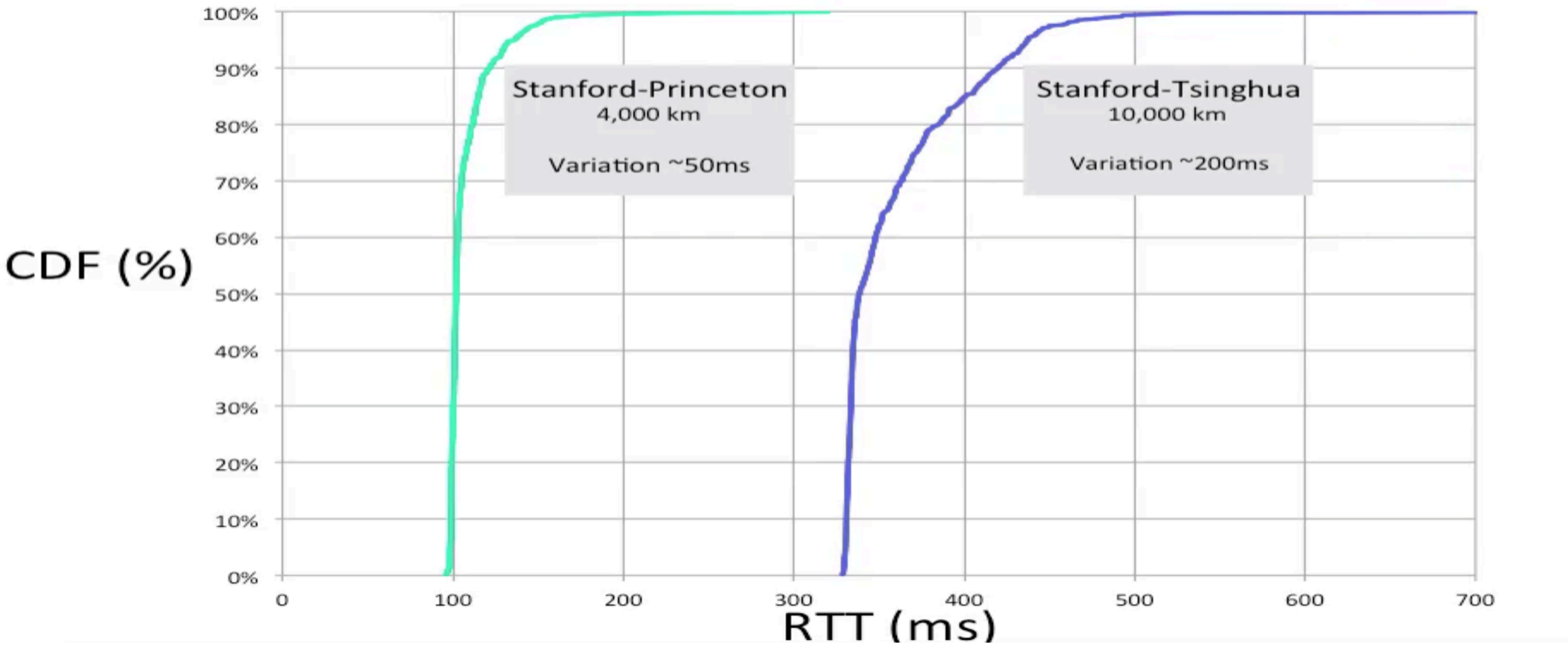
* means no response (probe lost, router not replying)

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

"Real" delay variations

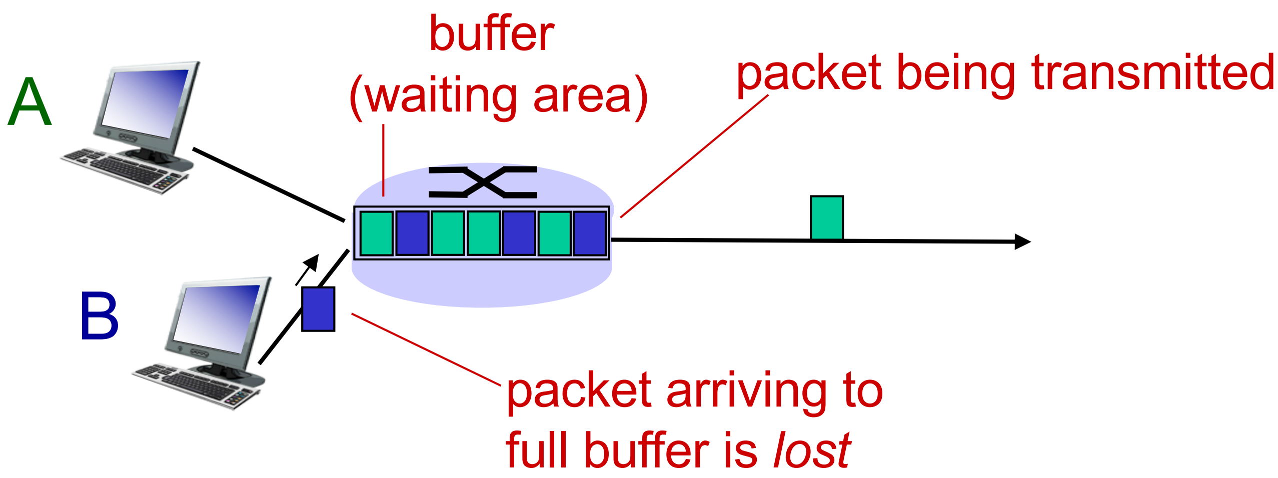
$$d_{nodal} = d_{proc} + d_{queue} + d_{trans} + d_{prop}$$

End-to-end delay = sum of all d_{nodal} along the path



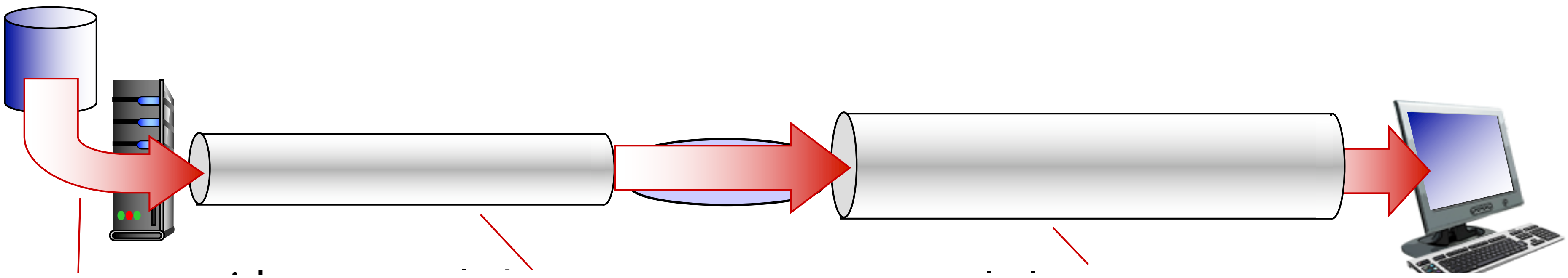
Packet loss

- queue (aka buffer) preceding link in buffer has finite capacity
- packet arriving to full queue dropped (aka lost)
- lost packet may be retransmitted



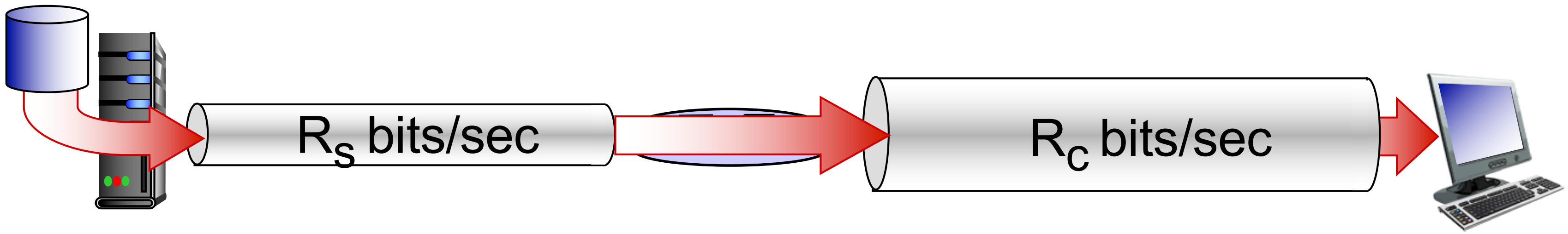
Throughput

- *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

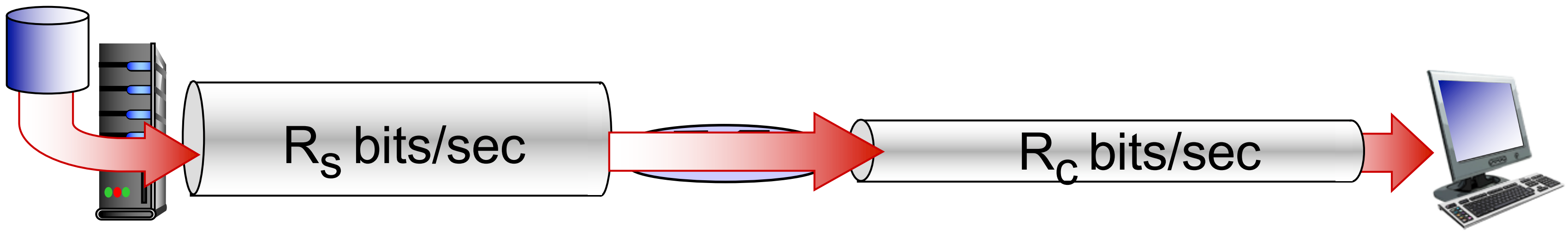


Throughput (more)

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



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

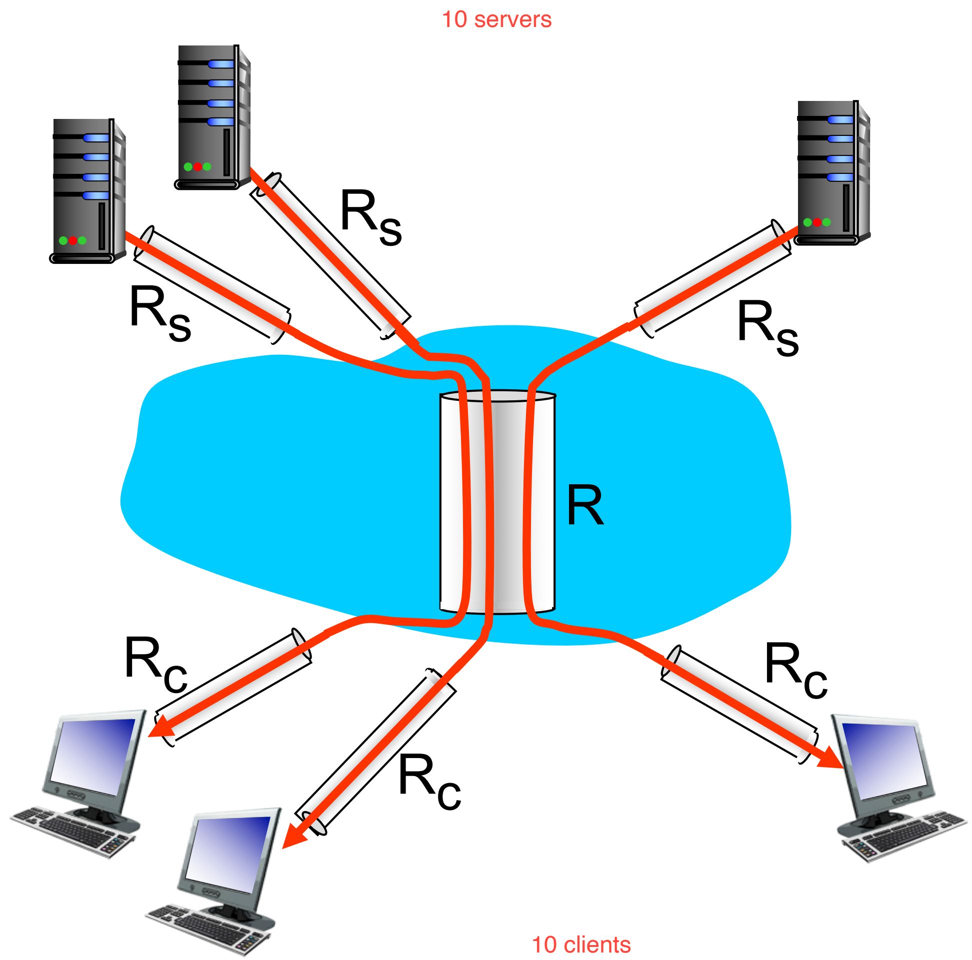


bottleneck link

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

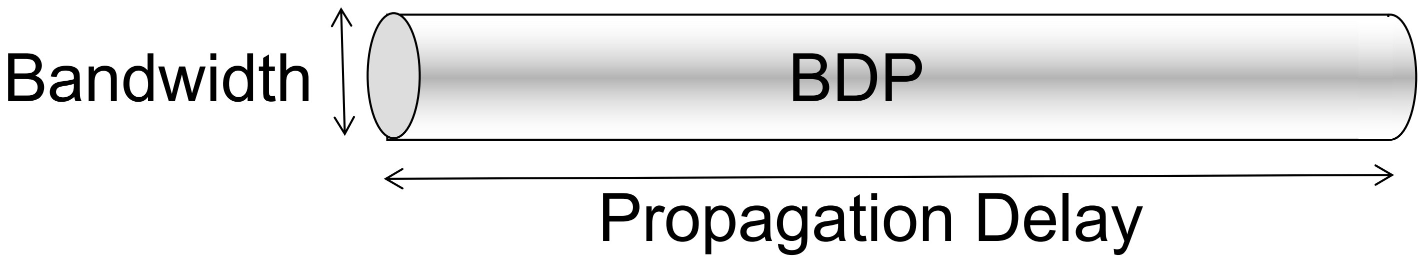
Throughput: Internet scenario

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



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

Key Properties of Links



- Bandwidth: "width" of the link
  - Number of bits sent per unit of time (bps)

- Propagation delay: "length" of the link
  - Propagation time to travel along the link (seconds)

- Bandwidth-Delay Product: "Volume" of the link
  - Amount of data in flight (bps*s=bits)

1. Introduction: 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

协议层次和服务模型

1.5 protocol layers, service models

Self study

1.6 networks under attack: security

1.7 history

Three (networking) design steps

- Break down the problem into tasks
- Organize these tasks
- Decide who does what

Tasks in Networking

- What does it take to send packets across?

- Prepare data (Application)
- Ensure that packets reach the dest process. (Transport)
- Deliver packets across *global* network (Network)
- Deliver packets to next hop within *local* network (Datalink)
- Put bits / packets on wire or trans. medium (Physical)

This is decomposition…

Now, how do we organize these tasks?

Let us have an example

Inspiration…

- CEO A writes letter to CEO B
  - 行政助手
  - Folds letter and hands it to administrative aide

```
Dear John,

Your days are numbered.

            --Pat
```

» Aide:

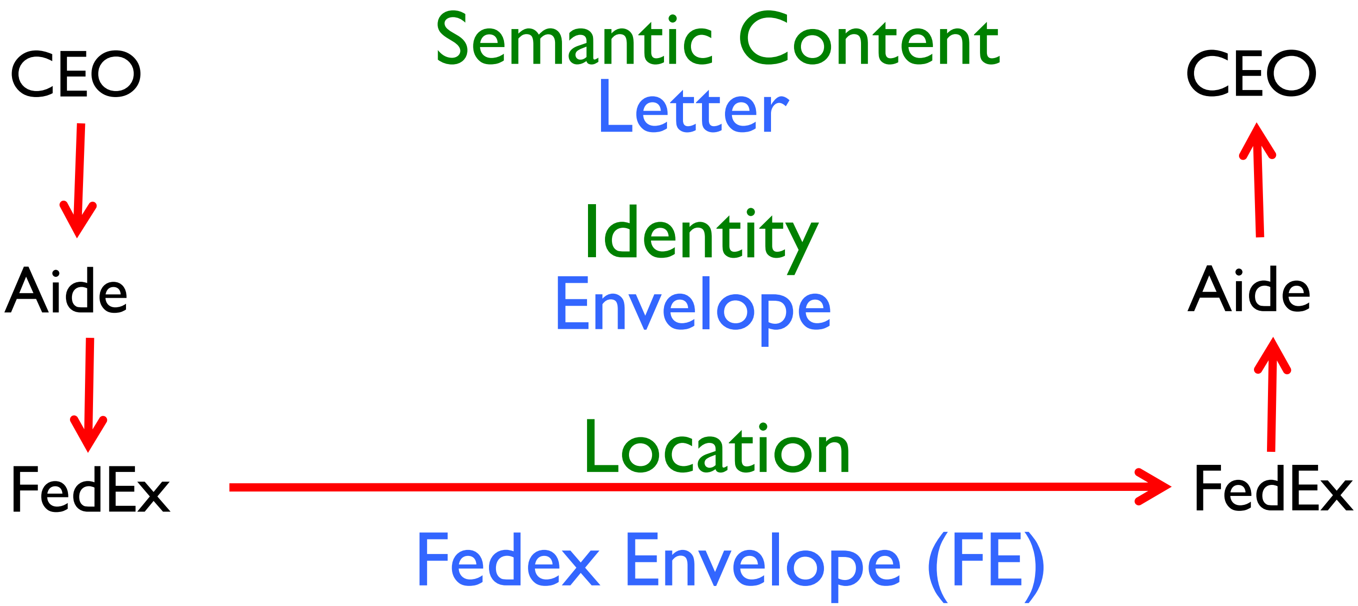
» Puts letter in envelope with CEO B's full name

» Takes to FedEx

- FedEx Office
  - Puts letter in larger envelope
  - Puts name and street address on FedEx envelope
  - Puts package on FedEx delivery truck
- FedEx delivers to other company

The Path of the Letter

"Peers" on each side understand the same things
No one else needs to (abstraction)
Lowest level has most packaging



The Path Through FedEx

Higher "Stack" at Ends

部分的

Partial "Stack" During Transit

Truck

FE

Sorting Office

Crate 箱

Airport

FE

Sorting Office

Crate

New Crate

Airport

Truck

FE

Sorting Office

Crate

Airport

Deepest Packaging (Envelope+FE+Crate) at the Lowest Level of Transport

In the context of the Internet

Applications

…built on…

Reliable (or unreliable) transport

…built on…

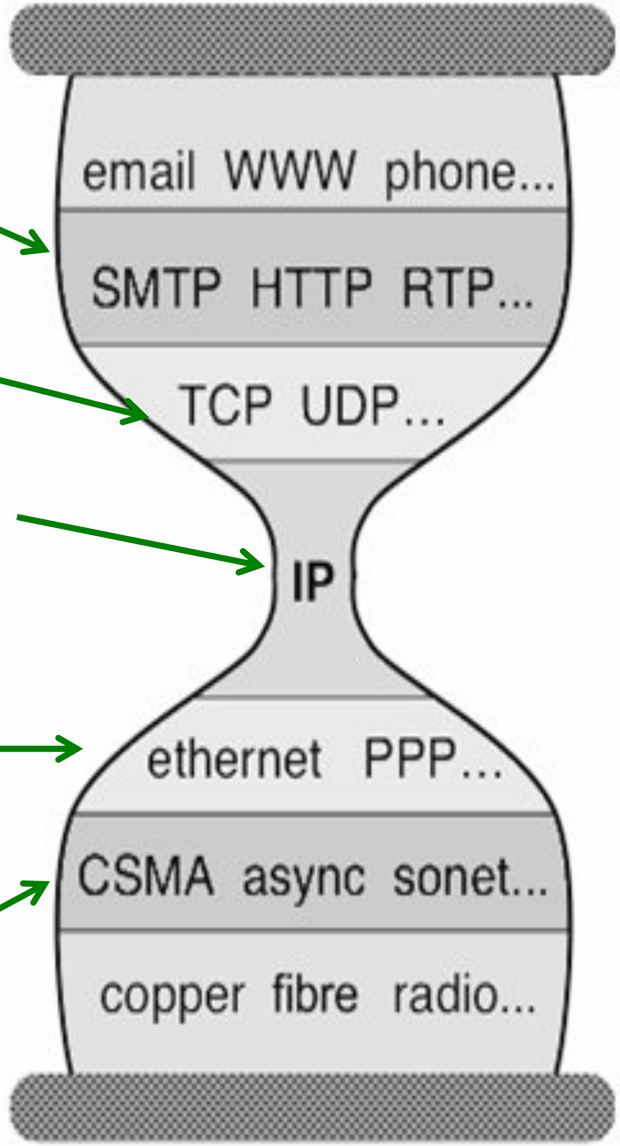
Best-effort global packet delivery

…built on…

Best-effort local packet delivery

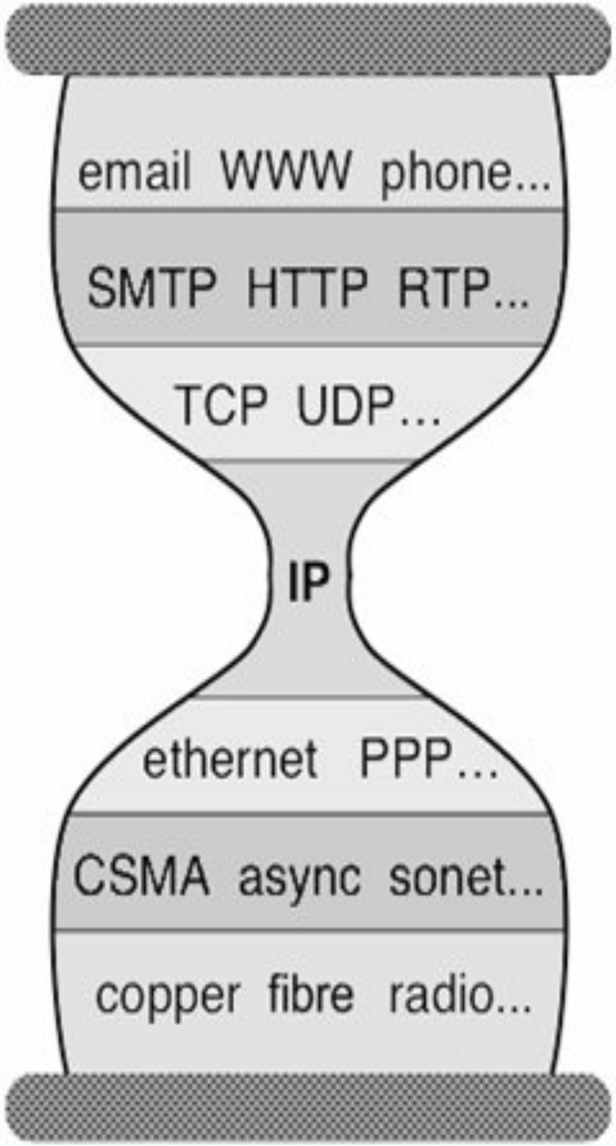
…built on…

Physical transfer of bits



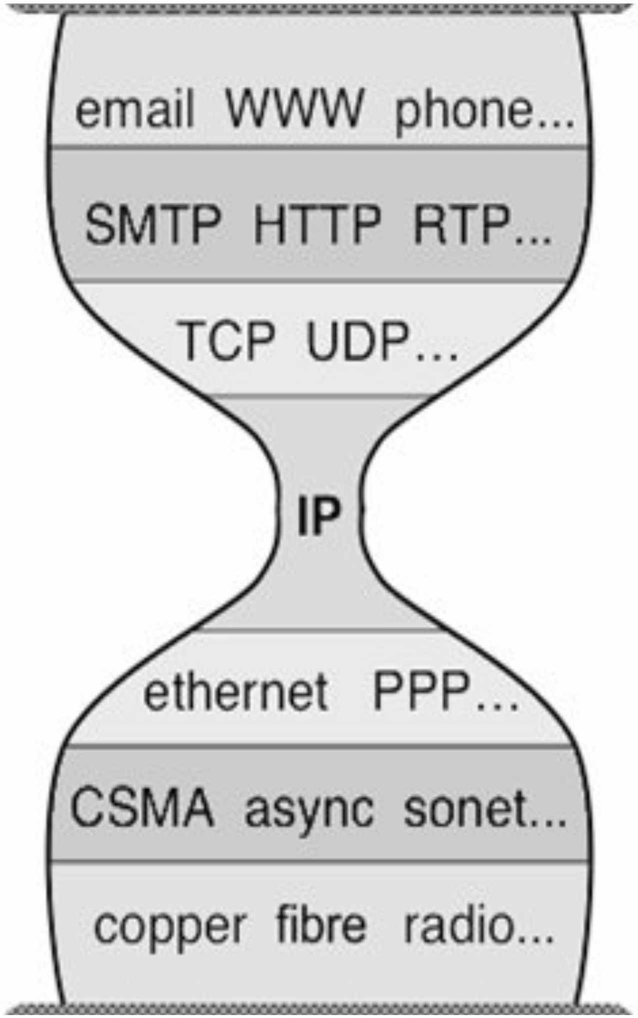
Internet protocol stack

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

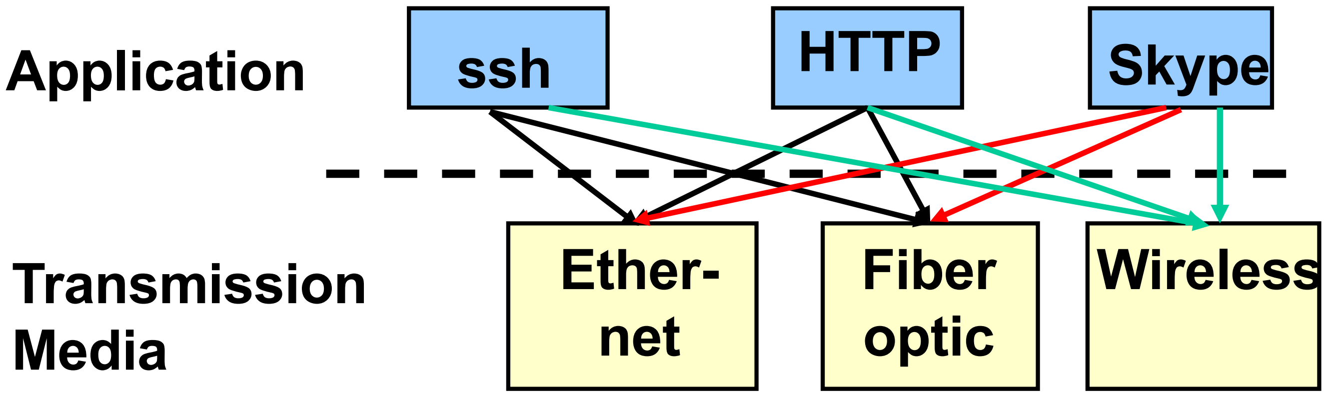


Three Observations

- Each layer:
  - Depends on layer below
  - Supports layer above
  - Independent of others
- Multiple versions in layer
  - Interfaces differ somewhat
  - Components pick which lower-level protocol to use
- But only one IP layer
  - Unifying protocol



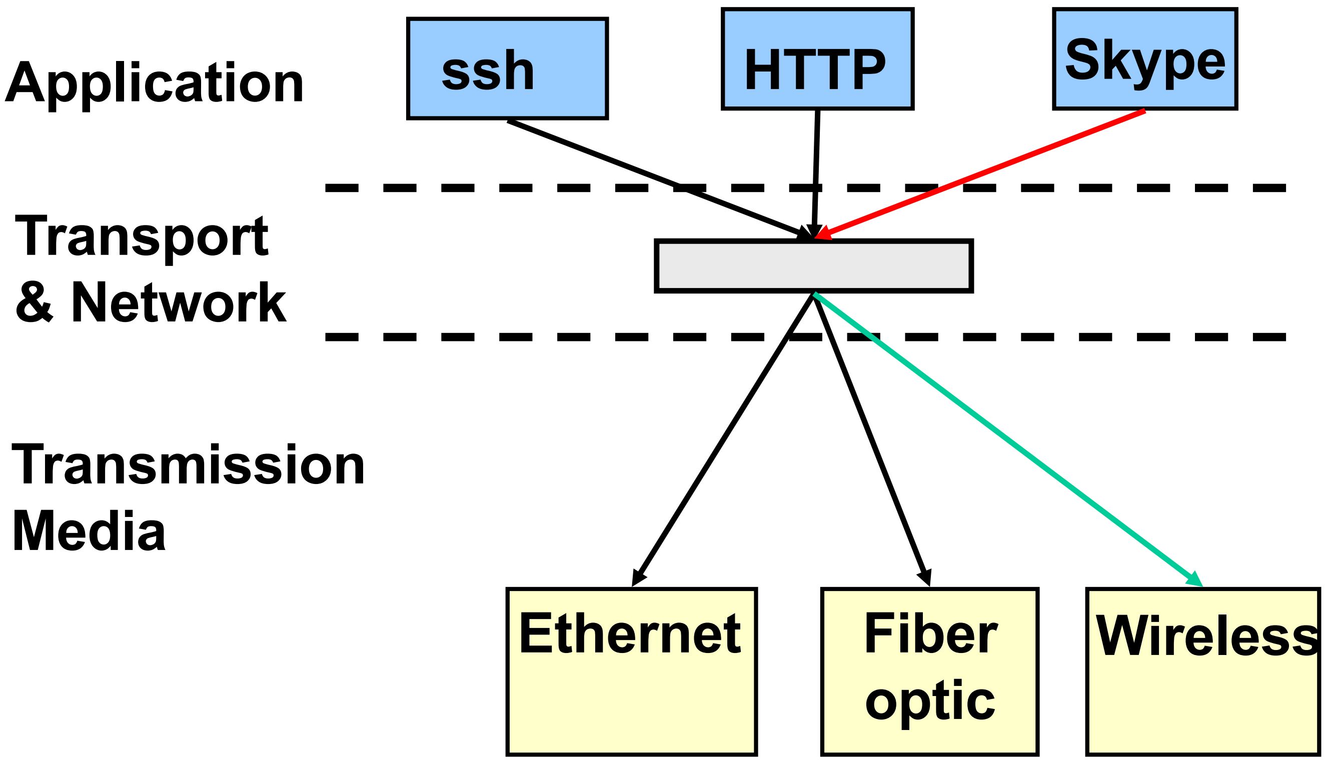
An Example: No Layering



- No layering: each new application has to be re-implemented for every network technology !

An Example: Benefit of Layering

- Introducing an intermediate layer provides a common abstraction for various network technologies



Is Layering Harmful?

- Layer N may duplicate lower level functionality
  - E.g., error recovery to retransmit lost data
- Information hiding may hurt performance
  - E.g. packet loss due to corruption vs. congestion
- Headers start to get really big
  - E.g., typically TCP + IP + Ethernet headers add up to 54 bytes
- Layer violations when the gains too great to resist
  - E.g., NAT
- Layer violations when network doesn't trust ends
  - E.g., Firewalls

Distributing Layers Across Network

- Layers are simple if only on a single machine
  - Just stack of modules interacting with those above/below

- But we need to implement layers across machines
  - Hosts
  - Routers
  - Switches

- What gets implemented where?

What Gets Implemented on Host?

- Hosts have applications that generate data/messages that are eventually put out on wire

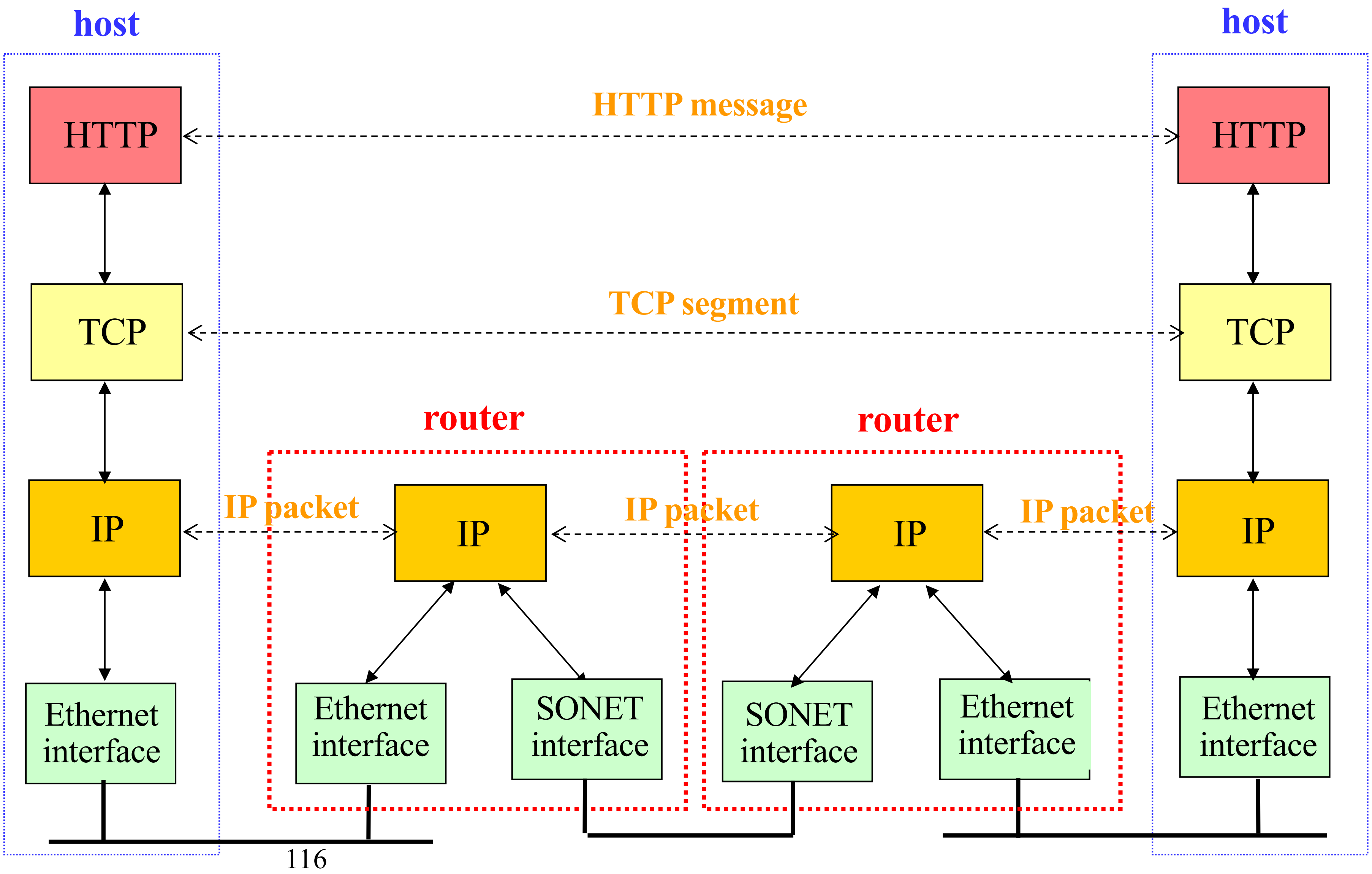
- At receiver host bits arrive on wire, must make it up to application

- Therefore, all layers must exist at host!

What Gets Implemented on Router?

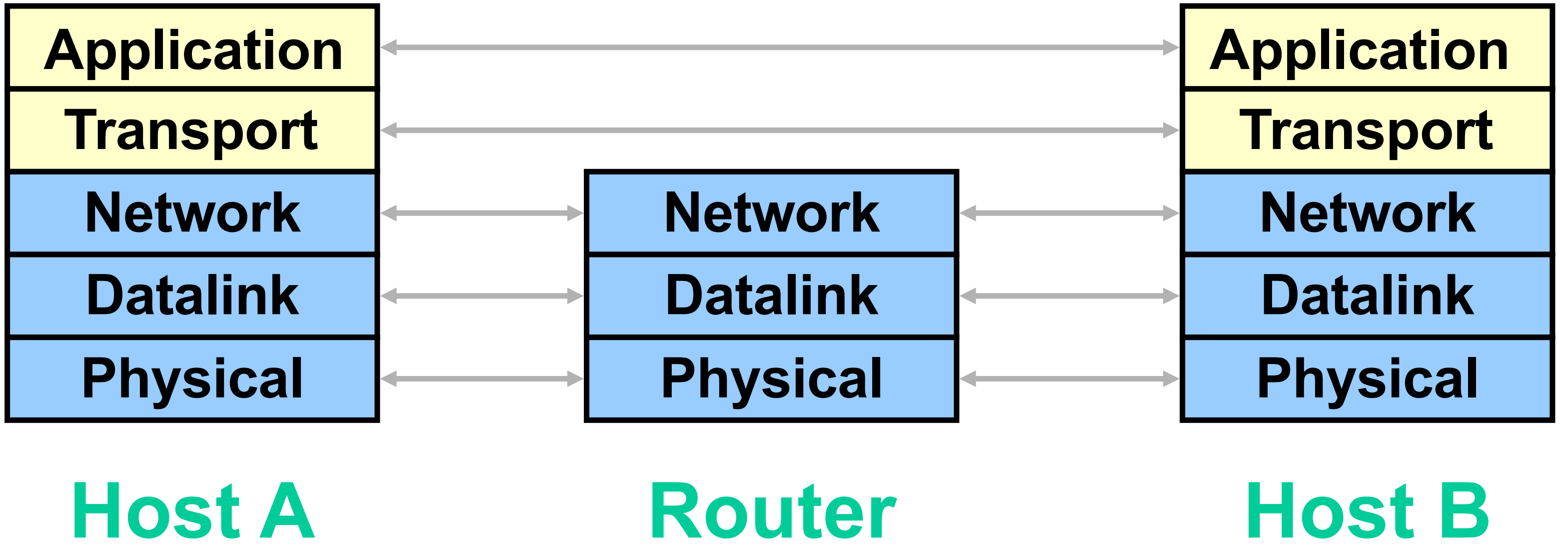
- Bits arrive on wire
  - Physical layer necessary
- Packets must be delivered to next-hop
  - datalink layer necessary
- Routers participate in global delivery
  - Network layer necessary
- Routers don't support reliable delivery
  - Transport layer (and above) ***not*** supported

Internet Layered Architecture



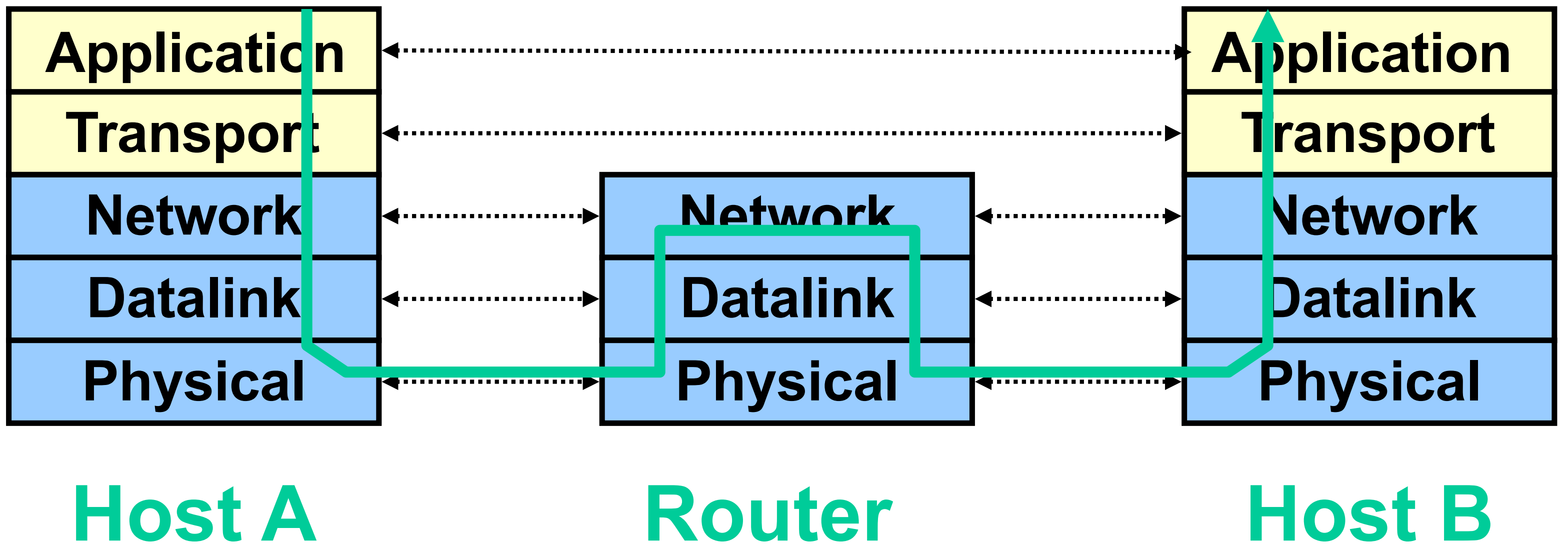
Logical Communication

- Layers interacts with peer's corresponding layer

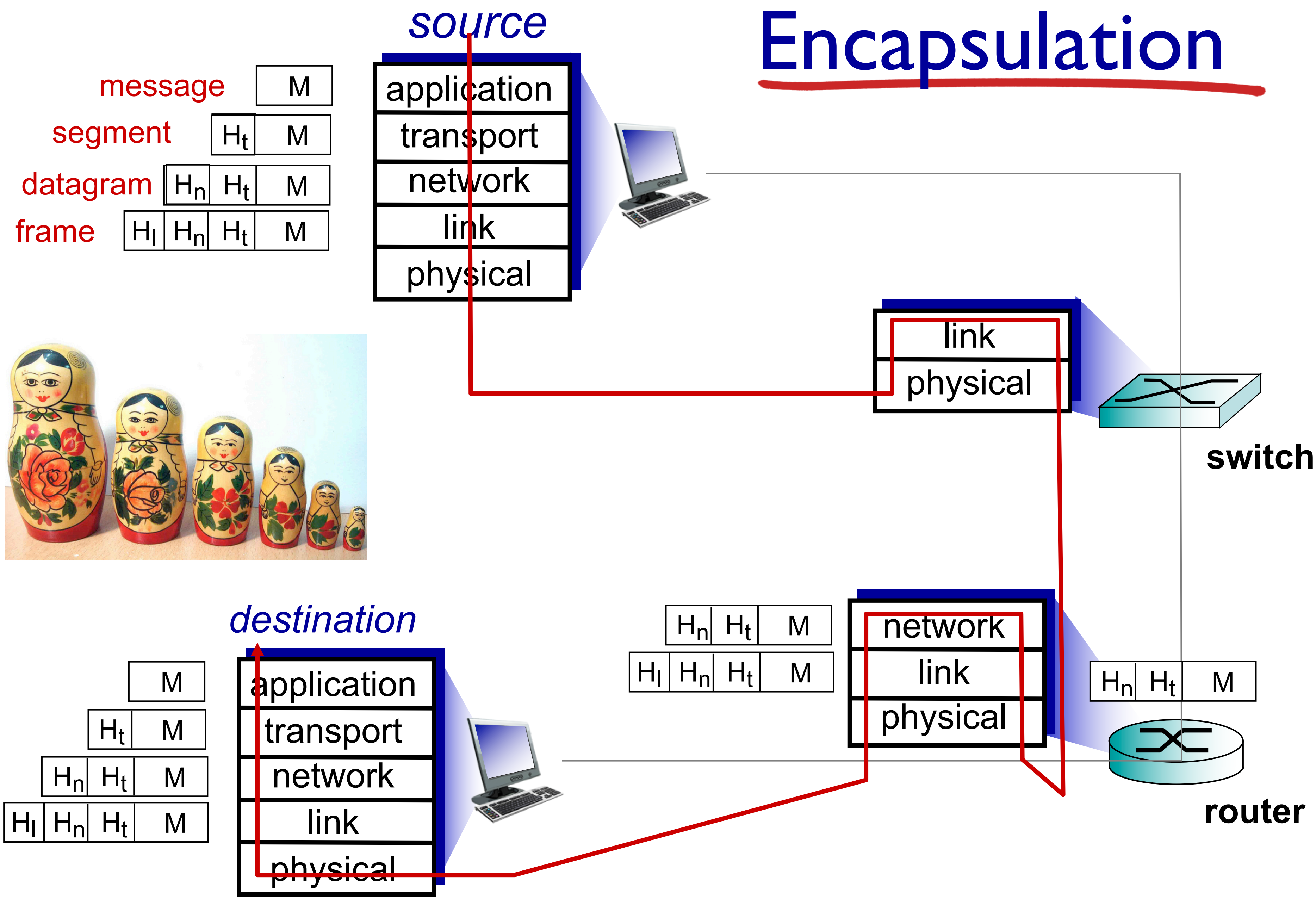


Physical Communication

- Communication goes down to physical network
- Then from network peer to peer
- Then up to relevant layer



Encapsulation



1. Introduction: 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

Self study

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
- Next Week
  - Application Layer