Chapter 1: 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 nets, physical media
- network core: packet/circuit switching, Internet structure
- performance: loss, delay, throughput
- protocol layers, service models
- security
- history

Chapter 1: roadmap

1.1 What *is* the Internet?
1.2 Network edge
- end systems, access networks, links

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

1.4 Delay, loss and throughput in packet-switched networks
1.5 Protocol layers, service models
1.6 Networks under attack: security
1.7 History

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

- PC
- server
- wireless laptop
- cellular handheld

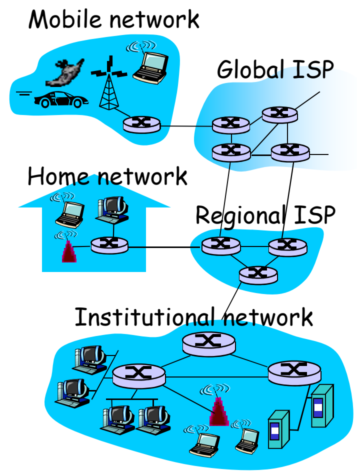
- ❑ billions of connected computing devices: *hosts = end systems*
  - running *network apps*

- access points
- wired links

- ❑ *communication links*
  - fiber, copper, coax, radio, satellite
  - transmission rate

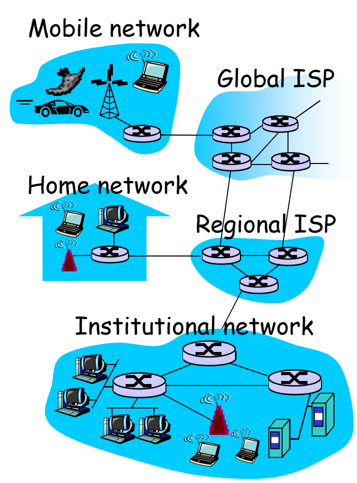
- router

- ❑ *Routers and switches* forward packets



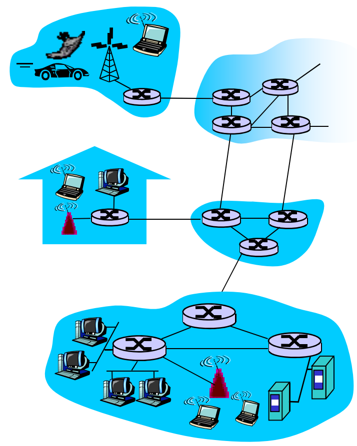
What's the Internet: architecture & protocols

- ❑ *Internet: "network of networks"*
  - loosely hierarchical
  - public Internet versus private intranet
- ❑ *protocols* control sending, receiving of msgs
  - e.g., TCP, IP, HTTP, Skype, Ethernet
- ❑ Internet standards
  - RFC: Request for comments
  - IETF: Internet Engineering Task Force



What's the Internet: a service view

- communication *infrastructure* enables distributed applications:
  - Web, VoIP, email, games, e-commerce, file sharing
- communication services provided to apps:
  - reliable data delivery from source to destination
  - "best effort" (unreliable) data delivery



What's a protocol?

human protocols:

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

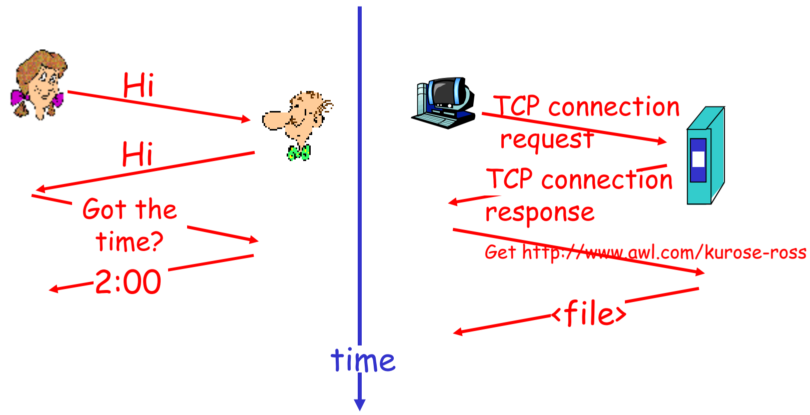
network protocols:

- 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, or timeout

What's a protocol?

a human protocol and a computer network protocol:

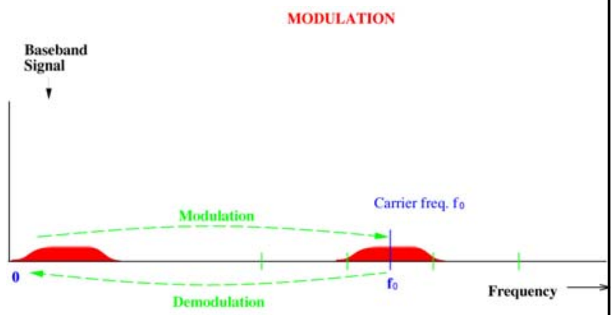


Q: Other human protocols?

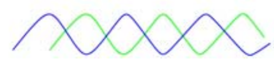
From physical media to communication channels—basic concepts (not in textbook)

Modulation and Demodulation

- Common examples: radio, television channels for analog signals
- Can also be used for digital signals (encoding binary data)



CARRIER WAVEFORM



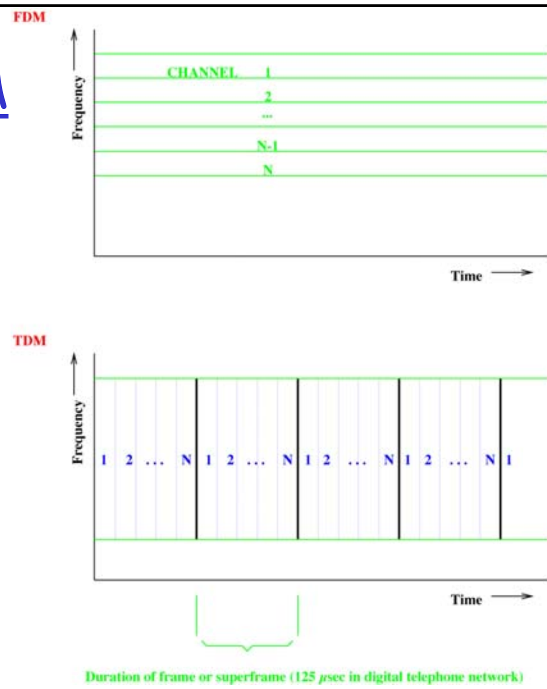
$A\cos(2\pi f_0 t+\theta)$

Shannon's Theorem

$$\boxed{C = B \log_2 (1 + S/N)}$$

where	C	max capacity in bits/sec
	B	bandwidth in hertz
	S/N	signal to noise ratio

FDM vs. TDM



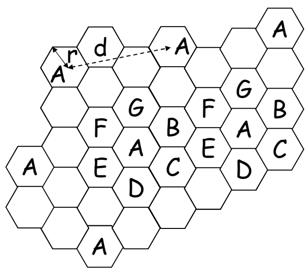
Duration of frame (or superframe) is 125 μsec in digital telephone networks

TDM in Telephone Networks

- Why 125 μsec for frame duration?
- Sampling Theorem: An analog signal can be reconstructed from samples taken at a rate equal to twice the signal bandwidth
- Bandwidth for voice signals is 4 Khz; for hi fidelity music, 22.05 Khz
- Sampling rate for voice = 8000 samples/sec or one voice sample every 125 μsec
- Digital voice channel (uncompressed), 8 bits x 8000/sec = 64 Kbps

Other Multiplexing Techniques

- Space division multiplex
  - Same frequency used in different cables
  - Same frequency used in different (nonadjacent) cells



- Wavelength division multiplex
  - Light pulses sent at different wavelengths in optical fiber
- Code division multiplex (in chapter 6 of text)

e.g., CDMA for cell phones

Chapter 1: roadmap

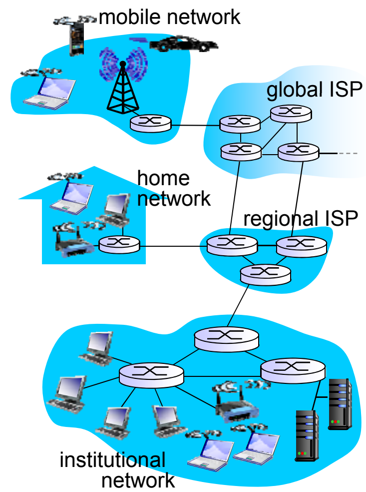
1.1 What *is* the Internet?
1.2 Network edge
- end systems, access networks, links

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

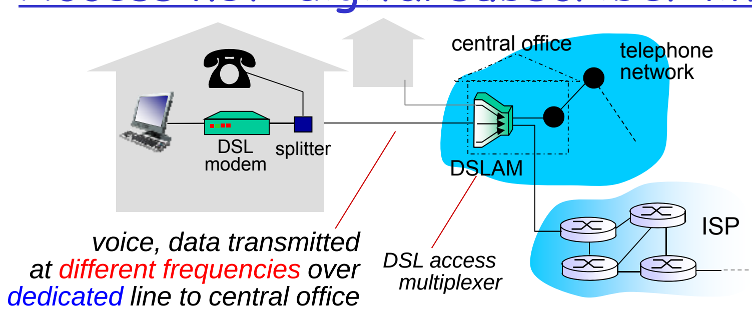
1.4 Delay, loss and throughput in packet-switched networks
1.5 Protocol layers, service models
1.6 Networks under attack: security
1.7 History

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

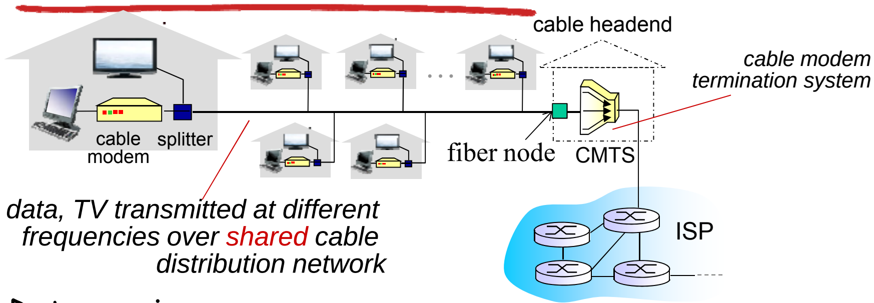


Access net: digital subscriber line (DSL)



- use FDM in telephone line to central office DSLAM
  - data over DSL line goes to Internet
  - voice over DSL line goes to telephone net
- asymmetric bandwidths/transmission rates (data download much faster than upload)

Access net - hybrid fiber coax (HFC)

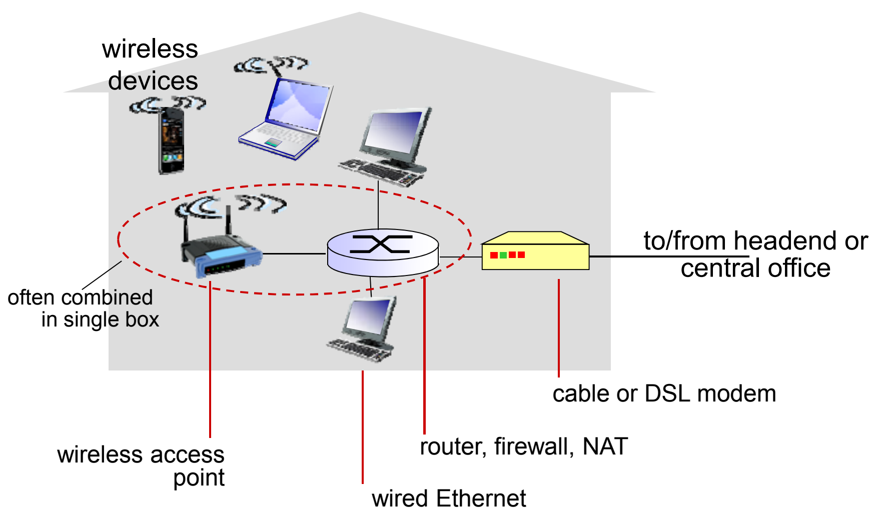


Data service

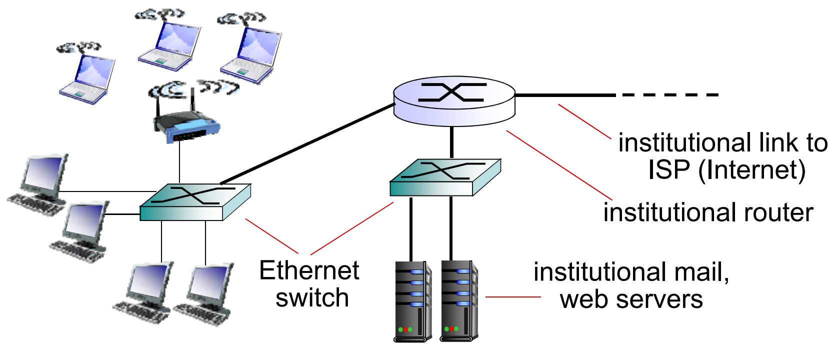
- homes *share* coax cable to cable headend
  (unlike DSL, which has dedicated access to central office)
- data channels have asymmetric rates and they are ***shared*** by homes - multiple access protocol required for uplink

Fiber to the home (Verizon, Google) – all optical switches

Access net: home network



Enterprise access networks (Ethernet)



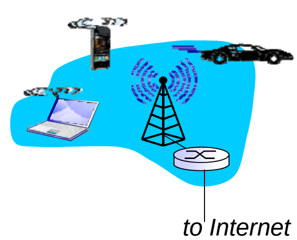
- today, end systems typically connect into Ethernet switch
  - 10 Mbps, 100Mbps, 1Gbps, 10Gbps transmission rates
- A large enterprise network is connected to multiple ISPs
  - multi-homing

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)

wide-area wireless access
- provided by telco (cellular) operators, 10's km
- up to 10s Mbps
- 3G, 4G: LTE



Chapter 1: roadmap

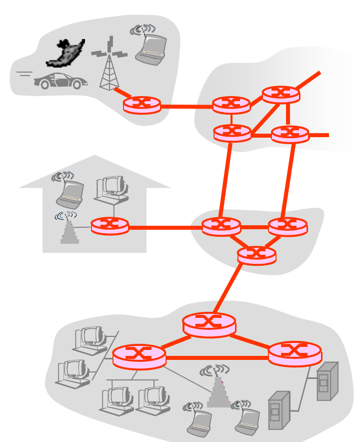
1.1 What *is* the Internet?
1.2 Network edge
- end systems, access networks, links

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

1.4 Delay, loss and throughput in packet-switched networks
1.5 Protocol layers, service models
1.6 Networks under attack: security
1.7 History

The Network Core

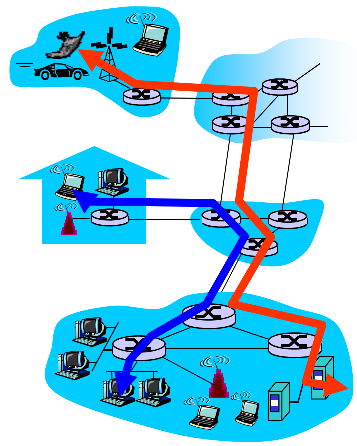
- mesh of interconnected routers
- *the* fundamental question: how is data transferred through net?
  - circuit switching: dedicated circuit per call: telephone net
  - packet-switching: data sent thru net in discrete "chunks"



Network Core: Circuit Switching

End-to-end resources reserved for each "call"

- E.g., link bandwidth
  - FDM, TDM
- end-to-end circuit-like (guaranteed) performance
- call setup required
  - resource piece *idle* if not used by the call *(no sharing)*

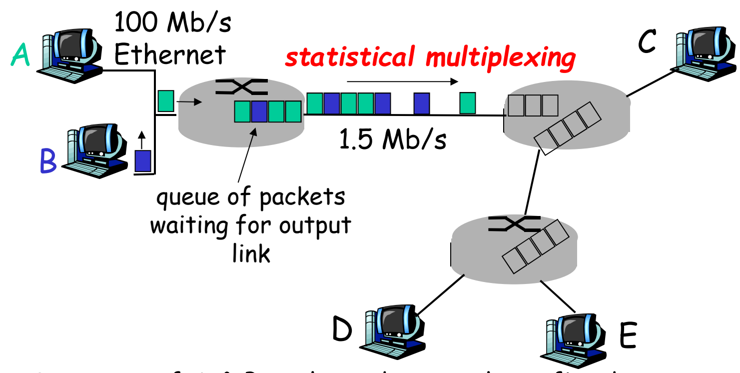


Numerical example

- How long does it take to send a file of 640,000 bits from host A to host B over a circuit-switched network?
  - all links are 1.536 Mbps
  - each link uses TDM with 24 slots/sec (i.e., one slot per circuit)
  - 500 msec to establish end-to-end circuit

Let's work it out!

Packet Switching: Statistical Multiplexing



- Sequence of A & B packets does not have fixed pattern
  bandwidth shared on demand → *statistical multiplexing*
- queueing delay, packet loss
- also called asynchronous time division multiplexing (ATDM)

Network Core: Packet Switching

each end-end data stream divided into packets

- packets of different users *share* network resources
- each packet uses full link bandwidth
- resources used *as needed*

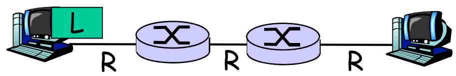
Bandwidth division into "pieces"
Dedicated allocation
Resource reservation

resource contention:

- aggregate resource demand can exceed amount available

  congestion: packets queue, wait for link use
- store and forward: packets move one hop at a time
  - Each node receives the complete packet before forwarding

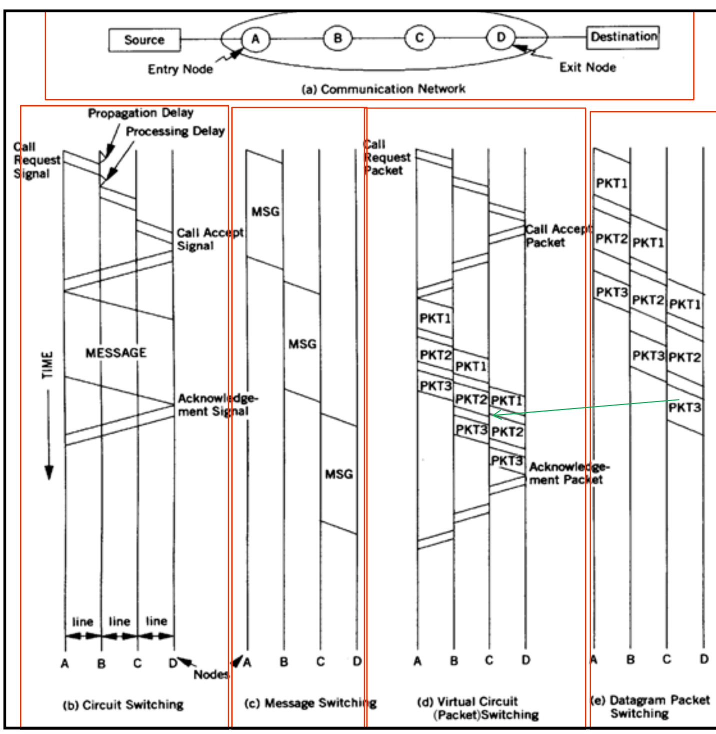
Disadvantage of store-and-forward



- takes L/R seconds to transmit (push out) a message of L bits on to link at R bps
- *store and forward:* entire message must arrive at router before it can be transmitted on next link

Example:
- L = 7.5 Mbits
- R = 1.5 Mbps
- End-to-end delay more than 15 seconds

- **A file/message larger than maximum packet size is transmitted as multiple packets**



Circuit vs. Message vs. Packet Switching

violates store-and-forward?

Packet Switching versus Message Switching

Advantages of packet switching

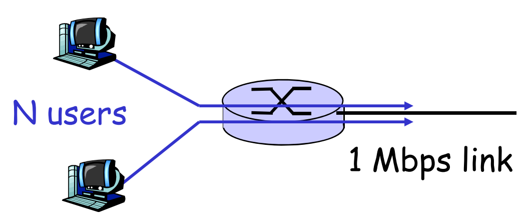
- Smaller end-to-end delay from pipelining
- Less data loss from transmission errors

Disadvantages of packet switching

- More header bits
- Additional work to do segmentation and reassembly

Packet switching versus circuit switching

- 1 Mb/s link
- each user:
  - 100 kb/s when "active"
  - active 10% of time (a "bursty" user)
- *circuit-switching:*
  - 10 users
- *packet switching:*
  - with 35 users, probability > 10 active at same time is less than .0004



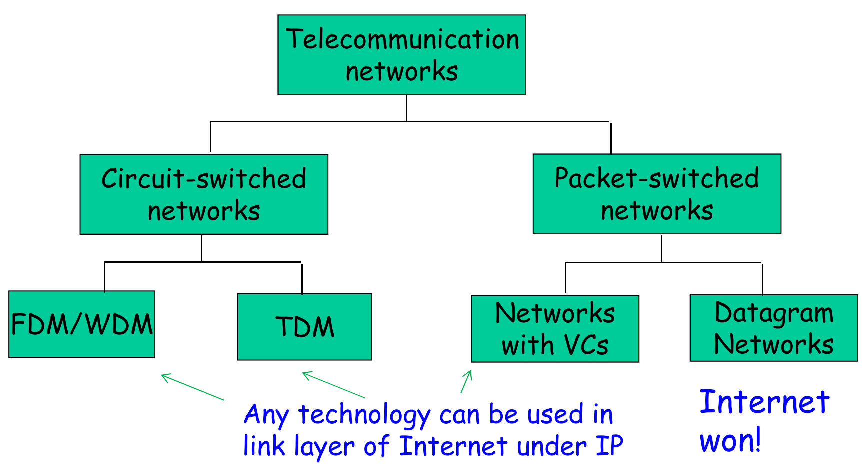
Q: how did we get value 0.0004?

Packet switching versus circuit switching

Is packet switching a "slam dunk winner?"

- great for bursty data
  - resource sharing
  - simpler, no call setup
- excessive congestion -> packet delay and loss
  - protocols needed for reliable data transfer, congestion control
- Q: How to provide circuit-like behavior?
  - bandwidth guarantees needed for *interactive* audio/video apps
  - solution may impact network neutrality

Network Taxonomy

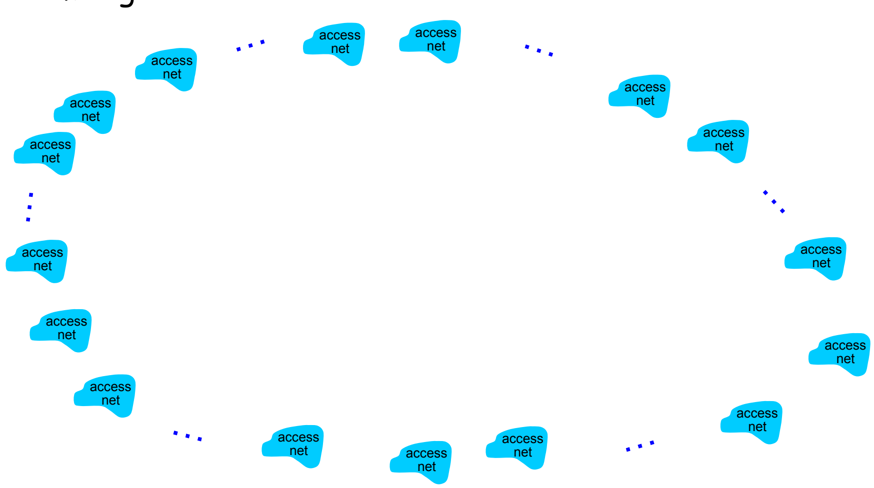


Internet structure: network of networks

- 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

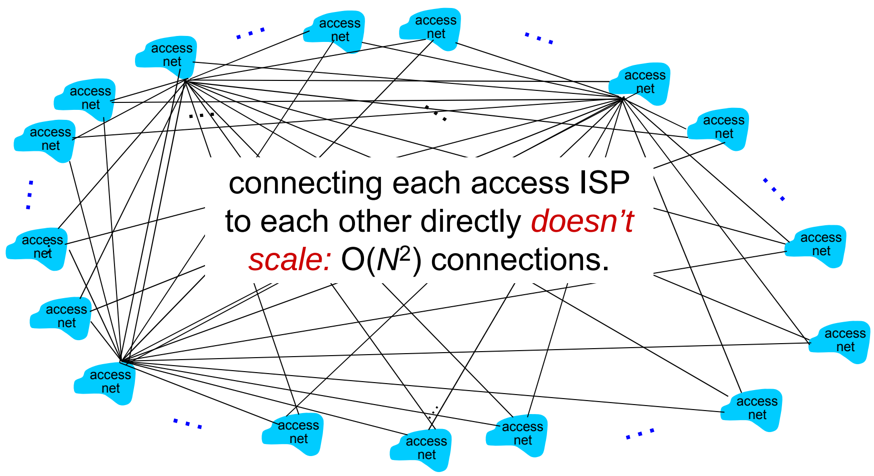
Internet structure: network of networks

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



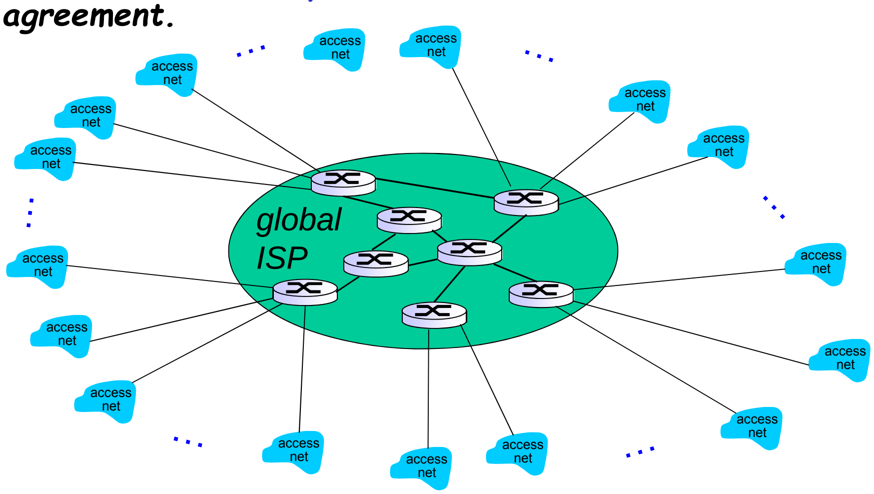
Internet structure: network of networks

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



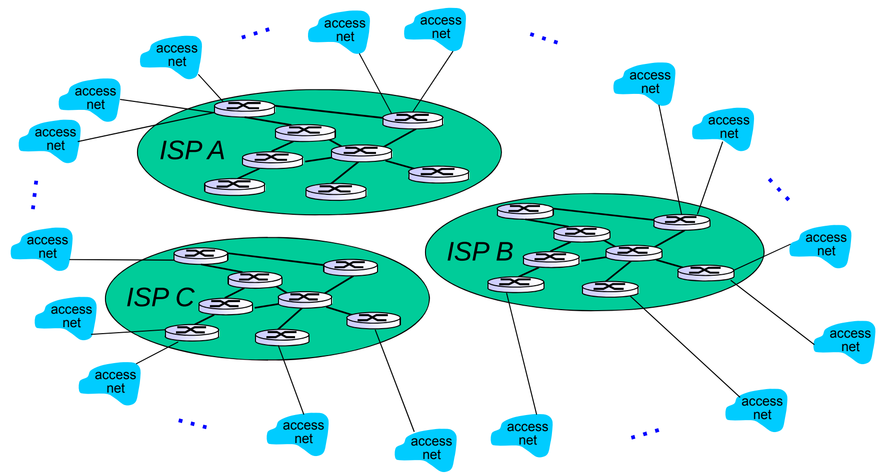
Internet structure: network of networks

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



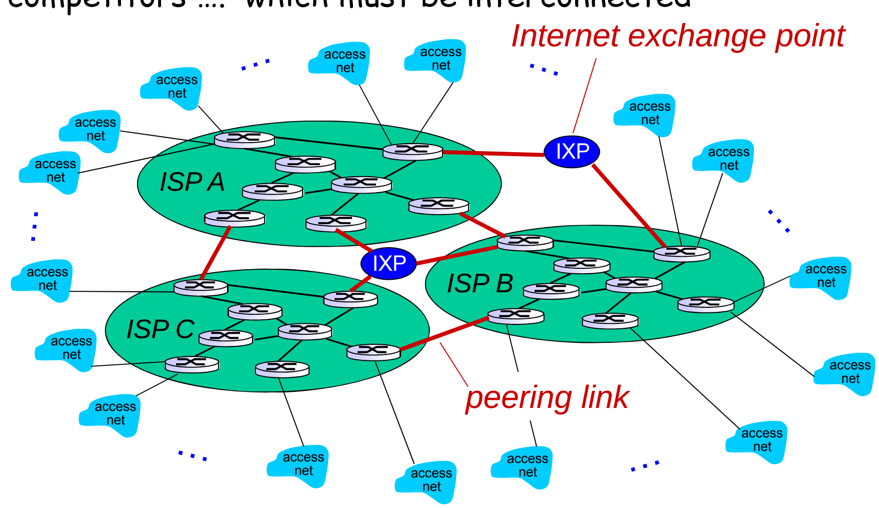
Internet structure: network of networks

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

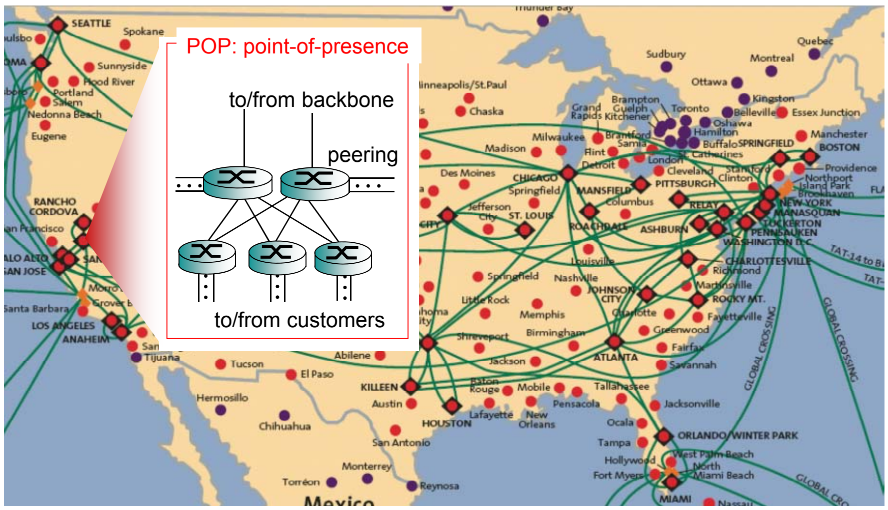


Internet structure: network of networks

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

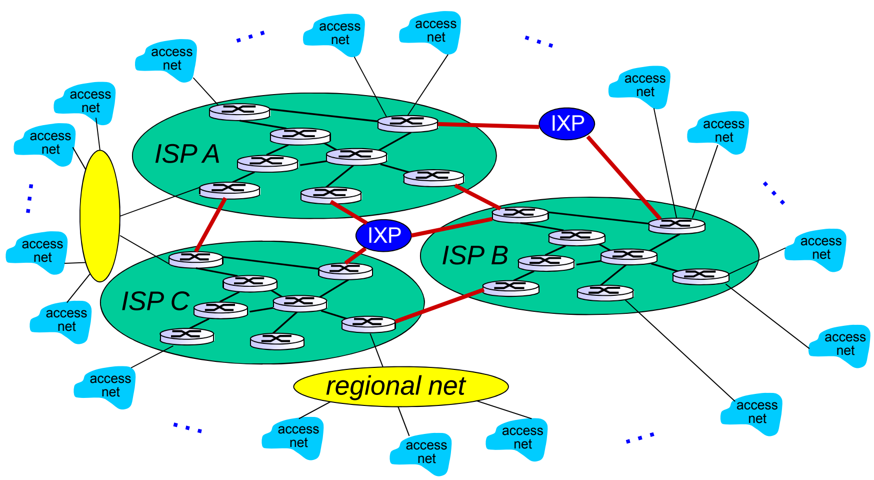


Tier-1 ISP: e.g., Sprint



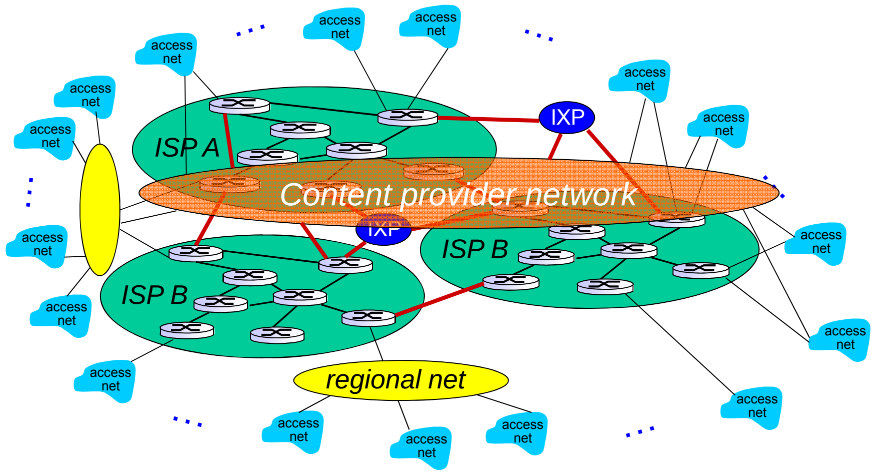
Internet structure: network of networks

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

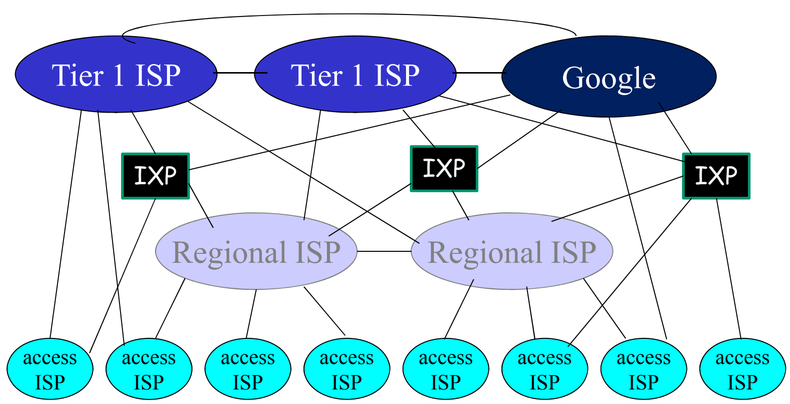


Internet structure: network of networks

... and a content provider network (e.g., Akamai, Google, Microsoft) may run its 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 networks (e.g., Google): private network that connects its **data centers** to Internet, often bypassing tier-1 and regional ISPs

Chapter 1: roadmap

1.1 What *is* the Internet?

1.2 Network edge

- end systems, access networks, links

1.3 Network core

- circuit switching, packet switching, network structure

1.4 Delay, loss and throughput in packet-switched networks

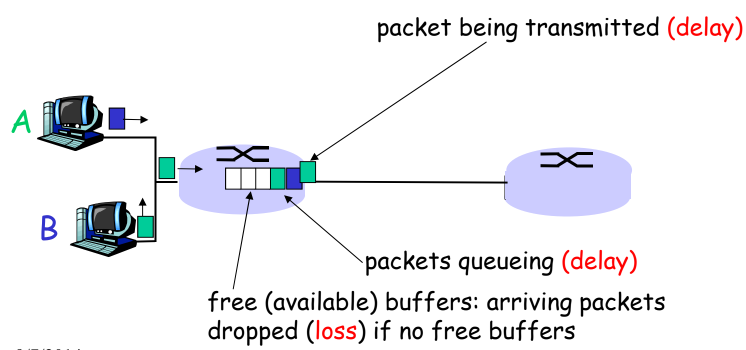
1.5 Protocol layers, service models

1.6 Networks under attack: security

1.7 History

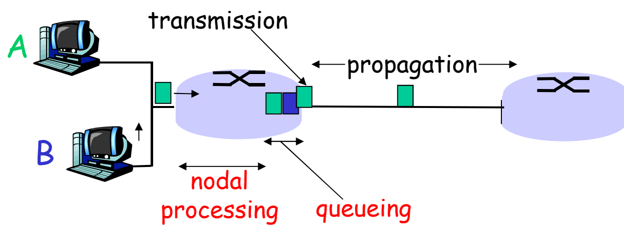
How do loss and delay occur?

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



Four sources of packet delay

- 1. nodal processing:
  - check bit errors
  - determine output link
- 2. queueing
  - time waiting at output link for transmission
  - depends on congestion level of router

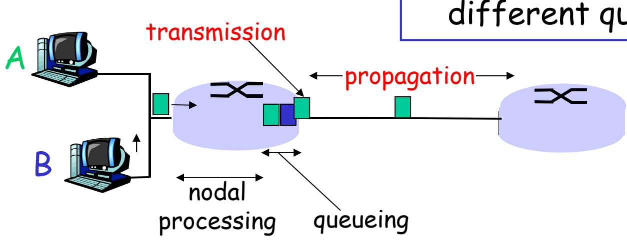


Delay in packet-switched networks

3. Transmission delay:
- R: link bandwidth (bps)
- L: packet length (bits)
- time to send bits into link = **L/R**

4. Propagation delay:
- d: length of physical link
- s: propagation speed in medium (~$2x10^8$ m/sec)
- propagation delay = **d/s**

Note: s and R are *very* different quantities!



End-to-End Delay

- Nodal delay (from when last bit of packet arrives at this node to when last bit arrives at next node)

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

- End-to-end delay over N identical nodes/links from client c to server s (from when last bit of packet leaves client to when last bit arrives at server)

  $d_{c\text{-}s} = d_{prop} + Nd_{nodal}$

- Round trip time (RTT)

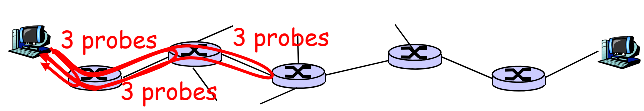
  $RTT = d_{c\text{-}s} + d_{s\text{-}c} + t_{server}$

  where t_{server} is server processing time

"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 and routes

traceroute: gaia.cs.umass.edu to www.eurecom.fr

Three 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

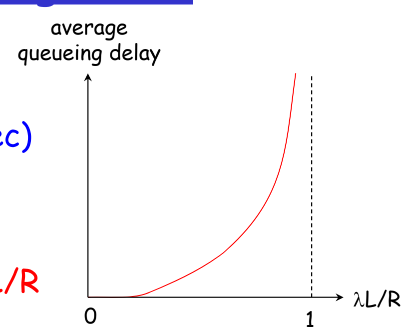
different packets

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

Queueing delay (waiting time)

- R: link bandwidth (bps)
- L: packet length (bits)
  - service rate = R/L (pkts/sec)
- λ: average packet arrival rate

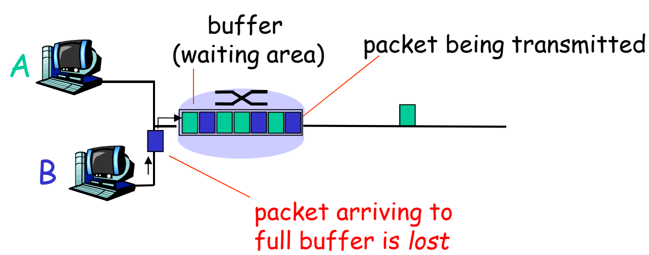
traffic intensity =
arrival rate/service rate = $\lambda L/R$



- $\lambda L/R \sim 0$: average queueing delay small
- $\lambda L/R \to 1$: delays become large
- $\lambda L/R > 1$: more "work" arriving than can be served, average delay infinite!
  - In reality, buffer overflow when $\lambda L/R \to 1$

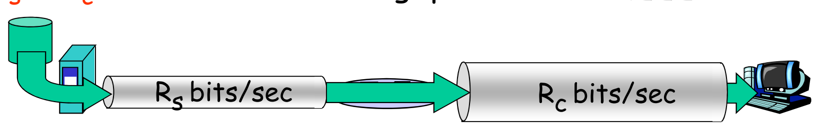
Packet loss

- buffer in router for each link has finite capacity
- lost packet may be retransmitted by previous node, by source end system, or not at all

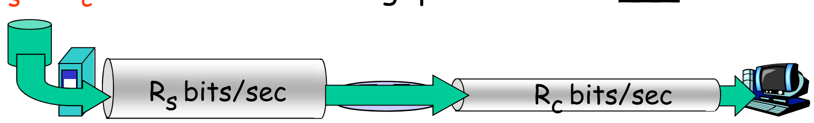


Throughput - rate at which bits are transferred from source to destination (in bits/sec.)

- $R_s < R_c$ end-end throughput less than ___ ?



- $R_s > R_c$ end-end throughput less than ___ ?

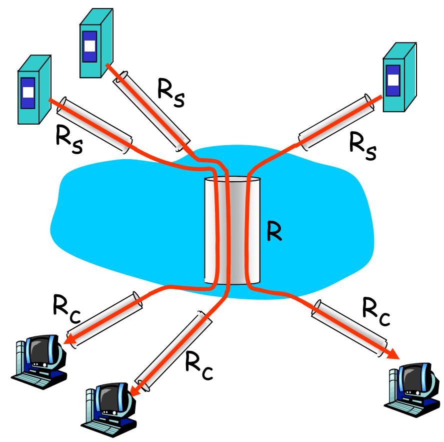


bottleneck link

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

Throughput: Internet scenario

- per-connection end-to-end throughput is approximately
  $\min(R_c, R_s, R/10)$
  - *Actually sharing a bottleneck equally is ideal but unrealistic*
- In practice: R_c or R_s is often the bottleneck
- or the server is the bottleneck

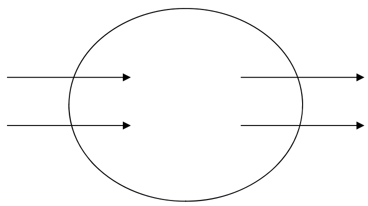


10 connections (fairly) share backbone bottleneck link R bits/sec

Little's law and a useful queueing delay formula

Little's Law

Average population
= (average delay) x
(throughput rate)



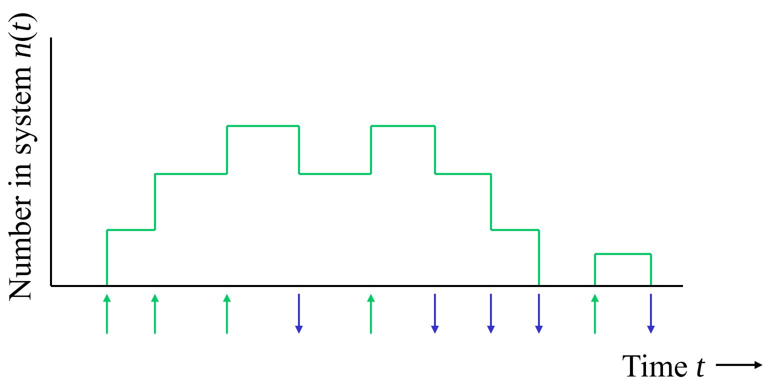
$$\text{average delay} = \frac{1}{N} \sum_{i=1}^{N} \text{delay}_i$$

where N is number of departures

throughput rate = N/T

where T is duration of experiment

average population
(to be defined)



$$\text{average population} = \frac{1}{\tau} \int_0^{\tau} n(t)dt$$

where τ is duration of the experiment random variable x

samples $x_1, x_2, \ldots, x_n$

mean (average) $\bar{x} = \frac{1}{n} \sum_{i=1}^{n} x_i$

second moment $\overline{x^2} = \frac{1}{n} \sum_{i=1}^{n} (x_i)^2 \geq (\bar{x})^2$

$$\text{mean residual life} = \frac{\overline{x^2}}{2\bar{x}} \geq \frac{\bar{x}}{2}$$

Special case: x is a constant

$$\overline{x^2} = (\bar{x})^2$$

$$\text{mean residual life} = \frac{(\bar{x})^2}{2\bar{x}} = \frac{\bar{x}}{2}$$

random variable x

with discrete values $x_1, x_2, \ldots, x_m$

let p_i = probability $[x = x_i]$ for $i = 1, 2, \ldots, m$

by definition

mean

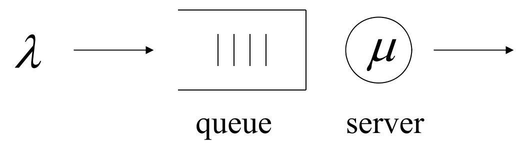
$$\bar{x} = \sum_{i=1}^{m} x_i \, p_i$$

second moment

$$\overline{x^2} = \sum_{i=1}^{m} x_i^2 \, p_i$$

(Aside: For a continuous random variable, use integration instead of summation.)

Single-Server Queue



$\bar{x}$ average service time, in seconds

μ service rate, in jobs/second ($\mu = 1/\bar{x}$)

λ arrival rate, in jobs/second

ρ utilization of server

Conservation of flow

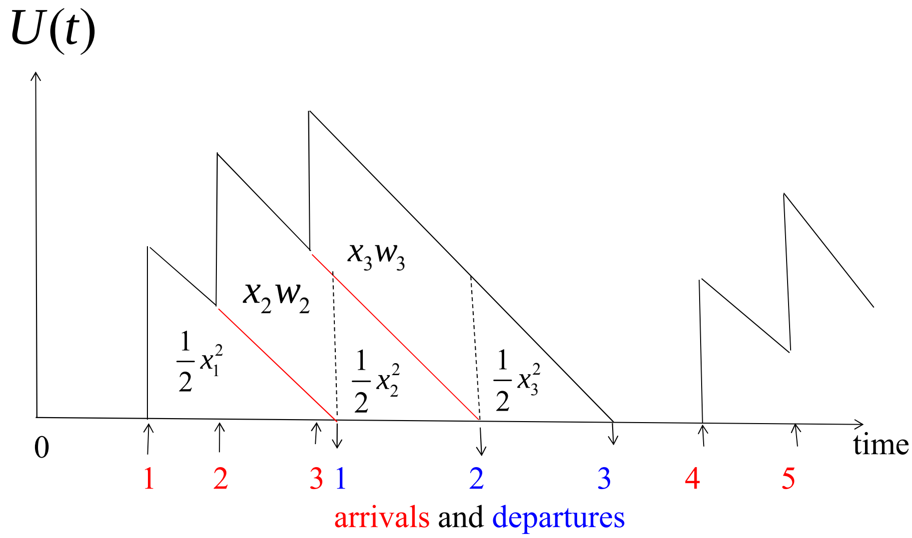
$$\lambda = \rho\mu$$

$$\rho = \frac{\lambda}{\mu} = \lambda\bar{x}$$

M/G/1 queue

- Single server
  - does not idle when there is work, no overhead, i.e., it performs 1 second of work per second
  - FIFO service
- Arrivals according to a Poisson process at rate λ jobs/second
- Service times of arrivals are $x_1, x_2, \ldots, x_i \ldots$ which are *independent, identically distributed* (with a general distribution)
- Average service time is $\bar{x}$, average wait is W, average delay is T = W + $\bar{x}$

Let $U(t)$ be the unfinished work at time t



Derivation of W

Time average of unfinished work is

$$\overline{U} = \frac{1}{\tau}\int_0^\tau U(t)dt$$

$$= \frac{1}{\tau}\left(\frac{1}{2}\sum_{i=1}^{n} x_i^2 + \sum_{i=1}^{n} x_i w_i\right)$$

x_i and w_i are independent

$$= \frac{n}{\tau}\left(\frac{1}{2}\overline{x_i^{\,2}} + \overline{x_i} \times \overline{w_i}\right) \quad \text{where } \overline{x_i w_i} = \overline{x_i} \times \overline{w_i}$$

For Poisson arrivals, the average wait is equal to $\overline{U}$ from the *Poisson arrivals see time average (PASTA)* Theorem

Derivation of W (cont.)

- The average wait is

$$W = \lambda\left(\frac{1}{2}\overline{x^2} + \bar{x}W\right) = \frac{\lambda\overline{x^2}}{2} + \lambda\bar{x}W$$

$$W(1-\rho) = \frac{\lambda\overline{x^2}}{2}$$

$$\boxed{W = \frac{\lambda\overline{x^2}}{2(1-\rho)}}$$

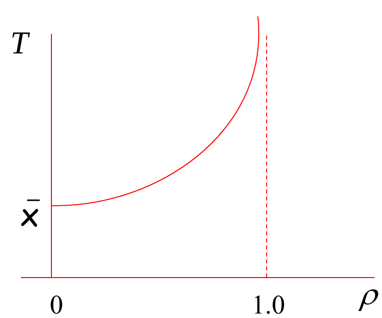
Pollaczek-Khinchin (P-K) mean value formula

M/G/1 queue

Markovian

General

Poisson



Average delay is

$$\boxed{T = \bar{x} + W = \bar{x} + \frac{\lambda\overline{x^2}}{2(1-\rho)}}$$

Also called Pollaczek-Khinchin (P-K) mean value formula

Special Cases

1. Service times have an exponential distribution (M/M/1). We then have

$$\overline{x^2} = 2(\overline{x})^2$$

$$W = \frac{\lambda(2)(\overline{x})^2}{2(1-\rho)} = \frac{\lambda(\overline{x})^2}{1-\rho} = \frac{\rho(\overline{x})}{1-\rho}$$

$$T = W + \overline{x}$$

$$= \frac{\rho\overline{x}}{1-\rho} + \overline{x} = \frac{\rho\overline{x} + \overline{x} - \rho\overline{x}}{1-\rho}$$

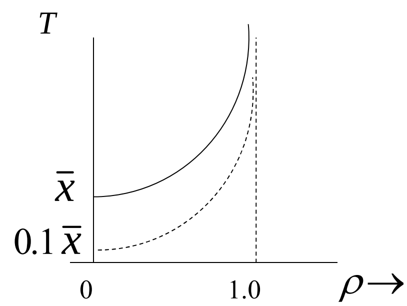
$$= \frac{\overline{x}}{1-\rho} = \boxed{\frac{\rho}{1-\rho}\frac{1}{\lambda}}$$

T decreases as λ increases

$$\lambda \to 10\lambda$$

$$\mu \to 10\mu$$

$$\boxed{\rho = \frac{10\lambda}{10\mu} = \frac{\lambda}{\mu}}$$



2. Service times are constant (deterministic)

$$\overline{x^2} = (\overline{x})^2$$

↓
M/D/1

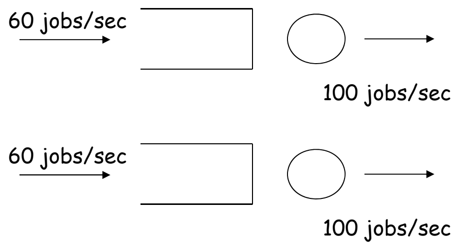
$$W = \frac{\lambda(\overline{x})^2}{2(1-\rho)} = \frac{\rho\overline{x}}{2(1-\rho)}$$

$$T = W + \overline{x}$$

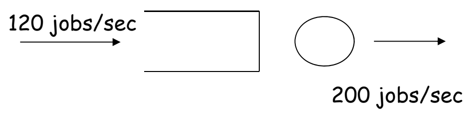
$$\boxed{T = \frac{\rho(2-\rho)}{2(1-\rho)}\frac{1}{\lambda}}$$

T decreases as λ increases

Two Servers and Two Queues:



Single Higher Speed Server:



Chapter 1: roadmap

1.1 What *is* the Internet?
1.2 Network edge
- end systems, access networks, links

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

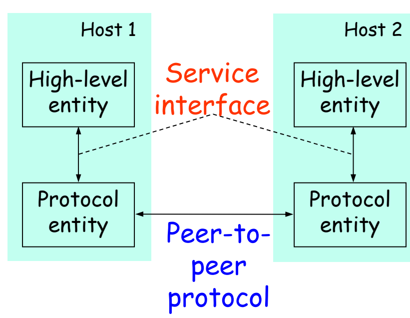
1.4 Delay, loss and throughput in packet-switched networks
1.5 Protocol layers, service models
1.6 Networks under attack: security
1.7 History

Layered architecture

- as **reference model** for protocol design by community effort
  - decompose a large system into smaller pieces which can be designed and implemented by different people/teams
- **modularity** eases maintenance and evolution of system
  - allows changes in implementation method so long as API remains the same, e.g., different Ethernet technologies
- strict layering **often violated** for efficient protocol implementation
  - cross-layer design

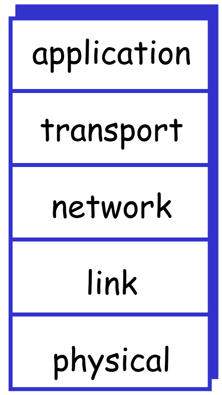
Each protocol

- involves two or more peers
- two kinds of specifications
  - service interface: operations a local user can perform on a protocol entity and get results
  - peer-peer protocol: format and meaning of messages exchanged by protocol entities (also called peers) to provide protocol service
- The term "protocol" generally refers to peer-peer spec



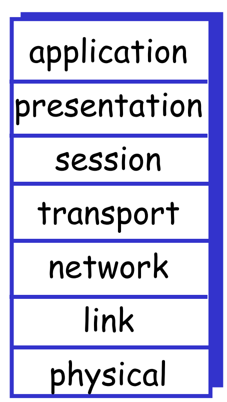
Internet protocol stack

- application: protocols that support 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
  - PPP, Ethernet, 802.11 (WiFi)
- physical: bits "on the wire"



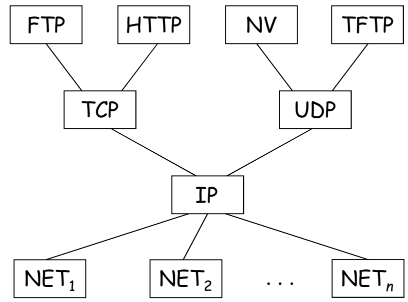
ISO/OSI reference model

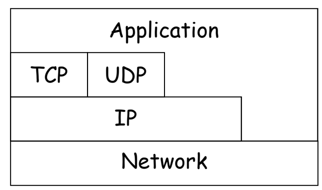
- presentation: allow applications to interpret meaning of data, e.g., encryption, compression, machine-specific conventions
- *session:* synchronization, checkpointing, recovery of data exchanged
- Internet stack "missing" these layers!
  - these services, *if needed*, must be implemented in application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



Internet Architecture

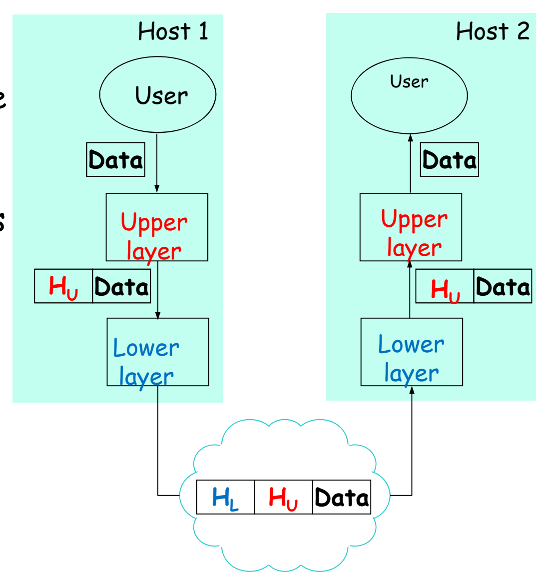
- Internet Engineering Task Force (IETF)
- application protocols support applications
- multiplexing and demultiplexing
- *hourglass* shape (only IP in network layer)
  - best effort service => any delivery service can be used by IP
- limitation of hourglass





Encapsulation

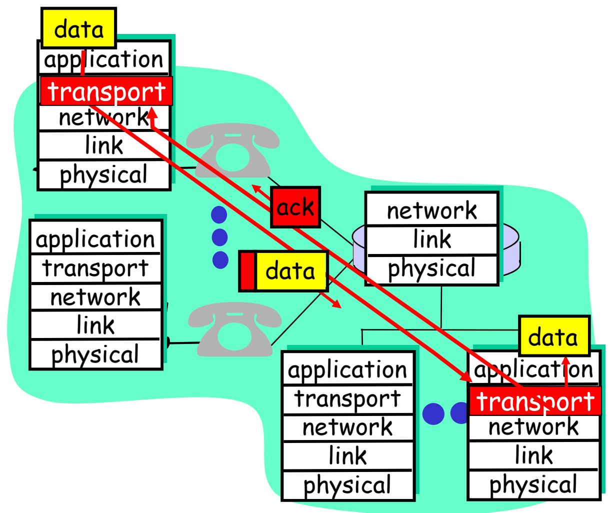
- Protocol peers provide a data delivery service
- How do protocol peers in different machines exchange protocol messages between themselves?
  - In protocol header encapsulated and de-encapsulated



Logical communication between peers

E.g.: transport

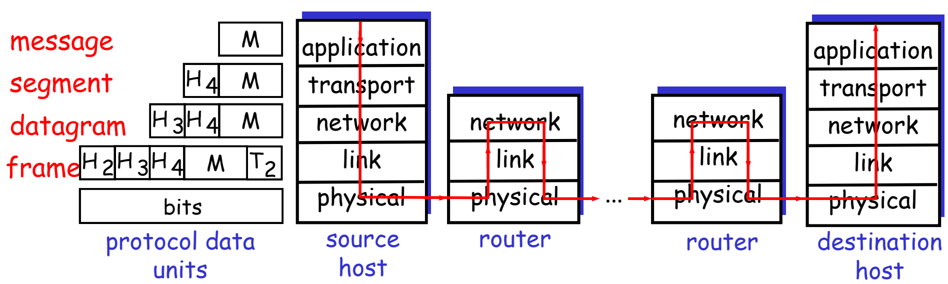
- accept data from application
- add addressing, reliability check info to form a message
- send message to peer via a delivery service
- wait for peer's reply (ack)



Physical path of data

Each layer takes data (service data unit) from above

- adds header to create its own protocol data unit
- passes protocol data unit to layer below



Note: In the past, a switch implements only two layers (physical and link). Nowadays many switches function as routers (3 layers)

Chapter 1: roadmap

1.1 What *is* the Internet?

1.2 Network edge

- end systems, access networks, links

1.3 Network core

- circuit switching, packet switching, network structure

1.4 Delay, loss and throughput in packet-switched networks

1.5 Protocol layers, service models

1.6 Networks under attack – please read on your own

1.7 History – please read on your own

End of Chapter 1