## Computer Networking

Lent Term M/W/F 11-midday
LT1 in Gates Building

Slide Set 1

Andrew W. Moore
andrew.moore@cl.cam.ac.uk
January 2013

1

## Topic 1 Foundation

- Administrivia
- Multiplexing
- Abstraction
- Layering
- Layers and Communications
- Entities and Peers
- Channels
- The Internet
- What is a protocol?
- Network edge; hosts, access net, physical media
- Network core: packet/circuit switching, Internet structure
- Performance: loss, delay, throughput

2

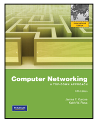
## Course Administration

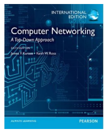
### Commonly Available Texts

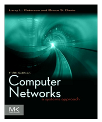
- Computer Networking: A Top-Down Approach
  Kurose and Ross, 6th edition 2013, Addison-Wesley
  (5th edition is also commonly available)
- Computer Networks: A Systems Approach
  Peterson and Davie, 5th edition 2011, Morgan-Kaufman

### Other Selected Texts (non-representative)

- Internetworking with TCP/IP, vol. I + II
  Comer & Stevens, Prentice Hall
- UNIX Network Programming, Vol. I
  Stevens, Fenner & Rudoff, Prentice Hall







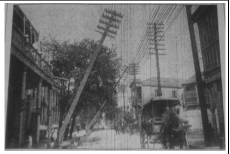

3

## Thanks

- Slides are a fusion of material from
  Ian Leslie, Richard Black, Jim Kurose, Keith Ross, Larry Peterson, Bruce Davie, Jen Rexford, Ion Stoica, Vern Paxson, Scott Shenker, Frank Kelly, Stefan Savage, Jon Crowcroft , Mark Handley and Adam Greenhalgh (and to those others I've forgotten, sorry.)
- Supervision material is largely drawn from
  Stephen Kell, Andy Rice
- Practical material remains Beta for this year
  But would be impossible without Nick McKeown, David Underhill, Andrew Ryrie and Antanas Uršulis
- Finally thanks to the Part 1b students past and Andrew Rice for all the tremendous feedback.

4

## Multiplexing



Sharing makes things efficient (cost less)

- One airplane/train for 100 people
- One telephone for many calls
- One lecture theatre for many classes
- One computer for many tasks
- One network for many users

5

## Multiplexing

Disadvantages of Multiplexing?

- Might have to wait
- Might not know how long you have to wait
- Might never get served

Multiplexing is the *action* of sharing of resources

*in contrast*

Concurrency is all about *how* resources are shared

6

## Abstraction

A mechanism for breaking down a problem

*what* not *how*

- eg Specification *versus* implementation
- eg Modules in programs

Allows replacement of implementations without affecting system behavior

*Vertical* versus *Horizontal*

*"Vertical"* what happens in a box "How does it attach to the network?"

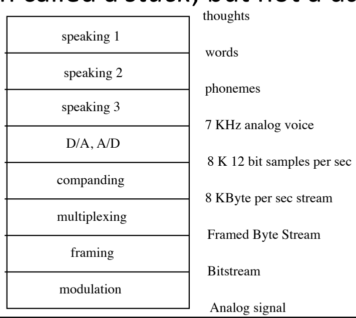
*"Horizontal"* the communications paths running through the system

**Hint:** paths are build on top of ("layered over") other paths

7

## Layering

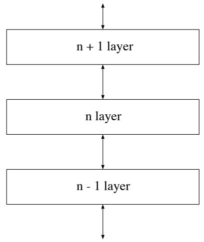
- A restricted form of abstraction: system functions are divided into layers, one built upon another
- Often called a *stack*; but not a data structure!




8

# Layers and Communications

- Interaction only between adjacent layers
- *layer n* uses services provided by *layer n-1*
- *layer n* provides service to *layer n+1*
- Bottom layer is physical media
- Top layer is application




9

# Entities and Peers

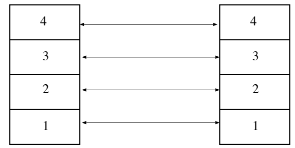
*Entity* – a *thing* (an independent existence)

Entities *interact* with the layers above and below

Entities *communicate* with *peer* entities

- same level but different place (eg different person, different box, different host)

Communications between peers is supported by entities at the lower layers




10

# Entities and Peers

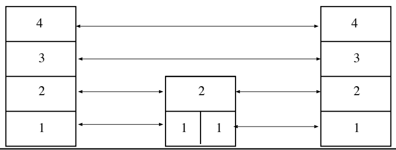
Entities usually do something useful

- Encryption – Error correction – Reliable Delivery
- Nothing at all is also reasonable

Not all communications is end-to-end

Examples for things in the middle

- IP Router – Mobile Phone Cell Tower
- Person translating French to English



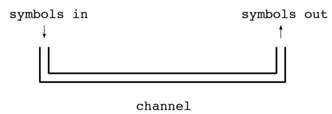

11

# Channels

(This channel definition is very abstract)

- Peer entities communicate over channels
- Peer entities provide higher-layer peers with higher-layer channels

*A channel is that into which an entity puts symbols and which causes those* symbols *(or a reasonable approximation) to appear somewhere else at a later point in time.*




12

# Channel Characteristics

**Symbol type**: bits, packets, waveform
**Capacity**: bandwidth, data-rate, packet-rate
**Delay**: fixed or variable
**Fidelity**: signal-to-noise, bit error rate, packet error rate

**Cost**: per attachment, for use
**Reliability**
**Security**: privacy, unforgability
**Order preserving**: always, almost, usually
**Connectivity**: point-to-point, to-many, many-to-many

Examples:
- Fibre Cable
- 1 Gb/s channel in a network
- Sequence of packets transmitted between hosts
- A telephone call (handset to handset)
- The audio channel in a room
- Conversation between two people

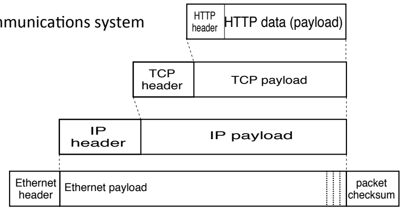
13

# Layering and Embedding

In Computer Networks we often see higher-layer information embedded within lower-layer information
- Such embedding can be considered a form of layering
- Higher layer information is generated by stripping off headers and trailers of the current layer
- eg an IP entity only looks at the IP headers

***BUT embedding is not the only form of layering***

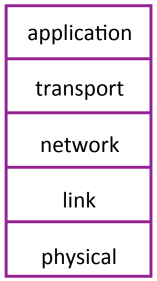
Layering is to help understand a communications system
**NOT**
determine implementation strategy




14

# Internet protocol stack

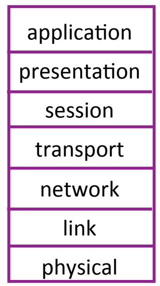
- application: supporting network applications
  - FTP, SMTP, HTTP
- transport: process-process data transfer
  - TCP, UDP
- network: routing of datagrams from source to destination
  - IP, routing protocols
- link: data transfer between neighboring network elements
  - Ethernet
- physical: bits "on the wire"




15

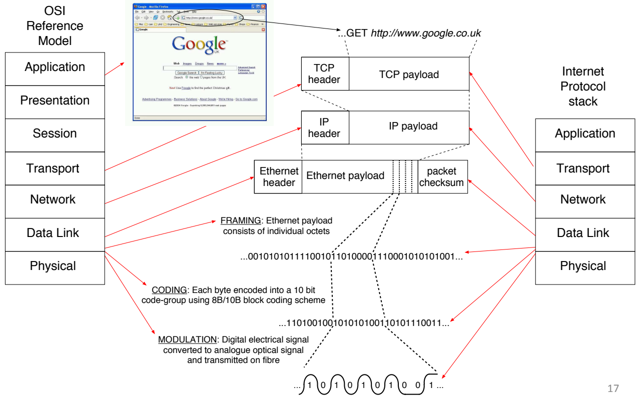
# ISO/OSI reference model

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



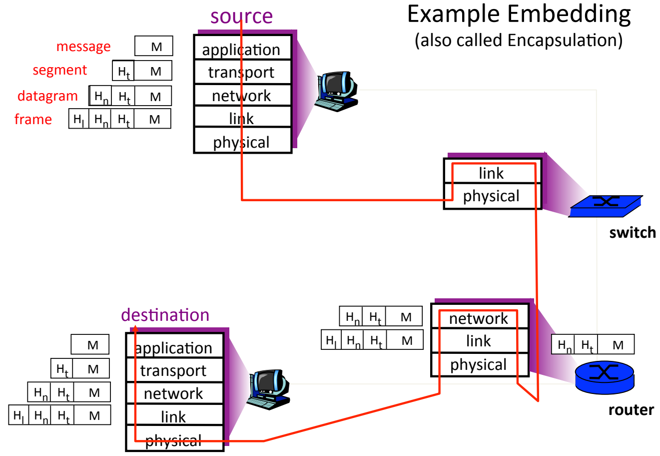

16

# Internet protocol stack *versus* OSI Reference Model

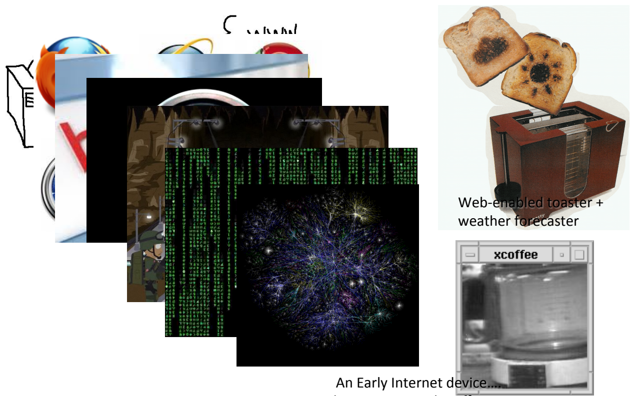





Example Embedding
(also called Encapsulation)




# What is the Internet: general public

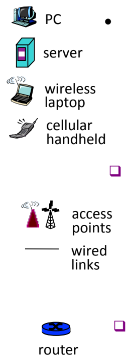


Web-enabled toaster + weather forecaster

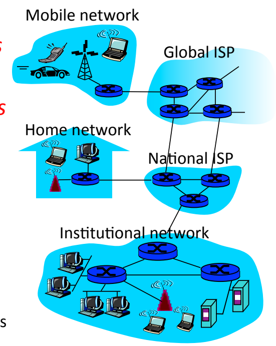
An Early Internet device....
The Computer Lab Coffee Pot
http://www.cl.cam.ac.uk/coffee/coffee.html



# What is the Internet: "nuts and bolts" view

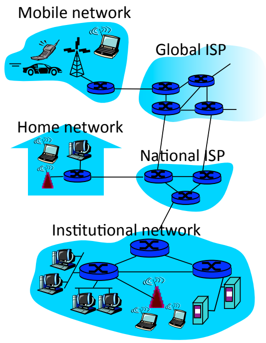



- millions of connected computing devices: *hosts = end systems*
  - running *network apps*
- *communication links*
  - fiber, copper, radio, satellite
  - transmission rate = *bandwidth*
- *routers:* forward packets (chunks of data)



## What is the Internet: "nuts and bolts" view

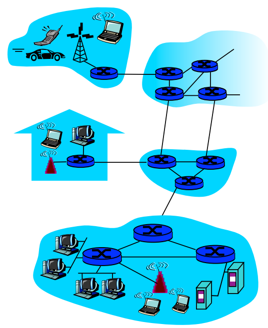
- *protocols* control sending, receiving of msgs
  - e.g., TCP, IP, HTTP, Skype, Ethernet
- *Internet:* "network of networks"
  - loosely hierarchical
  - public Internet versus private intranet
- Internet standards
  - RFC: Request for comments
  - IETF: Internet Engineering Task Force






## What is 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 is 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

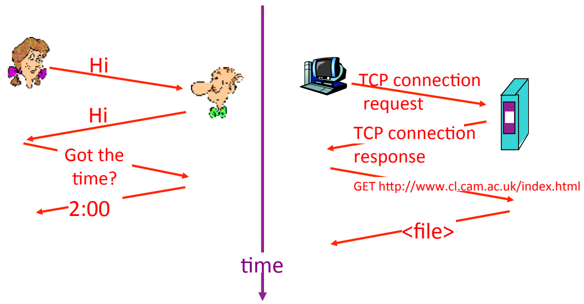
network protocols:
- machines rather than humans
- all communication activity in Internet governed by protocols

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



## What is a protocol?

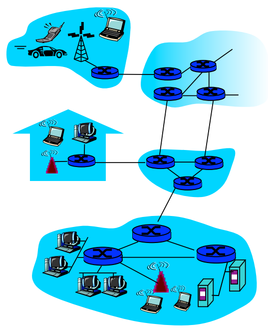
a human protocol and a computer network protocol:




Q: Other human protocols?

## A closer look at network structure:

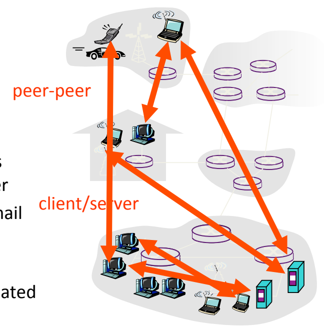
- network edge: applications and hosts
- access networks, physical media: wired, wireless communication links
- network core:
  - interconnected routers
  - network of networks



25

## The network edge:

- end systems (hosts):
  - run application programs
  - e.g. Web, email
  - at "edge of network"
- client/server model
  - client host requests, receives service from always-on server
  - e.g. Web browser/server; email client/server
- peer-peer model:
  - minimal (or no) use of dedicated servers
  - e.g. Skype, BitTorrent




26

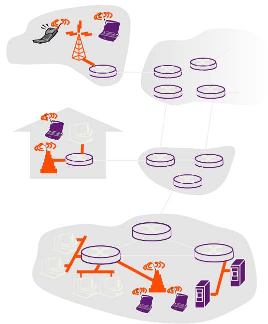
## Access networks and physical media

*Q: How to connect end systems to edge router?*

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

*Keep the channel in mind:*

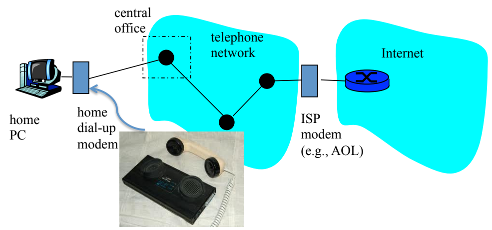
- Access bandwidth?
- shared or dedicated?



27

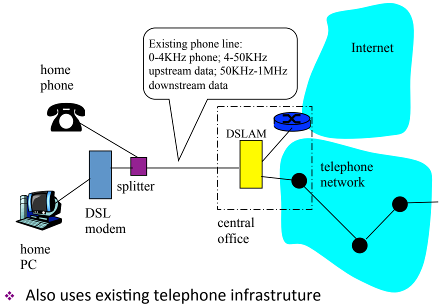
## Dial-up Modem

(archeology – unless you travel a lot)



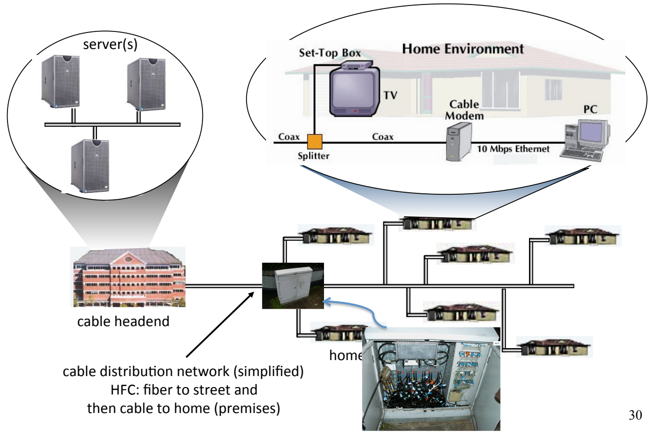

- Uses existing telephony infrastructure
  - Home is connected to central office
- up to 64Kbps direct access to router (often less)
- Can't surf and phone at same time: not "always on"

## Digital Subscriber Line (DSL)




- Also uses existing telephone infrastruture
- up to 1 Mbps upstream (today typically < 256 kbps)
- up to 8 Mbps downstream (today typically < 1 Mbps)
- dedicated physical line to telephone central office

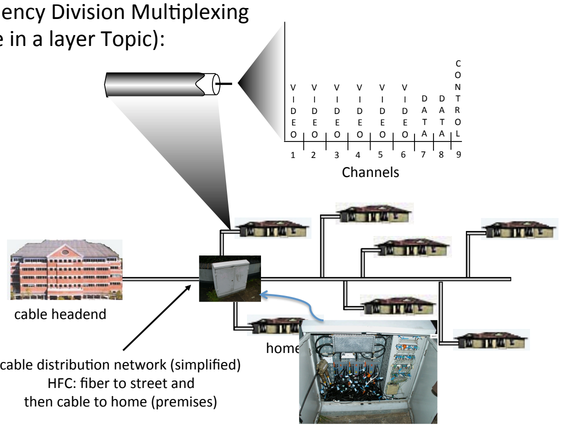
## Cable Network Architecture: Overview




30

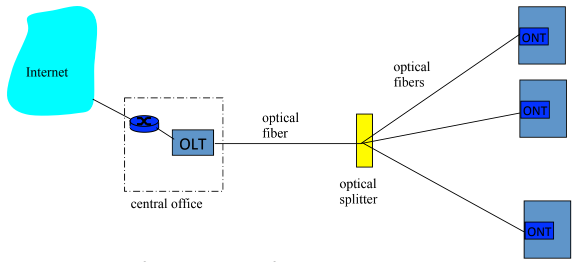
## Cable Network Architecture: Overview

Frequency Division Multiplexing
(more in a layer Topic):




31

## Fiber to the Home




- Optical links from central office to the home
- Two competing optical technologies:
  - Passive Optical network (PON)
  - Active Optical Network (PAN)
- Much higher Internet rates; fiber also carries television and phone services

32

## Physical Channels

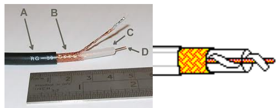
also known as *Physical Media*

Twisted Pair (TP)

- two insulated copper wires
  - Category 3: traditional phone wires, 10 Mbps Ethernet
  - Category 6: 1Gbps Ethernet
- Shielded (STP)
- Unshielded (UTP)

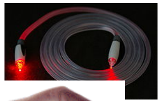
Coaxial cable:

- two concentric copper conductors
- bidirectional
- baseband:
  - single channel on cable
  - legacy Ethernet
- broadband:
  - multiple channels on cable
  - HFC (Hybrid Fiber Coax)



Fiber optic cable:

- high-speed operation
- point-to-point transmission
- (10's-100's Gps)
- low error rate
- immune to electromagnetic noise



33

## Physical media: radio

- Bidirectional and multiple access
- propagation environment effects:
  - reflection
  - obstruction by objects
  - interference

Radio link types:

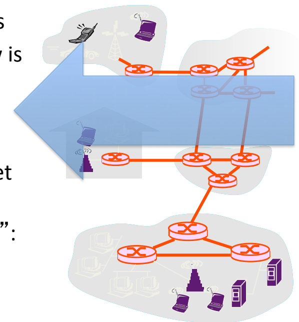
- terrestrial microwave
  - e.g. 45 Mbps channels
- LAN (e.g., Wifi)
  - 11Mbps, 54 Mbps, 200 Mbps
- wide-area (e.g., cellular)
  - 3G cellular: ~ 1 Mbps
- satellite
  - Kbps to 45Mbps channel (or multiple smaller channels)
  - 270 msec end-end delay
  - geosynchronous versus low altitude

34

## The Network Core

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

**Packet-switching = Asynchronous Time Division Multiplexing**



35

## Network Core: Circuit Switching

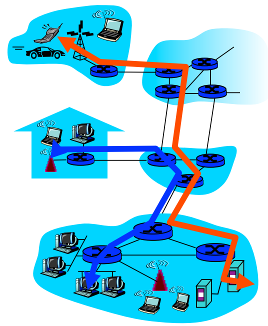
End-end resources reserved for "call"

- link bandwidth, switch capacity
- dedicated resources: no sharing
- circuit-like (guaranteed) performance
- call setup required

network resources (e.g., bandwidth) are divided into "pieces"

- pieces allocated to calls
- resource piece *idle* if not used by owning call (no sharing)

- How?
  - frequency division multiplexing
  - time division multiplexing



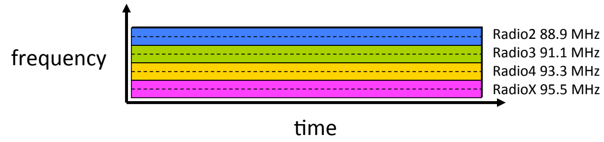
36

## Circuit Switching: FDM and TDM

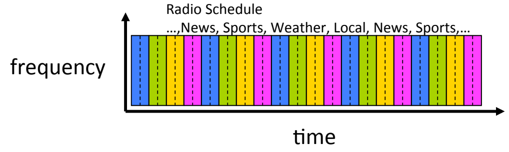
Example:

4 users

Frequency Division Multiplexing




Time Division Multiplexing






## 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
  - 500 msec to establish end-to-end circuit

Let's work it out!



## Network Core: Packet Switching

each end-end data stream divided into *packets (datagrams)*

- user A&B packets *share* network resources
- each packet uses full link bandwidth
- resources used *as needed*

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
  - Node receives complete packet before forwarding

*Fun* new problems/properties

1. How does a program(mer) compute available resources e.g., bandwidth, delay, variation in delay (jitter)?

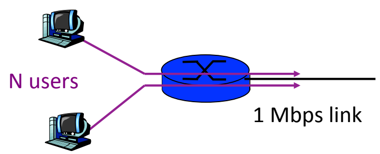
2. Sharing resources (contention-resolution) is now done by protocol (and precedent) – and all that this implies...



## Packet switching versus circuit switching

*Packet switching may (does!) allow more users to use network*

- 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?

## Packet switching versus circuit switching

Q: how did we get value 0.0004?

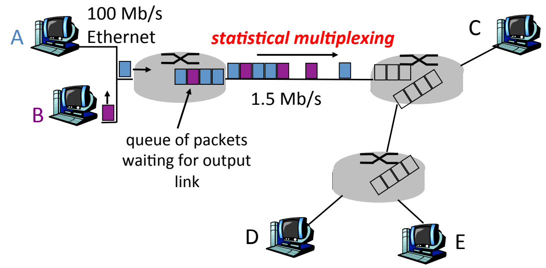
- 1 Mb/s link
- each user:
  - 100 kb/s when "active"
  - active 10% of time

**HINT:** Binomial Distribution

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



## Packet Switching: Statistical Multiplexing

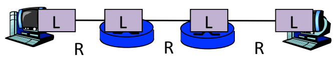



Sequence of A & B packets does not have fixed pattern, bandwidth shared on demand ➡ ***statistical multiplexing***.

(Recall in TDM: each host gets same slot in revolving TDM frame)



## Packet Switching: store-and-forward




- takes L/R seconds to transmit (push out) packet of L bits on to link at R bps
- *store and forward:* entire packet must arrive at each hope before it can be transmitted on next link
- delay = 3L/R (assuming zero propagation delay)

<u>Example:</u>

- L = 7.5 Mbits
- R = 1.5 Mbps
- transmission delay = 15 sec

} more on delay shortly …



## Packet switching versus circuit switching

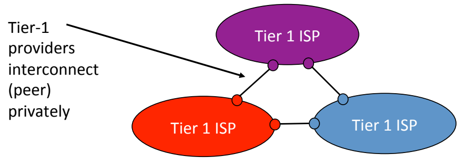
Is packet switching a "slam dunk winner?"

- 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 audio/video apps
  - still a problem (KR: Ch7 PD: Sec 6.5)

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

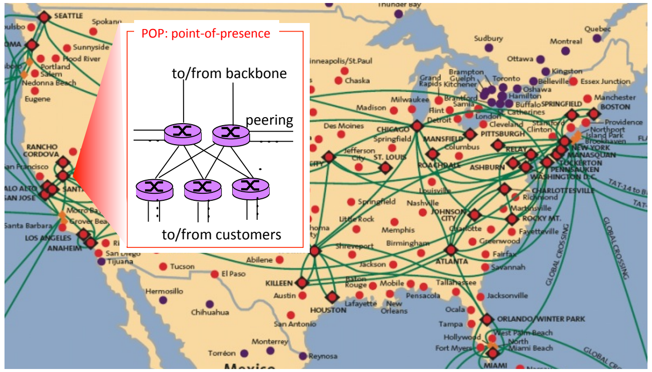
## Internet structure: network of networks

- roughly hierarchical
- at center: "tier-1" ISPs (e.g., Verizon, Sprint, AT&T, Cable and Wireless), national/international coverage
  - treat each other as equals





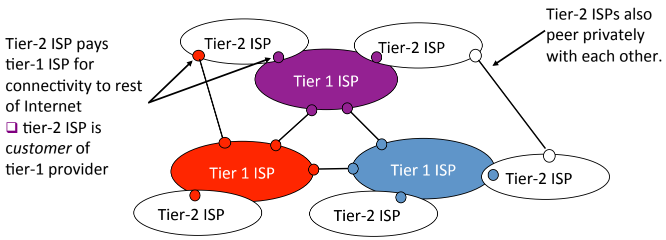

## Tier-1 ISP: e.g., Sprint






## Internet structure: network of networks

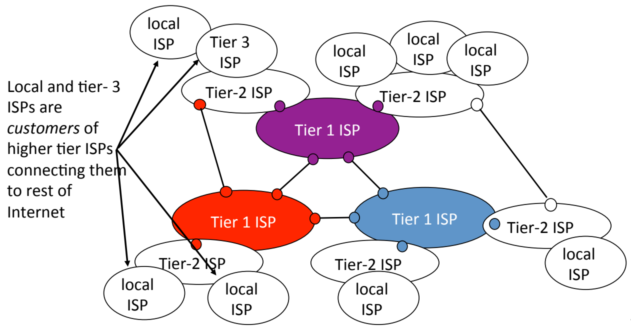
- "Tier-2" ISPs: smaller (often regional) ISPs
  - Connect to one or more tier-1 ISPs, possibly other tier-2 ISPs





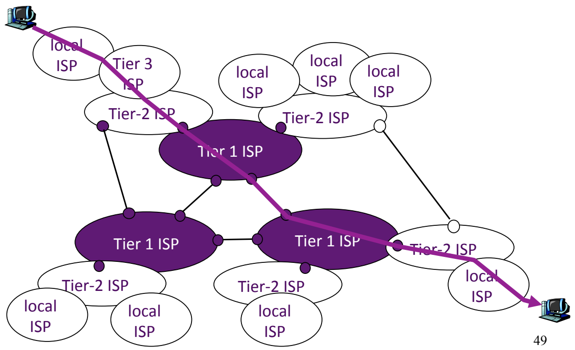

## Internet structure: network of networks

- "Tier-3" ISPs and local ISPs
  - last hop ("access") network (closest to end systems)



## Internet structure: network of networks

- a packet passes through many networks!






## "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
```

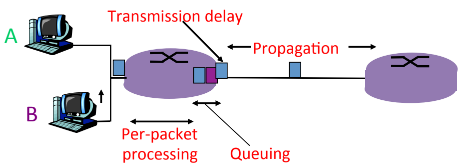
trans-oceanic link (hop 8)

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



## Four sources of packet delay

1. Per-packet processing:
   errors check & lookup output link
   depends on per-packet overhead
2. Queuing delay:
   time waiting for output link
   (depends on link-congestion)
3. Transmission delay:
   packet length / link bandwidth
4. Propagation:
   length of physical link /
   propagation speed in medium
   (e.g. $2x10^8$ m/sec)

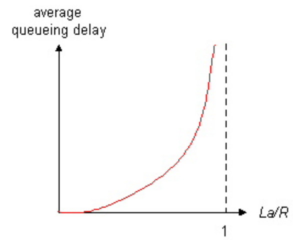





## 2. Queuing delay

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

traffic intensity = La/R



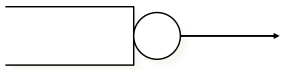

- La/R ~ 0: average queuing delay small
- La/R -> 1: delays become large
- La/R > 1: more "work" arriving than can be serviced, average delay infinite!

## Queuing Delay

Does not happen if
- packets are evenly spaced
- And arrival rate is less than service rate

Smooth Arrivals = No Queuing Delays

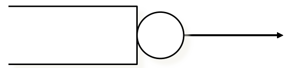
La/R ~ 0: average queuing delay small





## Queuing Delay

- Queuing delay caused by “packet interference”
  - Burstiness of arrival schedule
  - Variations in packet lengths



There is significant queuing delay even though link is underutilized



## Queuing Delay

- Does not happen if packets are evenly spaced
  - And arrival rate is less than service rate
- Queuing delay caused by “packet interference”
  - Burstiness of arrival schedule
  - Variations in packet lengths
- Made worse at high load
  - Less “idle time” to absorb bursts
  - Think about traffic jams in rush hour….

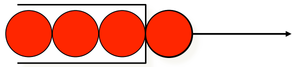


## Jitter

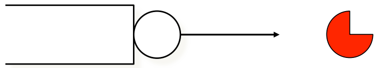
- Difference between minimum and maximal delay
- **Latency plays no role in jitter**
  - Nor does transmission delay for **same sized packets**
- Jitter typically just differences in queuing delay
- Why might an application care about jitter?

## Packet Loss

Packet Loss due to a full queue
    Statistical Multiplexing at High Load *may* lead to packet loss



Packet Loss due to corruption

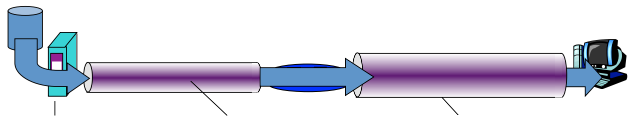


Recall the delay due to per-packet processing;
This processing detect/discards corruption



## Throughput

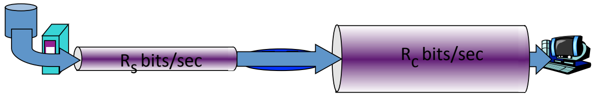
- *throughput:* rate (bits/time unit) at which bits transferred between sender/receiver
  - *instantaneous:* rate at given point in time
  - *average:* rate over longer period of time



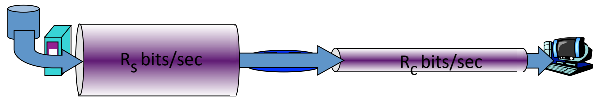



## Throughput (more)

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




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



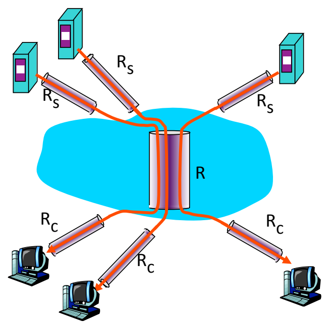

*bottleneck link*

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



## Throughput: Internet scenario

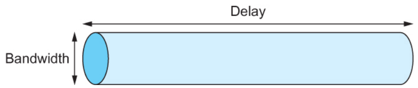
- per-connection end-end throughput: min $(R_c, R_s, R/10)$
- in practice: $R_c$ or $R_s$ is often bottleneck




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

# (Propagation) Delay × Bandwidth
## also called Bandwidth Delay Product

- Consider the channel between entities as a pipe
- Properties:
  - Propagation delay due to length of the pipe, and
  - Bandwidth aka the width of the pipe



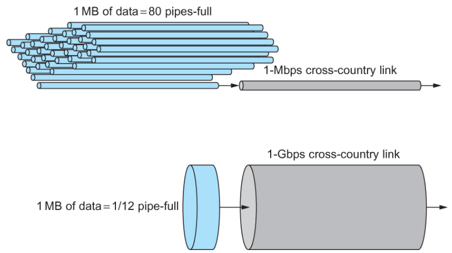

- Delay of 50 ms and bandwidth of 45 Mbps

⇒ $50 \times 10^{-3}$ seconds x $45 \times 10^{6}$ bits/second

⇒ $2.25 \times 10^{6}$ bits = 280 KB data

61

Relationship between bandwidth and link-latency




A MB file would fill the Mbps link 80 times, but only fill the Gbps link 1/12 of one time

62

# Performance

Second order effects
- Image/Audio quality

Other metrics…
- Network efficiency (good-put *versus* throughput)
- User Experience? (World Wide Wait)


- Network connectivity expectation


- Others?

63

# Lessons to take away

- Multiplexing is the key
- Lots of 'definitions'
- Consider the Task not the Technology
- Internet is an example of a packet network
- Why are we waiting? – Delay
- Where did my data go? – Loss
- Bandwidth vs Throughput
- Bandwidth delay product

64