Computer Networking

Lent Term
M/W/F 11:00-12:00
LT1 in Gates Building

Slide Set 1

Andrew W. Moore
Andrew.Moore@cl.cam.ac.uk
2018-2019

Topic 1 Foundation

- Administrivia
- Networks
- Channels
- Multiplexing
- Performance: loss, delay, throughput

Course Administration

Commonly Available Texts

- ❑ Computer Networking: A Top-Down Approach
  Kurose and Ross, 7th edition 2016, Addison-Wesley
  (6th and 5th edition is also commonly available)
- ❑ Computer Networks: A Systems Approach
  Peterson and Davie, 5th edition 2011, Morgan-Kaufman

Other Selected Texts (non-representative)

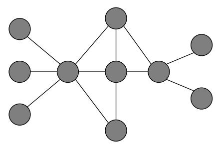
- ❑ Internetworking with TCP/IP, vol. I + II
  Comer & Stevens, Prentice Hall
- ❑ UNIX Network Programming, Vol. I
  Stevens, Fenner & Rudoff, Prentice Hall

Thanks

- Slides are a fusion of material from
  Evangelia Kalyvianaki, Brad Smith, 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, Sylvia Ratnasamy, and Adam Greenhalgh.
- Supervision material is drawn from
  Stephen Kell, Andy Rice, and the fantastic TA teams of 144 and 168
- Finally thanks to the Part 1b students past and Andrew Rice for all the tremendous feedback.

What is a network?

- A system of “links” that interconnect “nodes” in order to move “information” between nodes



- Yes, this is very vague

There are *many* different types of networks

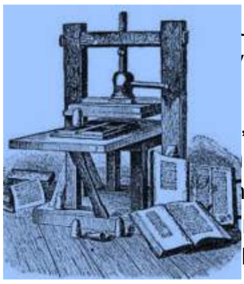
- Internet
- Telephone network
- Transportation networks
- Cellular networks
- Supervisory control and data acquisition networks
- Optical networks
- Sensor networks

We will focus almost exclusively on the Internet

The Internet has transformed everything

- The way we do business
  - E-commerce, advertising, cloud-computing
- The way we have relationships
  - Facebook friends, E-mail, IM, virtual worlds
- The way we learn
  - Wikipedia, search engines
- The way we govern and view law
  - E-voting, censorship, copyright, cyber-attacks

The Internet transforms everything



usin
ertis
rela
, E-m
eng
rn a
hip,
ks

Taking the dissemination of information to the next level

The Internet is big business

- Many large and influential networking companies
  - Huawei, Broadcom, AT&T, Verizon, Akamai, Cisco, …
  - $132B+ industry (carrier and enterprise alone)
- Networking central to most technology companies
  - Apple, Google, Facebook, Intel, Amazon, VMware, …

Internet research has impact

- The Internet started as a research experiment!
- 5 of 10 most cited authors work in networking
- *Many* successful companies have emerged from networking research(ers)

But why is the Internet *interesting*?

"What's your formal model for the Internet?" -- *theorists*

"Aren't you just writing software for networks" – *hackers*

"You don't have performance benchmarks???" – *hardware folks*

"Isn't it just another network?" – *old timers at AT&T*

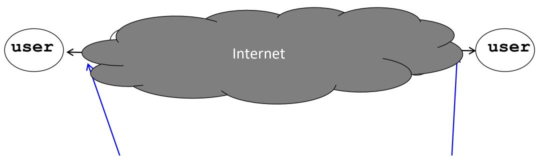
"What's with all these TLA protocols?" – *all*

"But the Internet seems to be working…" – *my mother*

A few defining characteristics of the Internet

A federated system

- The Internet ties together different networks
  - >18,000 ISP networks



Tied together by IP -- the "Internet Protocol" : a single common interface between users and the network and between networks

A federated system

- The Internet ties together different networks
  - >18,000 ISP networks

- A single, common interface is great for interoperability…
- …but tricky for business

- Why does this matter?
  - ease of interoperability is the Internet’s most important goal
  - practical realities of incentives, economics and real-world trust drive topology, route selection and service evolution

Tremendous scale

- 4.15 Billion users (55.1% of world population)
- 1.3+ Trillion unique URLs from 1.8 Billion web servers
- 269 Billion emails sent per day
- 2.5 Billion smartphones
- 846 Million Tweets a day
- 65 Billion WhatsApp messages per day
- 5 Billion YouTube videos watched per day
- 400 hours of Youtube video added per minute
- Switches that move 300+ Terabits/second
- Network links that carry 67.5 Terabits/second

Tremendous scale

- 4.15 Billion users (55.1% of world population)
- 1.3+ Trillion unique URLs from 1.8 Billio
  servers
- 269 Billion emails sent
- 2.5 Billion sm
- 846 M… a day
- …WhatsApp messages per day
- … Billion YouTube videos watched per day
- 400 hours of Youtube video added per minute
- Switches that move 300+ Terabits/second
- Network links that carry 67.5 Terabits/second

“Internet Scale” refers to such systems

Enormous diversity and dynamic range

- Communication latency: microseconds to seconds (10^6)
- Bandwidth: 1Kbits/second to 400 Gigabits/second (10^7)
- Packet loss: 0 – 90%

- Technology: optical, wireless, satellite, copper

- Endpoint devices: from sensors and cell phones to datacenters and supercomputers
- Applications: social networking, file transfer, skype, live TV, gaming, remote medicine, backup, IM
- Users: the governing, governed, operators, malicious, naïve, savvy, embarrassed, paranoid, addicted, cheap …

Constant Evolution

1970s:
- 56kilobits/second “backbone” links
- <100 computers, a handful of sites in the US (and one UK)
- Telnet and file transfer are the “killer” applications

Today
- 400+Gigabits/second backbone links
- 40B+ devices, all over the globe
- *20M* Facebook (new?) app installs per day

Asynchronous Operation

- Fundamental constraint: **speed of light**

- Consider:
  - How many cycles does your 3GHz CPU in Cambridge execute before it can possibly get a response from a message it sends to a server in Palo Alto?
    - Cambridge to Palo Alto: 8,609 km
    - Traveling at 300,000 km/s: 28.70 milliseconds
    - Then back to Cambridge: 2 x 28.70 = 57.39 milliseconds
    - 3,000,000,000 cycles/sec * 0.05739 = 172,179,999 cycles!

- Thus, communication feedback is always *dated*

Prone to Failure

- To send a message, **all** components along a path must function correctly
  - software, wireless access point, firewall, links, network interface cards, switches,…
  - Including human operators
- Consider: 50 components, that work correctly 99% of time → 39.5% chance communication will fail
- Plus, recall
  - scale → lots of components
  - asynchrony → takes a long time to hear (bad) news
  - federation (**inter**net) → hard to identify fault or assign blame

An Engineered System

- Constrained by what technology is practical
  - Link bandwidths
  - Switch port counts
  - Bit error rates
  - Cost
  - …

Recap: The Internet is…

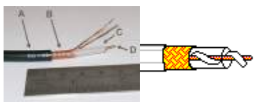
- A complex federation
- Of enormous scale
- Dynamic range
- Diversity
- Constantly evolving
- Asynchronous in operation
- Failure prone
- Constrained by what's practical to engineer
- Too complex for theoretical models
- "Working code" doesn't mean much
- Performance benchmarks are too narrow

Example Physical Channels

these example physical channels are 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)



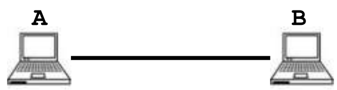
Fiber optic cable:
- high-speed operation
- point-to-point transmission
- (10's-100's Gps)
- low error rate
- immune to electromagnetic noise

More Physical media: Radio

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

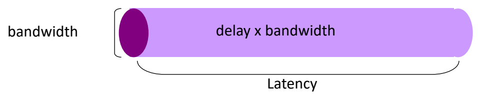
Radio link types:
- terrestrial microwave
  - e.g. 45 Mbps channels
- LAN (e.g., Wifi)
  - 11Mbps, 54 Mbps, 200 Mbps
- wide-area (e.g., cellular)
  - 4G cellular: ~ 4 Mbps
- satellite
  - Kbps to 45Mbps channel (or multiple smaller channels)
  - 270 msec end-end delay
  - geosynchronous versus low altitude

Nodes and Links



Channels = Links
Peer entities = Nodes

Properties of Links (Channels)



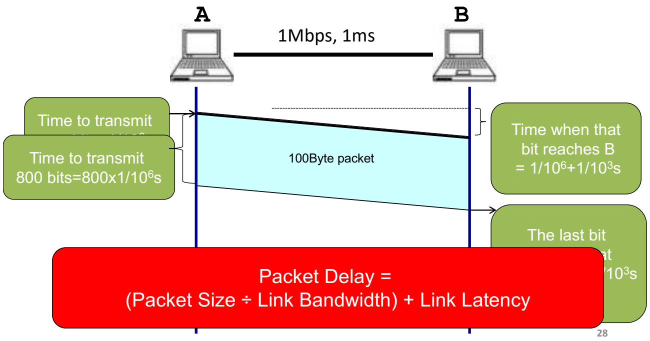
- Bandwidth (capacity): “width” of the links
  - number of bits sent (or received) per unit time (bits/sec or bps)
- Latency (delay): “length” of the link
  - propagation time for data to travel along the link (seconds)
- Bandwidth-Delay Product (BDP): “volume” of the link
  - amount of data that can be “in flight” at any time
  - propagation delay × bits/time = total bits in link

Examples of Bandwidth-Delay

- Same city over a slow link:
  - BW~10Mbps
  - Latency~0.1msec
  - BDP ~ 10^6bits ~ 125KBytes
- Cross-country over fast link:
  - BW~10Gbps
  - Latency~10msec
  - BDP ~ 10^8bits ~ 12.5MBytes

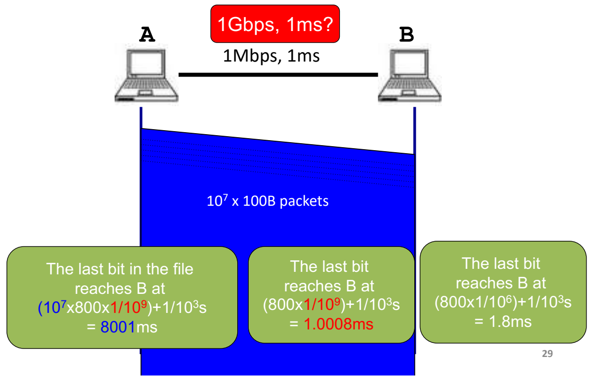
Packet Delay

Sending a 100B packet from A to B?



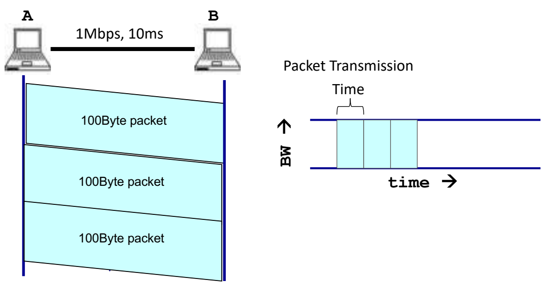
1GB file in 100B packets ay

Sending a 100B packet from A to B?



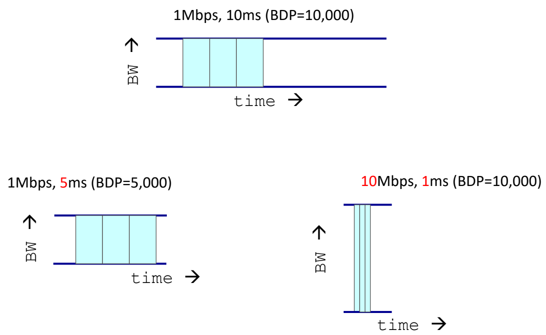
Packet Delay: The “pipe” view

Sending 100B packets from A to B?



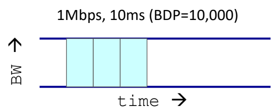
Packet Delay: The “pipe” view

Sending 100B packets from A to B?

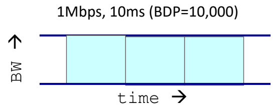


Packet Delay: The "pipe" view

Sending 100B packets from A to B?



What if we used *200Byte packets??*



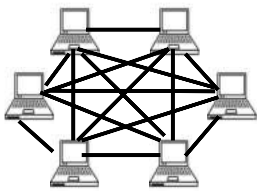
32

Recall Nodes and Links

33

What if we have more nodes?

One link for every node?

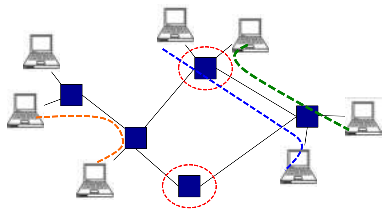


Need a <u>scalable</u> way to interconnect nodes

34

Solution: A switched network

Nodes <u>share</u> network link resources



How is this sharing implemented?

35

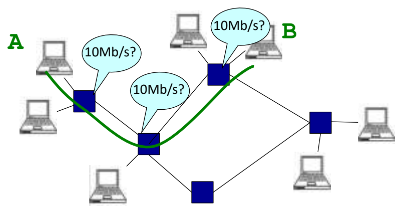
Two forms of switched networks

- Circuit switching (used in the *POTS*: Plain Old Telephone system)
- Packet switching (used in the Internet)

36

Circuit switching

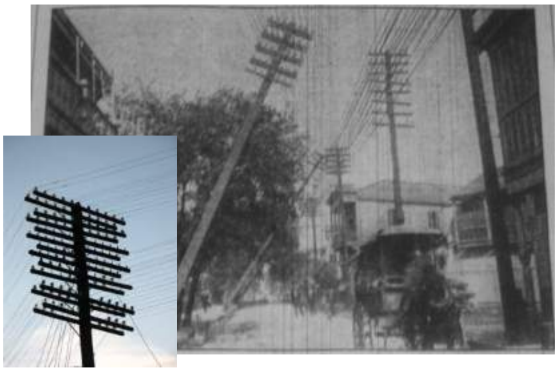
Idea: source reserves network capacity along a path



(1) Node A sends a reservation request
(2) Interior switches establish a connection -- i.e., "circuit"
(3) A starts sending data
(4) A sends a "teardown circuit" message

37

Old Time Multiplexing

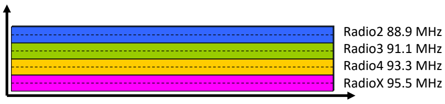


Circuit Switching: FDM and TDM

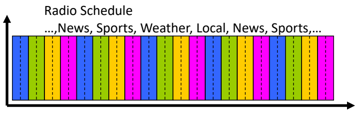
Example:

4 users

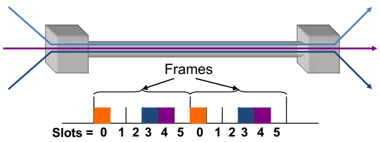
Frequency Division Multiplexing



Time Division Multiplexing

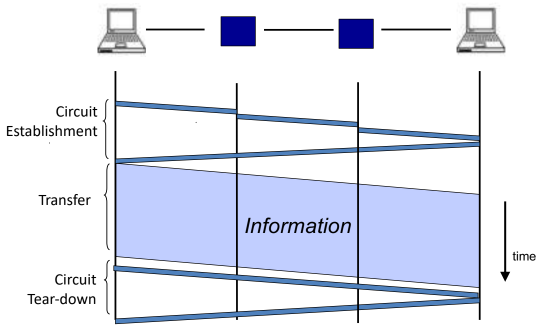


Time-Division Multiplexing/Demultiplexing



- Time divided into frames; frames into slots
- Relative slot position inside a frame determines to which conversation data belongs
  - e.g., slot 0 belongs to orange conversation
- Slots are reserved (released) during circuit setup (teardown)
- If a conversation does not use its circuit **capacity is lost!**

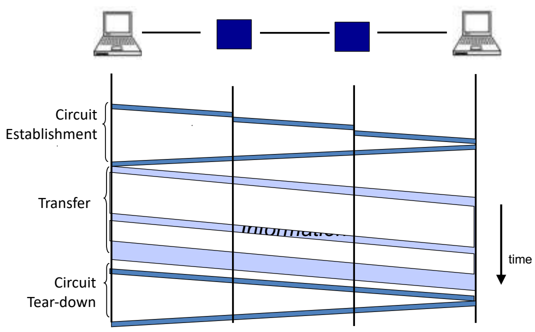
Timing in Circuit Switching



Circuit switching: pros and cons

- Pros
  - guaranteed performance
  - fast transfer (once circuit is established)
- Cons

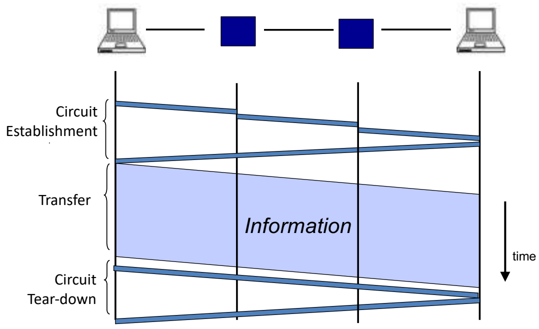
Timing in Circuit Switching



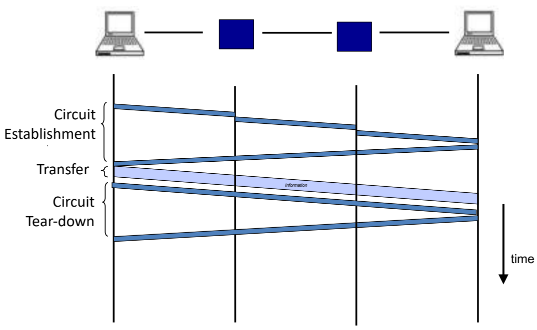
Circuit switching: pros and cons

- Pros
  - guaranteed performance
  - fast transfer (once circuit is established)
- Cons
  - **wastes bandwidth if traffic is "bursty"**

Timing in Circuit Switching



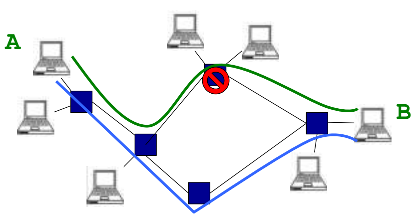
Timing in Circuit Switching



Circuit switching: pros and cons

- Pros
  - guaranteed performance
  - fast transfers (once circuit is established)
- Cons
  - wastes bandwidth if traffic is "bursty"
  - **connection setup time is overhead**

Circuit switching



Circuit switching doesn't "route around failure"

Circuit switching: pros and cons

- Pros
  - guaranteed performance
  - fast transfers (once circuit is established)
- Cons
  - wastes bandwidth if traffic is "bursty"
  - connection setup time is overhead
  - **recovery from failure is slow**

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

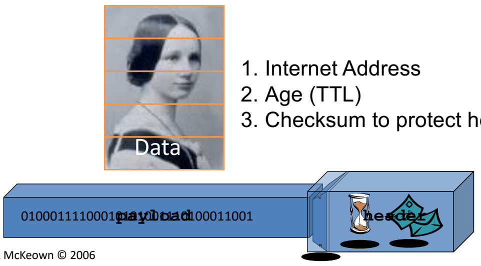
Let's work it out!

Two forms of switched networks

- Circuit switching (e.g., telephone network)
- Packet switching (e.g., Internet)

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

After Nick McKeown © 2006

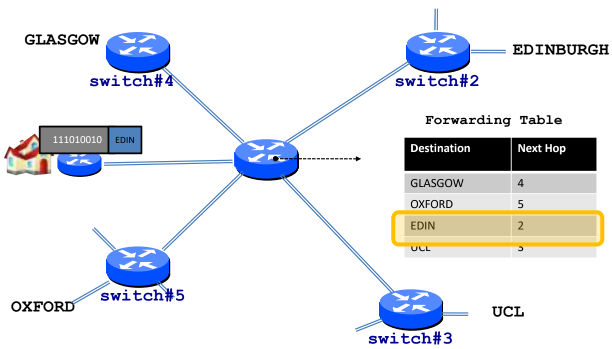
Packet Switching

- 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)

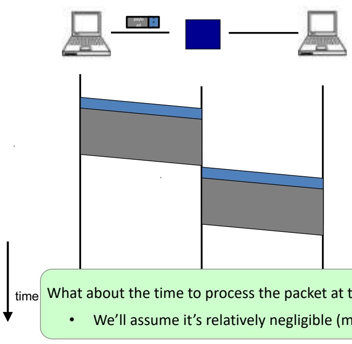
Packet Switching

- 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



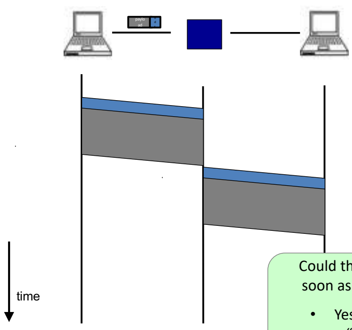
Timing in Packet Switching



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

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

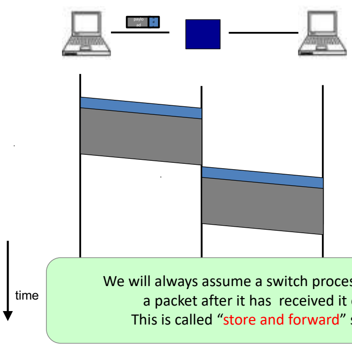
Timing in Packet Switching



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

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

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

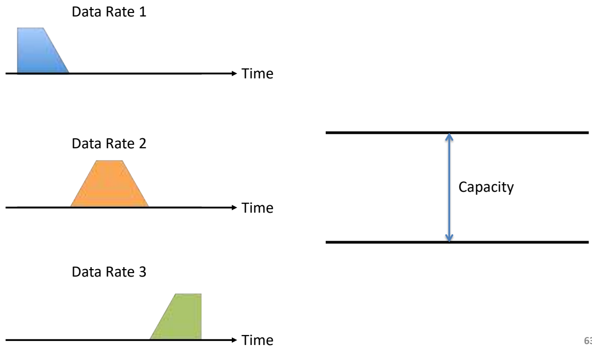
- Data is sent as chunks of formatted bits (Packets)
- Packets consist of a "header" and "payload"
- Switches "forward" packets based on their headers
- Each packet travels independently
- No link resources are reserved in advance. Instead packet switching leverages statistical multiplexing (stat muxing)

Multiplexing

Sharing makes things efficient (cost less)

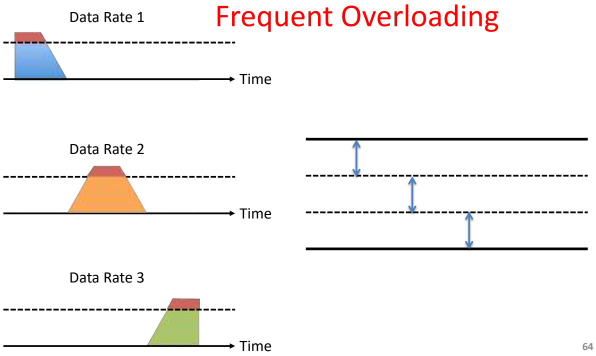
- One airplane/train for 100's of people
- One telephone for many calls
- One lecture theatre for many classes
- One computer for many tasks
- One network for many computers
- One datacenter many applications

Three Flows with Bursty Traffic

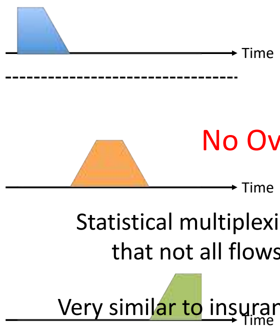


When Each Flow Gets 1/3rd of Capacity

Frequent Overloading



When Flows Share Total Capacity

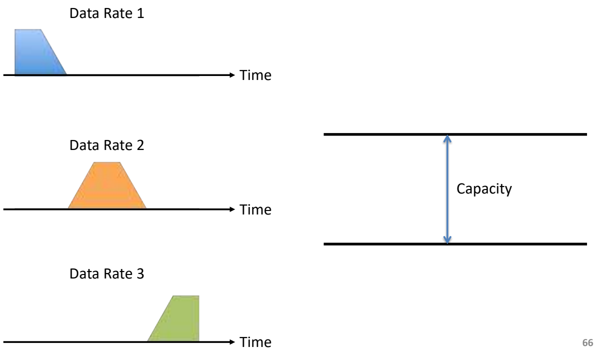


No Overloading

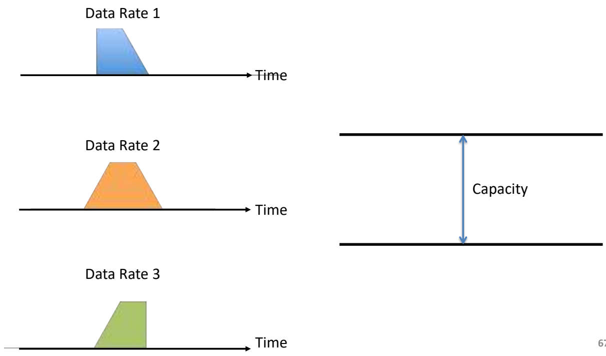
Statistical multiplexing relies on the assumption that not all flows burst at the same time.

Very similar to insurance, and has same failure case

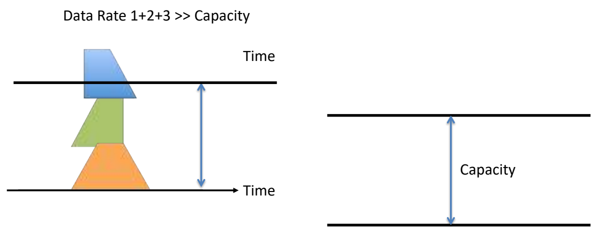
Three Flows with Bursty Traffic



Three Flows with Bursty Traffic

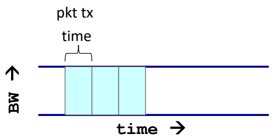


Three Flows with Bursty Traffic

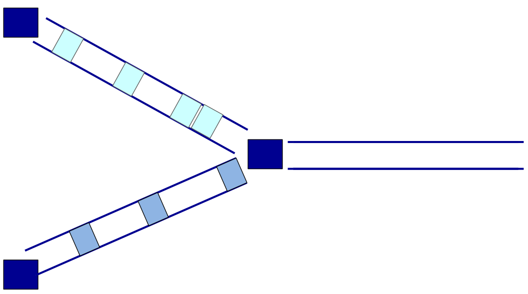


What do we do under overload?

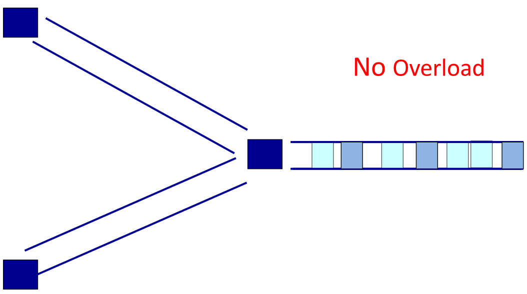
Statistical multiplexing: pipe view



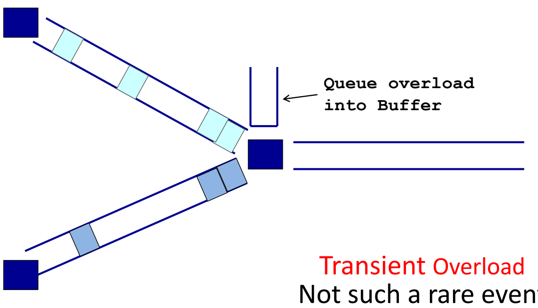
Statistical multiplexing: pipe view



Statistical multiplexing: pipe view

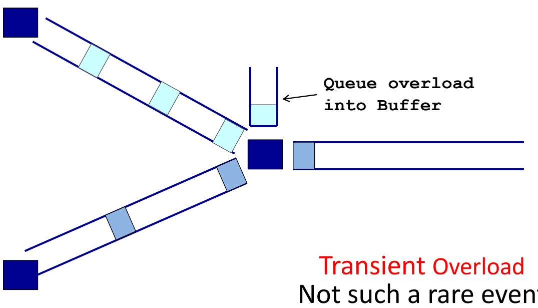


Statistical multiplexing: pipe view



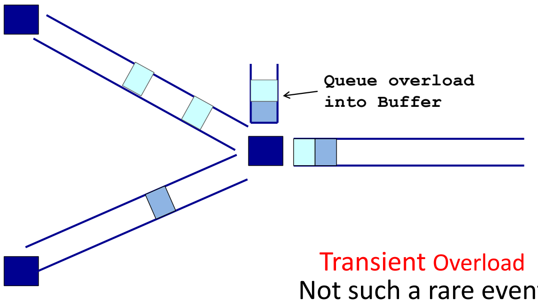
Transient Overload
Not such a rare event

Statistical multiplexing: pipe view



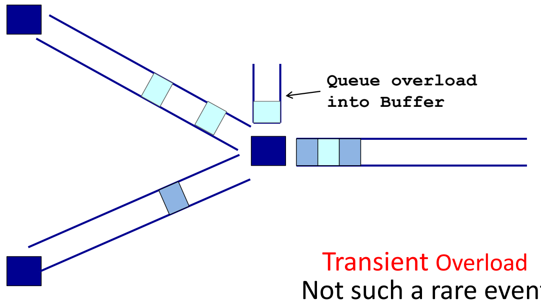
Transient Overload
Not such a rare event

Statistical multiplexing: pipe view



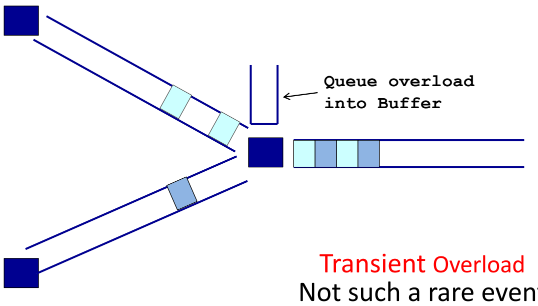
Transient Overload
Not such a rare event

Statistical multiplexing: pipe view



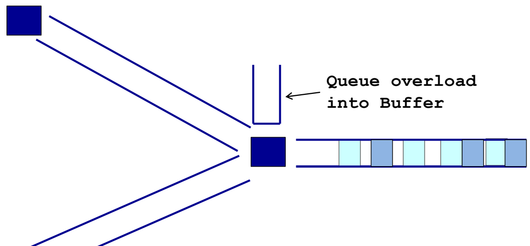
Transient Overload
Not such a rare event

Statistical multiplexing: pipe view



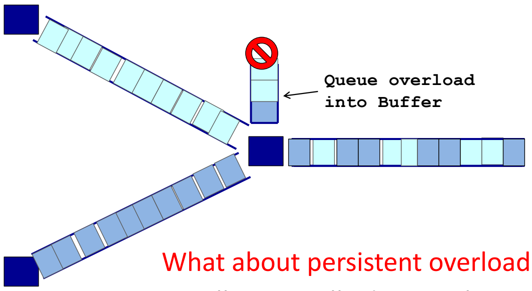
Transient Overload
Not such a rare event

Statistical multiplexing: pipe view



Buffer absorbs transient bursts

Statistical multiplexing: pipe view



What about persistent overload?
Will eventually drop packets

Queues introduce queuing delays

- Recall,

  packet delay = transmission delay + propagation delay (*)

- With queues (statistical multiplexing)

  packet delay = transmission delay + propagation delay + queuing delay (*)

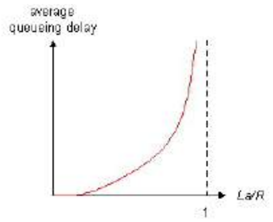
- Queuing delay caused by “packet interference”

- Made worse at high load
  - less “idle time” to absorb bursts
  - think about traffic jams at rush hour
    or rail network failure

(* plus per-hop *processing* delay that we define as negligible)

Queuing delay

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

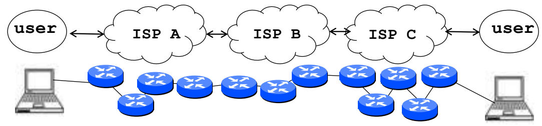


traffic intensity = La/R

- 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 – or data is lost (*dropped*).

Recall the Internet *federation*

- The Internet ties together different networks
  – >18,000 ISP networks



We can see (hints) of the nodes and links using traceroute…

"Real" Internet delays and routes

traceroute: rio.cl.cam.ac.uk to munnari.oz.au

(tracepath on winows is similar)

Three delay measurements from rio.cl.cam.ac.uk to gatwick.net.cl.cam.ac.uk

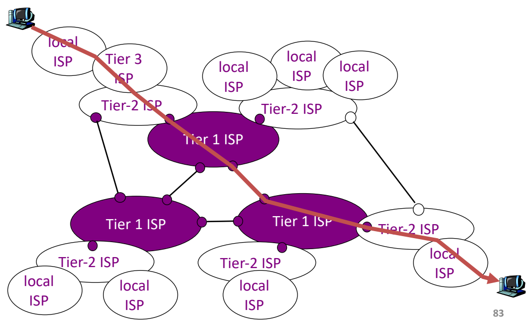
trans-continent link

```
traceroute munnari.oz.au
traceroute to munnari.oz.au (202.29.151.3), 30 hops max, 60 byte packets
 1  gatwick.net.cl.cam.ac.uk (128.232.32.2)  0.416 ms  0.384 ms  0.427 ms
 2  cl-sby.route-nwest.net.cam.ac.uk (193.60.89.9)  0.393 ms  0.440 ms  0.494 ms
 3  route-nwest.route-mill.net.cam.ac.uk (192.84.5.137)  0.407 ms  0.448 ms  0.501 ms
 4  route-mill.route-enet.net.cam.ac.uk (192.84.5.94)  1.006 ms  1.091 ms  1.163 ms
 5  xe-11-3-0.camb-rbr1.eastern.ja.net (146.97.130.1)  0.300 ms  0.313 ms  0.350 ms
 6  ae24.lowdss-sbr1.ja.net (146.97.37.185)  2.679 ms  2.664 ms  2.712 ms
 7  ae28.londhx-sbr1.ja.net (146.97.33.17)  5.955 ms  5.953 ms  5.901 ms
 8  janet.mx1.lon.uk.geant.net (62.40.124.197)  6.059 ms  6.066 ms  6.052 ms
 9  ae0.mx1.par.fr.geant.net (62.40.98.77)  11.742 ms  11.779 ms  11.724 ms
10  ae1.mx1.mad.es.geant.net (62.40.98.64)  27.751 ms  27.734 ms  27.704 ms
11  mb-so-02-v4.bb.tein3.net (202.179.249.117)  138.296 ms  138.314 ms  138.282 ms
12  sg-so-04-v4.bb.tein3.net (202.179.249.53)  196.303 ms  196.293 ms  196.264 ms
13  th-pr-v4.bb.tein3.net (202.179.249.66)  225.153 ms  225.178 ms  225.196 ms
14  pyt-thairen-to-02-bdr-pyt.uni.net.th (202.29.12.10)  225.163 ms  223.343 ms  223.363 ms
15  202.28.227.126 (202.28.227.126)  241.038 ms  240.941 ms  240.834 ms
16  202.28.221.46 (202.28.221.46)  287.252 ms  287.306 ms  287.282 ms
17  * * *
18  * * *
19  * * *
20  coe-gw.psu.ac.th (202.29.149.70)  241.681 ms  241.715 ms  241.680 ms
21  munnari.OZ.AU (202.29.151.3)  241.610 ms  241.636 ms  241.537 ms
```

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

Internet structure: network of networks

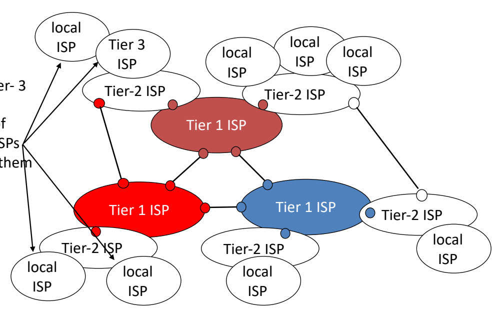
- a packet passes through many networks!



Internet structure: network of networks

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

Local and tier- 3 ISPs are *customers* of higher tier ISPs connecting them to rest of Internet

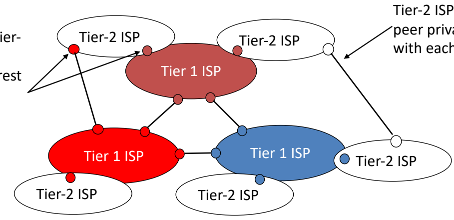


Internet structure: network of networks

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

Tier-2 ISP pays tier-1 ISP for connectivity to rest of Internet

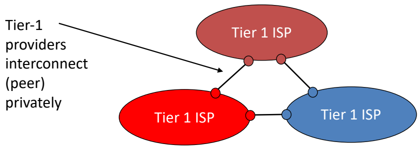
- tier-2 ISP is *customer* of tier-1 provider



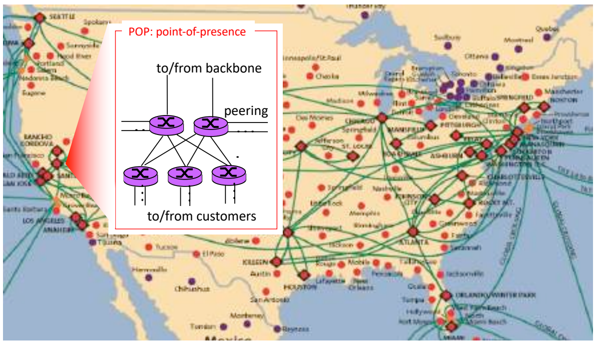
Tier-2 ISPs also peer privately with each other.

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



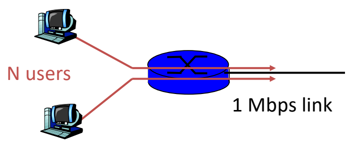
Packet Switching

- 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
  - allows efficient use of resources
  - but introduces queues and queuing delays

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

$$\Pr(K = k) = \binom{n}{k} p^k (1-p)^{n-k}$$

$$\Pr(K \le k) = 1 - \sum_{n=0}^{\lfloor k \rfloor} \binom{n}{k} p^k (1-p)^{n-k}$$

$$\Pr(K \le k) = 1 - \sum_{n=1}^{9} \binom{35}{k} (0.1)^k (0.9)^{35-k}$$

$$\Pr(K \le k) \approx 0.0004$$

Circuit switching: pros and cons

- Pros
  - guaranteed performance
  - fast transfers (once circuit is established)
- Cons
  - wastes bandwidth if traffic is "bursty"
  - connection setup adds delay
  - recovery from failure is slow

Packet switching: pros and cons

- Cons
  - no guaranteed performance
  - header overhead per packet
  - queues and queuing delays

- Pros
  - efficient use of bandwidth (stat. muxing)
  - no overhead due to connection setup
  - resilient -- can `route around trouble'

Summary

- A sense of how the basic `plumbing' works
  - links and switches
  - packet delays= transmission + propagation + queuing + (negligible) per-switch processing
  - statistical multiplexing and queues
  - circuit vs. packet switching

Topic 2 – Architecture and Philosophy

- Abstraction
- Layering
- Layers and Communications
- Entities and Peers
- What is a protocol?
- Protocol Standardization
- The architects process
  - How to break system into modules
  - Where modules are implemented
  - Where is state stored
- Internet Philosophy and Tensions

Abstraction Concept

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 built ("layered") on top of other paths

Computer System Modularity

Partition system into modules & abstractions:
- Well-defined interfaces give flexibility
  - ***Hides*** implementation - can be freely changed
  - Extend functionality of system by adding new modules
- E.g., libraries encapsulating set of functionality
- E.g., programming language + compiler abstracts away how the particular CPU works …

Computer System Modularity (cnt' d)

- Well-defined interfaces hide information
  - Isolate assumptions
  - Present high-level abstractions
- **But can impair performance!**
- Ease of implementation vs worse performance