If the Internet is the answer, then what was the question?

EE122 Fall 2012

Scott Shenker

http://inst.eecs.berkeley.edu/~ee122/

Materials with thanks to Jennifer Rexford, Ion Stoica, Vern Paxson and other colleagues at Princeton and UC Berkeley

1

Administrivia

- Participation: administrivia questions don't count
  - And don't send your email during class (duh!)
  - Math: 340 students/ 27 lectures ~ 12.5 comments/lecture
- Sections start this week
  - If you asked about a switch, should have heard from me
- Instructional account forms sent by email
  - Should have them by now
- Midterm clash:
  - Is Oct 11^{th} ok?

2

Outline for today's class

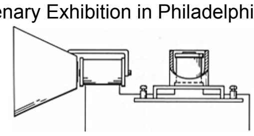
- The telephone network
- Taxonomy of networks
- Some basics of packet switching
- Statistical multiplexing
  - This is something you should know deep in your soul…

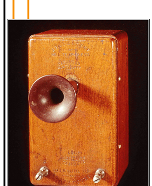
3

Review of Telephone Network

Telephones

- Alexander Graham Bell
  - 1876: Demonstrates the telephone at US Centenary Exhibition in Philadelphia





Telephone was an app, not a network!

- The big technological breakthrough was to turn voice into electrical signals and vice versa.
  - Great achievement
  - One of the nastiest patent battles in history
- The demonstration of this new device involved two phones connected by a single dedicated wire.

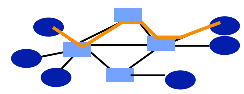
6

What about the phone "network"?

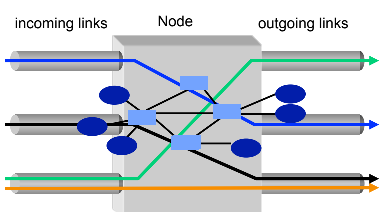
- You can't have a dedicated wire between every two telephones
  - Doesn't scale
  - Most wires will go unused….
- You need a "shared network" of wires
  - Much like the highway is shared by cars going to different destinations
- The telephone network grew into the first large-scale electronic network

Telephone network uses circuit switching

- Establish: source creates circuit to destination
  - Nodes along the path store connection info
  - And reserve resources for the connection
  - If circuit not available: "Busy signal"
- Transfer: source sends data over the circuit
  - No destination address in msg, since nodes know path
  - Continual stream of data
- Teardown: source tears down circuit when done

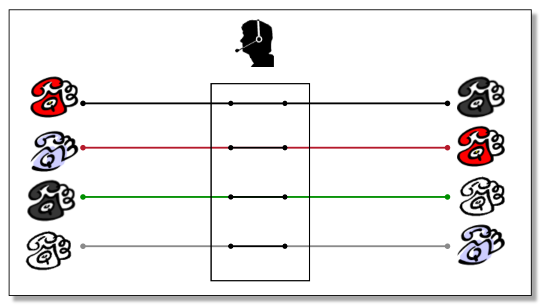


The switch in "circuit switching"

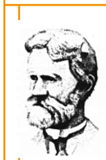


How does the node connect the incoming link to the outgoing link?

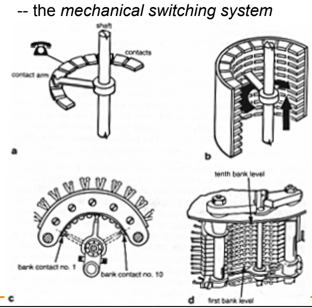
Circuit Switching With Human Operator



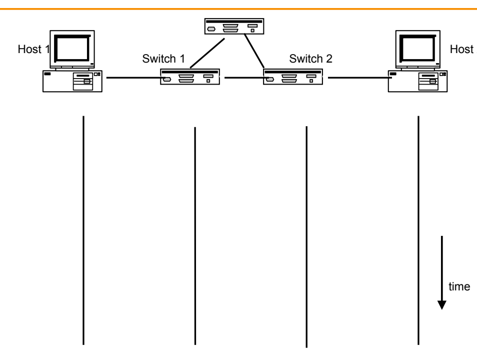
"Modern" switches



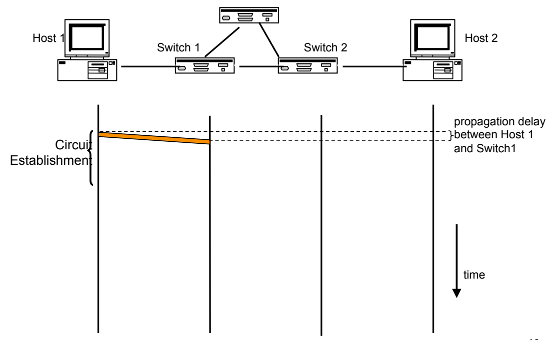
- Almon Brown Strowger (1839 - 1902)
  - 1889: Invents the "girl-less, cuss-less" telephone system -- the *mechanical switching system*



Timing in Circuit Switching

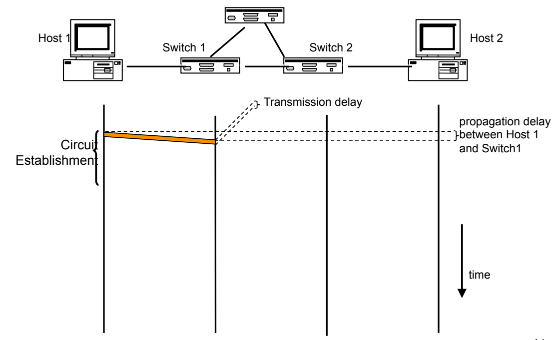


Timing in Circuit Switching



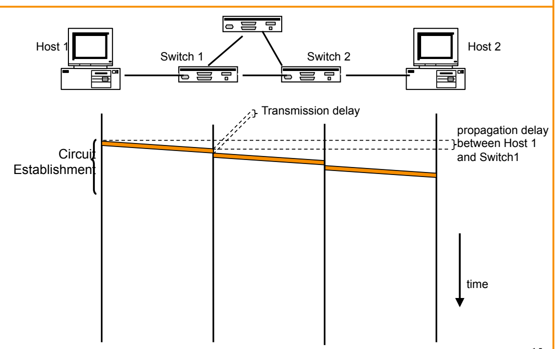
13

Timing in Circuit Switching



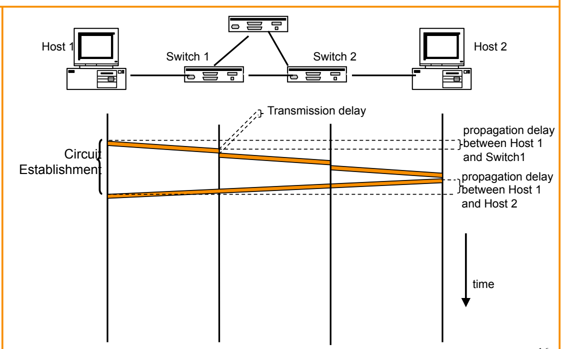
14

Timing in Circuit Switching



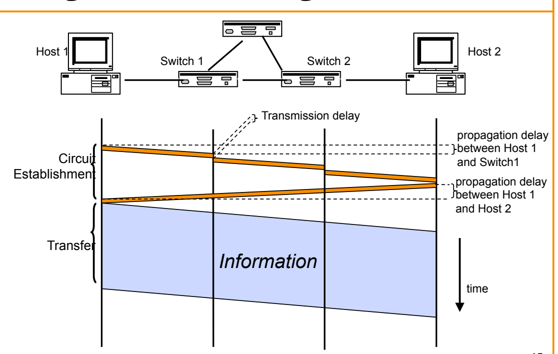
15

Timing in Circuit Switching



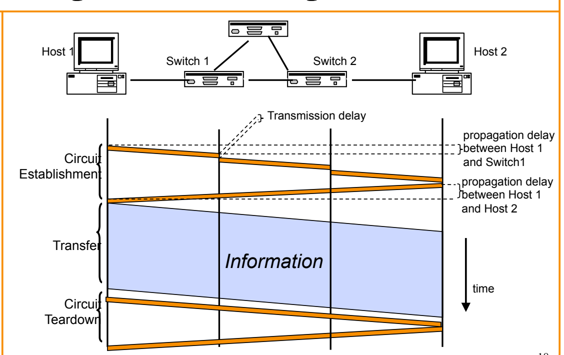
16

Timing in Circuit Switching



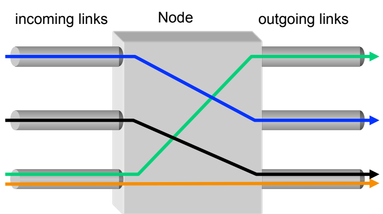
17

Timing in Circuit Switching



18

Sharing a link

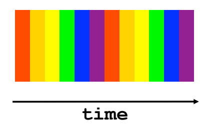


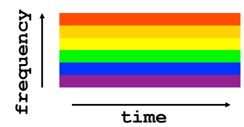
How do the black and orange circuits share the outgoing link?

19

Circuit Switching: *Multiplexing* a Link

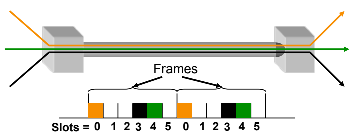
- Time-division
  - Each circuit allocated certain time slots
- Frequency-division
  - Each circuit allocated certain frequencies





20

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
- Requires synchronization between sender and receiver
- Need to dynamically bind a slot to a conversation
- If a conversation does not use its circuit **capacity is lost!**

21

Strengths of phone system

- Predictable performance
  - Known delays
  - No drops
- Easy to control
  - Centralized management of how calls are routed
- Easy to reason about
- Supports a crucial service

What about weaknesses?

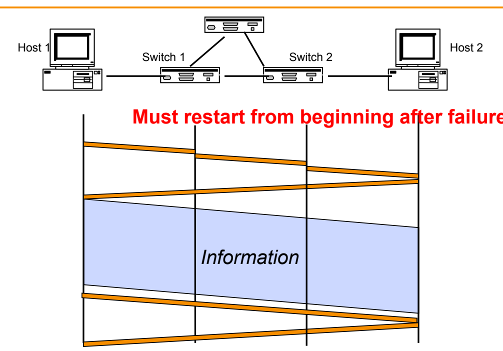
22

Weakness #1: Not resilient to failure

- Any failure along the path prevents transmission
- Entire transmission has to be restarted
  - "All or nothing" delivery model

23

All-or-Nothing Delivery



24

Weakness #2: Wastes bandwidth

- Consider a network application with:
  - Peak bandwidth P
  - Average bandwidth A
- How much does the network have to reserve for the application to work?
  - The peak bandwidth
- What is the resulting level of utilization?
  - Ratio of A/P

25

Smooth vs Bursty Applications

- Some applications have relatively small P/A ratios
  - Voice might have a ratio of 3:1 or so
- Data applications tend to be rather bursty
  - Ratios of 100 or greater are common
- Circuit switching too inefficient for bursty apps
- Generally:
  - Don't care about factors of two in performance
  - But when it gets to several orders of magnitude….

26

Statistical Multiplexing

- Will delve into this in more detail later
- But this is what drives the use of a shared network
- And it is how we could avoid wasting bandwidth

27

Weakness #3: Designed Tied to App

- Design revolves around the requirements of voice
- Not general feature of circuit switching
  - But definitely part of the telephone network design
  - Switches are where functionality was implemented

28

Weakness #4: Setup Time

- Every connection requires round-trip time to set up
  - Slows down short transfers
- In actuality, may not be a big issue
  - TCP requires round-trip time for handshake
  - No one seems to mind….
- This was a big issue in the ATM vs IP battle
  - But I think it is overemphasized as a key factor

29

How to overcome these weaknesses?

- There were two independent threads that led to a different networking paradigm….

30

What if we wanted a resilient network?

- How would we design it?
- This is the question **Paul Baran** asked….

31

Paul Baran

- Baran investigated survivable networks for USAF
  - Network should withstand almost any degree of destruction to individual components without loss of end-to-end communications.
- "On Distributed Communications" (1964)
  - Distributed control
  - Message blocks (packets)
  - Store-and-forward delivery

32

What about a less wasteful network?

- How would we design it?
- This is the question **Len Kleinrock** asked…..
  - Analyzed packet switching and statistical multiplexing

33

Returning to title of lecture

- If the Internet is the answer, then what was the question?
- There were two questions:
  - How can we build a more reliable network?
  - How can we build a more efficient network?
- Before considering nature of Internet, let's consider the broader design space for networks
  - Term "network" already implies we are sharing a communications infrastructure (i.e. not dedicated links)

34

Taxonomy of Networks

Taxonomy of Communication Networks

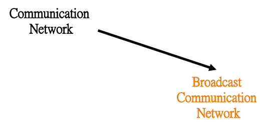
- Communication networks can be classified based on the way in which the ***nodes*** exchange information:

Communication
Network

36

Taxonomy of Communication Networks

- Communication networks can be classified based on the way in which the ***nodes*** exchange information:

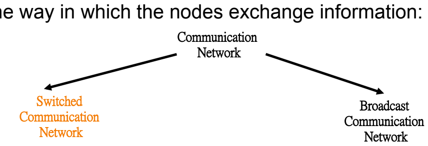


Broadcast Communication Networks

- Information transmitted by any node is received by every other node in the network
  - Usually only in LANs (*Local Area* Networks)
    - E.g., WiFi, Ethernet (classical, but not current)
    - E.g., lecture!
- What problems does this raise?
- Problem #1: limited range
- Problem #2: coordinating access to the shared communication medium
  - *Multiple Access Problem*
- Problem #3: privacy of communication

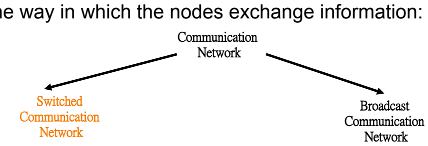
Taxonomy of Communication Networks

- Communication networks can be classified based on the way in which the nodes exchange information:



Taxonomy of Communication Networks

- Communication networks can be classified based on the way in which the nodes exchange information:

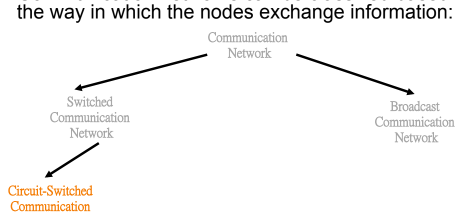


The term “switched” means that communication is directed to specific destinations

The question is how that “switching” is done

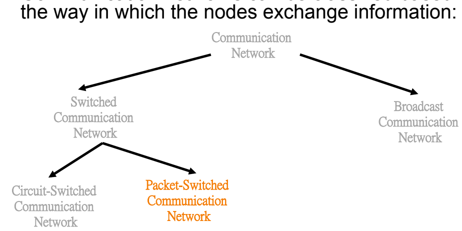
Taxonomy of Communication Networks

- Communication networks can be classified based on the way in which the nodes exchange information:



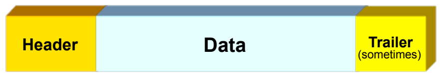
Taxonomy of Communication Networks

- Communication networks can be classified based on the way in which the nodes exchange information:



Packet Switching

- Data sent as chunks of formatted bit-sequences (**Packets**)
- Packets have following structure:

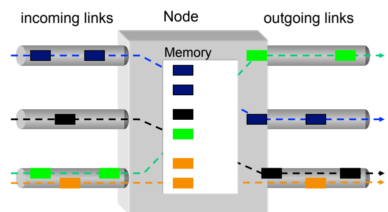


  - Header and Trailer carry control information (e.g., destination address, checksum)
- Each packet traverses the network from node to node along some path (**Routing**) based on header info.
- Usually, once a node receives the entire packet, it stores it (hopefully briefly) and then forwards it to the next node (**Store-and-Forward Networks**)

43

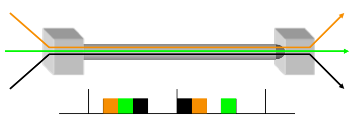
Packet Switching

- Node in a packet switching network



44

Packet Switching: Multiplexing/Demultiplexing

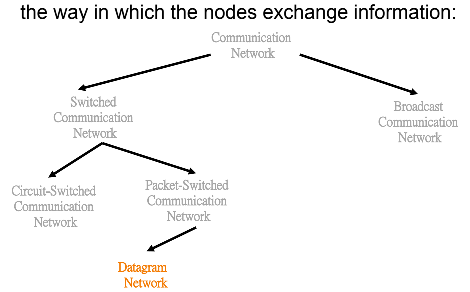


- How to tell packets apart?
  - Use meta-data (header) to describe data
- No reserved resources; dynamic sharing
  - Single flow can use *the entire link capacity* if it is alone
  - This leads to increased efficiency

45

Taxonomy of Communication Networks

- Communication networks can be classified based on the way in which the nodes exchange information:



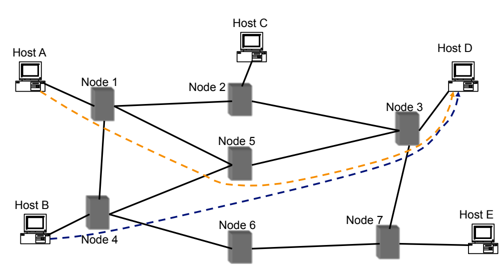
46

Datagram Packet Switching

- Each packet is independently switched
  - Each packet header contains full destination address
  - Routers/switches make independent routing decisions
- No resources are pre-allocated (reserved) in advance
- Leverages "statistical multiplexing"
  - Gambling that packets from different conversations won't all arrive at the same time, so we don't need enough capacity for all of them at their peak transmission rate
  - *Assuming independence of traffic sources*, can compute probability that there is enough capacity

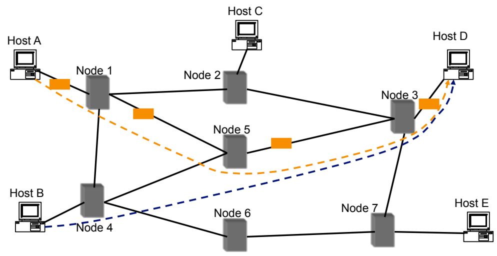
47

Datagram Packet Switching



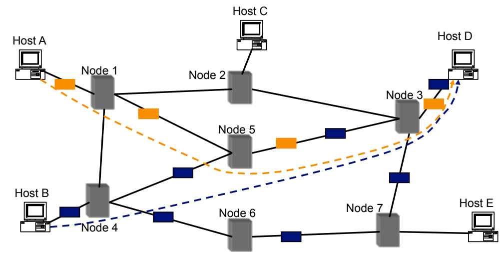
48

Datagram Packet Switching



49

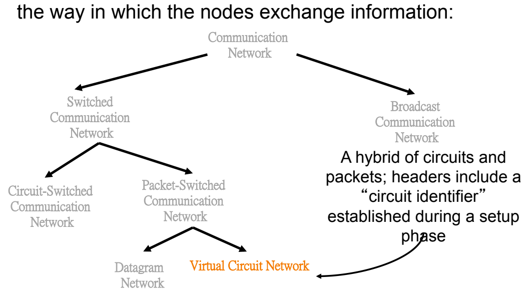
Datagram Packet Switching



50

Taxonomy of Communication Networks

- Communication networks can be classified based on the way in which the nodes exchange information:



51

5 Minute Break

Questions Before We Proceed?

52

Basics of Datagram Networks

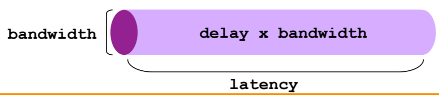
Nodes and Links

- Link: transmission technology
  - Twisted pair, optical, radio, whatever
- Node: computational devices on end of links
  - Host: general-purpose computer
  - Network node: switch or router

54

Properties of Links

- Latency (delay)
  - Propagation time for data sent along the link
  - Corresponds to the “length” of the link
- Bandwidth (capacity)
  - Amount of data sent (or received) per unit time
  - Corresponds to the “width” of the link
- Bandwidth-delay product: (BDP)
  - 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 slow link:
  - B~100mbps
  - L~.1msec
  - BDP ~ 10000bits ~ 1.25MBytes
- Cross-country over fast link:
  - B~10Gbps
  - L~10msec
  - BDP ~ 10^8bits ~ 12.5GBytes

Examples of Transmission Times

- 1500 byte packet over 14.4k modem: ~1 sec
- 1500 byte packet over 10Gbps link: ~10^{-6}sec

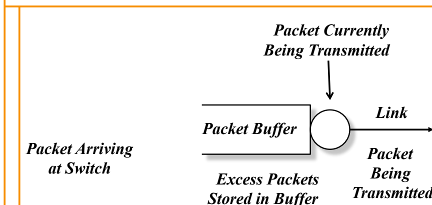
Utilization

- Fraction of time link is busy transmitting
  - Often denoted by ρ
- Ratio of arrival rate to bandwidth
  - Arrival: A bits/sec on average
  - Utilization = A/B = Arrival/Bandwidth

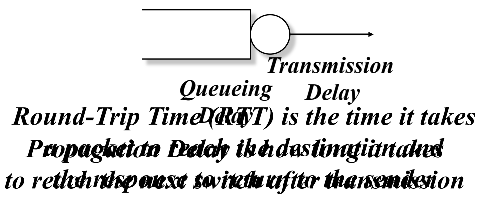
Packets

- Payload (Body)
  - Data being transferred
- Header
  - Instructions to the network for how to handle packet
  - Think of the header as an interface!

The Lifecycle of Packets

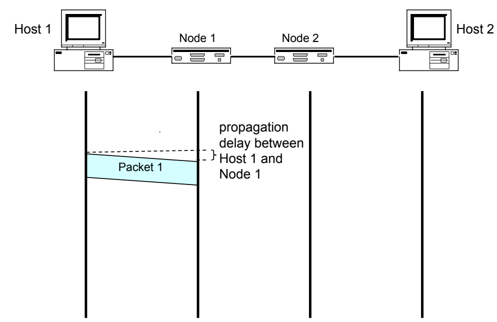


The Delays of Their Lives



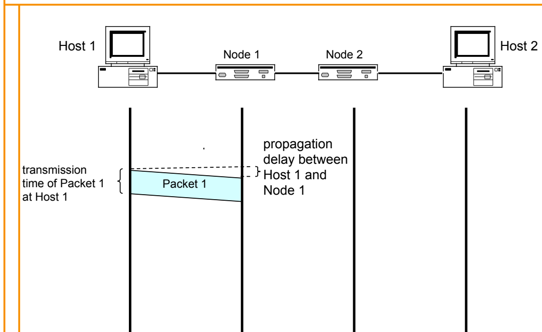
61

Timing of Datagram Packet Switching



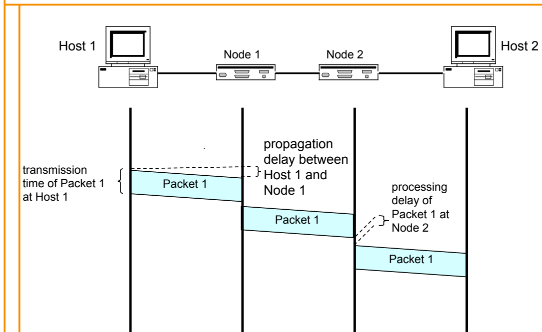
62

Timing of Datagram Packet Switching



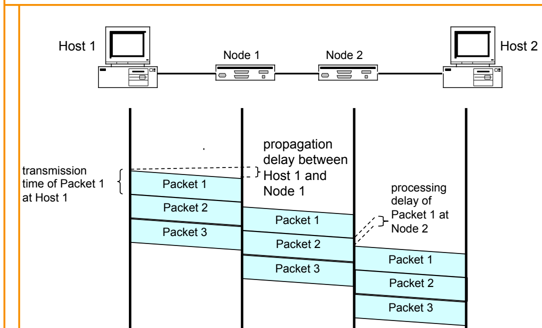
63

Timing of Datagram Packet Switching



64

Timing of Datagram Packet Switching



65

Review of Networking Delays

- Propagation delay: latency
  - Time spent in traversing the link
  - "speed of propagation" delay
- Transmission delay:
  - Time spent being transmitted
  - Ratio of packet size to bandwidth
- Queueing delay:
  - Time spent waiting in queue
  - Ratio of total packet bits ahead in queue to bandwidth
- Roundtrip delay (RTT)
  - Total time for a packet to reach destination and a response to return to the sender

66

Trends

- Propagation delay?
  - No change
- Transmission delay?
  - Getting smaller!
- Queueing delay?
  - Usually smaller
- How does this affect applications?
  - CDNs work very hard to move data near clients
  - Reduces backbone bandwidth requirements
  - But also decreases latency
  - Google: time is money!

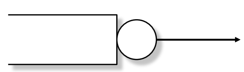
67

Queueing Delay

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

68

Smooth Arrivals = No Queueing Delays



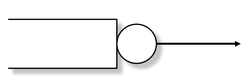
69

Queueing Delay

- Does not happen if packets are evenly spaced
  - And arrival rate is less than service rate
- Queueing delay caused by “packet interference”
  - Burstiness of arrival schedule
  - Variations in packet lengths

70

Bursty Arrivals = Queueing Delays



There is substantial queueing delay even though link is underutilized

71

Queueing Delay Review

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

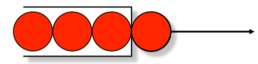
72

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 queueing delay
- Why might an application care about jitter?

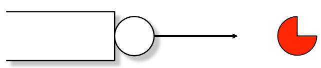
73

Packet Losses: Buffers Full



74

Packet Losses: Corruption



75

Basic Queueing Theory Terminology

- Arrival process: how packets arrive
  - Average rate A
  - Peak rate P
- Service process: transmission times
  - Average transmission time
  - For networks, function of packet size
- W: average time packets wait in the queue
  - W for “waiting time”
- L: average number of packets waiting in the queue
  - L for “length of queue”
- Two different quantities

76

Little’s Law (1961)

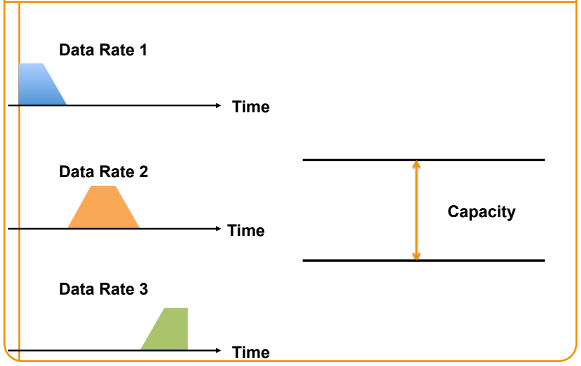
$$L = A \times W$$

- Compute L: count packets in queue every second
  - How often does a single packet get counted? W times
- Could compute L differently
  - On average, every packet will be counted W times
  - The average arrival rate determines how frequently this total queue occupancy should be added to the total
- Why do you care?
  - Easy to compute L, harder to compute W

77

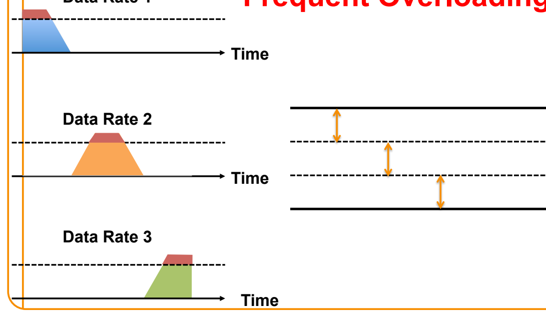
Statistical Multiplexing

Three Flows with Bursty Arrivals

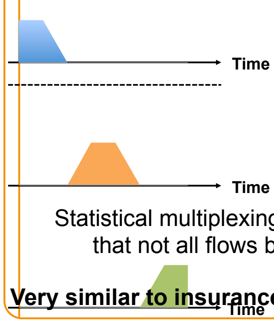


When Each Flow Gets 1/3rd of Capacity

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

Classroom Demonstration of Stat Mux

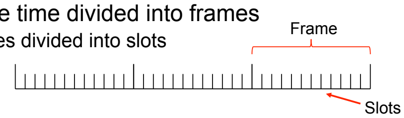
I need 8 volunteers!

- One group of 4:
  - Each generates either 0, 1, or 2 packets per cycle
  - But your link only handles 1 packet per cycle
  - How much of your link do you use (on average)?
- Other group of 4:
  - Each generates either 0, 1, or 2 packets per cycle
  - You share your links, so you can handle 4 packets/cycle
  - How much of your combined link do you use (average)?
- Which team will win?

83

Another Take on "Stat Mux"

- Assume time divided into frames
  - Frames divided into slots



- Flows generate packets during each frame
  - Peak number of packets/frame P
  - Average number of packets/frame A
- Single flow: must allocate P slots to avoid drops
  - But P might be much bigger than A
  - Very wasteful!
- Use the "Law of Large Numbers"….

84

Law of Large Numbers (~1713)

- Consider any probability distribution
  - Can be highly variable, such as varying from 0 to P
- Take N samples from probability distribution
  - In this case, one set of packets from each flow
- Thm: the sum of the samples is very close to N×A
  - And gets percentage-wise closer as N increases
- Sharing between many flows (high aggregation), means that you only need to allocate slightly more than average A slots per frame.
  - Sharing smooths out variations

Simple Example: M/M/1 Queue

- Consider n flows sharing a single queue
- Flow: random (Poisson) arrivals at rate λ
- Random (Exponential) service with average $1/\mu$
- Utilization factor: $\rho = n\lambda/\mu$
  - If $\rho > 1$, system is unstable
- Case 1: Flows share bandwidth
  - Delay = $1/(\mu - n\lambda)$
- Case 2: Flows each have $1/n^{th}$ share of bandwidth
  - No sharing
  - Delay = $n/(\mu - n\lambda)$ **Not sharing increases delay by n**

If you still don't understand stat mux

- Will cover in section….

Recurrent theme in computer science

- Greater efficiency through "sharing"
  - Statistical multiplexing
- Phone network rather than dedicated lines
  - Ancient history
- Packet switching rather than circuits
  - Today's lecture
- Cloud computing
  - Shared datacenters, rather than single PCs

General Lesson: scaling involves

- How you share resources
- How you deal with failures
- …..

Thursday's lecture….

- Layering, principles, the "good stuff"
- Read K&R 1.4-1.8 (mostly for context)