SC/CSE 3213 Winter 2014

L2: Basic Networking Principles

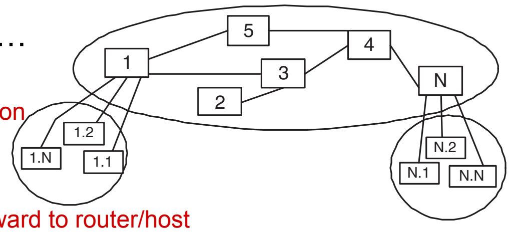
Sebastian Magierowski
York University

Outline

- Sharing
  - To reduce the cost of networking +1 billion devices you must share communication links
- Metrics
  - How do you measure and quantify the performance of a network? (Use some physics and use some math!)
- Scalability
  - The network is huge, organize it in a hierarchy such that changes only have local effects

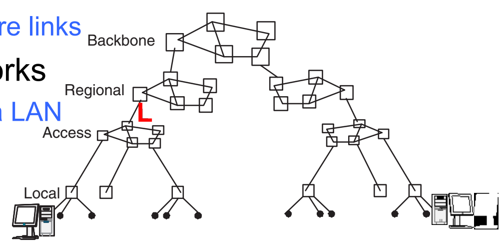
General Structure

- Distributed collections of…
  - hosts
    - sources & sinks of information
    - at the edges
  - routers
    - receive information and forward to router/host
  - links
    - copper/cable/optical/wireless
- For this to work you need to think about arrangement more deeply



Sharing

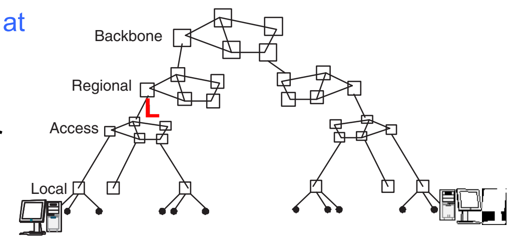
- Given the large amount of hosts…
  - You are going to have to share links
- Build a hierarchy of networks
  - Attach nearby computers in a LAN
  - Merge LANs with an access network
  - Then a regional network
  - Then a cross-country/globe backbone network
- Many computers share a link L



[©Morgan&Claypool, Walrand&Parekh]

Multiplexing Gain

- What makes sharing possible?
  – Computers don't all transmit at the same time
- If you send data into the network for 20 mins. over 8 hours
  – Then you are active only 20/(60•8) = 1/24 the time
  – For a link with transmission rate, R [bps], shared by A computers
    - We have to share A/24 computers
    - Thus each computer can in principle transmit at…
      – R/(A/24) = 24•R/A
    - "Multiplexing Gain"



[©Morgan&Claypool, Walrand&Parekh]

Metrics

- To clarify network characteristics we must precisely define some basic measures of performance
  – Rate
  – Bandwidth
  – Capacity
  – Throughput
  – Delay
  – Delay Jitter
  – Queues

Rate

- How many bits a link can handle per unit time
  - DSL
    - downlink rate (768 kbps)
    - uplink rate (256 kbps)
- Very loosely a "broadband" link > 100 kbps
  - A non-universal value
- We'll (usually) use *R* to denote link rate

Bandwidth

- Spectral extent of a signal
- "slow/fast" -> "low/high"

Fourier Series

- Finding the spectrum of a periodic signal

Fourier Transform

- Spectrum of non-periodic signals

Spectrum of a Pulse

- A pulse in the time domain gives a sinc spectrum
- Imagine a bit stream with *a* = 1 us
- As *R* increases *W* increases

Spectrum of a Sinc

- Spectrum of different pulse
- Bandwidth clear for this

Link Bandwidth

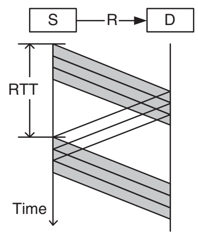
- Links with sufficient bandwidth, *W*, pass spectral components of a signal without excessive attenuation

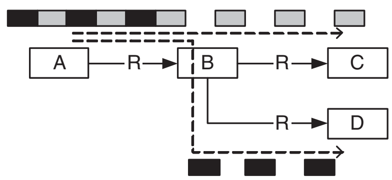
Link Capacity

- What is the biggest R I can stuff through a link…
- …and completely recover ALL the data I send through it
  - i.e. zero error communication
- $R_{max}|_{no\ error} = C =$

Throughput

- Rate of data transfer but…
  - …accounts for overhead of communication and network details





- 3 MB file takes 2 minutes to download
  - *T* is?

Delay

- Time elapsed between two points of interest
- Consists of
  - transmission time
  - propagation time
  - queuing time
  - processing time

Delay Jitter

- Delay varies
- Delay jitter, J = max delay – min delay
- Important for streaming or real-time apps
- Buffer for at least J

Queue Delay Estimation

- λ: packets arriving per second
- μ: packets processed per second
- ρ: utilization = λ/μ
- Rate at which a queue is emptied?
- Avg. time to empty a queue?
- Avg. wait time to get to front of queue?
- Avg. number of packets in queue?
- R = 10 Mbps, P = 1 KB, λ = 1000
  - avg. delay, T? (transmission and queuing components?)
  - avg. queue length, L?
  - delay jitter ~ 3x average delay

Congestion

- If λ goes up to 1150
  - T now?
  - L now?
  - J now?
- Average packet delay and jitter increase quickly as λ approaches μ

Link Sharing

- What if we have N computers share a link of speed $N\lambda$
  - $T_N = T/N$
    - Our average delay (and jitter drops)
  - $L_N = L$
    - Same number of average packets queued
- Sharing a faster link (instead of using a slower dedicated link)
  - reduces delay
  - transmission time is only $1/\mu N$

Little's Result

- $N = \lambda \cdot T$
- N: avg. number of packets in the system (i.e. network)
- T: avg. time spent in the system
- λ: avg. packet arrival per second in the system
- How many packets in Internet?
  - 1B users, 100 MB/day, 0.1 seconds in Internet
- How many bits stored in a fiber?
  - 2.4 Gbps, 20% capacity, 100 km long
- How many bits in a router?
  - 16 1-Gbps ports, 10% capacity, each bit in router for 5 ms

Fairness

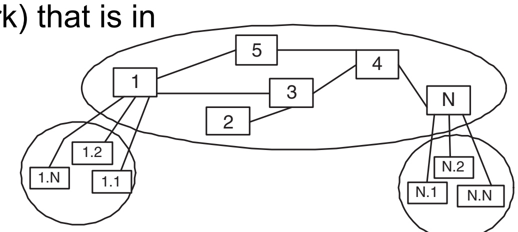
- What rates should be allocated to different flows
- For example…
  - 1 Mbps link
  - Two flows: 1.6 Mbps and 0.4 Mbps
  - How do you partition the resource?
    - In proportion to desired throughput?
      - 0.8 Mbps/0.2 Mbps
      - Fair to 0.4 Mbps? Can't get through even though it asks for less?
    - To the lowest request?
      - 0.6 Mbps/0.4 Mbps
      - "max-min allocation"
- Achieving fairness a complex issue in real networks

Scalability

- For a network to grow large in size modifications must have only limited effect
- Assuming 50% of the 1B Internet computers added in the last 10 years
  - ~100,000 computers added every day!
- ARPANET routers stored the address of each computer on the network
- In light of scaling rates this was unsustainable
- A number of other basic network functions need to account for scaling…

Location-Based Addressing

- Define address of a computer in terms of the group (network) that is in



- $M = N^2$ devices (N groups of N devices each)
- How many devices does each router have to be aware of?
- What's it look like as you scale?

Two-Level Routing

- How do you actually learn the best way to route a message?
- Routers have to exchange messages with each other to learn the metrics of their links (and keep updating this information)
- For *N* routers, *N* messages must be sent
- If routers are partition into…
  - $\sqrt{N}$ domains and
  - $\sqrt{N}$ sub-domains (inside domains)
- …much less routing messages are required

Best Effort Service

- Internet does not guarantee any precise property (e.g. delay, throughput, etc.)
- Provides best effort service
  - Try to deliver packets as well as possible
- As technology improves more and more demanding applications become supported
- Applications adapt to changing quality of services

End-to-End Principle & Stateless Routers

- Best effort service does not require routers to keep track of connection details (stateless)
- Message errors are corrected by source & destination
  - Arrange for retransmissions
  - End-to-end principle
    - Tasks should not be performed by routers if they can be performed by end devices
- Routers perform tasks on individual packets
  - No information on state of end-to-end connection
  - Only looks at errors in individual packets and forwards if correct
  - Does not keep copy of packet for retransmission
  - Simplifies design and keeps Internet scalable