

# ICS 351: Today's plan

- Performance
- Peer-to-Peer Networking

# Latency

- latency is measured in seconds:
  - one-way (useful, but hard to measure)
  - round-trip (also useful, and easy to measure)
- usually measured by sending the smallest possible transmission unit
- latency matters when there are lots of round-trips
  - web transfers (less so with HTTP/2)
  - remote login
- latency matters when things must happen in real time
  - telephony, gaming, trading, remote control, telemedicine

# Throughput

- throughput is measured in bits/second or Bytes/second
  - network equipment generally rated in bits/second
  - network downloads generally list Bytes/second
  - in this class and in much (but not all) of the world:
    - b/s stands for bits/second
    - B/s stands for Bytes/second
- throughput matters when lots of data must be sent
- `ttcp`, `netperf`, and `iperf` measure and report throughput

# Considerations

- at each level in the network, it is usually cheaper to provide less performance than more
  - in other words, it is more profitable to serve more people given the same infrastructure
- HTTP headers consume multiple hundred bytes, as opposed to about 40-60 bytes for TCP/IP headers
- but until HTTP/2, nobody was concerned enough to try to compress them
  - so the inefficiency must be negligible compared to the available throughput

# A more advance performance test

- send two successive pings as fast as possible
- at the slowest link in the path, the second will be delayed in proportion to the size of the first
- this allows us to estimate the speed of the *bottleneck link*
- `bwping` performs this test

# Peer-to-peer Networking

- in this class, it has been clear that a PC can perform as a router, and vice-versa
- why not use our PCs as routers?
- for many purposes this is adequate

# Content-Addressable Networking

- if we can change from IP addresses, there is less need for hierarchical control
  - content-addressable networks!
  - “Give me X”
- to increase the incentive, every node that has content and makes it available might get priority for obtaining other content
  - P2P networking, e.g. bittorrent
- the decentralization and lack of control are attractions for many of the users

# Managing Peer-to-peer Networks

- management of a network requires some authorities that cooperate
- in a peer-to-peer network, both the management and the authority are made as small as possible
- because there is no hierarchical assignment of addresses, each peer can decide what content to provide, i.e. what "address" to use
- this minimizes management

# Connecting Peer-to-peer Networks

- the network is connected by having one peer exchange addresses about other peers
- then, if the original peer stops collaborating, the other peers can be used to connect to the network
- the expectation is that some useful data exchange will take place, not necessarily that there will be continuous end-to-end connectivity with all parties involved

# Examples of Peer-to-Peer

- BitTorrent
- Distributed Hash Tables
- Freenet
- AllNet
- sneakernet
  - maybe: Skype

# Peer-to-peer vs. Client-Server

- a server is a program that provides a service
- typically, a server is found at a given address and port number, and is maintained by an individual or an organization
- clients may be anonymous or unidentified
- peer-to-peer (P2P) often brings to the infrastructure the anonymity of clients
- could imagine authenticated P2P networks, set up for reliability rather than anonymity

# Peer-to-peer vs. Client-Server II

- for a very different example, routing is a peer-to-peer process
  - routers identified by IP address or ID
- the ultimate appeal of peer-to-peer for some networking people is a self-organizing, self-managing scalable network
- peer-to-peer networks seem to be widely used