Internet Protocols
Fall 2004

Final Review

Andreas Terzis

Final

- Thursday Dec 16th
  - 9-11 am
  - Shaffer 304
- Details
  - Closed books
  - Bring a calculator
  - Combination of quantitative and qualitative questions

Transport Protocols

- Provide end-to-end services on top of raw, unreliable packet delivery
- UDP: User Datagram Protocol
- TCP: Transmission Control Protocol

Common Functionality

- Demultiplexing: Source/Destination ports
- Checksum (optional in UDP)

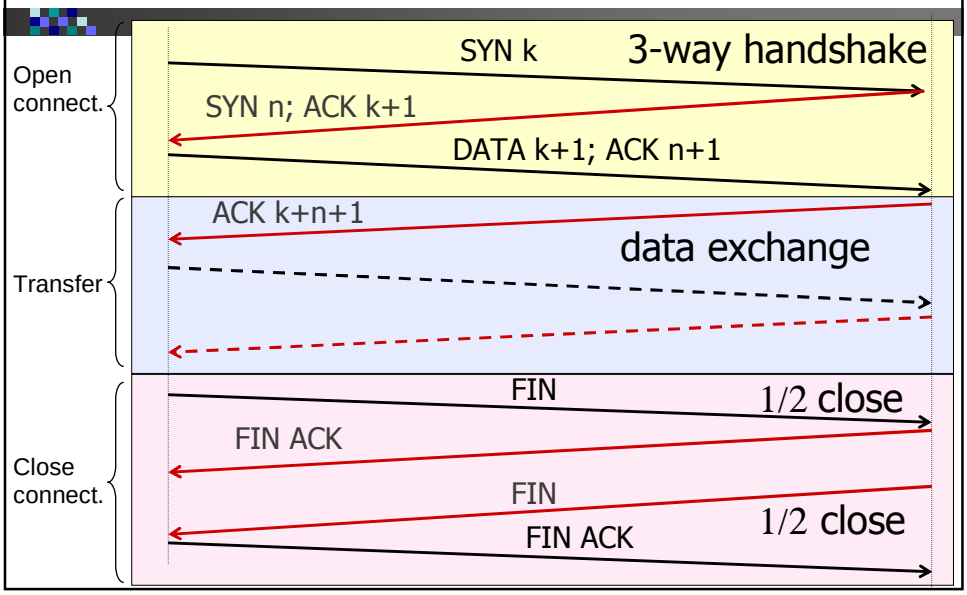
How Data Gets End-to-End

- MAC address (from ARP)
- IP address (from DNS)
- Protocol (in IP header)
- Port # (from OS)

Additional TCP Functionality

- Reliable delivery
- Congestion control

Timing Diagram



TCP Reliable Delivery

- Sender waits for ACK of data
  - if data not ACK'ed, retransmit
- Sliding window approach:
  - Allow W packets in flight at any time
  - ACK last packet received
- Two signs for retransmission:
  - timeout
  - duplicate ACKs

Congestion Control

- Signals of congestion:
  - Packet drops: timeouts or duplicate ACKs
  - Explicit signals: ECN
- Three goals:
  - quickly approach the available bandwidth
  - adjust to small (and large) variations in bandwidth
  - share bandwidth fairly with other flows

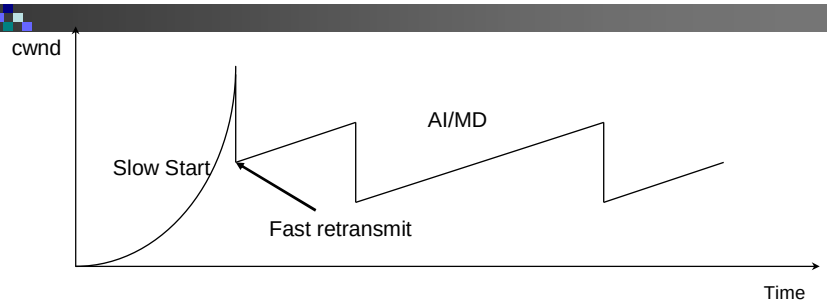
Slow-Start

- Increase cwnd by 1 for every ACK until cwnd hits ssthresh
- Sending rate increases exponentially (until queueing sets in or sshtresh is hit)

Congestion-Avoidance

- Cwnd increase by 1/cwnd for every ACK
- Set ssthresh = cwnd/2 upon loss
- Cwnd decreases by 1/2 upon 3 duplicate ACKs
- Cwnd goes to 1 upon timeout

Window Dynamics

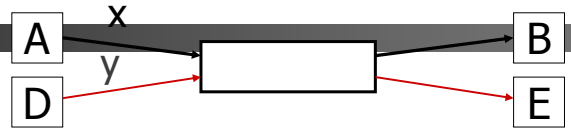


- Cwnd increases quickly
- Begins oscillating around appropriate value
- As long as no timeout, can usually keep pipe full

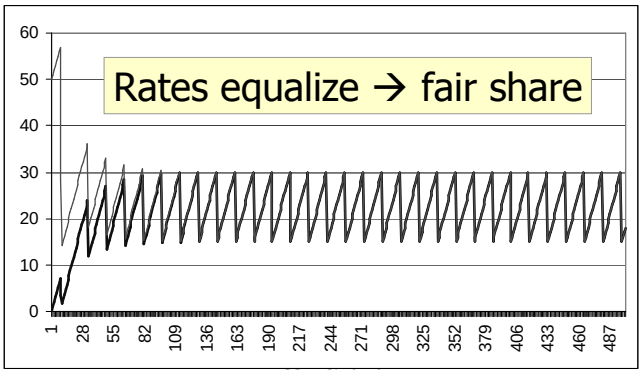
Why AIMD?

- MIMD and AIAD could also work
- But neither results in fair bandwidth allocation

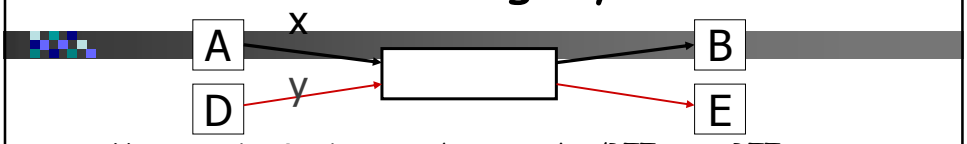
AIMD Sharing Dynamics



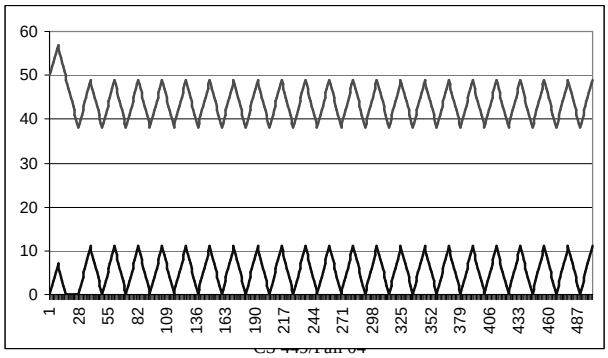
- No congestion → rate increases by one packet/RTT every RTT
- Congestion → decrease rate by factor 2



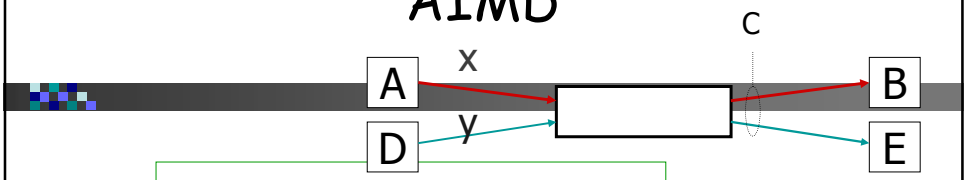
AIAD Sharing Dynamics

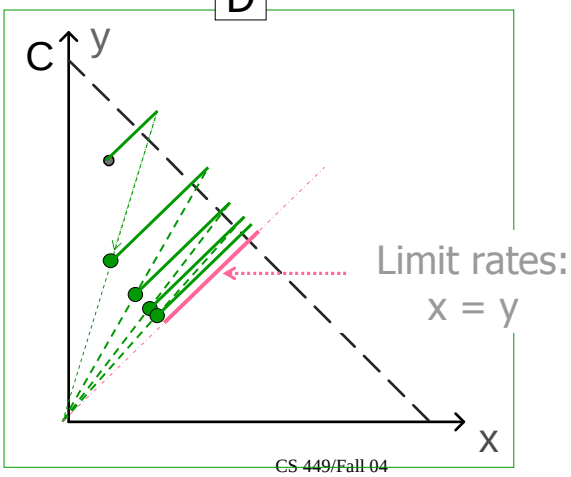


- No congestion → x increases by one packet/RTT every RTT
- Congestion → decrease x by 1

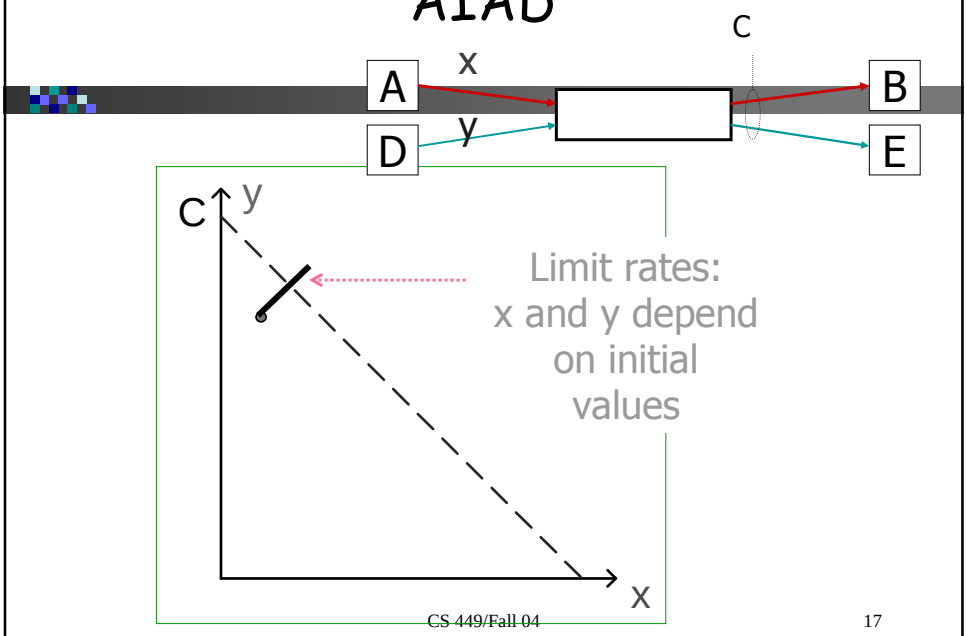


AIMD





AIAD



TCP Congestion Control Summary

- Measure available bandwidth
  - slow start: fast, hard on network
  - AIMD: slow, gentle on network
- Detecting congestion
  - timeout based on RTT
    - robust, causes low throughput
  - Fast Retransmit: avoids timeouts when few packets lost
    - can be fooled, maintains high throughput
- Recovering from loss
  - Fast recovery: don't set cwnd=1 with fast retransmits

Quality-of-Service

- Integrated Services (IntServ)
  - services: controlled load and guaranteed service
  - RSVP (reservation protocol)
  - packet scheduling algorithms
  - admission control algorithms
    - measurement-based for controlled load
    - worst-case for guaranteed service
- Differentiated Services (DiffServ)
  - per-hop behaviors
  - think of priority levels (specified by ToS bits)

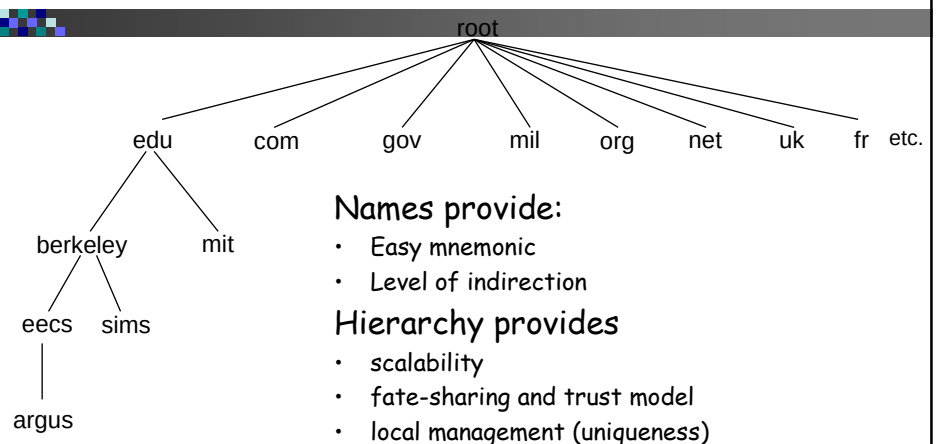
Application Layer

- DNS
- HTTP/CDN
- Peer-to-Peer
- Security
- Multimedia protocols

From Names to Addresses

- Users use hostnames
- Protocols use IP addresses
- DNS translates between them

Naming Hierarchy



Names provide:

- Easy mnemonic
- Level of indirection

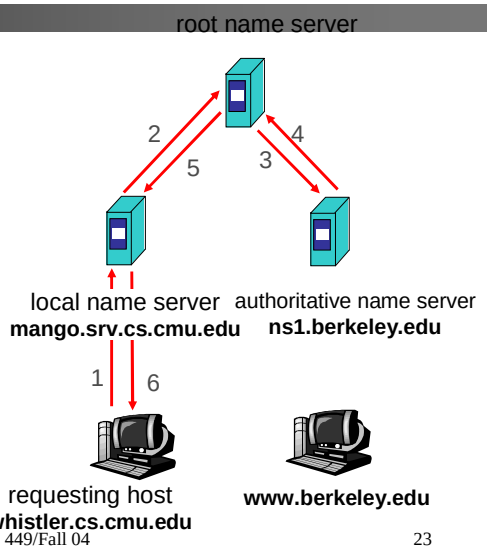
Hierarchy provides

- scalability
- fate-sharing and trust model
- local management (uniqueness)

Simple DNS Example

Host **`whistler.cs.cmu.edu`** wants IP address of **`www.berkeley.edu`**

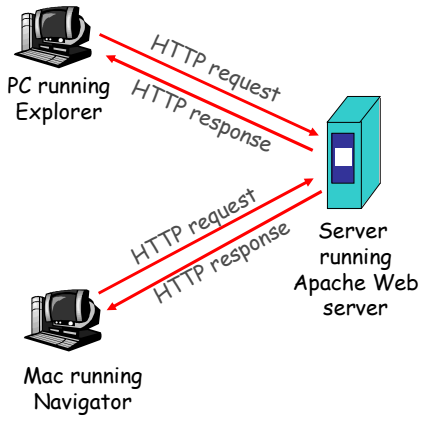
1. Contacts its local DNS server, **`mango.srv.cs.cmu.edu`**
2. **`mango.srv.cs.cmu.edu`** contacts root name server, if necessary
3. Root name server contacts authoritative name server, **`ns1.berkeley.edu`**, if necessary



HTTP overview

HTTP: hypertext transfer protocol

- Web's application layer protocol
- client/server model
  - *client:* browser that requests, receives, "displays" Web objects
  - *server:* Web server sends objects in response to requests
- HTTP 1.0: RFC 1945
- HTTP 1.1: RFC 2068



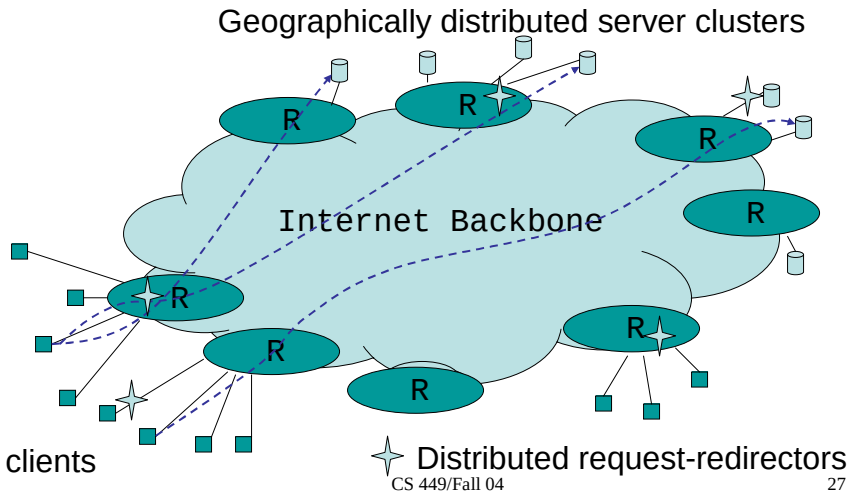
HTTP Topics

- Non-persistent vs. persistent connections
- Authorization
- Cookies
- Client-side Caching and conditional GETs

HTTP-TCP Interactions

- Effects of losses on HTTP
  - If first segment is lost long retx timer
  - HTTP responses are short->no duplicate acks
- Slow-start restart
- TIME_WAIT state on servers
- Nagle's algorithm and HTTP
- Delayed ACKs

Redirection Overlay



Story for CDNs

- Traditional: *Performance*
  - move content closer to the clients
  - avoid server bottlenecks
- New: *DDoS Protection*
  - dissipate attack over massive resources
  - multiplicatively raise level of resources needed to attack

Redirection Techniques

- DNS
  - one name maps onto many addresses
  - works for both servers and reverse proxies
- HTTP
  - requires an extra round trip
- Router
  - one address, select a server (reverse proxy)
  - content-based routing (near client)
- URL Rewriting
  - embedded links

Peer-to-Peer

- DNS maps names into addresses
  - host-centric
- Peer-to-peer allows users to search directly for content
  - data-centric
- Two basic styles:
  - unstructured (Gnutella): flood requests
  - structured (DHTs): exact matching

Multimedia Applications

- Multimedia applications are a different class of applications
  - Examples: A/V (streaming and conferencing), games
  - Have different requirements
  - Require different application protocols
    - Signaling (RSVP)
    - Transport (RTP)
    - Session Control (SIP)

The end

- Thank you!
  Good luck!