15-441: Computer Networking

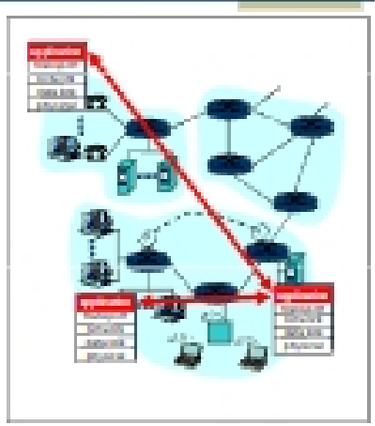
Lecture 4: Design Philosophy & Applications

Lecture Overview

- Last time:
  - Protocol stacks and layering
  - OSI and TCP/IP models
- Application requirements
- Application examples
  - http
- Project information
- Internet Architecture

Applications and Application-Layer Protocols

- Application: communicating, distributed processes
  - Running in network hosts in "user space"
  - Exchange messages to implement app
  - e.g., email, file transfer, the Web
- Application-layer protocols
  - One "piece" of an app
  - Define messages exchanged by apps and actions taken
  - User services provided by lower layer protocols



Client-Server Paradigm

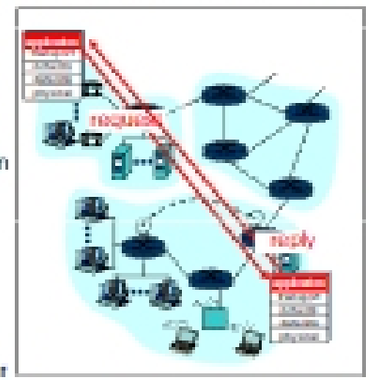
Typical network app has two pieces: *client* and *server*

Client:

- Initiates contact with server ("speaks first")
- Typically requests service from server,
- For Web, client is implemented in browser; for e-mail, in mail reader

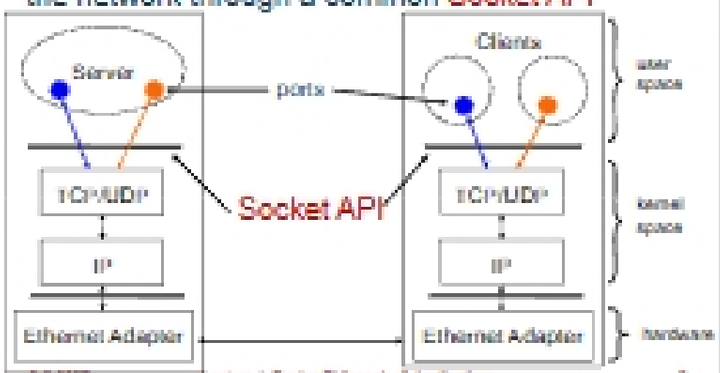
Server:

- Provides requested service to client
- e.g., Web server sends requested Web page, mail server delivers e-mail

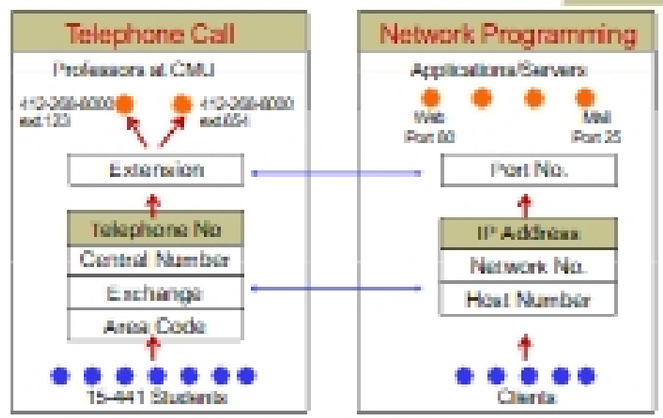


Server and Client

Server and Client exchange messages over the network through a common Socket API



Network Addressing Analogy



What Service Does an Application Need?

Data loss

- Some apps (e.g., audio) can tolerate some loss
- Other apps (e.g., file transfer, telnet) require 100% reliable data transfer

Timing

- Some apps (e.g., Internet telephony, interactive games) require low delay to be "effective"

Bandwidth

- Some apps (e.g., multimedia) require minimum amount of bandwidth to be "effective"
- Other apps ("elastic apps") make use of whatever bandwidth they get

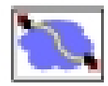
Transport Service Requirements of Common Apps

Application	Data loss	Bandwidth	Time Sensitive
file transfer	no loss	elastic	no
e-mail	no loss	elastic	no
web documents	no loss	elastic	no
real-time audio/ video	loss-tolerant	audio: 5Kb-1Mb video:10Kb-5Mb	yes, 100's msec
stored audio/video	loss-tolerant	same as above	yes, few secs
interactive games	loss-tolerant	few Kbps	yes, 100's msec
financial apps	no loss	elastic	yes and no

Other Requirements

- Network reliability
  - Network service must always be available
- Security: privacy, denial of service, authentication, …
- Scalability.
  - Scale to large numbers of users, traffic flows, …
- Manageability: monitoring, control, …

User Datagram Protocol(UDP): An Analogy



UDP	Postal Mail
• Single socket to receive messages • No guarantee of delivery • Not necessarily in-order delivery • Datagram – independent packets • Must address each packet	• Single mailbox to receive letters • Unreliable ☺ • Not necessarily in-order delivery • Letters sent independently • Must address each reply

Example UDP applications
Multimedia, voice over IP

Transmission Control Protocol (TCP): An Analogy

TCP	Telephone Call
• Reliable – guarantee delivery • Byte stream – in-order delivery • Connection-oriented – single socket per connection • Setup connection followed by data transfer	• Guaranteed delivery • In-order delivery • Connection-oriented • Setup connection followed by conversation

Example TCP applications
Web, Email, Telnet

HTTP Basics

- HTTP layered over bidirectional byte stream
  - Almost always TCP
- Interaction
  - Client sends request to server, followed by response from server to client
  - Requests/responses are encoded in text
- Stateless
  - Server maintains no information about past client requests