Lecture 2: Networks Models

Networks

- Independent machines connected to each other in some topology.

- Machines as:
  - Computers
  - Switches
  - Routers
  - Base stations

- Types of networks:
  - Local Area Network (LAN)
  - Wide Area Network (WAN)
  - Internet
  - Wireless LAN
  - Global System for Mobile (GSM) communication
Issues to consider

- Performance
- Mobility
- Scalability
- Reliability
- Availability
- Quality of Service

Communication over a network

Message passing

- Sequence of bits sent over the network.

- Purpose of messages:
  - Send requests
  - Send data
  - Send code to another machine
  - Instruct the machine to perform a task

- Messages can’t just be ad-hoc,
  - A protocol has to be adopted

The Internet protocol stack

- Application layer supports network applications (HTTP, SMTP, FTP).
- Transport layer provides transport of messages between client and server (TCP and UDP).
- Network layer is responsible for routing datagrams from one host to another (IP).
- Link layer moves frames from node to node (Ethernet, PPP).
- Physical layer moves individual bits of within a frame from node to node.
To build a network application, we must understand how processes running in multiple end systems communicate?

Interprocess communication is governed by the end system’s operating system (CS341).

Communication between end systems is accomplished by exchanging messages across a computer network (CS242).

Distributed computations depends on communication between end systems.
Network architectures

- Client-Server

- Peer-to-peer

Hybrid

Client-server architecture

- For each pair of communicating processes there is a client side and a server side.

- The host that initiates the session is labeled the client, while the process that waits to be contacted is the server.

- The Web, file transfer, remote login, and e-mail all fit this bill.
Socket programming

- A process sends messages into, and receives messages from, the network through its socket or API.
- A socket is the interface between the application layer and the transport layer within a host.

TCP services

- Provides a connection-oriented service.
- Handshakes set up a full duplex TCP connection between sockets prior to the exchange of information.
- Provides reliable transport service.
- The communication processes can rely on TCP to deliver all data sent without error and in the proper order.
- Includes a congestion control mechanism.
  - But at a cost and TCP does not guarantee a minimum transmission rate.
UDP services

- No frills delivery.
- There is no sequencing of data.
  - If ordering is required, it has to be managed by the application layer.
- The delivery of data to the destination cannot be guaranteed.
- There is no retransmission of lost packets in User Datagram Protocol (UDP).
- UDP is lightweight.
- UDP is faster, simpler and more efficient than TCP.

Demo time!

Get up and come to the board
Is this the only way?

- No!

- Several programming models exist to ease communication.
- Such as:
  - Remote Procedure Call (RPC)
    - Lightweight method to execute code on remote machine
  - Remote Method Invocation (RMI)
    - Object oriented version of RPC
  - Common Object Request Broker Architecture (CORBA)
    - Provides remote procedure invocation through an intermediate layer called the object request broker.

Let’s continue the demo!