Networking

Introduction to Computer Science
Robert Sedgewick and Kevin Wayne

Lecture Outline

Client/Server Programming in Java
- Socket Abstraction
- TCP/IP Protocols

The Internet: The view from your laptop
- www.broadband.gov
- www.measurementlab.net
- traceroute, ping, dig

Some Important Concepts
- Statistical Multiplexing, Contention, and Congestion
- Congestion Control (AIMD Algorithm)
- Latency, Bandwidth, and Delay x Bandwidth Product
- Jitter and Audio/Video Playback

Internet

Internet.
- Global communication network containing million of computers.
- Computer networks communicate using TCP/IP protocols.
- Provides access to services: email, chat, world wide web, BitTorrent.
- Started by military around 1969 as ARPANET: survivability, robustness, efficiency.
- Operating system and hardware independent.

Everybody but you grew up without it!

Client/Server

Stream socket.
- ADT for two-way communication over the Internet.
  - read from socket input and write to socket output
- IP address: identifies computer.
- Port number: identifies application.
- Ex: IP address = 128.112.129.71, Port = 25.

- Purpose of a Socket is to communicate with another Socket.

Client/server model.
- Client = creates socket to communicate with specified server.
- Server = creates one socket to listen for connection requests;
  creates another socket to communicate with each client.

Ex: client = web browser, server = web server.
Echo Client

Echo client: connect with server, read text from standard input, send text to server, print whatever server sends back.

```java
import java.net.Socket;
public class EchoClient {
    public static void main(String[] args) throws Exception {
        Socket socket = new Socket(args[0], 4444);  // open socket
        In stdin = new In();
        In in = new In(socket);  // server name
        Out out = new Out(socket);
        String s;
        while ((s = stdin.readLine()) != null) {  // read from stdin
            out.println(s);  // send to server
            System.out.println(in.readLine());  // get from server
        }
        out.close();  // close socket
        in.close();
        socket.close();
    }
}
```

Echo Server

Echo server: use ServerSocket to listen for connection requests; connect with a client; read text that client sends; and send it back.

```java
public class EchoServer {
    public static void main(String[] args) throws Exception {
        ServerSocket serverSocket = new ServerSocket(4444);  // open server socket and listen on port 4444
        while (true) {  // listen and wait for client to request connection
            Socket socket = serverSocket.accept();  // read data from client and echo it back
            In in = new In(socket);  // server name
            Out out = new Out(socket);
            String s;
            while ((s = in.readLine()) != null) {  // read data from client
                out.println(s);  // send to server
            }
            out.close();  // close input streams and socket
            in.close();
            socket.close();
        }
    }
}
```

Protocols

Internet Protocol (IP)
- Rules for routing packets from machine A to machine B.
- Each packet forwarded independently, possibly on different paths.
- No guarantee packets arrive in order, or even arrive.

Transmission Control Protocol (TCP)
- Rules to provide communication between client/server programs.
- Fragment application data into packets and send using IP.
- Reassemble IP packets into ordered, reliable byte-stream.
- Acknowledge received packets; retransmit missing packets.

Application Layer Protocols
- Send/Receive user data using TCP/IP.
- HTTP (web), SMTP (email), SSH (secure login), RTMP (video)…

Internet: View from your laptop

Broadband Subscribers

Internet Service Provider (ISP)

Access Technology (e.g., FIOS, WiFi, WiMax, Cellular…)

Rest of the Internet

YouTube.com

Netflix.com

cnn.com

Router

Peering Point
Multiplexing

Network-speak for "resource sharing"

- time division multiplexing
- frequency division multiplexing
- statistical multiplexing

Contention – multiple packets want service at the same time

- packets wait in a queue (FIFO, Priority)

Congestion – too many packets waiting for service

- drop a packet (tail-drop, random drop)

TCP Congestion Control

TCP source adjusts sending rate to match network's capacity

- Sender reacts to implicit "signals" from the network

  - Receiving an ACK for a packet says "send faster"
  - Not receiving an ACK after a period of time says "send slower"

- Speed-up conservatively & slow down aggressively

  - Additive Increase / Multiplicative Decrease (AIMD)

Performance

Latency (aka Delay)

- Time it takes one bit to travel end-to-end (e.g., 300ms)
- Propagation delay (speed of light) + queuing delay
- Also interested in round-trip-time (RTT)

Bandwidth (aka Throughput)

- Number of bits that can be transmitted in a period of time (e.g., 1Mbps)

Delay-Bandwidth Product

- Number of bits that can be "in flight" before sender hears back
  - E.g., 100ms RTT x 16bps = 100Mb ~ 10MBytes

Audio/Video Applications

Jitter – Variations in network delay

Playback Buffer – Big enough to "absorb" variable delay
Access Network Revisited

Content Distribution

Telco Backbone

Subscribers

More Information

**COS 318 Operating Systems**
- How to write multi-threaded servers

**COS 461 Computer Networks**
- TCP, IP, and everything in between