Networking Fundamentals

How Computers Talk to Each Other

Fundamentals of Communication

- Computers need to be told everything in intimate detail because they are dumb.
  - We want to make computers talk to each other.
  - We must specify in very low level details exactly each computer must do to!

- What defines communication?
  - What must two entities agree on?
  - What do we take for granted on an everyday basis?
The Things We Need…

- **Naming Convention**
  - Who am I talking to?
- **Common Language**
  - What is it that I’m trying to say?
- **Transport Medium**
  - How is my message sent from A to B?

Naming Convention

- This is paramount, equally as important as the somewhat more obvious things.
- Names must be unique within our universe to prevent collision problems:
  - Perhaps we might want to make names easy to remember and/or understand.
  - May need multiple kinds of names (SSN).
- The concept of a broadcast and/or multicast is also equally as important in many cases.
More About Names...

- Sometimes we need physical/logical names
- Physical names are *inherent* unique identifiers
  - Social Security Number
  - DNA patterns
- Logical names are *assigned* for grouping and convenience
  - First name – convenience
  - Last name – grouping

Common Language

- Both parties must agree to a common language
- Generally two aspects to this:
  - Protocol
  - Semantics
- Protocol is basically about who can speak when
- Semantics are about what the messages mean
- Everybody must agree to the same common language otherwise nothing will work
Protocol

- In the olden days, if you went to see the King unannounced, two things could possibly happen:
  - You are executed
  - The King grants you anything you want
- A computer protocol might be:
  - Client initiate a connection
  - Server responds with magic number
  - Client responds received magic number
  - Client asks for data, sends magic number as key

Semantics

- There must be agreement about which symbols to use to represent which concepts.
  - American English: tic-tac-toe
  - British English: naughts-and-crosses
- What if the same symbol means two different things on different systems?
  - 3/4/2000… March 4th or April 3rd?
  - 2000-MAR-03 is a better idea
A Transport Medium

• It's a physical necessity...
  • Human speech is carried through compression waves occurring in the air
  • Whale songs are carried as compressions waves through water
• Computers can be much fancier...
  • Wire guided EM radiation (copper)
  • Light pipe guided EM radiation (fiber optics)
  • Open air broadcast EM radiation (wireless)

Two Computers Talking

• Each computer has a unique name
• Both computers must have the same network protocol installed
  • TCP/IP
  • NetBIOS/NetBEUI
  • Appletalk
• Application programs make system calls to the operating system to setup / sendinfo / breakdown the communication system
ISO OSI Model

- Application
- Presentation
- Session
- Transport
- Network
- Data Link
- Physical

- Model the communication between two entities as going through a series of layers
- Layers can be changed easily, allows for a very flexible system design
- Often called a protocol "stack," (not the same as stack data structure)

Physical Layer

- This is where the hardware of the network infrastructure lives
- Communication is solely in terms of 0 and 1
- Examples include:
  - 10Base2 - coaxial cable
  - 10/100/1000BaseT(X) - copper twisted pair
  - 100BaseFX/1000BaseSX - fiber optics
  - ISM band (2.4 Ghz) wireless
Data Link Layer

- Sometimes called Media Access layer
- Deals with detecting physical layer connection outages
- Handles physical device naming
  - Names are assigned at the factory and never changed
  - Names must be globally unique
  - Ethernet MAC addresses - 00:A0:00:B3:46:78
- We must encode all this into the packet

Network Layer

- Here we control how packets are routed from one place to another
- Most network connections in use today are point to point
  - Broadcast mediums only work for short distances, long links are always P2P
- Here we decide which route to take and encode that information into the packet
- The Internet Protocol (IP) is an example
Transport Layer

- Communication systems are inherently fragile ... they are susceptible to errors
- We want to make sure that our packets get from point A to point B (in the right order too)
- The transport layer breaks messages into packets and marks them with sequence numbers
  - If a packet is missing, it is retransmitted
  - Before the packets move up the stack, they are sorted, reordered and recombined

Session Layer

- Here we handle multiple people talking to us at the same time
- We also take care of the language we will use to request/send information
- Common session layer protocols:
  - SQL
  - HTTP
  - X-Windows
Presentation Layer

- Used to place/decode data to/from an agreed upon format
- Common presentation formats:
  - Images - JPEG/GIF
  - Audio - WAV/MP3
  - Text - ASCII/UTF/HTML

Application Layer

- This is the layer that has the code which defines why we are doing the communication
- However, from a communications perspective, it is like the physical layer, it just needs to be present because it is there
MAC Addresses

- All network interfaces have a globally unique MAC address assigned at the factory
- This address can/should never be changed because that may cause a naming conflict
- Since they cannot be changed, it becomes very difficult for us to keep track of which machine belongs to whom
- Also, routing becomes a big problem

Routing

Each entity must know where to send its packets!
**IP Numbers**

- Globally unique, but logically assigned
- Permits us to group machines together by assigning a sequence of numbers
- Typical IP address: 128.59.16.1
- ARP - protocol to resolve IP addresses into MAC addresses
- Last leg of communication can only take place using MAC addresses

**IP Subnets/Supernets**

- By grouping machines together, we can have routing tables a reasonable # of routes
  - 128.59.16.0 - 128.59.23.255 - go south
  - 128.59.15.0 - 128.59.15.255 - go north
  - Everybody else - go east
- We actually specify these with subnet masks
  - 128.59.16.0 / 255.255.248.0
  - 128.59.15.0 / 255.255.255.0
All these numbers!

- 128.59.21.155 is pretty hard to remember
- Also, address blocks are linear, we want a heirarchal system similar to way we think
- Enter the DNS (Domain Name System)
  - Map heirarchal names to IP numbers
  - www.cs.columbia.edu -> 128.59.16.149
- There is a central root, but domains are delegated so that the actual maintenance of the database is handled by the site IT people

We also need ports!

- An IP address specifies a machine, but that machine might have many programs and/or users on it doing different things
- A port is an Apt #, IP# like a street address
- Ports usually range from 0 to 65537
- Below 1024 are Well Known Services
  - 80 - WWW
  - 25 - SMTP (email)
  - 43 - DNS
The Socket API

• Works across many platforms
• Available in many programming languages
• It's the "low level" API in JAVA
  • RMI is the medium level one
  • JAVASpaces, JINI, etc. are the high level ones
• Just like writing to a file... except it goes across the network.
• Don’t forget to try and catch!

Client API

```
Socket theServer = new Socket("www.server.com", 4567);
InputStreamReader ISR = new InputStreamReader(theServer.getInputStream());
BufferedReader fromServer = new BufferedReader(ISR);
PrintWriter toServer = new PrintWriter(theServer.getOutputStream());
String messageFromServer = fromServer.readLine();
toServer.println("Hello, world!");
```
Server API

```java
ServerSocket srvSock = new ServerSocket(4567);
Socket clientConn = srvSock.accept();
InputStreamReader ISR = new
    InputStreamReader(clientConn.getInputStream());
BufferedReader fromClient = new
    BufferedReader(ISR);
PrintWriter toClient = new
    PrintWriter(clientConn.getOutputStream());
String messageFromClient =
    fromClient.readLine();
toClient.println("Hello, world!");
```

API Caveats

- Don’t forget to import!
  - import java.net.*;
  - import java.io.*;
- Don’t forget to try/catch everything!
  - Network communication is prone to failure
  - Cable cuts, server down, DNS failure, etc.
- Get your server names and ports straight!
  - IP address 127.0.0.1 and name localhost always refer to machine you are on
A Little Known Fact...

- The world wide web operates on a very simple and easy to use protocol that is usually hidden from the user by the web browser.
  - Client initiates a connection (socket) to server
  - Client sends the string “GET /path/file.html”
  - Server responds by transmitting the html page (as text) back to the server
- Let’s create a simple program that retrieves web pages… less than 40 lines of JAVA!

The usual prerequisites...

```java
import java.io.*;
import java.net.*;

public class GetWebPage {
  private static final int wwwport = 80;
  public static void main(String [] args) {
    if (args.length < 1) {
      System.err.println("Must specify host as an arg");
      System.exit(-1);
    }
    Socket wwsrv = null;
    InputStreamReader ISR = null;
    BufferedReader fromsrv = null;
    PrintWriter tosrv = null;
```

```
Here’s where the action is...

```java
try {
    wwwsrv = new Socket(args[0], wwwport);
    ISR = new InputStreamReader(wwwsrv.getInputStream());
    fromsrv = new BufferedReader(ISR);
    tosrv = new PrintWriter(wwwsrv.getOutputStream());
    tosrv.println("GET /")
    tosrv.flush();
    String theLine = null;
    do {
        theLine = fromsrv.readLine();
        System.out.println(theLine);
    } while (theLine != null);
}
```

Cleanup

```java
fromsrv.close();
ISR.close();
tosrv.close();
} catch (Exception e) {
    System.err.println("Could retrieve the web page.");
    System.err.println(e);
}
```

In Summary

• Communications requires standards.
• Current standards in computer networks:
  • TCP - the protocol aspect of the language
  • HTML - the agreed upon encoding of semantics
  • IP - the local naming scheme
  • MAC address - the physical naming scheme
  • The Internet - the medium
• Sockets are the low level platform independent API for networks