Networking:
Computers & Computers

Important caveat: Today’s class is a 10,000 meter view of networking. Before attempting to write your own server software, please take a full networking class or at least read a few books.
Basic Concepts:
Basic Concepts:

A few considerations:

1. Points “A” and “B” might too far apart...
2. There might be many computers in between them...
3. The links between those computers might be unreliable...

1. *Routing* within a networks
2. *Switching* between networks...
3. *Packetization* of data
Basic Vocabulary:

Host: fancy term for computer
   “hostname”, “connect to host”

Server: host that a program might wish to connect to

Client: program connecting to a particular server
Basic Vocabulary:

Socket: endpoint of a network connection

Port: point on a host that a socket is bound to

Protocol: set of rules governing traffic over a network connection
OSI Reference Model
OSI Reference Model
OSI Reference Model

- Packets
  - Network
    - Path Determination and IP (Logical Addressing)
- Frames
  - Data Link
    - MAC and LLC (Physical Addressing)
- Bits
  - Physical
    - Media, Signal, and Binary Transmission

http://www.tutorial5.com/images/stories/osi-model-7-layers.png
OSI Reference Model

- **Segments**: End-to-End Connections and Reliability
- **Packets**: Path Determination and IP (Logical Addressing)
- **Frames**: MAC and LLC (Physical addressing)
- **Bits**: Media, Signal, and Binary Transmission
OSI Reference Model
OSI Reference Model

[Diagram showing the OSI model with layers:
- Data
- Presentation: Data Representation and Encryption
- Session: Interhost Communication
- Transport: End-to-End Connections and Reliability
- Network: Path Determination and IP (Logical Addressing)
- Data Link: MAC and LLC (Physical Addressing)
- Physical: Media, Signal, and Binary Transmission]
OSI Reference Model

- Application: Network Process to Application
- Presentation: Data Representation and Encryption
- Session: Interhost Communication
- Transport: End-to-End Connections and Reliability
- Network: Path Determination and IP (Logical Addressing)
- Data Link: MAC and LLC (Physical addressing)
- Physical: Media, Signal, and Binary Transmission

http://www.tutorial5.com/images/stories/osi-model-7-layers.png
Basic Vocabulary:

IP Address: a host’s numeric address on an Internet Protocol network.

137.53.44.68

There are a few special IP addresses:

127.0.0.1 is the “loopback” address: on any computer, if you connect to 127.0.0.1, you’re connecting to yourself. The host name “localhost” points to the same place.
Basic Vocabulary:

IP Address: a host’s numeric address on an Internet Protocol network.

137.53.44.68

There are a few special IP addresses:

192.168.x.x
172.16-32.x.x
10.x.x.x

These are “private” IP blocks that are not for public routing, and are often used for internal networks within a home or office.
Basic Vocabulary:

IP Address: a host’s numeric address on its network.

137.53.44.68

DNS Server: Domain Name System responsible for mapping host names to IP addresses.

skynet.ohsu.edu -> 137.53.250.53
A Simple Python Server:

```python
from socket import *
host = ''  # short-hand for 'localhost'
port = 50007
sock = socket(AF_INET, SOCK_STREAM)
sock.bind((host, port))  # takes a tuple
sock.listen(5)

while 1:
    connection, address = sock.accept()
    print 'Server connected by ', address
    while 1:
        data = connection.recv(1024)
        if not data: break
        print 'Client said: ' + data
        connection.send('You said: ' + data)
    connection.close()
print "closed..."
```
Networking in Python:

[obsidian:~/]% python echo_server.py
Server connected by ('127.0.0.1', 64283)
client said: hello world

closed...
Server connected by ('127.0.0.1', 64286)
client said: hello, again

closed...
Simple Networking tools:

**Netcat (nc):**

```
[obsidian:~] steven% echo 'hello world' | nc localhost 50007
You said: hello world
[obsidian:~] steven% xt
```

**Telnet (telnet):**

```
[obsidian:~] steven% telnet localhost 50007
Connected to localhost.
Escape character is '^]'.
Hello, world!
You said: Hello, world!
Hello, again!
You said: Hello, again!
Connection closed.
```
A Simple Python Client

from socket import *
host = 'localhost'
port = 50007

sock = socket(AF_INET, SOCK_STREAM)
sock.connect((host, port))  # takes a tuple

message = ['From a python client!','Another line!']

for line in message:
    sock.send(line)
    data = sock.recv(1024)
    print 'Server said: ' + data

sock.close()
A Simple Python Client

[obsidian:~/]% python echo_client.py
Server said: You said: From a python client!
Server said: You said: Another line!
In real life:

This is an overly simplistic example, especially at the server side.

Python provides several much more robust networking approaches; consult a quality textbook.
In real life:

Another important thing to know: using this approach, everything sent over the wire is unencyrypted...

... therefore, do not — under any circumstances — use this technique to work with passwords, credit card data, HIPAA-covered PHI, etc.
In real life:

Luckily for us, most network tasks we’d care to accomplish have established protocols...

... and Python has libraries to use those protocols!
In real life:

... and Python has libraries to use those protocols!

<table>
<thead>
<tr>
<th>Protocol</th>
<th>Description</th>
<th>Library</th>
</tr>
</thead>
<tbody>
<tr>
<td>HTTP</td>
<td>WWW traffic</td>
<td>httplib</td>
</tr>
<tr>
<td>FTP</td>
<td>File transfers</td>
<td>ftplib</td>
</tr>
<tr>
<td>SMTP</td>
<td>Sending email</td>
<td>smtplib</td>
</tr>
<tr>
<td>POP3</td>
<td>Recv. email</td>
<td>poplib</td>
</tr>
<tr>
<td>IMAP4</td>
<td>Recv. email</td>
<td>imaplib</td>
</tr>
</tbody>
</table>
A more interesting example:

The Hypertext Transfer Protocol is very simple:
An HTTP Request:

Verb: GET
URL: /HTTP/1.1
Version: HTTP/1.1

Headers:
- Host: www.google.com

Double blank line
An HTTP Response:

```
<!doctype html><html><head>...
```

**Status code**

HTTP/1.1 200 OK
Date: Fri, 20 Nov 2009 21:49:06 GMT
Content-Type: text/html; charset=ISO-8859-1
Server: gws
Transfer-Encoding: chunked

```
<!doctype html><html><head>...
```