

# The Internet

- Internet Protocol (IP)
  - each computer has an IP address
    - e.g. 130.65.86.66
    - 4 bytes
    - 127.0.0.1 is localhost
    - 10.x.x.x is in a local network
  - routers send packets to their destination
- Transmission Control Protocol (TCP)
  - used to automatically resend lost data
  - usually used together with IP: TCP/IP
  - port numbers (16 bit unsigned)
    - Web uses port 80
    - SSH uses port 22
- Network Address Translation (NAT)
  - used at BSU

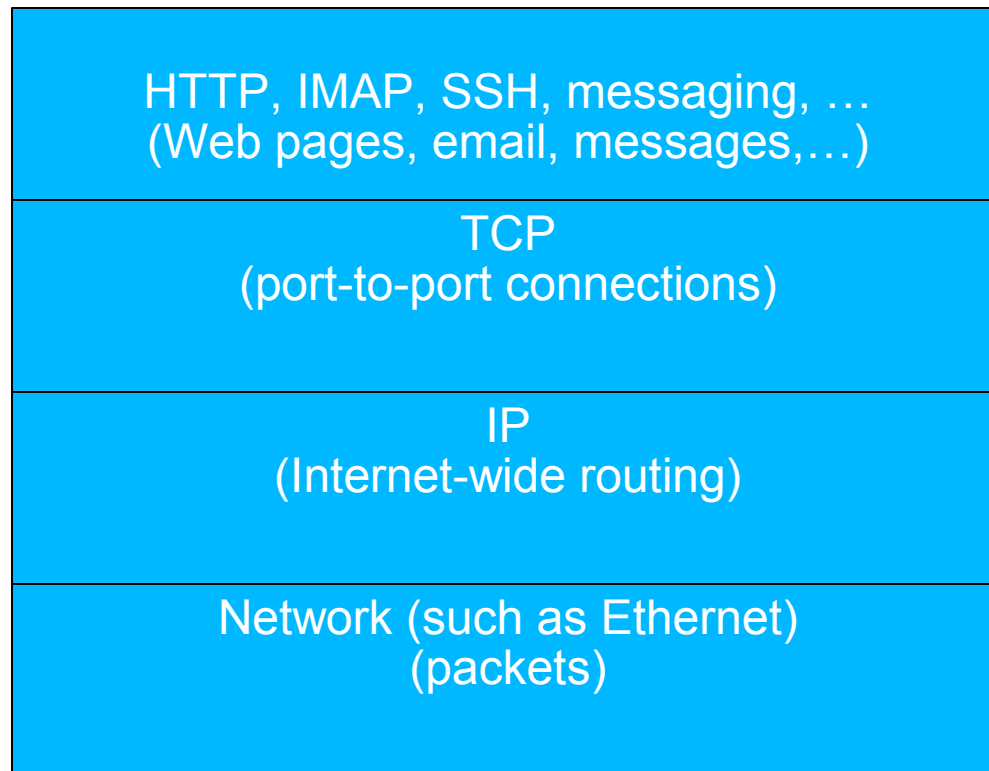
# Internet Addresses

- IP addresses
- Domain names
  - human readable
  - example: `www.bsu.edu`
  - DNS (Domain Name System)
    - translates domain names into IP addresses
- Uniform Resource Identifiers (URI)
  - also known as Uniform Resource Locator (URL)
  - example: `http://www.bsu.edu/cs/`
    - `http` protocol
    - `www.bsu.edu` domain name
    - `/cs/` path
    - port may be specified after domain (`:80`)

# Internet Applications

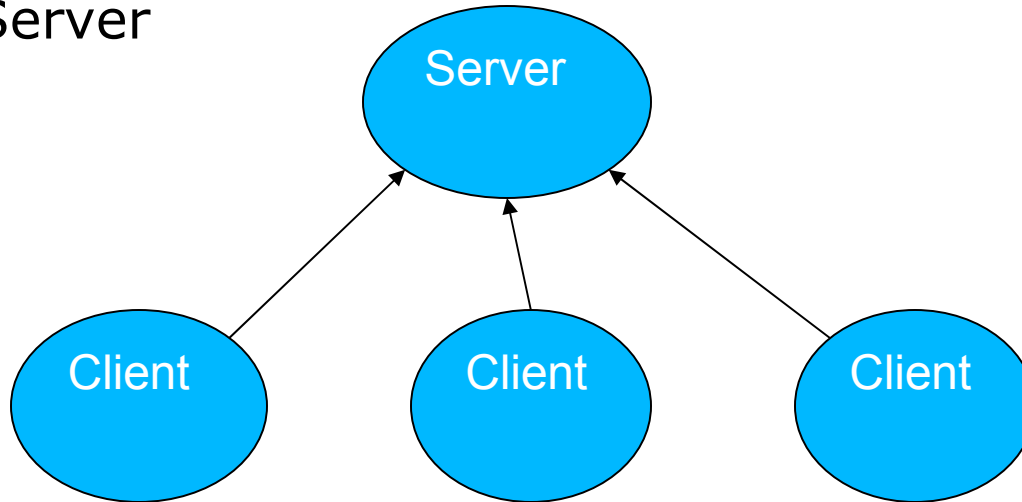
- Email
  - Post Office Protocol (POP)
  - Simple Mail Transfer Protocol (SMTP)
  - Internet Message Access Protocol (IMAP)
- SSH/Telnet
- FTP/SFTP
- Web
  - Hypertext Transfer Protocol (HTTP)
  - for transmitting Web pages
- Applications are usually on top of TCP/IP
  - design their own protocol
  - what's a protocol?

# Internet Application Stack

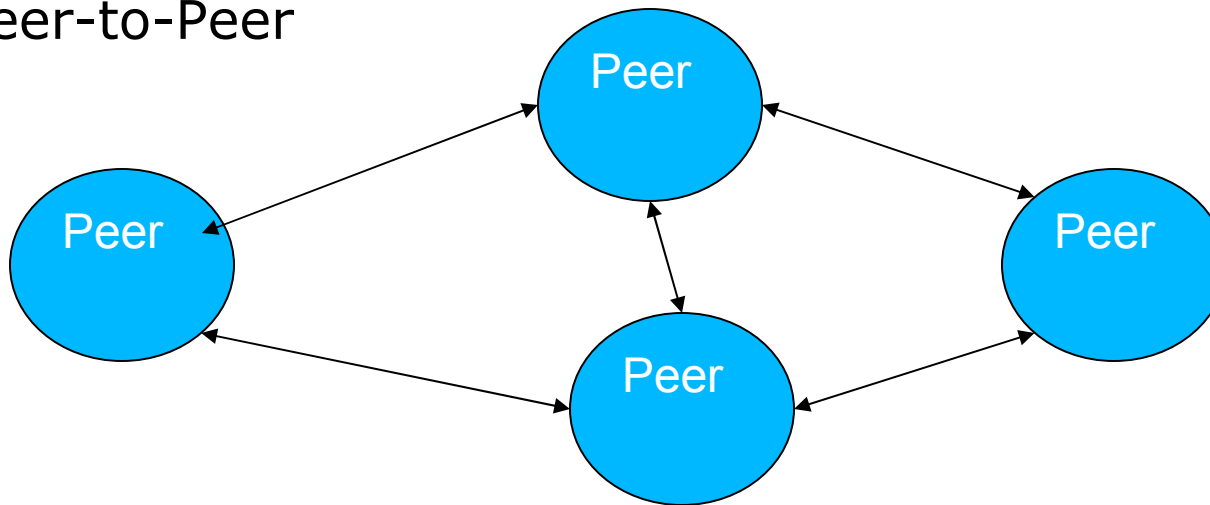


# Architectures

- Client-Server



- Peer-to-Peer



# Sockets

- Socket
  - one end of a TCP connection
  - java.net.Socket

```
try {
    Socket s = new Socket( hostname, port );
    InputStream inputStream = s. getInputStream();
    Scanner in = new Scanner( inputStream );
    String line = in. nextLine();
    //...
    s. close();
} catch( IOException ex ) {
    //...
}
```

# WebGet: Talking to a Web Server

```
final int HTTP_PORT = 80;
Socket s = new Socket( host, HTTP_PORT );

InputStream instream = s. getInputStream();
OutputStream ostream = s. getOutputStream();

Scanner in = new Scanner( instream );
PrintWriter out = new PrintWriter( ostream );

String command = "GET " + resource + " HTTP/1.0\n\n";
out. print( command );
out. flush();

while( in. hasNextLine() )
{
String input = in.nextLine();
System. out. println( input );
}
s. close();
```

# Testing Internet Connections

- Ping
  - tests if a host is reachable  
`ping www.bsu.edu`
- Traceroute
  - gives information about hops and delays  
`traceroute www.bsu.edu` (Linux)  
`tracert www.bsu.edu` (Linux)  
`tracert www.bsu.edu` (MS Windows)
- Finding out your IP address
  - `ifconfig` (Linux)
  - `ipconfig /all` (MS Windows)

# Accepting a Connection

```
final int portNumber = 8888;  
ServerSocket server = new ServerSocket( portNumber );  
Socket s = server.accept();  
//continue reading and writing socket as before
```

- `ServerSocket.accept()` blocks
  - i.e.: waits until someone requests a connection
- Often used with threads
  - threads allow several connections to be handled at once