Objective

Network Programming

- To understand the concept of sockets
- To learn how to send and receive data through sockets
- To implement network clients and servers
- To communicate with web servers and server-side applications.
- Application level protocol

Study Chapter 23 of BiG Java (Web Only): Early Object 6th .

Network

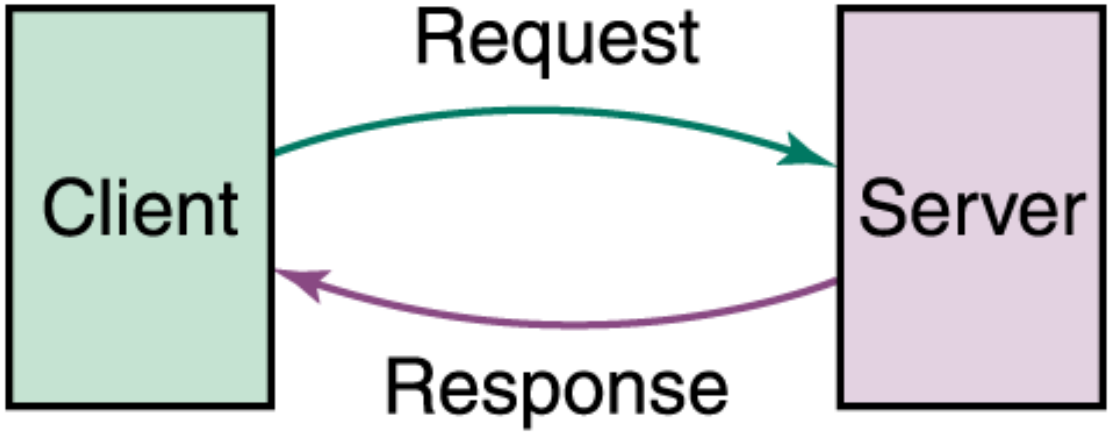
A computer network is a collection of nodes all connected together so they can share information, or *talk* to each other.

The generic term node or host refers to any device on a network

Network of computers is like network of highways.
If one bridge washed out, traffic could always find an alternate route to reach its destination.

Client/ Server

Computer networks have opened up an entire frontier in the world of computing called the **client/server model**



Protocols

- Computer needs some kind of communication software.
- To communicate with each other, two machines must follow the same Protocol.
- Protocol is a set of rules for the exchange of data between computers.

Internet Protocol

Internet Protocol (IP): A collection of rules to be used for governing the transmission of information from a computer to another.

- Each computer on the network has a unique address known as IP address
- Each address has four parts that are separated by dots.
- Each part can be a number between 0 to 255
- For example: 202.47.104.3

TCP/IP

- The protocol at the heart of the Internet is TCP/IP.
- TCP/IP (Transmission Control Protocol/ Internet Protocol).
- TCP/IP became the "language" of the Internet, allowing cross-network communication for almost every type of computer and network.
- TCP protocols define a system similar to postal systems.

IP Protocol

- When a message is sent to the Internet, It is broken into packets in the same way you might pack your belongings in several individually addressed box.
- Each packet has all the information to travel to its destination.
- Different packets may take different routes.
- In the destination the packets are reassembled to its original packets by TCP protocol.

IP Protocol

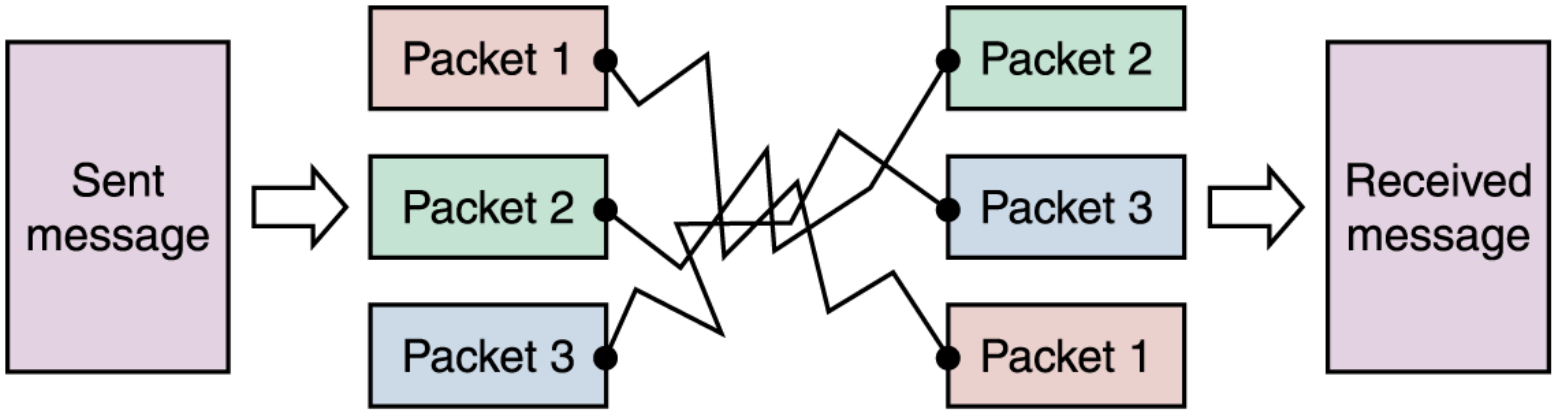
Each message, (text, picture, sound file, or anything else that can be represented as digital form) would broken up into Packets of equivalent size to about 1500 bytes.

Each packet has its source and destination IP address, along with the number indication of the packet number that shows which part of the message this packet represents.

For example:

FROM:	202.47.104.3 The IP address of the source(Sender)
TO:	101.78.44.19 The address of the destination (Receiver)
NUMBER:	14 of 27 The Packet number in the message

Packets



Message is divided into packets

Packets are sent over the Internet by the most expedient route

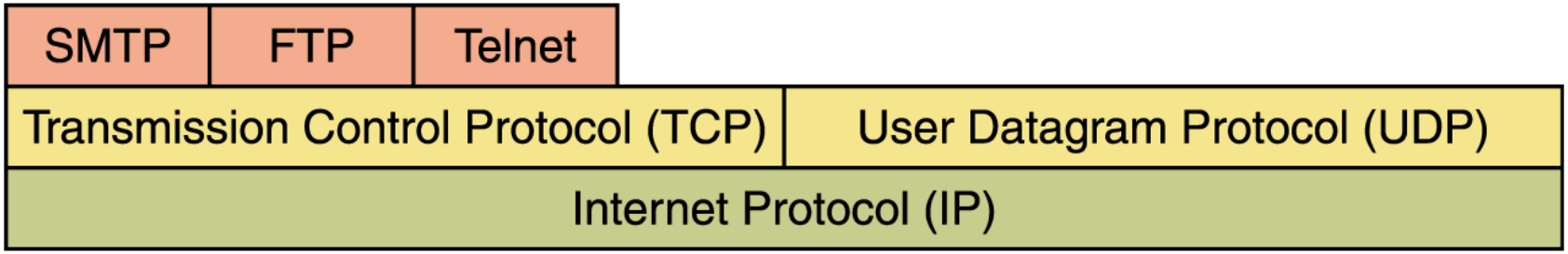
Packets are reordered and then reassembled

Transmission Control Protocol

- Internet Protocol (IP) does not notify the sender if data is lost or garbled
- This is the job of a higher level protocol *Transmission Control Protocol* (TCP)
- The most commonly used Internet services use TCP with IP (TCP/IP)

Protocol Stack

- Network protocols are layered such that each one relies on the protocols that underlie it
- Sometimes referred to as a **protocol stack**



DNS

DNS (Data Name Server)

- Every computer has an IP address
- We use IP address to access the computer.

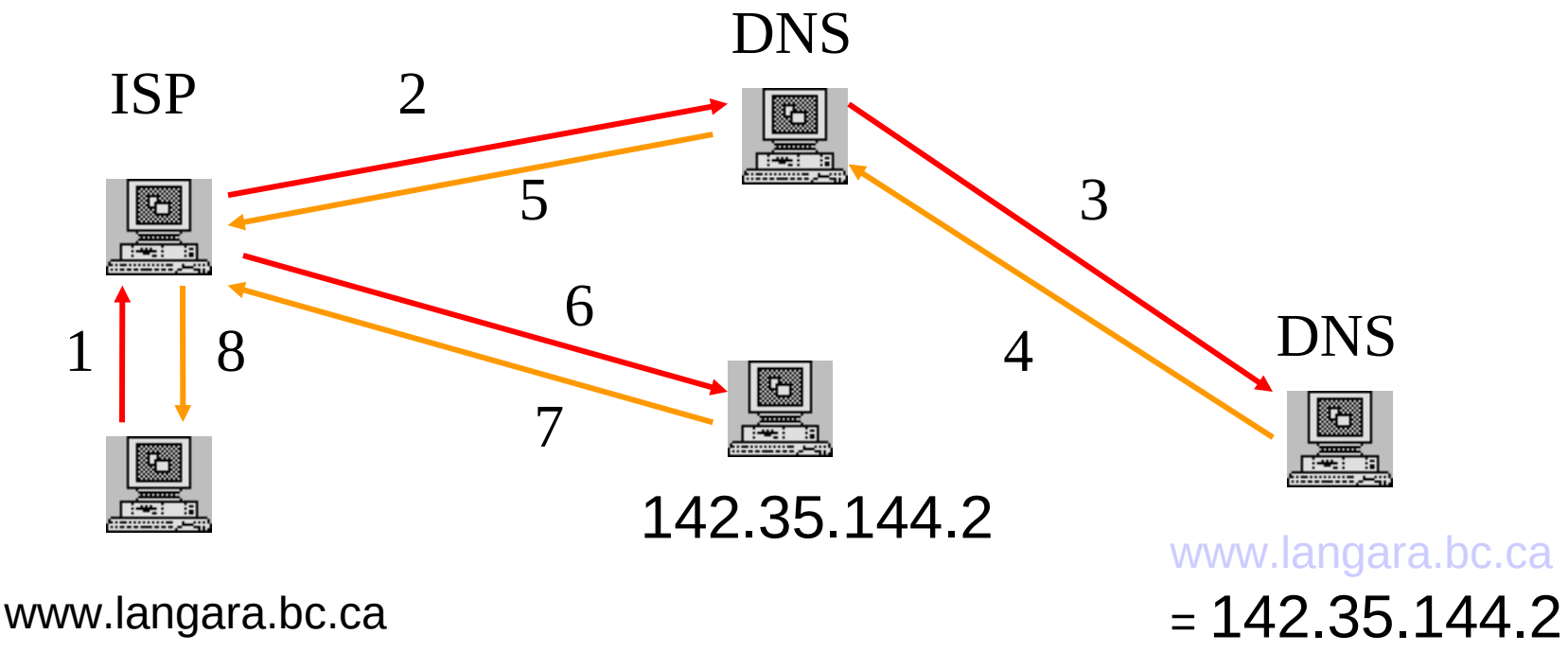
E.g: the IP address of the computer that host

langara.ba.ca is 142.35.144.2

But how computers know about this IP address?

DNS

Data Name Server are computers that stores name and IP address of the servers.



Port Numbers

One computer can offer multiple services over the Internet

- For example, both a web server and an email server

When data are sent to that computer, they need to indicate which program is to receive the data

Continued

Port Numbers

- IP uses *port numbers* for this
  - A port number is an integer between 0 and 65,535
  - The sending program must know the port number of the receiving program
  - This number is included in the transmitted data

Contents of TCP Packet

- The Internet address of the recipient
- The port number of the recipient
- Internet address of the sender
- The port number of the sender

Self Check

- What is the difference between an IP address and a domain name?
- Why do some streaming media services not use TCP?

Answers

- An IP address is a numerical address, consisting of four or sixteen bytes. A domain name is an alphanumeric string that is associated with an IP address.
- TCP is reliable but somewhat slow. When sending sounds or images in real time, it is acceptable if a small amount of the data is lost. But there is no point in transmitting data that is late.

A Client Program – Sockets

- A socket is an object that encapsulates a TCP/IP connection
- There is a **socket** on both ends of a connection
- Syntax to create a socket in a Java program

```
Socket s = new Socket(hostname, portnumber);
```

Continued

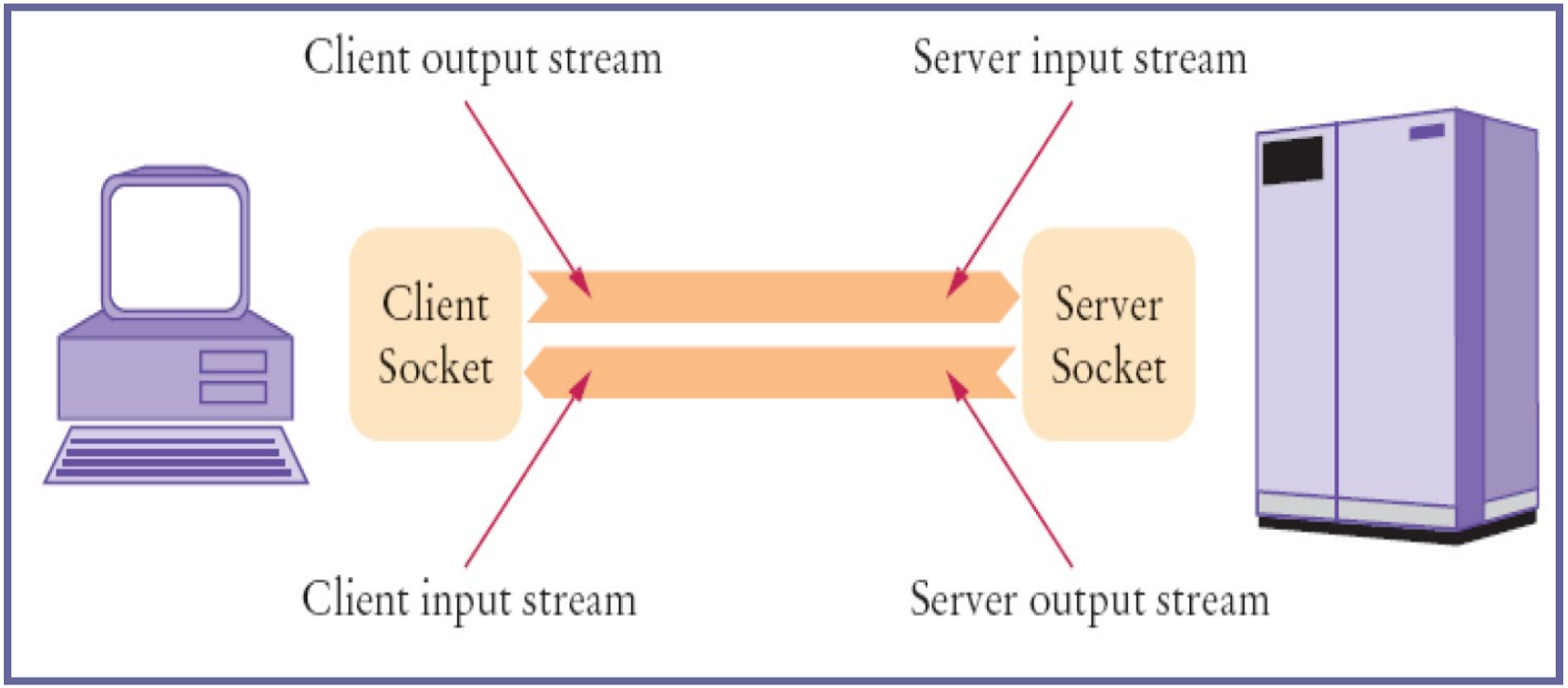
A Client Program – Sockets

Code to connect to the HTTP port of server,

```
final int HTTP_PORT = 80;
Socket s = new Socket("java.sun.com", HTTP_PORT);
```

If it can't find the host, the **`Socket`** constructor throws an **`UnknownHostException`**

Client and Server Sockets



A Client Program – Input and Output Streams

- Use the input and output streams attached to the socket to communicate with the other endpoint
- Code to obtain the input and output streams

```
InputStream instream = s.getInputStream();
OutputStream outstream = s.getOutputStream();
```

Continued

A Client Program – Input and Output Streams

- When you send data to **`outstream`**, the socket forwards them to the server
- The socket catches the server's response and you can read it through **`instream`**
- When you are done communicating with the server, close the socket

```
s.close();
```

A Client Program – Scanners and Writers

- **`InputStream`** and **`OutputStream`** send and receive bytes

  To send and receive text, we can use a scanner and a Printwriter

```
Scanner in = new Scanner(instream);
PrintWriter out = new PrintWriter(outstream);
```

Continued

A Client Program – Scanners and Writers

A **`PrintWriter`** *buffers* the characters and only sends when the buffer is full

- Buffering increases performance

When sending a command, you want the whole command to be sent now

- *Flush* the buffer manually:

```
out.print(command);
out.flush();
```

Retrieve Web Content

- This program lets you retrieve any item from a web server
- You specify the host and item from the command line
- For example:

```
java WebGet java.sun.com /
```

The "/" denotes the root page of the web server that listens to port 80 of `java.sun.com`

Continued

Retrieve Web Content

1. Establishes a connection to the host
2. Sends a `GET` command to the host
3. Receives input from the server until the server closes its connection

Check Example 1: WebGet.java

Retrieve Web Content using URLConnection

- Alternatively you can use URLConnection to retrieve web content.
- You do not need to send HTTP commands using URLConnection.
- `URLConnection` Class
  - Provides convenient support for HTTP
  - Can also handle FTP *(file transfer protocol*)
  - Takes care of socket connection for you
  - Makes it easy to communicate with a web server without giving HTTP commands

URL Connections

Construct a `URL` object from a URL starting with the http or ftp prefix

```
URL u = new URL("http://java.sun.com/");
```

Use the URL's **`openConnection()`** method to get the **`URLConnection`**

```
URLConnection connection = u.openConnection();
```

URL Connections

- Call the **`getInputStream`** method to obtain an input stream

```
InputStream instream = connection.getInputStream();
```

- You can turn the stream into a scanner in the usual way

Check Example 1 : URLc.java

Server Side Programming

- The server waits for clients to connect on a certain port. For example port: 8888
- To listen for incoming connections, use a server socket
- To construct a server socket, provide the port number

```
ServerSocket server = new ServerSocket(8888);
```

- Use the accept method to wait for client connection and obtain a socket

```
Socket s = server.accept();
```

Communication between client and server

Scanner and **PrintWriter** uses **tex**t format to sent and receive data.

Sending a number as a text is not efficient.

For example to send number like 20093456, you need to sent/receive 14 bytes of data.

However, 20093456 is an integer number, and can be handled by only four bytes of data in binary format.

As the result we **do not** generally use Scanner or PrintWriter for network communication.

Communication between client and server

Instead we use

DataOutputStream to send data

DataInputStream to receive data

between client and server.

```
DataOutputStream out = new
   DataOutputStream(client.getOutputStream());
DataInputStream in = new
   DataInputStream(client.getInputStream());
```

Study Java API for both classes

Protocol

We also need an application level protocol for communication between two peers of nodes in network.

Example:

```
public interface Protocol{
   int NUMBER = 1;
   int DATA = 2;
   int SUCCEED = 3;
   int INVALID = 4;
   int RESULT = 5;
   int QUIT = 6;
}
```

Client		Server
DATA message	Message: String	SUCCEED, INVALID
NUMBER n	n: double	RESULT n, INVALID
QUIT		Close connection

Examples

Check example 2 : A simple one-way client server connection

example 3 :

A server generally responses for multiple clients simultaneously.

To achieve this goal we span a thread for each client.

Example 4

Check example 4 : Two way communication between client and server

Check example 5 :Sample server program: enables clients to manage bank accounts in a bank

Self Check

- Why didn't we choose port 80 for the bank server?
- Can you read data from a serverSocket?

Self Check

- Why is it better to use an `URLConnection` instead of a socket when reading data from a web server?
- What happens if you use the `URLGet` program to request an image (such as `http://java.sun.com/im/logo_java.gif`)?

Answers

- The `URLConnection` class understands the HTTP protocol, freeing you from assembling requests and analyzing response headers.
- The bytes that encode the images are displayed on the console, but they will appear to be random gibberish.

Summary

The Internet is a worldwide collection of nodes using a common set of protocols.

TCP/IP is the abbreviation for Transmission Control Protocol and Internet Protocol.

TCP/IP designed to establish reliable transmission of data between two computers on the Internet.

A TCP connection requires the Internet addresses and port numbers of both end point.

A socket is an object that encapsulates a TCP connection.

When transmission over a socket is complete, remember to close the socket.

Summary

For text protocols, turn the socket streams into scanners and writers.

Flush the writer attached to a socket at the end of every command. Then the command is sent to the server, even if the writer's buffer is not completely filled.

The ServerSocket class is used by server applications to listen for client connections.

Use the URLConnection class to read data from a web server.

The URLConnection class makes it easy to communicate with a web server without having to issue HTTP commands.