Advanced Computer Programming

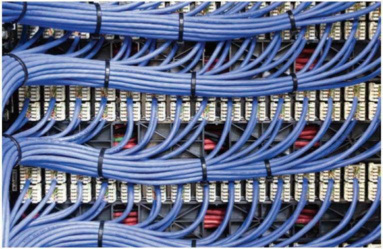
[Lecture 13]

Saeed Reza Kheradpisheh

kheradpisheh@ut.ac.ir

Department of Computer Science
Shahid Beheshti University
Spring 1397-98

Internet Programming



In this chapter, you will see what goes on “under the hood” when you send an e-mail message or when you retrieve a web page from a remote server. You will also learn how to write programs that fetch data from sites across the Internet and how to write server programs that can serve information to other programs.

The Internet Protocol

Computers can be connected with each other through a variety of physical media:

- Computers connected by network cabling.
- Computers connected via Internet network.

The Internet Protocol

Computers can be connected with each other through a variety of physical media:

- Computers connected by network cabling.
- Computers connected via Internet network.

Transmited data are of two kinds:

- Application Data: the data that one computer actually wants to send to another.
- Network Protocol Data: the data that describe how to reach the intended recipient and how to check for errors and data loss in the transmission.

The Internet Protocol

Computers can be connected with each other through a variety of physical media:

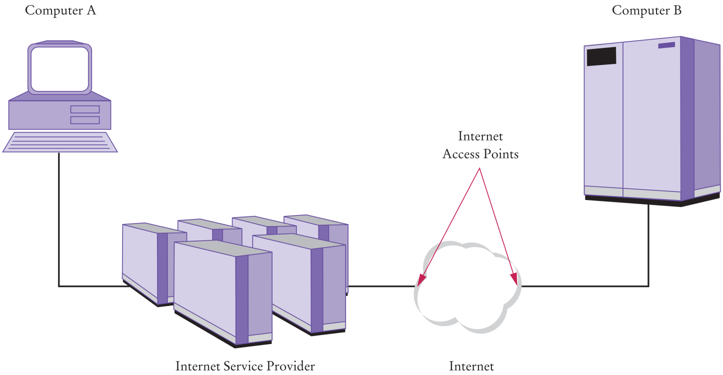
- Computers connected by network cabling.
- Computers connected via Internet network.

Transmited data are of two kinds:

- Application Data: the data that one computer actually wants to send to another.
- Network Protocol Data: the data that describe how to reach the intended recipient and how to check for errors and data loss in the transmission.

TCP/IP:
The protocol data follow certain rules set forth by the Internet Protocol Suite.

Connection via Internet



Internet itself is a complex collection of pathways on which a message can travel from one Internet access point to another.

Internet Protocol (IP) address

- For the data to arrive at its destination, it must be marked with a destination address.

Internet Protocol (IP) address

- For the data to arrive at its destination, it must be marked with a destination address.
- In IP, addresses are denoted by sequences of four numbers, each one byte, for example: 130.65.86.66

Internet Protocol (IP) address

- For the data to arrive at its destination, it must be marked with a destination address.
- In IP, addresses are denoted by sequences of four numbers, each one byte, for example: 130.65.86.66
- In order to send data, A needs to know the Internet address of B and include it in the protocol portion when sending the data across the Internet.

Internet Protocol (IP) address

- For the data to arrive at its destination, it must be marked with a destination address.
- In IP, addresses are denoted by sequences of four numbers, each one byte, for example: 130.65.86.66
- In order to send data, A needs to know the Internet address of B and include it in the protocol portion when sending the data across the Internet.
- The routing software that is distributed across the Internet can then deliver the data to B.

Internet Protocol (IP) address

- For the data to arrive at its destination, it must be marked with a destination address.
- In IP, addresses are denoted by sequences of four numbers, each one byte, for example: 130.65.86.66
- In order to send data, A needs to know the Internet address of B and include it in the protocol portion when sending the data across the Internet.
- The routing software that is distributed across the Internet can then deliver the data to B.
- On the Internet, computers can have socalled Domain Names that are easier to remember than IP address.

Internet Protocol (IP) address

- For the data to arrive at its destination, it must be marked with a destination address.
- In IP, addresses are denoted by sequences of four numbers, each one byte, for example: 130.65.86.66
- In order to send data, A needs to know the Internet address of B and include it in the protocol portion when sending the data across the Internet.
- The routing software that is distributed across the Internet can then deliver the data to B.
- On the Internet, computers can have socalled Domain Names that are easier to remember than IP address.
- A special service called the Domain Name System (DNS) translates between domain names and Internet addresses.

IP

- One interesting aspect of IP is that it breaks large chunks of data up into more manageable packets.

IP

- One interesting aspect of IP is that it breaks large chunks of data up into more manageable packets.
- Each packet is delivered separately, and different packets that are part of the same transmission can take different routes through the Internet.

IP

- One interesting aspect of IP is that it breaks large chunks of data up into more manageable packets.
- Each packet is delivered separately, and different packets that are part of the same transmission can take different routes through the Internet.
- Packets are numbered, and the recipient reassembles them in the correct order.

IP

- One interesting aspect of IP is that it breaks large chunks of data up into more manageable packets.
- Each packet is delivered separately, and different packets that are part of the same transmission can take different routes through the Internet.
- Packets are numbered, and the recipient reassembles them in the correct order.
- If some data get lost or garbled in the process, IP has safeguards built in to make sure that the recipient is aware of that unfortunate fact and doesnt rely on incomplete data.

Transmission Control Protocol (TCP)

- This protocol attempts reliable delivery of data, with retries if there are failures, and it notifies the sender whether or not the attempt succeeded.

Transmission Control Protocol (TCP)

- This protocol attempts reliable delivery of data, with retries if there are failures, and it notifies the sender whether or not the attempt succeeded.
- Exceptions are streaming media services, which bypass the slower TCP for the highest possible throughput and tolerate occasional information loss.

Transmission Control Protocol (TCP)

- This protocol attempts reliable delivery of data, with retries if there are failures, and it notifies the sender whether or not the attempt succeeded.
- Exceptions are streaming media services, which bypass the slower TCP for the highest possible throughput and tolerate occasional information loss.
- TCP is independent of the Internet Protocol; it could in principle be used with another lower-level network protocol.

Transmission Control Protocol (TCP)

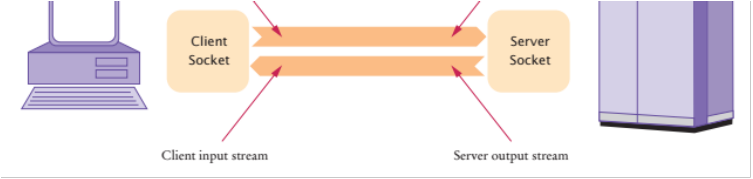
- This protocol attempts reliable delivery of data, with retries if there are failures, and it notifies the sender whether or not the attempt succeeded.
- Exceptions are streaming media services, which bypass the slower TCP for the highest possible throughput and tolerate occasional information loss.
- TCP is independent of the Internet Protocol; it could in principle be used with another lower-level network protocol.
- However, in practice, TCP over IP (often called TCP/IP) is the most commonly used combination.

Transmission Control Protocol (TCP)

- This protocol attempts reliable delivery of data, with retries if there are failures, and it notifies the sender whether or not the attempt succeeded.
- Exceptions are streaming media services, which bypass the slower TCP for the highest possible throughput and tolerate occasional information loss.
- TCP is independent of the Internet Protocol; it could in principle be used with another lower-level network protocol.
- However, in practice, TCP over IP (often called TCP/IP) is the most commonly used combination.
- When data are sent to that computer, they need to be marked so that they can be forwarded to the appropriate program. TCP uses port numbers for this purpose.

A Client Program

- We will see how to write a Java program that establishes a TCP connection to a server, sends a request to the server, and prints the response.
- In the terminology of TCP/IP, there is a socket on each side of the connection.



- In Java, a client establishes a socket with a call:

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

A Client Program

- The socket constructor throws an `UnknownHostException` if it can't find the host.
- Once you have a socket, you obtain its input and output streams:

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

- When you are done communicating with the server, you should close the socket:

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

- If you want to communicate with the server by sending and receiving text, you should turn the streams into scanners and writers, as follows:

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

A Client Program

- A print writer *buffers* the characters that you send to it.
- If you are communicating with a server that responds to requests, you want to make sure that the server gets a complete request at a time, you need to flush the buffer manually:

  ```
  out.print(command);
  out.flush();
  ```
- The flush method empties the buffer and forwards all waiting characters to the destination.

A Server Program

- The server program waits for clients to connect to a particular port.
- This port number should not be preassigned to another service.
- To listen to incoming connections, you use a server socket:

  ```
   ServerSocket server = new
  ServerSocket(PortNumber);
  ```
- The `accept` method of the `ServerSocket` class waits for a client connection:

  ```
   Socket s = server.accept();
  ```
- Now, the server program can send and receive messages from the client.

Example 1

Write a message passing client and server program.

Example 2

Write a Bank server and Client program.