

# Lab 11: Socket Programming

## What you will do:

- Set up the local network with two laptops and a Linksys router
- Implement Server and Client java program
- Use Wireshark to capture/explore/analyze communication packets.

## Things that you will need to know or learn:

- Everything that you learned in previous labs you will need to complete this lab.
- Understand sockets and ports
- Create comprehensive network applications using sockets

## What you need to submit and when:

- InLab Activities, including instructor signoffs before the end of your lab period (refer to the instructions below). This part is to be completed **with a partner**.

## Required Equipment and Software:

- Equipment requirements:
  - Network cables: two straight-through
  - One Linksys router
  - Wireshark installed and working on your laptop (done in Lab 01)
  - Lab 10 documents downloaded to your laptop
  - Two laptops
- Programming IDE and software:
  - Eclipse for Java
  - Wireshark

## References and Resources:

- Lab 01 - 10
- Cisco Chapter 7-10
- Chapter 28 Networking from “Java How to Program”
- Socket programming reference document

## TCP/IP SOCKET PROGRAMMING

The two key classes from the **java.net** package used in creation of server and client programs are:

- ServerSocket
- Socket

A server program creates a specific type of socket that is used to listen for client requests (server socket), in the case of a connection request, the program creates a new socket through which it will exchange data with the client using input and output streams. The socket abstraction is very similar to the file concept: developers have to open a socket, perform I/O, and close it.

A simple client program normally needs the following steps to communicate with a server program:

1. Create a Socket Object:

```
Socket client = new Socket(server, port_id);
```

2. Create I/O streams for communicating with the server.

```
is = new DataInputStream(client.getInputStream());  
os = new DataOutputStream(client.getOutputStream());
```

3. Perform I/O or communication with the server:

```
Receive data from the server: String line = is.readLine();  
Send data to the server: os.writeBytes("Hello\n");
```

4. Close the socket when done:

```
client.close();
```

## Task 0: Preparations

0.1 Find a partner to work with.

0.2 Confirm you have downloaded the following from BB “Labs -> Lab 11” to your computer:

0.2.1 CST8109 “Lab 11 – In-Lab Activities.pdf” (this document)

0.3 Disable the Wireless Network Interface of your Laptop computer. Your only connection to the network must be via the Ethernet (wired) interface.

0.4 Do not start until you have completed ALL steps in this task.

## Task 1: Build Network with Linksys Router

In this task you will build and test a network which consists of three physical devices: your laptop, your partner’s laptop and a Linksys router. Reset the router back to factory defaults.

## Task 2: Write the Server/Client program

In this task, you will write two simple server/client programs: SimpleServer and SimpleClient. SimpleServer creates a socket on port **1254**. SimpleClient establishes a connection to the server, read the message sent by the server and display it on the console. You can assign roles to each laptop. You will need one laptop to play the role of client and the other the role of server. Run the SimpleServer code on the server and SimpleClient on the client.

## Task 3: Wireshark Capture Analysis

In this task you will capture the traffic resulting from socket communication between the server and the client. In particular you will focus on the HTTP and TCP PDU.

1. Start a Wireshark capture on your laptop.
2. Run the code SimpleServer on the server laptop.
3. From the Command Prompt window, enter the following command:  

```
netstat -a | findstr 1254
```

  
Take a screen capture and save it as **Netstat\_TCP**.
4. Run the code SimpleClient on the client laptop.
5. Filter Wireshark capture by tcp.
6. Stop and save your capture as **WS-task3**.

7. You will now examine the capture to ensure that the desired traffic has been captured.

#### **Task 4: Demo, Cleanup and Other Tasks**

1. Demo your Wireshark captured file and Java code to your lab instructor.
2. Re-enable your firewall
3. Re-enable your Wireless Network and confirm you are able to access the College network.
4. Return the borrowed equipment and cables to your instructor.

Instructor Signoff \_\_\_\_\_