Lab 1

CS 640

LEARNING OBJECTIVES

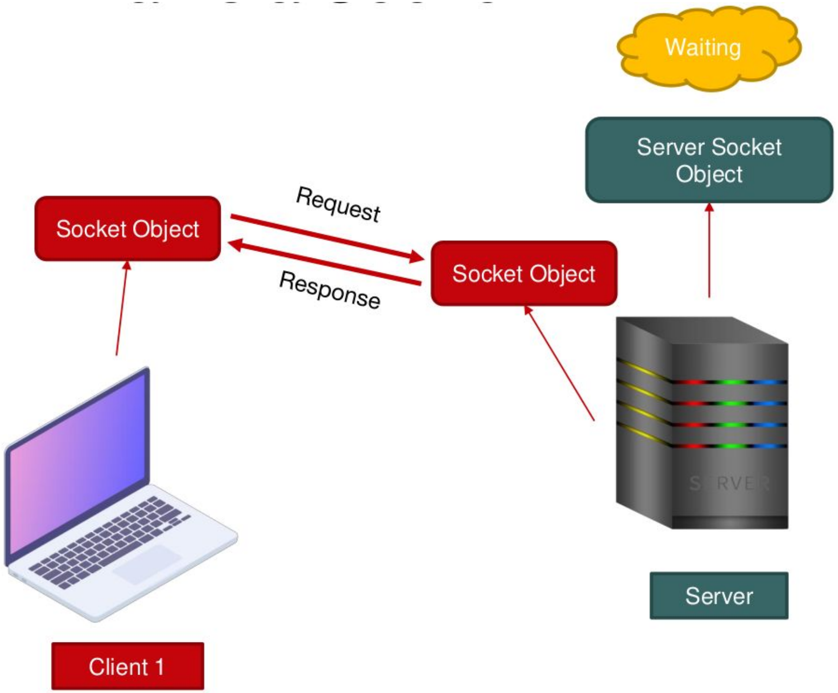
- Write applications that use sockets to transmit and receive data across a network
- Describe how latency and throughput can be measured
- Explain how latency and throughput are impacted by link characteristics and multiplexing

THINGS TO WATCH OUT FOR

- Use Java >= 8 and Java Sockets
- Work in a team of 1-2
- Use correct units (Bytes in KB and rate in Mbps)
- Use proper documentation practices within code
- Write a README with names, CS usernames, and assumptions made
- Write a Makefile to compile everything.

WHAT IS A SOCKET?

Sockets allow communication between two different processes on the same or different machines..



METHODS IN SOCKET API

Client

```
Socket socket = new Socket (address, port); // Create Socket
PrintWriter out = new PrintWriter (
    socket.getOutputStream(), true); // Create writer
out.println(msg); // Write message
out.close(); socket.close(); // Close connection
```

METHODS IN SOCKET API

Server

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

Socket socket = server.accept(); // Listen for a connection

br = new BufferedReader(new
InputStreamReader(socket.getInputStream())); // Create BufferedReader

line = br.readline(); // Read message

in.close(); socket.close(); server.close(); // Close connection
```

PART 1: WRITE `Iperfer`

Client

```
java Iperfer -c -h <server hostname> -p <server port> -t
<time>
```

Server

```
java Iperfer -s -p <listen port>
```

Communication

- Clients sends data (1000 bytes of 0x0) in chunk continuously
- Server reads data in 1000 byte chunks
- Close connection and print total bytes sent/received (KB) and rate (Mbps)

Error Messages

- If arguments are Invalid
- If port is not between 1024 and 65535

PART 1: WRITE `Iperfer`

Testing Wired

- On two machines connected by cable. (e.g. CSL)
- Take screenshots of measurements

Testing Wireless

- On two machines connected to the same wireless router (can be UW Net)
- Take screenshots of measurements

Predict throughput in two environments and verify.

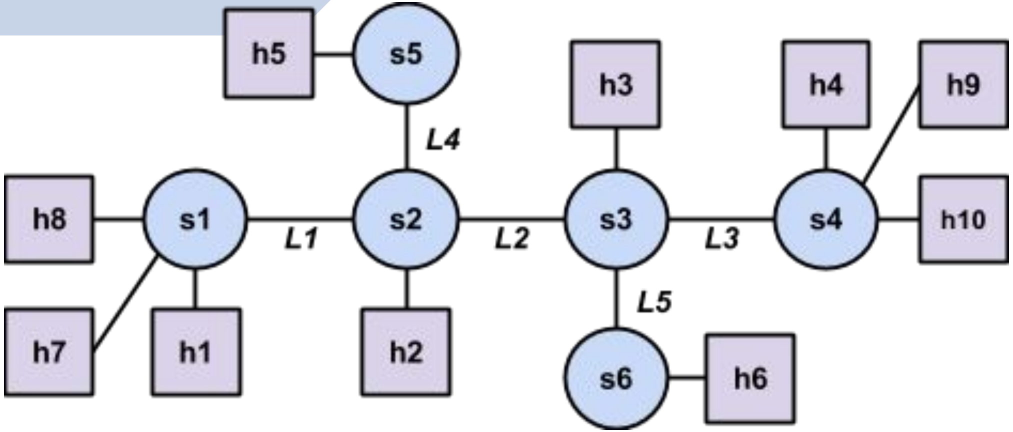
PART 2: MININET TUTORIAL

- Use your personal machines or CSL
- Follow instructions in description at http://mininet.org/vm-setup-notes/
- Download a VM launcher such as VirtualBox
- Install Java >=8 on mininet VM
- Follow the walkthrough and complete tasks in the lab description

PART3: MEASUREMENTS IN MININET

Measure:

- Use host endpoints
- Run ping with 30 packets
- Run Iperfer for 30 seconds



Conditions:

- Link latency and throughputs: L1, L2, … , L5
- Path latency and throughputs: between H1 and H4
- Path latency and throughputs with **multiplexing**:

  What if multiple hosts connected to s1 want to simultaneously talk to hosts connected to s4?

- Path latency and throughputs with **link sharing**:

  What if h1 wants to communicate with h4 at the same time h5 wants to communicate with h6?

Prediction & Evaluation:

- Expected values? Why?
- Compare measured values with expected values

RUBRIC

Section 1: Prerequisites (5 points)

- 1 - Correct submission format
- 2 - Code compiles
- 1 - Code is properly structured
  - 0.5 - Code has useful comments
  - 0.5 - Code has proper indentation
- 1 - Command takes valid inputs

RUBRIC

Section 2: Client (10 points)

- 2 - args parsed correctly
  - 1 - Error shown on missing/additional/incorrect args
  - 1 - Error shown on illegal port number
- 4 - ClientSocket is correct, and works
  - 1 - Socket object
  - 1 - Connects to correct endpoint
  - 1 - Socket sends data
  - 0.5 - Each data chunk is 1000 bytes
  - 0.5 - data is initialized to 0x0
- 2 - data is sent for time period t (specified in command line args)
- 2 - Throughput calculation including display
  - 1.5 - throughput is calculated correctly with correct units
  - 0.5 - output format matches that in description

RUBRIC

- **Section 3: Server (10 points)**

- 2 - args are parsed correctly
  - 1 - Error shown on missing/additional/incorrect args
  - 1 - Error shown on illegal port number
- 4 - ServerSocket is correct, and works
  - 1 - ServerSocket object bound to correct port
  - 1 - Accepts connections
  - 1 - Received data
  - 0.5 - Tries to read 1000 bytes each time
  - 0.5 - count correct # of bytes received (not necessarily 1000)
- 2 - Time duration
  - 1 - Start time on accept or on arrival of 1st trunk of data
  - 1 - End time on client closing connection (receiving EOF)
- 2 - Throughput calculation including display
  - 1.5 - Rate calculated correctly with correct units
  - 0.5 - Output format matches that in description

RUBRIC

- **Section 4: Testing (6 points)**

- 2 - Wired Environment
  - 1 - Throughput values on client and server
  - 1 - Screenshot
- 4 - Wireless Environment
  - 1 - wrote a prediction
  - 1 - Throughput values on client and server
  - 1 - screen shot
  - 1 - Explanation of results

RUBRIC

- **Section 5: Measurements in Mininet (29 points)**

  All measurements for 30 packets and 30 seconds (if incorrect deduction 1*num main question)
- 5 - Q1 Latency
  - 1 - Latency_L1.txt
  - 1 - Latency_L2.txt
  - 1 - Latency_L3.txt
  - 1 - Latency_L4.txt
  - 1 - Latency_L5.txt
- 5 - Q1 Throughput
  - 1 - Throughput_L1.txt
  - 1 - Throughput_L2.txt
  - 1 - Throughput_L3.txt
  - 1 - Throughput_L4.txt
  - 1 - Throughput_L5.txt
- 5 - Q2
  - 2 - Prediction written
  - 1 - Average RTT found
  - 1 - Measured throughput found
  - 1 - explanation of results

RUBRIC

- 7 - Q3
  - 2 - Prediction written for 2 hosts
  - 2 - Prediction written for 3 hosts
  - 1 - Latency and throughput measurement for 2 hosts
  - 1 - Latency and throughput measurement for 3 hosts
  - 1 - Explanation of results
- 7 - Q4
  - 2 - Prediction written
  - 1 - Latency_h1-h4.txt
  - 1 - Latency_h5-h6.txt
  - 1 - Throughput_h1-h4.txt
  - 1 - Throughput_h5-h6.txt
  - 1 - Explanation of results