ECE 361 Lab #3: UDP Bandwidth Estimation

Overview

In this lab you will write a client application using **C Socket Programming**, which utilizes UDP packets to estimate the propagation delay and transmission rate of a simulated link.

NOTE: This is a programming assignment. You are provided (at least) **2 weeks** to complete your client application and demonstrate it during your lab practicals. Please start early and use the discussion board for any questions.

Figure 1 illustrates the timing diagram to transmit a single frame and receive an acknowledgement over a link.

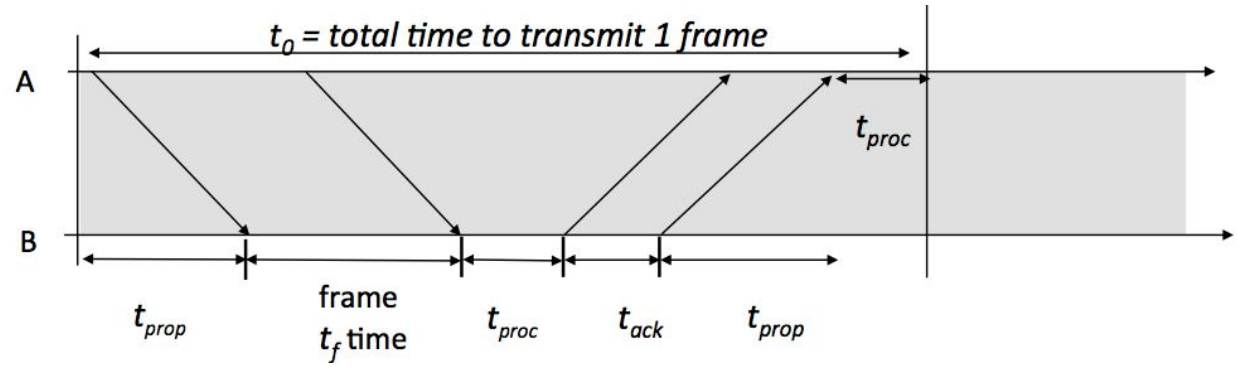

Figure 1: Timing Diagram of a frame transmission and acknowledgement

You will be given **3 simulators** (compiled C programs). The simulator's functionality is depicted in figure 2 below. Your UDP client will communicate with the simulator, which is a UDP echo server and a simulated link with an unknown transmission rate ***R*** and propagation delay $\boldsymbol{t_{prop}}$.

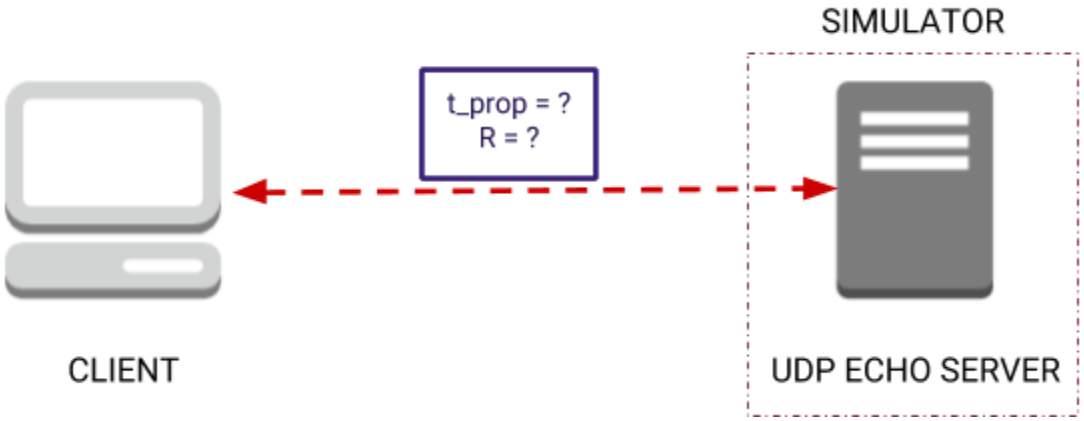

Figure 2: Simulator functionality

Your objective is to determine the values of ***R*** and t_{prop} by sending UDP datagrams *of* varying size to the echo server and measuring the amount of time until the echo datagram (duplicate) is received. Assume the processing times t_{proc} (in figure 1) are negligible and that t_{ack} is equal to t_{frame} since the acknowledgement is an echo.

The Lab

You will be given **3** simulators: **sim1.o**, **sim2.o**, **sim3.o**.
The simulators can be run using the command:

```
./sim<X>.o <port_number>
```

```
Example: ./sim1.o 5555
```

Your client can then communicate with the simulator by sending datagrams to the address **localhost** and to the **port number** you specified when starting the simulator.

NOTE: Please close your programs using **Ctl + c** instead of **Ctl + z** as the programs may not exit properly and the ports will not be released for future uses. If you want to kill suspended processes, refer to the following link: https://www.linux.com/learn/intro-to-linux/2017/5/how-kill-process-command-line
You can also use the following command to check which port the simulator is listening on: **`netstat -tulpn`**

TASK: Use your client program (written in C) to find the transmission rate in ***bits/s*** and propagation delay in ***microseconds*** for each of the simulators.

Hints: Beej's Guide

http://beej.us/guide/bgnet/html/multi/index.html

Beej's guide is an extremely useful tutorial for network programming in C. It provides a walkthrough from network basics to building applications using complex socket programming. It is highly recommended to read the guide and use it's C source code examples as starter code for your applications. You will use these mechanisms (and more) for the next lab as well. For this assignment, we suggest you to go over sections **2 - 6**.

Hints: C functions

You may find C struct **timeval** useful as well as the function **gettimeofday** for measuring the propagation and frame transmission times.

Hints: Use of Debuggers

It is highly recommended that you use either **NetBeans Debugger** or **GDB** to debug your program (especially segmentation faults). This will be extremely useful for the upcoming labs as well.

Demonstration

Once you have completed the client application and have measured the transmission rate and propagation delay for each simulator, **signal to the Lab TA that you are ready to perform the demo**. You must demo the functionality of your client application and describe how you obtained your estimates. The Lab TA will ask you about your approach and specifics about your code.

Submission Instructions

You must submit the following files:

- Your client application (must be named **client.c**)
- A Makefile to compile your program. Refer to the link for creating Makefiles: https://www.cs.swarthmore.edu/~newhall/unixhelp/howto_makefiles.html
- A text file to explain your findings you observed from this lab. It must be named **explanations_lab3.txt** and should be at most **100 words**. This file must also contain the names and student numbers of the group members (at the beginning of the file) prefixed by **#**. Please do not prefix other lines by **#** as this would confuse the automated scripts.

```
#first1 last1, studentnum1
#first2 last2, studentnum2
```

You must combine all these files into a tarball (must be named **lab3.tar.gz**) using the following command:

```
tar –czvf lab3.tar.gz <project directory>
```

Note: Only one student in the group needs to submit.

You can submit the file using the following command:

```
submitece361s <lab_number> <filename>
```

Example: **`submitece361s 3 lab3.tar.gz`**

You can verify if you have submitted successfully by using the following command: **`submitece361s -l 3`**

For more information regarding the command, please refer to it's man page: **`submitece361s`**

The submitted files will be used to verify your findings and check for plagiarism.

Marking

This lab is worth a total of **3** marks with the following breakdown:

- Client application/ Bandwidth Estimates: **2** marks (marked as group)
- Demo: **1** mark (marked individually)

All marks will be assigned by the end of the lab session.