CS5310.251/252, Spring 2020, Programming Project

Simulation of Fast Ethernet

Kai Qi

1. Project Objective

This project is to simulate the Fast Ethernet. The goal of the project is to practice basic socket API programming through a client/server application. It is a simulation in the sense that the fast Ethernet is simulated by multiple processes on multiple machines. Each station in the Ethernet is simulated by a process. The switch of the fast Ethernet is also simulated by a process.

2. Implementation of Communication Switch Process (CSP)

a. I use a struct named FrameData to represent the data exchanging between the CSP and SP.

b. I use a function named process_frame to convert the client's buffer data to the struct of FrameData. The CSP will respond to SP's request according to different situations with using the function of process_frame.

c. I use a function named partition_buffer to solve the TCP sticky packet problem.

3. Implementation of Station Process (SP)

a. I use a function named wait to let the SP read data from socket instead of reading data from the input file.

b. I use a function named read_file to read what the SP should do. SP can request the CSP to send some data to other SP. SP might need to wait for some frames.

c. I use a function named process_data to convert buffer data to the struct of FrameData. The SP will respond to CSP according to different situations with using the function of process_data.

d. I use a function named partition_buffer to solve the TCP sticky packet problem.

4. Implementation of I/O Multiplexing

I use the select function to implement the I/O Multiplexing both in CSP and SP. The select function allows the process to instruct the kernel to either wait for any one of multiple events to occur and to wake up the process only when one or more of these events occurs, or when a specified amount of time has passed.

5. Observations During The Testing

a. SP can send request frame to CSP.
SP can send data frame to the other SP.
CSP can send positive reply to respond to SP's request.
CSP can send negative reply to respond to SP's request.
CSP can help SP to forward the data frame to the destination SP.

b. This program can handle multiple SP simultaneously.

c. This program can handle TCP sticky packet problem.