Syd Lynch

slynch2

CIS 415 Project 2

Report
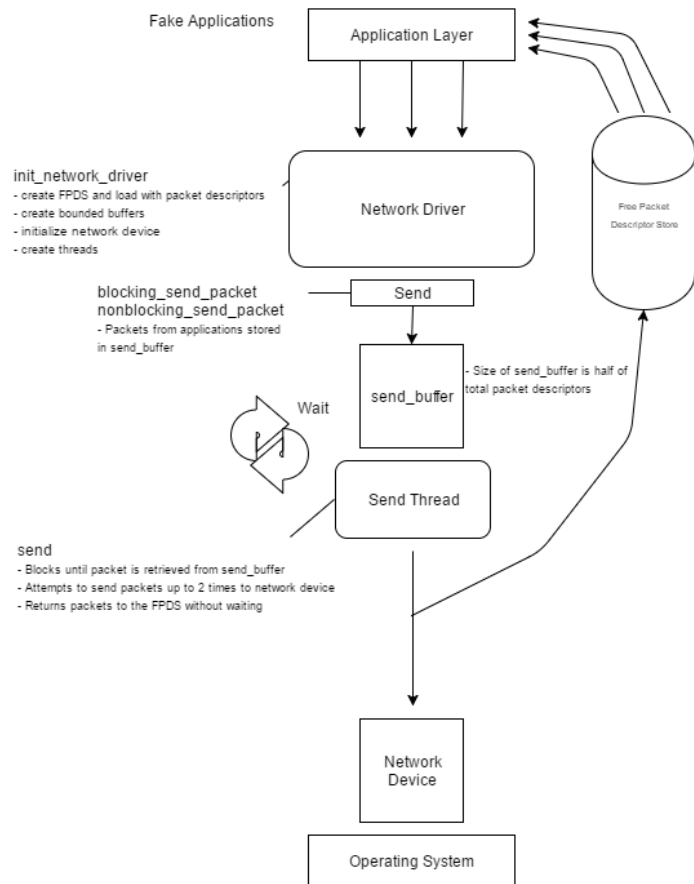
My network driver seems to be working to the desired specifications. I have run ./mydemo several times at different time intervals, at most 20 minutes, writing the output to a file log.txt. After, I perform the command: cat log.txt | grep … with Bug, Error, and Warning in place of the ellipses. I receive none of these in my output. I do not receive any segmentation faults. My output seems to be similar to that of the provided ./demo application, although I cannot say whether they perform similarly. I attempt to shield against bad input in init_network_driver by checking if the network device is null, and returning if so. I planned on checking if the number of packet descriptors created was appropriate for the value of MAX_PID, but did not implement this safeguard.

I discussed the assignment with Danny Lu, Jeremy Unck, and Robert Macy, but all code was completely written by me.

My diagram shows the general flow of the driver, each side representing a thread. The send thread describes applications sending packets to the driver to be stored in a single send buffer, and a sending thread that will send a packet then return that packet to the FPDS. The receive thread shows an array of bounded buffers, one for each application, that stores any received packets. Each side briefly explains the flow of the functions.

Diagram

SEND

Fake Applications

Application Layer

init_network_driver
- create FPDS and load with packet descriptors
- create bounded buffers
- initialize network device
- create threads

Network Driver

Free Packet
Descriptor Store

blocking_send_packet
nonblocking_send_packet
- Packets from applications stored
in send_buffer

Send

Wait

send_buffer

- Size of send_buffer is half of
total packet descriptors

send
- Blocks until packet is retrieved from send_buffer
- Attempts to send packets up to 2 times to network device
- Returns packets to the FPDS without waiting

Send Thread

Network
Device

Operating System

RECEIVE

Fake Applications

Application Layer

init_network_driver
- create FPDS and load with packet descriptors
- create bounded buffers
- initialize network device
- create threads

Network Driver

Free Packet
Descriptor Store

Receive

blocking_get_packet
nonblocking_get_packet
- Retrieve packets from corresponding
receive_buffer based on application's PID

receive_buffer
- Array of bounded buffers to hold
packets for each application
- Size of each is evenly distributed
based on half of the total packet
descriptors divided by total number
of PIDs

Wait

receive
- Only blocks until empty packet descriptor is retrieved from FPDS
- Initialize packets and register with network device
- Wait for incoming packet from network device
- Write to corresponding receive buffer based on PID, return
to FPDS if write failure occurs

Receive Thread

Network
Device

Operating System