Dylan Secreast
CIS 415 Prof. Sventek
Project 2: Report
5/23/16

Project 2: Report

State of Solution

At the time of project submission, my program appears to be working correctly and properly sending and receiving packets between all 11 fake applications. No errors are displayed from DIAGNOSTICS while running besides the occasional dropped packet which seems to be normal behavior. Final solution was tested against the fake applications by successfully running uninterrupted for 30 minutes.

Design

Included in my submission is design.pdf, a workflow showing interactions between applications, network driver, and network device. The design shows how concurrency is achieved by use of incoming and outgoing threads, free packet descriptor store, and incoming and outgoing bounded buffers. The threads created by the network driver utilize non-blocking function calls as to become available ASAP by incoming and outgoing packets from the applications.

Implementation

Since this code is intended to live in an OS, the data structures used are two bounded buffers for incoming and outgoing packets, of size MAX_PID and MAX_PID + 1 respectively. The incoming bounded buffer is an array of bounded buffers where each array element (BB) is initialized to size 4. Two pthreads are then used for concurrency to independently handle incoming and outgoing packets. These threads are initialized with their respective helper functions which process the packets. Both helper functions utilize non-blocking calls except for awaiting packet, and if a nonblocking_put_pd fails after the first attempt then blocking_put_pd is called to assure proper return of packet descriptor. After each helper function has finished its process, the used packet descriptor is returned to the free packet descriptor store. When the outgoing thread attempts to send a packet and fails, it tries again 3 more times if necessary. The packet is then dropped if it is not able to successfully send after 4 tries total.