CMPUT 379
OPERATING SYSTEM CONCEPTS

Assignment #2
Modelling a Software Defined Network using FIFOs

Due: Thursday, November 1, 2018, 09:00 PM

By Hee Soo Park (1389532)

Objectives

a2sdn: To implement a peer-to-peer program that utilize signals for examining the progress of the running processes, using FIFOs for communication and I/O multiplexing for nonblocking I/O. To read from a shared FIFO file, and directing each packets to a appropriate Switch and to handle communication between switch-controller and switch-switch. List the status of controller and switches, with switches in appropriate switch with appropriate IP_range.

Acknowledgements

Referecd fork()/pipe from "https://www.geeksforgeeks.org/named-pipe-fifo-example-c-program/"

Professor Ehab Elmallah lecture slide and examples.

TA's help duding lab session.

Design Overview

a2sdn

- command line arguments for controller and switch

```
HeeSoos-MacBook-Pro:submit heesoopark$ ./a2sdn cont 2
```

```
HeeSoos-MacBook-Pro:submit heesoopark$ ./a2sdn sw1 t1.dat null sw2 100-110
```

- Success run of a2sdn (cont/swi)
- Welcome Message with instructions

```
#########################################################################
#                                                                       #
#            CMPUT 379 -- a2sdn (Controller and Switch)                 #
#                                                                       #
#            (1) Please Turn on Controller                              #
#            (2) Please Turn on Switches in increasing order            #
#            (3) Please wait around 3-6 seconds to turn on              #
#                the next switch (depending on size of trafficFile)     #
#            (4) Type 'list' to view the status                         #
#            (5) Type 'quit' to view the status and terminate           #
#                                                                       #
#                                                                       #
#             User Command List:                                        #
#            (1) list - View the status of Contorller or Switch         #
#            (2) exit - View teh status and terminate                   #
#                                                                       #
#                                                                       #
#                Thank you -by Hee Soo Park                             #
#                                                                       #
#########################################################################
```

- Success run of switch in a2sdn
- Displays the relevant FIFO the switch is reading and writing

```
Switch Reads From Controller : myfifo-0-1
Switch Writes To Controller  : myfifo-1-0

READ FROM LEFT               : myfifo-0-1
READ FROM RIGHT              : myfifo-2-1
WRITING FROM LEFT            : myfifo-1-0
WRITING FROM RIGHT           : myfifo-1-2
```

- Display of Controller status upon “list” or “exit” command

```
Switch Information (nSwitch 2):
[sw1] port1 = -1, port2 = 2, port3 = 100-110
Packet Stats:
        Received     :   OPEN: 2, QUERY: 5
        Transmitted  :   ACK : 2, ADD  : 5
```

- Display of Switch status upon “list” or “exit command

```
Flow Table:
[0] (srcIP= 0-1000, destIP= 100-110, action= FORWARD: 3, pri= 4, pktCount= 3)
[1] (srcIP= 0-1000, destIP= 200-200, action= DROP: 0, pri= 4, pktCount= 1)
[2] (srcIP= 0-1000, destIP= 300-300, action= DROP: 0, pri= 4, pktCount= 1)
[3] (srcIP= 0-1000, destIP= 200-200, action= DROP: 0, pri= 4, pktCount= 1)

Packet Stats:
        Received     :   ADMIT:5,    ACK:1,     ADDRULE:3,      RELAYIN:0
        Transmitted  :   OPEN :1,    QUERY:3,   RELAYOUT:0
```

Project Status

a2sdn have been finished with all the required functions, they are stable, consistent and correct. I have implemented all the required function as modular as possible. List, exit and work flawlessly. But please note that, user needs to wait until next execution of switch, as switch needs time to read the trafficFIle (time is dependent of the size of trafficFile) and controller needs time to absorb the data from switch. I have also implemented a user GUI (welcome message) up on execution of the program.

I am satisfied with both my programs and their outputs and GUI.

I had hard time with figuring out with reading and writing of FIFO with synchronization, and relaying between switches, as switches need to poll from multiple FIFO files.
I have hard time synchronizing the I/O multiplex, since switch needs time to read the trafficFile and controller needs time to read from switch. I had a solution to sleep(), therefore the program is synchronized, due to this conflict, user needs to wait after execution of switch until executing next switch. Also I had great difficulty coding with C, as I had to manage dynamic allocated memory and was really hard to debug for errors such as "segmentaion fault" or "abort 6", I used "gdb" debugger to over come this problem

Testing and Results

I have tested a2sdn by

running function by, running example from eClass (2 switches and 3 switches), and executing the program commands such as "list" and "exit".
And my program have matched the output of examples correctly.

List function have been tested by checking if successfully ran packets have been append to list and display properly.

Exit function have been tested by exiting a2sdn, and check if the status of controller/switch is synchronized with the up-to-date status and to exit correctly.