**EECE.7290 Special Topics on SDN and Data Plane Programming**

**Lab2-TX**

**Introduction:**

In lab2-RX we learned how to receive the packets. In this lab2-TX, we are going to learn how to send packets, since in some applications we need to forward the packet out after the packet is processed.

**Tasks:**

1. Read the main.c source file and grasp the big picture of DPDK APIs.

2. Know to how to modify the main.c so that it does the job as we wish.

**Submissions:**

A single report which includes the following

1. A snapshot of the screen after you successfully start the application. How to compile and run the code can be found in the README file.

2. Summarize the main.c, write down all the important DPDK APIs(with prefix rte) used in the main.c in time-sequential order. Answer all the following questions in the report:

Some questions which depict the Big-Picture of a DPDK application

* Which API is used to setup EAL?
* Which API is used to configure the device (hardware port), and specify the number of RX queues and TX queues etc?
* Which API is used to setup RX queue, and in order to setup RX queue which API is used to setup memory pool?
* Which API is used to setup TX queue?
* Which API is used to enable the promiscuous mode?
* Which API is used to start the link?
* Which API is used to receive the packets from the device?
* Which API is used to send the packets out to the device?

Some minor questions which could help you program

* Which API can tell you the current core ID?
* Which API can tell if the given core ID is enabled by the “-c” option in the command line?
* Which API can tell you the ID of the master core?
* Which API can tell you the socket ID of the current port?
* Which API is used to release the packets back to the memory pool?
* Which API is used to the get data address of each packet?
* Which API can tell you the current CPU cycles?
* Which API can tell you the frequency of the CPU?
* Which API is used to let all the cores to start to run an user application?

3. The current main.c can generate and send random UDP (with correct checksums) packets. Modify the main.c so that it only generates TCP (with correct checksums) packets with given source IP address 10.0.0.1 and destination IP address 10.0.0.2. You can still keep other header fields random. Attach only your added/modified code in the report and explain.