Week 6 – 10.07.23 – 14.07.23

Installation of DPDK

- We install dpdk from the official website and follow steps from this guide
- To confirm if installation is done correctly, Run hello world program in the examples by;
    - cd examples/helloworld && make && cd ./build
    - sudo ./helloworld –l 0-3 –n 2 ; it will print helloworld.

Configure Hugepage

We need to enable hugepages as it is required by DPDK. To do so, go to /usertools directory and run;

```
dpdk-hugepages.py –p 2M –setup 20M
```

Bind the Network Interface with DPDK compatible driver (vfio-pci or uio_pci_generic)

We need to bind the network interfaces with DPDK compatible drivers (vfio, uio) instead of kernel drivers. This helps the dpdk to skip the kernel space and directly talk to NIC. Run;

```
lsmod | grep vfio # if it outputs nothing, we need to load the driver

modprobe vfio-pci # if still output of lsmod | grep vfio is nothing, switch to uio_pci_generic.

modprobe uio_pci_generic
```

See what interfaces are available by running in usertools directory;

```
sudo dpdk-devbind.py –s
```

Bind interface, for example, you want to dpdk;

```
sudo dpdk-devbind.py -b uio_pci_generic 0000:00:1f.6
```

Running l2fwd example with dpdk

sudo dpdk-hugepages.py -p 2M --setup 50M

sudo ./l2fwd -l 0-1 -- -q 1 -p 1

Running l2fwd with t4p4s

Run from t4p4s directory;

./t4p4s.sh :l2fwd model=v1model cores=2 verbose=1

P4-DPDK-Target

This tool can be used for our use-case to run P4 programs on NIC. This uses P4C-DPDK as its backend compiler. The workflow it follows is;

P4 program -> P4 compiler creates .spec file based on p4 program -> .spec file is used by dpdk -> dpdk creates c program based on .spec file -> C compiler compiles it -> a binary representation to be deployed on NIC.

https://www.youtube.com/watch?v=dPvH_joaScA