Lab 2 – send and receive packets with DPDK

Handout: March 12, 2019
Deadline: March 24 23:00, 2019 (No extension)

Assignment overview:

In this assignment, you will learn what DPDK is and what it's used for, you will learn how to set up a DPDK development environment and understand the implementation of DPDK modules and programs. In part 1, you are asked to answer some questions about DPDK. In part 2, you need to write a DPDK application to construct UDP packets and send them to NIC using DPDK.

Prerequisite:

In this lab, DPDK must be installed in your virtual machine. VMware and Ubuntu16.04 are suggested.

The steps for building DPDK development environment are as follows:

1. Download DPDK's source code from http://core.dpdk.org/download/

```
$ tar xf dpdk-stable-16.11.8.tar.gz                # 解压源码

$ cd dpdk-stable-16.11.8                           # 进入源码目录

$ make install T=x86_64-native-linuxapp-gcc        # 安装 DPDK
```

2. Configure and build DPDK runtime environment

```
$ sudo modprobe uio                      # 挂载 Linux 内核的 UIO 模块

$ sudo insmod ./build/kmod/igb_uio.ko    # 挂载编译生成的 igb_uio 模块

                                           （build 为 x86_64-native-linuxapp-gcc）

$ sudo ./tools/dpdk-devbind.py -s    # 列出所有网卡

$ sudo ./tools/dpdk-devbind.py \     # 为 1.3 中添加的网卡绑定 igb_uio 驱动（这步可以跳过）

$    --bind=igb_uio enoX             # 其中 enoX 由具体环境决定(enoX 一般为 02:00.1 之类的)

$ mkdir -p /mnt/huge                     # 创建 hugetlbfs 挂载点

$ mount -t hugetlbfs nodev /mnt/huge     # 挂载 hugetlbfs

$ echo 64 > /sys/kernel/mm/hugepages/hugepages-2048kB/nr_hugepages  # 分配大页
```

3. Run the sample program to make sure you have successfully installed DPDK

```
$ export RTE_SDK=/path/to/your/dpdk            # 设置 DPDK SDK 路径

$ export RTE_TARGET=x86_64-native-linuxapp-gcc # 设置保存 DPDK 编译生成内容的目录

                                                 (一般为 x86_64-native-linuxapp-gcc)
```

```
$ cd $RTE_SDK/examples/helloworld              # 进入 DPDK 中的 helloworld 例子

$ make                                         # 编译

$ sudo ./build/helloworld                      # 运行
```

For more information about installing DPDK,refer to the guidance (http://doc.dpdk.org/guides/linux_gsg/).

Part 1: Get familiar with DPDK

Take DPDK official website http://www.dpdk.org/
and Chapter 1 of "深入浅出DPDK" as references, answer the questions below:

Q1: What's the purpose of using hugepage?

Q2: Take examples/helloworld as an example, describe the execution flow of DPDK programs?

Q3: Read the codes of examples/skeleton, describe DPDK APIs related to sending and receiving packets.

Q4: Describe the data structure of 'rte_mbuf'.

Part 2: send packets with DPDK

1. Construct UDP packets with DPDK according to the definition of UDP/IP/Ethernet header. Contents of packets can be filled as you wish.
   (Tips: refer to codes of examples/skeleton and construct UDP packets in the array of rte_mbuf, you may use the function rte_pktmbuf_mtod
   https://doc.dpdk.org/api/rte__mbuf_8h.html#a2a8b10263496c7b580e9d0c7f2a1f073)

```
2.   /* Ethernet 协议头的定义在 lib/librte_net/rte_ether.h 中: */
3.  /**
4.   * Ethernet header: Contains the destination address, source address
5.   * and frame type.
6.   */
7.  struct ether_hdr {
8.      struct ether_addr d_addr; /**< Destination address. */
9.      struct ether_addr s_addr; /**< Source address. */
10.     uint16_t ether_type;      /**< Frame type. */
11. } __attribute__((__packed__));
12.
13. /* IP 协议头的定义在 lib/librte_net/rte_ip.h 中: */
14. /**
15.  * IPv4 Header
16.  */
17. struct ipv4_hdr {
18.     uint8_t version_ihl;        /**< version and header length */
19.     uint8_t type_of_service;    /**< type of service */
20.     uint16_t total_length;      /**< length of packet */
21.     uint16_t packet_id;         /**< packet ID */
22.     uint16_t fragment_offset;   /**< fragmentation offset */
23.     uint8_t time_to_live;       /**< time to live */
24.     uint8_t next_proto_id;      /**< protocol ID */
25.     uint16_t hdr_checksum;      /**< header checksum */
26.     uint32_t src_addr;          /**< source address */
27.     uint32_t dst_addr;          /**< destination address */
28. } __attribute__((__packed__));
29.
```

```c
/* UDP 协议头的定义在 lib/librte_net/rte_UDP.h 中: */
/**
 * UDP Header
 */
struct UDP_hdr {
    uint16_t src_port;      /**< UDP source port. */
    uint16_t dst_port;      /**< UDP destination port. */
    uint16_t dgram_len;     /**< UDP datagram length */
    uint16_t dgram_cksum;   /**< UDP datagram checksum */
} __attribute__((__packed__));
```

2. Write a DPDK application to construct and send UDP packets. (Refer to examples/skeleton). Before writing any code, you need to enable the virtual NIC in your virtual machine. If you are using VMware, refer to this website to enable a host-only mode virtual NIC: https://blog.csdn.net/ka_ka314/article/details/78936105
After virtual NIC is enabled, use following command to bind it to DPDK:

```
ifconfig                    #查看所有网卡的状态

sudo ifconfig ens33 down    #停止 ens33 网卡的工作，ens33 为刚激活的 host-only 模式下的虚拟网卡

   #（ens33 可替换成其他网卡，在绑定网卡到 DPDK 前需要使其从活跃状态变为不活跃状态）

sudo tools/dpdk-devbind --bind=igb_uio enoX # enoX 一般为 02:01.0 之类(与 ens33 对应)

sudo tools/dpdk-devbind -s #验证虚拟网卡是否被成功绑定到 DPDK 上
```

For binding ports to DPDK modules, refer to the contents of chapter 5.4 of http://doc.dpdk.org/guides/linux_gsg/linux_drivers.html

3. To verify the correctness of your program, you need to capture the packets and display the contents of them. **Wireshark** is the world's foremost and widely-used network protocol analyzer. It can help to capture the packets sent to NIC. Install it in your physical machine (not in virtual machine) and test your program.
https://www.wireshark.org/

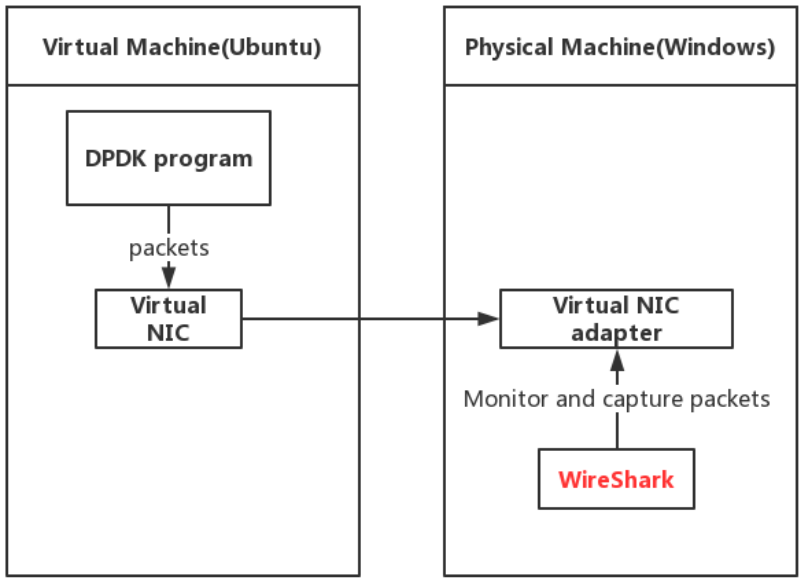


Diagram used to show how to test your program with Wireshark

Tips:

If your enabled virtual NIC is in host-only mode, you need to capture packets from the same virtual NIC in your physical machine using wireshark. If your enabled virtual NIC is in bridge-connection mode, you need to capture packets from the NIC your virtual NIC bridged to in your physical machine using wireshark. NAT mode is not advised.

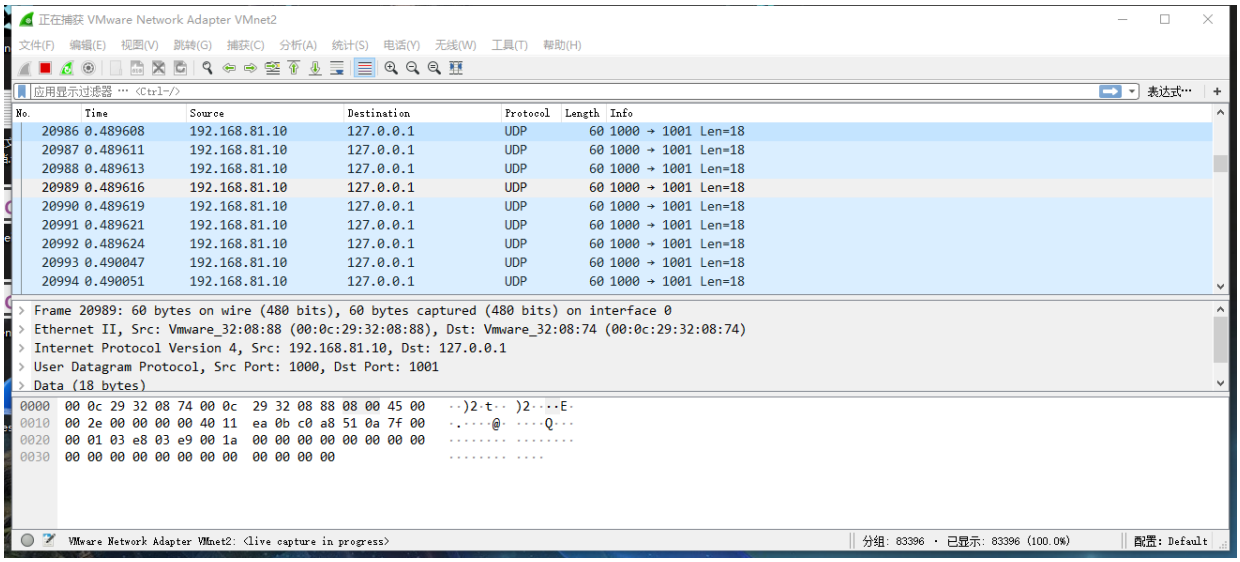
Grading:

- 50% for answers to questions in part one.
- 50%forthe code of DPDK program.Code will be checked manually and results will be verified by wireshark.

Handin procedure:

You need to submit the source code of the DPDK application, and a report including the answers to the questions in part 1 and a description of how you verify the correctness of the application with the screenshot of Wireshark in part 2.

The screenshot is supposed to look like this:



Send your report and source code as a gzipped tar file, named as {Your studentID}.tar.gz, to zhuangshujun@163.com

Reference:

1. DPDK official website: http://www.dpdk.org/
2. Helloworld 例程 http://doc.dpdk.org/guides/sample_app_ug/hello_world.html
3. 深入浅出DPDK https://book.douban.com/subject/26798275/
4. DPDK APIs http://doc.dpdk.org/api/