实验二 利用DPDK构造并发送数据包

1. **实验目的**
2. 学会编写和测试DPDK数据包发送程序。
3. 理解DPDK实现高性能数据包处理的原理。
4. **实验内容**
5. 编写一个DPDK程序，实现如下功能：**直接构造一个数据包**(内容任意)，要求数据包的协议头符合UDP/IP/Ethernet协议规范，并将其发送出去。该程序**基于skeleton程序**修改得到，附录中给出一个供参考的编程框架。
6. 使用tcpdump命令，抓取数据包，观察各协议字段值，确认实际输出与程序意图是否一致。

过程举例：

1. //在命令行窗口运行如下命令 来 抓取经过指定网卡含有特定端口号的数据包
2. sudo tcpdump -vv -A -e -i <网卡名称> port <端口号>
3. //然后在另一窗口运行DPDK程序
4.
5. //注意：tcpdump只能在内核驱动网卡上抓取数据包，更多用法可参阅相关资料

**三、注意事项**

1. 由于构造DPDK UDP包时需要构造以太网首部，IP首部以及UDP首部，可能会比较复杂，要注意各字节对应的协议内容以及主机**字节顺序**和网络字节顺序的转换。

 2. 可以先使用tcpdump命令抓取UDP包，观察收到的UDP包的首部和payload结构，合理分析自己构造的UDP数据包是否符合协议规范（源目的MAC地址，源目的IP地址，源目的端口，以及协议类型字段是否按照要求构造）。

 3. 在DPDK程序运行时，使用tcpdump命令过滤端口和协议类型后如果收不到包，可以适当放宽过滤条件，在观察到自己程序中构造的包后再分析是哪个字段的构造出了问题。最后应该能够保证抓到的包的各部分字段与预期一致，且tcpdump命令能够在过滤了端口以及协议类型的情况下接收到包。

 4. 由于运行DPDK程序的网卡应该挂载igb\_uio的驱动，不能够使用tcpdump命令抓包。所以推荐的配置为使用一个虚拟机，其中一张网卡配置为NAT模式，用于上网。再通过virtualbox设置界面新开辟出两张网卡，都设置为桥接网卡，并指定为同一个物理网卡。在做实验时，其中一张桥接网卡挂载DPDK的驱动运行DPDK的程序，另一张桥接网卡挂载内核驱动并开启混杂模式，使用tcpdump命令用于抓包。（进行虚拟机网卡设置时应该将虚拟机关机）。

 5. 增加新网卡后需要重新安装DPDK，编译新的igb\_uio.ko。总之如果出现莫名其妙的挂载错误，可以尝试重新编译一次DPDK，大概率可以解决问题。

 6. 进行UDP包构造时，应该先使用rte\_pktmbuf\_append函数在rte\_mbuf后申请内存空间，然后再进行数据填入，具体使用方法请参考手册。

1. **回答问题**

官方文档：[DPDK documentation](http://doc.dpdk.org/guides-18.11/), [DPDK API](http://doc.dpdk.org/api-18.11/)

注意：不同版本DPDK对应的文档内容也不完全一样，应对应查看

1. UDP协议头的cksum是必须的吗？不是的话应当如何设置？
2. DPDK为IP协议头的cksum计算提供了专用函数，请在相关文档中找到它，并应用在程序中，调用该函数的注意事项是什么？
3. DPDK实现高性能收发包的原理是什么？
4. **思考题**

 参照本实验，一个接收数据包并处理其中数据的DPDK程序应该如何实现？

1. **进展报告**

本实验的完成情况(需要附tcpdump抓到构造的udp包的截图)以及对以上3个问题的回答作为第二阶段的进展报告（进展报告中可以附关键代码），于 10 月 25日23:59之前在bb系统提交，会在提交后的下次课进行课堂讨论。

进展报告中需注明小组成员（组长排在第一个），以及每位成员对该报告的贡献比例。上课前各小组做好PPT，每个小组派一位同学上台主讲，其余同学可以补充。

提交文件命名及格式：进展报告：第二阶段\_组长姓名.pdf，代码：（随意命名）.c，PPT仅做上课展示用，不做提交要求。

实验过程中遇到任何问题，请及时发邮件给助教（ltli@mail.ustc.edu.cn），并抄送一份给主讲老师。

1. **附录**

 首先，可以在example目录下新建一个目录用于本实验，可以看到，example/skeleton目录中有Makefile和meson.build这两个文件，各拷贝一份到新建的目录中，将其中的文件名修改为你的主程序文件名。

注意：

* 1. [主机字节顺序和网络字节顺序的转换](https://doc.dpdk.org/api/rte__byteorder_8h.html#a6ec4cb000fc67785792d18728cf45563)，8bit不需要转换，16bit及以上一般需要进行转换
	2. 某些字段值可使用宏定义，如IPVERSION
	3. 可以使用的常用函数如inet\_addr()
	4. 各协议头对应的结构体有不同的版本，既可以使用经典版本(如[Ethernut](http://www.ethernut.de/api/annotated.html))，也可以使用DPDK自带的版本([DPDK\_Data\_Structure](http://doc.dpdk.org/api-18.11/annotated.html))，这里更推荐使用后者。各头部结构体中包含的数据项可以参阅相关文档，也可以从基础程序的头文件中获取信息。

基础程序链接：<https://rec.ustc.edu.cn/share/41077710-47cd-11ed-b851-11334779ec7e> 密码：z8k4

以上程序代码基于skeleton修改得来，主要是增加了一个build\_udp\_packet函数，本次实验的主要任务就是补充该函数中的相关代码，从而构造出一个UDP数据包。

1. **一些Q&A**

Q: dpdk的port号与什么对应？

A: 网卡，且port从0开始编号，本次实验中port设置为0即可。

Q: 在安装dpdk后，重新进行虚拟机配置，增加新网卡后，igb\_uio.ko模块挂载失败

A: 增加新网卡后需要重新安装dpdk，编译新的igb\_uio.ko

Q: 在使用tcpdump抓包时，enp0s3网卡找不到?

A: 应当配置对宿主机的同一块物理网卡，虚拟出2块网卡桥接到其上，从而使其处在同一网段，然后将其中一块虚拟网卡挂载igb\_uio模块，然后在另外一块上设置为混杂模式后进行抓包。

Q: 关于checksum，需要自己编写吗？

A: 对于ip包的checksum，可以使用dpdk自带的rte\_ipv4\_cksum来进行计算。对udp包的checksum不必关心。