一、工程概述

1.背景

在 5g 通信系统中的服务器与 B7 通信环节，需要一个高吞吐量，低时延要求的稳定的解决方案——DPDK 技术很好的满足了这个需求。

2.软硬件环境要求

【硬件要求】

多于八逻辑核心，64 位的服务器

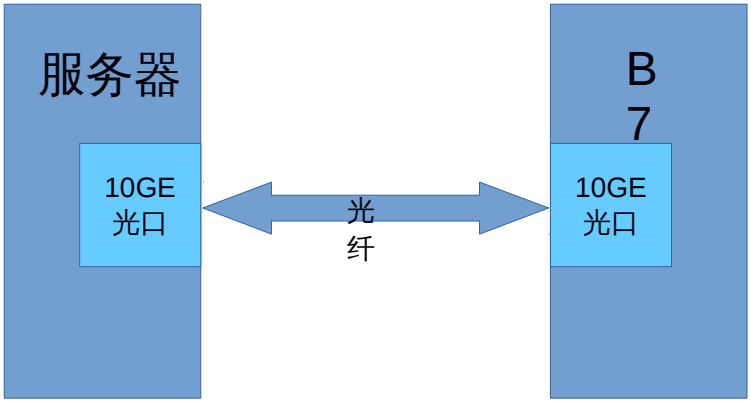
服务器配一张 dpdk 支持的网卡(http://dpdk.org/doc/nics)

【软件要求】

linux 操作系统(推荐 centos 或 ubuntu)，kernel 内核版本高于 2.6.33

dpdk 软件包 dpdk-stable-17.0.2.1(http://dpdk.org/download)

3.硬件连接



硬件连接图

4.技术指标

通信方式:全双工

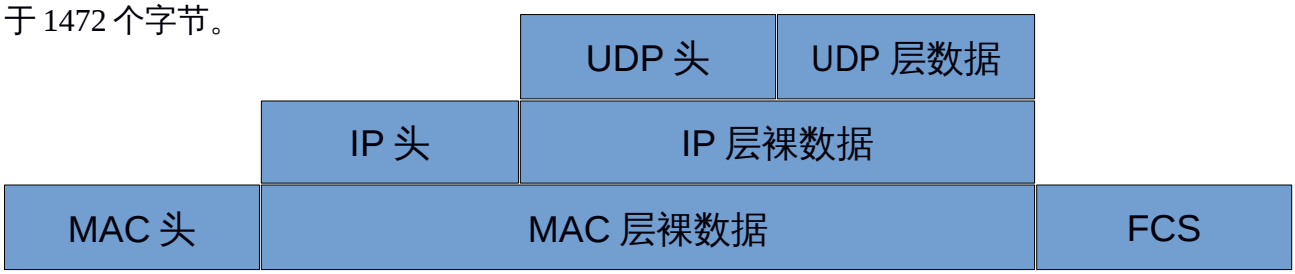
通信速率:双向都达到 9Gb 以上

时延:单向 12us 以内

二、帧协议结构

1.协议概述

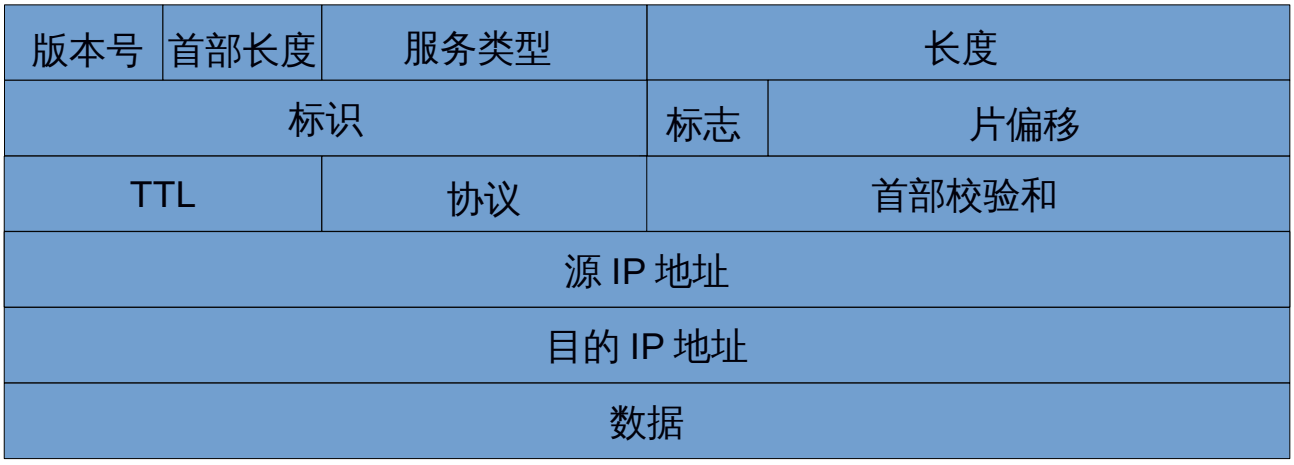
本程序中采用 UDP/IP/Ethernet V2 帧结构，并要求但个数据包的 UDP 层的裸数据长度不得长于 1472 个字节。



2.L4-UDP 协议

源端口号:2 字节。
目的段口号:2 字节。
字节数:整个 UDP 包的字节数，包括包头和数据，2 字节。
校验和:校验范围包括伪首部，首部和数据，2 字节。
数据:UDP 层的裸数据，在此程序中，UDP 层的裸数据不得高于 1472 字节。

3.L3-IP 协议



版本号:IP 包所使用的 IP 协议的版本号 4bit。
首部长度:IP 包的首部长度，以字节为单位，4bit。
服务类型:此 IP 包的服务类型，1 字节。
总长度:IP 数据报的总字节数，包含头部，2 字节。
标识:IP 包的 identification，2 字节。
标志:最高位保留；次高位为 1:不允许分片；最低位为 0:此数据报是整个数据报的最后一片，3bit。
片偏移:每偏移动 8 个字节，此处加 1，13bit。
TTL:最高允许的跳数，1 字节。
协议:此包的上层协议，1 字节。运行程序需要外部参数 -c ff -n 2
-c 是允许程序最多使用的核心数 ff 代表 8(掩码 8‘b1111_1111)

-n 是内存通道数，作用未知
首部校验和:2 字节。
源 IP 地址:4 字节。
目的 IP 地址:4 字节。
数据:IP 层的裸数据。

注:根据 MAC 协议，IP 层的裸数据不得少于 26 字节。若小于 46 字节，则需要 padding；在此工程中 IP 层的裸数据不得长于 1480 字节。

4.L2-Ethernet V2 协议

目的 MAC 地址	源 MAC 地址	类型	数据	FCS

目的 MAC 地址:6 字节
源 MAC 地址:6 字节
类型:上层协议类型代码，2 字节
数据:数据长度要求为 46 字节～1500 字节，若小于 46 字节，则需要 padding；MAC 层的裸数据不得长于 1500 字节。
FCS:MAC 层的校验字节，由硬件添加和剥除，4 字节。

5.一个典型的包

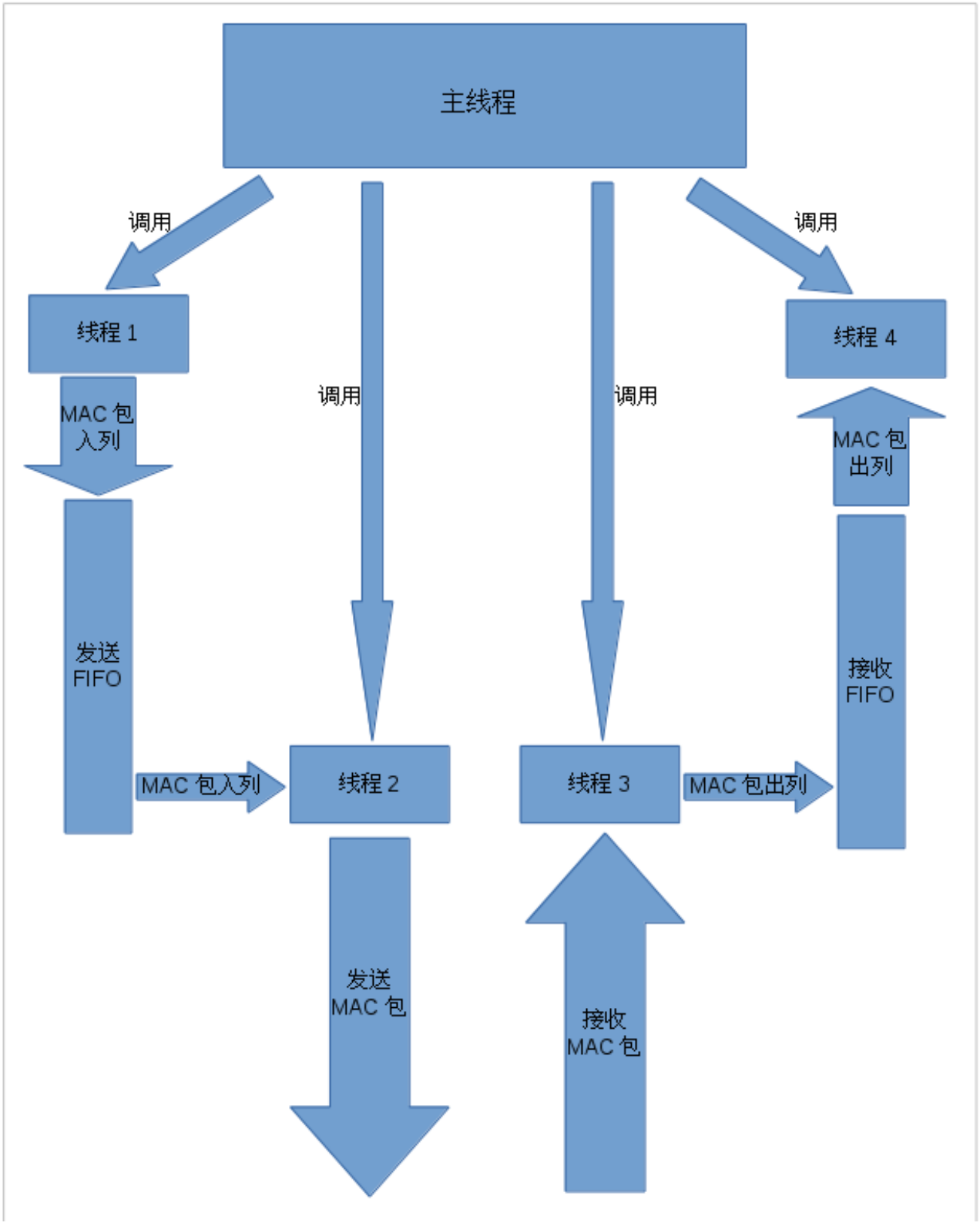
以太头
ff ff ff ff ff ff dc fe 18 90 f8 8b 08 00

IP 头
45 00 00 91
00 00 40 00
40 11 79 B3
C0 A8 00 01
FF FF FF FF
UDP 头
81 94 13 89 00 7D 76 60

UDP 层裸数据
01 01 0E 00 E1 2B 83 C7 EE 81 00 67 00 00 00 06 00 0A 54 4C 2D 57 44 52 35 36 32 30 00 0B 00 03 31 2E 30 00 07 00 01 01 00 05 00 11 44 43 2D 46 45 2D 31 38 2D 39 30 2D 46 38 2D 38 42 00 08 00 0B 31 39 32 2E 31 36 38 2E 30 2E 31 00 09 00 0A 74 70 6C 6F 67 69 6E 2E 63 6E 00 0A 00 0E 54 4C 2D 57 44 52 35 36 32 30 20 31 2E 30 00 0C 00 05 31 2E 37 2E 33

三、UDP-DPDK 程序解析

1.程序结构



2.程序解析
主线程:在主线程中进行 DPDK 环境的初始化，并调用 4 个子线程，最后等待 4 个子线程结束。
线程 1:在这个线程中读取 txt 文件中 UDP 裸数据，并使用几个类和函数构造出 MAC 包并存放在一个 mbuf 中，随后将这个 mbuf 放入发送 FIFO。

线程 2:在这个线程中每隔一定时间(可调)尝试从发送 FIFO 中拿出 32 个包，如果发送 FIFO 中的 MAC 包多于 32 个，则只拿出 32 个 MAC 包；若发送 FIFO 中的 MAC 包少于 32 个，则尽可能的多拿出。随后将所有拿出的 MAC 包发送出去。

线程 3:以轮询的方式从网卡中获取网卡收到的 MAC 包，将获取到的包放入接收 FIFO 中。

线程 4:从接收 FIFO 中拿出 MAC 包并按照协议进行解包。

四、示例用法

在 dpdk_udp_1.0 中，线程 1 不断产生包序递增的 UDP-IP-Ethernet V2 包并放入发送 FIFO；线程 2 每隔一定时间(可调)从发送队列中读取线程 1 所产生的包并交给网卡发送出去；线程 3 以轮询的方式读取网卡收到的包并放入接收 FIFO；线程 4 读取接收 FIFO 中的包并按照协议解包，最终将裸数据打印到名为 luodataint8_t.txt 的文件中。

运行程序./l2fwd -c ff -n 2
外部参数 -c ff -n 2
-c 是允许程序最多使用的核心数 ff 代表 8(掩码 8'b1111_1111)
-n 是内存通道数，作用未知

五、附录 A-部分函数功能及接口
static void print_mbuf_send(struct rte_mbuf *m)
功能 将一个 mbuf 的信息打印到 send_data 中
参数
m 想打印的 mbuf 的指针

static void print_mbuf_recieve(struct rte_mbuf *m)
功能 将一个 mbuf 的信息打印到 recieve_data 中
参数
m 想打印的 mbuf 的指针

unsigned short cal_ip_checksum(struct ip_hdr hdr)
功能 计算 ip checksum
参数
hdr ip 头信息，用来计算 ip checksum
返回值 ip checksum

unsigned short cal_udp_checksum(struct udp_fhdr_hdr hdr, unsigned char *buffer)
功能 计算 udpchecksum

参数
hdr udp 头信息(包含伪首部)
buffer udp 层裸数据的首指针，其长度信息蕴含在 hdr 中
返回值 udp checksum

int package(struct mac_hdr mhdr, struct ip_hdr ihdr, struct udp_fhdr_hdr uhdr, unsigned char* data, struct rte_mbuf *m)
mhdr 此包的 mac 头信息
ihdr 此包的 ip 头信息
uhdr 此包的 udp 头信息(包含伪首部)
data 此包的裸数据，数据长度包含在 uhdr 和 ihdr 中
m 通过上述信息构造出的 mac 包指针，可以直接填入发送队列
返回值 整个 mac 包的字节数
注 此函数带有 padding 功能

int read_from_txt(char* a,int num)
功能 将一定数量的数据从 txt 中读取到全局数组 data_to_be_sent 中
参数
a txt 的名字
num 读取数据的数量，以 byte 为单位，num 不可以大于 2048(unsigned char data_to_be_sent[2048]={0};)

static int l2fwd_main_loop_send(void)
功能 每隔 BURST_TX_DRAIN_US 尝试从发送 fifo 中拿出 32 个包并将这些包放入发送队列并排空发送队列。如果发送 fifo 中不足 32 个包则能拿出多少就拿出多少。

static void l2fwd_main_loop_recieve(void)
功能 轮询网卡，如果有新的 mac 包则放入接收 fifo

static void l2fwd_main_p(void)
功能 按照需求产生 mac 包并放入发送 fifo

static void l2fwd_main_c(void)
功能 将接收队列中的包拿出并按照 udp 协议进行解包，解出的裸数据放入 luodataint8_t.txt 中(测试用，可以按需修改)

static int l2fwd_launch_one_lcore_send(__attribute__((unused)) void *dummy)
功能 封装 static int l2fwd_main_loop_send，以便线程调用

static int l2fwd_launch_one_lcore_recieve(__attribute__((unused)) void *dummy)
功能 封装 static int l2fwd_main_loop_recieve，以便线程调用

l2fwd_launch_one_lcore_p(__attribute__((unused)) void *dummy)
功能 封装 static int l2fwd_main_loop_p，以便线程调用

l2fwd_launch_one_lcore_c(__attribute__((unused)) void *dummy)
功能 封装 static int l2fwd_main_loop_c，以便线程调用

以“rte_”开头的函数详情请见 DPDK 官方 API DOCUMENTATION
http://dpdk.org/doc/api/