异构网络DPDK实施方案

1.方案要求

利用intel的数据平面开发工具DPDK搭建一个基本的异构通信平台。

2.实施用途环境示意图



图 1异构逻辑层实施用途环境示意图

terminal\_send是一辆移动通信车，为异构逻辑层的发送端，terminal\_recv是一台服务器，为异构逻辑层的接收端，发送端和接收端通过若干通信网或通信链路连接。异构逻辑层需要根据业务需要和各条通信链路的连接情况决定传输路径，从而保持发送端terminal\_send和接收端terminal\_recv数据能够有效送达，保持通信正常。

3.开发环境

发送端为移动通信车，接收端为linux服务器

Linux发行版：ubuntu16.04

4.DPDK优势

DPDK是intel公司推出的一款开源的数据平面开发工具，利用DPDK进行开发可以提高网络包的收发吞吐量。

5.移动通信车工作流程



图2.移动通信车工作流程图

移动通信车工作流程如下：

一．移动通信车启动，首先系统探测各个网卡的信息，构造链路探测包，先封装UDP和IP头部，然后封装MAC帧，先判断传输线路类型：

1.1以太网，查询ARP缓存，若无ARP缓存信息，则需构造ARP请求包在局域网内广播，对应接收端收到ARP请求包后构造ARP响应包发给移动通信车，移动通信车收到ARP响应包后更新ARP缓存，根据更新后的ARP缓存封装MAC帧。交付给对应线路的网关，由网关将链路探测包发送出去。

1.2广电DTMB线路。构造对应线路的硬件地址查询帧并封装，交付给对应线路的网关，由网关将链路探测包发送出去。

1.3广电FM—CDR线路。构造对应线路的硬件地址查询帧并封装，交付给对应线路的网关，由网关将链路探测包发送出去。

二．服务器端收到链路探测包后解封转链路探测包获取序号信息后先构造对应的链路探测包回包的数据部分，然后根据链路探测包解封装的信息在封装链路探测包回包的UDP、IP帧头，然后在封装链路探测包回包的MAC帧，先判断传输线路类型：

2.1以太网，查询ARP缓存，若无ARP缓存信息，则需构造ARP请求包在局域网内广播，对应接收端收到ARP请求包后构造ARP响应包发给移动通信车，移动通信车收到ARP响应包后更新ARP缓存，根据更新后的ARP缓存封装MAC帧。交付给对应线路的网关，由网关将链路探测包回包发送给移动通信车。

2.2广电DTMB线路。构造对应线路的硬件地址查询帧并封装，交付给对应线路的网关，由网关将链路探测包回包发送给移动通信车。

2.3广电FM—CDR线路。构造对应线路的硬件地址查询帧并封装，交付给对应线路的网关，由网关将链路探测包回包发送给移动通信车。

三．移动通信车等待一定时延判断接收网卡有没有收到链路探测包回包，若收到链路探测包回包则判断该传输线路为通路其STATE=ON并算出对应线路的传输时延。若过了一定时间后仍未收到某条线路的链路探测包回包则判断该线路为断路。将其STATE置为OFF，STATE为OFF的通信线路上依然尝试发送链路探测包，直到收到链路探测包回包后才将其STATE置为ON。

四．移动通信车在收发链路探测包的同时网卡从处于局域网的终端设备接收数据包，先解封装数据包的MAC帧头部。在链路探测包已探测的STATE=ON中的线路中通过策略选择算法选择最优传输线路，封装LLC帧头，然后封装数据包UDP/IP帧。再封装MAC帧，先判断传输线路类型：

4.1以太网，查询ARP缓存，若无ARP缓存信息，则需构造ARP请求包在局域网内广播，对应接收端收到ARP请求包后构造ARP响应包发给移动通信车，移动通信车收到ARP响应包后更新ARP缓存，根据更新后的ARP缓存封装MAC帧。交付给对应线路的网关，由网关将数据包发送出去。

4.2广电DTMB线路。构造对应线路的硬件地址查询帧并封装，交付给对应线路的网关，由网关将数据包发送出去。

4.3广电FM—CDR线路。构造对应线路的硬件地址查询帧并封装，交付给对应线路的网关，由网关将数据包发送出去。

五.服务器端接收到业务数据包后，解封装业务数据包后获取数据包LLC帧中的序号信息，构造相应确认序号的ACK确认包，，在一定情况下服务器端将构造数据包和ACK合包进行发送。先封装ACK确认包的UDP、IP帧头，然后封装MAC帧，判断传输线路类型：

5.1以太网，查询ARP缓存，若无ARP缓存信息，则需构造ARP请求包在局域网内广播，对应接收端收到ARP请求包后构造ARP响应包发给移动通信车，移动通信车收到ARP响应包后更新ARP缓存，根据更新后的ARP缓存封装MAC帧。交付给对应线路的网关，由网关将数据包发送到移动通信车端。

5.2广电DTMB线路。构造对应线路的硬件地址查询帧并封装，交付给对应线路的网关，由网关将数据包发送到移动通信车端。

5.3广电FM—CDR线路。构造对应线路的硬件地址查询帧并封装，交付给对应线路的网关，由网关将数据包发送到移动通信车端。

在构造发送ACK确认包的同时，服务器端将解封装后的原始业务数据包先判断业务数据包是否存在分片情况，如果存在分片情况则将其进行放入接收端窗口进行整序，然后进行转发。

六.移动通信车接收端接收ACK确认包后根据确认信息调整发送端窗口。若迟迟未收到某个数据包的ACK确认包，则重传该数据包。

6.UDP封装帧

6.1UDP封装数据帧结构



图3.UDP封装数据帧结构

6.1.1数据包封装UDP隧道帧格式（49+若干字节）

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| MAC帧头 | IP帧头部 | UDP帧头部 | LLC帧头 | IP数据包 | MAC帧尾部 |

MAC帧头部：14字节，包含发送帧的MAC头部信息。

IP帧头部：20字节，包含发送帧的IP信息。

UDP帧头部：8字节，包含发送帧的UDP信息。

LLC帧头部：4字节，包含发送帧的LLC控制信息。

IP数据包：业务数据，若干字节。

MAC帧尾部：4字节，包含发送帧的MAC帧尾部信息。

6.1.2业务数据包格式（4+若干字节）

|  |  |  |  |
| --- | --- | --- | --- |
| TYPE | 数据包序号 | 分片号 | IP数据包 |

TYPE：1个字节，表明数据类型。

数据包序号：2个字节，表明数据包序号。

分片号：1个字节，若数据包存在分片表明分片号。

IP数据包：若干字节。

6.1.3ACK包格式（3个字节）

|  |  |
| --- | --- |
| TYPE | ACK序号 |

TYPE：1个字节，表明数据类型。

ACK序号：2个字节，表明ACK确认收到的数据包序号。

6.1.4数据和ACK合包（34+若干字节）

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| TYPE | 数据包序号 | 分片号 | IP数据包 | ACK序号 |

TYPE：1个字节，表明数据类型。

数据包序号：2个字节，表明数据包序号。

分片号：1个字节，若数据包存在分片表明分片号。

IP数据包：若干字节。

ACK序号：2个字节，表明ACK确认序号。

6.1.5RESET LLC帧格式（4字节）

|  |  |
| --- | --- |
| TYPE | 参数 |

TYPE：1个字节，表明数据类型。

参数：3个字节，表明要重置的参数。

6.2链路控制帧UDP封装帧结构



图4.链路控制帧UDP封装帧结构

6.2.1链路控制帧UDP隧道封装格式（53字节）

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| MAC帧头部 | IP帧头部 | UDP帧头部 | 数据部分（TLV） | MAC帧尾部 |

MAC帧头部：14字节，包含发送帧的MAC头部信息。

IP帧头部：20字节，包含发送帧的IP信息。

UDP帧头部：8字节，包含发送帧的UDP信息。

数据包：包含ARP请求包和ARP响应包控制帧，7字节。

MAC帧尾部：4字节，包含发送帧的MAC帧尾部信息。

6.2.2链路探测包格式（7字节）

|  |  |  |
| --- | --- | --- |
| Tag（数据类型） | Length（长度） | Value（事务号） |

数据类型：1个字节，表明链路探测包类型。

长度：4个字节，表明包格式中各部分长度。

事务号：2个字节，发送的链路探测包序号。

6.2.3链路探测包回包格式（7字节）

|  |  |  |
| --- | --- | --- |
| Tag（数据类型） | Length（长度） | Value（事务号） |

数据类型：1个字节，表明链路探测包回包类型。

长度：4个字节，表明包格式中各部分长度。

事务号：2个字节，确认的链路探测包序号。

6.2.4ARP请求/响应包格式（27字节）

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 硬件类型 | 协议类型 | 硬件地址长度 | 协议地址长度 | 数据类型 | 源MAC地址 | 目的MAC地址 | 源IP地址 | 目的IP地址 |

硬件类型：2个字节，表明ARP实现在何种类型的网络上。值为1，表示以太网。

协议类型：2个字节，表示要映射的协议地址类型。IP：0800。

硬件地址长度：1个字节，表示MAC地址长度，其值为6个字节。

协议地址长度：1个字节，表示IP地址长度，其值为4个字节。

操作类型：2个字节，表示ARP数据包类型，值为1表示ARP请求，值为2表示ARP应答。

源MAC地址：6个字节，表示发送端MAC地址。

源IP地址:4个字节，表示发送端IP地址。

目的MAC地址：6个字节，表示目标设备的MAC物理地址，在ARP请求包里为空。

目的IP地址：4个字节，表示目标设备的IP地址。

7.异构逻辑层转发示意图

异构逻辑层提供异构网络接入能力，包括网络状况测量、网络策略选择、数据包的分类与整序、数据包的切片与重组等。

Terminal\_send与terminal\_recv的之间包传输的基本架构如下：



图 5异构逻辑层包类型

包类型一种有7种，分别为数据包、数据ACK包、链路探测包、链路探测回包、ARP请求包、ARP响应包、数据和ACK确认合包。每种分别在发送端和接收端都有响应的处理流程。

7.1发送端帧处理总体流程图



图 6发送端处理流程图

7.2接收端帧处理总体流程图



图 7接收端总体流程图

7.3数据包

发送端：



图 8数据包发送端流程图

发送端从发送队列中取数据，首先查询网络状态信息，选择状态STATE=ON的传输线路，然后从STATE=ON的传输线路中根据网络策略选择算法选择相应的网络通路进行发送。先判断选择的传输线路是否为以太网，若为以太网先封装LLC帧头，然后构造相应得UDP/IP数据帧头，之后查询ARP缓存，如果有构造相应的MAC帧从相应网卡发送，如果没有则申请ARP查询，查询之后更新ARP缓存，之后通知相应的处理流程查询ARP缓存，构造相应MAC帧，传递到网卡进行发送。

接收端：



图 9数据包接收端流程图

轮询网卡从网络接收数据，首先先根据UDP端口号判断是否为数据帧，若不为数据帧则启动链路控制帧的处理流程，然后解封装数据帧根据LLC帧头判断是否为数据包，若不为数据包再判断是否为数据包ACK合包，若为数据包ACK合包则读取包中的ACK信息，同时启动ACK处理流程。然后判断数据包是否存在分片情况，若数据包存在分片情况则将数据包分片放入接收窗口进行整序。再将接收的数据包进行整序，同时启动发送数据包ACK的处理流程。最后启动下一步数据包的转发流程。

7.4数据ACK包

接收端：



图10数据ACK包发送流程图

启动数据ACK包的构造与发送流程，先从数据包中提取相关的参数信息，如UDP/IP信息、参数信息等。然后判断是否需要构造数据ACK合包，若需要的话则构造数据ACK包合包，否则构造ACK包，再在包头添加LLC、UDP、IP帧头，然后根据数据包的LLC信息判定传输线路类型是否为以太网，若不是以太网则进行相应线路的MAC帧封装，否则查询ARP缓存，若无ARP缓存则需申请ARP查询并更新ARP缓存表。最后构造相应的MAC帧头，将数据ACK包发送出去。

发送端：



图11数据ACK包接收流程图

发送端的ack模块从网卡上读取mbuf，先根据端口号判断是否为数据帧，然后读取数据包的LLC帧头，判断是否为ack确认包，若不是则判断是否为数据包ACK合包，若是数据包ACK合包则启动数据包的接收流程。再进行帧解析，此时已获得数据ack包的确认信息，根据已得到的ack信息进行对发送端发送窗口的操作。最后根据发送定时器决定新传还是重传数据帧。

7.5链路探测包

发送端：



图12链路探测包发送流程图

定时器每隔固定时间给CPU一个中断，启动心跳探测包发送流程。首先选择一个传输线路类型，先判断要发送的线路是否为以太网，若是则根据帧格式构造心跳探测包。再封装UDP/IP帧头，之后查询ARP缓存，找下一跳的MAC地址，如果有则构造相应的MAC帧发送，如果没有则申请ARP查询操作，待获得下一跳MAC地址后，更新ARP缓存，通知相应的流程构造MAC帧，最后通过网卡发送出去。

接收端：



图13链路探测包接收流程图

轮询网卡接收到数据之后，先根据端口号判断是否为链路控制帧，再判断是否为链路探测包，如果是链路探测包，则进行帧的参数解析，提取参数之后就启动链路探测包回包的处理流程，将参数传递到链路探测包回包。

7.6链路探测包回包

接收端：



图14链路探测包回包发送流程图

由前面的程序链路探测包回包启动流程之后，开始链路探测包回包构造过程。首先选择传输线路类型，同时获取前面链路探测包的UDP/IP等信息。之后根据信息构造链路探测包回包，封装UDP/IP帧头，再判断传输线路是否为以太网，若为以太网然后查询ARP缓存。如果存在，则申请ARP查询，查询到结果之后更新ARP缓存，提请相应的处理程序构造对应网络的MAC帧，之后选择从对应网卡发出。

发送端：



图15链路探测包回包接收处理流程图

轮询网卡接收数据，首先根据端口号判断是否为链路控制帧，若为则判断是否为链路探测包回包，若是则启动探测包回包的处理流程，首先进行帧解析，提取帧里面的参数信息，然后根据算法计算链路时延，最后更新相关的状态参数信息。

7.7ARP请求包

发送端：

****

图 16 ARP请求包发送端处理流程图

发送端申请ARP查询，先构造ARP请求包，构造成功后判断目的IP地址与发送端设备是否处于同一局域网，若处于同一局域网则在局域网内广播发送ARP请求包。否则则将ARP请求包交付给发送端的网关进行处理。

接收端：



图 17 ARP请求包接收端处理流程图

接收端的轮训网卡接收数据包，先根据端口号判断是否为链路控制帧，若是则判断是否为ARP请求包，若是则进行帧解析，根据数据包中信息判断ARP请求包中的目标IP地址与接收端设备的IP地址是否一致，若不一致则丢弃数据包。然后根据包中信息更新接收端ARP缓存表。最后启动ARP响应包发送流程。

7.8ARP响应包

发送端：



图 18 ARP响应包发送端处理流程图

发送端的轮训网卡接收数据包，先根据端口号判断是否为链路控制帧，若是则判断是否为ARP应答包，若不是则转到其他操作，再进行帧解析，根据数据包里的信息更新发送端的ARP缓存表。

接收端：



图 19 ARP响应包接收端处理流程图

接收端先从网卡接收ARP请求包，然后根据帧解析出的源IP地址和源MAC地址以及自身的IP地址和MAC地址构造ARP响应包，封装MAC帧，最后通过轮训网卡发送出去。

8.累计确认与超时重传机制

8.1累计确认

8.1.1接收端处理模块

接收端建立一定大小的发送窗口，接收端对接收的数据包mbuf进行累积确认，当收到某个序号数据包后，将该数据包按序插入接收端窗口，同时判断该数据包序号前的所有数据包mbuf是否收到，若该序号前所有数据包都收到，则发送该序号数据包的ack确认包。若该序号前有数据包mbuf未收到，则发送第一个未收到数据包的前一个序号的ack确认包。

示意图如下（假设接收端部分窗口size为6）：

1.起始接收端部分窗口大小为6，接收序号1-6的mbuf

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |

2.假定收到了mbuf1，2，3，6的数据包mbuf，则将mbuf1，2，3，6按序插入接收窗口。

并发送序号为3的ack确认包。

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| mbuf1 | mbuf2 | mbuf3 |  |  | mbuf6 |

3.1若之后接收端收到了mbuf4,5的数据包，则将mbuf4,5按序插入接收窗口。并发送序号为6的ack确认包。

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| mbuf1 | mbuf2 | mbuf3 | mbuf4 | mbuf5 | mbuf6 |

3.2若在时延期间内还未收到mbuf4,5的数据包，则仍发送序号为3的ack确认包不变。

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| mbuf1 | mbuf2 | mbuf3 |  |  | mbuf6 |

8.1.2发送端处理模块

发送端建立一个大小为size的发送窗口，窗口中可储存size个封装好的数据包mbuf，发送端对发送的数据包mbuf的ack进行累积确认，当收到某个数据包的ack包，则将其移出发送窗口，同时该数据包右边的队列右移，窗口进入一个新的数据包mbuf。

示意图如下（假设发送端窗口size为6）：

1.起始发送窗口大小为6，发送序号1-6的mbuf

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| mbuf1 | mbuf2 | mbuf3 | mbuf4 | mbuf5 | mbuf6 |

2.假定收到了mbuf3的ack确认包，将mbuf3及之前的mbuf移出窗口，同时窗口右移，发送窗口进入新的mbuf7，8，9。

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| mbuf4 | mbuf5 | mbuf6 | mbuf7 | mbuf8 | mbuf9 |

8.2超时重传



图 20 超时重传流程图

发送端给发送窗口中将要发送的数据设置N秒的定时器，若N秒后仍未收到发送窗口中数据包的确认，则重传该数据包。

9．链路时延测量算法

9.1链路探测响应包测量时延

9.1.1按序到达



图 21 链路探测包测量时延流程图

由于传输线路在某个时段可能出现一定波动，所以不能仅通过单个序号包的RTT(往返时延)去判定线路的通断情况以及线路时延。而应该通过平滑RTT来判定线路时延。

当发送端接收到一个链路探测包回包后，计算其往返时延RTT，其平滑RTT估计值SRTT：

其中g为新RTT样本占SRTT估计值权重，取为0.125。

9.1.2失序到达

9.1.3帧丢失



图 22 帧丢失判断通断流程图

根据帧丢失情况判断线路通断情况：接收端设置起始线路状态STATE=ON，当线路断开时，发送端无法收到接收端返回的链路探测包，每当发送一个序号为N的链路探测包后，设置一个计时器T，在T=4\*（SRTT+100ms）时间后还未收到对应的链路探测包回包。则依次判定序号为N-N+4的链路探测包是否收到了回包。若均为收到回包则判定该线路已断开，STATE=OFF，同时将SRTT置零。否则为维持该线路STATE=ON，同时未收到ACK响应的包不计入线路时延计算。

9.2链路探测包回包与数据ACK包混合测量