第五周实验报告

交换机学习实验

2015K8009922021

李一苇

**一、实验内容**

实现一个具有转发表学习功能的交换机，使得在非广播网络中连接到同一交换机的主机能通信。

并在此基础上进行iperf，对比交换机转发和集线器广播的性能

**二、实验流程**

本实验分以下步骤进行

1. 实现mac.c里缺失的mac表操作：查找、插入、删除

结合上一周实验中关于链表操作的部分，可以较简单地写出三个操作的函数。

需要注意的点：

visited属性在插入和查找时都需要更新成time(NULL)，即最新时间；

插入和删除操作都需要对链表结构进行修改，在多线程下可能会有问题：如插入到一半的链表结点的前一个结点被恰巧删除，等等潜在风险，因此需要用pthread中的上锁去锁；

涉及到mac的快速查找，老师已经对映射表加了简单的哈希链表结构，因此直接利用hash8函数得到mac的哈希值，存到对应的链表结构中。

具体实现细节在代码中说明。

1. 实现交换机处理数据包的操作handle\_packet函数

该函数的实现代码如下：

struct ether\_header \*eh = (struct ether\_header \*)packet; ETHER\_STRING ".", ETHER\_FMT(eh->ether\_dhost), ETHER\_FMT(eh->ether\_shost), ETHER\_FMT(iface->mac));

iface\_info\_t \*find\_result = lookup\_port(eh->ether\_dhost);

if (find\_result != NULL) {

iface\_send\_packet(find\_result, packet, len);

} else {

broadcast\_packet(iface, packet, len);

}

if (lookup\_port(eh->ether\_shost) == NULL) {

insert\_mac\_port(eh->ether\_shost, iface);

}

需要说明的几个点（也是我曾因理解不够彻底而犯的错）：

储存目的主机条目和储存源主机条目的操作是独立的，一次发送中不能二选其一；

但直接发送到目的主机和广播发送操作是二选其一的。

在储存源主机条目时，也需要先查表，否则会存入重复的值。

iface->mac是和源主机有关的值，但并不是数据包里的源主机的mac值。

1. 正确处理多线程删除

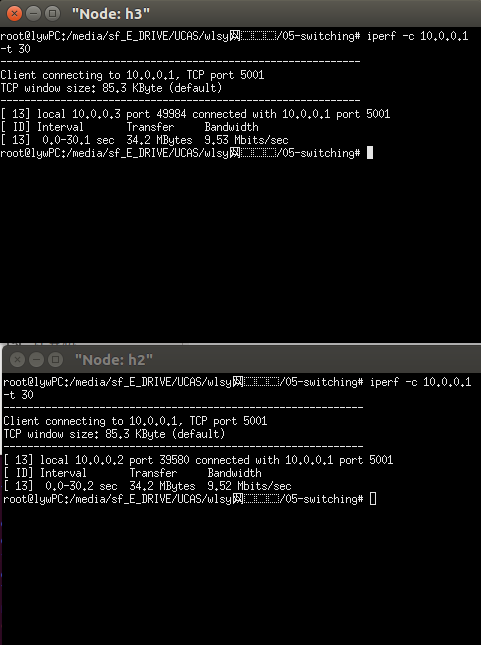
这一部分，因为老师已经事前给出了多线程示例程序和写好了sweeping\_mac\_port\_thread这个始终运行的线程，所以没有特殊要写的代码。

只是需要注意一点：在删掉老化的条目时，应该及时执行list\_delete\_entry(&entry->list)和free(entry)，以释放内存。

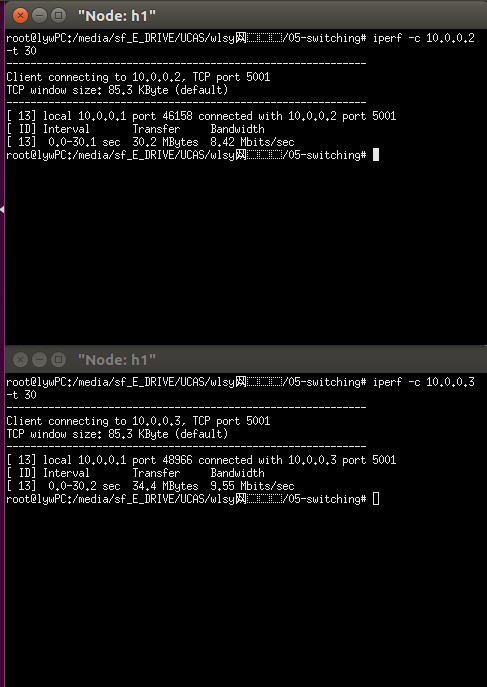
1. **实验结果和分析**

用iperf验证广播网络的效率

结果1）h1作服务器，h2和h3同时连接h1的效果：



h2和h3作服务器，h1做客户端同时连接的效果



分析：在前一种情况下，h2和h3共同占用了h1到集线器的20Mb/s带宽，以及他们各自到集线器的10Mb/s带宽，因此发挥了最大效能，带宽评估接近10Mb/s。所以这个结果里交换机和集线器是一样的。

在后一种情况下，由于转发器只将数据包发送给目的主机，不再广播占用其他带宽，所以也可以达到二者的近似最大带宽10Mb/s，和上周的实验不一样。