

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。

- 2) 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。

3) 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。

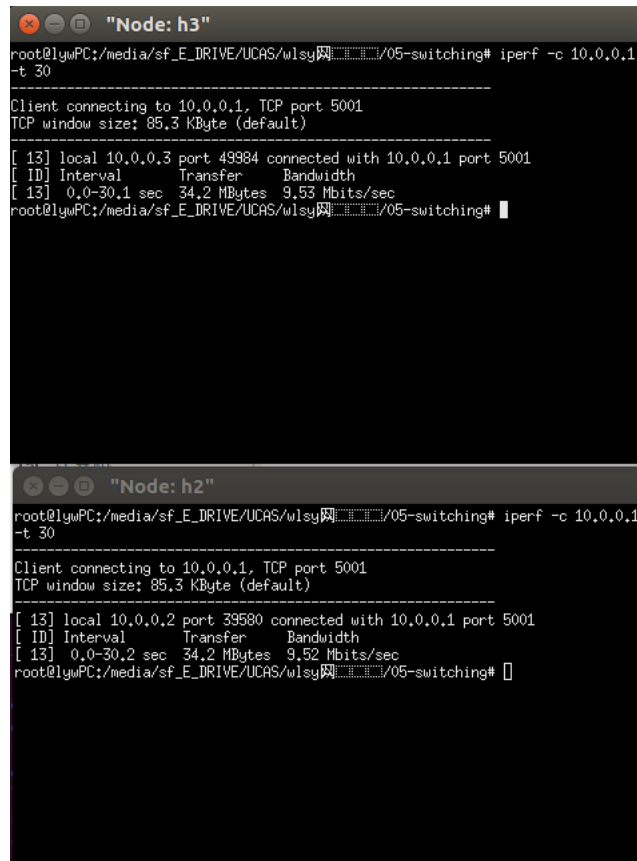
只是需要注意一点：在删掉老化的条目时，应该及时执行

list_delete_entry(&entry->list)和 free(entry)，以释放内存。

三、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：



The image shows two terminal windows. The top window is titled "Node: h3" and shows the command 'iperf -c 10.0.0.1 -t 30' being executed. It displays a successful connection to 10.0.0.1 on port 5001 and a performance summary: 34.2 MBytes transferred at 9.53 Mbits/sec over a 30-second interval. The bottom window is titled "Node: h2" and shows the same command being executed. It also displays a successful connection to 10.0.0.1 on port 5001 and a performance summary: 34.2 MBytes transferred at 9.52 Mbits/sec over a 30-second interval.

```
root@lywPC:/media/sf_E_DRIVE/UCAS/wlsy网[REDACTED]/05-switching# iperf -c 10.0.0.1 -t 30
Client connecting to 10.0.0.1, TCP port 5001
TCP window size: 85.3 KByte (default)
[ 13] local 10.0.0.3 port 49984 connected with 10.0.0.1 port 5001
[ ID] Interval      Transfer    Bandwidth
[ 13] 0.0-30.1 sec  34.2 MBytes  9.53 Mbits/sec
root@lywPC:/media/sf_E_DRIVE/UCAS/wlsy网[REDACTED]/05-switching#
```

```
root@lywPC:/media/sf_E_DRIVE/UCAS/wlsy网[REDACTED]/05-switching# iperf -c 10.0.0.1 -t 30
Client connecting to 10.0.0.1, TCP port 5001
TCP window size: 85.3 KByte (default)
[ 13] local 10.0.0.2 port 39580 connected with 10.0.0.1 port 5001
[ ID] Interval      Transfer    Bandwidth
[ 13] 0.0-30.2 sec  34.2 MBytes  9.52 Mbits/sec
root@lywPC:/media/sf_E_DRIVE/UCAS/wlsy网[REDACTED]/05-switching#
```

h2 和 h3 作服务器，h1 做客户端同时连接的效果

```
"Node: h1"
root@lywPC:/media/sf_E_DRIVE/UCAS/wlsy网[REDACTED]/05-switching# iperf -c 10.0.0.2
-t 30
-----
Client connecting to 10.0.0.2, TCP port 5001
TCP window size: 85.3 KByte (default)
-----
[ 13] local 10.0.0.1 port 46158 connected with 10.0.0.2 port 5001
[ ID] Interval      Transfer    Bandwidth
[ 13] 0.0-30.1 sec  30.2 MBytes  8.42 Mbits/sec
root@lywPC:/media/sf_E_DRIVE/UCAS/wlsy网[REDACTED]/05-switching#

"Node: h1"
root@lywPC:/media/sf_E_DRIVE/UCAS/wlsy网[REDACTED]/05-switching# iperf -c 10.0.0.3
-t 30
-----
Client connecting to 10.0.0.3, TCP port 5001
TCP window size: 85.3 KByte (default)
-----
[ 13] local 10.0.0.1 port 48966 connected with 10.0.0.3 port 5001
[ ID] Interval      Transfer    Bandwidth
[ 13] 0.0-30.2 sec  34.4 MBytes  9.55 Mbits/sec
root@lywPC:/media/sf_E_DRIVE/UCAS/wlsy网[REDACTED]/05-switching#
```

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。