

06-switch-report

作者：苗屹松

1.Overview:

老师在PPT中已经把实验分割成很清晰的 sub-tasks了(i.e. 查询、插入、老化操作)
所以我只说一些自己踩过的坑。

2.踩过的坑:

(a)关于源地址、目的地址的获取:

```
struct ether_header *eh = (struct ether_header *)packet;
```

就可以直接获得mac地址。

但是我依然疑惑，packet是char* 类型的，怎么就能获得源地址、目的地址呢？

我猜测，packet虽然是char*，但里面包含了这些信息！

(b)关于entry结构中的next指针:

```
struct mac_port_entry {
    uint8_t mac[ETH_ALEN];
    iface_info_t *iface;
    time_t visited;
    struct mac_port_entry *next;
};

typedef struct {
    mac_port_entry_t *hash_table[HASH_8BITS];
    pthread_mutex_t lock;
    pthread_mutexattr_t attr;
    pthread_t tid;
} mac_port_map_t;
```

在这个数据结构里，有趣的是mac_port_entry_t *hash_table[HASH_8BITS];
它是一个长度256的数组，每个都包含了一个指向mac_port_entry_t 的指针。

我一开始以为*next就是把所有的entry都连接起来，但这样是错误的。

其实是：我们先做hash()，把hash结果相同的entry放到相同的hash_table[hash_index]里

*next只不过是把hash_table[hash_index]里的entry连在一起！

(谢谢武老师10月26日在教室告诉我正确的思路)

(c)删除指针:

老化并free()之后，我发现在map里还有这个entry！

我debug了很久，最后发现，应该让

```
ptr = NULL;
```

(d)删除之后，next指针的操作：

```
if(current_entry != tmp && current_entry->next){
    previous_entry->next = current_entry->next;
}
if(current_entry == tmp && current_entry->next){
    mac_port_map.hash_table[i] = current_entry->next;
}
```

如果删除的entry在中间位置，就把它的左右邻居连起来

如果删除的entry在头部，就把链表的头置为它的next.

3.测试结果

(a)ping测试：

为了加快测试速度，我们这里让老化时间为3 seconds

我们先 h1 ping h3

过了两秒

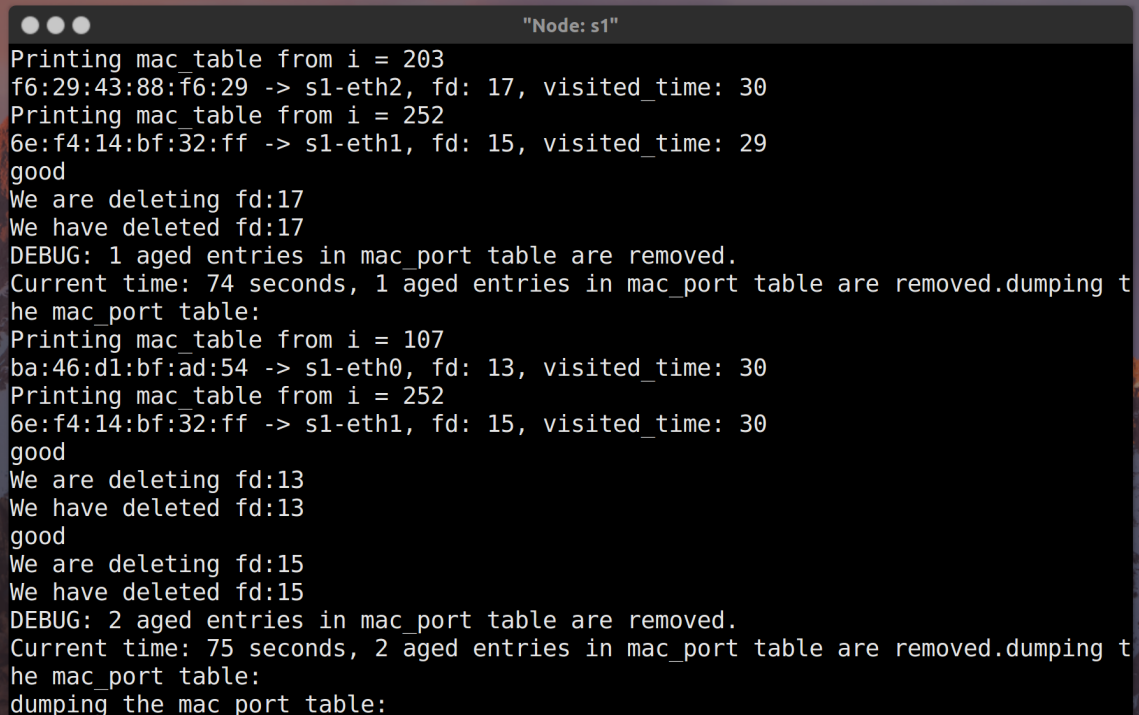
h2 ping h1

在 switch_ping_result.txt 里，我们可以看到，h3(fd = 17)最先被老化掉

然后h1 和 h2 在同一此sweep操作中被删去。

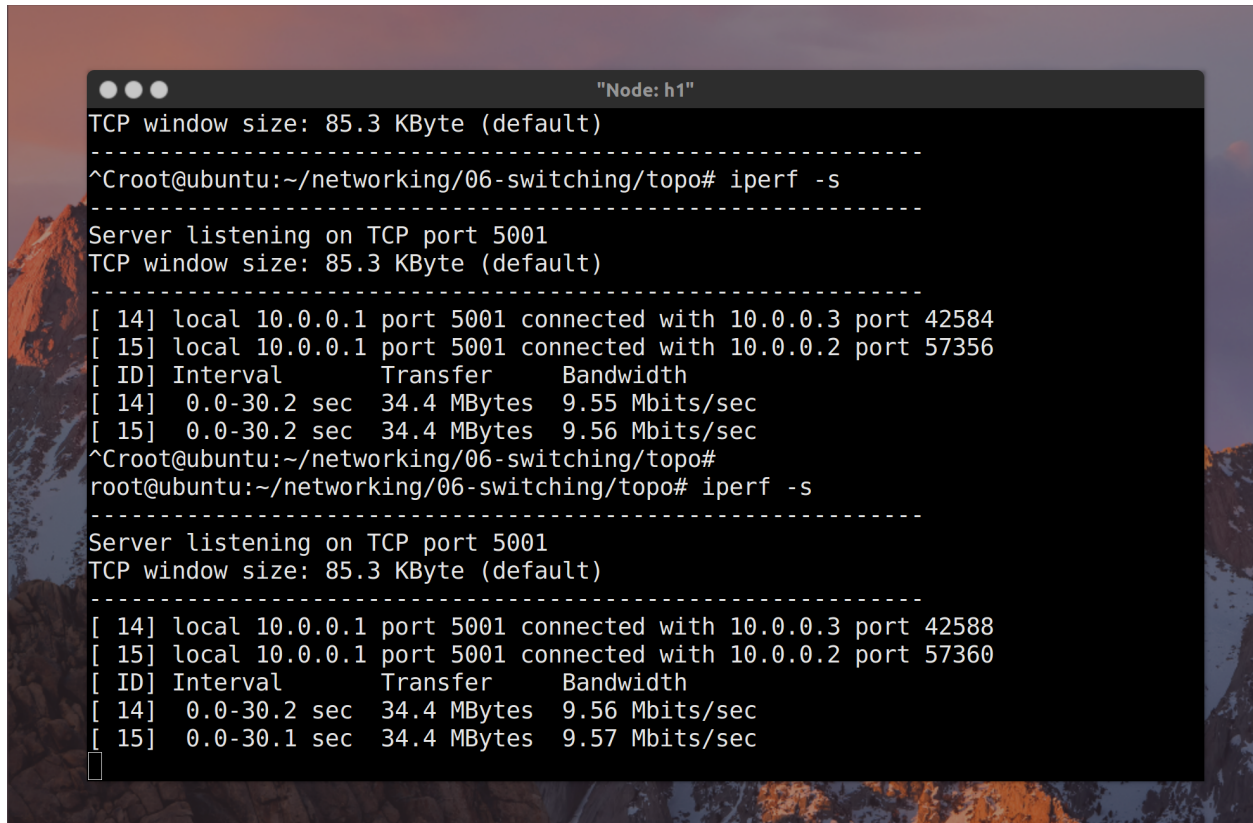
(b)iperf测试：

h1 为 server, h2, h3 为client, 并同时与h1连接。



```
"Node: s1"
Printing mac table from i = 203
f6:29:43:88:f6:29 -> s1-eth2, fd: 17, visited_time: 30
Printing mac_table from i = 252
6e:f4:14:bf:32:ff -> s1-eth1, fd: 15, visited_time: 29
good
We are deleting fd:17
We have deleted fd:17
DEBUG: 1 aged entries in mac_port table are removed.
Current time: 74 seconds, 1 aged entries in mac_port table are removed.dumping t
he mac_port table:
Printing mac_table from i = 107
ba:46:d1:bf:ad:54 -> s1-eth0, fd: 13, visited_time: 30
Printing mac table from i = 252
6e:f4:14:bf:32:ff -> s1-eth1, fd: 15, visited_time: 30
good
We are deleting fd:13
We have deleted fd:13
good
We are deleting fd:15
We have deleted fd:15
DEBUG: 2 aged entries in mac_port table are removed.
Current time: 75 seconds, 2 aged entries in mac_port table are removed.dumping t
he mac_port table:
dumping the mac_port table:
```

在上图我们可以看到，在74秒，h3被老化(因为它比h2稍早一点连接)
在75秒，h1,h2同时被老化。



```
"Node: h1"
TCP window size: 85.3 KByte (default)
-----
^Croot@ubuntu:~/networking/06-switching/topo# iperf -s
-----
Server listening on TCP port 5001
TCP window size: 85.3 KByte (default)
-----
[ 14] local 10.0.0.1 port 5001 connected with 10.0.0.3 port 42584
[ 15] local 10.0.0.1 port 5001 connected with 10.0.0.2 port 57356
[ ID] Interval      Transfer    Bandwidth
[ 14]  0.0-30.2 sec  34.4 MBytes  9.55 Mbits/sec
[ 15]  0.0-30.2 sec  34.4 MBytes  9.56 Mbits/sec
^Croot@ubuntu:~/networking/06-switching/topo#
root@ubuntu:~/networking/06-switching/topo# iperf -s
-----
Server listening on TCP port 5001
TCP window size: 85.3 KByte (default)
-----
[ 14] local 10.0.0.1 port 5001 connected with 10.0.0.3 port 42588
[ 15] local 10.0.0.1 port 5001 connected with 10.0.0.2 port 57360
[ ID] Interval      Transfer    Bandwidth
[ 14]  0.0-30.2 sec  34.4 MBytes  9.56 Mbits/sec
[ 15]  0.0-30.1 sec  34.4 MBytes  9.57 Mbits/sec
█
```

我们可以看到，带宽比广播网络提高了50%（上次为6.3Mbits/sec）