交换机转发实验

孙佳钰 2015K8009929051

2018 年 9 月 29 日

1 实验内容

实现交换机转发表的学习算法，共包括三个操作：

- 查询操作：每收到一个数据包，根据目的 MAC 地址查询转发表：如果查询到对应条目，就将数据包转发到相应端口，并更新访问时间；否则就广播该数据包。
- 插入操作：每收到一个数据包，根据源 MAC 地址查询转发表：如果查询到对应条目，就更近访问时间；否则就将源 MAC 地址插入转发表。
- 老化操作：删除超过30秒未访问的转发表条目。

2 实验流程

1. 网络拓扑已实现好。传入一个数据包时，从 main.c 里的 *handle_packet* 函数开始处理，需要先查找目的地址，然后插入源地址。

```
1  void handle_packet(iface_info_t *iface, char *packet, int len)
2  {
3      struct ether_header *eh = (struct ether_header *)packet;
4      log(DEBUG, "the dst mac address is " ETHER_STRING ".\n", ETHER_FMT
5  (eh->ether_dhost));
6
7      iface_info_t *res = lookup_port(eh->ether_dhost);
8      if (res)
9          iface_send_packet(res, packet, len);
10     else
11         broadcast_packet(iface, packet, len);
12     insert_mac_port(eh->ether_shost, iface);
13     // dump_mac_port_table();
14 }
```

2. mac.c 中的 *lookup_port* 函数负责在转发表中查询 MAC 地址，转发表使用哈希储存，采用链地址法处理冲突。所以将 MAC 地址哈希后对链进行逐项比对：找到则更新使用时间并返回对应的 iface；否则返回 NULL。

```
iface_info_t *lookup_port(u8 mac[ETH_ALEN])
{
    uint8_t index = hash8((char*)mac, ETH_ALEN);
    int flag;
    pthread_mutex_lock(&mac_port_map.lock);
    mac_port_entry_t *entry;
    list_for_each_entry(entry, &mac_port_map.hash_table[index], list) {
        flag = 1;
        for (int i = 0; i < ETH_ALEN; i++) {
            if (mac[i] != entry->mac[i])
            flag = 0;
        }
        if (flag) {
            printf("Index %u exists.\n", index);
            entry->visited = time(NULL);
            pthread_mutex_unlock(&mac_port_map.lock);
            return entry->iface;
        }
    }
    printf("Index %u does not exist.\n", index);
    pthread_mutex_unlock(&mac_port_map.lock);
    return NULL;
}
```

3.mac.c 中的 *insert_mac_port* 函数负责在转发表中插入 MAC 地址。它先在转发表中查询 MAC 地址：若找到，则更新使用时间并返回；否则就新创建一个转发表条目，将当前 MAC 地址和 iface 内容存入后插入到转发表中。

```
void insert_mac_port(u8 mac[ETH_ALEN], iface_info_t *iface)
{
    iface_info_t *res = lookup_port(mac);
    if (res) {
        (list_entry(res, mac_port_entry_t, iface))->visited = time(NU
LL);
        return;
    }
```

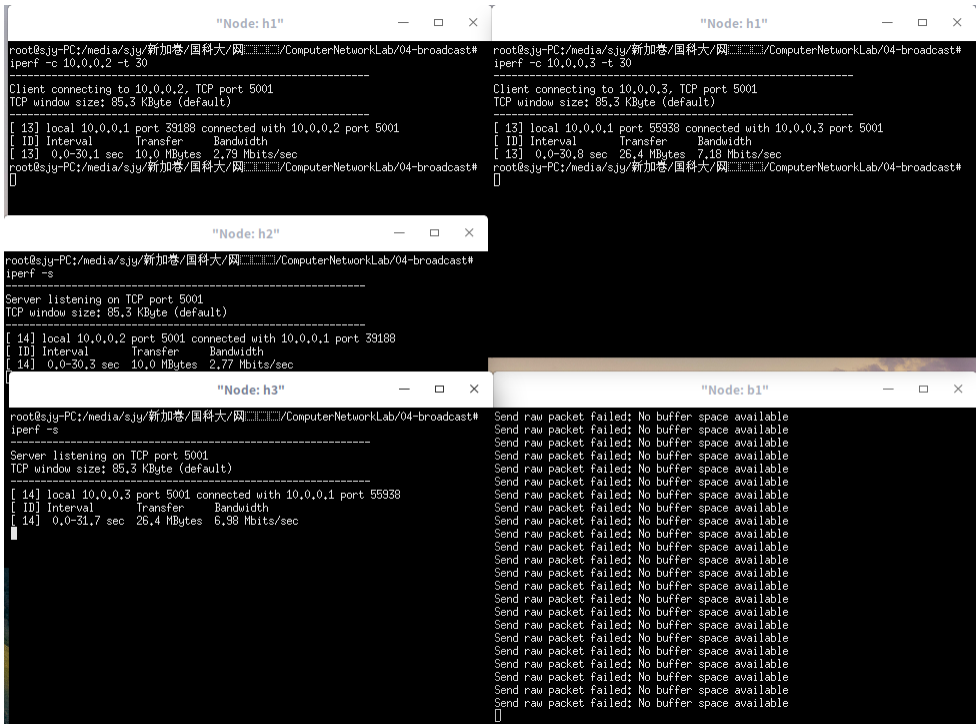
```
    uint8_t index = hash8((char*)mac, ETH_ALEN);
    pthread_mutex_lock(&mac_port_map.lock);
    mac_port_entry_t *entry;
    list_for_each_entry(entry, &mac_port_map.hash_table[index], list);
    mac_port_entry_t *tmp = malloc(sizeof(mac_port_entry_t));
    for (int i = 0; i < ETH_ALEN; i++)
        tmp->mac[i] = mac[i];
    tmp->iface = iface;
    tmp->visited = time(NULL);
    tmp->list.next = entry->list.next;
    tmp->list.prev = &(entry->list);
    entry->list.next->prev = &(tmp->list);
    entry->list.next = &(tmp->list);
    printf("Iface %s inserted.\n", iface->name);
    pthread_mutex_unlock(&mac_port_map.lock);
}
```

4. mac.c 中的 *sweep_aged_mac_port_entry* 函数与查询插入函数并发运行，负责扫描所有查询表条目，发现超过30没有使用的条目就将其删除。

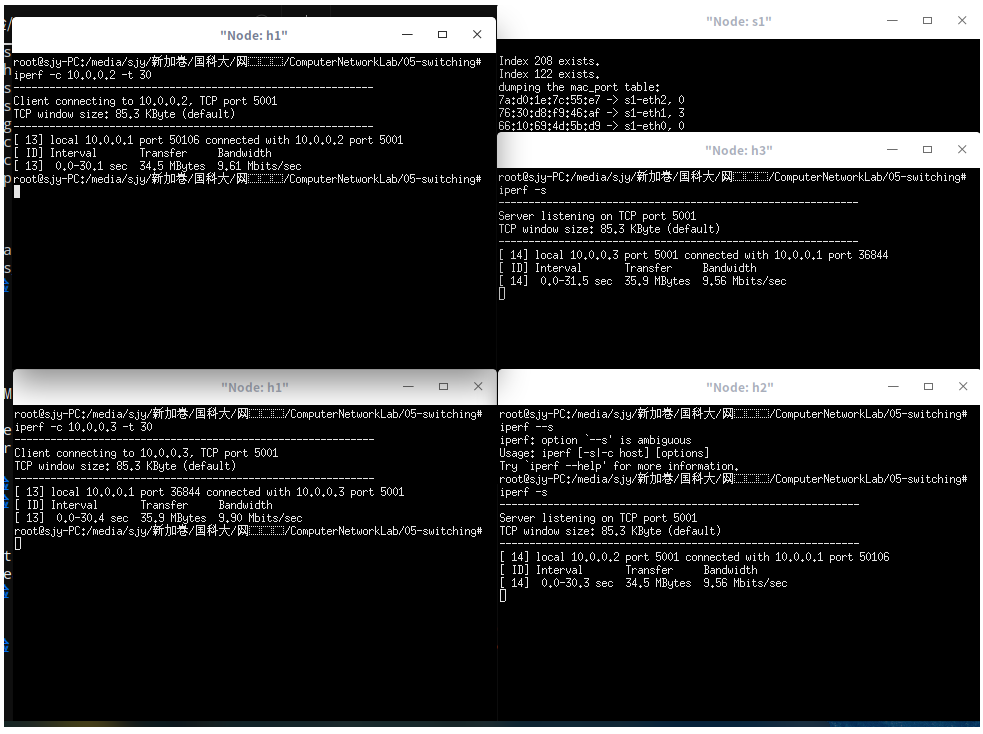
```
int sweep_aged_mac_port_entry()
{
    int num = 0;
    time_t now = time(NULL);
    pthread_mutex_lock(&mac_port_map.lock);
    mac_port_entry_t *entry;
    for (int i = 0; i < HASH_8BITS; i++) {
        list_for_each_entry(entry, &mac_port_map.hash_table[i], list) {
            if (now - entry->visited > MAC_PORT_TIMEOUT) {
                printf("Iface %s deleted.\n", entry->iface->name);
                list_delete_entry(&entry->list);
                // free(entry);
                num++;
            }
        }
    }
    pthread_mutex_unlock(&mac_port_map.lock);
    return num;
}
```

完整实验代码可见所附文件。

3 实验结果及分析



h1 作为 client，h2 和 h3 作为 server。上图是采用 hub 时利用 iperf 测量的带宽。



上图是采用 switch 时利用 iperf 测量的带宽。发现带宽的确增加了。