

计算机网络实验报告-05

阮星程

2015K8009929047

一、 实验题目

交换机转发实验。

二、 实验内容

实现对数据结构 `mac_port_map` 的所有操作，以及对数据包的转发和广播操作，从而搭建一个较为完整的交换机。

利用 `iperf` 和给定的拓扑网络进行实验，对比交换机转发与集线器广播的性能差异。

三、 实验过程

开始逐步构建各个函数

```
//search for corresponding iface. if not exist, return NULL
iface_info_t *lookup_port(u8 mac[ETH_ALEN])
{
    pthread_mutex_lock(&mac_port_map.lock);
    iface_info_t *iface = NULL;

    //if corresponding iface exist
    if(mac_port_map.hash_table[mac[0]])
    {
        for(i = 0; i < ETH_ALEN; i++)
        {
            if(mac_port_map.hash_table[mac[0]]->mac[i] != mac[i])
                return NULL;
            iface = mac_port_map.hash_table[mac[0]]->iface;
            mac_port_map.hash_table[mac[0]]->visited = time(NULL);
        }
        pthread_mutex_unlock(&mac_port_map.lock);
        return iface;
    }

    //insert new transform table entry
    void insert_mac_port(u8 mac[ETH_ALEN], iface_info_t *iface)
    {
        static int i;

        //malloc new entry
        mac_port_entry_t *entry = (mac_port_entry_t *)malloc(sizeof(mac_port_entry_t));
        pthread_mutex_lock(&mac_port_map.lock);

        //free existed entry
        if(mac_port_map.hash_table[mac[0]])
            free(mac_port_map.hash_table[mac[0]]);

        //assign hash_table entry and its predecessor
        mac_port_map.hash_table[mac[0]] = entry;
        if(mac_port_map.hash_table[(i-1)%HASH_8BITS])
            mac_port_map.hash_table[(i-1)%HASH_8BITS]->next = entry;

        //assign entry details
        for(i = 0; i < ETH_ALEN; i++)
        {
            mac_port_map.hash_table[mac[0]]->mac[i] = mac[i];
            mac_port_map.hash_table[mac[0]]->iface = (iface_info_t *)malloc(sizeof(iface_info_t));
            memcpy(mac_port_map.hash_table[mac[0]]->iface, iface, sizeof(iface_info_t));
            mac_port_map.hash_table[mac[0]]->next = mac_port_map.hash_table[(mac[0] + 1) % HASH_8BITS];
            mac_port_map.hash_table[mac[0]]->visited = time(NULL);
        }
        pthread_mutex_unlock(&mac_port_map.lock);
    }
}
```

```

//remove aged entry
int sweep_aged_mac_port_entry()
{
    static int i;
    time_t now = time(NULL);
    pthread_mutex_lock(&mac_port_map.lock);

    //traversal the table
    for(i = 0; i < HASH_8BITS; i++)
    {
        //if not exist
        if(!mac_port_map.hash_table[i])
            continue;

        //if aged
        if((now - mac_port_map.hash_table[i]->visited) > 30)
        {
            free(mac_port_map.hash_table[i]);
            mac_port_map.hash_table[i] = NULL;
            if(mac_port_map.hash_table[(i-1)%HASH_8BITS])
                mac_port_map.hash_table[(i-1)%HASH_8BITS]->next = NULL;
        }
    }
    pthread_mutex_unlock(&mac_port_map.lock);
    return 0;
}

//deal with input packet, if it is not in the transform table, broadcast it, else send it.
void handle_packet(iface_info_t *iface, char *packet, int len)
{
    struct ether_header *eh = (struct ether_header *)packet;
    iface_info_t *tmp;
    insert_mac_port(eh->ether_shost, iface);
    tmp = lookup_port(eh->ether_dhost);
    if(!tmp)
        broadcast_packet(iface, packet, len);
    else
        iface_send_packet(tmp, packet, len);
}

//broadcast packet to all iface except the source one
void broadcast_packet(iface_info_t *iface, const char *packet, int len)
{
    iface_info_t *iface2 = NULL;
    list_for_each_entry(iface2, &instance->iface_list, list) {
        if (iface2->fd == iface->fd)
            continue;
        iface_send_packet(iface2, packet, len);
    }
}

```

至此所有函数实现完毕，开始进行虚拟机测试，运行 python 脚本，启动终端

```

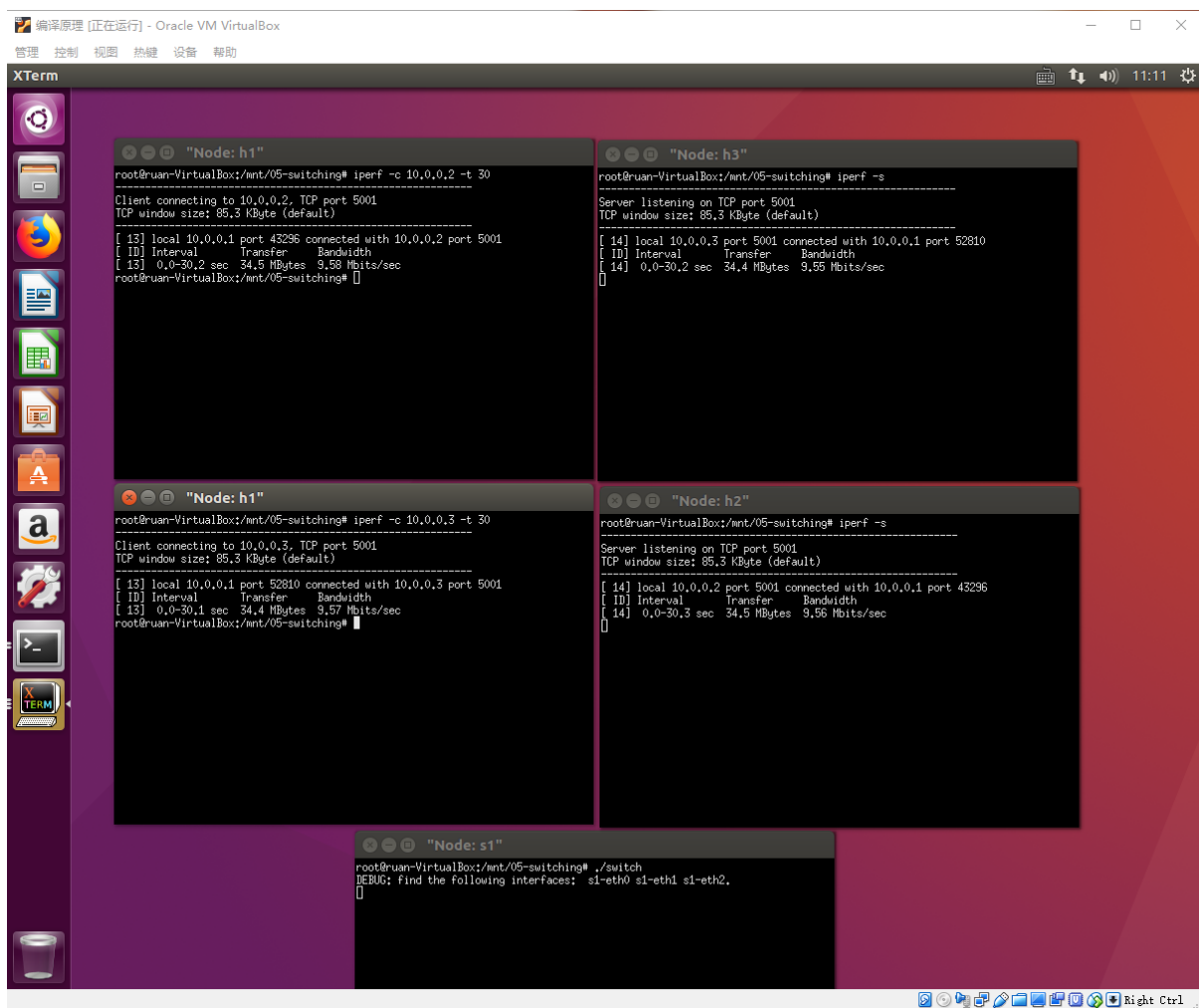
ruan@ruan-VirtualBox:/mnt/05-switching$ sudo python three_nodes_bw.py
mininet> xterm h1 h2 h3 s1
mininet> xterm h1

```

运行 switch 程序，h2,h3 作为 server，h1 作为 client，开始 iperf 测试，测试结果见下一部分。

四、 实验结果

运行完成后的截图见下页，可见 h1 同时访问 h2 与 h3 都达到了 9Mbps 的速率。



五、 实验总结

对比起上一个试验中实现的集线器广播最终两个线路加和不到 10Mbps 的速率，交换机明显更加高效地利用了链路，达到了接近链路速度限制的速率。可见交换机技术在今日的普及是有其丰厚的内涵的。

本次实验内容依然较为容易上手，很快便解决了实验内容，只是我书写的代码中设计的 next 是循环指向的，和 destroy 函数有些冲突，不过说实话我没有怎么看懂 destroy 函数的实现逻辑，是否有设计错误，好在程序中也并没有调用过它，不至于造成什么困扰。