

交换机转发实验报告

姓名：钟赞

学号：2016K8009915009

实验内容

- 实现一个具有转发表学习功能的交换机，使得：连接到同一交换机的主机可以相互通信。
- 实现对数据结构`mac_port_map`的所有操作(包括查询操作`lookup_port`、插入操作`insert_mac_port`和老化操作`sweep_aged_mac_port_entry`)，以及数据包的转发`handle_packet`和广播`broadcast_packet`操作。
- 使用`iperf`和给定的拓扑进行实验，对比交换机转发与集线器广播的性能。

实验步骤

补全`mac.c`中对`mac_port_map`的三种操作

- 查找 `lookup_port`

```
iface_info_t *lookup_port(u8 mac[ETH_ALEN])
{
    // TODO: implement the lookup process here
    fprintf(stdout, "implement the lookup process here.\n");
    mac_port_entry_t * entry;
    u8 hash_val = hash8(mac, ETH_ALEN);
    iface_info_t * result;

    pthread_mutex_lock(&mac_port_map.lock);
    list_for_each_entry(entry, &mac_port_map.hash_table[hash_val], list) {
        if (memcmp(entry->mac, mac, ETH_ALEN * sizeof(u8))) {
            // if found, get its iface info and the time
            entry->visited = time(NULL);
            result = entry->iface;
            break;
        }
    }
    pthread_mutex_unlock(&mac_port_map.lock);
    return result;
}
```

- 插入 `insert_mac_port`

```

void insert_mac_port(u8 mac[ETH_ALEN], iface_info_t *iface)
{
    // TODO: implement the insertion process here
    fprintf(stdout, "implement the insertion process here.\n");
    // initialize a new entry
    mac_port_entry_t * entry = (mac_port_entry_t *)malloc(sizeof(mac_port_entry_t));
    entry->iface = iface;
    entry->mac = mac;
    entry->visited = time(NULL);
    u8 hash_val = hash8(mac, ETH_ALEN);

    pthread_mutex_lock(&mac_port_map.lock);
    // add entry into list
    list_add_tail(&entry->list, &mac_port_map.hash_table[hash_val]);
    pthread_mutex_unlock(&mac_port_map.lock);
}

```

- 老化 **sweep_aged_mac_port_entry**

```

int sweep_aged_mac_port_entry()
{
    // TODO: implement the sweeping process here
    fprintf(stdout, "implement the sweeping process here.\n");
    mac_port_entry_t *entry, *q;
    time_t now = time(NULL);

    while (1) {
        pthread_mutex_lock(&mac_port_map.lock);
        for (int i = 0; i < HASH_8BITS; i++) {
            list_for_each_entry_safe(entry, q, &mac_port_map.hash_table[i], list) {
                if (now - entry->visited >= MAC_PORT_TIMEOUT) {
                    // delete the entry
                    list_delete_entry(&entry->list);
                    free(entry);
                }
            }
        }
        // cycling every 1 second
        sleep(TIME_SLICE);
        pthread_mutex_unlock(&mac_port_map.lock);
    }
    return 0;
}

```

实验结果

结果分析

参考资料

