

- 广播网络实验
 - 实验内容

广播网络实验

2024年11月12日

2024E8013282087-陈潇

实验内容

实现对数据结构`mac_port_map`的所有操作，以及数据包的转发和广播操作

- 设计思路 为了更方便的补全相关操作，我实现了以下两个函数：

```
// 比较两个Mac地址有多少位相同
int mac_euqal_len(u8 *m1, u8 *m2, int len)
{
    for(int i = 0; i < len; i++)
    {
        if(m1[i] != m2[i])
            return i+1;
    }
    return 0;    //没有一位相同
}

// 复制Mac地址
void mac_cpy (u8* mac_dst, u8* mac_src, int len) {
    for (int i = 0; i < len; i++) {
        mac_dst[i] = mac_src[i];
    }
}
```

`iface_info_t *lookup_port(u8 mac[ETH_ALEN]);`

函数作用：若在转发表中查找对应`mac`地址和`iface`映射的表项，则返回查询的`mac`和对应的`iface`。交换机在转发过程中会有另一个线程进行超时表项的清理工作，所以在查找的时候需要上锁来保持原子性。

代码如下：

```

iface_info_t *lookup_port(u8 mac[ETH_ALEN])
{
    // TODO: implement the lookup process here
    // fprintf(stdout, "TODO: implement the lookup process here.\n");
    pthread_mutex_lock(&mac_port_map.lock);
    u8 hash_value = hash8((char *)mac, ETH_ALEN);
    mac_port_entry_t * mac_port_entry_pos = NULL;
    list_for_each_entry(mac_port_entry_pos,
&mac_port_map.hash_table[hash_value], list) {
        if (mac_equal_len(mac_port_entry_pos->mac, mac, ETH_ALEN) == 0)
        {
            mac_port_entry_pos->visited = time(NULL);
            pthread_mutex_unlock(&mac_port_map.lock);
            return mac_port_entry_pos->iface;
        }
    }
    pthread_mutex_unlock(&mac_port_map.lock);
    return NULL;
}

```

```
void insert_mac_port(u8 mac[ETH_ALEN], iface_info_t *iface);
```

函数作用：在转发表没有源mac和对应iface的映射表项时，将源mac地址与对应的iface插入到转发表中。

代码如下：

```

void insert_mac_port(u8 mac[ETH_ALEN], iface_info_t *iface)
{
    // TODO: implement the insertion process here
    // fprintf(stdout, "TODO: implement the insertion process here.\n");
    pthread_mutex_lock(&mac_port_map.lock);
    mac_port_entry_t * new_mac_port_entry =
safe_malloc(sizeof(mac_port_entry_t));
    bzero(new_mac_port_entry, sizeof(mac_port_entry_t));
    mac_cpy(new_mac_port_entry->mac, mac, ETH_ALEN);
    new_mac_port_entry->iface = iface;
    new_mac_port_entry->visited = time(NULL);
    u8 hash_value = hash8((char *)mac, ETH_ALEN);
    list_add_tail(&new_mac_port_entry->list,
&mac_port_map.hash_table[hash_value]);
    pthread_mutex_unlock(&mac_port_map.lock);
}

```

```
int sweep_aged_mac_port_entry();
```

函数作用：清理转发表中超过30s没有被查询的冗余表项。因为多线程，所以清理操作也需要上锁。

代码如下:

```
int sweep_aged_mac_port_entry()
{
    // TODO: implement the sweeping process here
    // fprintf(stdout, "TODO: implement the sweeping process here.\n");
    pthread_mutex_lock(&mac_port_map.lock);
    mac_port_entry_t *entry = NULL;
    mac_port_entry_t *q = NULL;
    time_t now = time(NULL);
    int rm_entry_num = 0;
    for (int i = 0; i < HASH_8BITS; i++) {
        list_for_each_entry_safe(entry, q, &mac_port_map.hash_table[i],
list) {
            if ((int)(now - entry->visited) > MAC_PORT_TIMEOUT) {
                list_delete_entry(&entry->list);
                free(entry);
                rm_entry_num ++;
            }
        }
    }
    pthread_mutex_unlock(&mac_port_map.lock);
    return rm_entry_num;
}
```

```
void broadcast_packet(iface_info_t *iface, const char
*packet, int len);
```

函数作用: 广播函数, 复用lab1的即可。

代码如下:

```
void broadcast_packet(iface_info_t *iface, const char *packet, int len)
{
    // TODO: broadcast packet
    iface_info_t *ifa = NULL;
    list_for_each_entry(ifa, &instance->iface_list, list){
        if(ifa->index!=iface->index)
        {
            iface_send_packet(ifa,packet,len);
        }
    }
}
```

```
void handle_packet(iface_info_t *iface, char *packet, int
len);
```

函数作用：调用`lookup_port`函数，检查`mac`与端口映射在不在映射表中，在映射表中则根据映射表发包，没有则广播，并且调用`insert_mac_port`函数将映射插入表用。

代码如下：

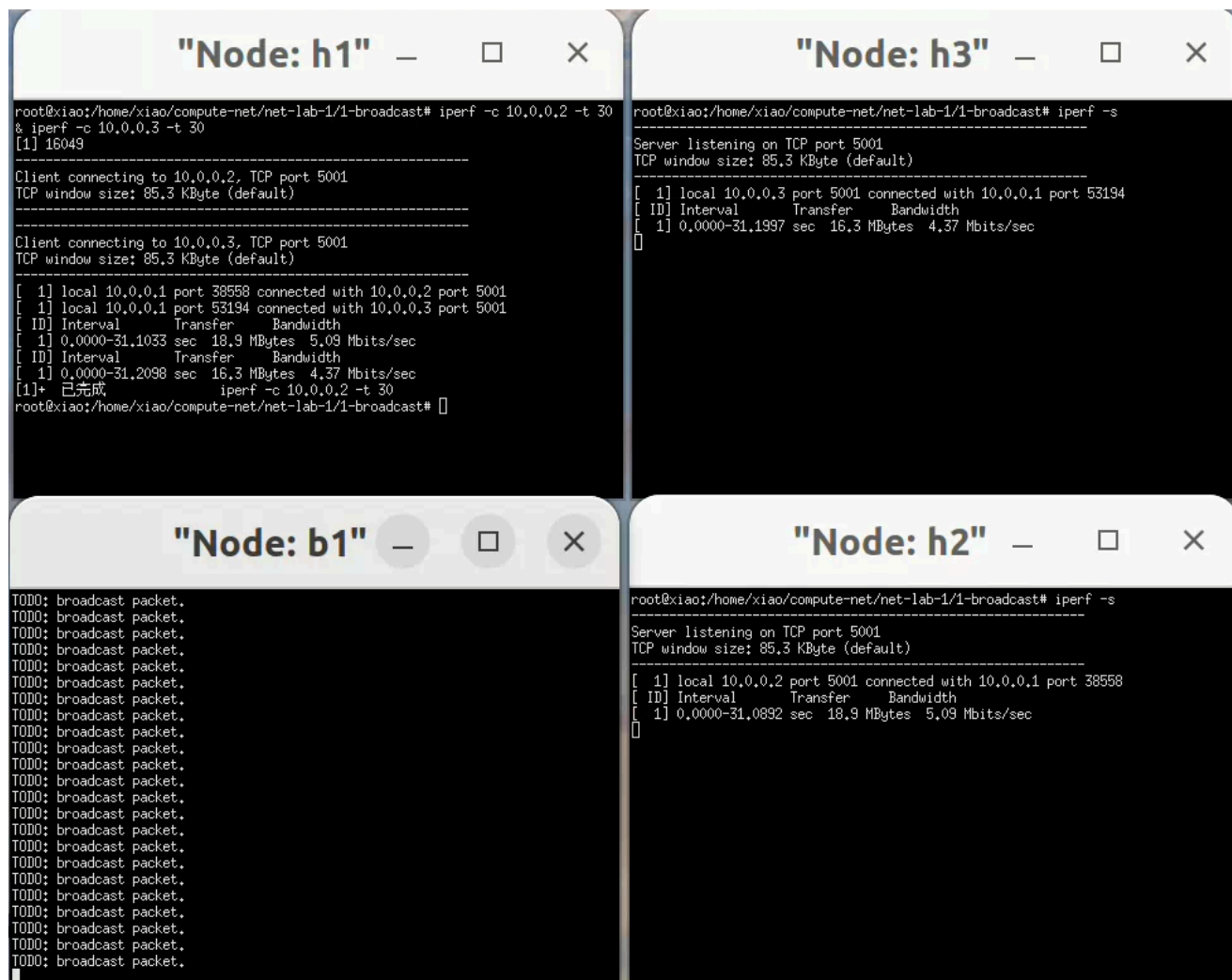
```
void handle_packet(iface_info_t *iface, char *packet, int len)
{
    // TODO: implement the packet forwarding process here
    // fprintf(stdout, "TODO: implement the packet forwarding process
    here.\n");

    struct ether_header *eh = (struct ether_header *)packet;
    // log(DEBUG, "the dst mac address is " ETHER_STRING ".\n",
    ETHER_FMT(eh->ether_dhost));
    iface_info_t * iface_entry = NULL;
    if ((iface_entry = lookup_port(eh->ether_dhost)) != NULL) {
        iface_send_packet(iface_entry, packet, len);
    } else {
        broadcast_packet(iface, packet, len);
    }

    if (lookup_port(eh->ether_shost) == NULL) {
        insert_mac_port(eh->ether_shost, iface);
    }
    free(packet);
}
```

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

- 使用`hub:H1: iperf client; H2, H3: servers`（`h1`同时向 `h2`和 `h3`测量）



上图中 h1向 h2、h3节点测量，可以看到 h1向 h2和 h3发送带宽分别为 5.09Mbps和 4.37Mbps。h2和 h3的接收带宽分别为 5.09Mbps和 4.37Mbps。在拓扑文件中，h1->b1的带宽为 20Mbps，b1->h2、b1->h3的带宽为 10Mbps。因此带宽利用率分别为：47.3%。

- 使用Switch:H1: `iperf client`; H2, H3: `servers` (h1同时向 h2和 h3测量)

The image displays four terminal windows arranged in a 2x2 grid, each showing the output of network tests performed on a switch node (s1) connected to three host nodes (h2, h1, h3).

- Node: s1**: Shows the switch configuration and the execution of the `iperf` command to initiate tests.
- Node: h2**: Shows the server listening on TCP port 5001 and the results of a test from h1 to h2, achieving a bandwidth of 9.34 Mbits/sec.
- Node: h1**: Shows the client connecting to h2 and h3, and the results of tests from h1 to h2 (9.47 Mbits/sec) and h1 to h3 (9.56 Mbits/sec).
- Node: h3**: Shows the server listening on TCP port 5001 and the results of a test from h1 to h3, achieving a bandwidth of 9.56 Mbits/sec.

```
"Node: s1"
root@xiao:/home/xiao/compute-net/net-lab-1/2-switching# ./switch
DEBUG: find the following interfaces: s1-eth0 s1-eth1 s1-eth2.
DEBUG: 3 aged entries in mac_port table are removed.
^C
root@xiao:/home/xiao/compute-net/net-lab-1/2-switching# ./switch-reference
.32
DEBUG: find the following interfaces: s1-eth0 s1-eth1 s1-eth2.
DEBUG: 3 aged entries in mac_port table are removed.
[]

"Node: h2"
root@xiao:/home/xiao/compute-net/net-lab-1/2-switching# iperf -s
Server listening on TCP port 5001
TCP window size: 85.3 KByte (default)
[ 1] local 10.0.0.2 port 5001 connected with 10.0.0.1 port 51614
[ ID] Interval      Transfer    Bandwidth
[ 1] 0.0000-31.3109 sec 34.9 MBytes 9.34 Mbits/sec
[ 2] local 10.0.0.2 port 5001 connected with 10.0.0.1 port 37274
[ ID] Interval      Transfer    Bandwidth
[ 2] 0.0000-30.4327 sec 34.4 MBytes 9.48 Mbits/sec
[]

"Node: h1"
root@xiao:/home/xiao/compute-net/net-lab-1/2-switching# iperf -c 10.0.0.2 -t 30 & iperf -c 10.0.0.3 -t 30
[1] 21905

Client connecting to 10.0.0.3, TCP port 5001
TCP window size: 85.3 KByte (default)

Client connecting to 10.0.0.2, TCP port 5001
TCP window size: 85.3 KByte (default)

[ 1] local 10.0.0.1 port 37274 connected with 10.0.0.2 port 5001
[ 1] local 10.0.0.1 port 46642 connected with 10.0.0.3 port 5001
[ ID] Interval      Transfer    Bandwidth
[ 1] 0.0000-30.4461 sec 34.4 MBytes 9.47 Mbits/sec
[ ID] Interval      Transfer    Bandwidth
[ 1] 0.0000-30.7111 sec 35.0 MBytes 9.56 Mbits/sec
[1]+ 已完成 iperf -c 10.0.0.2 -t 30
root@xiao:/home/xiao/compute-net/net-lab-1/2-switching#

"Node: h3"
root@xiao:/home/xiao/compute-net/net-lab-1/2-switching# iperf -s
Server listening on TCP port 5001
TCP window size: 85.3 KByte (default)
[ 1] local 10.0.0.3 port 5001 connected with 10.0.0.1 port 52002
[ ID] Interval      Transfer    Bandwidth
[ 1] 0.0000-31.2694 sec 35.6 MBytes 9.56 Mbits/sec
[ 2] local 10.0.0.3 port 5001 connected with 10.0.0.1 port 46642
[ ID] Interval      Transfer    Bandwidth
[ 2] 0.0000-30.6970 sec 35.0 MBytes 9.56 Mbits/sec
[]
```

上图中 h1向 h2、h3节点测量，可以看到 h1向 h2和 h3发送带宽分别为 9.47Mbps和 9.56Mbps。h2和 h3的接收带宽分别为 9.48Mbps和 9.56Mbps。在拓扑文件中，h1->b1的带宽为 20Mbps，b1->h2、b1->h3的带宽为 10Mbps。因此带宽利用率分别为：95.18%。

可以看到，使用交换机转发的带宽是使用集线器广播的201.2%