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_euqal_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：

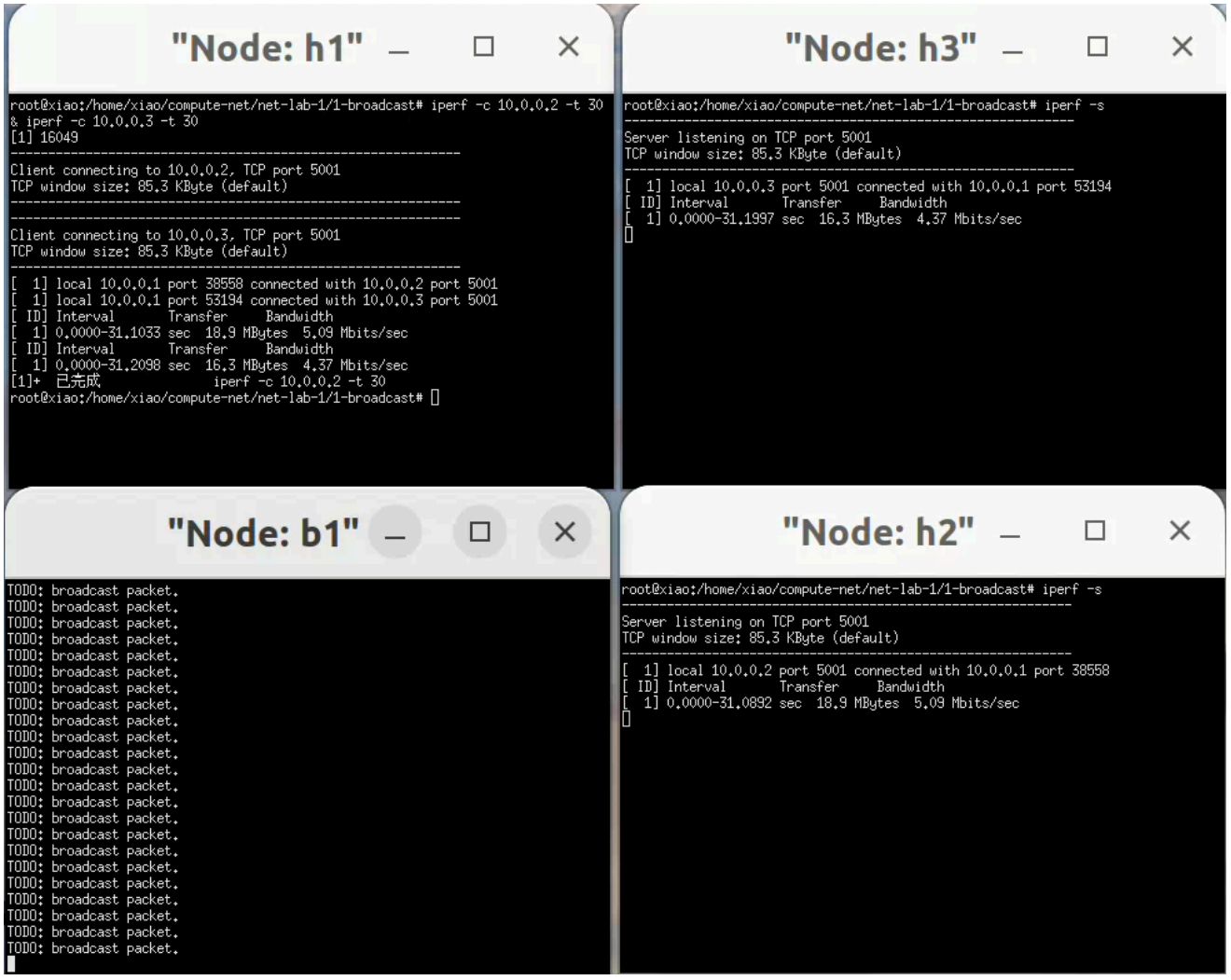
```
void handle_packet(iface_info_t *iface, char *packet, int len)
{
  // TODO: implement the packet forwarding process here
  // fprintf(stdout, "TODO: implement the packet forwarding process
here.\n");

  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）

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%