实验报告

孔静 2014K8009929022

October 25, 2017

1 实验题目

交换机学习实验

2 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
- 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

3 实验流程

- broadcast_packet

```
1 //收到每个数据包，将该包从所有其它端口转发出去
2 void broadcast_packet(iface_info_t *iface, const char *packet, int
      len){
3   iface_info_t *ifaces = NULL;
4   list_for_each_entry(ifaces, &instance->iface_list, list) {
5     if(iface->index != ifaces->index){
6       iface_send_packet(ifaces, packet, len);
7     }
8   }
9 }
```

- lookup_port

```
1 //哈希查找mac
2 iface_info_t *lookup_port(u8 mac[ETH_ALEN]){
3   pthread_mutex_lock(&mac_port_map.lock);
4
5   iface_info_t *iface = NULL;
6   mac_port_entry_t *entry;
7
```

```
  for(entry = mac_port_map.hash_table[hash8(mac, ETH_ALEN)]; entry
      != NULL; entry = entry->next){
    if (memcmp(mac, entry->mac, ETH_ALEN) == 0){
      iface = entry->iface;
      entry->visited = 0;
      break;
    }
  }

  pthread_mutex_unlock(&mac_port_map.lock);
  return iface;
}
```

- insert_mac_port

```
//哈希查找mac,若查找不到则插入
void insert_mac_port(u8 mac[ETH_ALEN], iface_info_t *iface){
  pthread_mutex_lock(&mac_port_map.lock);

  mac_port_entry_t *entry;
  entry = mac_port_map.hash_table[hash8(mac, ETH_ALEN)];

  if(entry == NULL){
    entry = (mac_port_entry_t *)malloc(sizeof(mac_port_entry_t));
    memcpy(entry->mac, mac, ETH_ALEN);
    entry->iface = iface;
    entry->visited = 0;
    entry->next = NULL;
    mac_port_map.hash_table[hash8(mac, ETH_ALEN)] = entry;
  }
  else{
    for( ; entry != NULL; entry = entry->next){
      if(memcmp(mac, entry->mac, ETH_ALEN) == 0){
        break;
      }

      if(entry->next == NULL){
        entry->next = (mac_port_entry_t *)malloc(sizeof(
    mac_port_entry_t));
        entry = entry->next;
        memcpy(entry->mac, mac, ETH_ALEN);
        entry->iface = iface;
        entry->visited = 0;
        entry->next = NULL;
        break;
      }
    }
  }
  pthread_mutex_unlock(&mac_port_map.lock);
}
```

- sweep_aged_mac_port_entry

```
//遍历，增加visitied次数，并判断是否超过30次，超过则free删除
int sweep_aged_mac_port_entry(){
  pthread_mutex_lock(&mac_port_map.lock);

```

```
  mac_port_entry_t *entry, *next_entry;
  int number = 0;

  for(int i = 0; i < HASH_8BITS; i++){
    entry = mac_port_map.hash_table[i];

    while(entry != NULL && entry->visited >= 30){
      next_entry = entry->next;
      free(entry);
      entry = next_entry;
    }

    mac_port_map.hash_table[i] = entry;

    if(entry != NULL){
      entry->visited++;
      next_entry = entry->next;

      while(next_entry != NULL){
        if(next_entry->visited >= 30){
          entry->next = next_entry->next;
          free(next_entry);
          next_entry = entry->next;
        }
        else{
          entry = next_entry;
          entry->visited++;
          next_entry = entry->next;
        }
      }
    }
  }
  pthread_mutex_unlock(&mac_port_map.lock);
  return number;
}
```

- 进行 iperf 测试

```
iperf -s #开启server
iperf -c 10.0.0.X -t 30 #在client连接server
```

4 实验结果

Figure 1: H1:iperf server; H2,H3:iperf clients

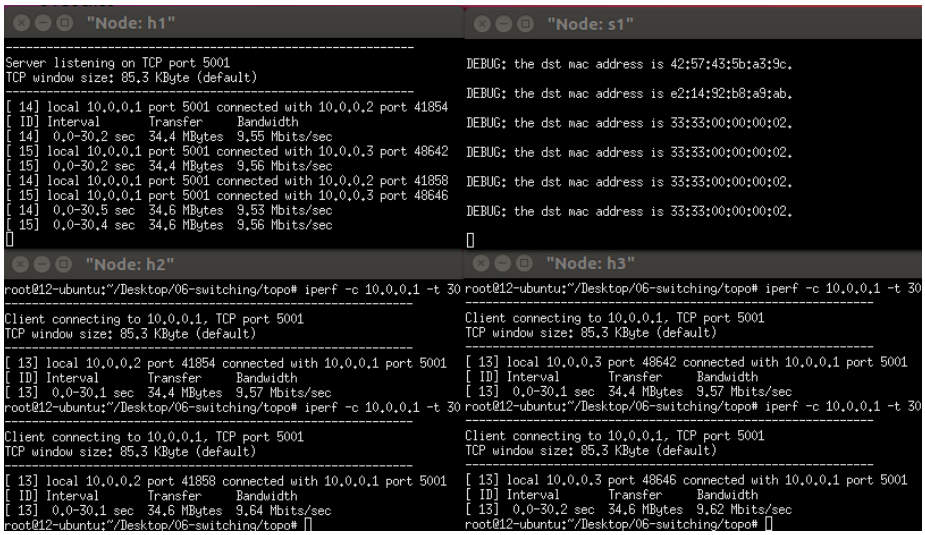
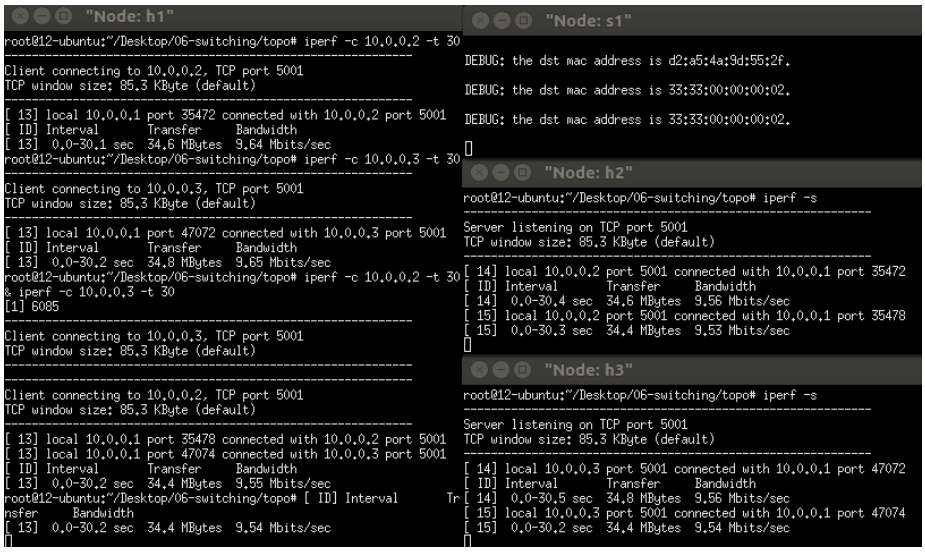


Figure 2: H1:iperf clients; H2,H3:iperf server



5 结果分析

交换机将收到的数据包转发给目的地址，在查找的到地址的情况下，并不会像广播网络那样发送无效数据包从而影响到交换机到其他主机的速度，只需考

虑各主机的带宽即可。

5.1 H1:iperf server; H2,H3:iperf clients

由 Fig1 可看出，H2/H3 单独或同时 iperf H1，带宽均约为 10(Mb/s)。

考虑到 hub 到 H1/H2/H3 的带宽分别是 20/10/10(Mb/s)，所以 H2/H3 同时 iperf H1，不会降速，速度均约等于 10(Mb/s)。

5.2 H1:iperf clients; H2,H3:iperf server

由 Fig2 可看出，H1 单独或同时 iperf H2/H3，带宽均约为 10(Mb/s)。

考虑到 hub 到 H1/H2/H3 的带宽分别是 20/10/10(Mb/s)，所以 H2/H3 同时 iperf H1，不会降速，速度均约等于 10(Mb/s)。

5.3 对比: 集线器广播

两次实验对比结果上，H1 作为服务器端，H2/H3 客户端时，结果差不多，集线器广播虽然 H2/H3 同时 iperf H1 时候，会影响 hub 到 H2/H3 的下行通路，但因为 hub 到 H1 的带宽是到 H2/H3 的两倍，所以对结果没有影响。

H1 作为客户端，H2/H3 作为服务器端的时候，交换机的结果是均为 10(Mb/s)，集线器广播的结果是两者相加为 10(Mb/s)。这是因为广播网络的多余数据包影响到了速度，而交换机只有在查找不到目的 mac 地址的时候才会广播，正常情况下并不会影响其他主机到交换机的速度，所以均为 10(Mb/s)。

综上分析，交换机的性能更优一些。