

实验报告

孔静 2014K8009929022

October 14, 2017

1 实验题目

广播网络实验

2 实验内容

- 实现 main.c 中的 TODO 部分
- 三个节点相互能够 ping 通
- 进行 iperf 测试

3 实验流程

- 实现 main.c 中的 TODO 部分

```
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 {
4     // TODO: implement the broadcast process here
5     iface_info_t *ifaces = NULL;
6     list_for_each_entry(ifaces, &instance->iface_list, list) {
7         if(ifaces->index != iface->index){
8             iface_send_packet(ifaces, packet, len);
9         }
10    }
11    fprintf(stdout, "TODO: implement the broadcast process here.");
12 }
```

- 三个节点相互能够 ping 通

```
1 ping 10.0.0.1 -c 4 #在h2,h3执行
2 ping 10.0.0.2 -c 4 #在h1,h3执行
3 ping 10.0.0.3 -c 4 #在h1,h2执行
```

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

Figure 1: 三个节点相互 ping 通

```

root@12-ubuntu:~/Desktop/05-broadcast/topo# ./hub
bash: ./hub: No such file or directory
root@12-ubuntu:~/Desktop/05-broadcast/topo# ./../hub
DEBUG: find the following interfaces: b1-eth0 b1-eth1 b1-eth2
TODO: implement the broadcast process here.
TODO: implement the broadcast process here.
TODO: implement the broadcast process here.
TODO: implement the broadcast process here.
TODO: implement the broadcast process here.
TODO: implement the broadcast process here.
TODO: implement the broadcast process here.
TODO: implement the broadcast process here.
TODO: implement the broadcast process here.
TODO: implement the broadcast process here.
TODO: implement the broadcast process here.
TODO: implement the broadcast process here.
TODO: implement the broadcast process here.
TODO: implement the broadcast process here.
TODO: implement the broadcast process here.
root@12-ubuntu:~/Desktop/05-broadcast/topo# ping 10.0.0.2 -c 4
PING 10.0.0.2 (10.0.0.2) 56(84) bytes of data:
 64 bytes from 10.0.0.2: icmp_seq=1 ttl=64 time=0.134 ms
 64 bytes from 10.0.0.2: icmp_seq=2 ttl=64 time=0.137 ms
 64 bytes from 10.0.0.2: icmp_seq=3 ttl=64 time=0.549 ms
 64 bytes from 10.0.0.2: icmp_seq=4 ttl=64 time=0.533 ms

--- 10.0.0.2 ping statistics ---
 4 packets transmitted, 4 received, 0% packet loss, time 3079ms
rtt min/avg/max/mdev = 0.134/0.338/0.549/0.203 ms
root@12-ubuntu:~/Desktop/05-broadcast/topo# ping 10.0.0.3 -c 4
PING 10.0.0.3 (10.0.0.3) 56(84) bytes of data:
 64 bytes from 10.0.0.3: icmp_seq=1 ttl=64 time=0.157 ms
 64 bytes from 10.0.0.3: icmp_seq=2 ttl=64 time=0.157 ms
 64 bytes from 10.0.0.3: icmp_seq=3 ttl=64 time=0.141 ms
 64 bytes from 10.0.0.3: icmp_seq=4 ttl=64 time=1.19 ms

--- 10.0.0.3 ping statistics ---
 4 packets transmitted, 4 received, 0% packet loss, time 3076ms
rtt min/avg/max/mdev = 0.141/0.413/1.199/0.454 ms
root@12-ubuntu:~/Desktop/05-broadcast/topo#

root@12-ubuntu:~/Desktop/05-broadcast/topo# ping 10.0,0,1 -c 4
PING 10.0,0,1 (10.0,0,1) 56(84) bytes of data:
 64 bytes from 10.0,0,1: icmp_seq=1 ttl=64 time=0.301 ms
 64 bytes from 10.0,0,1: icmp_seq=2 ttl=64 time=0.586 ms
 64 bytes from 10.0,0,1: icmp_seq=3 ttl=64 time=0.150 ms
 64 bytes from 10.0,0,1: icmp_seq=4 ttl=64 time=0.311 ms

--- 10.0,0,1 ping statistics ---
 4 packets transmitted, 4 received, 0% packet loss, time 3019ms
rtt min/avg/max/mdev = 0.150/0.337/0.586/0.157 ms
root@12-ubuntu:~/Desktop/05-broadcast/topo# ping 10.0,0,3 -c 4
PING 10.0,0,3 (10.0,0,3) 56(84) bytes of data:
 64 bytes from 10.0,0,3: icmp_seq=1 ttl=64 time=0.129 ms
 64 bytes from 10.0,0,3: icmp_seq=2 ttl=64 time=0.162 ms
 64 bytes from 10.0,0,3: icmp_seq=3 ttl=64 time=0.478 ms
 64 bytes from 10.0,0,3: icmp_seq=4 ttl=64 time=0.223 ms

--- 10.0,0,3 ping statistics ---
 4 packets transmitted, 4 received, 0% packet loss, time 3048ms
rtt min/avg/max/mdev = 0.129/0.248/0.478/0.137 ms
root@12-ubuntu:~/Desktop/05-broadcast/topo#

root@12-ubuntu:~/Desktop/05-broadcast/topo# ping 10.0.0,1 -c 4
PING 10.0.0,1 (10.0.0,1) 56(84) bytes of data:
 64 bytes from 10.0.0,1: icmp_seq=1 ttl=64 time=0.259 ms
 64 bytes from 10.0.0,1: icmp_seq=2 ttl=64 time=0.103 ms
 64 bytes from 10.0.0,1: icmp_seq=3 ttl=64 time=0.054 ms
 64 bytes from 10.0.0,1: icmp_seq=4 ttl=64 time=0.221 ms

--- 10.0.0,1 ping statistics ---
 4 packets transmitted, 4 received, 0% packet loss, time 3056ms
rtt min/avg/max/mdev = 0.054/0.193/0.259/0.084 ms
root@12-ubuntu:~/Desktop/05-broadcast/topo# ping 10.0,0,2 -c 4
PING 10.0,0,2 (10.0,0,2) 56(84) bytes of data:
 64 bytes from 10.0,0,2: icmp_seq=1 ttl=64 time=0.256 ms
 64 bytes from 10.0,0,2: icmp_seq=2 ttl=64 time=0.275 ms
 64 bytes from 10.0,0,2: icmp_seq=3 ttl=64 time=0.235 ms
 64 bytes from 10.0,0,2: icmp_seq=4 ttl=64 time=0.166 ms

--- 10.0,0,2 ping statistics ---
 4 packets transmitted, 4 received, 0% packet loss, time 3075ms
rtt min/avg/max/mdev = 0.166/0.233/0.275/0.041 ms
root@12-ubuntu:~/Desktop/05-broadcast/topo#

```

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

```

Node: h1
root@t2-ubuntu:/Desktop/05-broadcast/topo# iperf -s

Server listening on TCP port 5001
TCP window size: 65.3 KByte (default)

[14] local 10.0.0.1 port 5001 connected with 10.0.0.2 port 38974
[ID] Interval Transfer Bandwidth
[14] 0.0-30.3 sec 33.9 MBytes 9.37 Mbits/sec
[15] local 10.0.0.1 port 5001 connected with 10.0.0.3 port 42756
[15] 0.0-30.4 sec 34.0 MBytes 9.37 Mbits/sec
[14] local 10.0.0.1 port 5001 connected with 10.0.0.2 port 38978
[15] local 10.0.0.1 port 5001 connected with 10.0.0.3 port 42760
[15] 0.0-30.8 sec 32.9 MBytes 8.95 Mbits/sec
[14] 0.0-31.2 sec 33.6 MBytes 9.05 Mbits/sec

Node: h2
root@t2-ubuntu:/Desktop/05-broadcast/topo# iperf -c 10.0.0.1 -t 30

Client connecting to 10.0.0.1, TCP port 5001
TCP window size: 65.3 KByte (default)

[13] local 10.0.0.2 port 38974 connected with 10.0.0.1 port 5001
[ID] Interval Transfer Bandwidth
[13] 0.0-30.2 sec 33.9 MBytes 9.42 Mbits/sec
root@t2-ubuntu:/Desktop/05-broadcast/topo# iperf -c 10.0.0.1 -t 30

Client connecting to 10.0.0.1, TCP port 5001
TCP window size: 65.3 KByte (default)

[13] local 10.0.0.2 port 38978 connected with 10.0.0.1 port 5001
[ID] Interval Transfer Bandwidth
[13] 0.0-30.3 sec 33.6 MBytes 9.31 Mbits/sec
root@t2-ubuntu:/Desktop/05-broadcast/topo#

Node: b1
TODO: implement the broadcast process here.
TODO: implement the broadcast process here.
TODO: implement the broadcast process here.
TODO: implement the broadcast process here.
TODO: implement the broadcast process here.
TODO: implement the broadcast process here.
TODO: implement the broadcast process here.
TODO: implement the broadcast process here.
TODO: implement the broadcast process here.
TODO: implement the broadcast process here.
TODO: implement the broadcast process here.
TODO: implement the broadcast process here.

Node: h3
root@t2-ubuntu:/Desktop/05-broadcast/topo# iperf -c 10.0.0.1 -t 30

Client connecting to 10.0.0.1, TCP port 5001
TCP window size: 65.3 KByte (default)

[13] local 10.0.0.3 port 42756 connected with 10.0.0.1 port 5001
[ID] Interval Transfer Bandwidth
[13] 0.0-30.3 sec 34.0 MBytes 9.43 Mbits/sec
root@t2-ubuntu:/Desktop/05-broadcast/topo# iperf -c 10.0.0.1 -t 30

Client connecting to 10.0.0.1, TCP port 5001
TCP window size: 65.3 KByte (default)

[13] local 10.0.0.3 port 42760 connected with 10.0.0.1 port 5001
[ID] Interval Transfer Bandwidth
[13] 0.0-30.1 sec 32.9 MBytes 9.18 Mbits/sec
root@t2-ubuntu:/Desktop/05-broadcast/topo#

```

Figure 3: H1:iperf clients; H2,H3:iperf server

```

Node: h1"
root@i2-ubuntu:/Desktop/05-broadcast/topo# iperf -c 10.0.0.2 -t 30
TODD: implement the broadcast process here.
TODD: implement the broadcast process here.
TODD: implement the broadcast process here.
TODD: implement the broadcast process here.
TODD: implement the broadcast process here.
TODD: implement the broadcast process here.
TODD: implement the broadcast process here.
TODD: implement the broadcast process here.
TODD: implement the broadcast process here.
TODD: implement the broadcast process here.
Client connecting to 10.0.0.2, TCP port 5001
TCP window size: 85.3 KByte (default)

[13] local 10.0.0.1 port 48312 connected with 10.0.0.2 port 5001
[ID] Interval Transfer Bandwidth
[13] 0.0-30.2 sec 33.9 MBytes 9.40 Mbits/sec
root@i2-ubuntu:/Desktop/05-broadcast/topo# iperf -c 10.0.0.3 -t 30
TODD: implement the broadcast process here.
TODD: implement the broadcast process here.
TODD: implement the broadcast process here.
TODD: implement the broadcast process here.
TODD: implement the broadcast process here.
TODD: implement the broadcast process here.
TODD: implement the broadcast process here.
TODD: implement the broadcast process here.
TODD: implement the broadcast process here.
TODD: implement the broadcast process here.
Client connecting to 10.0.0.3, TCP port 5001
TCP window size: 85.3 KByte (default)

[13] local 10.0.0.1 port 41326 connected with 10.0.0.3 port 5001
[ID] Interval Transfer Bandwidth
[13] 0.0-30.1 sec 33.9 MBytes 9.43 Mbits/sec
root@i2-ubuntu:/Desktop/05-broadcast/topo# iperf -c 10.0.0.3 -t 30
& iperf -c 10.0.0.2 -t 30
[1] 5700

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

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

[13] local 10.0.0.1 port 41330 connected with 10.0.0.3 port 5001
[13] local 10.0.0.1 port 48316 connected with 10.0.0.2 port 5001
[ID] Interval Transfer Bandwidth
[13] 0.0-30.2 sec 24.9 MBytes 6.91 Mbits/sec
[ID] Interval Transfer Bandwidth
[13] 0.0-30.6 sec 9.50 MBytes 2.61 Mbits/sec
[1] Done
root@i2-ubuntu:/Desktop/05-broadcast/topo#

Node: b1"
root@i2-ubuntu:/Desktop/05-broadcast/topo# iperf -s
Server listening on TCP port 5001
TCP window size: 85.3 KByte (default)

[14] local 10.0.0.2 port 5001 connected with 10.0.0.1 port 48312
[ID] Interval Transfer Bandwidth
[14] 0.0-30.3 sec 33.9 MBytes 9.37 Mbits/sec
[15] local 10.0.0.2 port 5001 connected with 10.0.0.1 port 48316
[15] 0.0-30.7 sec 9.50 MBytes 2.60 Mbits/sec

Node: h2"
root@i2-ubuntu:/Desktop/05-broadcast/topo# iperf -s
Server listening on TCP port 5001
TCP window size: 85.3 KByte (default)

[14] local 10.0.0.3 port 5001 connected with 10.0.0.1 port 41326
[ID] Interval Transfer Bandwidth
[14] 0.0-30.3 sec 33.9 MBytes 9.37 Mbits/sec
[15] local 10.0.0.3 port 5001 connected with 10.0.0.1 port 41330
[15] 0.0-30.4 sec 24.9 MBytes 6.87 Mbits/sec

```

## 5 结果分析

广播网络将收到的数据包全数发送给其它端口，所以在不同情况下效率会有

影响。

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

由 Fig2 可看出，H2/H3 不管是分开单独，还是一起同时 iperf H1，带宽都很接近，约为 10(Mb/s)。

这是因为 H1 作为服务器端，H2/H3 作为客户端时，H2 发给 hub 再传给 H1 的数据包，hub 也会传给 H3，会影响下行至 H3 的网络，同理 H3 发送的数据包也会影响下行至 H2 的网络，但两者至 hub 的上行通路并未被影响。因此考虑到 hub 到 H1/H2/H3 的带宽分别是 20/10/10(Mb/s)，所以即使 H2/H3 同时 iperf H1，仍然不会降速，速度均约等于 10(Mb/s)。

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

由 Fig3 可看出，H1 分开单独 iperf H2/H3，带宽约为 10(Mb/s)，但同时 iperf H2/H3，带宽均减小许多，相加约为 10(Mb/s)。

这个现象是因为，当 H1 作为客户端，H2/H3 作为服务器端时，H1 发送给 hub 再传送给 H2 的数据包，hub 也会发送给 H3，但这条链路上还要传递 H1 发送给 H3 的数据包，再加上 hub 至 H2/H3 的带宽均为 10(Mb/s)，所以出现了都有所降速，但有带宽之和约等于 10(Mb/s) 的现象。