

# 广播网络实验报告

张磊 2017K8009922027

## 一、实验题目

广播网络实验

## 二、实验内容

1. 实现节点广播网络的 `broadcast_packet` 函数;
2. 验证广播网络能够正常运行;  
    从一个端点 ping 另一个节点;
3. 验证广播网络效率;  
    在 `three_nodes_bw.py` 进行 `iperf` 测量;  
    两种场景:  
        H1:iperf client; H2,H3:servers;(h1 同时向 h2 和 h3 测量)  
        H1:iperf server; H2,H3:clients;(h2 和 h3 同时向 h1 测量)
4. 自己动手构建环形拓扑, 验证该拓扑下节点广播网络产生数据包环路;

## 三、实验流程

1. 完成 `main.c` 中的 `broadcast_packet` 函数的编写;
2. 启动 h1, h2, h3, b1;
3. 验证 h1, h2, h3 三个节点可以互相 ping 通;
4. 使用 `iperf` 测试验证广播网络链路的利用效率, 分为 h1 做客户端, h2, h3 做服务端以及 h1 做服务端, h2, h3 做客户端两种情况;
5. 修改 `three_nodes_bw.py` 文件构造环路;
6. 使用 h1 向 h2 发送一个数据包, 在 h2 节点使用 `wireshark` 工具抓包, 观察数据包在环路中不断转发;
7. 调研分析广播网络链路效率低于设置带宽的原因;

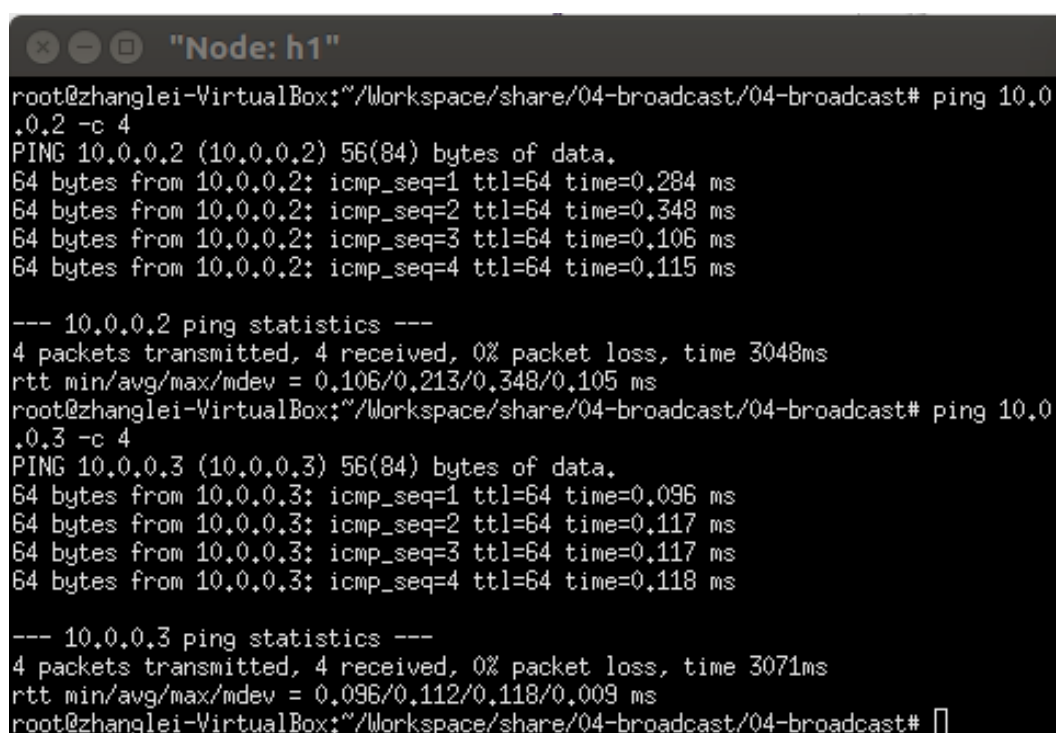
## 四、实验结果

### 1. 实现节点广播:

```
void broadcast_packet(iface_info_t *iface, const char *packet, int len)
{
    // TODO: broadcast packet
    // fprintf(stdout, "TODO: broadcast packet here.\n");
    // printf("\n\nThe %d times call broadcast\n", count_broadcast++);

    struct list_head list = instance->iface_list;
    struct list_head *list_ptr = &instance->iface_list;
    iface_info_t *entry;
    // printf("%d\n", iface->index);
    list_for_each_entry(entry, list_ptr, list) {
        // printf("index = %d\n", entry->index);
        if(entry->index != iface->index)
            iface_send_packet(entry, packet, len);
    }
}
```

Broadcast\_packet 函数



The terminal window shows the execution of ping commands from a node named 'h1'. The first command is 'ping 10.0.0.2 -c 4', which results in four successful pings with times ranging from 0.106 ms to 0.348 ms. The second command is 'ping 10.0.0.3 -c 4', which also results in four successful pings with times ranging from 0.096 ms to 0.118 ms. Both tests show 0% packet loss.

```
"Node: h1"
root@zhanglei-VirtualBox:~/Workspace/share/04-broadcast/04-broadcast# 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=0.284 ms
64 bytes from 10.0.0.2: icmp_seq=2 ttl=64 time=0.348 ms
64 bytes from 10.0.0.2: icmp_seq=3 ttl=64 time=0.106 ms
64 bytes from 10.0.0.2: icmp_seq=4 ttl=64 time=0.115 ms

--- 10.0.0.2 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3048ms
rtt min/avg/max/mdev = 0.106/0.213/0.348/0.105 ms
root@zhanglei-VirtualBox:~/Workspace/share/04-broadcast/04-broadcast# 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=0.096 ms
64 bytes from 10.0.0.3: icmp_seq=2 ttl=64 time=0.117 ms
64 bytes from 10.0.0.3: icmp_seq=3 ttl=64 time=0.117 ms
64 bytes from 10.0.0.3: icmp_seq=4 ttl=64 time=0.118 ms

--- 10.0.0.3 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3071ms
rtt min/avg/max/mdev = 0.096/0.112/0.118/0.009 ms
root@zhanglei-VirtualBox:~/Workspace/share/04-broadcast/04-broadcast#
```

h1 ping h2, h3

```
"Node: h2"
root@zhanglei-VirtualBox:~/Workspace/share/04-broadcast/04-broadcast# 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=0.061 ms
64 bytes from 10.0.0.1: icmp_seq=2 ttl=64 time=0.117 ms
64 bytes from 10.0.0.1: icmp_seq=3 ttl=64 time=0.110 ms
64 bytes from 10.0.0.1: icmp_seq=4 ttl=64 time=0.125 ms

--- 10.0.0.1 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3069ms
rtt min/avg/max/mdev = 0.061/0.103/0.125/0.025 ms
root@zhanglei-VirtualBox:~/Workspace/share/04-broadcast/04-broadcast# 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=0.296 ms
64 bytes from 10.0.0.3: icmp_seq=2 ttl=64 time=0.125 ms
64 bytes from 10.0.0.3: icmp_seq=3 ttl=64 time=0.127 ms
64 bytes from 10.0.0.3: icmp_seq=4 ttl=64 time=0.139 ms

--- 10.0.0.3 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3078ms
rtt min/avg/max/mdev = 0.125/0.171/0.296/0.073 ms
root@zhanglei-VirtualBox:~/Workspace/share/04-broadcast/04-broadcast#
```

h2 ping h1, h3

```
"Node: h3"
root@zhanglei-VirtualBox:~/Workspace/share/04-broadcast/04-broadcast# 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=0.067 ms
64 bytes from 10.0.0.1: icmp_seq=2 ttl=64 time=0.113 ms
64 bytes from 10.0.0.1: icmp_seq=3 ttl=64 time=0.083 ms
64 bytes from 10.0.0.1: icmp_seq=4 ttl=64 time=0.180 ms

--- 10.0.0.1 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3079ms
rtt min/avg/max/mdev = 0.067/0.110/0.180/0.045 ms
root@zhanglei-VirtualBox:~/Workspace/share/04-broadcast/04-broadcast# 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=0.274 ms
64 bytes from 10.0.0.2: icmp_seq=2 ttl=64 time=0.113 ms
64 bytes from 10.0.0.2: icmp_seq=3 ttl=64 time=0.077 ms
64 bytes from 10.0.0.2: icmp_seq=4 ttl=64 time=0.127 ms

--- 10.0.0.2 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3054ms
rtt min/avg/max/mdev = 0.077/0.147/0.274/0.076 ms
root@zhanglei-VirtualBox:~/Workspace/share/04-broadcast/04-broadcast#
```

h3 ping h1, h2

## 2. iperf 测试(h1:client; h2,h3:servers):

```
root@zhanglei-VirtualBox:~/Workspace/share/04-broadcast/04-broadcast# iperf -c 10.0.0.2 -t 30
-----
Client connecting to 10.0.0.2, TCP port 5001
TCP window size: 85.3 KByte (default)
-----
[ 13] local 10.0.0.1 port 36180 connected with 10.0.0.2 port 5001
[ ID] Interval      Transfer    Bandwidth
[ 13] 0.0-30.5 sec  24.0 MBytes  6.60 Mbits/sec
root@zhanglei-VirtualBox:~/Workspace/share/04-broadcast/04-broadcast#
```

h1-h2

```
root@zhanglei-VirtualBox:~/Workspace/share/04-broadcast/04-broadcast# iperf -c 10.0.0.3 -t 30
-----
Client connecting to 10.0.0.3, TCP port 5001
TCP window size: 85.3 KByte (default)
-----
[ 13] local 10.0.0.1 port 39820 connected with 10.0.0.3 port 5001
[ ID] Interval      Transfer    Bandwidth
[ 13] 0.0-30.2 sec  13.6 MBytes  3.78 Mbits/sec
root@zhanglei-VirtualBox:~/Workspace/share/04-broadcast/04-broadcast#
```

h1-h3

```
root@zhanglei-VirtualBox:~/Workspace/share/04-broadcast/04-broadcast# 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 36180
[ ID] Interval      Transfer    Bandwidth
[ 14] 0.0-30.9 sec  24.0 MBytes  6.51 Mbits/sec

```

h2

```
root@zhanglei-VirtualBox:~/Workspace/share/04-broadcast/04-broadcast# 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 39820
[ ID] Interval      Transfer    Bandwidth
[ 14] 0.0-30.2 sec  13.6 MBytes  3.78 Mbits/sec

```

h3

3. iperf 测试(h1:server; h2,h3:clients):

```

Node: h2
root@zhanglei-VirtualBox:~/Workspace/share/04-broadcast/04-broadcast/1.0# iperf
-c 10.0.0.1 -t 30

-----
Client connecting to 10.0.0.1, TCP port 5001
TCP window size: 85.3 KByte (default)
-----
[ 13] local 10.0.0.2 port 58836 connected with 10.0.0.1 port 5001
[ ID] Interval      Transfer    Bandwidth
[ 13] 0.0-30.3 sec  32.6 MBytes  9.04 Mbits/sec
root@zhanglei-VirtualBox:~/Workspace/share/04-broadcast/04-broadcast/1.0#
```

h2-h1

```

Node: h3
root@zhanglei-VirtualBox:~/Workspace/share/04-broadcast/04-broadcast/1.0# iperf
-c 10.0.0.1 -t 30

-----
Client connecting to 10.0.0.1, TCP port 5001
TCP window size: 85.3 KByte (default)
-----
[ 13] local 10.0.0.3 port 55000 connected with 10.0.0.1 port 5001
[ ID] Interval      Transfer    Bandwidth
[ 13] 0.0-30.1 sec  33.9 MBytes  9.44 Mbits/sec
root@zhanglei-VirtualBox:~/Workspace/share/04-broadcast/04-broadcast/1.0#
```

h3-h1

```


Node: h1
root@zhanglei-VirtualBox:~/Workspace/share/04-broadcast/04-broadcast/1.0# iperf
-s

-----
Server listening on TCP port 5001
TCP window size: 85.3 KByte (default)
-----
[ 14] local 10.0.0.1 port 5001 connected with 10.0.0.2 port 58836
[ 15] local 10.0.0.1 port 5001 connected with 10.0.0.3 port 55000
[ ID] Interval      Transfer    Bandwidth
[ 14] 0.0-31.1 sec  32.6 MBytes  8.81 Mbits/sec
[ 15] 0.0-31.1 sec  33.9 MBytes  9.12 Mbits/sec

```

h1

#### 4. 数据包环路:



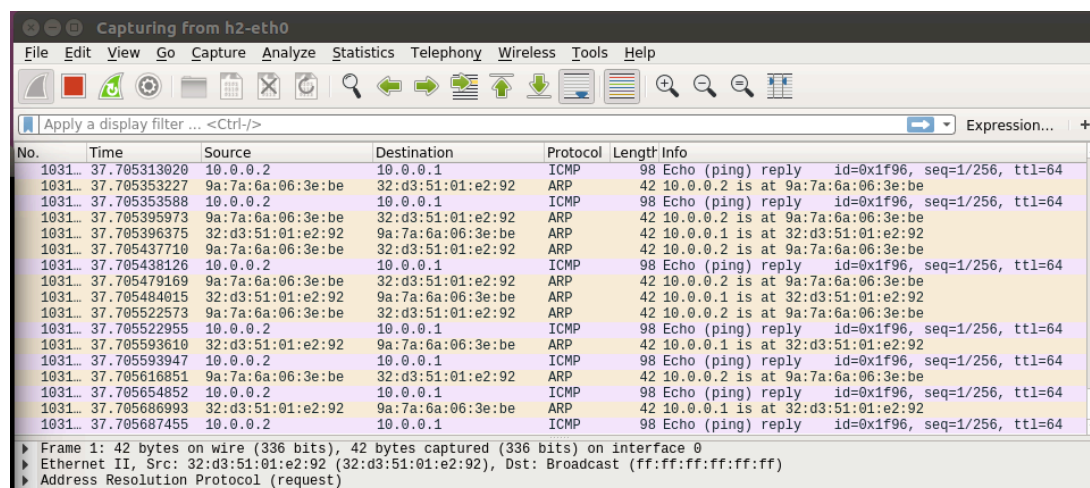
```

root@zhanglei-VirtualBox:~/Workspace/share/04-broadcast/04-broadcast/ring# ping -c 1 10.0.0.2
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=0.395 ms

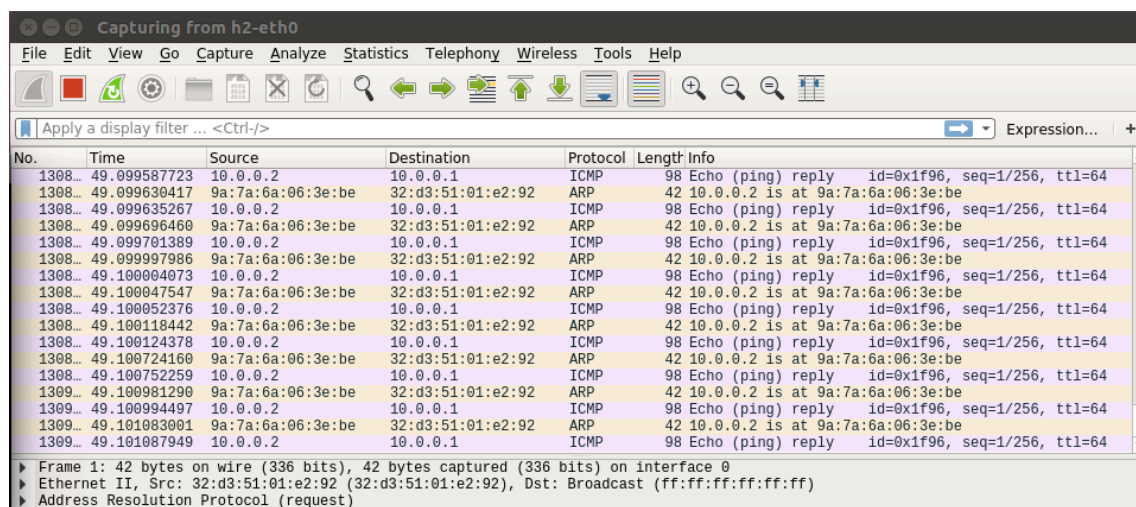
--- 10.0.0.2 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 0.395/0.395/0.395/0.000 ms
root@zhanglei-VirtualBox:~/Workspace/share/04-broadcast/04-broadcast/ring#

```

h1 发送 1 个数据包到 h2



Wireshark 抓包结果-1



Wireshark 抓包结果-2

## 五、实验分析

1. 成功完成 `broadcast_packet` 函数的编写, h1, h2, h3 三个节点能够互相 ping 通;
2. `iperf` 测试(h1:client)时, h1 连接 h1, h1 连接 h3 的实际带宽都明显低于设置带宽 10Mb/s;
3. `iperf` 测试(h1:server)时, h2 连接 h1, h3 连接 h1 的实际带宽都接近但略低于设定带宽 10Mb/s;
4. 修改 `three_nodes_bw.py` 并搭建环路成功;
5. 成功观察到数据包在环路中传播;

## 六、调研解释

1. `iperf` 测试(h1:client)时, 带宽显著低于设定带宽 10Mb/s:

在网上查阅了很多资料, 但没有发现说明这种现象的文章, 所以我根据我们自己编写的 `broadcast_packet` 函数以及设定带宽谈一下我的想法:

在我们设定的广播网络带宽中, h1 带宽为 20Mbps, h2 和 h3 的带宽为 10Mbps, 当 h1 做客户端时, 集线器 b1 可以将收到的来自 h1 的数据在一个循环里广播给 h2 和 h3, 同时将来自 h2 和 h3 的数据广播到 h1, 由于 h2 和 h3 的带宽只有 10Mbps, 所以在 h1 接受来自 h2 和 h3 的数据时, 理想的最大的带宽也只能达到 10Mbps。

但是集线器 b1 并不能同时收集并广播来自 h2 和 h3 的数据包, 而必须等待到其中一个数据包广播完毕才能进行下一次的调用, 所以两次调用 `broadcast_packet` 函数会出现时间间隔, 这个时间间隔降低了有效的数据传输的带宽;

并且当集线器 b1 收到来自 h2 的数据包时并不是只广播给 h1, 而是将数据包广播到网络链路中的所有其他节点, 也就是 h1 和 h3, 这也造成了 h1 接收数据包的带宽达不到 10Mbps 甚至减半的结果;

这也能很好的解释为什么当 h1 做服务端的时候, h2 和 h3 的带宽基本能够达到 10Mbps, h1 的带宽也基本能达到 20Mbps 的现象;

基于以上分析, 可以猜想, 当广播网络中的节点数继续增加时, 广播网络的效率就会越低, 且广播网络中的数据传输带宽约等于发送方的带宽除以广播网络中的节点个数。

## 七、反思总结

1. 本次实验加深了我对广播网络的理解, 让我对广播网络链路如何传播数据包有了更加清晰的认识, 同时我也通过调研的过程了解到了广播网络的优点和缺点;

## 八、 参考文献

i

ii

---

<sup>i</sup> 中国科学院大学 2020 春计算机网络研讨课 04-广播网络实验课件

<sup>ii</sup> 中国科学院大学 2020 春计算机网络研讨课 04-广播网络实验样例代码