

---

# 广播网络实验

学号：2016K8009908007

姓名：薛峰

## 一、实验内容

### 1、实现节点广播

- (1)实现 main.c 中的 broadcast\_packet 函数；
- (2)使用 three\_nodes\_bw.py 拓扑文件, 三个节点相互能够 ping 通。

### 2、广播网络传输效率

- (1)利用 iperf 测试网络链路的利用效率；
- (2)两种场景：H1: iperf client; H2, H3: servers; H1: iperf server; H2, H3: clients。

### 3、数据包在环路中不断转发

- (1)构建环形拓扑；
- (2)验证该拓扑下节点广播会产生数据包环路。

## 二、实验流程

### 1、broadcast\_packet 函数

```
void broadcast_packet(iface_info_t *iface, const char *packet, int len)
{
    // TODO: broadcast packet
    fprintf(stdout, "TODO: broadcast packet here.\n");

    iface_info_t *entry = NULL;
    list_for_each_entry(entry, &instance->iface_list, list) {
        if (entry->fd != iface->fd)
            iface_send_packet(entry, packet, len);
    }
}
```

## 2、环形网络拓扑

```
#!/usr/bin/python

import sys
import os.path
from mininet.topo import Topo
from mininet.net import Mininet
from mininet.link import TCLink
from mininet.cli import CLI

def clearIP(n):
    for iface in n.intfList():
        n.cmd('ifconfig %s 0.0.0.0' % (iface))

class BroadcastTopo(Topo):
    def build(self):
        h1 = self.addHost('h1')
        h2 = self.addHost('h2')
        b1 = self.addHost('b1')
        b2 = self.addHost('b2')
        b3 = self.addHost('b3')

        self.addLink(h1, b1)
        self.addLink(h2, b2)
        self.addLink(b1, b2)
        self.addLink(b1, b3)
        self.addLink(b2, b3)

if __name__ == '__main__':
    if not os.path.exists('/sbin/ethtool'):
        print 'ethtool not found, please install it using `apt install ethtool`'
        sys.exit(1)

    topo = BroadcastTopo()
    net = Mininet(topo = topo, link = TCLink, controller = None)

    h1, h2, b1, b2, b3 = net.get('h1', 'h2', 'b1', 'b2', 'b3')
    h1.cmd('ifconfig h1-eth0 10.0.0.1/8')
    h2.cmd('ifconfig h2-eth0 10.0.0.2/8')
    clearIP(b1)
    clearIP(b2)
    clearIP(b3)

    for h in [ h1, h2]:
        h.cmd('./disable_offloading.sh')
        h.cmd('./disable_ipv6.sh')

    net.start()
    CLI(net)
    net.stop()
```

## 三、实验结果及分析

### 1、实验结果

测试 3 台 host 之间是否能够 ping 通



H1: iperf client; H2, H3: iperf servers

```
"Node: h1"
root@feng-VirtualBox:~/Lab/P04/04-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 54596 connected with 10.0.0.2 port 5001
[ ID] Interval      Transfer    Bandwidth
[ 13] 0.0-30.5 sec  16.6 MBytes  4.58 Mbits/sec
root@feng-VirtualBox:~/Lab/P04/04-broadcast#
```

```
"Node: h1"
root@feng-VirtualBox:~/Lab/P04/04-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 41024 connected with 10.0.0.3 port 5001
[ ID] Interval      Transfer    Bandwidth
[ 13] 0.0-30.3 sec  17.8 MBytes  4.91 Mbits/sec
root@feng-VirtualBox:~/Lab/P04/04-broadcast#
```

数据包在环路中不断转发

\*h2-eth0

文件(F) 编辑(E) 视图(V) 跳转(G) 捕获(C) 分析(A) 统计(S) 电话(Y) 无线(W) 工具(T) 帮助(H)

应用显示过滤器 ... <Ctrl-/> 表达式...

| No. | Time        | Source            | Destination       | Protocol | Length | Info                             |
|-----|-------------|-------------------|-------------------|----------|--------|----------------------------------|
| 254 | 0.023496680 | 32:47:22:e2:e9:73 | 0a:92:df:d3:2f:60 | ARP      | 42     | 10.0.0.2 is at 32:47:22:e2:e9:73 |
| 255 | 0.023530444 | 32:47:22:e2:e9:73 | 0a:92:df:d3:2f:60 | ARP      | 42     | 10.0.0.2 is at 32:47:22:e2:e9:73 |
| 256 | 0.023541640 | 32:47:22:e2:e9:73 | 0a:92:df:d3:2f:60 | ARP      | 42     | 10.0.0.2 is at 32:47:22:e2:e9:73 |
| 257 | 0.023548825 | 32:47:22:e2:e9:73 | 0a:92:df:d3:2f:60 | ARP      | 42     | 10.0.0.2 is at 32:47:22:e2:e9:73 |
| 258 | 0.023557051 | 32:47:22:e2:e9:73 | 0a:92:df:d3:2f:60 | ARP      | 42     | 10.0.0.2 is at 32:47:22:e2:e9:73 |
| 259 | 0.023563970 | 32:47:22:e2:e9:73 | 0a:92:df:d3:2f:60 | ARP      | 42     | 10.0.0.2 is at 32:47:22:e2:e9:73 |
| 260 | 0.023572211 | 32:47:22:e2:e9:73 | 0a:92:df:d3:2f:60 | ARP      | 42     | 10.0.0.2 is at 32:47:22:e2:e9:73 |
| 261 | 0.023579185 | 32:47:22:e2:e9:73 | 0a:92:df:d3:2f:60 | ARP      | 42     | 10.0.0.2 is at 32:47:22:e2:e9:73 |
| 262 | 0.023587060 | 32:47:22:e2:e9:73 | 0a:92:df:d3:2f:60 | ARP      | 42     | 10.0.0.2 is at 32:47:22:e2:e9:73 |
| 264 | 0.023602969 | 32:47:22:e2:e9:73 | 0a:92:df:d3:2f:60 | ARP      | 42     | 10.0.0.2 is at 32:47:22:e2:e9:73 |
| 266 | 0.023618154 | 32:47:22:e2:e9:73 | 0a:92:df:d3:2f:60 | ARP      | 42     | 10.0.0.2 is at 32:47:22:e2:e9:73 |
| 267 | 0.023625091 | 32:47:22:e2:e9:73 | 0a:92:df:d3:2f:60 | ARP      | 42     | 10.0.0.2 is at 32:47:22:e2:e9:73 |
| 268 | 0.023633011 | 32:47:22:e2:e9:73 | 0a:92:df:d3:2f:60 | ARP      | 42     | 10.0.0.2 is at 32:47:22:e2:e9:73 |
| 270 | 0.023648158 | 32:47:22:e2:e9:73 | 0a:92:df:d3:2f:60 | ARP      | 42     | 10.0.0.2 is at 32:47:22:e2:e9:73 |
| 272 | 0.023663225 | 32:47:22:e2:e9:73 | 0a:92:df:d3:2f:60 | ARP      | 42     | 10.0.0.2 is at 32:47:22:e2:e9:73 |
| 273 | 0.023670023 | 32:47:22:e2:e9:73 | 0a:92:df:d3:2f:60 | ARP      | 42     | 10.0.0.2 is at 32:47:22:e2:e9:73 |
| 274 | 0.044693481 | 32:47:22:e2:e9:73 | 0a:92:df:d3:2f:60 | ARP      | 42     | 10.0.0.2 is at 32:47:22:e2:e9:73 |
| 278 | 0.044750614 | 32:47:22:e2:e9:73 | 0a:92:df:d3:2f:60 | ARP      | 42     | 10.0.0.2 is at 32:47:22:e2:e9:73 |
| 280 | 0.044774611 | 32:47:22:e2:e9:73 | 0a:92:df:d3:2f:60 | ARP      | 42     | 10.0.0.2 is at 32:47:22:e2:e9:73 |

Frame 268: 42 bytes on wire (336 bits), 42 bytes captured (336 bits) on interface 0  
Ethernet II, Src: 32:47:22:e2:e9:73 (32:47:22:e2:e9:73), Dst: 0a:92:df:d3:2f:60 (0a:92:df:d3:2f:60)  
Address Resolution Protocol (reply)

```
"Node: h1"
root@feng-VirtualBox:~/Lab/P04/04-broadcast# 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=134 ms

--- 10.0.0.2 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 134.142/134.142/134.142/0.000 ms
root@feng-VirtualBox:~/Lab/P04/04-broadcast#
```

## 2、结果分析

- 
- broadcast\_packe 函数部分编写正确，由结果看出连接到 hub 的节点能够相互 ping 通。
  - 对于测试链路效率，当 h1 作为 Client，h2、h3 作为 Server 时，h1 同时向 h2 和 h3 发包，发给 h2 的包发到 hub 之后会同时发向 h2 和 h3，发给 h3 的包发给 hub 之后也会同时发向 h2 和 h3，因此，hub 到 h2 和 h3 的链路上都分别有两个包，因为 hub 到 h2 和 h3 的带宽都为 10MB/s，因此链路上两个包的带宽之和应该小于 10，在图中也可以看出，一个带宽为 4.58Mb/s，另一个为 4.91Mb/s；当 h1 作为 Server，h2、h3 作为 Client 时，h2 发出去的包发到 hub 之后会同时发向 h1 和 h3，因为链路带宽是双向的，因此发向 h3 的包不会影响 h3 发出包，因此 h3 发出的包的带宽仍然为 10Mb/s 左右，同理 h2 发出去的包也为 10MB/s 左右。
  - 在环路中，一个包发出后，三个 hub 不断转发，导致该包在环路中不停地旋转，浪费资源，因此在设计中应避免环路。