

计算机网络研讨课实验报告

冯吕 2015K8009929049

2018 年 4 月 10 日

实验题目

广播网络实验

实验内容

在本次实验中，需要构建多节点网络。实验内容主要分为三个部分：

- 实现一个节点广播，在该网络结构中，有一个 *hub* 节点 *b1*，另外还有三个主机节点 *h1*, *h2*, *h3*，每个主机节点均与 *b1* 相连，*b1* 收到每一个数据包以后，需要将该包从其他端口转发出去，从而实现广播的功能。
- 使用 *iperf* 工具测量上面创建的广播网络的链路利用效率。
- 自定义一个网络拓扑结构，在该结构中，有三个 *hub* 节点，分别为 *b1*, *b2*, *b3*，另外，还有两个主机节点：*h1*, *h2*，*h1* 连接到 *b1*，*h2* 连接到 *b2*。由 *h1* 向 *h2* 发送一个数据包，通过抓包观察一个数据包不断被转发的现象。

实验需要实现的内容主要有：*main.c* 中实现将收到的每一个包从其他端口转发出去；另外，需要写一个 *Python* 脚本实现上面写到的自定义网络拓扑，进行环路转发实验。

实验流程

本次实验中，首先需要实现一个广播网络，需要实现 *main.c* 中的 *TODO* 部分：将收到的包从其他端口转发出去。由于所有端口都已经存储在 *instance->iface_list* 数据结构中，因此，通过宏定义 *list_for_each_entry(pos, head, member)* 找出所有其他端口，然后通过 *iface_send_packet* 函数将包发送出去即可。实现代码如下：

```
1 void broadcast_packet(iface_info_t *iface, const char *packet, int len)
2 {
3     iface_info_t *iface_t = NULL;
4     list_for_each_entry(iface_t, &instance->iface_list, list)
5         if (iface_t->fd != iface->fd)
6             iface_send_packet(iface_t, packet, len);
7 }
```

之后，通过脚本 *three_nodes_bw.py* 创建网络拓扑，然后在 *b1* 中运行 *main.c* 编译后的 *hub* 程序，之后，通过 *ping* 命令来查看 *h1*, *h2*, *h3* 三个节点两两之间是否能够 *ping* 通，如果能够 *ping* 通，则说明实现了节点广播。

节点广播网络构建好之后，需要使用 *iperf* 工具来测试网络的链路利用率。分两种方式测量，一种方式是 *h1* 作为 *client*，*h2,h3* 作为 *servers*；另一种方式是 *h1* 作为 *server*，*h2,h3* 作为 *clients*。

在最后一部分实验内容中，需要创建一个环路网络，网络具体结构见**实验内容**部分。因此，需要完成一个创建该网络拓扑的脚本，脚本内容如下：

```

1  #!/usr/bin/env python2
2
3  from mininet.topo import Topo
4  from mininet.net import Mininet
5  from mininet.link import TCLink
6  from mininet.cli import CLI
7
8  # Mininet will assign an IP address for each interface of a node
9  # automatically, but hub or switch does not need IP address.
10 def clearIP(n):
11     for iface in n.intfList():
12         n.cmd('ifconfig %s 0.0.0.0' % (iface))
13
14 class BroadcastTopo(Topo):
15     def build(self):
16         h1 = self.addHost('h1')
17         h2 = self.addHost('h2')
18         b1 = self.addHost('b1')
19         b2 = self.addHost('b2')
20         b3 = self.addHost('b3')
21
22         self.addLink(b1, b2, bw=20)
23         self.addLink(b1, b3, bw=20)
24         self.addLink(b2, b3, bw=20)
25         self.addLink(h1, b1, bw=20)
26         self.addLink(h2, b2, bw=20)
27
28 if __name__ == '__main__':
29     topo = BroadcastTopo()
30     net = Mininet(topo = topo, link = TCLink, controller = None)
31
32     h1, h2, b1, b2, b3 = net.get('h1', 'h2', 'b1', 'b2', 'b3')
33     h1.cmd('ifconfig h1-eth0 10.0.0.1/8')
34     h2.cmd('ifconfig h2-eth0 10.0.0.2/8')
35     clearIP(b1)
36     clearIP(b2)
37     clearIP(b3)
38
39     h1.cmd('./disable_offloading.sh')

```

```
40     h2.cmd( './disable_offloading.sh' )  
41  
42     net.start()  
43     CLI(net)  
44     net.stop()
```

创建好网络拓扑之后，由 *h1* 向 *h2* 发送一个数据包，通过抓包来观察一个数据包不断被转发的现象（使用 *wireshark*）来进行观察。

实验结果

略。

结果分析

略。