

# Lab3 实验报告

## 一. 实验任务

1. 实现节点广播的 `broadcast_packet` 函数，将收到的包广播给其他节点。
2. 利用 `iperf` 验证广播网络的效率。
3. 自己动手构建环形拓扑，验证该拓扑下节点广播会产生数据包环路。

## 二. 实验过程

### 1. 实现节点广播

我们需要实现 `broadcast_packet` 函数，这个函数读入传入端口，传入的包和包的长度，我们需要把这个包广播到除了传入端口以外的其他端口，伪代码如下所示：

```
foreach iface in iface_list:  
    if iface != rx_iface:  
        iface_send_packet(iface, packet, len);
```

图 1：节点广播的伪代码

实现也很好实现，我们把伪代码翻译成 C 语言代码即可：

```
void broadcast_packet(iface_info_t *iface, const char *packet, int Len)  
{  
    iface_info_t *iface_entry;  
    list_for_each_entry(iface_entry, &instance -> iface_list, list) {  
        if (iface_entry -> fd != iface -> fd)  
            iface_send_packet(iface_entry, packet, Len);  
    }  
}
```

图 2：节点广播的 C 语言代码

实现完成后，节点 h1, h2, h3 可以相互 ping 通。

### 2. 测试广播网络的传输效率

当 h1 作为 client, h2, h3 作为 server 时，iperf 结果如下：

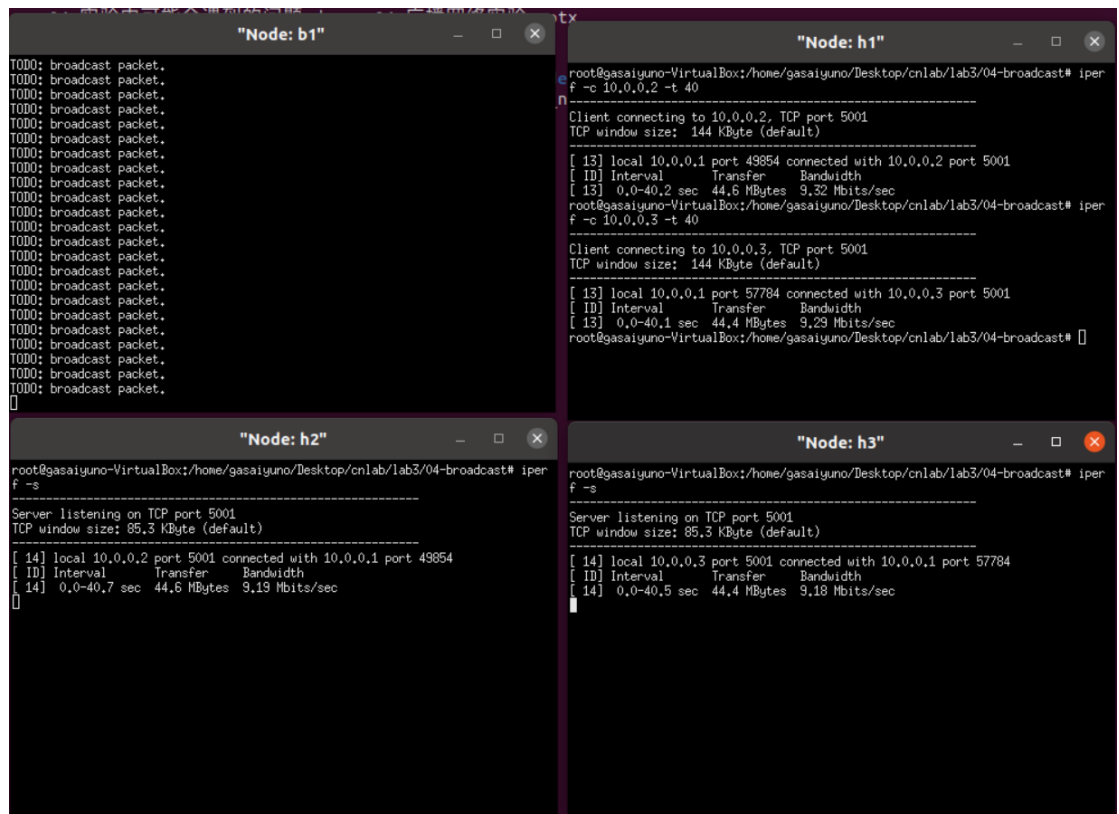


图 3: 一个 client 两个 server 的 iperf 结果

我们可以看出，我们最大带宽是 20Mbps，这里的传输速率为 9.2Mb/s，效率大约为 50%。

我们再把 h1 当作 server, h2, h3 当作 client:

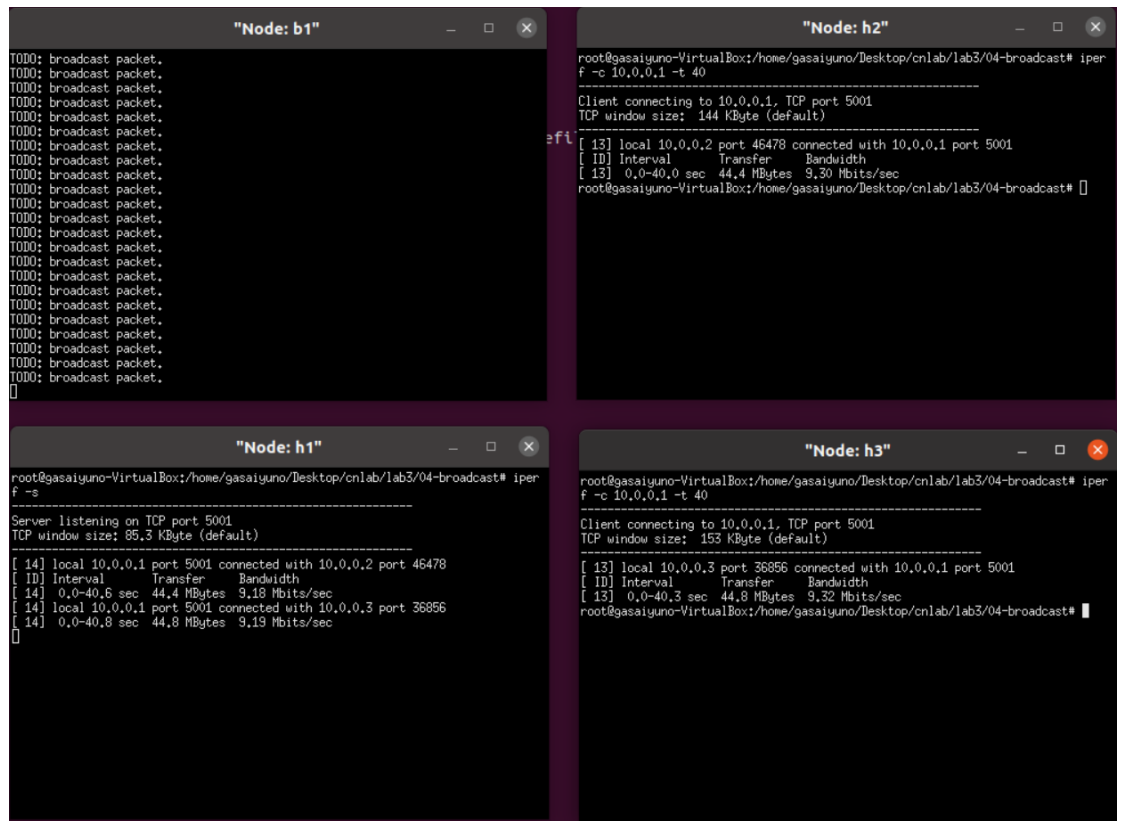


图 4：一个 server 两个 client 的 iperf 结果

效率和之前的差不多。

具体原因我们分析，是由于广播网络损失了一半的带宽，具体如下图所示：

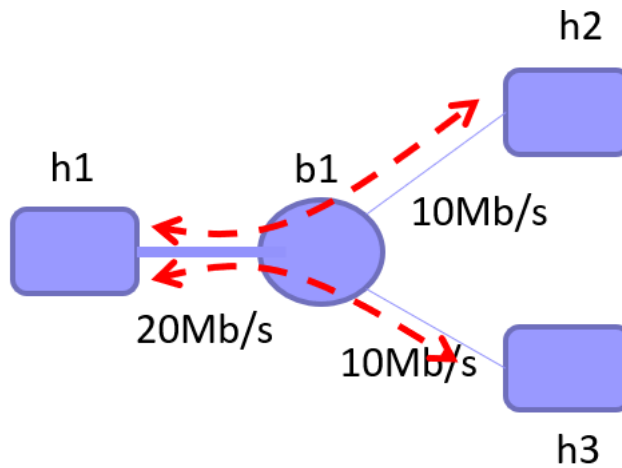


图 5：广播网络带来的带宽损失

由此我们可以看出，如果节点数目更多的话，广播网络带来的带宽损失会更大，广播网络不是一个效率高的设计。

### 3. 构建环路拓扑

我们构建如下图的环路拓扑：

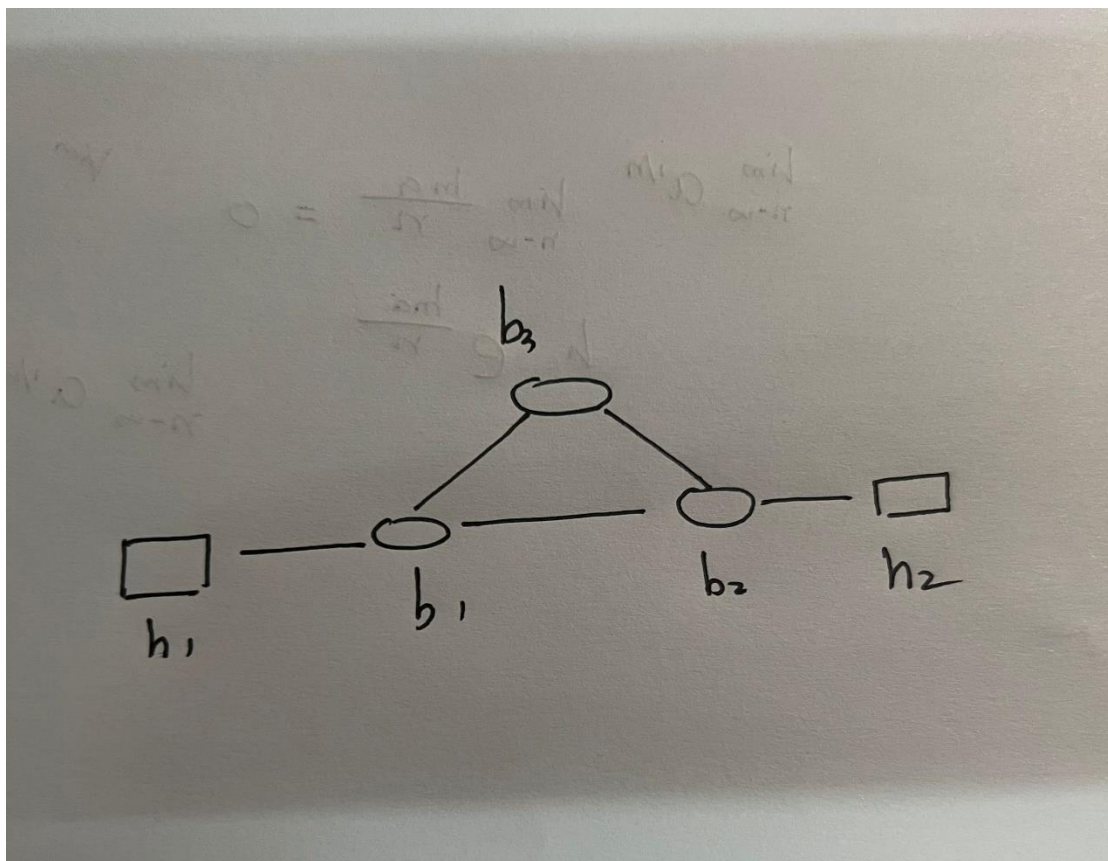


图 6：构建出的拓扑结构

我们对原本的 python 脚本进行少许更改，在 h1 节点 ping h2 节点，会发现数据包在 b1, b2, b3 形成的环路中循环无限传播：

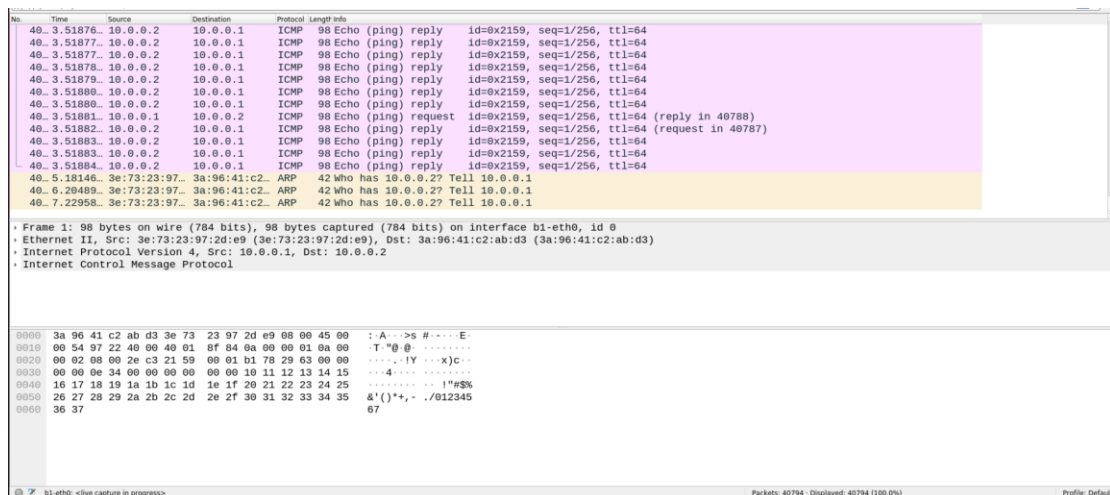


图 7：数据包在环路中无限传播

产生了数据包环路，我们在搭建广播网络的拓扑结构时一定要注意这一点，不要形成环路。

### 三. 实验总结

本次实验主要构建了一个简单的直连网络，使用广播方式传递数据包。同时我们也通过 iperf 工具探究了直连网络的效率，得出了直连网络由于广播传递数据包而效率低下的结论。然后探究了直连网络由于广播发送数据包会导致数据包在环路中无限传播的情况。得到了直连网络的一些特性和结论。