

# 00-broadcast

---

## 实验内容

实现节点广播，并验证广播网络能够正常运行

## 实验步骤

1. 构建Mininet网络拓扑和配置并启动

```
sudo ./three_nodes_bw.py
```

2. 使用ping或iperf进行Host之间的连通性测试
3. 编写**broadcast.c**内的函数**broadcast\_packet**，完成Packet的广播（对于Switch而言广播就是将该数据包原封不动的发给除了发来接口以外的其他接口）
4. 编译

```
make
```

生成可执行程序**hub**，此时的Switch只会执行广播，因此目前它还只是一个Hub（集线器）

5. 根据Mininet CLI内的dump命令得到Switch b1的PID，然后用"**Mininet 外部节点命令**"的方法，让其运行可执行程序**hub**（此处提供了可执行程序**hub-reference**作为参考实现）
6. 使用ping或iperf进行Host之间的连通性测试（Screenshots）

## 文件列表

broadcast.c # 待实现部分  
device\_internal.c # 网口管理等内部实现  
example # list\_head数据结构使用例子  
hub-reference(.32) # hub参考实现  
include # 所有相关头文件  
main.c # 主程序逻辑  
Makefile scripts # 禁止TCP Offloading, IPv6等  
three\_nodes\_bw.py # Mininet topo脚本

## 参考实现

```
void broadcast_packet(iface_info_t *iface, const char *packet, int len)
{
    iface_info_t* tx_iface = NULL;
```

```
# 遍历所有的interface，参考该语句学习该部分实验中给出的数据结构list的使用方法
list_for_each_entry(tx_iface, &instance->iface_list, list)
{
    if (tx_iface->index != iface->index) {

        # 向指定端口发包的接口函数
        iface_send_packet(tx_iface, packet, len);
    }
}
```

此次实验的实例实验已经给出，可以参考该代码学习这一部分实验的主要思路，对照结合代码学习几个关键类如**iface\_info\_t**,**ustack\_t**，及所实现的list（链表）数据结构的接口函数的使用方法（**include\list.h**）。

## 实验报告

在**writeups.md**完成实验报告,建议写完之后导出成**writeups.pdf**。

1. Implementation Explanation部分请对实验过程中实现的每一个函数进行简要的思路解释  
例如：**broadcast\_packet**函数中通过遍历比较每个接口和接收到该packet的接口的index确定是不是同一个包，将收到的packet发送给接收到出发来的接口以外的其他接口。
2. Screenshots部分请给出实验步骤中用（Screenshots）标出的步骤的实验过程给出实验截图
3. Remaining Bugs部分如果有的话则给出