

南京大学本科生实验报告

课程名称：计算机网络 任课教师：黄程远 助教：刘松岳

学院	人工智能学院	专业（方向）	人工智能
学号	211300022	姓名	刘梦杰
Email	2757400745@qq.com	开始/完成日期	2024.9.27/2024.10.10

- 1. 实验名称 计网 lab2
- 2. 实验目的 实现三种机制的交换机操作
- 3. 实验内容与核心代码

a) **basic**: 在 `myswitch.py` 文件中，在主函数 `main` 中新增一个字典名为 `table`，源 `mac` 地址、数据包的接收端口作为字典的键值对，如果数据包的目标地址在 `table` 中则直接发包，否则进行广播操作；

```
if eth is None:
    log_info("Received a non-Ethernet packet?!")
    return
else:
    table[eth.src] = fromiface
if eth.dst in mymacs:
    log_info("Received a packet intended for me")
else:
    if eth.dst in table:
        net.send_packet(table[eth.dst], packet)
    else:
        for intf in my_interfaces:
            if fromiface != intf.name:
                log_info(f"Flooding packet {packet} to {intf.name}")
                net.send_packet(intf, packet)
```

b) **time out** 机制：在 `myswitch_to.py` 中，字典 `table` 的键值对改为源 `mac` 地址、[接收端口，接收时间]，且在每个循环中遍历字典，查找时间差大于 `10s` 的并删除其键值对；

```

    table[eth.src] = (fromlface, time.time())
for mac in list(table.keys()):
    if time.time()-table[mac][1] >= 10:
        del table[mac]

```

- c) lru 机制：table 的键值对改为源 mac 地址、[接收端口，优先级]，若源 mac 地址已存在，且与接收端口不一致则改变接收端口且优先级不变，否则优先级加一，若源 mac 地址不存在则 table 键值对数满 5 就删去优先级最高的，将其余的优先级加一，再新增初始优先级为 0 的键值对；

```

if eth.src in list(table.keys()):
    if table[eth.src][0] != fromlface:
        table[eth.src] = fromlface
    else:
        table[eth.src][1] += 1
else:
    if len(table) == 5:
        m = -1
        s = eth.src
        for mac in list(table.keys()):
            if m < table[mac][1]:
                m = table[mac][1]
                s = mac
        del table[s]
    for mac in list(table.keys()):
        table[mac][1] += 1
    table[eth.src] = [fromlface, 0]

```

- d) traffic 机制：table 的键值对改为源 mac 地址、[接收端口，流量]，在目标地址在 table 中时进行发包操作后给对应流量加一，每次删除流量最小的键值对；

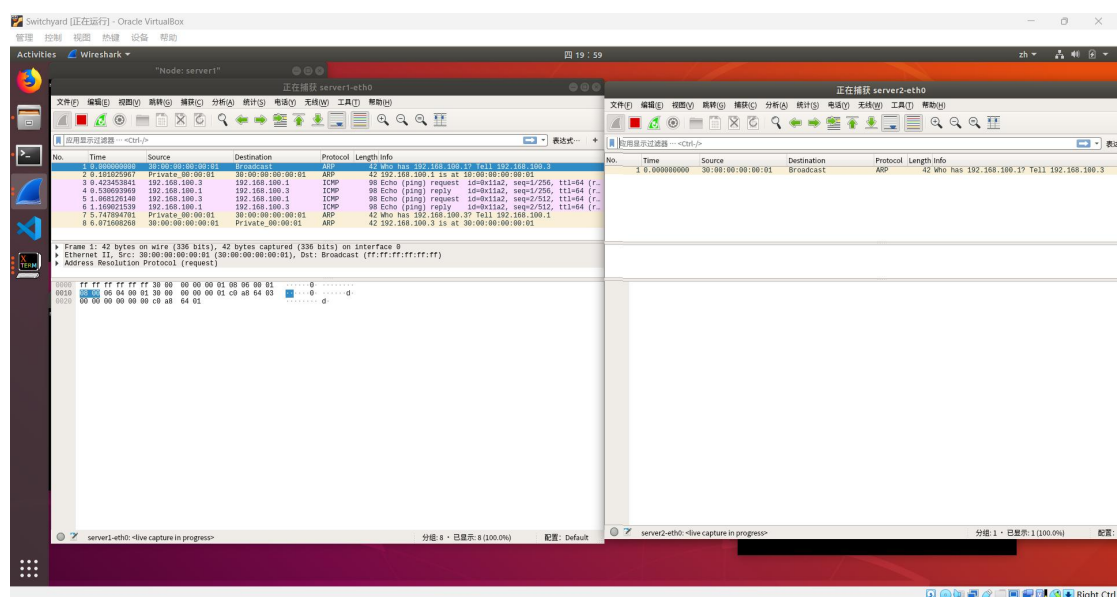
```

if len(table) == 5:
    m = float("inf")
    s = eth.src
    for mac in list(table.keys()):
        if m > table[mac][1]:
            m = table[mac][1]
            s = mac
    del table[s]
table[eth.src] = [fromiface, 0]
if eth.dst in mymacs:
    log_info("Received a packet intended for me")
else:
    if eth.dst in table:
        net.send_packet(table[eth.dst][0], packet)
        table[eth.dst][1] += 1

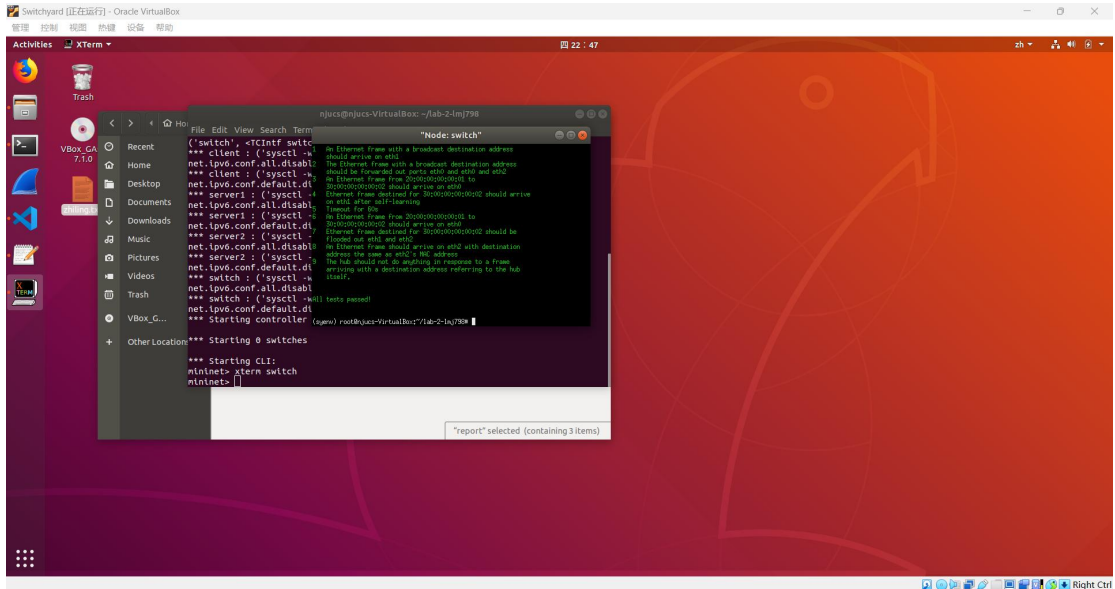
```

4. 实验测试方式与结果（与上文中序号表示的代码一致）：

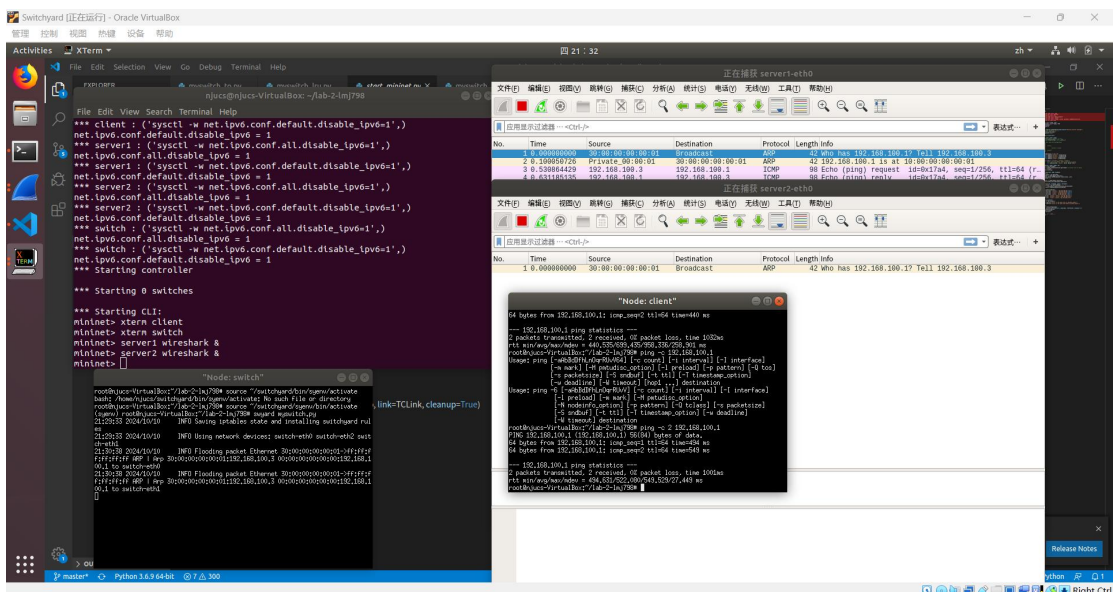
a) 如图所示：

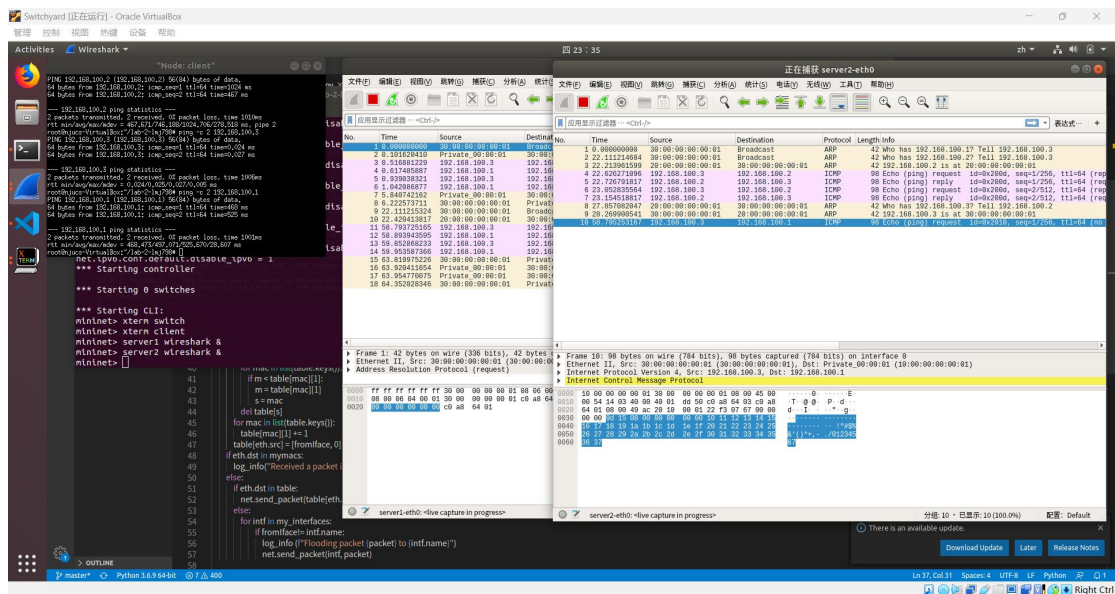


b) test 结果如下图：

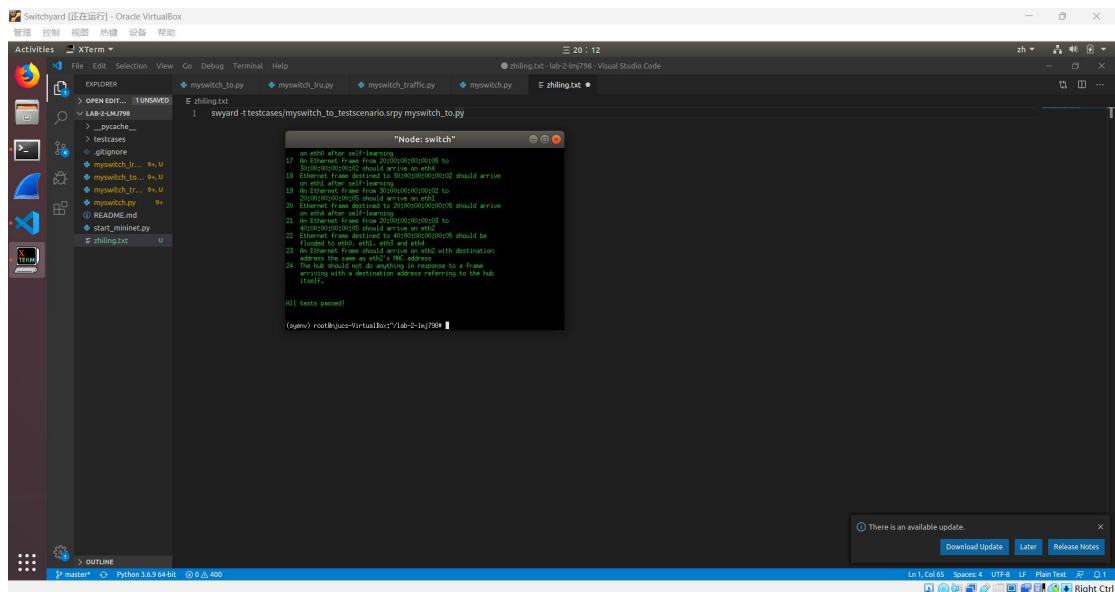


此外，在本地用 mininet 验证时，通过在 client 中发送两次指令 ping -c 2 192.168.100.1 来验证，其中两次指令的发送间隔大于 10s，在原 myswitch.py 中只有第一次产生了广播操作，而在 time out 机制下则会由于 10s 的删除机制出现两次广播操作，由 server2 的抓包结果中显示此次结果，二者如下(server1 和 server2 命名反了)；



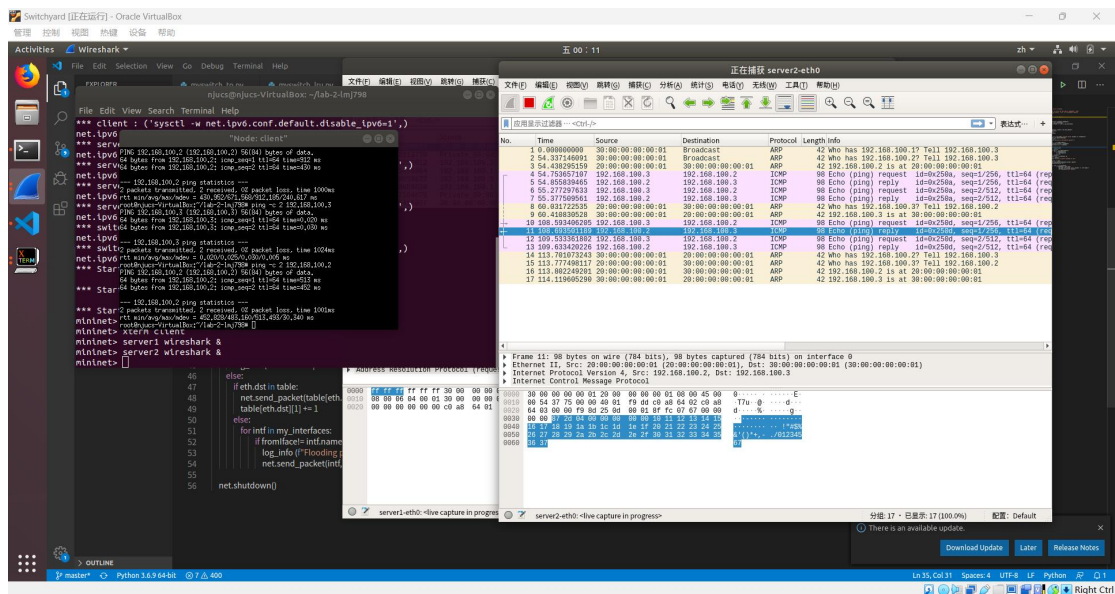


d) test 结果如下：



（为了方便测试把 table 最大数量改为 2）

依次执行 ping -c 2 192.168.100.1、ping -c 2 192.168.100.1、ping -c 2 192.168.100.1、ping -c 2 192.168.100.2、ping -c 2 192.168.100.3、ping -c 2 192.168.100.2, 192.168.100.1 即 server1 为转发流量最多的接口，故 server2 中会再次出现广播操作，如图：



6. 总结与感想

a) 用给定文件和指令测试很简单，自己抓包很复杂。