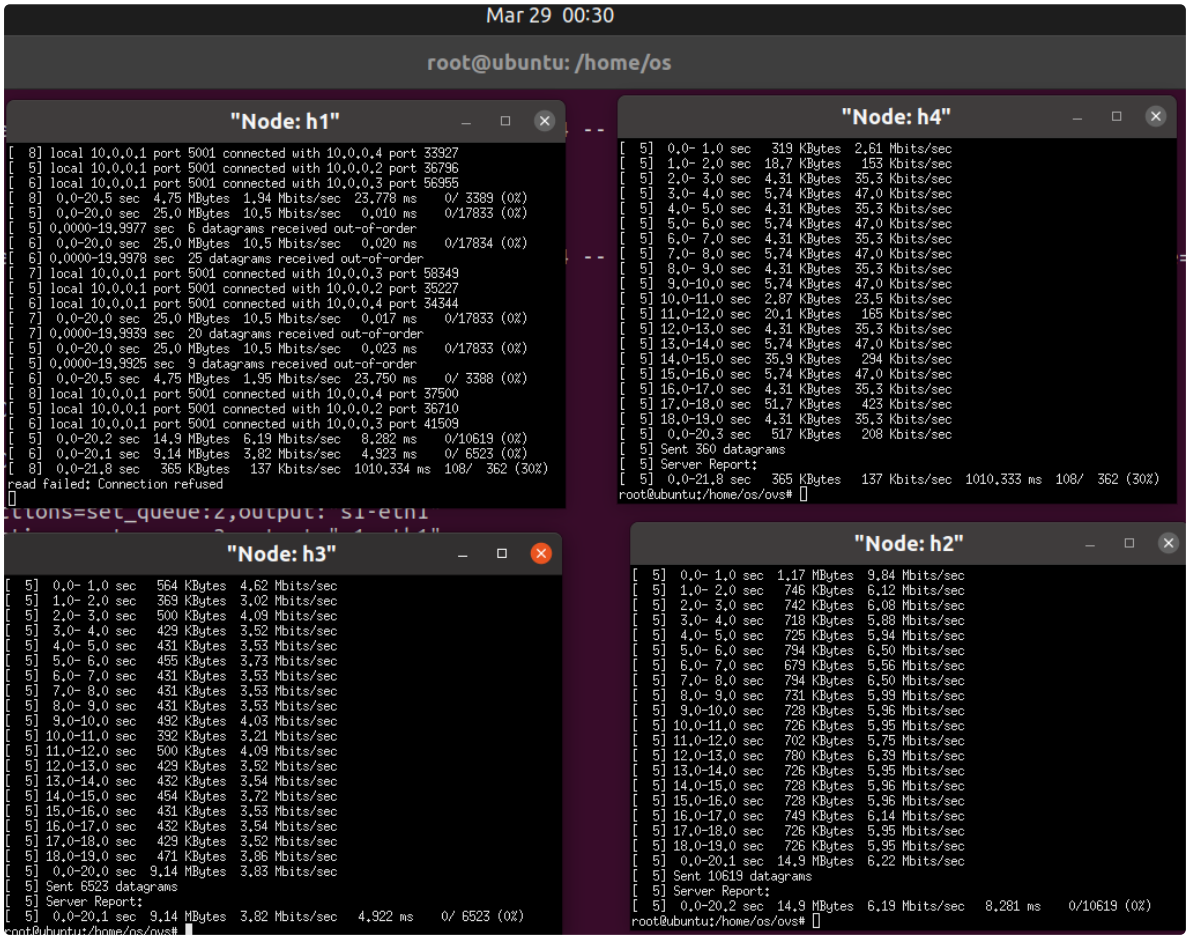
Task4

Task4 你可以通过上述三种限速的方法来达成目标，请记录你的设计过程（思路及运行指令），并将你稳定后的三个 Client 的带宽结果截图。

由于网卡限速和 meter 表限速并不能进行精确控制，因此采用队列限速。设计四个队列，总队列限速 10Mbps，其中两个限制最低速度，最后一个限制最大速度为 2Mbps。

```python
ovs-vsctl set port s1-eth1 qos=@newqos1 --
--id=@newqos1 create qos type=linux-htb \
other-config:max-rate=10000000 queues=2=@q2,3=@q3,4=@q4 -- \
--id=@q2 create queue other-config:min-rate=5000000 -- \
--id=@q3 create queue other-config:min-rate=3000000 -- \
--id=@q4 create queue other-config:max-rate=2000000
```



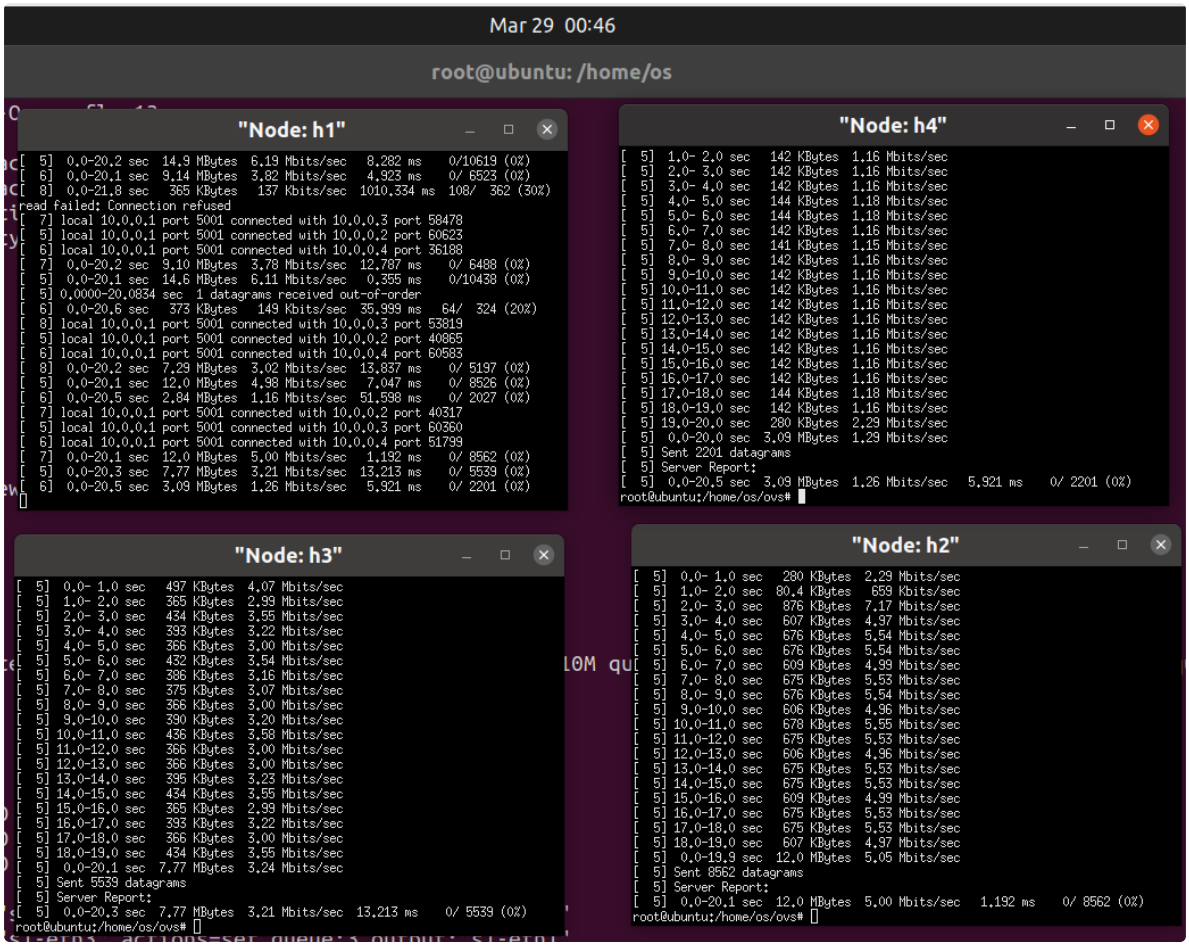
可以看到，这样设置的三个队列只设置了两个条件，q2 最低带宽是 5Mbps,q3 最低带宽是 3Mbps,但有两个缺点。

1. 没有指定队列给指定的端口
2. H4 竞争力过低，并且有较高的丢包率，不符合在保障其他的情况下尽量快的要求。

改进方法为分别指定不同队列给不同端口,并且限制 h2 和 h3 的速度

```Python
ovs-vsctl set port s1-eth1 qos=@newqos1 --
--id=@newqos1 create qos type=linux-htb \
other-config:max-rate=10000000 queues=2=@q2,3=@q3,4=@q4 -- \
--id=@q2 create queue other-config:max-rate=5500000 other-config:min-rate
=5000000 -- \
--id=@q3 create queue other-config:max-rate=3300000 other-config:min-rate
=3000000 -- \
--id=@q4 create queue other-config:max-rate=2000000
```

```Python
ovs-ofctl add-flow s1 in_port=2,action=set_queue:2,output:1 -O openflow13
ovs-ofctl add-flow s1 in_port=3,action=set_queue:3,output:1 -O openflow13
ovs-ofctl add-flow s1 in_port=4,action=set_queue:4,output:1 -O openflow13
```



由于是手动启动发送数据，h2 最后启动，因此平均带宽会相对实际较低，但这里也符合最低 5Mbps 的要求，丢包率为 0，三条队列的带宽情况也符合要求。总带宽为 9.47Mbps，接近于要求的 10Mbps.