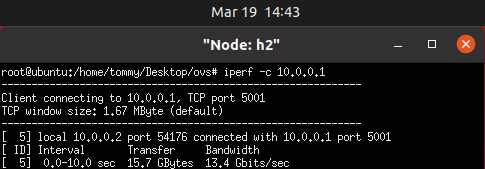
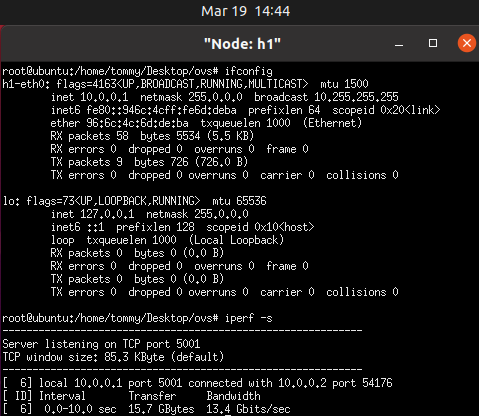
Report for Lab3: QoS Implementation with OvS

519021910913 黄喆敏

**Task1**请在你自己的环境中完成上面的连通性测试，并以截图的形式分别记录Node:h1和Node:h2中iperf的输出结果。

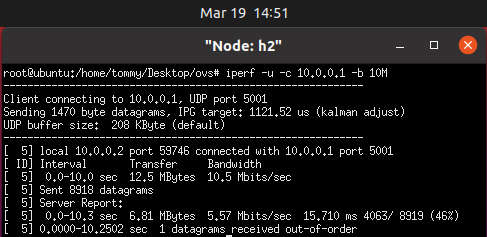
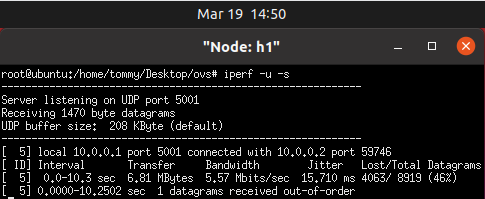
答：截图如下所示。可以看到Node:h1与Node:h2已正常联通，带宽等指标正常。



**Task2.1** 请截图记录输出结果，截图要求同Task1，并着重关注其中的带宽、抖动、丢包率等数据。

答：截图如下所示。可以看到，发送端h2的发送带宽约为10Mbps，但由于我们对接收端h1的收包速率进行了限速，当收包速率超过5Mbps时，将多余的包直接丢掉。

因此接收带宽为5.57Mbps，约为5Mbps；抖动为15.710ms；丢包率为46%，即丢掉了约一半的包，符合预期。

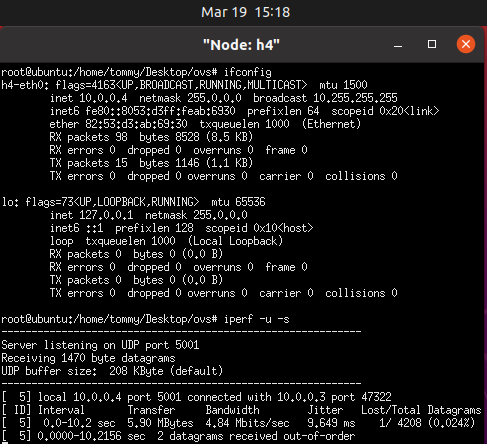
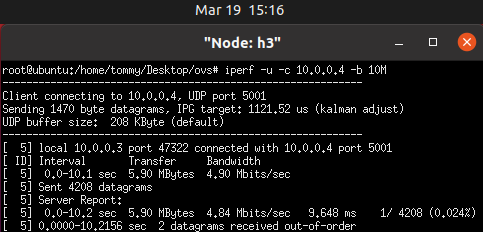


**Task2.2** 同上，此处也需要截图记录实验结果。（队列限速）

答： h3、h4的截图分别如下所示。可以看到，由于指定了队列最大速率为5Mbps，因此接收端带宽为4.84Mbps，约为5Mbps。

抖动为9.649ms，小于网卡限速的抖动。

由于**队列限速会将数据包缓存起来，而不会像网卡限速那样简单的把数据包丢弃**，因此丢包率远小于网卡限速，实验中丢包率为0.024%。



**Question 1** 尝试理解Line19,20两条指令，指出每条指令的具体工作是什么，并逐个分析其中各个参数的具体含义。

答：

Line19：为s1加入新的流内容，对于来自端口5的包，使用Meter表限速，丢弃超过5M的数据包，然后从端口6转发出去，协议为openflow13。（修改流表以使用Meter表）

Line20：输出s1上的所有流内容，包括隐藏的流，协议为openflow13。

参数含义：

s1: 指交换机

in\_port：数据包输入端口号

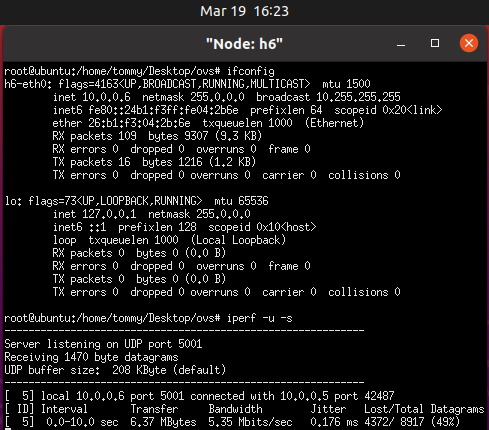
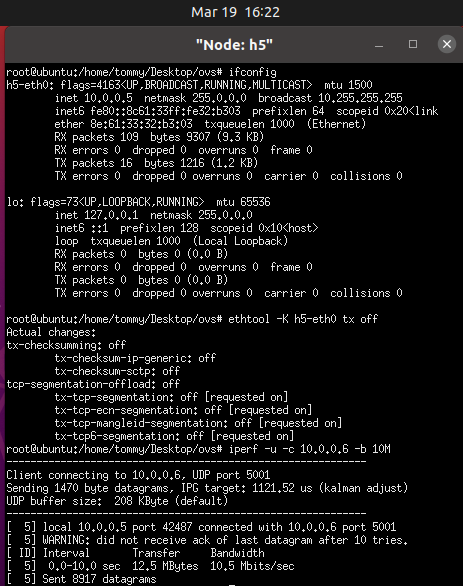
action：转发动作，meter:1指1号Meter表

output：数据包输出端口号

-O: 后面跟使用的协议

**Task2.3** 同上，请将此处的实验结果按要求截图。（Meter表）

答：h5与h6的截图如下所示。可以看到使用Meter表成功进行了限速，带宽为5.35Mbps。抖动率较低，为0.176ms。丢包率则为49%左右。



**Question 2** 到这里，你已经完成了三种限速方式的实验，并获得了三组测试数据，请你就三组数据中的带宽、抖动和丢包率等参数，对三种限速方式进行横向比较，并适当地分析原因。

答：比较如下表所示，我们同时计算出带宽与限制速度（5M）的误差率：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | 带宽 | 抖动 | 丢包率 | 带宽误差率 |
| 网卡限速 | 5.57Mbps | 15.710ms | 45.554% | 11.4% |
| 队列限速 | 4.84Mbps | 9.649ms | 0.024% | -3.2% |
| Meter表限速 | 5.35Mbps | 0.176ms | 49.030% | 7.0% |

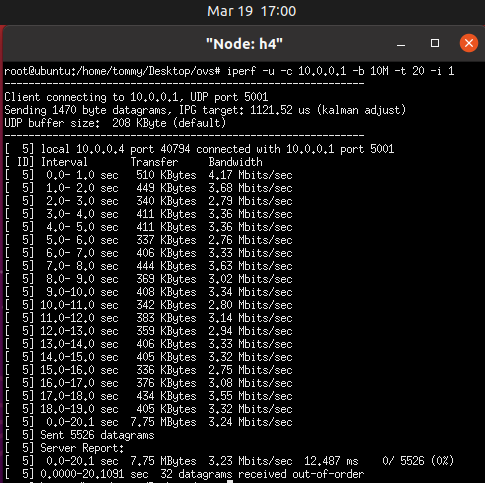
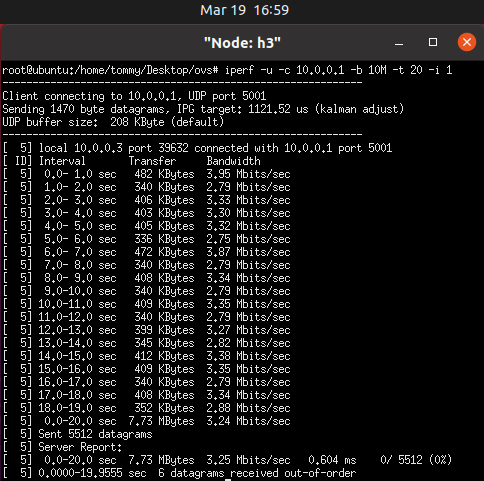
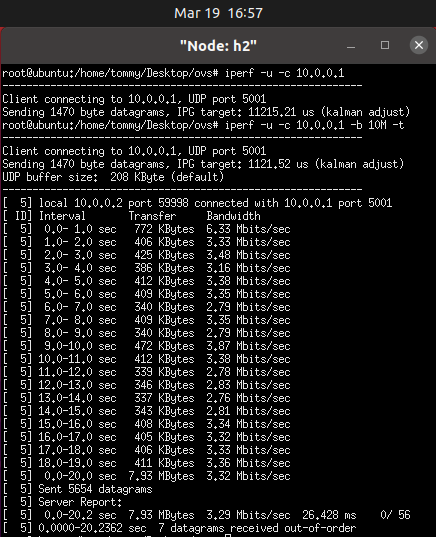
对于网卡限速，其表现较差，其实现最简单，控制粒度较粗。

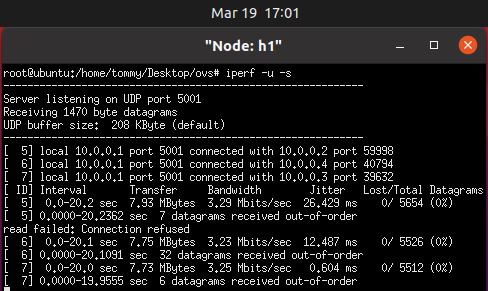
对于队列限速，其丢包率远小于另两种方法，因为当速率超过配置速率时，队列限速会将数据包缓存，而不会简单地丢弃；并且带宽误差率最小，表现最好。

对于Meter表限速，其抖动远小于另两种方法，但是丢包率最高；而带宽误差率介于网卡限速与Meter表限速两者之间。

**Task3** 在限制Server端（h1）的带宽为10Mb的前提下，观察稳定后的三个Client的带宽，将结果截图并简单分析。

答：我们首先将之前实验的配置清除，再运行命令。我们采用**队列限速**的方法限制h1的带宽。结果截图如下：





可以看到，稳定后三个Client的带宽分别为3.29Mb、3.23Mb和3.25Mb左右，三者相差不大。但三个Client的总带宽为9.77Mb，没有达到限速带宽，有一定的损耗。

**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。

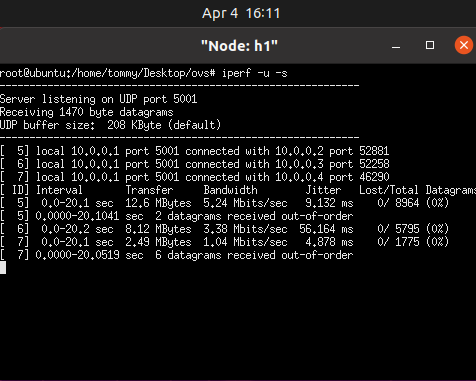
答：通过查询文档，我们发现**队列限速中可以设置最小的速率**。因此为了保证h2和h3的带宽，我们首先设置两条队列，最小速率分别为5Mb和3Mb，接着下发流表，设置对应的队列。

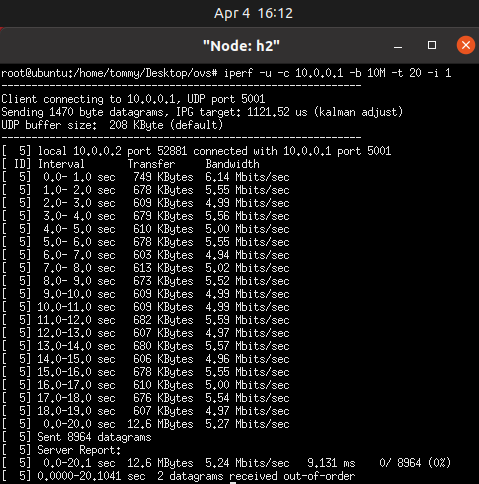
在进行第一步的实验，即只设置了min-rate，没有设置max-rate后，我们发现**h4的带宽过小，经常在几十k到几百k之间浮动**。因此为了在保证h2和h3的前提下，使h4的带宽尽量多，我们设置了**h2的带宽上限为5.5Mb，h3的带宽上限为3.5Mb**。最终h4的带宽可以达到1Mb左右。

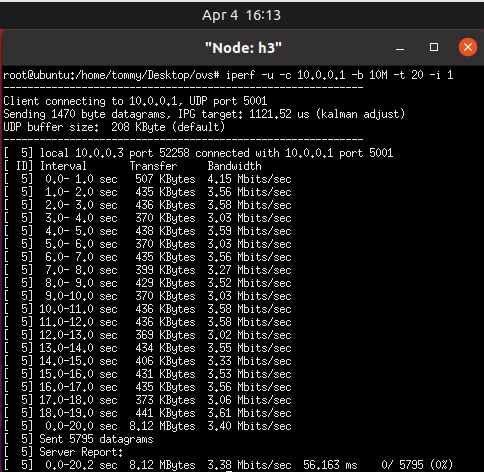
我们采取的命令如下：

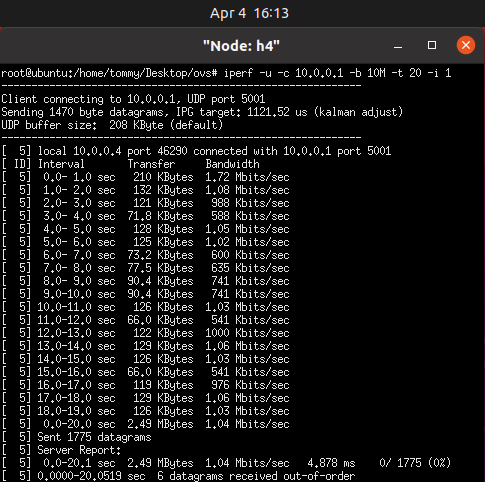
1. $ ovs-vsctl set port s1-eth1 qos=@newqos -- \
2. --id=@newqos create qos type=linux-htb
3. other-config:max-rate=10000000 queues=1=@q1,2=@q2 -- \
4. --id=@q1 create queue other-config:min-rate=5000000
5. other-config:max-rate=5500000 -- \
6. --id=@q2 create queue other-config:min-rate=3000000
7. other-config:max-rate=3500000
9. $ ovs-ofctl add-flow s1 in\_port=2,actions=set\_queue:2,output:1 -O openflow13
10. $ ovs-ofctl add-flow s1 in\_port=3,actions=set\_queue:3,output:1 -O openflow13

最终结果如下图所示，h2带宽为5.24Mb，h3带宽为3.38Mb，h4带宽为1.04Mb，符合要求。









**Reference**

[1] <https://docs.pica8.com/pages/viewpage.action?pageId=52207258>

[2] <https://docs.pica8.com/display/PicOS422sp/Configuring+Meter>

[3] <https://www.cnblogs.com/goldsunshine/p/11720310.html>

[4] <https://www.cnblogs.com/goldsunshine/p/13056429.html>