Lab3: QoS Implementation with OvS

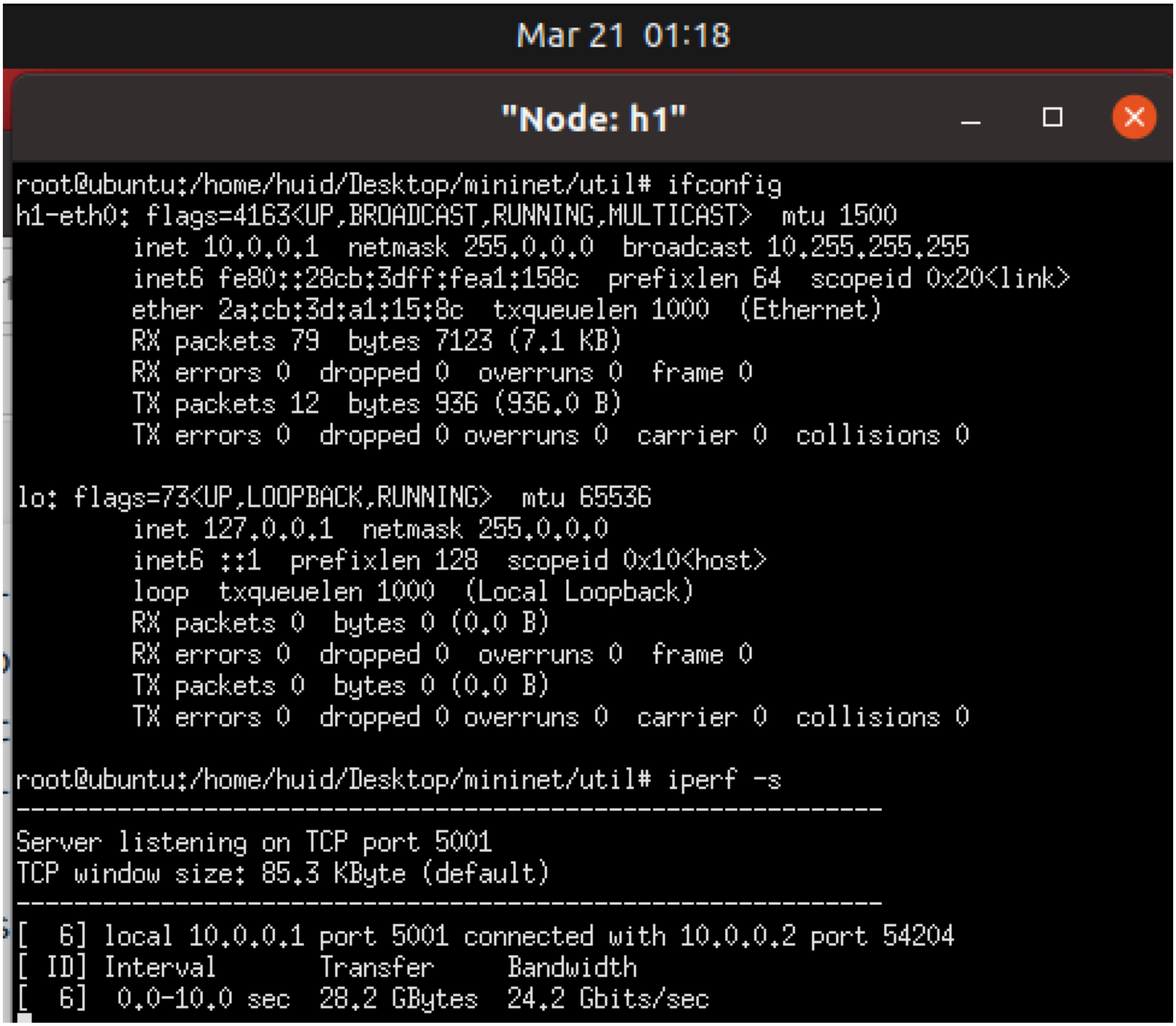
name: xuhuidong
id: 519021910861
email: 391678792hd@sjtu.edu.cn

目录

Part1: 创建网络拓扑

使用 iperf 测试两台虚拟主机 `h1` 和 `h2` 之间的连通性。

Task 1: Node: h1 的输出结果为：

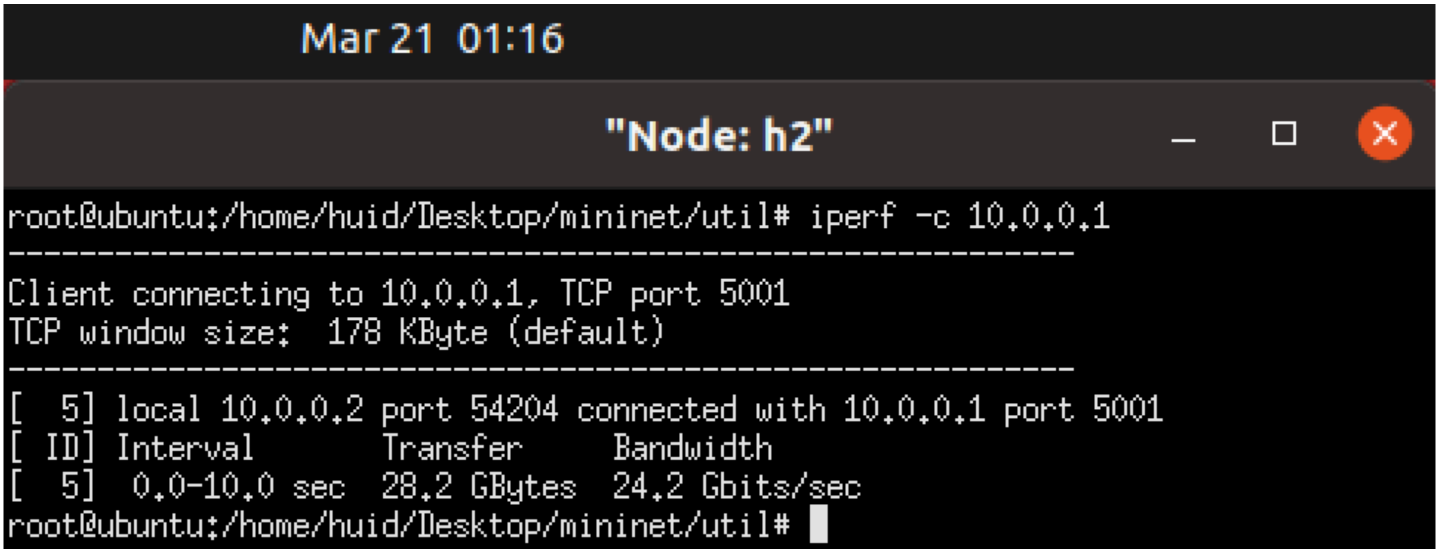


```
root@ubuntu:/home/huid/Desktop/mininet/util# ifconfig
h1-eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 10.0.0.1  netmask 255.0.0.0  broadcast 10.255.255.255
        inet6 fe80::28cb:3dff:fea1:158c  prefixlen 64  scopeid 0x20<link>
        ether 2a:cb:3d:a1:15:8c  txqueuelen 1000  (Ethernet)
        RX packets 79  bytes 7123 (7.1 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 12  bytes 936 (936.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 0  bytes 0 (0.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

root@ubuntu:/home/huid/Desktop/mininet/util# iperf -s
------------------------------------------------------------
Server listening on TCP port 5001
TCP window size: 85.3 KByte (default)
------------------------------------------------------------
[  6] local 10.0.0.1 port 5001 connected with 10.0.0.2 port 54204
[ ID] Interval       Transfer     Bandwidth
[  6]  0.0-10.0 sec  28.2 GBytes  24.2 Gbits/sec
```

Task 1: Node: h2 的输出结果为：



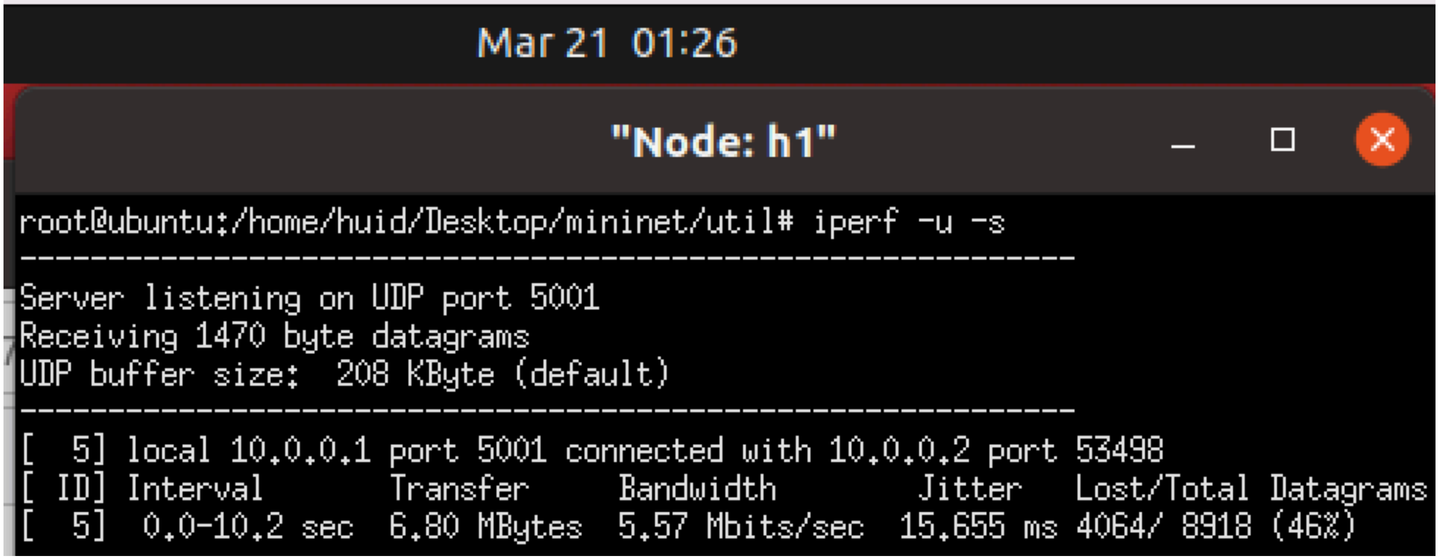
```
root@ubuntu:/home/huid/Desktop/mininet/util# iperf -c 10.0.0.1
------------------------------------------------------------
Client connecting to 10.0.0.1, TCP port 5001
TCP window size:  178 KByte (default)
------------------------------------------------------------
[  5] local 10.0.0.2 port 54204 connected with 10.0.0.1 port 5001
[ ID] Interval       Transfer     Bandwidth
[  5]  0.0-10.0 sec  28.2 GBytes  24.2 Gbits/sec
root@ubuntu:/home/huid/Desktop/mininet/util# 
```

Part2: 三种限速方式

网卡限速

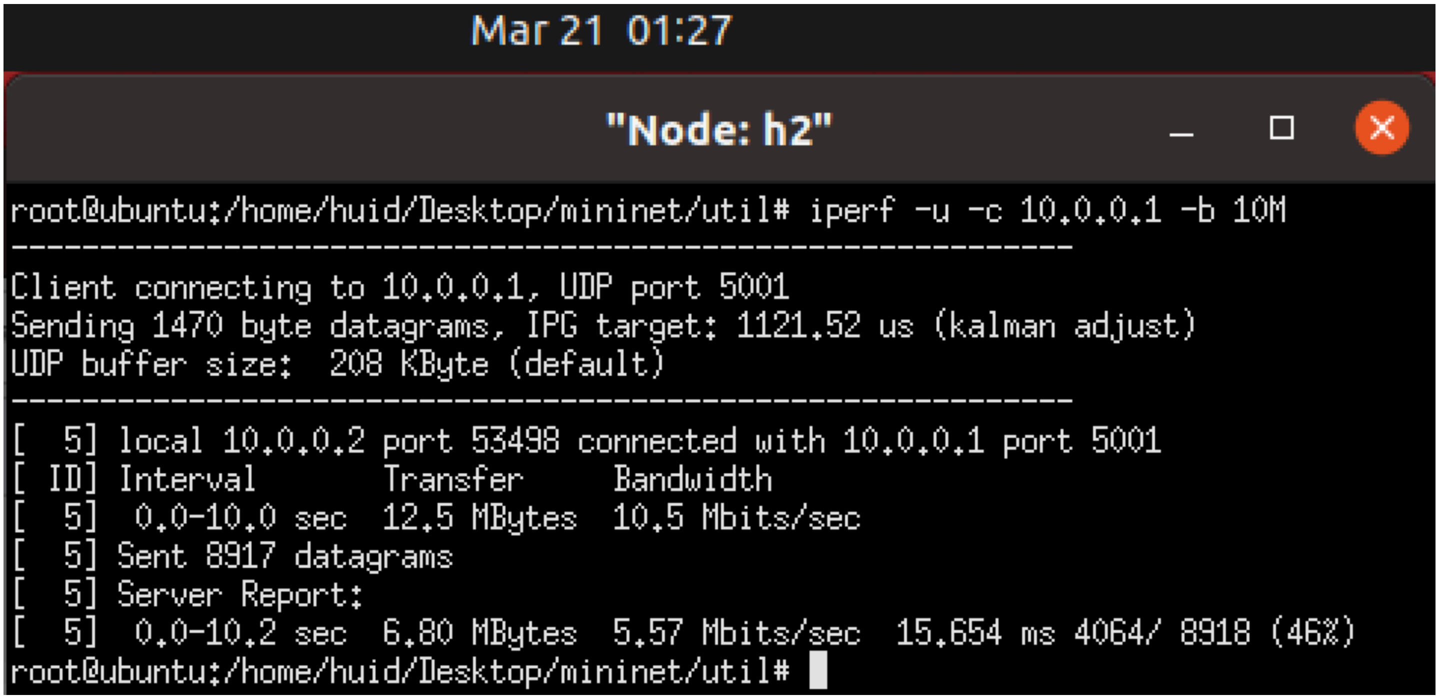
- 带宽：5.57 Mbits/sec
- 抖动：15.655 ms
- 丢包率：46 %

Task 2.1: Node: h1 的输出结果为



```
root@ubuntu:/home/huid/Desktop/mininet/util# iperf -u -s
------------------------------------------------------------
Server listening on UDP port 5001
Receiving 1470 byte datagrams
UDP buffer size:  208 KByte (default)
------------------------------------------------------------
[  5] local 10.0.0.1 port 5001 connected with 10.0.0.2 port 53498
[ ID] Interval       Transfer     Bandwidth        Jitter   Lost/Total Datagrams
[  5]  0.0-10.2 sec  6.80 MBytes  5.57 Mbits/sec  15.655 ms 4064/ 8918 (46%)
```

Task 2.1: Node: h2 的输出结果为

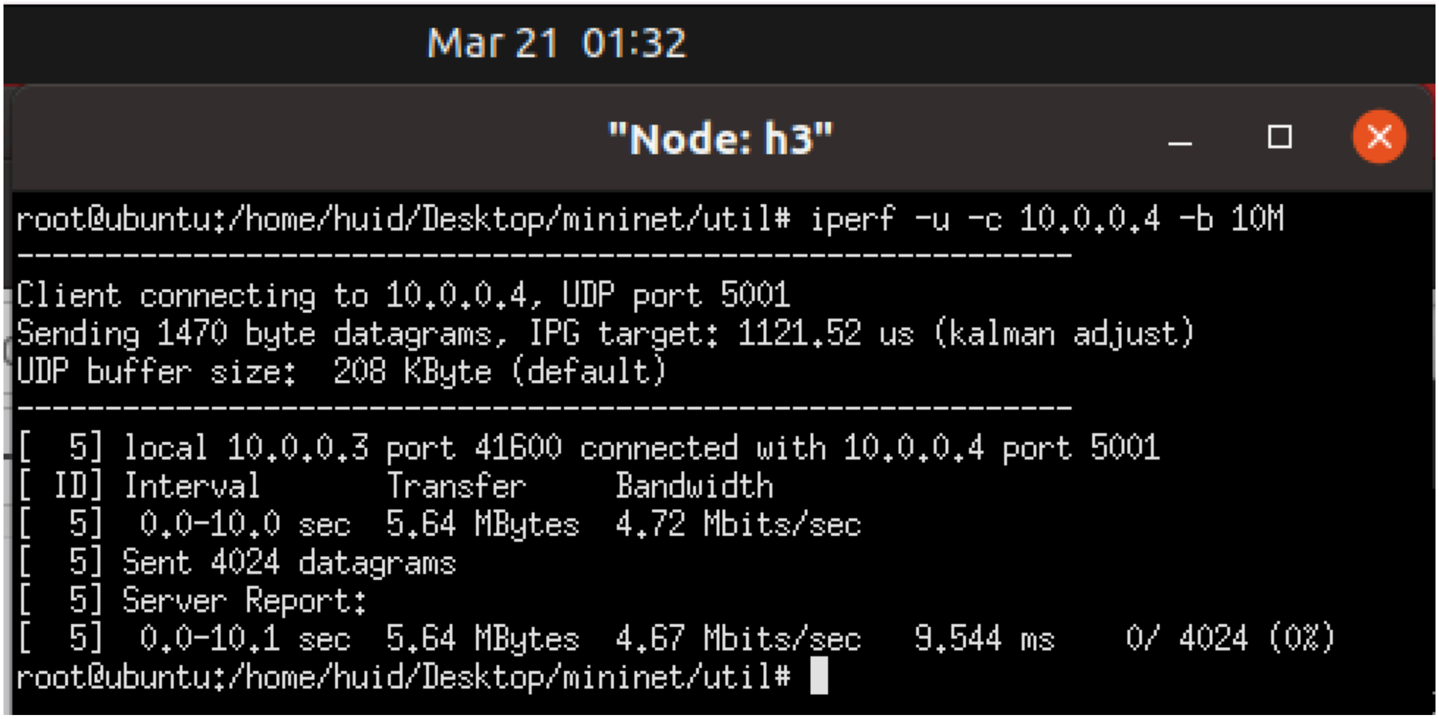


```
root@ubuntu:/home/huid/Desktop/mininet/util# iperf -u -c 10.0.0.1 -b 10M
------------------------------------------------------------
Client connecting to 10.0.0.1, UDP port 5001
Sending 1470 byte datagrams, IPG target: 1121.52 us (kalman adjust)
UDP buffer size:  208 KByte (default)
------------------------------------------------------------
[  5] local 10.0.0.2 port 53498 connected with 10.0.0.1 port 5001
[ ID] Interval       Transfer     Bandwidth
[  5]  0.0-10.0 sec  12.5 MBytes  10.5 Mbits/sec
[  5] Sent 8917 datagrams
[  5] Server Report:
[  5]  0.0-10.2 sec  6.80 MBytes  5.57 Mbits/sec  15.654 ms 4064/ 8918 (46%)
root@ubuntu:/home/huid/Desktop/mininet/util# 
```

队列限速

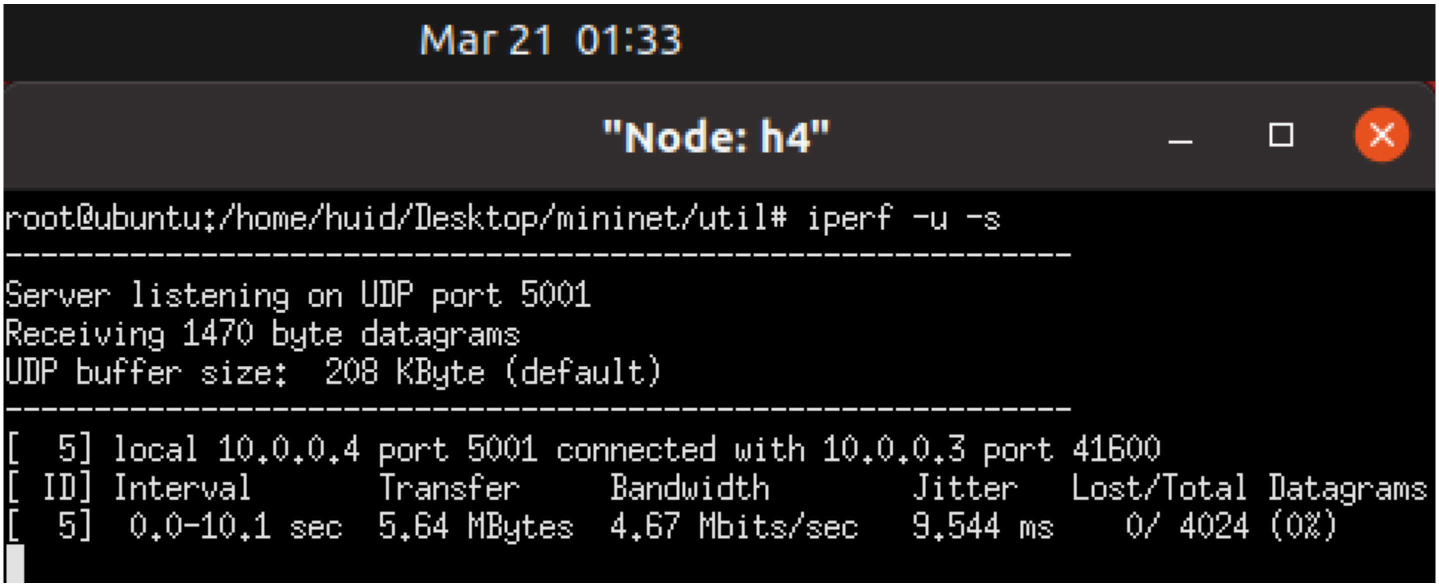
- 带宽：4.67 Mbits/sec
- 抖动：9.544 ms
- 丢包率：0 %

Task 2.2: Node: h3 的输出结果为



```
root@ubuntu:/home/huid/Desktop/mininet/util# iperf -u -c 10.0.0.4 -b 10M
------------------------------------------------------------
Client connecting to 10.0.0.4, UDP port 5001
Sending 1470 byte datagrams, IPG target: 1121.52 us (kalman adjust)
UDP buffer size:  208 KByte (default)
------------------------------------------------------------
[  5] local 10.0.0.3 port 41600 connected with 10.0.0.4 port 5001
[ ID] Interval       Transfer     Bandwidth
[  5]  0.0-10.0 sec  5.64 MBytes  4.72 Mbits/sec
[  5] Sent 4024 datagrams
[  5] Server Report:
[  5]  0.0-10.1 sec  5.64 MBytes  4.67 Mbits/sec   9.544 ms    0/ 4024 (0%)
root@ubuntu:/home/huid/Desktop/mininet/util# 
```

Task 2.2: Node: h4 的输出结果为



```
root@ubuntu:/home/huid/Desktop/mininet/util# iperf -u -s
------------------------------------------------------------
Server listening on UDP port 5001
Receiving 1470 byte datagrams
UDP buffer size:  208 KByte (default)
------------------------------------------------------------
[  5] local 10.0.0.4 port 5001 connected with 10.0.0.3 port 41600
[ ID] Interval       Transfer     Bandwidth        Jitter   Lost/Total Datagrams
[  5]  0.0-10.1 sec  5.64 MBytes  4.67 Mbits/sec   9.544 ms    0/ 4024 (0%)
```

Meter 表限速

- 带宽：5.22 Mbits/sec
- 抖动：15.764 ms
- 丢包率：49 %

Q1: 理解 Line 15, Line 16 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。

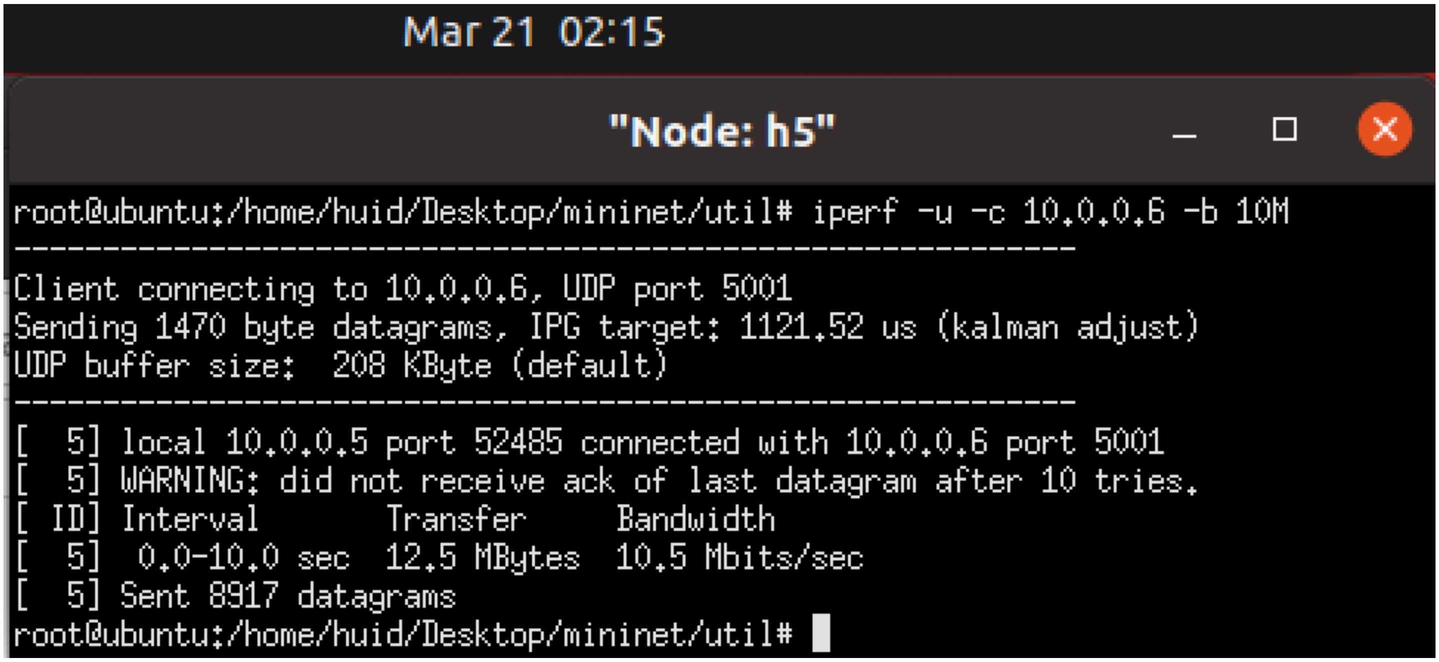
```
$ ovs-ofctl add-flow s1 in_port=5,action=meter:1,output:6 -O openflow13
```

下发转发的流表。`in_port` 匹配进端口为 5，`action` 指明转发动作为 `meter:1, output:6`，表示匹配到的流表首先交给 meter 表处理，超过 5M 的数据包丢弃，然后再交给 `output: 6` 从端口 6 转发出去。`-O` 指明了 OpenFlow 的版本 13。

```
$ ovs-ofctl dump-flows s1 -O openflow13
```

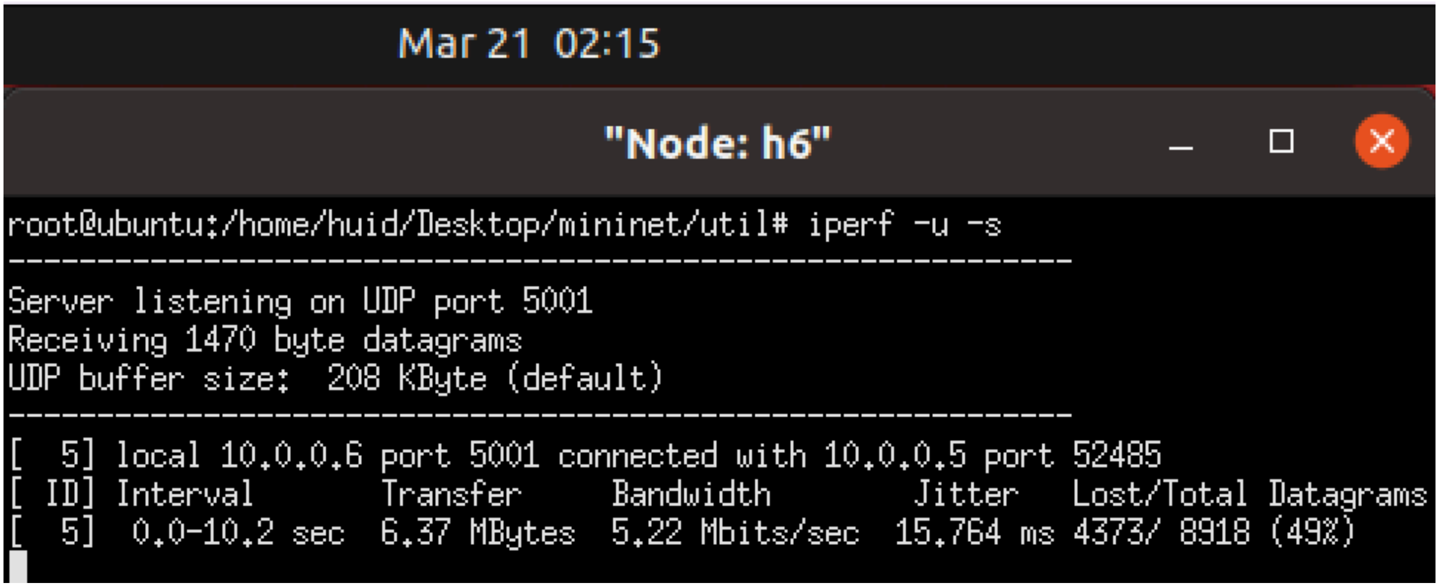
查看交换机中的流表的条目。`dump-flows` 打印出 `s1` 流表的条目。`-O` 指明了 OpenFlow 的版本 13。

Task 2.3: Node: h5 的输出结果为



```
root@ubuntu:/home/huid/Desktop/mininet/util# iperf -u -c 10.0.0.6 -b 10M
------------------------------------------------------------
Client connecting to 10.0.0.6, UDP port 5001
Sending 1470 byte datagrams, IPG target: 1121.52 us (kalman adjust)
UDP buffer size:  208 KByte (default)
------------------------------------------------------------
[  5] local 10.0.0.5 port 52485 connected with 10.0.0.6 port 5001
[  5] WARNING: did not receive ack of last datagram after 10 tries.
[ ID] Interval       Transfer     Bandwidth
[  5]  0.0-10.0 sec  12.5 MBytes  10.5 Mbits/sec
[  5] Sent 8917 datagrams
root@ubuntu:/home/huid/Desktop/mininet/util# 
```

Task 2.3: Node: h6 的输出结果为



```
root@ubuntu:/home/huid/Desktop/mininet/util# iperf -u -s
------------------------------------------------------------
Server listening on UDP port 5001
Receiving 1470 byte datagrams
UDP buffer size:  208 KByte (default)
------------------------------------------------------------
[  5] local 10.0.0.6 port 5001 connected with 10.0.0.5 port 52485
[ ID] Interval       Transfer     Bandwidth        Jitter   Lost/Total Datagrams
[  5]  0.0-10.2 sec  6.37 MBytes  5.22 Mbits/sec  15.764 ms 4373/ 8918 (49%)
```

Q2: 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。

	带宽 Mbits/sec	抖动 ms	丢包率	带宽误差率
网卡限速	5.57	15.655	46 %	11.4 %
队列限速	4.67	9.544	0 %	6.6 %
Meter 表限速	5.22	15.764	49 %	4.4 %

从数据来看，队列限速的表现最好，带宽限制在了 5 Mb 以内，且抖动最小，丢包率也最小。

而网卡限速表现最不好，带宽的误差率最高，抖动也很高，这与网卡限速的实现方式有关，其控制精度比较粗粒度。

Meter 表作为 SDN 模式限速的代表，表现中等，这可能与 ovs 交换的流表控制能力有关。交换机中流表的匹配，数据流计数，动作的执行等都是影响其控制粒度的原因。软件实现的交换机对流表的控制比不上硬件交换机，如果使用硬件交换机其效率表现应该有所提高。

Part3: 拓展与应用

我们首先先清理之前 part 遗留下来的限速队列。

```
$ ovs-vsctl clear port s1-eth1 qos
$ ovs-vsctl clear port s1-eth2 qos
$ ovs-vsctl clear port s1-eth3 qos
$ ovs-vsctl clear port s1-eth4 qos
```

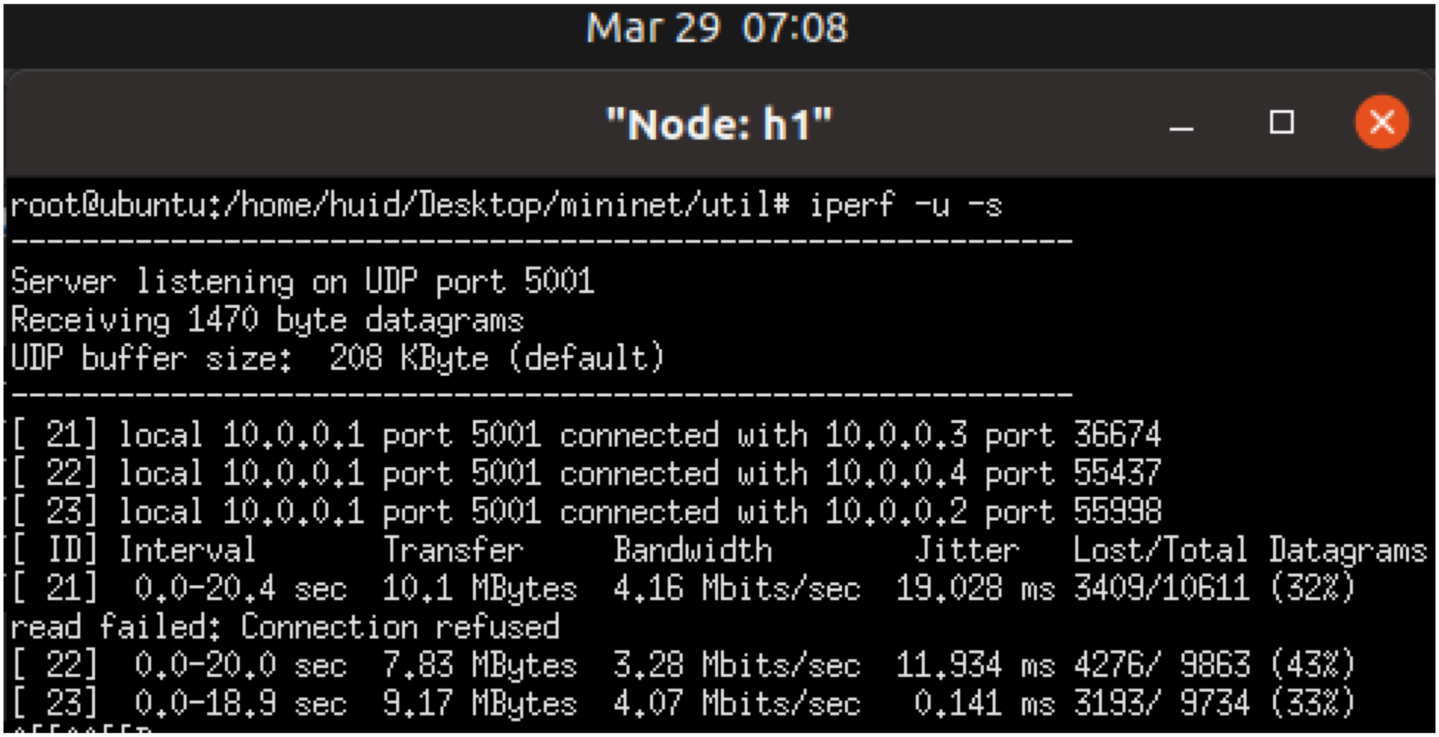
我们可以用如下命令清理所有限速队列并检查是否清理干净。

```
$ ovs-vsctl -- --all destroy qos -- --all destroy queue
$ ovs-vsctl list qos
$ ovs-vsctl list queue
```

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。

使用队列限速对 Server 端控制带宽为 10 Mb。

```
$ ovs-vsctl set port s1-eth4 qos=@newqos -- --id=@newqos create qos type=linux-htb
queues=0=@q0 -- --id=@q0 create queue other-config:max-rate=5000000
$ h1 iperf -u -s
```

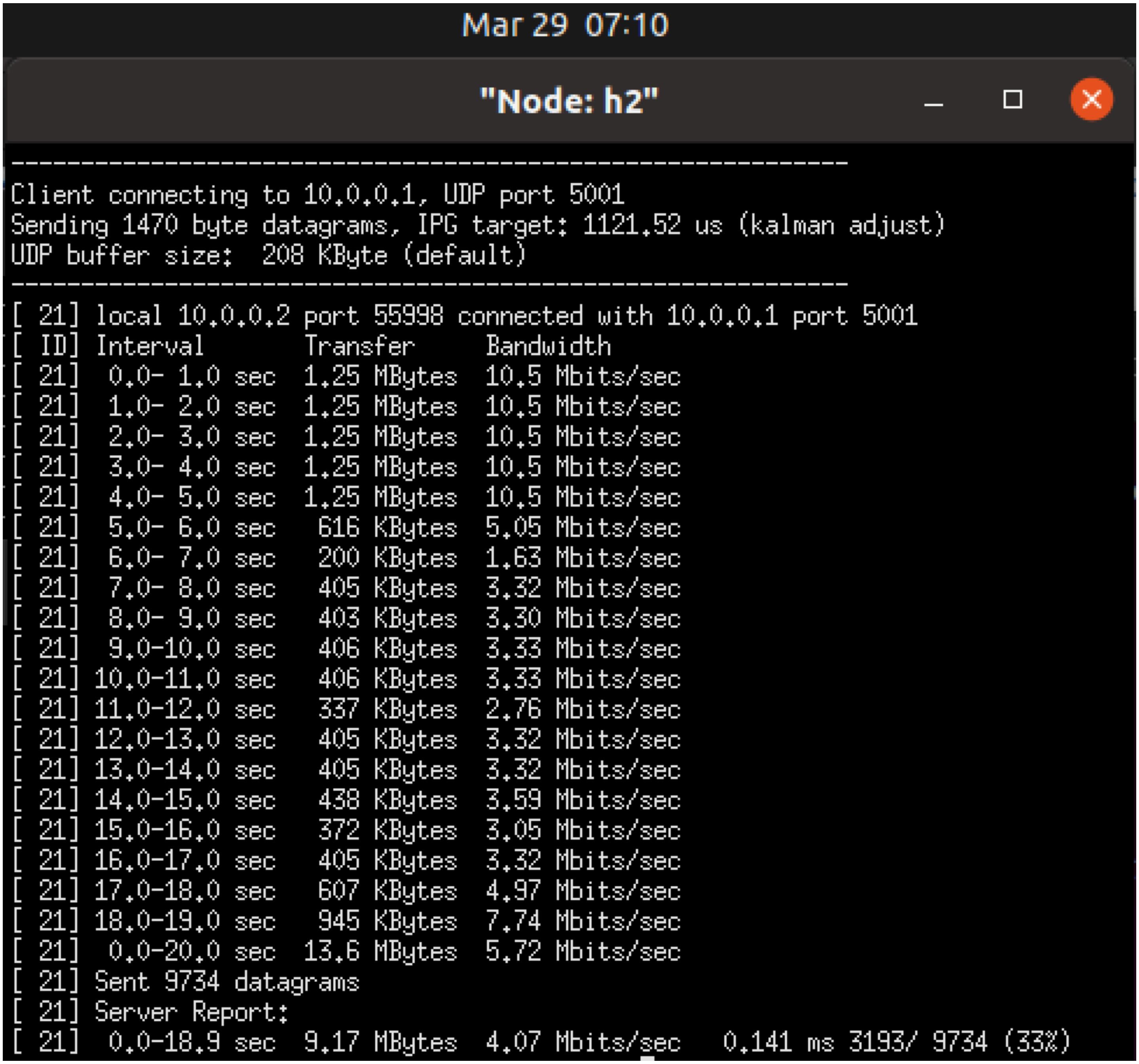


```
root@ubuntu:/home/huid/Desktop/mininet/util# iperf -u -s
------------------------------------------------------------
Server listening on UDP port 5001
Receiving 1470 byte datagrams
UDP buffer size:  208 KByte (default)
------------------------------------------------------------
[ 21] local 10.0.0.1 port 5001 connected with 10.0.0.3 port 36674
[ 22] local 10.0.0.1 port 5001 connected with 10.0.0.4 port 55437
[ 23] local 10.0.0.1 port 5001 connected with 10.0.0.2 port 55998
[ ID] Interval       Transfer     Bandwidth        Jitter   Lost/Total Datagrams
[ 21]  0.0-20.4 sec  10.1 MBytes  4.16 Mbits/sec  19.028 ms 3409/10611 (32%)
read failed: Connection refused
[ 22]  0.0-20.0 sec  7.83 MBytes  3.28 Mbits/sec  11.934 ms 4276/ 9863 (43%)
[ 23]  0.0-18.9 sec  9.17 MBytes  4.07 Mbits/sec   0.141 ms 3193/ 9734 (33%)
```

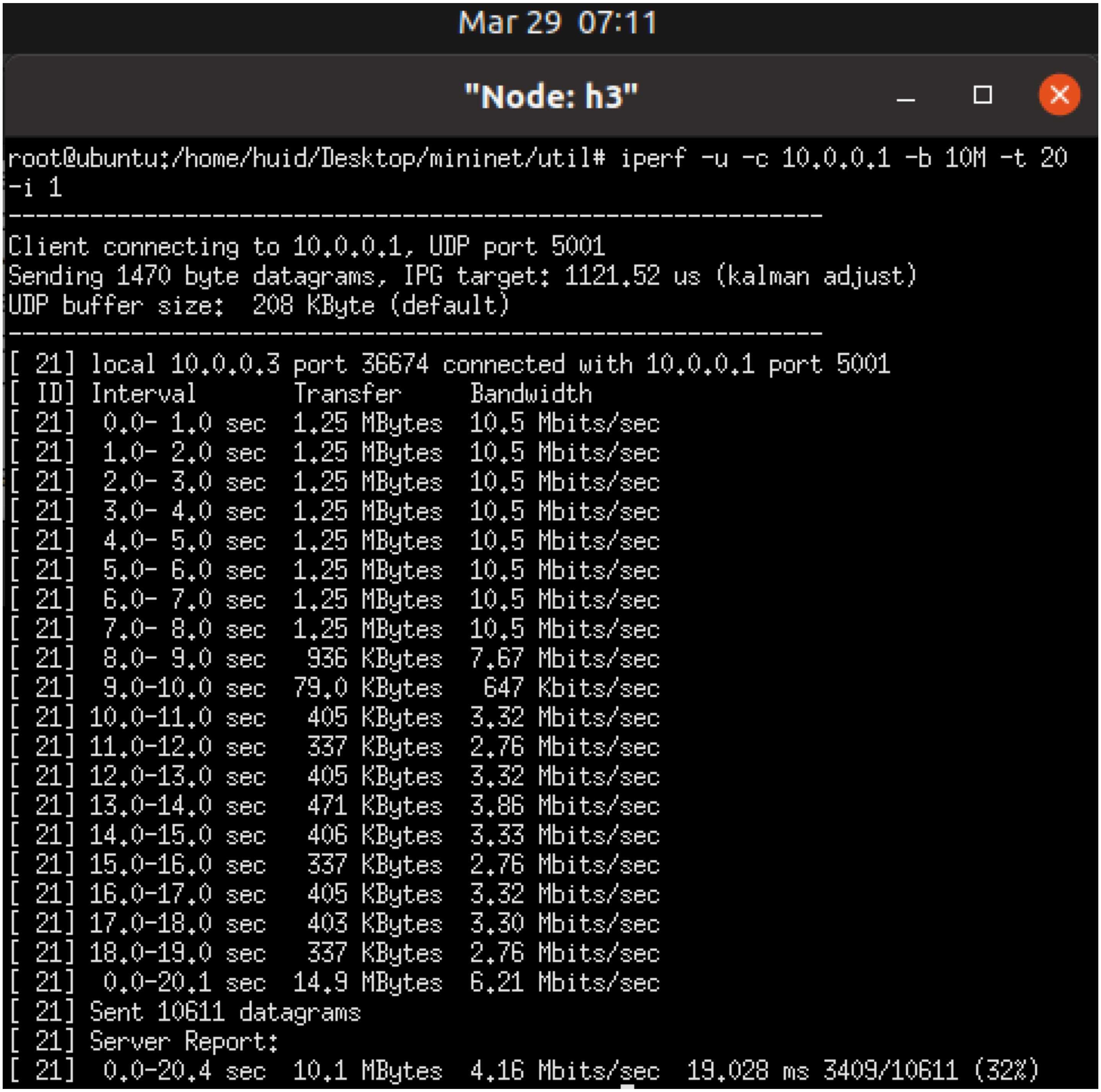
然后 "手动同时" 运行三个 Client。

```
$ h2 iperf -u -c 10.0.0.1 -b 10M -t 20 -i 1
$ h3 iperf -u -c 10.0.0.1 -b 10M -t 20 -i 1
$ h4 iperf -u -c 10.0.0.1 -b 10M -t 20 -i 1
```

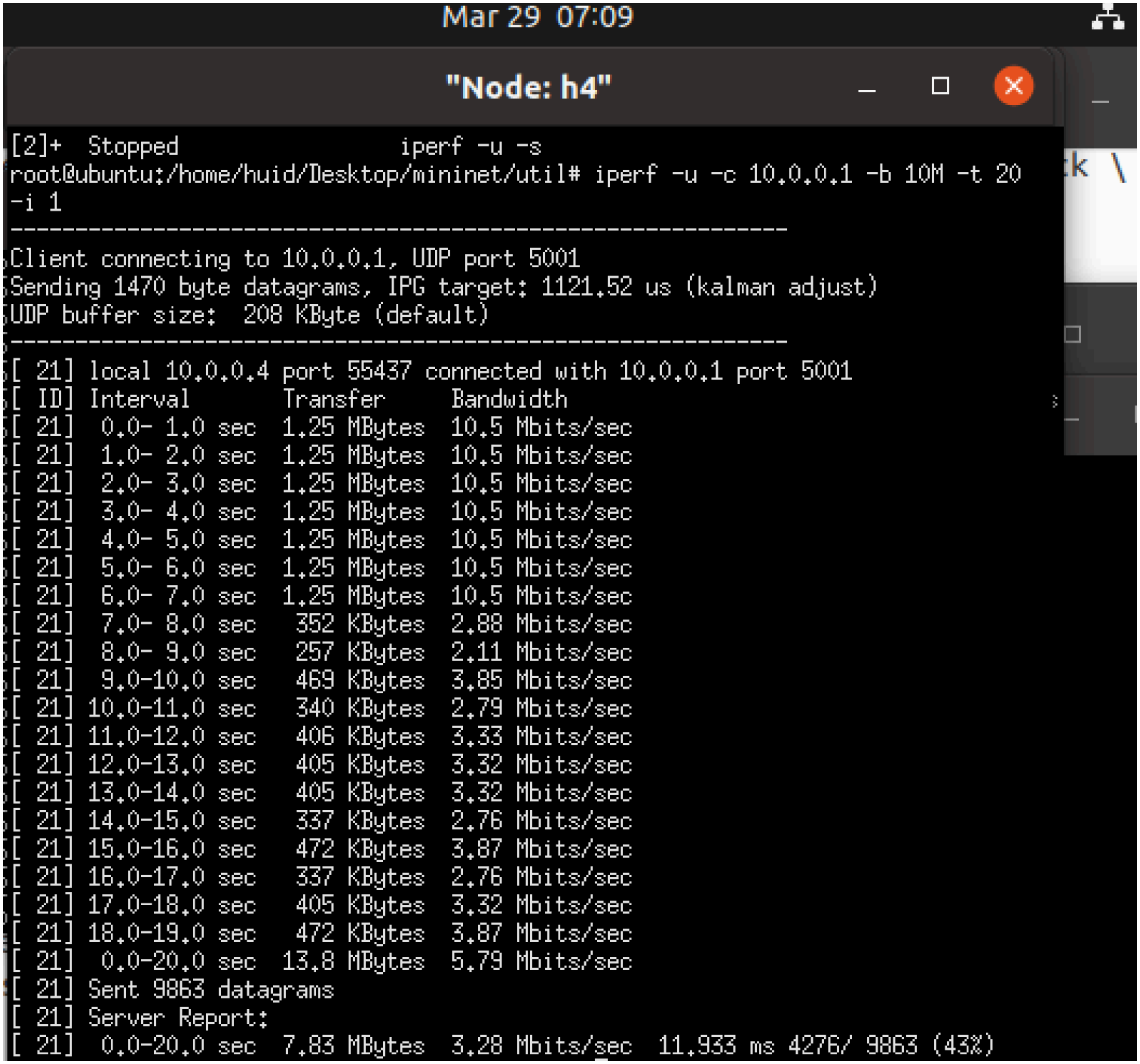
并观察稳定后（即三个 Client 并行时）的带宽。可以发现，三个 Client 几乎均分了 Server 的 10 M 带宽。



```
------------------------------------------------------------
Client connecting to 10.0.0.1, UDP port 5001
Sending 1470 byte datagrams, IPG target: 1121.52 us (kalman adjust)
UDP buffer size:  208 KByte (default)
------------------------------------------------------------
[ 21] local 10.0.0.2 port 55998 connected with 10.0.0.1 port 5001
[ ID] Interval       Transfer     Bandwidth
[ 21]  0.0- 1.0 sec  1.25 MBytes  10.5 Mbits/sec
[ 21]  1.0- 2.0 sec  1.25 MBytes  10.5 Mbits/sec
[ 21]  2.0- 3.0 sec  1.25 MBytes  10.5 Mbits/sec
[ 21]  3.0- 4.0 sec  1.25 MBytes  10.5 Mbits/sec
[ 21]  4.0- 5.0 sec  1.25 MBytes  10.5 Mbits/sec
[ 21]  5.0- 6.0 sec   616 KBytes  5.05 Mbits/sec
[ 21]  6.0- 7.0 sec   200 KBytes  1.63 Mbits/sec
[ 21]  7.0- 8.0 sec   405 KBytes  3.32 Mbits/sec
[ 21]  8.0- 9.0 sec   403 KBytes  3.30 Mbits/sec
[ 21]  9.0-10.0 sec   406 KBytes  3.33 Mbits/sec
[ 21] 10.0-11.0 sec   406 KBytes  3.33 Mbits/sec
[ 21] 11.0-12.0 sec   337 KBytes  2.76 Mbits/sec
[ 21] 12.0-13.0 sec   405 KBytes  3.32 Mbits/sec
[ 21] 13.0-14.0 sec   405 KBytes  3.32 Mbits/sec
[ 21] 14.0-15.0 sec   438 KBytes  3.59 Mbits/sec
[ 21] 15.0-16.0 sec   372 KBytes  3.05 Mbits/sec
[ 21] 16.0-17.0 sec   405 KBytes  3.32 Mbits/sec
[ 21] 17.0-18.0 sec   607 KBytes  4.97 Mbits/sec
[ 21] 18.0-19.0 sec   945 KBytes  7.74 Mbits/sec
[ 21]  0.0-20.0 sec  13.6 MBytes  5.72 Mbits/sec
[ 21] Sent 9734 datagrams
[ 21] Server Report:
[ 21]  0.0-18.9 sec  9.17 MBytes  4.07 Mbits/sec   0.141 ms 3193/ 9734 (33%)
```



```
root@ubuntu:/home/huid/Desktop/mininet/util# iperf -u -c 10.0.0.1 -b 10M -t 20
-i 1
------------------------------------------------------------
Client connecting to 10.0.0.1, UDP port 5001
Sending 1470 byte datagrams, IPG target: 1121.52 us (kalman adjust)
UDP buffer size:  208 KByte (default)
------------------------------------------------------------
[ 21] local 10.0.0.3 port 36674 connected with 10.0.0.1 port 5001
[ ID] Interval       Transfer     Bandwidth
[ 21]  0.0- 1.0 sec  1.25 MBytes  10.5 Mbits/sec
[ 21]  1.0- 2.0 sec  1.25 MBytes  10.5 Mbits/sec
[ 21]  2.0- 3.0 sec  1.25 MBytes  10.5 Mbits/sec
[ 21]  3.0- 4.0 sec  1.25 MBytes  10.5 Mbits/sec
[ 21]  4.0- 5.0 sec  1.25 MBytes  10.5 Mbits/sec
[ 21]  5.0- 6.0 sec  1.25 MBytes  10.5 Mbits/sec
[ 21]  6.0- 7.0 sec  1.25 MBytes  10.5 Mbits/sec
[ 21]  7.0- 8.0 sec  1.25 MBytes  10.5 Mbits/sec
[ 21]  8.0- 9.0 sec   936 KBytes  7.67 Mbits/sec
[ 21]  9.0-10.0 sec  79.0 KBytes   647 Kbits/sec
[ 21] 10.0-11.0 sec   405 KBytes  3.32 Mbits/sec
[ 21] 11.0-12.0 sec   337 KBytes  2.76 Mbits/sec
[ 21] 12.0-13.0 sec   405 KBytes  3.32 Mbits/sec
[ 21] 13.0-14.0 sec   471 KBytes  3.86 Mbits/sec
[ 21] 14.0-15.0 sec   406 KBytes  3.33 Mbits/sec
[ 21] 15.0-16.0 sec   337 KBytes  2.76 Mbits/sec
[ 21] 16.0-17.0 sec   405 KBytes  3.32 Mbits/sec
[ 21] 17.0-18.0 sec   403 KBytes  3.30 Mbits/sec
[ 21] 18.0-19.0 sec   337 KBytes  2.76 Mbits/sec
[ 21]  0.0-20.1 sec  14.9 MBytes  6.21 Mbits/sec
[ 21] Sent 10611 datagrams
[ 21] Server Report:
[ 21]  0.0-20.4 sec  10.1 MBytes  4.16 Mbits/sec  19.028 ms 3409/10611 (32%)
```



```
[2]+  Stopped                 iperf -u -s
root@ubuntu:/home/huid/Desktop/mininet/util# iperf -u -c 10.0.0.1 -b 10M -t 20
-i 1
------------------------------------------------------------
Client connecting to 10.0.0.1, UDP port 5001
Sending 1470 byte datagrams, IPG target: 1121.52 us (kalman adjust)
UDP buffer size:  208 KByte (default)
------------------------------------------------------------
[ 21] local 10.0.0.4 port 55437 connected with 10.0.0.1 port 5001
[ ID] Interval       Transfer     Bandwidth
[ 21]  0.0- 1.0 sec  1.25 MBytes  10.5 Mbits/sec
[ 21]  1.0- 2.0 sec  1.25 MBytes  10.5 Mbits/sec
[ 21]  2.0- 3.0 sec  1.25 MBytes  10.5 Mbits/sec
[ 21]  3.0- 4.0 sec  1.25 MBytes  10.5 Mbits/sec
[ 21]  4.0- 5.0 sec  1.25 MBytes  10.5 Mbits/sec
[ 21]  5.0- 6.0 sec  1.25 MBytes  10.5 Mbits/sec
[ 21]  6.0- 7.0 sec  1.25 MBytes  10.5 Mbits/sec
[ 21]  7.0- 8.0 sec   352 KBytes  2.88 Mbits/sec
[ 21]  8.0- 9.0 sec   257 KBytes  2.11 Mbits/sec
[ 21]  9.0-10.0 sec   469 KBytes  3.85 Mbits/sec
[ 21] 10.0-11.0 sec   340 KBytes  2.79 Mbits/sec
[ 21] 11.0-12.0 sec   406 KBytes  3.33 Mbits/sec
[ 21] 12.0-13.0 sec   405 KBytes  3.32 Mbits/sec
[ 21] 13.0-14.0 sec   405 KBytes  3.32 Mbits/sec
[ 21] 14.0-15.0 sec   337 KBytes  2.76 Mbits/sec
[ 21] 15.0-16.0 sec   472 KBytes  3.87 Mbits/sec
[ 21] 16.0-17.0 sec   337 KBytes  2.76 Mbits/sec
[ 21] 17.0-18.0 sec   405 KBytes  3.32 Mbits/sec
[ 21] 18.0-19.0 sec   472 KBytes  3.87 Mbits/sec
[ 21]  0.0-20.0 sec  13.8 MBytes  5.79 Mbits/sec
[ 21] Sent 9863 datagrams
[ 21] Server Report:
[ 21]  0.0-20.0 sec  7.83 MBytes  3.28 Mbits/sec  11.933 ms 4276/ 9863 (43%)
```

Task4: 通过上述三种限速的方式来达成目标，记录设计过程（思路及运行指令），并将稳定后的三个 Client 的带宽结果截图。

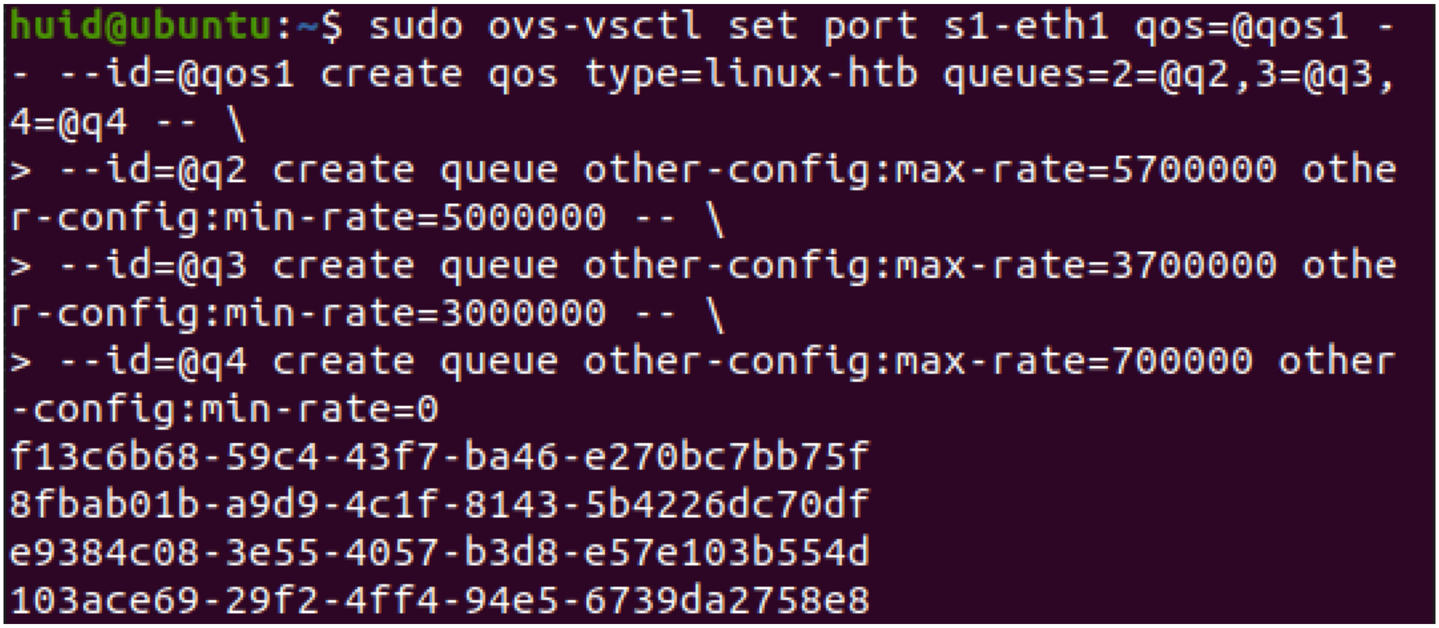
考虑到网卡限速并不能细粒度限制不同 Client 的带宽，因此网卡限速不适合此场景。

所以，我们使用 meter 表限速结合队列限速来控制三个 Client 的带宽。我们限制 Server 端（`h1`）总带宽为 10Mb，并且根据三个 Client 的要求分别限制。

- `h2` 优先级高：5 Mb 及以上
- `h3` 优先级中等：3 Mb 及以上
- `h4` 优先级低：在保证 `h2` 和 `h3` 的前提下尽量多。

因此我们对可以在 Server 端限制三个队列，分别对应于三个 Client。每个队列根据 Client 要求分别设置带宽下限 `min-rate` 分别为 5 Mb、3 Mb 和 0 Mb。而还有 2 Mb 带宽我将其均分给三个 Client，因此设置带宽上限 `max-rate` 分别为 5.7 Mb、3.7 Mb 和 0.7 Mb。

```
$ sudo ovs-vsctl set port s1-eth1 qos=@qos1 -- --id=@qos1 create qos type=linux-htb
queues=2=@q2,3=@q3,4=@q4 -- \
--id=@q2 create queue other-config:max-rate=5700000 other-config:min-rate=5000000 -- \
--id=@q3 create queue other-config:max-rate=3700000 other-config:min-rate=3000000 -- \
--id=@q4 create queue other-config:max-rate=700000 other-config:min-rate=0
```

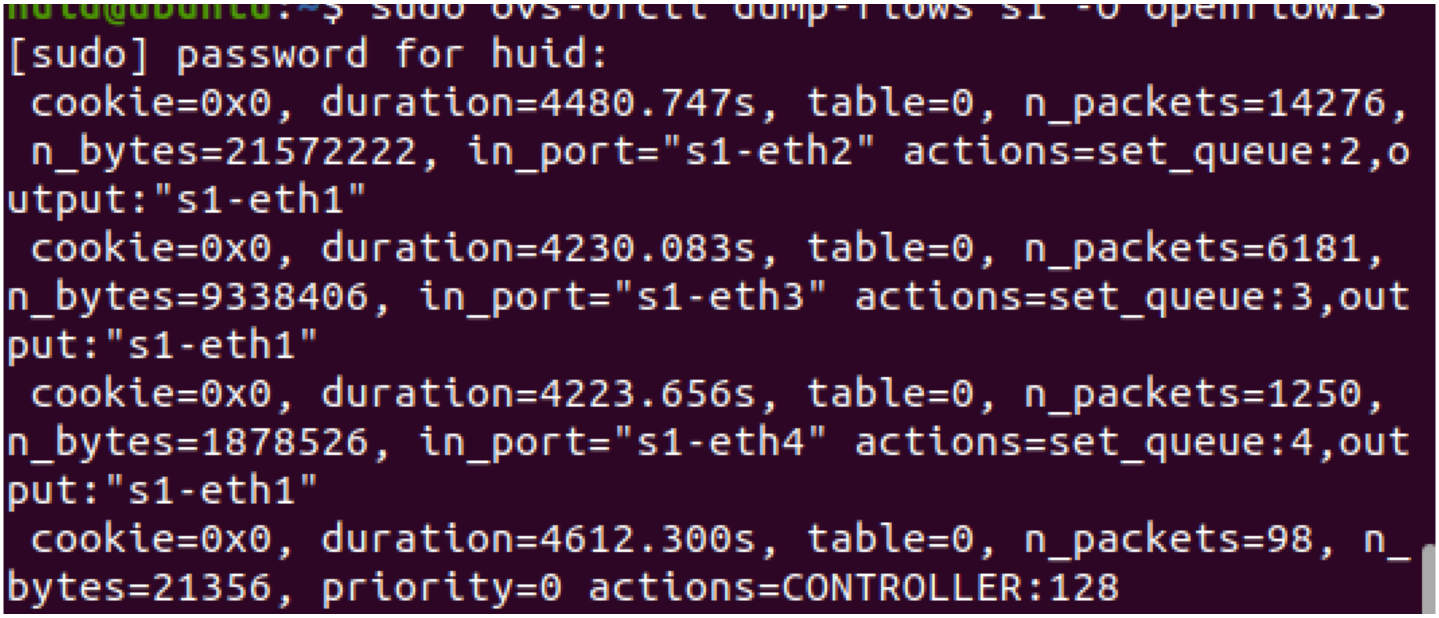


```
huid@ubuntu:~$ sudo ovs-vsctl set port s1-eth1 qos=@qos1 -
- --id=@qos1 create qos type=linux-htb queues=2=@q2,3=@q3,
4=@q4 -- \
> --id=@q2 create queue other-config:max-rate=5700000 othe
r-config:min-rate=5000000 -- \
> --id=@q3 create queue other-config:max-rate=3700000 othe
r-config:min-rate=3000000 -- \
> --id=@q4 create queue other-config:max-rate=700000 other
-config:min-rate=0
f13c6b68-59c4-43f7-ba46-e270bc7bb75f
8fbab01b-a9d9-4c1f-8143-5b4226dc70df
e9384c08-3e55-4057-b3d8-e57e103b554d
103ace69-29f2-4ff4-94e5-6739da2758e8
```

然后我们下发流表并且分别指定不同队列给不同 Client，并查看。

```
$ 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
$ ovs-ofctl dump-flows s1 -O openflow13
```

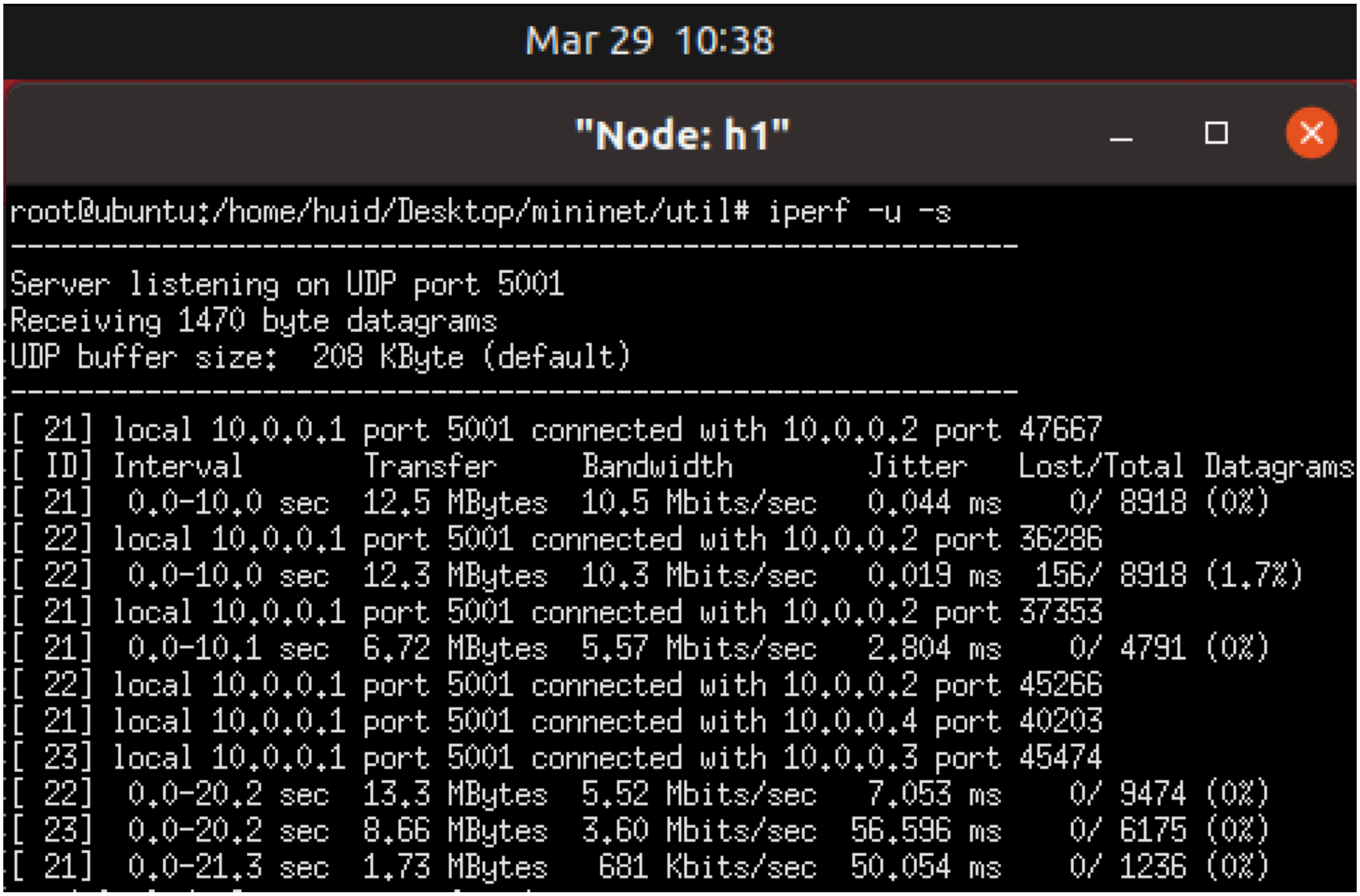
```
huid@ubuntu:~$ sudo ovs-ofctl dump-flows s1 -O openflow13
```



```
huid@ubuntu:~$ sudo ovs-ofctl dump-flows s1 -O openflow13
[sudo] password for huid:
 cookie=0x0, duration=4480.747s, table=0, n_packets=14276,
 n_bytes=21572222, in_port="s1-eth2" actions=set_queue:2,o
utput:"s1-eth1"
 cookie=0x0, duration=4230.083s, table=0, n_packets=6181,
n_bytes=9338406, in_port="s1-eth3" actions=set_queue:3,out
put:"s1-eth1"
 cookie=0x0, duration=4223.656s, table=0, n_packets=1250,
n_bytes=1878526, in_port="s1-eth4" actions=set_queue:4,out
put:"s1-eth1"
 cookie=0x0, duration=4612.300s, table=0, n_packets=98, n_
bytes=21356, priority=0 actions=CONTROLLER:128
```

在 Server 端（`h1`）运行命令

```
$ iperf -u -s
```



Mar 29 10:38

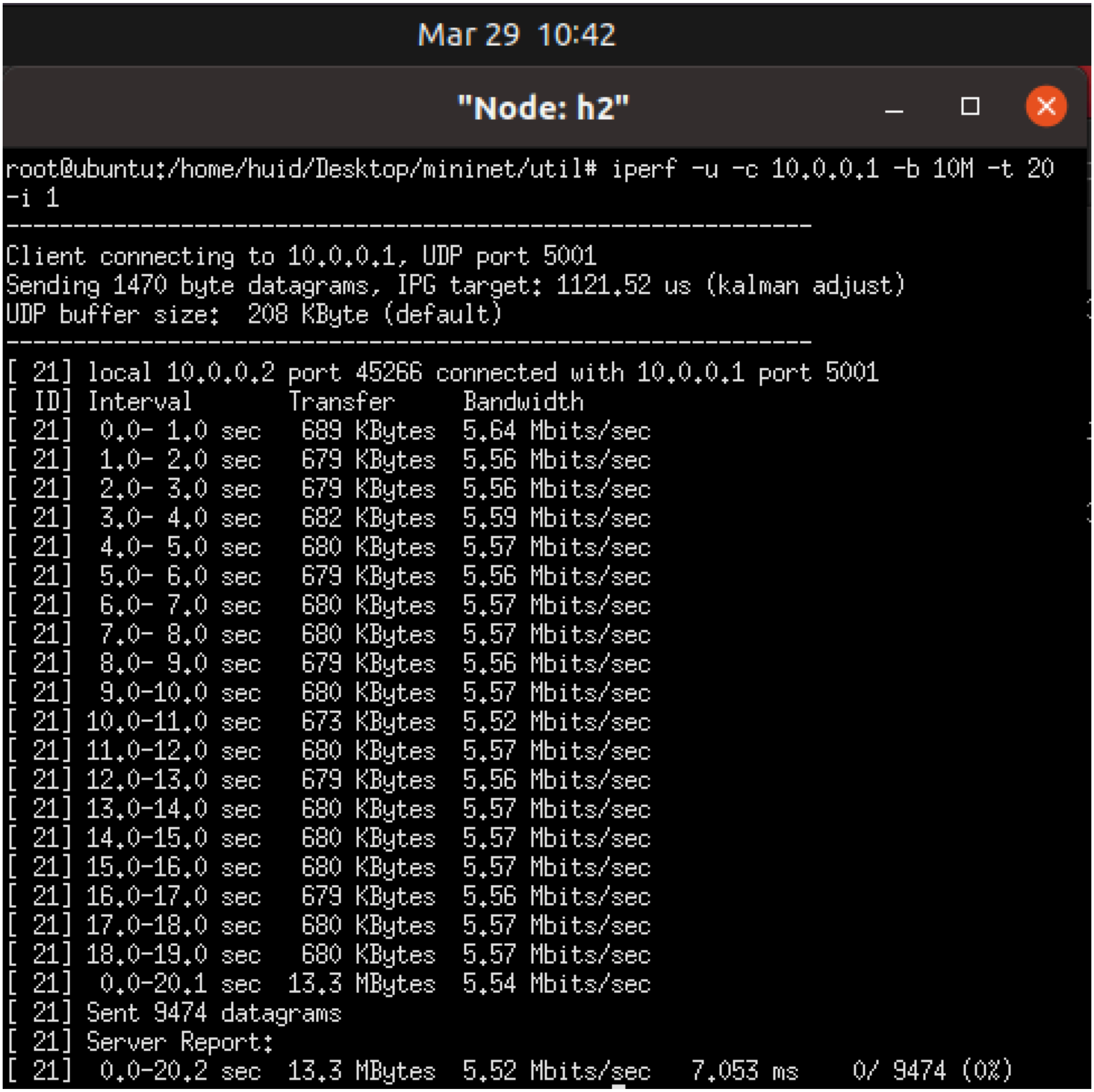
"Node: h1"

```
root@ubuntu:/home/huid/Desktop/mininet/util# iperf -u -s
------------------------------------------------------------
Server listening on UDP port 5001
Receiving 1470 byte datagrams
UDP buffer size:  208 KByte (default)
------------------------------------------------------------
[ 21] local 10.0.0.1 port 5001 connected with 10.0.0.2 port 47667
[ ID] Interval       Transfer     Bandwidth        Jitter   Lost/Total Datagrams
[ 21]  0.0-10.0 sec  12.5 MBytes  10.5 Mbits/sec   0.044 ms    0/ 8918 (0%)
[ 22] local 10.0.0.1 port 5001 connected with 10.0.0.2 port 36286
[ 22]  0.0-10.0 sec  12.3 MBytes  10.3 Mbits/sec   0.019 ms  156/ 8918 (1.7%)
[ 21] local 10.0.0.1 port 5001 connected with 10.0.0.2 port 37353
[ 21]  0.0-10.1 sec  6.72 MBytes  5.57 Mbits/sec   2.804 ms    0/ 4791 (0%)
[ 22] local 10.0.0.1 port 5001 connected with 10.0.0.2 port 45266
[ 21] local 10.0.0.1 port 5001 connected with 10.0.0.4 port 40203
[ 23] local 10.0.0.1 port 5001 connected with 10.0.0.3 port 45474
[ 22]  0.0-20.2 sec  13.3 MBytes  5.52 Mbits/sec   7.053 ms    0/ 9474 (0%)
[ 23]  0.0-20.2 sec  8.66 MBytes  3.60 Mbits/sec  56.596 ms    0/ 6175 (0%)
[ 21]  0.0-21.3 sec  1.73 MBytes   681 Kbits/sec  50.054 ms    0/ 1236 (0%)
```

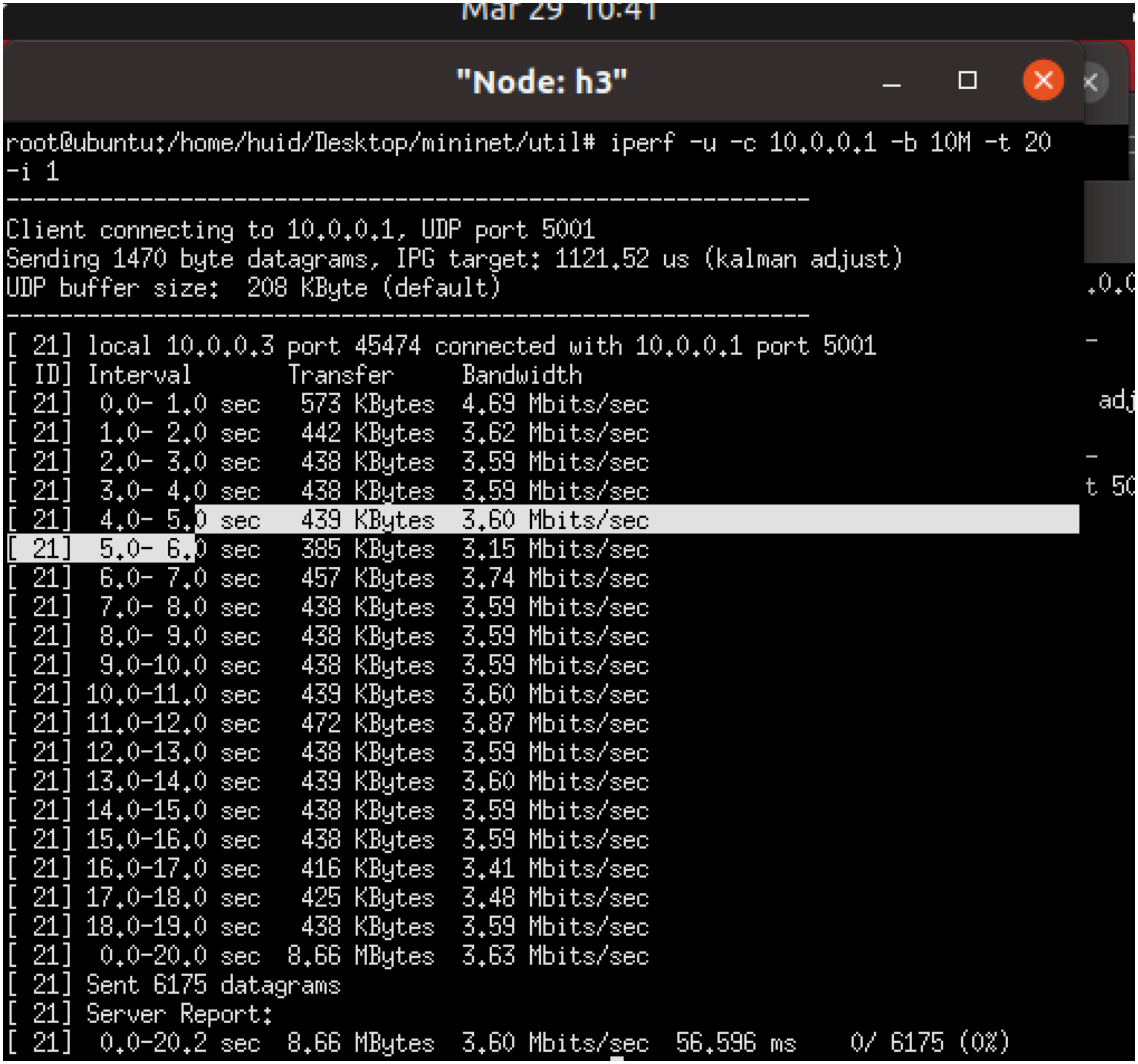
在三个 Client 端（`h2`、`h3`、`h4`）分别同时运行命令

```
$ iperf -u -c 10.0.0.1 -b 10M -t 20 -i 1
```

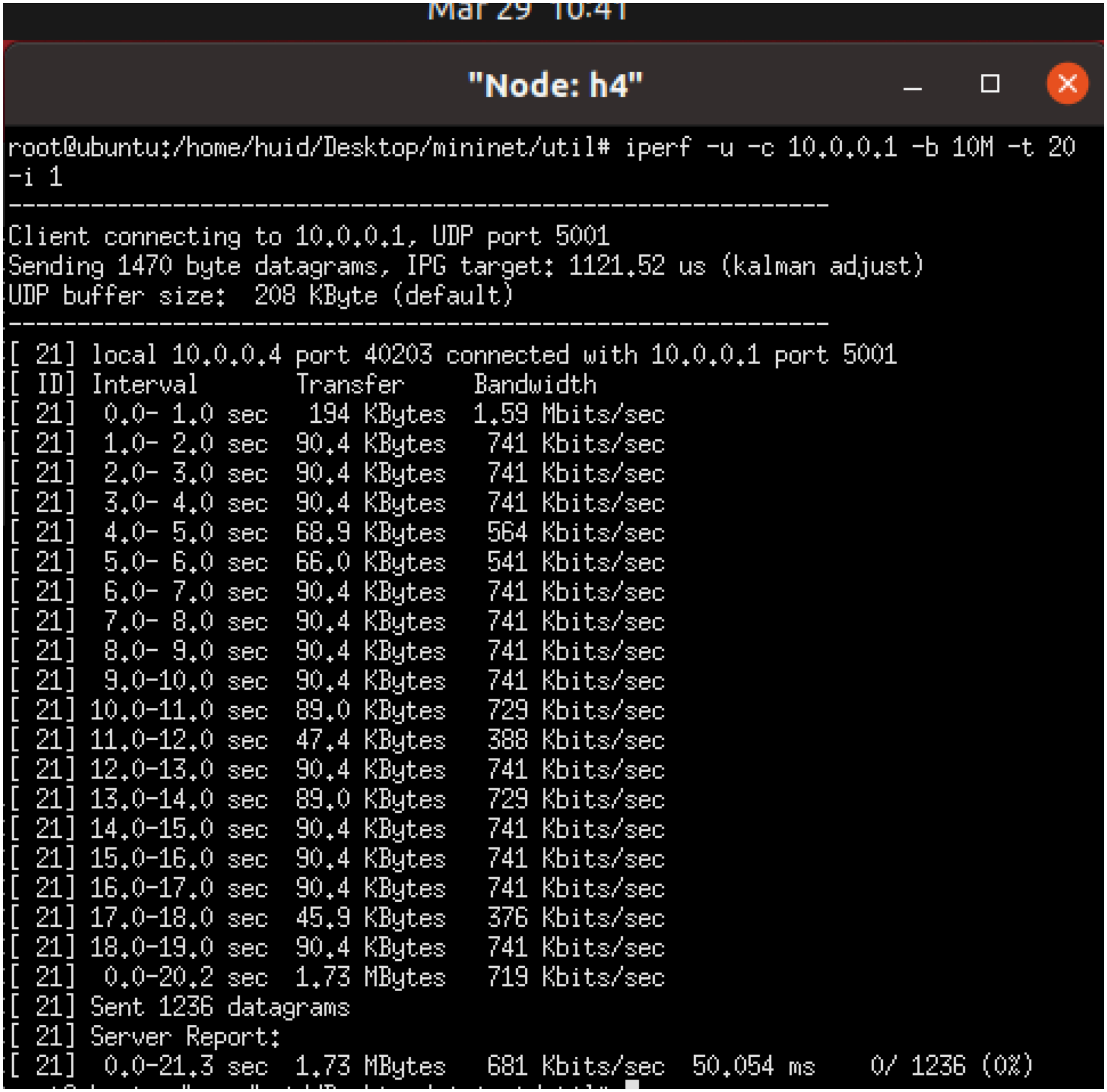
可以看出，Client `h2` 带宽稳定在 5.57 Mb 左右，Client `h3` 带宽稳定在 3.6 Mb 左右，Client `h4` 带宽稳定在 0.7 Mb 左右，符合题目要求且几乎用尽了 Server `h1` 的全部带宽。



```
root@ubuntu:/home/huid/Desktop/mininet/util# iperf -u -c 10.0.0.1 -b 10M -t 20
-i 1
------------------------------------------------------------
Client connecting to 10.0.0.1, UDP port 5001
Sending 1470 byte datagrams, IPG target: 1121.52 us (kalman adjust)
UDP buffer size:  208 KByte (default)
------------------------------------------------------------
[ 21] local 10.0.0.2 port 45266 connected with 10.0.0.1 port 5001
[ ID] Interval       Transfer     Bandwidth
[ 21]  0.0- 1.0 sec   689 KBytes  5.64 Mbits/sec
[ 21]  1.0- 2.0 sec   679 KBytes  5.56 Mbits/sec
[ 21]  2.0- 3.0 sec   679 KBytes  5.56 Mbits/sec
[ 21]  3.0- 4.0 sec   682 KBytes  5.59 Mbits/sec
[ 21]  4.0- 5.0 sec   680 KBytes  5.57 Mbits/sec
[ 21]  5.0- 6.0 sec   679 KBytes  5.56 Mbits/sec
[ 21]  6.0- 7.0 sec   680 KBytes  5.57 Mbits/sec
[ 21]  7.0- 8.0 sec   680 KBytes  5.57 Mbits/sec
[ 21]  8.0- 9.0 sec   679 KBytes  5.56 Mbits/sec
[ 21]  9.0-10.0 sec   680 KBytes  5.57 Mbits/sec
[ 21] 10.0-11.0 sec   673 KBytes  5.52 Mbits/sec
[ 21] 11.0-12.0 sec   680 KBytes  5.57 Mbits/sec
[ 21] 12.0-13.0 sec   679 KBytes  5.56 Mbits/sec
[ 21] 13.0-14.0 sec   680 KBytes  5.57 Mbits/sec
[ 21] 14.0-15.0 sec   680 KBytes  5.57 Mbits/sec
[ 21] 15.0-16.0 sec   680 KBytes  5.57 Mbits/sec
[ 21] 16.0-17.0 sec   679 KBytes  5.56 Mbits/sec
[ 21] 17.0-18.0 sec   680 KBytes  5.57 Mbits/sec
[ 21] 18.0-19.0 sec   680 KBytes  5.57 Mbits/sec
[ 21]  0.0-20.1 sec  13.3 MBytes  5.54 Mbits/sec
[ 21] Sent 9474 datagrams
[ 21] Server Report:
[ 21]  0.0-20.2 sec  13.3 MBytes  5.52 Mbits/sec   7.053 ms    0/ 9474 (0%)
```



```
root@ubuntu:/home/huid/Desktop/mininet/util# iperf -u -c 10.0.0.1 -b 10M -t 20
-i 1
------------------------------------------------------------
Client connecting to 10.0.0.1, UDP port 5001
Sending 1470 byte datagrams, IPG target: 1121.52 us (kalman adjust)
UDP buffer size:  208 KByte (default)
------------------------------------------------------------
[ 21] local 10.0.0.3 port 45474 connected with 10.0.0.1 port 5001
[ ID] Interval       Transfer     Bandwidth
[ 21]  0.0- 1.0 sec   573 KBytes  4.69 Mbits/sec
[ 21]  1.0- 2.0 sec   442 KBytes  3.62 Mbits/sec
[ 21]  2.0- 3.0 sec   438 KBytes  3.59 Mbits/sec
[ 21]  3.0- 4.0 sec   438 KBytes  3.59 Mbits/sec
[ 21]  4.0- 5.0 sec   439 KBytes  3.60 Mbits/sec
[ 21]  5.0- 6.0 sec   385 KBytes  3.15 Mbits/sec
[ 21]  6.0- 7.0 sec   457 KBytes  3.74 Mbits/sec
[ 21]  7.0- 8.0 sec   438 KBytes  3.59 Mbits/sec
[ 21]  8.0- 9.0 sec   438 KBytes  3.59 Mbits/sec
[ 21]  9.0-10.0 sec   438 KBytes  3.59 Mbits/sec
[ 21] 10.0-11.0 sec   439 KBytes  3.60 Mbits/sec
[ 21] 11.0-12.0 sec   472 KBytes  3.87 Mbits/sec
[ 21] 12.0-13.0 sec   438 KBytes  3.59 Mbits/sec
[ 21] 13.0-14.0 sec   439 KBytes  3.60 Mbits/sec
[ 21] 14.0-15.0 sec   438 KBytes  3.59 Mbits/sec
[ 21] 15.0-16.0 sec   438 KBytes  3.59 Mbits/sec
[ 21] 16.0-17.0 sec   416 KBytes  3.41 Mbits/sec
[ 21] 17.0-18.0 sec   425 KBytes  3.48 Mbits/sec
[ 21] 18.0-19.0 sec   438 KBytes  3.59 Mbits/sec
[ 21]  0.0-20.0 sec  8.66 MBytes  3.63 Mbits/sec
[ 21] Sent 6175 datagrams
[ 21] Server Report:
[ 21]  0.0-20.2 sec  8.66 MBytes  3.60 Mbits/sec  56.596 ms    0/ 6175 (0%)
```



```
root@ubuntu:/home/huid/Desktop/mininet/util# iperf -u -c 10.0.0.1 -b 10M -t 20
-i 1
------------------------------------------------------------
Client connecting to 10.0.0.1, UDP port 5001
Sending 1470 byte datagrams, IPG target: 1121.52 us (kalman adjust)
UDP buffer size:  208 KByte (default)
------------------------------------------------------------
[ 21] local 10.0.0.4 port 40203 connected with 10.0.0.1 port 5001
[ ID] Interval       Transfer     Bandwidth
[ 21]  0.0- 1.0 sec   194 KBytes  1.59 Mbits/sec
[ 21]  1.0- 2.0 sec  90.4 KBytes   741 Kbits/sec
[ 21]  2.0- 3.0 sec  90.4 KBytes   741 Kbits/sec
[ 21]  3.0- 4.0 sec  90.4 KBytes   741 Kbits/sec
[ 21]  4.0- 5.0 sec  68.9 KBytes   564 Kbits/sec
[ 21]  5.0- 6.0 sec  66.0 KBytes   541 Kbits/sec
[ 21]  6.0- 7.0 sec  90.4 KBytes   741 Kbits/sec
[ 21]  7.0- 8.0 sec  90.4 KBytes   741 Kbits/sec
[ 21]  8.0- 9.0 sec  90.4 KBytes   741 Kbits/sec
[ 21]  9.0-10.0 sec  90.4 KBytes   741 Kbits/sec
[ 21] 10.0-11.0 sec  89.0 KBytes   729 Kbits/sec
[ 21] 11.0-12.0 sec  47.4 KBytes   388 Kbits/sec
[ 21] 12.0-13.0 sec  90.4 KBytes   741 Kbits/sec
[ 21] 13.0-14.0 sec  89.0 KBytes   729 Kbits/sec
[ 21] 14.0-15.0 sec  90.4 KBytes   741 Kbits/sec
[ 21] 15.0-16.0 sec  90.4 KBytes   741 Kbits/sec
[ 21] 16.0-17.0 sec  90.4 KBytes   741 Kbits/sec
[ 21] 17.0-18.0 sec  45.9 KBytes   376 Kbits/sec
[ 21] 18.0-19.0 sec  90.4 KBytes   741 Kbits/sec
[ 21]  0.0-20.2 sec  1.73 MBytes   719 Kbits/sec
[ 21] Sent 1236 datagrams
[ 21] Server Report:
[ 21]  0.0-21.3 sec  1.73 MBytes   681 Kbits/sec  50.054 ms    0/ 1236 (0%)
```

参考

https://www.sdnlab.com/23289.html

https://www.cxyzjd.com/article/u012707739/85337640

https://blog.51cto.com/u_13958494/2177890

https://www.sdnlab.com/23289.html

https://github.com/mininet/mininet

http://www.openvswitch.org/support/dist-docs/ovs-vsctl.8.txt

[illegible]

https://mailman.stanford.edu/pipermail/mininet-discuss/2015-March/005837.html

https://lists.linuxfoundation.org/pipermail/ovs-discuss/2015-November/019565.html

https://mailman.stanford.edu/pipermail/mininet-discuss/2014-February/004089.html

https://www.eet-china.com/mp/a82420.html

https://cloud.tencent.com/developer/article/1667220

https://docs.pica8.com/pages/viewpage.action?pageId=3083175