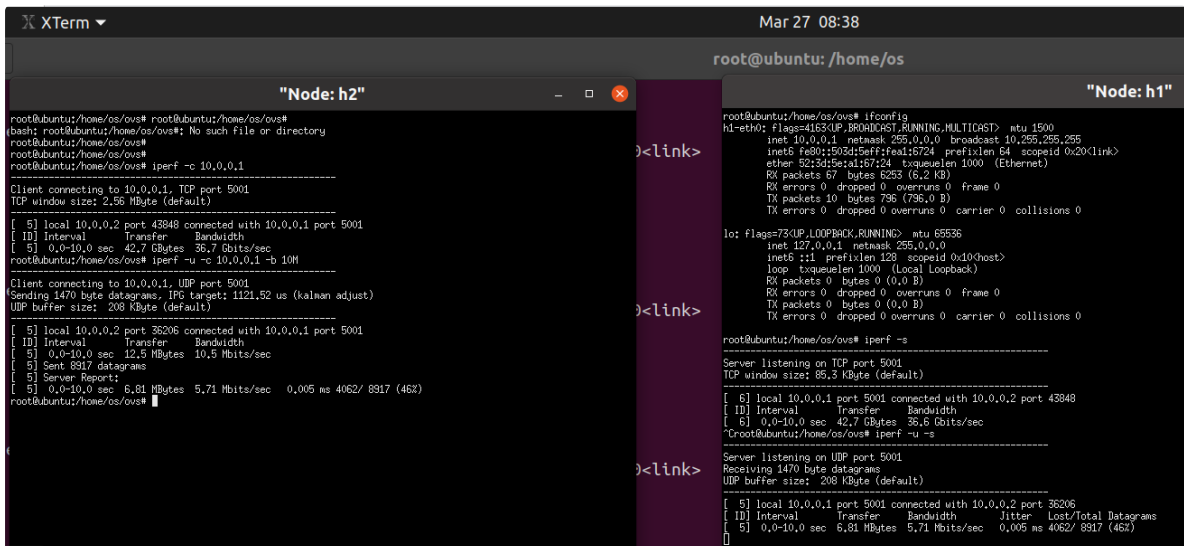


Task2

网卡限速：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。



```
root@ubuntu:/home/os/ovs# root@ubuntu:/home/os/ovs#
bash: root@ubuntu:/home/os/ovs#: No such file or directory
root@ubuntu:/home/os/ovs#
root@ubuntu:/home/os/ovs#
root@ubuntu:/home/os/ovs# iperf -c 10.0.0.1
Client connecting to 10.0.0.1, TCP port 5001
TCP window size: 2.56 MByte (default)
[ 5] local 10.0.0.2 port 43848 connected with 10.0.0.1 port 5001
[ ID] Interval      Transfer      Bandwidth
[ 5] 0.0-10.0 sec  42.7 GBytes  36.7 Gbits/sec
root@ubuntu:/home/os/ovs# 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 (kaiman adjust)
UDP buffer size: 208 KByte (default)
[ 5] local 10.0.0.2 port 36206 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] 0.0-10.0 sec  6.81 MBytes  5.71 Mbits/sec  0.005 ms 4062/ 8917 (46%)
root@ubuntu:/home/os/ovs#

root@ubuntu:/home/os/ovs# ifconfig
hi-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::505d:5eff:feaf:6724 prefixlen 64 scopeid 0x20<link>
    ether 52:3d:3d:a1:67:24 txqueuelen 1000 (Ethernet)
    RX packets 67 bytes 6253 (6.2 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 10 bytes 736 (736.0 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/os/ovs# 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 43848
[ ID] Interval      Transfer      Bandwidth
[ 6] 0.0-10.0 sec  42.7 GBytes  36.6 Gbits/sec
root@ubuntu:/home/os/ovs# 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 36206
[ ID] Interval      Transfer      Bandwidth      Jitter  Lost/Total Datagrams
[ 5] 0.0-10.0 sec  6.81 MBytes  5.71 Mbits/sec  0.005 ms 4062/ 8917 (46%)
```

分析：带宽为 5.71Mbps，比设定的稍高,可能是因为，抖动 jitter 为 0.005ms，非常低，因为没有延迟策略，丢包率为 46%，因为发送的是 10Mbps，是上限的两倍。

队列限速：Task2.2

```
ities  Terminal  Mar 28 02:29
root@ubuntu: /home/os

TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

s1-eth5: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
inet6 fe80::5852:edff:fed9:181d prefixlen 64 scopeid 0x20<link>
ether 5a:52:ed:d9:18:1d txqueuelen 1000 (Ethernet)
RX packets 13 bytes 1006 (1.0 KB)
RX errors 0 dropped 0 overruns 0 frame 0
TX packets 90 bytes 7962 (7.9 KB)
TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

s1-eth6: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
inet6 fe80::6c7c:78ff:fe6a:58a0 prefixlen 64 scopeid 0x20<link>
ether 6e:7c:78:6a:58:a0 txqueuelen 1000 (Ethernet)
RX packets 13 bytes 1006 (1.0 KB)
RX errors 0 dropped 0 overruns 0 frame 0
TX packets 90 bytes 7962 (7.9 KB)
TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

root@ubuntu:/home/os# ovs-vsctl set interface s1-eth1 ingress_policing_rate=5000
root@ubuntu:/home/os# ovs-vsctl set interface s1-eth2 ingress_policing_rate=5000
root@ubuntu:/home/os# 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
373b008d-0b2b-4979-ad66-5505ad8b3292
034c284b-72db-46b3-806a-687853996842
root@ubuntu:/home/os# ovs-vsctl list qos
_uuid          : 373b008d-0b2b-4979-ad66-5505ad8b3292
external_ids   : {}
other_config   : {}
queues         : {0=034c284b-72db-46b3-806a-687853996842}
type           : linux-htb
root@ubuntu:/home/os# ovs-vsctl list queue
_uuid          : 034c284b-72db-46b3-806a-687853996842
dscp           : []
external_ids   : {}
other_config   : {max-rate="5000000"}
```

```
Mar 28 09:35
root@ubuntu: /home/os/ovs

"Node: h4"
root@ubuntu:/home/os/ovs# 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 57638
[ ID] Interval      Transfer    Bandwidth    Jitter    Lost/Totals  Datagrams
[ 5] 0.0-10.2 sec   5.91 MBytes  4.86 Mbits/sec  9.592 ms   0/ 4216 (0%)

"Node: h3"
root@ubuntu:/home/os/ovs# 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 57638 connected with 10.0.0.4 port 5001
[ ID] Interval      Transfer    Bandwidth    Jitter    Lost/Totals  Datagrams
[ 5] 0.0-10.1 sec   5.91 MBytes  4.91 Mbits/sec
[ 5] Sent 4216 datagrams
[ 5] Server Report:
[ 5] 0.0-10.2 sec   5.91 MBytes  4.86 Mbits/sec  9.592 ms   0/ 4216 (0%)
root@ubuntu:/home/os/ovs#
```

分析：带宽为 4.86Mbps，比设定的稍高,可能是因为，抖动 jitter 为 9.592ms，比较高，因为进入队列后会有延迟，丢包率为 0，因为使用队列技术。

Meter 表限速：Task2.3

Q1:

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

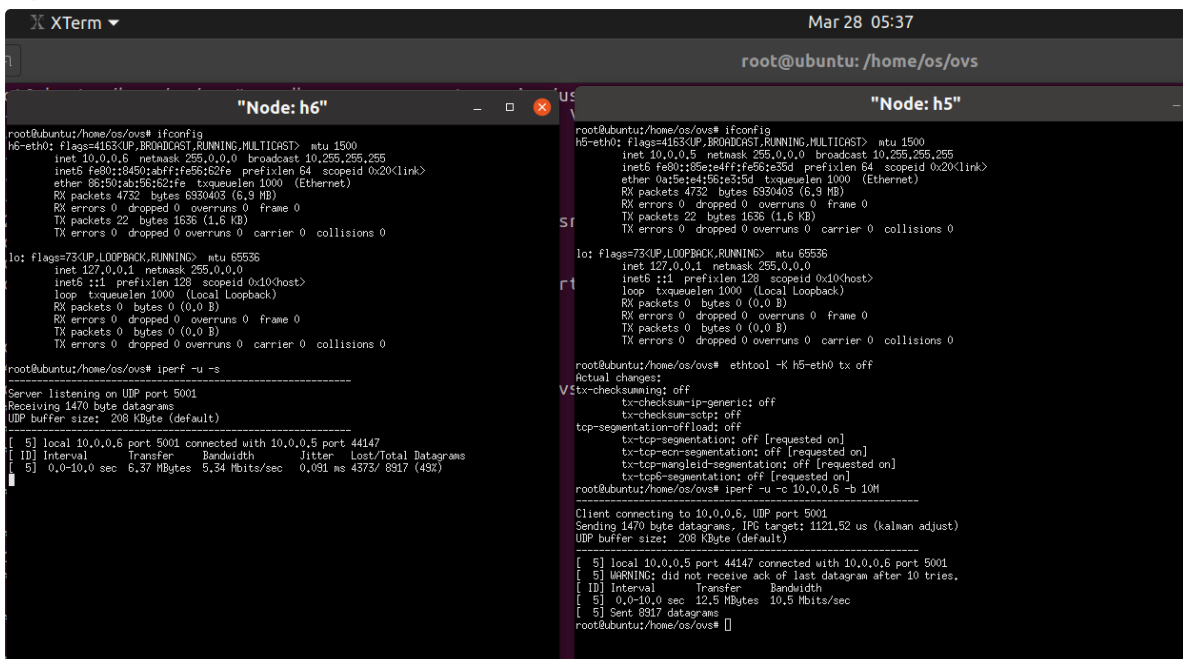
这条指令的作用是向 Open vSwitch 交换机 s1 添加一条流表项。当交换机收到从端口 5 (h5) 进入的数据包时，将会执行 meter ID 为 1 的流量测量，并将数据包从端口 6 输出。

其中，`in_port=5` 表示匹配数据包的输入端口为 5；`action=meter:1,output:6` 表示匹配成功后执行 meter ID 为 1 的流量测量，并将数据包从端口 6 输出；`-O openflow13` 表示使用 OpenFlow1.3 进行操作。

2. `ovs-ofctl dump-flows s1 -O openflow13`

这条指令的作用是从 Open vSwitch 交换机 s1 中获取当前所有的流表项，并将其输出到控制台。它会显示出每个流表项的匹配条件、处理动作、优先级等信息。

其中，`dump-flows` 表示获取所有流表项的操作；`s1` 表示要获取流表项的交换机名称；`-O openflow13` 表示使用 OpenFlow1.3 进行操作。



The screenshot shows two terminal windows side-by-side. The left window, titled "Node: h6", shows the configuration of interface h6-eth0 and a local loopback interface lo. It then shows the output of the `iperf -u -s` command, indicating it is listening on UDP port 5001. The right window, titled "Node: h5", shows the configuration of interface h5-eth0 and a local loopback interface lo. It then shows the output of the `iperf -u -c 10.0.0.5 -b 10M` command, indicating it is connecting to 10.0.0.5 on UDP port 5001. The output of the `iperf` command shows a bandwidth of 12.5 MB/sec, which is slightly above the 5 Mbps limit, but the packet loss rate is 0%.

```
root@ubuntu:/home/os/ovs# ifconfig
h6-eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 10.0.0.6 netmask 255.0.0.0 broadcast 10.255.255.255
    inet6 fe80::18450:abff:fe56:1b2fe prefixlen 64 scopeid 0x20<link>
    ether 86:15:0a:b5:6321fe txqueuelen 1000 (Ethernet)
    RX packets 4732 bytes 6930403 (6.9 MB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 22 bytes 1636 (1.6 KB)
    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/os/ovs# 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 44147
[ ID] Interval      Transfer      Bandwidth      Jitter  Lost/Total Datagrams
[ 5] 0.0-10.0 sec  6.37 MBytes  5.34 Mbits/sec  0.091 ms 4373/ 8917 (49%)

root@ubuntu:/home/os/ovs# ifconfig
h5-eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 10.0.0.5 netmask 255.0.0.0 broadcast 10.255.255.255
    inet6 fe80::18450:abff:fe56:1b2fe prefixlen 64 scopeid 0x20<link>
    ether 86:15:0a:b5:6315d txqueuelen 1000 (Ethernet)
    RX packets 4732 bytes 6930403 (6.9 MB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 22 bytes 1636 (1.6 KB)
    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/os/ovs# ethtool -K h5-eth0 tx off
Actual changes:
V:tx-checksumming: off
   tx-checksum-ip-generic: off
   tx-checksum-sctp: off
tcp-segmentation-offload: off
   tx-tcp-segmentation: off [requested on]
   tx-tcp-ecn-segmentation: off [requested on]
   tx-tcp-mangleid-segmentation: off [requested on]
   tx-tcp-segmentation: off [requested on]
root@ubuntu:/home/os/ovs# iperf -u -c 10.0.0.5 -b 10M
Client connecting to 10.0.0.5, UDP port 5001
Sending 1470 byte datagrams, IPG target: 1121.52 us (kcalman adjust)
UDP buffer size: 208 KByte (default)

[ 5] local 10.0.0.5 port 44147 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/os/ovs#
```

实验截图如图所示，带宽还是没有控制在 5Mbps 以下，但是相对误差比网卡限速更好，说明控制粒度更精细，同时抖动远远小于队列限速。丢包率为 0。

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(百分比)	是否成功限制带
网卡限速	5.71	0.005	46%	否
队列限速	4.86	9.592	0	是
Meter 表限速	5.34	0.091	49%	否

从带宽来看，只有队列限速成功限制带宽在 5Mbps 以下，说明控制粒度最好，并且队列限速丢包率为 0，因此是最好的方案。

网卡限速相对粗粒度，因此带宽超过限额最多，但是由于不需要队列等机制抖动相对较低。由于丢包率较高，只适合实时性相对较高的场景。

Meter 表限速超过限额但是相对较少，说明控制粒度适中，同时因为是软件模拟，软件实现的交换机对流表的控制比不上硬件交换机。但是 Meter 表控制的优势是抖动也比较低。适合简单方案。