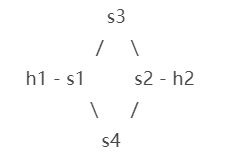
**Lab：Write SDN Controller**

叶增渝 519030910168

1.构建如下拓扑结构的文件为test.py（可放在任意文件下执行）



2.符合要求的remote controller文件为6\_1.py，文件放在~/ryu/app文件夹下执行如下命令开启controller



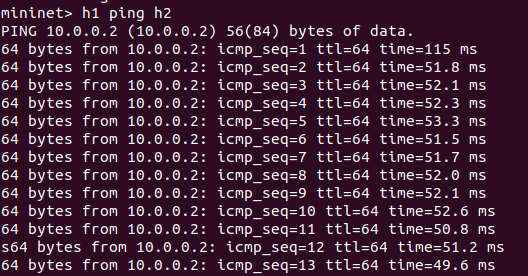
然后我们再使用如下命令构建网络拓扑



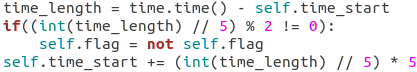
在mininet中使用如下命令测试网络转发

C:\Users\HP\AppData\Local\Microsoft\Windows\INetCache\Content.Word\屏幕截图 2021-10-31 204950.png

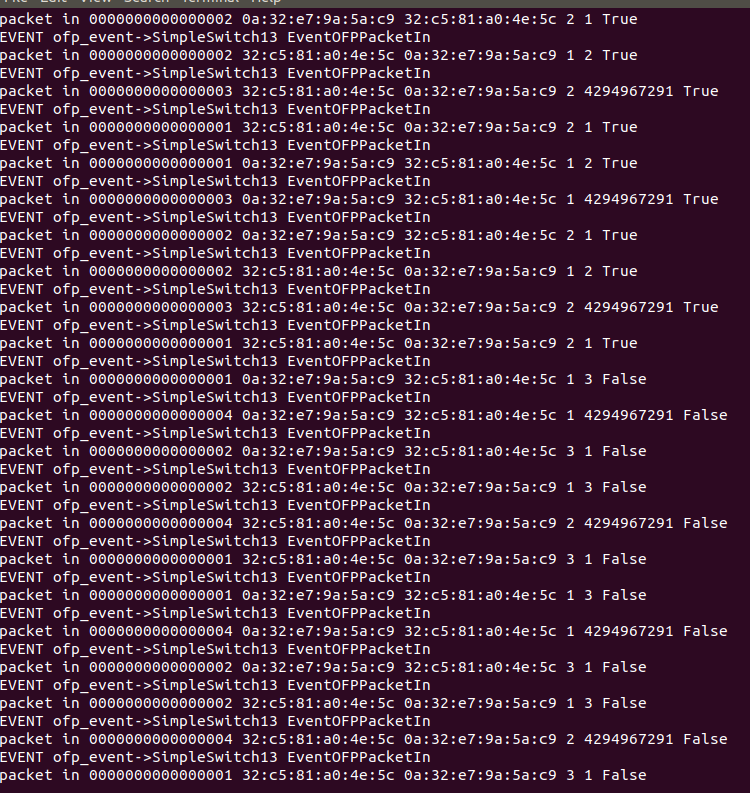
首先我们可以看到是能够ping通的



然后我们查看packet\_in\_handler发出的信息，我们可以看到在flag为True时，我们使用经过s3的线路进行转发，在时间到达后，我们的flag被取反，此时使用经过s4的线路进行转发，说明达到目标。其中计时我们使用作为计时开始，每达到5s更改flag切换线路（由于频繁切换线路，我们不添加流表项）



p.s.这里输出的信息按照顺序分别为16位dp\_id、source、destination、in\_port、out\_port（4294967291为OFPP\_FLOOD的输出）、flag



3. 符合要求的remote controller文件为6\_2.py，文件放在~/ryu/app文件夹下执行如下命令开启controller



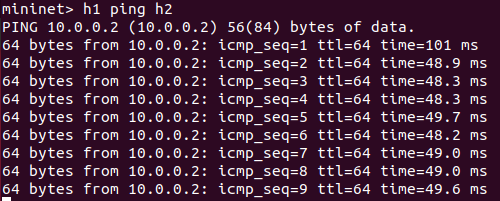
然后我们再使用如下命令构建网络拓扑

C:\Users\HP\AppData\Local\Microsoft\Windows\INetCache\Content.Word\屏幕截图 2021-10-31 204746.png

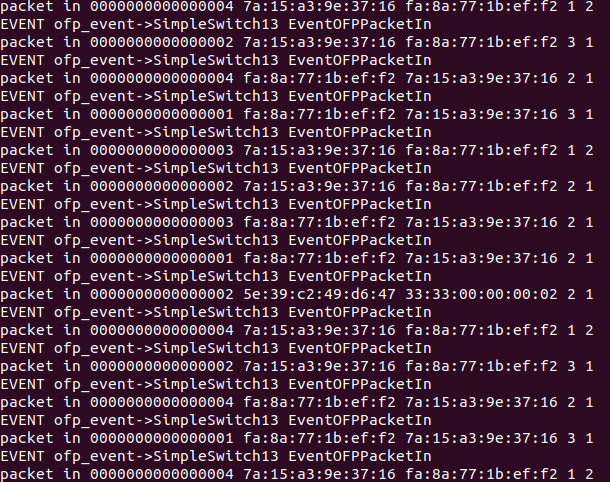
在mininet中使用如下命令测试网络转发

C:\Users\HP\AppData\Local\Microsoft\Windows\INetCache\Content.Word\屏幕截图 2021-10-31 204950.png

首先我们可以看到是能够ping通的



然后我们查看packet\_in\_handler发出的信息，我们可以看到在ping测试时，同时能接收到来自s3与s4的消息，说明两路同时在转发， 由于ping通，我们可以知道是各50%的两路转发



4. 符合要求的remote controller文件为6\_3.py，文件放在~/ryu/app文件夹下执行如下命令开启controller（这里在s1与s3使用了OFPGT\_FF，当link断开时，s1自然地选择下方通路，而已经从s2传至s3的packet会改为流向s2，s2根据记录的h2的eth与packet的source进行比较，将返回的包重新发往s4）

C:\Users\HP\AppData\Local\Microsoft\Windows\INetCache\Content.Word\屏幕截图 2021-10-31 211836.png

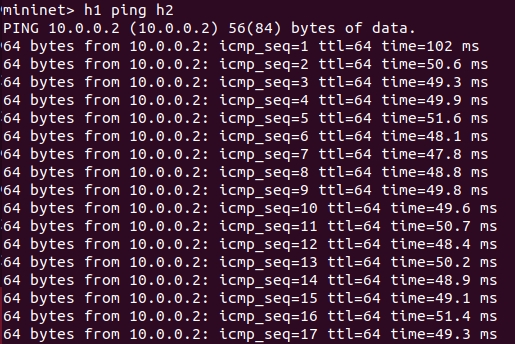
然后我们再使用如下命令构建网络拓扑

C:\Users\HP\AppData\Local\Microsoft\Windows\INetCache\Content.Word\屏幕截图 2021-10-31 204746.png

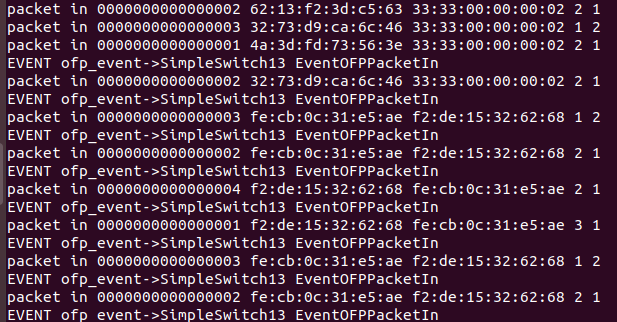
在mininet中使用如下命令测试网络转发

C:\Users\HP\AppData\Local\Microsoft\Windows\INetCache\Content.Word\屏幕截图 2021-10-31 204950.png

首先我们可以看到是能够ping通的



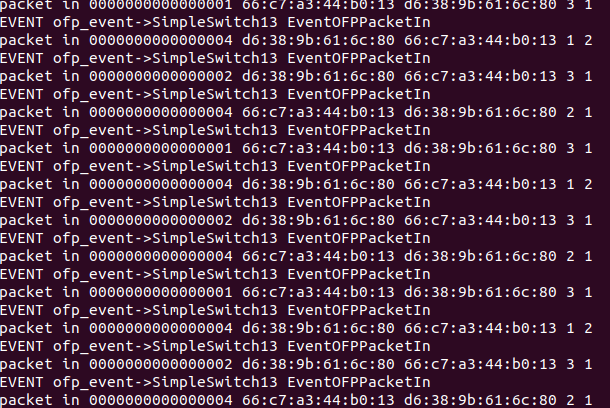
此时的输出信息如下



我们使用如下命令将连接s3的端口2断开



此时我们可以看到依然每一个packet都ping通，但是此时，我们只能看到含s4的线路转发



说明完成了fast failover操作