1. 实验中如何获取吞吐率随时间变化的结果？
   * 如果考察吞吐率，可以将iperf的结果输出出来，命令格式如下： iperf -c 10.0.0.2 -t $duration -i 0.1 | tee iperf\_result.txt，每0.1秒输出一次吞吐率结果。iperf的使用手册是错误的，使用-o参数不会输出到文件中。
2. 按照上面的命令样式修改start\_iperf函数，还看不到输出文件呢？
   * start\_iperf函数中，h1启动iperf命令的方式是popen(就是Linux管道的封装)，我们在iperf命令后又通过管道(“|”)将输出重定向到tee。前面的管道已将输出重定向到Mininet终端，因此后面的管道不再起作用。正确的用法应该是，将h1.popen一行改为如下（注意命令结尾的“&” ）：

h1.cmd('iperf -c %s -t %d -i 0.1 | tee iperf\_result.txt &' % (h2.IP(), duration+ 5))

1. 随着maxq值逐渐变大， ping.txt文件中的报文的总数量在逐渐变小，为什么？
   * 在60秒内，每0.1秒发送一个ping请求，获得一个ping应答，理论上应该返回600个结果，有两种情形会导致结果数目变少：1. ping的请求或应答数据包丢失；2. RTT过大，导致结果没来得及返回。对于丢包导致的ping结果丢失，可以通过查看每条结果的序列号来判断哪个条目丢失，通过前后取平均值来（近似）补齐该值。
2. 有什么办法来查看数据包队列的每一项（业务）占据多少字节的空间？
   * 没办法查看。实验中的队列不区分任何数据包，对所有数据包遵从先进先出排队策略。队列可以设置的再复杂一点儿，即公平队列：把一个队列分成几个虚拟队列，新进来的数据包按IP地址+端口号hash到某个虚拟队列。这种情况下，只能保证同一业务放到同一虚拟队列中，但不保证一个队列中只有一个业务的数据包。核心问题还是在于，数据包队列的处理需要极快的时间(~ns)，不能实现复杂的逻辑。
3. 复现实验中，tail drop曲线一直出现“尖端”情况，和ppt中稳定的一段峰值不同，为什么？
   * Mininet环境下的taildrop结果确实有“尖端”现象，和PPT中要复现的图不完全一样，这是仿真环境的差异导致的。在Mininet中，maxq参数实际上是一个模拟延迟和丢包的批处理大小，与真实的队列大小有区别，只是近似仿真。