

1. 实验中如何获取吞吐率随时间变化的结果?
 - 可以将iperf每隔一段时间的吞吐率结果输出出来,把h1.popen一行改为:
h1.cmd('iperf -c %s -t %d -i 0.5 | tee iperf_result.txt &' % (h2.IP(), duration+5)), 每 0.1 秒输出一次吞吐率结果。
 - 注意: 1、popen 需要改成 cmd, 否则 popen 生成的管道会将输出重定向到 Mininet 终端, 因此后面的管道不再起作用; 2、iperf 的使用说明手册有错误, 使用-o 参数不会输出到文件中。
2. 随着 maxq 值逐渐变大, rtt.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时间戳和 RTT 值来判断哪个条目丢失, 通过前后取平均值来(近似)补齐该值。
3. 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。
4. 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 结果中的“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。