

Test TCP Performance

计算机网络 CS339

李子龙 518070910095

2021 年 10 月 20 日

目录

1	TCP Vegas	2
2	测试拥塞控制算法	2
3	多因素分析	3
3.1	带宽	3
3.2	延迟	3
3.3	丢包	3
4	多机分析	3

In this lab, we test TCP performance using Mininet. There are different TCP congestion control algorithms, each was designed with different optimization purpose. You can enable different TCP algorithms easily in Ubuntu.

1 TCP VEGAS

Pick one TCP congestion control algorithm other than TCP Reno, and explain how it works.
Vegas 的基本思想是：

- (1) 在分组丢失发生之前，在源与目的地之间检测路由器中的拥塞（通过 RTT 预测）；
- (2) 当检测出快要发生的分组丢失时，线性地降低发送速率。

2 测试拥塞控制算法

Enable TCP Reno and your selected TCP congestion control algorithm, and test them in Mininet.

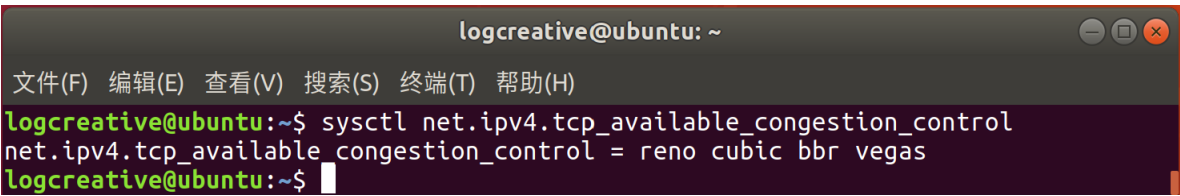


图 1: 开启TCP拥塞控制算法

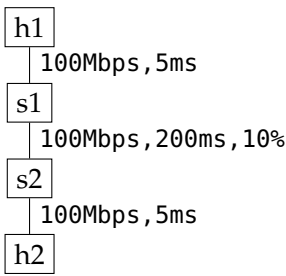


图 2: 测试结构

表 1: TCP 吞吐量 (Mbps)

Alg	Throughput
reno	20.3
vegas	56.3

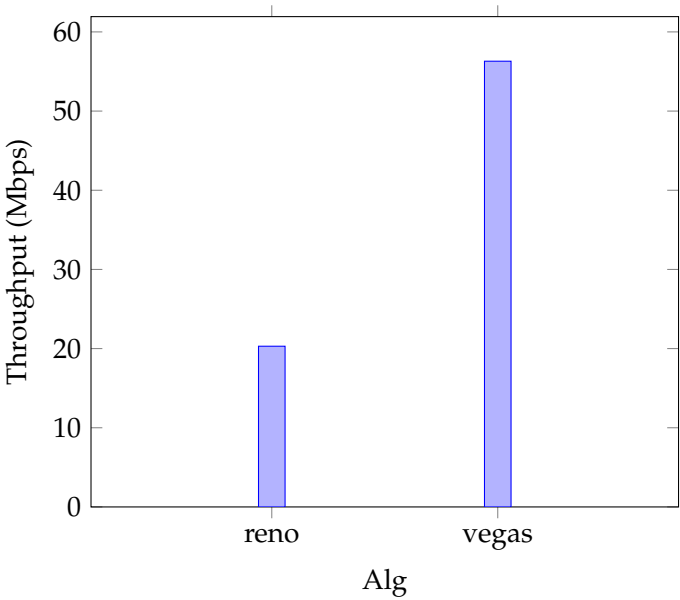


图 3: TCP 吞吐量

首先开启 TCP 拥塞控制算法 TCP Vegas，见图 1。之后对两种算法进行测试，架构如图 2，测试代码见 `task2.py`。

3 多因素分析

Construct a network with only one pair of sender and receiver. Study how TCP throughput varies with respect to link bandwidth/link delay/loss rate for the above two TCP versions.

默认状态就是上一节图 2 所对应的状态。本节测试的目的是为了对拥塞进行分析，所以对于两个交换机之间的连接进行单变量限制，也就是在默认值左右进行调整，以查看相关结果。

3.1 带宽

带宽进行 ± 80 Mbps 测试，每 10 Mbps 分为一个间隔。

3.2 延迟

延迟进行 ± 200 ms 测试，每 40 ms 分一个间隔。

3.3 丢包

丢包进行 $-10\% \sim +50\%$ 测试，每 5% 分一个间隔。

4 多机分析

Construct a network with a bottleneck link shared by multiple pairs of senders and receivers. Study how these sender-receiver pairs share the bottleneck link.