

实验七 TCP 协议的拥塞控制机制观察

161130118 尹浚宇

实验目的

熟练掌握 TCP 协议的拥塞控制机制

学会捕获和处理数据包获得 TCP 拥塞窗口曲线

能够根据相应图像分析出 TCP 拥塞控制的各个阶段以及 TCP 流的吞吐量和丢包率

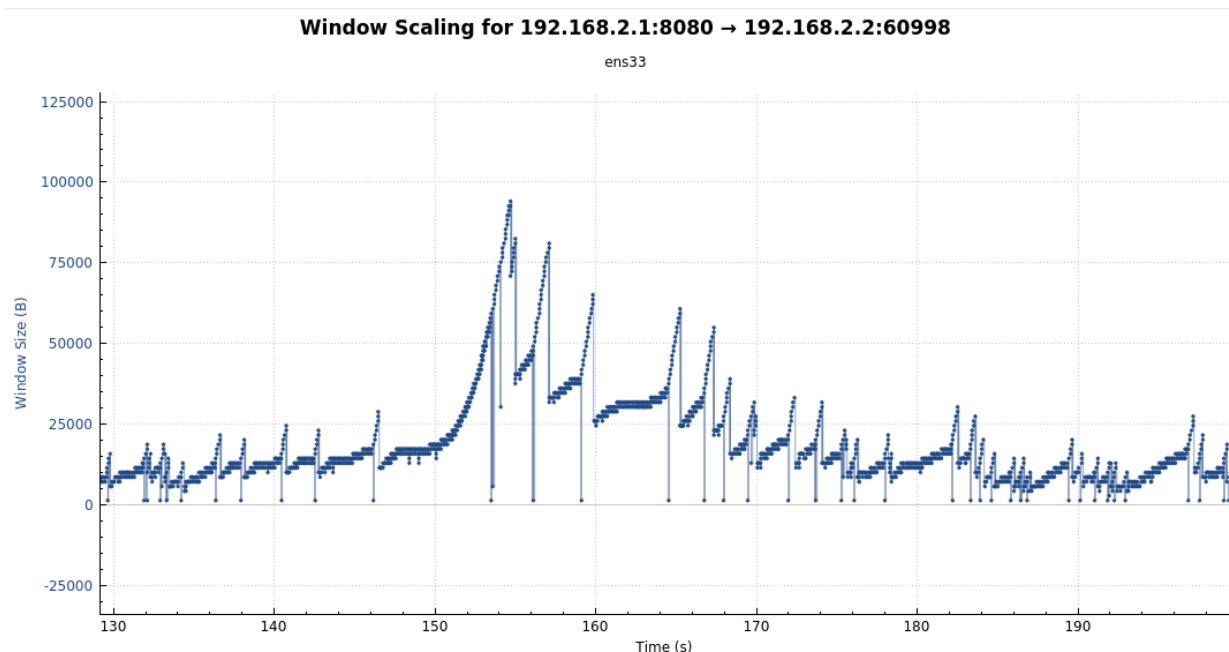
实验过程

因为实际网络情况错综复杂，难以观察到理想的 TCP 拥塞控制情况，故这里使用两台虚拟机搭建文件服务器并限制带宽的做法来获取相对可控的网络环境。在该网络环境中观察 TCP 拥塞控制机制就显得相对容易，并且 wireshark 获取的图像也相对直观明了。

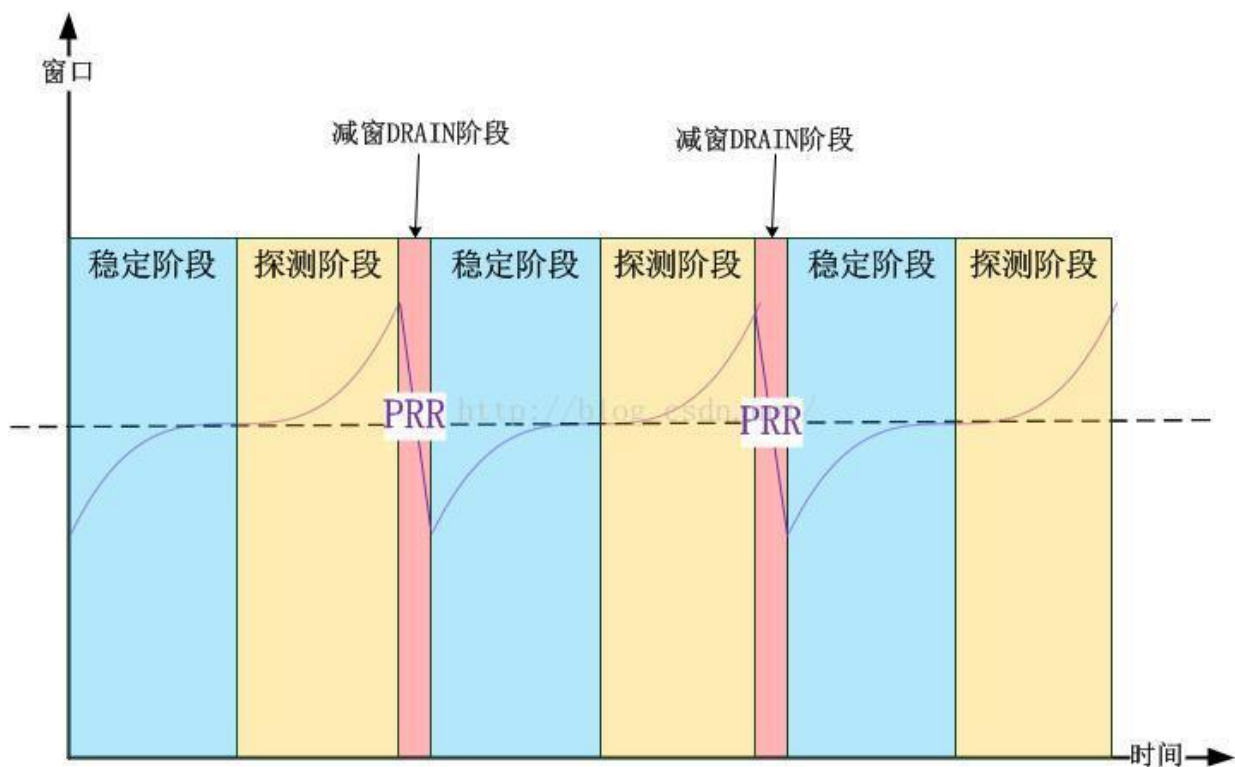
具体做法为，一台虚拟机向另一台虚拟机发送文件，同时使用 wireshark 抓包，结束后直接调用 wireshark 的统计功能生成图像。

congestion window 变化曲线分析

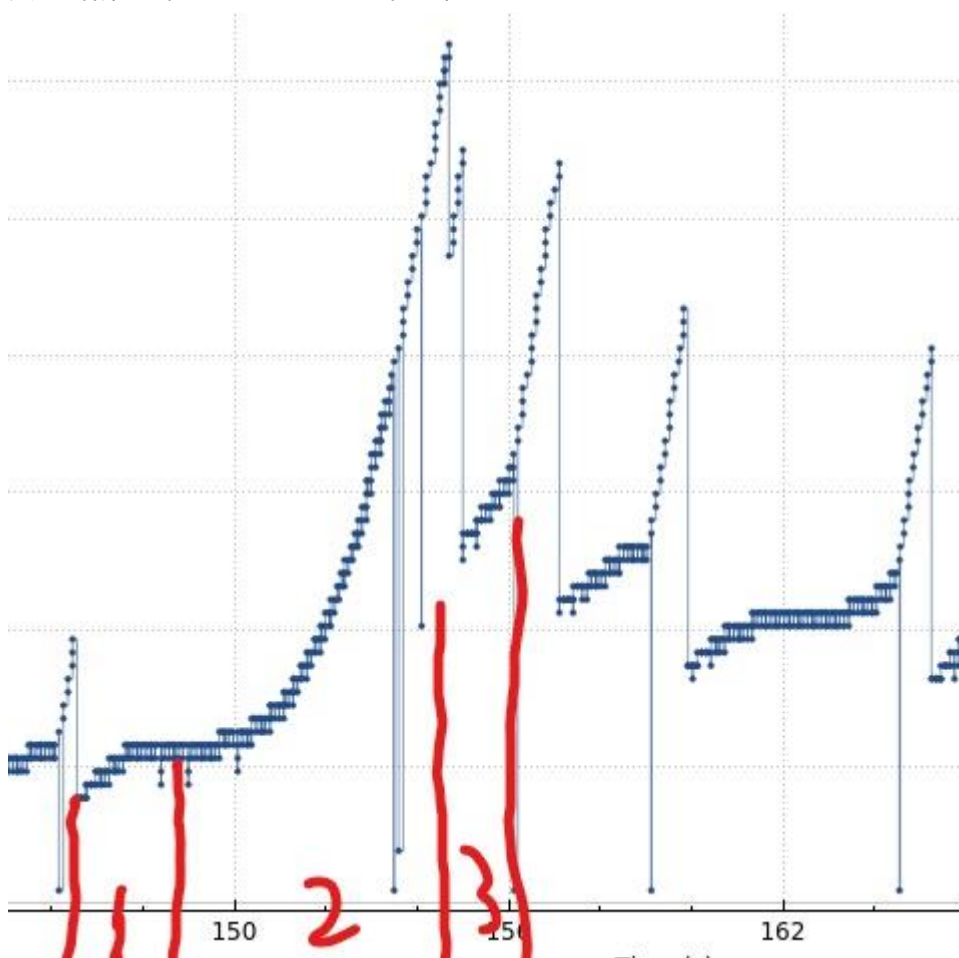
下图是传文件过程中整体窗口的变化情况



由于 ubuntu 18.04 上使用的 tcp 拥塞控制协议为 tcp cubic，与教材上所述的拥塞控制协议有所不同，故这里生成的图像和标准的 AIMD 带来的锯齿形图像有所区别。但 cubic 算法也可大致分为稳定，探测和减窗三个阶段。该算法的核心大致是以三次函数来拟合整个 cwnd-time 曲线，并以三次函数的拐点来区分稳定和探测阶段。具体见下图。



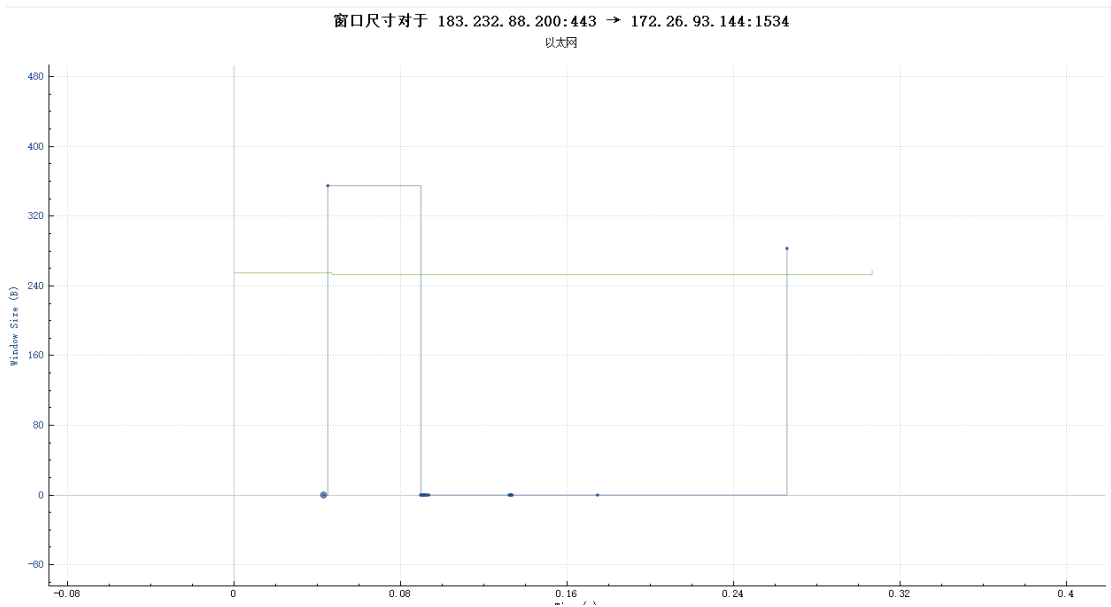
具体的阶段划分见下面放大后的局部图.



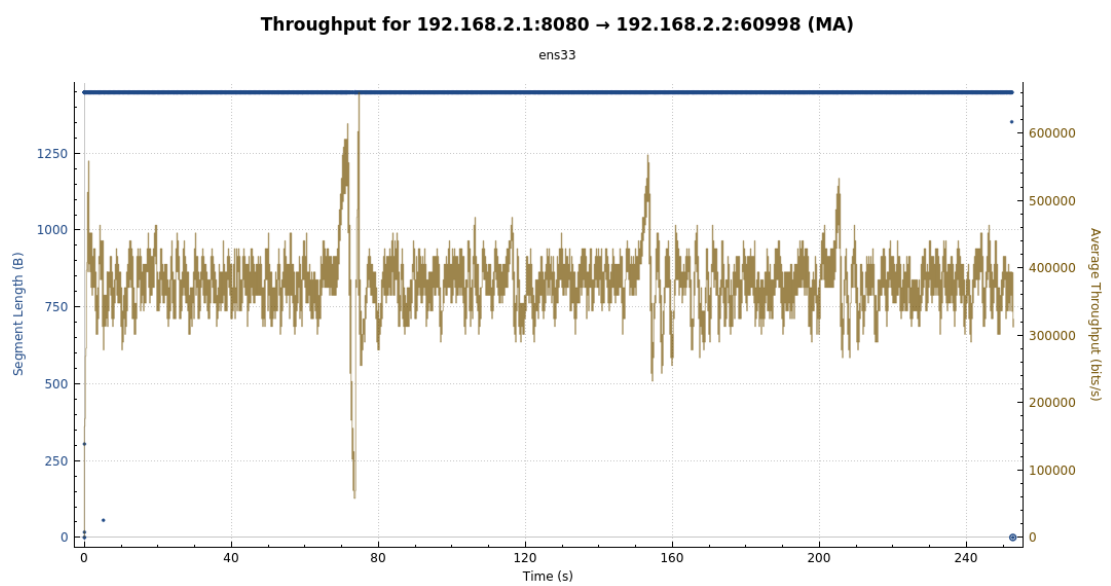
这里第一阶段对应稳定阶段, 第二阶段对应探测阶段, 第三阶段对应收到三个连续冗余 ACK

后的窗口减半和恢复阶段.

对于短流而言，这里尝试了多次都未能捕获到合理的曲线图。下面是一个发邮件时捕获的短流曲线图。



吞吐量分析



该图是 wireshark 生成的瞬时吞吐量-时间图，从图中可以大致看出吞吐量随时间波动的情况。总体而言是比较无规律的，但可以观察到几个抖动明显的时间段与窗口大小-时间图里曲线抖动明显的时间段是相对应的。

Statistics			
Measurement	Captured	Displayed	Marked
Packets	14616	14600 (99.9%)	—
Time span, s	254.265	252.687	—
Average pps	57.5	57.8	—
Average packet size, B	912	913	—
Bytes	13323699	13322859 (100.0%)	0
Average bytes/s	52 k	52 k	—
Average bits/s	419 k	421 k	—

该图是 wireshark 给出的整个传输过程的数据统计情况. 可以得到整个过程的平均吞吐量大约为 52k bytes/s.

丢包率分析

根据 TCP 拥塞控制协议, 发生丢包时窗口大小会直接减小到一个很小的值而不是简单减半. 故观察窗口大小-时间图, 曲线减少到一个很小的值时, 可判断发生了一次丢包. 于是观察整个图, 统计其中曲线减少到很小值的次数, 就可得到丢包次数.

对于该次实验文件传输对应的图像而言, 经过肉眼计数, 我得到丢包次数大约为 153 次. 发送包的总个数由 wireshark 的数据统计情况可知, 为 14600 个, 于是计算丢包率为 0.0105. 这与我配置虚拟机网络时设定的网络丢包率大致相当, 接近百分之 1.

网络适配器高级设置

传入传输

带宽(B): 自定义

Kbps(K): 500

数据包丢失(%) (P): 1.0

延迟 (毫秒)(I): 0

传出传输

带宽(A): 自定义

Kbps(S): 500

数据包丢失(%) (L): 1.0

延迟 (毫秒)(E): 0

MAC 地址(M)

00:0C:29:CB:88:0D

生成(G)

确定

取消

帮助