华东师范大学数据科学与工程学院实践报告

课程名称: 计算机网络	年级: 2022级	上机实践成绩:	
指导教师 : 赵明昊	姓名 : 田亦海	学号: 10225101529	
上机实践名称 : TCP拥塞控制和缓冲区膨胀		上机实践日期: 2023/10/28	
上机实践编号:	组号:	上机实践时间 : 12:34 a.m.	

I.实验任务

创建自己的网络模拟,测试不同的缓冲区最大队列长度,完成代码,绘图,分析结果

□.使用环境

使用Vagrant配置的虚拟机中运行jupyter notebook,主机浏览器上进行实验。

皿.实验过程

1) 填写代码

这部分其实并不太难,虽然是未接触过的一套体系,但简单查阅文档并推敲代码意思即可填写出来。代码量很小,而且有许多提示。主要需要查阅一些函数的用法,以及写最后那个调用的时候注意变量名称即可。

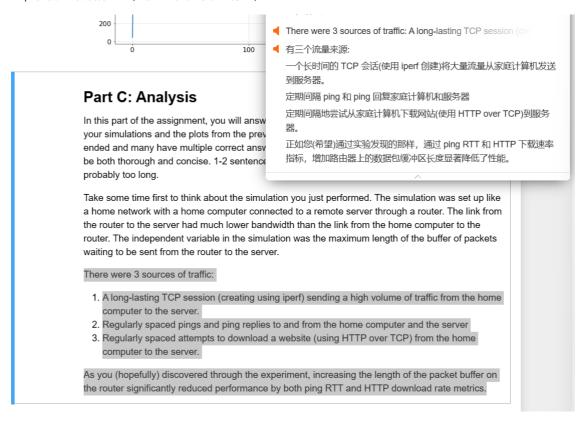
主要发生的错误是意外写出了什么bug,可以在下一阶段的绘图运行时找到错误发生点,仔细阅读代码可以很方便debug。

2) 画图

调用一次函数即可,看清楚传参为你的自定义的名称的列表.

3) 分析

这里有一个让我有些疑惑的点,就是似乎我们的代码中没有保存关于http下载的信息,这里却让我们观察http下载速率指标。(当然可能是我写漏了)



在分析阶段,重点应该是Q3Q4,需要我们结合图例与知识,阐述缓冲区膨胀效应的原理。这部分我经过了一些网上资料的查找(比如CSDN)与思考,加深了对TCP控制流量方法以及缓冲区膨胀的理解。

A3.

我们来看队列大小为100的图,这三张图表现的很明显。

起初,h1端尝试以很高的速率发送数据,但很快把s0的队列缓冲区堆满了,这表现为队列中包数和RTT急剧上升。由于此时较高的RTT,tcp无法及时感知到网络堵塞,因此持续了一小段时间后才控制堵塞窗口(cwnd)逐步减小,使网络传输逐步恢复正常。

从packets=50开始的一段逐步上升是因为缓冲区设置的过大,因此TCP错误的认为网络容量较大,因此尝试上调cwnd来加快数据发送。包数的不断上涨导致了路由器处理信息的滞后性增加,因此RTT也逐步增加,web页面下载因为排队延迟也会增加。这导致网络性能的降低。

直到缓冲区被堆满,落后的TCP感知意识到网络堵塞,尝试降低cwnd来降低发送速率。对应的路由器的接受<发送,因此表现为packets减小,RTT减小。

但是随之, TCP又尝试增大cwnd来加快传输速率, 这就是新一轮的循环。

队列数目20的结果跟队列数100的结果对比,我们可以发现,max=20时,由于平均缓冲区的队列数较小,因此RTT更短,TCP感知网络也更及时,cwnd与RTT的波动也更小。网络传输的性能更高,因此我们发现此时queue_max_length为20是更合适的选择。

Q4. ¶

Re-describe the cause of the bufferbloat effect using a non-technical analogy to something other than computer networking. It is important to be able to describe technical content such that a layperson can understand, and generating analogies often helps your own reasoning.

A4.

假设一个电梯模型,每层都有一个等候区,可以容纳一定的人来等待。如果缓冲区设置的过大会导致 什么结果呢?

- 1.等待时间增加:如果等候区设置地过大,且此时有很多人已经在等候,那么人们需要等待更久的时间来进入电梯,类似于网络中数据包传输效率降低。
- 2.混乱与拥挤:人们可能尝试挤入电梯,这会导致不必要的混乱和阻塞,就像当数据缓冲区中数据过 多时,会导致其他的比如数据丢失的问题。
- 3.不恰当的管理(指人们还会尝试涌入等候区):由于在大楼外面,看起来等待区的人还没有排到大楼外面,因此仍旧有更多的人尝试进入大楼开始等候。就像TCP算法很难适应大型大型缓冲区,导致了不合适的数据包发送,加重了堵塞问题。

电梯模型我觉得还是很贴切地描述了缓冲区膨胀效益。

IV.总结

收获:

对TCP拥塞控制的方法,以及缓冲区膨胀效应的理解程度大大加深了

对mininet等的python代码方法有了较初步的认识

查阅资料时稍微拓展了一些知识,比如CDN和AQM