

数据包队列管理实验

学号： 2021K8009929010

姓名： 贾城昊

一、实验题目：数据包队列管理实验

二、实验任务：

了解为什么需要数据包队列以及数据包队列的原理、功能，了解如何根据环境设置数据包队列的大小。认识数据包队列过大过小引起的问题，了解 BufferBloat 问题的现象和原因。学习解决 BufferBloat 问题的三种方法：

改变队列大小、改进传输控制策略、改进队列管理策略。了解 2 种典型的队列管理策略：RED (Random Early Detection) 和 CoDel (Controlled Delay)。

最后根据给定的实验环境,重现 BufferBloat 问题，改变数据包队列大小,观察其对拥塞窗口值(cwnd)、队列长度(qlen)、往返延迟(rtt)、吞吐率的影响。使用不同队列管理策略,对比在动态带宽下的往返延迟(rtt),重现讲义中实验结果。

三、实验流程

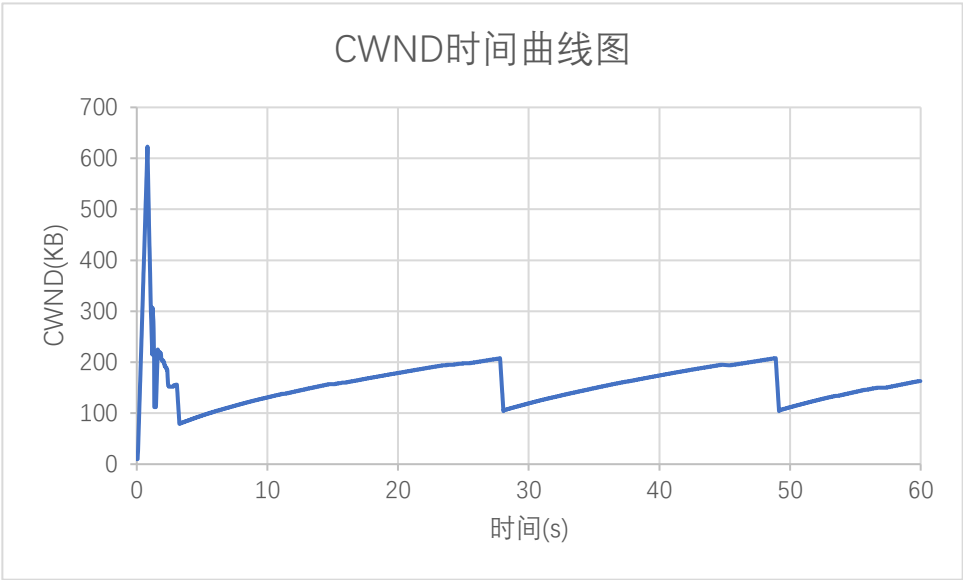
1. 重现 BufferBloat 问题,得到 CWND、Qlen、RTT 这 3 个时间曲线图。
2. 改变数据包队列大小,观察其对 CWND、Qlen、RTT 和吞吐率图像的影响。
3. 重现不同队列管理策略在动态带宽下的往返延迟结果。

四、实验结果与分析

(一) 重现 BufferBloat 问题

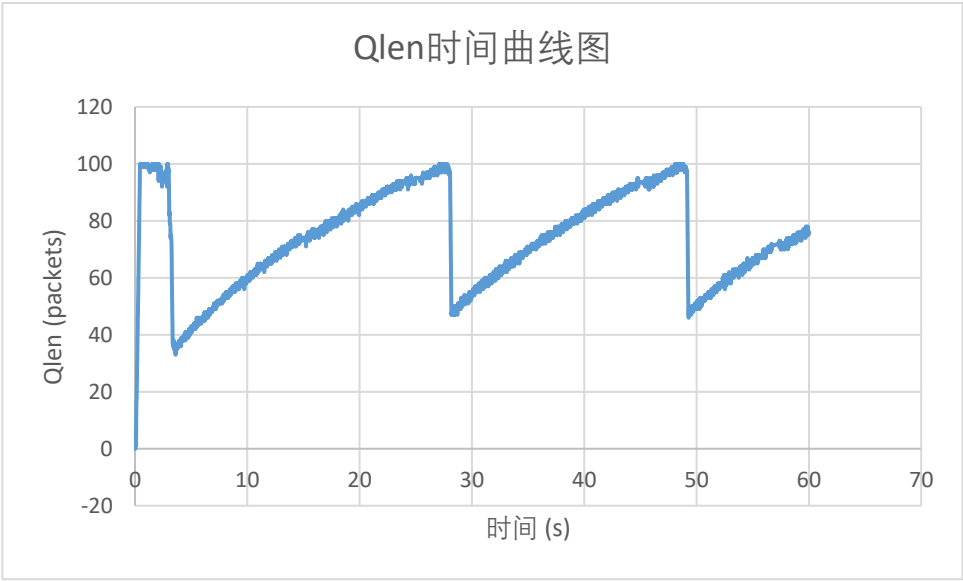
1. CWND 实验结果

取 maxq=100，作出 cwnd 与时间的曲线图如下，与讲义中实验结果一致：



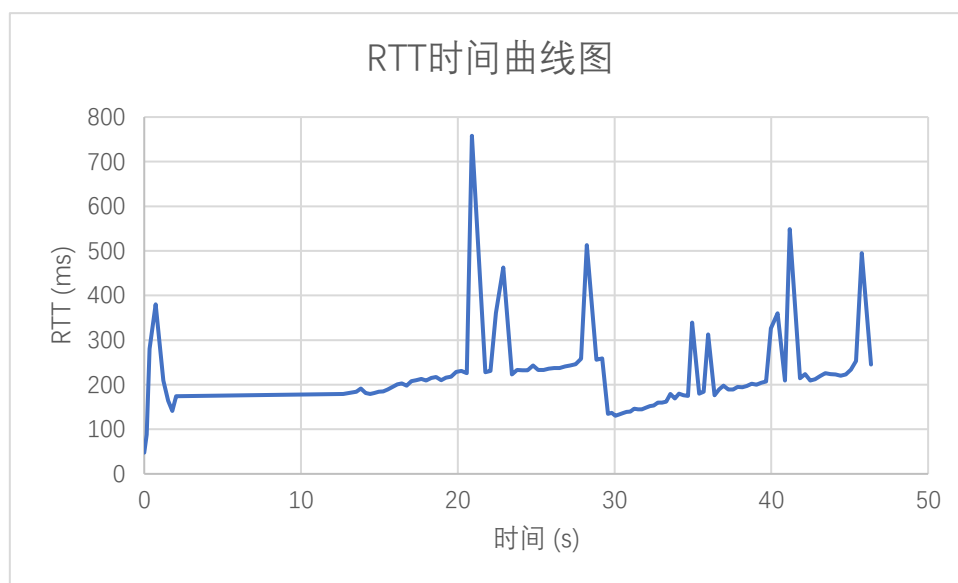
2. Qlen 实验结果

取 maxq=100，作出 qlen 与时间的曲线图如下，与讲义中实验结果一致：



3. RTT 实验结果

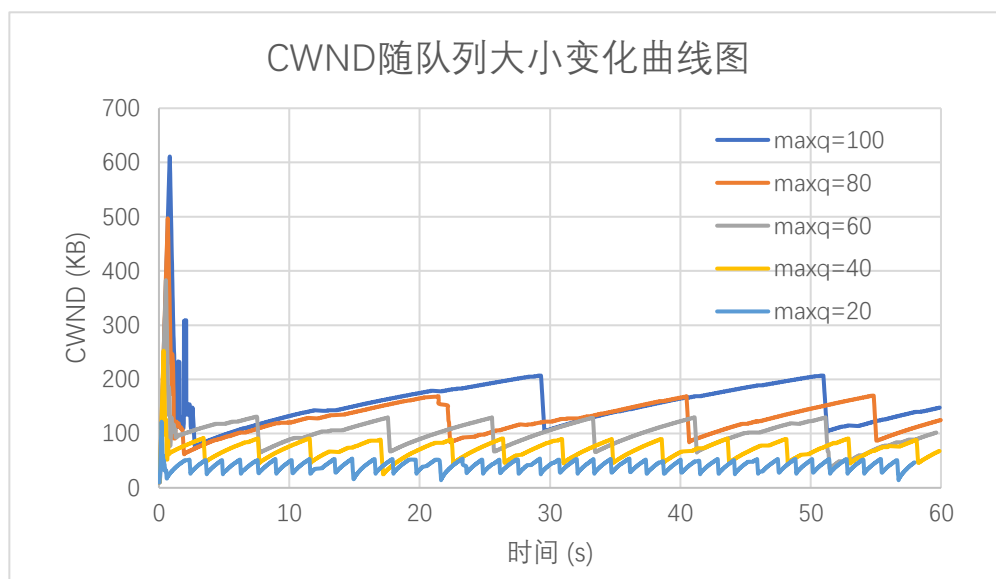
取 $\max q=100$ ，作出 rtt 与时间的曲线图如下，由于实验的 ping 程序实现没有很好的并行机制，相邻 ping 之间的间隔是变动的，低至百毫秒，高至数秒，所以实验有部分采样点相隔较远，从而得到的图像有部分没有锯齿，但图像与讲义中实验结果在曲线走势上基本一致。



（二）观察队列大小对 CWND、Qlen、RTT 和吞吐率图像的影响

1. CWND 随队列大小变化实验结果

分别取 $\max q=20, 40, 60, 80, 100$ ，作出 $cwnd$ 与时间的曲线图如下：

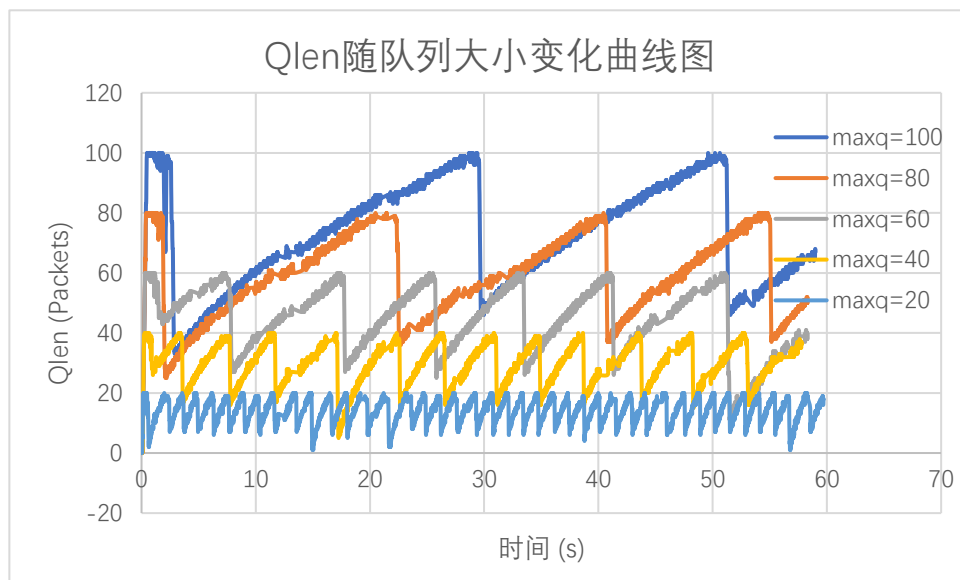


以 maxq=100 为基准，我们看到 cwnd 随时间呈现周期性变化，周期较小，变化幅度也较小。测试刚开始时，而拥塞窗口迅速增长到峰值，接着拥塞窗口迅速回落到队列大小一致的水平（100），接着再次增长。拥塞窗口随收包增长，随丢包下降。随着 maxq 增大，周期变大，每个周期的变化幅度也加剧。平均 CWND 随 maxq 逐渐增大，BufferBloat 问题加剧。

可以注意到，测试刚开始时，队列大小为 40 的拥塞窗口迅速增长达到 250 的峰值，队列大小为 80 的拥塞窗口迅速增长达到 500 的峰值。可见该实验环境下，拥塞窗口峰值大小为队列大小的两倍。另外可以看到大队列后续的波动较小队列不明显。

2. Qlen 随队列大小变化实验结果

取 maxq=20、40、60、80、100，作出 qlen 与时间的曲线图如下图所示：

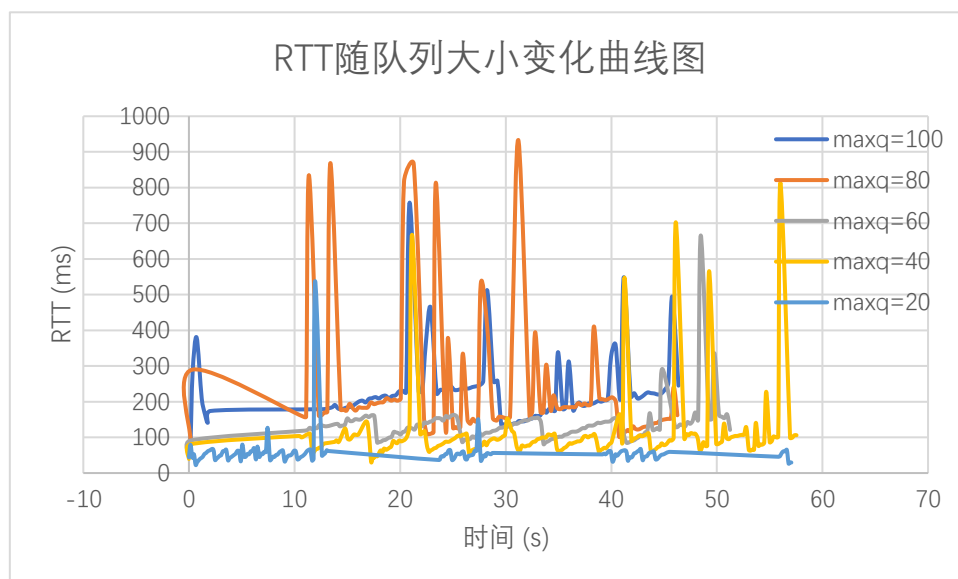


上图为即时队列长度随时间变化的曲线。以 $\text{maxq}=100$ 为基准，我们看到队列大小为 100 的曲线的上界为 100。测试刚开始时队列快速上升达到峰值，当队列满时，直接丢弃新接收的包，由于发送速率骤降，使得队列长度降低至最大队列长度的 40% 左右。随后一直这次循环下去。

以 $\text{maxq}=20$ 为基准，qlen 随时间呈现周期性变化，周期较小，变化幅度也较小。而随着 maxq 增大，周期变大，每个周期的变化幅度也加剧平均。qlen 随 maxq 逐渐增大，BufferBloat 问题加剧。

3. RTT 随队列大小变化实验结果

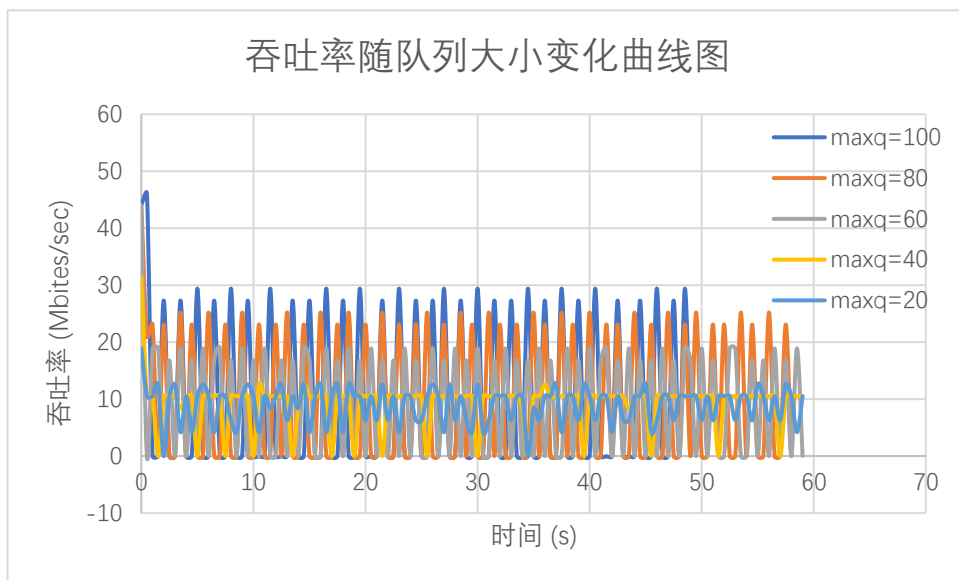
取 $\text{maxq}=20、40、60、80、100$ ，作出 qlen 与时间的曲线图如下图所示：



上图为 RTT 随时间变化的曲线。以 $\text{maxq}=20$ 为基准，我们看到 rtt 随时间呈现周期性变化，测试刚开始时，由于接收队列的快速增长，网络延时急剧增加，随后也随着队列缩短而减少，之后的时间内，延时随着队列长度的变化而变化，波动明显，但总体上仍然存在一个周期。随着 maxq 增大，这个周期变大，每个周期的变化幅度也加剧。平均 RTT 随 maxq 逐渐增大，BufferBloat 问题加剧。注意到曲线中有些突变的部分，例如 $\text{maxq}=20$ 的 15s 到 25s 段，这是由于 RTT 输出结果不稳定所致。

4. 吞吐率随队列大小变化实验结果

取 $\text{maxq}=20, 40, 60, 80, 100$ ，作出 qlen 与时间的曲线图如下图所示：



对于每一个 maxq, 我们都可以看到在最初有一个吞吐率峰值, 这是刚开始传输产生。之后吞吐率下降, 并且不同队列大小的吞吐率都在 10Mbits/s 附近波动, 与链路带宽 10Mbits/s 一致。然后注意到吞吐率有一些周期性的向下波动, 这与 cwnd/qlen/rtt 图像中, 达到峰值后下降的时间点一致。同时也可以看出, 队列越长, iperf Transfer 为 0 的可能性越大, 但是 Transfer 成功时的带宽也越大

5. 总结

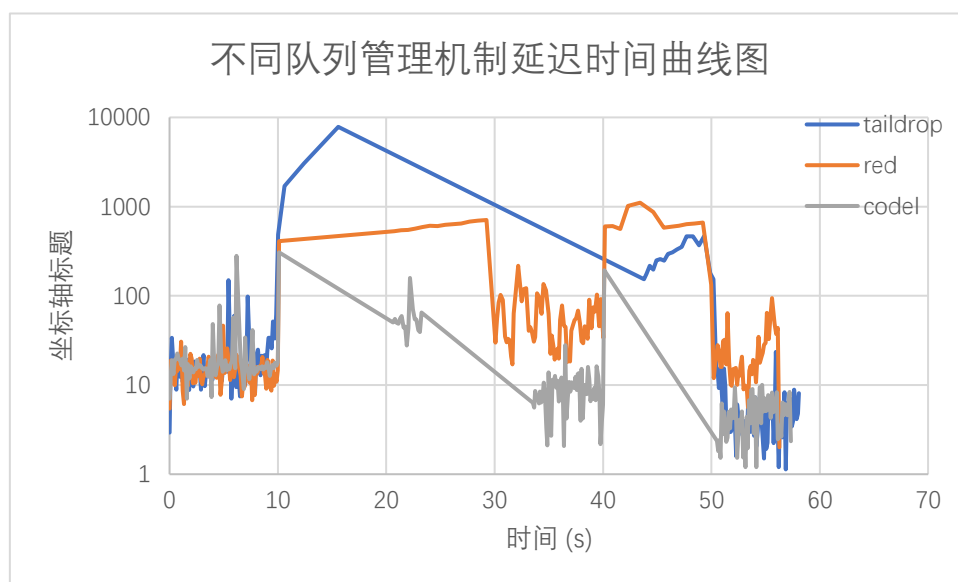
接收队列的长度受到拥塞窗口的影响, 因其长度取决于发送方的发包速率, 而发包速率又同时受到滑动窗口和拥塞窗口的制约。因此, 当拥塞窗口较大时, 可以实现更高的发送速率, 接收队列的增长也会随之加快。但是从实验结果中可以看到随着拥塞窗口增大, 接收队列增速反而减缓, 这可能是因为时延增大引起的。

另外, 随着接收队列增长, 数据包的接收延时会线性增长。当队列长度很长时, 会产生 BufferBloat 问题。tail drop 机制可以减少延时。但 tail drop 启动时, 网络会产生 BufferBloat 问题, 而且这一问题会随着最大接收队列长度的增加而增加。但减小最大队列长度也会导致不停地重启, 从而影响传输效率, 还可能引发 TCP Incast 问题。因此需

要改进此方法来解 BufferBloat 问题。

（三）解决 BufferBloat 问题

在给定实验环境下，分别运行 3 种队列管理策略，作出延迟时间随带宽变化的曲线图如下所示：



可以看出, tail drop 的队列延时远高于 RED 和 CoDel。其中 CoDel 的性能表现最佳。

另外值得注意的是, tail drop 曲线一直出现“尖端”情况, 和 ppt 中稳定的一段峰值不同, 这是因为 Mininet 环境下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。因此结果存在差异。

五、实验总结

通过本次实验, 我了解了数据包队列的原理和工作方式, 以及如何根据环境设置数据包队列的大小。然后了解到 BufferBloat 问题的现象和原因, 知道了解决 BufferBloat 问题的三种方法: 改变队列大小、改进传输控制策略、改进队列管理策略。在实验环节, 通过改变队列

大小、使用不同队列管理策略，直观的认识这两种方法是如何解决 BufferBloat 问题的。