

# Lab3-4基于可靠UDP的实验数据分析

许健2013018

本次实验在已经完成 3 个版本的可靠 UDP 传输的基础上，通过调节实现代码中的参数获得不同的传输数据并对其进行分析。通过本次实验，可以对计算机网络中的传输算法、拥塞控制算法有更为清晰的了解和认识。

在本次实验中，使用的传输文件为课程组提供的图片1.jpg，图片大小为 1857352byte，重传时间设置为 100ms。每组测试中，采用控制变量法进行性能对比实验，测评指标为吞吐率(KB/s)，每一组都会测多次取平均值。

## 停等机制与滑动窗口的性能比较

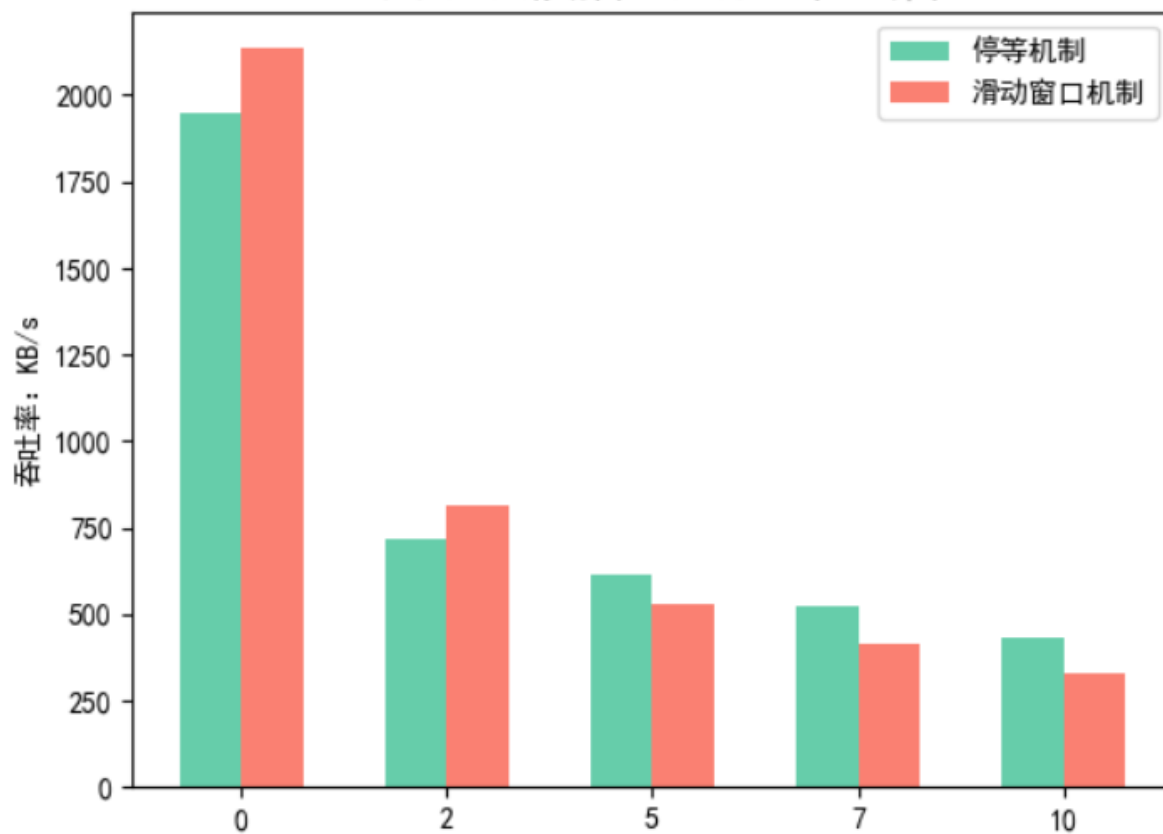
比较停等机制与滑动窗口机制对传输性能的影响，使用滑动窗口为 4 的情况进行对照组实验。

固定 router 的时延不变，动态调整丢包率，观察传输的性能。

滑动窗口	丢包率	时延	吞吐率
1（停等协议）	0%	0	1946
1	2%	0	715
1	5%	0	614
1	7%	0	526
1	10%	0	430
4	0%	0	2134
4	2%	0	815
4	5%	0	528
4	7%	0	415
4	10%	0	328

从直方图中可以看出，在丢包率较低时，滑动窗口的传输速率高于停等机制。但当丢包率较大时，滑动窗口的性能会下降，因为重发的包过多。

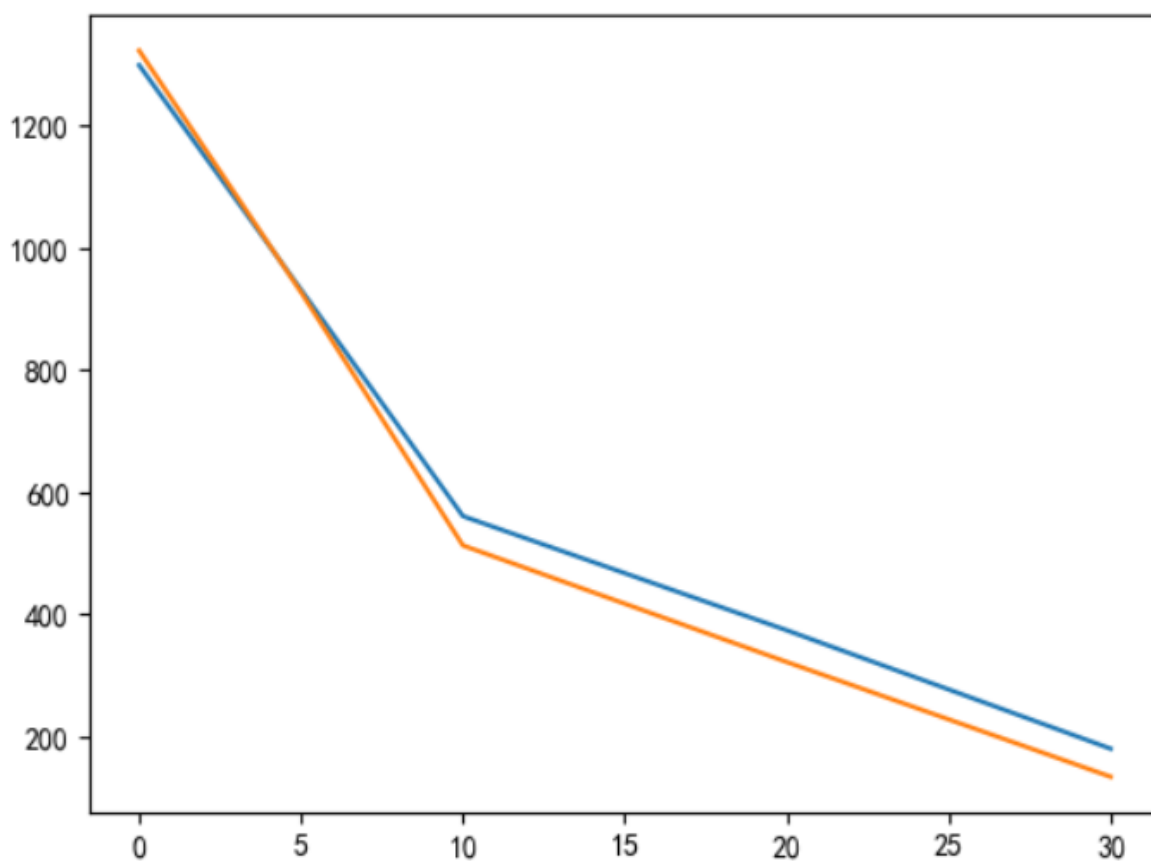
停等机制-滑动窗口机制吞吐率比对图



接下来，我们固定 router 的丢包率为 5%，通过改变传输时延来检测其性能变化。

设置延时为 0ms、5ms、10ms、20ms、30ms 来观察其变化对传输的影响。

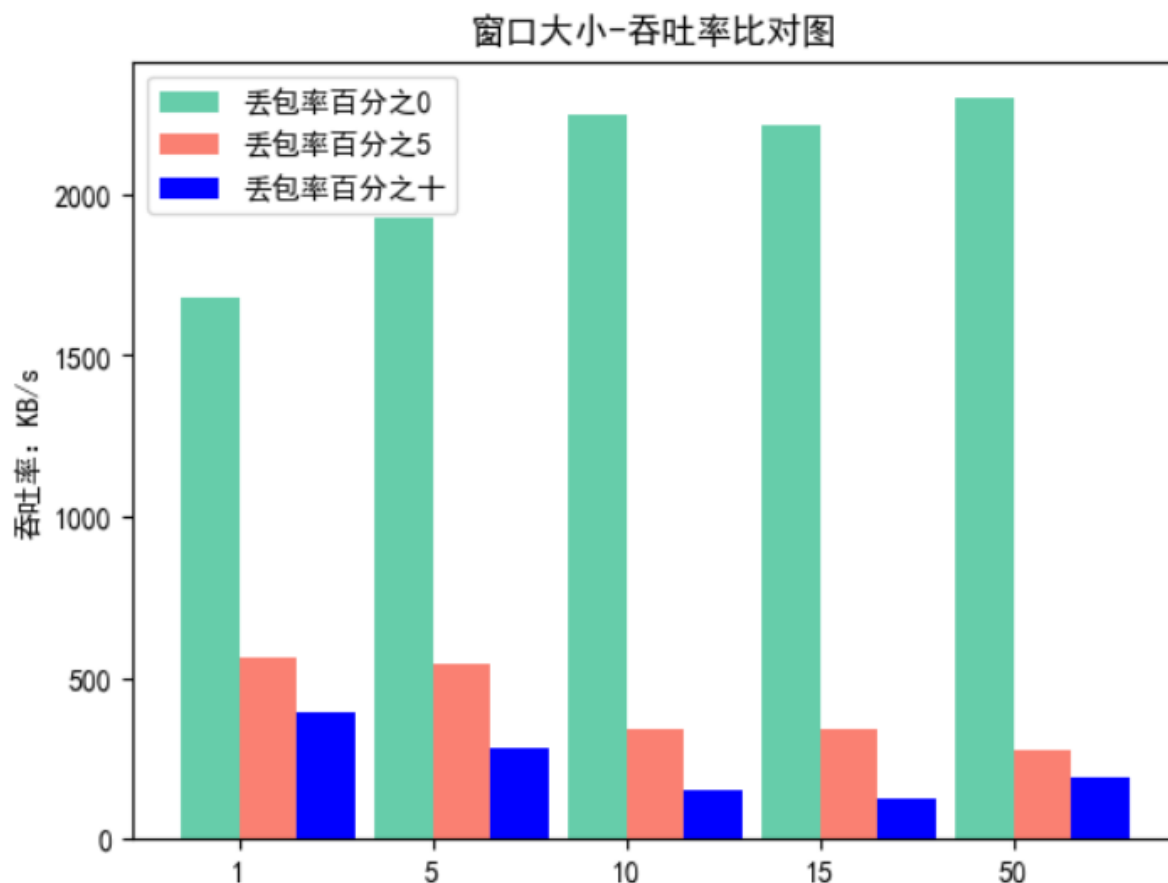
其中橙色的线代表滑动窗口机制，蓝色的线代表停止等待机制。



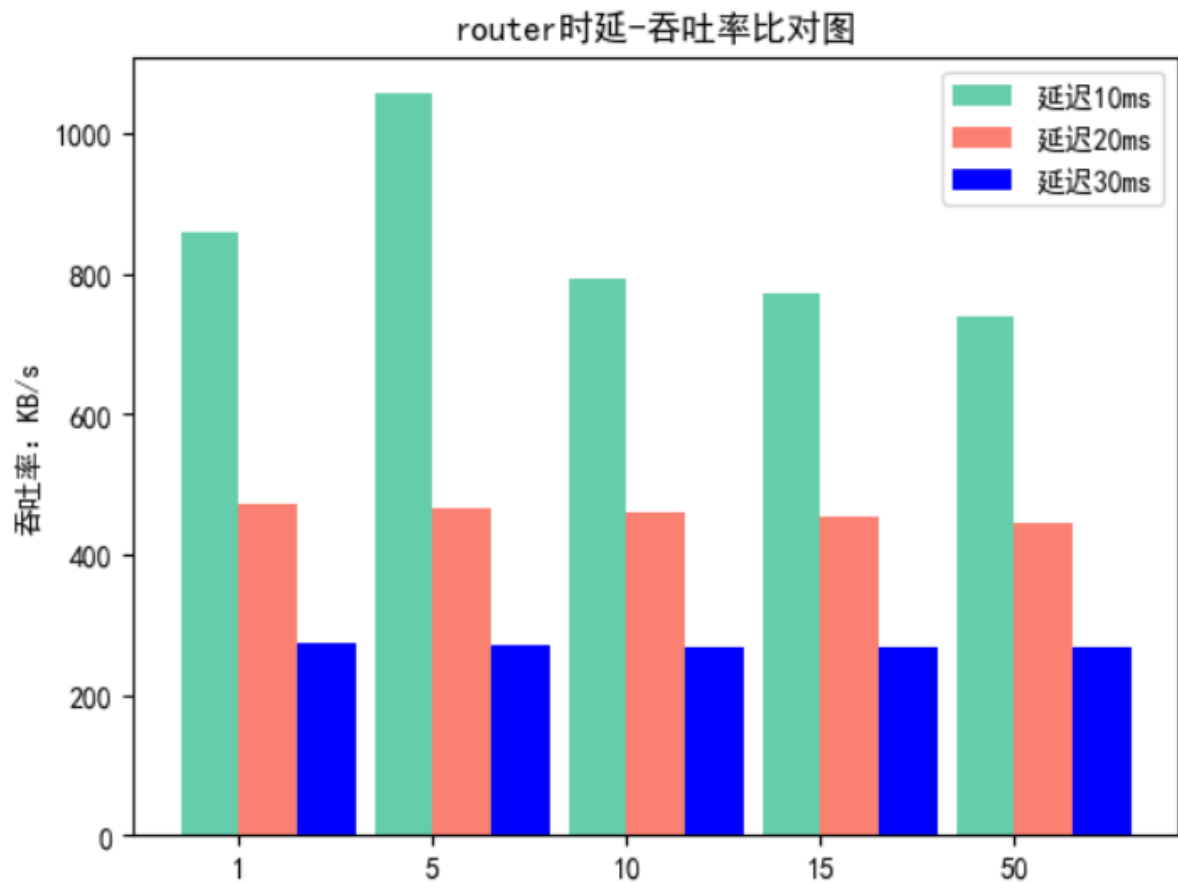
从固定 router 的丢包率改变时延以及固定 router 的时延改变丢包率的结果来看，滑动窗口和停等机制的性能差距似乎不是很大，甚至当丢包率较大时停等机制的传输吞吐率要大于滑动窗口 GBN。这是由于 GBN 的实现中需要同时具备收发两个线程，程序编写者设计的不够精妙，所以两个线程之间的交互对效率产生了一些影响；其次我们的测试只是在本机，并不是在实际大规模网络中，不能反映真实情况。

## 滑动窗口机制中不同窗口大小性能比较

在没有丢包的情况下，窗口的增大会带来传输速率的提升。在有丢包的情况下，随着窗口的增大，传输速率会有一定提升，但当窗口增长到一定大小时，其带来的性能增长减小，原因在于GBN重发的数据包量增大，拖慢发送速率。



固定 router 丢包率不变，改变 router 转发时延，测试不同窗口的吞吐率情况

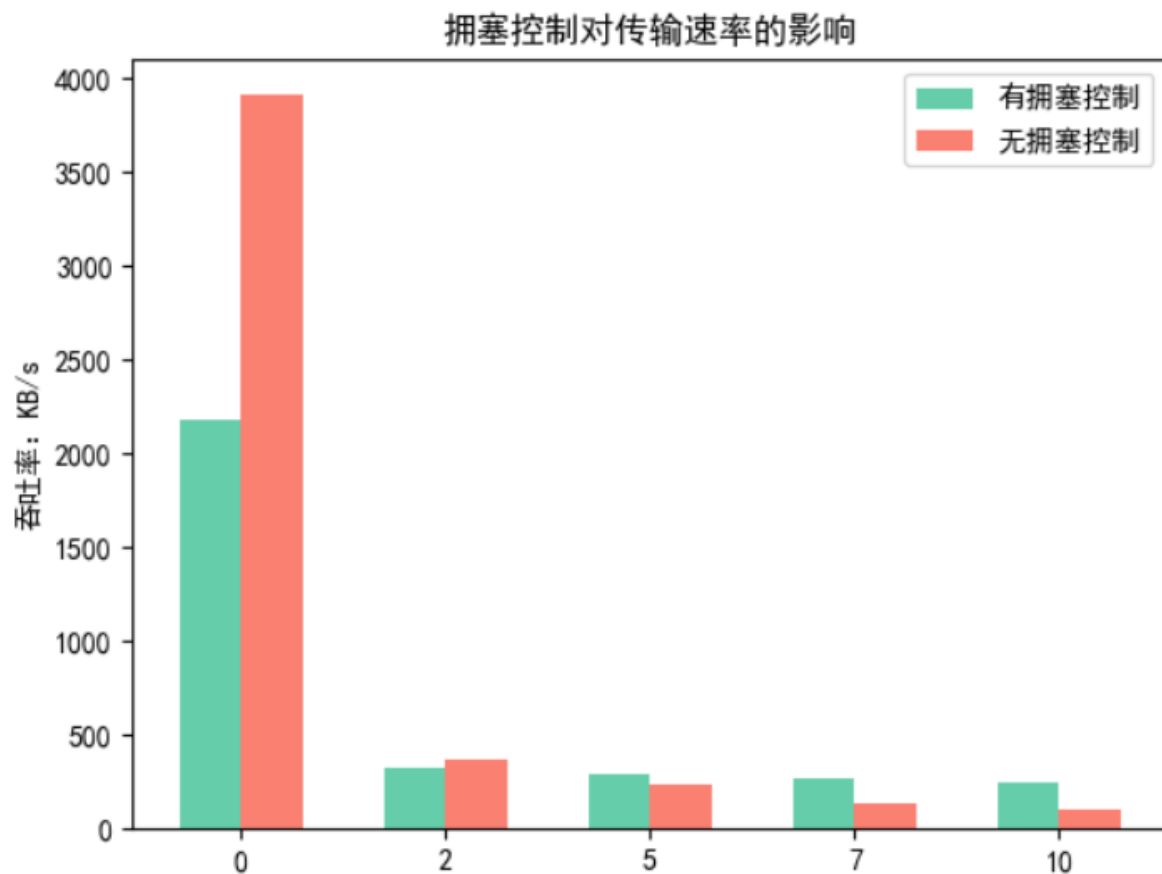


从两组对比实验中可以看出，丢包率从 0 增长到 5 时，当 router 的转发时延从 0ms 增长到 20ms 时，效率降低的最大。但是当我们逐步增大丢包率或者 router 转发时延时，效率降低的速率逐渐趋缓，这是因为分母增大的比率在减小。

## 拥塞控制对传输性能的影响

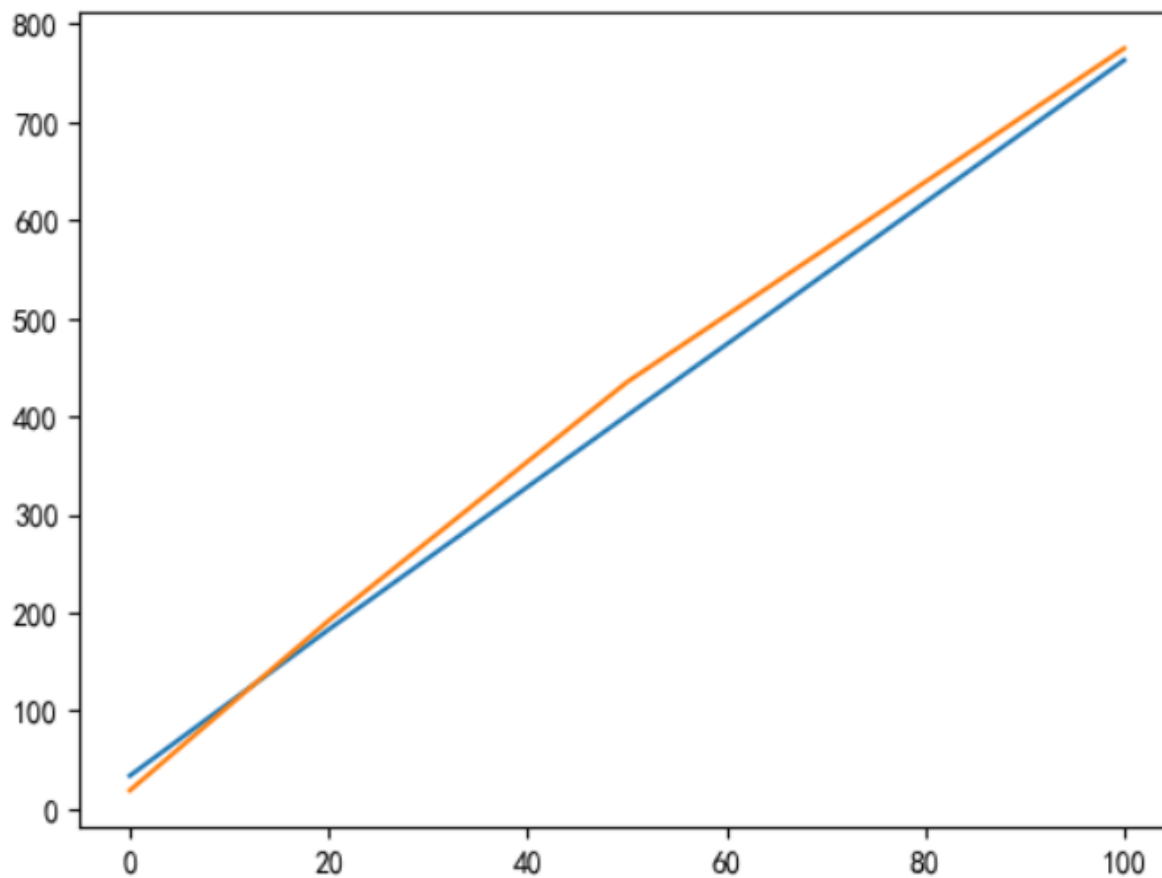
本次实验中，我们通过比较是否拥有拥塞控制对于传输性能的影响。使用控制变量法进行数据测量，没有拥塞控制时我们使用 GBN 实现的滑动窗口为 10 的数据进行对照组。

理论上讲，采用拥塞控制的效果应该更好，因为避免了传输超过接收方接收能力的数据包。但从实际测得结果，无拥塞控制的效率还要高于拥塞控制，可能与我们实验模拟网络有关，无法反映实际中的情况，其次实验测试的数据量偏小，传输时间很短，RENO还包括慢启动过程，反而导致效率下降。



接下来固定 router 的丢包率为 0%，然后改变 router 的转发时延，观察文件在传输时的性能。

蓝色的线是有拥塞控制时的传输时延变化折线，橙色的折线是没有拥塞控制时的时延折线。



## 总结

通过本次对可靠 UDP 传输程序性能的测试实验。我加深了对停等机制、GBN、滑动窗口、拥塞控制等概念的理解，也明白真实网络环境是复杂的，我们测试的只是最简单情况，无法反映出流量控制和拥塞控制在实际应用中在网络性能的提升。其次编写的程序并未完全发挥算法的性能，待优化地方有很多，可能某一个线程函数未处理好，就会导致程序性能的下降。在实际编程中，也要养成测试的好习惯，不光是为了防止程序出错，也是为了程序能够适用现实中复杂的场景，具有很好的鲁棒性。