个人信息

学号：1911410

姓名：付文轩

专业：信息安全

实验要求

基于给定的实验测试环境，通过改变延迟时间和丢包率，完成下面3组性能对比实验：

（1）停等机制与滑动窗口机制性能对比；

（2）滑动窗口机制中不同窗口大小对性能的影响；

（3）有拥塞控制和无拥塞控制的性能比较

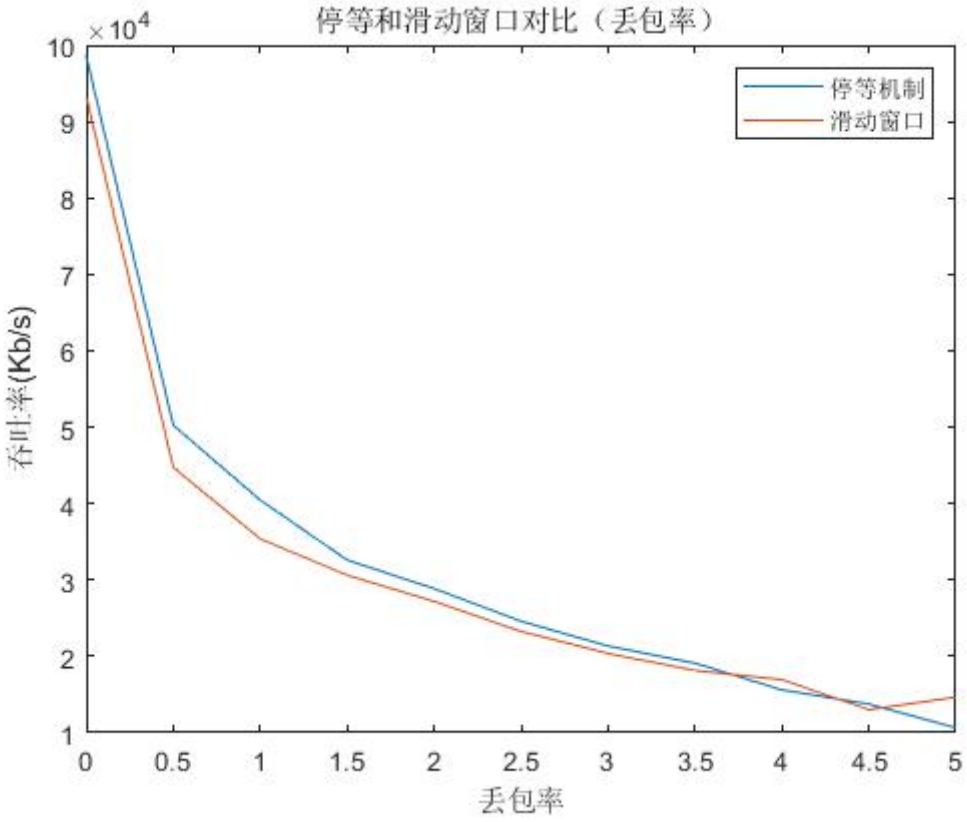
本次实验中我们以1.jpg作为样例进行性能测试

（1）停等机制和窗口机制对比

得到如下实验数据

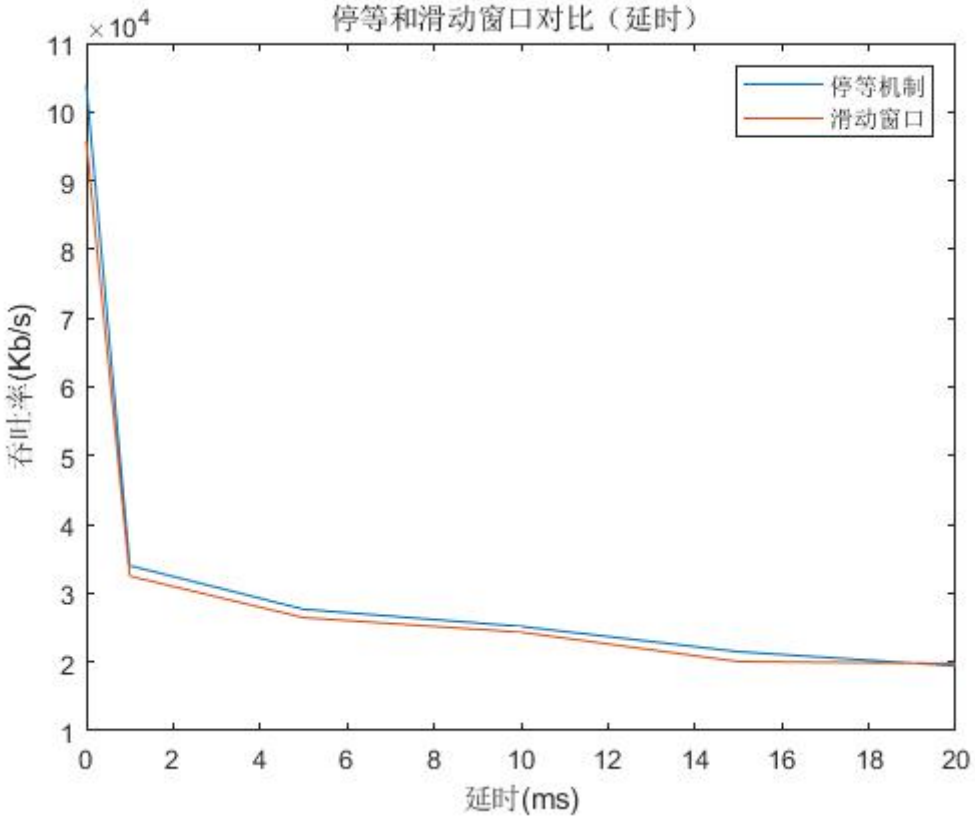
丢包率：

丢包率	0	0.5	1	1.5	2	2.5	3	3.5	4	4.5	5
停等机制	98596.1	50254.4	40456.4	32616.6	28867.8	24594.5	21346.4	19113.7	15605.4	13754.4	10651.5
滑动窗口	93287.4	44738.7	35389.4	30620.8	27201.5	23260.8	20375.9	18144.4	16946.7	13016.5	14656



延时：

延迟(ms)	0	1	5	10	15	20
停等机制	103962	34009.4	27667.2	25171.8	21494.6	19432.4
滑动窗口	95724.3	32489.6	26453.1	24323.2	20059.3	19783.2



结果分析：

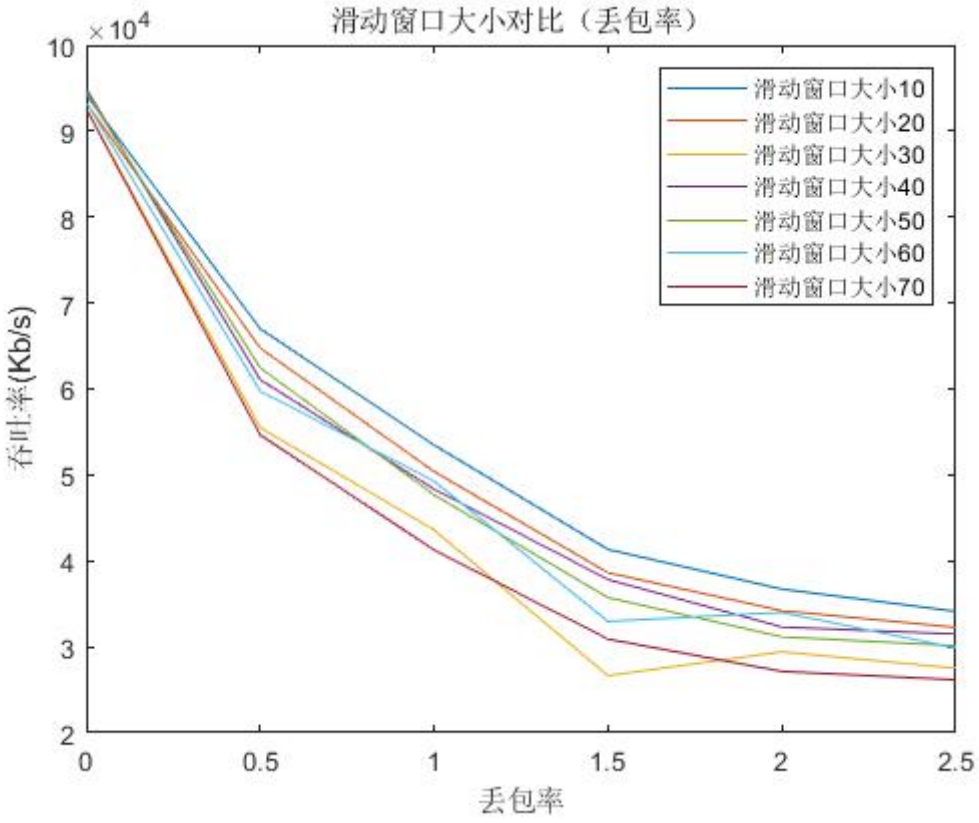
根据如上实验数据可以发现，停等机制和滑动窗口机制在0延时和0丢包率的情况下都非常的迅速，此时停等机制的性能要优于滑动窗口。当开始出现丢包和延时时，二者的效率都迅速下降。但是本次实验中比较奇怪的是：滑动窗口的性能居然不如停等机制。不过当丢包率和延时慢慢增大的时候我们发现一个趋势：滑动窗口降低的速率相对较慢，后期逐渐趋于缓和；而停等机制则是下降趋势相对较快。由此可见，当网络环境较差时，滑动窗口的抗压能力应该是比停等机制要更强的。

（2）滑动窗口机制中不同窗口大小对性能的影响

得到如下实验数据

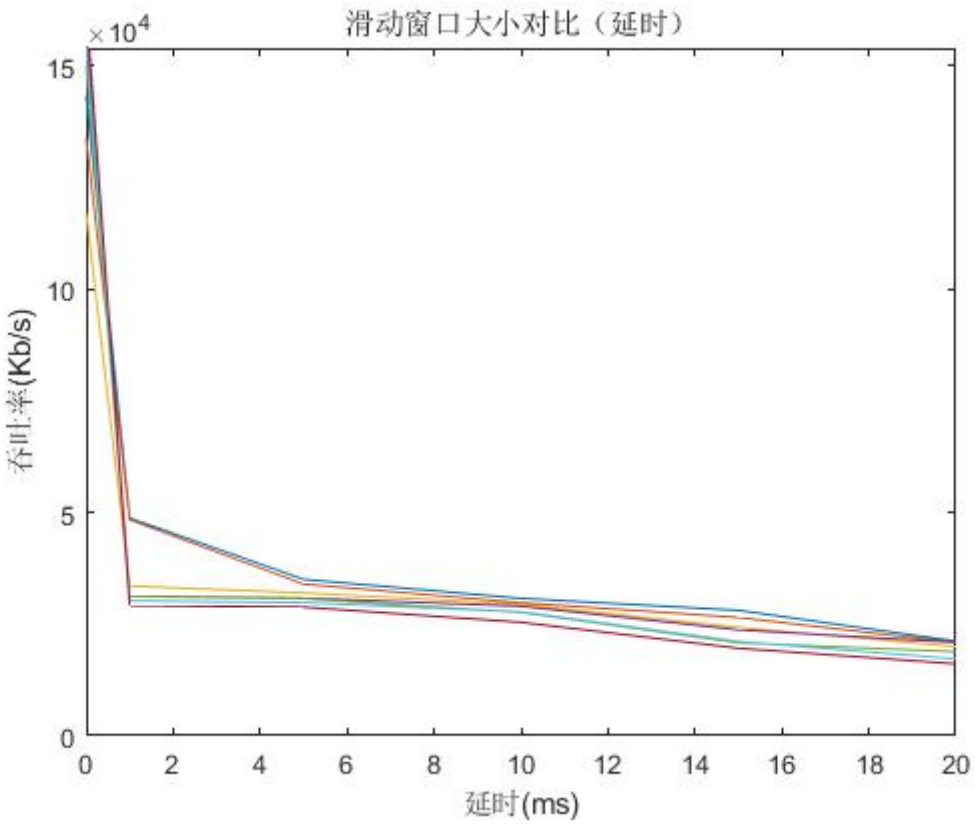
丢包率：

丢包率	滑动窗口10	滑动窗口20	滑动窗口30	滑动窗口40	滑动窗口50	滑动窗口60	滑动窗口70
0	94181.5	93287.4	94903.3	94517	93348.4	92636.1	92576
0.5	67008.9	64738.7	61077.9	62451.2	59757.5	55498.9	54648.6
1	53472.2	50389.4	48342.5	47678.4	49273.2	43587.5	41267.8
1.5	41326.8	38620.8	37824.1	35721.4	32953.8	26648.5	30895.5
2	36711.7	34201.5	32301.4	31156.2	34001.9	29416	27133.3
2.5	34124.2	32260.8	31482.4	30124.1	29857.4	27508.9	26168.9



延时：

延迟(ms)	滑动窗口10	滑动窗口20	滑动窗口30	滑动窗口40	滑动窗口50	滑动窗口60	滑动窗口70
0	143027	133793	117269	156964	149907	154663	163600
1	48748.1	48379.9	33532.9	31306.5	31229.1	30234.9	29168.9
5	35023.6	33973.4	31997.7	30845.9	30710.2	29821.8	28809
10	30758.5	29801.6	29454.7	29006.2	27637.1	27765.2	25416.4
15	28089.4	26487.8	24275.3	23728.3	20805.1	21106.8	19600.6
20	21180.9	20876.2	19969.5	20900.8	18825.8	17211.3	16120.3



结果分析：

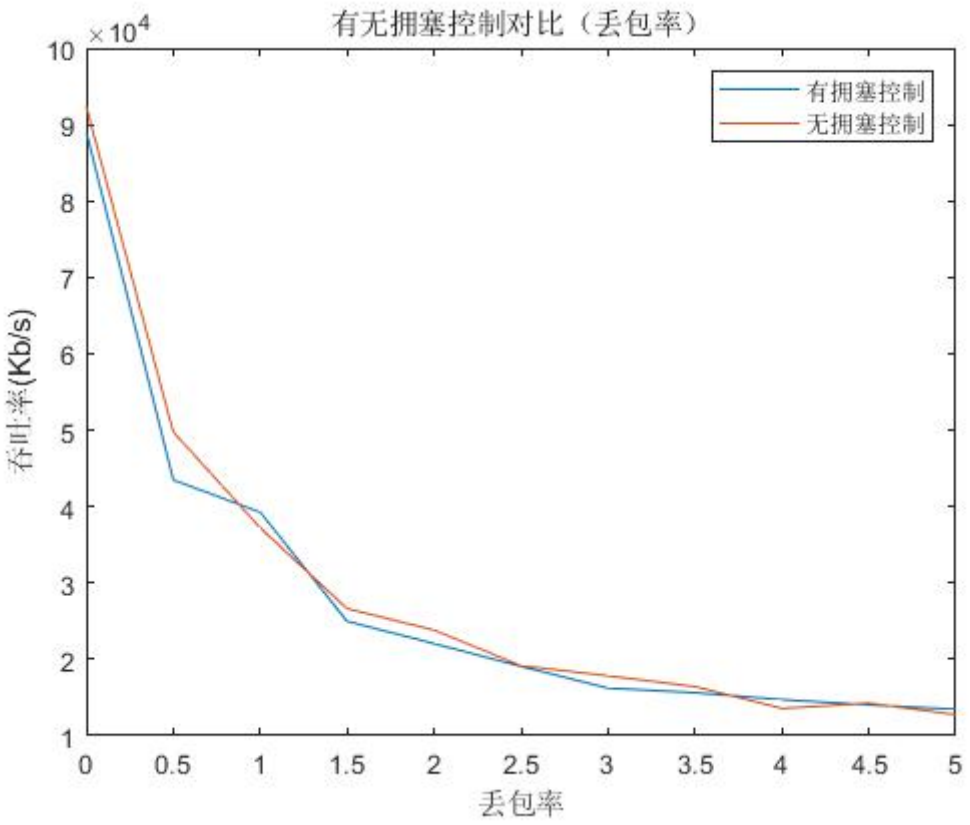
根据实验数据可以发现，当滑动窗口越大，其传输效率反而越低，其下降的趋势都是逐渐趋于平缓的。这里我猜测可能是因为我们程序或者是路由器的原因导致出现这样的问题。或者是当窗口越大，每次出现丢包或者超时的时候需要重传的分组个数越多，所以相对消耗的时间就会越长，从而导致出现窗口越大效率越低的情况。

（3）有拥塞控制和无拥塞控制的性能比较

得到如下实验数据

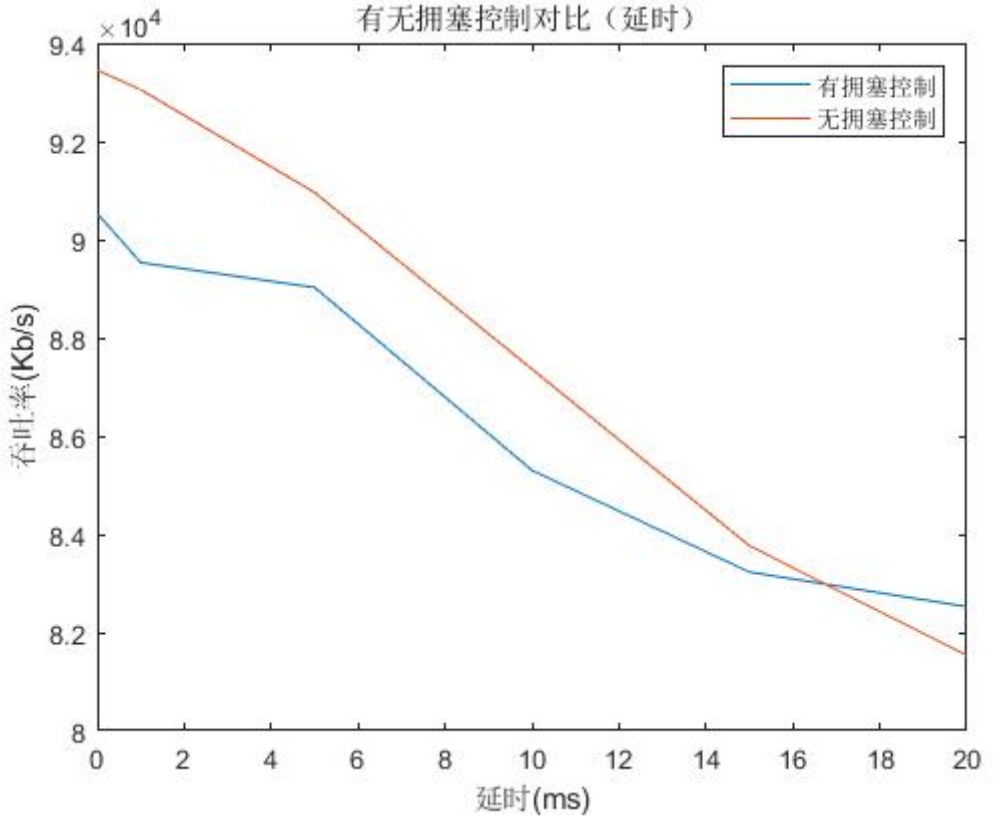
丢包率：

丢包率	0	0.5	1	1.5	2	2.5	3	3.5	4	4.5	5
有拥塞控制	89044.1	43485.3	35261.2	24989.2	22058.6	19083.1	16241.3	15635.2	14725.3	14041.7	13463.4
无拥塞控制	92354.3	49782.1	37214.6	26632.5	23834.3	19163.2	17847.2	16425.9	13574.2	14261.6	12765.5



延时：

延迟(ms)	0	1	5	10	15	20
有拥塞控制	90531.1	89532.2	89025.2	85301.3	83231.4	82526.3
无拥塞控制	93451.2	93052.3	90959.1	87361.4	83766.3	81531.4



结果分析：

整体结果上来看，无拥塞控制的性能要优于有拥塞控制的性能。但两者下降趋势的比较之下我们可以发现：有拥塞控制的曲线相对较为平缓，无拥塞控制的曲线则较为陡峭。基于此我们可以猜测，无拥塞控制时，发送速度会不受到限制从而疯狂增长，当网络状态良好时是可以接受的，那么此时自然传输速度就会快；而拥塞控制是有一定的自我约束能力，他不会让自己的传输速度疯狂增长，所以在网络状态良好时反而表现不如没有拥塞控制。但是当网络状态逐渐变差时，无拥塞控制受到网络状态的影响就会更加明显，同时也会更加容易影响到网络状态从而使得网络状态更差；而有拥塞控制在良好的自我约束情况下，不会对网络状态造成太大影响，从而能够更容易的适应当前网络状态，所以在后期网络状态越来越差时，有拥塞控制的性能会趋于平缓，并且优于无拥塞控制。