

# 实验3-4：性能对比实验

学院：网络空间安全学院

专业：密码科学与技术

学号：2112155

姓名：梁婧涵

## 实验要求

实验3-4：基于给定的实验测试环境，通过改变延时和丢包率，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（累计确认和选择确认两种情形）；（3）滑动窗口机制中相同窗口大小情况下，累计确认和选择确认的性能比较。

## 实验过程

该报告中数据图表绘制由chartcool图表在线绘制而成

### 1、停等机制与滑动窗口机制性能对比

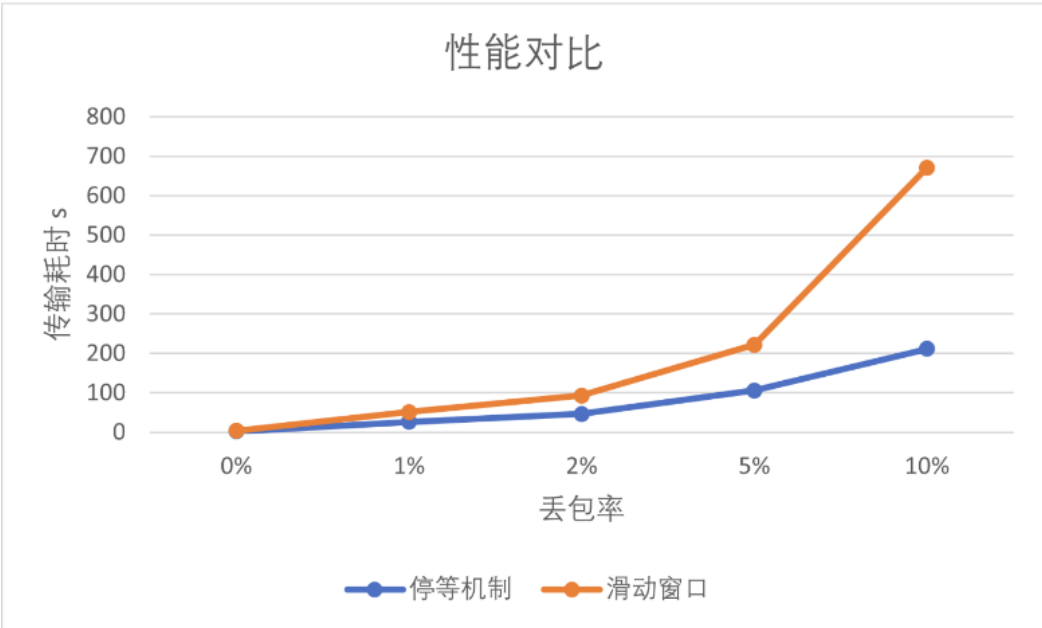
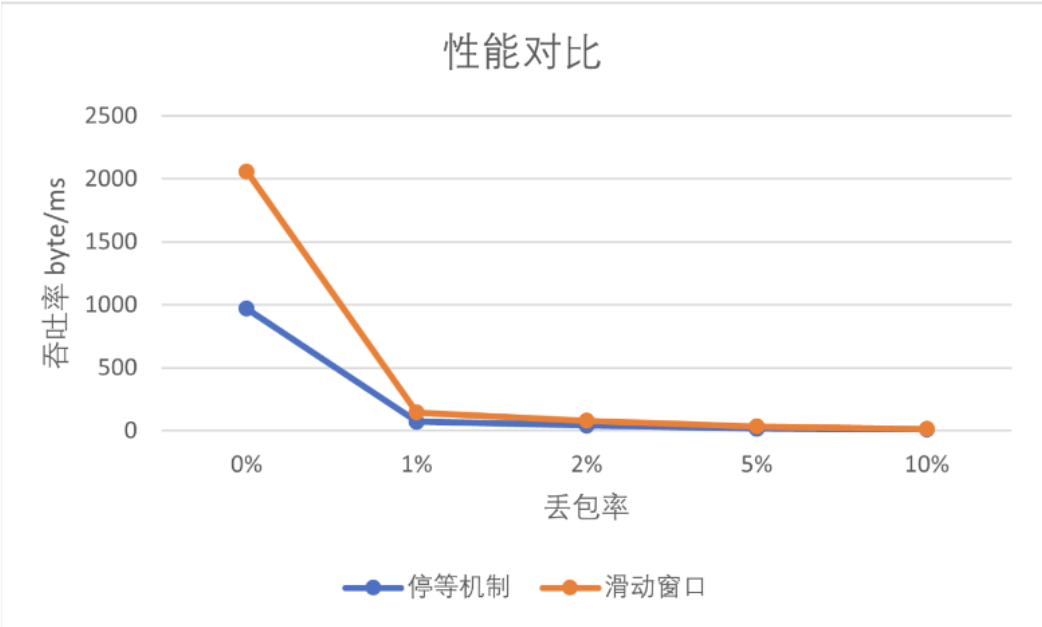
#### 1.1 实验数据

滑动窗口统一使用滑动窗口为6、超时重传>0.5s，动态调整传输性能。

实验使用传输文件1.jpg，图片大小1857352byte

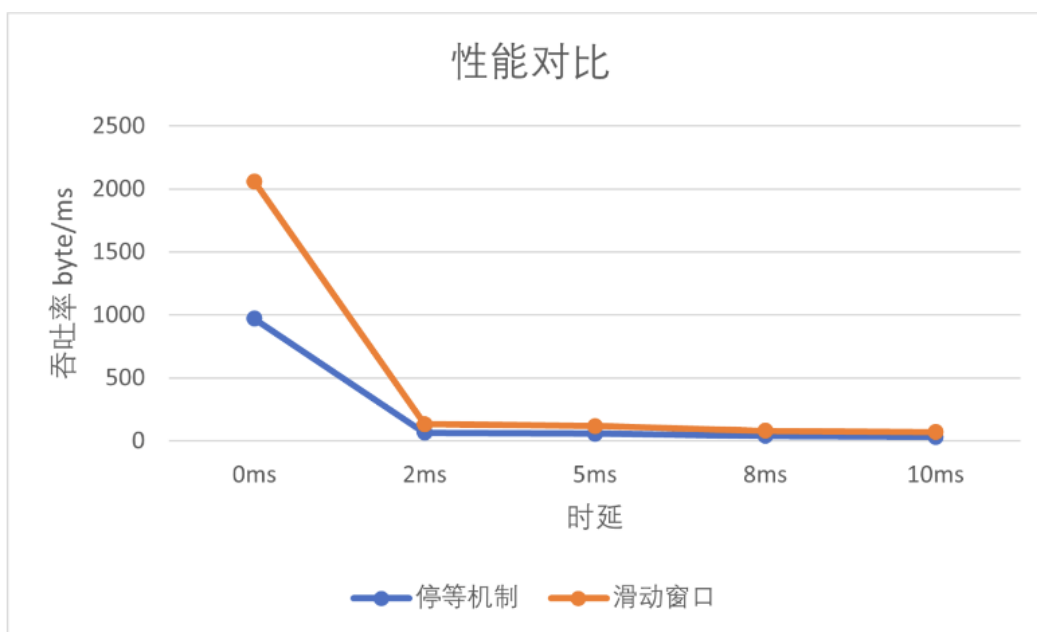
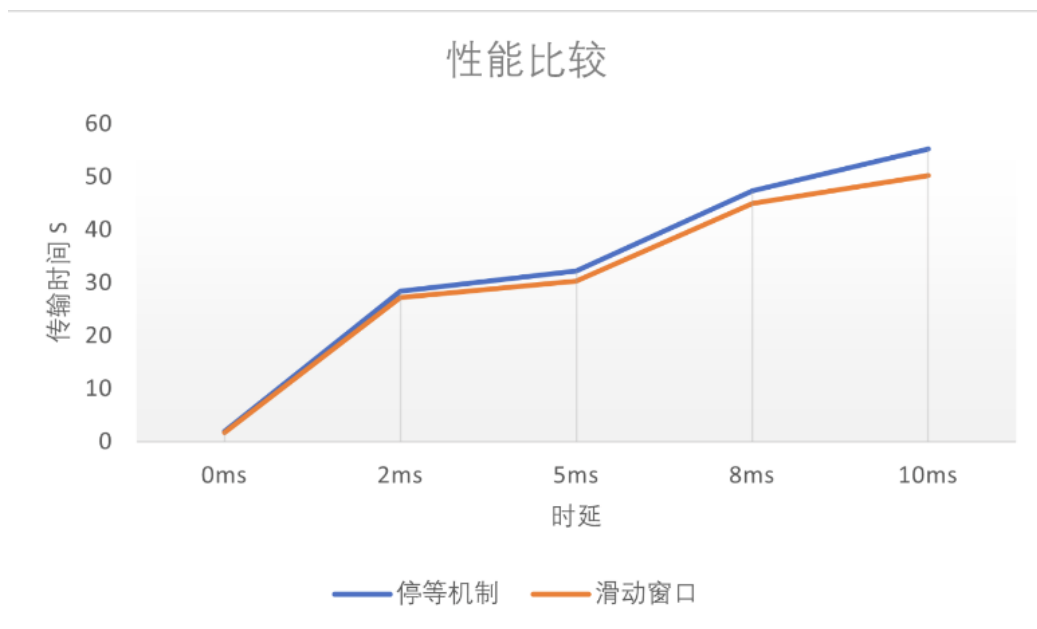
延时为0更改丢包率：

机制\丢包率	0%	1%	2%	5%	10%
停等机制-传输时间s	1.914	26.386	46.153	105.826	211.202
停等机制-吞吐率byte/ms	970.4038	70.3916	40.2434	17.551	8.7942
滑动窗口-传输时间s	1.704	25.211	47.028	115.73	460.2
滑动窗口-吞吐率byte/ms	1090	73.6723	39.4946	16.049	4.0359



丢包率0改变延时:

机制\延时	0ms	2ms	5ms	8ms	10ms
停等机制-传输时间s	1.914	28.374	32.245	47.349	55.249
停等机制-吞吐量byte/ms	970.4038	65.4597	57.6013	39.2268	33.6179
滑动窗口-传输时间s	1.704	27.187	30.356	44.930	50.238
滑动窗口-吞吐量byte/ms	1090	68.3176	61.1856	41.3388	36.9711



## 1.2 对比结果

- 1、停等机制和滑动窗口的传输性能随丢包率提高而降低
- 2、没有丢包或丢包率低时，滑动窗口性能优于停等机制（滑动窗口允许发送多条同时等待ack，减少RTT在总时间的影响）
- 3、丢包率较大时，滑动窗口性能低于停等机制（GBN在丢包较大数据重传严重，网络大量数据报拥堵）
- 4、延时对停等机制和滑动窗口传输性能影响没有丢包率改变明显，但延时时滑动窗口性能更好（停等机制需要每条数据包单独等待时延和RTT，窗口可以同时等待）

## 2、滑动窗口机制中不同窗口大小对性能的影响

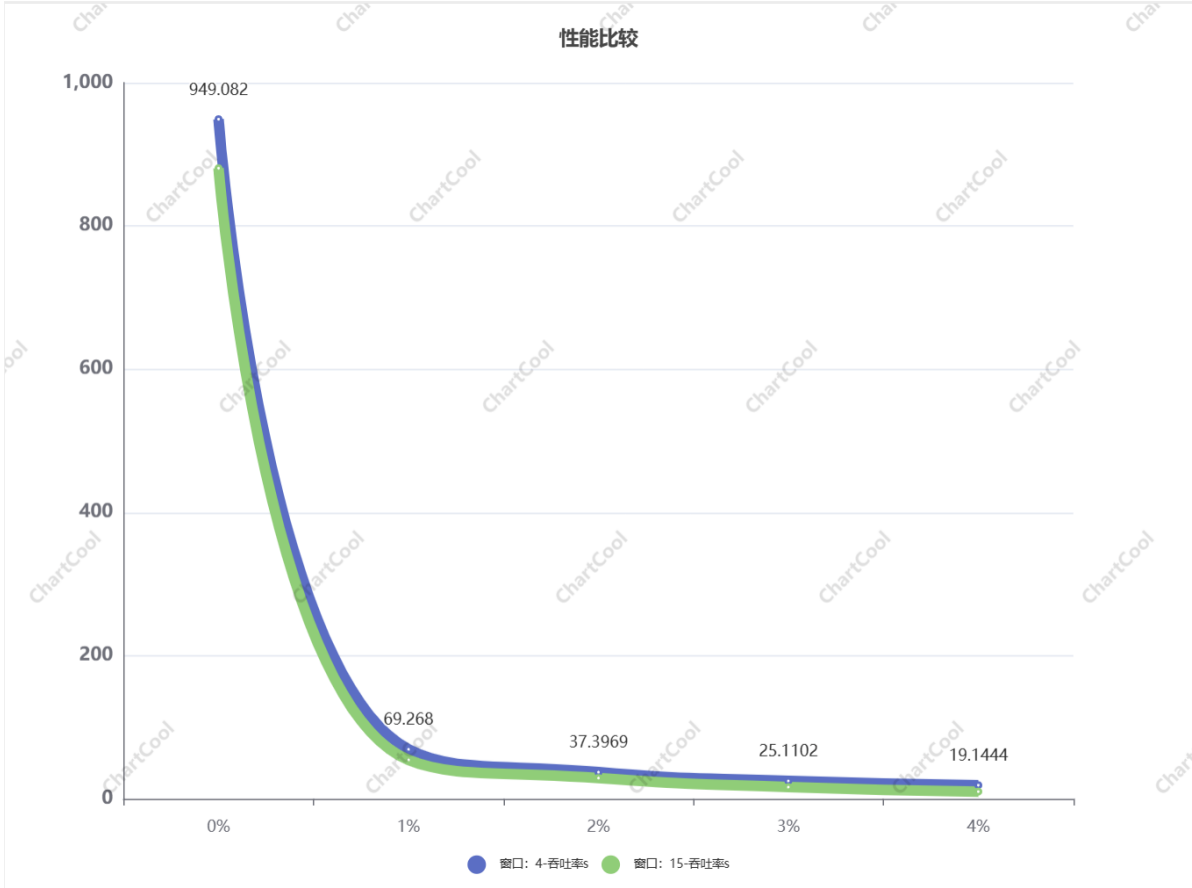
### 2.1 累计确认

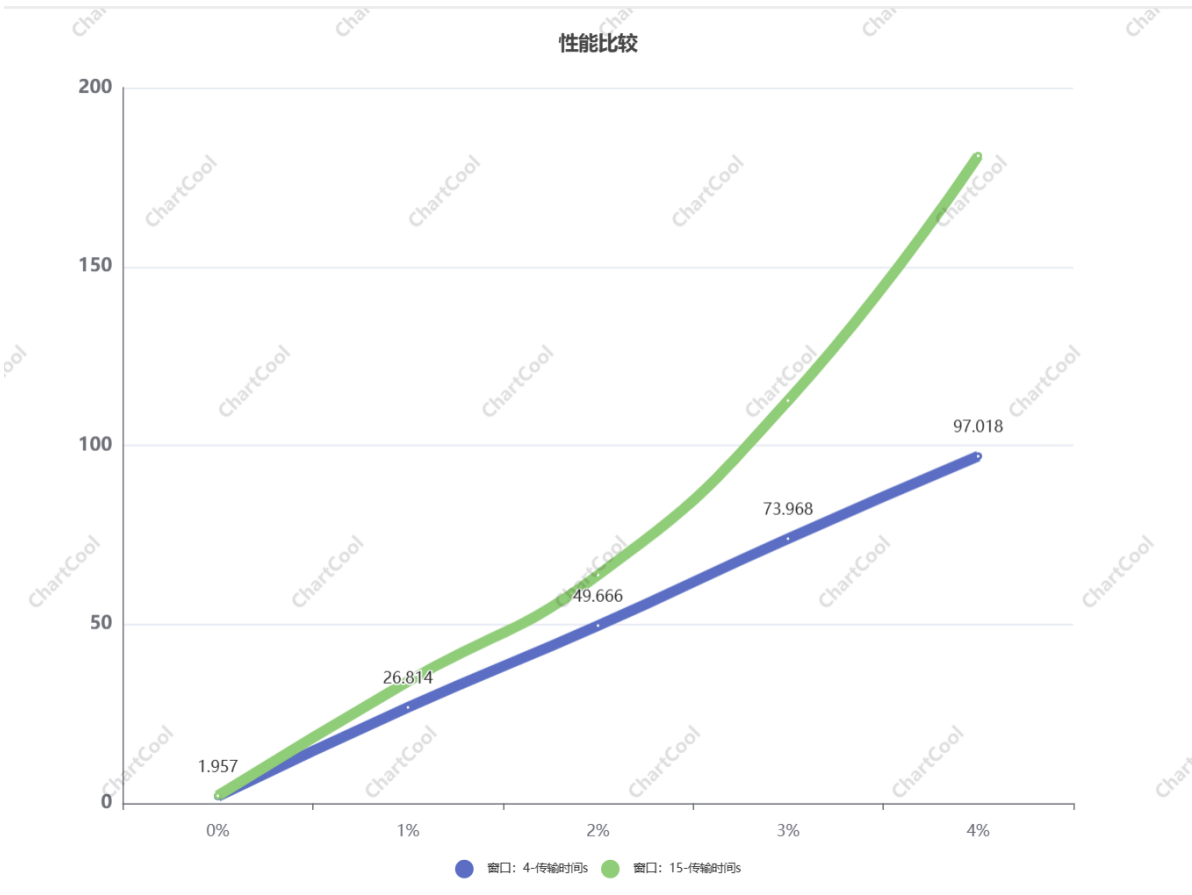
#### 2.1.1 比较结果

选择窗口4、15：

延时为0更改丢包率：

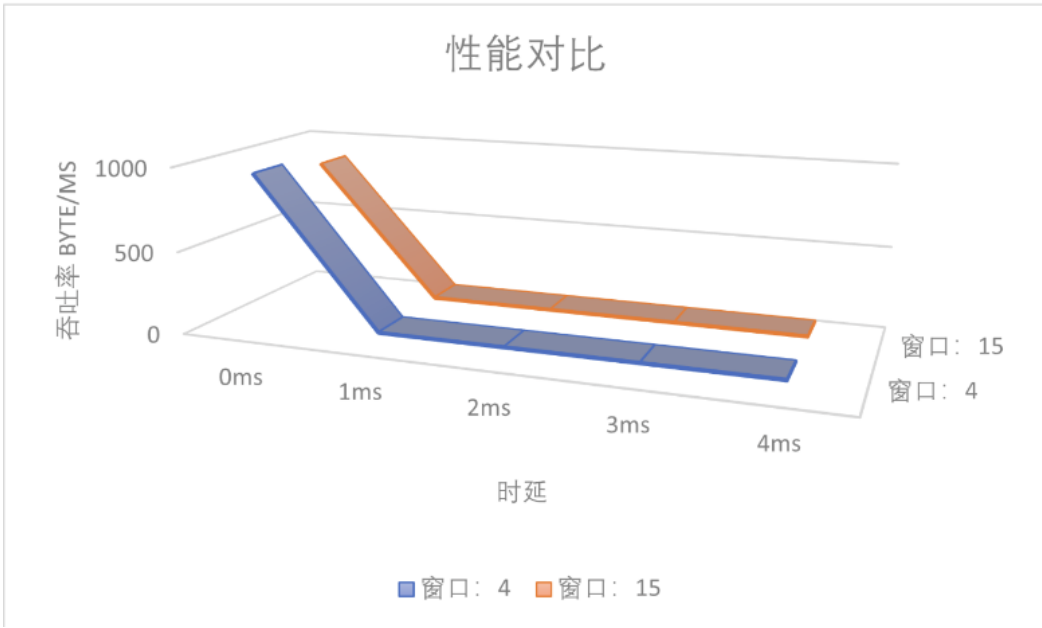
机制\丢包率	0%	1%	2%	3%	4%
窗口：4-传输时间s	1.957	26.814	49.666	73.968	97.018
窗口：4-吞吐率byte/ms	949.082	69.268	37.3969	25.1102	19.1444
窗口：15-传输时间s	2.109	34.065	63.788	112.557	180.971
窗口：15-吞吐率byte/ms	880.679	54.5238	29.1176	16.5014	10.2633

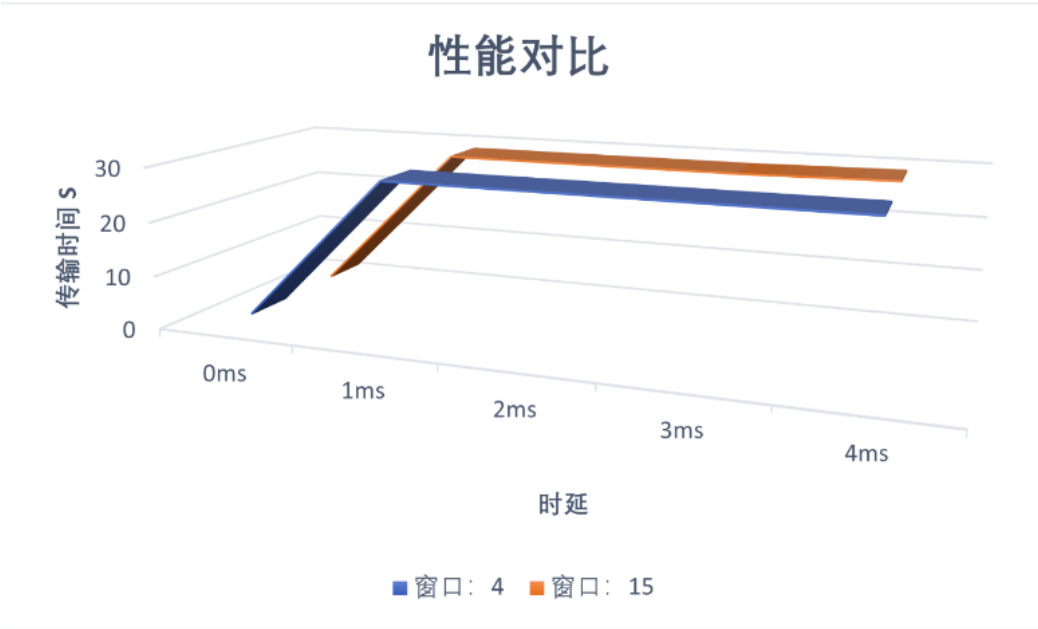




丢包率为0更改延时:

机制\延时	0ms	1ms	2ms	3ms	4ms
窗口: 4-传输时间s	1.957	28.767	28.787	28.793	28.826
窗口: 4-吞吐率byte/ms	949.082	64.5654	64.5205	64.5071	64.4333
窗口: 15-传输时间s	2.109	28.747	28.762	28.787	29.177
窗口: 15-吞吐率byte/ms	880.679	64.6103	64.5766	64.5205	63.6581





2.1.2 结论

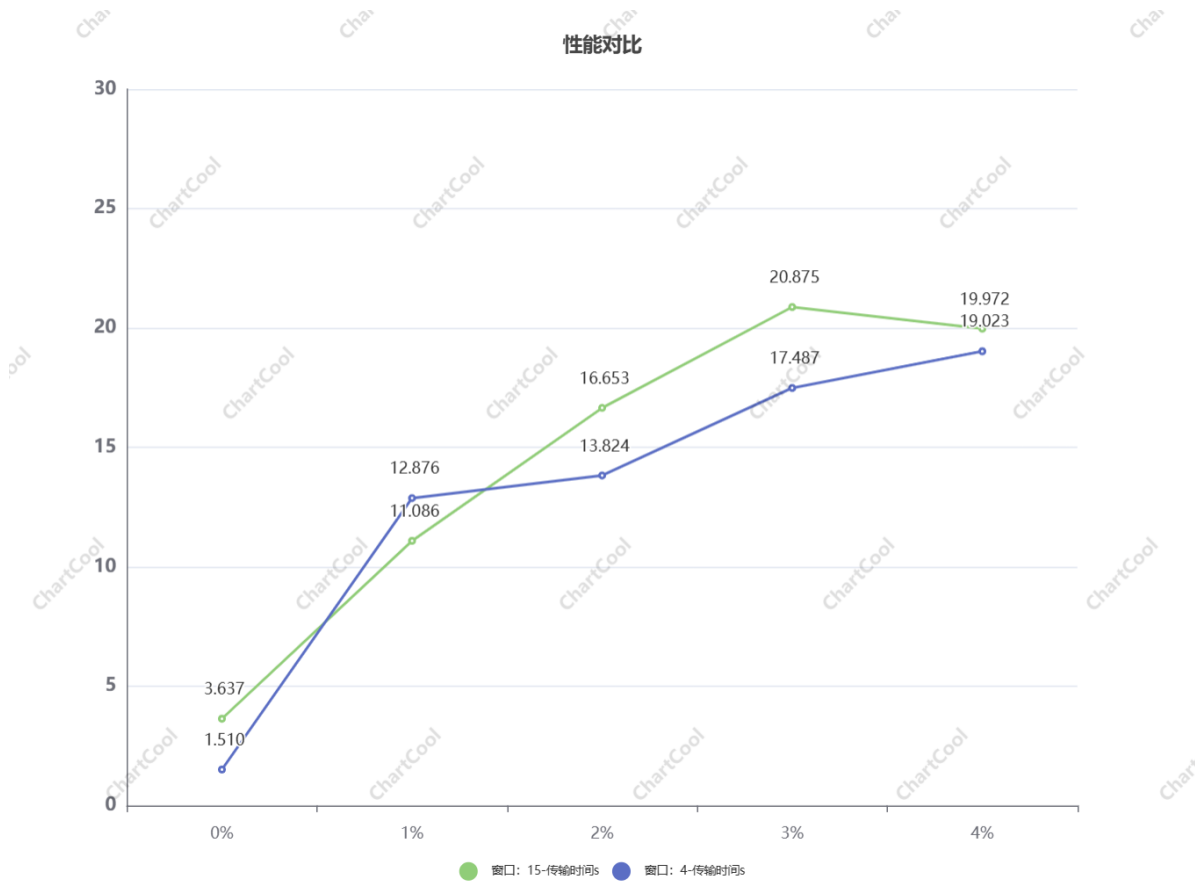
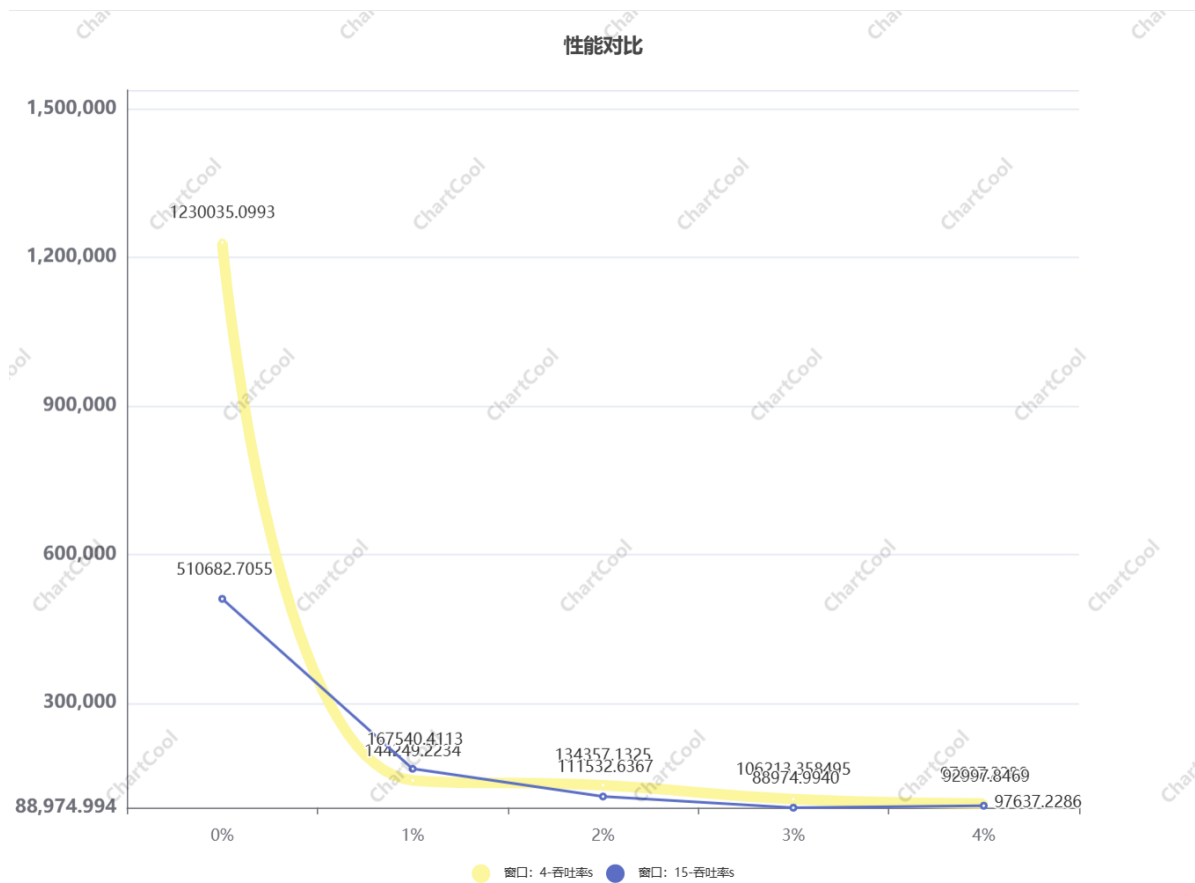
- 1、不同滑动窗口大小在丢包率和时延的改变下，性能较低变化整体一致
- 2、丢包率改变增大时，滑动窗口大小越大性能下降越快、降低更多（大的窗口重传代价大）  
丢包率为0或较低时，大窗口性能更高
- 3、时延改变时，滑动窗口大小区别在这里影响不大，降低。

2.2 选择确认

选择窗口4、15：

延时为0更改丢包率：

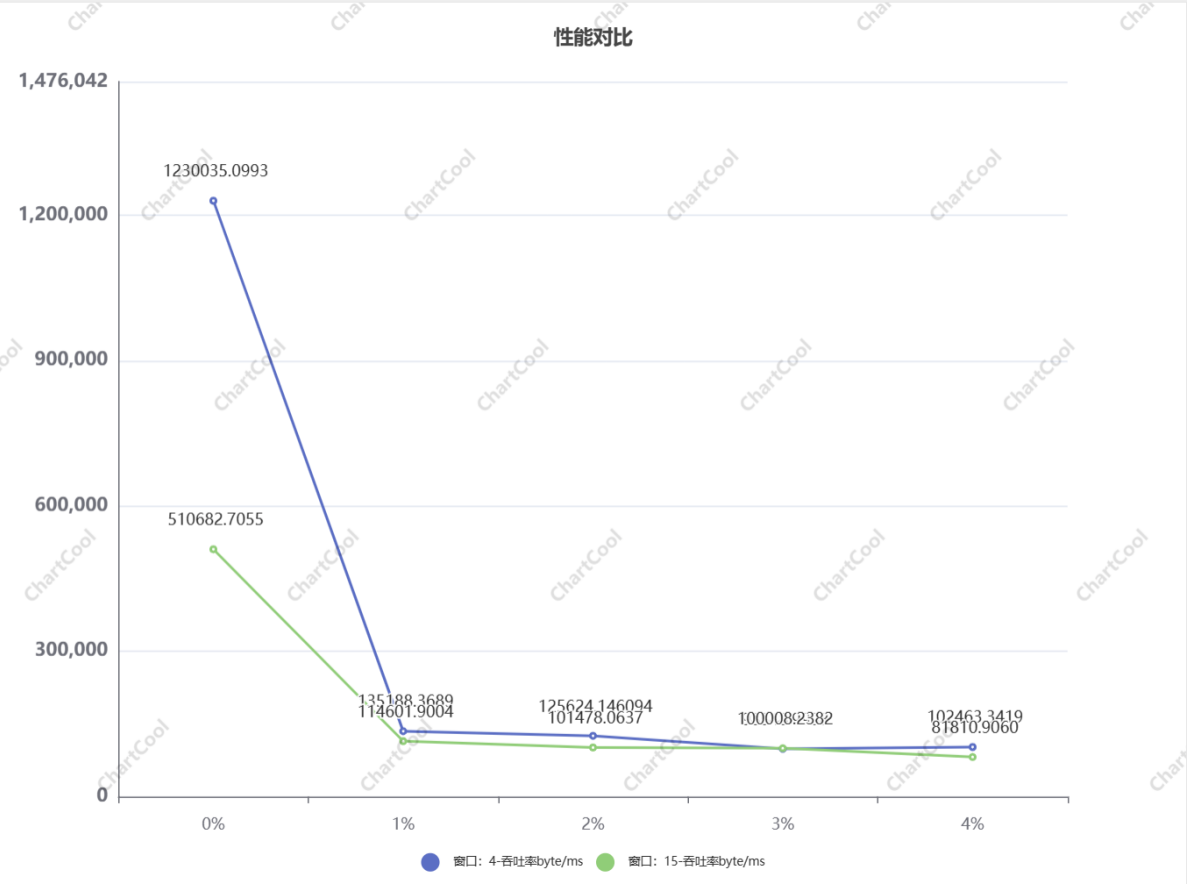
机制\丢包率	0%	1%	2%	3%	4%
窗口：4-传输时间s	1.510	12.876	13.824	17.487	19.023
窗口：4-吞吐量byte/ms	1230035.0993	144249.2234	134357.1325	106213.358495	97637.2286
窗口：15-传输时间s	3.637	11.0860	16.653	20.875	19.972
窗口：15-吞吐量byte/ms	510682.7055	167540.4113	111532.6367	88974.9940	92997.8469



丢包率为0更改延时:

机制\丢包率	0ms	1ms	2ms	3ms	4ms
窗口: 4-传输时间s	1.510	13.739	14.785	18.847	18.127

机制\丢包率	0ms	1ms	2ms	3ms	4ms
窗口: 4-吞吐率byte/s	1230035.0993	135188.3689	125624.146094	98548.9998	102463.3419
窗口: 15-传输时间s	3.637	16.207	18.303	18.572	22.703
窗口: 15-吞吐率byte/s	510682.7055	114601.9004	101478.0637	100008.238208	81810.9060







## 2.2.2 结论

- 1、不同滑动窗口大小在丢包率和时延的改变下，性能较低变化整体一致
- 2、丢包率改变增大时，滑动窗口大小越大性能下降越快、降低更多（大的窗口重传代价大）
- 3、时延改变时，滑动窗口大小影响不大，降低

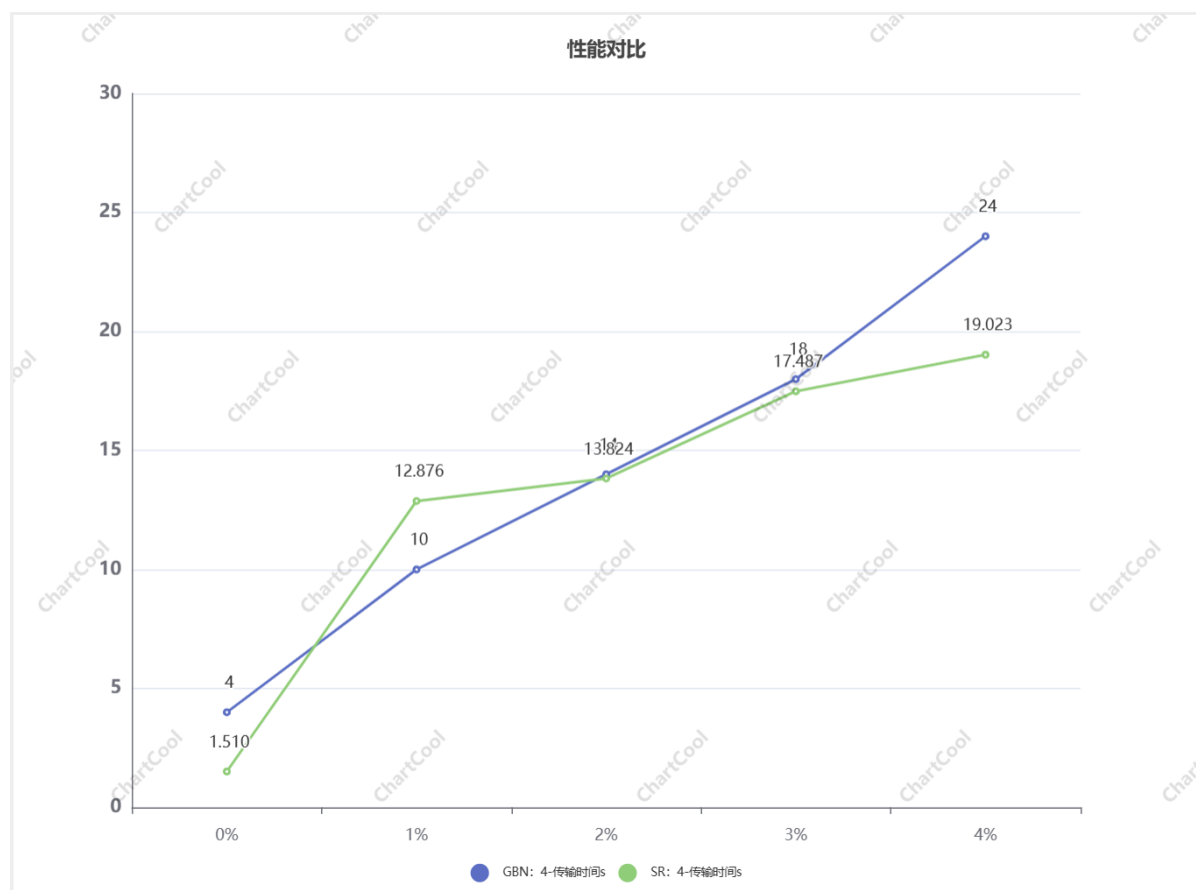
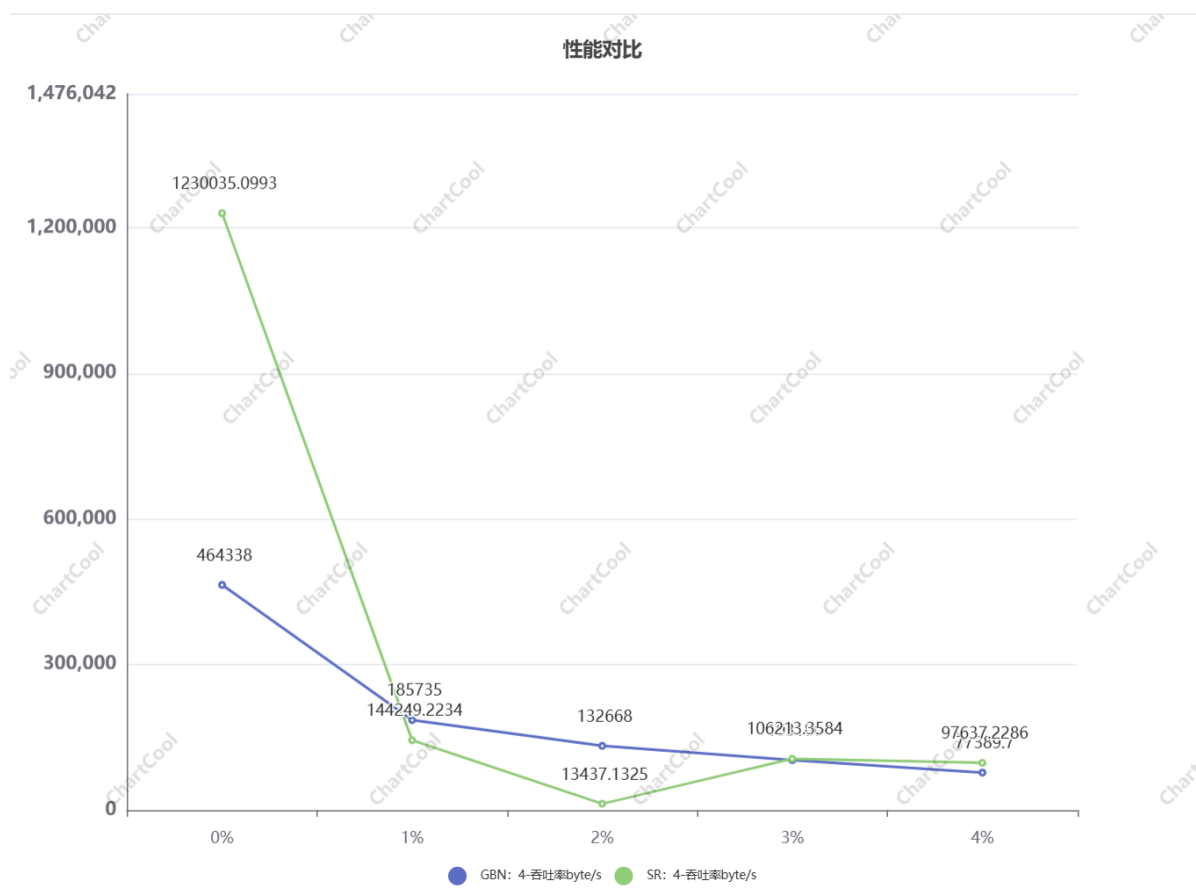
## 3、滑动窗口机制中相同窗口大小情况下，累计确认和选择确认的性能比较

### 3.1 比较结果

选择窗口4,#define MSL 500,#define max\_size 4096

延时为0更改丢包率：

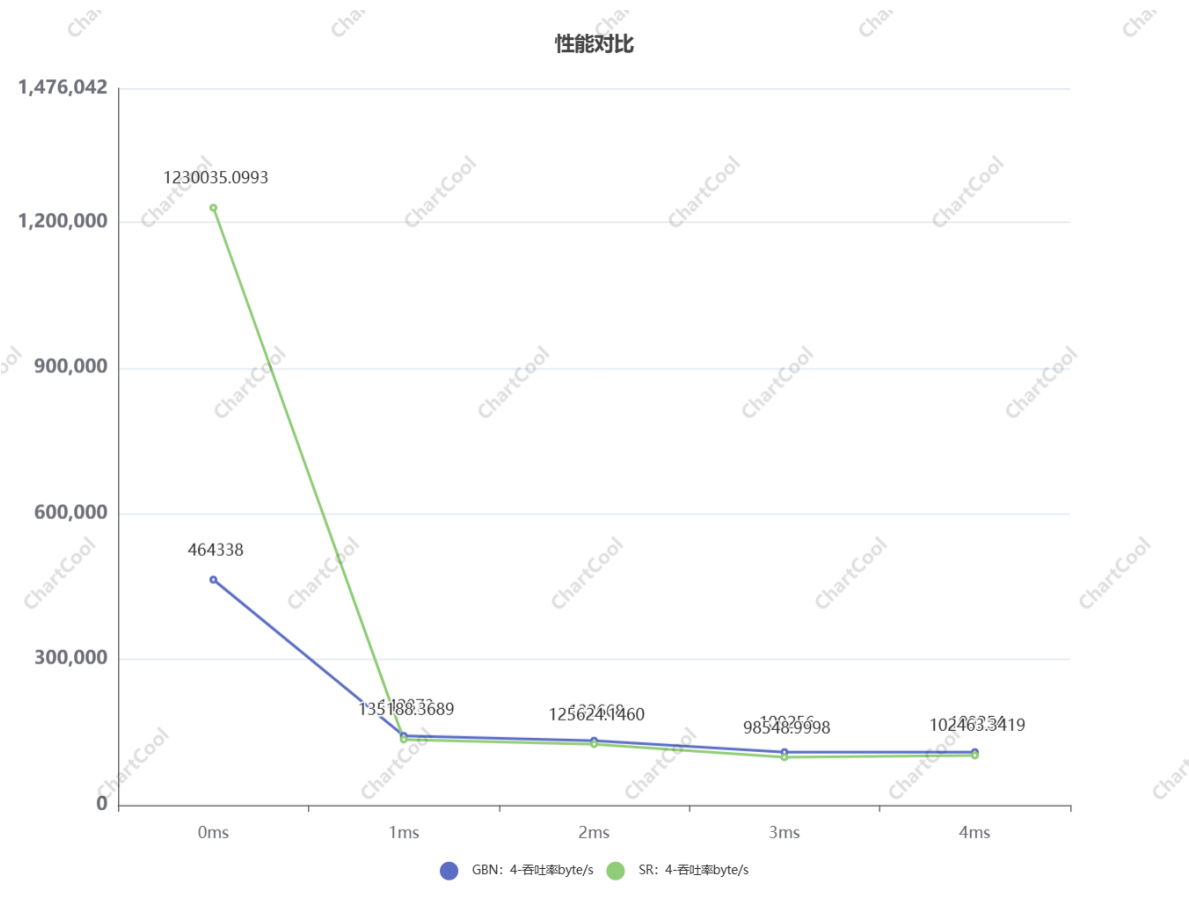
机制\丢包率	0%	1%	2%	3%	4%
GBN: 4-传输时间s	4	10	14	18	24
GBN: 4-吞吐率byte/s	464338	185735	132668	103186	77389.7
SR: 4-传输时间s	1.510	12.876	13.824	17.487	19.023
SR: 4-吞吐率byte/s	1230035.0993	144249.2234	134357.1325	106213.358495	97637.2286



丢包率为0更改延时:

机制\丢包率	0ms	1ms	2ms	3ms	4ms
GBN: 4-传输时间s	4	13.896	14.898	17.987	17.998

机制\丢包率	0ms	1ms	2ms	3ms	4ms
GBN: 4-吞吐量 率byte/s	464338	142873	132668	109256	109256
SR: 4-传输时 间s	1.510	13.739	14.785	18.847	18.127
SR: 4-吞吐量 byte/s	1230035.0993	135188.3689	125624.146094	98548.9998	102463.3419





### 3.2 结论

- 1、不同丢包率/延时时间，SR机制优于GBN机制，在延时越大、丢包率越大时，GBN机制虽然能稳定吞吐率在较高水平，但传输时间较长。
- 2、GBN协议中的一个最大的问题就是，当窗口大小N非常大时（例如为1000的话），而最开始的分组0错误了，就需要重传之后的999个分组，因此造成了时间上的损失，延时增加
- 3、SR协议而言，发送方和接收方并不是总能看到相同的结果，这也就说明两者的窗口不总是一致的。而SR协议会面临到的一个最重要的问题就是面临有序序号范围的现实时，发送方和接收方的窗口不同步会产生严重后果。这样的后果会导致接收方无法判断出该分组是一次重传还是下一个新的分组的，因而就SR的窗口不能够很大
- 4、SR机制更能适应延迟高、丢包率高的情况

## 4、总结

不同丢包率/延时时间，SR机制优于GBN机制，在延时越大、丢包率越大时，GBN机制虽然能稳定吞吐率在较高水平，但传输时间较长

- 停等机制：网络顺畅时可以将发送端和接收端性能发挥极致，实际性能受网络时延和丢包影响大
- GBN机制：接收端需要维护队列对发送的数据包缓存，发送一次数据包的成本提高；延迟时，传输时间所受影响较小，但丢包每一次重传往往伴随窗口所有数据的重传，导致虽然维持吞吐率但占据很多网络资源
- SR机制：发送端和接收端均需要维护队列保存发、接的数据包，进一步提高发送一次数据包的成本，但更能适应延迟较高、丢包率较高的情况

实际情况，影响GBN、SR机制的性能的因素有很多，比较结果不具有普遍性。