

计算机网络实验报告

实验三：基于UDP服务设计可靠传输协议并编程实现

姓名：孙悦

学号：2110052

专业：物联网工程

实验3-4 性能测试对比

实验要求：

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。

一、停等机制与滑动窗口机制性能对比

这里分别测试Lab3-1中的停等机制和Lab3-2与Lab3-3的滑动窗口机制进行性能对比。窗口大小选择window = 10、超时重传设定为>0.5s。

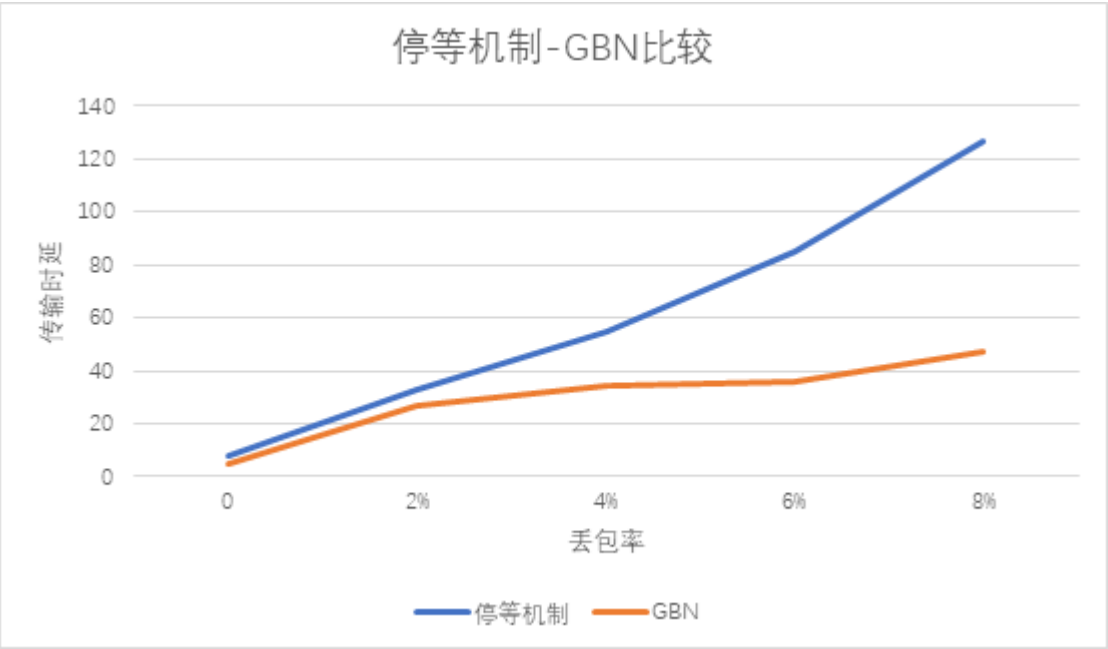
实验设置：

- (1) 测试文件：测试文件2，总文件大小为5898611Bytes。
- (2) 滑动窗口：GBN和SR机制,滑动窗口固定大小为10
- (3) 丢包率梯度设置（即控制延时为0时）：0%，2%，4%，6%，8%
- (4) 延时梯度设置（即控制丢包率为0时）：0ms，2ms，4ms，6ms，8ms

1.1 控制延时为0，改变丢包率，对比停等机制和滑动窗口GBN累计确认的性能

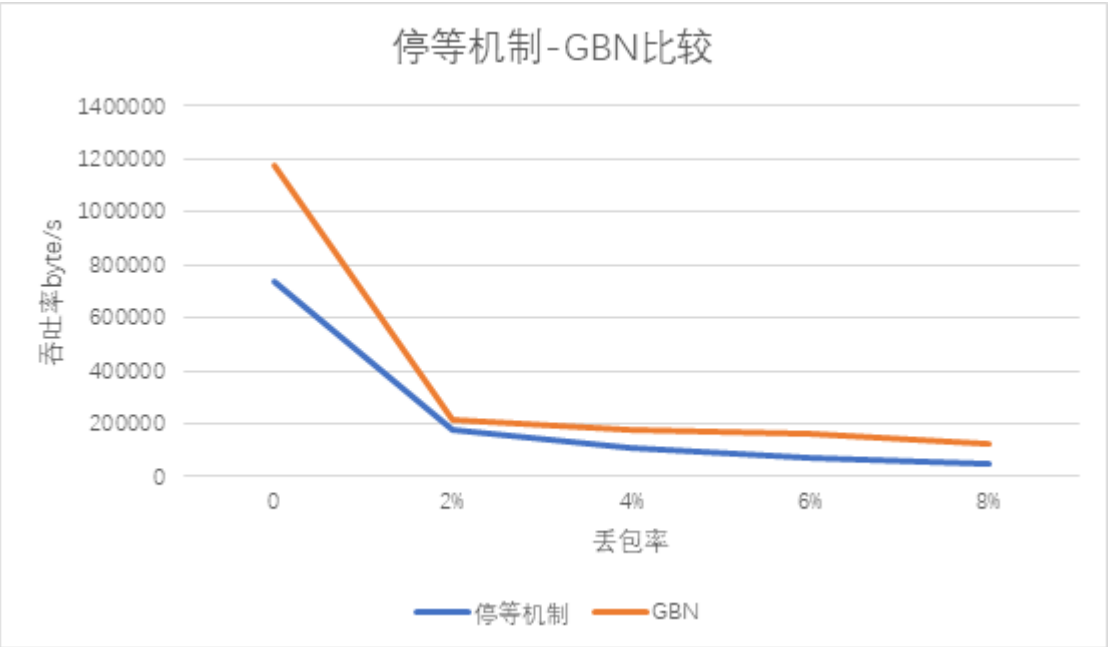
- 以传输时延为指标：

机制\丢包率	0	2%	4%	6%	8%
停等机制传输时间	8s	33s	55s	85s	127s
GBN传输时间	5s	27s	34s	36s	47s



- 以吞吐率为指标：

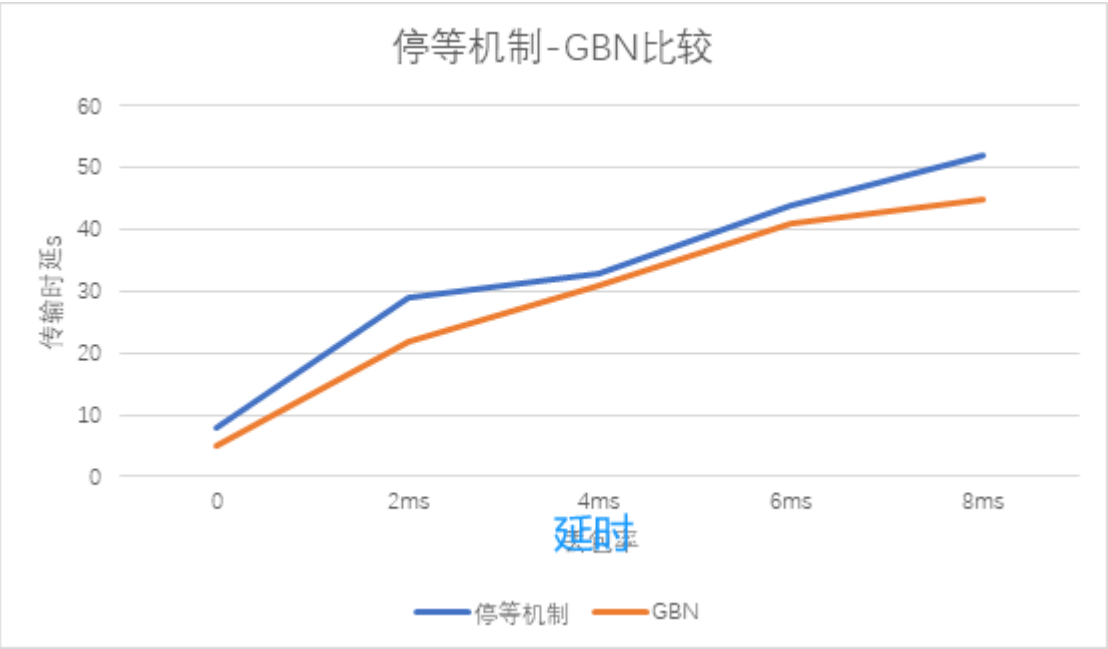
机制\丢包率	0	2%	4%	6%	8%
停等机制吞吐率	737313byte/s	178743byte/s	107246byte/s	69394.2byte/s	46444.9byte/s
GBN吞吐率	1.1797e+06byte/s	218463byte/s	173485byte/s	163847byte/s	123185byte/s



1.2 控制丢包率为0，改变延时，对比停等机制和滑动窗口GBN累计确认的性能

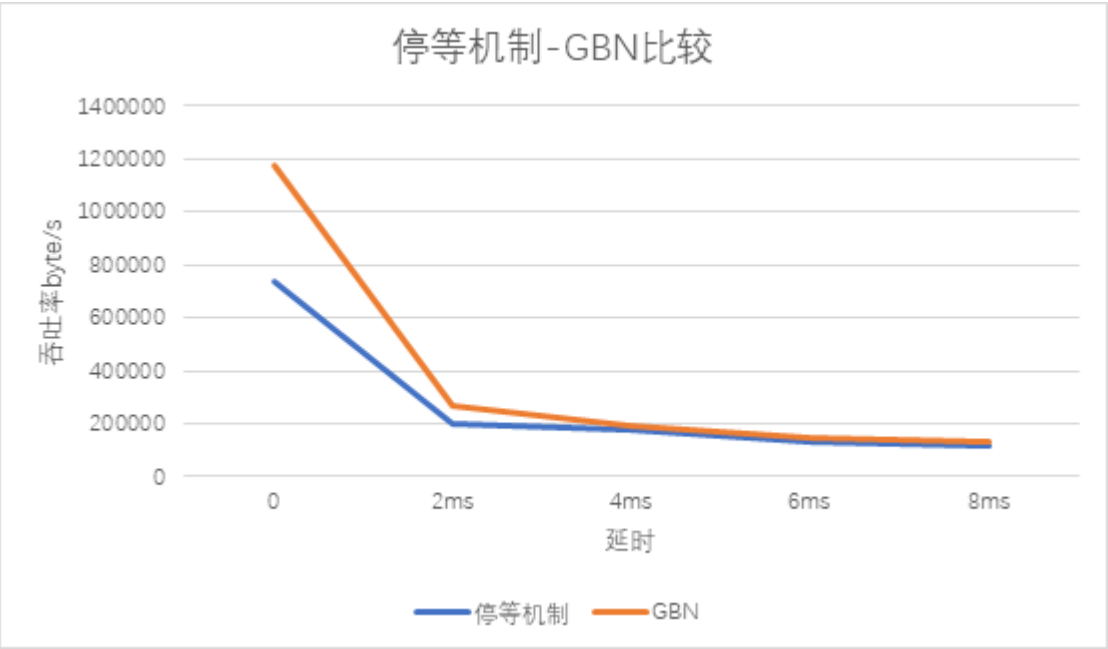
以传输时延为指标：

机制\延时	0	2ms	4ms	6ms	8ms
停等机制传输时间	8s	29s	33s	44s	51s
GBN传输时间	5s	22s	31s	41s	45s



以吞吐率为指标：

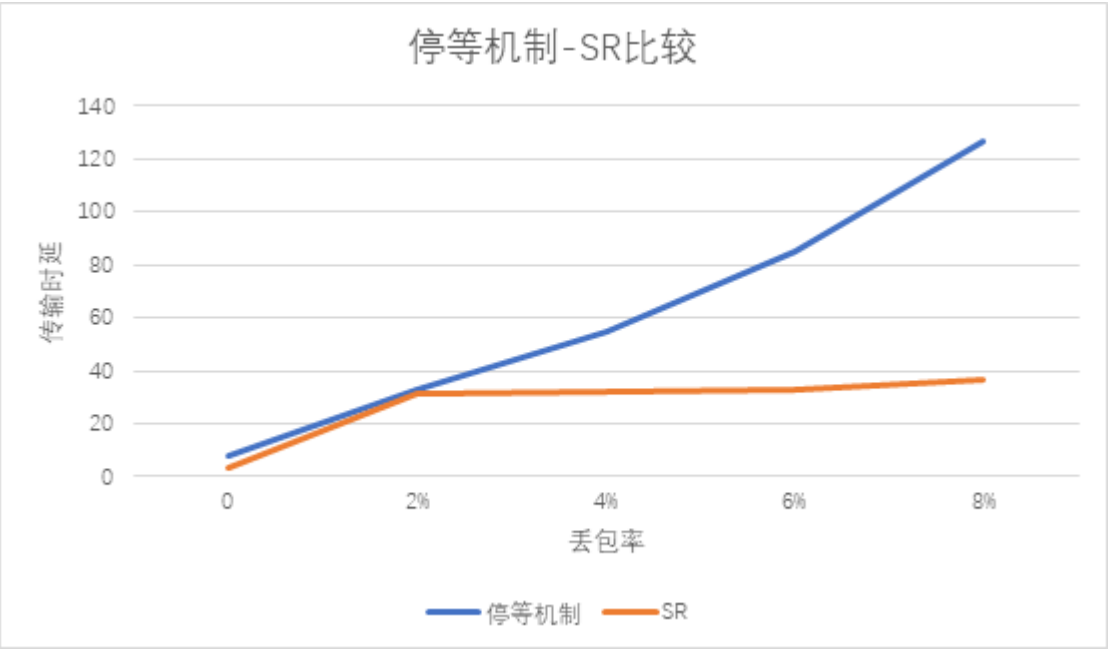
机制\延时	0	2ms	4ms	6ms	8ms
停等机制吞吐率	737313byte/s	203504byte/s	178986byte/s	134057byte/s	115657byte/s
GBN吞吐率	1.1797e+06byte/s	268114byte/s	190274byte/s	143866byte/s	131078byte/s



1.3控制丢包率为0，改变延时，对比停等机制和滑动窗口SR选择确认的性能

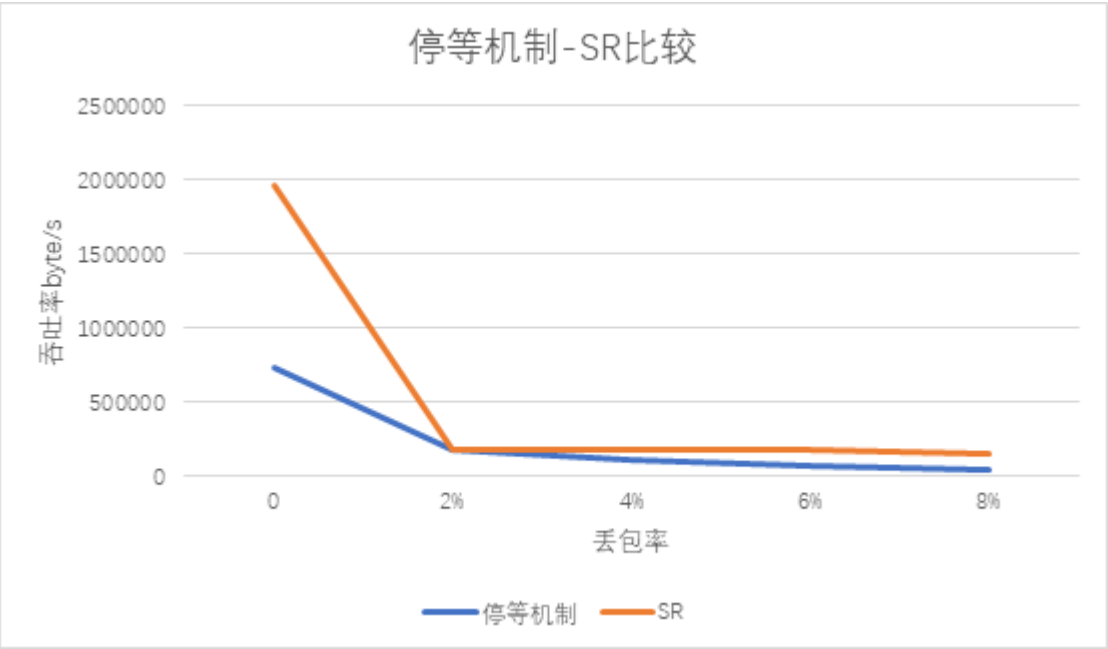
以传输时延为指标：

机制\丢包率	0	2%	4%	6%	8%
停等机制传输时间	8s	33s	55s	85s	127s
SR传输时间	3s	31s	32s	33s	37s



以吞吐率为指标：

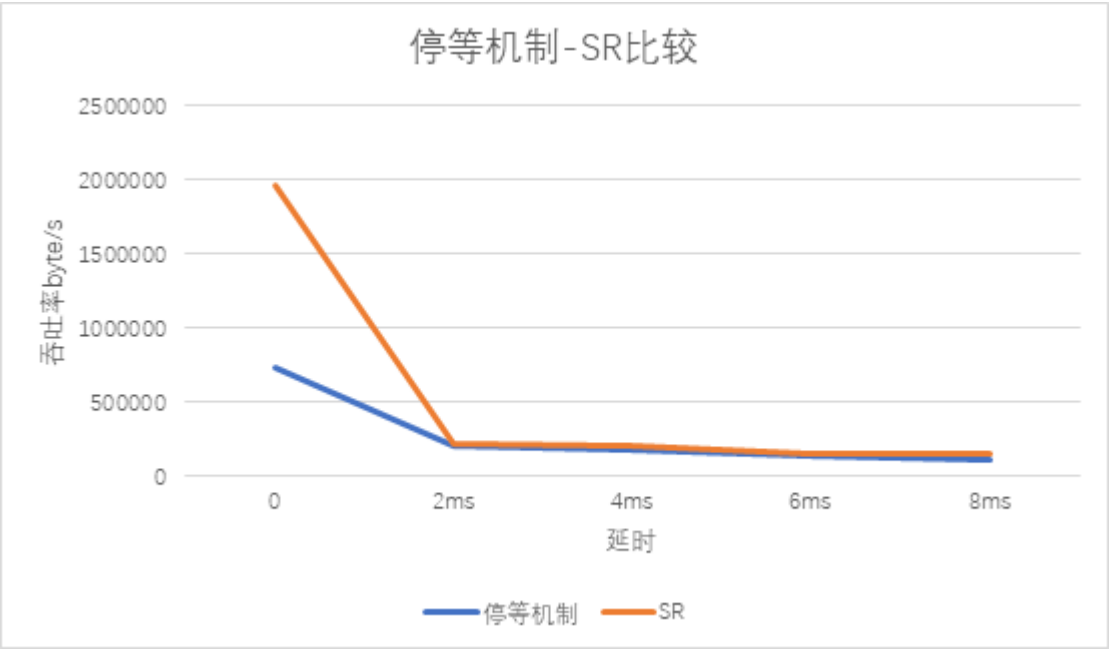
机制\丢包率	0	2%	4%	6%	8%
停等机制吞吐率	737313byte/s	178743byte/s	107246byte/s	69394.2byte/s	46444.9byte/s
SR吞吐率	1.96617e+06byte/s	186025byte/s	178986byte/s	178304byte/s	159419byte/s



1.4控制丢包率为0，改变延时，对比停等机制和滑动窗口SR选择确认的性能

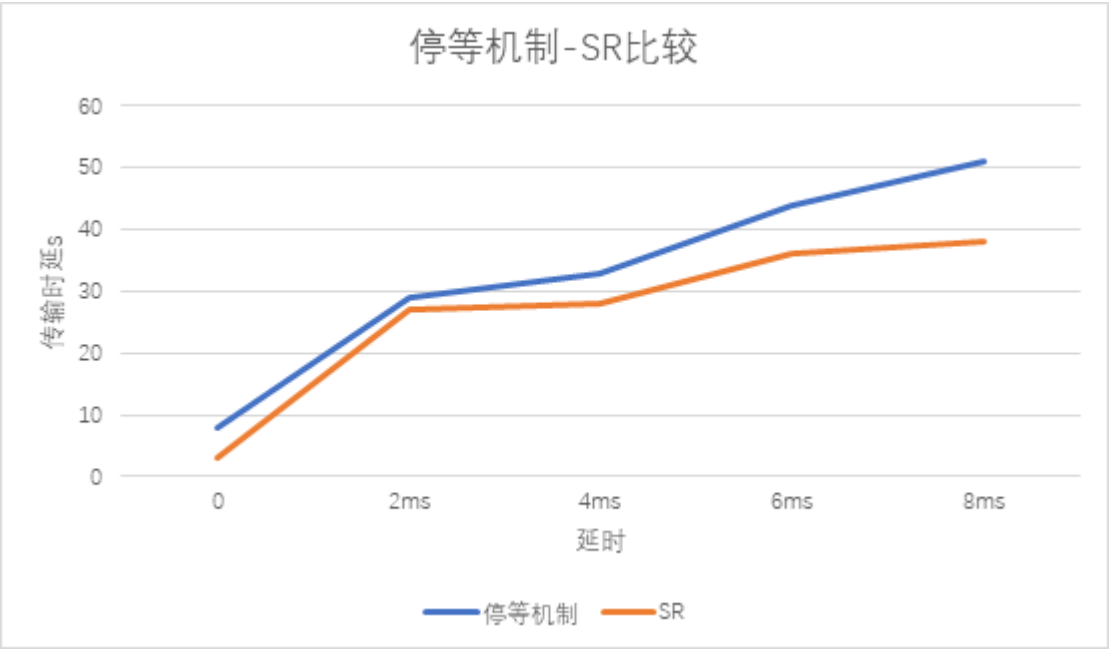
以吞吐率为指标：

机制\延时	0	2ms	4ms	6ms	8ms
停等机制吞吐率	737313byte/s	203504byte/s	178986byte/s	134057byte/s	115657byte/s
SR吞吐率	1.96617e+06byte/s	216570byte/s	210834byte/s	159998byte/s	156054byte/s



以传输时延为指标：

机制\延时	0	2ms	4ms	6ms	8ms
停等机制传输时间	8s	29s	33s	44s	51s
SR传输时间	3s	27s	28s	36s	38s



1.5 停等机制与滑动窗口的比较总结：

- 二者的传输性能均严重的依赖丢包率。随丢包率提高而显著降低。

- 在没有丢包或丢包率较低的情况下：滑动窗口的性能优于停等机制。
 - 滑动窗口可以允许发送多条，同时等待ACK，减少了RTT在总时间上的影响。
- 丢包率较大的情况下：滑动窗口性能低于停等机制。
 - 滑动窗口GBN在丢包情况下的数据重传严重、重传耗时代价过高、网络大量数据报拥塞。
- 延时对于二者的影响改变的没有丢包率提高那么严重，但是延迟确实所有数据的延迟降低。
- 延时状况下：滑动窗口的性能更优一些。
 - 停等机制需要每条消息单独等待时延和RTT，而窗口可以同时等待多条。

二、滑动窗口机制中不同窗口大小对性能的影响

实验设置：

在这里我们测试一下不同滑动窗口大小的滑动窗口协议累积确认GBN与SR选择确认会带来的变化。部分已经重复测试过的数据会直接沿用。

- (1)测试文件：测试文件2，文件大小为5898611Bytes。
- (2)滑动窗口：GBN和SR机制
- (3)滑动窗口大小梯度设置：4，10，20
- (4)丢包率梯度设置（即控制时延为0时）：0%，2%，4%，6%，8%
- (5)时延梯度设置（即控制丢包率为0时）：0ms，2ms，4ms，6ms，8ms

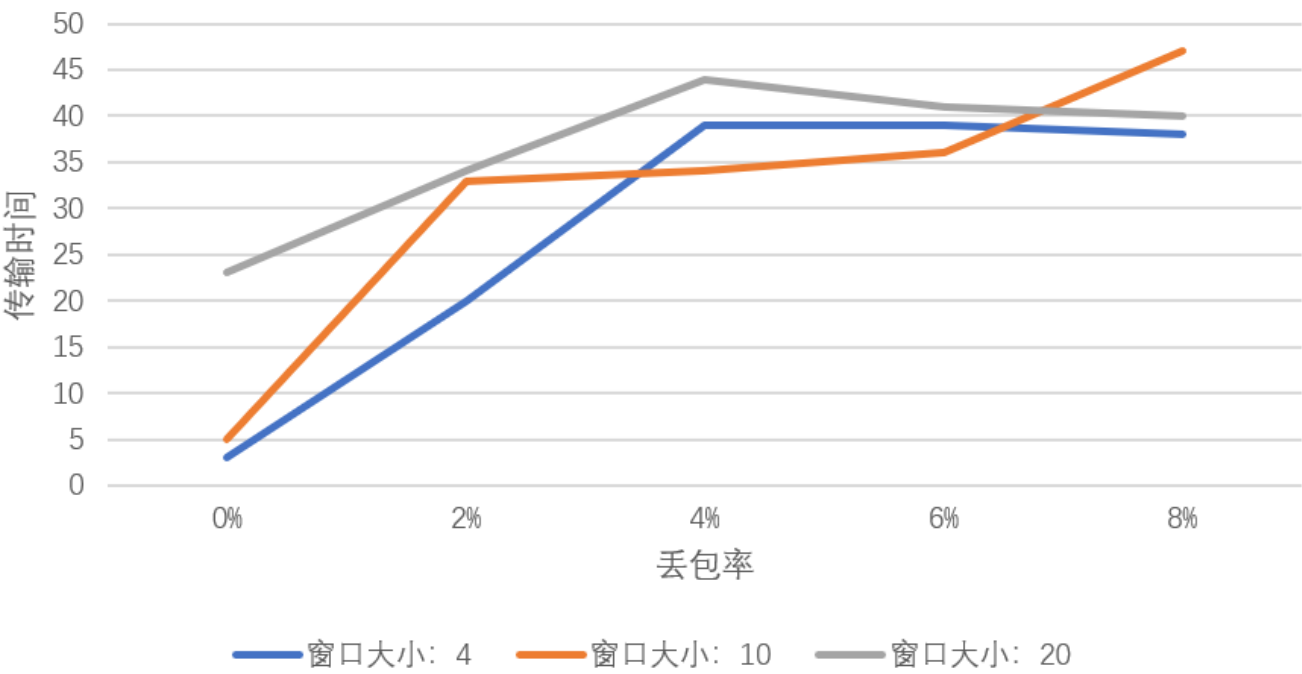
2.1滑动窗口GBN不同窗口大小对性能的影响

2.1.1延时为0 更改丢包率

以传输时延为指标：

窗口\丢包率	0	2%	4%	6%	8%
4	3s	30s	39s	39s	38s
10	5s	33s	34s	36s	47s
20	23s	34s	44s	41s	40s

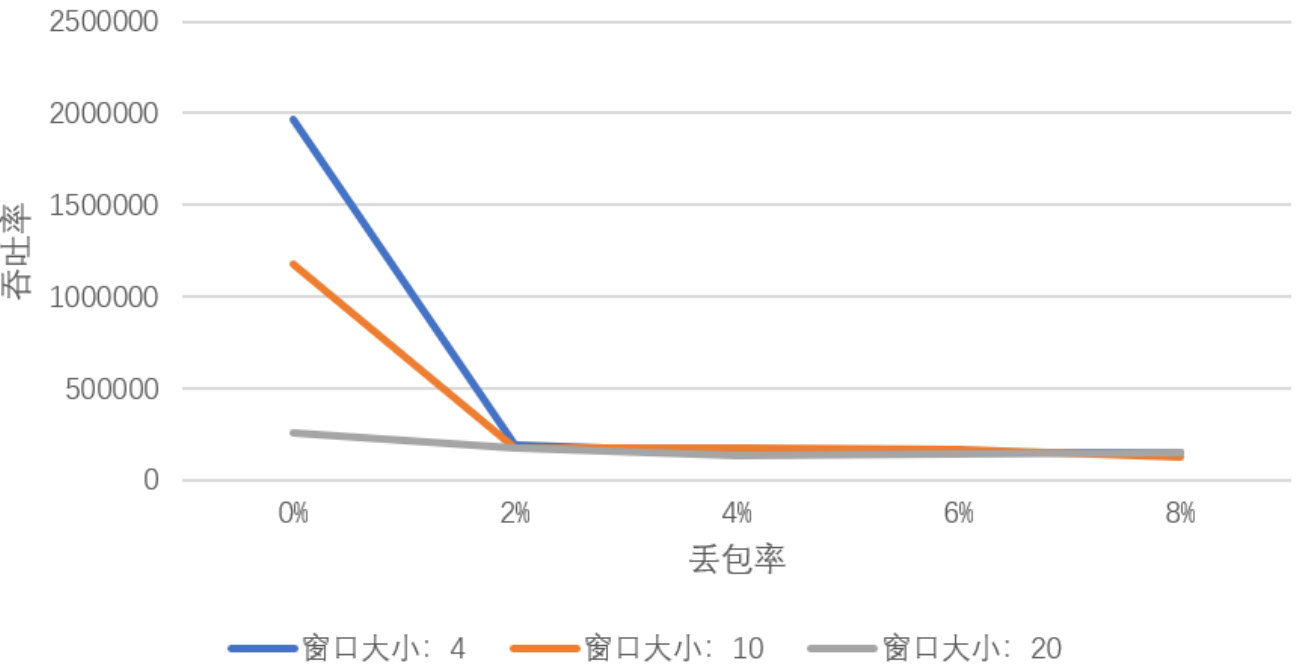
GBN传输时间



以吞吐率为指标:

窗口\丢包率	0	2%	4%	6%	8%	
4	1.96617e+06byt	196617byte/s	151244byte/s	151244byte/s	155224byte/s	
10	1.1797e+06byte	178743byte/s	173485byte/s	163847byte/s	123185byte/s	
20	256457byte/s	173485byte/s	134057byte/s	143866byte/s	147463byte/s	

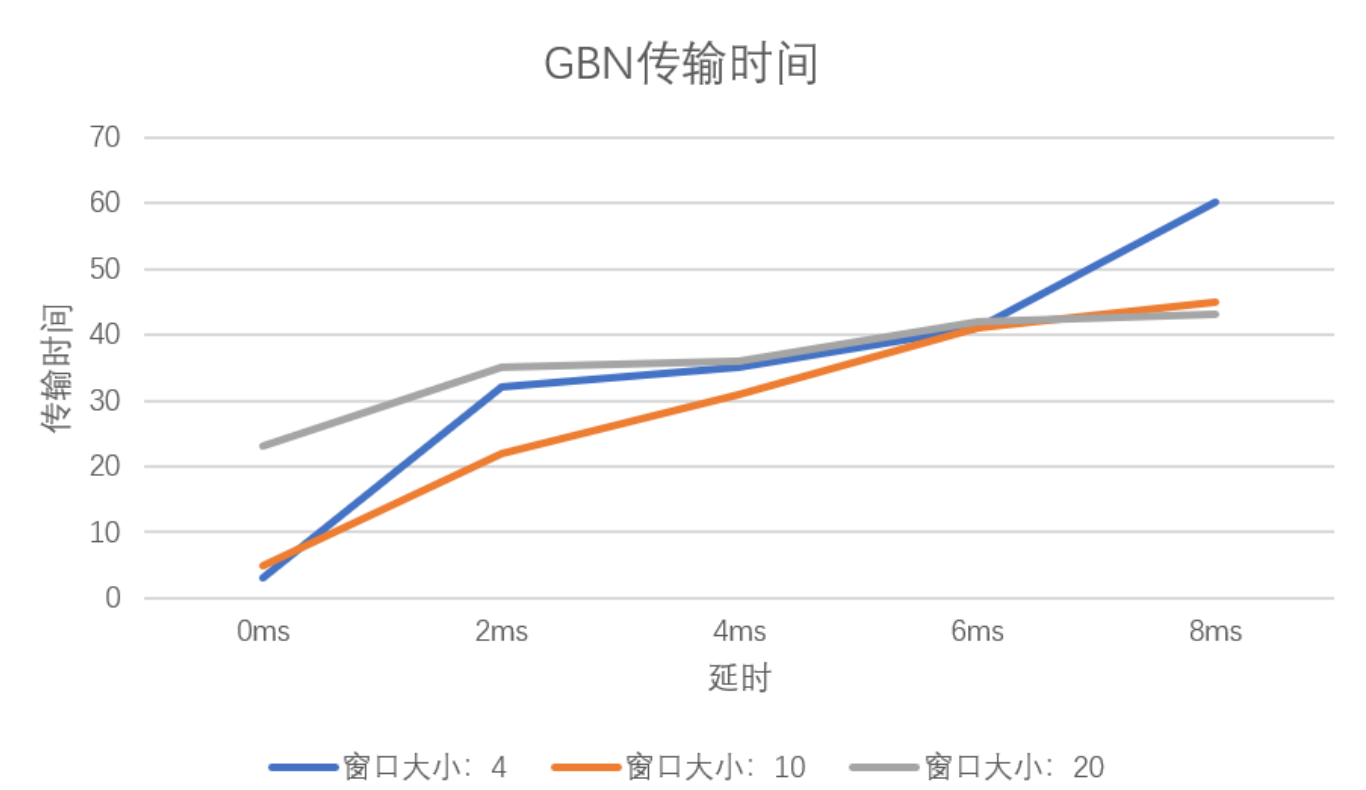
GBN吞吐率



2.1.2丢包率为0 更改延时

以传输时延为指标：

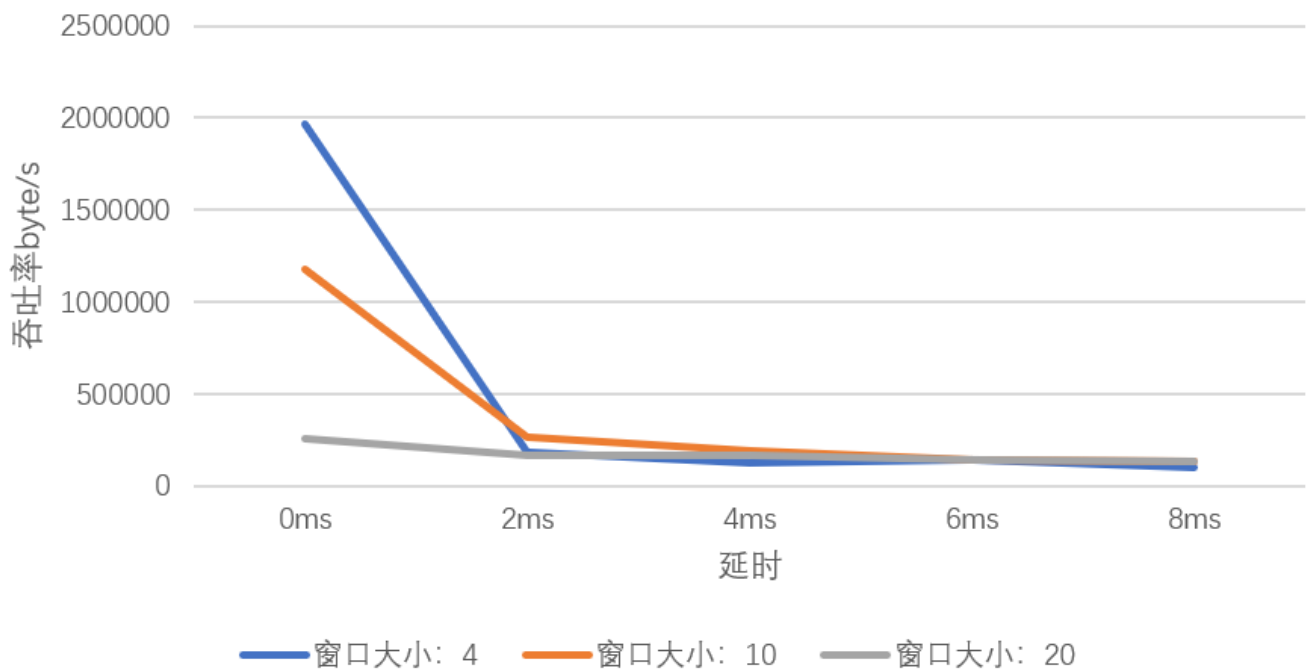
窗口\延时	0	2ms	4ms	6ms	8ms
4	3s	32s	33s	41s	60s
10	5s	22s	31s	41s	45s
20	23s	35s	36s	42s	43s



以吞吐率为指标：

窗口\延时	0	2ms	4ms	6ms	8ms	
4	1.96617e+06byt	184328byte/s	180186byte/s	143866byte/s	98308.4byte/s	
10	1.1797e+06byte	268114byte/s	190274byte/s	143866byte/s	131078byte/s	
20	256457byte/s	168529byte/s	163847byte/s	140441byte/s	137175byte/s	

GBN吞吐率



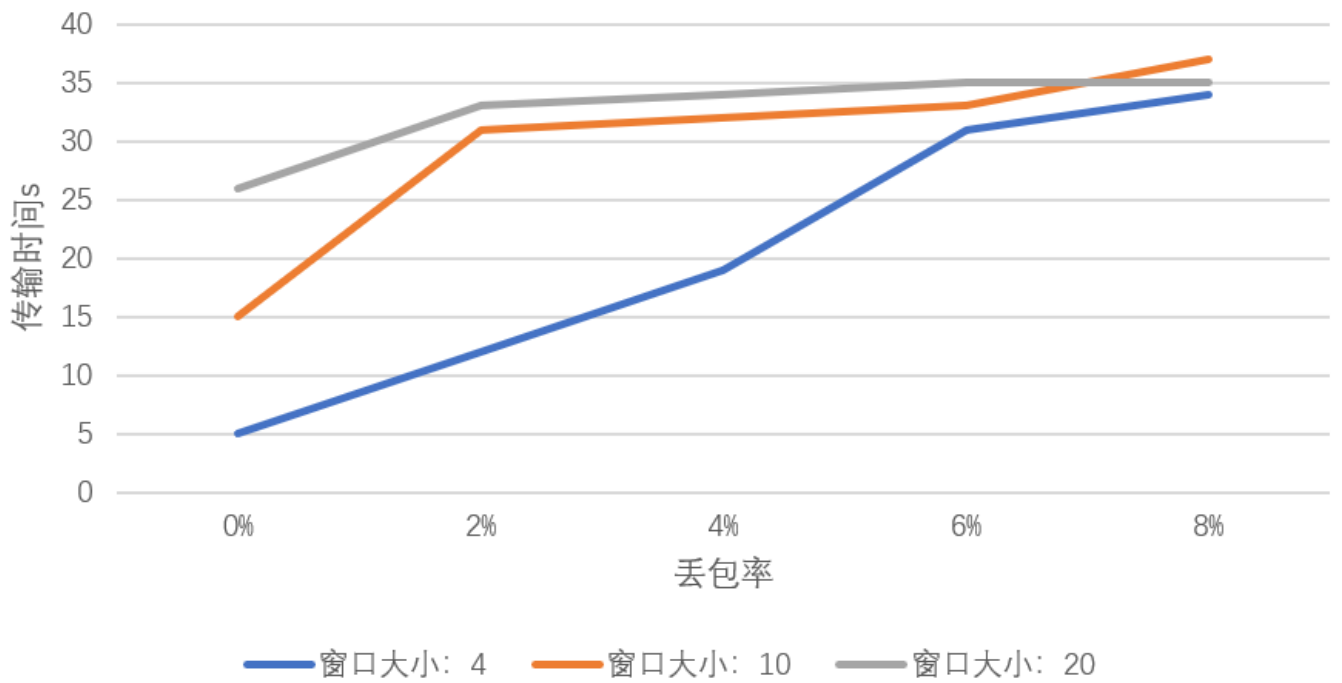
2.2滑动窗口SR不同窗口大小对性能的影响

2.2.1延时为0 更改丢包率

以传输时延为指标：

窗口\丢包	0	2%	4%	6%	8%
4	8s	12s	19s	31s	34s
10	15s	31s	32s	33s	37s
20	26s	33s	34s	35s	35s

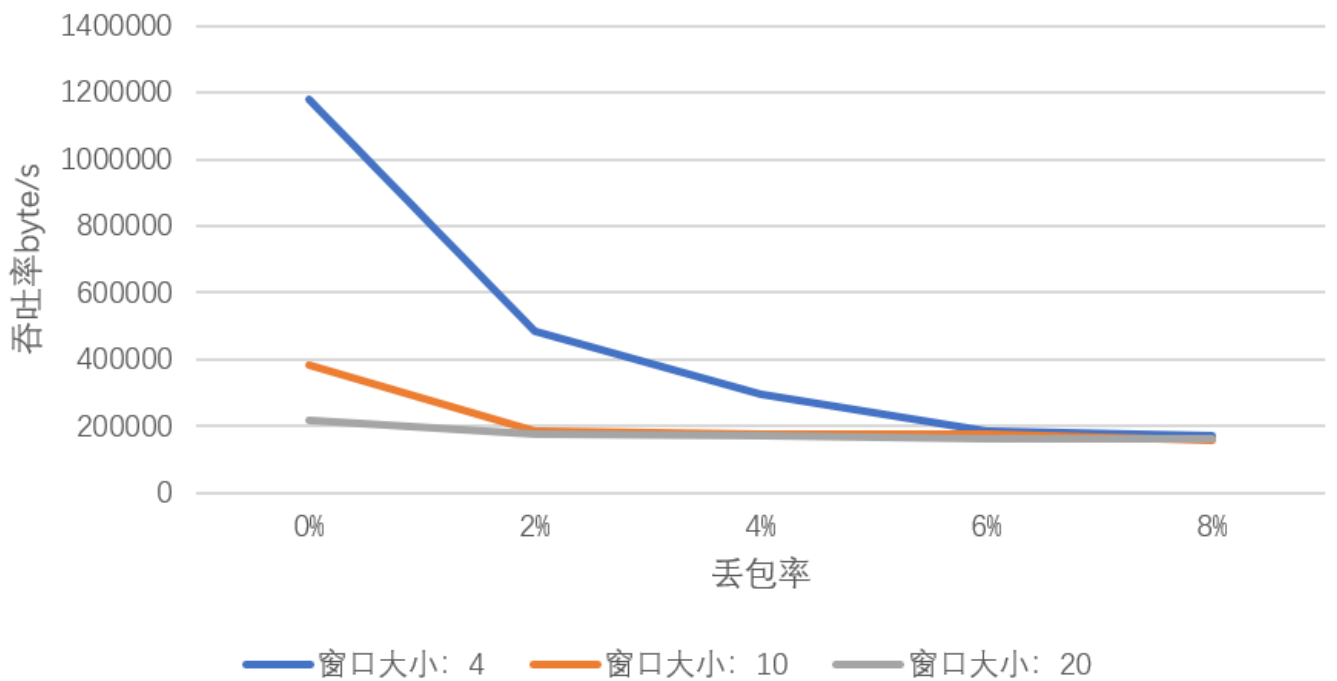
SR传输时间



以吞吐率为指标:

窗口\丢包	0	2%	4%	6%	8%
4	737313byte/s	485913byte/s	296243byte/s	185318byte/s	169995byte/s
10	383567byte/s	186025byte/s	178986byte/s	178304byte/s	159419byte/s
20	219128byte/s	178655byte/s	172951byte/s	165103byte/s	164381byte/s

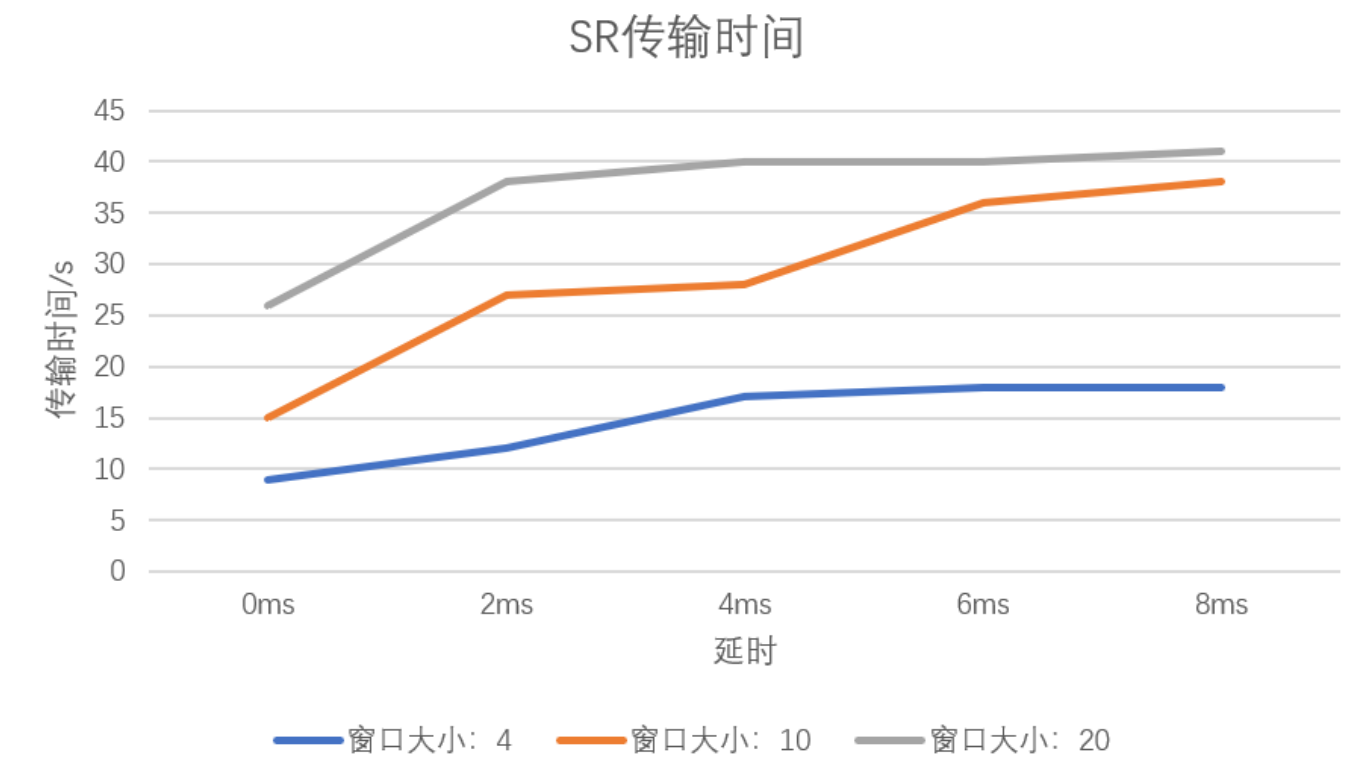
SR吞吐率



2.2.2丢包率为0 更改延时

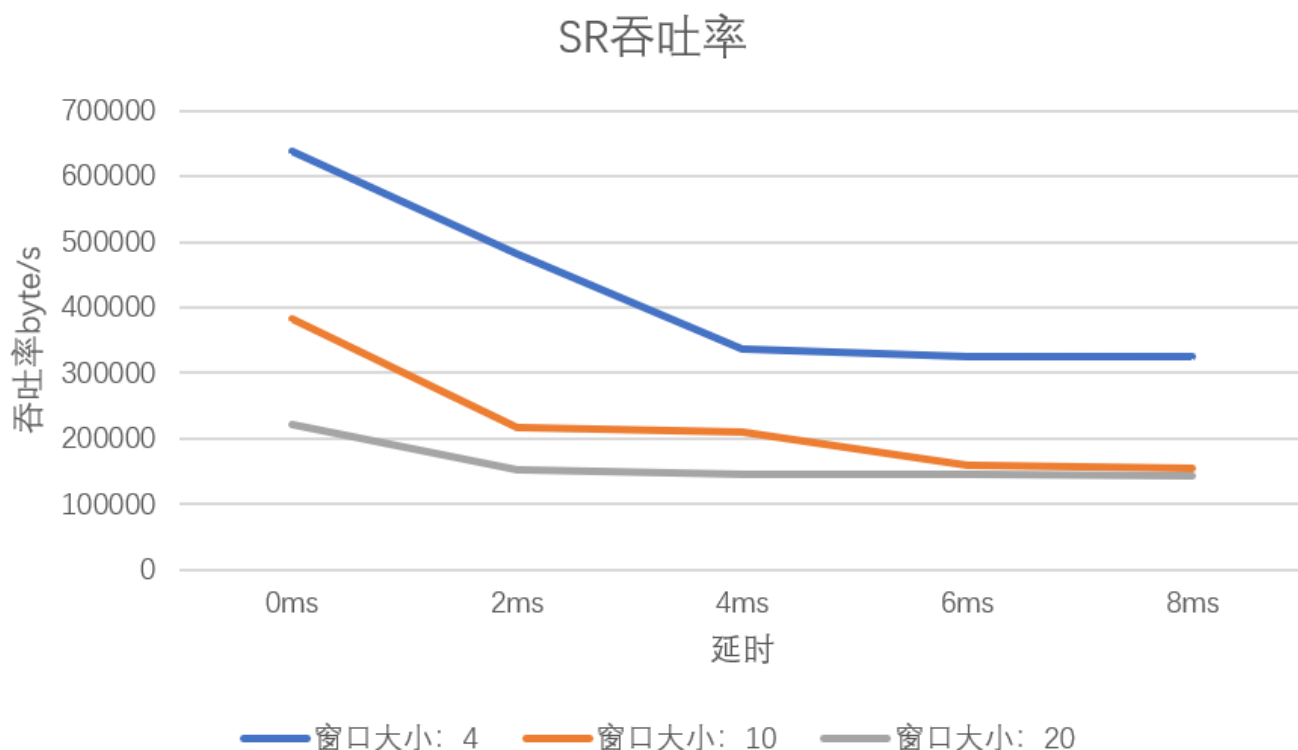
以传输时延为指标：

窗口\延时	0	2ms	4ms	6ms	8ms
4	9s	12s	17s	18s	18s
10	15s	27s	28s	36s	38s
20	26s	38s	40s	40s	41s



以吞吐率为指标：

窗口\延时	0	2ms	4ms	6ms	8ms
4	636918byte/s	481589byte/s	337597byte/s	325560byte/s	325650byte/s
10	383567byte/s	216570byte/s	210834byte/s	159998byte/s	156054byte/s
20	221373byte/s	152451byte/s	146736byte/s	146390byte/s	144309byte/s



2.3 不同的滑动窗口大小性能对比总结: (SR 与GBN二者结论相似)

- 不同的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
- 在丢包率改变增大的情况下：
 - 滑动窗口大小越大的性能下降越快、降低更多。
 - 大的窗口带来的重传代价更大。
 - 在丢包率为0或者较低的时候，大窗口的性能、效率高一些
- 在时延的改变情况下：滑动窗口的大小区别在这里影响区别不大，保持着同样的速率降低趋势

三、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

实验设置:

在这里我们测试一下在窗口大小一致的情况下对比下滑动窗口协议中的累积确认GBN和选择确认SR的性能。部分已经重复测试过的数据会直接沿用。

(1)测试文件：测试文件2，即总文件大小（包括文件路径）为5898611Bytes。

(2)滑动窗口：GBN和SR机制

(3)滑动窗口大小固定设置：10

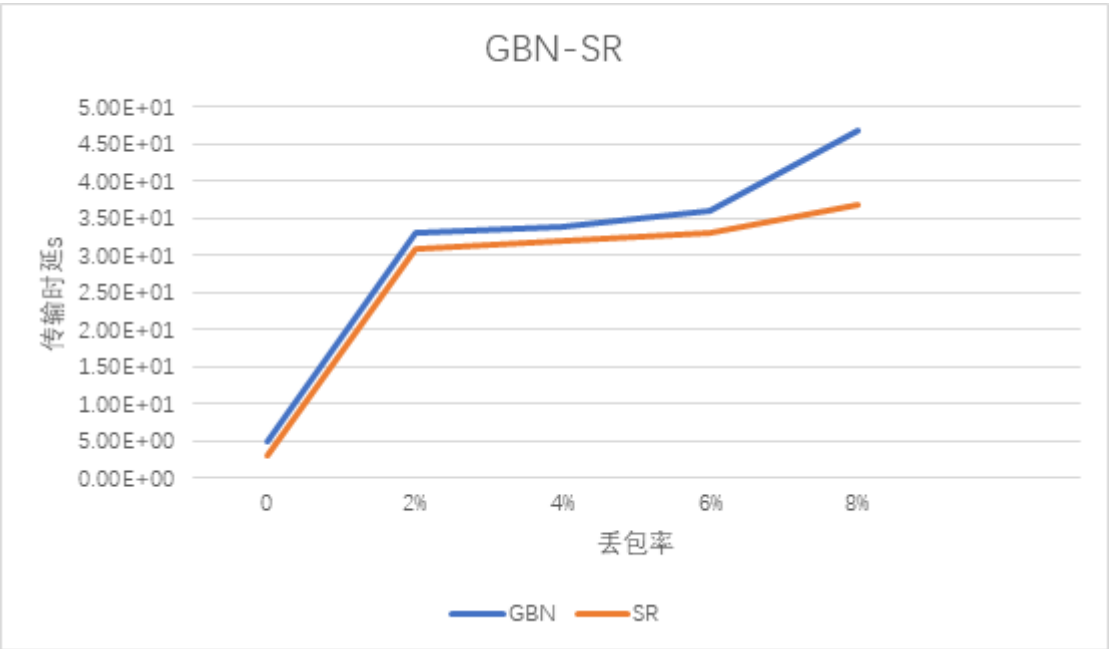
(4)丢包率梯度设置（即控制时延为0时）：0%，2%，4%，6%，8%

(5)时延梯度设置（即控制丢包率为0时）：0ms，2ms，4ms，6ms，8ms

3.1延时为0 更改丢包率

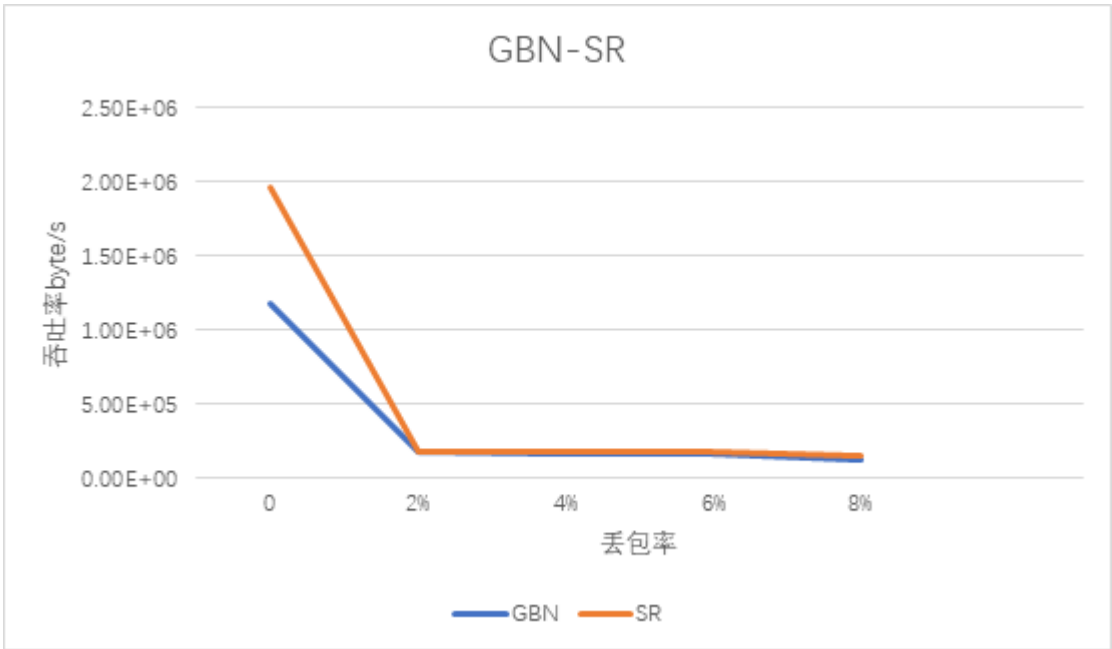
以传输时延为指标：

机制\丢包率	0	2%	4%	6%	8%
GBN传输时间	5s	33s	34s	36s	47s
SR传输时间	3s	31s	32s	33s	37s



以吞吐率为指标：

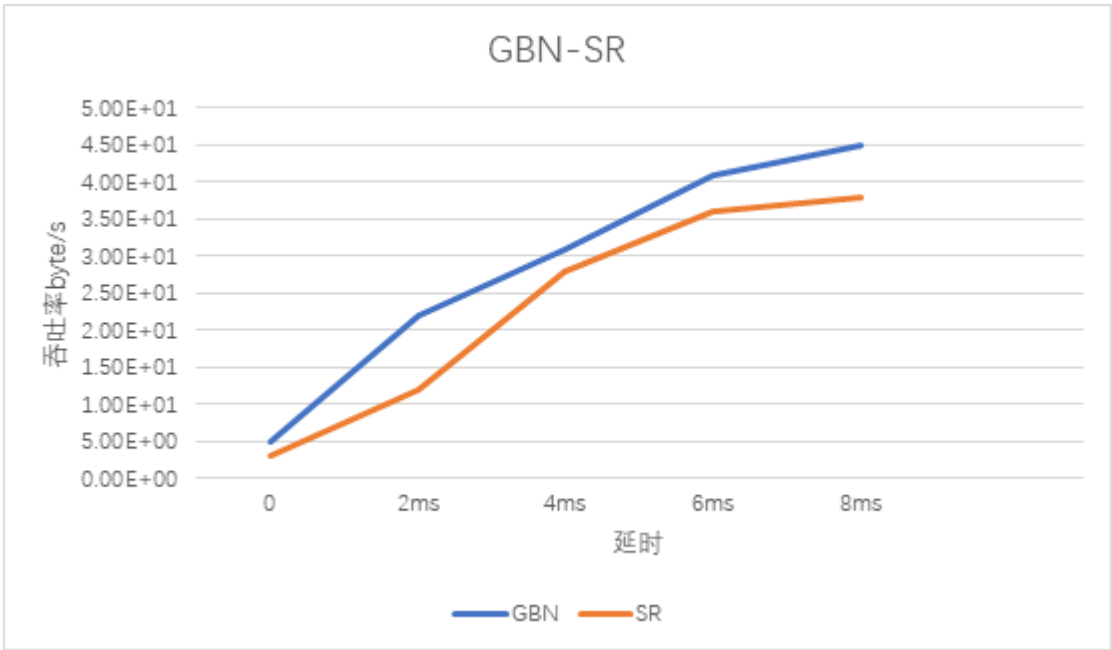
机制\丢包率	0	2%	4%	6%	8%
GBN吞吐率	1.1797e+06byte/s	178743byte/s	173485byte/s	163847byte/s	123185byte/s
SR吞吐率	1.96617e+06byte/s	186025byte/s	178986byte/s	178304byte/s	159419byte/s



3.2丢包率为0 更改延时

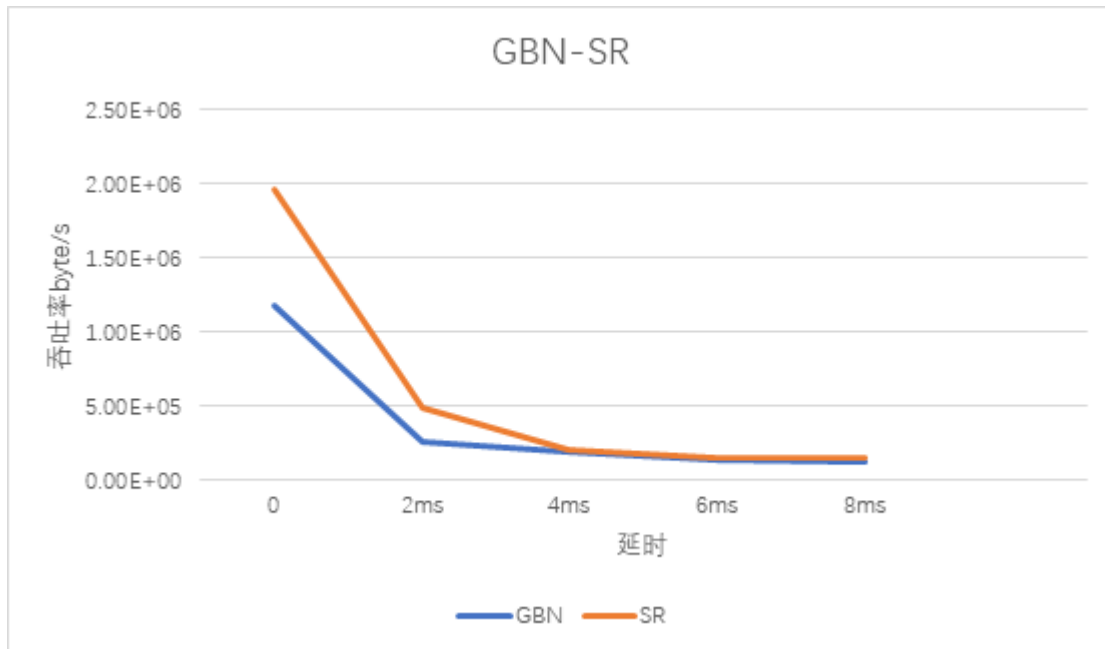
以传输时延为指标:

机制\延时	0	2ms	4ms	6ms	8ms
GBN传输时间	5s	22s	31s	41s	45s
SR传输时间	3s	12s	28s	36s	38s



以吞吐率为指标:

机制\延时	0	2ms	4ms	6ms	8ms
GBN吞吐量	1.1797e+06byte/s	268114byte/s	190274byte/s	143866byte/s	131078byte/s
SR吞吐量	1.96617e+06byte/s	491542byte/s	210834byte/s	159998byte/s	156054byte/s



3.3 累计确认和选择确认的性能比较：

- 当丢包率较低（0%）时，SR性能高于GBN。

因为窗口大小毕竟只有10，那么窗口大小满了后导致的延时等情况对于GBN而言就要全部超时重传，而SR只需要超时重传部分即可。因此带来了性能的优异。同时SR实现接收方可以同时发送ACK和接收报文。

- 随着丢包率增大，SR性能和GBN性能之间的差异增大。因为对GBN而言是需要全部超时重传，而SR只需要超时重传部分即可
- 改变延时，SR性能由于GBN，SR与GBN变化与改变丢包率相似
- 当延时增大导致超时重传成为主旋律，SR就可以省下更多的资源，实现更为合理的重传。