

实验3-4

学号：2112066 姓名：于成俊

一、实验题目

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。

性能测试指标：时延、吞吐率，要给出图、表并进行分析。

二、性能测试

为了方便进行测试，本次测试均选择传输图片1。此外，为了减小偶然性对测试的影响，每次测试的结果均是五次测试后取的平均值。

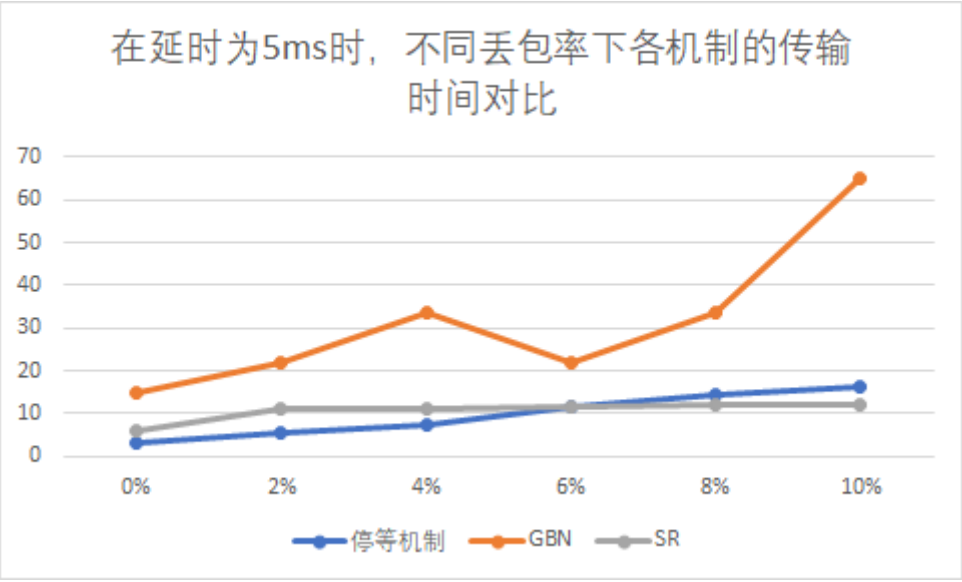
(1) 停等机制与滑动窗口机制性能对比：

1.将滑动窗口大小设为10，延时设为5ms，更改丢包率，来比对传输时间和吞吐率：

传输时间（单位：s）

丢包率	停等机制	GBN	SR
0%	3.068	14.772	5.882
2%	5.603	22.035	11.142
4%	7.344	33.589	11.268
6%	11.643	21.992	11.71
8%	14.382	33.474	12.071
10%	16.462	64.836	12.2

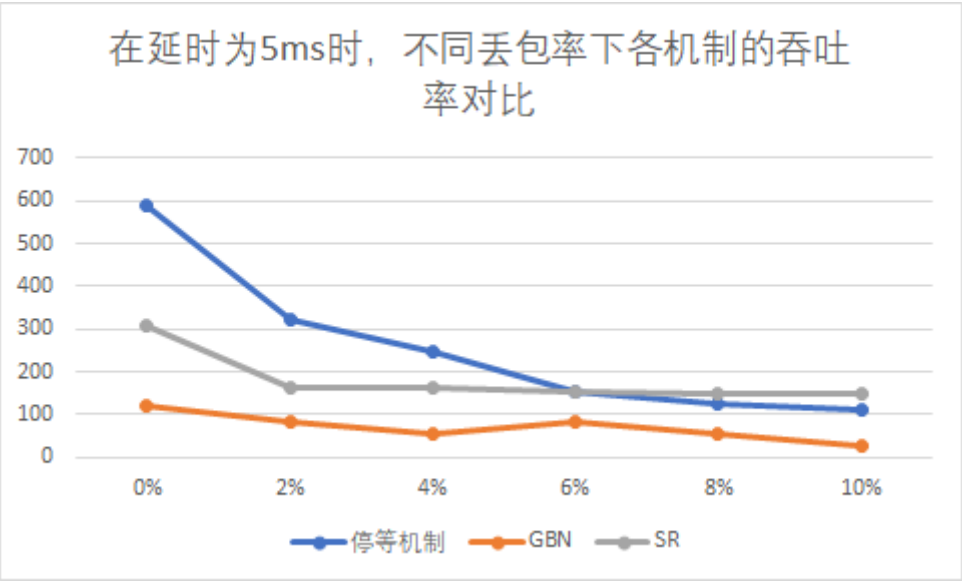
根据数据绘制图像：



吞吐量 (单位: KB/s)

丢包率	停等机制	GBN	SR
0%	590.939	122.732	308.228
2%	323.577	82.2782	162.718
4%	246.868	53.976	160.898
6%	155.716	82.4391	154.825
8%	126.06	54.1614	150.195
10%	110.132	27.9629	148.607

根据数据绘制图像:

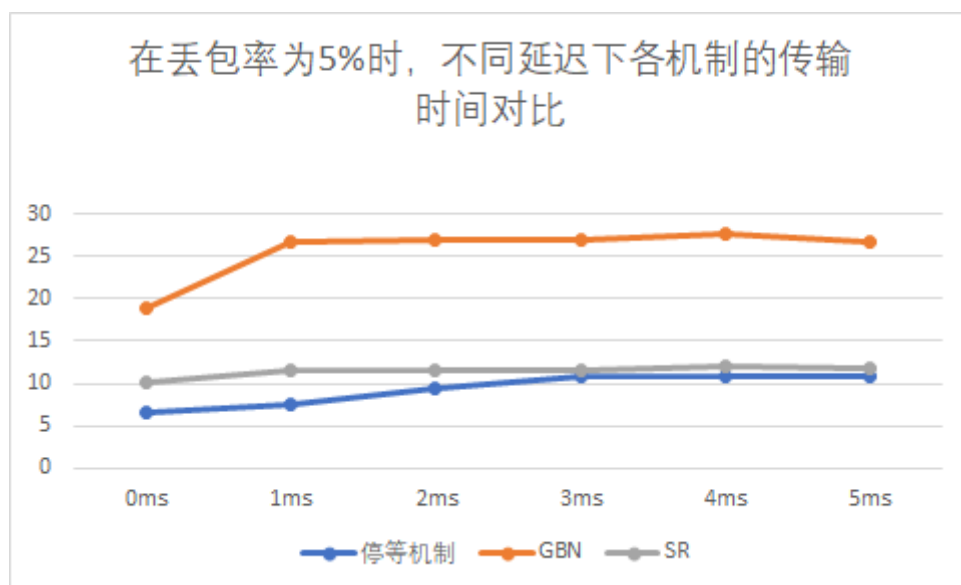


2.将滑动窗口大小设为10，丢包率设为5%，更改延迟，来比对传输时间和吞吐率：

传输时间（单位：s）

延迟	停等机制	GBN	SR
0ms	6.468	19.01	10.06
1ms	7.64	26.835	11.477
2ms	9.355	26.868	11.456
3ms	10.748	27.044	11.536
4ms	10.94	27.597	12.042
5ms	10.958	26.782	11.903

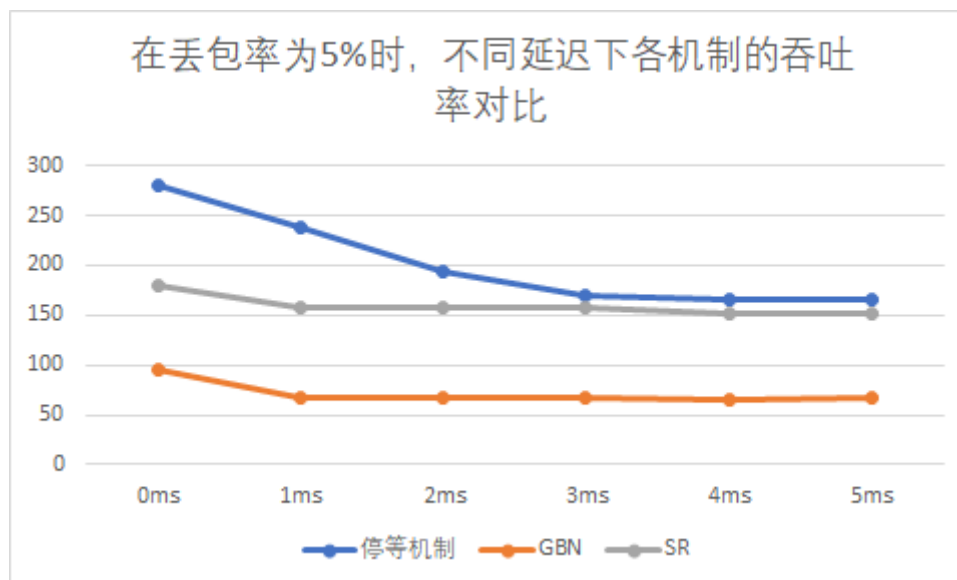
根据数据绘制图像：



吞吐率（单位：KB/s）

延时	停等机制	GBN	SR
0ms	280.303	95.3709	180.219
1ms	237.304	67.561	157.968
2ms	193.8	67.478	158.258
3ms	168.683	67.0389	157.16
4ms	165.722	65.6955	150.556
5ms	165.45	67.6947	152.315

根据数据绘制图像：



3.结果分析:

- 从图中看出，停等机制的性能较高于滑动窗口机制。我认为，由于RTT较小，停等机制与滑动窗口的性能本身差距就不大，再加上网络环境较差，或者网络带宽较低，导致停等机制的性能优于滑动窗口机制。当然，也与滑动窗口实现的代码效率有一定的关系，因为滑动窗口实现的逻辑较为复杂。其中GBN的性能最低，可能是我用单线程实现的原因。
- 当延迟固定时，随着丢包率的增加，三个机制的传输时间和吞吐率都呈下降趋势。其中，GBN下降的幅度最大，因为丢包率越高，GBN重传次数增加，而GBN每次重传付出的额外代价还很多，导致性能急速下降。而停等机制和SR下降幅度较缓，因为重传代价较小。
- 当丢包率固定时，停等机制的性能先随着延迟增加，而下降，然后趋于稳定。SR和GBN几乎不受延迟增加的影响，基本趋于稳定。

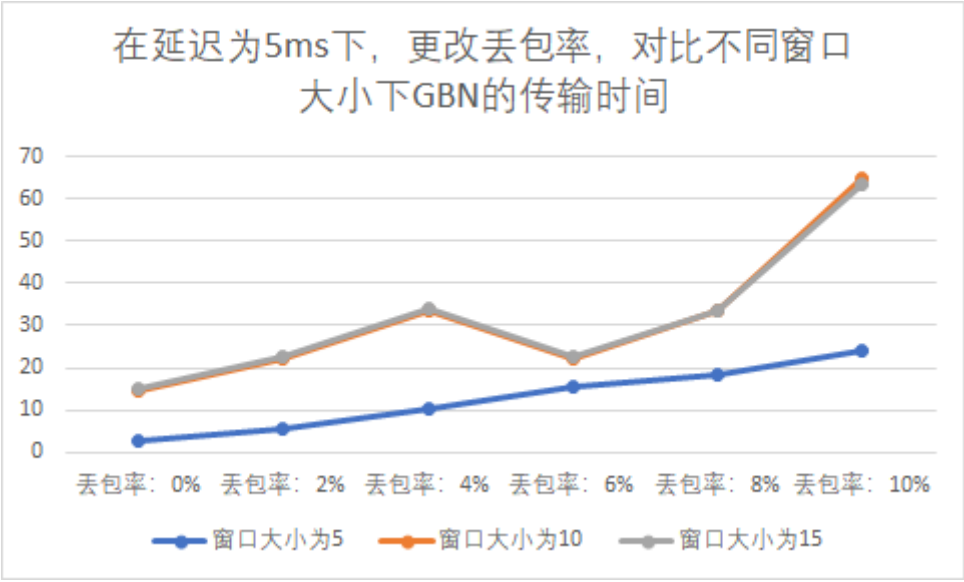
(2) 滑动窗口机制中不同窗口大小对性能的影响（累计确认和选择确认两种情形）

1.对于累计确认（即GBN），将延迟设为5ms，更改丢包率，比对传输时间和吞吐率

传输时间（单位：s）

	丢包率： 0%	丢包率： 2%	丢包率： 4%	丢包率： 6%	丢包率： 8%	丢包率： 10%
窗口大小 为5	2.845	5.763	10.307	15.709	18.286	24.03
窗口大小 为10	14.772	22.035	33.589	21.992	33.474	64.836
窗口大小 为15	14.999	22.636	33.994	22.675	33.338	63.4

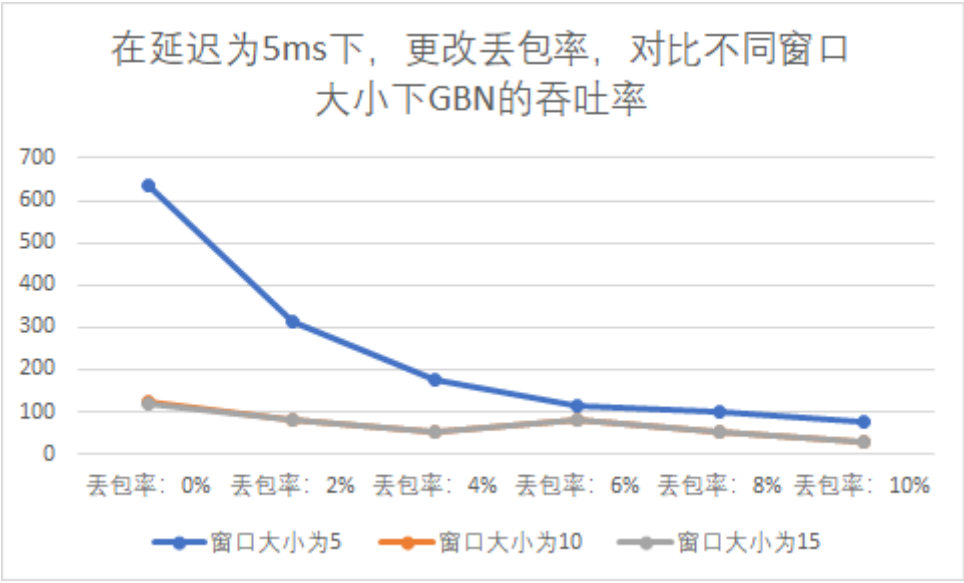
根据数据绘制图像：



吞吐率 (单位: KB/s)

	丢包率：0%	丢包率：2%	丢包率：4%	丢包率：6%	丢包率：8%	丢包率：10%
窗口大小为5	637.258	314.593	175.9	115.412	99.1469	75.447
窗口大小为10	122.732	82.2782	53.976	82.4391	54.1614	27.9629
窗口大小为15	120.875	80.0937	53.3329	79.9559	54.3824	28.5962

根据数据绘制图像：

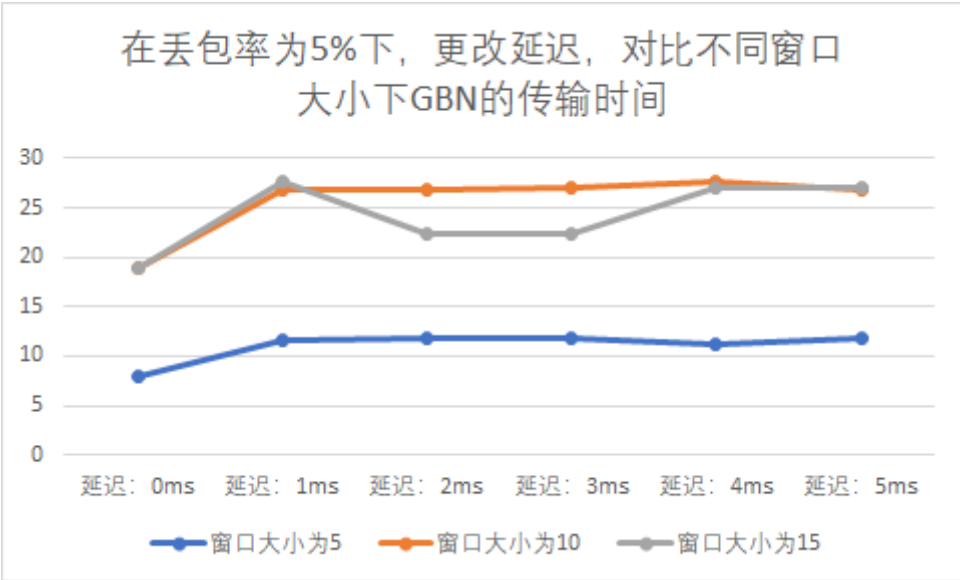


2.对于累计确认（即GBN），将丢包率设为5%，更改延迟，比对传输时间和吞吐率

传输时间（单位：s）

	延迟： 0ms	延迟： 1ms	延迟： 2ms	延迟： 3ms	延迟： 4ms	延迟： 5ms
窗口大小为5	8.018	11.689	11.777	11.725	11.268	11.785
窗口大小为10	19.01	26.835	26.868	27.044	27.597	26.782
窗口大小为15	18.991	27.663	22.355	22.44	26.954	27.097

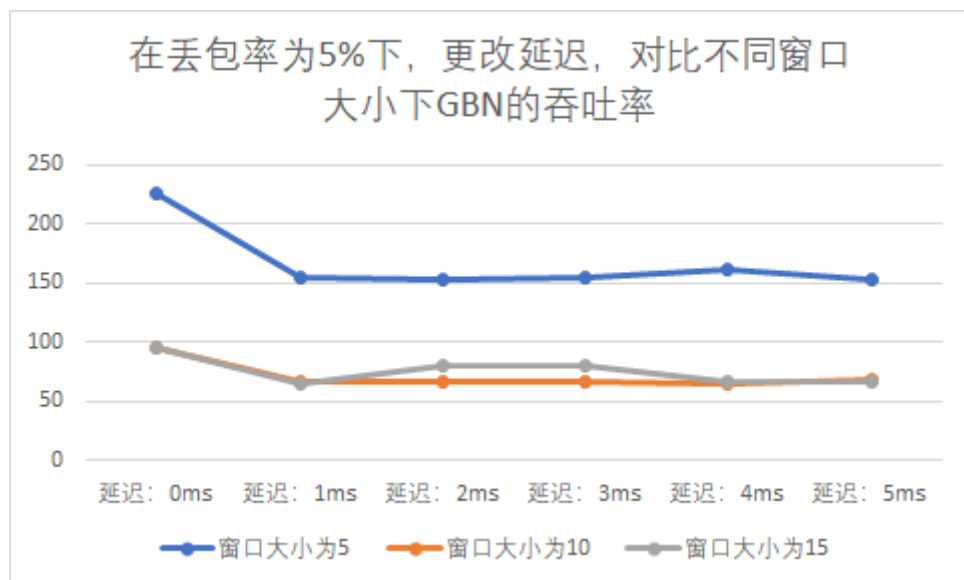
根据数据绘制图像：



吞吐率（单位：KB/s）

	延迟： 0ms	延迟： 1ms	延迟： 2ms	延迟： 3ms	延迟： 4ms	延迟： 5ms
窗口大小为5	226.116	155.103	153.944	154.627	160.898	153.84
窗口大小为10	95.3709	67.561	67.478	67.0389	65.6955	67.6947
窗口大小为15	95.4663	65.5388	81.1004	80.7932	67.2627	66.9078

根据数据绘制图像：

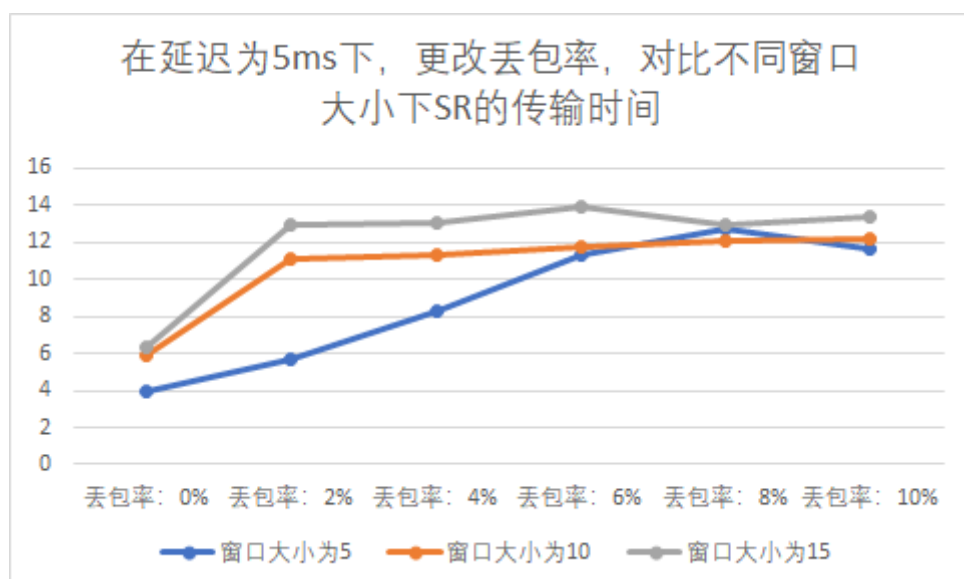


3.对于选择确认（即SR），将延迟设为5ms，更改丢包率，比对传输时间和吞吐率

传输时间（单位：s）

	丢包率：0%	丢包率：2%	丢包率：4%	丢包率：6%	丢包率：8%	丢包率：10%
窗口大小为5	3.967	5.713	8.311	11.286	12.693	11.613
窗口大小为10	5.882	11.142	11.268	11.71	12.071	12.2
窗口大小为15	6.355	12.951	13.058	13.901	12.931	13.41

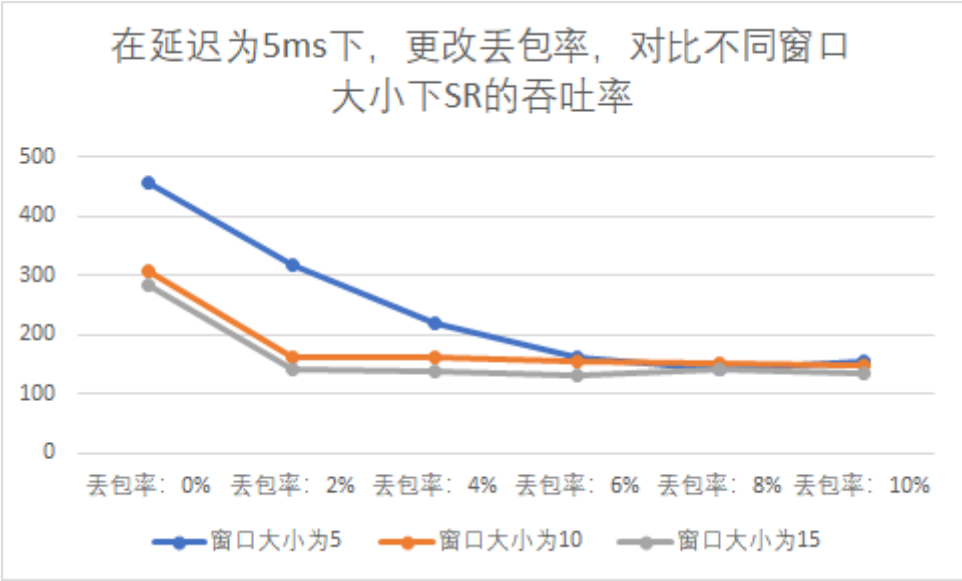
根据数据绘制图像：



吞吐率（单位：KB/s）

	丢包率： 0%	丢包率： 2%	丢包率： 4%	丢包率： 6%	丢包率： 8%	丢包率： 10%
窗口大小 为5	457.02	317.346	218.145	160.642	142.835	156.118
窗口大小 为10	308.228	162.718	160.898	154.825	150.195	148.607
窗口大小 为15	285.287	139.989	138.842	130.422	140.206	135.198

根据数据绘制图像：

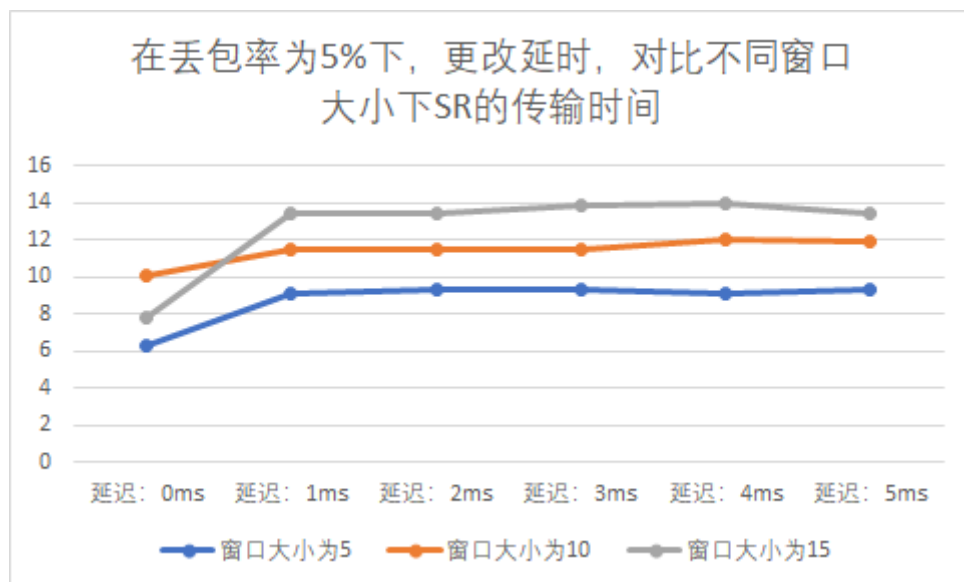


4.对于选择确认（即SR），将丢包率设为5%，更改延迟，比对传输时间和吞吐率

传输时间（单位：s）

	延迟： 0ms	延迟： 1ms	延迟： 2ms	延迟： 3ms	延迟： 4ms	延迟： 5ms
窗口大小为 5	6.293	9.064	9.305	9.313	9.077	9.286
窗口大小为 10	10.06	11.477	11.456	11.536	12.042	11.903
窗口大小为 15	7.823	13.444	13.403	13.911	13.991	13.406

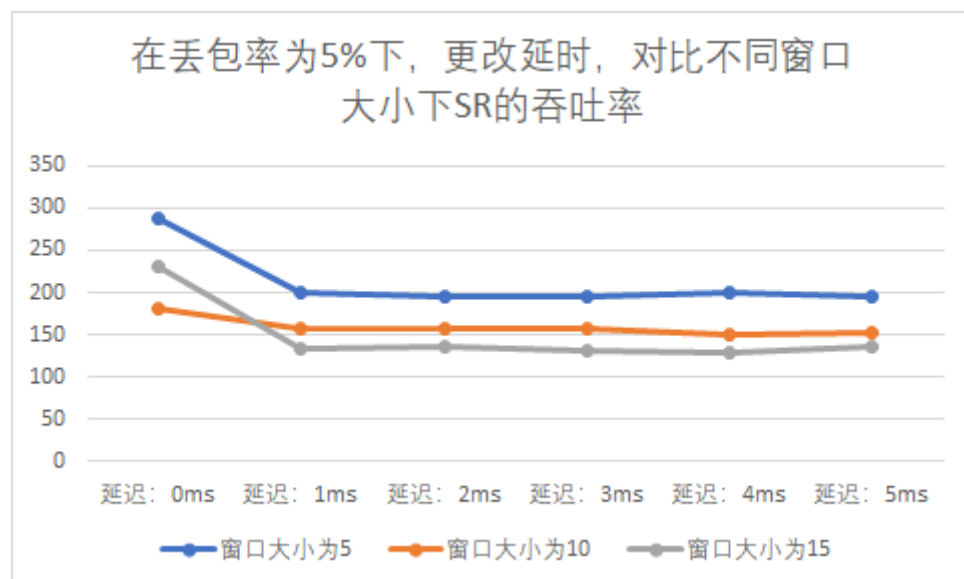
根据数据绘制图像：



吞吐率 (单位: KB/s)

	延迟: 0ms	延迟: 1ms	延迟: 2ms	延迟: 3ms	延迟: 4ms	延迟: 5ms
窗口大小为 5	288.098	200.022	194.841	194.674	199.736	195.24
窗口大小为 10	180.219	157.968	158.258	157.16	150.556	152.315
窗口大小为 15	231.753	134.856	135.268	130.329	129.583	135.238

根据数据绘制图像:



5.结果分析：

- 对于GBN来说：
 - 窗口大小为5的性能比窗口大小为10、15的性能更好。窗口大小为10和15的性能几乎是一样的。
 - 当丢包率越高时，窗口大小较大的下降幅度较大，窗口大小较小的下降幅度较小。
 - 一开始延迟增加时，不同窗口大小的GBN的性能都下降，且窗口大小为5的下降幅度较高。但是在后来GBN的性能几乎不受延迟增加的影响。
- 对于SR来说：
 - 窗口大小为5的性能>窗口大小为10的性能>窗口大小为15的性能。这应该和我代码实现逻辑有关，当窗口大小越大时，一次传的很多，那么接收方处理的压力就会较大，这就可能影响传输时间，而影响性能。
 - 但当丢包率较高时，三者的性能相差不多（即传输时间和吞吐率）。
 - 与GBN类似，一开始延迟增加时，不同窗口大小的SR的性能都下降，但窗口大小为15的下降幅度较高，窗口大小为10的下降幅度较缓。但是在后来SR的性能几乎不受延迟增加的影响。

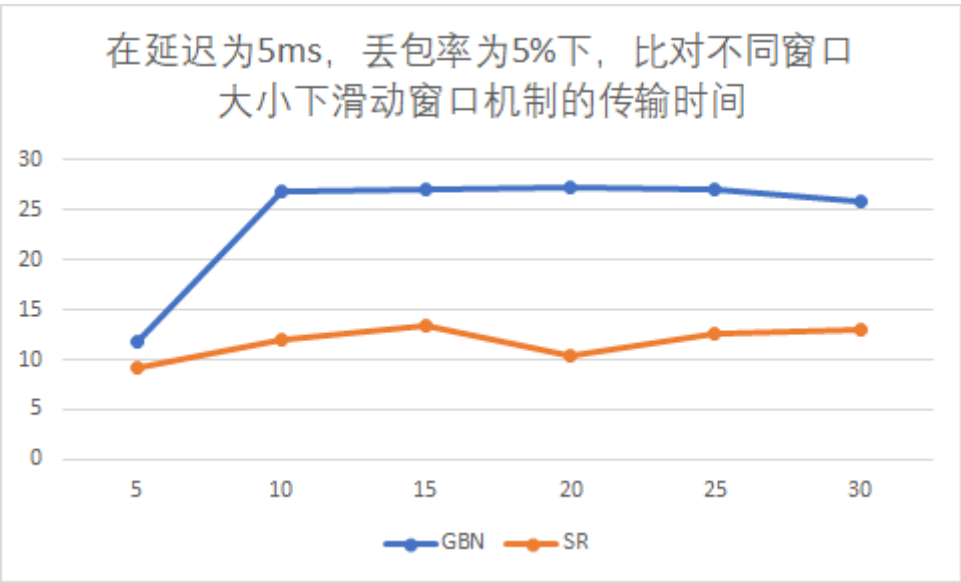
(3) 滑动窗口机制中相同窗口大小情况下，累计确认和选择确认的性能比较

1.将丢包率设为5%，延迟设为5ms，更改窗口大小，比对传输时间和吞吐率

传输时间（单位：s）

窗口大小	GBN	SR
5	11.785	9.286
10	26.782	11.903
15	27.097	13.406
20	27.327	10.474
25	26.992	12.597
30	25.909	13.028

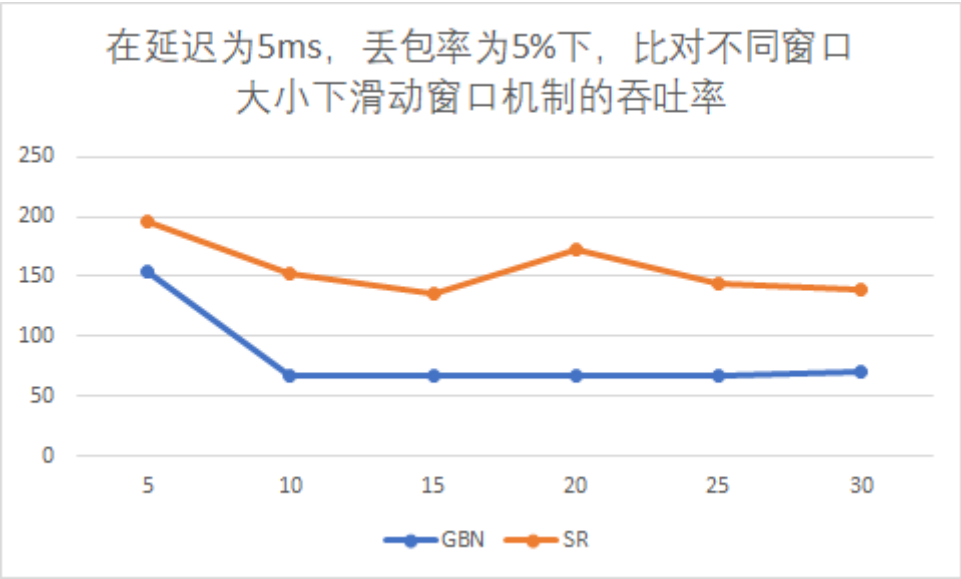
根据数据绘制图像：



吞吐率 (单位: KB/s)

窗口大小	GBN	SR
5	153.84	195.24
10	67.6947	152.315
15	66.9078	135.238
20	66.3446	173.095
25	67.168	143.923
30	69.9757	139.162

根据数据绘制图像:

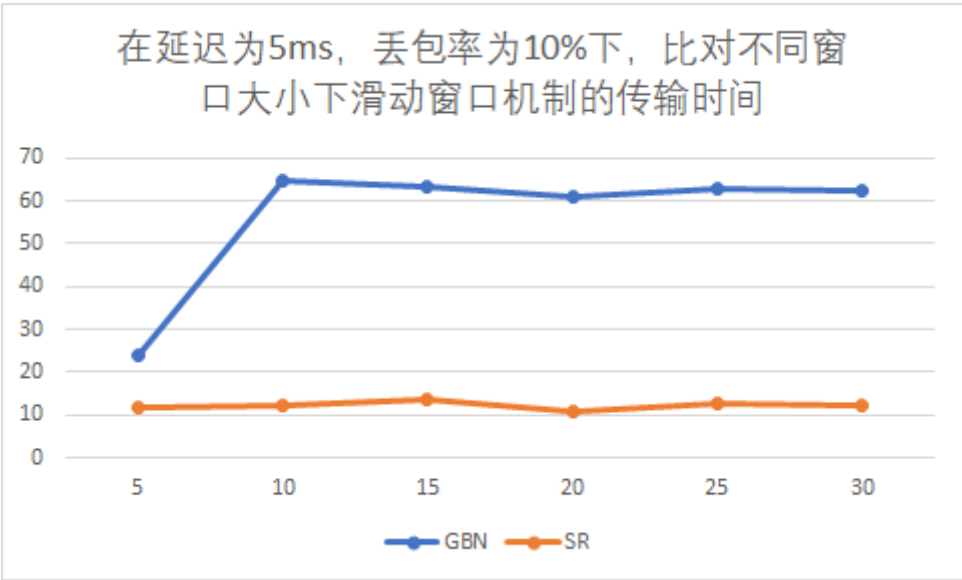


2.将丢包率设为10%，延迟设为5ms，更改窗口大小，比对传输时间和吞吐率

传输时间（单位：s）

窗口大小	GBN	SR
5	24.03	11.613
10	64.836	12.2
15	63.4	13.41
20	60.757	10.97
25	63.04	12.858
30	62.323	12.378

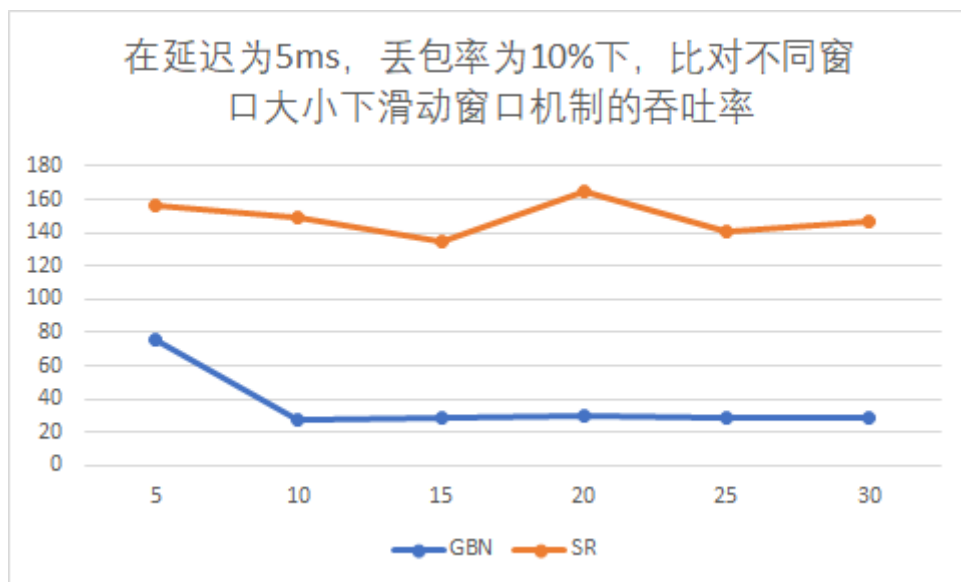
根据数据绘制图像：



吞吐率（单位：KB/s）

窗口大小	GBN	SR
5	75.447	156.118
10	27.9629	148.607
15	28.5962	135.198
20	29.8402	165.269
25	28.7595	141.002
30	29.0904	146.47

根据数据绘制图像：



3.结果分析:

- 可以看出，当丢包率越高时，SR比GBN的性能高的越多。
- 对于SR和GBN，性能最好的都是窗口大小为5时。当窗口大小为10时，性能出现明显下降，GBN下降幅度非常大。但是后来，随着滑动窗口大小增加，两者的性能都趋于稳定。

三、总结

- 本次实验是这学期最后一次实验，是对前面三次实验的总结。通过对三种机制的性能测试，让我更清楚地了解三种机制的区别。且对我自己写的代码的效率，有了更进一步的了解。