



南開大學
Nankai University

计算机网络

3-4 性能对比实验报告

姓名：高祎珂

学号：2011743

专业：计算机科学与技术

2022 年 12 月 30 日

目录

1 实验简介	2
2 停等机制与滑动窗口机制性能对比	2
2.1 分析	3
3 滑动窗口机制中不同窗口大小对性能的影响	4
3.1 分析	5
4 有拥塞控制和无拥塞控制的性能比较	5
4.1 分析	7

1 实验简介

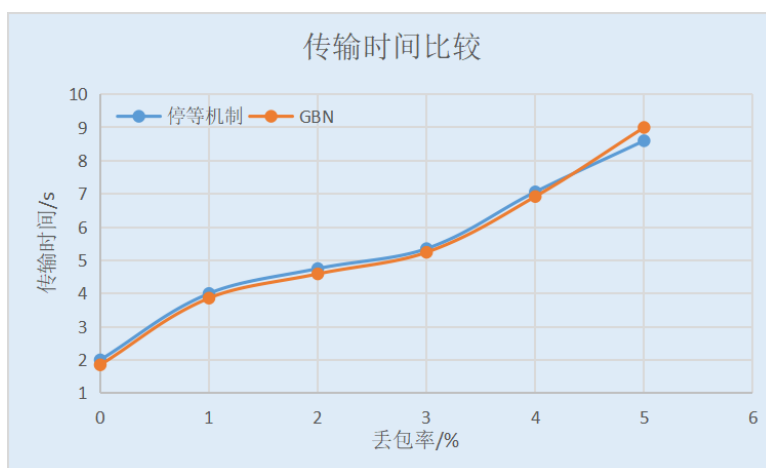
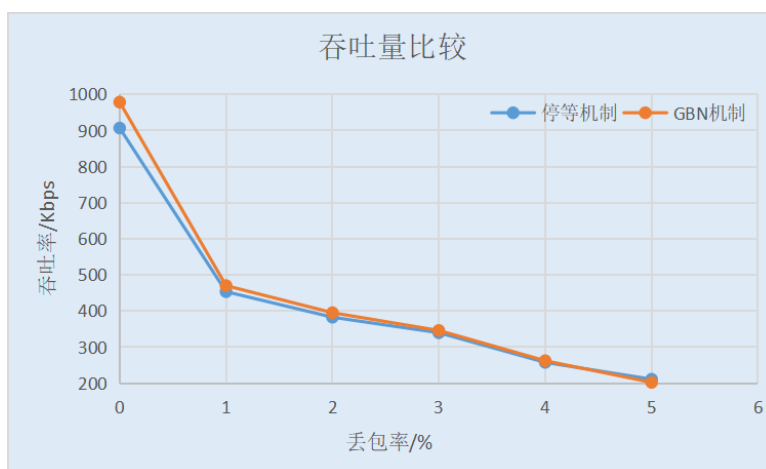
本次实验在 Windows 系统 X86 平台进行实验，使用编译器为 VS2022 使用的 MSVC，丢包使用所给的路由器程序。在实验设计过程中，对于不同的策略，保证每次传输的数据包最大都为 8160byte，设置超时时间均为 100ms，通过控制相同时延改变丢包率，和控制相同丢包率改变时延进行实验，进行三方面的结果对比。

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

2 停等机制与滑动窗口机制性能对比

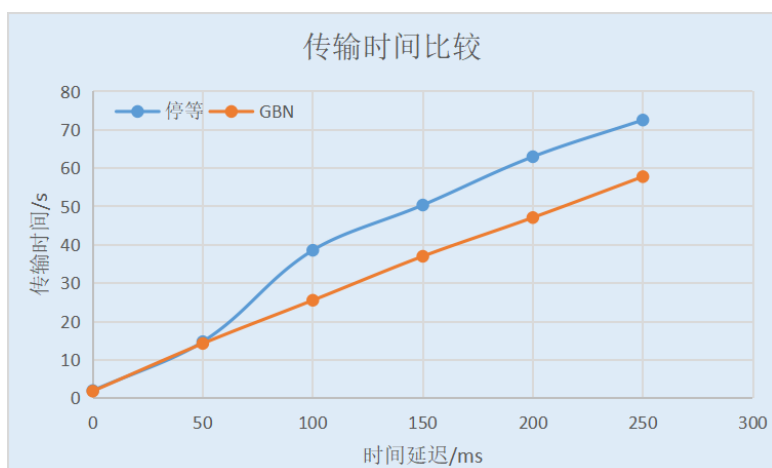
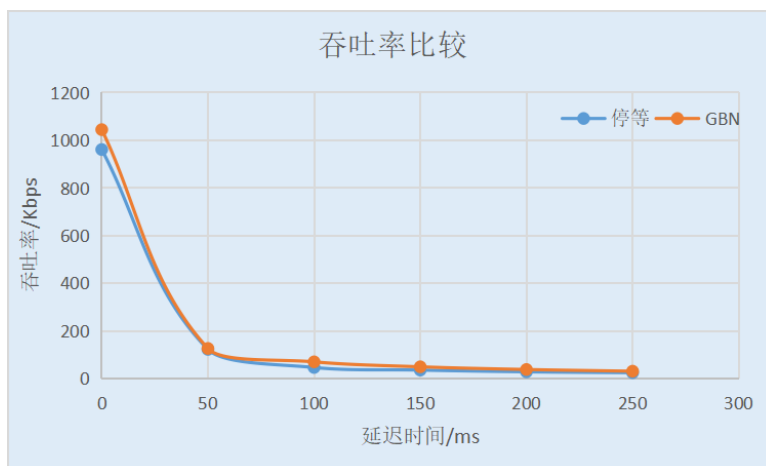
设置延迟时间为 0，改变丢包率

延迟时间为 0，不同丢包率不同机制的吞吐量，单位 Kbps						
机制	0%(不丢包)	1%	2%	3%	4%	5%
停等	906.56	453.4555	382.7815	339.78	257.71	211.155
滑动窗口 (GBN)	978.266	470.25	395.28	346.1295	262.3545	202.269



设置丢包率为 0，改变延迟时间

丢包率为 0，不同延迟时间不同机制的吞吐量，单位 Kbps						
机制	0ms	50ms	100ms	150ms	200ms	250ms
停等	906.825	122.832	47.02	36.1274	28.8211	25.0782
滑动窗口 (GBN)	978.26	126.833	70.5729	49.8144	38.5412	31.393



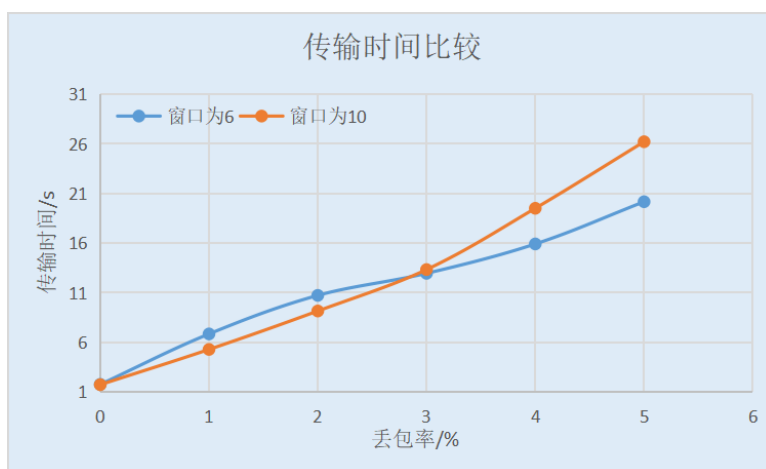
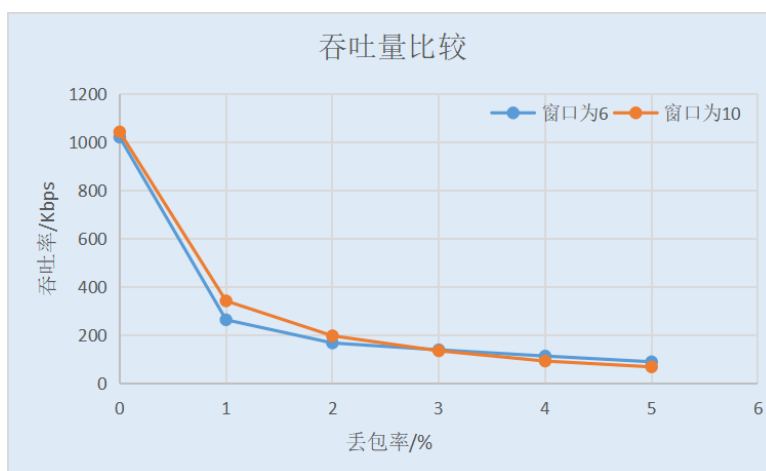
2.1 分析

- 总体上滑动窗口机制比停等机制的效率更高
 - 滑动窗口机制允许发送多条消息，同时等待对方回复的 ACK，减少 RTT 的影响
- 在有延时的情况下，GBN 表现更好
 - 原因同上，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
- 丢包率大时 GBN 效率比停低
 - 更大的窗口意味着更高的重传代价，大大降低性能

3 滑动窗口机制中不同窗口大小对性能的影响

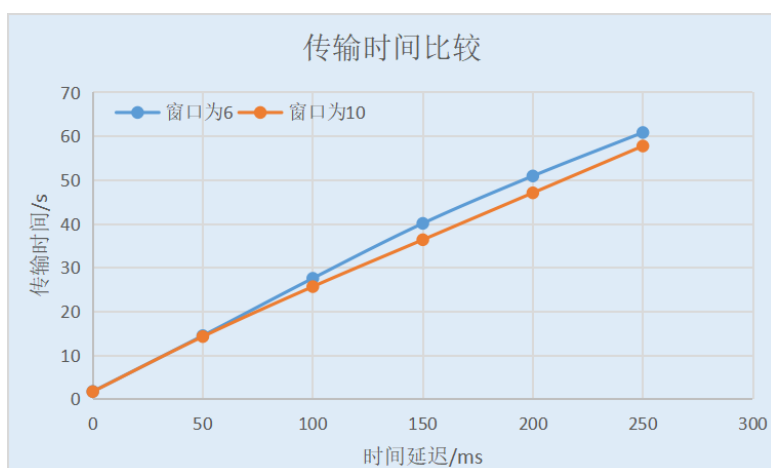
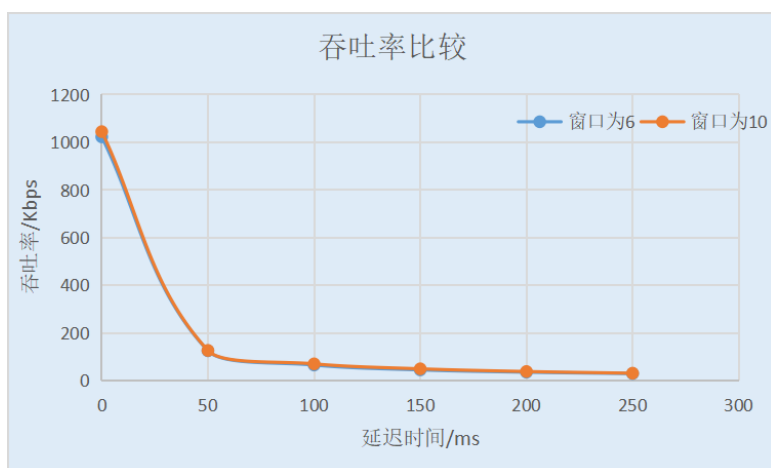
设置延迟时间为 0，改变丢包率

延迟时间为 0，不同丢包率不同窗口的吞吐量，单位 Kbps						
窗口大小	0%(不丢包)	1%	2%	3%	4%	5%
6	1022.322	264.897	168.695	140.271	114.535	90.6781
10	1044.26	343.323	198.482	136.259	93.7098	69.5384



设置丢包率为 0，改变延迟时间

丢包率为 0，不同延迟时间不同机制的吞吐量，单位 Kbps						
窗口大小	0ms	50ms	100ms	150ms	200ms	250ms
6	1022.32	125.453	65.6478	45.2397	35.629	29.7989
10	1044.26	126.833	70.5729	49.8144	38.5412	31.393



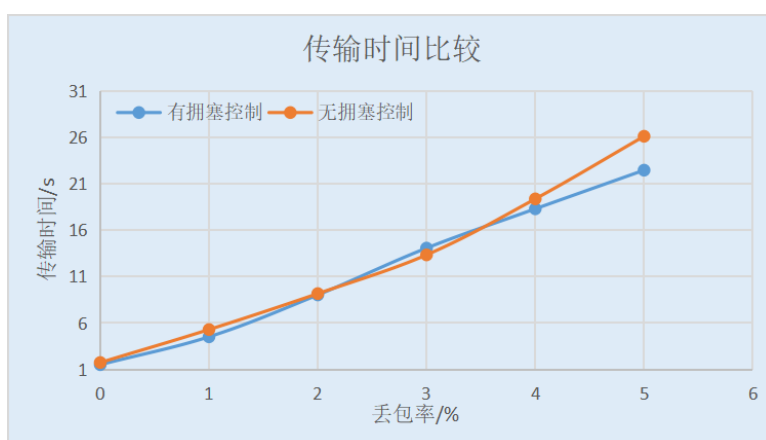
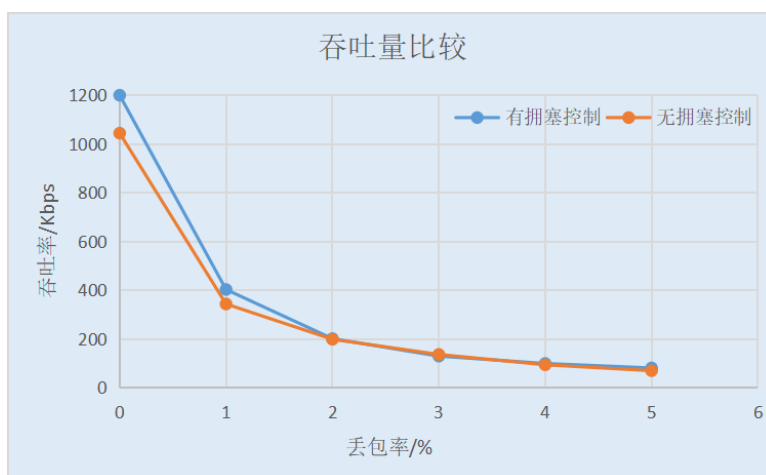
3.1 分析

- 不同窗口大小在不同网络环境下的效率变化总体上区域一致
- 在网络情况较好的时候窗口大的效率更高
 - 因为更大的窗口可以允许同时发送更多条消息并同时等待对方的 ACK，即减少等待的周期数，更好的应对时延问题。
- 对于丢包问题，当丢包率较高时由于大的窗口会增加重传代价，效率降低

4 有拥塞控制和无拥塞控制的性能比较

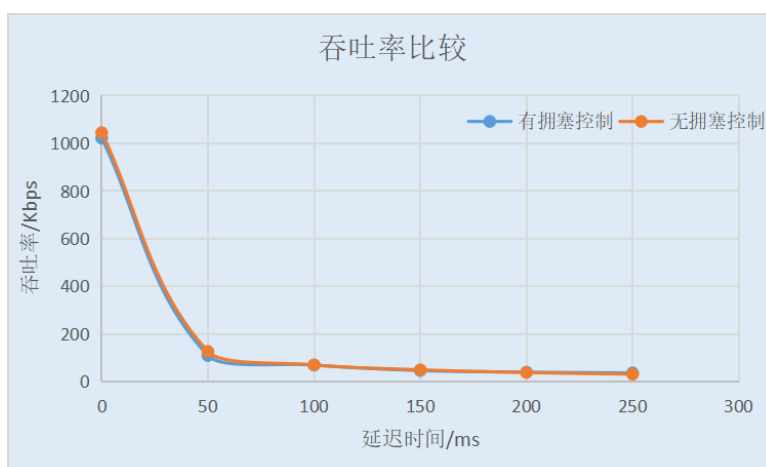
设置延迟时间为 0，改变丢包率

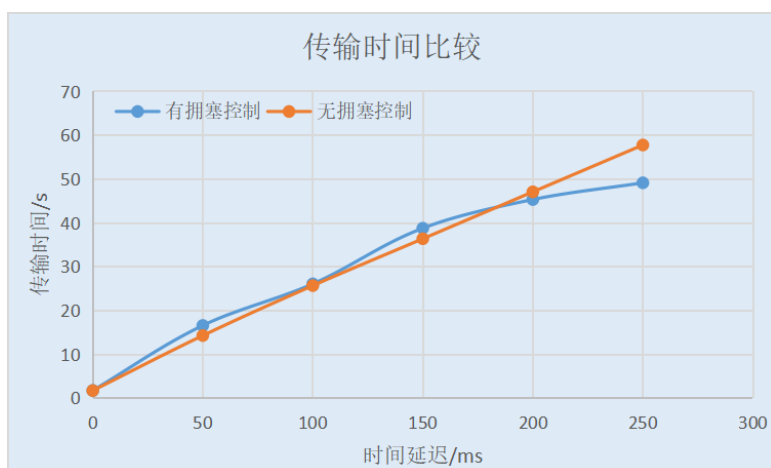
延迟时间为 0，不同丢包率有无拥塞控制的吞吐量，单位 Kbps						
拥塞控制	0%(不丢包)	1%	2%	3%	4%	5%
有	1200.11	402.673	201.129	129.408	99.187	80.806
无	1044.26	343.323	198.482	136.259	93.7098	69.5384



设置丢包率为 0，改变延迟时间

丢包率为 0，不同延迟时间有无拥塞控制的吞吐量，单位 Kbps						
拥塞控制	0ms(不延迟)	50ms	100ms	150ms	200ms	250ms
有	1021.178	109.642	69.5348	46.711	40.013	36.909
无	1044.26	126.833	70.5729	49.8144	38.5412	31.393





4.1 分析

- 在网络情况良好时，有拥塞控制相对于没有拥塞控制效率更高
 - 因为拥塞控制可以允许有更大的窗口
- 当网络情况逐渐变差时，拥塞控制机制表现可能并不比没有拥塞控制好
 - 频繁缩小窗口导致窗口较小，小于没有拥塞控制机制的设定值，导致受时延的影响较大
- 实验过程中也发现，当网络情况变得更差时，拥塞控制机制可能使窗口由几百突然降至 1，并进入重传，使得重传代价飞速增加。