

实验 3：基于 UDP 服务设计可靠传输协议并编程实现（任务 3-4）

姓 名： 杨科迪 学 号： 1813828
专 业： 计算机科学与技术 指导老师： 徐敬东
实验时间： 2020 年 12 月 25 日 实验地点： 实验楼 A 区 204

目录

1 实验目的	2
2 实验环境	2
3 实验数据	2
4 实验评价指标	2
5 实验结果	2
5.1 停等机制和滑动窗口机制性能对比	2
5.2 不同滑动窗口大小性能对比	3
5.3 有拥塞控制和无拥塞控制的性能对比	4
6 实验结论	4

1 实验目的

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 实验环境

- 所有的算法都是用 C++ 实现的
- 编译器:g++
- CPU: Intel Core i7-8550U (1.99GHZ)
- 内存:8GB
- OS: Windows 10
- 硬盘:512GB SSD

3 实验数据

大小为 1655808Byte 的传输文件

4 实验评价指标

方法的性能是通过传输时延和吞吐率来衡量的。实验时，将文件传输 5 次，传输时延和吞吐率取平均值

5 实验结果

5.1 停等机制和滑动窗口机制性能对比

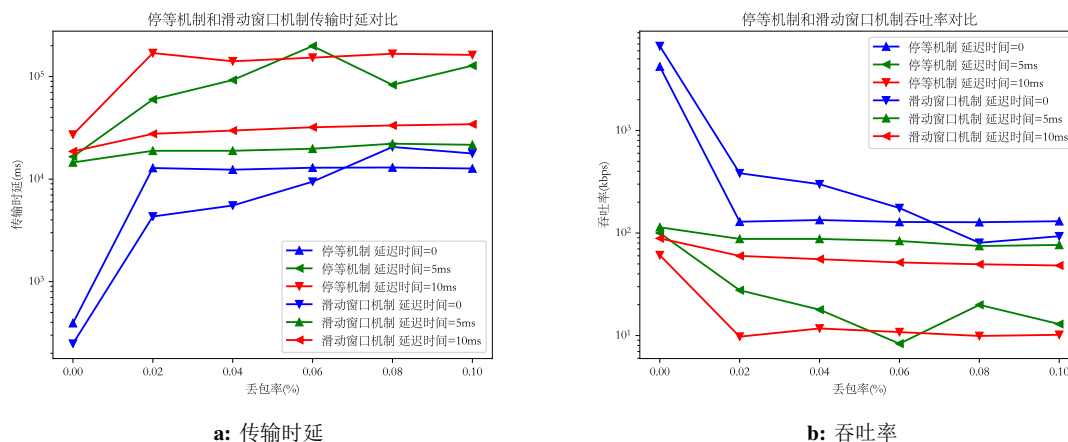


图 1: 停等机制和滑动窗口机制性能对比。图a为两种机制传输时延的比较，图b为两种机制吞吐率的比较。

图1为停等机制和滑动窗口机制性能对比，滑动窗口机制的窗口大小为 32。可以看出，在延迟时间为 5ms 和 10ms 下，滑动窗口机制的性能在不同的丢包率下均明显优于停等机制，大约有一个数量级的差别；但在延迟时间为 0 时，滑动窗口机制和停等机制的性能差别不明显，在丢包率大于 0.06% 时，停等机制还略优于滑动窗口机制。

5.2 不同滑动窗口大小性能对比

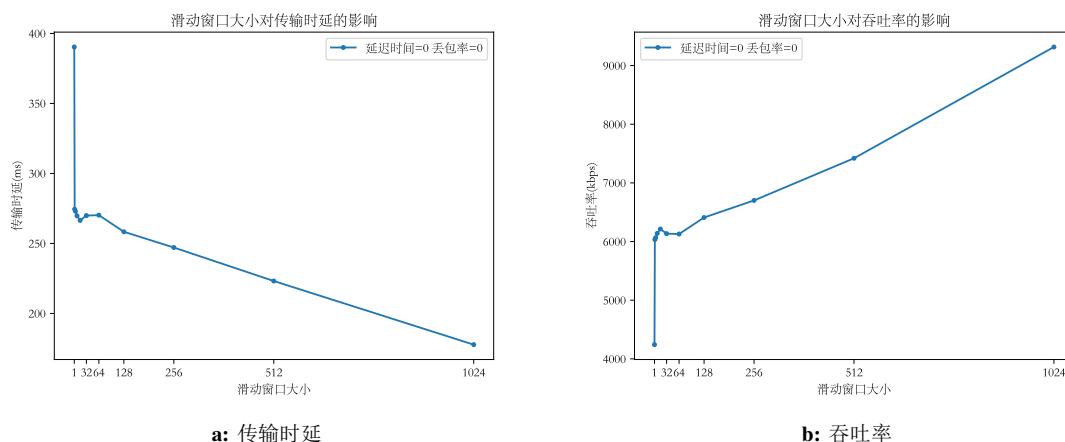


图 2: 不同滑动窗口大小性能对比 (延迟时间 = 0, 丢包率 = 0)。图a为传输时延的对比，图b为吞吐率的对比。

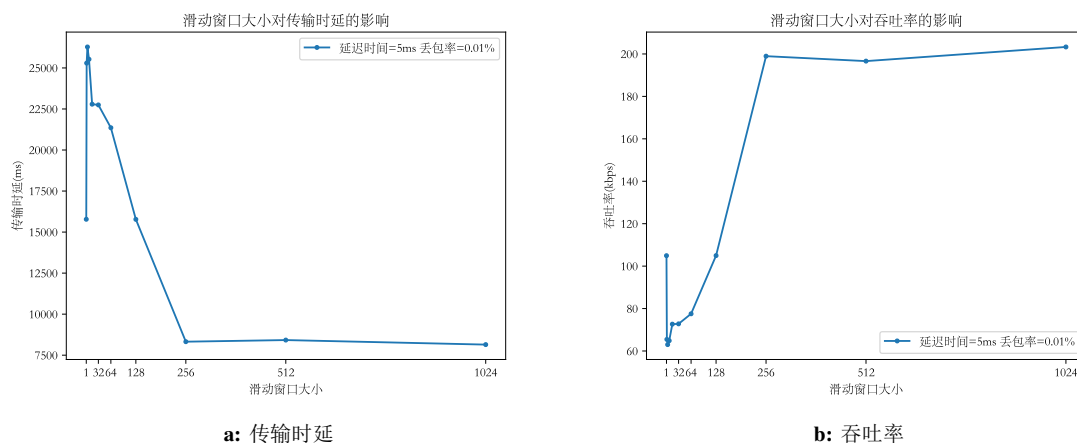


图 3: 不同滑动窗口大小性能对比 (延迟时间 = 5ms, 丢包率 = 0.01%)。图a为传输时延的对比，图b为吞吐率的对比。

延迟时间为 0，丢包率为 0 时，不同窗口大小的性能对比如图2所示，延迟时间为 5ms，丢包率为 0.01% 时，不同窗口大小的性能对比如图3所示，滑动窗口机制带有拥塞控制。

在延迟时间为 0，丢包率为 0 时，滑动窗口大小由一到多时，性能有非常大的提升，之后性能随滑动窗口大小的增加而提高，基本上呈现线性关系。

在延迟时间为 5ms，丢包率为 0.01% 时，滑动窗口大小由一到多时，性能有所下降，之后性能随着滑动窗口大小的增加而提高，但在滑动窗口大小为 256 之后性能不再提升。可能是因为拥塞控制机制，有丢包情

况发生，超时后进入慢启动状态，拥塞窗口大小变为 1，发送的报文数是滑动窗口大小和拥塞窗口大小的最小值，因此受限于拥塞窗口大小，滑动窗口大小增加到一定程度后，性能不再提升。

5.3 有拥塞控制和无拥塞控制的性能对比

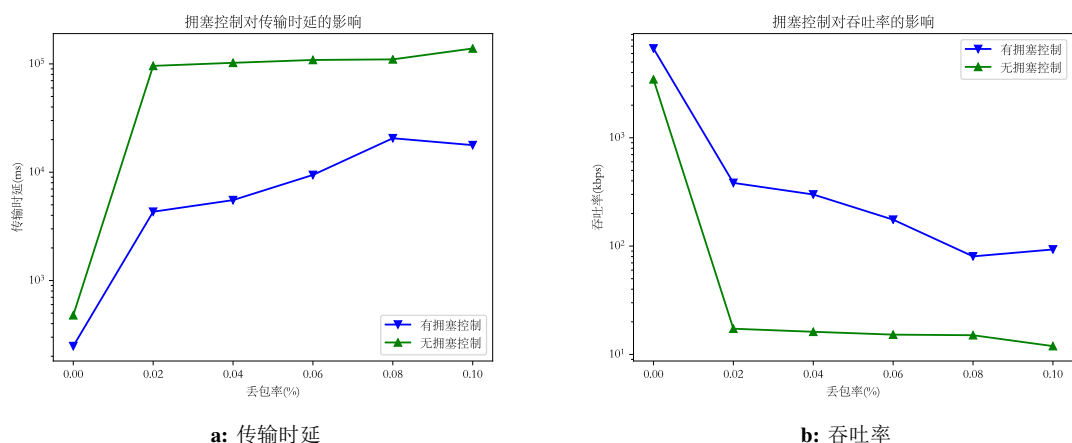


图 4: 有拥塞控制和无拥塞控制性能对比。图a为两种机制传输时延的对比，图b为两种机制吞吐率的对比。

有拥塞控制和无拥塞控制性能对比如图4所示。在丢包率为 0 时，性能差距不大，有丢包率之后，有拥塞控制较无拥塞控制性能约有一个数量级的提升。在无拥塞控制时，发生丢包后还会继续发送，使得拥塞更为严重，因此拥塞控制能够提升性能。

6 实验结论

1. 滑动窗口机制显著优于停等机制
2. 一般情况下，滑动窗口大小在一定范围内增大时，性能也随之提高；滑动窗口大小增大到一定程度后，性能不再提升
3. 有拥塞控制时的性能优于无拥塞控制时的