



南开大学
Nankai University

南 开 大 学

网 络 空 间 安 全 学 院

计算机网络实验报告

编程作业 3 基于 UDP 服务设计可靠传输协议并编程
实现₄

于文明

年级：2020 级

专业：信息安全

指导教师：徐敬东，张建忠

2022 年 12 月 30 日

摘要

关键字：socket UDP rdt3.0

目录

一、 停等机制和滑动窗口性能对比	1
(一) 丢包率影响	1
(二) 延迟时间影响	2
二、 滑动窗口机制中不同窗口大小对性能的影响	2
三、 有无拥塞控制的性能对比	3
(一) 丢包率影响	3
(二) 延迟时间影响	4
四、 附录	5

一、 停等机制和滑动窗口性能对比

(一) 丢包率影响

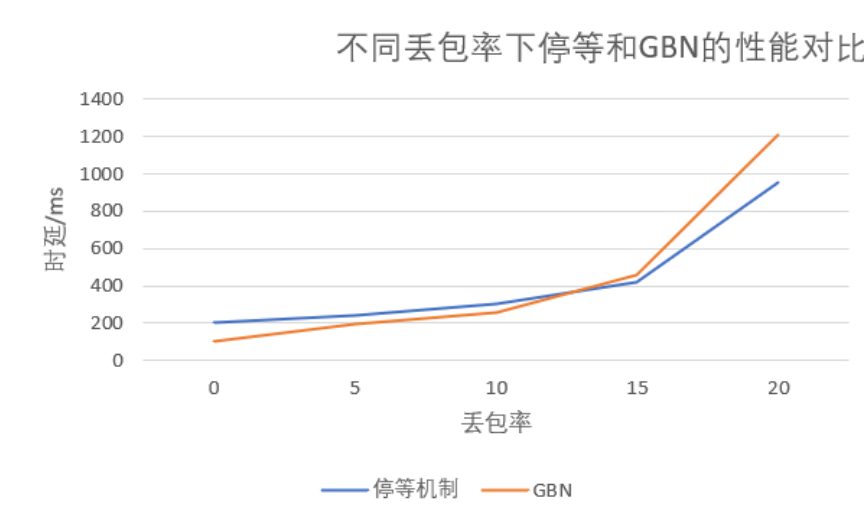


图 1: 停等机制和 GBN 在不同丢包率下的时延对比

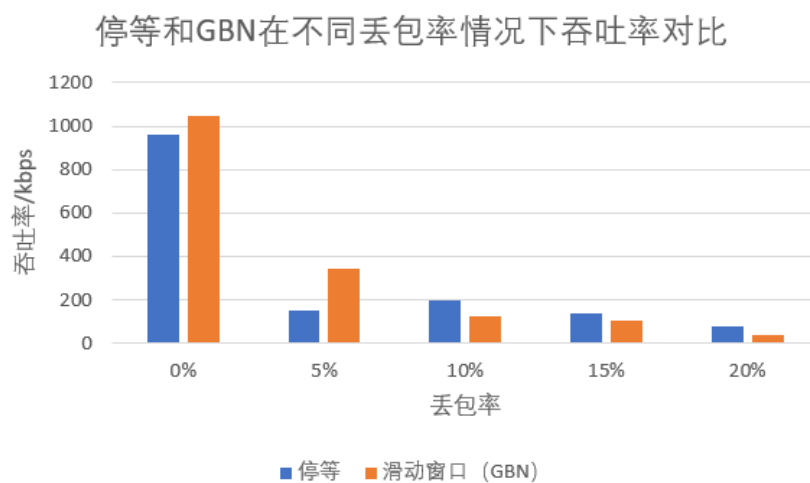


图 2: 停等机制和 GBN 在不同丢包率下的吞吐率对比

分析

- 大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

(二) 延迟时间影响

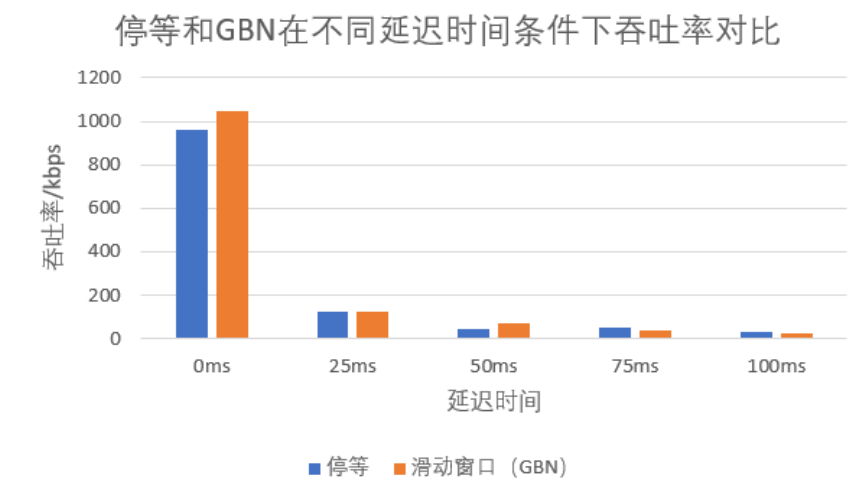


图 3: 停等机制和 GBN 在不同延迟时间下的吞吐率对比

二、 滑动窗口机制中不同窗口大小对性能的影响

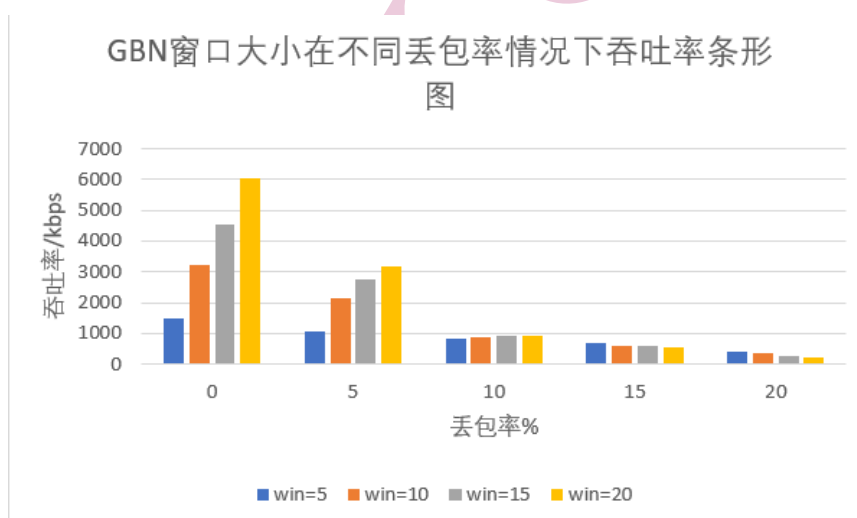


图 4: 滑动窗口机制中不同窗口大小对性能的影响

分析

从图表可以看出，窗口大小对于吞吐量的影响情况会收到丢包率的影响。当丢包率较小时，吞吐率随着窗口大小的增加而提升，这是因为丢包率小的时候，影响 GBN 效率的不必要重传很少发生。

而随着丢包率的上升，随着窗口大小增长性能反而降低。这是因为丢包率较大时，会产生大量超时重传，窗口越大，重传的开销越大，性能越差。

三、 有无拥塞控制的性能对比

(一) 丢包率影响

有无拥塞控制在不同丢包率情况下的时延折线图

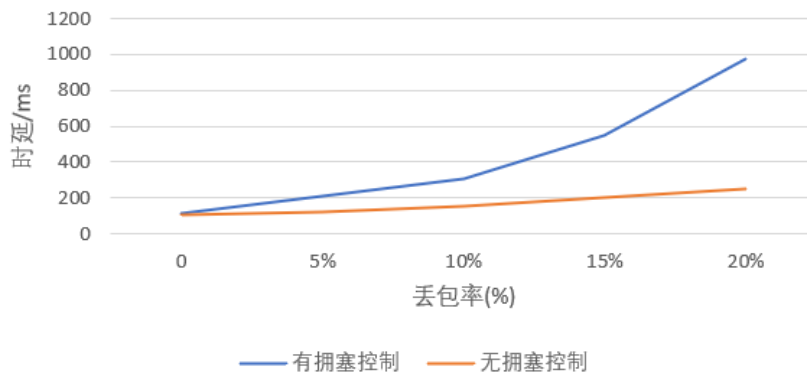


图 5: 有无拥塞控制在不同丢包率下的时延对比折线图

有无拥塞控制在不同丢包率下的吞吐率对比

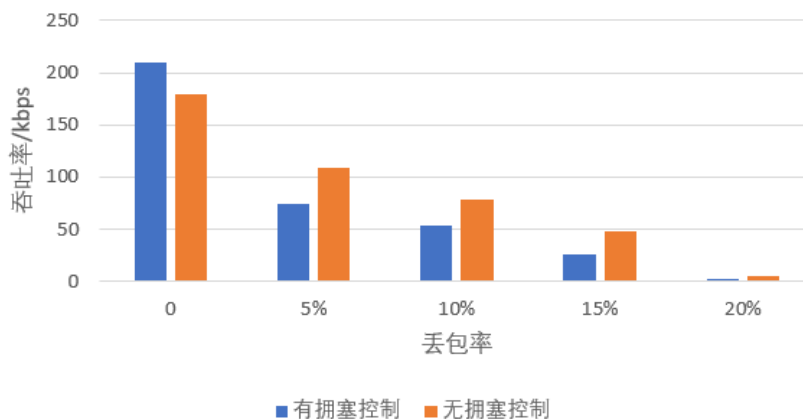


图 6: 有无拥塞控制在不同丢包率下的吞吐率对比图

分析

从图表中可以看出, 增加拥塞控制后, 性能有很大降低。原因如下: 对于无拥塞控制的算法, 超时后会重传所有未确认的数据报文, 但是并不会改变窗口的大小, 因此在后续中还能保持一个较大的吞吐量。相反对于有拥塞控制的 RENO 算法, 如果遇到丢包, 三次重复 ACK 会提前超时, 此时窗口大小折半, 如果遇到超时重传, 窗口大小甚至会被设置为 1, 会影响到吞吐量。

并且随着丢包率不断增加, 有拥塞控制的算法性能会快速下降, 这是因为频繁出现冗余 ACK 会导致窗口始终维持在一个较低的水平, 严重影响性能。

(二) 延迟时间影响

在本次实验中，固定丢包率为 0，设置不同的延迟时间，测量平均时延，比较有无拥塞控制的性能差异

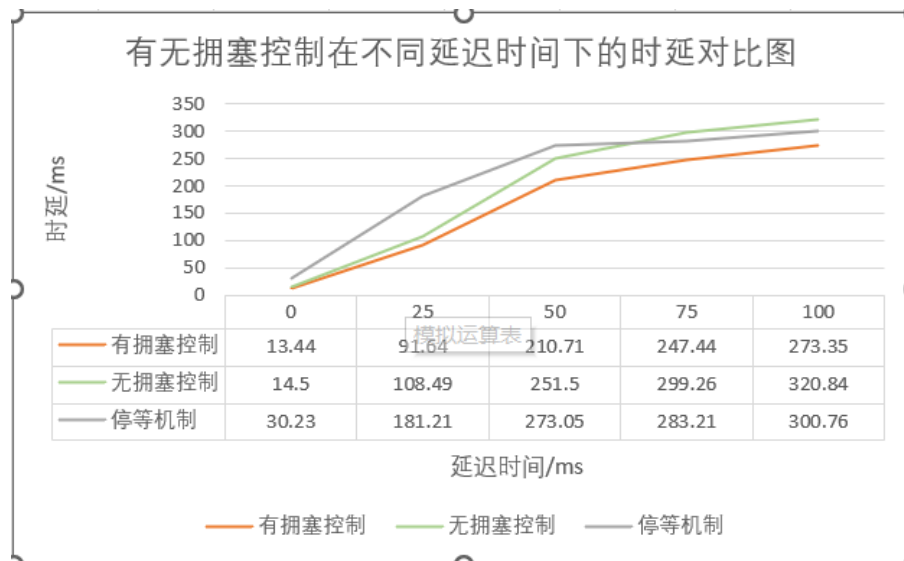


图 7: 时延对比折线图

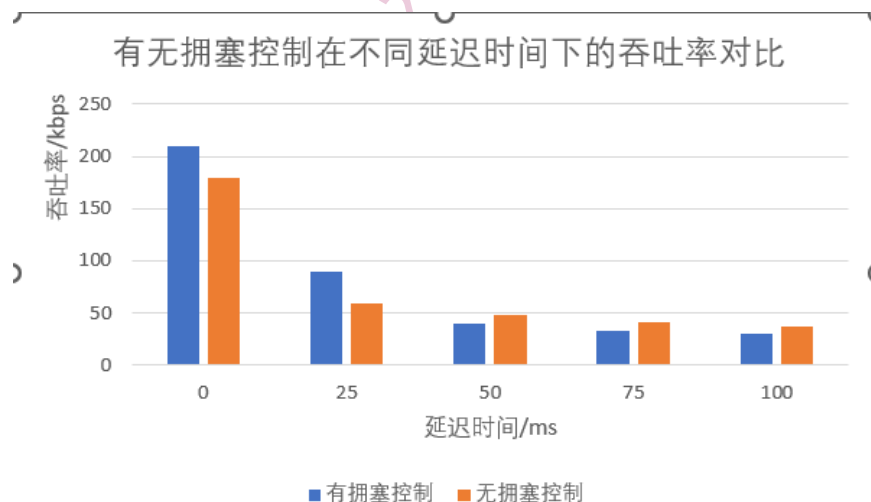


图 8: 有无拥塞控制下的时延对比图

为了能够对于拥塞控制的性能提升有更直观的理解，我在实验中额外对停等机制进行了对比。

可以看到当延迟时间较小的时候，有拥塞和无拥塞控制的性能十分接近，同时由于流水线机制的存在，两者的性能都优于停等机制。随着延迟时间的增加，超过某个 RTO 时，会发生超时重传。无拥塞控制的算法超时会重传窗口内的所有未确认的报文，会造成一定的性能损失。而有拥塞控制的算法会在超时重传时调整窗口大小，减小由于超时重传造成的性能开销。所以在结果中表现为随着延迟时间的增加，无拥塞控制的算法性能甚至比停等机制还差，而有拥塞控制的算法优于停等机制。

四、 附录

本实验相关的资源 (源文件,测试文件,实验结果截图等) 都已上传至 github(<https://github.com/Metetor/network>),

NIKU

参考文献

NKUD