



南开大学  
Nankai University

南 开 大 学

## 计算机网络实验报告

---

### 实验 3-4

---

姓名：李雅帆

学号：2213041

年级：2022 级

专业：信息安全

2024 年 12 月 21 日

## 目录

|                         |    |
|-------------------------|----|
| 一、 实验要求                 | 1  |
| 二、 停等机制与滑动窗口机制性能对比      | 2  |
| (一) 单一变量: 丢包率 . . . . . | 2  |
| (二) 单一变量: 时延 . . . . .  | 4  |
| 三、 滑动窗口机制中不同窗口大小对性能的影响  | 6  |
| (一) 网络情况较好 . . . . .    | 6  |
| (二) 网络情况较差 . . . . .    | 9  |
| 四、 有拥塞控制和无拥塞控制的性能比较     | 11 |
| (一) 单一变量: 时延 . . . . .  | 11 |
| (二) 单一变量: 丢包率 . . . . . | 13 |
| 五、 总结                   | 15 |

## 一、 实验要求

实验 3-4: 基于给定的实验测试环境, 通过改变网络的延迟时间和丢包率, 完成下面 3 组性能对比实验:

- 停等机制与滑动窗口机制性能对比;
- 滑动窗口机制中不同窗口大小对性能的影响;
- 有拥塞控制和无拥塞控制的性能比较。
- 本次实验以 2.jpg 为测试文件。
- 实验中丢包率的单位为%, 吞吐率的单位为 Byte/s, 时延的单位为 ms。实验过程主要通过控制变量法进行性能对比测试。

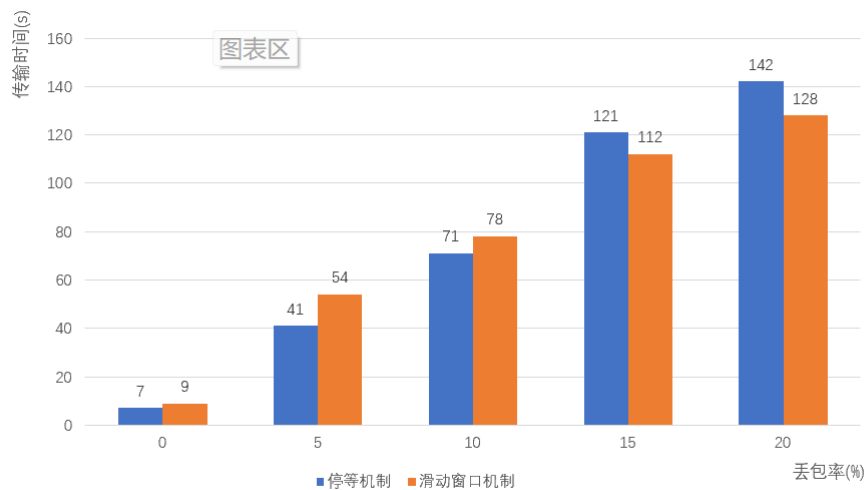
NIUB

## 二、 停等机制与滑动窗口机制性能对比

### (一) 单一变量：丢包率

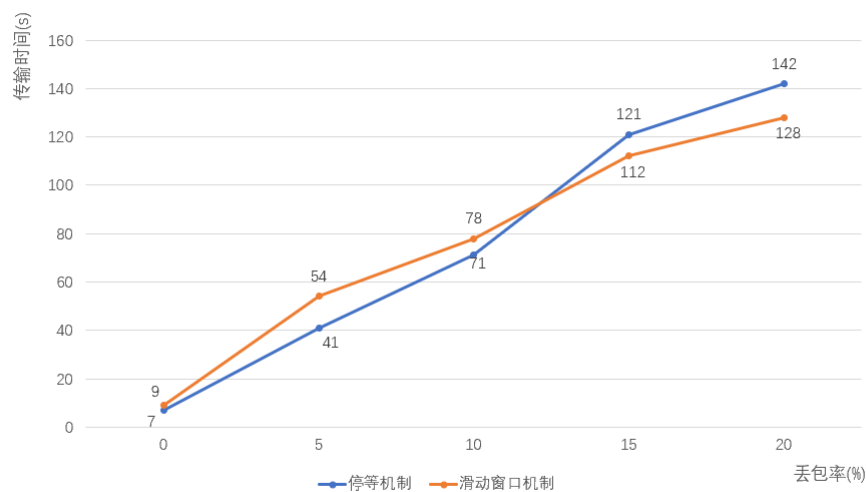
设置时延为 0，改变丢包率，比较停等机制和滑动窗口机制的时延和吞吐率。

#### 1. 图形化分析



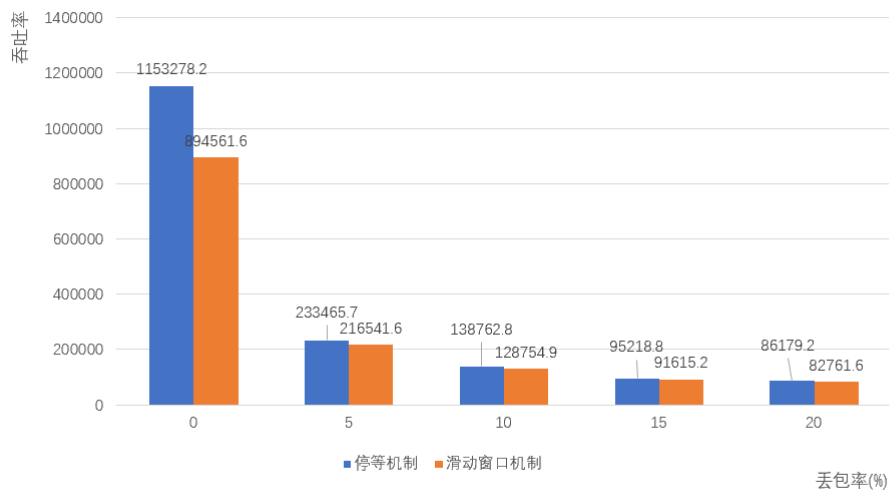
不同丢包率下传输时间对比柱状图

图 1: 时延为 0



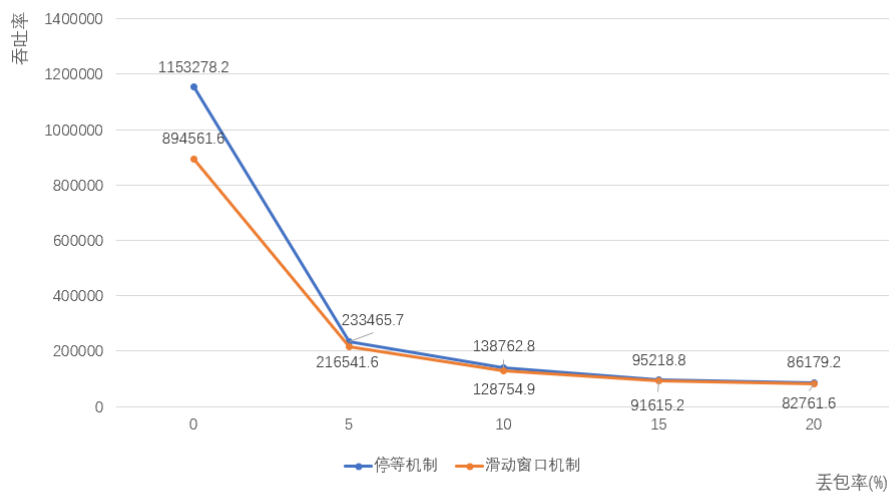
不同丢包率下传输时间对比折线图

图 2: 时延为 0



不同丢包率下吞吐量对比柱状图

图 3: 时延为 0



不同丢包率下吞吐量对比折线图

图 4: 时延为 0

## 2. 结果分析

- 传输总时长: 观察柱状图, 可以看出, 在相同延时的情况下, 低丢包率情况下, 停等协议优于滑动窗口机制; 高丢包率情况下, 滑动窗口和停等协议的性能差不多, 并且滑动窗口的性能略微差于停等协议。这是因为, 由于滑动窗口采用的是 GBN 协议, 当丢包率较高时, 由于 GBN 协议每次都要将发送缓存区中的分组全部重发, 所以传输时间反而比停等协议更长。
- 吞吐量: 在无丢包的情况下, 停等机制的吞吐量高于滑动窗口机制。但随着丢包率的

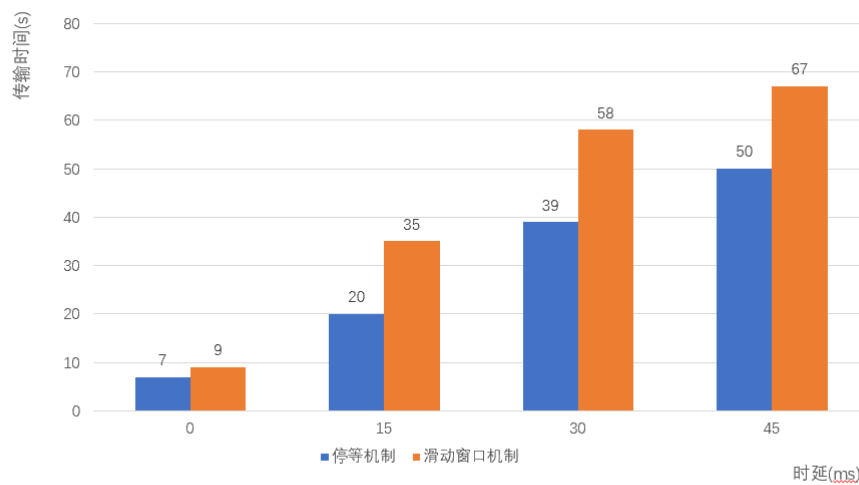
增加，两者的吞吐率都显著下降，且停等机制的下降幅度更大。

- 性能差异分析；在该实验条件下传输速度主要取决于接收端的接受速度，在发送端发送数据后，接收端需要等待和停等协议接近相同的超时时间接收丢失的包，因此滑动窗口和停等协议传输速度并没有相差多少，但是滑动窗口使得接收端不得不处理失序的包，所以耽误了更多的时间。更大的窗口意味着更高的重传代价，将大大降低滑动窗口性能。

## (二) 单一变量：时延

设置丢包率为 0，改变时延，比较停等机制和滑动窗口机制的时延和吞吐率。

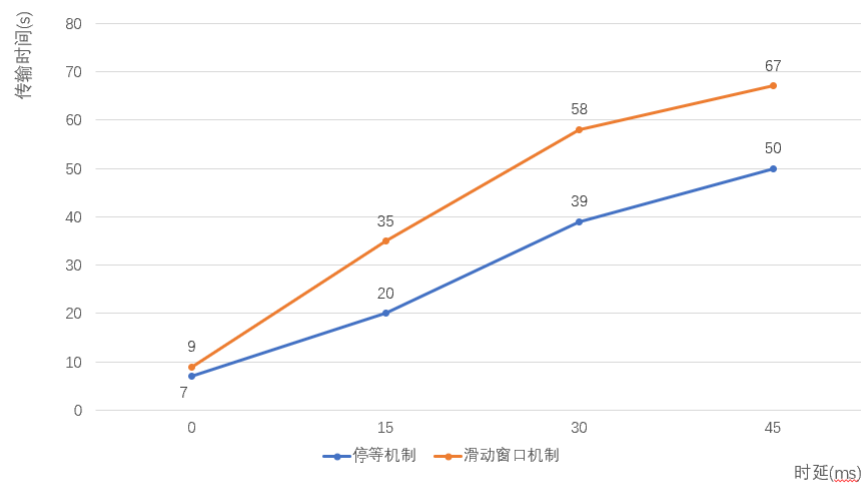
### 1. 图形化分析



不同时延下传输时间对比柱状图

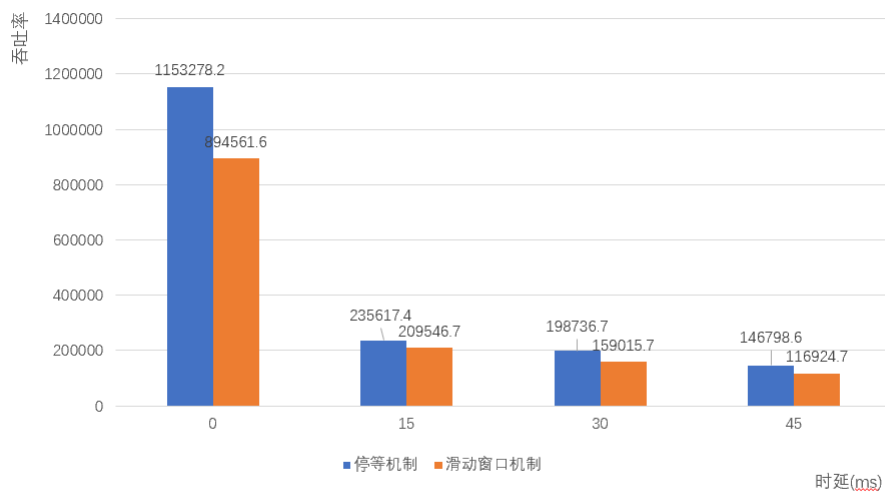
图表反 I

图 5: 丢包率为 0



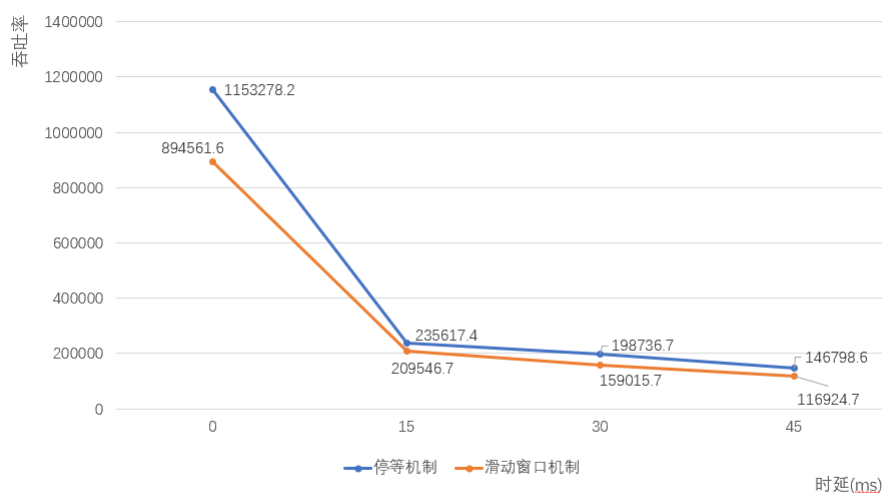
不同时延下传输时间对比折线图

图 6: 丢包率为 0



不同时延下吞吐量对比柱状图

图 7: 丢包率为 0



不同延迟下吞吐量对比折线图

图 8: 丢包率为 0

## 2. 结果分析

- 传输总时长：在无时延情况下，滑动窗口的传输总时长略长于停等机制。这是因为，当时延超过阈值时会发生超时重传现象，停等机制会重发该分组，但滑动窗口机制会重发窗口内所有未确认的分组，相对于停等机制效率会低。
- 吞吐量：在无时延的情况下，停等机制的吞吐量高于滑动窗口协议。但随着时延的增加，两种协议的吞吐量都显著降低，且下降幅度相似。这表明时延对两种协议的影响是类似的。
- 性能差异分析；在无时延的情况下，停等机制在吞吐量上优于滑动窗口。然而，随着时延的增加，两种协议的性能差异逐渐减小。这表明在高时延环境中，滑动窗口协议的稳定性稍好于停等机制。

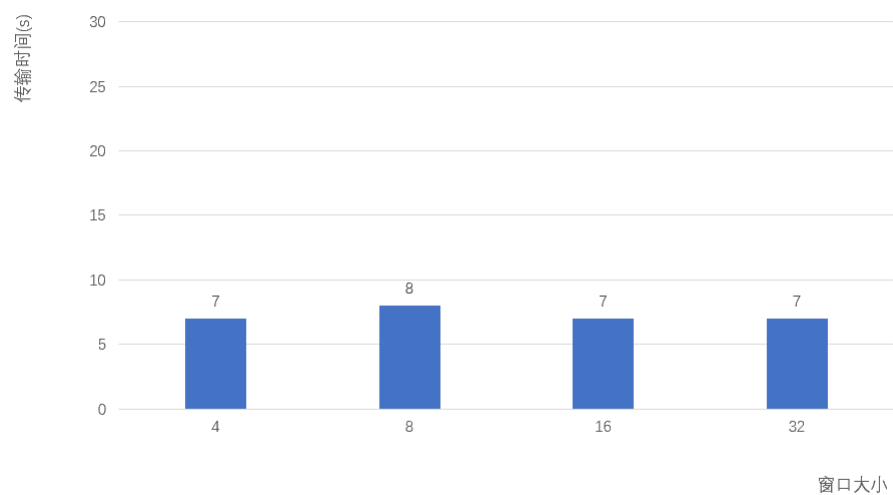
## 三、 滑动窗口机制中不同窗口大小对性能的影响

### (一) 网络情况较好

分析在网络环境较好时，在 0 时延和 0 丢包率条件下，滑动窗口大小对性能的影响。

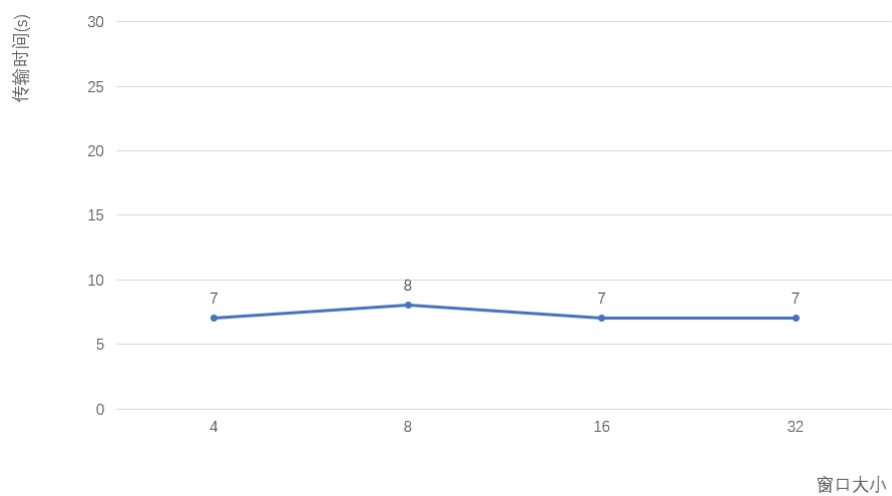
#### 1. 图形化分析





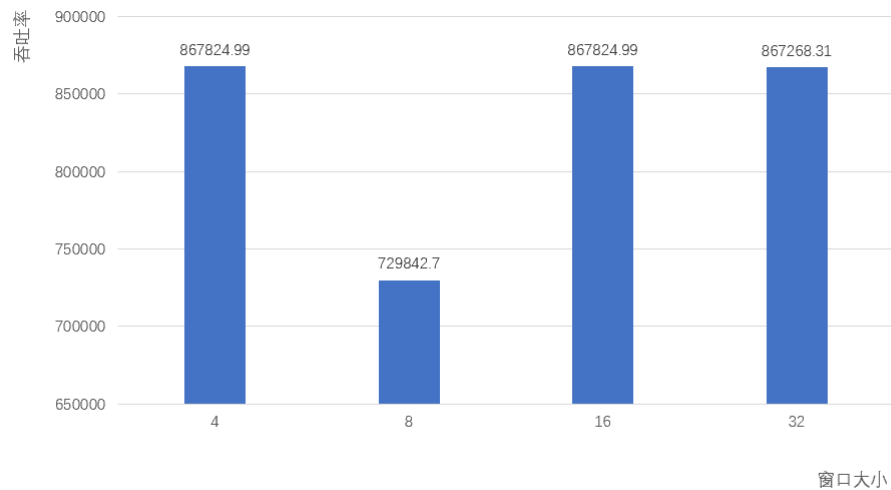
不同窗口大小传输时间对比柱状图

图 9



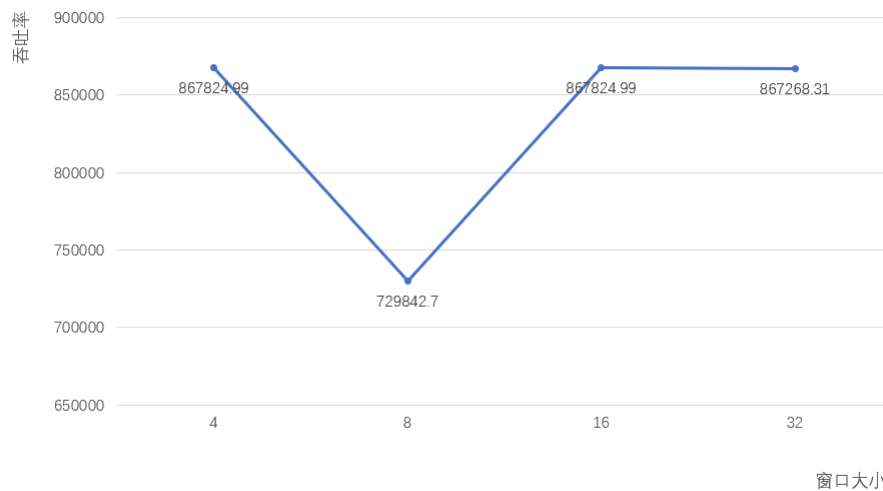
不同窗口大小下传输时间对比折线图

图 10



不同窗口大小下吞吐率对比柱状图

图 11



不同窗口大小下吞吐率对比折线图

图 12

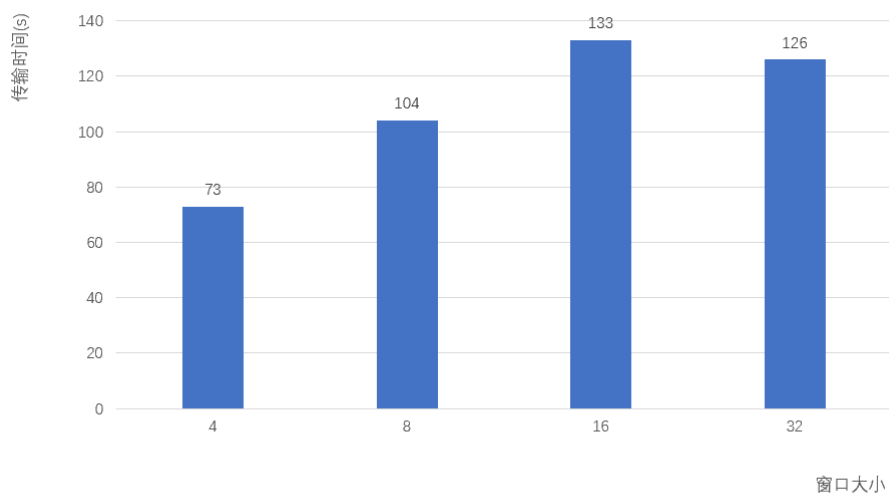
## 2. 结果分析

- 在窗口大小由 4 增加到 32 的过程中，传输时间和吞吐率变化都不大。
- 分析：当窗口大小较大时，制约传输速度的因素主要是接收端接收的速度。如果接收端接收的速度快，发送端接收到 ACK 的时间变短，当前窗口大小长时间不到最大窗口大小，窗口大小的效用就不显著，因此单纯的增大窗口大小无益于增加传输速度，提高性能。本身将每个数据包的大小增加，数据包个数减少，以本次实验为例，总共传输了 576 个数据包，结果不显著。

## (二) 网络情况较差

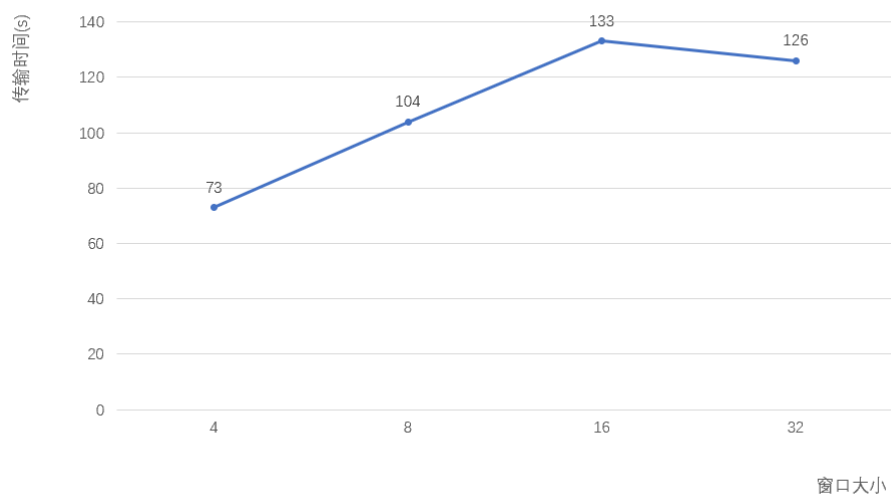
在丢包率为 10%，延时为 30ms 的情况下进行实验。

### 1. 图形化分析



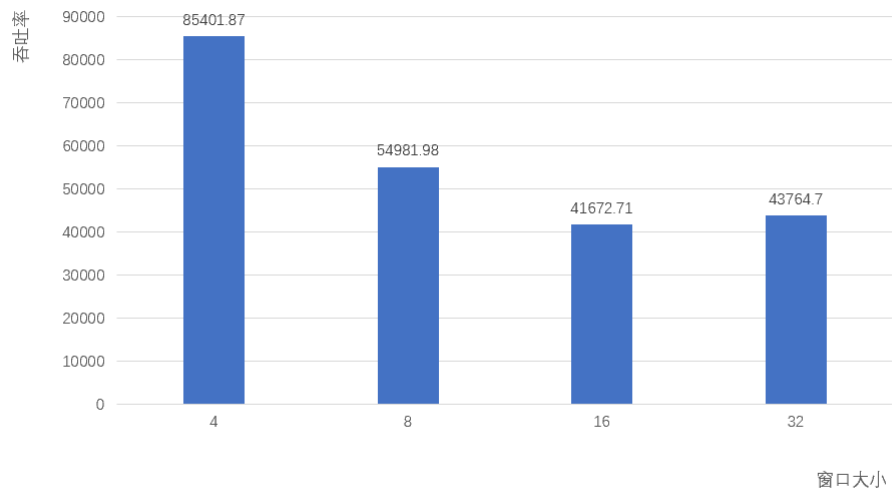
不同窗口大小下传输时间对比柱状图

图 13



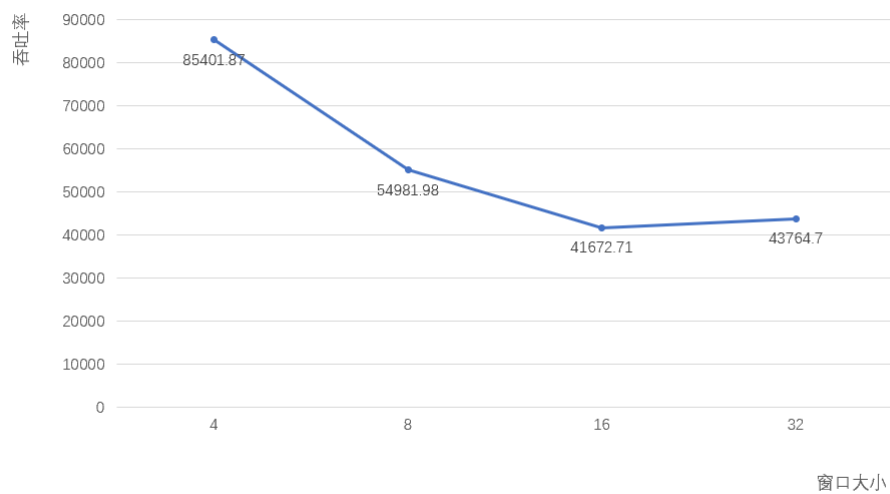
不同窗口大小下传输时间对比折线图

图 14



不同窗口大小下吞吐率对比柱状图

图 15



不同窗口大小下吞吐率对比折线图

图 16

## 2. 结果分析

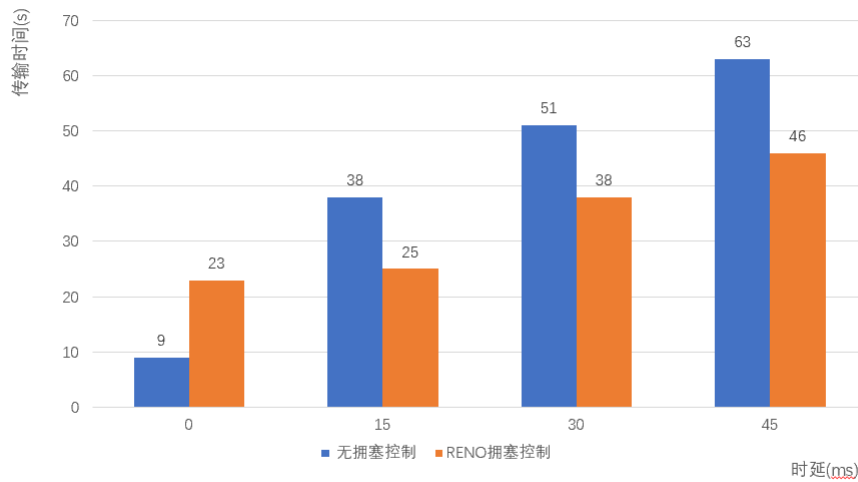
- 传输总时长：网络情况较差时，在窗口大小由 4 增加到 32 的过程中，传输时间不断升高。
- 吞吐率：网络情况较差时，在窗口大小由 4 增加到 32 的过程中，传输时间不断升高。吞吐率不断降低。

## 四、有拥塞控制和无拥塞控制的性能比较

### (一) 单一变量：时延

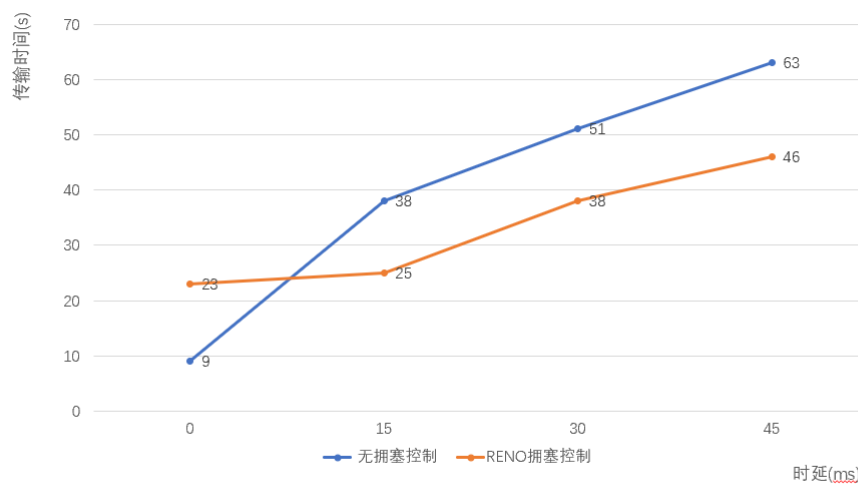
设置丢包率 0，改变时延，比较无拥塞控制和有拥塞控制的时延和吞吐率。

#### 1. 图形化分析



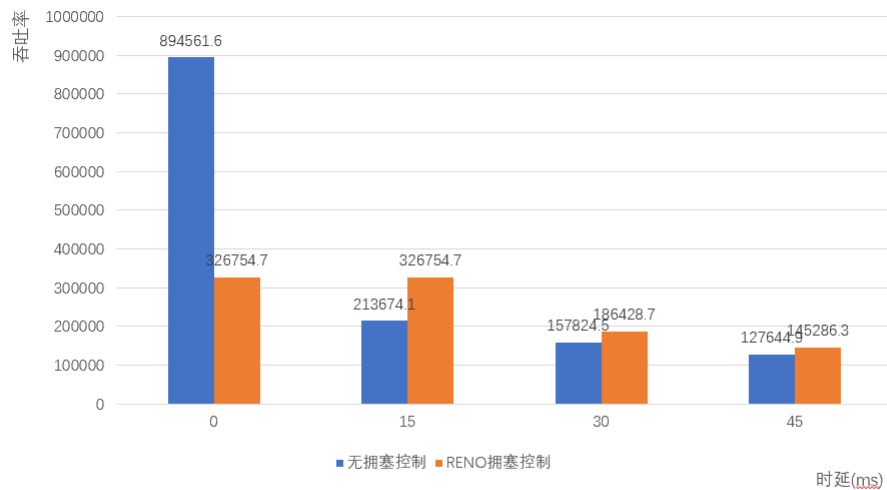
不同时延下传输时间对比柱状图

图 17: 丢包率为 0



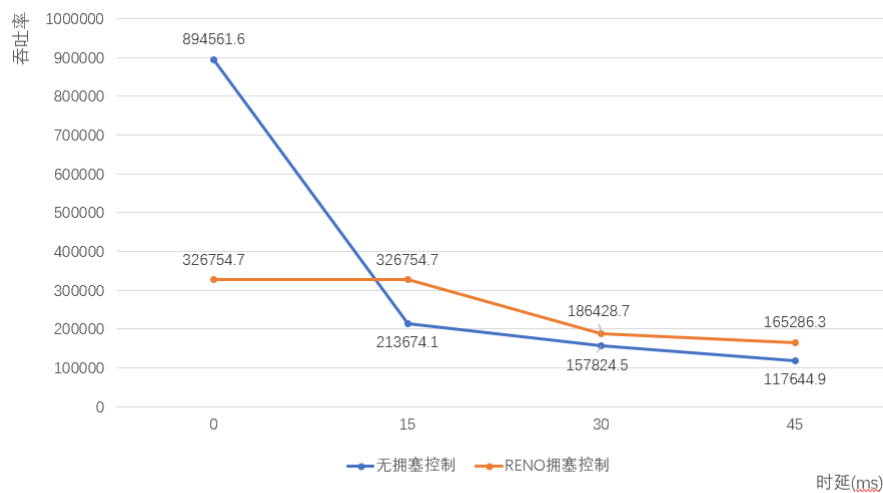
不同时延下传输时间对比折线图

图 18: 丢包率为 0



不同时延下吞吐量对比柱状图

图 19: 丢包率为 0



不同时延下吞吐量对比折线图

图 20: 丢包率为 0

## 2. 结果分析

### • 有拥塞控制的性能表现：

在低时延下，拥塞控制机制的影响较小，可以维持接近最大吞吐量，但同时能保证网络的稳定性。传输时长比较短，数据传输较为流畅。吞吐量接近带宽的最大值，表现较好。

### • 无拥塞控制的性能表现：

在低时延的情况下，网络较为稳定，数据传输较为顺畅。由于没有拥塞控制的限制，吞吐量接近最大带宽，但无法保证长期稳定性。传输时长较长，吐率表现较好，但缺乏

对突发网络变化的适应性。

高时延无拥塞控制时，传输时长显著增加。

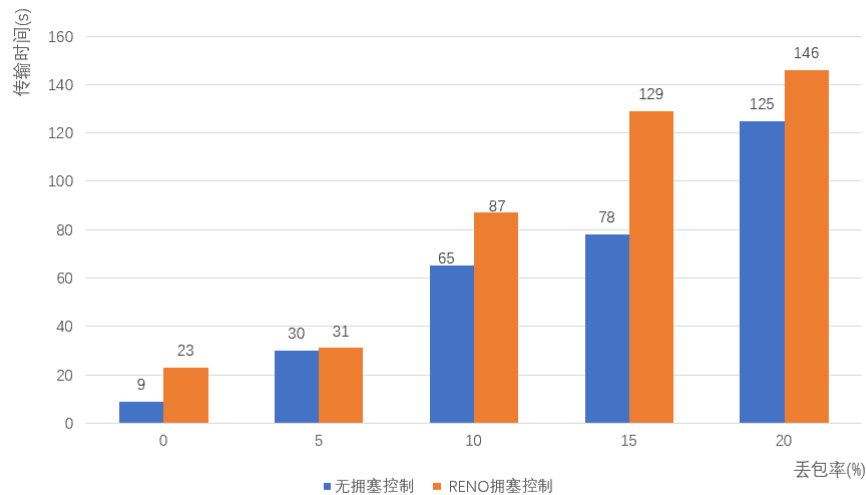
- 总结：

有拥塞控制性能均明显优于滑动窗口。有拥塞控制可以在下保持稳定，无拥塞控制在高时延下性能大幅下降，吞吐率和传输时长受丢包影响较大，但在低时延情况下，吞吐率较高。

## (二) 单一变量：丢包率

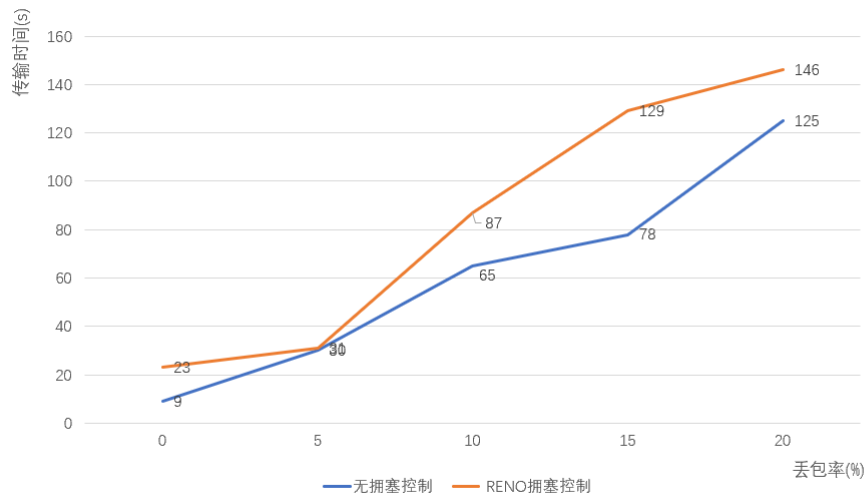
设置时延 0，改变丢包率，比较无拥塞控制和有拥塞控制的时延和吞吐率。

### 1. 图形化分析



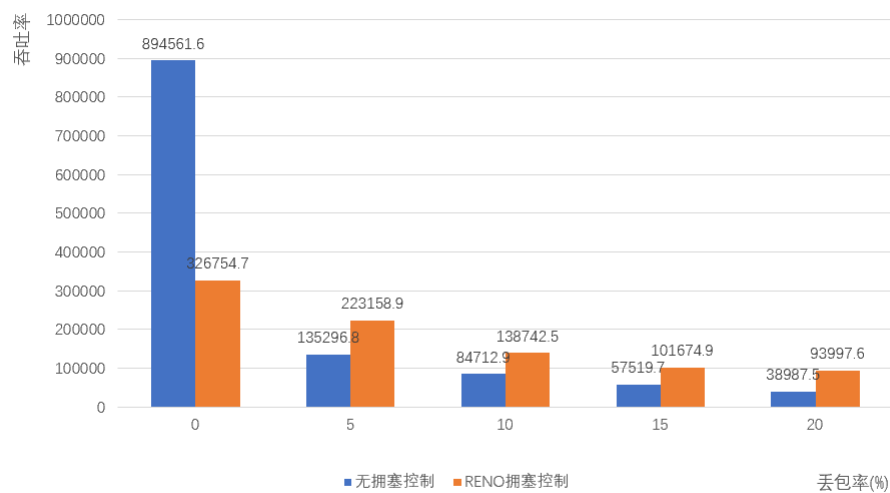
不同丢包率下传输时间对比柱状图

图 21: 时延为 0



不同丢包率下传输时间对比折线图

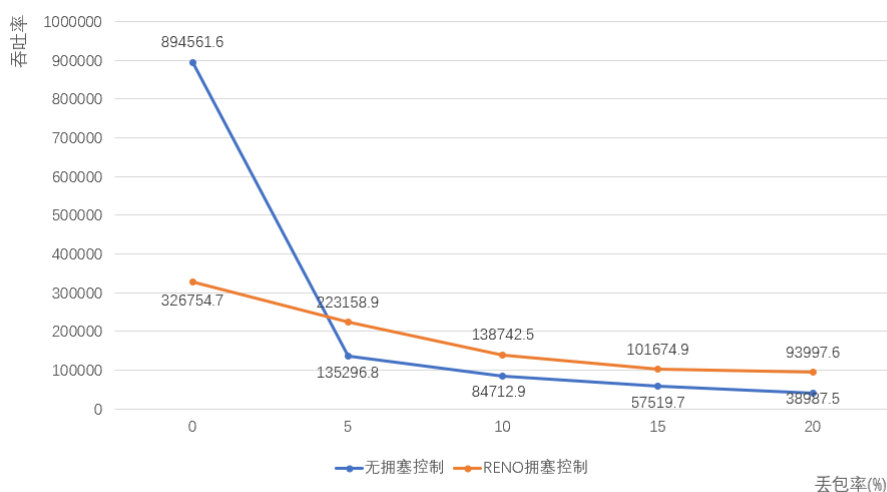
图 22: 时延为 0



不同丢包率下吞吐量对比柱状图

图 23: 时延为 0





不同丢包率下吞吐率对比折线图

图 24: 时延为 0

## 2. 结果分析

- 短期或低丢包环境下,无拥塞控制可以取得更高的瞬时吞吐率和更短的传输时间;Reno 虽不及其,但依然能保证较高效率。
- 高丢包场景,无拥塞控制极易导致大量重传,实际可用吞吐率往往锐减;Reno 可以自行调节,提供较为稳定的性能。Reno 则会在慢启动后逐步探测带宽,尤其在检测到潜在丢包时就会减小拥塞窗口,所以其最终吞吐率略低,传输耗时相对更长。
- 有拥塞控制的传输时间更可控,长时间运行更平稳。

## 五、 总结

- 滑动窗口机制相较于停等机制,能显著提高吞吐率,特别是在网络延迟和丢包较高的环境下;
- 滑动窗口大小对性能有较大影响,过大或过小的窗口都可能导致性能下降,最佳窗口大小应根据具体网络情况进行调整;
- 拥塞控制对于提高网络稳定性和吞吐率至关重要,能够有效避免网络拥塞,减少丢包,提高传输效率。