



南开大学
Nankai University

南 开 大 学

计 算 机 学 院

计算机网络实验报告

Lab 3.4 基于 UDP 服务设计可靠传输协议并编程实现

徐俊智

年级：2022 级

专业：计算机科学与技术

指导教师：吴英

2024 年 12 月 15 日

目录

| | |
|----------------------------------|---|
| 一、 实验目的 | 1 |
| 二、 实验要求 | 1 |
| 三、 实验内容 | 1 |
| 1. 停等机制与滑动窗口机制性能对比 | 1 |
| 3.1.1 时延相同, 丢包率不同 | 1 |
| 3.1.2 丢包率相同, 时延不同 | 2 |
| 3.1.3 分析结果 | 3 |
| 2. 滑动窗口机制中不同窗口大小对性能的影响 | 3 |
| 3.2.1 时延相同, 丢包率不同 | 3 |
| 3.2.2 丢包率相同, 时延不同 | 3 |
| 3.2.3 分析结果 | 4 |
| 3. 有拥塞控制和无拥塞控制的性能比较 | 5 |
| 3.3.1 时延相同, 丢包率不同 | 5 |
| 3.3.2 丢包率相同, 时延不同 | 5 |
| 3.3.3 分析结果 | 6 |

一、 实验目的

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: (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。

二、 实验要求

1. 控制变量法: 对比时要控制单一变量 (算法、窗口大小、延时、丢包率)
2. Router: 可能会有较大延时, 传输速率不作为评分依据, 也可自行设计
3. 延时、丢包率对比设置: 要有梯度 (例如 30ms, 50ms, ...; 5)
4. 测试文件: 必须使用助教发的测试文件 (1.jpg、2.jpg、3.jpg、helloworld.txt)
5. 性能测试指标: 时延、吞吐率, 要给出图、表并进行分析

三、 实验内容

1. 停等机制与滑动窗口机制性能对比

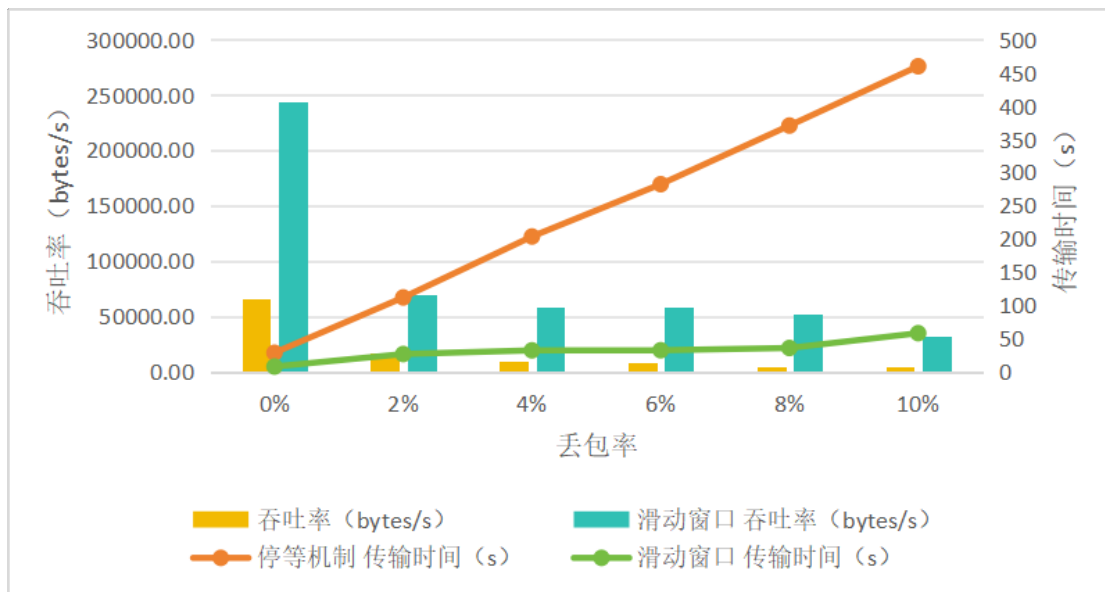
固定滑动窗口大小为 5, 进行测试。

3.1.1 时延相同, 丢包率不同

延时设为 0, 结果如下所示:

表 1: 丢包率对传输时间和吞吐率的影响

| 丢包率 | | 0% | 2% | 4% | 6% | 8% | 10% |
|------|---------------|----------|----------|---------|---------|---------|---------|
| 停等机制 | 传输时间 (s) | 28.37 | 111.65 | 203.43 | 282.21 | 370.49 | 459.81 |
| | 吞吐率 (bytes/s) | 65468.90 | 16635.49 | 9130.18 | 8457.25 | 5013.23 | 4039.42 |
| 滑动窗口 | 传输时间 (s) | 7.63 | 26.44 | 31.98 | 32.15 | 35.58 | 57.87 |
| | 吞吐率 (bytes/s) | 243491 | 70261.1 | 58082.2 | 57769.7 | 52196.3 | 32096.9 |

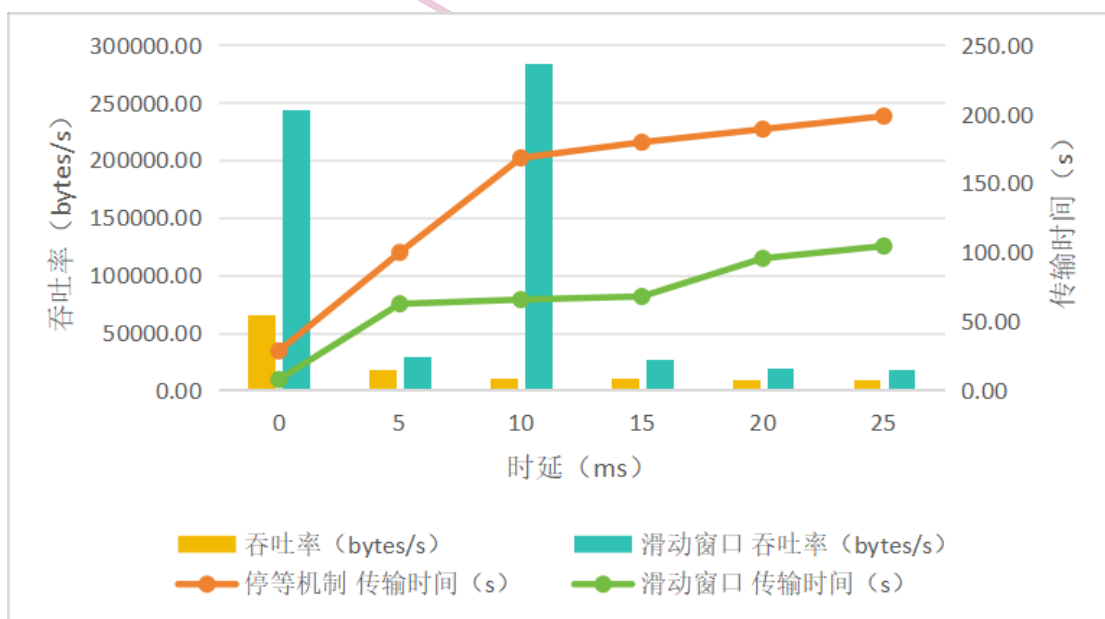


3.1.2 丢包率相同，时延不同

丢包率设为 0，结果如下所示：

表 2: 时延对传输时间和吞吐率的影响

| 时延 (ms) | | 0 | 5 | 10 | 15 | 20 | 25 |
|---------|---------------|----------|-----------|----------|----------|---------|--------|
| 停等机制 | 传输时间 (s) | 28.37 | 99.70 | 168.16 | 179.48 | 188.93 | 198.37 |
| | 吞吐量 (bytes/s) | 65468.90 | 18,628.53 | 11044.80 | 10348.27 | 9830.83 | 9362.7 |
| 滑动窗口 | 传输时间 (s) | 7.63 | 62.45 | 65.51 | 67.79 | 95.419 | 104.32 |
| | 吞吐量 (bytes/s) | 243491 | 29743.3 | 283550.5 | 27399 | 19465.2 | 17804 |



3.1.3 分析结果

停等机制中，每次发送一个数据包后必须等待确认，如果发生丢包，整个传输过程会被阻塞，导致传输时间增加；随着丢包率的增加，需要等待确认的次数增多，吞吐率大大降低。

滑动窗口机制允许发送方一次发送多个数据包，大大优化了停等机制中每发一个数据包都要等一个 ACK 的情况，但如果出现丢包，需要重传窗口内的所有数据包，因此丢包率对性能的影响较大；随着丢包率的增加，重传次数增多，导致传输时间增加，并降低吞吐率。

总体来说，滑动窗口机制比停等机制的效率更高，在不同网络条件下表现更为稳定和鲁棒。

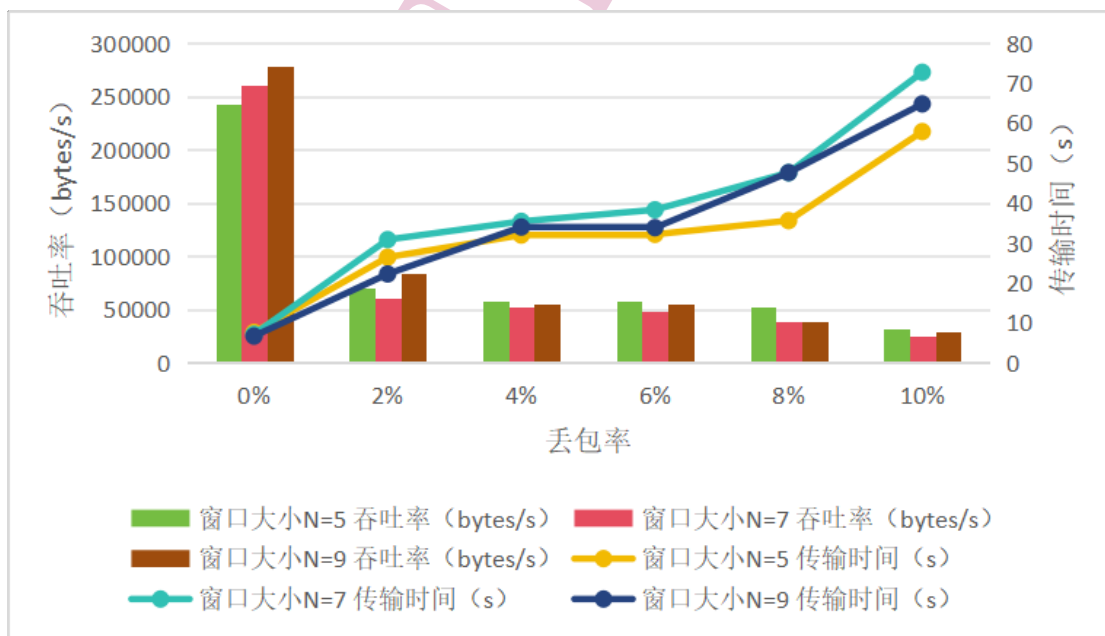
2. 滑动窗口机制中不同窗口大小对性能的影响

3.2.1 时延相同，丢包率不同

延时设为 0，结果如下所示：

表 3: 丢包率对不同窗口大小下传输时间和吞吐率的影响

| 丢包率 | | 0% | 2% | 4% | 6% | 8% | 10% |
|----------|---------------|--------|---------|---------|---------|---------|---------|
| 窗口大小 N=5 | 传输时间 (s) | 7.63 | 26.44 | 31.98 | 32.15 | 35.58 | 57.87 |
| | 吞吐率 (bytes/s) | 243491 | 70261.1 | 58082.2 | 57769.7 | 52196.3 | 32096.9 |
| 窗口大小 N=7 | 传输时间 (s) | 6.68 | 30.84 | 35.4 | 38.3 | 47.6 | 72.72 |
| | 吞吐率 (bytes/s) | 277922 | 60223.8 | 52461.2 | 48485.1 | 39015.4 | 25539.6 |
| 窗口大小 N=9 | 传输时间 (s) | 7.15 | 22.24 | 33.98 | 33.82 | 47.6 | 64.8 |
| | 吞吐率 (bytes/s) | 260706 | 83478 | 54651.3 | 54909.8 | 39015.4 | 28658.5 |

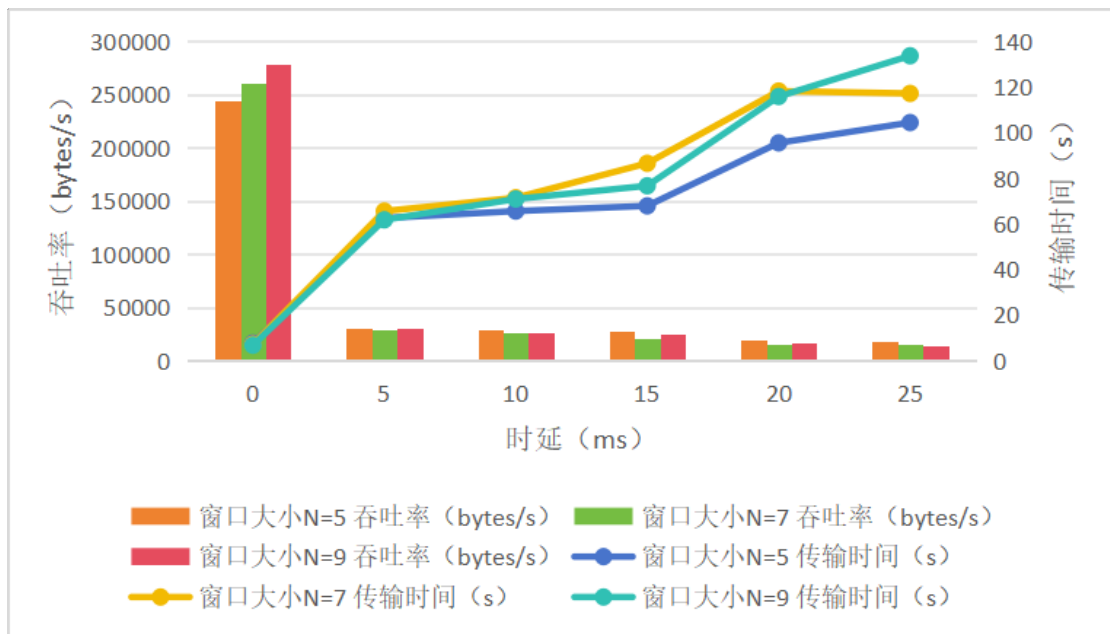


3.2.2 丢包率相同，时延不同

丢包率设为 0，结果如下所示：

表 4: 时延对不同窗口大小下传输时间和吞吐率的影响

| 时延 (ms) | | 0 | 5 | 10 | 15 | 20 | 25 |
|----------|---------------|--------|----------|----------|----------|----------|----------|
| 窗口大小 N=5 | 传输时间 (s) | 7.63 | 62.45 | 65.51 | 67.79 | 95.419 | 104.32 |
| | 吞吐率 (bytes/s) | 243491 | 29743.3 | 283550.5 | 27399 | 19465.2 | 17804 |
| 窗口大小 N=7 | 传输时间 (s) | 6.68 | 65.51 | 71.35 | 86.46 | 118.13 | 117.09 |
| | 吞吐率 (bytes/s) | 277922 | 283550.5 | 26028.99 | 21481.29 | 15721.93 | 15861.43 |
| 窗口大小 N=9 | 传输时间 (s) | 7.15 | 61.68 | 70.8 | 76.6 | 115.74 | 133.6 |
| | 吞吐率 (bytes/s) | 260706 | 30111.9 | 26230.6 | 24242.55 | 16048.45 | 13896.1 |



3.2.3 分析结果

总体来说,不同窗口大小在不同网络环境下的效率变化总体上趋于一致。较大的窗口在低延时和低丢包率条件下表现较好,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,此时丢包率较小,能一次发出更多的包,而没有太多的丢包导致的重传,所以能起到较好的加速效果,即减少等待的周期数,更好的应对时延问题。

但当丢包率较高时由于较大的窗口会增加重传代价,使用更大的滑动窗口相对于较小的滑动窗口性能优势明显减弱。当有较多丢包时,滑动窗口需要累积确认ACK、重传失序未确认的数据包,当这些失序的数据包ACK收到后才能移动窗口,因此有较多的性能损失。尤其是当路由延时设置较大时,性能损失更明显。

较小的窗口可能会导致较高的传输时间,但在某些情况下能够提高吞吐率。但是较小的窗口在高延时条件下可能更为稳定,减少了等待确认的时间。

选择合适的窗口大小需要综合考虑网络条件,以平衡传输时间和吞吐率的需求。

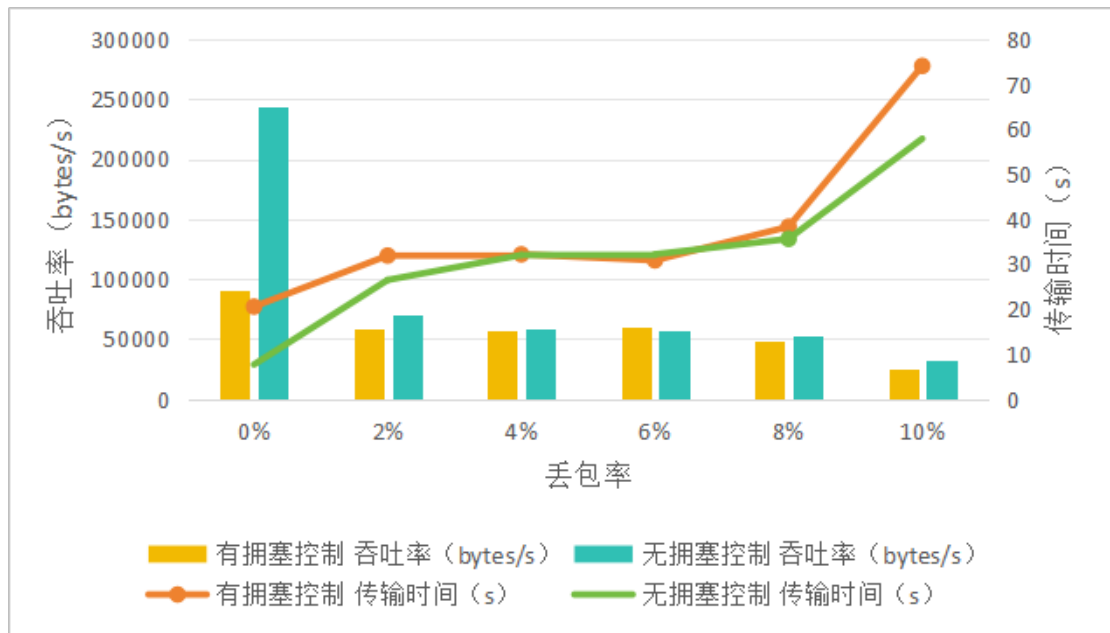
3. 有拥塞控制和无拥塞控制的性能比较

3.3.1 时延相同，丢包率不同

延时设为 0，结果如下所示：

表 5: 丢包率对有无拥塞控制下传输时间和吞吐率的影响

| 丢包率 | | 0% | 2% | 4% | 6% | 8% | 10% |
|-------|---------------|---------|---------|---------|---------|---------|---------|
| 有拥塞控制 | 传输时间 (s) | 20.51 | 31.86 | 32.14 | 30.8 | 38.3 | 74.05 |
| | 吞吐率 (bytes/s) | 90531.9 | 58323 | 57787.7 | 60288 | 48484.7 | 25079.7 |
| 无拥塞控制 | 传输时间 (s) | 7.63 | 26.44 | 31.98 | 32.15 | 35.58 | 57.87 |
| | 吞吐率 (bytes/s) | 243491 | 70261.1 | 58082.2 | 57769.7 | 52196.3 | 32096.9 |

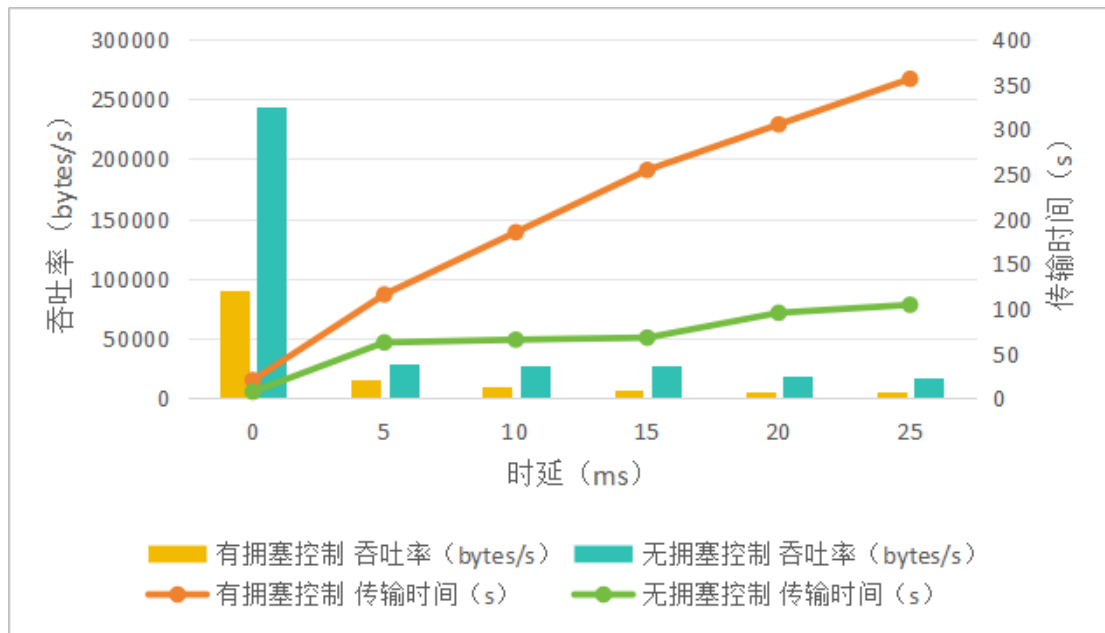


3.3.2 丢包率相同，时延不同

丢包率设为 0，结果如下所示：

表 6: 时延对有无拥塞控制下传输时间和吞吐率的影响

| 时延 (ms) | | 0 | 5 | 10 | 15 | 20 | 25 |
|---------|---------------|---------|----------|----------|---------|---------|---------|
| 有拥塞控制 | 传输时间 (s) | 20.51 | 115.819 | 185.075 | 254.331 | 305.066 | 355.801 |
| | 吞吐率 (bytes/s) | 90531.9 | 16036.68 | 10035.67 | 7302.89 | 6088.36 | 5220.2 |
| 无拥塞控制 | 传输时间 (s) | 7.63 | 62.45 | 65.51 | 67.79 | 95.419 | 104.32 |
| | 吞吐率 (bytes/s) | 243491 | 29743.3 | 28355.05 | 27399 | 19465.2 | 17804 |



3.3.3 分析结果

在低延时和低丢包率条件下，渡过慢启动阶段的有拥塞控制和无拥塞控制的效率差不多，拥塞控制的窗口可以不断增大，快重传机制只需要重传丢失的包，并不需要像超时重传那样重传窗口内所以已发送未确认的包，进一步提升了效率。

当延时丢包率逐渐增加时，拥塞控制机制表现不如无拥塞控制，频繁的丢包导致窗口不断缩小，可能会小于无拥塞控制的窗口大小，导致性能急速下降。