

# 计算机网络实验3-4报告

学号：2213034

姓名：辛杰

专业：物联网工程

## 计算机网络实验3-4报告

- 一、实验目的
- 二、实验内容
  - 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
- 三、收获感悟

## 一、实验目的

基于给定的实验测试环境，通过改变延时和丢包率，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、实验内容

说明：传输文件为helloworld.txt，大小为1,655,808 字节，因为传输的字节固定，所以对吞吐率不作对比图

### 1、停等机制与滑动窗口机制性能对比

停等机制用3-1实验的程序，滑动窗口机制用3-2实验的程序，滑动窗口发送窗口为20，接收窗口为1

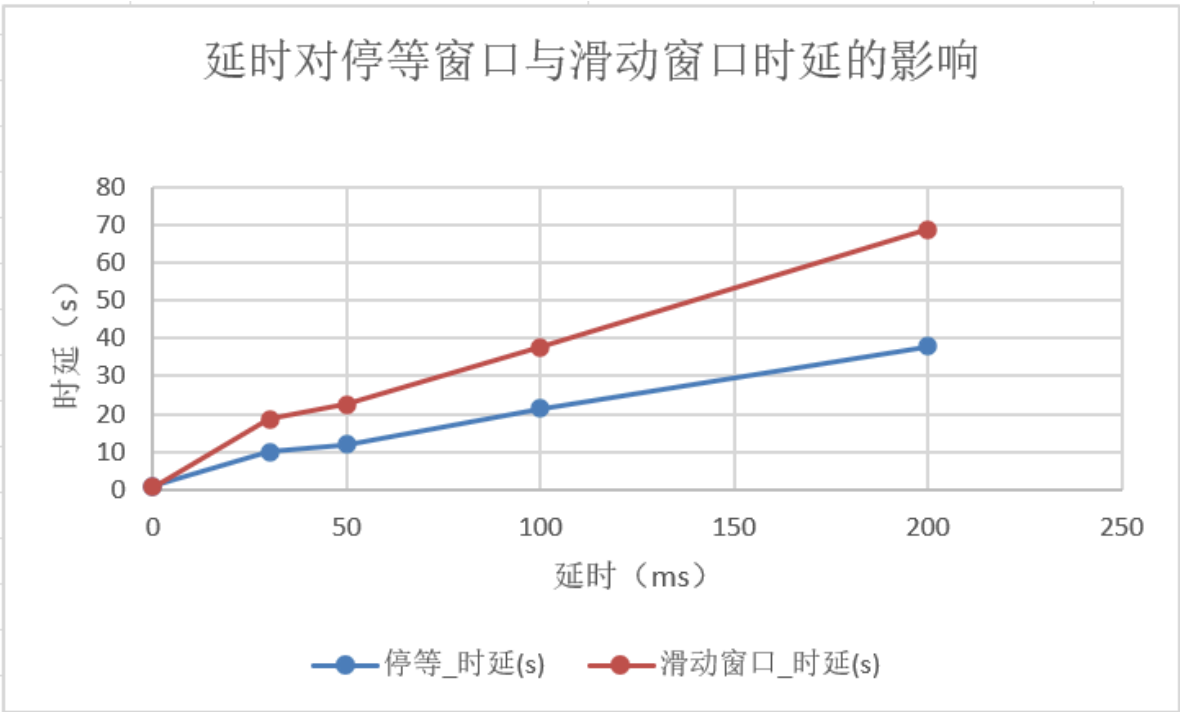
利用控制变量法，采用多组相同的延时和丢包率，对比停等机制与滑动窗口机制性能

```
C:\Users\辛杰\Desktop\1.exe x + v - □ ×
12:47:01 [send] Seq: 162 Ack: 0 Length: 10000ACK: 0 SYN: 0 FIN: 0 ST: 0 OV: 0
12:47:01 [recv] Seq: 0 Ack: 163 Length: 0 ACK: 1 SYN: 0 FIN: 0 ST: 0 OV: 0
接收报文的校验和: 0xa0e7
发送报文的校验和: 0x1ac
12:47:01 [send] Seq: 163 Ack: 0 Length: 10000ACK: 0 SYN: 0 FIN: 0 ST: 0 OV: 0
12:47:01 [recv] Seq: 0 Ack: 164 Length: 0 ACK: 1 SYN: 0 FIN: 0 ST: 0 OV: 0
接收报文的校验和: 0xa0e6
发送报文的校验和: 0x6b31
12:47:01 [send] Seq: 164 Ack: 0 Length: 10000ACK: 0 SYN: 0 FIN: 0 ST: 0 OV: 0
12:47:01 [recv] Seq: 0 Ack: 165 Length: 0 ACK: 1 SYN: 0 FIN: 0 ST: 0 OV: 0
接收报文的校验和: 0xa0e5
发送报文的校验和: 0x1aa
12:47:01 [send] Seq: 165 Ack: 0 Length: 10000ACK: 0 SYN: 0 FIN: 0 ST: 0 OV: 0
12:47:01 [recv] Seq: 0 Ack: 166 Length: 0 ACK: 1 SYN: 0 FIN: 0 ST: 0 OV: 0
接收报文的校验和: 0xa0e4
发送报文的校验和: 0x7c8c
12:47:01 [send] Seq: 166 Ack: 0 Length: 5808 ACK: 0 SYN: 0 FIN: 0 ST: 0 OV: 1
12:47:01 [recv] Seq: 0 Ack: 167 Length: 0 ACK: 1 SYN: 0 FIN: 0 ST: 0 OV: 0
接收报文的校验和: 0xa0e3
*****文件发送成功*****
文件传输时间: 0.874 s
吞吐率: 1.80675MB/s
*****开始关闭连接*****
发送报文的校验和: 0x7a37
12:47:01 [send] Seq: 6334 Ack: 0 Length: 0 ACK: 0 SYN: 0 FIN: 1 ST: 0 OV: 0
12:47:01 [recv] Seq: 18467Ack: 6335 Length: 0 ACK: 1 SYN: 0 FIN: 1 ST: 0 OV: 0
接收报文的校验和: 0x40b8
接收报文的校验和: 0x40b8
*****连接结束*****
请按任意键继续...
```

(这是停等机制在0丢包0延时的数据)

延时 (ms)	丢包率 (%)	停等_时延 (s)	滑动窗口_时延 (s)	停等_吞吐率 (MB/s)	滑动窗口_吞吐率 (MB/s)
0	0	0.874	0.689	1.80675	2.2918715
30	0	9.92	18.549	0.159184	0.0851314
50	0	12.008	22.482	0.131504	0.0702385
100	0	21.482	37.711	0.0735081	0.0418738
200	0	37.721	68.742	0.0418627	0.0229714
0	0.1	1.829	119.398	0.863369	0.0132255
0	1	3.985	104.062	0.396261	0.0151746
0	3	11.899	154.793	0.132709	0.0102014
0	5	19.872	156.952	0.0794636	0.010061
200	5	54.504	91.84	0.0289722	0.0171941

(这是总数据表)



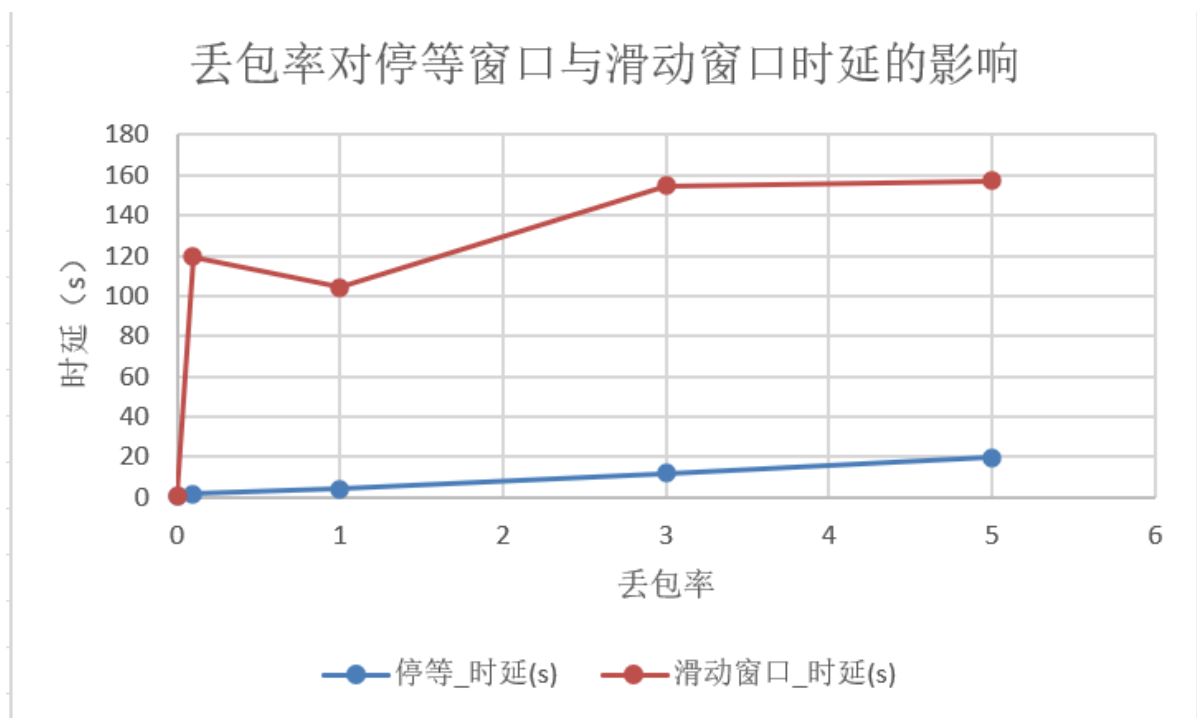
(这里丢包率设置为0)

分析:

随着延时 (ms) 的增加，**停等协议**和**滑动窗口协议**的总时延 (s) 均显著增加。

延时为0 ms时，滑动窗口协议的时延小于停等协议，显示其在低延时环境下的效率更高。

随着延时升高，两种协议的时延增长趋势不同，滑动窗口协议的时延增长幅度大于停等协议。



(这里延时设置为0)

分析：

**停等协议：**随着丢包率从0%增加到5%，停等协议的时延逐渐增加，呈非线性上升趋势，但总体增长较为平稳。

**滑动窗口协议：**滑动窗口协议的时延对丢包率非常敏感，时延显著增加，尤其在低丢包率（如0.1%）时时延激增，表现出明显的性能劣化。

结论：

**停等协议的适用性：**

- 停等协议虽然在低丢包率条件下效率较低，但其简单的设计使得在高丢包率环境下具有更好的稳定性。
- 适用于丢包率较高或网络质量波动较大的环境，例如无线网络或长距离传输链路。

**滑动窗口协议的适用性：**

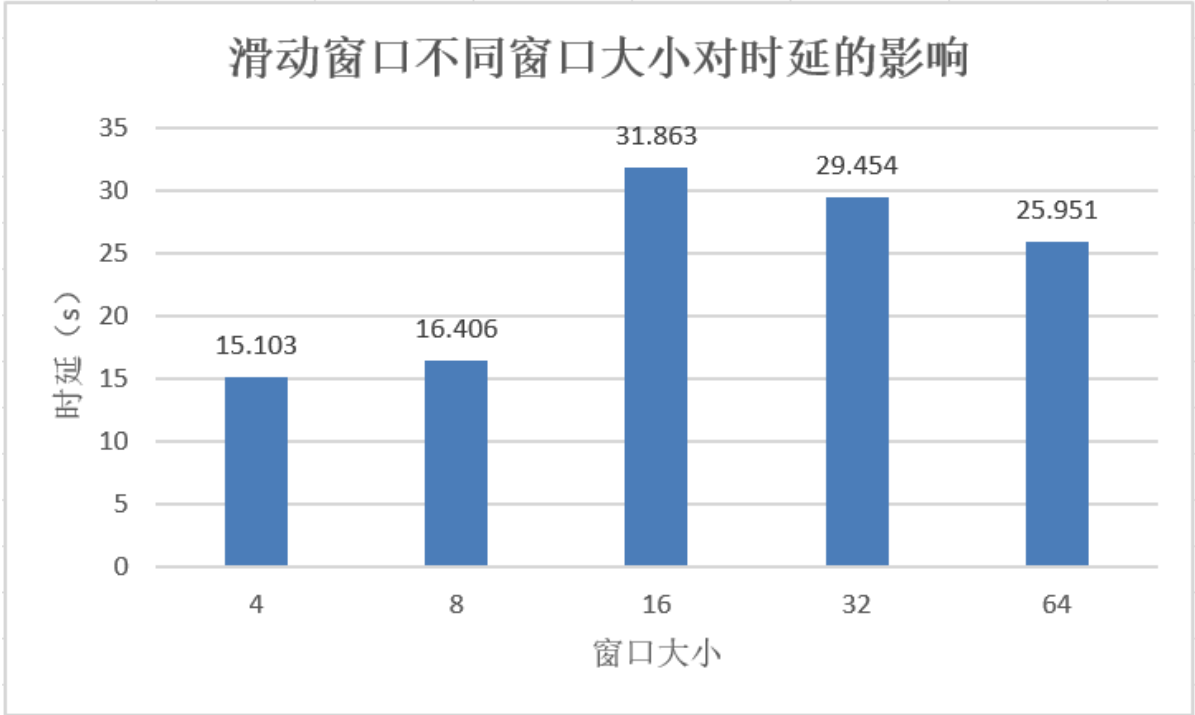
- 滑动窗口协议在低丢包率条件下能够充分利用网络带宽，具有较低时延，适用于高质量网络（丢包率接近0%）。
- 然而，当丢包率升高时，滑动窗口协议性能迅速下降，尤其在丢包率超过0.1%后表现较差，可能导致严重的延迟积累

## 2、滑动窗口机制中不同窗口大小对性能的影响

滑动窗口用3-2的实验，修改窗口大小，观察在延时为50ms、丢包率为0.1%和延时为0和丢包率为0.1%两种情况下的延时

窗口大小	延时 (ms)	丢包率 (%)	时延 (ms)	吞吐率 (Mbps)
4	50	0.1	15.103	0.104555
8	50	0.1	16.406	0.0962515
16	50	0.1	31.863	0.0495591
32	50	0.1	29.454	0.0536125
64	50	0.1	25.951	0.0608494
4	0	0	60.844	0.0259533
8	0	0	24.739	0.0638305
16	0	0	0.798	1.97882
32	0	0	0.731	2.16019
64	0	0	0.707	2.23352

(这是总数据表)

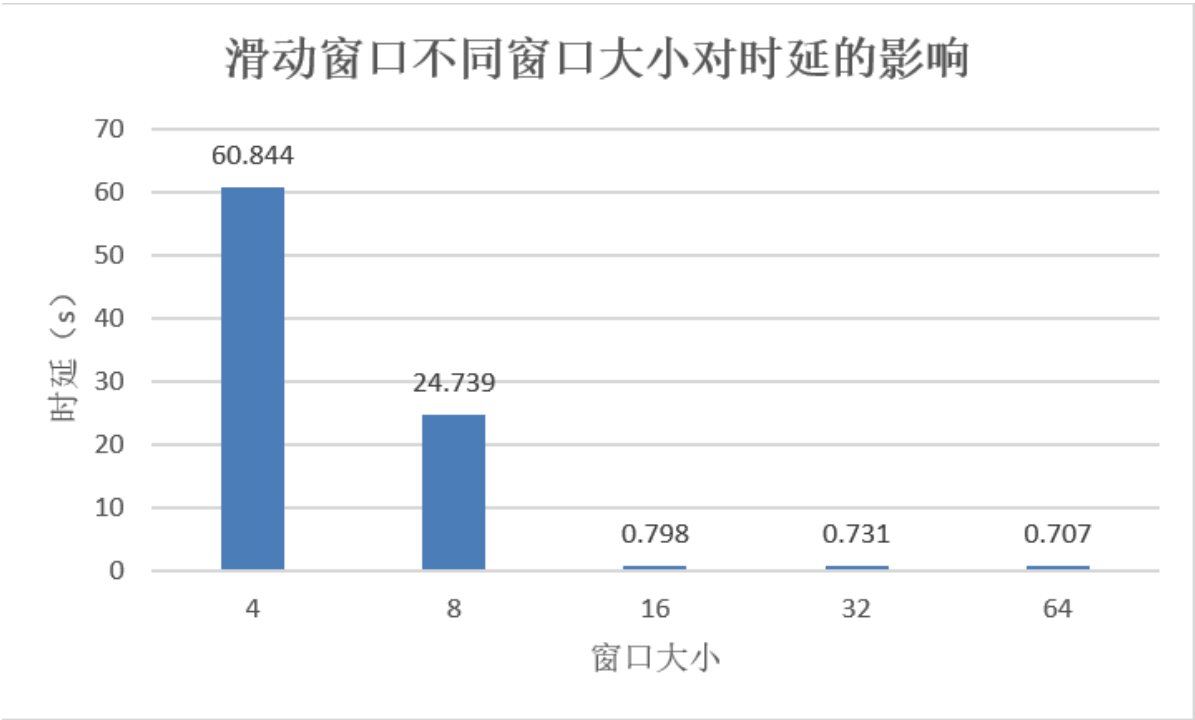


(丢包率为0.1%，延时为50ms)

分析：

随着窗口大小增加，时延总体呈现上升趋势，但并不线性增长。

小窗口（4和8）的时延较低，但窗口从16到64时，时延开始波动，32的时延较低，而64的时延下降。



(丢包率为0%，延时为0ms)

分析：

窗口大小为4时，时延较高（60.844ms）。

随着窗口大小的增加，时延显著下降，在窗口为16时降到不到1ms，并在32和64时进一步略微降低。

结论：

在无延迟和无丢包的理想网络中，窗口大小是影响时延和吞吐率的关键因素。

窗口大小为16时，时延和吞吐率已有明显优化，窗口大小为32或64时接近性能上限

延时和丢包率的存在显著影响网络性能，吞吐率与时延表现不再随窗口大小单调优化。

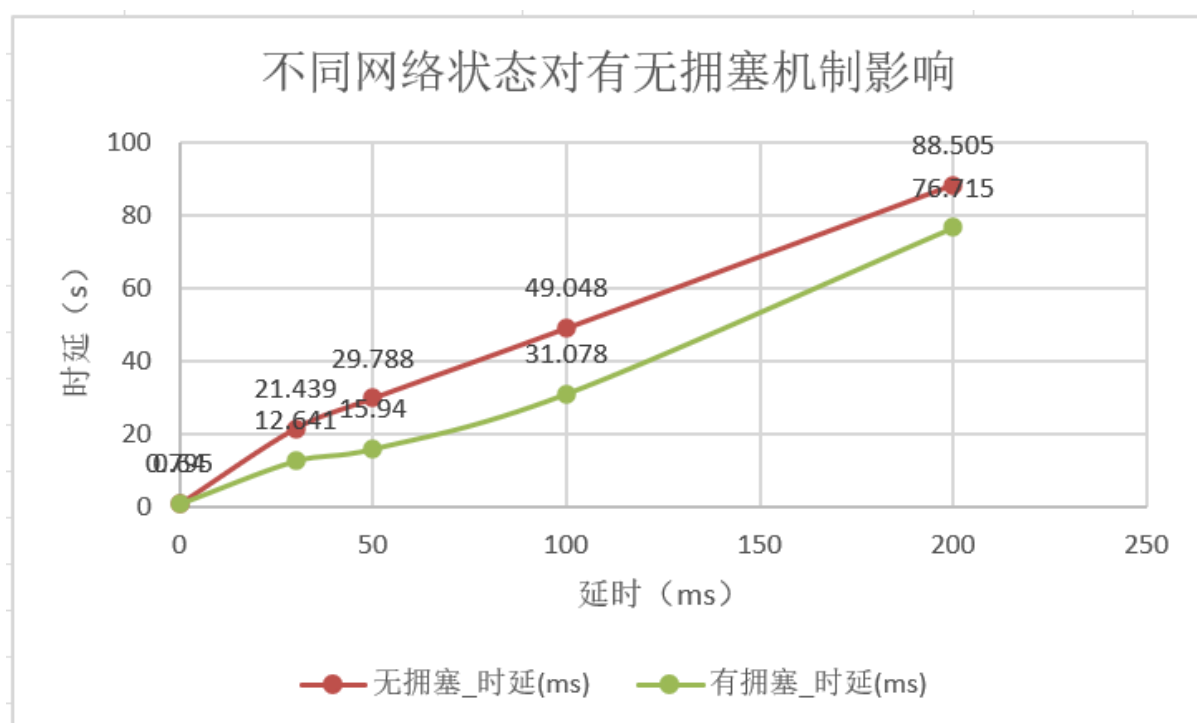
窗口大小过大（16及以上）反而可能因拥塞和重传导致性能下降，建议在此环境下选择较小窗口（如4或8）

### 3、有拥塞控制和无拥塞控制的性能比较

用3-2和3-3中的实验，在多组不同的延时和丢包率，比较有拥塞控制和无拥塞控制的时延

延时 (ms)	丢包率 (%)	无拥塞 时延 (ms)	有拥塞 时延 (ms)	无拥塞 吞吐率 (Mbps)	有拥塞 吞吐率 (Mbps)
0	0	0.64	0.795	2.46735	1.98629
30	0.1	21.439	12.641	0.0736556	0.124919
50	1	29.788	15.94	0.0530113	0.0990653
100	3	49.048	31.078	0.032195	0.0508109
200	5	88.505	76.715	0.0178419	0.020584

(这是总数据表)



(网络状态以时延为主，丢包率随时延增加也增加)

分析：

随着延时和丢包率增加，时延显著增长。延时从0增加到200ms时，无拥塞时延从0.64ms增至88.505ms，呈明显的线性增长趋势。

有拥塞条件下，时延相比无拥塞条件始终更低（如延时100ms、丢包率3%时，有拥塞时延31.078ms < 无拥塞时延49.048ms）。

在任何延时和丢包率条件下，无拥塞控制的时延始终高于有拥塞控制的时延

结论：

在**稳定、可靠的网络环境**（如局域网）中，可优先选择无拥塞控制以实现更高的吞吐率和更低的时延。

在**复杂、不稳定的网络环境**（如广域网、互联网）中，推荐采用拥塞控制机制，以提高网络的稳定性和抗拥塞能力。

## 三、收获感悟

这次实验不仅让我更好地理解网络协议与机制，还培养了我的实验设计和数据分析能力。通过对不同网络条件下的性能测试和比较，我认识到了网络性能优化中的多种因素，并学会了如何在实践中进行调整和优化。