

# 计算机网络实验报告

## Lab3-4 基于 UDP 服务设计可靠传输协议

### 网络空间安全学院 信息安全专业

2112492 刘修铭 1063

## 一、实验要求

实验要求主要包括三个方面：

1. **停等机制与滑动窗口机制性能对比**：通过改变延时和丢包率，完成停等机制和滑动窗口机制性能的对比实验。
2. **滑动窗口机制中不同窗口大小对性能的影响**：分别测试滑动窗口机制中不同窗口大小对性能的影响，包括累计确认和选择确认两种情形。
3. **滑动窗口机制中相同窗口大小情况下，累计确认和选择确认的性能比较**：在相同窗口大小的情况下，比较累计确认和选择确认的性能表现。

## 二、实验环境

本次实验基于前面三次实验完成，并在其基础上进行细微改动以更好适应本次实验：

- 修改日志输出，仅输出每次传输的**吞吐率及时延**，避免过多的 I/O 操作影响传输性能。

本次实验基于前面三次实验而实现，为避免实验报告冗长，三种协议设计、实现部分未在此次报告中给出。各个协议的具体实现请见附件：

- lab3-1.pdf
- lab3-2.pdf
- lab3-3.pdf

本次实验在本地 Windows 11 系统完成，使用 g++ 进行编译。

## 三、实验内容

本次实验主要使用**控制变量法**完成性能对比测试：

- 停等机制与滑动窗口机制性能对比：丢包率及时延相同的情况下，测试两种传输机制（需分别使用两种确认机制进行测试）的吞吐率及时延；
- 滑动窗口机制中不同窗口大小对性能的影响（累计确认和选择确认两种情形）：丢包率及时延相同的情况下，通过改变窗口大小，分别测试两种确认机制的吞吐率及时延；

- 滑动窗口机制中相同窗口大小情况下，累计确认和选择确认的性能比较：窗口大小相同的情况下，分别改变吞吐率或时延，测试并比较两种确认机制下传输文件的吞吐率及时延。

本实验报告中的数据均为多次测试的**平均值**，能够避免一些偶然因素导致的数据误差。

## 四、实验结果及分析

实验中丢包率的单位为 %，吞吐率的单位为 *Byte/ms*，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，分为以下几个步骤：

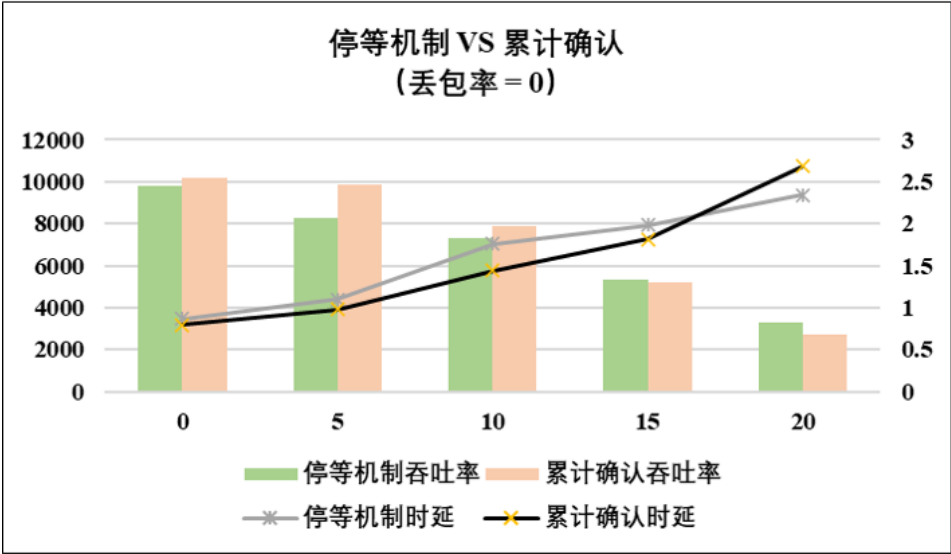
### （一）停等机制与滑动窗口机制性能对比

- 设置丢包率为 0，改变时延，比较停等机制和滑动窗口机制（累计确认）的吞吐率和时延。

◦ **实验结果：**

时延	停等机制 吞吐率	累计确认 吞吐率	停等机制 时延	累计确认 时延
0	9768.7	10210.5	0.863	0.795
5	8265.2	9863.4	1.1	0.982
10	7321.1	7893.5	1.756	1.437
15	5326.1	5185	1.986	1.813
20	3264.9	2689.3	2.342	2.6813

◦ **图形化分析：**



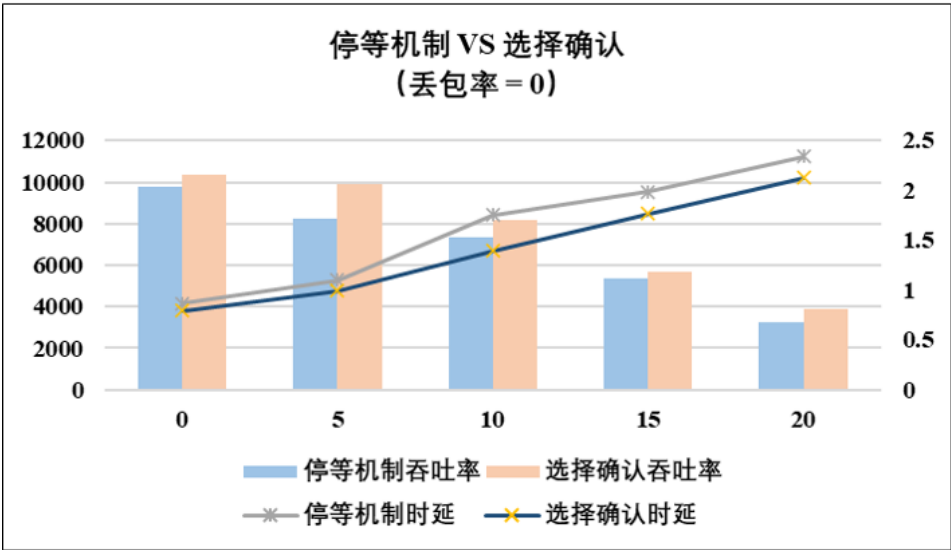
◦ **分析：**

- 本次实验设定的超时重传时间阈值为 10 *ms*。当设定的时延小于该阈值时，使用累计确认机制的基于滑动窗口的时延低于停等机制，吞吐率高于停等机制；但当设定的时延超过该阈值时，累计确认机制的性能将会降低，各项指标均低于停等机制。
  - 分析可知，如果时延过高，导致频繁发生超时现象。而累计确认机制中超时重传整个缓冲区的数据，故而性能大幅度降低；而低时延下，滑动窗口机制能够充分利用网络带宽，减少传输时间，使得各项指标均高于停等机制。
- 设置丢包率为 0，改变时延，比较停等机制和滑动窗口机制（选择确认）的吞吐率和时延。

○ 实验结果：

时延	停等机制 吞吐率	选择确认 吞吐率	停等机制 时延	选择确认 时延
0	9768.7	10356	0.863	0.794
5	8265.2	9934.5	1.1	0.994
10	7321.1	8165.2	1.756	1.395
15	5326.1	5689.4	1.986	1.764
20	3264.9	3896.7	2.342	2.126

○ 图形化分析：



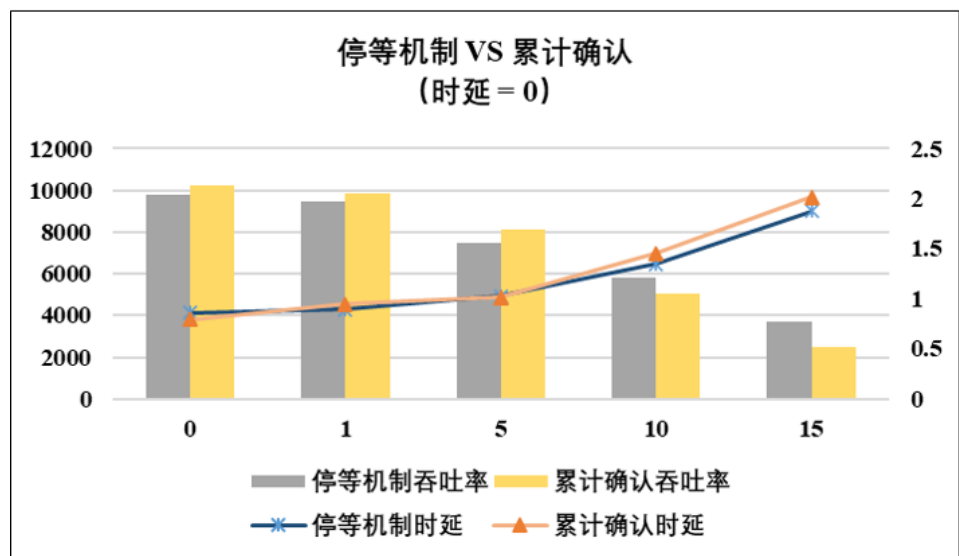
○ 分析：

- 可以看到，随着设定的时延增高，两种机制的性能均有所下降，但选择确认机制性能要优于停等机制。
  - 分析可知，当时延较低时，滑动窗口能够充分利用网络带宽，减少传输时间，使得各项指标均高于停等机制；但是当时延较高时，选择确认机制仅仅重传超时数据包，且能够充分利用网络带宽，故而性能指标也高于停等机制。
- 设置时延为 0，改变丢包率，比较停等机制和滑动窗口机制（累计确认）的吞吐率和时延。

○ 实验结果：

丢包率	停等机制 吞吐率	累计确认 吞吐率	停等机制 时延	累计确认 时延
0	9768.7	10210.5	0.863	0.795
1	9456.2	9834.1	0.896	0.945
5	7456.1	8125.2	1.025	1.015
10	5789.5	5046.2	1.345	1.456
15	3697.4	2465.1	1.876	2.016

○ 图形化分析：



○ 分析:

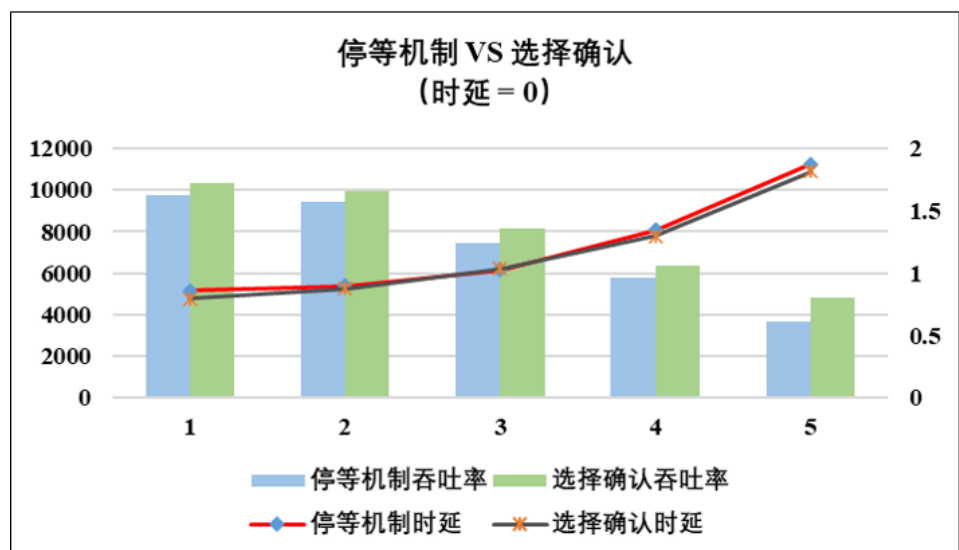
- 当丢包率较小时, 使用累计确认机制的基于滑动窗口的时延低于停等机制, 吞吐率高于停等机制; 但当丢包率较高时, 累计确认机制的性能将会降低, 各项指标均低于停等机制。
- 分析可知, 如果丢包率过高, 导致频繁发生重传现象, 累计确认机制将重传整个缓冲区的数据, 故而性能大幅度降低; 而低丢包率下, 滑动窗口机制能够充分利用网络带宽, 减少传输时间, 使得各项指标均高于停等机制。

- 设置时延为 0, 改变丢包率, 比较停等机制和滑动窗口机制 (选择确认) 的吞吐率和时延。

○ 实验结果:

丢包率	停等机制吞吐率	选择确认吞吐率	停等机制时延	选择确认时延
0	9768.7	10356	0.863	0.794
1	9456.2	9934.5	0.896	0.876
5	7456.1	8165.2	1.025	1.034
10	5789.5	6354.5	1.345	1.297
15	3697.4	4831	1.876	1.816

○ 图形化分析:



○ 分析:

- 可以看到，随着设定的丢包率增高，两种机制的性能均有所下降，但选择确认机制性能要优于停等机制。
- 分析可知，当丢包率较低时，滑动窗口能够充分利用网络带宽，减少传输时间，使得各项指标均高于停等机制；但是当丢包率较高时，选择确认机制仅仅重传丢失数据包，且能够充分利用网络带宽，故而性能指标也高于停等机制。

• **综合分析：**

- 随着丢包率的增加，停等机制、累计确认机制和选择确认机制的吞吐率均有所减小，传输时延均有所增加。
- 随着传输时延的增大，三种机制的吞吐率均有所减小，传输时延均有所增加。
- 两种确认机制下的基于滑动窗口的发送机制整体上性能要优于停等机制，具体表现为吞吐率更高，时延更小。但是当时延或丢包率较高时，累计确认机制的性能又会由于数据的大量重复传输而急剧下降。

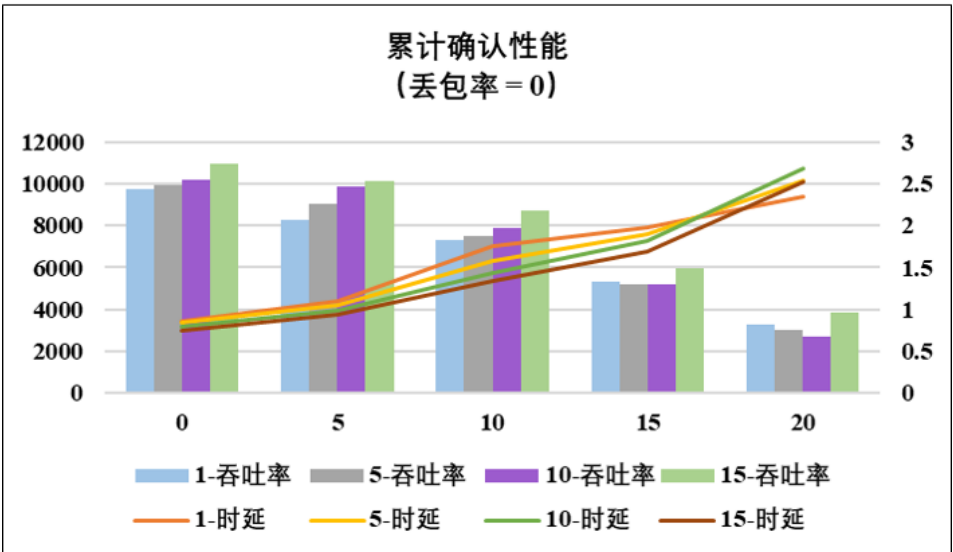
## (二) 滑动窗口机制中不同窗口大小对性能的影响

- 设置丢包率为 0，改变窗口大小及时延，比较滑动窗口机制（累计确认）的吞吐率和时延。

◦ **实验结果：**

窗口大小		时延				
		0	5	10	15	20
1	吞吐率	9768.7	8265.2	7321.1	5326.1	3264.9
	时延	0.863	1.1	1.756	1.986	2.342
5	吞吐率	9925.3	9031.1	7532.4	5221.5	2986.8
	时延	0.842	1.045	1.582	1.896	2.548
10	吞吐率	10210.5	9863.4	7893.5	5185	2689.3
	时延	0.795	0.982	1.437	1.813	2.684
15	吞吐率	10986.4	10158.1	8698.3	5957.6	3836.4
	时延	0.746	0.934	1.344	1.697	2.519

◦ **图形化分析：**



◦ **分析：**

- 将其与前面数据比较可知，当窗口大小为 1 时，支持累计确认机制的基于窗口的文件传输机制将退化成停等机制；且随着窗口的增大，各项指标均有所升高（高吞吐率，低时延）。

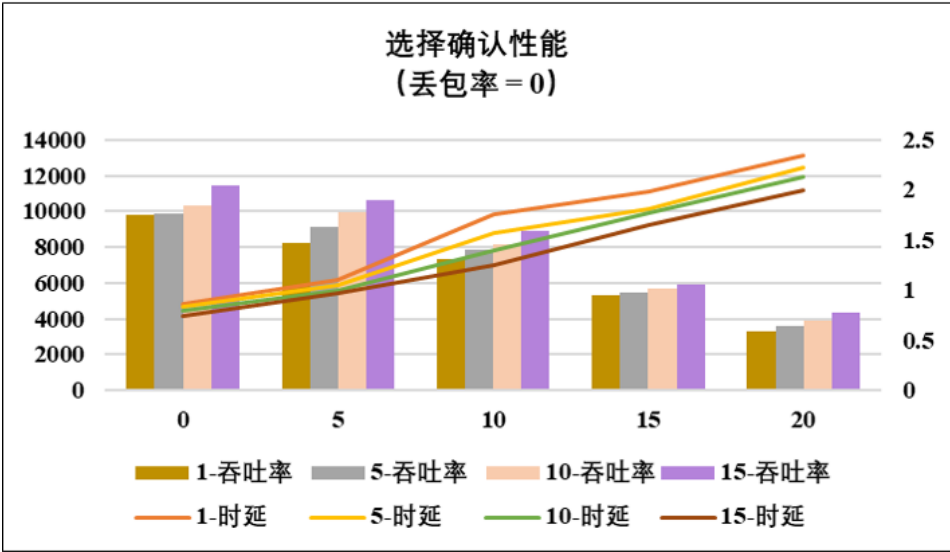
- 但是随着设定时延的增大，各种窗口大小的传输机制的性能均有所降低，且下降幅度较大。分析依旧由于其重传整个缓冲区导致。

• 设置丢包率为 0，改变窗口大小及时延，比较滑动窗口机制（选择确认）的吞吐率和时延。

◦ 实验结果：

窗口大小		时延				
		0	5	10	15	20
1	吞吐率	9768.7	8265.2	7321.1	5326.1	3264.9
	时延	0.863	1.1	1.756	1.986	2.342
5	吞吐率	9859.4	9163.4	7831.5	5497.7	3593.4
	时延	0.833	1.044	1.574	1.816	2.219
10	吞吐率	10356	9934.5	8165.2	5689.4	3896.7
	时延	0.794	0.994	1.395	1.765	2.126
15	吞吐率	11463.4	10654.1	8926.4	5934.1	4321.1
	时延	0.739	0.964	1.254	1.653	2.001

◦ 图形化分析：



◦ 分析：

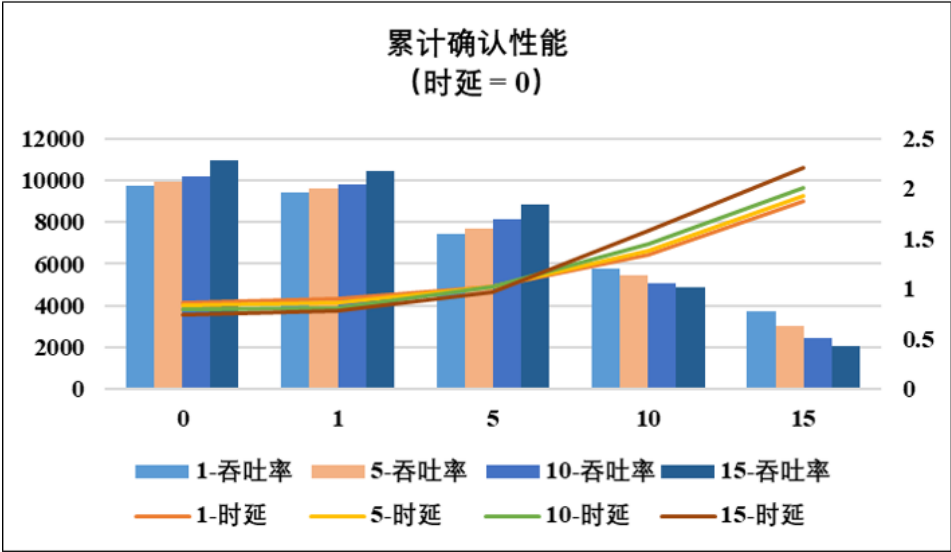
- 将其与前面数据比较可知，当窗口大小为 1 时，支持选择确认机制的基于窗口的文件传输机制将退化成停等机制；且随着窗口的增大，各项指标均有所升高（高吞吐率，低时延）。
- 但是随着设定时延的增大，各种窗口大小的传输机制的性能均有所降低，但下降幅度较累计确认相比更为均匀。这是由于选择确认仅重传为正确收到的数据包。

• 设置时延为 0，改变丢包率及窗口大小，比较滑动窗口机制（累计确认）的吞吐率和时延。

◦ 实验结果：

窗口大小		丢包率				
		0	1	5	10	15
1	吞吐率	9768.7	9456.2	7456.1	5789.5	3697.4
	时延	0.863	0.896	1.025	1.345	1.876
5	吞吐率	9925.3	9598.1	7682.1	5436.1	3024.5
	时延	0.842	0.867	1.023	1.379	1.926
10	吞吐率	10210.5	9834.1	8125.2	5046.2	2465.1
	时延	0.795	0.813	1.018	1.456	2.016
15	吞吐率	10986.4	10432.4	8846.4	4863.1	2035.6
	时延	0.746	0.787	0.968	1.589	2.215

图形化分析：



分析：

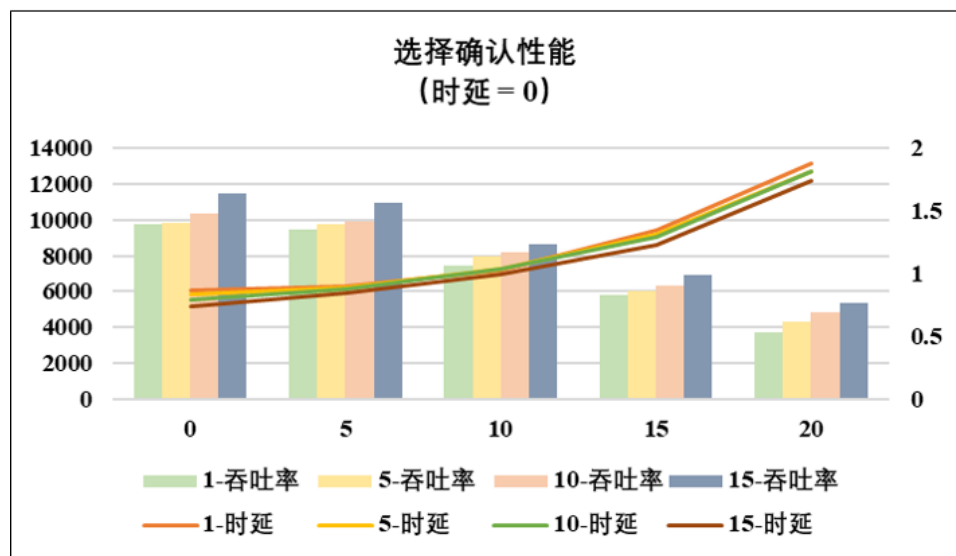
- 当丢包率较小时，传输性能随着窗口大小的增加而提升，具体表现为高吞吐率和低传输时延。而当丢包率较大时，其性能反而随着窗口大小的增加而递减，且窗口越大的性能越差。
  - 分析可知，当丢包率较高时，将频繁激发数据重传，而累计确认机制将重传整个缓冲区，故而丢包率高时性能会差；且如果窗口越大，则需要重传的数据包也就越多，性能也就越差。
- 设置时延为 0，改变丢包率及窗口大小，比较滑动窗口机制（选择确认）的吞吐率和时延。

实验结果：

窗口大小		丢包率				
		0	1	5	10	15
1	吞吐率	9768.7	9456.2	7456.1	5789.5	3697.4
	时延	0.863	0.896	1.025	1.345	1.876
5	吞吐率	9859.4	9724.6	7983.4	6024.1	4326.1
	时延	0.833	0.883	1.033	1.316	1.816
10	吞吐率	10356	9934.5	8165.2	6354.5	4831
	时延	0.794	0.876	1.034	1.297	1.816
15	吞吐率	11463.4	10987.4	8653.4	6946.8	5369.4
	时延	0.739	0.843	0.989	1.234	1.743

图形化分析：





- 分析：
  - 将其与前面数据比较可知，当窗口大小为 1 时，支持选择确认机制的基于窗口的文件传输机制将退化成停等机制；且随着窗口的增大，各项指标均有所升高（高吞吐率，低时延）。
  - 但是随着设定时延的增大，各种窗口大小的传输机制的性能均有所降低，但下降幅度较累计确认相比更为均匀，且未出现累计确认机制中窗口大小与性能成反比的现象。这是由于选择确认仅重传为正确收到的数据包。
- 综合分析：
  - 整体来看，在不同的丢包率与时延下，窗口越大，性能越高；但是当丢包率或时延较大时，累计确认机制会出现窗口大小与性能成反比的现象，即窗口大小越大，性能反而越低。分析可知，这是由于累计确认机制重发数据时，重新发送整个缓冲区中所有的数据包，使得出现大量数据包被重复发送的问题，而窗口越大，浪费的资源也就越多，重传开销越大，故而性能也就越低。

### （三）滑动窗口机制中相同窗口大小情况下，累计确认和选择确认机制的性能比较

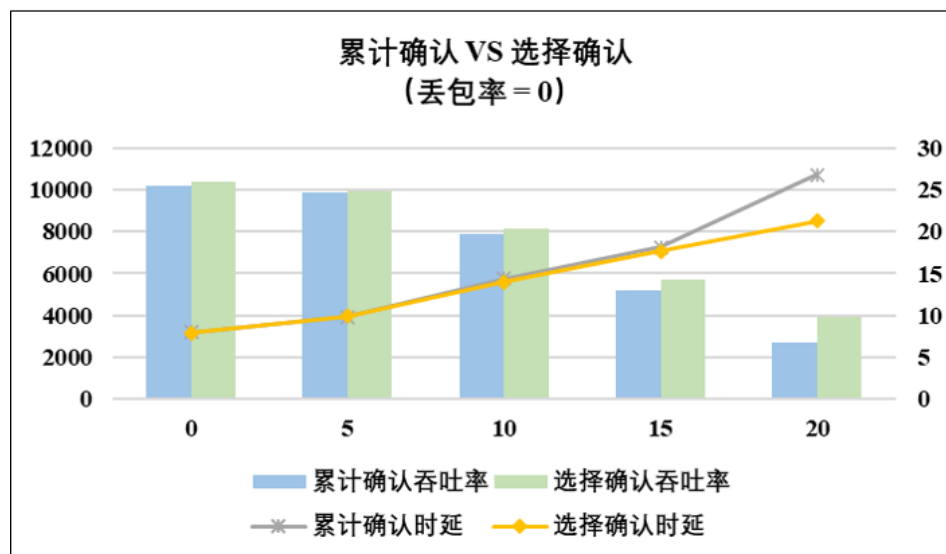
- 设置丢包率为 0，保持窗口大小相同，改变时延，比较滑动窗口机制的累计确认和选择确认的吞吐率和时延。

#### ○ 实验结果：

时延	累计确认 吞吐率	选择确认 吞吐率	累计确认 时延	选择确认 时延
0	10210.5	10356	7.95	7.94
5	9863.4	9934.5	9.82	9.94
10	7893.5	8165.2	14.37	13.95
15	5185	5689.4	18.13	17.64
20	2689.3	3896.7	26.84	21.26

#### ○ 图形化分析：





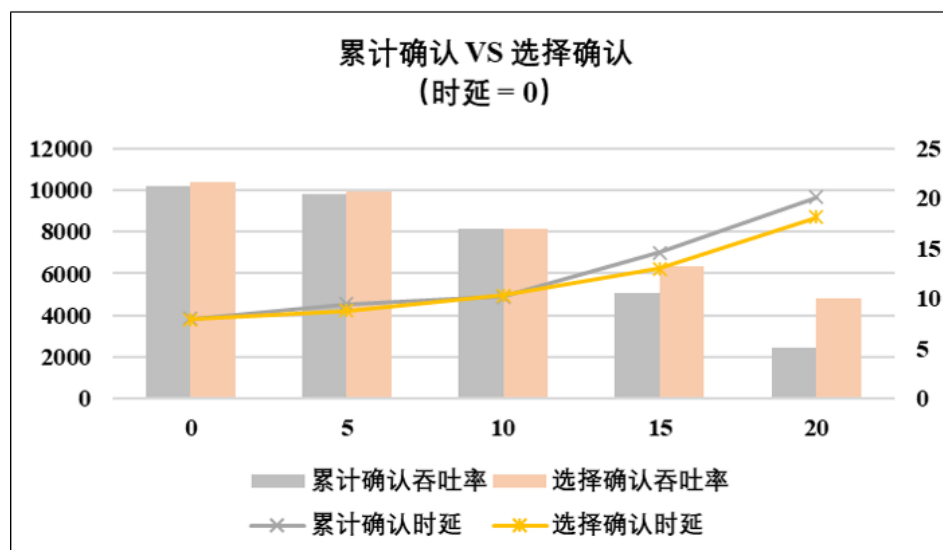
○ 分析:

- 在低时延下, 累计确认与选择确认的性能接近; 但是当设定的时延较高时, 选择确认机制的性能要明显高于累计确认机制。
  - 分析可知, 高时延下, 累计确认机制将频繁重发缓冲区中全部数据包, 使得大量数据包被重复发送, 浪费资源, 降低性能。
- 设置时延为 0, 保持窗口大小相同, 改变丢包率, 比较滑动窗口机制的累计确认和选择确认的吞吐率和时延。

○ 实验结果:

丢包率	累计确认吞吐率	选择确认吞吐率	累计确认时延	选择确认时延
0	10210.5	10356	7.95	7.94
1	9834.1	9934.5	9.45	8.76
5	8125.2	8165.2	10.18	10.34
10	5046.2	6354.5	14.56	12.97
15	2465.1	4831	20.16	18.16

○ 图形化分析:



○ 分析:

- 在低丢包率下，可以看到累计确认与选择确认的性能接近；但是当设定的丢包率较高时，选择确认机制的性能要明显高于累计确认机制。
  - 分析可知，高丢包率下，累计确认机制将频繁重发缓冲区中全部数据包，使得大量数据包被重复发送，浪费资源，降低性能。
- 综合分析：
    - 当丢包率或时延较低时，可以看到两种确认机制的文件传输性能较为接近，二者均能较大程度利用网络带宽进行传输；但是当丢包率或时延较高时，由于累计确认机制重发整个缓冲区的全部内容，使得累计确认的性能急剧下降，显著低于选择确认。

## 五、实验总结

---

综合以上实验结果和分析，可以得出以下结论：

- 基于滑动窗口的文件传输机制整体上性能要优于停等机制；
- 当窗口过小时，基于滑动窗口的文件传输机制将退化成停等机制；
- 当丢包率或时延等较高时，支持累计确认机制的基于滑动窗口的文件传输机制性能将**显著降低**，且可能会低于停等机制；
- 当窗口较小时，无法激发快速重传机制，使得二者的性能较低，与停等机制相近；随着窗口的增大，两种确认机制的性能整体上会提高；但是当窗口增大到一定程度时，由于网络带宽、文件读取等因素的限制，其性能不会有太大的提升，最终会趋于动态平衡；
- 选择确认机制的性能整体上要优于累计确认机制。
- 本次实验也说明，使用何种机制要视网络的实际情况而决定，适合的才是最好的。

## 六、实验思考

---

- 本次实验仅仅基于有限情况进行比较说明，无法代表三种传输机制的真实情况，存在一定局限性。
- 本实验在个人电脑完成，实验数据仅对本实验报告负责，存在一定偶然性。
- 由于编程逻辑等问题，导致各种设计在实验中均会不同程度地产生离群值，以及违背理论的数据。
- 选择确认机制的性能谷点未能发现，需要再多加了解。

## 七、说明

---

一个学期转瞬即逝，感谢各位老师、助教学长学姐的悉心指导！自己也和网络编程有了初步的了解！

本学期实验所涉及源代码、可执行文件及实验报告等均已上传至[Github仓库](#)，且均已开源。