

# 计算机网络第三次作业第四部分

张昊星 2113419

## 实验要求

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（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。

## 实验设计

传输文件以为1.jpg为例。传输速率以吞吐率(byte/s)作为对比量。主要利用表格数据进行对比。

## 实验步骤

### （1）停等机制与滑动窗口机制性能对比

滑动窗口机制以累计确认方式（GBN）为例。

首先设置丢包率为0%，延迟分别为5、10、15、20、25，对比如下：

延迟	停等机制	滑动窗口机制
5	928676	928676
10	371471	371471
15	309559	309559
20	232169	265336
25	206373	232169

再设置延迟为0ms，丢包率分别为1、2、3、4、5，对比如下：

丢包率	停等机制	滑动窗口机制
1	309559	464338
2	265336	371471
3	206373	309559
4	168850	232169
5	142873	142873

## 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
- 丢包率越大时GBN效率与停等机制越接近并有可能效率低于停等机制
  - 更大的窗口意味着更高的重传代价，大大降低性能

## (2) 滑动窗口机制中不同窗口大小对性能的影响

### 1. 累计确认

首先设置丢包率为0%，延迟分别为5、10、15、20、25，对比如下：

延迟	窗口=6	窗口=10
5	464338	928676
10	371471	371471
15	309559	309559
20	265336	265336
25	232169	232169

再设置延迟为0ms，丢包率分别为1、2、3、4、5，对比如下：

丢包率	窗口=6	窗口=10
1	928676	464338
2	464338	371471
3	309559	309559
4	232169	232169
5	142873	142873

### 2. 选择确认

首先设置丢包率为0%，延迟分别为5、10、15、20、25，对比如下：

延迟	窗口=6	窗口=10
5	464338	464338
10	371471	371471
15	309559	309559
20	265336	265336
25	232169	232169

再设置延迟为0ms，丢包率分别为1、2、3、4、5，对比如下：

丢包率	窗口=6	窗口=10
1	619118	464338
2	464338	371471
3	371471	265336
4	309559	232169
5	232169	232169

## 分析

- 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
- 在延迟较低的时候**窗口大的效率更高**
  - 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，即减少等待的周期数，更好的应对时延问题。
- 对于丢包问题，当丢包率较低时由于大的窗口会**增加重传代价**，效率降低

## (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

窗口设置为6。

首先设置丢包率为0%，延迟分别为5、10、15、20、25，对比如下：

延迟	累计确认	选择确认
5	464338	464338
10	371471	371471
15	309559	309559
20	265336	265336
25	232169	232169

再设置延迟为0ms，丢包率分别为1、2、3、4、5，对比如下：

丢包率	累计确认	选择确认
1	928676	619118
2	464338	464338
3	309559	371471
4	232169	309559
5	142873	232169

## 分析

- 不同确认方式的效率变化总体上趋于一致
- 延迟不同时不同确认方式的效率差别不大
- 当丢包率较低时，累计确认的方式可以更高效地利用带宽和传输资源；当丢包率较高时，选择确认能够更快地检测到丢失的数据段，并通过快速的确认触发重传机制。这样可以更及时地补救丢失的数据，避免等待超时导致的延迟，提高了传输效率。