

实验三：基于 UDP 服务设计可靠传输协议并编程实现

- 姓名：马永田
- 年级：2020 级
- 专业：计算机科学与技术
- 指导教师：张建忠 & 徐敬东

实验要求

基于给定的实验测试环境，通过改变延迟时间和丢包率，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

其中：router转发会有较大延时，文件传输速率不作为评分依据；控制变量法；性能测试指标：吞吐率、时延，给出图形结果并进行分析

性能对比

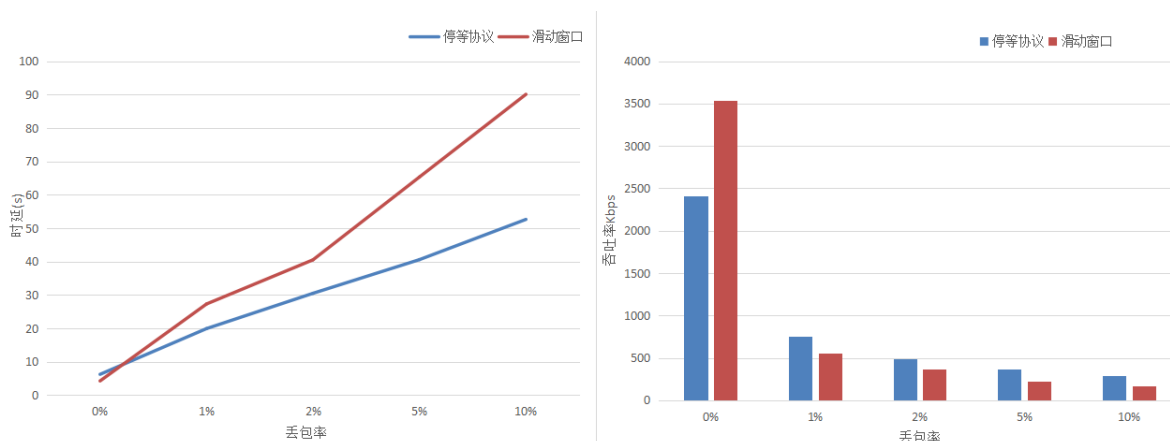
使用文件 1.jpg 进行测试，文件大小为18.73MB，采用控制变量法进行性能对比实验：

停等机制与滑动窗口机制性能对比

测试停等机制与滑动窗口机制性能对比，其中停等机制超时重传时间为200ms，滑动窗口机制中服务端累积确认大小为5，窗口大小设置为15，超时重传时间为650ms

丢包率对性能的影响

设置延迟时间为0ms，改变丢包率，探究不同丢包率对性能的影响：



如图可见，当丢包率为0%时，滑动窗口机制的性能优于停等协议，滑动窗口机制的时延更低，吞吐率更高，性能大约1.4倍；而当丢包率逐渐提高时，停等机制与滑动窗口机制的都性能开始大幅度下降，其中滑动窗口机制的时延更高，吞吐率更低，性能更差。

分析停等协议与滑动窗口协议的流程

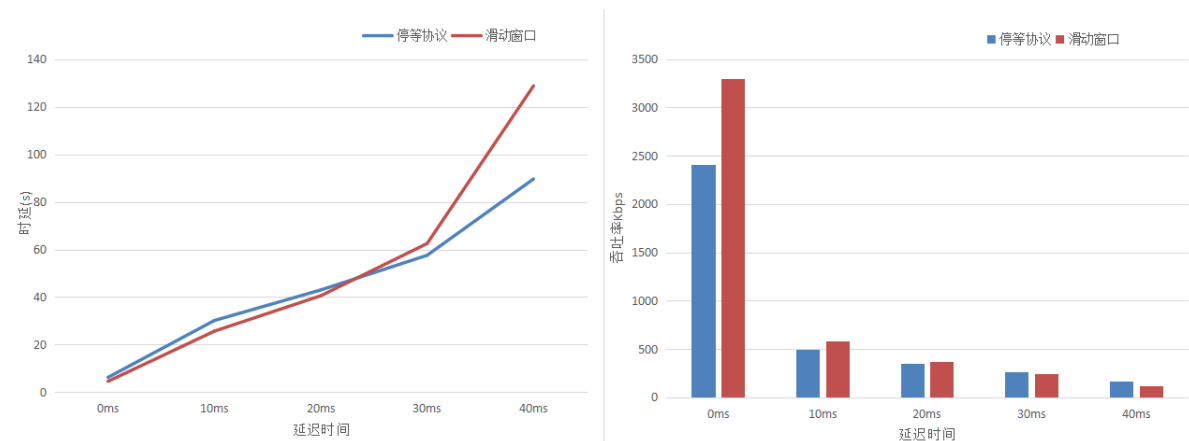
- 假设如果客户端打包和发送分组的时间相对服务端处理和回复的时间更慢，则客户端的发送线程实际是全程满载状态，则双线程滑动窗口机制中缓冲区发送的时间可掩盖客户端接收ACK和处理的时间，即停等协议中的等待时间被掩盖
- 而若是服务端的处理和回复的时间相对较长的话，客户端缓冲区发送的时间无法完全掩盖其他时间(实际报文往返时间对称，因此流水线机制下依旧可以掩盖很长的时间)，由于滑动窗口机制中服务端采用累积确认，因此相对停等机制也会节省很多时间

所以理论上讲滑动窗口机制的性能应当比停等机制更好，且根据理论分析，两种机制的性能应该会相差较大，滑动窗口的流水线机制大约会有5倍以上的性能提升，但实际实验中还会有一些其他因素的影响，例如多线程会存在一些额外的开销，日志打印的I/O时间较长，滑动窗口机制中的处理逻辑更复杂，时间测量存在一定误差等原因，因此实测的性能差别远未达到理论值。

在丢包的情况下，停等协议只需要等待超时后重传当前丢失的分组；但由于滑动窗口采用的是GBN协议，在丢包的情况下可能会重传大量分组，一方面这样会占用大量资源，另一方面由于路由程序设定的是累积满一定数量的数据包后就会进行丢包，因此重传的分组也很有可能出现再次丢包的情况。尤其是在丢包率较高的情况下，客户端会重发大量的数据包，且由于接收方是累积确认，按序接收，一旦发生丢包，之后到来的数据包对接收方来说都是无意义的，最终严重影响到协议的性能，因此随着丢包率的提高，滑动窗口机制的性能大幅度下降，且比停等协议更差。

延时时间对性能的影响

设置丢包率为0%，改变延迟时间，探究不同延迟时间对性能的影响：



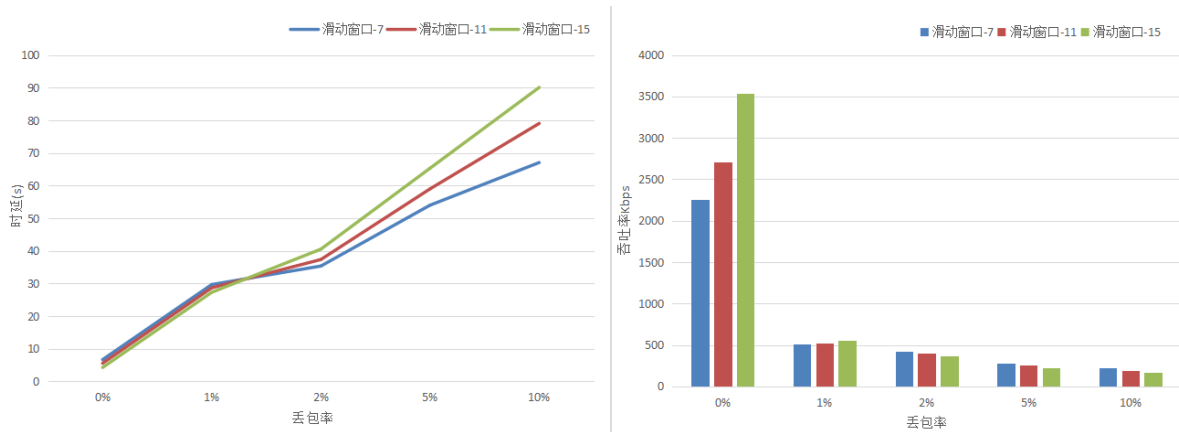
如图可见，当延迟时间较小的时候，滑动窗口机制的性能要优于停等机制，且时延与延迟时间大致呈线性增长；但当延迟时间超过30ms的时候，滑动窗口协议的性能下降的更快，且比停等机制更差。

分析原因这是由于滑动窗口机制采用的GBN协议，当延迟时间设置较高的时候，可能会触发超时重传机制，重传大量分组，而接收方也会接收大量冗余分组，造成资源的浪费，而停等协议中只会重传当前分组，且由于延迟时间未到达设定的重传时间，因此停等协议的时延随延迟时间大致呈线性变化；而延迟时间较低的时候，根据前文进行的分析，滑动窗口的性能会比停等协议更高。

滑动窗口机制中不同窗口大小对性能的影响

丢包率对性能的影响

设置延迟时间为0ms，改变丢包率和窗口大小，探究不同丢包率与窗口大小对性能的影响：

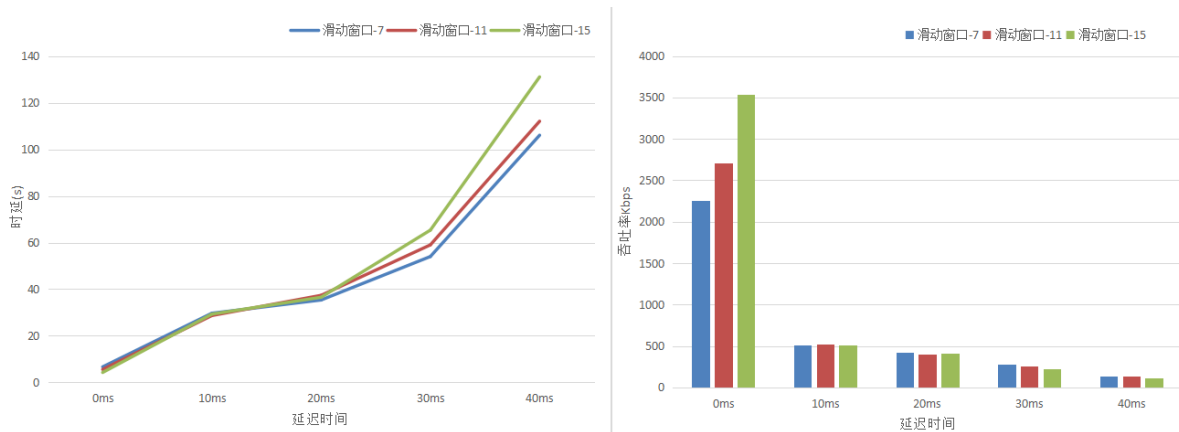


如图可见，随着丢包率的提高，不同窗口大小的性能都会逐渐变差。其中当丢包率较小的时候，窗口越大，吞吐率相对越高，性能相对越好；而当丢包率较高时则反过来，窗口越大，吞吐率相对越低，性能相对越差。

根据前文中我们对停等机制与滑动窗口机制的流程分析可知，当丢包率较大时，窗口越大，丢包后重传的分组也就会越多，重传的分组也更有可能会再次丢包，也就会导致更多的资源浪费，因此性能也就会更差；但当丢包率较小时，重传对GBN协议造成的影响变小，流水线机制对性能的影响则相对更大，窗口越大，流水线效率更高，性能因此也就越好。

延时时间对性能的影响

设置丢包率为0%，改变延迟时间和窗口大小，探究不同延迟时间和窗口大小对性能的影响：



如图可见，随着延迟时间的变大，时延也变大，性能越来越差，且时延提高的速度越来越快；延迟较低时，窗口越大性能越好，延迟较高时则相反，窗口越大性能越差。

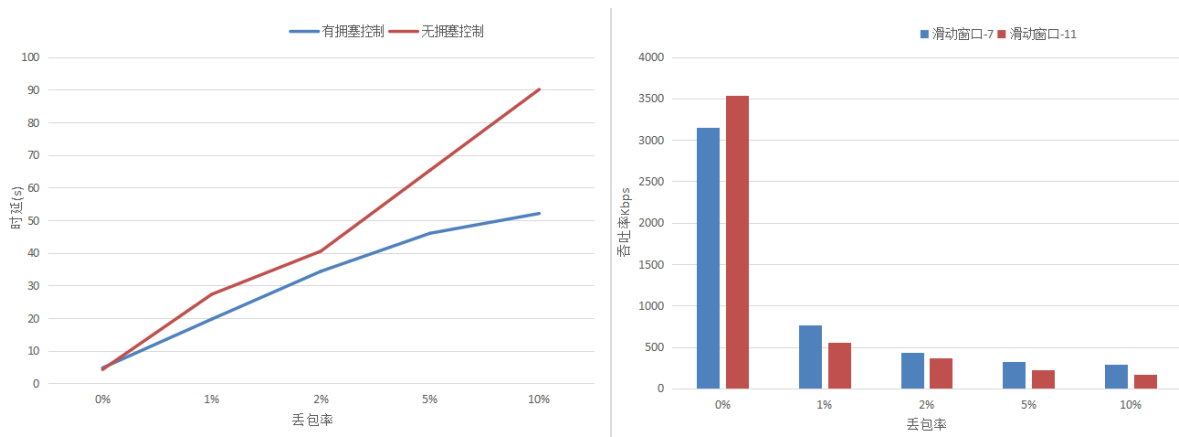
分析原因是当延迟时间较小时，流水线对性能的影响较大，因此窗口越大性能越好，而当延迟时间较大时可能会触发超时重传，造成资源的浪费，且窗口越大，重传的分组越多，性能也就越差。

有拥塞控制和无拥塞控制的性能比较

实验中设置接收方累积确认大小为5，发送方的超时重传等待时间为650ms，其中无拥塞控制的窗口大小为15，有拥塞控制的慢启动门限 `ssthresh` 为24。

丢包率对性能的影响

设置延迟时间为0ms，改变丢包率，探究不同丢包率对性能的影响：

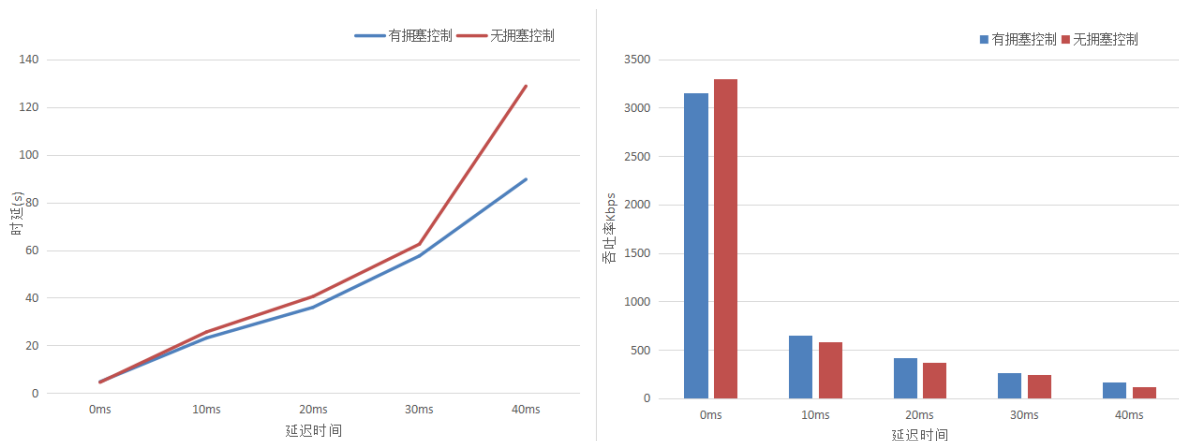


如图可见，随着丢包率的提高，两种协议的性能都会下降，其中有拥塞控制的协议性能优于无拥塞控制的协议性能，且随着丢包率的提高，性能的差别变大。

分析原因是由于在添加了拥塞控制之后，窗口的大小会不断变大，很快就会超过无拥塞控制的固定窗口大小，而当丢包时，虽然拥塞控制的窗口大小会减半，但之后仍会逐渐增大。因此丢包率较低时，有拥塞控制的窗口相对更大，性能也就相对更高；而在丢包率较大的情况下，根据前文中的分析，固定窗口大小会导致大量重传以及重传丢包后再次重传，这就会严重影响到性能，而有拥塞控制的情况下频繁丢包会导致窗口越来越小，GBN协议需要重传的分组数也就更少，重传的分组再次丢包的概率也更低，因此丢包的代价也就更小，性能也就比固定窗口的情况更好。

延迟时间对性能的影响

设置丢包率为0%，改变延迟时间，探究不同延迟时间对性能的影响：



如图可见，随延迟时间的增加，性能逐渐变差，且最初大致呈线性增长，当延迟时间较高时，时延提高的速度变快，其中有拥塞控制的性能一直比无拥塞控制的性能更好。分析原因是当延迟时间较高时，可能会产生超时重传的情况，因此时延变大的速度提高(即斜率更高)，而有拥塞控制后超时重传会导致窗口变小，这样GBN协议重传的分组数就会更少，因此性能相对更好。