

计算机网络3-4实验

姓名：聂志强 学号：2012307 专业：信息安全

一. 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

二. 报告说明

本次实验采用控制变量法，做三组对比实验（停等机制与滑动窗口，滑动窗口大小的影响，有无拥塞控制），通过分别控制延迟时间和丢包率，以吞吐率为指标得出实验结果（这里不以传输时间作为指标，因为每次传输的内容都一样，所以仅仅以吞吐率作为指标即可）。选择图片1（1.77 MB）作为本次实验的测试传输文件。

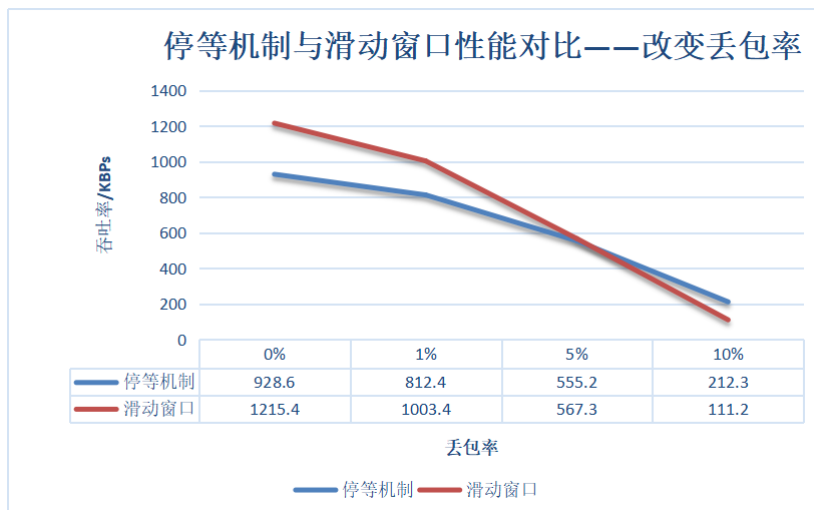
- 对比停等机制与滑动窗口性能
- 对比不同滑动窗口对性能的影响
- 有无拥塞控制

三. 停等机制与滑动窗口机制性能对比

滑动窗口大小采用固定值10，滑动窗口采用的协议为GBN协议。

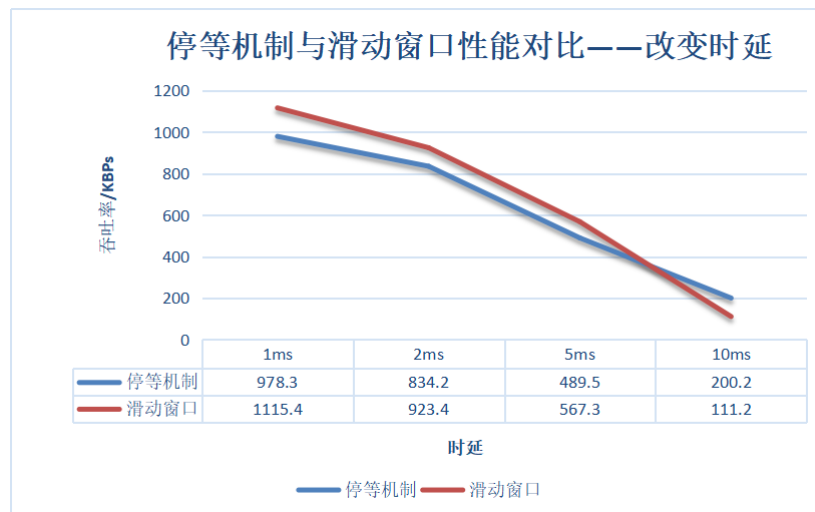
1. 控制时延为0，改变丢包率，对比停等机制和滑动窗口的吞吐率

- 分析：当丢包率较低时，滑动窗口的传输时间低于停等协议，吞吐率高于停等协议；当丢包率较高时，滑动窗口的传输时间高于停等协议，吞吐率低于停等协议。
- 原因：由于滑动窗口采用的是 GBN 协议，当丢包率较高时，由于 GBN 协议每次都要将发送缓存区中的包全部重发，所以传输时间反而比停等协议更长。



2. 控制丢包率，改变时延，对比停等机制和滑动窗口的传输时间和吞吐率

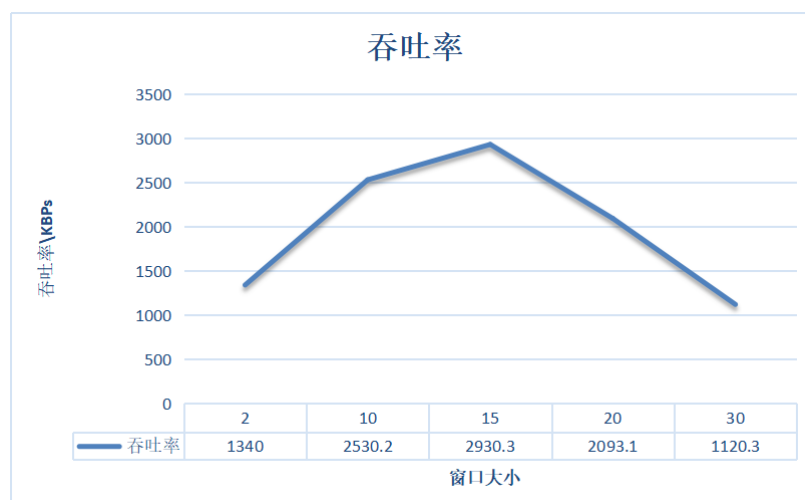
- 结果：本程序设定的超时重传时间阈值为4ms，时延在阈值之前时，滑动窗口的传输时间低于停等协议，吞吐率高于停等协议；当时延超过阈值时，滑动窗口的传输时间高于停等协议，吞吐率低于停等协议。
- 原因：当时延较小小于阈值时，因为滑动窗口机制允许连续发送几个报文，而不是等到回复报文回复之后才继续发送下一个报文，因此相当停等机制效率更高。当时延超过阈值时会发生超时重传现象，停等机制会重发该报文，但滑动窗口机制会重发窗口内所有未确认的报文段，相对于停等机制效率会低。



四. 滑动窗口机制中不同窗口大小对性能的影响

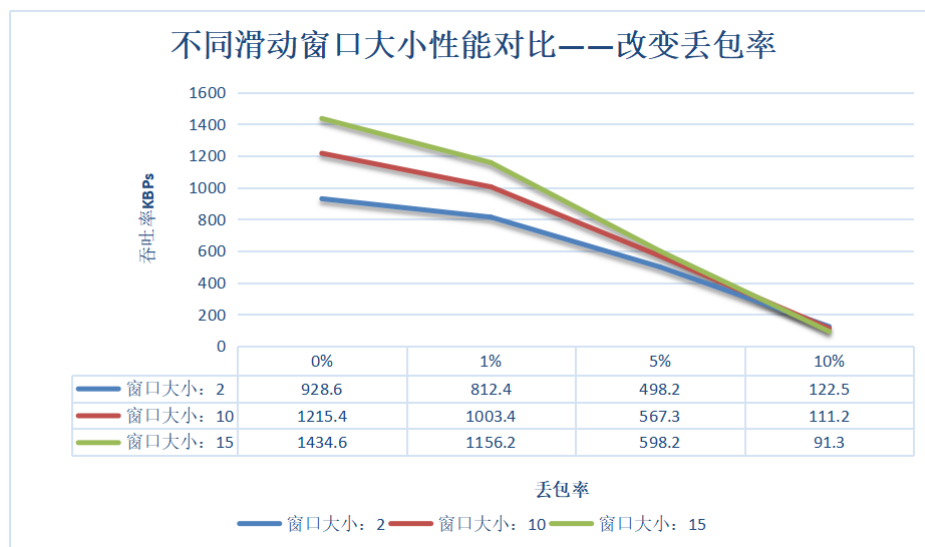
1. 时延和丢包率均为0，对比不同滑动窗口大小的吞吐率

根据图表可以看出，当窗口逐渐增大时，发送端可以发送多个包，发包效率提升，吞吐率也增加，当到达窗口数15左右，吞吐率达到顶峰，之后继续增大窗口，容易造成网络堵塞，接收端来不及接收数据包，会造成丢包，延时，从而不断重复发包，吞吐率反而开始下降，窗口数45左右，吞吐率骤降。



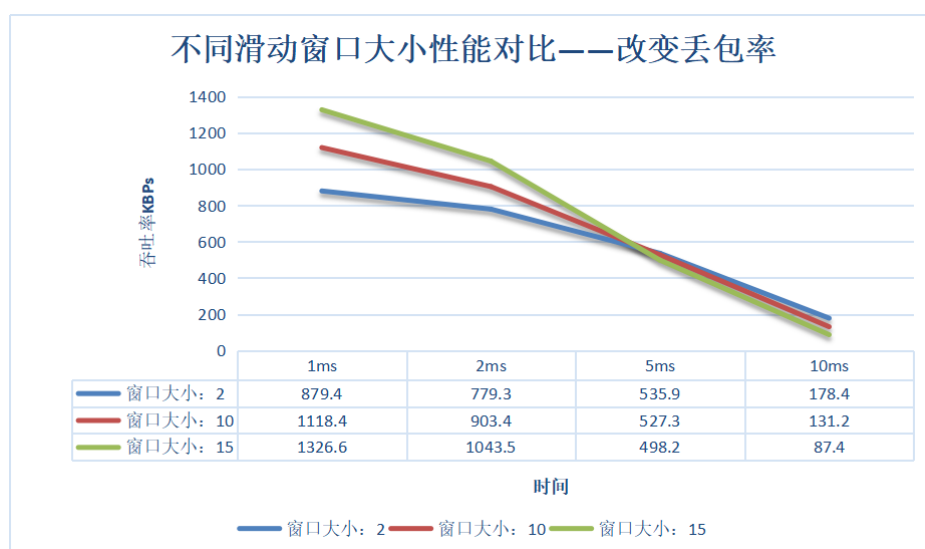
2. 控制时延为0，改变丢包率，对比不同滑动窗口大小的吞吐率

- 结果：当丢包率较低时，不同窗口大小的传输时间差别不大；当丢包率较高时，随着窗口大小增大，传输时间增大，吞吐率减小。
- 原因：因为如果一个包丢失，那么在发送缓冲区中后面发送的包都是无意义的，所以窗口越大，一旦数据报丢失，要重发的数据报就越多，这就导致了窗口变大，性能反而下降。



3. 控制丢包率为0，改变时延，对比不同滑动窗口大小的传输时间和吞吐率

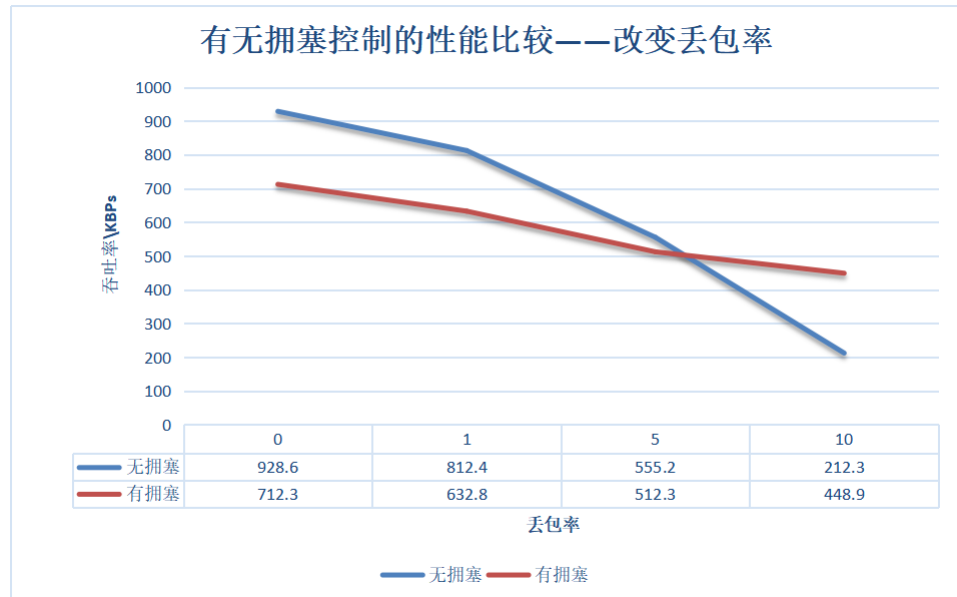
- 结果：由表可知，当窗口比较小时，性能比较好，而当窗口逐渐增大后，性能会下降，但增大到一定程度，继续增大窗口，性能又会有所提升。



五. 有拥塞控制和无拥塞控制的性能比较

1. 控制时延为0，改变丢包率，对比有无拥塞控制的传输时间和吞吐率

- 结果：当丢包率较低时，无拥塞控制性能较有拥塞控制更好；当丢包率较高时，有拥塞控制性能比无拥塞控制更好。
- 原因：无拥塞控制时，窗口大小固定，随着丢包率增大性能降低；而有拥塞控制时，丢包率大时RENO 算法会自动下调窗口大小，使性能得到提升。



2. 控制丢包率为0，改变时延，对比有无拥塞控制的传输时间和吞吐率

- 结果：当时延较低时，无拥塞控制与有拥塞控制性能相当；当时延较高时，有拥塞控制性能比无拥塞控制更好。
- 原因：有拥塞控制时，时延大时RENO算法会有快速重传，使性能得到提升，并且会自动下调窗口大小，避免超时后需要重传的过多。

