



南開大學

Nankai University

计算机学院

计算机网络实验 3-4 报告

基于传输协议的性能比对实验

姓名：林帆

学号：2113839

专业：计算机科学与技术

2023 年 12 月 23 日

目录

1 实验介绍	2
2 对比实验 1: 停等机制与滑动窗口机制性能比对	2
2.1 调整丢包率	2
2.2 调整延时	3
3 对比实验 2: 滑动窗口机制中不同窗口大小对性能的影响	3
3.1 GBN 机制	3
3.2 SR 机制	4
4 对比实验 3: 累计确认和选择确认的性能比较	4
4.1 调整丢包率	5
4.2 调整延时	5

1 实验介绍

本次实验是完成了 3-1 到 3-3 共 3 个版本的基于 UDP 的传输基础后, 对不同传输机制和拥塞控制机制进行的对比实验。根据实验要求, 完成了以下三组对比实验:

- 停等机制与滑动窗口机制性能比对
- 滑动窗口机制中不同窗口大小对性能的影响
- 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。

实验中主要控制的自变量为**延时**和**丢包率**, 性能测试的指标为**传输时长**和**吞吐率**。同时, 固定传输的文件为图片 1.jpg, 大小为 1857353 字节。

2 对比实验 1: 停等机制与滑动窗口机制性能比对

在实验 1 中, 我对停等机制 (3-1) 和 GBN 传输 (3-2) 进行了比较, 在滑动窗口中, 固定了窗口大小 win=4 进行对照实验。

2.1 调整丢包率

首先固定时延 timer=5ms, 从丢包率 0 开始, 以梯度 5% 动态调整丢包率, 观察图片 1 在两种机制下传输的性能, 所得吞吐率 (Byte/s) 和传输时间 (ms) 如下所示:

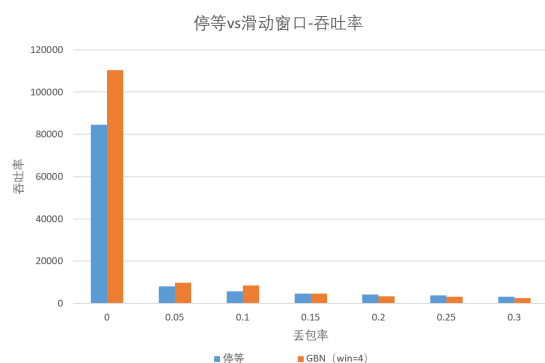


图 2.1: 吞吐率对比

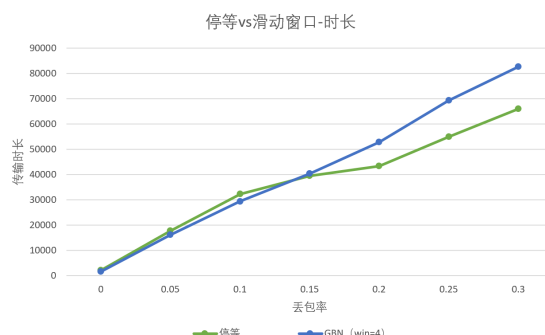


图 2.2: 传输时长对比

观察上述测量结果, 我们可以发现, 在没有丢包或丢包率较小的情况下, 滑动窗口的传输吞吐率高于停等机制, 传输时间也更短。但是随着丢包率逐渐增大, 两种机制下的吞吐率都迅速下降, 传输时长也不断增加。到丢包率 >20% 时出现了滑动窗口机制的传输效率小于停等机制的情况。

分析可知, 停等协议的序列号只有 0 和 1, 即发送端每次只确认前一个数据包的 ack, 且必须确认后才能发送下一个数据包; 而滑动窗口机制拥有更多序列号以及容纳性强的窗口, 使得即便在等待确认某个数据包的 ack 时, 也能并行发送下一个数据包, 从而提升了传输效率。这便是前期滑动窗口机制的吞吐率高于停等机制, 传输时间更短的原因。

对于丢包率较高时出现的滑动窗口效率不如停等效率的反常情况, 有多个原因

- 丢包数较多时, 滑动窗口的机制会陷入持续大量的超时重传, 且每次重传需要重传窗口内的全部数据包, 从而占据了大量的带宽资源。

- 同时, 每次重传的过程又可能带来新的丢包, 又进一步增加了丢包的可能性。
- 由于 GBN 按序确认 ack, 即发送端每次必须确认窗口内第一个序列号的数据包的 ack, 才能移动窗口, 那么当丢包数较多时, 对于待确认 ack 之后发来的数据包, 又将全部丢弃, 造成了更多的丢包情况。
- GBN 是多线程的, 在编程实现的过程中, 为了保证三个线程中的一些全局变量同步, 加入了很多锁, 因而上锁会抑制并行传输的效率。

2.2 调整延时

固定 5% 的丢包率, 对延时从 0 到 100ms 按照梯度 25ms 开始递增调整, 最终得到停等机制与 GBN 机制的吞吐率和传输时间如下所示:

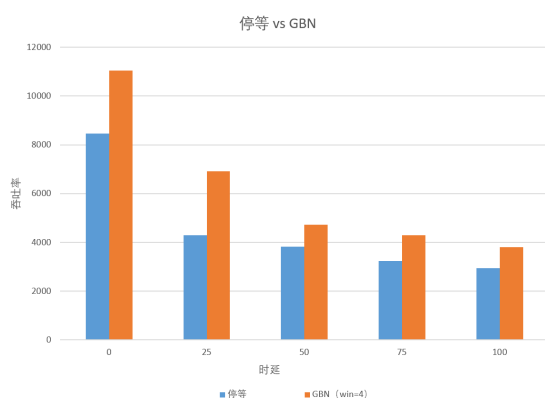


图 2.3: 吞吐率对比

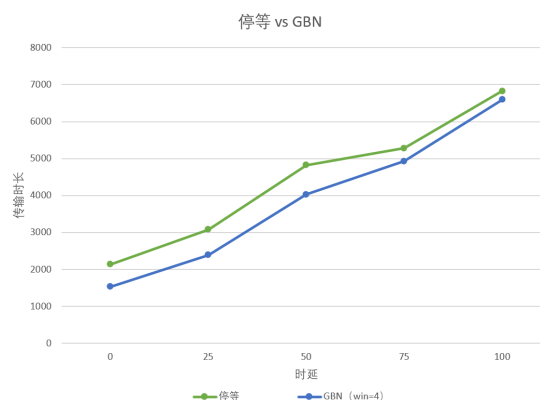


图 2.4: 传输时间对比

分析可以看到, 随着延时逐渐增加, 无论是停等机制还是 GBN 机制的传输效率都会逐渐降低, 这与延时增加了处理单个包的时间有关。但 GBN 的效率还是会高于停等机制, 说明 GBN 的多序列号流水线机制的优势还是较为稳定的。

3 对比实验 2: 滑动窗口机制中不同窗口大小对性能的影响

3.1 GBN 机制

首先对 GBN 机制中不同窗口大小对性能的影响进行测试, 分别设置了窗口大小 win=4, 8 和 16, 依然是控制时延 timer=5ms, 设置 5% 的丢包率梯度, 对不同窗口下传输图片 1.jpg 的吞吐率和传输时间进行测试, 所得结果如下所示 (在下面那一页):

从上述结果中可以看到, 在丢包率为 0 或者较小的情况下, 吞吐率大小: win=16>win=8>win=4, 传输时间: win=16<win=8<win=4, 即**传输效率随着窗口的增大而增大**。分析原因是丢包率较小的情况下, 非必要重传的次数较少, 因而传输的效率与非堵塞状态的流水线效率直接挂钩, 窗口大小越大, 流水线的速率越高, 传输效率也就越高, 从而提升了吞吐率, 减少了传输时间。

但当丢包率较大时, 情况为, 吞吐率大小: win=16<win=8<win=4, 传输时间: win=16>win=8>win=4, 即**传输效率随着窗口增大而减小**, 这看上去是个反常现象。分析原因主要有以下两点:

- 在丢包率较高的情况下, 超时重传的次数增多, 而 GBN 的超时重传需要重发整个窗口中的数据包, 当窗口越大, 需要重发的数据包就越多, 重发的代价越高, 大大增加了传输时间。

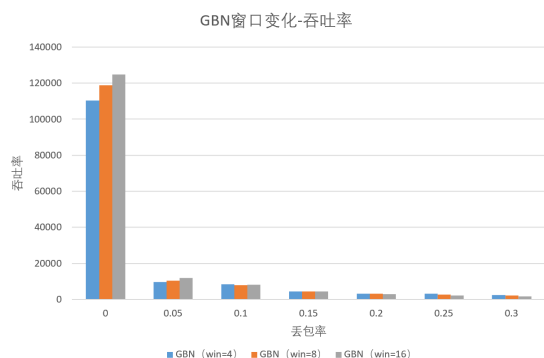


图 3.5: 吞吐量对比

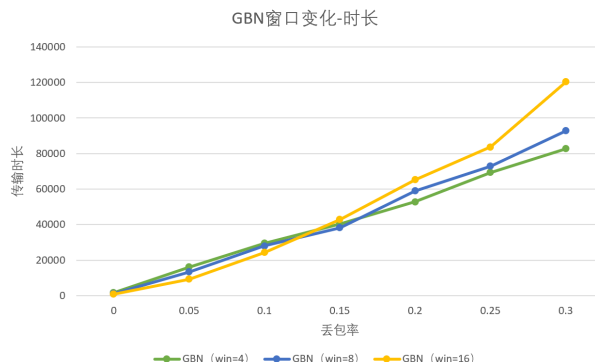


图 3.6: 传输时间对比

- 窗口越大，一次性发送的数据越多，过程中丢包的可能性就越大，造成网络拥塞的可能性也增大，同时会消耗更多的带宽资源。

3.2 SR 机制

对 SR 机制的测试，采取与前文所述 GBN 相同的变量设置，得到下面的结果：

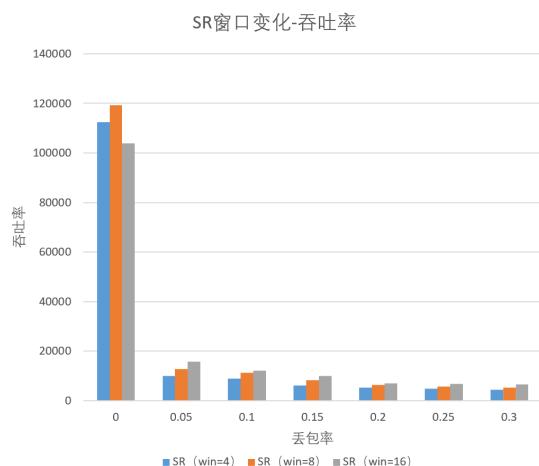


图 3.7: 吞吐量对比

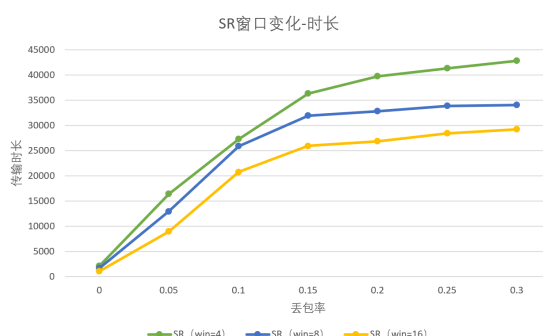


图 3.8: 传输时间对比

从上面的两张图可以看到，SR 出现了与 GBN 不同的情况：**传输效率始终随着窗口增大而增大**（丢包率为 0 的情况可能是线程开销不稳定导致的）。

分析原因主要是，SR 在超时情况下的选择重传机制不再重传窗口内全部数据包，而是只重传未确认的数据包，减少了重传的开销代价，且重传过程中丢包的可能性也大大降低，同时发送端也不会因为必须按序接收而随意丢弃接收到的数据包导致进一步重传的连锁反应，从而大大发挥了流水线的机制，规避了 GBN 中的异常情况，让较大的窗口提升至更高的传输效率。

4 对比实验 3: 累计确认和选择确认的性能比较

该实验比对了 GBN 协议和 SR 协议的传输性能。

4.1 调整丢包率

在对比实验 2 中可以看到, GBN 协议在不同丢包率下, 窗口大小对传输性能的影响不太相同, 因而对不同丢包率下, GBN 和 SR 的多种窗口大小均进行了比对, 依然控制延时 = 5ms, 调控丢包率的梯度未 5%, 所得结果如下:

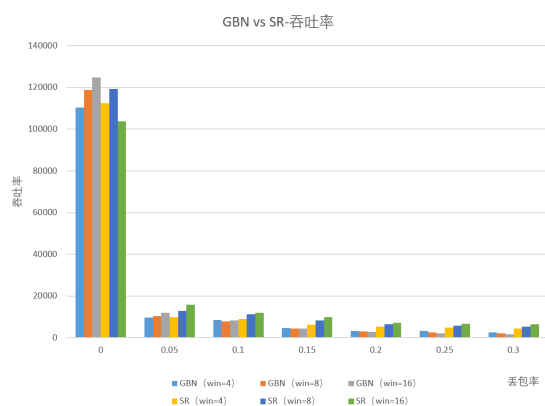


图 4.9: 吞吐率对比

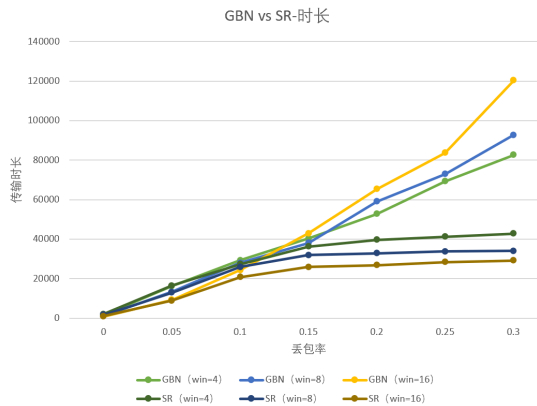


图 4.10: 传输时间对比

从传输时间上可以更直观地看出, 丢包率为 0 或较低的情况下, GBN 和 SR 的传输性能没有太大的区别, 原因主要是在该情况下没有过多超时重传事件发生, 因而二者的流水线的效率都发挥到了较高水平。

但丢包率随着逐渐上升, 可以看到 SR 的优势越来越明显, 传输时间也较为稳定地缓慢增加, 而 GBN 的传输时间增加非常明显, 大窗口的传输时间增加尤其快 (这一点在前文的对比实验 2 已经做过了解释)。分析 SR 的优势原因, 主要还是前文提到的——相比于累计确认, 选择确认不需要重发窗口内的所有数据包, 因而重传的开销代价小; 同时, 选择确认可以确认待接收序号之后的数据包, 而不是选择丢弃, 因而避免了 GBN 大量丢弃可能引发的丢包的连锁反应。

4.2 调整延时

然后固定丢包率为 5%, 对延时进行梯度为 25ms 的调整, 测得窗口为 4 的 GBN 和 SR 协议传输图片 1.jpg 的吞吐率和传输时间如下所示:

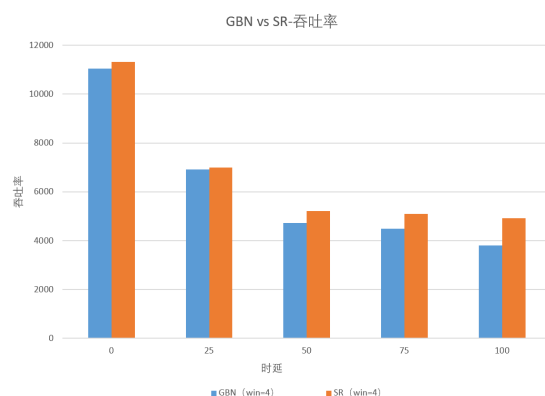


图 4.11: 吞吐率对比

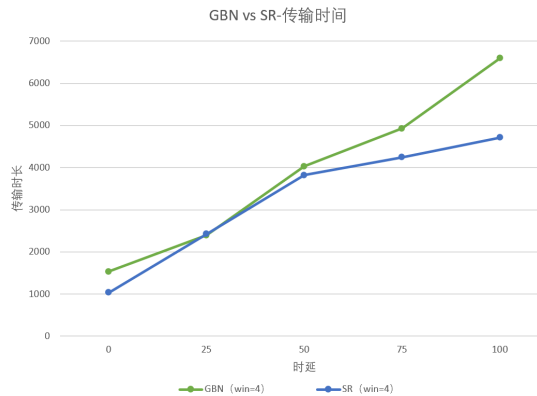


图 4.12: 传输时间对比

可以看到, 两种传输方式下随着延时增加, 传输性能都会变差, 这点不再作解释。发现较低延时

下, 二者的传输性能并没有太大的区别; 但高时延下, SR 的传输性能较高于 GBN。分析原因可能是, SR 协议的选择重传可以更好地利用网络带宽, 从而降低重传的开销代价, 也能更快地相应超时丢包的事件, 效率较高, 重传丢包可能也更小, 使得与路由器的重传事件较为有序 (GBN 可能会不断丢弃包而不断向路由器重传), 从而使得延时在整个传输事件中开销影响较小, 从而更好地发挥自己的优势。