



南開大學
Nankai University

计算机学院
计算机网络实验报告

lab3.4-性能测试

姓名：曹骜天

学号：2114036

专业：计算机科学与技术

2023 年 12 月 20 日

目录

| | |
|--|----------|
| 1 实验介绍 | 2 |
| 2 实验内容 | 2 |
| 2.1 停等机制与滑动窗口机制的性能对比 | 2 |
| 2.1.1 延时不变, 改变丢包率进行测试 | 2 |
| 2.1.2 丢包率不变, 改变延时进行测试 | 3 |
| 2.2 滑动窗口机制中不同窗口的大小对性能的影响 | 4 |
| 2.3 相同窗口大小情况下, 累计确认和选择确认的性能比较。 | 4 |

1 实验介绍

本次实验使用了 UDP 实现了可靠数据传输的功能，分别实现了以下三种不同的可靠传输方式：

1. 停止等待协议：发送方在传输每个数据帧后等待接收方的确认帧，如果在超时前收到确认，则发送下一帧；否则，超时后假设帧丢失，重新发送。该协议简单易实现，但效率较低，因为发送方必须等待确认才能发送下一帧，而且超时重发机制导致较长等待时间。
2. Go-Back-N：采用滑动窗口机制，允许发送方连续发送多个帧而无需等待确认。发送方启动计时器等待接收方的累积确认，接收方按序接收帧并发送累积确认。一旦确认帧的序列号达到窗口右边界，发送方停止计时器，滑动窗口，继续发送下一窗口内的帧。尽管提高了网络利用率，GBN 协议可能造成带宽浪费，因为在发生错误时，整个窗口内的帧都需要重新发送。
3. Selective Repeat：SR 协议允许发送方一次性发送多个数据帧，而接收方仅对出错或缺失的帧发送相应的确认，从而提高了网络利用率。接收方维护一个窗口，按序接收帧，并向上层交付已成功接收的数据，同时发送单个确认消息。相比于 GBN，SR 协议更灵活，因为只有发生错误的帧需要重传，避免了整个窗口的重传，提高了传输效率。

停等协议、GBN 协议以及 SR 协议的实验内容均使用测试样例 1.jpg 完成，性能评价指标为传输时延以及吞吐率，接下来将进行详细分析。

2 实验内容

2.1 停等机制与滑动窗口机制的性能对比

在本组实验中，我们采用控制变量法，首先控制延时不变，然后对丢包率进行梯度的比对；然后再控制丢包率不变，对延时进行梯度比对。分别比较了停等机制、滑动窗机制 (GBN)、滑动窗机制 (SR) 的时延以及吞吐率。

2.1.1 延时不变，改变丢包率进行测试

在测试中，控制最大报文段 (MAXBUFSIZE) 为 4096 字节，延时为 5ms，滑动窗的窗口大小为 10，分别得到了时延和吞吐率的测试结果：

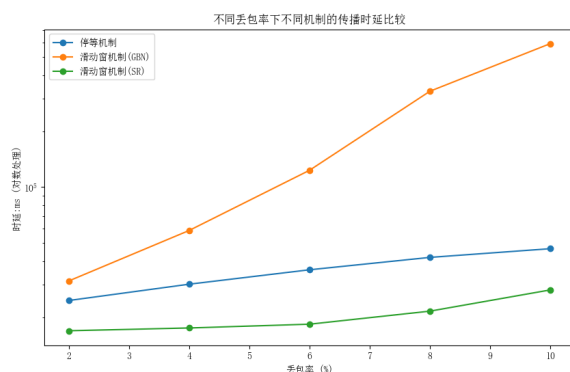
| 丢包率 (%) | 停等机制 | 滑动窗机制 (GBN) | 滑动窗机制 (SR) |
|---------|-------|-------------|------------|
| 2 | 24558 | 31275 | 16886 |
| 4 | 30056 | 58456 | 17480 |
| 6 | 35935 | 123222 | 18317 |
| 8 | 41897 | 328037 | 21530 |
| 10 | 46669 | 591808 | 28005 |

表 1: 不同丢包率下的传输时延对比 (单位:ms)

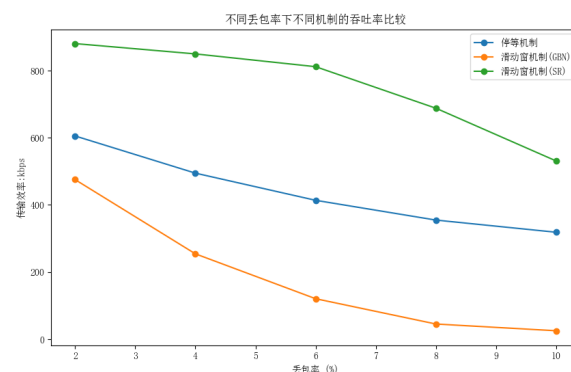
通过表格1、2以及图2.1中的数据可以分析出：从总体的趋势上看，随着丢包率的增加，传输的总时延在不断地增加，吞吐率也在不断变小，符合客观的规律认知。但是比较发现，在三种协议中，SR 的性能较为优越，传输时间的增加和吞吐率的减小相比于另外两个协议较为平缓。**特别注意的是：在图中我们用了取对数处理传输的总时间。**这可以看出虽然使用了滑动窗机制，但 GBN 协议的性能较差，在对数处理后仍然保持较大的增长速率，因此原增长速率应是指数级的，而另外两个的增长是幂级的，

| 丢包率 (%) | 停等机制 | 滑动窗机制 (GBN) | 滑动窗机制 (SR) |
|---------|---------|-------------|------------|
| 2 | 605.05 | 475.102 | 879.949 |
| 4 | 494.371 | 254.188 | 849.254 |
| 6 | 413.492 | 120.586 | 811.204 |
| 8 | 354.651 | 45.2962 | 687.582 |
| 10 | 318.387 | 25.1963 | 530.578 |

表 2: 不同丢包率下的传输吞吐率对比 (单位:kbps)



(a) 传输时延随丢包率变化图。



(b) 吞吐率随丢包率变化图。

图 2.1: 给定延时, 性能测试指标与丢包率的关系

这说明 GBN 收到丢包率的影响很大。从总的性能上来看, SR 的性能较优, 其次是停等协议, 再其次才是 GBN。

原因分析 GBN 在一次丢包之后, 会重传在滑动窗中在其之后的所有报文, 而无论是停等协议还是 SR 协议, 都只需要重传出错的数据。因此, 当丢包率不断上升的过程中, GBN 一开始相比于停等协议可能有优势, 因为不用等待上一个报文的 ACK, 而是采用累计确认, 但是一旦丢包率上升, 就需要频繁重传窗口里的所有报文, 而停等协议只需重传当前丢失的报文, 在相同的超时时间下, 此时的停等协议将会优于 GBN 协议, 而至于 SR 协议, 和停等协议相比, 在发生丢包的过程中不进行等待, 而是先传输后续的报文, 这样节省了一段后续报文的传输时间, 因此优于停等协议, 产生上面的实验情况。

2.1.2 丢包率不变, 改变延时进行测试

在测试中, 固定丢包率为 2%, MAXBUFSIZE 以及滑动窗大小与之前一致。

| 延时:ms | 停等机制 | 滑动窗机制 (GBN) | 滑动窗机制 (SR) |
|-------|-------|-------------|------------|
| 2 | 25168 | 30276 | 17441 |
| 4 | 25638 | 31599 | 17687 |
| 6 | 26631 | 38411 | 36605 |
| 8 | 27269 | 45762 | 39505 |
| 10 | 27677 | 49123 | 41754 |

表 3: 不同延时下的传输总时延对比 (单位:ms)

我们通过表3、4以及图2.2看出, 随着延时的增加, 三种协议的效率都是在下降的。不同的是, 停

| 延时:ms | 停等机制 | 滑动窗机制 (GBN) | 滑动窗机制 (SR) |
|-------|---------|-------------|------------|
| 2 | 590.386 | 490.779 | 851.948 |
| 4 | 579.563 | 468.383 | 840.099 |
| 6 | 557.952 | 385.747 | 405.923 |
| 8 | 544.898 | 323.156 | 376.125 |
| 10 | 536.865 | 302.176 | 355.866 |

表 4: 不同延时下的传输吞吐率对比 (单位:kbps)

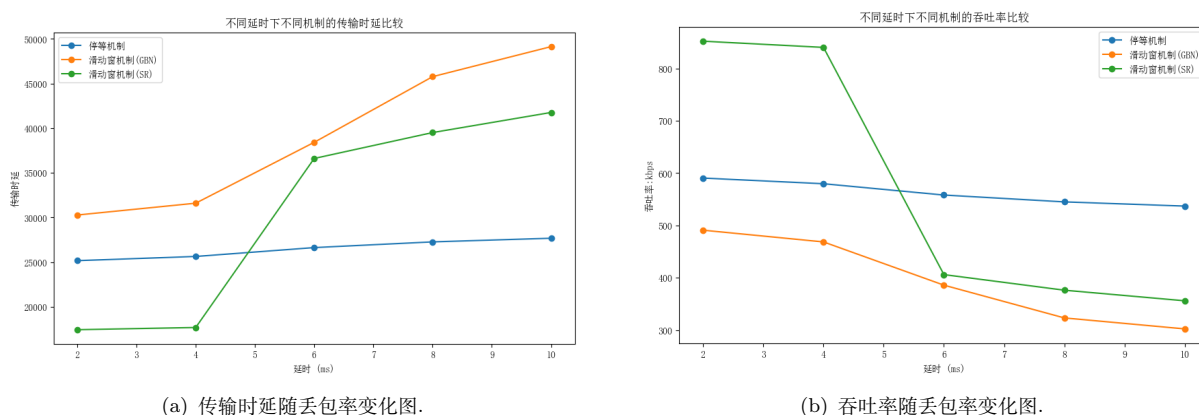


图 2.2: 给定丢包率, 性能测试指标与延时的关系

等协议的传播延时以及吞吐率是在以线性的速度分别上升以及下降, 且延时增加的幅度经计算后**恰好等于每个数据报的延时 X 发送数据报的个数**, 就意味着增加的时间以及改变的吞吐率仅仅为路由器转发的延时而已。但是反观 GBN 以及 SR 协议, 上升的幅度并不等于上面描述的那样。这是由于代码的实现方式导致的, 在代码中, 我们并没有使用真正意义上发送端以及接收端的多线程, 而是仅在发送端实现了发送线程以及接受并检查重传线程, 在增加延时的过程中, 可能会导致执行流异常行为, 例如本来不需要重传的报文因延时的改变而需要超时重传了, 导致效率变化不是那么均匀。

2.2 滑动窗口机制中不同窗口的大小对性能的影响

同样地, 在本组实验中, 我们会采用控制变量法, 固定丢包率以及路由器延时, 改变滑动窗口大小以测量性能。

注: 由于此测量目标和下一目标十分相关, 于是统一在下方说明。

2.3 相同窗口大小情况下, 累计确认和选择确认的性能比较。

在此处, 我们会改变窗口大小, 进行 SR 与 GBN 的自身的纵向对比以及横向对比, 固定丢包率为 2%, 路由器延时为 2ms, 最大报文长度为 4096 字节。

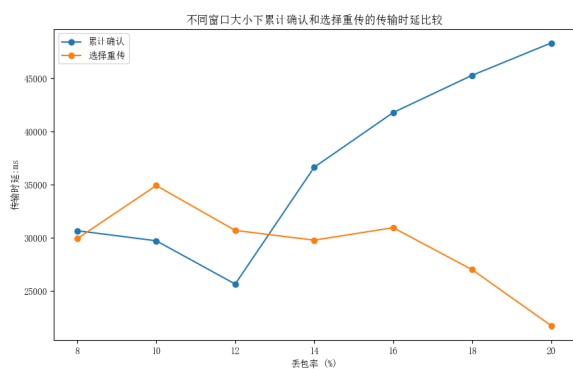
首先回答上一点的问题, 根据表5、6以及图2.3, 在其它参数固定而窗口大小增大的过程中, 采用累计确认的时延为增大趋势, 吞吐率为减小趋势, 总体性能下降, 这是由于在一定的丢包率下, 此次传输**需要重传的的报文总数是不变的**, 也就是说要重传的次数不变, 但是随着窗口的增加, 需要重传的数据包由于 GBN 的缘故变多了, 因此性能下降了, 相反地对于选择确认来说, 若窗口较小, 可能会出现窗口阻塞的情况, 阻止后续报文发送, 而窗口增大, 后续的报文可以继续发送并收到确认, 而由于丢包率不变, 需要重传的报文总数不变, 也就是需要重传的次数不变。若在重传等待 ACK 的过

| 窗口大小 | 累计确认 | 选择重传 |
|------|-------|-------|
| 8 | 30676 | 29932 |
| 10 | 29726 | 34928 |
| 12 | 25649 | 30704 |
| 14 | 36651 | 29786 |
| 16 | 41805 | 30960 |
| 18 | 45313 | 27006 |
| 20 | 48356 | 21740 |

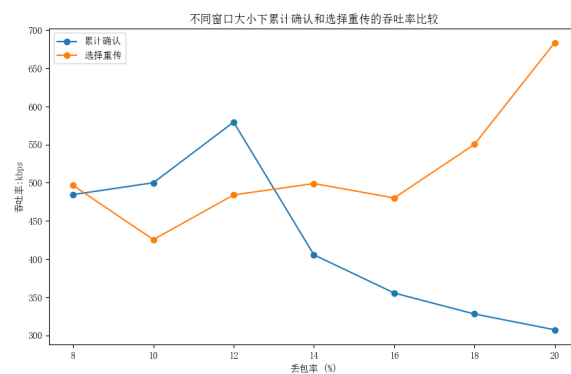
表 5: 不同窗口大小下的传输时延对比 (单位:ms)

| 窗口大小 | 累计确认 | 选择重传 |
|------|---------|---------|
| 8 | 484.379 | 496.419 |
| 10 | 499.86 | 425.413 |
| 12 | 579.314 | 483.938 |
| 14 | 405.414 | 498.853 |
| 16 | 355.432 | 479.936 |
| 18 | 327.915 | 550.205 |
| 20 | 307.28 | 683.479 |

表 6: 不同窗口大小下的吞吐率对比 (单位:kbps)



(a) 传输时延随丢包率变化图.



(b) 吞吐率随丢包率变化图.

图 2.3: 两种重传机制的效率与窗口大小的关系

程中窗口满了，则不能继续发送；当增大了窗口数量，则可以继续发送，这样也就提高了发送的效率，因此在一定丢包率的情况下时延下降、吞吐率上升。

由于累计确认配合的是 GBN 协议使用，选择确认配合 SR 协议使用，在大部分情况下，相同窗口时，选择确认是要比累计确认较为高效的，特别是在丢包率上升以及窗口大小增加时，选择确认的优势会体现得淋漓尽致，从另一种角度上来看，选择确认通过对窗口中的每一个报文添加一个计时器，增加了额外的空间开销，属于一种宏观意义上的“空间换时间”的做法。