

华东师范大学数据科学与工程学院报告

课程名称:计算机网络与编程	年级:22级	实践成绩:
指导教师:张召	姓名:郭夏辉	学号:10211900416
实践名称:GBN/SR窗口大小和序号空间大小的关系	实践日期:2023年5月17日	实践编号:No.001
组号:1-416	实践时间:2023年5月17日	

一、作业内容

说明窗口大小和序号空间大小的关系，并解释原因。

二、完成过程

对于这次作业的完成，我尽量结合所学知识，给出一个较为严谨的证明。

2.1 知识回顾

首先我们来简单回顾一下流水线式的数据传输协议，一般采用两种方法来进行差错恢复——回退N步(GBN)和选择重传(SR).

2.1.1 GBN

发送方:

1. 维护了一个大小为N的窗口，动态地挪动它。在这个窗口中，只能放置已发送未确认、未发送的分组，在这个窗口之前的分组都是已发送且已被确认了的。
2. 可以在未接受到接收方 ACK 的情况下把这个窗口中的多个分组全部发送出去
3. 收到了 ACK_n 之后窗口会前移(n 为窗口最左端元素之标号):这里的确认是一种累计确认——发送方认为接收方不仅接受了标号为 n 的分组，而且接收到了之前的那些分组。
4. 如果已发送的分组确认时间超时:发送方会重传该分组和在这个分组之后所有的"发送但未确认"的分组，这也就叫回退N步。

接收方:

1. 维护了一个大小为1的接受窗口:为什么为1呢? 这代表了接收方希望接收到的下一个分组
2. 如果按序接收到了希望收到的分组 n ，窗口前移,并且发送 ACK_n

3. 如果没有按序接收到分组:丢弃, 并且重发最近发送的那个 ACK_n

2.1.2 SR

发送方:

1. 与GBN类似, 维护了一个窗口, 可以在未接受到接收方 ACK 的情况下把这个窗口中的多个分组全部发送出去。
2. 只有按序收到已发送的分组的确认时, 窗口才能向前滑动, 滑动多少看窗口左端确认了多少个。(其实比如说收到了 ACK_n , 如果这个 n 不是窗口最左端元素之标号, 不滑动; 否则滑动。)如果收到了的确认不是按序的, 也要记录下来。
3. 如果已发送的分组确认时间超时:发送方会重传该分组。

接收方:

1. 接收方维持了一个窗口, 这个窗口的大小可以不为1。
2. 如果按序接收到了分组 n (这个 n 是窗口最左端元素之标号), 窗口前移, 并且发送 ACK_n 。
3. 如果没有按序接受到分组, 也发送 ACK_n , 顺便记录一下。

2.2分析过程

根据课本上面的实例, 围绕老师上课所讲的那个窗口大小可能导致的问题, 我来对窗口的大小分析一下吧。

2.2.1 SR

其实有一个小小的细节, 就是接收方的窗口尺寸可以和发送方不一样吗? 我在纸上写了很多的例子, 发现不一样也是可以的。但是需要满足接收方的窗口大小不超过发送方的窗口大小(否则在逻辑上是不成立的)。为了方便论述, 我在这里只讨论接收方和发送方的窗口大小一样, 如果不一样, 也是同理的, 毕竟求的是上界。

假设序号空间的大小为 N , 窗口的大小是 M , 我们将发送的分组按次序这样去标号:

0 1 2 N-1 N N+1 2N-1

当然它们的序号是:

0 1 2 N-1 0 1 N-1

这个很容易用取模运算来描述:序号是 $h(u)$, 标号是 u $h(u)=u \bmod N$

在这样的情况下, 我们如何来描述问题呢? 什么时候会出现不成立的情况?

出现矛盾的本质其实是接收方无法判断来自发送方的重传到底是真的重传呢还是新的分组呢, 这又是因为接收方相对于发送方的窗口来说, 存在一个序号一样, 但是标号不一样的元素!

假设 u, v 是非负整数标号, u 是来自发送方的窗口之标号, v 是来自接收方的窗口之标号, 满足 $h(u) = h(v), u \neq v$

u, v 之间的距离 $|v - u|$ 最大能有多大?

接收方在发回ACK确认时可能会出现无序、丢失现象, 这个会使得发送方和接收方的窗口移动并不总是同步。这个不同步使得接收方的窗口移动相较于最初移动的程度大于等于发送方的窗口, 则 $u < v$ 。

我们取一个最极端的情况, 让发送端的那个窗口一直不移动, 但是接收端的窗口总是能移动——接收端发出的ACK全部都丢失了。在这个情况下, 两个窗口能恰恰好好地交错开:

发送窗口: $i \ i+1 \ i+2 \ \dots \ u \ \dots \ i+M-1$

接收窗口: $i+M \ i+M+1 \ \dots \ v \ \dots \ i+2M-1$

交错开之后, 两个窗口共有 $2M$ 个元素, 当然如果继续让接收窗口往右边走一点也是可以的, 这样从发送数组第一个元素到接收窗口最后一个元素(包括这两个)就有了 $2M + t$ 个元素。然后我们可以发现一个不等式

$[u, v]$ 之间的元素个数 $v - u + 1$ 应该不超过 $2M + t$

即 $v - u + 1 \leq 2M + t, t \geq 0$

其中 $v - u = kN, k \geq 1, k \in \mathbb{N}^+$

由恒成立性和一般性, 可以导出不成立情况下的解: $2M \geq N + 1$

也很容易就可以证明这个解是不成立情况下解的全集。

取补集, 得 $2M < N + 1$ 由于皆为整数, $2M \leq N$

2.2.2 GBN

其实这个老师并没有特别要求去做, 只不过我还是想分析一下。

相对来说这个更容易去讨论, 但是其实也是有点复杂的。

由于接收方的窗口大小只为1, 这个很方便去讨论:

1. 如果接受窗口在发送窗口之间, 这时只要能满足发送窗口内没有重复的序号之分组, 就可以区分(否则就不能, 因为接收方发送一个有重复序号的分组的ACK, 接收方就分不清楚了), 可以得到 $M \leq N$
2. 如果接受窗口在发送窗口最左侧, 发送方发送一个分组正在等待确认, 这个时候与第一种情况类似, 同样只要能满足发送窗口内没有重复的序号之分组, 就可以区分, 即 $M \leq N$
3. 如果接受窗口在发送窗口最右侧的下一个, 接收方收到了所有的分组, 还没来得及确认完。在这种情况下要求发送方和接收方内部没有重复的序号之分组, 否则, 若发送方发送了一个有重复标号的分组, 接收方会以为收到了, 但其实不对呢, 这时即 $M + 1 \leq N$

综上所述 $M \leq N - 1$

2.3结果

更为一般地，假设发送方的窗口大小 W_s ,接收方的窗口大小 W_r

SR协议:

$$W_r \leq W_s$$

$$1 \leq W_s$$

$$2W_s \leq N$$

$$W_s + W_r \leq N$$

这个的证明过程同理，只不过变换一下长度罢了。

GBN协议:

$$W_s + W_r \leq N$$

只不过 $W_r = 1$

三、总结

通过这一系列的分析，我对GBN,SR的原理有了一个深入的理解之后，对于课本上没有给出详细证明的窗口大小和序号空间的关系在数学推理之后有了更深刻的认识。看到一个通式的导出我还是十分激动的:

$$W_s + W_r \leq N, 1 \leq W_r \leq W_s$$