5 回退 N 帧协议

为了提高传输效率，当发送端等待确认时，必须传输多个分组。换言之，当发送端等待确认时，我们需要让不止一个分组处于未完成状态，以此确保信道忙碌。为此有两种协议。第一个协议称为回退 N 帧协议（Go-Back-N，GBN）。回退 N 帧的关键是我们在接收到确认之前，可以发送多个分组，但是接收端只能缓冲一个分组。我们保存被发送分组的副本直到确认到达。注意，很多数据分组以及确认可以同时处于信道中。

序号

如前所述，序号是模 2m 的，这里 m 是序号字段的大小，单位是比特（位）。

确认号

这个协议中的确认号是累积的，并且定义了预期接收的下一个分组序号。例如，如果确认号（ackNo）是 7，这意味着序号在 6 以内的分组都已经安全完整到达，并且接收方等待序号为 7 的分组。

在回退 N 帧协议中，确认号是累积的并且定义了预期接收的下一个分组序号。

发送窗口

发送窗口是一个想象的盒子，它覆盖了处于运送途中的以及可以被发送的数据分组序号。在每个窗口位置，某些序号定义了已经被发送的分组；其他序号定义了可以被发送的分组。窗口最大为 2m - 1。协议的窗口大小可以变化。

在任何时候，发送窗口都可能将序号分成四部分。第一部分，窗口左侧，定义了已经确认的分组的序号。发送方不需要担心这些分组并且不需要保存它们的副本。第二部分，定义了已经被发送的分组的序号，但是这些分组状态未知。发送方需要等待，从而发现这些分组究竟是已经被接收还是丢失。我们把这些分组称为未完成（outstanding）分组。第三部分，定义了可以发送的分组的序号；然而，相应数据还没有从应用层接收到。最后，第四部分，窗口右侧，定义了直到窗口滑动前都不能使用的序号。

接收窗口

接收窗口确保正确的数据分组被接收，并且确保正确的确认被发送。在回退 N 帧中，接收窗口的大小总是 1。接收方总是寻找特定分组是否到达。任何失序分组到达都会被丢弃并需要被重发。注意，我们只需要一个变量，即 Rn（接收窗口，预期接收的下一个分组），

来定义这种抽象窗口。窗口左侧的序号属于已经被接收和确认的分组；窗口右侧的序号定义了不能被接收的分组。任何序号在这两区域中的分组都被丢弃。只有序号符合 Rn 值的分组才能被接收和确认。接收窗口也滑动，但是一次只滑动一个槽。当正确的分组被接收时，窗口滑动 Rn =（Rn + 1）modulo 2m。

参考 http://book.51cto.com/art/201212/375305.htm