

这就是说, 发送方只有万分之 2.7 时间是忙的。从其他角度来看, 发送方在 30.008ms 内只能发送 1000 字节, 有效的吞吐量仅为 267kbps, 即使有 1Gbps 的链路可用! 想象一个不幸的网络经理购买了一条千兆比容量的链路, 但他仅能得到 267kbps 吞吐量的情况! 这是一个形象的网络协议限制底层网络硬件所提供的能力的示例。而且, 我们还忽略了在发送方和接收方的底层协议处理时间, 以及可能出现在发送方与接收方之间的任何中间路由器上的处理与排队时延。考虑到这些因素, 将进一步增加时延, 使其性能更糟糕。

解决这种特殊的性能问题的一个简单方法是: 不使用停等方式运行, 允许发送方发送多个分组而无需等待确认, 如在图 3-17b 图示的那样。图 3-18b 显示了如果发送方可以在等待确认之前发送 3 个报文, 其利用率也基本上提高 3 倍。因为许多从发送方向接收方输送的分组可以被看成是填充到一条流水线中, 故这种技术被称为**流水线** (pipelining)。流水线技术对可靠数据传输协议可带来如下影响:

- 必须增加序号范围, 因为每个输送中的分组 (不计算重传的) 必须有一个唯一的序号, 而且也许有多个在输送中未确认的报文。
- 协议的发送方和接收方两端也许必须缓存多个分组。发送方最低限度应当能缓冲那些已发送但没有确认的分组。如下面讨论的那样, 接收方或许也需要缓存那些已正确接收的分组。
- 所需序号范围和对缓冲的要求取决于数据传输协议如何处理丢失、损坏及延时过大的分组。解决流水线的差错恢复有两种基本方法是: **回退 N 步** (Go-Back- N , GBN) 和**选择重传** (Selective Repeat, SR)。

3.4.3 回退 N 步

在回退 N 步 (GBN) 协议中, 允许发送方发送多个分组 (当有多个分组可用时) 而不需等待确认, 但它也受限于在流水线中未确认的分组数不能超过某个最大允许数 N 。在本节中我们较为详细地描述 GBN。但在继续阅读之前, 建议你操作本书配套 Web 网站上的 GBN Java 小程序 (这是一个非常好的 Java 程序)。

图 3-19 显示了发送方看到的 GBN 协议的序号范围。如果我们将基序号 (base) 定义为最早的未确认分组的序号, 将下一个序号 (nextseqnum) 定义为最小的未使用序号 (即下一个待发分组的序号), 则可将序号范围分割成 4 段。在 $[0, \text{base} - 1]$ 段内的序号对应于已经发送并被确认的分组。 $[\text{base}, \text{nextseqnum} - 1]$ 段内对应已经发送但未被确认的分组。 $[\text{nextseqnum}, \text{base} + N - 1]$ 段内的序号能用于那些要被立即发送的分组, 如果有数据来自上层的话。最后, 大于或等于 $\text{base} + N$ 的序号是不能使用的, 直到当前流水线中未被确认的分组 (特别是序号为 base 的分组) 已得到确认为止。

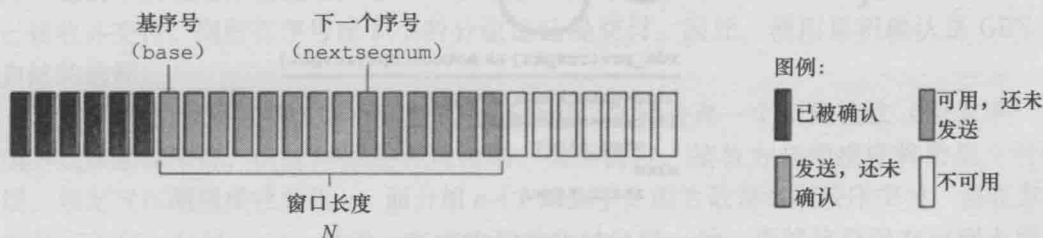


图 3-19 在 GBN 中发送方看到的序号