



# [12] 发明专利说明书

专利号 ZL 200510089995.4

[45] 授权公告日 2009年2月18日

[11] 授权公告号 CN 100463445C

[22] 申请日 2005.8.9

[21] 申请号 200510089995.4

[73] 专利权人 大唐移动通信设备有限公司

地址 100083 北京市海淀区学院路29号

[72] 发明人 王鑫 周杰

[56] 参考文献

WO 2005/006704A1 2005.1.20

CN1311590A 2001.9.5

EP1271886A2 2003.1.2

WO01/28180A2 2001.4.19

Compressing IP/UDP/RTP Headers for Low-Speed Serial Links. S. Casner, V. Jacobson. Network Working Group Request for Comments:2508. 1999

IP Header Compression. M. Degermark, B. Nordgren, S. Pink. Network Working Group Request for Comments:2507. 1999

审查员 常交法

[74] 专利代理机构 北京集佳知识产权代理有限公司

代理人 胡晶 逯长明

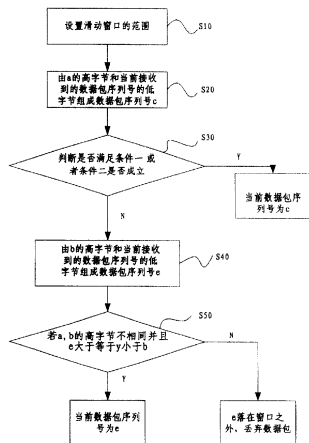
权利要求书4页 说明书12页 附图4页

## [54] 发明名称

数据包序列号计算方法及数据包传输方法

## [57] 摘要

一种数据包序列号计算方法，用于接收端获得全头数据包的数据包序列号，包括：(1) 设置滑动窗口的范围 [a, b)，滑动窗口的起点 a 为期望数据包序列号，由预设滑动窗口长度 L 确定滑动窗口的终点 b；(2) 由 a 的高字节和当前接收到的数据包序列号的低字节组成数据包序列号 c；(3) 判断是否满足 a 的高字节和 b 的高字节相同且 c 落在窗口之内或者 a、b 的高字节不相同且 c 落在 [a, x] 范围内的其中条件之一，若是，当前数据包序列号为 c，否则，进行步骤(4)；(4) 由 b 的高字节和当前接收到的数据包序列号的低字节组成数据包序列号 e；(5) 若 a、b 的高字节不相同且 e 落在 [y, b) 范围内，则当前数据包序列号为 e，所述 x 为 a 的高字节和低字节为 0xFF 组成，所述 y 为 b 的高字节和低字节为 0x00 组成，通过上述方法可以正确计算全头数据包的序列号。



1、一种数据包序列号计算方法，用于接收端获得全头数据包的数据包序列号，其特征在于，包括：

(1)设置滑动窗口的范围 $[a,b)$ ，所述滑动窗口的起点  $a$  为期望数据包序列号，由预设滑动窗口长度  $L$  确定滑动窗口的终点  $b$ ；

(2)由  $a$  的高字节和当前接收到的数据包序列号的低字节组成数据包序列号  $c$ ；

(3)判断是否满足条件一： $a$ 的高字节和 $b$ 的高字节相同且 $c$ 落在窗口之内，或者满足条件二： $a,b$ 的高字节不相同且 $c$ 落在 $[a,x]$ 范围内，若是，当前数据包序列号为 $c$ ，否则，进行步骤(4)；

(4)由  $b$  的高字节和当前接收到的数据包序列号的低字节组成数据包序列号  $e$ ；

(5)若  $a,b$  的高字节不相同且  $e$  落在  $[y,b)$  范围内，则当前数据包序列号为  $e$ ，所述  $x$  为  $a$  的高字节和低字节为  $0xFF$  组成，所述  $y$  为  $b$  的高字节和低字节为  $0x00$  组成。

2、如权利要求 1 所述的数据包序列号计算方法，其特征在于，步骤(1)中，若  $a+L-1 \geq 65535$ ，则  $b = (a+L+1) \& 0xFFFF$ ，否则  $b = a+L$ 。

3、如权利要求 1 或 2 所述的数据包序列号计算方法，其特征在于，步骤(3)中判断条件一或者条件二是否满足具体为：

判断  $(c-a) \& 0xffff > (c-b) \& 0xffff$  是否成立，若是，则条件一或者条件二成立，否则，条件一不成立且条件二也不成立。

4、如权利要求 1 或 2 所述的数据包序列号计算方法，其特征在于，步骤(3)判断是否满足条件一或条件二具体为：

a1: 若  $a \leq c < b$  且  $(a \& 0xff00) = (b \& 0xff00)$  成立时，条件一成立，否则进行步骤 a2；

a2:当  $a \leq c < x$  成立时, 则条件二成立。

5、如权利要求1所述的数据包序列号计算方法, 其特征在于, 所述L的取值范围为  $0 < L < 128$ 。

6、一种数据包序列号计算方法, 用于接收端获得全头数据包的数据包序列号, 其特征在于, 包括:

(1)预设滑动窗口长度L, L的取值范围为  $0 < L < 128$ ;

(2)若  $a + L - 1 \leq 65535$ , 则进行步骤(3), 否则进行步骤(6);

(3)计算D, D为a的低字节减去当前接收数据包序列号的低字节;

(4)若  $D > (256 - L)$ , 则低字节发生进位, 所述当前数据包序列号是由(a的高字节+1)得到的高字节和当前接收到的数据包序列号的低字节组成, 否则, 进行步骤(5);

(5)若  $D < (L - 256)$ , 则高字节发生退位, 所述当前数据包序列号是由(a的高字节-1)得到的高字节和当前接收到的数据包序列号的低字节组成, 否则, 当前数据包序列号是由a的高字节和当前接收到的数据包序列号的低字节组成;

(6)若当前接收数据包的序列号的低字节大于等于a的低字节, 则当前数据包序列号是由a的高字节和当前接收到的数据包序列号的低字节组成, 否则, 进行步骤(7);

(7)若当前接收数据包的序列号的低字节小于等于  $(a + L - 65535)$  的低字节, 则当前数据包序列号是高字节为0和当前接收到的数据包序列号的低字节组成, 其中, a为期望数据包序列号。

7、如权利要求6所述的数据包序列号计算方法, 其特征在于, 还包括: 当计算的当前数据包序列号不在滑动窗口内时, 丢弃所述数据包, 所述当前数据包序列号是否在滑动窗口是通过以下条件判断的: 满足  $(k = \text{ReceiveSn} - a) < L$  且  $\text{ReceiveSn} \geq a$  或者  $(k = \text{ReceiveSn} + 65535 - a) < L$  且  $\text{ReceiveSn} < a$  时, 所述数据包序列号落在滑动窗口内, 否则所述数据包序列号落在滑动窗口之

外,其中 ReceiveSn 为收到数据包的序列号, a 为期望接收到数据包的序列号, L 为滑动窗口的长度。

8、一种数据包传输方法, 其特征在于, 包括:

(1)接收端接收数据包, 判断所述数据包为压缩数据包还是全头数据包, 若是压缩数据包, 则提取出接收数据包序列号, 若是全头数据包, 则按照下述方法计算接收数据包序列号:

(b1)设置滑动窗口的范围 $[a,b)$ , 所述滑动窗口的起点 a 为期望数据包序列号, 由预设滑动窗口长度 L 确定滑动窗口的终点 b;

(b2)由 a 的高字节和当前接收到的数据包序列号的低字节组成数据包序列号 c;

(b3)判断是否满足条件一: a 的高字节和 b 的高字节相同且 c 落在窗口之内, 或者满足条件二: a,b 的高字节不相同且 c 落在 $[a,x]$ 范围内, 若是, 当前数据包序列号为 c, 否则, 进行步骤(b4);

(b4) 由 b 的高字节和当前接收到的数据包序列号的低字节组成数据包序列号 e;

(b5)若 a,b 的高字节不相同且 e 落在 $[y,b)$ 范围内, 则当前数据包序列号为 e, 所述 x 为 a 的高字节和低字节为 0xFF 组成, 所述 y 为 b 的高字节和低字节为 0x00 组成;

(2)接收端按照数据包序列号来解压缩接收到的数据包, 从而得到正确的数据流。

9、如权利要求 8 所述的一种数据包传输方法, 其特征在于, 步骤(1)还包括: 若当前接收到的数据包是期望接收到的数据包, 则期望数据包序列号加 1, 否则, 对所述数据包进行缓存, 并继续接收期望接收到的数据包。

10、如权利要求 8 或 9 所述的一种数据包传输方法, 其特征在于, 若期望接收到的数据包在预设时间内未接收到, 则将缓存中已经收到其后连续的数据包一并处理, 并修改期望数据包序列号。

11、一种数据包传输方法，其特征在于，包括：

(1)接收端接收数据包，判断所述数据包为压缩数据包还是全头数据包，若是压缩数据包，则提取出接收数据包序列号，若是全头数据包，则按照下述方法计算接收数据包序列号：

(c1)预设滑动窗口长度  $L$ ， $L$  的取值范围为  $0 < L < 128$ ；

(c2)若  $a + L - 1 \leq 65535$ ，则进行步骤(c3)，否则进行步骤(c6)；

(c3)计算  $D$ ， $D$  为  $a$  的低字节减去当前接收数据包序列号的低字节；

(c4)若  $D > (256 - L)$ ，则低字节发生进位，所述当前数据包序列号是由 ( $a$  的高字节+1)得到的高字节和当前接收到的数据包序列号的低字节组成，否则，进行步骤(c5)；

(c5)若  $D < (L - 256)$ ，则高字节发生退位，所述当前数据包序列号是由 ( $a$  的高字节 - 1)得到的高字节和当前接收到的数据包序列号的低字节组成，否则，当前数据包序列号是由  $a$  的高字节和当前接收到的数据包序列号的低字节组成；

(c6) 若当前接收数据包的序列号的低字节大于等于  $a$  的低字节，则当前数据包序列号是由  $a$  的高字节和当前接收到的数据包序列号的低字节组成，否则，进行步骤(c7)；

(c7)若当前接收数据包的序列号的低字节小于等于  $(a + L - 65535)$  的低字节，则当前数据包序列号是高字节为 0 和当前接收到的数据包序列号的低字节组成，其中， $a$  为期望数据包序列号；

(2)接收端按照数据包序列号来解压缩接收到的数据包，从而得到正确的数据流。

## 数据包序列号计算方法及数据包传输方法

### 技术领域

本发明涉及通信领域的数据传输技术，尤其涉及移动通信中分组数据汇聚层数据传输控制技术，特别是数据传输过程中的数据包序列号提取方法及相应的PDCP（packet data convergence protocol, 分组数据协议汇聚协议）层乱序情况下数据包传输方法。

### 背景技术

目前，对TCP/IP数据包的处理主要是以RFC2507协议为主，由于RFC2507协议采用增量压缩技术，因此接收端在解压缩时对压缩数据包的连续性要求很严格。但是由于实际的通讯环境中通常无法保证良好的无线传输条件，因此经常发生接收端接收到的数据包乱序的情况，进而影响后续的解压缩过程，从而导致整个数据通信过程的失败。

为此，在RFC2507协议中引入了压缩数据包重排序的机制。发送端在每个需要发送的数据包中增加序列号，接收端通过数据包中携带的序列来获得压缩数据包传输顺序的信息，由此保证接收端能够正确地解压缩数据包。

在RFC2507协议中，发送端经过处理的TCP/IP报流一般分为COAPRESS\_TCP（压缩数据包）和FULL\_HEAD（全头数据包）两种数据包类型。对于COAPRESS\_TCP数据包，发送端通常用两个字节来存储数据包序列号。初始值为1，范围为1-65535。当序列号增加到65535时发生环回重新从1开始累计。而对于FULL\_HEAD数据包，目前协议规定在支持重排序的条件下只有一个字节用来存储数据包序列号，考虑到数据发送的关联性选取序列号的低字节。比如，接收端期望接收到的下一个数据包序列号为0x0012，而接

收端当前接收到的数据包序列号的低字节为0x12, 则计算当前数据包序列号: 期望数据包序列号的高字节和当前接收到的数据包序列号的低字节组成当前数据包序列号0x0012, 并且设置期望接收到的下一个数据包序列号为0x0013。

但是, 接收端进行数据包序列号计算过程中, 需要考虑以下几个问题:

### 1、单字节的进位与退位情况

由于全头数据包只有一个字节用来存放数据包的序列号, 在接收端用全头数据包中传输的序列号作为低字节与期望数据包序列号的高字节进行组合作为当前接收数据包的序列号时就需要考虑进位和退位的情况。例如: 当前期望的数据包序列号是 0x0100, 收到全头数据包中携带的数据包序列号低字节为 0xFF (该数据包完整的序列号为 0x00FF), 在不考虑高字节退位的情况下组合生成的序列号为 0x01FF, 这种情况就会出现序列号的计算错误。另一种情况, 当前期望的数据包序列号为 0x00FF, 收到全头数据包中携带的数据包序列号低字节为 0x01 (该包数据完整的序列号为 0x0101), 不考虑低字节进位的情况下生成的序列号 0x0001 同样是错误的。

也就是说, 在实际复杂的通讯环境下容易发生数据包乱序的情况, 而现有的数据包序列号计算方法容易发生序列号计算出错的问题。

### 2、高字节的环回情况

不管是全头数据包还是压缩数据包, 数据包序列号都占用两个字节, 当计数器累计超过了65535的时候会发生数据包序列号的环回情况。即, 该计数器重新从1开始累计。而现有的数据包序列号计算方法不能保证能够正确获得高字节环回的数据包序列号。比如, 接收端期望接收到的全头数据包的当前数据包序列号为65534 (0xFFFFE), 而由于外界无线传输条件的影响导致后序的两个数据包传输丢失, 则接收到的是数据包序列号为0002(0x0002)的数据包。由于全头数据包只能传输低字节0x02, 则通过现有的数据包序列号的计

算方法，得到的数据包序列号为0xFF02。很显然，这种数据包序列号计算方法得到的数据包序列号是错误的。

总之，现有的全头数据包的序列号计算方法，是将期望数据包序列号的高字节和接收到的数据包序列号的低字节组成当前接收到的全头数据包的序列号，而实际的通讯环境较为复杂，容易在数据传输过程中发生数据包乱序甚至丢包的情况，由此容易造成按照现有的全头数据包的序列号计算方法获得的全头数据包序列号错误情况发生，进而造成接收端不能正确解压缩数据包的后果。

## 发明内容

本发明的目的在于提供一种数据包序列号计算方法及数据包传输方法，以解决现有计算方法未考虑到数据传输过程中发生数据包乱序甚至丢包的情况，由此发生进位和退位及高字节的环回等情况，进而造成按照现有的全头数据包的序列号计算方法获得的全头数据包序列号错误的技术问题。

为了解决上述问题，本发明公开了一种数据包序列号计算方法，用于接收端获得全头数据包的数据包序列号，包括：

(1)设置滑动窗口的范围 $[a,b)$ ，所述滑动窗口的起点  $a$  为期望数据包序列号，由预设滑动窗口长度  $L$  确定滑动窗口的终点  $b$ ；

(2)由  $a$  的高字节和当前接收到的数据包序列号的低字节组成数据包序列号  $c$ ；

(3)判断是否满足条件一： $a$ 的高字节和 $b$ 的高字节相同且 $c$ 落在窗口之内，或者满足条件二： $a,b$ 的高字节不相同且 $c$ 落在 $[a,x]$ 范围内，若是，当前数据包序列号为 $c$ ，否则，进行步骤(4)；

(4)由  $b$  的高字节和当前接收到的数据包序列号的低字节组成数据包序列号  $e$ ；

(5)若  $a,b$  的高字节不相同且  $e$  落在 $[y,b)$ 范围内，则当前数据包序列号为  $e$ ，



所述  $x$  为  $a$  的高字节和低字节为  $0xFF$  组成, 所述  $y$  为  $b$  的高字节和低字节为  $0x00$  组成。若  $a+L-1 \geq 65535$ , 则  $b = (a+L+1) \& 0xFFFF$ , 否则  $b = a+L$ 。

步骤(3)中判断条件一或者条件二是否满足具体为: 判断  $(c-a) \& 0xffff > (c-b) \& 0xffff$  是否成立, 若是, 则条件一或者条件二成立, 否则, 条件一不成立且条件二也不成立。

步骤(3)判断是否满足条件一或条件二具体为: a1: 若  $a \leq c < b$  且  $(a \& 0xff00) = (b \& 0xff00)$  成立时, 条件一成立, 否则进行步骤 a2; a2: 当  $a \leq c < x$  成立时, 则条件二成立。

所述  $L$  的取值范围为  $0 < L < 128$ 。

本发明公开了另一种数据包序列号计算方法, 用于接收端获得全头数据包的数据包序列号, 包括:

(1) 预设滑动窗口长度  $L$

(2) 若  $a+L-1 \leq 65535$ , 则进行步骤(3), 否则进行步骤(6);

(3) 计算  $D$ ,  $D$  为  $a$  的低字节减去当前接收数据包序列号的低字节;

(4) 若  $D > (256 - L)$ , 则低字节发生进位, 所述当前数据包序列号是由  $(a$  的高字节+1) 得到的高字节和当前接收到的数据包序列号的低字节组成, 否则, 进行步骤(5);

(5) 若  $D < (L - 256)$ , 则高字节发生退位, 所述当前数据包序列号是由  $(a$  的高字节-1) 得到的高字节和当前接收到的数据包序列号的低字节组成, 否则, 当前数据包序列号是由  $a$  的高字节和当前接收到的数据包序列号的低字节组成;

(6) 若当前接收数据包的序列号的低字节大于等于  $a$  的低字节, 则当前数据包序列号是由  $a$  的高字节和当前接收到的数据包序列号的低字节组成, 否则, 进行步骤(7);

(7) 若当前接收数据包的序列号的低字节小于等于  $(a+L-65535)$  的低字节, 则当前数据包序列号是高字节为  $0$  和当前接收到的数据包序列号的低字

节组成，其中， $a$ 为期望数据包序列号。

当计算的当前数据包序列号不在滑动窗口内时，丢弃所述数据包，所述当前数据包序列号是否在滑动窗口是通过以下条件判断的：满足  $(k=ReceiveSn-a) < L$  且  $ReceiveSn \geq a$  或者  $(k=ReceiveSn+65535-a) < L$  且  $ReceiveSn < a$  时，所述数据包序列号落在滑动窗口内，否则所述数据包序列号落在滑动窗口之外，其中  $ReceiveSn$  为收到数据包的序列号， $a$  为期望接收到数据包的序列号， $L$  为滑动窗口的长度。

一种数据包传输方法，包括：

(1)接收端接收数据包，判断所述数据包为压缩数据包还是全头数据包，若是压缩数据包，则提取出接收数据包序列号，若是全头数据包，则按照下述方法计算接收数据包序列号：

(b1)设置滑动窗口的范围 $[a,b)$ ，所述滑动窗口的起点  $a$  为期望数据包序列号，由预设滑动窗口长度  $L$  确定滑动窗口的终点  $b$ ；

(b2)由  $a$  的高字节和当前接收到的数据包序列号的低字节组成数据包序列号  $c$ ；

(b3)判断是否满足条件一： $a$  的高字节和  $b$  的高字节相同且  $c$  落在窗口之内，或者满足条件二： $a,b$  的高字节不相同且  $c$  落在 $[a,x]$ 范围内，若是，当前数据包序列号为  $c$ ，否则，进行步骤(b4)；

(b4) 由  $b$  的高字节和当前接收到的数据包序列号的低字节组成数据包序列号  $e$ ；

(b5)若  $a,b$  的高字节不相同且  $e$  落在 $[y,b)$ 范围内，则当前数据包序列号为  $e$ ，所述  $x$  为  $a$  的高字节和低字节为  $0xFF$  组成，所述  $y$  为  $b$  的高字节和低字节为  $0x00$  组成；

(2)接收端按照数据包序列号来解压缩接收到的数据包，从而得到正确的数据流。

步骤(1)还包括：若当前接收到的数据包是期望接收到的数据包，则期望

数据包序列号加 1，否则，对所述数据包进行缓存，并继续接收期望接收到的数据包。若期望接收到的数据包在预设时间内未接收到，则将缓存中已经收到其后连续的数据包一并处理，并修改期望数据包序列号。

一种数据包传输方法，包括：

(1)接收端接收数据包，判断所述数据包为压缩数据包还是全头数据包，若是压缩数据包，则提取出接收数据包序列号，若是全头数据包，则按照下述方法计算接收数据包序列号：

(c1)预设滑动窗口长度  $L$

(c2)若  $a+L-1 \leq 65535$ ，则进行步骤(c3)，否则进行步骤(c6)；

(c3)计算  $D$ ， $D$  为  $a$  的低字节减去当前接收数据包序列号的低字节；

(c4)若  $D > (256 - L)$ ，则低字节发生进位，所述当前数据包序列号是由 ( $a$  的高字节+1)得到的高字节和当前接收到的数据包序列号的低字节组成，否则，进行步骤(c5)；

(c5)若  $D < (L - 256)$ ，则高字节发生退位，所述当前数据包序列号是由( $a$  的高字节 - 1)得到的高字节和当前接收到的数据包序列号的低字节组成，否则，当前数据包序列号是由  $a$  的高字节和当前接收到的数据包序列号的低字节组成；

(c6) 若当前接收数据包的序列号的低字节大于等于  $a$  的低字节，则当前数据包序列号是由  $a$  的高字节和当前接收到的数据包序列号的低字节组成，否则，进行步骤(c7)；

(c7)若当前接收数据包的序列号的低字节小于等于( $a + L - 65535$ )的低字节，则当前数据包序列号是高字节为 0 和当前接收到的数据包序列号的低字节组成，其中， $a$  为期望数据包序列号；

(2)接收端按照数据包序列号来解压缩接收到的数据包，从而得到正确的数据流。

与现有技术相比,本发明具有以下优点:本发明结合滑动窗口对 TCP/IP 全头数据包序列号进行计算,充分考虑到进位与退位情况、高字节环回情况,由此保证全头数据包序列号计算的正确性,进而保证接收端能够正确解压缩数据包。

## 附图说明

图 1 是本发明的数据包序列号计算方法的一实施例流程图

图 2 是本发明的另一数据包序列号计算方法的一种实施例流程图;

图 3 是本发明的另一数据包序列号计算方法的第二种实施例流程图;

图 4 是本发明的一种数据包传输方法的流程图。

## 具体实施方式

以下结合附图,具体说明本发明。

本发明提供的一种数据包序列号计算方法,用于接收端获得全头数据包的数据包序列号。请参阅图 1,其为数据包序列号计算方法的流程图。它包括以下步骤:

S10: 设置滑动窗口的范围 $[a,b)$ ,所述滑动窗口的起点  $a$  为期望数据包序列号,由预设滑动窗口长度  $L$  确定滑动窗口的终点  $b$ 。当  $a+L-1 \geq 65535$  时,  $b = (a+L+1) \& 0xFFFF$ 。因为协议中规定数据包的序列号从 1 开始,不能为 0。当  $a+L-1 < 65535$  时,  $b = a+L$ 。

S20: 由  $a$  的高字节和当前接收到的数据包序列号的低字节组成数据包序列号  $c$ 。

S30: 判断是否满足条件一:  $a$  的高字节和  $b$  的高字节相同且  $c$  落在窗口之内,或者满足条件二:  $a,b$  的高字节不相同且  $c$  落在 $[a,x]$ 范围内,若是,当前数据包序列号为  $c$ ,否则,进行步骤 S40。

步骤 S30 中判断条件一或条件二是否满足具体为：判断  $(c-a) \& 0xffff > (c-b) \& 0xffff$  是否成立，若是，则条件一或者条件二满足。

步骤 S30 判断条件一或条件二是否满足还可以通过以下方式获得：

a1: 若  $a \leq c < b$  且  $(a \& 0xff00) = (b \& 0xff00)$  成立时，条件一成立，否则进行步骤 a2；a2: 当  $a \leq c < x$  成立时，则条件二成立。

S40: 由 b 的高字节和当前接收到的数据包序列号的低字节组成数据包序列号 e。

S50: 若 a, b 的高字节不相同且 e 落在  $[y, b)$  范围内，则当前数据包序列号为 e，所述 x 为 a 的高字节和低字节为 0xFF 组成，所述 y 为 b 的高字节和低字节为 0x00 组成。比如，判断  $(e-a) \& 0xffff > (e-b) \& 0xffff$ ，若是，则 a, b 的高字节不相同且 e 落在  $[y, b)$  范围内，e 即为所求的当前数据包序列号，否则，e 落在窗口之外，丢弃所述数据包。

通过上述的方法，即便数据传输过程中发生数据包乱序甚至丢包的情况，也能正确获得当前数据包的序列号。

请参阅图 2，其为本发明的数据包序列号计算方法的流程图。它主要用于获得全头数据包序列号，它包括以下步骤：

S110: 预设滑动窗口长度 L，考虑到全头数据包序列号采用单字节传输，且为了保证序列号提取的正确性，滑动窗口的长度通常为  $0 < L < 128$ 。在接收端接收数据过程中，可以根据具体情况，调节滑动窗口长度 L 的大小。滑动窗口长度 L 越长，进行数据包序列号计算时所占用的资源越多；

S120: 若  $(65535 - a) \geq (L-1)$ ，则进行步骤 S130，否则进行步骤 S160。当  $(65535 - a) \geq (L-1)$  成立，而说明当前接收的数据包序列号没有发生高字节环回，否则当前接收到的数据包序列号有可能发生高字节环回。

S130: 计算 D，D 为 a 的低字节减去当前接收数据包序列号的低字节。若期望数据包序列号的低字节和当前接收到的数据包序列号的低字节相同，

则说明当前接收到的数据包序列号即为期望数据包的序列号。当期望数据包序列号的低字节和当前接收到的数据包序列号的低字节不相同，则当前接收到的数据包不是当前期望接收到的数据包，就有可能发生当前接收数据包序列号的低字节发生进位或退位问题。为此，本发明通过计算的 D 值来判断是否存在进位或退位问题。

S140: 若  $D > (256 - L)$ ，则表明接收到数据包序列号的低字节发生进位，接收数据包序列号的高字节为 a 的高字节 + 1，否则，进行步骤 S150；

S150: 若  $D < (L - 256)$ ，则表明接收到的数据包序列号的高字节发生退位，接收数据包序列号的高字节为 a 的高字节 - 1，否则，接收数据包序列号的高字节为 a 的高字节；

S160: 若当前接收数据包的序列号的低字节大于等于 a 的低字节，则接收数据包序列号的高字节为 a 的高字节，否则，进行步骤 S170；

a 的低字节可以通过以下方式获得： $a \text{ 的低字节} = a \& 0x00FF$ ，a 的高字节也可以通过以下方式获得： $a \text{ 的高字节} = a \& 0xFF00$ ，其中 & 为与运算。

S170: 若当前接收数据包的序列号的低字节小于等于  $(a + L - 65535)$  的低字节，则当前接收数据包序列号的高字节为 0。

同样道理， $(a + L - 65535)$  的低字节可以通过以下方式获得： $(a + L - 65535) \& 0x00FF$ 。

通过上述步骤，接收到包头数据包的数据包序列号。接收到数据包可能是期望接收到的数据包，也可能不是期望接收到的数据包。并且，还需判断当前接收数据包序列号是否落在滑动窗口内，若是，将所述数据包序列号的高字节和接收得到的数据包序列号的低字节组成当前数据包序列号，否则，丢弃所述数据包（请参阅图 3）。满足  $(k = \text{ReceiveSn} - a) < L$  且  $\text{ReceiveSn} \geq a$  或者满足  $(k = \text{ReceiveSn} + 65535 - a) < L$  且  $\text{ReceiveSn} < a$ ，则所述数据包序列号落在滑动窗口内，其中 ReceiveSn 为收到数据包的序列号，a 为期望接收到数据

包的序列号，L为滑动窗口的长度。

请参阅图4，其为本发明的一种数据包传输方法的流程图。它包括：

S210: 接收端接收数据包，得到数据包的序列号

接收端判断接收到的数据包为压缩数据包还是全头数据包，若是压缩数据包，则提取出接收数据包序列号，若是全头数据包，则按照下述方法计算接收数据包序列号：

(b1)设置滑动窗口的范围 $[a,b)$ ，所述滑动窗口的起点a为期望数据包序列号，由预设滑动窗口长度L确定滑动窗口的终点b；

(b2)由a的高字节和当前接收到的数据包序列号的低字节组成数据包序列号c；

(b3)判断是否满足条件一：a的高字节和b的高字节相同且c落在窗口之内，或者满足条件二：a,b的高字节不相同且c落在 $[a,x]$ 范围内，若是，当前数据包序列号为c，否则，进行步骤(b4)；

(b4)由b的高字节和当前接收到的数据包序列号的低字节组成数据包序列号e；

(b5)若a,b的高字节不相同且e落在 $[y,b)$ 范围内，则当前数据包序列号为e，所述x为a的高字节和低字节为0xFF组成，所述y为b的高字节和低字节为0x00组成；

S220:接收端按照数据包序列号来解压缩接收到的数据包，从而得到正确的数据流。

另外，全头数据包还可以按照下述方法计算接收数据包序列号：

(c1)预设滑动窗口长度L

(c2)若 $a+L-1 \leq 65535$ ，则进行步骤(c3)，否则进行步骤(c6)；

(c3)计算D，D为a的低字节减去当前接收数据包序列号的低字节；

(c4)若  $D > (256 - L)$ , 则低字节发生进位, 所述当前数据包序列号是由(a 的高字节+1)得到的高字节和当前接收到的数据包序列号的低字节组成, 否则, 进行步骤(c5);

(c5)若  $D < (L - 256)$ , 则高字节发生退位, 所述当前数据包序列号是由(a 的高字节 - 1)得到的高字节和当前接收到的数据包序列号的低字节组成, 否则, 当前数据包序列号是由 a 的高字节和当前接收到的数据包序列号的低字节组成;

(c6) 若当前接收数据包的序列号的低字节大于等于 a 的低字节, 则当前数据包序列号是由 a 的高字节和当前接收到的数据包序列号的低字节组成, 否则, 进行步骤(c7);

(c7)若当前接收数据包的序列号的低字节小于等于(a + L - 65535)的低字节, 则当前数据包序列号是高字节为 0 和当前接收到的数据包序列号的低字节组成, 其中, a 为期望数据包序列号。

发送端发送数据包时, 在每一个数据包中增加一数据包序列号。接收端接收数据包时, 只有从数据包中获得序列号, 才能得到数据包传输顺序信息, 进而完成数据包的解压缩。

若当前接收到的数据包不是期望接收到的数据包, 则对所述数据包进行缓存, 并继续接收期望接收到的数据包。当前接收到的数据包序列号等于期望数据包序列号, 则当前接收到的数据包是期望接收到的数据包, 否则当前接收到的数据包不是期望接收到的数据包。

若期望接收到的数据包在预设时间内未接收到, 则查找其后缓存中已经收到的连续数据包一并处理, 并修改期望数据包序列号。比如, 需要接收数据包为 A2, 则数据包 A2 未在预设时间内收到, 则查找缓存器中是否有后续的数据包 A3、A4、A5, 若有, 则处理所述数据包 A3、A4、A5, 并将期望数据包序列号为 A5 的数据包序列号加 1。



若当前接收到的数据包是期望接收到的数据包，则将期望数据包序列号加 1，当所述期望数据包序列号大于 65535，则期望数据包序列号为 1。

若当前接收到的数据包为重传数据包，则直接丢弃所述数据包。

以上公开的仅为本发明的几个具体实施例，但本发明并非局限于此，任何本领域的技术人员能思之的变化都应落在本发明的保护范围内。

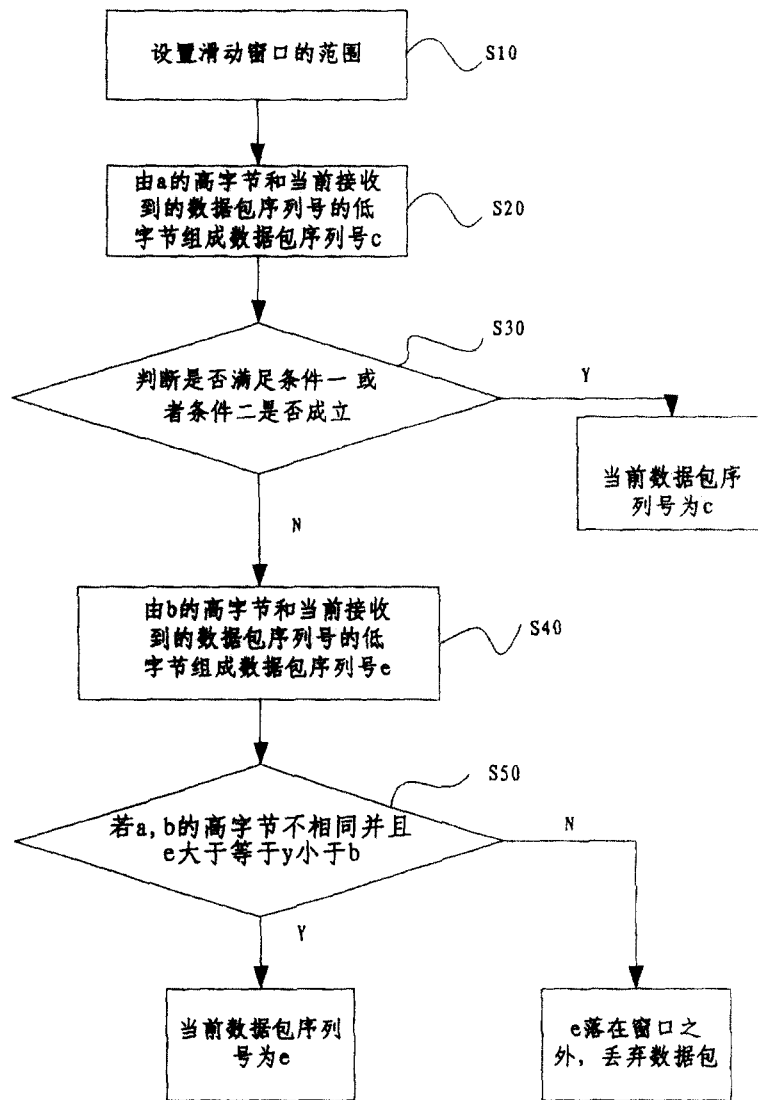


图 1

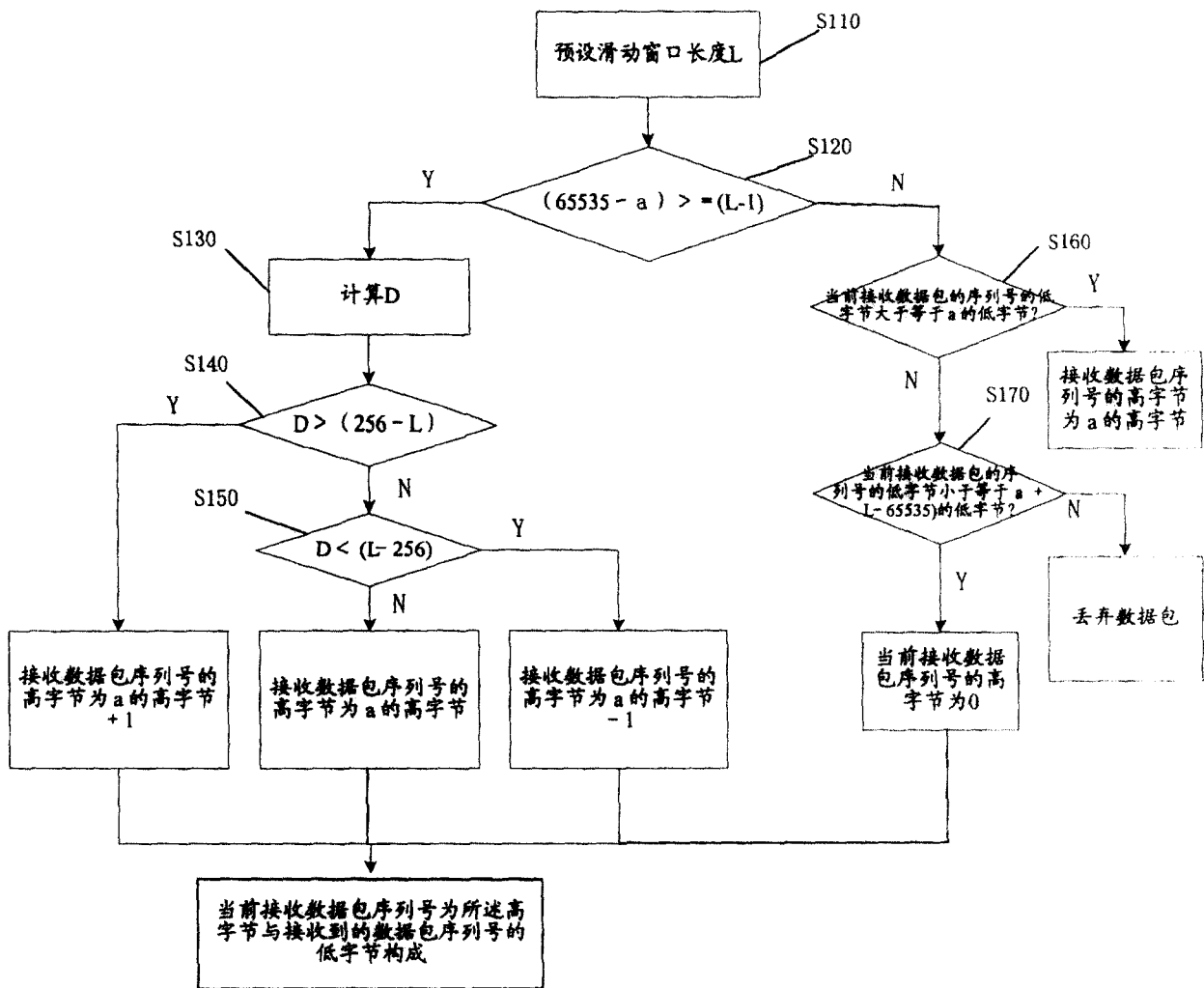


图 2

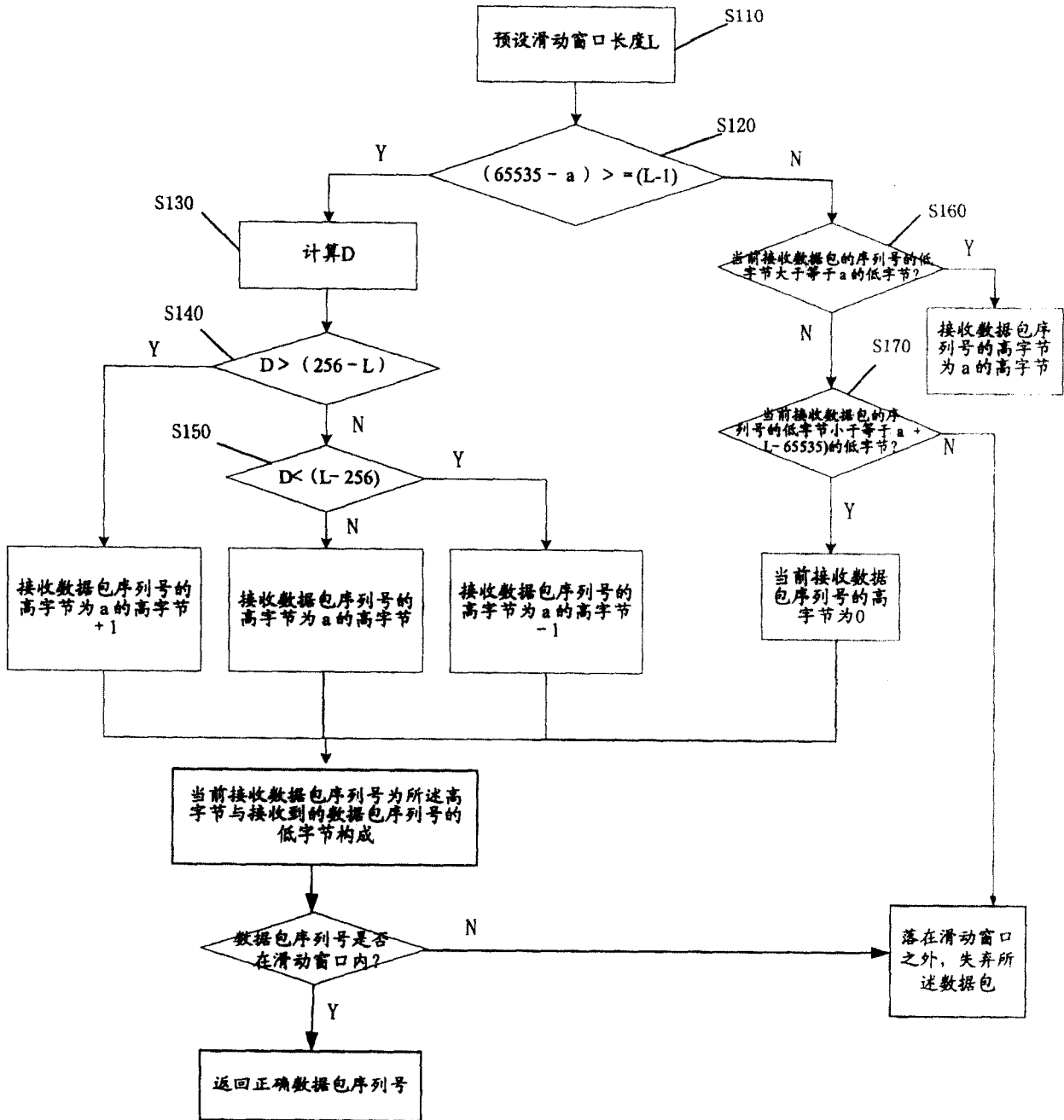


图 3

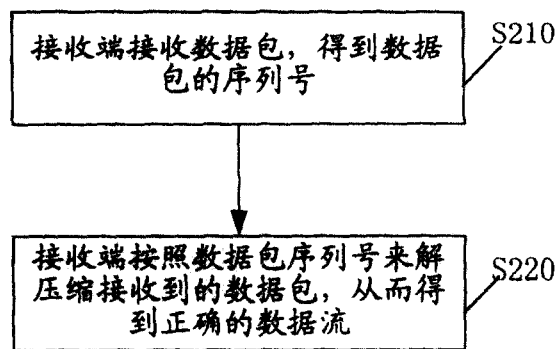


图 4