



[12] 发明专利说明书

[21] ZL 专利号 94118823.X

[45] 授权公告日 2004 年 11 月 3 日

[11] 授权公告号 CN 1174641C

[22] 申请日 1994. 11. 24 [21] 申请号 94118823. X

[30] 优先权

[32] 1993. 11. 30 [33] US [31] 159871

[71] 专利权人 美国电报电话公司

地址 美国纽约

[72] 发明人 理查德·P·伊扎克 塞义夫·南达

审查员 罗世娜

[74] 专利代理机构 中国国际贸易促进委员会专利
商标事务所

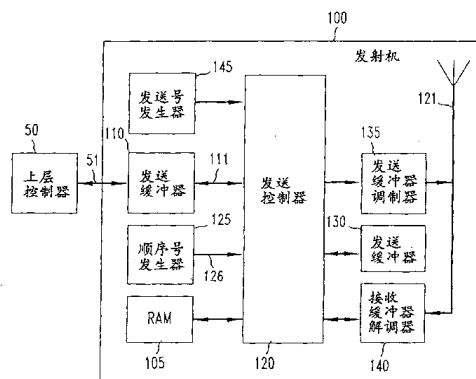
代理人 陆立英

权利要求书 2 页 说明书 22 页 附图 9 页

[54] 发明名称 无线通信的重发协议

[57] 摘要

设置一个用以发送数据包的发射机(100)来提高一个无线电传输链路的信息通过量,它响应发送窗口关闭,重复地重发所有未确认的数据包,一直这样作到发送窗口重新打开时为止,或直到接收到一个重发数据包的收信机状态消息确认时为止。



1. 一种在通信系统中应用的数据包重发设备，它包括有：
调制器，用于向一个接收机发送数据包；及

控制器，用于（a）将所述发送的数据包与其被发送时各自相应的发送次序号相关联；（b）响应接收到的用以识别所述接收机正确地和不正确地接收所述发送数据包的状态消息，当发送数据包的窗口打开时，重发仅是那些所述不正确接收的数据包，该数据包的相关发送次序号处在从所述状态消息中识别出来是正确地接收数据包中最后一个数据包的那个发送的数据包有关的发送次序号之前，

其特征在于：

每当发送数据包的窗口关闭时，依次重发所述接收机在该窗口打开时不正确地接收到的或根本没有接收到的所有未确认的数据包，并重复重发每个这样的数据包，直至所述窗口打开以发送数据包时为止或接收到一个指明该数据包被正确地接收到了的状态消息时为止。

2. 根据权利要求1所述的设备，其特征在于，所述状态消息是一个全状态消息，它包含至少一个比特映像字段、一个用以识别正确地接收到最后的数据包的字段、以及一个用以识别按正当顺序正确接收最后数据包的字段，所述比特映像字段是由与各自相应的顺序号相关联的许多比特组成的，所述的顺序号是按照所述数据包被发送时的次序添加到所述数据包中相应

的数据包上的。

3. 根据权利要求1所述的设备，其特征在于，所述的状态消息是一个局部状态消息，它插在用户信息包中，所述局部状态消息包括至少：(a) 一个比特映像字段，它是由与一组各自相应的顺序号相关联的一组比特形成的，所述的各自相应的顺序号是按照所述数据包组被发送时的次序指配到一组数据包的各自相应的数据包上的；(b) 一个组号，用以确定所述顺序号组与另一顺序号组之间的关系。

4. 根据权利要求3所述的设备，其特征在于，所述的局部状态消息还包括一个用以识别正确地接收到并在正当顺序中的最后一个数据包的字段。

5. 根据权利要求1所述的设备，其特征在于，所述的通信系统是一个无线电话系统。

无线通信的重发协议

本发明涉及无线通信。

众所周知，无线传输是有干扰的，并且通常遭受到可持续几百毫秒的强的衰落。在这样的情况下在不利的传输条件期间通过无线传输链路发送的数据包还需要重发一次或多次。可以理解，数据包频繁地重发会降低无线传输链路的信息通过量。还能理解，每当发送窗口关闭时停止重发数据包也会降低信息通过量。

另外，在反向模式下（从主接收机到用户发射机），从主接收机发送到用户发射机的信息经常由一个状态包预先占用，它表明哪些包由该接收机正确地和/或不正确地接收到。这时，可用于信息传输的反向传输通路的带宽因传输状态包而减小了。在某些应用场合中，这种带宽的减少是能接受的。但是，在要求高等级的持续双工通过量时，这种减少将是不容许的。

图1和图2分别概括性地示出可实践本发明原理的发射机和接收机的方框图；

图3示例性地示出图2所示的接收机发送给图1所示的发射机

的状态控制包的格式例子；

图 4 示例性地示出图 2 所示接收机按照本发明原理在一个待发送给图 1 所示发射机的信息包上添加一个局部状态控制消息的格式例子；

图 5 和图 6 示例性地示出按照本发明的原理由图 1 的发射机保留的一个发送表的例子；

图 7 示出局部状态控制消息的几个例子；

图 8 和图 9 示出图 2 所示的接收机实现本发明原理的程序流程图；

图 10 和图 11 示出图 1 所示的发射机实现本发明原理的程序流程图。

在图 1 所示的本发明的一个示例性实施例中，发射机 100 可构成一个便携式无线终端的发射机部分，设在一个无线通信装置内。其中，该无线终端可以自适应地按照一个预定的协议来工作。这个协议可以规定一个所谓的“上层控制器”50，它通过总线 51 连接到发送缓冲器 110 上，该缓冲器 110 可以是一个常规的存储器件，例如是一个先进先出 (FIFO) RAM 存储器。一个数据产生处理器起上层控制器 50 的作用，其内存放由缓冲器 110 的按顺序的存储器单元中的信息字形成的数据包。在一个数据包已经存入缓冲器 110 之后，它就可由控制器 120 通过总线 111 进行检索来得到。也就是象下面将要讨论的，为了把这些数据包发送给接收机 (图 2)，控制器 120 按

照这些数据包已被存储入缓冲器 110 的次序转储这些数据包。

上述的协议还可以规定逻辑链路控制层，用以限定将一个无线终端通过一条无线通信通路与一个主终端接口的许多功能，达到交换数据包的目的。在接收机 200 处，这样的功能包括，特别是，检查与已接收数据包伴随的序号，差错校验，产生状态控制消息，以及将这些消息发送给发射机 100。按照本发明的一个方面，为下面将讨论的，这样的状态消息可以是局部的也可以是全部的状态消息。在发射机 100 处，这样的功能包括，特别是，产生序号，将序号数添加到从缓冲器 110 转储的有关数据包上，将这些数据包发送给接收机 200，以及重发接收机 200 未正确接收到的那些数据包。

(请注意，这里所用的“添加”一词和该词的派生词意指加上一个表，使人们始终监视发送数包的次序。还请注意，下面有关发射机 100 (图 2 的接收机 200) 工作的讨论在发射机与接收机以全双工模式工作时同样地涉及到接收机 200 (图 1 的发射机 100)。也就是说，发射机 100 包括一个接收机 200，或接收机 200 包括一个发射机 100。)

特别是利用控制器 120、RAM 缓冲器 105、序号发生器 125 和发送号发生器 145，可在发射机 100 中实现逻辑链路控制层功能。更具体地说，序号发生器 125 可以是一个常规的模 M 计数器，其中，M 的值例如可以是 128。控制器 120 通过总线 126 接收序号发生器 125 产生的当前值，并将此值作为包的序号加到由控制器 120

从发送缓冲器 110 转储的最新版数据包上。此外，控制器 120 产生出一种差错校验码，可使这样得出数据包的接收机（例如，接收机 200）判定在无差错的情况下数据包是否被正确地接收。差错校验码可以是通常称作“循环冗余码（CRC）”的校验码。

在控制器 120 组成一个数据包之后，它可将该数据包装入发送缓冲器调制器 135 中。然后，发送缓冲器调制器 135 按常规方式利用一个由主接收机 200 指配的无线电频道所限定的载波信号，使该数据包对它调制。调制器 135 再通过天线 121 发送该调制产物。此外，控制器 120 将已发送的数据包的拷贝存入 RAM105 中由当前顺序号编索引的一个存储位置上。据此，控制器 120 利用与该数据包有关的顺序号作为 RAM 105 的索引号，很容易地访问存储在 RAM 105 中的一个数据包。

发射机 100 还包括有发送表缓冲器 130，该缓冲器可以由多个按顺序的 RAM 存储器单元组成的。为了便于描述并无限制性起见，上述的存储器单元的个数可以相应于发生器 125 所产生的号数的范围。按照本发明的一个方面，控制器 120 将数据包的顺序号存储入缓冲器 130，它与发生器 145 产生的一个号数结合在一起地发送出去，所述的号数表明控制器 120 发送的每个数据包（还包括控制器 120 重发的数据包）的次序。保留发送表 130 的理由将在下文详细解释。

参看图 1 和图 2，接收机 200 包括一付天线 221，用以接收已对

上述无线电频道的载波进行调制的数据包。所述的接收是由解调器 225 对该无线电频道的内容进行解调来实现的，所得的解调结果形成一个数据包，该数据包存储在一个有关的缓冲器中。然后，接收机控制器 205 将数据包从解调器 225 的缓冲器中按其存储的次序转储。控制器 205 检查每一个此类数据包是否含有任何差错。发生差错的原因通常是数据包通过空气以无线电波形式传播时遭受噪声而造成的。如果接收控制器 205 发现已接收的数据包含有差错，则控制器 205 丢弃该数据包。如果已接收的数据包无差错，则控制器 205 将该数据包通过总线 206 存入接收缓冲器 210 中由发射机 100 添加在该数据包上的顺序号为索引的存储器单元内。而后，控制器 205 再从缓冲器 210 中依序转储这些数据包，让它们通过总线 211 到达上层控制器 250。例如，如果号为 21 至 24 和 26 的包存储在缓冲器 210 中，则控制器 205 让包 21 至 24 传到上层控制器 250，但因包 26 在顺序以外而不传送。只有在包 25 已正确地收到并存入缓冲器 210 之后，控制器 205 才让包 26 传到上层控制器 250。

为了探踪哪些数据包已正确地、不正确地或根本没有接收到，接收控制器 205 在 RAM 215 中保存比特映象表 216，它是由与上述顺序号范围中的一些号相对应的许多比特组成的。当接收控制器 205 正确地收到一个数据包时，它便设定一个特定的二进制值，例如二进制的“1”，该比特在比特映象表中的位置对应于该数据包中所含的顺序号。例如，如果正确地接收到的数据包的顺序号是 27，则控

制器 205 将比特映象表 26 中位于存储单元 27 的比特置为二进制“1”。另一方面,如果不管由于什么原因数据包不正确地被接收或根本未接收到,则比特映象表 216 中相应比特的值置为二进制的另一个值,例如是二进制“0”。

接收机 200 周期性地通过缓冲器调制器 220 向发射机 100 发送一个状态控制消息,用以表明特别是哪些包是正确地和不正确地(或者根本没有)被接收到的。(在下文中,“不正确地接收的数据包”也包括根本未接收到的数据包。)图 3 示出一个示例性的状态控制消息的例子,具体地说,差错校验字段包含一个常规的差错校验码可使发射机 100 判定状态控制消息是否含有因通道噪声和衰落而可能引起的差错。NR 字段含有一个顺序号,它比接收机传送到其有关的上层控制器的最新数据包的顺序号数大 1。NL 字段含有接收机正确接收的最大顺序号数据包。比特映象字段“bmf”是由比特映象表 216 (图 1) 形成的,其含义是由接收机 200 “告诉”发射机 100,哪些数据包是正确地或不正确地被接收。“bmf”字段中的比特位置与按照上面讨论方式相对于在有关状态消息的 NR 字段内含顺序号的数据包顺序号相对应。同样地,如上所述,bmf 字段中比特值表明,相应的数据包是正确地接收到(例如比特值为二进制的 1),还是不正确地被接收到(例如比特值为二进制的 0)。例如,如果 NR 字段中的值为 8,则比特 NR+1 对应于数据包 9,比特 NR+2 对应于数据包 10,比特 NR+3 对应于数据包 11,依此类推。

如果接收信机 200 发送到发射机 100 的业务量 (traffic) 等级不是“重”，则接收机 200 周期性地通过缓冲器 220 发送图 3 的控制或状态消息。如果这样的业务量相当重而且在两个方向上都要求 (双工运行) 高等级持续通过量，则按照本发明的一个方面接收机 200 发送它的一个局部的式样 (version)，以使接收机 200 把这个局部的状态控制消息添加在用户信息包上。图 4 示出一个示例性的局部控制消息“a”的例子，它是由多个字段例如三个字段组成的。字段 b、c 和 d 分别含有一个组号；一个比特用以表明 NR 是否在这个组内；以及一个局部比特映像它包含例如 8 个比特。更具体地说，该组号是一个地址用以识别该比特映像表 216 内含的比特组 (例如 8 比特组)，形成字段 d 的诸多比特分别代表所识别的比特组的状态。

据此，在业务量重时，按照本发明的一个方面，接收机 200 可以发送多个局部状态控制消息 (每个局部状态控制消息添加在各有关的信息包上)，而不发送全状态控制消息。按照这样的方式，接收机 200 在重的业务量阶段期间不使接收机到发射机的反向通道的带宽过载，这将在下文详细讨论。

返回来看图 1，缓冲器解调器 140 监视从接收机 200 到发射机 100 的反向无线电通道，并对在该通道上发送的载波信号解调。通过解调产生一个消息 (包)，已由接收机 200 发送出去。缓冲器解调器 140 将每个这样接收的消息存入有关的存储器，再由控制器 120 转储。如上所述，这样的消息可以是图 3 所示的状态控制消息形式

(下文也称之为“全”状态控制消息)。这样的消息还可以是一个信息包，它可以添加也可以不添加局部状态控制消息。

当控制器 120 从缓冲器 140 转储一个消息（包）时，它检查该消息，以确定它是信息包还是全状态控制消息。如果是前者，则控制器 120 把信息包提供给相关的接收机 200（未示出）和解调器 225。在做这之前，控制器 120 将可能已添加在信息包上的任何局部状态控制消息都除掉。如果接收的包是一个全状态控制消息，则控制器 120 利用该状态控制消息的内容重写 RAM 105 内含的比特映像表，其中，所述的比特映像表设置得与比特映像表 216 相似。然后，控制器 120 确定哪些发送的数据包作为存在 RAM 105 内的比特映像表内容的一个函数已被接收机 200 正确地或不正确地接收到，还确定接收的状态控制消息内含的 NR 和 NL 的值。然后，发射机 100 重发接收机不正确地接收到的那些数据包。可是，为了保证同一数据包避免不必要地多次被重发，如上所述，控制器 120 在发送缓冲器 130（图 1）中保存控制器 120 已发送的相应数据包顺序号表。如图 5 所示并如上所述，控制器 120 还将通过缓冲器 135 发送的每个数据包与一个发送次序号相关联。

为便于下面的讨论，假定例如控制器 120 已通过缓冲器调制器 135 向接收机 200 分别发送与顺序号 50 到 54 有关的多个数据包。在这样做时，控制器 120 将每个包按它们被发送的次序存入发送缓冲器 130。控制器 120 还将这样存储的每个包与一顺序发送次序号

相联系，如图 5 所示，例如，可以看出控制器 120 将已发送包的顺序号 50 到 54 分别与发送次序号 831 到 835 相关联。假定在发送包 54 之后但在发送包 55 之前发射机 100 从接收机 200 接收到一个全状态控制制消息。还假定接收的消息内含的 NR 和 NL 值分别是 51 和 54，又除了比特位置 NR 和 NR+1 之外，包括接收到的比特映像的 NR+3（即 54）在内的所有比特位置都置为二进制“1”。

控制器 120 响应这样的内容，确定哪些发送包被接收到：(a) 正确地接收到并传送到上层控制器 250（例如包 50）；(b) 正确地接收到但在顺序之外（例如包 53、54）；(c) 不正确地接收到或根本未接收到（例如包 51 和 52）。根据 NR 的值，控制器 120 从表格中消去一直到和包括顺序号 50 在内的顺序号，这是因为这些包正确地和顺序地已被接收到了的缘故。控制器 120 还从发送缓冲器 130 中消去有关的包。但是，控制器 120 不从该表中消去顺序号 53 和 54，而是标记它们以作确认了。这样做的理由是，虽然这些包被正确接收到了，但它们在顺序之外。然后控制器 120 利用表 500 确定哪些包未被确认，并与具有其数值小于最大发关次序号（下文称作“SL”）且与一个确认的包（即 54）有关的发送次序号相关联。在这样做时，控制器 120 将 SL 置为 835，并把 SL 的当前值存入有关的存储器。然后，控制器 120 确定，顺序号 51 和 52 与发送次序号 832 和 833 相关联，它们小于 SL，标记包 51 和 52，以供重发。根据这个结果，控制器 120 将包 51 提供给发送缓冲器 135，以便重发给接收机 200。当

下一个包已被发送时控制器 120 将包 52 提供给缓冲器 135, 以供重发。在这样做时, 控制器 120 在表 500 中在进入 (entry) 顺序号 54 之后再进入 (re-enter) 较新的顺序号, 如在 501 和 502 处分别示出的那样。此外, 控制器 120 将上述的较新进入的号与接续的发送次序号 836 和 837 分别地相关联, 以便识别那些包的最新发送顺序。接着, 控制器 120 发送与顺序号 55 到 57 相关联的包, 并在把它们提供给缓冲 135 时将这些号列入表 500 中。此外, 如图 5 中所示, 控制器 120 将较新的顺序号分别与发送次序号 838 到 840 相关联。

假定这时在发送包 57 之后、但在发送包 58 之前发射机 100 接收一个状态控制消息。还假定 NR 和 NL 分别等于 51 和 54, 并且除了 NR 外, 比特映像表中一直到和包括 NR+3 在内的所有比特都具有二进制值“1”。同样地, 控制器 120 从缓冲器中消去与顺序号小于 NR 有关的所有包。但是, 如前所述, 控制器 120 不消去正确接收到的但不在顺序内的那些包。然后, 控制器 120 检查表 500, 以确定较新确认的包中哪一个包与最高发送次序号相关联。在这样做时, 控制器 120 发现最高发送次序号与重发包 52 (入口 502) 相关联, 并因此将 SL 置为 837。控制器 120 再确定与未确认的重发包 51 相关联的发送次序号是 836, 它小于 SL。据此, 控制器 120 将包 51 提供给缓冲器 135, 用于重发, 并当该包已经发送时, 也即缓冲器 135 空时, 控制器 120 第一次将包 58 提供给缓冲器 135。此外, 顺序号 51 和 58 按照它们被发送的次序列到表 500 中, 并分别与发送次序号

841 和 842 相关联。

假定这时发射机 100 接收一个状态控制消息是在发送包 58 之后、但在发送数据包 59 之前，还假定 NR 和 NL 分别等于 51 和 57，并除了 NR 和 NR+5 外，比特映像表中一直到和包括 NR+6 在内都具有二进制值“1”。同样地，控制器 120 从缓冲器中消去其序号小于 NR 有关的所有包，但控制器 120 不消去被正确接收到的但不在顺序内的那些包，即包 52 到 55 和 57。然后，控制器 120 检查表 500，以确定在 NR 和 NL 之间较新确认的包中哪些包与最高发送次序号相关联。在这样做时，控制器 120 发现最高发送次序号与发送的包 57 相关联，如入口 503 处指示的，并因此将 SL 置为 840。据此，控制器 120 将包 56 (NR+5) 提供给缓冲器 135 以便重发，这是因为所接收的比特映像表明，该包未被正确地接收到，还因为与之相关的发送次序号 (即 839) 小于当前的 SL 的缘故。然而，即使新比特映像仍然表明该包被不正确地接收，控制器 120 也不重发数据包 51。不重发包 51 (即 841) 的理由是当前与重发包 51 (即 841) 相关联的发送次序号大于 SL 的当前值 (840) 的缘故。

假定这时在重发包 56 之后发送窗口 (此后假定它的尺寸是 8 号) 是关闭的。这意味着：在具有序号 51 (59 减窗口号) 的数据包被确认之前，发射机不能发送具有序号 59 的包。再者，依照 SL 的当前值，所有已发送的但未被确认的包 (例如 51、56 和 58) 不能被重发。在先前的安排中，当窗口关闭和无包重发时，发射机停

止其发送数据包，其中包括保持为未被确认的包，诸如包 56、51 和 58。我们业已承认，按照本发明的一个方面，在窗口关闭和无包被标志以供发送期间，发送全部未确认的包是较为有利的，而不允许发射机进入空闲状态并在此期间结束其包的发送，如前所述，这样，当窗口关闭和无包被标志以供发送时，按照本发明的一个方面，发射机 100 进入优先的重发状态，在此期间它自动地和连续地以循环的方式重发未被确认的包，在目前的示例性例子中，它们是包 56、51 和 58。

据此，这时，控制器 120 将包 56、51 和 58 依次地提供给发送缓冲器 135 以供发送（重发）给接收机 200，并且同样地将那些号数与各自的顺序发送次序号如在 505 到 507 处所示的一起加到表 500 上。这些数据包按次序从最低发送次序号到最高发送次序号发送。此外，按照本发明的一个方面，控制器 120 为那些优先重发包的每个包制定多个拷贝比特，作为一种识别的方式，以识别哪些包已优先地重发，如表 500 的栏目 510 中所示。按照本发明的一个方面，控制器 120 再优先地依次重发包 56 和 51，并同样地随着相关的拷贝比特的制定，将那些号列入表 500 中，如 508 和 509 处所示。

假定这时在控制器 120 第二次优选地发送包 58 之前，发射机 100 接收到一个状态控制消息，还假定新接收到的状态控制消息内含的 NR 和 NL 分别地等于 51 和 57，并且除了 NR 之外在比特映像表中一直到并包括 NR+7 在内的所有比特都具有二进制值“1”。同

样地, 控制器 120 从缓冲器中消去与顺序号小于 NR 有关的所有包。还同样地, 控制器 120 不消去已被正确地接收到但不在顺序内的那些包。然而, 按照本发明的一个方面, 只要窗口关闭, 控制器 120 就不改变 SL, 保持 SL 置于 840。据此, 当与一个确认包的发送次序号有关的拷贝比特设定时, 发射机 100 不利用该发送次序号来更新 SL。这样做的理由是, 因该包的多个拷贝比特可能已被发送, 故难以确定用哪一个发送次序号来更新 SL 的缘故。

因此时接收机 200 接收的包 56 和 58 已被确认, 故发射机 100 只优先重发包 51。在这样做时, 控制器 120 将包 51 与每次重发的发送次序号 851、852 和 853 相关联, 并据此设定该拷贝比特, 如表 500 所示。假定这时从接收机 200 接收到一个状态控制消息, 其中 NR 等于 59, NL 等于 58。此时, 因为窗口已关闭, 故 SL 的值保持为 840。作为窗口关闭的结果, 控制器 120 已处在优先重发模式, 并无包被标志可供重发用。然而, 根据接收状态控制消息的结果, 假定发送窗口是打开的。据此, 控制器 120 按上文讨论的方式发送下一个顺序的包, 也即为表 500 所示的以顺序号 59 开始的包。

如上所述, 在发射机 100 与接收机 200 之间繁重业务量期间, 接收机 200 与开始发送局部状态控制消息, 而不发送全状态控制消息。如上所述, 一个局部控制消息涉及一组数据包, 例如 8 个数据包, 它们的接收状态由代表局部比特映像的一组比特例如 8 个比特来限定。如上所述, 接收机 200 将该组比特和一个相应的组号添加到一

个用户信息包上,以发送给发射机 100。发射机 100 响应对这样的信息包的接收,“剥掉”(strip off)该局部控制消息,然后将此信息包传送所收信机缓冲器 225。发射机 100 随后只针对该消息中被间接识别的那些数据包处理其局部状态控制消息。

具体地说,假定这时顺序号的范围是 0 到 127,控制器 120 刚已发送的数据包分别具有顺序号 45 到 48。还假定控制器 120 已将较新的数据包分别与发送次序号 731 到 743 相关联,并将该信息存入有关的发送表 500 中,如图 6 所示。从图 6 可以看出,它或多或少与图 5 相似,只是图 6 现在包括一个附加的发送启动字段(enable field) 520,这将在下文讨论。(此外,还如下文将要讨论的,图 6 所示的表 500 包含 128 行,分别与顺序号 0 到 127 有关。)

还假定此时在发送包 48 之后、但在发送包 49 之前,发射机 100 从接收机 200 接收到一个具有局部比特映像的信息包 8。较新的包的一个示例性实施例示于图 7。参照图 6 和图 7 可以看到,在包 8 中的局部比特映像“a”指定到组号 5,借此表明有关的 8 个比特表示与该组相关的包(也即分别与顺序号 40 到 47 有关的数据包)的接收状态(被确认(ACK)或未被确认(VNAK))。子字段“C”置为二进制“1”时,表明 NR 由有关的局部比特映像中第一个 0 比特位置来代表,当其置为二进制“0”时,指明 NR 不在该组中。(包 8 中指定为“e”的字段代表该包的其余部分。)

还假定接收机 200 正确接收到顺序号为 0 到 44 的包,即组 0 到

4, 并通过有关的局部比特映像通知发射机 100 这一事实。这样, 接收机 200 不重复发送较新的局部比特映像, 因为那些包已正确地接收到, 并且发射机 100 已得到这方面的通知以及 NR 现在与组 5 相关联。比时, 假定发射机 100 已接收到指明 NR 等于 45 和 NL 等于 44 的比特映像。然而, 如下文所讨论的, 如果那些包中任一个或多个包被不正确地接收到, 则接收机 200 连续发送一组包的局部比特映像。再者, 如果多于一组的包未能正确地接收到, 则接收机 200 以循环方式向发射机 100 发送有关的局部比特映像, 这也将在下文讨论。当接收机 200 已正确地接收到全部未决的 (outstanding) 数据包并且 $NR = NL + 1$ 时, 则只要接收机 200 连续地接收任何类型的数据包, 它继续通过字段 “C” 发送识别 NR 的比特映像。

具体地说, 发射机 100 响应接收到图 7 所示的信息包 8 (它在发送图 6 所示的包 48 之后被接收到), 如上所述, 剥掉局部比特映像 “a”。根据接收到的局部比特映像, 使 $NR = 46$, $NL = 45$ 。此外, SL 等于 731 (它大于 SL 的先前值)。据此, SL 被置位等于 731。发射机 100 利用该信息, 按照处理全比特映像的方式处理该局部比特映像。在这样做时, 控制器 120 对于要重发给接收机 200 的数据包顺序号相关联的那些重发启动字段中记入 (enter) 二进制 “1”。然而, 即使接收到的比特映像 8 指出那些包是未确认的 (UNAK), 控制器 120 也不标志 (启动) 包 46 和 47 用于重发。不标志这些包用于重发的原因是因为接收到的最大顺序号 NL 等于 45, 该信息是模糊的 (am-

biguous) 缘故。这种情况发生在接收机 200 尚未接收到包 46 和 47 的时候。然后, 控制器 120 继续向缓冲器 135 依次提供发送包 49 到 51, 以便发送到接收机 200。假定在发送较新的包之后、但在发送包 52 之前, 发射机 100 接收到信息包 9。同样地, 控制器 120 从信息包 9 中剥掉局部比特映像“a”, 并将该包传送到预定的目的地。

可以看出, 包 9 的字段“a”指明该比特映像与含有 48 到 55 的组 6 相关联, 并且 NR 不在此组内。尚且, NR 依然等于 45, NL 等于 49, SL 等于 735。(因 735 大于 SL 的值 (即 731), 故 SL 被置为 735。) 因其发送次序号 734 小于 SL (735), 故, 作为处理该比特映像的结果, 控制器 120 标记与顺序号 48 相关联的重发启动字段 (利用数值“1”来启动它), (尽管从当前的局部比特映像中可能得不到具有顺序号 46 和 47 的数据包的状态, 但一接收到组 5 的局部比特映像, 或者根据 NR 是否移到组 6, 就可得到这样的数据包状态。) 在作出这样的标记之后, 控制器 120 根据这样标记的有关的重发启动字段, 重发包 48。可以认为控制器 120 然后继续发送包 52, 并在发送较新的包之后接收信息包 10。

再有, 包 10 携带内含 NR 的一个局部比特映像, 并隐含地确认具有顺序号小于 NR 的所有包。这样, 作为控制器 120 处理该局部比特映像的结果, NR 等于 48, 包 46 和 47 被隐含地确认。因与包 46 和 47 相关联的发送次序号小于 735, 故 SL 保持等于 735。据此, 控制器 120 不重发任何包, 但继续发送包 53。

假定此时发射机 100 从接收机 200 接收到信息包 11，剥掉比特映像“a”而将其余的部分传送到预定接收器。作为处理该较新的比特映像的结果，NR 仍保持等于 48，但 NL 和 SL 分别改变到 52 和 739。据此，由于接收机 200 没有确认接收有关的包，但确认了接收与发送次序号 739 相关联的包 52，所以控制器 120 标记与发送次序号 737 相关联的发送启动字段。在作出这样的标记之后，控制器 120 依次地重发包 48 和 51，并将这些包分别与发送次序号 741 和 742 相关联。随后，假定发射机 100 接收到信息包 12，它由控制器 120 同样地进行处理。上述处理的结果是 NR、NL 和 SL 分别置为等于 53、52 和 739。根据这些数值，控制器 120 断定经过顺序号 52 的所有数据包已由接收机 200 确认了，因而没有必要重发的未决的数据包。据此，发射机 100 通过向发送缓冲器 135 提供包 54，继续发送它的数据包。

在本发明的一个较佳实施例中，如上所述，表 500 只含有与顺序号 0 到 127 分别关联的 128 个入口。这样，表 500 具有固定的大小，其中，仅有发送次序号、确认状态、重发启动、以及一行的拷贝比特字段的内容变化，例如，图 6 列出包 48 三次，分别与发送次序号 734、738 和 741 相关联。实际上，顺序号 48 即使重发许多次，它在表 500 中仅列出一次。也就是说，按照上例，第二次发送包 48 时，发送次序号 734 由入口 48 上的号 738 代替。同样地，第三次发送包 48 时，发送次序号由 741 代替，等等。

图 8 和图 9 示出在接收机 200 内实施本发明的软件流程图。先参看图 8，当它接收到一个不含有差错的数据包时程序在方框 800 处进入并进到方框 801。在方框 801 处，程序将接收到的数据包内含的序号（下文称为“SN”）与 NR 的当前值进行比较。如果 SN 不在 NR 与 $(NR + \text{窗口} - 1)$ 之间，意味着接收机 200 已经接收到该包，则程序放弃此接收的包，然后经方框 802 退出，以等候接收下一个包。（注意，在变量 SN、NR、NL 和 NG 上的所有操作都是由每个变量即分别是 128、128 和 16 的范围决定的按模计算的操作。还有，比较操作被认为是针对在当前窗口亦即 NR 与 $(\text{窗口} - 1)$ 之间有效的值而言的。如果窗口跨越数值 0，则适合的操作被认为是保持了该操作的固有意义。）否则，程序进到方框 803，在这里确定该比特映像“RCVR__状态”在对应于序号 SN 的位置上是否其值为“1”。如果“是”，接收机 200 断定，它已经接收到一个载有此序号的包，从而在方框 802 处退出。如果“否”，则程序到方框 804，在这里将接收的包按顺序包存入缓冲器 210，然后，如上所述，更新存在存储器 215 中的比特映像，然后程序再前进到方框 805，在这里检查 SN 是否大于 NL。如果“是”，则程序进到方框 806，在这里将 NL 更新到等于 SN。程序再进到方框 807。

在方框 807 处，程序判定 NR 是否等于 SN。如果“否”，则没有新的顺序内的数据可得到用以传送到上层控制器 250，因此在方框 811 处退出。否则，程序断定这样的数据可以得到，并进到方框 808。

在方框 808 处，根据可得到的用于传送到上层控制器 250 的顺序内的数据包号，程序确定 NR 的新值。上述操作之后，程序确定对应于“RCVR__状态”中的第一个进入的序号（它等于 0）。这是 NR 的新数值。然后，程序前进到方框 809，在这里它将其号数在 NR 的先前值与比 NR 的新值小 1 的数值之间的全部顺序内的数据包传送到上层控制器。程序便经由方框 812 退出。

现在参照图 9，产生状态控制消息（局部的和全部的）的接收机程序在方框 900 处进入，然后进到方框 901。在方框 901 处，根据存在存储器 RAM 215 中的比特映像和 NR 与 NL 的当前值，该程序产生出一个全状态控制消息。然后，程序在方框 902 处将该全状态控制消息存入缓冲器 220。在方框 903 处，程序将 NG 加 1，并在方框 904 处检查 NL 的值是否小于 $NG \times 8$ ，或是在方框 906 处检查 NG 的值是否小于 $NR/8$ 所得商的整数部分。如果“是”，则程序进到方框 905，将 NG 置为 $NR/8$ 所得商的整数部分，并指明 NR 由比特映像中的第一个 0 来代表。在方框 907 处，应用与 NG 值所确定的组相关联的比特映像表 216 中比特状态，程序产生一个局部状态控制消息。然后，在方框 908 处，程序将该局部状态控制消息存入缓冲器 220。程序最后经由方框 909 退出。

在缓冲器 220 处，如果在缓冲器 220 内未存储用户信息包，则发送全状态控制消息。如果用户信息包存储了，缓冲器 220 便对之添加上局部状态控制消息，再将所得出的结果发送出去。

图 10 示例出当发射机 100 接收到一个状态控制消息时进入的程序。具体地说，程序在方框 1000 处进入，前进到方框 1001。在方框 1001 处，如果状态消息是一个全状态控制消息，则程序进到方框 1002。否则，程序进到方框 1004。

在方框 1002 处，程序更新 RAM 105 中存储的全比特映像的内容，并按接收到的状态消息中数据的规定更新 NR 和 NL 的值。程序随后确定 SL 值，它是当前的 SL 和新确认数据包的发送次序号中较大者，然后，程序进到方框 1003，在这里它标记待重发的且它们的发送次序号小于 SL 值的那些数据包中的启动比特。程序乃经由方框 1011 退出。

在方框 1004 处，程序检查 NR 是否含在接收的局部状态控制消息（通过图 4 中的字段“c”）中。如果“是”，则程序在方框 1005 处根据组成有关的 8 比特组的二进制比特值确定 NR 的值。接着，程序在方框 1006 处为较新的 8 比特组确定新的 NL 值。程序在方框 1007 处再检查 $NL_{\text{新}}$ 是否大于 NL 的当前值。如果“是”，则程序进到方框 1008，将 NL 设定到等于 $NL_{\text{新}}$ 。否则，程序在方框 1009 处标记确认的数据包，并根据局部状态控制消息的内容更新 SL 的值。程序在方框 1010 处对于指明较新的控制消息是未被确认的、且它们的发送次序号 (SSN) 小于 SL 的那些数据包，设定重发启动比特。然后，程序进到方框 1011 退出。

每当发送缓存器 135 空时，图 11 的程序在方框 1100 处进入。具

体地说,程序首先确定如有任何包则哪个包将要重发。程序经由方框 1101~1103、1107 和 1108 实现这一点,这些步骤方框代表对表 500 (图 5 和图 6) 进行搜索的程序环路,从 NR 开始为一个未确认的数据包(其重发启动比特置为“1”)寻找出最小顺序号。如果该顺序号被找到,则程序在方框 1104 为该顺序号清除重发启动比特,并将该顺序号与下一个顺序发送次序号(SSN)相关联。然后,程序在方框 1005 处将有关的包存入缓冲器 135,然后退出。此后,当缓冲器 135 再空时,程序在方框 1100 处重新进入。

如果没有发现具有顺序号数在 NR 和 NR_{max} 之间的、其有关的发送启动比特置定于 1 的数据包,则程序进到方框 1109。在方框 1109 处,程序检查 NR_{max} 是否小于 $NR +$ (发送窗口)。如果“是”,则程序在方框 1110 处检查缓冲器 110 是否含有一个数据包。若“有”,则程序在方框 1111 处将该包与下一个顺序号及下一个顺序发送次序号相关联,并把该结果存入 RAM 105。然后,程序在方框 1112 处将该数据包存入缓冲器 135,用于向接收机 200 发送,然后程序退出。

如果在方框 1109 处发现发送窗口是关闭的,或者在方框 1110 处发现没有新的数据包要发送,则程序进到方框 1113。方框 1113 到 1118 代表一个环路的程序,它搜索与一个与未确认包相关联的最小发送次序号(SSN)。当程序找到该号时,它便优先地重发有关的数据包。在这样做时,程序在方框 1119 和 1120: (a) 为该数据包设定拷贝比特为二进制值“1”; (b) 将该包与下一个发送次序号相关联;

(c) 将该包存入缓冲器 135。如上所述，此后，当发送缓冲器 135 变空时，程序在方框 1100 处重新进入。

上文仅是本发明原理的示例说明。本领域技术人员将能设想出无数的安排，它们虽然在这里没有明显地示明或描述，但都体现在本发明权利要求书的精神和范围之内。

图 1

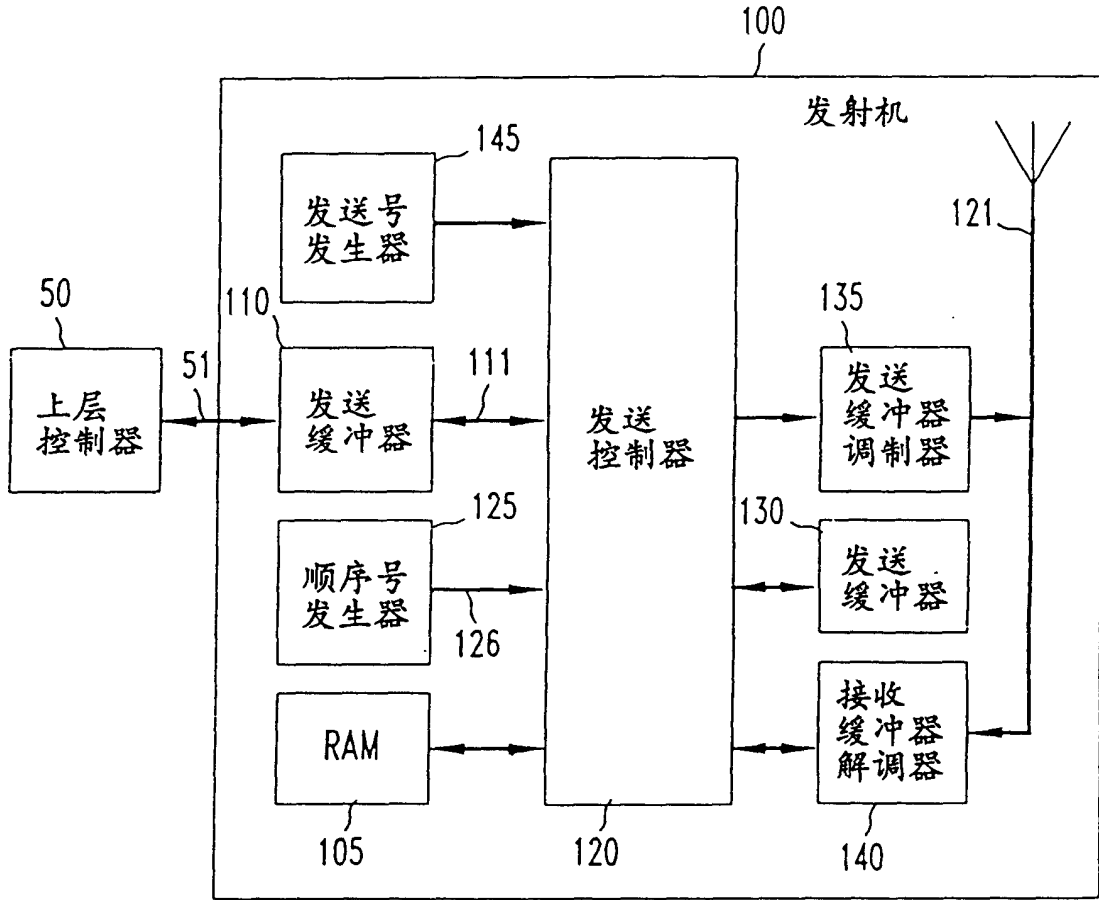


图 2

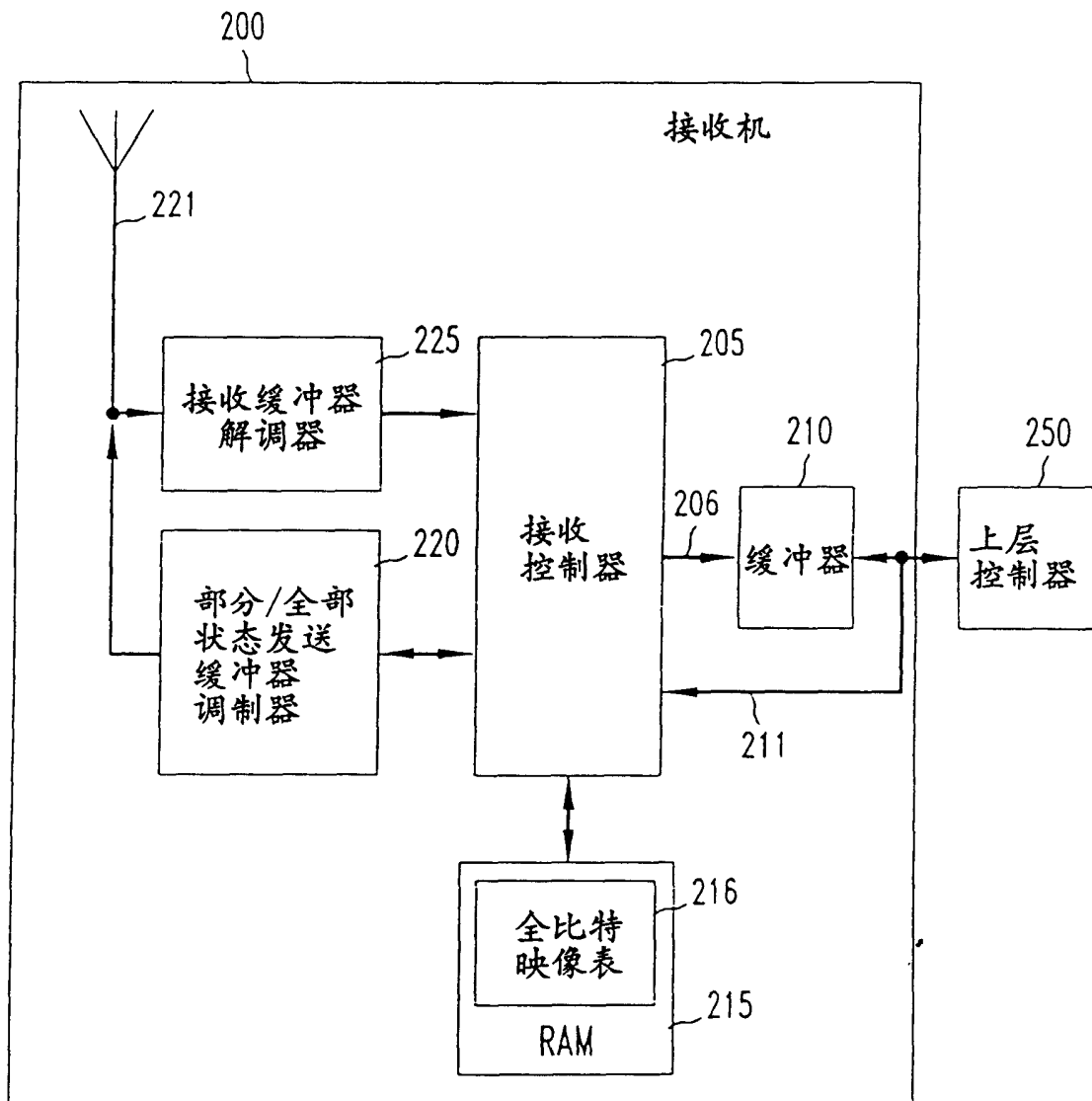


图 3

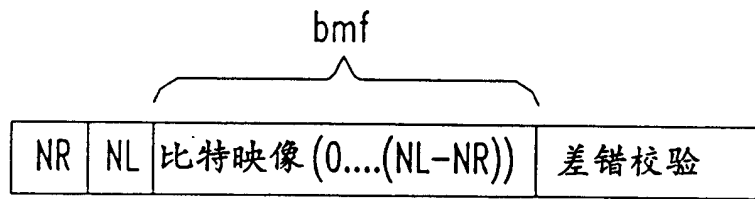


图 4

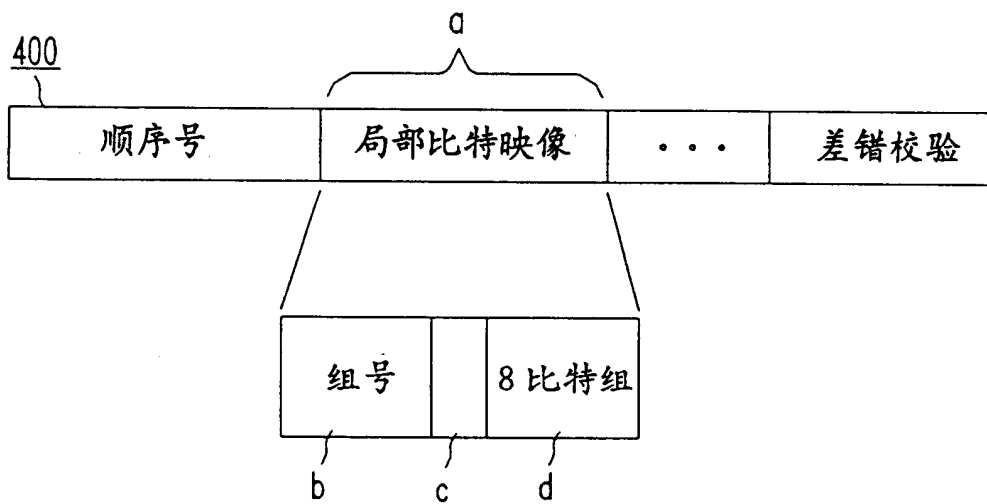


图 5

	顺序号	发送 次序号	CB
	50	831	0
	51	832	
	52	833	
	53	834	
	54	835	
501	51	836	
502	52	837	
	55	838	
503	56	839	
	57	840	
	51	841	
	58	842	
504	56	843	
	51	844	
	58	845	
505	56	846	1
506	51	847	1
507	58	848	1
508	56	849	1
509	51	850	1
	51	851	1
	51	852	1
	51	853	1
	59	854	

图 6

顺序号	发送次序号	重发启动	CB
45	731		
46	732		
47	733		
48	734	1	
49	735		
50	736		
51	737	1	
48	738	1	
52	739		
53	740		
48	741		
51	742		
54	743		

图 7

		a		e	
		40	47		
顺序号	5	1	11111100	-----	
48 55					
顺序号	6	0	01000000	-----	
40 47					
顺序号	6	1	01000000	-----	
48 55					
顺序号	6	1	01101000	-----	
48 55					
顺序号	6	1	11111000	-----	

图 8

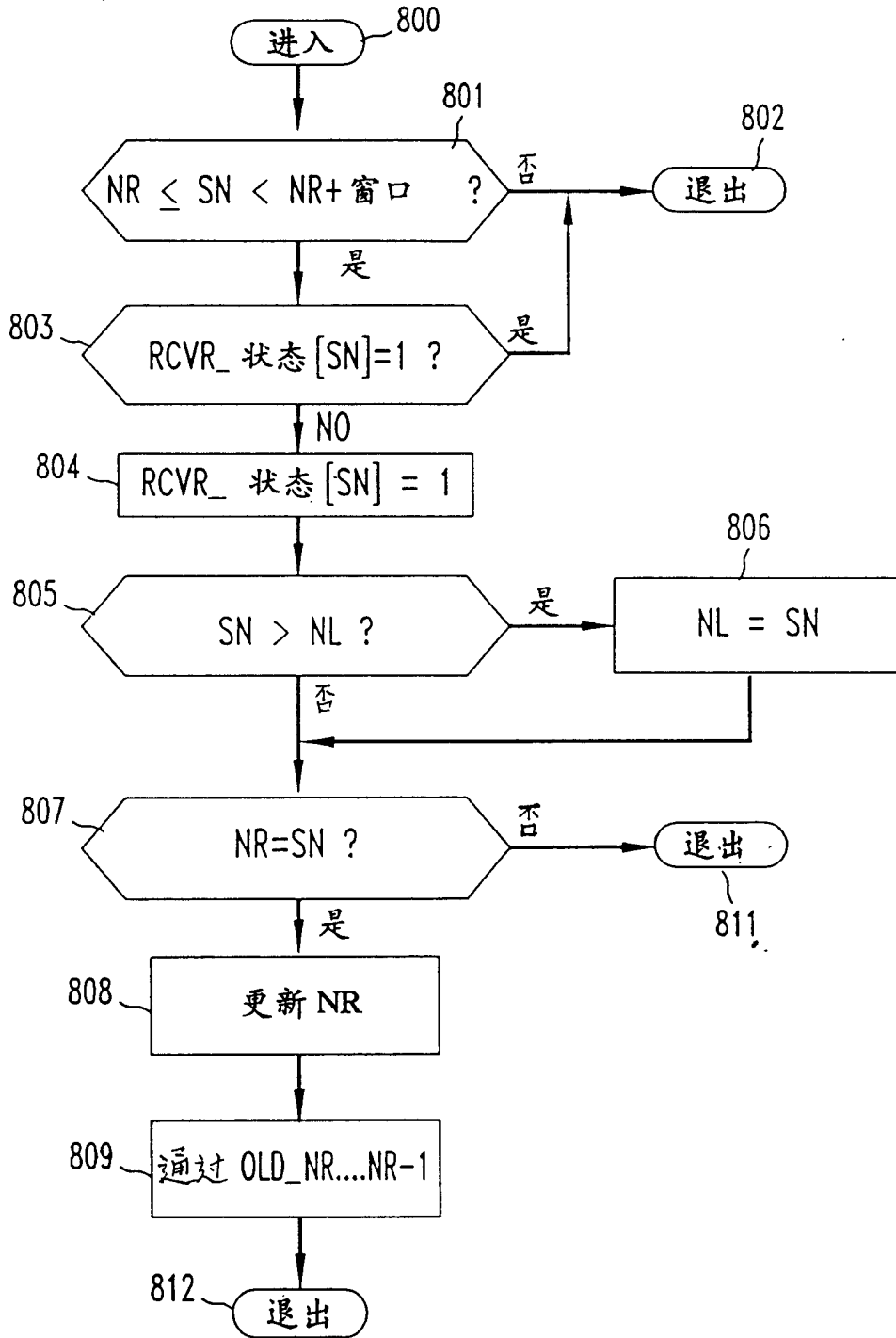


图 9

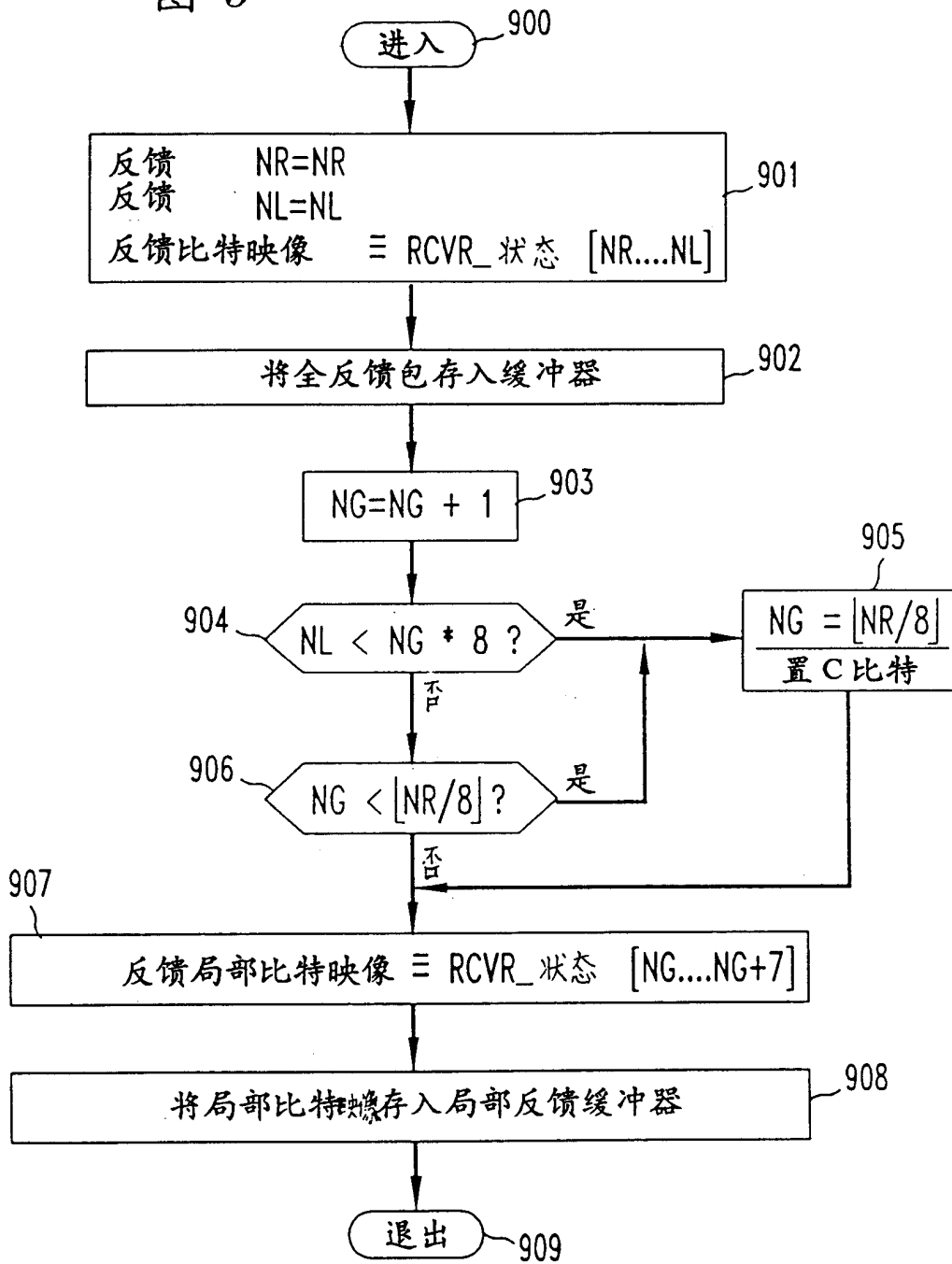


图 10

