

[19] 中华人民共和国国家知识产权局

[51] Int. Cl⁷

H04Q 7/22

H04L 1/18 H04L 1/12



[12] 发明专利说明书

[21] ZL 专利号 99811169.4

[45] 授权公告日 2004 年 1 月 14 日

[11] 授权公告号 CN 1135014C

[22] 申请日 1999.7.13 [21] 申请号 99811169.4

[30] 优先权

[32] 1998.7.21 [33] US [31] 09/120,163

[86] 国际申请 PCT/SE99/01263 1999.7.13

[87] 国际公布 WO00/05911 英 2000.2.3

[85] 进入国家阶段日期 2001.3.21

[71] 专利权人 艾利森电话股份有限公司

地址 瑞典斯德哥尔摩

[72] 发明人 B·拉通伊 H·奥洛夫松

F·U·卡恩

审查员 徐 刚

[74] 专利代理机构 中国专利代理(香港)有限公司

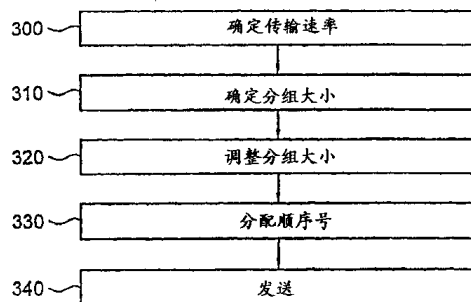
代理人 程天正 陈景峻

权利要求书 2 页 说明书 18 页 附图 11 页

[54] 发明名称 用于使通信系统中的开销最小的方法与设备

[57] 摘要

提供用于使通信系统中的分组重发的开销最小的一种方法与设备。根据当前传输速率、每个分组的大小和以前分配的顺序号给此分组指定一个顺序号。能调整此分组大小,以使整个分组适合单个传输块,也可以根据通过量来调整分组大小。可以根据传输速率和/或通过量、是否正在发送分组或是否正在重发此分组来调整分组大小。或者,如果正在重发此分组,则以其原始传输速率发送此分组而不管当前传输速率如何。



ISSN 1008-4274

1. 用于在通信系统中发送分组的一种方法，此方法包括以下步骤：
 - 确定当前传输速率；
 - 5 根据当前传输速率、一个分组的大小和以前发送的分组的顺序号来分配一个顺序号给此分组以便发送；和
 - 以当前传输速率发送此分组。
2. 权利要求1的方法，还包括：
 - 调整分组大小，以使整个分组适合当前传输速率上的单个传输块。
- 10 3. 权利要求2的方法，其中此调整步骤还包括：
 - 确定此发送分组是否具有与能以当前传输速率在单个传输块中进行发送的分组相同的大小；和
 - 如果此分组较大，则将此发送分组划分为4个较小分组。
4. 权利要求3的方法，还包括：如果此发送分组小于能在单个
- 15 传输块中发送的分组，则在此传输块中附加上另外的分组。
5. 权利要求1的方法，还包括：
 - 接收一个发送块，该发送块包括一个或多个分组；
 - 检测所述一个或多个分组中的差错；和
 - 重发已检测到差错的块中的所述一个或多个分组中的任一分组，
 - 20 其中如果在重发时传输速率低于原始发送这些分组的速率，则将此分组划分为较小分组，并给这些较小分组重新分配顺序号。
6. 权利要求5的方法，其中此检测步骤包括：
 - 检查被传输的块中的所述一个或多个分组的差错；和
 - 释放未检测到差错的被传输的块中的那些分组。
- 25 7. 权利要求5的方法，还包括根据在某一时间周期期间重发的分组的数量来调整分组大小以使通过量最大的步骤。
8. 权利要求1的方法，还包括以下步骤：
 - 确定是否正在重发分组，其中如果未在重发分组，则执行确定当前传输速率、分配顺序号和以当前传输速率发送分组的步骤，或如果
 - 30 正在重发此分组，则以其原始传输速率重发此分组。
9. 一种收发信机，包括：
 - 处理器，用于确定当前传输速率，并根据当前传输速率、一个分

组的大小和以前发送的分组的顺序号来给此发送分组分配一个顺序号；和

发射机，用于以当前传输速率发送此分组。

5 10. 权利要求 9 的收发信机，其中此处理器调整分组大小，以使整个分组适合当前传输速率上的单个传输块。

11. 权利要求 10 的收发信机，其中此处理器确定此发送分组是否具有与能以当前传输速率在单个传输块中发送的分组相同的大小，而如果此分组较大，则此处理器将此发送分组划分为较小分组。

10 12. 权利要求 11 的收发信机，其中如果此发送分组小于能在单个传输块中发送的分组，则此处理器在此传输块中附加上另外的分组。

13. 权利要求 9 的收发信机，其中接收此发送块，并且此处理器检测此被发送的块中的差错，而如果在此发送块中检测到任何差错，则此发射机重发已检测到差错的分组，和其中如果在重发时传输速率低于原始发送这些分组的速率，则此处理器将此分组划分为较小分组并重新分配顺序号给这些较小分组。

14. 权利要求 13 的收发信机，其中此处理器通过检查此被传输的块中的分组中的差错和释放未检测到差错的被传输的块中的那些分组来检测差错。

20 15. 权利要求 14 的收发信机，其中此处理器根据某一时间周期期间重发的分组的数量来调整分组大小以使通过量最大。

16. 权利要求 9 的收发信机，其中此处理器确定是否正在重发分组，和其中如果未在重发分组，则此处理器确定当前传输速率并分配顺序号，而且此发射机发送此分组，或如果正在重发此分组，则此发射机以其原始传输速率重发此分组。

25

用于使通信系统中的开销最小的方法与设备

背景

5 本发明一般涉及用于使通信系统中的开销最小的方法与设备。更具体地，本发明涉及用于使通信系统中分组重发的开销最小的方法与设备。

10 图 1 是包括示例性基站 110 和移动站 120 的示例性蜂窝无线电话系统的方框图。虽然站 120 表示为“移动站”，但此站 120 也可以是固定的蜂窝站。基站包括连接到移动交换中心 (MSC) 140 的控制与处理单元 130，此单元 130 又连接到 PSTN (未示出)。这样的蜂窝无线电话系统总的方面是本领域公知的。基站 110 通过由控制与处理单元 130 控制的话音信道收发信机 150 管理多个话音信道。每个基站也包括可能管理多于一个的控制信道的控制信道收发信机 160。此控制信道收发信机 160 由控制与处理单元 130 进行控制，此控制信道收发信机 160 在基站或小区的控制信道上广播控制信息给锁定到那个控制信道上的移动站。将明白：收发信机 150 与 160 能实施为单个设备，类似于话音与控制收发信机 170，以便和共享同一无线电载波的控制与业务信道一起使用。

20 移动站 120 在其话音与控制信道收发信机 170 上接收在控制信道上广播的信息。随后，处理单元 180 评估接收的包括是此移动站锁定到的候选者的小区的特性的控制信道消息，并确定此移动站应锁定到哪个小区。有益地，此接收的控制信道信息不仅包括涉及与之相关的小区绝对信息，也包含涉及此控制信道与之相关的小区附近的其他小区的相对信息，例如，如授予 Raith 等人的题为“用于无线电话系统中的通信控制的方法与设备”的美国专利号 5353332 中所述的。

诸如蜂窝与卫星无线电系统的现代通信系统采用各种操作模式 (模拟、数字、双模式等) 和诸如频分多址 (FDMA)、时分多址 (TDMA)、码分多址 (CDMA) 的接入技术以及这些技术的混合。

30 在北美，使用 TDMA 的数字蜂窝无线电话系统称为数字先进移动电话业务 (D-AMPS)，其一些特性规定在电信工业协会与电子工业协会 (TIA/EIA) 出版的 TIA/EIA/IS-136 标准中。使用直接序列 CDMA 的另

一数字通信系统利用 TIA/EIA/IS-95 标准来规定, 而跳频 CDMA 通信系统利用 EIA SP3389 标准 (PCS 1900) 来规定。PCS 1900 标准是在北美之外是公用的为个人通信业务 (PCS) 系统引入的 GSM 系统的应用。

当前在包括国际电信联盟 (ITU)、欧洲电信标准研究会 (ETSI) 和日本无线电工业与商业协会 (ARIB) 的各个标准设定组织中正在讨论用于下一代数字蜂窝通信系统的几个建议。除了发送话音信息之外, 还能期望下一代系统传送分组数据并与也通常根据诸如开放系统接口 (OSI) 模型或传输控制协议/互联网协议 (TCP/IP) 组的工业范围数据标准设计的分组数据网络相互协作。多年来, 无论正式还是事实上都一直在开发这些标准, 并且使用这些协议的应用已经能获得。基于标准的网络的主要目的是实现与其他网络的互连。互联网是追求此目标的这样的基于标准的分组数据网络的当今一个最明显的示例。

在蜂窝系统中引入分组数据协议的优点包括支持高数据速率传输和同时实现无线电接口上无线电频率带宽的灵活与有效利用的能力。作为用于全球移动通信系统 (GSM) 标准的分组模式的通用分组无线电业务 (GPRS) 被设计用于所谓的“多时隙操作”, 其中允许单个用户同时占用多于一个的传输资源。

在图 2A 中表示出 GPRS 网络结构的概况。来自外部网络的信息分组在 GGSN (网关 GPRS 业务节点) 10 上进入此 GPRS 网络。然后, 通过骨干网络 12 为分组从 GGSN 选择路由至正服务于寻址的 GPRS 远程站驻留在其中的区域的 SGSN (服务 GPRS 支持节点) 14。在专用 GPRS 传输中从 SGSN 14 为这些分组选择路由至正确的 BSS (基站系统)。BSS 包括多个基站收发信机 (BTS) (只示出其中之一, BTS18) 和基站控制器 (BSC) 20。BTS 与 BSC 之间的接口称为 A-bis 接口。此 BSC 是 GSM 特定的表示, 而对于其他示例性系统, 术语无线电网络控制 (RNC) 用于具有类似于 BSC 的功能的节点。然后, 由 BTS18 使用选择的信息传输速率通过空中接口将分组发送给远程站 21。

一个 GPRS 寄存器保持所有 GPRS 预约数据。GPRS 寄存器可以与 GSM 系统的 HLR (归属位置寄存器) 22 合并, 也可以不合并。用户数据可以在 SGSN 与 MSC/VLR 24 之间进行交换以保证诸如受限制的漫游的业务那样的相互作用。BSC20 与 MSC/VLR 24 之间的接入网络接口是基于 CCITT 信令系统 7 号的移动应用部分的称为 A 接口的标准接口。MSC/VLR

24 也通过 PSTN 26 接入陆地有线系统。

在大多数的数字通信系统中，通过频率调制具有接近 800 兆赫 (MHZ)、900MHZ 和 1900MHZ 的无线电载波信号来实施通信信道。在 TDMA 系统中和甚至在 CDMA 系统的变化范围中，将每个无线电信道划分为一
5 系列时隙，每个时隙包含来自用户的信息脉冲串。这些时隙组成为连续帧，每帧具有预定的持续时间，并且连续帧可以组成为一系列通常称为的超帧。通信系统使用的这种类型的接入技术(例如，TDMA 或 CDMA)影响如何在时隙或帧中表示用户信息，但是目前的接入技术全部使用时隙/帧结构。

10 分配给同一用户的可能不是无线电载波上的连续时隙的时隙可以认为是分配给此用户的逻辑信道。在每个时隙期间，根据此系统使用的特定接入技术(例如，CDMA)来发送预定数量的数字比特。除了用于话音或数据业务的逻辑信道之外，蜂窝无线电通信系统也为控制消息提供逻辑信道，诸如用于基站与移动站交换的呼叫建立消息的寻呼/
15 接入信道。一般地，这些不同信道的传输比特率不必一致，而且这些不同信道中的时隙的长度不必均匀。一个信道可能的传输比特率的设置一般是有限的整数值并且对于使用此信道的发射机与接收机来说是公知的。

在蜂窝无线电系统中，为了允许移动站与基站和移动交换中心
20 (MSC) 通信而需要空中接口协议。此空中接口协议用于始发或接收蜂窝电话呼叫。物理层(层 1)定义物理通信信道的参数，例如，载波无线电频率间距、调制特性等。链路层(层 2)定义处于物理信道的限制之内的信息的准确传输所需要的技术，例如，纠错与检错等。无线电资源控制(RRC)层 3 定义了物理信道上发送的信息的接收与处理的程
25 序。例如，TIA/EIA/IS-136 和 TIA/EIA/IS-95 规定了空中接口协议。层 2 协议的功能包括对可以在驻留在移动站与蜂窝交换系统内的通信层 3 对等实体之间发送的层 3 消息的定界或组帧。

远程站与基站之间的物理信道一般划分为时帧，如图 2B 所示。在一个时帧期间发送的信息单元能称为传输块。在下一代系统中，数据
30 能组合为分组以便发送。一个或几个数据分组能在一个传输块内进行发送。

许多通信系统的一个特性是：为了例如响应将发送的信息量的变

化，远程站与基站能发送用户信息比特的速率对于系统中不同的信道可能是不同的并且甚至在一个信道内也能随时间而变化。有时允许传输速率只在帧开始时变化并且必须在整个帧期间保持为恒定值。因而，根据特定帧的传输速率，传输块的大小能随帧而变化，这是 ETSI 与 ARIB 目前正在考虑的第三代系统的代表。给远程站分配能用于信息比特的传输的一组不同速度。此基站控制每个信道的速度分配，并向远程站发送消息以便将允许此移动站使用哪些传输速率通知此移动站。可以由基站在它发送的每个帧中发送这样的消息。

传输速率如何变化取决于系统。例如，在 TDMA 系统中，分配给用户的不同数量的时隙可以随时间而变化。不同的调制与编码方案也能导致不同的传输速率，如公开在例如申请日为 1997 年 8 月 29 日的共同转让的美国专利 No. 6208663 中。对于 CDMA 系统，能改变扩展因数以改变业务信道的速率，这是正在开发的 IMT2000 标准中的一个建议。在 TDMA 与 CDMA 系统中，带有凿孔 (puncturing) 方案 (即，用于选择抛弃哪些比特和传输之前的信道编码器输出的方法) 的信道编码也能用于获得不同的速率。

假定一个分组在长度上小于或等于一个传输块。假定不同的分组大小的数量为有限的整数值。这些分组的大小取决于当前用于此信道的传输速率设置并因此对于发射机与接收机二者是公知的。给这些分组指定顺序号并在一个传输块内以这些顺序号的顺序来发送这些分组。顺序号的分配是任意的并且对于接收机是未知的。此接收机只知道用于每个接收帧的总的传输块有多大。因而，因为接收机总是知道传输块在哪开始与结束，所以无需组帧。发射机不发送有关此传输块中分组组成或安排的任何信息。

在层 2 级上，一个分组一般包括标题部分、信息部分 (I 部分) 和检错码部分。此标题部分一般包括用于请求已破坏的分组的重发的信息，此重发方案通常称为自动重发请求 (ARQ) 方案。称为循环冗余码 (CRC) 的检错部分用于确定此分组的其余部分当在信道上发送时是否已以某一方式被破坏。如果是这样的话，则给发射机发送重发请求信号，从而重发原始数据。

混合 ARQ 方案利用冗余来提供低误码率。具有诸如类型 I 混合 ARQ 方案的固定冗余 ARQ 方案和诸如软组合的可变冗余 ARQ 方案以及类型

II 与类型 III 混合 ARQ 方案。

在图 2C 所示的软组合中，当第一次解码尝试失败时，发送已被原始发送的信息的一个拷贝。如果与此拷贝组合的原始发送的信息的第二次解码尝试失败，则发送此信息的一个新拷贝以增加在接收实体上成功解码的概率。能根据需要而执行同一信息的多次重发，直至成功解码此信息。

在图 2D 所示的类型 II 混合 ARQ 方案中，能增加数据冗余度，直至成功解码所接收的信息。通过从信息解码器中发送系统信息来增加冗余度，并且如果需要的话，也发送信道编码器输出的（由生成此码多项式的另一生成器生成的）冗余信息。如图 2D 所示，如果第一次解码尝试失败，则发送冗余信息 R1。如果原始发送信息与此冗余信息 R1 的第二次解码尝试失败，则发送冗余信息 R2，等等。通过从接收的具有最低 C/I 的块开始重发所接收的具有弱载波干扰比（C/I）的块，可以进一步增强类型 II 混合 ARQ 方案。在每个块重发时新近发送的信息与先前信息的分集/软组合之后，进行解码尝试。此处理继续进行，直至解码成功。

这些方案具有的问题是需存储以前发送的信息，以便此信息能与新近发送的信息进行组合来解码，这将消耗大量的存储器。

在类型 III 混合 ARQ 方案中，每个子块自身是可解码的。因而，在没有存储空间可利用时，总是能抛弃以前发送的块。

可变传输速率对于分组重发将呈现出问题，这是因为发送的速率可能不同于重发的速率。已提议了处理可变传输速率的分组重发的技术。根据一种技术，分组顺序编号总是递增 1，并且此分组大小总是固定为对应于可能的传输速率设置内最小可能的速率的分组大小。否则，在改变传输速率时不可能处理分组的重发，这由于较高的传输速率而会导致许多开销，固定的数据分组大小小于实际能发送的分组的大小。由于此开销因而浪费了许多不必要的无线电资源。

根据另一种技术，使用固定的分组长度，但对于较低传输速率能使用交错在几个时帧中发送一个分组，这产生大的延迟。另外，这浪费信道资源，这是因为不管用户数据量如何，分组大小总是相同的。

通信系统中信道差错特性变化很大并取决于例如移动站的速度、来自诸如建筑物、山脉等的物体的不同反射、噪声干扰、小区业务负

载和传输速率。这些信道特性根据移动站正在其中四处移动或其中固定蜂窝站安装的环境而可能不同。这些信道特性影响通过量。

在上述的常规解决方法中，分组大小对于不同的传输速率是固定的。不存在对重发分组的数量的连续评估，并且也不存在通过对分组大小的调整来使通过量最大。因而，这些分组大小在某些业务信道条件下将不是最佳的。

具有各种公知的用于调整传输速率的方法。例如，对于使用不止一个的调制和/或信道编码方案（MCS）的系统，例如，GPRS GSM、D-AMPS++、EDGE GSM，转换到较高级调制（HLM）或更少的信道编码会导致较高的信息比特率，并且反之亦然。同样地，在宽带 CDMA（WCDMA）系统中，利用不同的传输格式（TF）来获得不同的信息速率。利用传输速率、代码、交错深度和使用的重复与凿孔方案来定义 TF。

为了无线电接口上有效的传输，将较高层 PDU（协议数据单元）分段为较小尺寸的 RLC（无线电链路控制）块或分组，这也允许在 RLC 块级别上根据表示为 RLC 协议（RLCP）的 ARQ 协议执行重发。

在例如图 1 中所示的基站与远程站的两个对等实体之间建立 RLC 连接用于 RLC 块的传输。每个实体具有接收 RLC 块的接收机和发送 RLC 块的发射机。通过一个窗口来控制这两个实体之间的块传输，即利用窗口大小来限制大量 RLC 块（未确认的块）的最大数量。

当前使用两种不同的方法来在使用不止一个的 MCS/TF 的系统中将较高层 PDU 帧中的块划分为 RLC 块。根据一种方法，较高层 PDU 帧中的整个块划分为其大小对应于当前使用的 MCS 或 TF 的 RLC 块，这使之不可能在较高层 PDU 帧的传输周期期间在不同的模式之间进行转换。根据第二种方法，将较高层 PDU 帧中的块划分为对应于最低传输速率的 RLC 块大小。然后，在一个块周期期间发送整数数量的这些相对较小的单元。将较高层 PDU 帧中的块分段为对应于最小传输速率的较小单元将会由于用于小尺寸单元的 RLC 标题与 CRC 而导致大的开销，例如，具有帧长等于 10 毫秒的系统中的 16 千比特/秒承载业务只有 160 比特的块大小，而开销占 25% 的容量（例如，16 比特 CRC 和 24 比特 LRC 标题）。此方法也因为一个传输块只应包含一个 RLC 块来支持类型 II 混合 ARQ 与软组合而使得使用软组合和/或类型 II 混合 ARQ 很困难。

因而，需要一种能减少分组重发开销的方法与设备，也需要一种

能支持类型 II 混合 ARQ 与软组合的可变速率分组重发的方法与设备。

发明概要

因此本发明的一个目的是使得资源的利用最大化并减少分组重发中的开销。本发明的另一目的是使通过量最大。本发明的还一目的是在可变速率分组重发中支持类型 II 混合 ARQ 与软组合。

根据本发明的示例性实施例，利用在通信系统中发送/重发分组的方法与设备来实现这个与其他的目的。

根据第一实施例，确定当前传输速率，根据当前传输速率、一个分组的大小和以前发送的分组顺序号给一个用于发送的分组分配一个顺序号，并发送此分组。调整此分组的大小，以使整个分组适合当前传输速率上的单个传输块。

根据第二实施例，根据通过量来调整用于发送的分组的大小。确定在某一时间周期期间重发的分组的数量，并且利用此测量结果来选择使通过量最大的分组大小。

根据第三实施例，确定是否在重发分组。如果不是的话，则确定当前传输速率，给此分组分配一个顺序号，并且与第一实施例中一样发送此分组。否则，以其原始速率重发此分组。

附图简述

本发明的特性、目的与优点通过结合附图阅读此描述将变得显而易见，其中相同的标号指相同的部分，其中：

图 1 是蜂窝示例性无线电话通信系统的方框图；

图 2A 表示 GSM/GPRS 网络结构；

图 2B 表示划分为帧的物理信道；

图 2C 表示软组合重发方案；

图 2D 表示类型 II 混合重发方案；

图 3A 与 3B 表示根据第一实施例的示例性分组顺序编号技术；

图 3C 与 3D 表示根据第一实施例的信道中的示例性分组发送/重发；

图 3E 表示根据第一实施例用于分组发送/重发的方法；

图 3F 表示根据第二实施例用于使通过量最大的方法；

图 4A - 4C 表示示例性纠错解码；

图 5A 表示根据第三实施例的示例性分组顺序编号；

图 5B 表示根据第三实施例的信道中的分组发送/重发；和
图 5C 表示根据第三实施例用于分组发送/重发的方法。

详细描述

根据本发明的示例性实施例，能调整分组大小、顺序编号和/或传
5 输速率以使通信系统中的分组重发开销最小。

根据第一实施例，能选择分组大小以使开销最小。例如，能选择
分组大小，以便对于每个传输速率，整个分组适合一个传输块。因而，
当以高的传输速率发送时不浪费额外的开销。另外，能以这样一种方
式给这些分组分配顺序号，以便能有效地以不同的传输速率来处理错
10 误分组的重发。

可以根据一个分组中信息部分（I 部分）的大小和一组传输速率
之中的当前传输速率来分配顺序号给此分组。适合最小分组的信息量
能对应于递增“1”（一个单位）的顺序号。因而，能给具有最小 I 部
分的分组分配一个比以前分配的顺序号大一个步长 1 的顺序号。能选
15 择用于较大分组大小（具有较大 I 部分）的顺序号，于是在此分组被
破坏或丢失时（即在接收的分组中检测到差错时）总是能执行重发，
并且此速率组中的最小速率能用于重发错误分组的整个 I 部分，这可
以通过在发射机上将长分组重新安排为几个较小分组来实现。然后，
除了那些保持原始 I 部分的第一或最后一部分的分组之外，在发送较
20 小分组时，能给这些较小分组分配新的顺序号，这将在下面参照图 3C
与 3D 更详细地进行解释。

假定能发送的最小 I 部分（对应于加 1 的顺序号）的大小为 L_I ，
则下面的规则适用于确定任何一个保持 I 部分的分组的顺序号增加 SEQ
- INCREASE:

$$25 \quad (\text{SEQ_INCREASE} - 1) * L_I < (\text{I-Part}) \text{ 的大小} \leq \text{SEQ_INCREASE} * L_I \quad (1)$$

因而，下面的等式能用于确定一个必须能在信道上利用长度大小
为 L 的传输块来重发的具有当前长度大小的分组的顺序号的增加:

$$\text{SEQ_INCREASE} = \lfloor \text{current_size (当前大小)} / L \rfloor \quad (2)$$

其中 $\lfloor \rfloor$ 是将变元舍入为最近的较大整数值的运算符。

30 分组的顺序号在下面将表示为 NS。此顺序号的表示通常用于基于
公知的高级数据链路控制（HDLC）协议的常规 ARQ 系统中。

能以许多种方式来给具有大于最小分组大小的、对应于大于 1 个

单位的顺序号增加的 I 部分的分组分配顺序号，在图 3A 与 3B 中表示出其中两种方式。

在图 3A 与 3B 中，第一行中的分组表示最大的 I 部分。根据示例性实施例，例如通过以常规的方式划分较高层 PDU 帧块就可以获得这些分组。这些分组例如可以存储在中间缓冲器中。

第一行中的分组分别隔开 4 个步长进行编号。例如，在图 3A 中，第一行中的分组编号为 $NS = 0$ 和 $NS = 4$ ，而在图 3B 中，第一行的分组编号为 $NS = 3$ 和 $NS = 7$ 。

在第二行中，这些分组包含着是最小 I 部分两倍大小的 I 部分。例如，通过将第一行中所示的分组一分为二就可以获得这些分组。第二行中的分组分别隔开 2 个步长进行编号。例如，在图 3A 中，第二行中的分组编号为 $NS = 0$ 、 $NS = 2$ 、 $NS = 4$ 和 $NS = 6$ 。图 3B 的第二行中的分组编号为 $NS = 1$ 、 $NS = 3$ 、 $NS = 5$ 和 $NS = 7$ 。

在第三行中，这些分组包含与最小 I 部分同样大小的 I 部分并连续进行编号。例如，通过将第二行中所示的分组一分为二可以获得这些分组。因此，在图 3A 与 3B 中，第三行中的分组编号为 $NS = 1$ ， $NS = 2$ ， $NS = 3$ ， \dots ， $NS = 7$ 。

为便于解释，参照图 3B 解释当以较低传输速率重发一个分组时执行顺序号重新分配的方式。假定在第一行中分配有顺序号 $NS = 7$ 的分组将要在只能以能发送第一行中的分组的速率的 $1/4$ 速率发送分组的信道上进行重发，则分配有 $NS = 7$ 的原始分组可被划分为编号为 $NS = 4$ 、 $NS = 5$ 、 $NS = 6$ 和 $NS = 7$ 的四个分组。

图 3C 与 3D 表示根据本发明的第一实施例如何以可变传输速率在信道中发送/重发分组的一个示例。这些分组在传输块中从例如移动站的发送实体中发送给例如基站的接收实体。在每个传输块中发送一个或多个分组。如上面参照图 2B 所述在每个时帧期间发送单个传输块给每个接收实体。接收实体检测接收分组中的差错。为便于解释，发送实体与接收实体在图 3D 中分别表示为 ARQ 实体 1 与 ARQ 实体 2。当然，应明白：ARQ 实体 1 与 ARQ 实体 2 都包括发送与接收能力。

在图 3C 与 3D 中，分组的顺序号表示为 NS ，并且传输速率组为 $(R, 2R, 3R, 4R)$ 。以这些速率发送的完整大小分组的顺序号的增加（即，顺序号步长）为 $(1, 2, 3, 4)$ 。换言之，在速率 R 上，在连续分配

的序号之间具有 1 的差值，在速率 $2R$ 上，在连续分配的序号之间具有 2 的差值，等等。

在图 3C 中，箭头表示速率变化发生的时刻。速率变化未在图 3D 中表示出，但从分组顺序编号中能推断出速率变化发生的时间。例如，
5 从接收实体发送给发送实体的信息能引起速率变化，这些速率变化能以任何常规方式来执行。图 3C 与 3D 也表示发送对所选择分组的重发请求的反向信道。

以速率 R 在第一时帧中以等于 1 的相应序号步长开始分组传输。给发送的第一分组分配序号 0。在下一帧中，给发送的分组分配序
10 号 1。

在第三帧中，此速率改变为对应于 2 的序号步长的 $2R$ 。给此帧中发送的分组分配序号 3（比 1 大 2 的步长）。在信道上丢失此分组，如从图 3D 中能看出，并在接收实体上检测到差错。在第四帧中，给发送的下一个分组分配序号 5（比 3 大 2 的步长）。重发请求信号或否
15 定确认（NAK）与序号 3 一起在下一帧中回送给发送实体。

在第五帧中，速率也改变为对应于序号步长 3 的 $3R$ 。给此帧中发送的分组分配序号 8（比 5 大 3 的步长）。如图 3D 所示，随后在此信道上丢失此分组，并在接收的分组中检测到差错。

由于此速率大得足以发送具有序号 3 的分组，所以接下来重发
20 此分组。这在此帧中在传输块中只留下足够发送具有能以速率 R 发送的大小的分组的空间。因而，给此大小的分组分配序号 9（比 8 大 1 的步长）并与分配有序号 3 的分组一起进行发送。

在后一帧中与序号 8 一起将 NAK 信号发送给发射机以请求重发。在第七帧中，速率也改变为对应于序号步长 2 的 $2R$ 。因而，给发送的下一分组分配序号 11（比 9 大 2 的步长）。
25

随后，在第八帧中，速率改变为对应于序号步长 1 的 R 。因此，给发送的下一分组分配序号 12（比 11 大 1 的步长）。

在第九帧中，将重发具有序号 8 的分组。然而，具有序号 8 的分组包括是能以速率 R 在一个传输块中发送的 3 倍的信息量。因而，
30 将具有序号 8 的分组划分为分别分配有序号 6、7 与 8 的 3 个分组，并在第九、第十与第十一帧中连续发送这些分组。

如图 3D 所示，在信道上丢失分配有序号 6 与 11 的分组，并在

接收的分组中检测到差错。在第十一帧期间，与这些顺序号一起发送 NAK 信号给发送侧。

在第十二帧中，速率改变为对应于顺序号步长 4 的 4R。在此速率上，在传输块中具有足够的空间来重发分配有顺序号 6 的分组和分配有顺序号 11 的分组，并具有足够的剩余空间来发送包含有能以速率 R 发送的信息的分组。因此，给下一分组分配顺序号 13（比 12 大 1 的步长）并在与具有顺序号 6 和 11 的分组相同的传输块中发送这下一分组。

为简化顺序编号，如果可能的话，两个或多个分组能合并为一个分组并能利用一个编号来替代这多个顺序号。例如，参见图 3C，如果正好在发送分组 NS = 7 之前表示等于 2R 的允许速率的速率变化消息到达，则 NS = 7 分组与 NS = 8 分组的 I 部分能合并在一起并利用只包括顺序号 NS = 8 的速率 2R 进行重发，这只适用于连续编号的分组。例如，图 3C 中 NS = 3 分组与 NS = 9 分组当在同一传输块中进行发送时因为 I 部分不是连续的而不能合并在一起。

以上述方式分配顺序号可以解决在必须以较低传输速率重发较高速率分组时如何处理重发的问题。不在不必要的开销信息上浪费传输信道资源，这将增加信道通过量。

另外，通过调整分组大小以使整个分组适合传输块，可以进一步减少开销。作为一个示例性示例，考虑 ETSI 与 ARIB 的开发 IMT2000 WCDMA 标准的建议，其中具有 10 毫秒时帧和一个速率组中不同的传输速率。假定此速率组包括传输速率（32，64，128）千比特/秒，这对应于一组（320，640，1280）比特的分组大小。每个 RLCP 分组保持一个顺序号、一些控制比特和一个 CRC，导致每个 RLCP 分组近似 32 比特的开销。利用常规的技术，这些分组大小将全部分配给 320 比特的最小分组大小，并且对于每个连续分组，分组顺序编号将递增 1。因此，在 32 千比特/秒的速率上，每个时帧将只发送一个分组，并且所导致的开销将是 32 比特。在 64 千比特/秒的速率上，每个时帧将发送两个分组，导致 64 比特的开销。在 128 千比特/秒的速率上，每个时帧将发送 4 个分组，导致 128 比特的开销。

根据第一实施例，能根据传输速率调整分组大小，以便在每个传输块中发送单个分组。不管传输速率如何，这将导致 32 比特的恒定开销。64 千比特/秒与 128 千比特/秒信道的通过量将分别增加 5.5%与

8.3%。将认识到：只在信道是无差错时才出现如此大的通过量增益。否则，通过量增益将因为重发的分组大小将更大而更小，从而降低通过量。

图 3E 表示根据第一实施例的分组发送/重发的一种方法。在步骤 300，确定传输速率，并在步骤 310，确定分组大小，例如 I 部分的大小。应明白：也能以相反的顺序执行这些步骤。在步骤 320，调整分组大小，以使整个分组能适合单个传输块。在步骤 330，给此分组（或一些分组，如果在步骤 320 将此分组划分为较小分组）分配一个顺序号。随后，在步骤 340 发送此分组（一些分组）。对于发送与重发的每个分组重复此方法。

根据本发明的第二实施例，除了根据每个分组中的信息量和传输速率给分组分配顺序号之外，还可以调整分组大小以使通过量最大。根据此实施例，例如，处理器中的连续测量算法可以评估当前的分组大小是否最适合当前信道情况。例如，此算法测量某一时间周期期间发送实体上重发分组的数量，并使用这些测量值来调整分组大小以获得最大通过量。确定此通过量对于较大/较小分组是否是较高的。如果是这种情况的话，则相应地增加/减少此分组大小。

根据此实施例，增加系统的通过量，并且每个 ARQ 实体能根据当前信道情况使它自己的通过量最大，能以常规方式执行通过量测量。例如，假定使用的一组速率是 {32, 64, 128} 千比特/秒，并假定使用的一组分组大小是 {320, 640 与 1280} 比特。也假定一个分组的开销是 40 比特，还假定发送实体以 128 千比特/秒的速率发送并使用包括此开销的 640 比特的分组大小。在时间周期 T 期间，此发送实体发送包括 3 个重发分组的总数为 100 的分组。能计算表示为 P 的通过量作为在时间周期 T 期间成功发送的有用比特的数量。其他分组大小的差错数量能预测为相同的，即，对于 200 个大小为 320 比特的分组和 50 个大小为 1280 比特的分组具有 3 个差错。各个分组大小的通过量 P 的值是能计算为：

$$P_{320} = (320 - 40) * (2 * 100 - 3) / T = 55160 / T$$

$$P_{640} = (640 - 40) * (100 - 3) / T = 58200 / T$$

$$P_{1280} = (1280 - 40) * (100 / 2 - 3) / T = 58280 / T$$

根据这些计算，此通过量对于大小为 1280 比特的分组 P_{1280} 是最

大的。因而，为使通过量最大，发送实体在以 128 千比特/秒的速率发送时能将分组大小调整为 1280 比特。

图 3F 表示根据第二实施例用于使通过量最大的一种方法。此方法可以作为图 3E 中的步骤 320 的一部分来执行。在步骤 322，确定通过量。在步骤 324，确定使此通过量最大的分组大小。在步骤 326，调整分组大小以使通过量最大。

为了检测差错，给这些分组附加上 CRC。例如，每个传输块中可以包括一个 CRC。在这种情况下，对于整个传输块的检错，只有一个 CRC。当传输块包括不止一个的分组时，给每个分组分配一个不同的顺序号，在发送实体与接收实体之间必须具有附加的信令来通知接收实体：不同的分组位于此传输块中的哪里。为此，在一个块标题内能包括一个块内的分组的顺序号，并且如果需要的话，也能包括长度指示符，它将对应于每个特定顺序号的 I 部分的长度通知接收实体。利用此方法，在接收实体上能对每个传输块进行 CRC 检查。

可选择地，CRC 可以包括在分配有不同的顺序号的每个分组中。例如，参见图 3C，当在第 12 帧中发送这些分组时，每个分组能保持有一个 CRC 值，即，在此传输块内能发送 3 个 CRC。这要求接收实体在进行 CRC 检查时要作一些额外处理，这将在下面具体进行描述。

接收实体在接收到此块时不知道整个传输块是如何利用 CRC 来保护的。此接收实体只知道被表示为 L 的整个传输块的长度。一个分组的大小称为 A_i ，其中 $i=1, \dots, M$ ，并且 M 是一个有限值，即，具有有限数量的分组大小。可以按递增顺序（即， $A_1 < A_2 < \dots < A_M$ ）将这些分组大小进行组合。对于长度为 L 的块，对于每个块中有限数量的分组 A_i ，下面的规则应用于每个传输块：

$$L = \sum A_i \quad (3)$$

分组放置在特定传输块中的顺序是任意的。例如，参见图 3C，能在此块中以 $3! = 6$ 种不同的方式在第 12 帧中安排这些分组：{6, 11, 12}，{6, 12, 11}，{11, 6, 12}，{11, 12, 6}，{12, 6, 11} 和 {12, 11, 6}。

假定一个传输块具有 10 个单位的长度 L，并且根据所使用的速率组，不同可能的分组大小的组为 {1, 2, 4, 5, 6, 10}，则此传输块能包括不同大小的分组的不同组合，例如，[2, 2, 2, 2, 2]，[5, 2, 2, 1]，[10]，

[6, 4], 等等。

根据一个示例性实施例, 从具有可能的大小的组内的最小尺寸的分组的 CRC-OK (CRC 通过) 的检查来开始此 CRC 检查。CRC-OK 表示在接收的分组中未检测到差错。如果具有最小尺寸的分组的检查结果不是 CRC-OK, 则检查下一分组大小, 等等, 直至已检查最大的分组大小。如果在检查最大可能的分组大小之后未获得 CRC-OK, 则在此数据块中最小分组大小的末尾之后 (即, 在 A_1 之后) 开始重复此过程, 并继续直至此块的末尾。在分组 A_1 的末尾之后至此传输块的末尾, 只检查具有适合于剩余子块的大小的分组。如果此第一子块未能导致得到 CRC-OK, 则此开始位置移到第二最小分组大小 A_2 的末尾后面。和前面一样检查此传输块的其余子块。当获得 CRC-OK 时, 释放此子块, 并且又从最小分组大小开始新的查找, 并继续直至检查完整个数据块。

此处理表示在图 4A 中, 图 4A 表示如何能实施传输块的 CRC 解码算法的流程图。图 4A 中使用的表示方式基于 C 编程语言语法, 但本发明并不局限于 C 编程实施。

此处理开始于步骤 400, 在步骤 400 例如由接收实体中的处理器发出 CRC 解码的指令。在步骤 405, 对分组大小变量和可调变量进行初始化。在步骤 410, 变量“first (第一)”设置为等于分组大小 $A[j]$ 。接着, 在步骤 420, 处理变量“i”设置为等于 0, 并且 CRC 寄存器设置为一个预定值, 例如 1 或 0。接下来, 在步骤 425, i 递增 1, 并确定 i 是否小于或等于 M, 其中 M 是可能的分组大小的数字。在步骤 430, 确定第 i 分组大小是否小于 $L - \text{first}(\text{第一})$, 即, 除了具有该变量 first 的长度的分组之外, 具有大小为 $A[i]$ 的第 i 分组是否适合此传输块的子块。如果这是真的话, 在步骤 435, 从变量 first 加上分组大小 $A[i - 1]$ 的偏差开始, 确定对于具有分组大小 $A[i]$ 减去 $A[i - 1]$ 的大小的传输块的子块是否产生了 CRC-OK。如果否的话, 此处理返回到步骤 425。如果是的话, 则在步骤 440 释放此子块, 并在步骤 445 将该变量 first 递增第 i 分组的大小。自此, 此处理返回到步骤 420。

如果在步骤 425 确定 $++i$ 不小于或等于 M, 则在步骤 450 确定 first 是否小于 L。如果是的话, 则在步骤 455 从该变量 first 加上分组大小 $A[i - 1]$ 的偏差开始, 确定对于具有 $L - \text{first} - A[i - 1]$ 的大小的传输块的子块是否产生了 CRC OK。如果是的话, 则在步骤 460 释放此子块。

如果在步骤 450 确定变量 first 不小于 L 或在步骤 455 未得到 CRC OK, 则在步骤 465 确定 first 是否等于 L. 如果否的话, 在步骤 470 则 j++, 即将 j 递增 1, 并且此处理返回到步骤 410. 如果在步骤 465 确定 first 等于 L, 即, 已经到达此输出块的末尾, 或如果在步骤 460 释放此子块, 5 则此处理在步骤 480 结束.

能使用线性反馈移位寄存器 (LFSR) 或利用硬件或软件来进行 CRC 检查. 要作 CRC OK 检查的比特流可以通过此 LFSR 来进行. 步骤 435 与 455 中的函数 CRC_Check(CRC 检查)(arg1, arg2)对应于在通过 LFSR 运行的数据单位流的位置 arg1 开始检查 arg2 数据单位的数量. 此数 10 据单位例如可以是比特或字节. 如果最后得到的寄存值是有效的, 则函数 CRC - CHECK () 返回 TRUE (真), 而如果这些值是无效的, 则此函数返回 FALSE (假).

作为 CRC 解码如何工作的一个示例, 考虑一组分组大小 {320, 480, 640} 比特和具有大小 L 等于 640 比特的发送的块的大小, 如图 4B 所示. 15 假定要检查的传输块包括每个大小为 320 比特的两个分组并且第一分组是有错误的, 即, 将不能导致有效的 CRC 检查. 图 4C 表示根据这些假定的 CRC 解码步骤.

参见图 4C, 在步骤 405 将处理变量 j 设置为 0, 并将分组大小变量 A[0]、A[1]、A[2]、A[3]与 A[4]分别设置为 0、320、480、640 与 640. 在步骤 410, 将处理变量 first 设置为 0. 在步骤 420, 将处理 20 变量 i 设置为 0, 并将 CRC 寄存器复位.

在步骤 425, 将 i 递增 1, 并确定 i (1) 小于或等于 M (3). 在步骤 430, 确定 A[1] (320) 小于 L-first (640). 接着, 在步骤 435, 检查此块的前 320 比特, 并确定对于被检查的这 320 比特未获得 CRC OK. 25 然后, 重复步骤 425, 并确定 i+1 (2) 小于或等于 M (3). 在步骤 430, 确定 A[2] (480) 小于 L-first (640). 在步骤 435, 检查相对此块的开头偏离了 320 比特的包含 160 比特的一个子块, 并确定: 对于检查的此块的总共 480 比特未获得 CRC-OK. 随后, 重复步骤 425, 并确定 A[3] (3) 小于或等于 M (3). 在步骤 430, 确定 A[3] (640) 不小于 L-first 30 (640). 因而, 执行步骤 450 并确定 first (0) 小于 L (640). 接下来, 在步骤 455, 检查相对此块的开头偏离了 480 比特的包含 160 比特的一个子块, 并确定: 对于检查的此块的总共 640 比特, 未获得 CRC-OK.

随后，在步骤 465，确定 $\text{first}(0)$ 不等于 $L(640)$ ，并在步骤 470 将 j 设置为等于 1。

接着，重复步骤 410，并将 first 设置为等于 320。接下来，在步骤 420，将 i 复位为 0，并将 CRC 寄存器复位。然后，在步骤 425，将 i 递增 1，并随后确定 i 小于或等于 $M(3)$ 。在步骤 430，确定 $A[1]$ 不小于 $L-\text{first}(320)$ ，并在步骤 450 确定 $\text{first}(320)$ 小于 $L(640)$ 。在步骤 455，确定对于此块的开头偏离了 320 比特的包含 320 比特的一个子块获得 CRC OK。在步骤 460 释放此子块，并且此处理在步骤 480 结束。

10 根据第三实施例，在上述的实施例具有低开销和增加通过量的优点的同时，可以调整分组大小，以使每个传输块只包含一个 RLCP 来支持类型 II 与类型 III 混合 ARQ 与软组合。这允许接收实体将新近发送的信息与以前发送的信息加以组合以增加成功解码的概率。

15 根据第三实施例，为了支持类型 II 与类型 III 混合 ARQ 与软组合，只针对原始传输调整分组大小，并以原始传输速率执行同一分组的所有重发。它的原因是：如果允许在重发时改变速率，则在一个传输块内重发的信息将不同于在一个传输块内原始发送的信息，这将使类型 II 与类型 III 混合 ARQ 与软组合相当复杂。

20 图 5A 表示根据第三实施例的分组顺序编号。根据第三实施例，至少对于分组的原始传输，顺序编号类似于第一与第二实施例。在图 5A 中，第一行中的分组具有比最小的 I 部分大四倍的信息部分。分别给这些分组分配顺序号 4、8、12 与 16。第二行中的分组表示实际上如何发送这些分组。

25 例如，假定第一行中分配有顺序号 $NS = 4$ 的分组将在只能以能发送第一行中的分组的速率的 $1/2$ 速率发送分组的信道上进行发送。于是能将分配有 $NS = 4$ 的原始分组划分为具有编号为 $NS = 2$ 与 $NS = 4$ 的两个分组。假定第一行中分配有顺序号 $NS = 8$ 与 $NS = 12$ 的分组将要在能以与第一行中的分组相同的速率发送分组的信道上进行发送。于是不必为了传输而划分这些分组。假定在第一行中分配有顺序号 $NS = 16$ 的分组将要在只能以能发送第一行中的分组的速率的 $1/4$ 速率发送分组的信道上进行发送。于是分配有 $NS = 16$ 的原始分组能划分为编号为 $NS=13$ 、 $NS = 14$ 、 $NS = 15$ 与和 $NS = 16$ 的四个分组。

30

图 5B 表示根据第三实施例的示例性发送/重发。在图 5B 中，在每个时帧中发送一个发送分组。箭头表示速率变化。虽然已参照软组合进行描述，但将明白这能扩展到类型 II 与类型 III 混合 ARQ 方案以及其他可变的冗余方案。

5 在图 5B 中，速率组为 $\{R, 2R, 3R, 4R\}$ 。在最大速率上，分组顺序号为 $\{4, 8, 12, 16\}$ 。在发送开始时，速率为 $2R$ ，这是最大速率的一半。因此，将第一分组划分为编号为 2 与 4 的两个分组。在第一帧中以速率 $2R$ 发送具有顺序号 2 的分组。在下一帧中，发送具有顺序号为 4 的分组（比 2 大 2 的步长）。在具有顺序号为 2 的接收分组中检测到一
10 个差错，并在第三帧中将 NAK 信号与顺序号 2 一起发送给发送实体。而且，在第 3 帧中，速率变为 $4R$ ，并发送分配有顺序号 8（比 4 大 4 的步长）的分组。

在第四帧中，具有顺序号 2 的分组以是此分组的原始传输速率的速率 $2R$ 进行重发。此发送与接收实体中以前的发送进行软结合，并成
15 功解码此分组。

在第五帧中，发送一个分配有顺序号 12 的分组（比 8 大 4 的步长）。随后，速率变为 R ，且此速率对于接下来的 3 个帧保持为固定。因而，将具有顺序号为 16 的分组划分为分配有顺序号为 13、14、15 与 16 的
20 四个分组，分别在第六、第七与第八帧中发送分配有连续顺序号为 13、14 与 15 的分组。在具有顺序号为 12 的接收分组中检测到一个差错，并在第八帧期间将 NAK 信号与顺序号 12 一起发送给发送实体。在第九帧中，一个分配有顺序号为 12 的分组以其原始传输速率 $4R$ 进行重发。接下来，在第十帧中，发送一个分配有顺序号为 16（比 15 大 1 的步长）的分组。

25 图 5C 表示根据第三实施例用于分组发送/重发的一种方法。此方法开始于步骤 500，在步骤 500 确定是否正在重发一个分组。如果否的话，则此处理进行到步骤 510、520、530 与 540，分别以类似于图 3E 中所示的步骤 300-340 的方式，确定传输速率、确定此分组大小、调整分组大小、分配一个（一些）顺序号、和发送此（这些）分组。如
30 果在步骤 500 确定正在重发此分组，则此处理进行到步骤 560，在步骤 560 确定原始传输速率，在步骤 570 以此速率重发此分组。

根据第三实施例的技术具有许多优点，包括低开销，这是因为对

于每个分组只要求一个标题与一个 CRC，这也使之更容易执行软组合和类型 II 与类型 III 混合 ARQ。过多的分组序号可以导致中间序号保持为未使用的情况，但这在开销方面是非常有限的，例如，两个额外的比特用于等于 4 的顺序编号步长。

- 5 例如，在收发信机的处理器中能执行用于上述的序列编号、分组大小调整和传输速率变化的所有处理。

虽然以 CDMA 系统的应用进行描述，但本领域技术人员将明白，本发明能以例如 TDMA 系统或混合 CDMA - DMA 系统的其它特定形式来实施而不背离其本质特征。上述的实施例因此应认为是示例性的而不是限制性的。

10

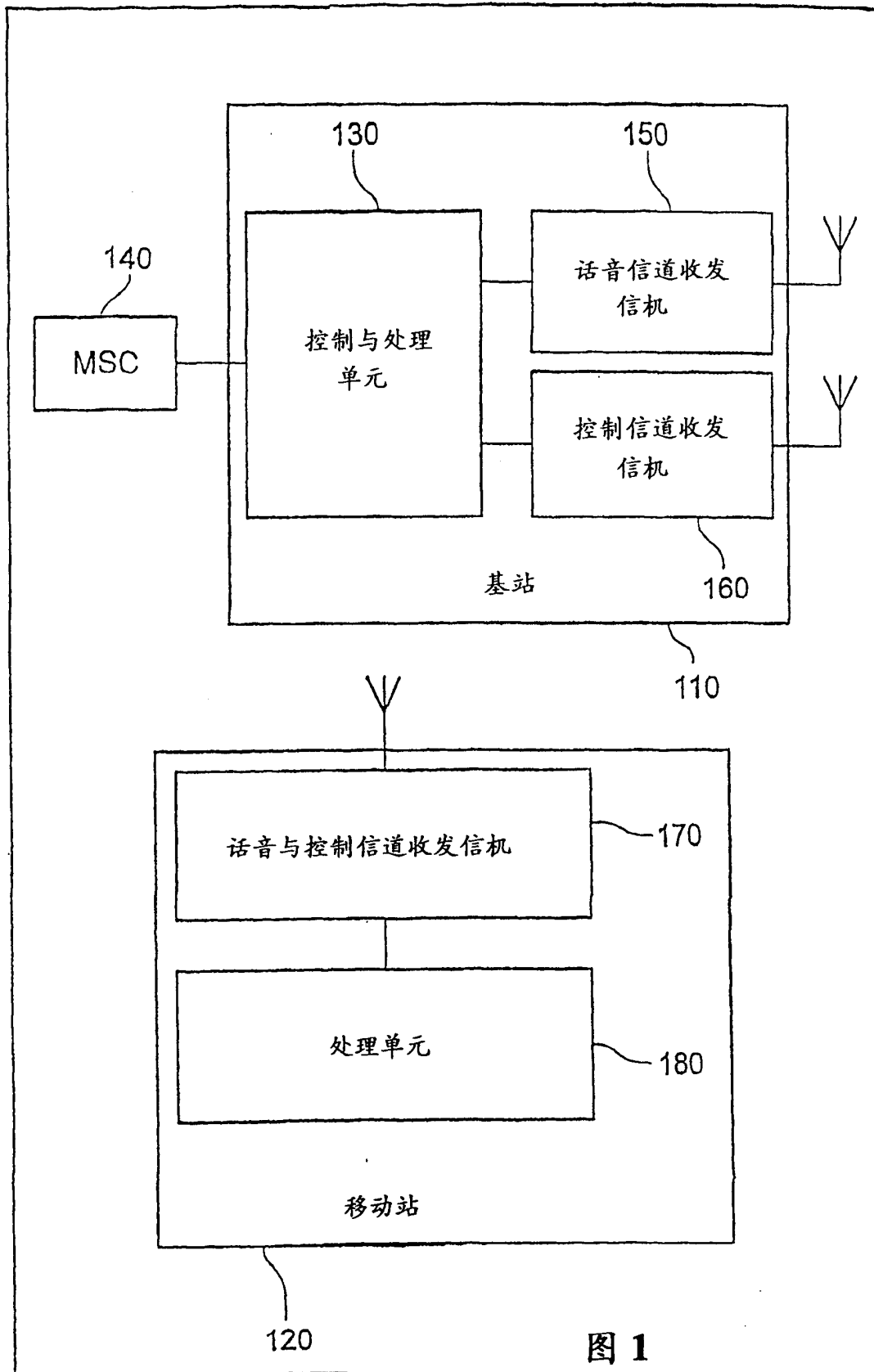


图 1
现有技术

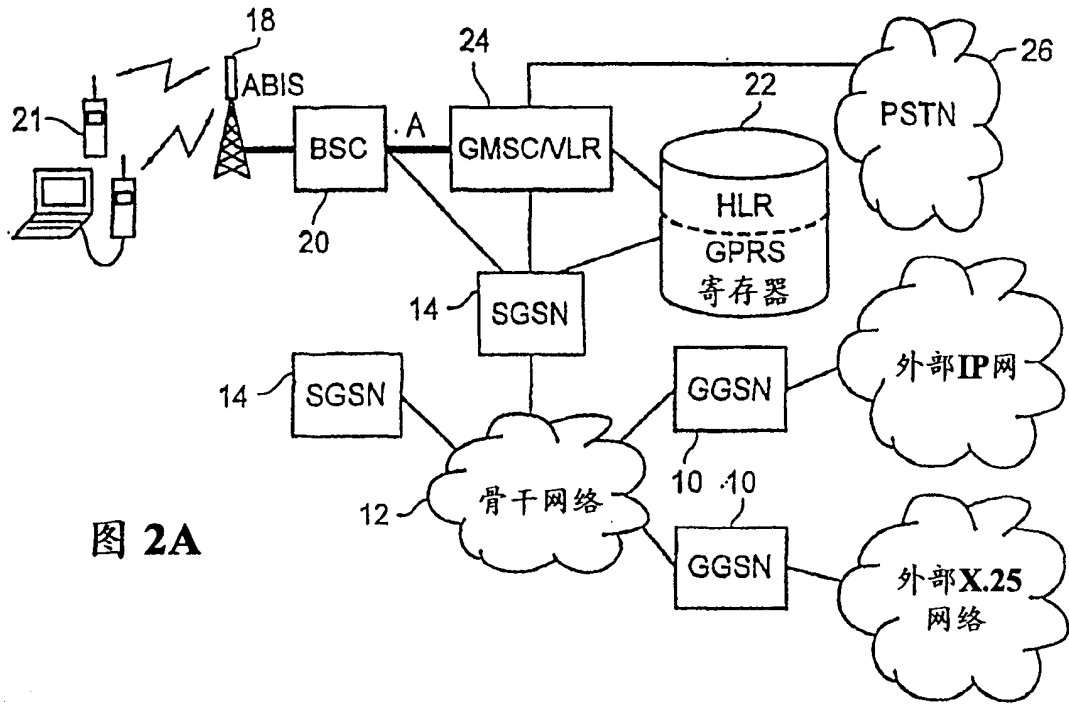


图 2A

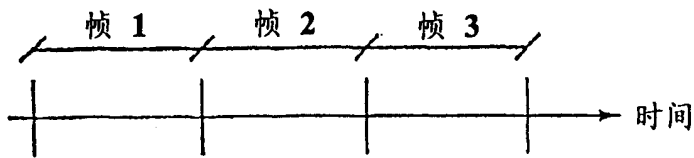


图 2B

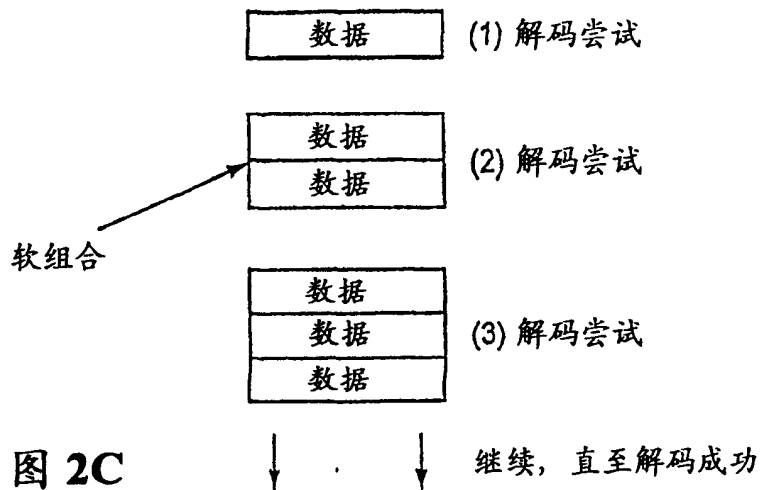


图 2C

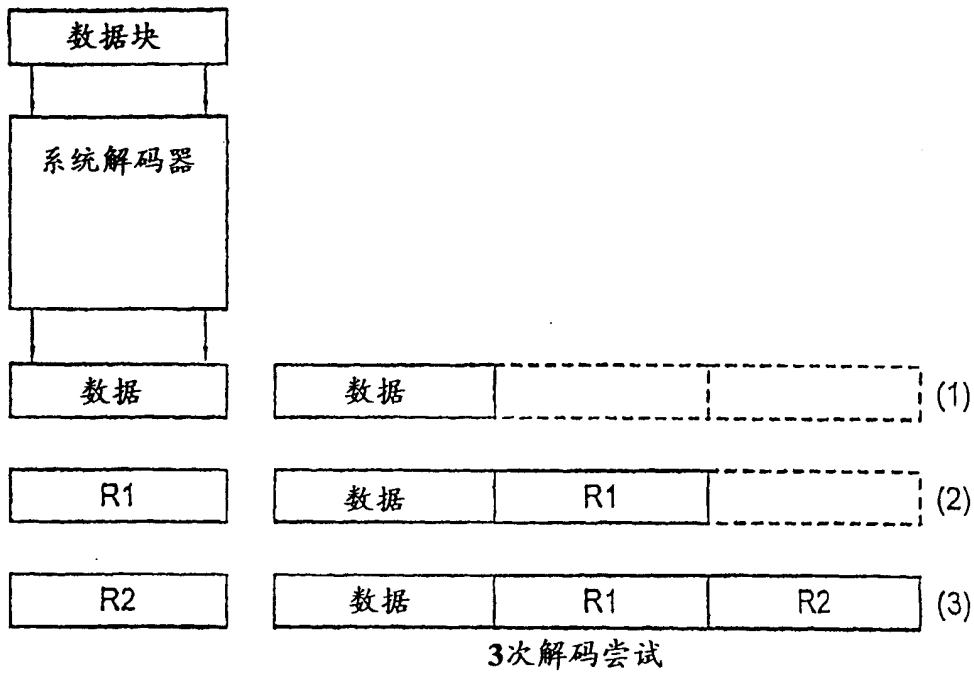


图 2D

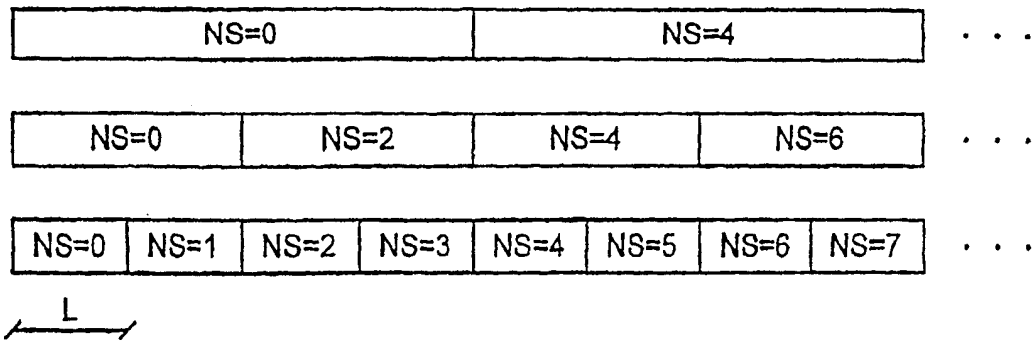


图 3A

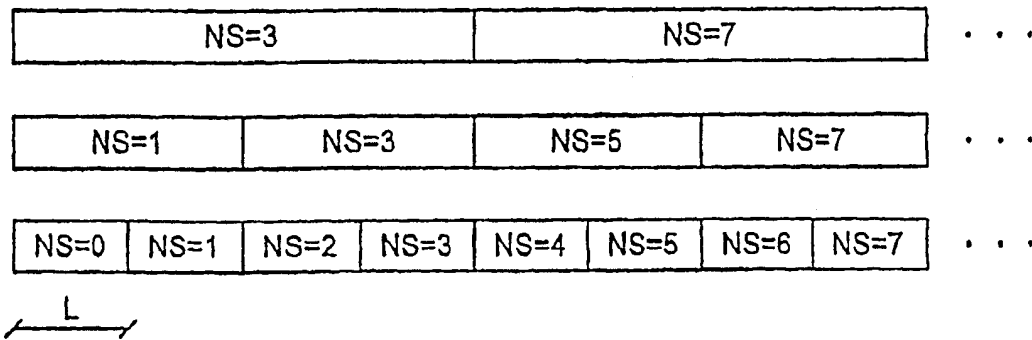


图 3B

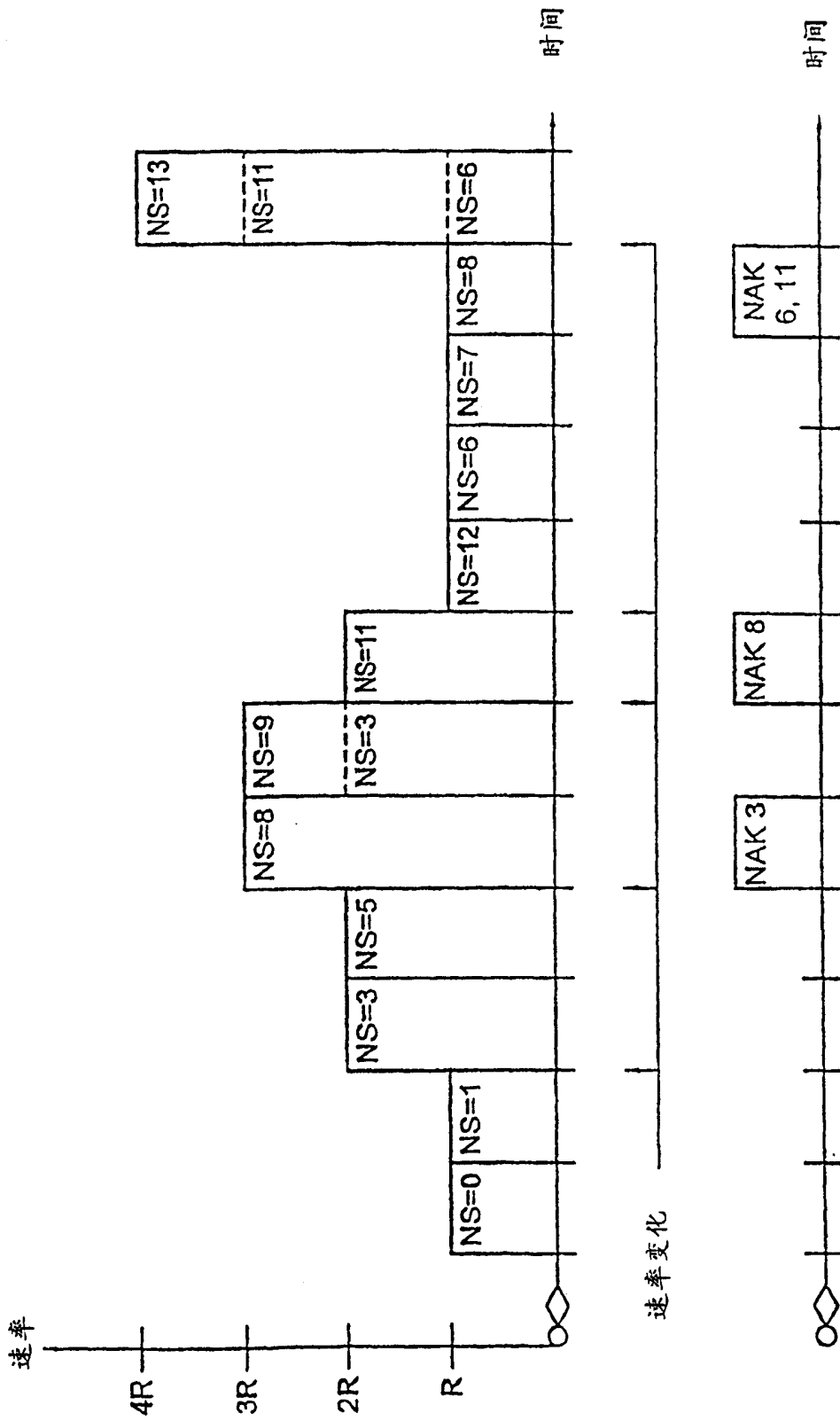


图 3C

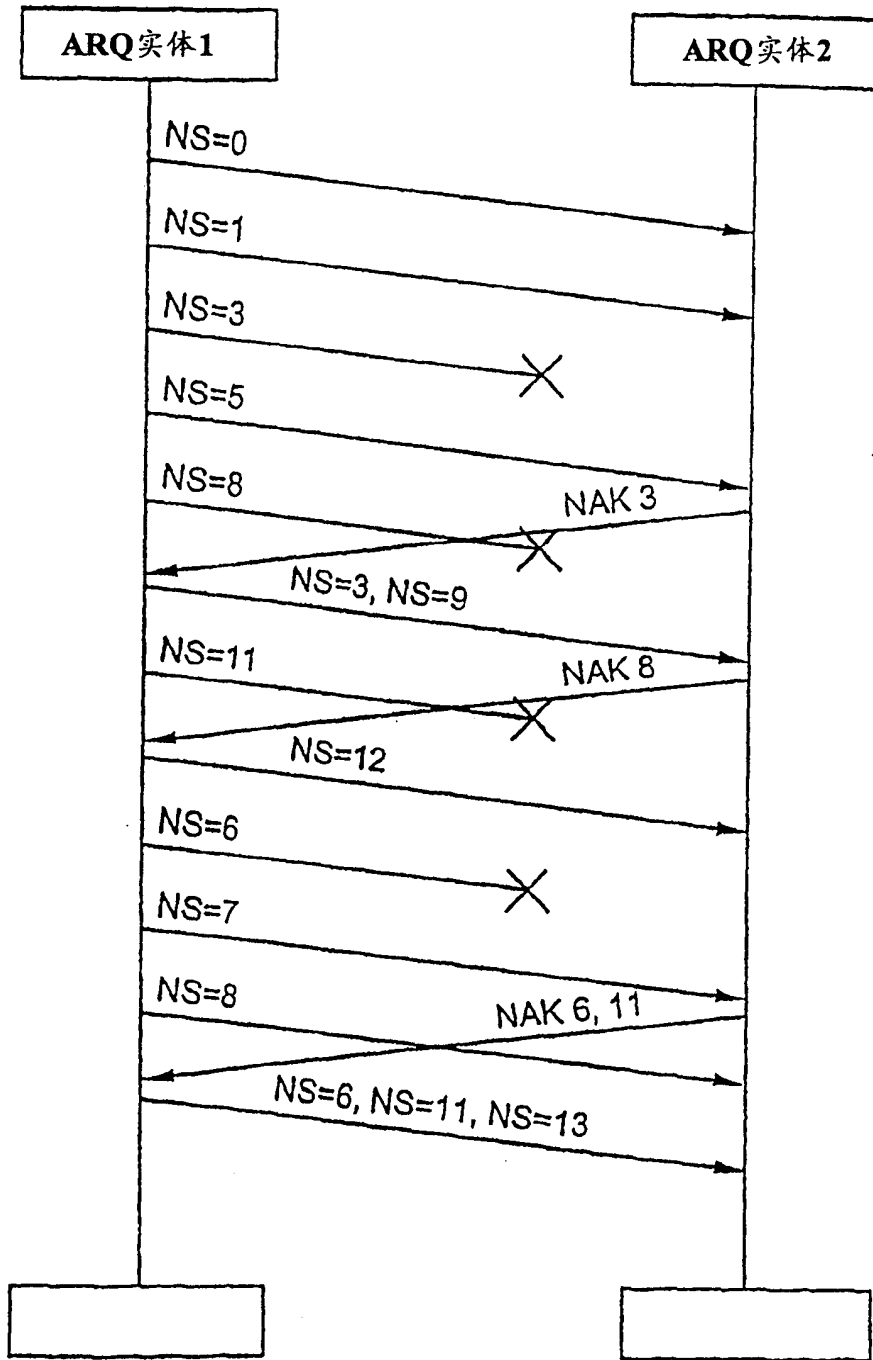


图 3D

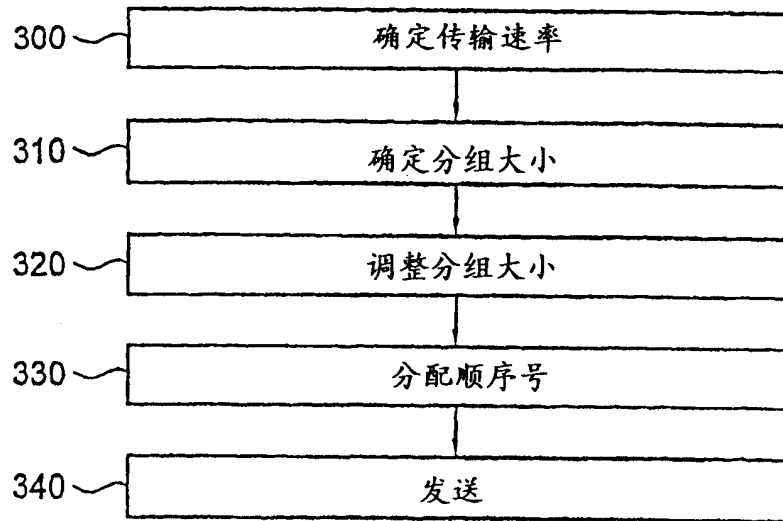


图 3E

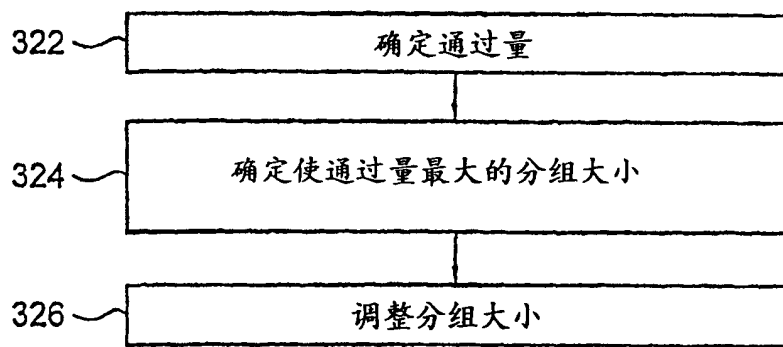


图 3F

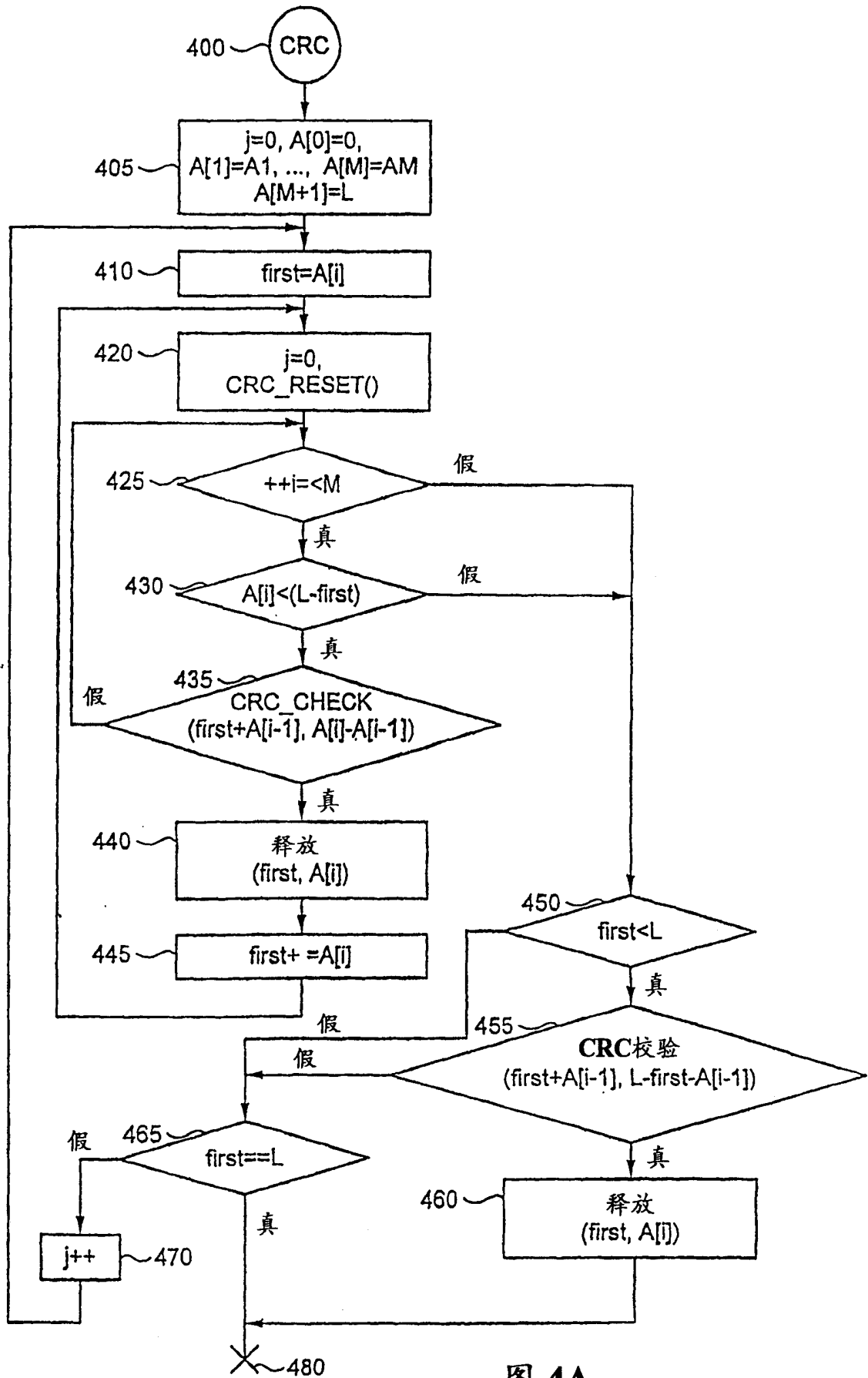
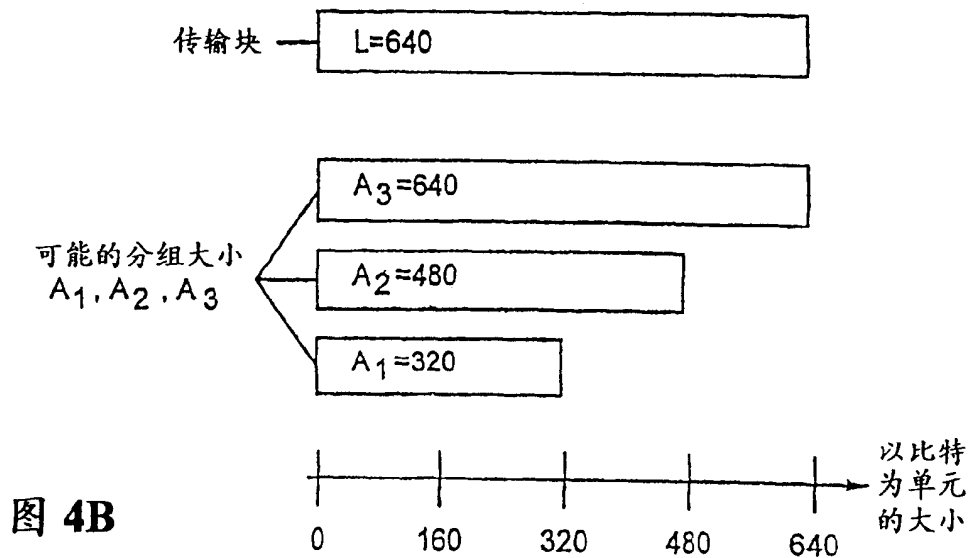


图 4A

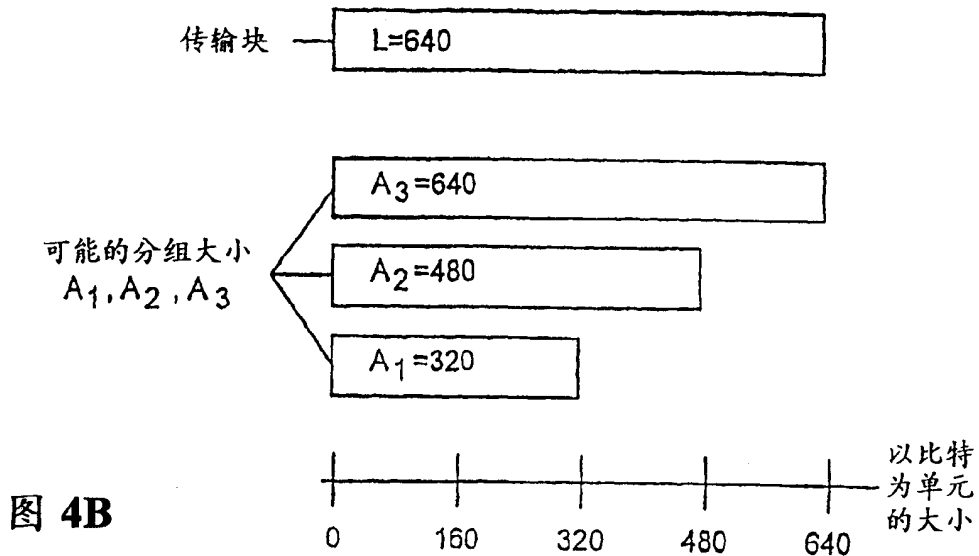


```

405: j=0, A[0]=0, A[1]=320, A[2]=480, A[3]=640, A[4]=640
410: first=0
420: i=0, CRC_RESET()
425: ? 1=<3 => 真
430: ? 320=<640 => 真
435: ? CRC_CHECK(0,320) => 假/*将前320个比特馈送给LFSR*/
425: ? 2=<3 => 真
430: ? 480=<640 => 真
435: ? CRC_CHECK(320,160) => 假/*将正好在前320个之后的160个比特馈送给LFSR*/
425: ? 3=<3 => 真
430: ? 640<640 => 假
450: ? 0<640 => 真
455: ? CRC_CHECK(480,160) => 假/*将正好位于前480个之后的160个比特馈送给LFSR*/
465: ? 0==640 => 假
470: j=1
410: first=320
420: i=0, CRC_RESET()
425: ? 1=<3 => 真
430: ? 320<320 => 假
450: ? 320<640 => 真
455: ? CRC_CHECK(320,320) => 真/*将正好位于前320个比特的320个比特馈送给LFSR*/
460: RELEASE(320,320)/*即, 释放第二分组, 因为此分组无差错*/
480: /* 结束*/

```

图 4C



```

405: j=0, A[0]=0, A[1]=320, A[2]=480, A[3]=640, A[4]=640
410: FIRST=0
420: i=0, CRC_RESET()
425: ? 1=<3 => 真
430: ? 320=<640 => 真
435: ? CRC_CHECK(0,320) =>假/*将前320个比特馈送给LFSR*/
425: ? 2=<3 => 真
430: ? 480=<640 => 真
435: ? CRC_CHECK(320,160) =>假/*将正好在前320个之后的160个比特馈送给LFSR*/
425: ? 3=<3 => 真
430: ? 640<640 => 假
450: ? 0<640 => 真
455: ? CRC_CHECK(480,160) =>假/*将正好位于前480个之后的160个比特馈送给LFSR*/
465: ? 0==640 => 假
470: j=1
410: FIRST=320
420: i=0, CRC_RESET()
425: ? 1=<3 => 真
430: ? 320<320 => 假
450: ? 320<640 => 真
455: ? CRC_CHECK(320,320) =>真/*将正好位于前320个之后的320个比特馈送给LFSR*/
460: RELEASE(320,320)/*即, 释放第二分组, 因为这是无差错的*/
480: /* 结束*/

```

图 4C

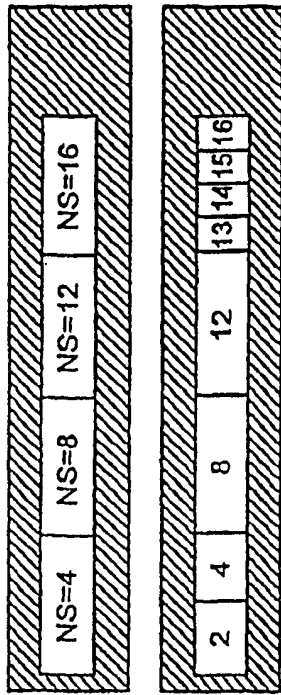
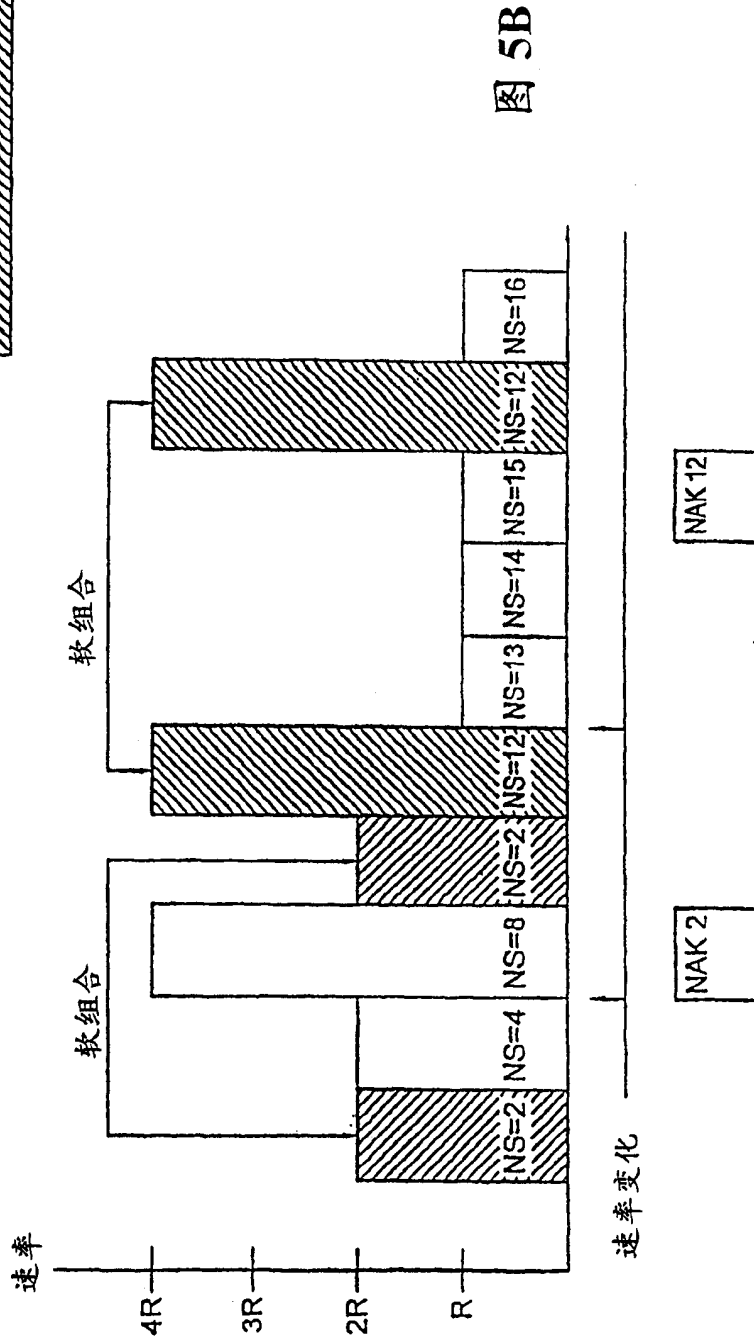


图 5A



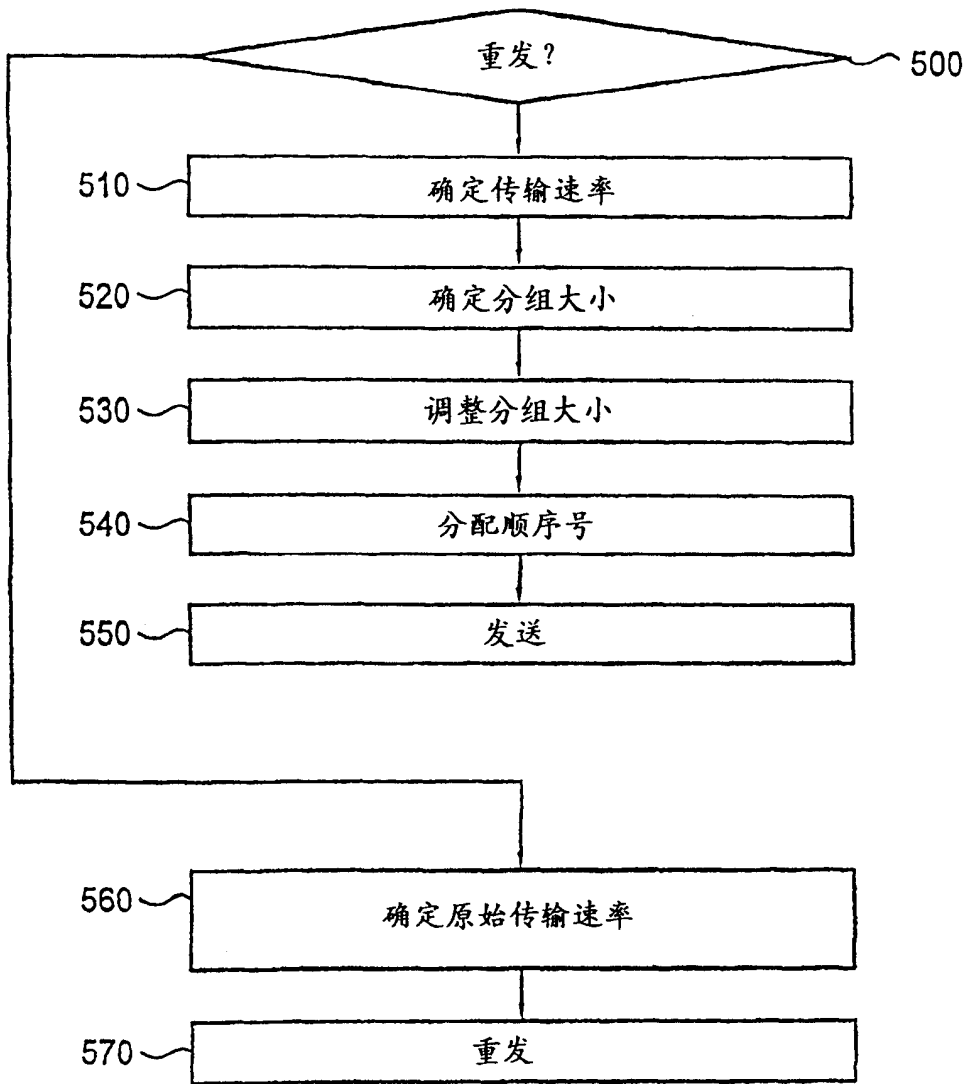


图 5C