

1. 一种设备,包括:

缓冲存储器存储单元,其被配置成存储数据的分组;和

用于缓冲自适应的装置,其被配置成将接收到的数据的分组的顺序标识符与再现顺序进行比较,所述用于缓冲自适应的装置被配置成选择接收到的分组的至少一个以用于再现。

2. 如权利要求 1 所述的设备,其中,所述用于缓冲自适应的装置进一步被配置成如果没有接收到具有预期再现顺序标识符的预期分组,则启动擦除的再现,其中,所述擦除在所述预期分组的预期再现时刻被再现。

3. 如权利要求 1 所述的设备,其中,所述用于缓冲自适应的装置进一步被配置成:

确定下一预期再现时刻;

将接收到的分组的顺序标识符与包括所述预期再现顺序标识符的多个顺序标识符进行比较;和

如果没有接收到所述预期分组和下一顺序的分组,则启动擦除的再现,其中,所述擦除在所述预期分组的下一预期再现时刻被再现。

用于语音 IP 传输的自适应去抖动缓冲器

[0001] 本申请是 2005 年 8 月 30 日提交的中国专利申请号为 200580036630.3 (国际申请号为 PCT/US2005/030894) 的题为“用于语音 IP 传输的自适应去抖动缓冲器”的国际申请的分案。

[0002] 本专利申请要求 2004 年 8 月 30 日提交的标题为“Adaptive De-Jitter Buffer For Voice Over IP for Packet Switched Communications”的美国临时申请第 60/606,036 号的优先权,该临时申请已被转让给其受让人。

技术领域

[0003] 本发明涉及无线通信系统,并具体涉及用于分组交换通信的语音因特网协议 (VoIP) 的自适应去抖动缓冲器。本发明适用于可能会丢失分组的任何系统。

背景技术

[0004] 在通信系统中,可将分组的端对端延迟定义为从该分组在信号源处被生成的时刻到该分组到达其目的地的时刻所经过的时间。在分组交换通信系统中,分组从信号源传播到目的地的延迟可能会取决于包括但不限于信道条件和网络负载的各种工作条件而变化。信道条件是指无线链路的质量。确定无线链路质量的一些因素是信号强度、移动速度和/或物理障碍。

[0005] 端对端延迟包括在网络和分组所经过的各种元件中引入的延迟。许多因素都能影响端对端延迟。端对端延迟中的变化被称为抖动。抖动会导致在分组不再有用之后才接收到分组。例如,在诸如语音的低等待时间应用中,如果分组被接收到太晚,则它可能会被接收器丢弃。这样的状况导致通信质量的恶化。

附图说明

[0006] 图 1 是现有技术通信系统的框图,其中接入终端包括去抖动缓冲器;

[0007] 图 2 示出了现有技术的去抖动缓冲器;

[0008] 图 3 是示出导致“下溢”的分组的传输、接收和再现的时序图;

[0009] 图 4A 和 4B 是示出两种情形下的最佳去抖动缓冲器长度的计算的时序图;

[0010] 图 5 是示出由延迟分组引起的“下溢”的形成过程的时序图;

[0011] 图 6 是示出目标去抖动缓冲器长度的计算的流程图;

[0012] 图 7A 是示出第一情形下的分组的传输的时序图;

[0013] 图 7B 是示出没有采用去抖动缓冲器自适应的分组的接收的时序图;

[0014] 图 7C 是示出采用去抖动缓冲器自适应的分组的接收的时序图,其中接收器可以在分组的预期时刻之后接收到该分组。

[0015] 图 8A 是示出允许接收器在分组的预期时刻之后接收到该分组的隐式缓冲器自适应的一个实例的流程图;

[0016] 图 8B 是自适应去抖动缓冲器的操作模式的状态图;

- [0017] 图 9 是示出根据另一实例的去抖动缓冲器自适应的应用的时序图；
- [0018] 图 10 是示出根据一个实例的话音突峰中的语音信息的传输的示图，其中去抖动缓冲器延迟不足以避免数据碰撞；
- [0019] 图 11 是含有自适应去抖动缓冲器的通信系统的框图；
- [0020] 图 12 是包括自适应去抖动缓冲器和时间规整单元的接收器的一部分的框图；
- [0021] 图 13A 示出了包括压缩和扩展阈值的自适应去抖动缓冲器的一个实例；
- [0022] 图 13B 示出了包括多个压缩和扩展阈值的自适应去抖动缓冲器的一个实例；
- [0023] 图 14 是示出对具有不同延迟的分组的接收进行时间规整的过程的时序图；
- [0024] 图 15 是示出以下实例的时序图：i) 语音片段的静音部分的压缩；和 ii) 语音片段的静音部分的扩展；
- [0025] 图 16 是示出语音信号的时序图，其中语音信号的一些部分会重复；
- [0026] 图 17A 是示出语音片段的示图，其中被称为 RWindowSize 的用于重叠相加 (add-overlap) 操作的参考窗口中的 PCM 样本的数目被标识，并且其中被称为 Segment 的目标片段大小或期望的片段大小被标识；
- [0027] 图 17B 是示出根据一个实例的应用重叠相加操作来压缩语音片段的方法的示图；
- [0028] 图 18A 是示出多个语音片段的示图，其中被称为 RWindowSize 的用于重叠相加操作的参考窗口中的 PCM 样本的数目被标识，并且其中被称为 Segment 的目标片段大小或期望的片段大小被标识以为扩展当前语音片段作准备；
- [0029] 图 18B 是示出根据一个实例的应用重叠相加操作来扩展语音样本的方法的示图；
- [0030] 图 18C 是示出根据可选实例的应用操作来扩展语音样本的方法的示图；
- [0031] 图 19 是示出对分组进行扩展以便为混合 ARQ 重传情况下延迟的分组和失序到达的分组的到达留出余地的方法的示图；
- [0032] 图 20 是示出两个用户之间的会话的时间线的示图；
- [0033] 图 21 是示出根据一个实例的在语音突峰开头处的增强的流程图；
- [0034] 图 22 是示出根据可选实例的在语音突峰开头处的增强的示图；
- [0035] 图 23 是示出语音突峰结尾的增强的示图；
- [0036] 图 24 是示出根据一个实例的在语音突峰结尾处的增强的流程图；
- [0037] 图 25 是示出现有技术的去抖动缓冲器和解码器系统的操作的示图，其中去抖动缓冲器以规则的时间间隔向解码器传递分组；
- [0038] 图 26 是示出根据一个实例的自适应去抖动缓冲器和解码器的操作的示图，其中自适应去抖动缓冲器以不规则的时间间隔向解码器传递分组；
- [0039] 图 27 是示出根据一个实例的包括自适应去抖动缓冲器和时间规整 (time wrap) 控制单元的接入终端 (AT) 的框图；
- [0040] 图 28 示出了根据一个实例的包括自适应去抖动缓冲器并适于对分组进行时间规整的接收器的一部分；
- [0041] 图 29 示出了根据另一实例的包括自适应去抖动缓冲器并适于对分组进行时间规整的接收器的可选实例；
- [0042] 图 30 是示出根据一个实例的包括自适应去抖动缓冲器并适于对分组进行时间规整的接收器的一个实例中的解码器中的调度器的一个实例的流程图；

- [0043] 图 31 是示出接收器的一个实例中的音频接口单元中的调度器的流程图；
- [0044] 图 32 示出了在解码器外部计算调度方案的时间规整单元；
- [0045] 图 33 示出了在解码器中的时间规整单元中计算调度方案的时间规整单元。

具体实施方式

[0046] 在分组交换系统中,数据被形成为分组并通过网络被路由。每个分组基于分组内(典型为信头中)包含的分配地址而被发送到网络中的目的地。分组的端对端延迟,或者分组在网络内从第一用户或“发送者”传播到第二用户或“接收者”所花费的时间,取决于以下因素而变化:信道条件,网络负载,系统的服务质量(QoS)能力,以及除这些因素之外的竞争资源的其它流。应注意,为了清楚起见,下面的讨论描述支持分组数据通信的扩频通信系统,其包括但不限于码分多址(CDMA)系统、正交频分多址(OFDMA)系统、宽带码分多址(W-CDMA)系统、全球移动通信系统(GSM)、支持诸如 802.11(A、B、G)、802.16 等的 IEEE 标准的系统。

[0047] 在无线通信系统中,每个分组可能会遭受与属于相同流的其它分组所经历的延迟不同的从源到目的地的延迟。延迟中的这种变化被称为“抖动”。抖动对接收器端的应用产生额外的复杂性。如果接收器不校正抖动,则接收到的消息在分组被重新组合时将会失真。一些系统在根据接收到的分组而重建消息时对抖动进行校正。这样的系统包含增加(被称为抖动缓冲器延迟的)等待时间的去抖动缓冲器。当去抖动缓冲器施加固定的较大的去抖动缓冲器延迟时,它可以容纳分组到达时间中的大抖动量;然而,这种使用方式并不高效,因为具有较小延迟的分组也通过使用较大的去抖动缓冲器延迟而被处理,尽管这些分组可能已在之前被处理过。对于这些分组而言,这会导致比通过使用较小的去抖动缓冲器延迟而得到的延迟更大的端对端延迟。

[0048] 为了防止这种情况,含有去抖动缓冲器的 VoIP 系统可以试图适应分组延迟中的变化。例如,去抖动缓冲器可以通过分析分组到达的统计数据来检测分组延迟中的变化。许多去抖动缓冲器实现方案根本不使它们的延迟自适应,并且被配置成具有保守的大延迟。在这种情况下,去抖动缓冲器可能会对分组加入过大的延迟,从而使用户的体验处于次优状态。

[0049] 下面的讨论描述的是通过改变去抖动缓冲器延迟来适应分组延迟行为中的变化的自适应去抖动缓冲器。该去抖动缓冲器利用话音时间规整来增强其跟踪分组的可变延迟的能力。下面的讨论适用于分组化的通信,诸如具有周期性数据传输、低等待时间要求、数据的顺序处理或指定再现速率的通信。特别地,下面的讨论详细描述语音通信,其中数据或者话音和静音源于信号源并被传输到目的地用于再现。原始数据被分组并使用已知的编码方案被编码。在接收器处,为数据的每个分组确定编码方案。在话音通信中,例如,对话音进行的编码的类型不同于对静音进行的编码的类型。这允许通信系统利用包括静音部分的话音的周期性本质。对于话音通信,数据呈现突发性,并且话音内容可能会呈现重复性。由于语音通信的参与者不希望听到延迟,而通信的质量仅允许有限的延迟,所以分组化的话音传输具有低等待时间要求。分组化的话音可能会采取不同的路径来到达接收器,然而,在接收后,分组将以它们的原始顺序被重新编译。因此,接收到的分组化话音被顺序地再现。如果分组在空中传输或者物理层处理过程中被丢失,则该分组无法恢复,但是接收器可以

估计或猜测该分组的内容。另外,话音通信的再现速率具有预定的再现速率或范围。如果再现速率在该范围之外,则接收器处的质量会恶化。本讨论的应用的一个实例是将其应用于话音通信。其它应用可包括视频通信、游戏通信或具有与话音通信类似的特性、规范和/或要求的其它通信。例如,视频通信可能会期望使再现加速或减速。本讨论对于这样的应用可能是合乎要求的。如在本文中提供的那样,自适应去抖动缓冲器可以允许接收器实现系统的抖动要求所指定的服务质量。自适应去抖动缓冲器使目标去抖动缓冲器长度(例如,去抖动缓冲器中存储的数据量)适应于在该自适应去抖动缓冲器处接收到的数据的定时和数据量。此外,自适应去抖动缓冲器使用该去抖动缓冲器的状态或大小(例如,自适应去抖动缓冲器中存储的数据的度量)来确定何时进行时间规整对接收到的数据的处理和再现是有益的。例如,如果数据以低速率到达自适应去抖动缓冲器,则自适应去抖动缓冲器将该信息提供给时间规整单元,从而允许时间规整单元扩展接收到的分组。如果自适应去抖动缓冲器中存储的数据超过阈值,则自适应去抖动缓冲器提醒时间规整单元压缩分组以便有效地跟上到来的数据。应注意,时间规整是在可由通信的应用和类型定义的限度内进行的。例如,在话音通信中,时间规整不应该压缩话音(即,提高音高),以免收听者无法理解通信内容。类似地,时间规整不应该将话音扩展出某一范围之外。在理想情况下,时间规整的范围被定义为使得收听者几乎没有不适或完全没有不适。

[0050] 通信系统

[0051] 图1是示出数字通信系统50的框图。两个接入终端(AT)52和82经由基站(BS)70进行通信。在AT 52内,发射处理单元64将语音数据发送给编码器60,该编码器60对语音数据进行编码和将其分组化并将经分组的数据发送给低层处理单元58。然后为了实现发射,数据被发送给BS 70。BS 70处理接收到的数据并将数据发射到AT 82,其中数据在低层处理单元88被接收。然后数据被提供给去抖动缓冲器86,该去抖动缓冲器86存储数据以便隐藏或减小抖动的影响。数据从去抖动缓冲器86被发送给解码器84,并接着被发送给接收处理单元92。

[0052] 为了从AT 82发射,数据/语音从发射处理单元94被提供给编码器90。低层处理单元88处理用于发射到BS 70的数据。为了在AT 52接收来自BS 70的数据,数据在低层处理单元58被接收。然后数据的分组被发送给去抖动缓冲器56,在这里它们被存储直到达到所需的缓冲器长度或延迟为止。一旦达到该长度或延迟,去抖动缓冲器56就开始向解码器54发送数据。解码器54将经分组的数据转换成语音数据分组并将分组发送给接收处理单元62。在本实例中,AT 52的行为与AT82类似。

[0053] 去抖动缓冲器

[0054] 存储器或去抖动缓冲器被用在诸如上述的AT中,以便隐藏抖动的影响。在一个实例中,自适应去抖动缓冲器被用于诸如VoIP通信的分组交换通信。去抖动缓冲器具有自适应缓冲存储器并使用话音时间规整来增强其跟踪可变延迟和抖动的能力。在该实例中,使去抖动缓冲器的处理与解码器的处理相协调,其中去抖动缓冲器确定对分组进行时间规整的时机或需要,并指示解码器对分组进行时间规整。解码器通过按照去抖动缓冲器的指示对分组进行压缩或扩展,来对分组进行时间规整。

[0055] 图2示出了去抖动缓冲器的一个实例。到来的编码分组被累积和存储在缓冲器中。在一个实例中,缓冲器是先进先出(FIFO)缓冲器,其中数据以特定顺序被接收并以相

同的顺序被处理；首先被处理的数据是首先被接收的数据。在另一实例中，去抖动缓冲器是记录下一个要处理的分组的有序列表。自适应去抖动缓冲器可以是存储器存储单元，其中去抖动缓冲器的状态是自适应去抖动缓冲器中存储的数据的度量（或分组数量）。由去抖动缓冲器处理的数据可从去抖动缓冲器发送给解码器或其它装置。经编码的分组可以对应于固定量（例如，与 8KHz 采样速率下的话音数据的 160 个样本相对应的 20msec）的话音数据。在本发明的一个实例中，具有时间规整功能的解码器所产生的样本数目，可以基于分组是否经过了时间规整而变化。当去抖动缓冲器指示解码器/时间规整器对分组进行扩展时，解码器/时间规整器可产生多于 160 个样本。另一方面，当去抖动缓冲器指示解码器/时间规整器对分组进行压缩时，解码器/时间规整器可产生少于 160 个样本。应注意，可选系统可以具有不同的再现方案，诸如不同于 20ms 语音编码的再现方案。

[0056] 到达去抖动缓冲器的分组可能不会以规则间隔到达。因此去抖动缓冲器的设计目标之一是调整到来的数据的不规则性。在本发明的一个实例中，去抖动缓冲器具有目标去抖动缓冲器长度。目标去抖动缓冲器长度是指在开始再现第一分组之前需要在去抖动缓冲器中累积的数据量。在另一实例中，目标去抖动缓冲器长度可以指去抖动缓冲器中的第一分组在再现之前需要被延迟的时间量。图 2 示出了目标去抖动缓冲器长度。通过在开始分组的再现之前在去抖动缓冲器中累积足够的分组，去抖动缓冲器能够以规则间隔再现随后的分组，同时将分组被用光的潜在可能性最小化。图 2 示出了去抖动缓冲器，其中首先接收到去抖动缓冲器中的声码器分组，是被调度为从去抖动缓冲器输出的下一分组。去抖动缓冲器包括充足的分组来实现所需的去抖动缓冲器延迟。这样，去抖动缓冲器使分组所经历的抖动变得平滑并隐藏分组到达接收器的时间的变化。

[0057] 图 3 示出了各种情形下的分组的传输、接收和再现的时间线。第一分组 PKT 1 在时刻 t_0 被传输，并在时刻 t_1 被接收，并被再现。随后的分组 PKT 2、PKT 3 和 PKT 4 在 PKT 1 之后以 20ms 的间隔被传输。

[0058] 在没有时间规整的情况下，解码器从第一分组的再现时刻起，以规则的时间间隔（例如，20ms）再现分组。例如，如果解码器以规则的 20ms 间隔再现分组，则第一个接收到的分组在时刻 t_1 得到再现，并且随后的分组将在时刻 t_1 后的 20ms、时刻 t_1 后的 40ms、时刻 t_1 后的 60ms 等时刻得到再现。如图 3 所示，PKT 2（在不采用去抖动缓冲器延迟的情况下）的预期再现时刻为 $t_2 = t_1 + 20\text{ms}$ 。PKT 2 在其预期再现时刻 t_2 之前被接收到。另一方面，分组 3 在其预期再现时刻 $t_3 = t_2 + 20\text{ms}$ 之后被接收到。这种状况被称为下溢。当再现装置准备好再现分组，而分组却没有存在于去抖动缓冲器中时，就会发生下溢。典型地，下溢会使解码器产生擦除（erasure）并使再现质量恶化。

[0059] 图 3 还示出了第二种情形，在该情形中，去抖动缓冲器在第一分组的再现之前引入延迟 t_{djb} 。在这种情形下，去抖动缓冲器延迟被加入以便使再现装置每 20ms 都能够接收到分组（或样本）。在这种情形下，尽管 PKT 3 在其预期再现时刻 t_3 之后才被接收到，但是去抖动缓冲器延迟的加入使得 PKT 3 在 PKT 2 再现之后的 20ms 得到再现。

[0060] 现在，PKT 1 在时刻 t_0 被发送，在时刻 t_1 被接收，并在时刻 $t_1 + t_{djb} = t_1'$ 被再现，而不是像先前那样，在时刻 t_1 被再现。再现装置在 PKT 1 之后以例如 20ms 的预定间隔再现 PKT 2，或者在时刻 $t_2' = t_1 + t_{djb} + 20 = t_2 + t_{djb}$ 再现 PKT 2，并在时刻 $t_3' = t_3 + t_{djb}$ 再现 PKT 3。将再现延迟 t_{djb} ，允许第三分组得到再现而不会导致下溢。因此，如图 3 所示，去抖动缓

冲器延迟的引入可减少下溢并防止语音质量的恶化。

[0061] 语音由语音突峰和静音段组成。静音段的扩展 / 压缩对话音质量具有极小的影响或没有影响。这就允许去抖动缓冲器针对每个语音突峰而对第一分组的再现进行不同的延迟。

[0062] 图 4A 和 4B 为不同的语音突峰示出了传输和接收的时间线。应注意,去抖动缓冲器延迟的量被确定以防止下溢。这被称为“最佳去抖动缓冲器延迟”。最佳去抖动缓冲器延迟涉及目标去抖动缓冲器长度。

[0063] 换言之,目标去抖动缓冲器长度被确定成允许足够的数据存储于缓冲器中,以便使分组的再现与再现装置的特性相一致。最佳去抖动缓冲器延迟可通过系统所经历的最大端对端延迟来确定。可选地,最佳去抖动缓冲器延迟可以是基于系统所经历的平均延迟。用于确定最佳去抖动缓冲器延迟的其它方法也可以具体针对给定的标准或系统设计来实现。此外,目标去抖动缓冲器长度被确定以便实现最佳去抖动缓冲器延迟,因此,目标去抖动缓冲器长度可以基于接收的分组速率、分组差错率 (PER) 或其它工作统计数据来计算。

[0064] 图 4A 和 4B 示出了两个实例的最佳去抖动缓冲器延迟。如图所示,相继的分组的传输和接收之间的时间随时间而变化。由于 PKT 3 具有从传输到接收最长的延迟,所以使用该差值来确定用于去抖动处理的最佳延迟。

[0065] 使用具有目标去抖动缓冲器长度的去抖动缓冲器,可以避免至少一些下溢状况。再参考图 3,第二种情形避免了(当解码器预期有分组到来并且再现装置准备好再现分组,而没有分组存在于分组存储缓冲器中时发生的)下溢。此处,在 t_1 之后的 20ms 预定间隔后,PKT 2 得到再现,其中 t_1 是 PKT 1 的再现时刻。尽管将 PKT 3 调度为在时刻 t_3 再现或者预期 PKT 3 在时刻 t_3 再现,但是 PKT 3 直到时刻 t_3 之后才被接收到。换言之,再现装置准备好再现 PKT 3,但是该分组并没有存在于存储缓冲器中。由于 PKT 3 在预期时刻不能用于再现,并且无法得到再现,所以产生了关于 PKT 3 的大抖动量和下溢。PKT 4 在 PKT 4 的预期再现时刻 t_4 得到再现。应注意,预期时刻 t_4 是根据时刻 t_3 计算的。由于每个分组可能会包含多于一个语音分组,所以由于下溢而引起的分组的丢失会使语音质量恶化。

[0066] 供考虑的另一情形涉及如图 5 所示的“由于延迟分组而引起的下溢”的形成过程,其中分组的传输、接收和预期再现时刻在时间上被示出。在这种情形下,每个分组在其预期再现时刻后的短时间内被接收到。例如,PKT 50 的预期再现时刻是 t_0 ,但是 PKT 50 直到 t_0 后的时刻 t_0' 才被接收到。下一分组 51 预期在时刻 t_1 被接收到,但是它直到 t_1 后的时刻 t_1' 才被接收到。这引起导致“延迟下溢”(即由于延迟分组而引起的下溢)的高百分比的下溢的形成,并因而引起更高的端对端延迟。

[0067] 显然,将再现延迟较大的量的去抖动缓冲器,在将下溢保持在最少这方面是成功的。然而,这样的去抖动缓冲器将较大的去抖动缓冲器延迟引入到分组的端对端延迟中。较大的端对端延迟会导致保持会话流中的困难。大于 100ms 的延迟会使收听方认为讲话方还没有结束讲话。因此,要获得良好的质量,在理想情况下既要考虑避免下溢也要考虑减小端对端延迟。由于解决一个问题会恶化另一个问题,所以存在着问题。换言之,较小的端对端延迟通常会导致更多的下溢,而较大的端对端延迟通常会导致更少的下溢。因此,需要平衡这些竞争的目标。具体而言,需要使去抖动缓冲器能够跟踪和避免下溢并同时减小端对端延迟。

[0068] 去抖动缓冲器目标长度

[0069] 自适应去抖动缓冲器的设计目标是允许系统以语音分组的特定“下溢率”为目标，同时实现低的端对端延迟。由于感觉到的质量是下溢的百分比的函数，所以以下溢的特定百分比为目标的能力使得语音质量能够得到控制。当存在缺失的分组时，在去抖动缓冲器处会发生分组下溢。当分组被丢失或延迟时，该分组是缺失的分组。当丢失的分组在其到达接收器前被丢弃时（诸如，当它在接入网中的某处（例如，在物理层或前向链路调度器上）被丢弃时）会导致下溢。在这种情形下，通过使用去抖动缓冲器延迟无法校正下溢，因为分组从未到达去抖动缓冲器。可选地，可能会由于被延迟并在其再现时刻之后到达的分组而发生下溢。除了跟踪由于延迟分组而引起的下溢之外，自适应去抖动缓冲器还可以跟踪由于丢失的分组而引起的下溢。

[0070] 可以通过对下溢进行权衡而得到去抖动缓冲器延迟，来控制由于延迟分组而引起的下溢的数目。表示由于延迟分组而引起的下溢的目标百分比的值被称为“下溢目标”。该值是用于去抖动缓冲器的操作的目标值，并被选择为能够将端对端延迟保持在合理的限度内。在一个实例中，1% (0.01) 的值可用作“下溢目标”。另一实例使用 0.5% (0.005) 的值。为了实现“下溢目标”，可以使去抖动缓冲器延迟自适应。

[0071] 在本发明的一个实例中，可以使用由于延迟分组而引起的下溢（下文中称为“延迟下溢”）的百分比的经过滤波的值，来使去抖动缓冲器延迟自适应。在每个静音段的结尾（或每个语音突峰的开头）处，去抖动缓冲器延迟如图 6 所示被更新。如图 6 所示，算法规定如下内容：

[0072] 1) If ($PER_{delay} < TARGET_VALUE$) then

[0073] $DEJITTER_DELAY = DEJITTER_DELAY - CONSTANT$;

[0074] 2) If ($PER_{delay} > TARGET_VALUE \&\& PER_{delay} \geq last_PER_{delay}$) then

[0075] $DEJITTER_DELAY = DEJITTER_DELAY + CONSTANT$;

[0076] 3) Set $DEJITTER_DELAY = MAX(MIN_JITTER, DEJITTER_DELAY)$;

[0077] AND

[0078] 4) $DEJITTER_DELAY = MIN(MAX_JITTER, DEJITTER_DELAY)$. (1)

[0079] 在本实例中，可将初始去抖动缓冲器延迟设置成诸如 40ms 的常数值。TARGET_VALUE 是“延迟下溢”的目标值（例如 1%）。 PER_{delay} 是在滤波器的参数允许实现 TARGET_VALUE 的情况下，分组的“延迟下溢”率的经过滤波的值。 $last_PER_{delay}$ 是去抖动缓冲器延迟的前一次更新时的 PER_{delay} 的值。DEJITTER_DELAY 是如上文中定义的目标去抖动缓冲器长度。在本实例中，CONSTANT 等于 20ms。MIN_JITTER 和 MAX_JITTER 是去抖动缓冲器延迟的最小值和最大值；根据一个实例，这些值分别被设置为 20ms 和 80ms。MIN_JITTER 和 MAX_JITTER 可以基于系统模拟来估计。(MIN_JITTER, MAX_JITTER, CONSTANT) 的值可取决于使用去抖动缓冲器的通信系统而被最优化。

[0080] PER_{delay} 可以在每个静音段的结尾或在每个语音突峰的开头处被更新，其中 PER_{delay} 被如下计算：

[0081] $PER_{delay} = PER_CONSTANT \times PER_{delay} + (1 - PER_CONSTANT) \times Current_PER_{delay}$ (2)

[0082] PER_CONSTANT 是供滤波器用于估计 PER_{delay} 的时间常数。该常数的值确定滤波器的存储，并允许实现 TARGET_VALUE。

[0083] $Current_PER_{delay}$ 是在 PER_{delay} 的上一次更新和当前更新之间观察到的“延迟下溢”率。

[0084] $Current_PER_{delay}$ 被定义为上一次更新和当前更新之间的延迟下溢分组的数目与 PER_{delay} 的接收到的分组的总数的比率。

[0085]

$$Current_PER_{delay} = \frac{\text{自上一次更新以来的延迟下溢的数目}}{\text{自上一次更新以来接收到的分组的数目}} \quad (3)$$

[0086] 参考图 6, 用于计算和更新去抖动缓冲器延迟的处理 100 通过初始化 DEJITTER_DELAY 而在步骤 101 开始。在步骤 102, 通过比较而将 PER_{delay} 与 TARGET_VALUE 进行比较。如果 PER_{delay} 小于 TARGET_VALUE, 则在步骤 104, 从 DEJITTER_DELAY 中减去值 CONSTANT。如果在步骤 102 中 PER_{delay} 大于 TARGET_VALUE, 并且在步骤 103 中 PER_{delay} 大于 TARGET_VALUE 并大于或等于 LAST_PER_DELAY, 并在步骤 103 中不小于 last_PER_DELAY, 则处理继续进行到决定 108。在步骤 108, DEJITTER_DELAY 被设置成 DEJITTER_DELAY 加上值 CONSTANT。从步骤 103 继续, 如果 PER_{delay} 不大于 TARGET_VALUE 并且不大于或等于 LAST_PER_DELAY, 则处理继续进行到步骤 110。此外, 从步骤 104 继续, 在步骤 110, DEJITTER_DELAY 被设置成等于 MIN_JITTER 和 DEJITTER_DELAY 中的最大值。从步骤 110, 处理继续进行到步骤 112, 以便在步骤 112, 将 DEJITTER_DELAY 设置成等于 MAX_JITTER 和 DEJITTER_DELAY 中的最小值。

[0087] 对延迟的跟踪

[0088] 去抖动缓冲器可以进入对延迟进行跟踪（而不是对下溢率进行跟踪）的模式。所跟踪的延迟可以是端对端延迟或去抖动缓冲器延迟。

[0089] 在一个实例中, 去抖动缓冲器在目标下溢率可以很容易满足时进入“延迟跟踪”模式。这意味着去抖动缓冲器能够在某段时间内实现比目标下溢率低的下溢率。该段时间可以是几百毫秒到几秒中的任何一段时间。

[0090] 在该模式下, 去抖动缓冲器具有目标延迟值。这类似于上述的下溢目标值。可以以类似的方式使用上文中用于确定目标下溢率的等式 (1) 来计算目标延迟值。当去抖动缓冲器进入确定目标延迟值的该模式时, 只要能够维持目标延迟, 这就可以允许去抖动缓冲器减小其目标下溢率。

[0091] 隐式缓冲器自适应

[0092] 在一些情形下, 解码器可能会期待再现还没有接收到的分组。图 5 中示出了这种情形, 其中 PKT 50 的预期再现时刻是 t_0 , 但是 PKT 50 在该时刻后才被接收到。类似地, PKT 51 在其预期再现时刻 t_1 后才被接收到, PKT 52 在其预期再现时刻 t_2 后才被接收到, 等等。此处应注意, 分组相当规则地到达, 但是由于 PKT 50 的接收时刻稍迟于其预期再现时刻, 所以它使所有随后的分组也错过了它们的再现时刻。另一方面, 如果解码器能够在时刻 t_0 插入擦除并仍能够在时刻 t_1 再现分组 PKT 50, 则它将使得所有分组都符合它们的再现时刻。通过在擦除之后再再现 PKT 50 而不是已经对 PKT 50 进行再现, 可以有效地使去抖动缓冲器长度自适应。

[0093] 应注意, 在 PKT 50 的擦除之后再再现 PKT 50 可能会导致不连续, 这可以通过使用 2005 年 7 月 7 日提交的标题为“PHASE MATCHING INVOCODERS”的共同未决的申请第 11/192, 231 号中描述的相位匹配技术来消除。

[0094] 如图 7A 所示,在分组的接收中可能会存在间隙,诸如 PKT 3 和 PKT 4 之间的时间间隙。对于每个分组而言,分组到达时间中的延迟可能是不同的。去抖动缓冲器可以立即采用调整以补偿该延迟,来做出响应。如图所示,PKT 1、PKT 2 和 PKT 3 分别在时刻 t_1 、 t_2 和 t_3 被接收到。在时刻 t_4 ,预期将接收到 PKT 4,但是 PKT 4 还没有到达。假定在图 7A 中,预期每 20ms 就会接收到分组。在本示例中,PKT 2 在 PKT1 后的 20ms 被接收到,并且 PKT 3 在 PKT 1 后的 40ms 被接收到。PKT4 预期在 PKT 1 后的 60ms 被接收到,但是直到 PKT 1 后的 80ms 才到达。

[0095] 在图 7B 中,在接收到的第一个分组 PKT 1 的再现之前,在去抖动缓冲器处引入初始延迟。此处,传输延迟是 D_{init} 。在这种情况下,PKT1 将在时刻 D_{init} 由缓冲器再现,PKT 2 将在时刻 $D_{init}+20ms$ 由缓冲器再现,PKT 3 将在时刻 $D_{init}+40ms$ 由缓冲器再现,等等。在图 7B 中,当 PKT 4 未能在预期时刻 $D_{init}+60ms$ 到达时,擦除可由去抖动缓冲器再现。在对分组进行再现的下一时刻,去抖动缓冲器将试图再现 PKT 4。如果 PKT 4 仍未到达,则可在时刻 $D_{init}+80ms$ 发送另一个擦除。擦除将继续被再现,直到 PKT 4 到达去抖动缓冲器为止。一旦 PKT 4 到达去抖动缓冲器,PKT 4 就接着被再现。这样的处理会导致延迟,因为直到接收到 PKT 4 才会对其它分组进行再现。当系统不能够恢复(即,从未接收到)PKT 4 时,系统可以对处理进行复位,从而允许再现 PKT 4 之后的分组而不再现 PKT 4。在上述情形下,去抖动缓冲器的端对端延迟具有增加的潜在可能,因为在 PKT 4 到达之前可能会长时间地继续发送擦除。

[0096] 相反,根据如图 7C 所示的实例,如果分组未能到达或者如果分组的接收被延迟,则在 PKT 4 的预期再现时刻将再现擦除。这类似于上面关于图 7B 所描述的情形,在该情形下,系统等待 PKT 4。在下一再现时刻,如果 PKT 4 仍未到达而下一分组 PKT 5 已经到达,则 PKT 5 得到再现。为了进一步说明,假设 PKT 4 的接收被延迟并且去抖动缓冲器预期在时刻 $D_{init}+80ms$ 接收到 PKT 4。当 PKT 4 被延迟时,将再现擦除。在时刻 $D_{init}+100ms$,如果 PKT 4 仍未到达,则对 PKT 5 进行再现而不再现另一擦除。在该第二情形下,立即做出了对延迟的调整并避免了通信网络中过大的端对端延迟。该处理可被称为 IBA,因为在再现之前存储在缓冲器中的数据的大小是根据数据的接收而增加和减小的。

[0097] 隐式缓冲器自适应 (IBA) 处理 200 由图 8A 中的流程图示出。处理 200 可以在自适应去抖动缓冲器内的控制器(诸如输出控制器 760 或去抖动缓冲器控制器 756)中。处理 200 可以驻留在支持自适应去抖动缓冲器的系统内的其它部分中。在步骤 202,在自适应去抖动缓冲器处接收请求以提供用于再现的下一分组。下一分组被标识为具有序列中的索引 i 的分组,具体为 PKT[i]。在步骤 204,如果隐式缓冲器自适应 (IBA) 模式被启用,则处理器继续进行到步骤 206 以根据 IBA 模式进行处理;如果 IBA 模式被禁用,则处理继续进行到步骤 226 以便不采用 IBA 模式进行处理。

[0098] 如果在步骤 206 接收到 PKT[i],则在步骤 208,自适应去抖动缓冲器提供 PKT[i] 以用于再现。IBA 模式在步骤 210 被禁用,并且索引 i 被增加,即, ($i = i+1$)。此外,如果在步骤 206 没有接收到 PKT[i] 并且如果在步骤 214 接收到 PKT[$i+1$],则处理继续进行到步骤 216 以对 PKT[$i+1$] 进行再现。IBA 模式在步骤 218 被禁用,并且在步骤 220,索引 i 被增加两次,即, ($i = i+2$)。

[0099] 如果在步骤 214,PKT[i] 和 PKT[$i+1$] 都没有接收到,则在步骤 222,控制器启动对

擦除的再现；并且在步骤 224,索引 i 被增加。应注意,在本实例中,当处于 IBA 模式中时,控制器总共检查两 (2) 个分组,来响应诸如在步骤 202 中接收到的对下一分组的请求。这有效地实现了分组窗口,其中控制器在该分组窗口的范围内搜索接收到的分组。可选实例可以实现不同的窗口大小,例如,搜索三 (3) 个分组,在该实例中将是分组序号为 i 、 $i+1$ 和 $i+2$ 的分组。

[0100] 返回到步骤 204,如果 IBA 模式没有被启用,则处理继续进行到步骤 226 以确定是否接收到 PKT[i]。如果接收到,则在步骤 228,提供 PKT[i] 用于再现,并且在步骤 230,索引 i 被增加。如果在步骤 226 没有接收到 PKT[i],则在步骤 232,自适应去抖动缓冲器提供擦除以用于再现。由于没有接收到 PKT[i] 并且擦除取代 PKT[i] 而得到再现,所以启用 IBA 模式。

[0101] 图 8B 是与 IBA 模式有关的状态图。当处于普通模式 242 时,如果自适应去抖动缓冲器提供 PKT[i] 用于再现,则控制器停留在普通模式。当擦除得到再现时,控制器从普通模式 242 转换到 IBA 模式 240。一旦处于 IBA 模式 240,控制器就停留在该模式中并继续再现擦除。在再现了 PKT[i] 或 PKT[$i+1$] 之后,控制器从 IBA 模式 240 转换到普通模式 242。

[0102] 图 9 是实现诸如图 8A 和 8B 所示的 IBA 的去抖动缓冲器的一个实例。在本示例中,再现装置从解码器请求用于再现的样本。然后,解码器从去抖动缓冲器请求足够的分组以允许再现装置能够进行不间断的再现。在本示例中,分组传送语音通信,并且再现装置每 20ms 就再现一样本。可选系统可以通过其它配置将来自去抖动缓冲器的分组化的数据提供给再现装置,并且分组化的数据可以不同于语音通信。

[0103] 去抖动缓冲器在图 9 中作为分组的堆叠而被示出。在该示例中,缓冲器首先接收到 PKT 49,接着随后接收到 PKT 50、PKT 51、PKT 52、PKT 53 等。该示例中的分组号指的是分组的顺序。然而,在分组化的系统中,并不能保证将以该顺序接收到分组。为了便于清楚地理解,在该示例中,分组以与传输顺序相同的数字顺序被接收,该顺序也是再现的顺序。为了说明的目的,在图 9 中,将随后接收到的分组堆在去抖动缓冲器中的先前接收到的分组的上面;例如,PKT 49 堆在 PKT50 的上面,PKT 51 堆在 PKT 50 的上面,等等。去抖动缓冲器中的位于堆叠底部的分组是首先要发送给再现装置的分组。还应注意,在本示例中,未示出目标去抖动缓冲器长度。

[0104] 在图 9 中,相对于时间而画出了分组的接收、分组的预期接收时刻以及分组的再现时刻。在每次接收到分组时,都示出了更新后的缓冲器状态。例如,PKT 49 在时刻 t_0 被接收到,其中预期在时刻 t_1 对 PKT49 进行再现。在时刻 t_0 (即 PKT 49 的接收时刻) 上方的图的顶部,示出了在接收到 PKT 49 之后的缓冲器状态。将去抖动缓冲器处接收到的每个分组的接收时刻图示为 RECEIVED。就在 RECEIVED 时刻的下面,图示了 ANTICIPATED PLAYBACK 时刻。再现时刻被标识为 PLAYBACK。

[0105] 在该实例中,在最初,下一个要再现的分组是 PKT 49,它被预期在时刻 t_0 得到再现。预期下一个相继的分组在时刻 t_1 得到再现,等等。第一分组 PKT 49 在预期再现时刻 t_0 之前被接收到。因此,PKT 49 按照预期在时刻 t_0 得到再现。预期下一分组 PKT 50 在时刻 t_1 得到再现。

[0106] 然而,PKT 50 的接收被延迟,并且擦除取代 PKT 50 而被发送给再现装置。PKT 50 的延迟导致了如前所述的下溢。PKT 50 在预期再现时刻 t_1 之后并在下一预期再现时刻 t_2

之前被接收到。一旦接收到 PKT 50, 就将 PKT 50 存储在去抖动缓冲器中。因此, 当在时刻 t_2 接收到下一个对用于再现的分组请求时, 系统在去抖动缓冲器中寻找序号最低的分组; 并且 PKT 50 被提供给再现装置以用于在时刻 t_2 再现。应注意, 通过使用 IBA, 即使没有及时接收到 PKT 50 以按照预期进行再现, PKT50 也能在之后得到再现, 并且具有从该点接续下来的剩余的序号的分组也能得到再现。如图所示, 随后的分组 PKT 51、PKT 52 等都被及时地接收和再现以避免更多的擦除。

[0107] 尽管看上去好像 IBA 增加了分组的端对端延迟, 但这并非实际情况。由于 IBA 导致更少数量的下溢, 所以可以将从上文中的等式 1 估计的去抖动缓冲器值维持在较小的值。因此, IBA 的整体效果将是分组整体的平均端对端延迟的减小。

[0108] IBA 可以增强具有话音突峰的通信的处理。话音突峰是指语音通信的话音部分, 其中语音通信包括符合正常话音模式 (speech pattern) 的话音部分和静音部分。在话音处理中, 声码器为话音生成一种类型的分组并为静音生成另一种类型的分组。话音分组以一种编码速率编码, 并且静音以不同的编码速率编码。当在去抖动缓冲器处接收到经编码的分组时, 去抖动缓冲器根据编码速率而识别分组类型。去抖动缓冲器假定话音帧是话音突峰的一部分。第一个非静音帧是话音突峰的开头。话音突峰在静音分组被接收到时结束。在不连续的传输中, 并非所有的静音分组都被传输, 这是因为接收器可能会实现模拟的噪声来占据通信的静音部分。在连续传输中, 所有的静音分组都被传输和接收。在一个实例中, 去抖动缓冲器根据接收到的分组的类型来调整去抖动缓冲器长度。换言之, 系统可以决定减小为通信的静音部分所需的去抖动缓冲器的长度。应注意, IBA 方法可以适用于根据预定的定时方案 (诸如固定速率等) 进行再现的任何通信。

[0109] 时间规整

[0110] 话音突峰通常由数据的多个分组构成。在一个实例中, 可以将话音突峰的第一分组的再现延迟与去抖动缓冲器延迟相等的长度。去抖动缓冲器延迟可以以各种方式被确定。在一种情形下, 去抖动缓冲器延迟可以是基于诸如上文中的等式 1 的算法而计算的去抖动缓冲器延迟。在另一情形下, 去抖动缓冲器延迟可以是接收与去抖动缓冲器延迟的长度相等的语音数据所花费的时间。可选地, 可将去抖动缓冲器延迟选择为前述值中的较小值。在该实例中, 假设去抖动缓冲器延迟通过使用等式 1 而被计算为 60ms, 并且话音突峰的第一个分组在第一时刻 t_1 被接收到。当在第一分组后的 50ms 接收到话音突峰的下一分组时, 自适应去抖动缓冲器数据等于去抖动延迟 60ms。换言之, 从在自适应去抖动缓冲器处接收到分组到分组的再现所经过的时间为 60ms。

[0111] 应注意, 可以设置自适应去抖动缓冲器的目标长度以实现 60ms 延迟。

[0112] 这样的计算确定的是为了满足延迟时间而将要存储的分组的数目。

[0113] 自适应去抖动缓冲器监视数据对缓冲器的填充和从缓冲器的腾空, 并调整缓冲器的输出以便将缓冲器维持在目标延迟长度, 即调整数据量以便实现目标延迟时间。当去抖动缓冲器发送话音突峰的第一分组以用于再现时, 存在着等于 Δ 的延迟, 其中 $\Delta = \text{MIN}(\text{去抖动缓冲器延迟}, \text{接收等于去抖动延迟的语音数据所花费的时间})$ 。话音突峰的随后的分组被延迟的时间为 Δ 加上再现先前分组所花费的时间。因此, 一旦已经定义了第一分组的去抖动缓冲器延迟, 就隐式地定义了相同话音突峰中的随后分组的去抖动缓冲器延迟。在实践中, 去抖动缓冲器延迟的定义可能会需要额外的考虑以便适应诸如图 10 中所示

的情况。

[0114] 图 10 示出了话音突峰中的语音信息的传输。话音突峰 150 在时刻 t_0 被接收, 并且话音突峰 154 在时刻 t_2 被接收。在话音突峰 150 和话音突峰 154 之间接收到 20ms 的静音段 152。在接收到之后, 自适应去抖动缓冲器可以存储接收到的数据并为每个话音突峰分组的再现确定延迟。在该实例中, 话音突峰 150 在时刻 t_0 在自适应去抖动缓冲器处被接收到, 其中自适应去抖动缓冲器延迟时间被计算为 80ms。将去抖动缓冲器延迟加到接收时刻上以产生再现时刻。这样, 话音突峰 150 在再现之前被自适应去抖动缓冲器延迟了 80ms。话音突峰 150 在时刻 t_1 开始再现, 其中 $t_1 = t_0 + 80\text{ms}$, 或者话音突峰 150 在接收到话音突峰 150 之后的 80ms 开始再现; 并且话音突峰 150 在时刻 t_4 完成再现。通过使用诸如等式 1 的算法来计算上述目标去抖动缓冲器长度, 施加给话音突峰 154 的去抖动缓冲器延迟是 40ms。这意味着话音突峰 154 的第一个分组将在时刻 t_3 得到再现, 其中 $t_3 = t_2 + 40\text{ms}$, 或者话音突峰 154 的第一个分组将在接收到话音突峰 154 之后的 40ms 得到再现。然而, 分组 154 在时刻 t_3 的再现, 与在时刻 t_4 结束再现的话音突峰 150 的最后一个分组的再现是相冲突的。因此, (为分组 154) 计算出的 40ms 的去抖动缓冲器延迟不能使话音突峰 150 有足够的时间结束再现。为了避免冲突并允许两个分组都正确地得到再现, 应该在话音突峰 150 的最后一个分组已经再现之后才对与话音突峰 150 之间具有静音段的话音突峰 154 的第一个分组进行再现。在该实例中, 话音突峰 150 和话音突峰 154 在时刻 t_3 至 t_4 之间相重叠。因此, 再现方法在这种情形下是不理想的。为了防止如本文中所述的分组再现之间的重叠, 需要检测前一个话音突峰的最后一个分组被再现的时刻。因此, 为分组进行去抖动缓冲器延迟的计算可以考虑前一个再现分组的再现时间, 以便避免重叠或冲突。

[0115] 如上所述, 在一个实例中, 去抖动缓冲器延迟是在话音突峰的开头处计算或更新的。然而, 将去抖动缓冲器延迟的更新限制到话音突峰的开头, 可能是有限制的, 因为话音突峰常常在长度上有所变化并且工作条件在话音突峰期间可能会发生变化。考虑图 10 的实例。因此, 需要在话音突峰期间更新去抖动缓冲器延迟。

[0116] 应注意, 合乎需要的是, 控制从自适应去抖动缓冲器中流出的数据流以便维持目标延迟长度。这样, 如果自适应去抖动缓冲器正在接收具有可变延迟的数据, 则从自适应去抖动缓冲器中流出的数据就可以得到调整, 以便允许缓冲器填充有足够的数据来满足目标自适应去抖动缓冲器长度。当自适应去抖动缓冲器所接收的分组不充足时, 可以使用时间规整来扩展分组以便维持目标延迟长度。类似地, 当自适应去抖动缓冲器所接收的分组过多并且正在存储超出目标延迟长度的分组时, 可以使用时间规整来压缩分组。自适应去抖动缓冲器可以与解码器配合工作以便对分组进行时间规整, 如本文中所述。

[0117] 图 11 是包括通过网络元件进行通信的两个接收器的系统的框图。接收器是 AT 252 和 AT 282; 如图所示, AT 252 和 282 适于通过 BS 270 进行通信。在 AT 252 中, 发射处理单元 264 将语音数据发送给编码器 260, 编码器 260 对语音数据进行数字化并将经分组的数据发送给低层处理单元 258。然后分组被发送给 BS 270。当 AT 252 接收到来自 BS 270 的数据时, 数据首先在低层处理单元 258 中被处理, 从这里, 数据的分组被提供给自适应去抖动缓冲器 256。接收到的分组被存储在自适应去抖动缓冲器 256 中, 直到达到目标去抖动缓冲器长度为止。一旦达到了目标去抖动缓冲器长度, 自适应去抖动缓冲器 256 就将数据发送给解码器 254。在该示出的实例中, 实现时间规整的压缩和扩展可以在解码器 254

中执行,该解码器 254 将分组化的数据转换成语音数据并将语音数据发送给接收处理单元 262。在本发明的另一实例中,时间压缩和扩展(时间规整)可以在自适应去抖动缓冲器内由(未示出的)控制器执行。AT 282 的行为类似于 AT 252。AT 282 在从发射处理单元 294 到编码器 290 到低层处理单元 288 并最终到 BS 270 的路径上发送数据。AT 282 在从低层处理单元 288 到自适应去抖动缓冲器 286 到解码器 284 到接收处理单元 292 的路径上接收数据。进一步的处理并未示出,但是可能会影响诸如语音的数据的再现并且可能会涉及音频处理、屏幕显示等。

[0118] 等式 1 中给出的去抖动缓冲器等式在话音突峰的开头处计算去抖动缓冲器延迟。去抖动缓冲器延迟可以表示诸如由话音突峰确定的特定分组数目,或者可以表示与(诸如语音数据的)数据的再现等效的预期时刻。此处应注意,去抖动缓冲器具有目标大小,并且这确定了预期在所有时间点处在去抖动缓冲器中看到的所存储的数据量。

[0119] 由于信道条件和其它工作条件而引起的分组延迟的变化,会导致分组到达自适应去抖动缓冲器的时间的差别。因此,自适应去抖动缓冲器中的数据量(分组数目)可能会小于或大于所计算的去抖动缓冲器延迟值 DEJITTER_DELAY。例如,分组可能会以比它们最初在编码器处被生成的速率更慢或更快的速率到达去抖动缓冲器。当分组以慢于预期速率的速率到达去抖动缓冲器时,去抖动缓冲器会开始耗尽,因为到来的分组将不能以相同的速率补充流出的分组。换句话说,如果分组以比编码器处的生成速率更快的速率到达,则去抖动缓冲器的大小会开始增加,因为分组离开去抖动缓冲器的速率无法与它们进入去抖动缓冲器的速率一样快。前一种状况会导致下溢,而后一种状况会由于去抖动缓冲器中更大的缓冲时间而导致较高的端对端延迟。后一种状况是重要的,因为如果分组数据系统的端对端延迟减小(AT 移动到负载更小的区域或者用户移动到具有更好信道质量的区域),则合乎需要的是将该延迟减小结合到话音的再现中。端对端延迟是重要的话音质量因素,并且再现延迟的任何减小都会被感觉为会话或话音质量的提高。

[0120] 为了在去抖动缓冲器中校正 DEJITTER_DELAY 和实际存在于去抖动缓冲器中的数据量之间的差别,去抖动缓冲器的一个实例采用时间规整。时间规整涉及扩展或压缩话音分组的持续时间。当自适应去抖动缓冲器开始耗尽时,去抖动缓冲器通过扩展话音分组来实现时间规整,并且当自适应去抖动缓冲器变得大于 DEJITTER_DELAY 时,去抖动缓冲器通过压缩话音分组来实现时间规整。自适应去抖动缓冲器可以与解码器配合工作以便对分组进行时间规整。时间规整提供话音质量的实质性的提高而不会增加端对端延迟。

[0121] 图 12 是实现时间规整的自适应去抖动缓冲器的一个实例的框图。物理层处理单元 302 将数据提供给数据堆栈 304。数据堆栈 304 将分组输出给自适应去抖动缓冲器和控制单元 306。前向链路 (FL) 媒体接入控制 (MAC) 处理单元 300 向去抖动处理单元 306 提供越区切换指示。MAC 层实现用于在物理层上(即通过空中)接收和发送数据的协议。MAC 层可以包括安全、加密、认证和连接信息。在支持 IS-856 的系统中,MAC 层包含管理控制信道、接入信道以及前向和反向业务信道的规则。目标长度估计器 314 通过使用等式 1 中给出的计算而向去抖动缓冲器提供目标去抖动缓冲器长度。对目标长度估计器 314 的输入包括分组到达信息和当前分组差错率 (PER)。应注意,可选配置可以包括位于自适应去抖动缓冲器和控制单元 306 内的目标长度估计器 314。

[0122] 在一个实例中,自适应去抖动缓冲器和控制单元 306 还包括控制为再现提供的数

据的速率的再现控制。从自适应去抖动缓冲器和控制单元 306, 分组被发送给不连续发射 (DTX) 单元 308, 其中 DTX 单元 308 在话音数据未被接收到时向解码器 310 提供背景噪声信息。应注意, 由自适应去抖动缓冲器和控制单元 306 提供的分组已准备好用于解码处理, 并且可被称为声码器分组。解码器 310 对分组进行解码并将脉冲编码调制 (PCM) 话样本提供给时间规整单元 312。在可选实例中, 可以将时间规整单元 312 实现在解码器 310 内。时间规整单元 312 从自适应去抖动缓冲器和控制单元 306 接收时间规整指示符。时间规整指示符可以是控制信号、指令信号或标记。在一个实例中, 时间规整指示符是具有例如压缩、扩展和不进行时间规整的多状态指示符。对于不同的压缩级别和 / 或不同的扩展级别, 可以有不同的值。在一个实例中, 时间规整指示符指示时间规整单元 312 扩展或压缩数据。时间规整指示符指示扩展、压缩或不进行时间规整。可以将时间规整指示符看作是启动时间规整单元 312 处的操作的控制信号。时间规整指示符可以是规定如何扩展或压缩分组的消息。时间规整指示符可以标识要进行时间规整的分组以及所采取的操作是压缩还是扩展。此外, 时间规整指示符可以向时间规整单元 312 提供对选项的选择。在静音间隔期间, DTX 模块将去抖动缓冲器所提供的擦除流修改成擦除和静音帧的流, 其中解码器使用该擦除和静音帧的流来重建更准确和更高质量的背景噪声。在可选实例中, 时间规整指示符开启和关闭时间规整。在另一实例中, 指示符标识用于再现的压缩和扩展的量。时间规整单元 312 可以修改来自解码器的样本并将样本提供给音频处理 316, 该音频处理 316 可以包括接口和转换单元以及音频驱动器和扬声器。

[0123] 尽管时间规整指示符标识了何时进行压缩或者何时进行扩展, 但仍需要确定将对给定分组应用多少时间规整。在一个实施例中, 时间规整的量是固定的, 其中分组根据话音周期或音高而被进行时间规整。

[0124] 在一个实施例中, 时间规整指示符是作为目标扩展级别或目标压缩级别的百分比来传递的。换言之, 时间规整指示符指示压缩给定的百分比或者扩展给定的百分比。

[0125] 在一种情形下, 可能有必要识别到来的数据的已知特性。例如, 编码器可以预期已知音调的数据或者具有例如特定长度特性的数据。

[0126] 在这种情况下, 由于可以预期特定特性, 所以使用时间规整来修改接收到的数据将不是理想的。例如, 编码器可以预期到来的数据具有特定的音调长度。然而, 如果时间规整被启用, 则音调长度可能会通过时间规整而被修改。因此, 在这种情形下, 不应该启用时间规整。基于音调的通信包括但不限于: 文本电话 / 聋哑人电话装置 (TTY/TDD) 信息、使用键盘输入的应用或者使用基于音调的通信的其它应用。在这样的通信中, 音调载波信息的长度, 并且因此, 修改音高或音调长度, 诸如在再现时的压缩或扩展, 会导致该信息的丢失。在使有听力障碍的接收者能够接收的 TTY、TDD 和其它应用中, 解码器也提供这种通信的其带内处理的状态。该指示被用于掩蔽去抖动缓冲器所提供的时间规整指示。如果解码器正在处理具有 TTY/TDD 信息的分组, 则应该禁用时间规整。这可以用 2 种方式来实现: 向去抖动缓冲器控制器提供 TTY/TDD 状态, 或者向时间规整单元提供 TTY/TDD 状态。如果解码器 TTY/TDD 状态被提供给去抖动缓冲器控制器, 则当声码器指示 TTY/TDD 的处理时, 控制器不应该指示任何扩展或压缩指示。如果解码器 TTY/TDD 状态被提供给时间规整单元, 则这会起到滤波器的作用, 并且如果解码器正在处理 TTY/TDD 信息, 则时间规整单元在接收到时间规整指示后不会采取操作。

[0127] 在如图 12 所示的系统中,自适应去抖动缓冲器和控制单元 306 监视到来数据的速率,并且在过多或过少的分组被获得或缓冲时生成时间规整指示符。自适应去抖动缓冲器和控制单元 306 确定何时进行时间规整以及采取什么操作。图 13A 示出了通过使用压缩和扩展阈值来做出时间规整决定的自适应去抖动缓冲器的一个实例的操作。自适应去抖动缓冲器累积可能已经以不规则的时间间隔到达的分组。去抖动目标长度估计器 314 产生目标去抖动缓冲器长度;然后目标去抖动缓冲器长度被施加给去抖动缓冲器。在实践中,自适应去抖动缓冲器和控制单元 306 使用去抖动缓冲器长度值来做出关于去抖动缓冲器操作的控制决定并控制再现。压缩阈值和扩展阈值分别指示何时触发压缩或扩展。可以将这些阈值指定为去抖动目标长度的一小部分。

[0128] 如图 13A 所示,目标去抖动缓冲器长度被给定为 L_{Target} 。压缩阈值被给定为 T_{Compress} ,并且扩展阈值被给定为 T_{Expand} 。当去抖动缓冲器长度增加到超出压缩阈值 T_{Compress} 时,去抖动缓冲器向解码器指示分组应该被压缩。

[0129] 以类似的方式,当去抖动缓冲器长度减小到低于扩展阈值 T_{Expand} 时,去抖动缓冲器向解码器指示分组应该被扩展并应该以更低的速率被有效地再现。

[0130] 扩展和压缩阈值之间的操作点避免了下溢以及端对端延迟的过大的增加。因此,目标操作是在 T_{Compress} 和 T_{Expand} 之间。在一个实例中,扩展和压缩阈值的值被分别设置成去抖动缓冲器的目标值的 50% 和 100%。尽管在一个实例中,时间规整可以在解码器内部执行,而在可选实例中,该功能可以在解码器外部执行,例如在解码之后执行。然而,在合成信号之前对信号进行时间规整可能会更简单。如果在解码信号之后应用这样的时间规整方法,则将需要估计信号的音高周期。

[0131] 在某些情形下,例如在 W-CDMA 系统中,去抖动缓冲器长度可以更大。时间规整阈值发生器可以生成多个压缩和扩展阈值。可以响应于工作条件而计算这些阈值。图 13B 中示出了多级阈值。 T_{C1} 是第一压缩阈值, T_{C2} 是第二压缩阈值,并且 T_{C3} 是第三压缩阈值。图中也示出了表示三个不同的压缩阈值的值的 T_{E1} 、 T_{E2} 和 T_{E3} 。阈值可以基于时间规整的百分比(经历时间规整的分组数目)、压缩分组的百分比、扩展分组的百分比或者这两个值的比率。阈值的数目可以根据需要而变化,换言之,可能会需要更多或更少的阈值。阈值中的每一个涉及不同的压缩或扩展速率,例如,对于要求更细粒度的系统,可以使用更多的阈值,而对于更粗糙的粒度,可以使用更少的阈值。 T_{E1} 、 T_{E2} 和 T_{E3} 等可以是目标延迟长度的函数。可以通过跟踪延迟下溢并基于诸如 PER 的差错统计数据来改变阈值。

[0132] 图 14 示出了采用时间规整和没有采用时间规整的分组的再现。在图 14 中,PKT 1 在时刻 t_1 被传输,PKT 2 在时刻 t_2 被发送,等等。分组按照所指示的那样到达接收器,其中 PKT 1 在 t_2' 到达,并且 PKT 2 在 t_2'' 到达。对于每个分组,将没有使用时间规整的再现时刻作为 PLAYBACK WITHOUT WARPING(没有采用规整的再现)给出。相反,将使用时间规整的再现时刻作为 PLAYBACK WITH WARPING(采用规整的再现)给出。由于本实例是关于诸如话音通信的实时数据的,所以分组的预期再现时刻具有固定的时间间隔。在再现期间,理想的是每个分组在预期再现时刻之前到达。如果分组的到达对于预期时刻的再现而言过迟,则会对再现质量造成影响。

[0133] PKT 1 和 2 被按时接收到,并且它们被再现而没有经历时间规整。PKT 3 和 PKT 4 二者在同一时刻 t_4' 被接收到。两个分组的接收时刻是满足要求的,因为每个分组都是在相

关的预期再现时刻（对于 PKT 3 为 t_4'' ，对于 PKT 4 为 t_5' ）之前接收到的。PKT 3 和 4 被按时再现而没有经历时间规整。当在 PKT 5 的预期再现时刻之后的时刻 t_6' 接收到 PKT 5 时，问题发生了。在预期再现时刻，擦除取代 PKT 5 而得到再现。随后，PKT 5 在擦除已经开始再现之后才到达。

[0134] 在没有采用规整的第一种情形下，PKT 5 被丢弃并且 PKT 6 被接收到并在下一预期再现时刻得到再现。应注意，在这种情况下，PKT 6 被及时接收到以用于再现。在第二种情形下，如果 PKT 5 和 PKT 5 之后的所有分组被延迟，则每个分组对于预期再现而言会到达得过迟，并会导致一串擦除。在这两种情形下，信息都被丢失，即，在第一种情形下 PKT 5 被丢弃；在第二种情形下 PKT 5 和随后的分组被丢失。

[0135] 可选地，使用 IBA 技术允许 PKT 5 在下一预期再现时刻得到再现，其中随后的分组从该点接续下来。IBA 防止了数据的丢失，但延迟了分组流。

[0136] 没有采用时间规整的这种再现会增加通信系统中的整个端对端延迟。如图 14 所示，分组间的延迟会导致丢失信息或者再现的延迟。

[0137] 通过实现时间规整，当 PKT 5 在其预期再现时刻之后到达时，分组被扩展并且可以避免擦除。例如，扩展 PKT 4 可以使再现以 23ms 完成而不是 20ms。PKT 5 在其被接收到时得到再现。这比发送擦除来进行再现的情况（如图 14 中所述的没有采用时间规整但采用了 IBA 的再现的可选实例所示）要更快。用扩展 PKT 4 来代替发送擦除，会使再现质量的恶化更小。因此，时间规整提供了更好的整体再现质量以及等待时间的减小。如图 14 所示，通过使用时间规整，与不使用时间规整技术的情况相比，PKT 5 之后的分组更早地得到了再现。在该具体实例中，当使用时间规整时，PKT 7 在时刻 t_9 得到再现，这要早于没有采用时间规整的情况。

[0138] 应用时间规整以改善再现质量，同时要考虑变化的工作条件以及话音传输中所传输信息的特性的变化。由于具有话音突峰和静音段的话音特性会变化，所以目标去抖动缓冲器延迟长度以及用于每种类型的数据的压缩和扩展阈值可能是不同的。

[0139] 图 15 示出了随着话音突峰的不同，由于去抖动延迟的差异而引起的“静音压缩”和“静音扩展”的实例。在图 15 中，阴影区域 120、124 和 128 表示话音突峰，而无阴影的区域 122 和 126 表示接收到的信息的静音段。在被接收时，话音突峰 120 起始于时刻 t_1 并结束于时刻 t_2 。在接收器处，去抖动缓冲器延迟被引入，因此话音突峰 120 的再现开始于时刻 t_1' 。去抖动缓冲器延迟被标识为时刻 t_1' 与时刻 t_1 之间的差。在被接收时，静音段 122 起始于时刻 t_2 并结束于时刻 t_3 。静音段 122 被压缩并被再现为从时刻 t_2' 到 t_3' 的静音段 132，这要小于接收到的静音段 122 的原始持续时间。在信号源处，话音突峰 124 起始于时刻 t_3 并结束于时刻 t_4 。在接收器处，话音突峰 124 从时刻 t_3' 到时刻 t_4' 被再现。静音段 126（时刻 t_4 到 t_5 ）在接收器处进行再现时被扩展为静音段 136，其中 $(t_5' - t_4')$ 大于 $(t_5 - t_4)$ 。当去抖动缓冲器需要更早地再现分组时可以压缩静音段，并且当去抖动缓冲器需要延迟分组的再现时可以扩展静音段。在一个实例中，静音段的压缩或扩展会导致语音质量微小的恶化。因此，可以实现自适应去抖动延迟而不使语音质量恶化。在图 15 的实例中，自适应去抖动缓冲器压缩和扩展由该自适应去抖动缓冲器识别和控制的静音段。

[0140] 应注意，如在本文中使用的的那样，时间规整是指响应于到达时间和接收到的数据的长度而对再现进行自适应控制。时间规整可以通过在再现时压缩数据、在再现时扩展数

据或者在再现时压缩和扩展数据来实现。在一个实例中,使用阈值来触发压缩。在另一实例中,使用阈值来触发扩展。在又一实例中,使用两个触发:一个用于触发压缩,一个用于触发扩展。其它实例可以采用指示时间规整的各种级别(例如,不同速率的快速再现)的多个触发。

[0141] 也可以在解码器内部执行时间规整。用于执行解码器时间规整的技术被记述在2005年5月5日提交的标题为“Time Warping Frames Inside the Vocoder by Modifying the Residual”的共同未决的申请第11/123,467号中。

[0142] 在一个实例中,时间规整包含用于“融合”话音片段的方法。融合话音片段涉及比较至少两个连续的话音片段中的话音样本,并且如果在所比较的片段之间发现相关性,则生成至少两个连续片段的单一片段。话音的融合是在试图保持话音质量时进行的。保持话音质量并使引入到输出的话音中的瑕疵(诸如是提供给用户的质量恶化的声音,包括“咯嚓声”和“砰响声”)最小化,是通过仔细地选择要融合的分组来实现的。话音片段的选择基于片段的相似性或相关性。话音片段的相似性越接近,产生的话音质量就越好并且引入话音瑕疵的概率就越低。

[0143] 图16示出了随时间绘出的话音信号。纵轴表示信号的幅度;并且横轴表示时间。应注意,话音信号具有区别性的模式,其中话音信号的一些部分随时间重复。在该实例中,话音信号包括从时刻 t_1 到 t_2 的第一片段,其在 t_2 到 t_3 期间作为第二片段重复。当发现片段的这种重复时,片段中的一个或多个,诸如从时刻 t_2 到时刻 t_3 的片段,可以被消除而几乎不会对样本的再现质量造成影响或者实际上不会造成影响。

[0144] 在一个实例中,可以使用如在下文中给出的等式4来找出两个话音片段之间的关系。相关性是两个片段之间的关系的强度的度量。等式4提供了绝对和边界相关因子(从-1到1)作为关系的强度的度量,其中与反映较强关系(即较大的相关性)的较高的正数相比,较低的负数反映较弱的关系,即较小的相关性。如果等式4的应用指示“良好的相似性”,则执行时间规整。如果等式4的应用显示出极弱的相似性,则瑕疵可能会存在于融合的话音片段中。相关性给出如下:

$$[0145] \quad Corr(d) = \frac{\sum_i [(x(i) - mx) \times (y(i - d) - my)]}{\sqrt{\sum_i (x(i) - mx)^2} \sqrt{\sum_i (y(i - d) - my)^2}} \quad (4)$$

[0146] 在等式4中, x 和 y 表示两个话音片段, m 表示在其范围内计算两个片段之间的相关性的窗口, d 表示相关性部分,并且 i 是索引。如果等式4的应用指示可以融合片段而不会引入瑕疵,则可以使用“重叠相加”技术来进行融合。重叠相加技术将被比较的片段结合在一起并从两个分离的话音片段中生成一个话音片段。使用重叠相加的结合,可以基于诸如等式5的等式,其中等式5给出如下:

$$[0147] \quad a) \quad OutSegment[i] = \frac{(Segment1(i) * (WindowSize - i) + (Segment2(i) * i))}{WindowSize}$$

$$[0148] \quad b) \quad OutSegment[i] = \frac{(Segment2(i) * (WindowSize - i) + (Segment1(i) * i))}{WindowSize} \quad (5)$$

$$[0149] \quad i = 0..WindowSize-1 \quad WindowSize = RWindowSize$$

[0150] 合成的样本可以是脉冲编码调制(PCM)样本。每个PCM样本具有定义位长度的预

定格式和 PCM 样本的格式。例如,16 位带符号数可以是表示 PCM 样本的格式。通过应用等式 5 而产生的重叠相加技术,包括进行加权以提供片段 1 的第一个 PCM 样本和片段 2 的最后一个 PCM 样本之间的平滑过渡。在等式 5 中,“RWindowSize”是参考窗口中的 PCM 样本的数目,并且“OutSegment”是所产生的重叠相加后的片段的大小。“WindowSize”等于参考窗口大小并且“Segment”是目标片段大小。这些变量是取决于采样速率、语音的频率内容以及质量和计算复杂度之间的期望折衷而确定的。

[0151] 图 17A 和 17B 中示出了上述的重叠相加技术。在图 17A 中,示出了由 160 个 PCM 样本组成的语音片段。在该实例中,RWindowSize 由 PCM 样本 0-47 表示。换言之,PCM 样本 0-47 与大小为 WindowSize 的参考窗口中的样本的数目相对应。Segment 是指目标搜索区域的大小并由 PCM 样本 10-104 表示。在该实例中,将 PCM 样本 0-47 与样本 10-104 进行比较,一次比较一个 PCM 样本,来找出参考样本与目标搜索区域之间的最佳相关性。目标搜索区域内发现最大相关性的位置被称为“偏移”。在偏移点处,可以将 RWindowSize 与对应于 RWindowSize 大小的 Segment 的部分结合。使对应于 PCM 样本 104-160 的语音片段保持不变。

[0152] 在图 17B 中,以一次比较一个 PCM 样本的方式将语音片段的第一个 RWindowSize 样本与语音片段的随后的部分进行比较。发现 RWindowSize 与目标搜索区域 (Segment) 内相应长度的样本之间的最大相关性的位置是“偏移”。偏移的长度是从语音片段的开头到发现 RWindowSize 与 Segment 之间的最大相关性的点之间的距离。一旦发现最大相关性,就(在偏移点处)将 RWindowSize 与相应长度的 Segment 融合。换言之,通过将 RWindowSize 加到 Segment 内的相同长度的部分上,来执行重叠相加。这是在如图所示的偏移点处进行的。如图所示,从原始片段中复制剩余的样本。所产生的语音样本是通过将从原始语音片段中复制的剩余样本附加在融合的样本上而构成的,如图所示。所产生的分组比原始片段短偏移的长度更短。该处理被称为语音压缩。压缩的语音片段越少,检测到任何质量恶化的概率就越低。

[0153] 当去抖动缓冲器包含较少数目的语音分组时,执行语音扩展。如果去抖动缓冲器具有较少数目的语音分组,则下溢的概率会增加。当发生下溢时,去抖动缓冲器可以将擦除馈送给解码器。然而,这会导致语音质量的恶化。为了防止语音质量的这种恶化,可以延迟去抖动缓冲器中的最后几个分组的再现。这是通过扩展分组来实现的。

[0154] 语音扩展可以通过重复语音片段的多个 PCM 样本来实现。重复多个 PCM 样本同时避免瑕疵或音高平坦 (pitch flatness) 是通过采用比执行语音时间压缩时更多的 PCM 语音样本进行工作来实现的。例如,用于实现语音扩展的 PCM 样本的数目可以是语音时间压缩中使用的 PCM 样本的数目的两倍。可以从先前再现过的语音分组中获得额外的 PCM 样本。

[0155] 图 18A 示出了语音扩展的一个实例,其中每个分组或语音片段的长度为 160 个 PCM 样本并且“预扩展的”语音片段被生成。在该实例中,将两个语音片段(即,“当前”语音片段和“先前”语音片段)进行比较。将当前语音片段的第一个 RWindowSize PCM 样本选择为参考样本。将这些 RWindowSize 样本与先前语音分组的 Segment 进行比较,其中具有最大相关性的点(或偏移)被确定。在偏移点处将 RWindowSize PCM 样本与先前分组内的相应大小的 Segment 重叠相加。通过复制并将其余的样本从先前语音片段附加到经过重叠相加的片段上来生成预扩展的语音片段,如图 18A 所示。则扩展后的语音片段的长度是预扩

展片段的长度加上当前话音片段的长度,如图 18A 所示。在该实例中,PCM 样本从话音片段的开头处被偏移。

[0156] 在另一实例中,如图 18B 所示扩展当前分组或话样本。参考样本 RWindowSize 位于当前话音片段的开头处。将 RWindowSize 与其余的当前话音分组进行比较直到定位出具有最大相关性的点(偏移)。将参考样本与当前话音片段内被发现具有最大相关性的相应 PCM 样本重叠相加。然后通过复制开始于分组的开头处并结束于偏移点的 PCM 样本,将重叠相加后的片段附加到该 PCM 样本上,并复制和附加来自当前分组的未修改的剩余的 PCM 样本,来生成扩展后的话音片段。扩展后的话音片段的长度等于偏移加上原始分组的长度的总和。

[0157] 在另一实例中,如图 18C 所示扩展话音,其中 RWindowSize 嵌在当前分组或话音片段内并且不发生在分组的开头处。Roffset 是与当前分组的开头到 RWindowSize 所起始的点之间的距离相对应的话音片段的长度。将 RWindowSize 与当前分组中具有最大相关性的点处发现的相应大小的 PCM 样本重叠相加。然后通过复制开始于原始分组或当前分组的开头处并结束于偏移处的 PCM 样本,并附加重叠相加后的片段和来自原始分组的剩余的 PCM 样本,来生成扩展后的话音片段。所产生的扩展后的话音片段的长度是原始分组的长度加上偏移减去 Roffset 样本(即如上面定义的 Roffset 中的 PCM 样本的数目)。

[0158] 经滤波的时间规整阈值

[0159] 为了避免压缩和扩展决定的摆动,当自适应去抖动缓冲器中所存储的分组的数目快速变化时,使用变量来估计自适应去抖动缓冲器的状态,即自适应去抖动缓冲器中所存储的分组数目,一个实例在采样窗口上滤波这样的变量。自适应去抖动缓冲器的状态可以指自适应去抖动缓冲器中所存储的分组数目或者用于估计自适应去抖动缓冲器中所存储的数据的任何变量。在支持突发数据传递的称为 1xEV-DO 的 IS-856 系统中,向给定接收器传递的分组被时分复用在向前链路上,接收器可能会在一个时刻接收若干分组,在之后的一端时间内没有分组被接收到。这导致在接收器的自适应去抖动缓冲器处接收到脉冲串形式的数据。接收到的数据实际上会经历“集束”(bundling),其中可能会存在两个或更多分组在时间上相互接近地一起到达的情况。这样的集束会很容易导致分组的扩展和压缩之间的摆动,其中自适应去抖动缓冲器响应于接收到的数据的速率和缓冲器的状态而提供时间规整指示。例如,考虑在话音突峰的开头时,去抖动缓冲器的计算出的值(延迟或长度)为 40ms 的情况。在稍后的时刻,去抖动缓冲器载荷落到扩展阈值以下,从而导致扩展数据分组的决定。紧随该分组的再现之后,集束的三个分组到达了;到达的数据填充了去抖动缓冲器的大小使得压缩阈值被超过。这将导致分组被压缩。由于集束的分组到达之后可能跟随着一段时间内没有分组到达的状态,所以去抖动缓冲器可能会再次被耗尽,从而导致分组被扩展。扩展和压缩之间的这种来回切换会导致要进行时间规整的分组的高百分比。这是不希望有的,因为我们希望将信号信息由于时间规整而被修改的分组的百分比限制到很小的值。

[0160] 一个实例通过使集束现象可能会对自适应去抖动缓冲器的自适应控制以及数据的时间规整和再现所造成的影响变平滑,来避免这种摆动。该实例使用平均值来确定何时进行时间规整。平均值是通过滤波这种计算中所使用的变量来计算的。在一个实例中,压缩和扩展阈值是通过对去抖动缓冲器的大小进行滤波或求平均来确定的。应注意,缓冲器

的大小是指缓冲器的当前状态。

[0161] 将缓冲器大小的经滤波的值与扩展阈值进行比较,会导致更高数目的下溢,因为通过使用未滤波的值可能已经被扩展的一些分组,在使用经滤波的值的条件下没被扩展。另一方面,将经滤波的值与压缩阈值进行比较,可以用来抑制大部分的摆动(或时间规整控制之间的来回切换)而只有最小的负面影响或实际上没有负面影响。因此,可以以不同的方式对待压缩和扩展阈值。

[0162] 在一个实例中,对照扩展阈值来检查自适应去抖动缓冲器大小的即时值。相反,对照压缩阈值来检查去抖动缓冲器的经滤波的值。一种配置使用无限脉冲响应(IIR)滤波器来确定自适应去抖动缓冲器的平均大小,其中自适应去抖动缓冲器具有可被周期性(诸如每 60ms 计算一次)重新计算的经滤波的值。滤波器时间常数可以从集束统计数据中导出并且用于 1xEV-DO Rev A 的该时间常数的一个实例可以是 60msec。使用集束统计数据来导出滤波器时间常数,这是因为它们与去抖动缓冲器的即时大小在工作期间的摆动情况之间具有很强的相关性。

[0163] 由于缺失分组而引起的扩展

[0164] 如上文中所述,自适应去抖动缓冲器和用于控制自适应去抖动缓冲器及控制接收到的数据的时间规整的各种方法可以适于具体的系统规范和工作条件。对于实现重复请求方案(诸如混合自动重复请求(H-ARQ)方案)以改善性能的通信系统,这样的重复处理具有关于如何扩展话音分组的暗示。具体而言,H-ARQ 会导致分组重新排序(即失序)地到达。考虑图 19,其示出了具有一定长度的去抖动缓冲器和给定为目标去抖动缓冲器长度的 50%的扩展阈值 T_{expand} 。正被再现的当前分组具有序号 20,它是 PKT 20。去抖动缓冲器包含分别具有序号 21、23 和 24 并标为 PKT 21、PKT 23 和 PKT 24 的三个分组。当再现装置在再现了 PKT 20 之后请求下一分组时,扩展阈值不会触发,因为去抖动缓冲器包含足以将缓冲器长度维持在计算出的去抖动缓冲器长度的 50%以上的分组。在本实例中,PKT 21 因此没有被扩展。如果 PKT22 直到 PKT 21 结束再现的时刻还未到达,则这会导致下溢,因为分组是按照顺序再现的,并且因此再现装置不会在 PKT 22 之前再现 PKT23。即使扩展阈值不会触发,一个实例也预期到接收到的分组中的不连续,并选择扩展 PKT 21 以便为 PKT 22 的到达留出更多的时间。这样,PKT 21 的扩展可以避免分组缺失和擦除。因此,即使去抖动缓冲器长度超出了扩展阈值 T_{expand} ,也可以扩展分组。

[0165] 可以增强应对分组进行扩展的条件。如上文中所述,如果去抖动缓冲器大小低于扩展阈值,则可以扩展分组。在另一种情形下,如果具有下一序号的分组没有存在于去抖动缓冲器中,则可以扩展分组。

[0166] 如前所述,可以在话音突峰的开头处计算去抖动缓冲器延迟。由于包括但不限于信道条件和负载条件的网络条件可能会在话音突峰期间(特别是在长话音突峰期间)发生变化,所以将一个实例配置成在话音突峰期间改变去抖动缓冲器延迟。这样,可以在话音突峰期间每过 CHANGE_JITTER_TIME 秒就周期性地重新计算上文中给出的去抖动缓冲器等式。可选地,可以在触发事件(诸如工作条件、负载、空中接口指示的显著变化或者其它事件)发生时重新计算变量。在一个实例中,可以将 CHANGE_JITTER_TIME 的值设置成 0.2 秒(200ms)。

[0167] 时间规整阈值,例如压缩和扩展阈值,可以在话音突峰期间提供关于如何改变取

值的引导。正常操作是指当自适应去抖动缓冲器状态在压缩和扩展阈值之间并在目标去抖动缓冲器长度周围时,接收器的操作。每个阈值充当触发。当阈值被达到或违反时,可以取决于阈值而扩展或压缩自适应去抖动缓冲器中的分组。自适应去抖动缓冲器的大小在它接收到分组时可以继续扩展或收缩。自适应去抖动缓冲器大小的这种持续的变化表明在通信期间可能会连续地接近压缩和扩展阈值。通常,系统试图将自适应去抖动缓冲器大小保持在扩展和压缩阈值之间,这被认为是稳定状态。在稳定状态下,自适应去抖动缓冲器的大小不变化;并且分组接收中的变化以及因此产生的自适应去抖动缓冲器大小的变化,可以自动地使压缩/扩展阈值触发并相应地对分组进行压缩/扩展,直到达到新的自适应去抖动缓冲器延迟。在这种情形下,自适应去抖动缓冲器目标延迟长度根据 CHANGE_JITTER_TIME 而被更新。可以不必要计算去抖动缓冲器的实际大小,因为去抖动缓冲器大小在由于达到时间规整扩展/压缩阈值而引起触发时会自动地变化。在一个实例中,可以将 CHANGE_JITTER_TIME 的值设置成 0.2 秒 (200ms)。

[0168] 越区切换预规整

[0169] 典型地,越区切换伴随着短时间的覆盖损失。当越区切换即将来临时,AT 会经历较差的信道条件和增加的分组延迟。一个实例以对话音分组应用时间规整的特殊方式来处理越区切换状况。AT 一决定越区切换到新的基站,该信息就被用于控制去抖动缓冲器。在接收到该越区切换信号后,AT 进入“预规整”模式。在该模式下,AT 扩展分组直到两个条件中的一个被满足为止。在第一种条件下,去抖动缓冲器继续累积分组并且累积起来的扩展产生为 PRE_WARPING_EXPANSION 的去抖动缓冲器大小。换言之,对分组进行扩展直到达到 PRE_WARPING_EXPANSION。可选地,在第二种条件下,时间段 WARPING_TIME 已被满足。计时器在接收到越区切换信号或中断指示符时开始计时;接收器在 WARPING_TIME 时停止计时。一旦这两个条件中的一个已被满足,AT 就退出预规整模式。在预规整模式期间,除非(稍后描述的)End_Talkspurt 条件被满足,否则不压缩任何分组,因为去抖动缓冲器希望累积足够的分组以便以规则的间隔将它们发送给再现装置。在分组以规则间隔(例如 20ms)被预期的实例中,可以将 PRE_WARPING_EXPANSION 的值设置成 40ms,并将 WARPING_TIME 的值设置成等于 100 个时隙 (166ms)。

[0170] 越区切换仅是中断事件的一种形式。去抖动缓冲器可以实现用于处理越区切换或其它类型的中断的机制。为此,所需的信息是:需要多少去抖动超额量来处理中断 (PRE_WARPING_EXPANSION) 和去抖动缓冲器将持续工作在这种中断避免模式中多长时间 (WARPING_TIME)。

[0171] 计算延迟下溢的数目

[0172] 由于上文中提供的自适应去抖动缓冲器等式被设计成以延迟下溢的百分比为目标,所以合乎需要的是准确地测量延迟下溢的数目。当发生下溢时,并不知道该下溢是由于分组延迟引起的还是由于在网络中某处(即,传输路径)被丢弃的分组引起的。因此需要准确地确定下溢的类型。

[0173] 在一个实例中,对于使用 RTP/UDP/IP 的通信,每个分组包括 RTP 序号。序号被用于以接收到的分组的传输顺序来排列接收到的分组。当发生下溢时,引起下溢的分组的 RTP 序号会存储在诸如存储器阵列的存储器中。如果具有识别出的序号的分组较迟地到达,则将该下溢作为“延迟下溢”而计数。

[0174] “延迟下溢率”是下溢的数目与接收到的分组的总数的比率。在每次更新去抖动缓冲器等式时,将下溢的数目和接收到的分组的数目都设置成零。

[0175] 对话音突峰开头和结尾处的增强

[0176] 考虑图 20,其示出了两个用户之间的会话的时间线。在该图中,纵轴表示时间。每个用户发送话音突峰和静音段,它们然后由另一用户接收。为了清楚起见,带阴影的块段 400 和 410 表示用户 1 的话音突峰(话音片段)。未带阴影的块段 405 表示用户 2 的话音突峰。时间线上的话音突峰外部的区域表示用户没有讲话但是可能在听另一用户讲话或在接收静音段的时间。片段 400 在用户 2 处得到再现。一旦话音片段 400 在用户 2 处结束再现,用户 2 就在开始讲话之前等待短时间间隔。用户 2 的第一个话音片段 405 的开头随后由用户 1 听到。用户 1 所感觉到的会话往返行程延迟(RTD)是用户 1 停止讲话的时刻到用户 1 听到用户 2 的话音片段的开头的时刻之间的时间间隔。会话 RTD 不是单向的端对端延迟,而是用户特定的并且从用户的观点来看是很重要的。例如,如果会话 RTD 对于用户 1 而言过大,这将促使用户 1 再次开始讲话而不等待用户 2 的话音片段得到再现。这打断了会话流并且会被感觉为会话质量的恶化。

[0177] 可以以不同的方式改变用户 1 所体验的会话 RTD。在一个实例中,可以改变用户 1 的话音片段的结尾被再现给用户 2 的时刻。在第二实例中,改变的是用户 2 的话音片段的开头被再现给用户 1 的时刻。应注意,只有话音突峰的开头和结尾的延迟才会影响会话中的语音质量。设计目标是进一步减小在话音突峰开头和结尾处的延迟。

[0178] 在一个实例中,目标是增强话音突峰的开头。可以通过操作用户 1 的话音突峰的第一个分组使得收听者用户 2 比实现默认自适应去抖动缓冲器延迟的情况更早地接收到分组,来实现这种增强。施加给自适应去抖动缓冲器中的分组的延迟可以是默认自适应去抖动缓冲器延迟、计算出的值或者是被选择成使得收听者能够在特定时刻接收到分组的值。在一种情形下,通过在每个接收到的话音突峰的开头处重新计算自适应去抖动缓冲器延迟来改变话音突峰的第一个分组的定时。当减小施加给话音突峰的第一个分组的自适应去抖动缓冲器延迟时,该第一个分组被加快地发送给收听者。当增加施加的延迟时,该第一个分组在较迟的时刻由收听者接收到。用于第一个分组的默认去抖动缓冲器延迟可以小于计算出的去抖动缓冲器延迟并且也可以大于计算出的去抖动缓冲器延迟。在所示出的实例中,通过可以以秒来测量的被称为 MAX_BEGINNING_DELAY 的值,来限制每个话音突峰的第一个分组的去抖动延迟。该值可以是重新计算的去抖动缓冲器延迟或被设计成使得收听者能够在指定时刻接收到分组的延迟。

[0179] MAX_BEGINNING_DELAY 的值可以小于实际计算的去抖动缓冲器延迟。当 MAX_BEGINNING_DELAY 小于计算出的去抖动缓冲器延迟并被施加给话音突峰的第一个分组时,话音突峰的随后的分组将被自动地扩展。随后分组的自动扩展会发生,是因为去抖动缓冲器不能以它再现分组的相同速率接收到分组。当去抖动缓冲器再现分组时,去抖动缓冲器的大小发生减小并且扩展阈值被接近。一旦达到扩展阈值,扩展就被触发并且话音突峰中的随后的分组被扩展直到去抖动缓冲器接收到足以超过扩展阈值的到来的分组。通过实现 MAX_BEGINNING_DELAY 值,收听者更早地接收到了话音突峰的第一个分组,同时随后的分组被扩展。收听者对更早地接收到初始分组感到满意。增强话音突峰的开头具有使下溢的数目发生小量增加的潜在可能;然而,适当的 MAX_BEGINNING_DELAY 值会减轻这种影响。

[0180] 在一个实例中,MAX_BEGINNING_DELAY 的值被计算为实际去抖动目标的一小部分;举例来说,大小为 TARGET DE-JITTER BUFFERLENGTH 的 0.7 倍的 MAX_BEGINNING_DELAY 值会导致下溢的微小的增加。在另一实例中,MAX_BEGINNING_DELAY 值可以是诸如 40ms 的固定数字,这在例如支持 1xEV-DO Rev A 的系统中会导致下溢的微小增加。

[0181] 扩展话音突峰中的随后的分组不会使整体语音质量恶化。这在图 20 中被示出,其中用户 2 接收到来自用户 1 的话音突峰的第一个分组并且初始或“单向延迟”被限制为 T_{d1} 。如图所示,话音片段 400 在用户 2 处被接收到而没有经过任何扩展或压缩,然而,话音片段 405 在接收时在用户 1 处被压缩。

[0182] 图 21 是示出对话音突峰的开头的增强的流程图。首先在步骤 510 中确定系统是否处于静音模式。静音模式可以与话音突峰之间的静音段或者去抖动缓冲器没有正在接收分组的时间相对应。如果系统不处于静音模式,则处理结束。如果它处于静音模式,则在步骤 520 中执行目标去抖动长度估计。然后,在步骤 530 中确定是否增强系统。根据一个实例,增强指示计算出的目标自适应去抖动长度大于给定值,该给定值在一个实例中被给定为诸如 MAX_BEGINNING_DELAY 的增强因子;在步骤 540 中,系统等待等于增强因子或目标长度的一小部分的时间段以开始再现。如果系统不被增强,则在步骤 550 中,系统等待新目标长度以开始再现。新目标的值可以等于计算出的目标去抖动缓冲器长度或最大去抖动缓冲器长度。

[0183] 图 22 也示出了对话音突峰的开头的增强。图中示出的处理 580 开始于对话音突峰的识别。考虑两种情形:i) 采用时间规整;和 ii) 没有采用时间规整。在该实例中,使用 20ms 长度的话音分组。可以实现任意长度的话音分组。此处,自适应去抖动缓冲器在再现分组之前等待 120ms。该值是自适应目标去抖动缓冲器长度并在步骤 582 中从自适应去抖动缓冲器目标估计器接收到。在本实例中,120ms 等效于在不采用时间规整的情况下接收六(6)个分组,每个分组的长度为 20ms。如果在步骤 584,时间规整没有被使用,则在 120ms 内提供了六(6)个分组。因此,在第一种情形下,去抖动缓冲器在接收到六个分组之后将开始再现分组。这在时间上等效于 120ms 的延迟。在第二种情形下,通过实施时间规整,去抖动缓冲器可以扩展接收到的前四(4)个分组并在接收到四(4)个分组之后开始再现分组。因此,即使这种情况下的 80ms 的去抖动缓冲器延迟小于估计的 120ms 的去抖动缓冲器延迟,也可以通过扩展前面几个分组来避免潜在的下溢。换言之,采用时间规整的分组再现可以比没有采用时间规整的分组再现开始得更早。因此,时间规整可以用于增强话音突峰的开头而不会影响下溢的数目。

[0184] 在另一实例中,可以增强话音突峰的结尾。这是通过压缩话音突峰的最后几个分组,从而减小端对端延迟来实现的。换言之,话音突峰结尾处的延迟被变得更小并且第二用户更快地从第一用户听到了回应。图 23 中示出了对话音突峰结尾的增强。此处,1/8 速率分组指示话音突峰的结尾。这不同于可以用于传输语音数据的全速率(速率 1)分组、半速率(速率 1/2)分组或四分之一速率(速率 1/4)分组。也可以将其它速率的分组用于静音段期间或话音突峰结尾处的传输。在语音通信中实施 1/8 速率分组作为静音指示符分组,被进一步记述在优先权日期为 2005 年 2 月 1 日、标题为“METHOD FOR DISCONTINUOUS TRANSMISSION AND ACCURATE REPRODUCTION OF BACKGROUND NOISE INFORMATION”的共同未决的美国专利申请第 11/123,478 号中。

[0185] 如图 23 所示,在没有采用时间规整的情况下,分组 N 至分组 N+4 在 100ms 中被再现。通过压缩话音突峰的最后几个分组,相同的分组 N 至分组 N+4 可以在 70ms 中得到再现而不是在 100ms 中得到再现。

[0186] 当实施时间压缩时,话音质量几乎没有恶化或实际上没有恶化。对话音突峰的结尾的增强,假定了接收器知晓对话音突峰结尾的识别,并能在接近结尾时预期到它。

[0187] 当在一个实例中通过实时传输协议 (RTP) 发送语音分组时,可以在每个话音突峰的最后—个分组中设置“话音突峰结尾”指示符。当正在提供分组以进行再现时,在去抖动缓冲器中的分组中检查“话音突峰结尾”指示符。如果该指示符被设置在其中一个分组中并且在正被提供以用于再现的当前分组和“话音突峰结尾”分组之间没有缺失的序号,则对正被提供以用于再现的分组以及当前话音突峰的所有将来的分组进行压缩。

[0188] 在另一实例中,如果系统处于话音突峰中并且 1/8 速率分组或设置有静音指示符描述 (SID) 位的分组被传递给再现装置,则系统转换到静音。可以通过检查 1/8 速率分组的大小而检测到 1/8 速率分组。SID 位被携带在 RTP 信头中。如果系统处于静音状态,并且既不是 1/8 速率分组也不是设置有 SID 位的分组被传递以用于再现时,系统转换到话音突峰状态。应注意,在一个实例中,如本文中提供的自适应去抖动缓冲方法可以在系统处于话音突峰状态时被执行,并且可以在系统处于静音段时被忽略。

[0189] 应注意,该方法可以正确地丢弃迟到的重复分组。如果重复分组到达,则它将被简单地丢弃,因为分组的第一个实例已在适当的时刻得到再现并且其序号没有被保存在包含“延迟下溢”候选者的阵列中。

[0190] 当在一个实例中通过 RTP 发送语音分组时,可以在每个话音突峰的最后—个分组中设置“话音突峰结尾”指示符。当正在提供分组以进行再现时,在去抖动缓冲器中的分组中检查“话音突峰结尾”指示符。如果该指示符被设置在其中一个分组中并且在正被提供以用于再现的当前分组和“话音突峰结尾”分组之间没有缺失的序号,则对正被提供以用于再现的分组以及当前话音突峰的所有将来的分组进行压缩。

[0191] 图 24 中示出了根据一个实例的对话音突峰结尾的增强的流程图。新分组开始于步骤 600。在步骤 605 中,如果去抖动缓冲器长度大于或等于压缩阈值,则在步骤 635 中生成压缩指示并且在步骤 600 中将结尾提供给新分组。在步骤 605 中,如果去抖动缓冲器不大于或等于压缩阈值,则在步骤 610 中确定去抖动缓冲器长度是否小于或等于扩展阈值。如果其小于或等于扩展阈值,则步骤 615 确定结尾是否等于表示静音段或话音突峰结尾的分组速率。在一个实例中,可以在静音段期间或在话音突峰结尾处,以例如 20ms 的恒定间隔发送连续的一串 1/8 速率分组。在图 24 中,如果在步骤 615 中确定结尾不等于 1/8 速率分组,则在步骤 620 中扩展片段并返回到步骤 600 中的新分组。步骤 625 确定结尾是否等于 1/8。在步骤 625 中,如果结尾等于 1/8 速率,则在步骤 635 中生成压缩指示。如果它不等于 1/8 速率,则在步骤 630 中,再现是正常的,而不采用任何时间规整。

[0192] 时间规整质量优化器

[0193] 当许多连续的分组被压缩(或扩展)时,这会显著地使音频加速(或减速)并会导致质量的恶化。可以通过隔开经时间规整的分组,即在经过时间规整的分组后跟随有一些未经过时间规整的分组以后再规整另一分组,来避免这种恶化。

[0194] 如果将上述隔开经规整的分组的方法应用于扩展,则会使一些本来会经历扩展的

分组不经历扩展。这会导致下溢,因为分组的扩展是在去抖动缓冲器耗尽分组时执行的。因此,在一个实例中,可以将上述隔开经规整的分组的方法应用于压缩分组,即在经过压缩的分组后可以跟随有一些未经过压缩的分组,之后才可以压缩另一个分组。典型地,在两个经压缩的分组之间不应被压缩的这种分组的数目可以设置成 2 或 3。

[0195] 触发时间规整的条件的设置

[0196] 此处描述的是用于触发语音分组的时间规整(扩展/压缩)的许多条件。以下是用于确定将对分组进行压缩、进行扩展还是既不压缩也不扩展的(伪码形式的)规则的组合集。

[0197] If(in Pre-Warping(Handoff Detected)Phase and no End of Talkspurt Detected)and DEJITTER_TARGET+PRE_WARPING_EXPANSION not reached)

[0198] Expand Packet

[0199] End If

[0200] Else

[0201] If(End of Talkspurt Detected)

[0202] Compress

[0203] End if

[0204] Else

[0205] If(Compress Threshold Triggered)

[0206] Compress

[0207] End if

[0208] Else If(Expand Threshold Triggered or Next Packet not in Queue)

[0209] Expand

[0210] End If

[0211] End If

[0212] End If.

[0213] 图 25 示出了与解码器功能耦合的常规去抖动缓冲器的实现方案。

[0214] 在图 25 中,预期分组以 20ms 间隔到达去抖动缓冲器。在该实例中可以观察到,分组以不规则的间隔到达,即具有抖动。去抖动缓冲器累积分组直到达到特定去抖动缓冲器长度,以便一旦其开始以诸如 20ms 的规则间隔发送出分组,去抖动缓冲器就不会被耗尽。在到达所需要的去抖动缓冲器长度时,去抖动缓冲器开始以 20ms 的规则间隔开始再现分组。解码器以规则间隔接收这些分组并将每个分组转换成每分组为 20ms 的语音。可选实例可以选择其它时间间隔。

[0215] 作为比较,图 26 示出了支持时间规整的自适应去抖动缓冲器的实例。此处,分组以不规则的间隔到达自适应去抖动缓冲器。然而在这种情况下,目标去抖动缓冲器长度小得多。这是因为时间规整允许在去抖动缓冲器开始耗尽时扩展分组,从而为自适应去抖动缓冲器得到补充留出了时间。解码器可以在自适应去抖动缓冲器开始耗尽时扩展分组,并在自适应去抖动缓冲器开始累积过多的分组时压缩分组。可以观察到,不均匀传递的语音分组从自适应去抖动缓冲器被输入到解码器和时间规整单元中。由于采用了时间规整,这

些分组被允许以不规则的间隔到达,解码器取决于原始分组的到达时间而将每个分组转换成不同长度的语音分组。例如,在该实例中,解码器将每个分组转换成每分组为 15-35ms 的语音。由于分组因为时间规整而可以更早地得到再现,所以所需的缓冲器大小变得更小,从而在网络中产生更小的等待时间。

[0216] 图 27 是示出根据一个实例的 AT 的框图。自适应去抖动缓冲器 706、时间规整控制单元 718、接收电路 714、处理器 722 的控制、存储器 710、发射电路 712、解码器 708、H-ARQ 控制 720、编码器 716、话音处理 724、话音突峰 ID 726、差错校正 704 可以如前面的实施例中所示耦合在一起。另外,它们可以经由图 27 中所示的通信总线 702 耦合在一起。

[0217] 图 28 示出了一个实例中的分组处理,其中分组由去抖动缓冲器接收并最终由扬声器再现。如图所示,分组在去抖动缓冲器处被接收。去抖动缓冲器在从解码器接收到分组请求后向解码器发送分组和时间规整信息。解码器在从输出驱动器接收到请求后向输出驱动器发送样本。

[0218] 去抖动缓冲器内的输入控制器保持对到来的分组的跟踪,并且如果在到来的分组中存在差错则给出指示。去抖动缓冲器可以接收具有序号的分组。例如,差错可以在到来分组的序号低于先前分组的序号时,由输入控制器检测到。图 28 中位于输入控制器内的分类单元对到来的分组进行分类。由分类单元定义的不同类别可以包括“良好的分组”、“延迟的分组”、“差分组”等。此外,输入控制单元可以对分组进行比较并将该信息发送给去抖动缓冲器控制器。

[0219] 图 28 中示出的去抖动缓冲器控制器接收来自去抖动缓冲器的输入和输出控制器的双向输入。去抖动缓冲器控制器接收来自输入控制器的数据,其中这样的数据指示到来数据的特性,诸如接收到的良好分组的数目、接收到的差分组的数目等。去抖动缓冲器可以使用该信息来确定何时需要使去抖动缓冲器收缩或增长,这会产生发送给时间规整控制器的用于压缩或扩展的信号。去抖动缓冲器控制器单元内的分组差错率 (PER) 单元计算 PER 延迟。去抖动缓冲器的输出控制器从去抖动缓冲器请求分组。去抖动缓冲器的输出控制器单元还可以指示最后一个被再现的分组。

[0220] 解码器将分组请求发送给去抖动缓冲器,并在这样的请求后接收来自去抖动缓冲器的分组。解码器内的时间规整控制单元从去抖动缓冲器的输出控制器接收时间规整控制信息。时间规整控制信息指示将对分组进行压缩、扩展还是使其保持不变。由解码器接收的分组被解码并转换成话音样本;并且在从输出驱动器内的缓冲器接收到请求后,样本被发送给输出驱动器。来自输出驱动器的样本请求由解码器内的输出控制器接收。

[0221] 相位匹配

[0222] 如前所述,在分组的预期再现时刻之后接收到分组,会导致擦除取代延迟分组而得到再现。接收到自适应去抖动缓冲器处的擦除或缺失的分组,会导致解码后的话音的不连续。当自适应去抖动缓冲器辨认出潜在的不连续时,自适应去抖动缓冲器可以请求解码器执行相位匹配。如图 28 所示,自适应去抖动缓冲器 750 可以包括接收来自输出控制器 760 的输入的相位匹配控制器。相位匹配控制信息被发送给可以位于解码器 762 中的相位匹配单元。在一个实例中,相位匹配控制信息可以包括“相位偏移”和“游程长度”(run length) 信息。相位偏移是解码器已经解码的分组数目和编码器已经编码的分组数目之间的差。游程长度是指就在当前分组的解码之前,解码器已经解码的连续擦除的数目。

[0223] 在一个实例中,相位匹配和时间规整都在具有公共控制代码或软件的解码器中实现。在一个实例中,解码器实现波形内插,其中:

[0224] a) 如果没有使用时间规整和相位匹配,则使用采用 160 个样本的波形内插 (waveform_interpolation) 来进行语音编码;

[0225] b) 如果使用了时间规整并且没有使用相位匹配,则使用采用 $(160+-N*$ 音高周期) 个样本的波形内插解码 (waveform_interpolation_decoding) 来进行语音编码,其中 N 可以是 1 或 2;

[0226] c) 如果没有使用时间规整并且使用了相位匹配,则使用采用 $(160-\Delta)$ 个样本的波形内插解码 (waveform_interpolation_decoding) 来进行语音编码,其中 Δ 是相位匹配量。

[0227] d) 如果使用了相位匹配和时间规整二者,则使用采用 $(160-\Delta+-N*$ 音高周期) 个样本的波形内插解码 (waveform_interpolation_decoding) 来进行语音编码,其中 Δ 是相位匹配量。

[0228] 对输出驱动器的时钟输入,确定了该输出驱动器内的缓冲器对数据进行请求的频率。这是系统中的主要时钟并且可以以许多不同的方式实现。系统的主时钟可以通过 PCM 样本的采样速率导出。例如,如果正在通信的是窄带语音,则系统每秒再现 8000 个 PCM 样本 (8KHz)。该时钟可以驱动系统的其余部分。一种方法是让音频接口 770 在需要更多样本时从解码器请求更多样本。另一种方法是让解码器/时间规整单元独立地运行,由于该模块知道先前传递了多少 PCM 样本,所以它知道接下来何时提供更多的样本。

[0229] 调度器可以位于解码器 762 中或音频接口和控制单元 810 中。当位于音频接口控制单元 810 中时,调度器基于接收到的 PCM 样本的数目来确定对分组的下一请求。当调度器位于解码器中时,调度器可以每过 t ms 就请求分组。例如,解码器调度器可以每 2ms 就从自适应去抖动缓冲器 750 请求分组。如果在解码器中没有启用时间规整,或者如果时间规整单元不位于解码器 762 中,则调度器将与 1 个分组中的样本数目恰好相对应的一组样本发送给音频接口和控制单元 770。例如,在音频接口单元 770 每过 2ms 就请求样本的情况下,解码器的输出控制器 766 发送 16 个 PCM 样本 (1 个分组对应于 20ms,即 8KHz 采样速率下的语音数据的 160 个样本)。换言之,当时间规整控制器位于解码器外部时,解码器的输出是用于样本转换的普通分组。音频接口单元 770 将样本的数目转换成在解码器执行时间规整的情况下它会接收到的样本的数目。

[0230] 在另一种情形下,当时间规整控制器位于解码器内并且时间规整被启用时,在压缩模式下,解码器可以输出更少的样本;并且在扩展模式下,解码器可以输出更多的样本。

[0231] 图 30 进一步示出了调度功能由解码器实现的情形。在步骤 902,解码器从去抖动缓冲器请求分组。在步骤 904,分组被接收。在步骤 906 中,分组被转换成“N”个样本。在步骤 908 中,“N”个产生的样本被传递给音频接口控制单元,并且在步骤 910 中,将下一分组请求调度为 N 的函数。

[0232] 图 31 示出了解码器外部的音频接口和控制单元中的调度。在步骤 1002,音频接口单元首先请求一组 PCM 样本。在步骤 1004,接收到所请求的 PCM 样本,并且在步骤 1006 中,将下一分组请求调度为 N 的函数。

[0233] 时间规整指示符 (诸如,不进行时间规整指示符) 可以是来自自适应去抖动缓冲

器的指令的一部分。图 32 示出了在解码器外部计算调度方案的例如音频接口和控制单元中的时间规整单元。分组类型、时间规整指示符和要进行的规整量被输入给时间规整单元。

[0234] 图 33 示出了在解码器中的时间规整单元中计算调度方案的时间规整单元。对时间规整单元的输入包括分组类型、时间规整指示符和要进行的规整量。规整量和启用信号被输入给时间规整单元的质量优化单元。时间规整信息被输出。

[0235] 尽管本说明书描述了本发明的特定实例,但是普通技术人员能在不脱离本发明概念的情况下提出本发明的变形。例如,本文中的教导是参照电路交换网络元件进行说明的,但是其等效地适用于分组交换域网络元件。此外,本文的教导不限于认证三元组对,还可适用于包括两个 SRES 值(一个是通常的格式,一个是本文公开的新格式)的单个三元组。

[0236] 本领域的专业技术人员可以理解,可以使用许多不同的工艺和技术中的任意一种来表示信息和信号。例如,上述说明中提到过的数据、指令、命令、信息、信号、比特、符号、及码片都可以表示为电压、电流、电磁波、磁场或磁性粒子、光场或光粒子、或以上的结合。

[0237] 本领域的专业技术人员还可以进一步意识到,结合本文中所公开的实例描述的各种示例的逻辑块、模块、电路、方法及算法,能够以电子硬件、计算机软件、或二者的结合被实现。为了清楚地说明硬件和软件的可互换性,在上述说明中已经按照功能一般性地描述了各种示例的组件、程序块、模块、电路、方法及算法。这种功能究竟以硬件还是软件方式来实现,取决于整个系统的特定的应用和设计约束条件。专业技术人员可以对每个特定的应用来使用不同方法来实现所描述的功能,但是这种实现决定不应被认为超出了本发明的范围。

[0238] 结合本文中所公开的实例描述的多种示例的逻辑块、模块和电路可以用通用处理器、数字信号处理器(DSP)、专用集成电路(ASIC)、现场可编程门阵列(FPGA)或其它可编程逻辑器件、分立门或晶体管逻辑、分立硬件部件、或设计成执行本文所述功能的以上的任意组合来实现或执行。通用处理器可以是微处理器,但是可替换地,处理器也可以是任何常规的处理器、控制器、微控制器、或状态机。处理器也可以被实现为计算器件的组合,例如,DSP和微处理器的组合、多个微处理器的组合、一个或多个微处理器与一个DSP核心的组合、或任意其它此类配置。

[0239] 结合本文中所公开的实例描述的方法或算法可以直接用硬件、处理器执行的软件模块、或二者的结合来实施。软件模块可置于RAM存储器、闪存、ROM存储器、EPROM存储器、EEPROM存储器、寄存器、硬盘、可移动硬盘、CD-ROM、或技术领域内所公知的任意其它形式的存储介质中。可将存储介质连接到处理器,以便处理器可从存储介质读取信息并向存储介质写入信息。可替换地,存储介质可以被集成在处理器中。处理器和存储介质可以置于ASIC中。

[0240] 对所公开的实例的上述说明,是为了使本领域的任何专业技术人员能够实现或使用本发明。对这些实例的多种修改对本领域的专业技术人员来说将是显而易见的,本文中所定义的一般原理可以在不脱离本发明的精神或范围的情况下,在其它实例中实现。因此,本发明将不会被限制于本文所示的这些实例,而是要符合与本文所公开的原理和新颖特点一致的最宽的范围。

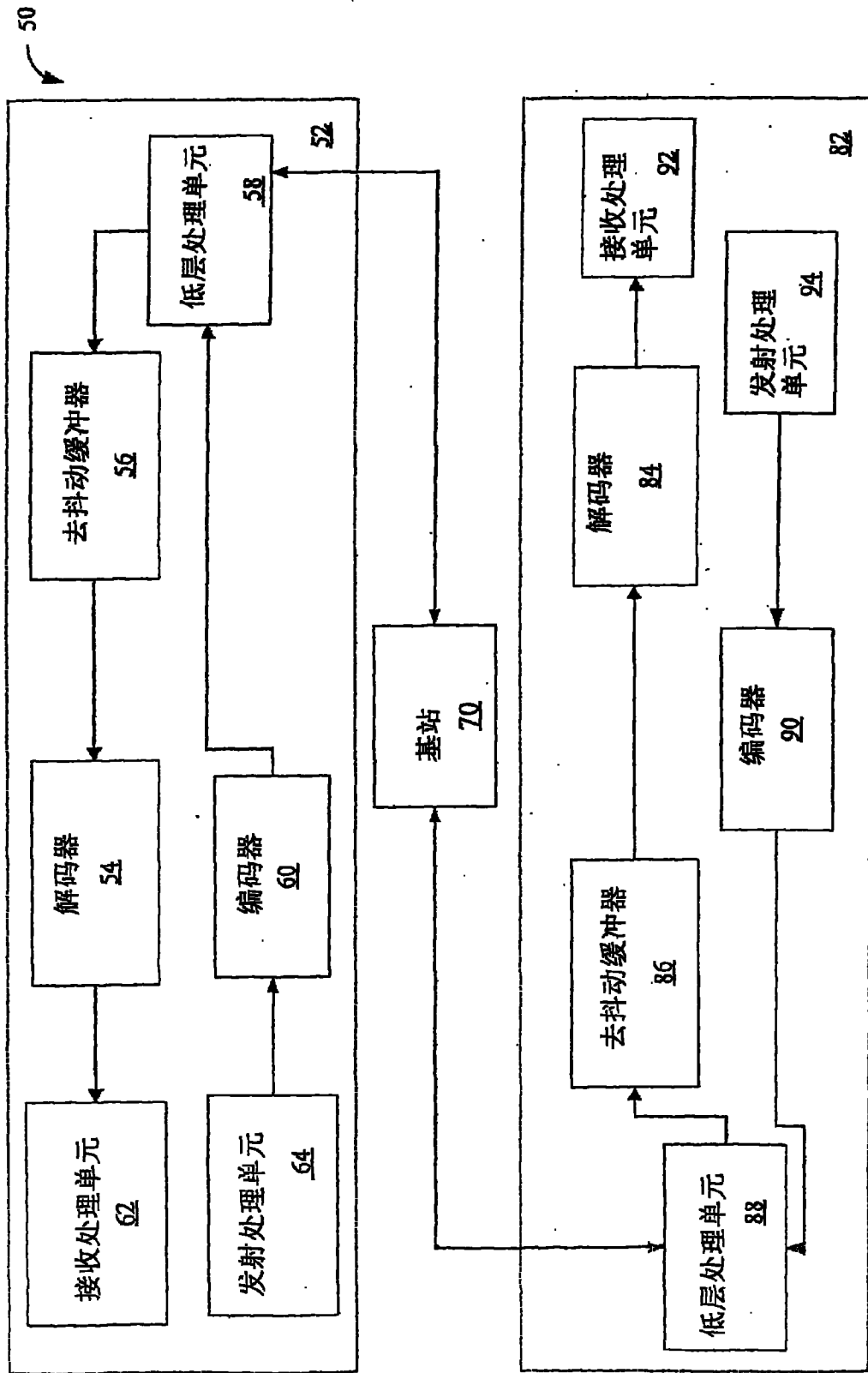


图 1

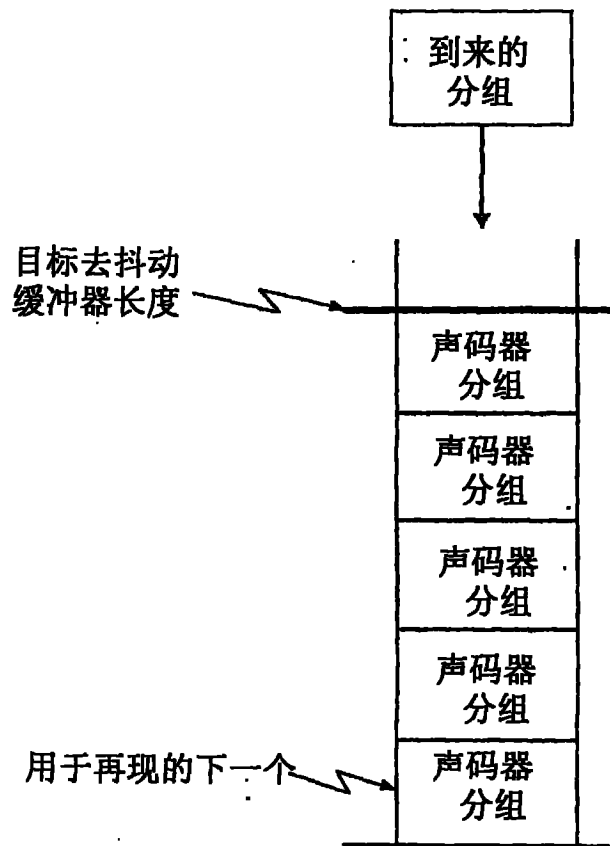


图 2

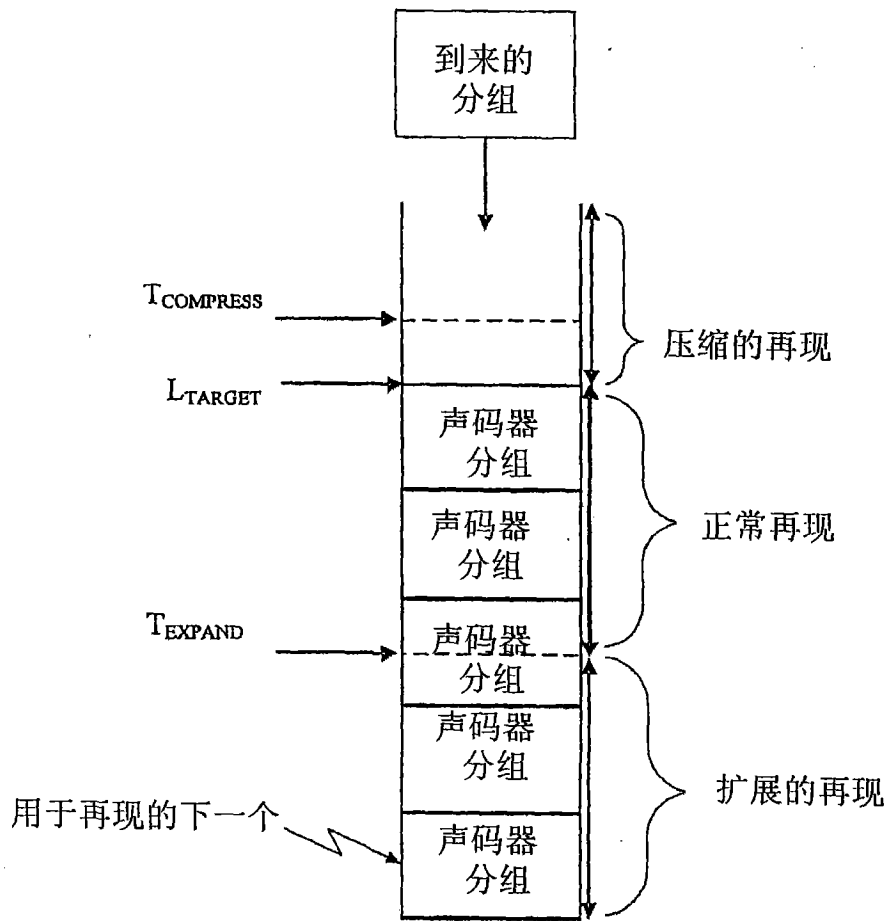


图 13A

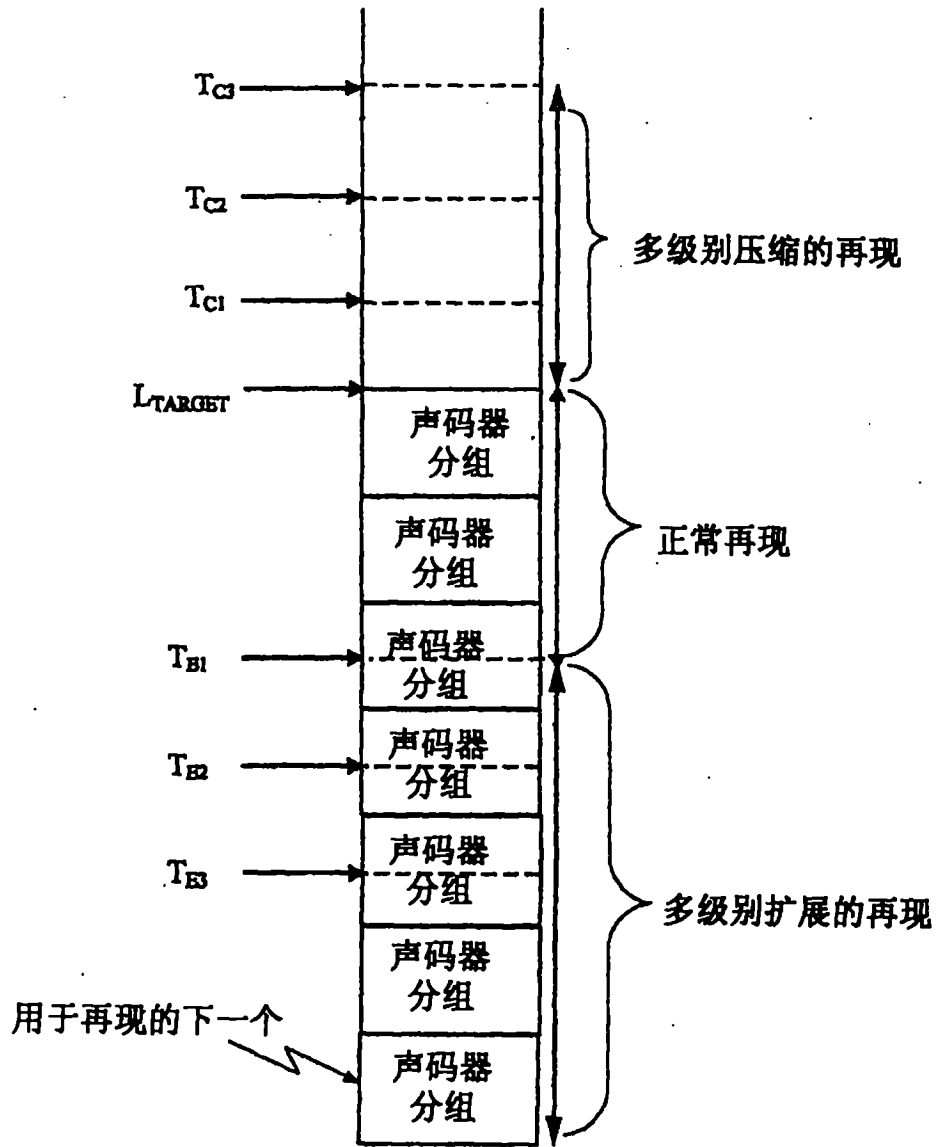


图 13B

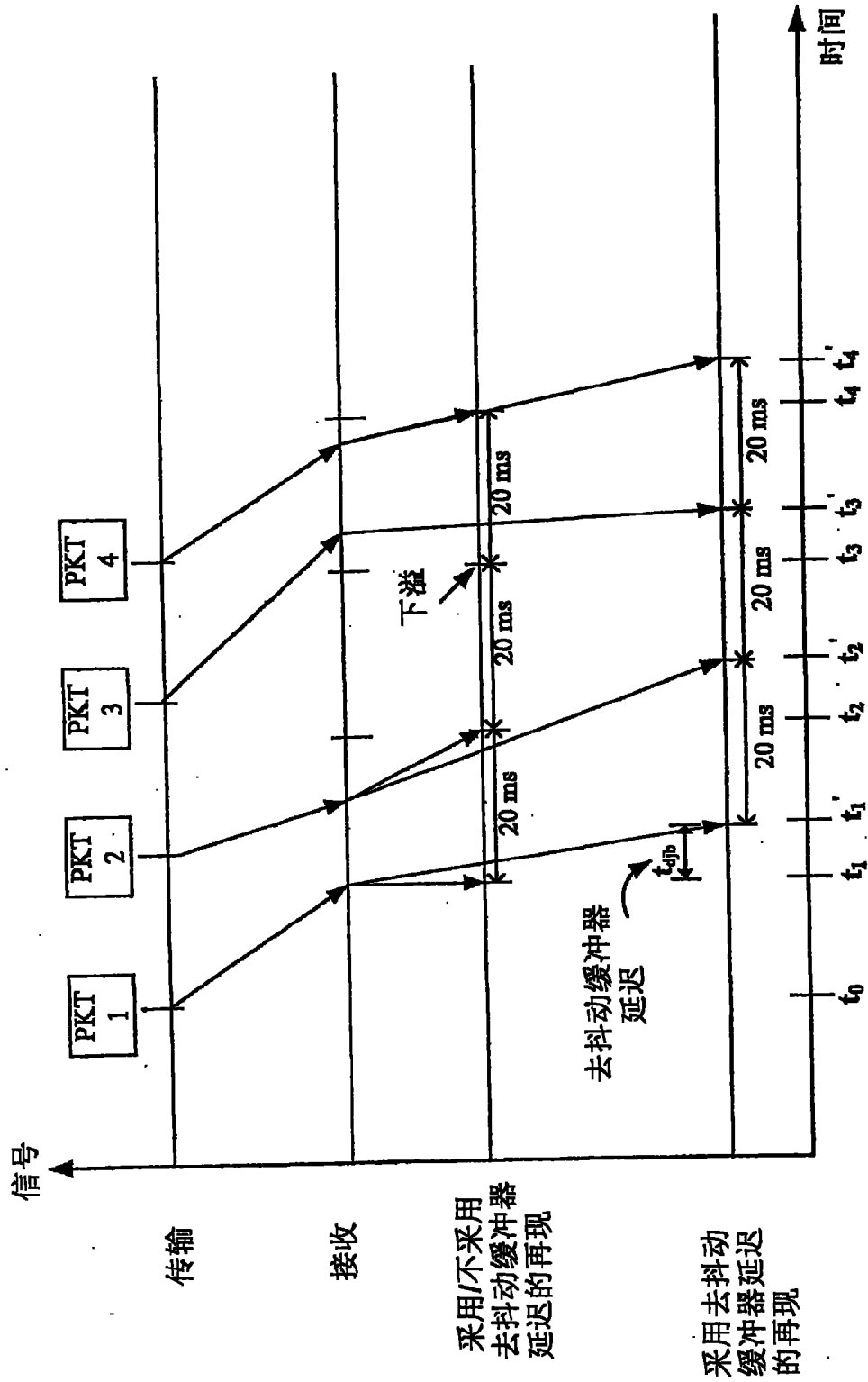


图 3

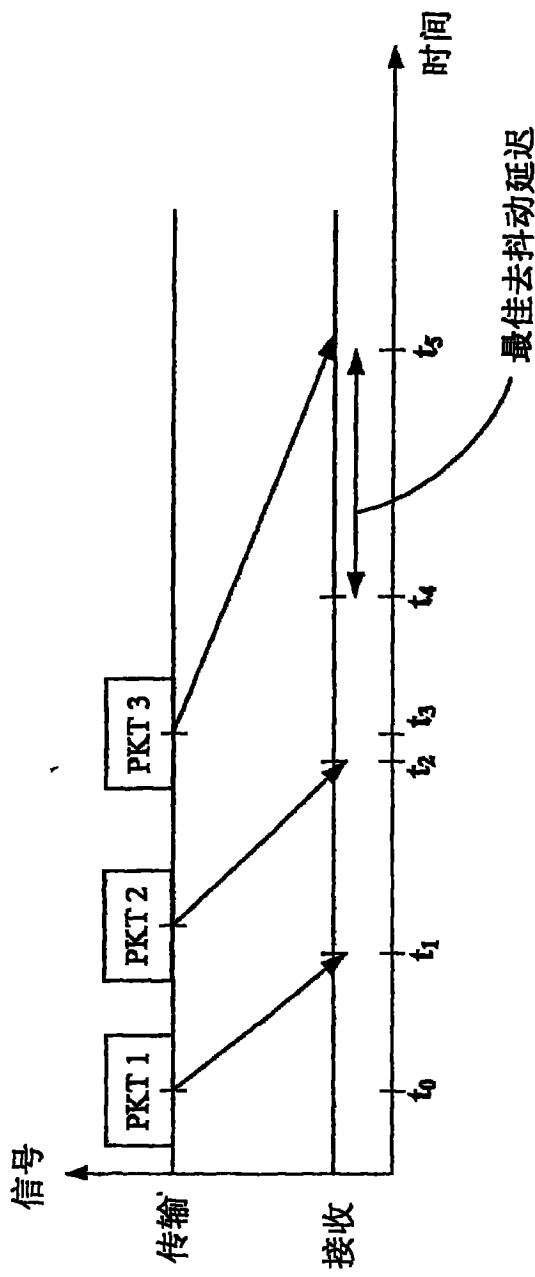


图 4A

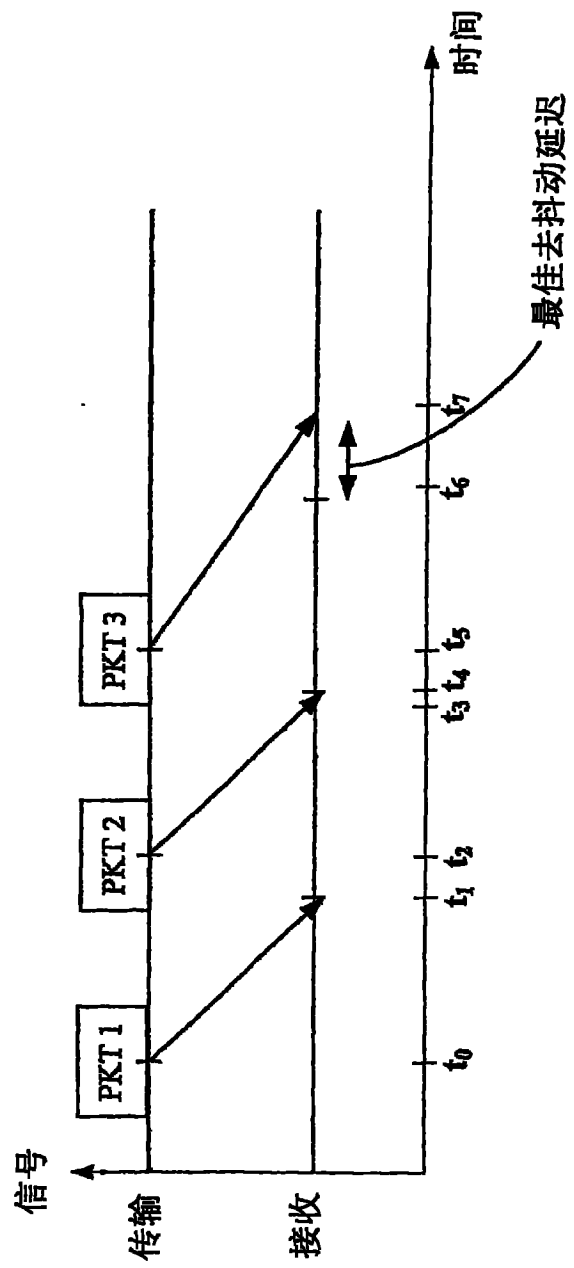


图 4B

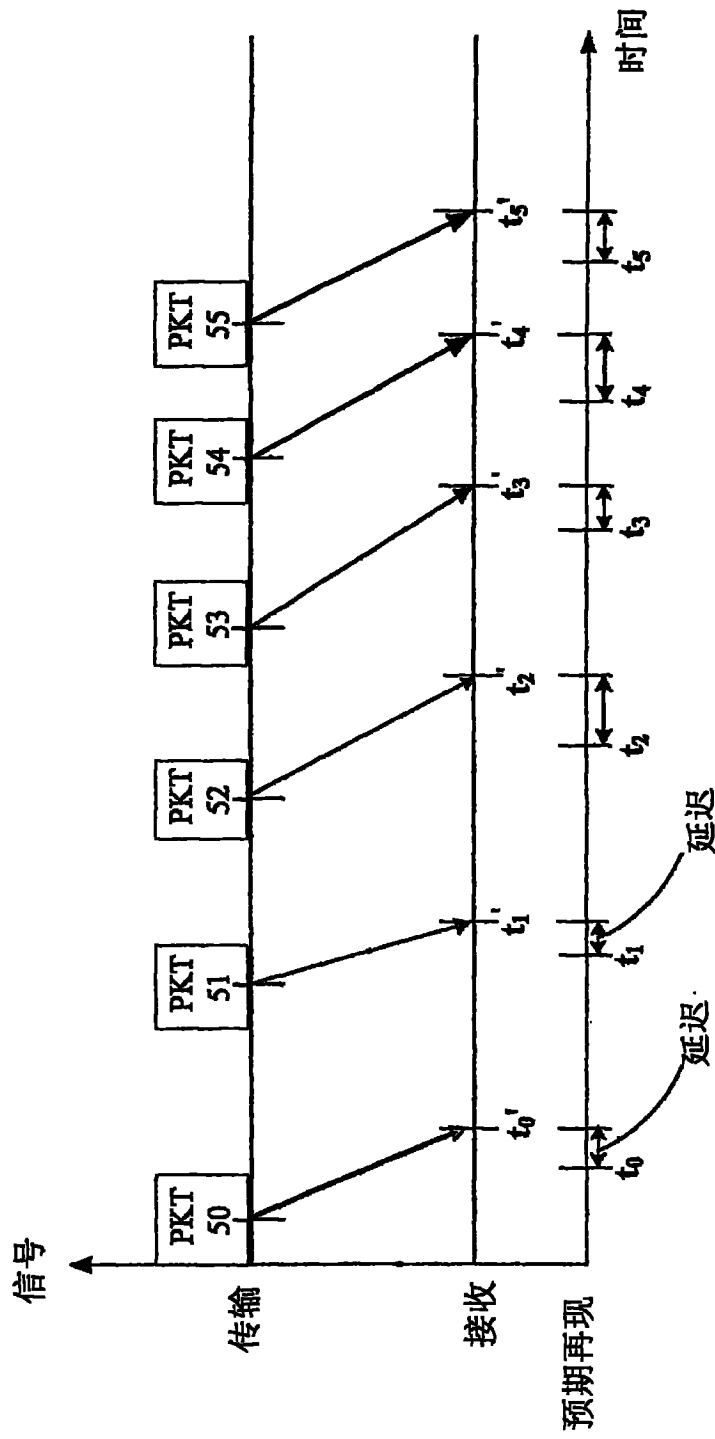


图 5

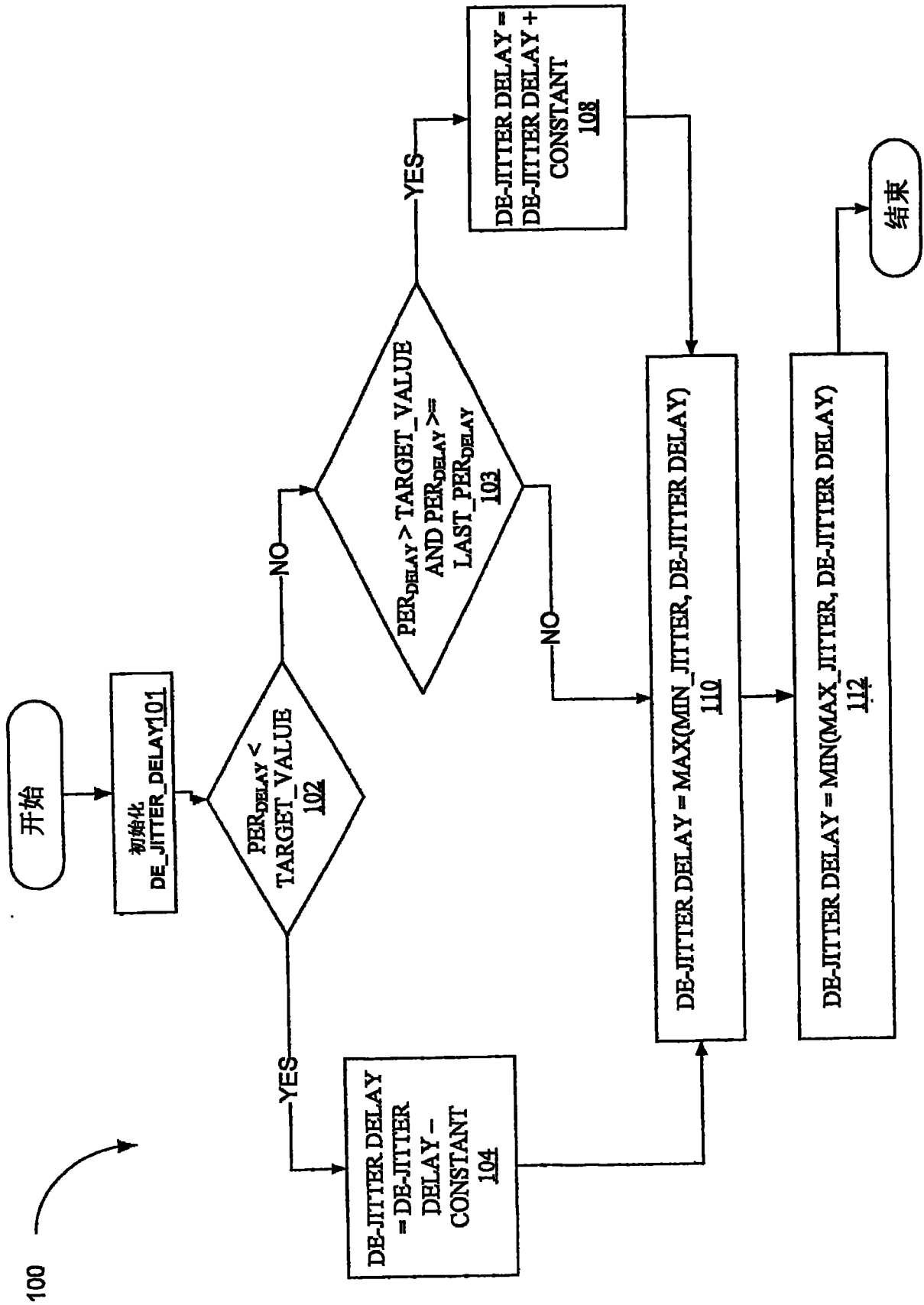


图 6

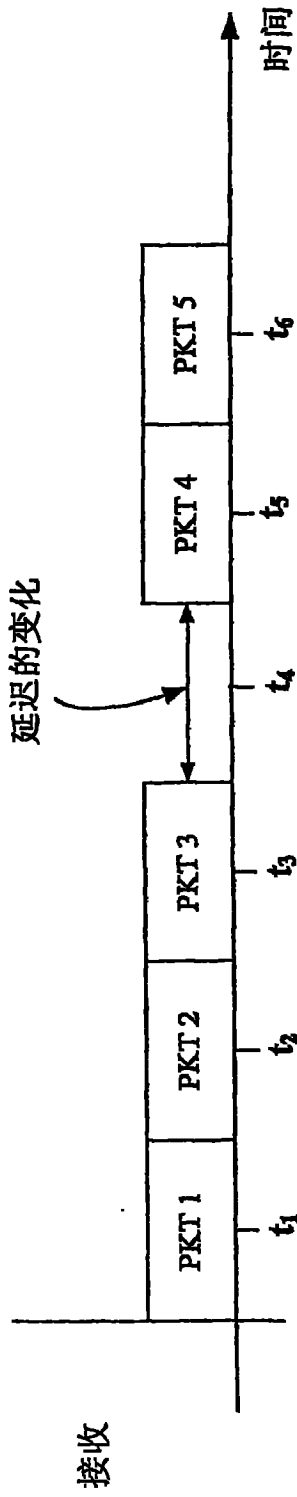


图 7A

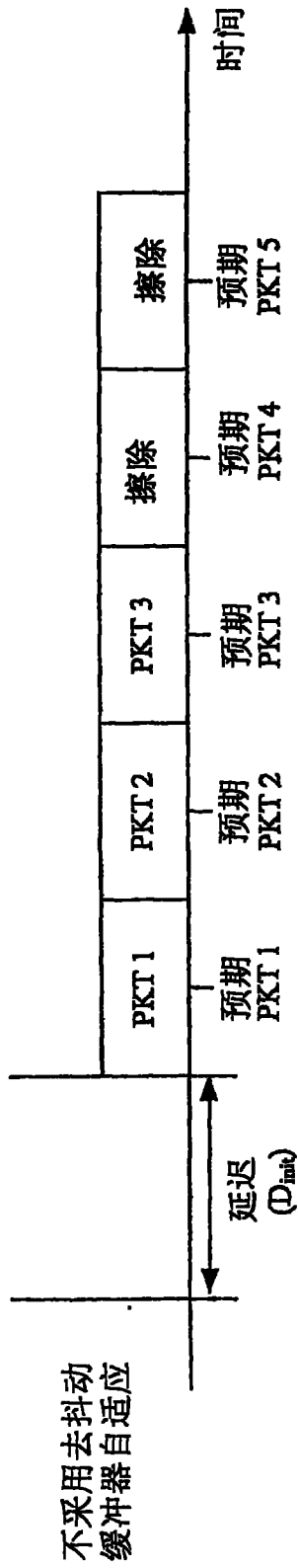


图 7B

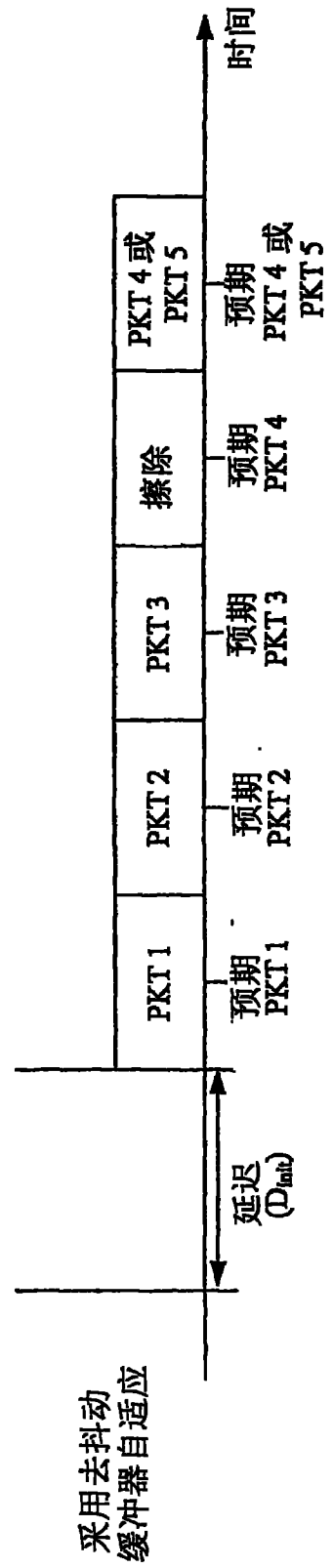


图 7C

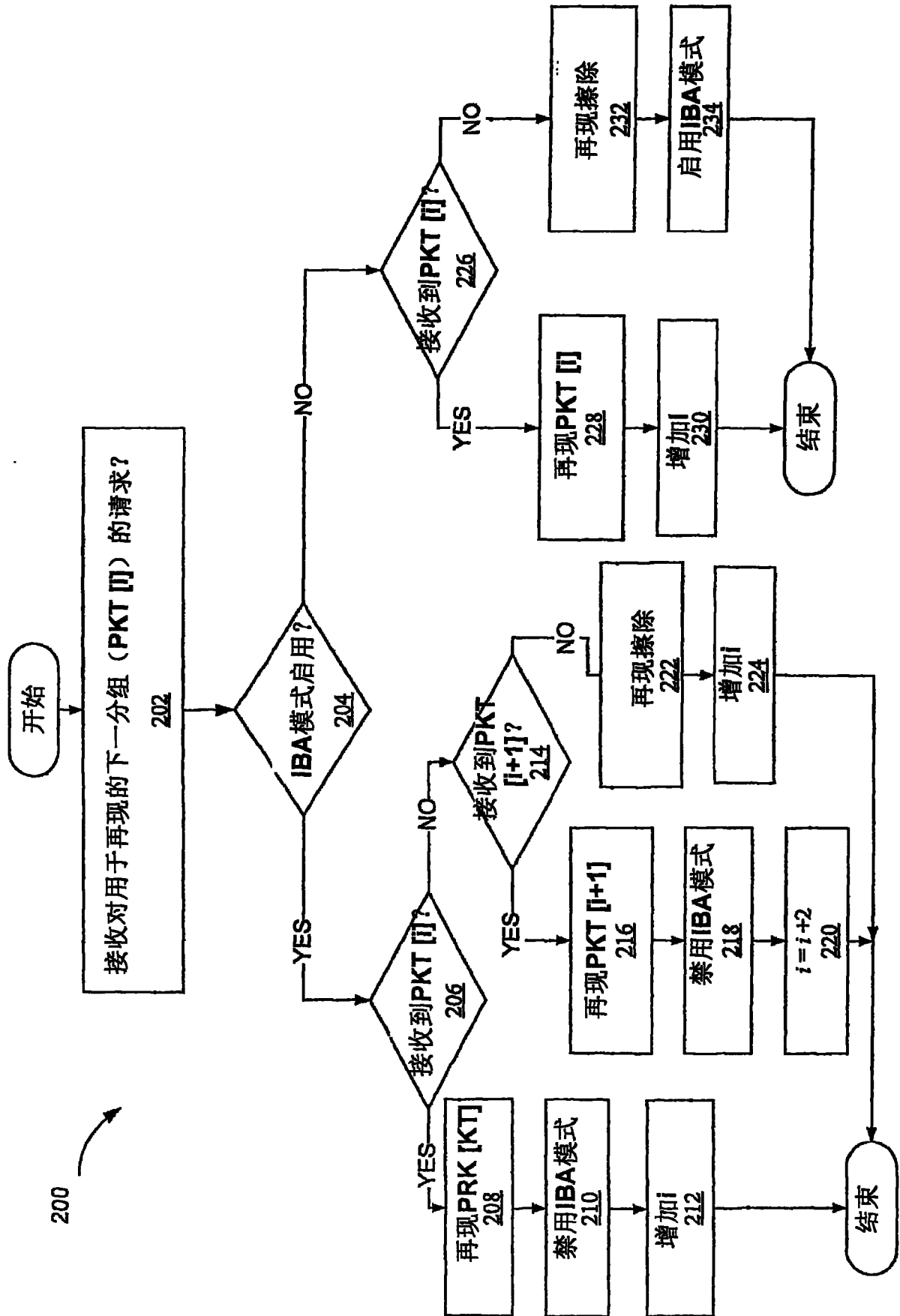


图 8A

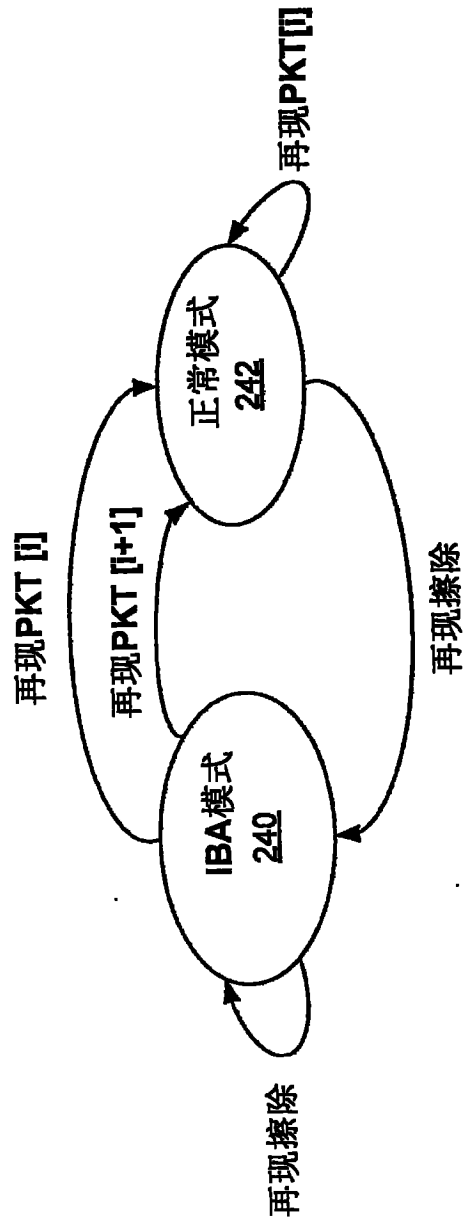


图 8B

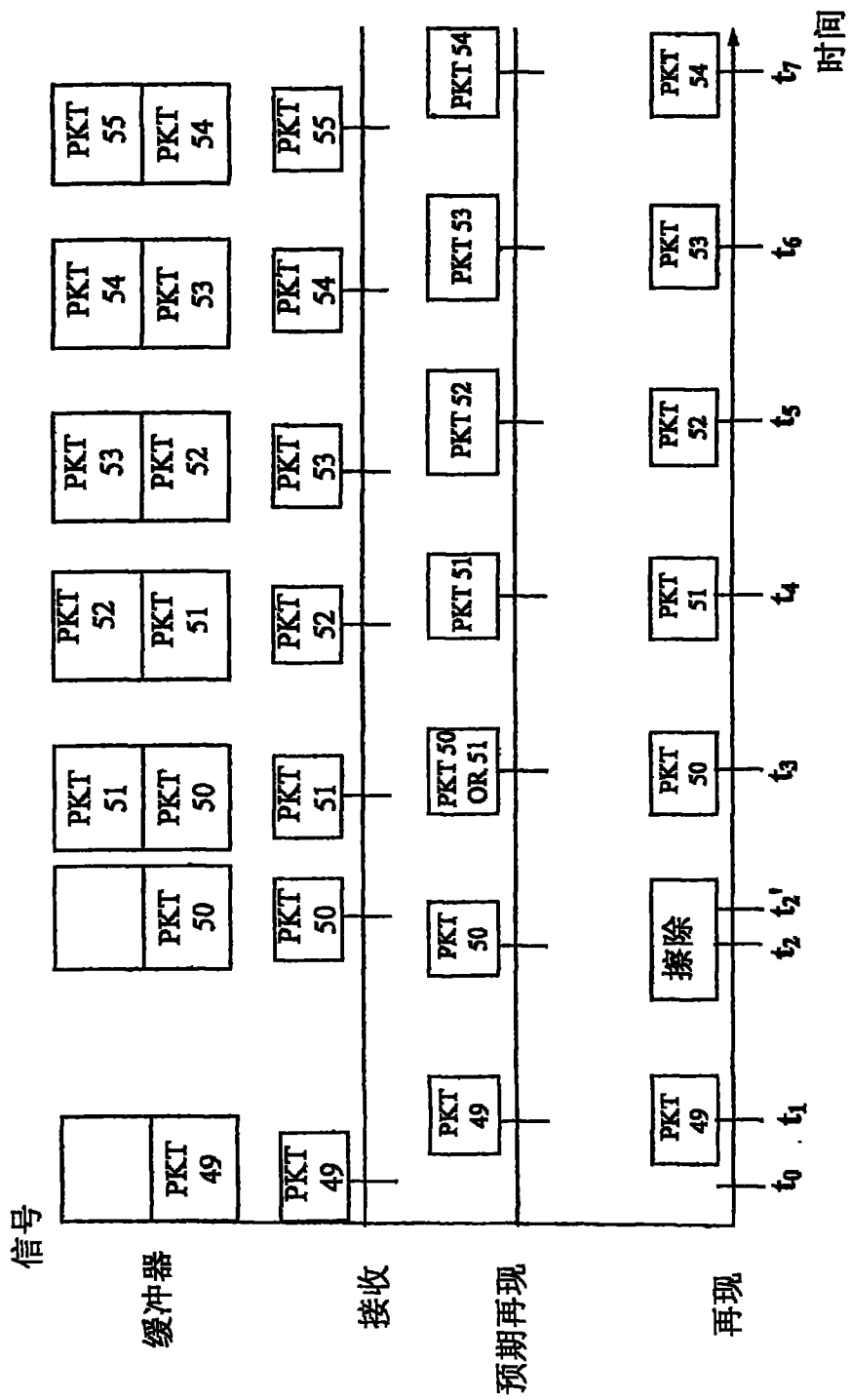


图 9

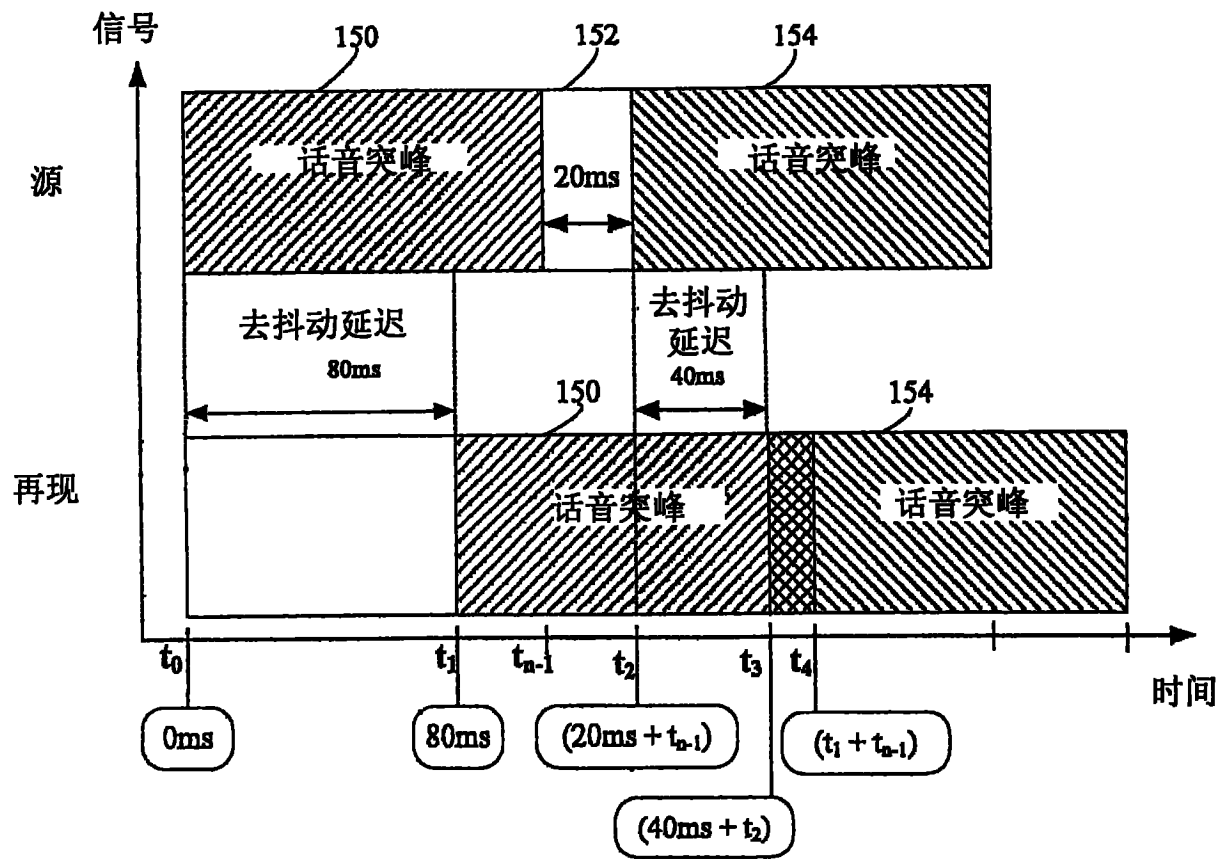


图 10

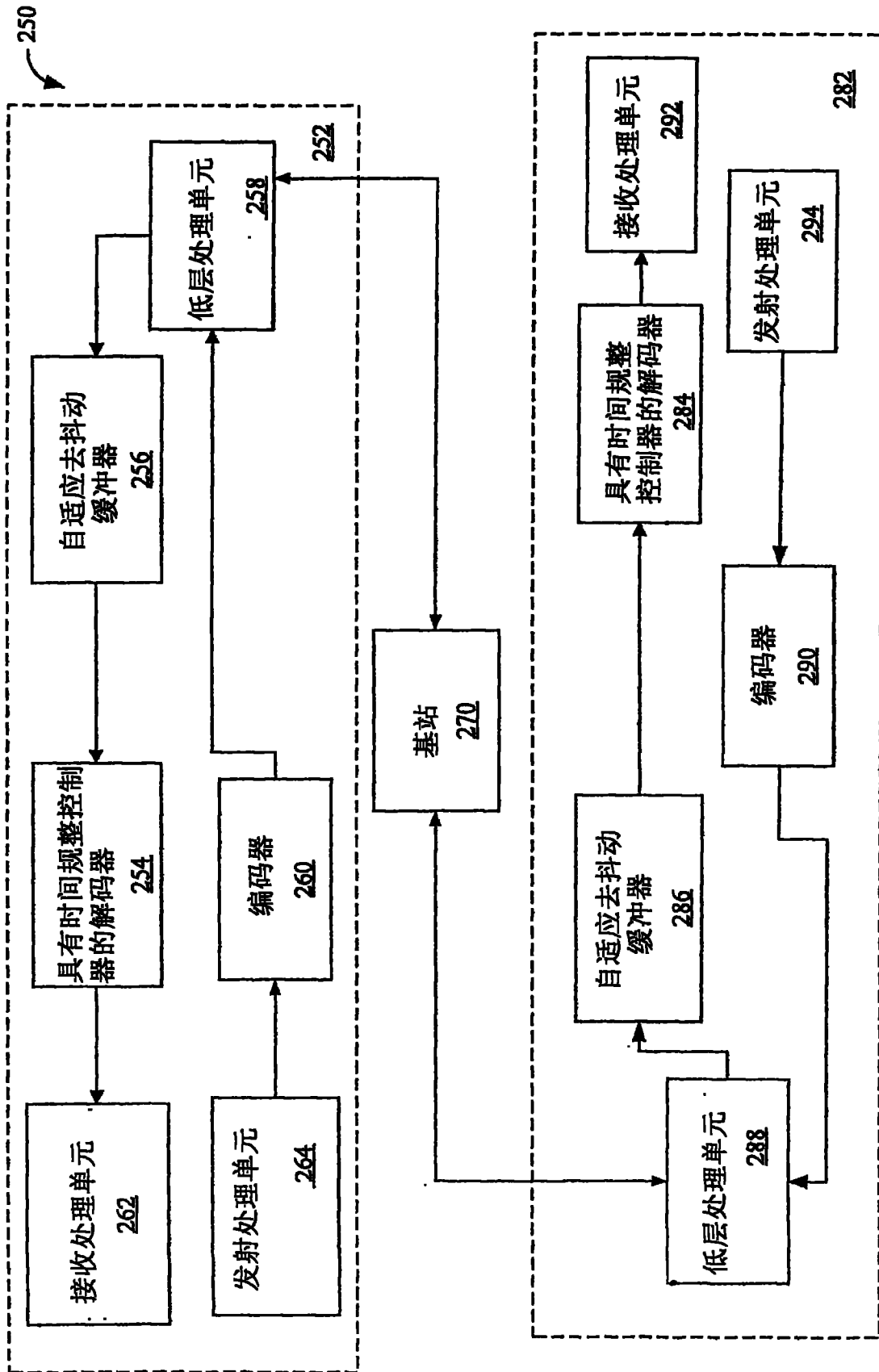


图 11

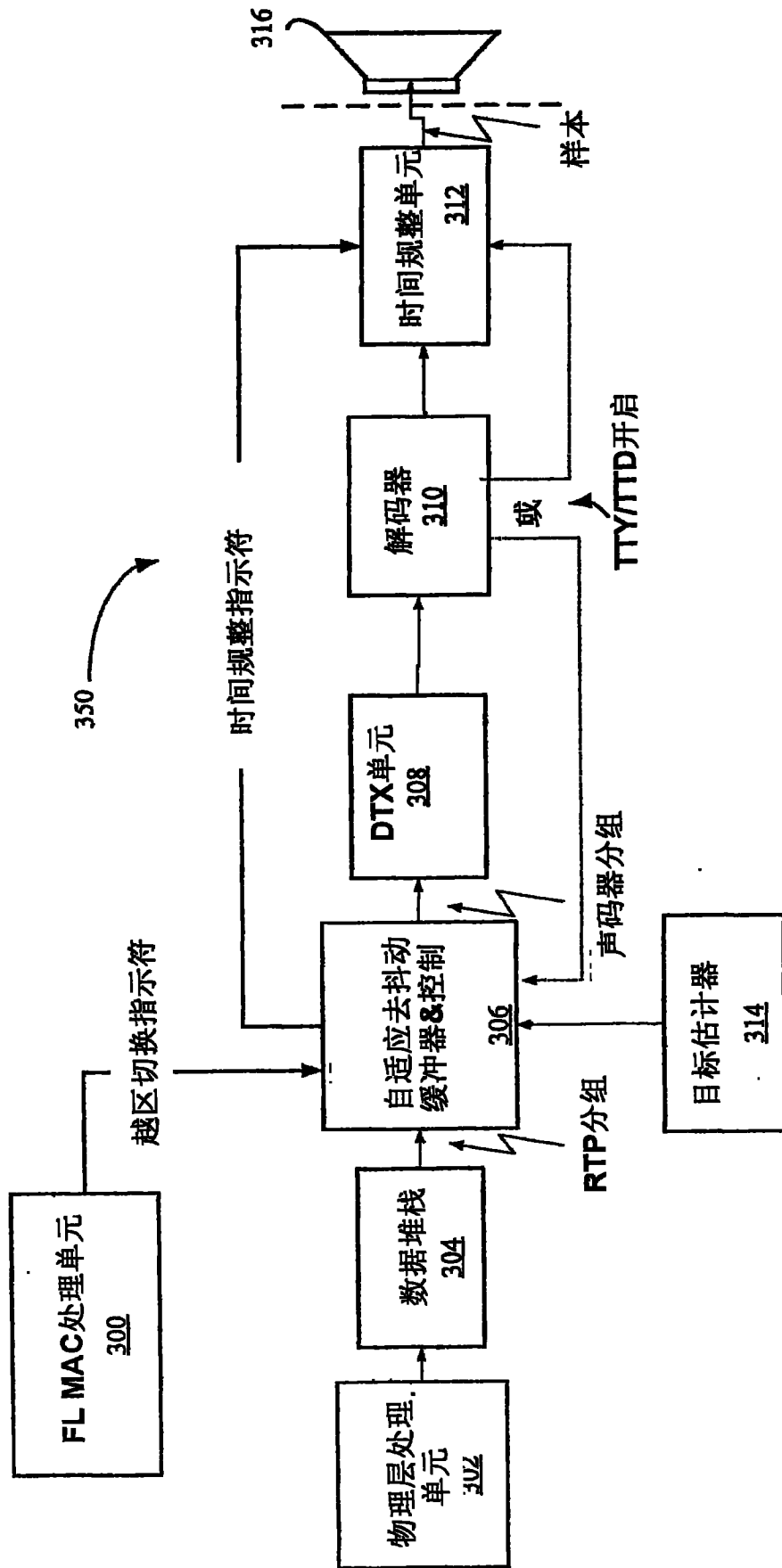


图 12

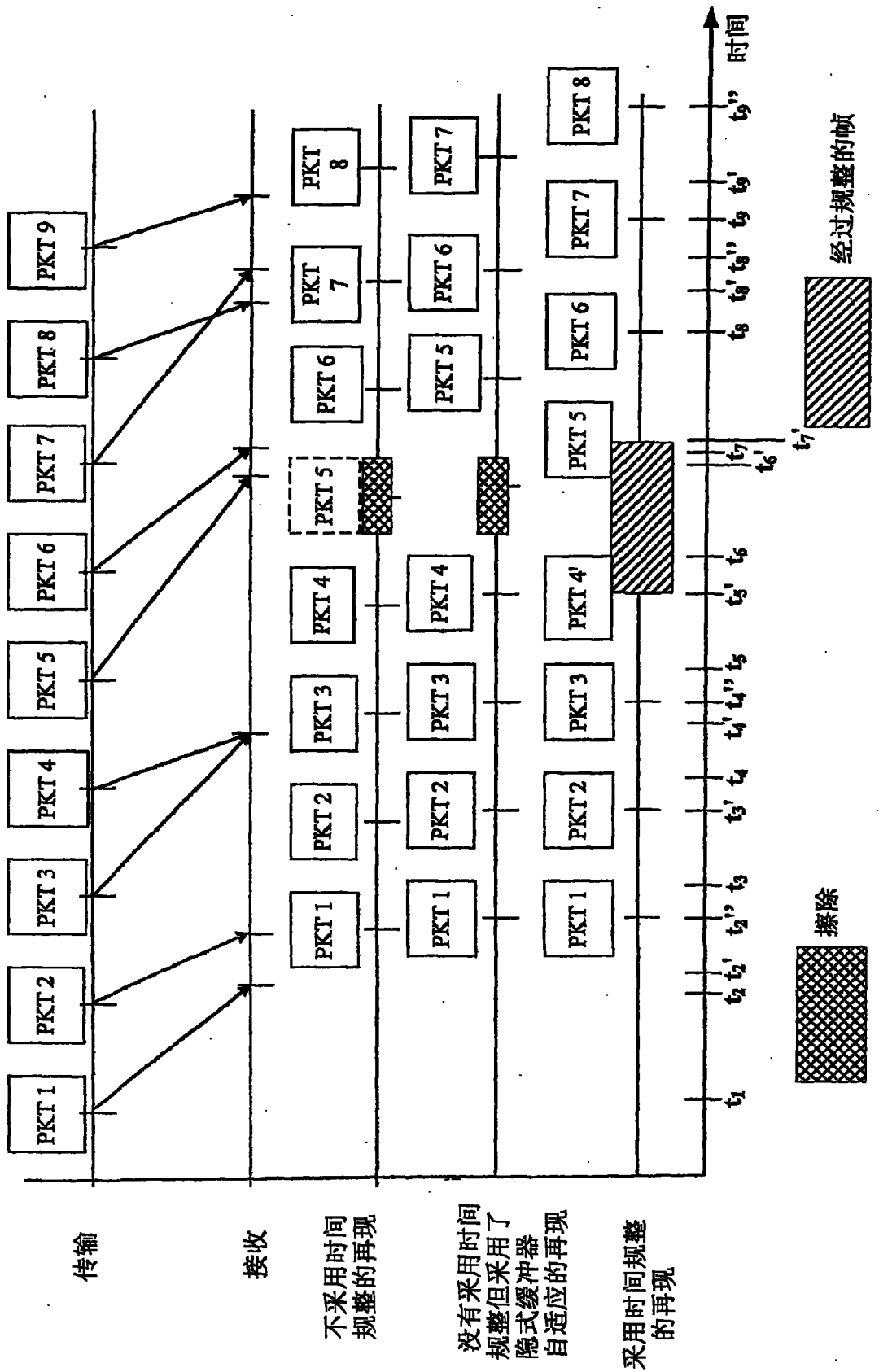


图 14

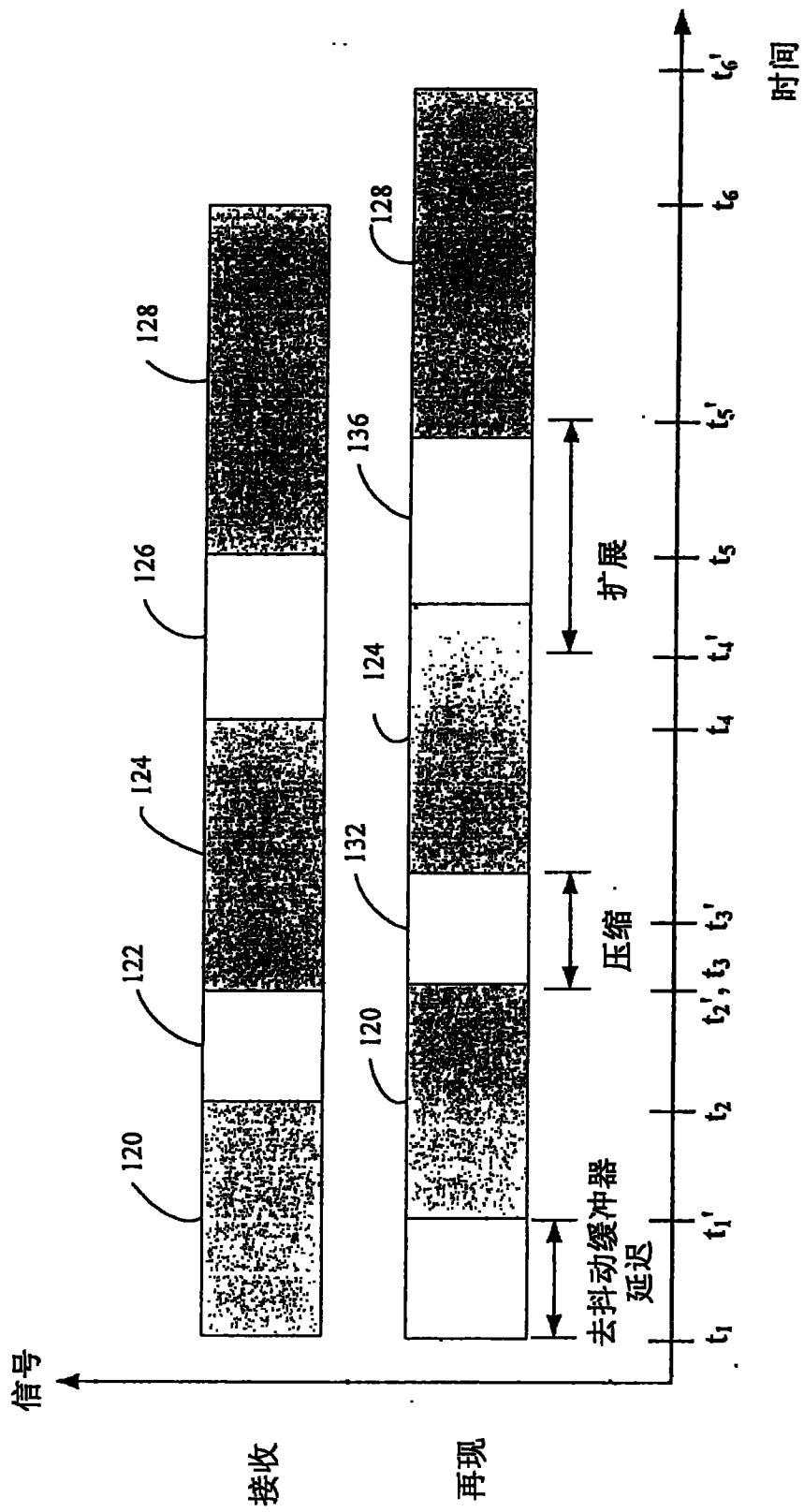


图 15

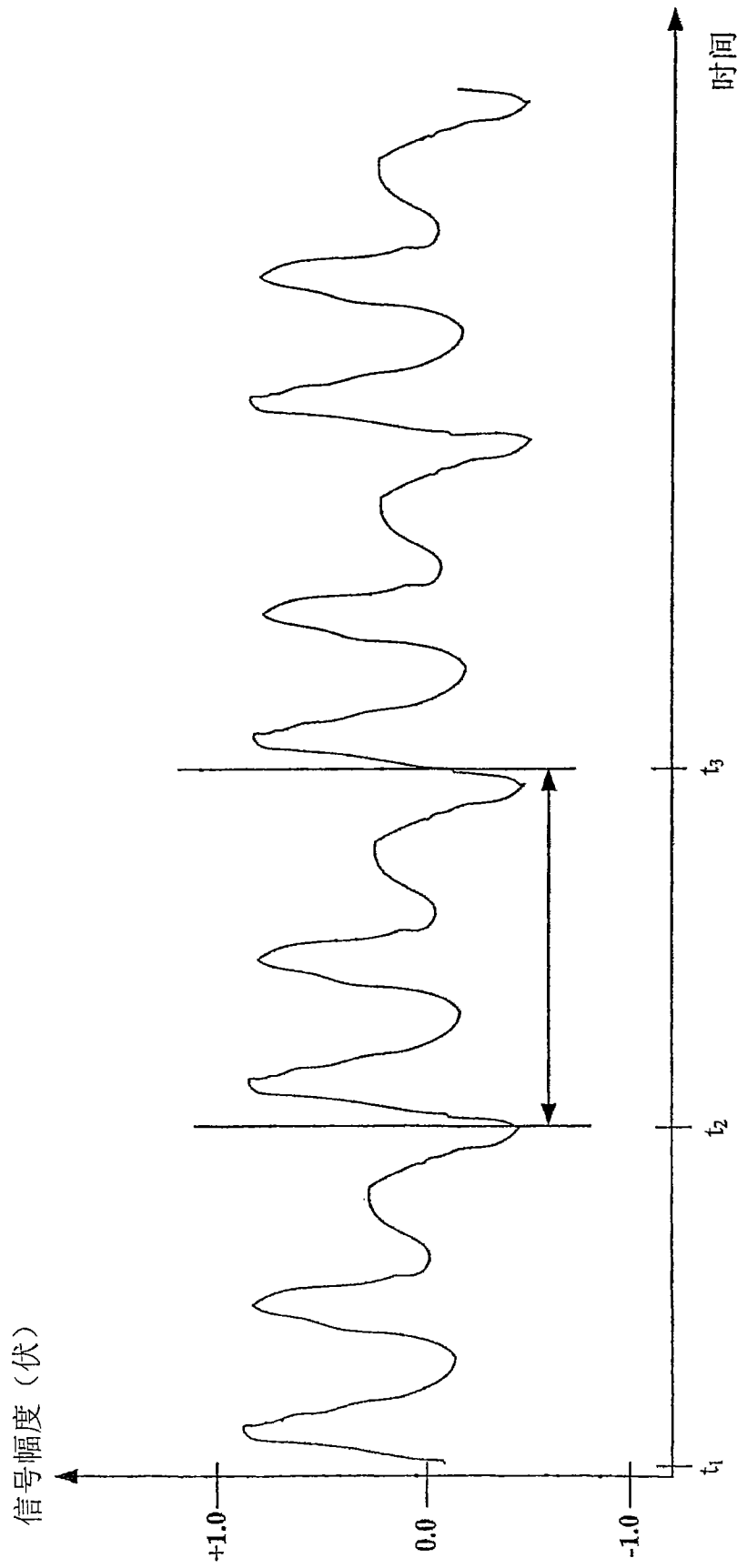


图 16

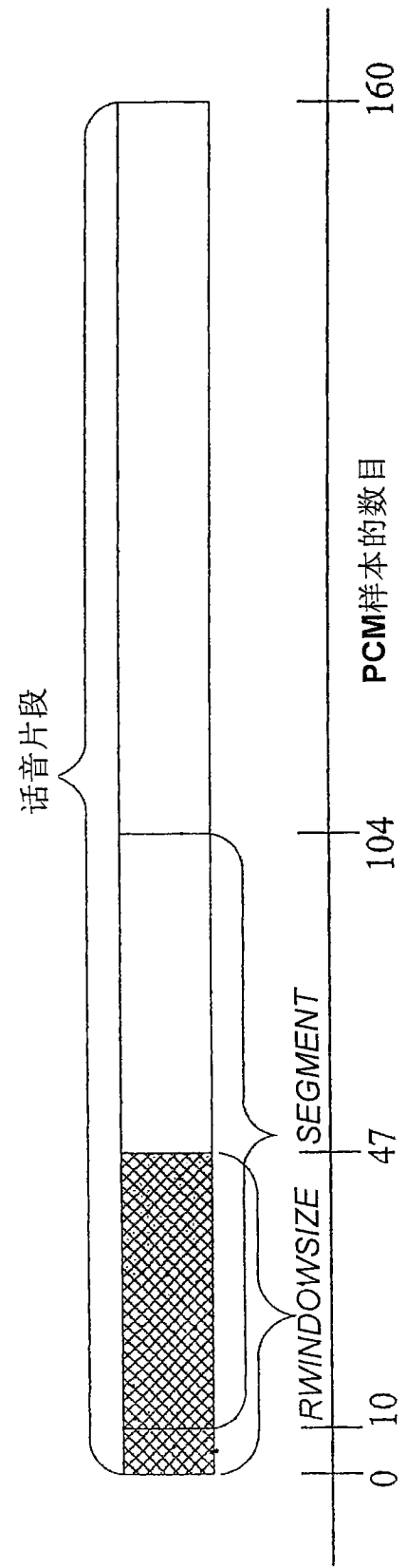


图 17A

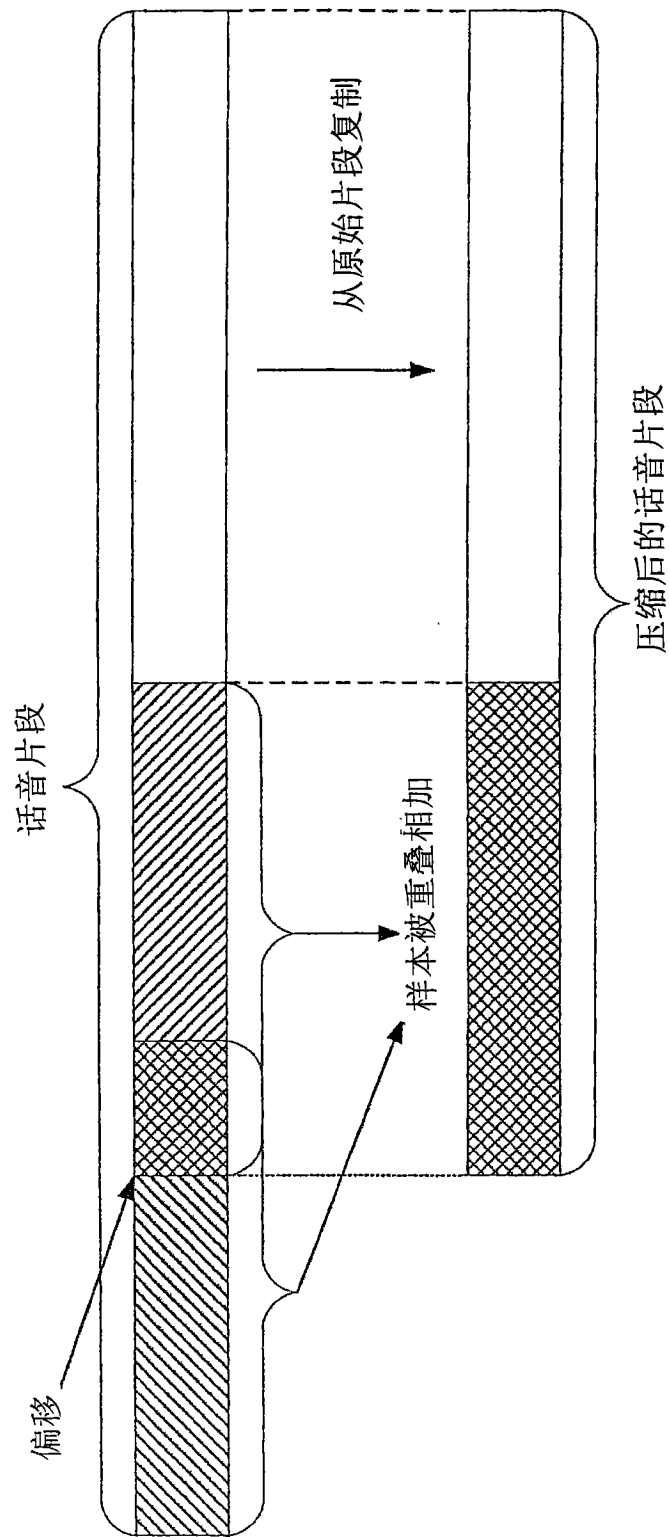


图 17B

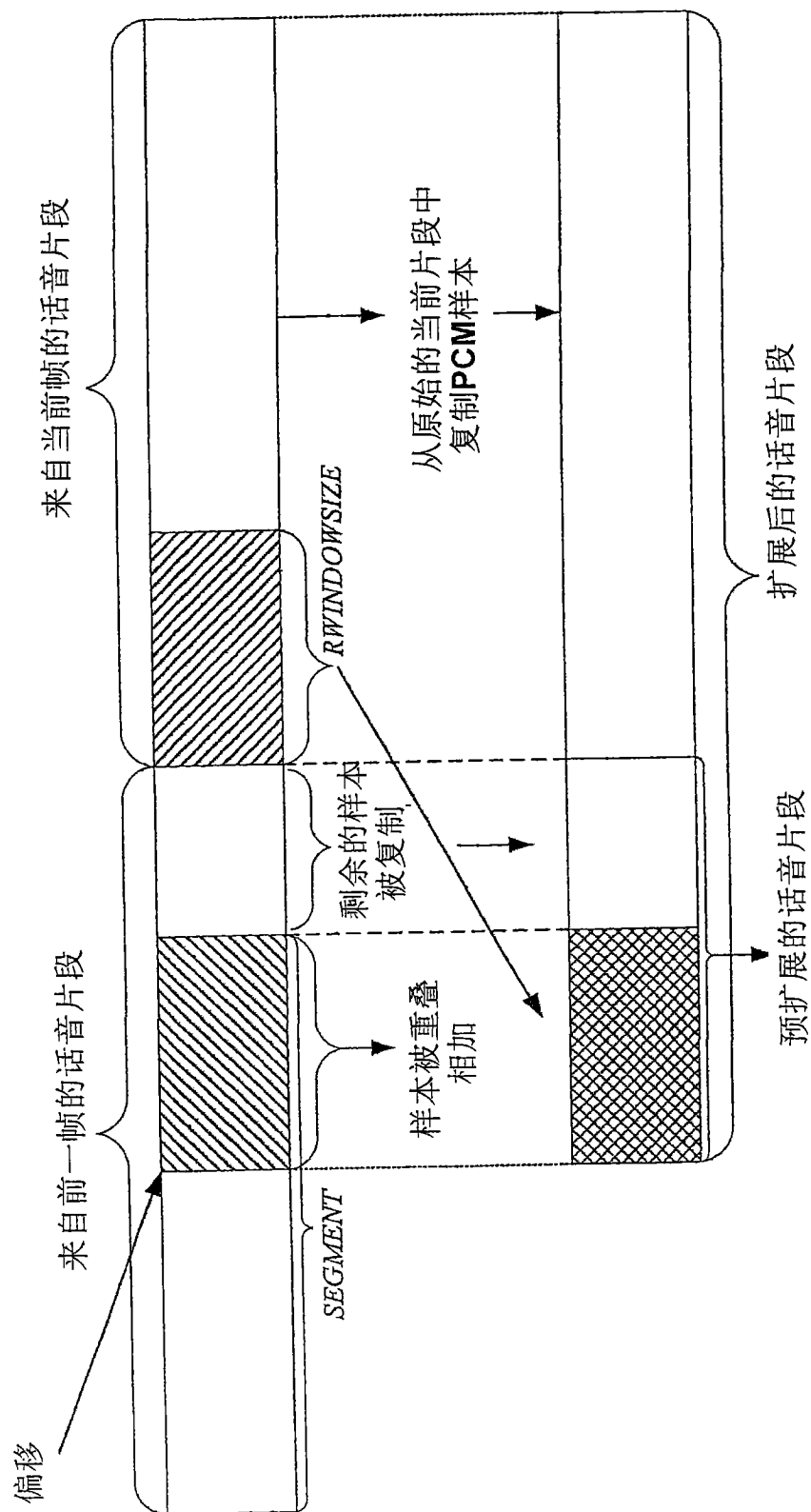


图 18A

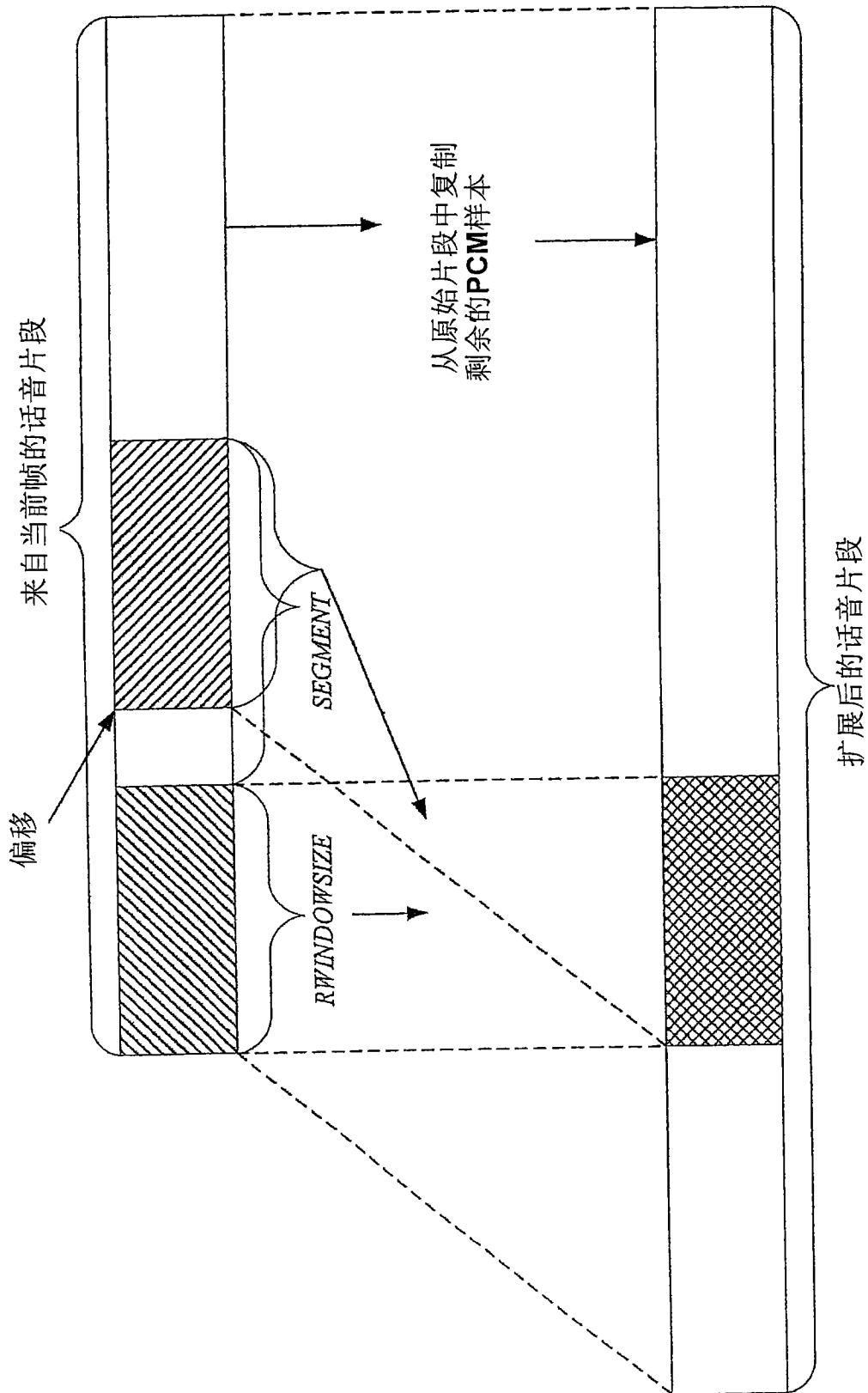


图 18B

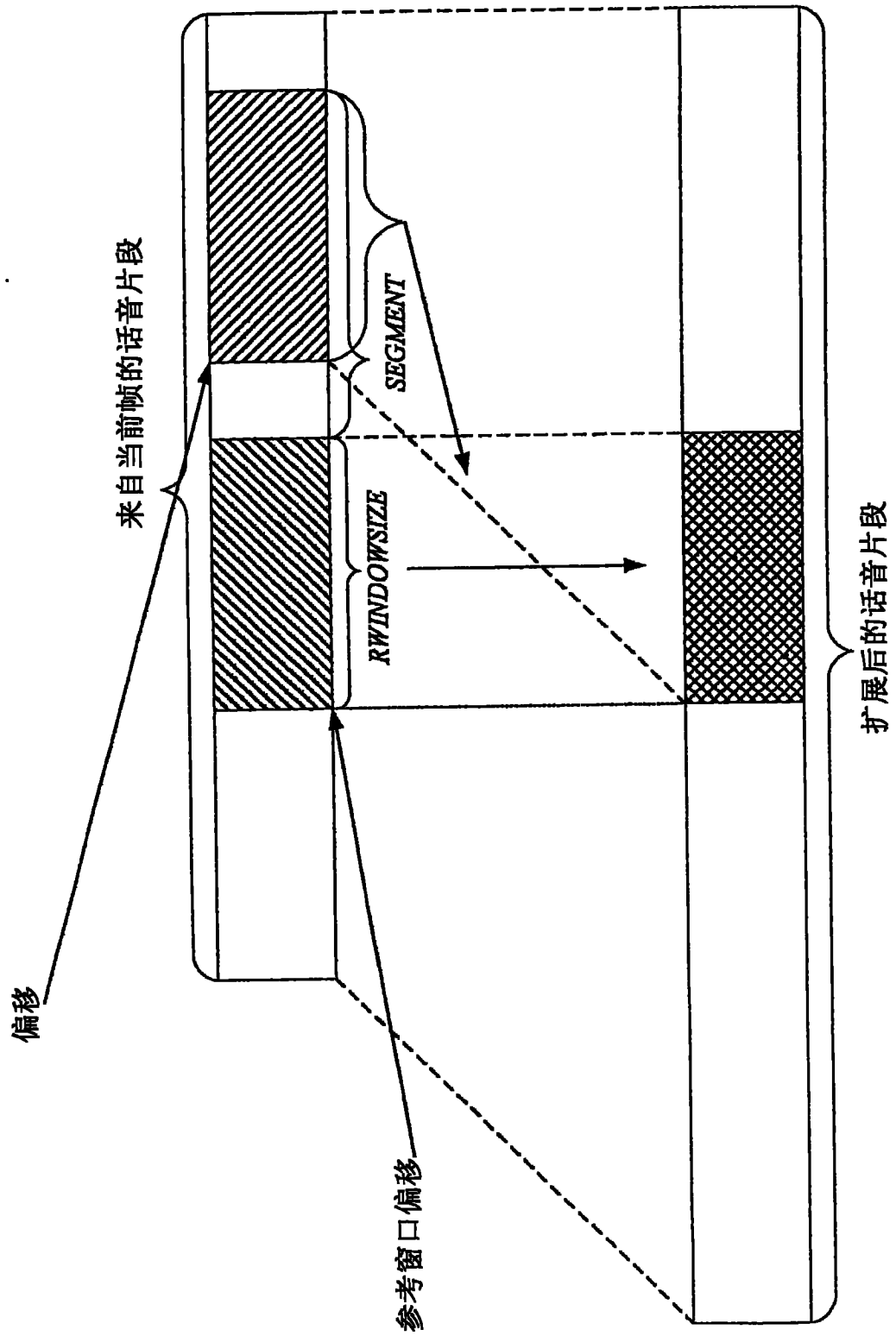


图 18C

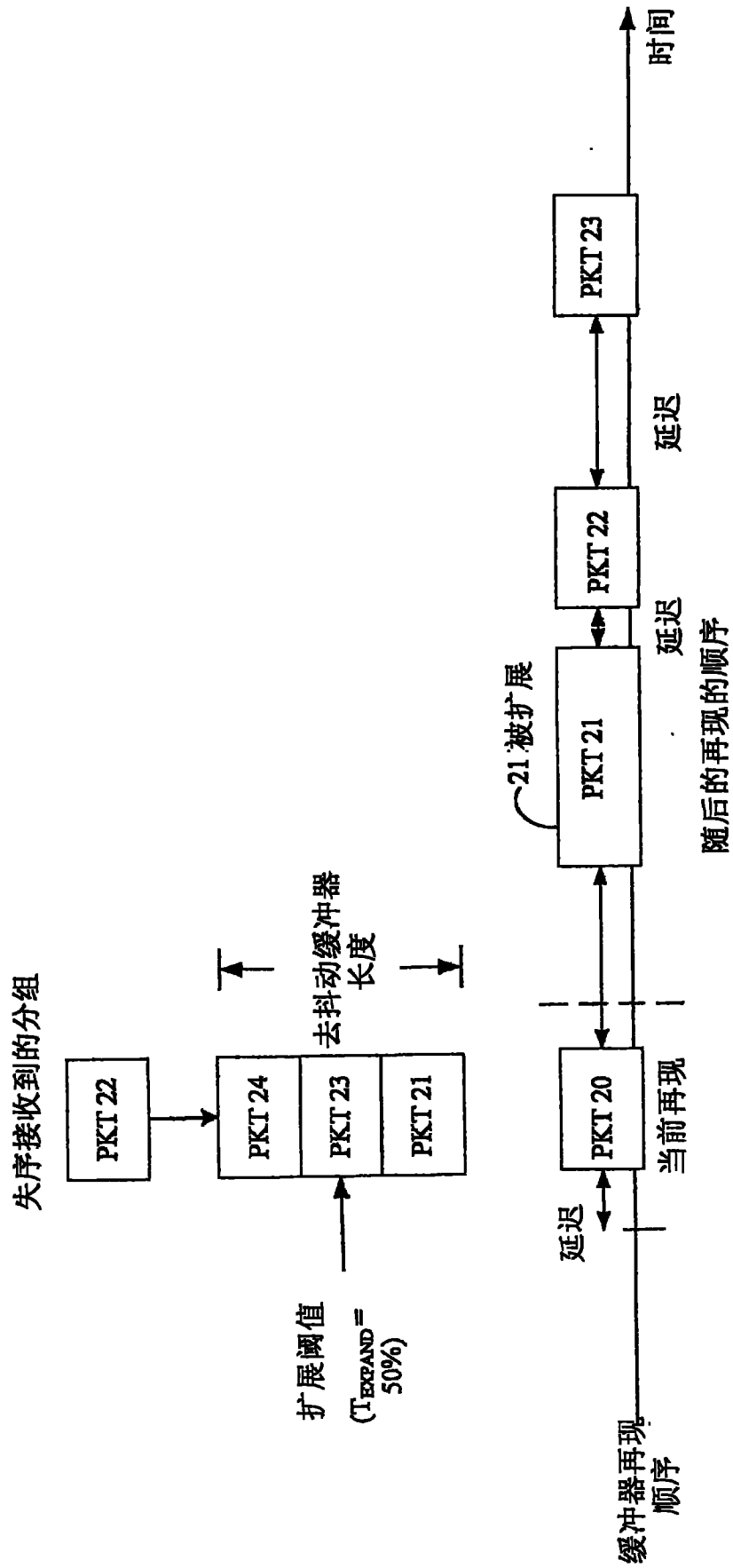


图 19

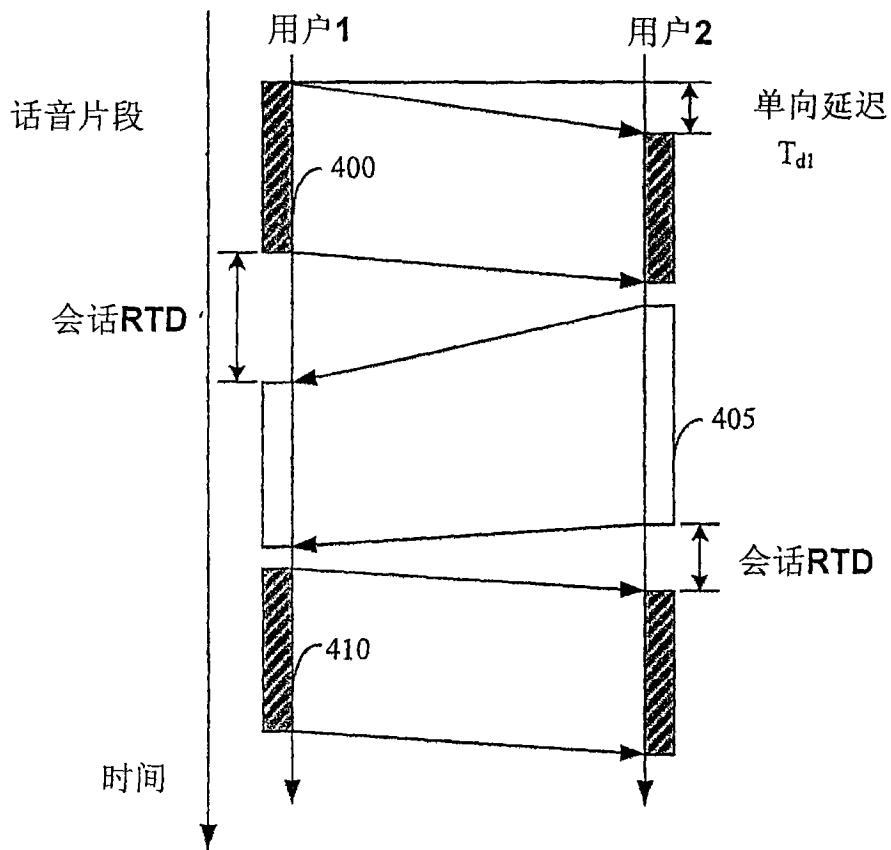


图 20

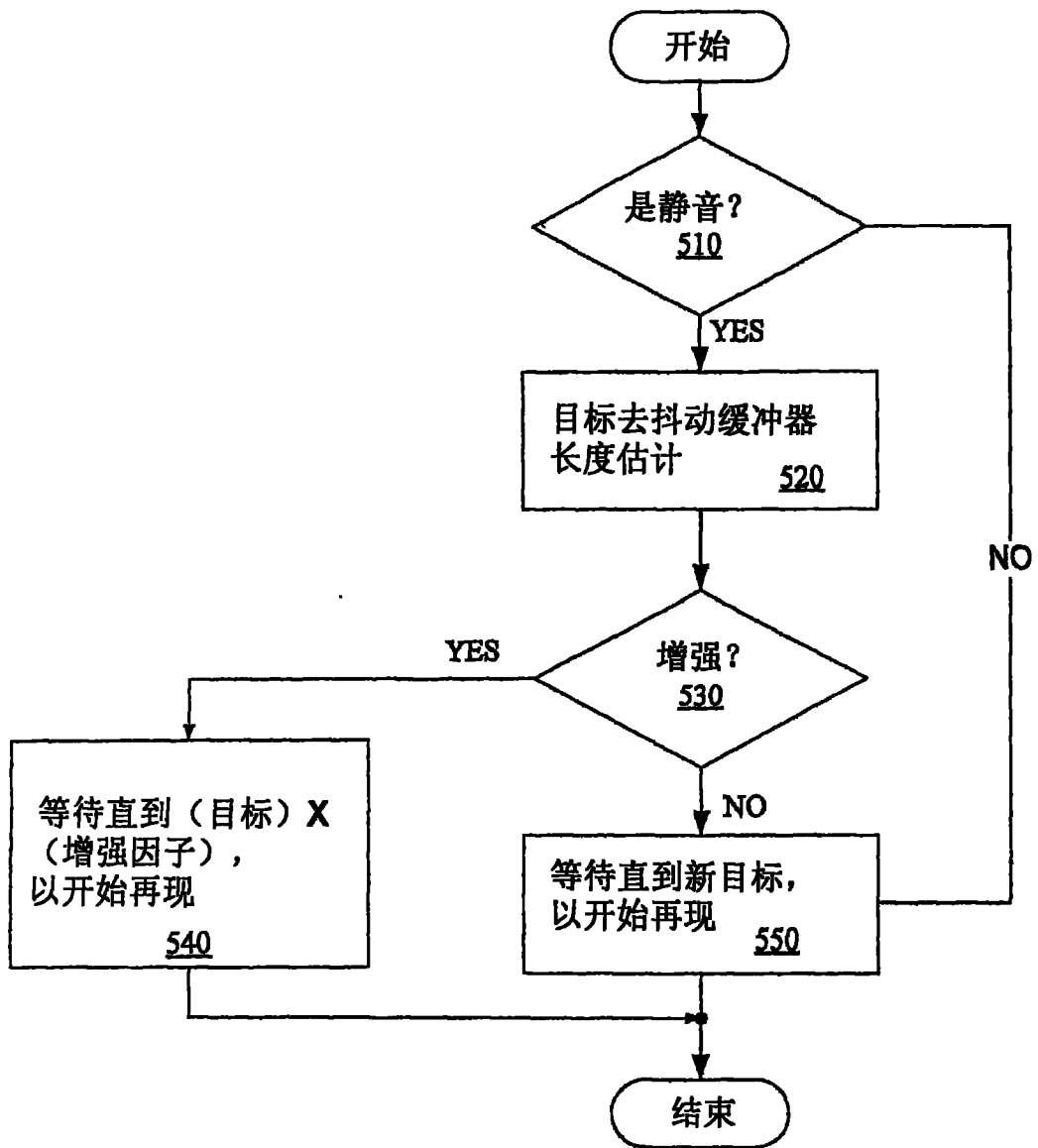


图 21

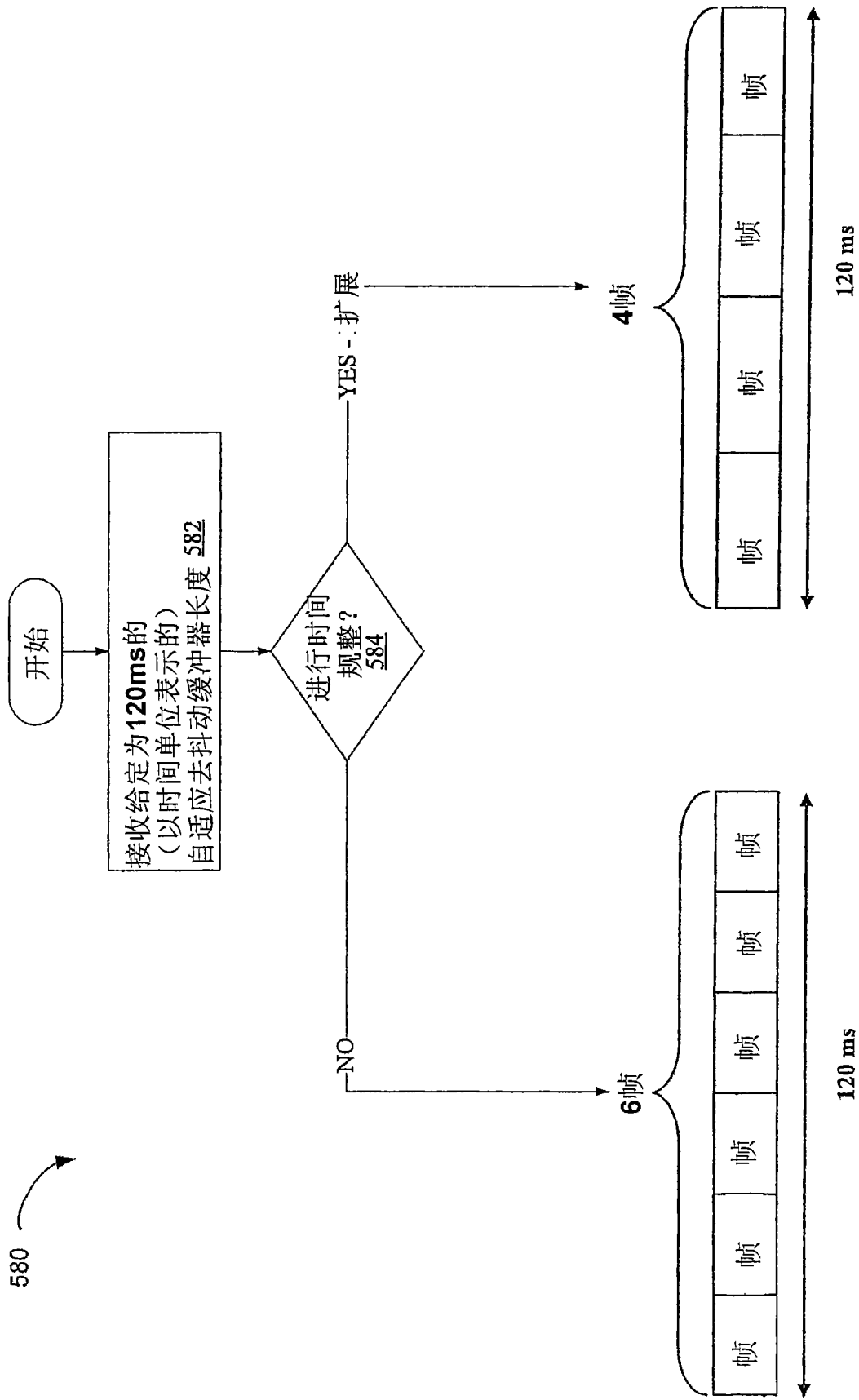


图 22

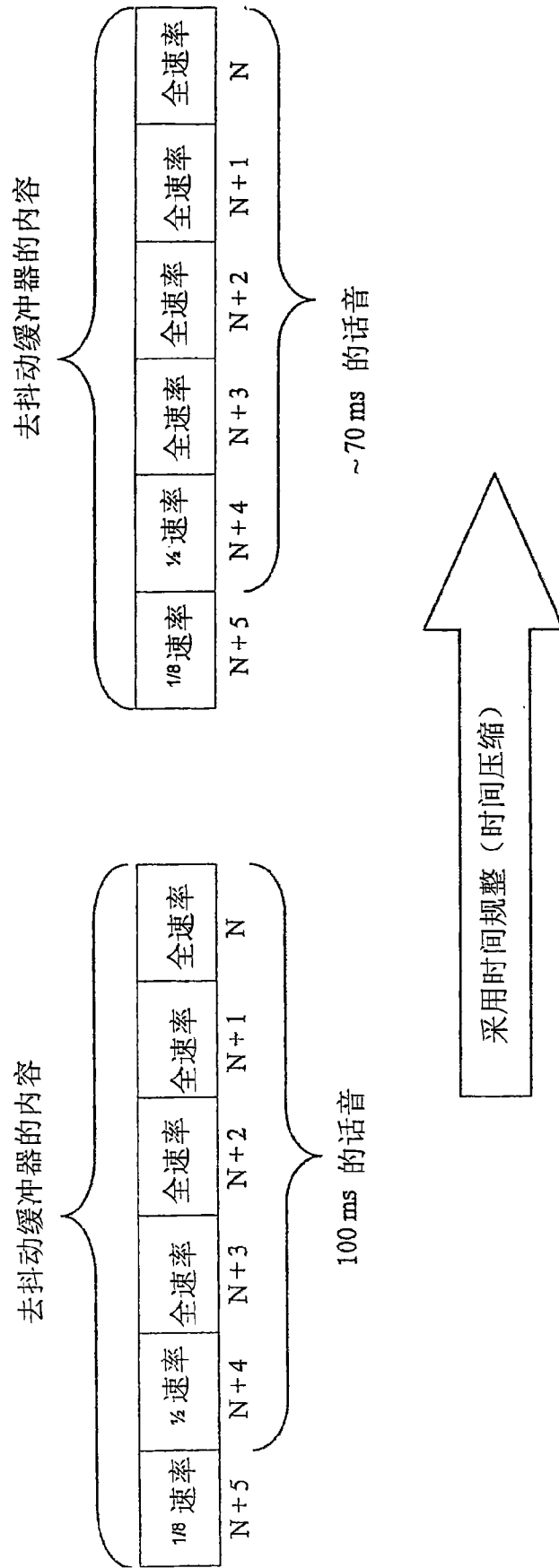


图 23

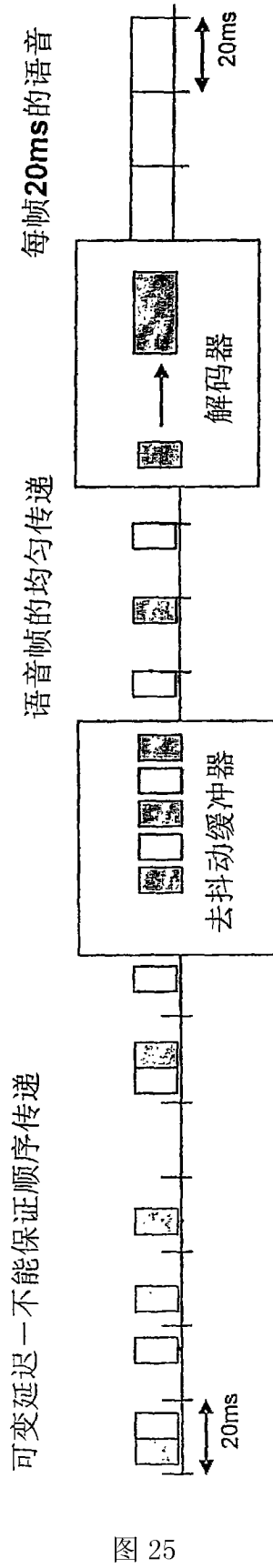


图 25

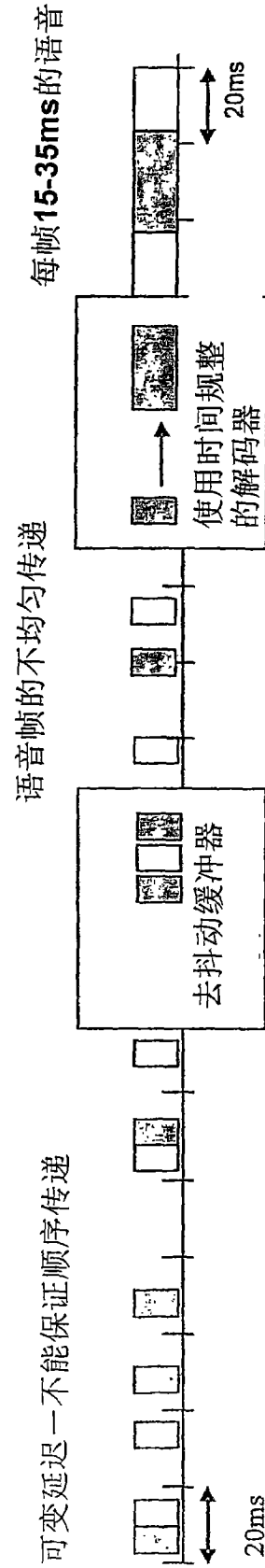


图 26

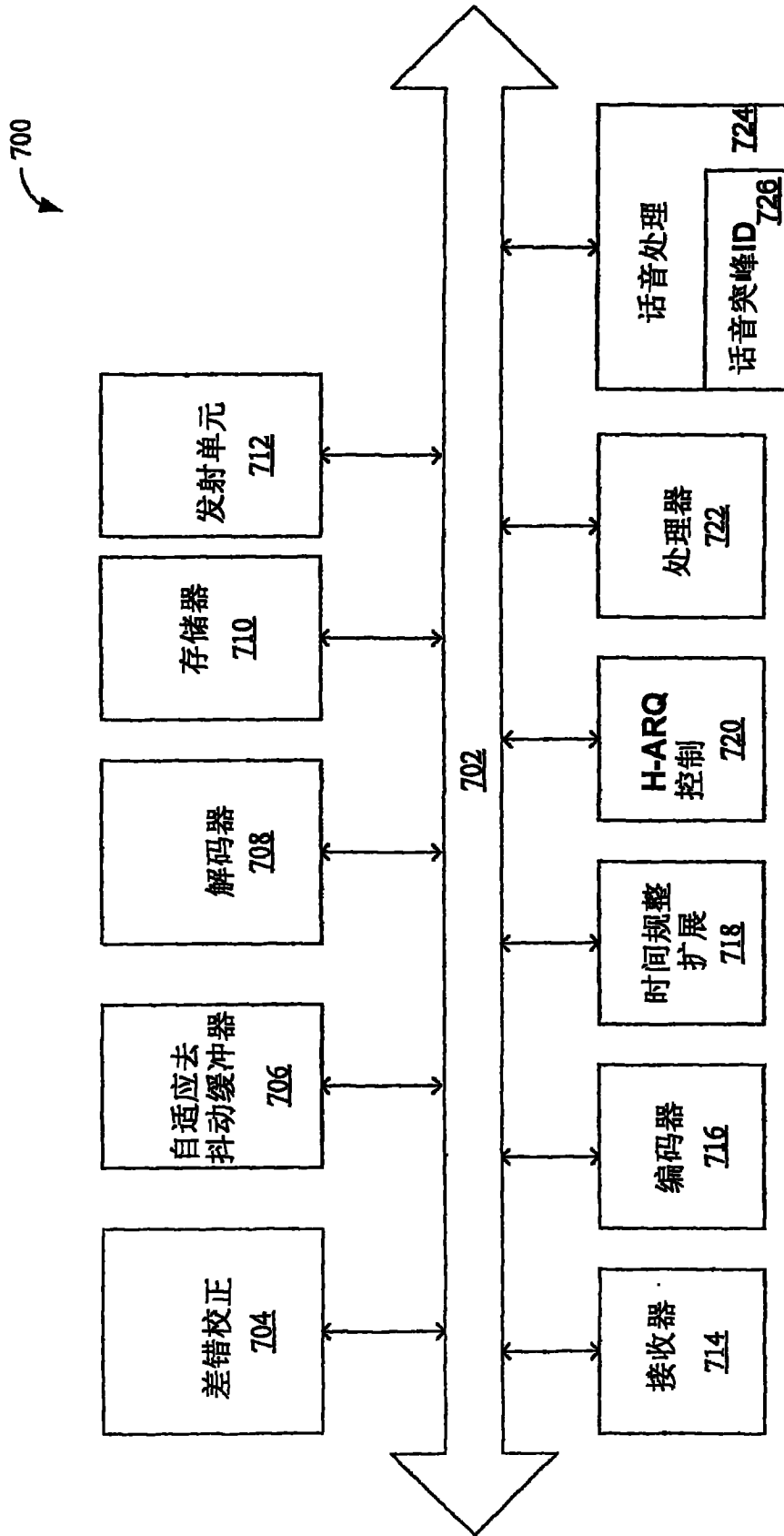


图 27

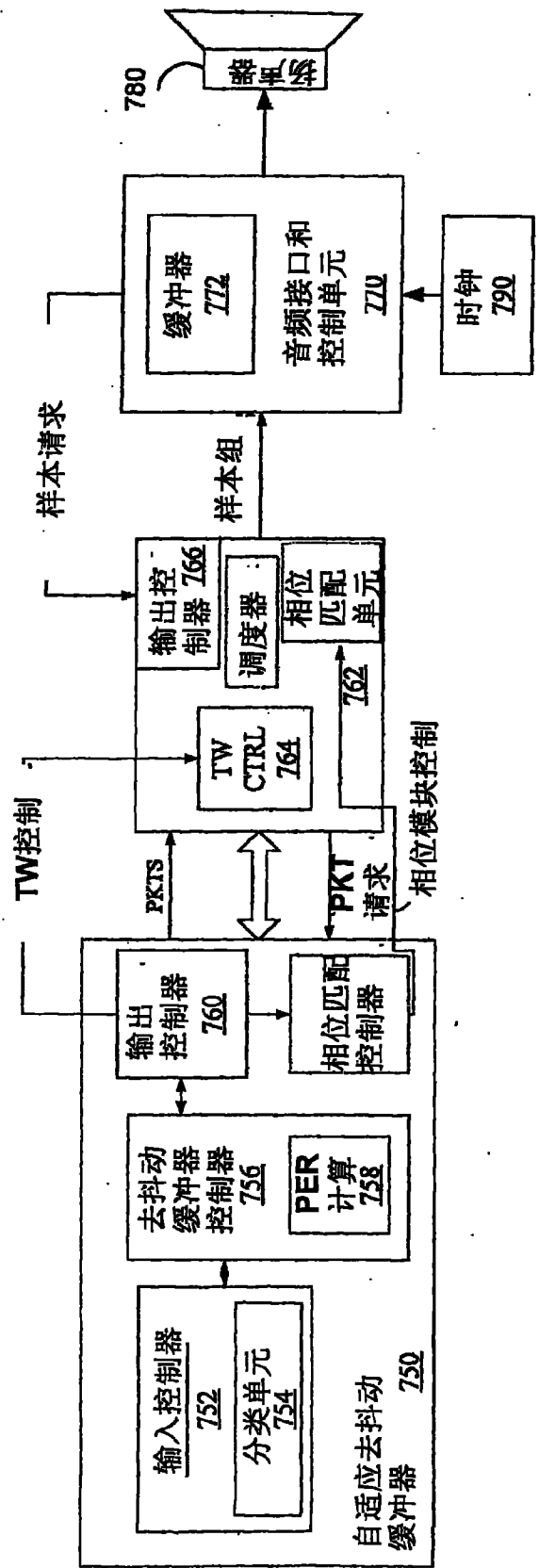


图 28

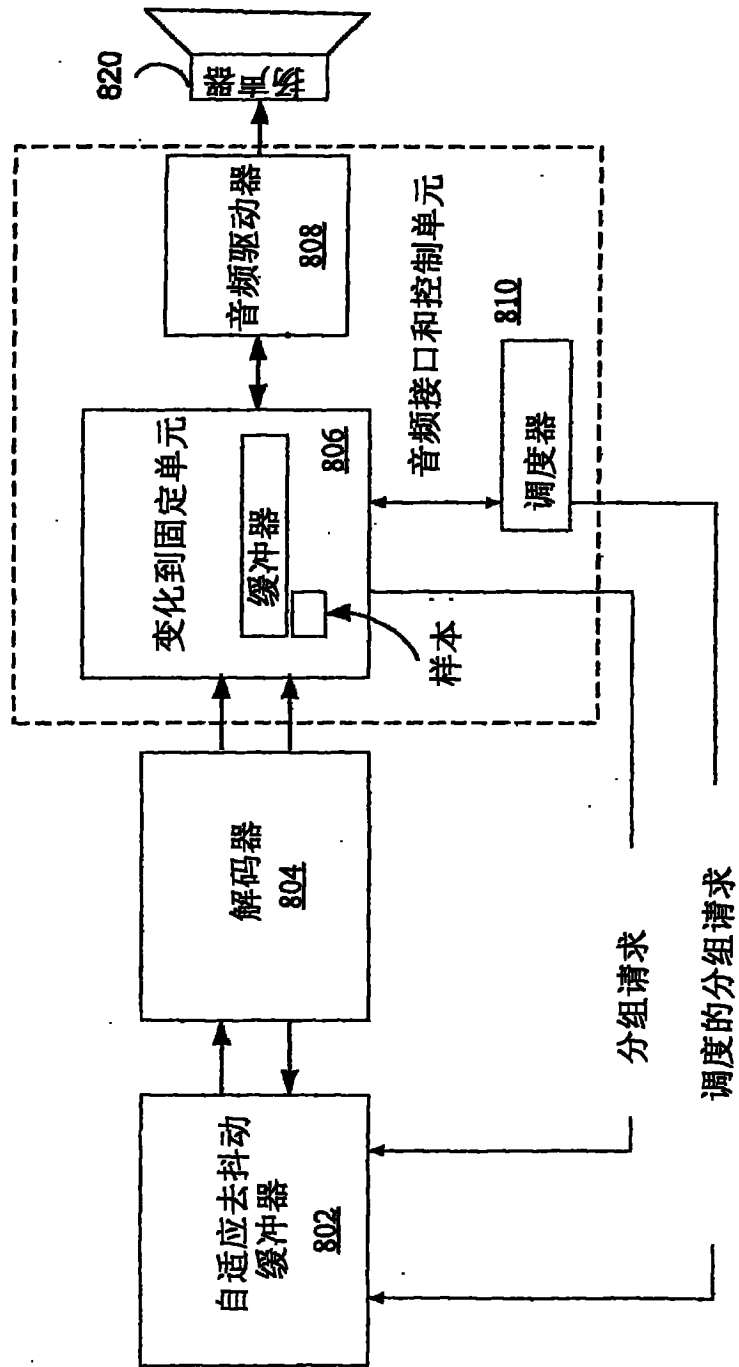


图 29

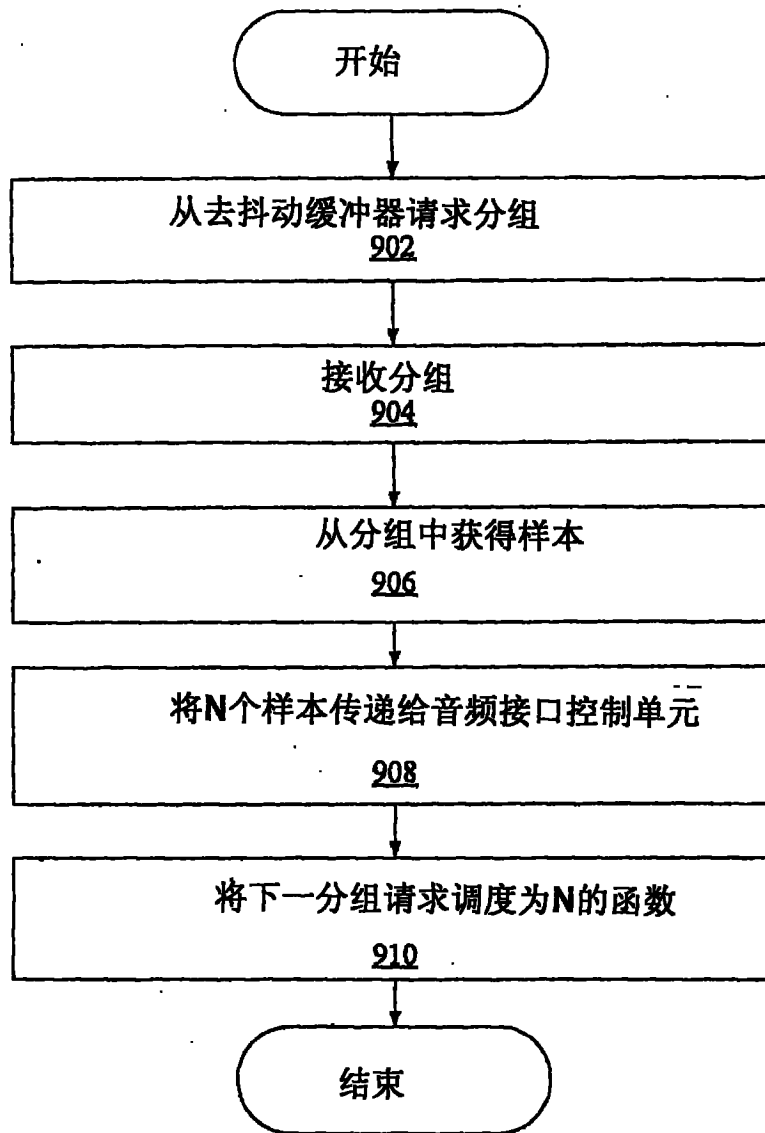


图 30

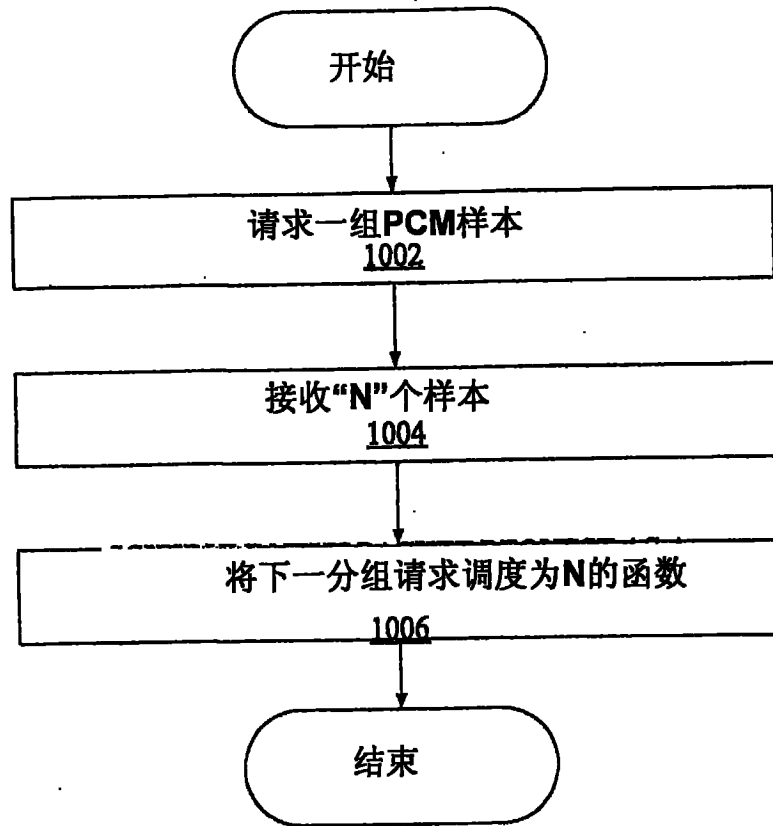


图 31

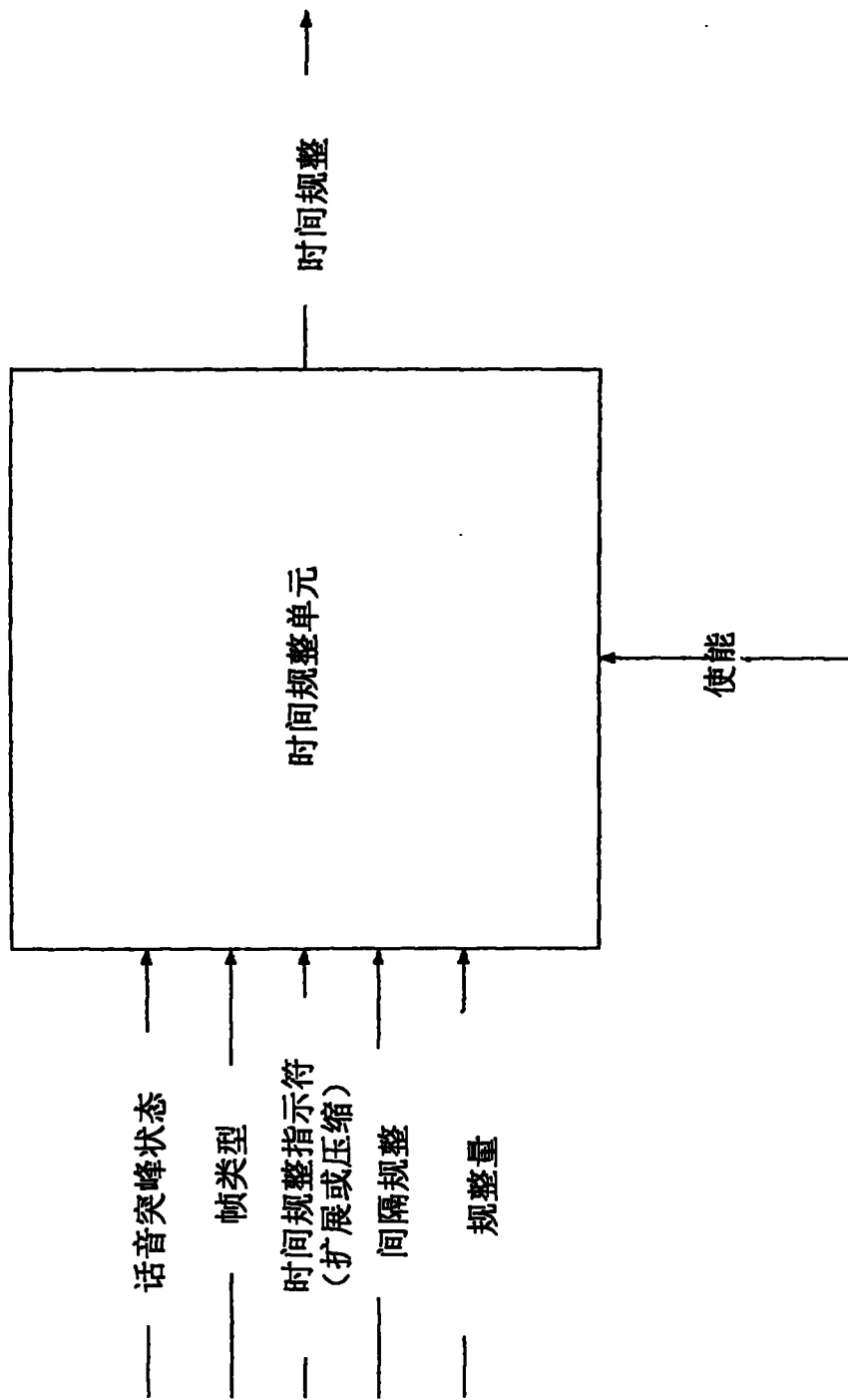


图 32

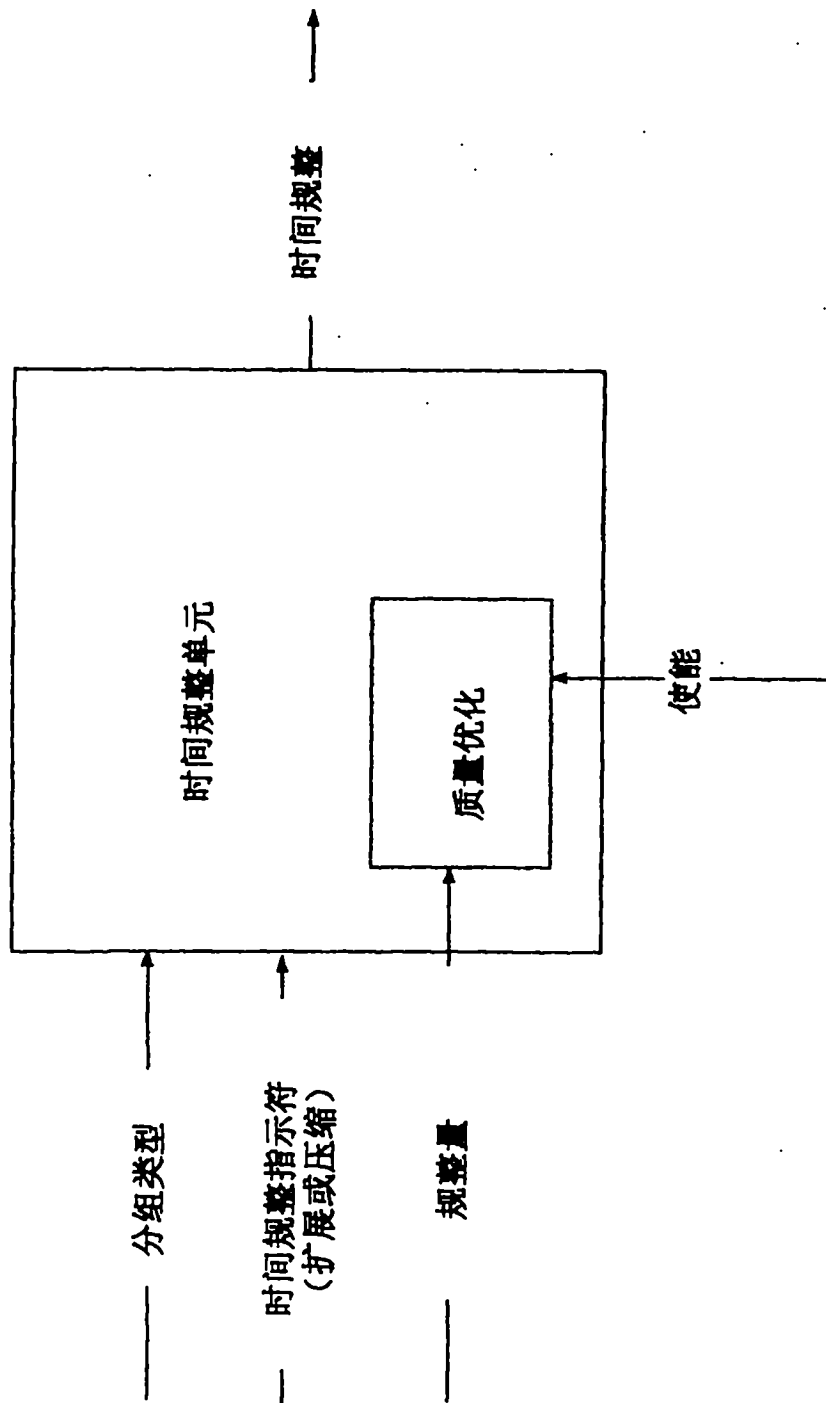


图 33