



[12] 发明专利说明书

专利号 ZL 200610005700.5

[45] 授权公告日 2008年7月9日

[11] 授权公告号 CN 100401665C

[22] 申请日 2006.1.19

[21] 申请号 200610005700.5

[73] 专利权人 华为技术有限公司

地址 518129 广东省深圳市龙岗区坂田华为总部办公楼

[72] 发明人 敬晓云

[56] 参考文献

CN1551552A 2004.12.1

US2002/0045449A1 2002.4.18

US2002/0167926A1 2002.11.14

GB2396998A 2007.7.7

US2003/0086427A1 2003.5.8

审查员 刘承恩

[74] 专利代理机构 北京德琦知识产权代理有限公司

代理人 宋志强 麻海明

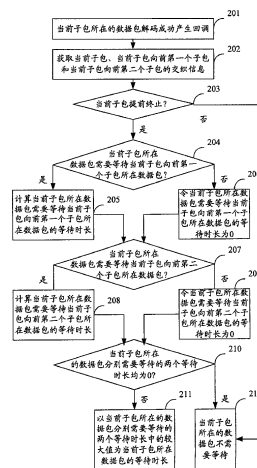
权利要求书 3 页 说明书 14 页 附图 4 页

[54] 发明名称

一种判断反向数据包字节丢失的方法

[57] 摘要

本发明公开了一种判断反向数据包字节丢失的方法，该方法的一种实现方式为：确定当前解码成功的反向数据包的等待时长，并判断在等待时长之内，是否收到在当前反向数据包中的字节之前发送的反向字节，如果没有收到则认为发生反向字节丢失。该方法的另一种实现方式为设置包缓存队列，包括：确定当前解码成功的反向数据包的等待时长，并将该反向数据包放入包缓存队列，在等待时长到达时，将包缓存队列中，上述解码成功的反向数据包及其之前的反向数据包选出，作为排好序的反向数据包，并判断在排好序的反向数据包中是否有字节缺失，如果是则认为发生反向字节丢失。采用本发明提供的技术方案，可以避免错误的将乱序反向数据包认定为反向字节丢失。



1、一种判断反向数据包字节丢失的方法，其特征在于，该方法包括：

步骤 A1，确定当前解码成功的反向数据包的等待时长，并判断在当前反向数据包中的字节之前发送的反向字节是否缺失，如果有则执行步骤 B1；

步骤 B1，判断在步骤 A1 中得到的等待时长之内，是否收到在当前反向数据包中的字节之前发送的反向字节，如果没有收到则认为发生反向字节丢失，通知发送方重发丢失的反向字节。

2、根据权利要求 1 所述的判断反向数据包字节丢失的方法，其特征在于，步骤 A1 所述确定当前解码成功的反向数据包的等待时长包括：

步骤 A11，基站收发信机将当前解码成功的反向数据包，以及当前反向数据包解码成功时产生的交织信息传送给基站控制器；

步骤 A12，基站控制器判断当前反向数据包是否提前结束，如果是则执行步骤 A13，否则认为等待时长为 0；

步骤 A13，基站控制器根据当前反向数据包解码成功时产生的交织信息，计算当前反向数据包的等待时长。

3、根据权利要求 1 所述的判断反向数据包字节丢失的方法，其特征在于，步骤 A1 所述确定当前解码成功的反向数据包的等待时长包括：

步骤 A21，基站收发信机判断当前反向数据包是否提前结束，如果是则执行步骤 A22，否则认为等待时长为 0；

步骤 A22，基站收发信机根据当前反向数据包解码成功时产生的交织信息，计算当前反向数据包的等待时长；

步骤 A23，基站收发信机将解码成功的反向数据包，以及计算出的等待时长传送给基站控制器。

4、根据权利要求 1 到 3 任意一项所述的判断反向数据包字节丢失的方法，其特征在于，步骤 B1 所述判断在步骤 A1 中得到的等待时长之内，是否收到在当前反向数据包中的字节之前发送的反向字节由基站控制器完成。

5、一种判断反向数据包字节丢失的方法，其特征在于，设置包缓存队列，该方法包括：

步骤 A2，确定当前解码成功的反向数据包的等待时长，并将当前解码成功的反向数据包放入包缓存队列，然后开始为当前解码成功的反向数据包计时；

步骤 B2，在步骤 A2 中得到的等待时长到达时，将包缓存队列中，步骤 A2 中解码成功的反向数据包及其之前的反向数据包选出，作为排好序的反向数据包；

步骤 C2，判断在排好序的反向数据包中是否有字节缺失，如果是则认为发生反向字节丢失，通知发送方重发丢失的反向字节。

6、根据权利要求 5 所述的判断反向数据包字节丢失的方法，其特征在于，步骤 A2 所述将当前解码成功的反向数据包放入包缓存队列为：

根据当前解码成功的反向数据包所携带的帧序号，判断当前解码成功的反向数据包，与包缓存队列中已有的反向数据包，在发送时的先后顺序，然后将当前解码成功的反向数据包放入包缓存队列。

7、根据权利要求 5 所述的判断反向数据包字节丢失的方法，其特征在于，步骤 A2 所述确定当前解码成功的反向数据包的等待时长包括：

步骤 A31，基站收发信机将当前解码成功的反向数据包，以及当前反向数据包解码成功时产生的交织信息传送给基站控制器；

步骤 A32，基站控制器判断当前反向数据包是否提前结束，如果是则执行步骤 A33，否则认为等待时长为 0；

步骤 A33，基站控制器根据当前反向数据包解码成功时产生的交织信息，计算当前反向数据包的等待时长。

8、根据权利要求 5 所述的判断反向数据包字节丢失的方法，其特征在于，步骤 A2 所述确定当前解码成功的反向数据包的等待时长包括：

步骤 A41，基站收发信机判断当前反向数据包是否提前结束，如果是则执行步骤 A42，否则认为等待时长为 0；

步骤 A42，基站收发信机根据当前反向数据包解码成功时产生的交织信息，

计算当前反向数据包的等待时长。

9、根据权利要求 8 所述的判断反向数据包字节丢失的方法，其特征在于，步骤 A2 所述将当前解码成功的反向数据包放入包缓存队列，然后开始为当前解码成功的反向数据包计时为：

基站收发信机将解码成功的反向数据包，以及步骤 A42 中计算出的等待时长传送给基站控制器，基站控制器将当前解码成功的反向数据包放入包缓存队列，然后开始为当前解码成功的反向数据包计时。

10、根据权利要求 8 所述的判断反向数据包字节丢失的方法，其特征在于，步骤 A2 所述将当前解码成功的反向数据包放入包缓存队列，然后开始为当前解码成功的反向数据包计时为：

基站收发信机根据将当前解码成功的反向数据包放入包缓存队列，然后开始为当前解码成功的反向数据包计时；

步骤 B2 所述将包缓存队列中，步骤 A2 中解码成功的反向数据包及其之前反向数据包选出，作为排好序的反向数据包为：

基站收发信机将包缓存队列中，步骤 A2 中解码成功的反向数据包及其之前反向数据包选出，作为排好序的反向数据包传送给基站控制器。

11、根据权利要求 5 到 10 任意一项所述的判断反向数据包字节丢失的方法，其特征在于，步骤 C2 所述判断在排好序的反向数据包中是否有字节缺失由基站控制器完成。

一种判断反向数据包字节丢失的方法

技术领域

本发明涉及移动通信技术，特别是涉及一种判断反向数据包字节丢失的方法。

背景技术

码分多址（Code Division Multiple Access, CDMA）2000 系统的新型数据业务（Evolution Data Optimized, EV-DO）已经由最初的 Rev 0 版本发展到了 Rev A 版本。图 1 示意了在 EV-DO Rev A 中，作为接受方的基站收到作为发送方的移动终端发来的反向数据包时的效果。

在 EV-DO Rev A 中，在移动终端上，每个要发送的反向数据包被分成 4 个子包，其中第一个子包携带了原数据包的所有有效信息和部分冗余信息，其他 3 个数据包均携带帮助解码的冗余信息；同时，有三个交织号分别为 0、1、2 的三个交织，这三个交织用于发送分属三个不同反向数据包的子包。一个反向数据包的所有子包，由交织号相同的交织发送。由于移动终端到基站之间的无线信道的变化很剧烈，在基站上，对于每个数据包而言，有可能收到第一个子包就能成功解码，也有可能是在收到第二、第三、第四子包后才能解码成功，甚至最终都没有解码成功。如果基站在收到第一子包后没有解码成功，那么会通过由基站到移动终端的前向反馈发送 Nak 给移动终端，让移动终端再发第二个子包；如果基站在收到第一子包后解码成功，那么会通过前向反馈发 Ack 给移动终端，移动终端开始第一子包所在交织发送一个新的数据包，依此类推。在图 1 中，反向数据包 A 由交织号为 0 的交织发送，并且在第一个子包 A0 就解码成功，基站通过前向反馈发送 Ack 给移动终端，这样移动终端下一次在交织号为 0 的交织发送的就是反向数据包 D 的第一

个子包 D0; 反向数据包 B 的 4 个子包 B0、B1、B2、和 B3, 全部由交织号为 1 的交织发送, 在第四个子包 B3 才解码成功, 因此在基站收到 B0、B1 和 B2 后都会通过前向反馈发送 Nak 给移动终端, 让移动终端接着发下一个子包; 反向数据包 C 由交织号为 2 的交织发送, 并且在第一个子包 C0 就解码成功, 基站通过前向反馈发送 Ack 给移动终端, 这样下一次在交织号为 2 的交织发送的就是反向数据包 E 的第一个子包 E0, 等等。采用这种交织发送的方法, 可以实现反向数据包的提前终止, 提高了反向数据速率, 但同时也使得反向数据包可能出现顺序错乱。例如在图 1 中, 在发送端, 反向数据包 B 是在反向数据包 C 之前开始发送的; 但是在接受端, 由于反向数据包 B 在第四个子包才解码成功, 而反向数据包 C 在第一个子包就解码成功, 因此 C 比 B 先完成解码。

反向数据包的解码工作是由基站收发信机 (Base Transceiver Station, BTS) 完成的, BTS 将解码完成的反向数据包传送到基站控制器 (Base Station Controller, BSC)。在现有技术中, 如果出现上述的反向数据包 C 比反向数据包 B 先完成解码的情况, BSC 上的无线链路协议 (Radio Link Protocol, RLP) 模块在处理反向数据包的字节时, 发现反向数据包 A 所包含的字节后面紧跟的是反向数据包 C 所包含的字节, 则认为反向数据包 B 所包含的字节丢失, 通知移动终端重发反向数据包 B。实际上反向数据包 B 只是延迟到达而已, 这样就会造成不必要的反向数据包重发, 影响反向通信的速率。

发明内容

有鉴于此, 本发明的主要目的在于提供一种反向判断数据包字节丢失的方法, 以防止在乱序反向数据包的情况下基站控制器错误认为反向数据包中的字节丢失, 从而避免不必要的反向数据包重复发送。

为了达到上述目的, 本发明提供了一种判断反向数据包字节丢失的方法, 该方法的第一种实现方式包括:

步骤 A1, 确定当前解码成功的反向数据包的等待时长, 并判断在当前反向

数据包中的字节之前发送的反向字节是否缺失，如果有则执行步骤 B1；

步骤 B1，判断在步骤 A1 中得到的等待时长之内，是否收到在当前反向数据包中的字节之前发送的反向字节，如果没有收到则认为发生反向字节丢失，通知发送方重发丢失的反向字节。

其中，步骤 A1 所述确定当前解码成功的反向数据包的等待时长包括：

步骤 A11，基站收发信机将当前解码成功的反向数据包，以及当前反向数据包解码成功时产生的交织信息传送给基站控制器；

步骤 A12，基站控制器判断当前反向数据包是否提前结束，如果是则执行步骤 A13，否则认为等待时长为 0；

步骤 A13，基站控制器根据当前反向数据包解码成功时产生的交织信息，计算当前反向数据包的等待时长。

其中，步骤 A1 所述确定当前解码成功的反向数据包的等待时长包括：

步骤 A21，基站收发信机判断当前反向数据包是否提前结束，如果是则执行步骤 A22，否则认为等待时长为 0；

步骤 A22，基站收发信机根据当前反向数据包解码成功时产生的交织信息，计算当前反向数据包的等待时长；

步骤 A23，基站收发信机将解码成功的反向数据包，以及计算出的等待时长传送给基站控制器。

其中，步骤 B1 所述判断在步骤 A1 中得到的等待时长之内，是否收到在当前反向数据包中的字节之前发送的反向字节由基站控制器完成。

该方法的第二种实施方式为，设置包缓存队列，该方法包括：

步骤 A2，确定当前解码成功的反向数据包的等待时长，并将当前解码成功的反向数据包放入包缓存队列，然后开始为当前解码成功的反向数据包计时；

步骤 B2，在步骤 A2 中得到的等待时长到达时，将包缓存队列中，步骤 A2 中解码成功的反向数据包及其之前的反向数据包选出，作为排好序的反向数据包；

步骤 C2，判断在排好序的反向数据包中是否有字节缺失，如果是则认为发

生反向字节丢失，通知发送方重发丢失的反向字节。

其中，步骤 A2 所述将当前解码成功的反向数据包放入包缓存队列为：

根据当前解码成功的反向数据包所携带的帧序号，判断当前解码成功的反向数据包，与包缓存队列中已有的反向数据包，在发送时的先后顺序，然后将当前解码成功的反向数据包放入包缓存队列。

其中，步骤 A2 所述确定当前解码成功的反向数据包的等待时长包括：

步骤 A31，基站收发信机将当前解码成功的反向数据包，以及当前反向数据包解码成功时产生的交织信息传送给基站控制器；

步骤 A32，基站控制器判断当前反向数据包是否提前结束，如果是则执行步骤 A33，否则认为等待时长为 0；

步骤 A33，基站控制器根据当前反向数据包解码成功时产生的交织信息，计算当前反向数据包的等待时长。

其中，步骤 A2 所述确定当前解码成功的反向数据包的等待时长包括：

步骤 A41，基站收发信机判断当前反向数据包是否提前结束，如果是则执行步骤 A42，否则认为等待时长为 0；

步骤 A42，基站收发信机根据当前反向数据包解码成功时产生的交织信息，计算当前反向数据包的等待时长。

其中，步骤 A2 所述将当前解码成功的反向数据包放入包缓存队列，然后开始为当前解码成功的反向数据包计时为：

基站收发信机将解码成功的反向数据包，以及步骤 A42 中计算出的等待时长传送给基站控制器，基站控制器将当前解码成功的反向数据包放入包缓存队列，然后开始为当前解码成功的反向数据包计时。

其中，步骤 A2 所述将当前解码成功的反向数据包放入包缓存队列，然后开始为当前解码成功的反向数据包计时为：

基站收发信机根据将当前解码成功的反向数据包放入包缓存队列，然后开始为当前解码成功的反向数据包计时；

步骤 B2 所述将包缓存队列中，步骤 A2 中解码成功的反向数据包及其之前

反向数据包选出，作为排好序的反向数据包为：

基站收发信机将包缓存队列中，步骤 A2 中解码成功的反向数据包及其之前反向数据包选出，作为排好序的反向数据包传送给基站控制器。

其中，步骤 C2 所述判断在排好序的反向数据包中是否有字节缺失由基站控制器完成。

采用本发明所提供的技术方案，对于乱序反向数据包，首先计算等待时长，只有在超出等待时长以后才认为发生反向数据包中所包含的字节丢失，这样就避免了错误的将乱序反向数据包认定为反向字节丢失。进一步，等待时长可以用于将乱序反向数据包排序，从而加快基站控制器中后续处理的速度。而计算等待时长和排序的工作可以由基站收发信机完成，这样就减轻了基站控制器的计算压力。

附图说明

图 1 是 CDMA2000 EV-DO Rev A 中反向数据包的发送方案；

图 2 是本发明提供的计算等待时长的流程图；

图 3 是本发明提供的判断反向数据包字节丢失方法实施例一的流程图；

图 4 是本发明提供的判断反向数据包字节丢失方法实施例二的流程图。

具体实施方式

本发明的核心思想在于，对于乱序反向数据包，首先计算等待时长，根据等待时长来判断是否发生了数据包中的字节丢失。

为使本发明的目的、技术方案和优点更加清楚，下面结合附图及具体实施例对本发明作进一步地详细描述。

请参考图 2，图 2 是本发明提供的计算等待时长的流程图。为了便于理解，且不失一般性，以图 1 所示的反向数据包 C 为例说明计算过程。根据图 1，从直观上看，反向数据包 C 不需要等待反向数据包 A，但需要等待反向数据包 B。

步骤 201, 当前子包所在的反向数据包解码成功产生回调。

这里, C0 是当前子包, 解码成功产生回调, 启动对于反向数据包 C 的等待时长计算流程。

步骤 202, 获取当前子包、当前子包向前第一个子包和当前子包向前第二个子包的交织信息。

对于当前子包 C0 来说, 当前子包向前第一个子包是子包 B0, 当前子包向前第二个子包是子包 A0。

所述的交织信息包括三个数据域: 交织号、子包号和交织状态。

交织号表示该子包位于哪个交织, 其取值范围是 0 到 2, 因为总共有 3 个交织。在图 1 中, 子包 A0 位于交织号 0, 子包 B0 位于交织号 1, 子包 C0 位于交织号 2。

子包号表示该子包是所在反向数据包的第几个子包, 其取值范围是 0 到 3, 因为对于作为接受端的基站来说, 一个反向数据包最多包括 4 个子包。在图 1 中, A0、B0 和 C0 都是各自所在的反向数据包的第 0 个子包。

子包状态表示该子包所在反向数据包的状态, 可以是以下这四个值:

CRCPass, 反向数据包解码成功, 且该反向数据包是当前解码子包所在的反向数据包;

CRCFail, 反向数据包没有解码成功;

Invalid, 反向数据包解码成功, 且该反向数据包不是当前解码子包所在的反向数据包; 或者该反向数据包没有数据发送; 或者第四子包解码失败, 且该反向数据包不是当前解码子包所在的反向数据包;

SubPacketFail, 该子包无效, 不能用于所在反向数据包的解码。

在图 1 中, 子包 A0 的子包状态值为 Invalid, 子包 B0 的子包状态值为 CRCFail, 子包 C0 的子包状态值为 CRCPass;

需要说明的是, 以上仅仅给出了子包状态的四种取值所表示的与计算等待时长有关的意义。

步骤 203, 判断当前子包是否提前终止, 如果是则执行步骤 204, 否则

执行步骤 212.

判断当前子包是否提前终止的方法是看当前子包的交织信息中的子包号是否为 3，如果为 3 表示没有提前终止，如果不为 3 表示提前终止。

当前子包是 C0 时，当前子包的交织信息中的子包号为 0，表示提前终止。

步骤 204，判断当前子包所在反向数据包是否需要等待当前子包向前第一个子包所在反向数据包，如果是则执行步骤 205，否则执行步骤 206。

判断当前子包所在反向数据包是否需要等待当前子包向前第一个子包所反向在数据包的方法是，如果以下三个条件中的任意一个条件成立，则认为当前子包所在反向数据包不需要等待当前子包向前第一个子包所在反向数据包，否则认为当前子包所在反向数据包需要等待当前子包向前第一个子包所在反向数据包。这三个条件是：

(1) 当前子包交织信息中的子包号，大于当前子包向前第一个子包的交织信息中的子包号；

(2) 当前子包向前第一个子包的交织信息中的子包状态为 Invalid；

(3) 当前子包向前第一个子包的交织信息中的子包状态为 SubPacketFail，且当前子包向前第一个子包的交织信息中的子包号为 3。

当前子包是 C0 时，当前子包向前第一个子包是 B0，且上述三个条件均不满足，因此当前子包 C0 所在的反向数据包需要等待当前子包向前第一个子包 B0 所在的反向数据包。

步骤 205，计算当前子包所在反向数据包需要等待当前子包向前第一个子包所在反向数据包的等待时长，然后执行步骤 207。

等待时长的计算方法是，首先将当前子包的交织信息中的交织号，减去当前子包向前第一个子包的交织信息中的交织号，再加上 3，将加法所得的结果除以 3 以后取余数；然后用 3 减去当前子包向前第一个子包的交织信息中的子包号，将得到的差乘以 3，并且用乘法的结果减去前面一步得到的余数，就是当前子包所在的反向数据包需要等待当前子包向前第一个子包所在

的反向数据包的等待时长。

根据以上计算方法，子包 C0 所在的反向数据包应该等待子包 B 0 所在的反向数据包的等待时长为 8，单位是子包。

步骤 206，令当前子包所在反向数据包需要等待当前子包向前第一个子包所在反向数据包的等待时长为 0。

步骤 207，判断当前子包所在反向数据包是否需要等待当前子包向前第二个子包所在反向数据包，如果是则执行步骤 208，否则执行步骤 209。

判断当前子包所在反向数据包是否需要等待当前子包向前第二个子包所在反向数据包的方法是，如果以下三个条件中的任意一个条件成立，则认为当前子包所在反向数据包不需要等待当前子包向前第二个子包所在反向数据包，否则认为当前子包所在反向数据包需要等待当前子包向前第二个子包所在反向数据包。这三个条件是：

(1) 当前子包交织信息中的子包号，大于当前子包向前第二个子包的交织信息中的子包号；

(2) 当前子包向前第二个子包的交织信息中的子包状态为 Invalid；

(3) 当前子包向前第二个子包的交织信息中的子包状态为 SubPacketFail，且当前子包向前第二个子包的交织信息中的子包号为 3。

当前子包是 C0 时，当前子包向前第二个子包是 A0，且上述三个条件中的 (2) 满足，因此当前子包 C0 所在的反向数据包不需要等待当前子包向前第二个子包 A0 所在的反向数据包。

步骤 208，计算当前子包所在反向数据包需要等待当前子包向前第二个子包所在反向数据包的等待时长，然后执行步骤 210。

等待时长的计算方法是，首先将当前子包的交织信息中的交织号，减去当前子包向前第二个子包的交织信息中的交织号，再加上 3，将加法所得的结果除以 3 以后取余数；然后用 3 减去当前子包向前第二个子包的交织信息中的子包号，将得到的差乘以 3，并且用乘法的结果减去前面一步得到的余数，就是当前子包所在的反向数据包需要等待当前子包向前第二个子包所在

的反向数据包的等待时长。

步骤 209, 令当前子包所在反向数据包需要等待当前子包向前第二个子包所在反向数据包的等待时长为 0。

步骤 210, 判断当前子包所在的反向数据包分别需要等待的两个等待时长是否均为 0, 如果是则执行步骤 212, 否则执行步骤 211。

步骤 211, 以当前子包所在的反向数据包分别需要等待的两个等待时长中较大的一个作为当前子包所在反向数据包的等待时长。

由于当前子包 C0 所在的反向数据包 C 需要等待当前子包向前第一个子包 B0 所在的数据包 B, 且等待时长为 8; 而当前子包 C0 所在的反向数据包 C 需要等待当前子包向前第二个子包 A0 所在的数据包 A 的等待时长为 0, 因此当前子包 C0 所在的反向数据包的等待时长为 8。

步骤 212, 当前子包所在的反向数据包不需要等待, 也就是说当前子包的等待时长为 0。

请参考图 3, 图 3 是本发明提供的判断反向数据包字节丢失方法实施例一的流程图。

步骤 301, BTS 将反向数据包解码, 并将交织信息传送给 BSC。

在移动通信基站系统中, 对反向数据包进行解码是由 BTS 完成的, 在对计算等待时长的流程的描述中提到, 在 BTS 对当前子包所在的反向数据包解码成功后, 会得到当前子包、当前子包向前第一个子包和当前子包向前第二个子包的交织信息, BTS 将成功解码的反向数据包, 以及对应的三个交织信息都传送给 BSC。

步骤 302, BSC 根据交织信息, 计算当前反向数据包的等待时长。

步骤 302 就是步骤 201 至步骤 212 所进行的流程。

步骤 303, BSC 将当前反向数据包中的字节放入重排缓冲区。

重排缓冲区是现有技术的 RLP 协议所用来对反向数据包中的字节进行排序用的。移动终端会按照 RLP 协议为所发送的每个字节分配一个 RLP 序号, RLP 序号按照字节发送的先后顺序递增; BSC 在接收到反向数据包以后,

将反向数据包所包含的字节，根据 RLP 序号放入重排缓冲区中。只有连续的、超过一定长度门限的、且最早进入重排缓冲区的一段字节才能被送入后续处理模块。

例如，假设长度门限是 20 字节，重排缓冲区中现有字节为 RLP 序号从 30 到 35 的总共 6 个连续字节，由于连续字节的长度没有超过长度门限，因此不能送入后续处理模块。现在收到第一个反向数据包，其中包括 RLP 序号为 40 到 50 的 11 个连续字节。按照 RLP 序号放入重排缓冲区后，连续字节的数目仍然没有超过 20 个。如果又收到第二个反向数据包，其中包括 RLP 序号为 36 到 39 的 4 个连续字节。按照 RLP 序号放入重排缓冲区后，连续字节的 RLP 序号从 30 到 40，连续字节的数目为 21 个，超过了长度门限，则将这 21 个字节全部发送到后续处理模块。

步骤 304，判断重排缓冲区中当前反向数据包中的字节之前是否有字节缺失，如果有则执行步骤 305，否则执行步骤 308。

所谓的字节缺失，即步骤 303 所举的例子中，收到第二个反向数据包之前的情况，RLP 序号为 36 到 39 的字节就是缺失字节。

步骤 305，BSC 将计时器置零后启动计时器。

计时器的作用在于，字节缺失有可能是由乱序反向数据包造成的，也有可能是反向数据包丢失造成的。如果是由乱序反向数据包造成的，则在步骤 302 所计算出的等待时长内，就应该收到包含缺失字节的反向数据包。如果超过这个时间还没有收到，则该反向数据包在从移动终端到基站的通信路径上丢失，需要移动终端重新发送该缺失字节。

步骤 306，BSC 判断计时器是否到达等待时长，如果是则执行步骤 311，否则执行步骤 307。

步骤 307，BSC 判断是否收到 BTS 传来的包含缺失字节的反向数据包，如果收到则执行步骤 308，否则返回执行步骤 306。

步骤 308，BSC 判断重排缓冲区中当前反向数据包中的字节所在的一段连续字节的长度是否超过长度门限，如果是则执行步骤 309，否则执行步骤

310.

步骤 309, BSC 将重排缓冲区中当前反向数据包中的字节所在的一段连续字节进行后续处理。

步骤 310, BSC 准备处理下一个反向数据包。

步骤 311, 认为 BTS 未能成功解码缺失字节所在的反向数据包, 要求移动终端重发缺失字节。

在实施例一中, 从步骤 302 到步骤 311, 都是由 BSC 完成的。在实际的移动通信基站系统中, 一个 BSC 往往有很多个 BTS, 对于每个 BTS 传送来的反向数据包, BSC 都要进行步骤 302 到步骤 311 的流程。这样就占用了 BSC 的大量处理能力。另一方面, BSC 计算等待时长需要 BTS 在传送当前反向数据包的同时发送交织信息, 这会降低 BTS 和 BSC 之间接口的利用效率。例如, 三个子包的交织信息大约为 3 字节, 而在基于互联网协议的语音业务 (Voice on Internet Protocol, VoIP) 中, 一个反向数据包的长度才 22 字节, 传送交织信息使得 BTS 和 BSC 之间的接口利用率降低了近 10%。

作为实施例一的一种改进, 步骤 301 中, BTS 可以先不将解码后的反向数据包以及交织信息传送给 BSC, 而是由 BTS 自身执行步骤 302, 然后将解码后的反向数据包和计算出的等待时长一起传送给 BSC。这样就可以提高 BTS 和 BSC 之间接口的利用效率, 同时也分担了 BSC 的一部分计算量。

但是, 对于反向数据包中字节的处理只能在 BSC 进行, 这样就使得无法将步骤 303 到步骤 311 移到 BTS 中进行。这样, BSC 的计算量还是很大, 并且传送等待时长还是需要占用 BTS 和 BSC 之间接口的资源。

本发明的实施例二可以解决这个问题。请参考图 4, 图 4 是本发明提供的判断反向数据包字节丢失方法实施例二的流程图。

为了对反向数据包排序, BTS 为三个交织各维护一个定时器, 每个定时器的初始值都是 0。每个定时器还各自对应一个启动帧序号 (Frame Sequence Number, FSN), 表示启动这个定时器的子包所在反向数据包的 FSN。FSN 是移动终端在发送反向数据包时为每个数据包分配的编号, FSN 与数据包是

一一对应的关系，分配 FSN 是按照发送数据包的先后顺序。BTS 还为维护一个反向数据包缓存队列，简称包缓存队列，包缓存队列中的反向数据包 FSN 由小到大排列。

步骤 401，BTS 收到子包。

步骤 402，BTS 判断是否在当前子包完成当前子包所在反向数据包的解码，如果是则执行步骤 410，否则执行步骤 403。

步骤 403，判断当前子包所在交织的定时器是否为 0，如果是则执行步骤 405，否则执行步骤 404。

步骤 404，执行步骤 420。

步骤 405，判断除当前子包所在的交织外，其他两个交织的定时器是否都不为 0，如果是则执行步骤 407；如果除当前子包所在的交织外，其他两个交织的定时器中的任意一个为 0 则执行步骤 406。

步骤 406，判断除当前子包所在的交织外，其他两个交织的定时器是否都为 0，如果是则执行步骤 409；如果除当前子包所在的交织外，其他两个交织的定时器中有且只有一个为 0 则执行步骤 408。

步骤 407，比较判断除当前子包所在的交织外，其他两个交织的定时器的启动 FSN，将包缓存队列中，值较小的启动 FSN 所对应的反向数据包之前的反向数据包发送到 BSC，执行步骤 420。

步骤 408，将包缓存队列中，不为 0 的定时器的启动 FSN 所对应的反向数据包之前的反向数据包发送到 BSC，执行步骤 420。

步骤 409，将包缓存队列中的反向数据包全部发送到 BSC，执行步骤 420。

步骤 410，将当前子包所在交织的定时器清零，并将当前子包按照 FSN 的顺序插入包缓存队列。

步骤 411，BTS 根据在成功解码反向数据包时获取的当前子包、当前子包向前第一个子包和当前子包向前第二个子包的交织信息，分别计算出当前子包所在反向数据包需要等待当前子包向前第一个子包的等待时长 t_1 ，和当

前子包所在反向数据包需要等待当前子包向前第二个子包的等待时长 t_2 。

步骤 412, 判断 t_1 和 t_2 是否均为 0, 如果是则执行步骤 414, 如果 t_1 和 t_2 中任意一个不为 0 则执行步骤 413。

步骤 413, 判断 t_1 是否为 0, 如果是则 t_2 必然不为 0, 执行步骤 418, 否则执行步骤 415。

步骤 414, 返回执行步骤 405。

步骤 415, 判断当前子包向前第一个子包所在交织的定时器是否为 0, 如果是则执行步骤 416, 否则执行步骤 417。

步骤 416, 将当前子包向前第一个子包所在交织的定时器的值设为 t_1 的值, 并将当前子包向前第一个子包所在交织的定时器的启动 FSN 设为当前子包所在数据包的 FSN。

步骤 417, 判断 t_2 是否为 0, 如果是则执行步骤 420, 否则执行步骤 418。

步骤 418, 判断当前子包向前第二个子包所在交织的定时器是否为 0, 如果是则执行步骤 419, 否则执行步骤 420。

步骤 419, 将当前子包向前第二个子包所在交织的定时器的值设为 t_2 的值, 并将当前子包向前第二个子包所在交织的定时器的启动 FSN 设为当前子包所在数据包的 FSN。

步骤 420, 将不为 0 的定时器减 1 后结束对当前子包的处理流程, 准备处理下一个子包。

这样, 从 BTS 传送到 BSC 的反向数据包都是按照发射端的发射顺序排好的。BSC 虽然仍然要执行步骤 303 到步骤 310, 但是由于反向数据包都是经过排序的, 所以 BSC 不用对从 BTS 传送来的反向数据包中的字节做任何重排缓冲处理, 这样就减少了 BSC 的计算量, 基本上不用 BSC 进行计算。又由于 BTS 只需要将数据包本身传送给 BSC, 这样就可以进一步提高 BTS 和 BSC 接口的利用效率。当然, 如果基于其他考虑, 也完全可以将对反向数据包进行排序的工作放到 BSC 中进行。在这种情况下, 步骤 407、步骤 408 和步骤 409 中所述将反向数据包发送到 BSC 则应该为将反向数据包从包

缓存队列中发送到后续处理模块。

以上所述仅为本发明的较佳实施例而已，并非用于限定本发明的保护范围。凡在本发明的精神和原则之内，所作的任何修改、等同替换、改进等，均应包含在本发明的保护范围之内。

前向反馈

Ack	Nak	Ack	Ack	Nak	Ack	Nak	Nak	Ack	Nak	Ack	Ack
-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----

反向数据包

A0	B0	C0	D0	B1	E0	F0	B2	G0	F1	B3	H0
----	----	----	----	----	----	----	----	----	----	----	----

交织号

0	1	2	0	1	2	0	1	2	0	1	2
---	---	---	---	---	---	---	---	---	---	---	---

图 1

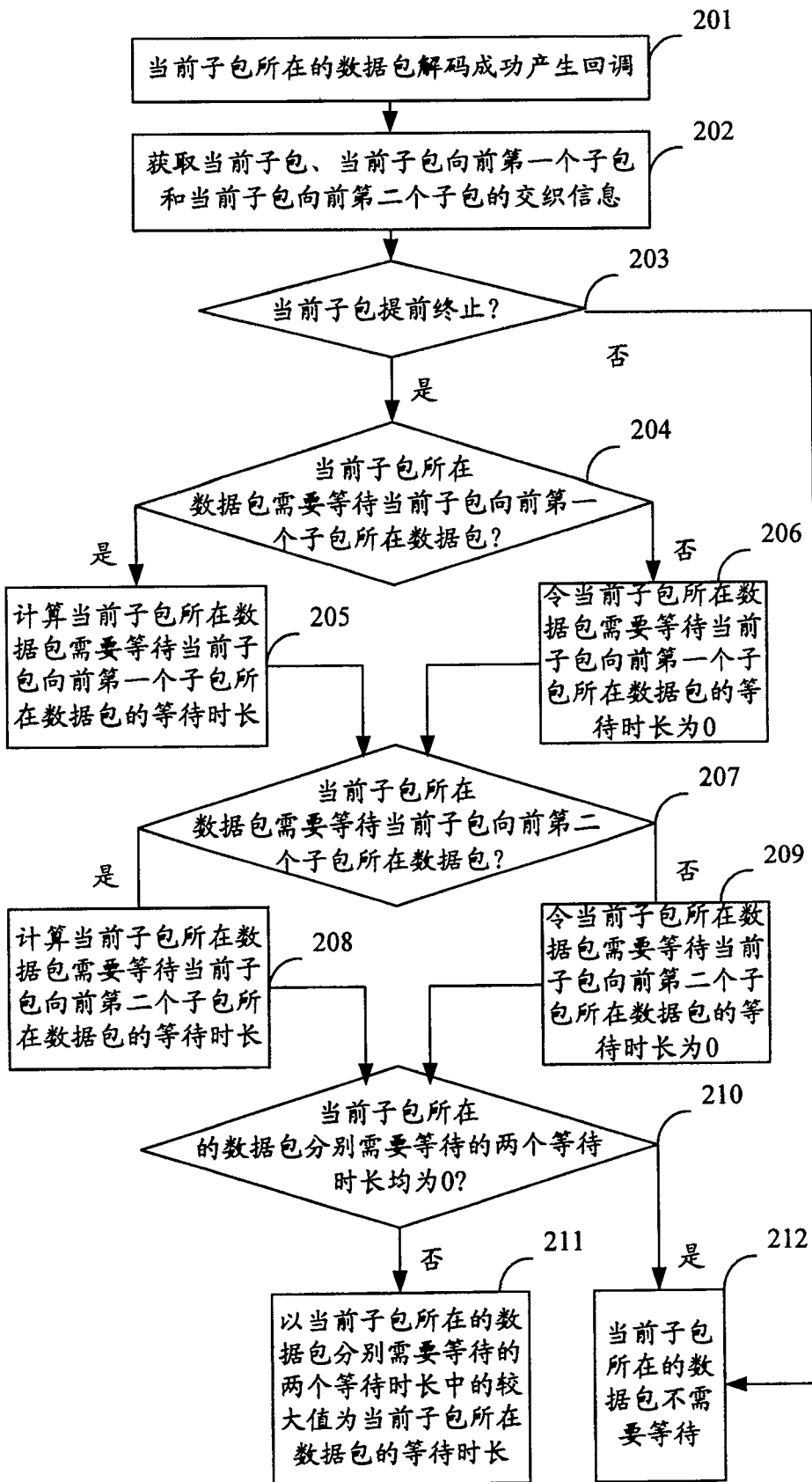


图 2

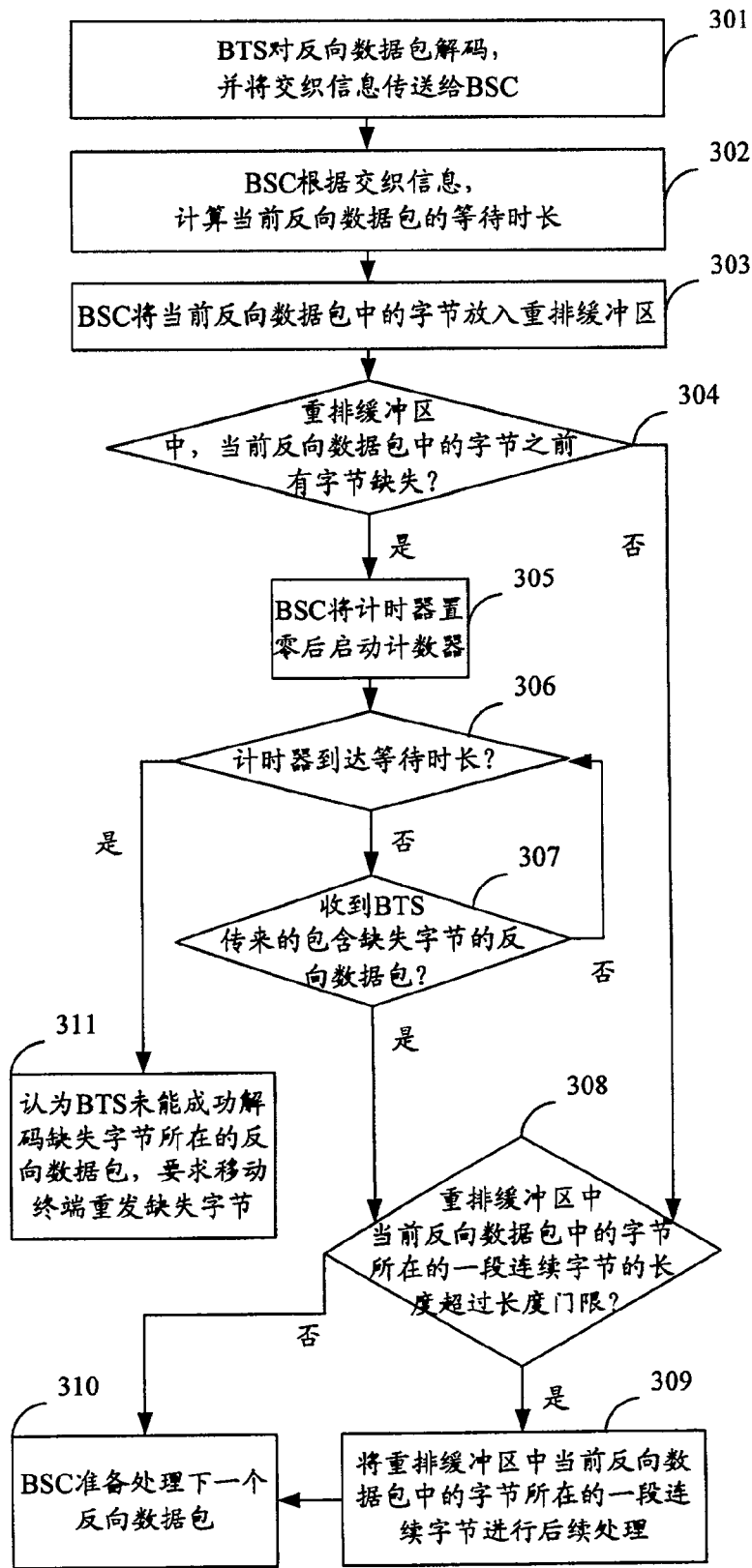


图 3

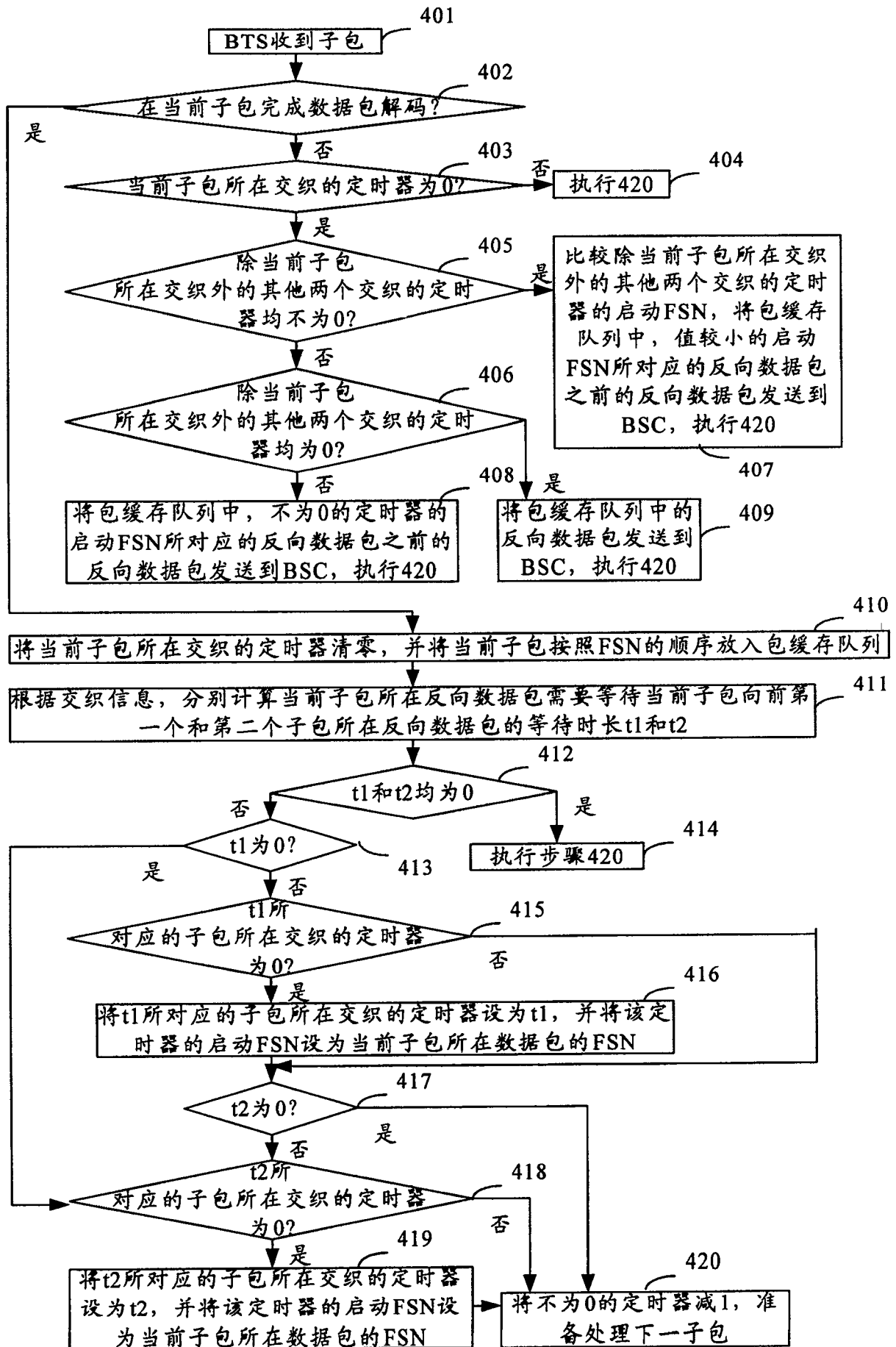


图 4