

上行传输的方法和装置

5 技术领域

本申请涉及通信领域，并且更具体地，涉及一种上行传输的方法和装置。

背景技术

10 目前，跨多时间单元传输块（transport block over multi-slot, TBoMS）速率匹配在每个时间单元执行一次，而且会根据上一次速率匹配比特选择的结束比特在循环缓存器中的位置实时计算下一次速率匹配比特选择的起始比特在循环缓存器中的位置。如果在一次数据传输中，网络设备通过物理下行控制信道（physical downlink control channel, PDCCH）调度终端设备上报上行控制信息（uplink control information, UCI），并且指示终端设备在物理上行共享信道（physical uplink shared channel, PUSCH）上承载 UCI，但是终端设备
15 没有检测该 PUCCH，则不会在 PUSCH 上承载 UCI，那么会导致网络设备和终端设备无法对齐其中一个时间单元速率匹配的比特选择的起始比特在循环缓存器中的位置，从而网络设备对于本次传输的编码比特序列中从该时间单元开始的编码比特序列都会译码错误。那么，在这样的情况下，网络设备和终端设备如何对其每次速率匹配中比特选择的起始比特在循环缓存器中的位置成为亟待解决的问题。

20

发明内容

本申请提供一种上行传输的方法和装置，以提高终端设备上行传输的译码成功率。

25 第一方面，提供了一种上行传输的方法，包括：终端设备接收来自网络设备的第一指示信息，该第一指示信息用于指示该终端设备在多个时间单元上发送第一传输块；该终端设备接收来自该网络设备的第二指示信息，该第二指示信息用于指示该终端设备在第一时间单元上发送上行控制信息 UCI，该第一时间单元为该多个时间单元中的一个；该终端设备根据该第一时间单元对应的参数 G 确定第一编码比特序列的起始比特在循环缓存中的位置，该第一编码比特序列为第二时间单元内经过速率匹配输出的编码比特序列，该参数 G 为一个时间单元内可用于传输该第一传输块和该 UCI 的编码比特总数， G 为正整数，该
30 第二时间单元为该第一时间单元在该多个时间单元中的下一个时间单元；该终端设备在该多个时间单元上向该网络设备发送该 UCI 和该第一传输块。

上述方案，网络设备指示终端设备在第一时间单元上承载 UCI 的指示信息，在终端设备检测到该指示的情况下，网络设备和终端设备在确定第二时间单元（第一时间单元的下一个时间单元）内经过速率匹配输出的编码比特序列的起始比特在循环缓存中的位置时使用的参数 G 都不扣除 UCI 的编码比特总数，可以使得从第二时间单元开始网络设备和终端设备可以对齐每次速率匹配中比特选择的起始比特在循环缓存器中的位置，从而提高译码成功率。

结合第一方面，在第一方面的某些实现方式中，该终端设备根据该第一时间单元对应的参数 G 确定第一编码比特序列的起始比特在循环缓存中的位置，包括：该终端设备根据该第一时间单元对应的参数 G 的 a 倍确定该第一编码比特序列的起始比特在循环缓存中的
40

位置, $0 < a \leq 1$ 。

结合第一方面, 在第一方面的某些实现方式中, 该方法还包括: 该终端设备根据该第一时间单元对应参数 G 确定第二编码比特序列的长度, 该第二编码比特序列为该第一时间单元内经过速率匹配输出的编码比特序列。

- 5 上述方案, 在终端设备接收到第二指示信息的情况下, 网络设备和终端设备可以使用与终端设备未接收到第二指示信息时相同的方式确定第一编码比特序列的起始比特在循环缓存中的位置和第二编码比特序列的长度, 扩展了本申请提供的上行传输的方法的应用范围。

- 10 结合第一方面, 在第一方面的某些实现方式中, a 等于以下任意一个数: $0.1x$, $0 < x < 10$, 或者, $0.2x$, $0 < x < 5$, 或者, $0.01x$, $0 < x < 100$, 其中, x 为整数。

结合第一方面, 在第一方面的某些实现方式中, 该终端设备根据该第一时间单元对应的参数 G 确定该第一编码比特序列的起始比特在循环缓存中的位置, 包括: 该终端设备根据该第一时间单元对应的参数 G 和 Z_c 确定该第一编码比特序列的起始比特在循环缓存中的位置, Z_c 为低密度奇偶校验 LDPC 因子。

- 15 上述方案, 终端侧在进行资源映射时, 减少了查找映射至上一个时间单元的最后一个比特在第一比特序列中的位置的步骤, 在根据 G 确定第一编码比特序列的起始比特在循环缓存中的位置的基础上结合 Z_c 确定该位置, 简化了映射过程中的计算, 节省了开销。

- 20 结合第一方面, 在第一方面的某些实现方式中, 该方法还包括: 该终端设备将第二编码比特序列映射至该第一时间单元上的时频资源, 该第二编码比特序列为该第一时间单元内经过速率匹配输出的编码比特序列, 该第二编码比特序列的比特数等于该第一时间单元对应的参数 G ; 该终端设备将该 UCI 映射至该第一时间单元上的部分时频资源, 以替换承载于该部分时频资源的该第一编码比特序列中的部分比特序列。

- 25 上述方案, 通过打孔的方式将 UCI 复用在该第一时间单元上, 可以使得第一时间单元上除 UCI 占用的时频资源外的时频资源上承载的包含于第一传输块的比特序列不改变, 从而可以提高网络设备译码的成功率和准确性。

结合第一方面, 在第一方面的某些实现方式中, 该第一指示信息还用于指示该网络设备能够在该多个时间单元上发送一个传输块, 该第一传输块为一个传输块, 其中, 该第一指示信息包括使能跨多时隙传输块传输字段, 或者, 该第一指示信息包括跨多时隙传输块映射类型参数。

- 30 上述方案, 通过多种形式向终端设备指示网络设备支持跨多时隙传输功能, 丰富信令设计, 提高上行传输的效率。

- 35 第二方面, 提供了一种上行传输的方法, 包括: 网络设备向终端设备发送第一指示信息, 该第一指示信息用于指示该终端设备在多个时间单元上发送第一传输块; 该网络设备向该终端设备发送第二指示信息, 该第二指示信息用于指示该终端设备在第一时间单元上发送上行控制信息 UCI, 该第一时间单元为该多个时间单元中的一个; 该网络设备接收来自该终端设备的该第一传输块; 该网络设备根据该第一时间单元对应的参数 G 确定第一编码比特序列的起始比特在循环缓存中的位置, 该第一编码比特序列为第二时间单元内经过速率匹配输出的编码比特序列, 该参数 G 为一个时间单元内能够用于传输该第一传输块和该 UCI 的编码比特总数, G 为正整数, 该第二时间单元为该第一时间单元在该多个时间单元中的下一个时间单元。
- 40

上述方案,网络设备指示终端设备在第一时间单元上承载 UCI 的指示信息,对于终端设备检测到或没有检测到该指示的情况下,网络设备和终端设备在确定第二时间单元(第一时间单元的下一个时间单元)内经过速率匹配输出的编码比特序列的起始比特在循环缓存中的位置时使用的参数 G 都不扣除 UCI 的编码比特总数,可以使得从第二时间单元开始网络设备和终端设备可以对齐每次速率匹配中比特选择的起始比特在循环缓存器中的位置,从而提高译码成功率。

结合第二方面,在第一方面的某些实现方式中,该网络设备根据该第一时间单元对应的参数 G 确定第一编码比特序列的起始比特在循环缓存中的位置,包括:该网络设备根据该第一时间单元对应的参数 G 的 a 倍确定该第一编码比特序列的起始比特在循环缓存中的位置, $0 < a \leq 1$ 。

上述方案,通过 G 乘上大于零且小于一的 a ,使得网络设备计算第一编码比特序列的起始比特在循环缓存中的位置可以提前,从而缩短第二编码比特序列的结束比特与第一编码比特序列的起始比特之间的间隙,提高信道质量。

结合第二方面,在第一方面的某些实现方式中,该方法还包括:该网络设备根据该第一时间单元对应参数 G 确定第二编码比特序列的长度,该第二编码比特序列为该第一时间单元内经过速率匹配输出的编码比特序列。

上述方案,进一步地,仍然根据 G 计算第二编码比特序列的长度,由于 G 大于 aG ,可以进一步确保消除上述间隙。

结合第二方面,在第一方面的某些实现方式中, a 等于以下任意一个数: $0.1x$, $0 < x < 10$, 或者, $0.2x$, $0 < x < 5$, 或者, $0.01x$, $0 < x < 100$, 其中, x 为整数。

结合第二方面,在第一方面的某些实现方式中,该网络设备根据该第一时间单元对应的参数 G 确定该第一编码比特序列的起始比特在循环缓存中的位置,包括:该网络设备根据该第一时间单元对应的参数 G 和 Z_c 确定该第一编码比特序列的起始比特在循环缓存中的位置, Z_c 为低密度奇偶校验 LDPC 因子。

上述方案,终端侧在进行资源映射时,减少了查找映射至上一个时间单元的最后一个比特在第一比特序列中的位置的步骤,在根据 G 确定第一编码比特序列的起始比特在循环缓存中的位置的基础上结合 Z_c 确定该位置,简化了映射过程中的计算,节省了开销。

结合第二方面,在第一方面的某些实现方式中,该第一指示信息还用于指示该网络设备能够在该多个时间单元上发送一个传输块,该第一传输块为一个传输块,其中,该第一指示信息包括使能跨多时隙传输块传输字段,或者,该第一指示信息包括跨多时隙传输块映射类型参数。

上述方案,通过多种形式向终端设备指示网络设备支持跨多时隙传输功能,丰富信令设计,提高上行传输的效率。

第三方面,提供了一种上行传输的方法,其特征在于,包括:终端设备接收来自网络设备的第一指示信息,该第一指示信息用于指示该终端设备在多个时间单元上发送一个传输块;该终端设备确定第一时间单元内的第一编码比特序列被取消传输,该第一时间单元为该多个时间单元中的一个时间单元,该第一编码比特序列为该第一时间单元内经过速率匹配输出的编码比特序列;该终端设备在第二时间单元上向该网络设备发送该第一编码比特序列,该第二时间单元为该多个时间单元中该第一时间单元以外的时间单元。

结合第三方面,在第三方面的某些实现方式中,该终端设备在该第二时间单元的下一

个时间单元上发送的比特序列的起始比特为该第一编码比特序列的结束比特的下一个比特。

结合第三方面，在第三方面的某些实现方式中，该方法还包括：该终端设备在第三时间单元上向网络设备发送第二编码比特序列，该第二编码比特序列该第三时间单元内经过速率匹配输出的编码比特序列，该第二编码比特序列的起始比特为该多个时间单元中的最后一个时间单元承载的编码比特序列的结束比特的下一个比特，该第三时间单元为该第一时域资源的下一个用于承载上行传输块的时间单元。

第四方面，提供了一种上行传输的装置，包括：收发模块，用于接收来自网络设备的第一指示信息，该第一指示信息用于指示该终端设备在多个时间单元上发送第一传输块；该收发模块，还用于接收来自该网络设备的第二指示信息，该第二指示信息用于指示该终端设备在第一时间单元上发送上行控制信息 UCI，该第一时间单元为该多个时间单元中的一个；处理模块，用于根据该第一时间单元对应的参数 G 确定第一编码比特序列的起始比特在循环缓存中的位置，该第一编码比特序列为第二时间单元内经过速率匹配输出的编码比特序列，该参数 G 为一个时间单元内可用于传输该第一传输块和该 UCI 的编码比特总数， G 为正整数，该第二时间单元为该第一时间单元在该多个时间单元中的下一个时间单元；该收发模块，还用于在该多个时间单元上向该网络设备发送该 UCI 和该第一传输块。

上述方案，网络设备指示终端设备在第一时间单元上承载 UCI 的指示信息，在终端设备检测到该指示的情况下，网络设备和终端设备在确定第二时间单元（第一时间单元的下一个时间单元）内经过速率匹配输出的编码比特序列的起始比特在循环缓存中的位置时使用的参数 G 都不扣除 UCI 的编码比特总数，可以使得从第二时间单元开始网络设备和终端设备可以对齐每次速率匹配中比特选择的起始比特在循环缓存器中的位置，从而提高译码成功率。

结合第四方面，在第四方面的某些实现方式中，该处理模块，具体用于根据该第一时间单元对应的参数 G 的 a 倍确定该第一编码比特序列的起始比特在循环缓存中的位置， $0 < a \leq 1$ 。

结合第四方面，在第四方面的某些实现方式中，该处理模块，还用于根据该第一时间单元对应参数 G 确定第二编码比特序列的长度，该第二编码比特序列为该第一时间单元内经过速率匹配输出的编码比特序列。

结合第四方面，在第四方面的某些实现方式中， a 等于以下任意一个数： $0.1x$ ， $0 < x < 10$ ，或者， $0.2x$ ， $0 < x < 5$ ，或者， $0.01x$ ， $0 < x < 100$ ，其中， x 为整数。

结合第四方面，在第四方面的某些实现方式中，

该处理模块，具体用于根据该第一时间单元对应的参数 G 和 Z_c 确定该第一编码比特序列的起始比特在循环缓存中的位置， Z_c 为低密度奇偶校验 LDPC 因子。

结合第四方面，在第四方面的某些实现方式中，该处理模块，还用于将第二编码比特序列映射至该第一时间单元上的时频资源，该第二比特序列为该第一时间单元内经过速率匹配输出的编码比特序列，该第二编码比特序列的比特数等于该第一时间单元对应的参数 G ；

该处理模块，还用于将该 UCI 映射至该第一时间单元上的部分时频资源，以替换承载于该部分时频资源的该第一编码比特序列中的部分比特序列。

结合第四方面，在第四方面的某些实现方式中，该第一指示信息还用于指示该网络设

备能够在该多个时间单元上发送一个传输块，该第一传输块为一个传输块，

其中，该第一指示信息包括使能跨多时隙传输块传输字段，或者，该第一指示信息包括跨多时隙传输块映射类型参数。

第五方面，提供了一种上行传输的装置，包括：收发模块，用于向终端设备发送第一指示信息，该第一指示信息用于指示该终端设备在多个时间单元上发送第一传输块；该收发模块，还用于向该终端设备发送第二指示信息，该第二指示信息用于指示该终端设备在第一时间单元上发送上行控制信息 UCI，该第一时间单元为该多个时间单元中的一个；该收发模块，还用于接收来自该终端设备的该第一传输块；处理模块，用于根据该第一时间单元对应的参数 G 确定第一编码比特序列的起始比特在循环缓存中的位置，该第一编码比特序列为第二时间单元内经过速率匹配输出的编码比特序列，该参数 G 为一个时间单元内能够用于传输该第一传输块和该 UCI 的编码比特总数， G 为正整数，该第二时间单元为该第一时间单元在该多个时间单元中的下一个时间单元。

上述方案，网络设备指示终端设备在第一时间单元上承载 UCI 的指示信息，在终端设备检测或没有检测到该指示的情况下，网络设备和终端设备在确定第二时间单元（第一时间单元的下一个时间单元）内经过速率匹配输出的编码比特序列的起始比特在循环缓存中的位置时使用的参数 G 都不扣除 UCI 的编码比特总数，可以使得从第二时间单元开始网络设备和终端设备可以对齐每次速率匹配中比特选择的起始比特在循环缓存器中的位置，从而提高译码成功率。

结合第五方面，在第五方面的某些实现方式中，该处理模块，具体用于根据该第一时间单元对应的参数 G 的 a 倍确定该第一编码比特序列的起始比特在循环缓存中的位置， $0 < a \leq 1$ 。

结合第五方面，在第五方面的某些实现方式中，该处理模块，还用于根据该第一时间单元对应参数 G 确定第二编码比特序列的长度，该第二编码比特序列为该第一时间单元内经过速率匹配输出的编码比特序列。

结合第五方面，在第五方面的某些实现方式中， a 等于以下任意一个数： $0.1x$ ， $0 < x < 10$ ，或者， $0.2x$ ， $0 < x < 5$ ，或者， $0.01x$ ， $0 < x < 100$ ，其中， x 为整数。

结合第五方面，在第五方面的某些实现方式中，

该处理模块，具体用于根据该第一时间单元对应的参数 G 和 Z_c 确定该第一编码比特序列的起始比特在循环缓存中的位置， Z_c 为低密度奇偶校验 LDPC 因子。

结合第五方面，在第五方面的某些实现方式中，该第一指示信息还用于指示该网络设备能够在该多个时间单元上发送一个传输块，该第一传输块为一个传输块，

其中，该第一指示信息包括使能跨多时隙传输块传输字段，或者，该第一指示信息包括跨多时隙传输块映射类型参数。

第七方面，提供了一种上行传输的方法，该方法包括：

终端设备接收来自网络设备的下行控制信息和第一信息，该下行控制信息用于指示终端设备在第一时域资源上传输非周期的信道状态信息 A-CSI，所述第一信息指示在第二时域资源上传输物理上行共享信道 PUSCH，该第一时域资源与该第二时域资源重叠；该终端设备取消传输该 A-CSI，或者，该终端设备取消该第一时域资源与该第二时域资源重叠的时域资源上的 PUSCH 传输。

结合第七方面，在第七方面的某些实现方式中，该第一时域资源与该第二时域资源包

含于同一个时间单元。

第八方面，提供了一种上行传输的方法，该方法包括：

- 5 终端设备接收来自网络设备的第一指示信息，该第一指示信息用于指示该终端设备在多个时间单元上发送第一传输块；该终端设备接收来自该网络设备的第二指示信息，该第二指示信息用于指示该终端设备在第一时间单元上发送 HARQ-ACK 和第一控制信息，该第一控制信息包括信道状态信息 CSI 和/或配置授权上行控制信息 CG-UCI，该第一时间单元为该多个时间单元中的一个；该终端设备根据该第一时间单元对应的参数 G 确定第一编码比特序列的起始比特在循环缓存中的位置，该第一编码比特序列为第二时间单元内经过速率匹配输出的编码比特序列，该参数 G 为一个时间单元内可用于传输该第一传输块和该 HARQ-ACK 的编码比特总数，G 为正整数，该第二时间单元为该第一时间单元在该多个时间单元中的下一个时间单元；该终端设备在该多个时间单元上向该网络设备发送该 HARQ-ACK、该第一信息和该第一传输块。

第九方面，提供了一种上行传输的方法，该方法包括：

- 15 终端设备接收来自网络设备的第一指示信息，该第一指示信息用于指示该终端设备在多个时间单元上发送第一传输块；

该终端设备接收来自该网络设备的第二指示信息，该第二指示信息用于指示该终端设备在第一时间单元上发送 HARQ-ACK，该第一时间单元为该多个时间单元中的一个；

该 HARQ-ACK 占用的比特数大于 2 比特时，该终端设备根据打孔的方式，将该 HARQ-ACK 复用到该第一传输块；

- 20 该终端设备在该多个时间单元上向该网络设备发送复用有该 HARQ-ACK 的该第一传输块。

第十方面，提供了一种上行传输的方法，该方法包括：

网络设备向终端设备发送第一信息，该第一信息用于指示该终端设备使用第一传输方式发送一个传输块；

- 25 在该网络设备未接收到或未正确接收到该传输块的情况下，该网络设备向该终端设备发送第二信息，该第二信息用于指示该终端设备使用第二传输方式发送该传输块，

其中，该第一传输方式为基于单码块的物理上行共享信道 PUSCH 或者基于单码块的重复类型 A 的 PUSCH 重复，该第二传输方式为多时隙传输块或者多时隙传输块重复，

- 30 或者，该第一传输方式为基于多码块的物理上行共享信道 PUSCH 或者基于多码块的重复类型 A 的 PUSCH 重复，该第一传输方式与该第二传输方式相同。

结合第十方面，在第十方面的某些实现方式中，该方法还包括：

该网络设备根据该第一传输方式确定该第二传输方式。

第十一方面，提供了一种上行传输的装置，该装置包括：

- 35 收发模块，用于接收来自网络设备的下行控制信息和第一信息，该下行控制信息用于指示终端设备在第一时间域资源上传输非周期的信道状态信息 A-CSI，该第一信息用于指示在第二时域资源上传输物理上行共享信道 PUSCH，该第一时间域资源与该第二时域资源重叠；

处理模块，用于取消传输该 A-CSI，或者，该终端设备取消该第一时间域资源与该第二时域资源重叠的时域资源上的 PUSCH 传输。

- 40 第十二方面，提供了一种上行传输的装置，该装置包括：

收发模块,用于接收来自网络设备的第一指示信息,该第一指示信息用于指示该终端设备在多个时间单元上发送第一传输块;

- 5 该收发模块,还用于接收来自该网络设备的第二指示信息,该第二指示信息用于指示该终端设备在第一时间单元上发送 HARQ-ACK 和第一控制信息,该第一控制信息包括信道状态信息 CSI 和/或配置授权上行控制信息 CG-UCI,该第一时间单元为该多个时间单元中的一个;

- 10 处理模块,用于根据该第一时间单元对应的参数 G 确定第一编码比特序列的起始比特在循环缓存中的位置,该第一编码比特序列为第二时间单元内经过速率匹配输出的编码比特序列,该参数 G 为一个时间单元内可用于传输该第一传输块和该 HARQ-ACK 的编码比特总数, G 为正整数,该第二时间单元为该第一时间单元在该多个时间单元中的下一个时间单元;

该收发模块,还用于在该多个时间单元上向该网络设备发送该 HARQ-ACK、该第一信息和该第一传输块。

第十三方面,提供了一种上行传输的装置,该装置包括:

- 15 收发模块,用于接收来自网络设备的第一指示信息,该第一指示信息用于指示该终端设备在多个时间单元上发送第一传输块;

该收发模块,还用于接收来自该网络设备的第二指示信息,该第二指示信息用于指示该终端设备在第一时间单元上发送 HARQ-ACK,该第一时间单元为该多个时间单元中的一个;

- 20 所述 HARQ-ACK 占用的比特数大于 2 比特时,处理模块,用于根据打孔的方式,将所述 HARQ-ACK 复用到所述第一传输块;

该收发模块,还用于在所述多个时间单元上向所述网络设备发送复用有所述 HARQ-ACK 的所述第一传输块。

第十四方面,提供了一种上行传输的装置,该装置包括:

- 25 收发模块,用于向终端设备发送第一信息,该第一信息用于指示该终端设备使用第一传输方式发送一个传输块;

在该收发模块未接收到或未正确接收到该传输块的情况下,该收发模块,还用于向该终端设备发送第二信息,该第二信息用于指示该终端设备使用第二传输方式发送该传输块,

- 30 其中,该第一传输方式为基于单码块的物理上行共享信道 PUSCH 或者基于单码块的重复类型 A 的 PUSCH 重复,该第二传输方式为多时隙传输块或者多时隙传输块重复,

或者,该第一传输方式为基于多码块的物理上行共享信道 PUSCH 或者基于多码块的重复类型 A 的 PUSCH 重复,该第一传输方式与该第二传输方式相同。

第十五方面,提供了一种上行传输的方法,包括:

- 35 终端设备接收来自网络设备的第一指示信息,所述第一指示信息用于指示所述终端设备在多个时间单元上发送第一传输块;

所述终端设备接收来自所述网络设备的第二指示信息,所述第二指示信息用于指示所述终端设备在第一时间单元上发送上行控制信息 UCI,所述第一时间单元为所述多个时间单元中的一个;

- 40 所述终端设备根据所述第一时间单元对应的参数 G 确定第一编码比特序列的起始比特在循环缓存中的位置,所述第一编码比特序列为第二时间单元内经过速率匹配输出的编

码比特序列, 所述参数 G 为一个时间单元内可用于传输所述第一传输块和所述 UCI 的编码比特总数, G 为正整数, 所述第二时间单元为所述多个时间单元中的一个;

所述终端设备在所述多个时间单元上向所述网络设备发送所述 UCI 和所述第一传输块。

- 5 上述方案中, 网络设备指示终端设备在第一时间单元上承载 UCI 的指示信息, 在终端设备检测到该指示的情况下, 网络设备和终端设备在确定第二时间单元内经过速率匹配输出的编码比特序列的起始比特在循环缓存中的位置时使用的参数 G 都不扣除 UCI 的编码比特总数, 可以使得从第二时间单元开始网络设备和终端设备可以对齐每次速率匹配中比特选择的起始比特在循环缓存器中的位置, 从而提高译码成功率。

- 10 一种实现方式中, 所述多个时间单元对应的参数 G 相等。

一种实现方式中, 所述终端设备根据所述第一时间单元对应的参数 G 确定第一编码比特序列的起始比特在循环缓存中的位置, 包括:

根据所述第一时间单元对应的参数 G 和比特选择的真实周期 N'_{cb} , 确定所述所述第一编码比特序列的起始比特在循环缓存中的位置。

- 15 一种实现方式中, 根据所述第一时间单元对应的参数 G 和比特选择的真实周期 N'_{cb} , 确定所述第一编码比特序列的起始比特在循环缓存中的位置, 包括:

根据所述第一时间单元对应的参数 G 和比特选择的真实周期 N'_{cb} , 确定第一起始比特索引 $k'_{r,n}$;

- 20 根据所述第一起始比特索引 $k'_{r,n}$ 确定所述第一编码比特序列的起始比特在循环缓存中的位置 $k_{r,n}$ 。

其中, n 表示所述第二时间单元在所述多个时间单元中的索引, 所述索引是所述多个时间单元按照时间顺序从 0 开始计数得到的。

一种实现方式中, 所述第一起始索引 $k'_{r,n}$ 满足:

$$k'_{r,n} = \text{mod}(k'_{r,0} + n \times G, N'_{cb});$$

- 25 其中, 所述 $k'_{r,0}$ 表示所述多个时间单元中第一个时间单元的第一起始比特索引, 所述 N'_{cb} 表示比特选择的真实周期, n 表示所述第二时间单元在所述多个时间单元中的索引, 所述索引是所述多个时间单元按照时间顺序从 0 开始计数得到的。

一种实现方式中, 所述根据所述 $k'_{r,n}$ 确定第一编码比特序列的起始比特在循环缓存中的位置 $k_{r,n}$ 包括:

- 30 若 $0 \leq k'_{r,n} < K'_{NULL} - 2Z_c$, 则 $k_{r,n} = k'_{r,n}$;

若 $K'_{NULL} - 2Z_c \leq k'_{r,n} < N'_{cb}$, 则 $k_{r,n} = k'_{r,n} + (K_{NULL} - K'_{NULL})$;

其中, $K'_{NULL} - 2Z_c$ 表示循环缓存中第一个填充比特的位置, $K_{NULL} - K'_{NULL}$ 表示循环缓存中填充比特的个数。

一种实现方式中, 所述比特选择的真实周期 N'_{cb} 满足:

- 35 $N'_{cb} = N_{cb} - (K_{NULL} - K'_{NULL})$, 其中, $K_{NULL} - K'_{NULL}$ 表示循环缓存中填充比特的个数, N_{cb} 表示循环缓存的大小。

一种实现方式中, 所述多个时间单元中第一个时间单元的第一起始比特索引 $k'_{r,n}$ 满足

若 $0 \leq k_0 < K'_{NULL} - 2Z_c$, 则 $k'_{r,0} = k_0$; 其中, $K'_{NULL} - 2Z_c$ 表示循环缓存中第一个填充比特的索引;

- 40 若 $K'_{NULL} - 2Z_c \leq k_0 < K_{NULL} - 2Z_c$, 则 $k'_{r,0} = K'_{NULL} - 2Z_c$; 其中, $K_{NULL} - 2Z_c$ 表示

循环缓存中第一个校验比特的索引；

若 $K_{NULL} - 2Z_c \leq k_0 < N_{cb}$ ，则 $k'_{r,0} = k_0 - (K_{NULL} - K'_{NULL})$ ，其中， $K_{NULL} - K'_{NULL}$ 表示循环缓存中填充比特的个数；

其中， k_0 表示所述多个时间单元中第一个时间单元对应的编码比特的起始比特在循环缓存中的位置，所述 k_0 由冗余版本 RV 和基图 BG 共同确定。

上述方案中，保证所述多个时隙中的每个时隙通过比特选择得到的比特序列是连续的，从而保证译码性能。

第十六方面，提供了一种通信装置，包括：处理器和存储器；该存储器，用于存储计算机程序；该处理器，用于执行该存储器中存储的计算机程序，以使得该通信装置执行第一方面至第三方面、第七方面至第十方面及第十五方面中任一方面所述的通信方法。

第十六方面，提供了一种计算机可读存储介质，该计算机可读存储介质上存储有计算机程序，当该计算机程序在计算机上运行时，使得该计算机执行如第一方面至第三方面、第七方面至第十方面及第十五方面中任一方面所述的通信方法。

第十七方面，提供了一种芯片系统，包括：处理器，用于从存储器中调用并运行计算机程序，使得安装有该芯片系统地通信设备执行如第一方面至第三方面、第七方面至第十方面及第十五方面中任一方面所述的通信方法。

附图说明

- 图 1 是适用于本申请实施例的无线通信系统 100 的一示意图。
- 图 2 示出了多时隙传输块的示意图。
- 图 3 示出了信号处理的流程示意图。
- 图 4 示出了传输时机的示意图。
- 图 5 示出了基于冗余版本循环的 TBoMS 资源映射的示意图。
- 图 6 示出了 TBoMS 中每个时隙执行一次速率匹配的示意图。
- 图 7 示出了网络设备和终端设备进行每次速率匹配比特选择的起始比特在循环缓存中的位置。
- 图 8 示出了本申请提供的上行传输的方法 100 的示意性交互图。
- 图 9 示出了本申请提供的上行传输的方法 200 的示意性交互图。
- 图 10 示出了两种可能的上下行时隙配比的示例。
- 图 11 示出了确定速率匹配比特选择的起始位置的一例的示意图。
- 图 12 示出了循环缓存器中的比特序列按顺序依次由 PUSCH 承载的示意图。
- 图 13 示出了循环缓存器中的比特序列按顺序依次由 PUSCH 承载的示意图。
- 图 14 示出了循环缓存器中的比特序列按顺序依次由 PUSCH 承载的示意图。
- 图 15 示出了跨多时隙传输的比特序列按顺序依次由 PUSCH 承载的示意图。
- 图 16 示出了 PUSCH 重复类型 A 的 RV 循环配置调度。
- 图 17 示出了 PUSCH 重复类型 A 的 RV 循环配置调度。
- 图 18 示出了 PUSCH 重复类型 A 的 RV 循环配置调度。
- 图 19 示出了 PUSCH 重复类型 A 的 RV 循环配置调度。
- 图 20 示出了上行传输的方法 500 的一例的示意图。
- 图 21 示出了上行传输的方法 500 的又一例的示意图。

图 22 示出了上行传输的方法 500 的再一例的示意图。

图 23 示出了 TBoMS 传输的一例的示意图。

图 24 是本申请实施例提供的用于上行传输的通信装置的示意性框图。

图 25 为本申请实施例提供的上行传输的装置 20 的示意图。

5 图 26 示出了 k_0 在填充比特的示意性框图。

图 27 中的 (a)、(b)、(c)、(d) 示出了 k_0' , $\bar{k}_0$ 和填充比特在循环缓存器中的位置的四种不同的情况的示意图。

图 28 中的 (a) 和 (b) 示出了长度分别为 N_{cb} 和 N'_{cb} 的循环缓存器。

10 具体实施方式

本申请实施例的技术方案可以应用于各种通信系统, 例如: 第五代 (5th generation, 5G) 系统或新无线 (new radio, NR)、长期演进 (long term evolution, LTE) 系统、LTE 频分双工 (frequency division duplex, FDD) 系统、LTE 时分双工 (time division duplex, TDD)、通用移动通信系统 (universal mobile telecommunication system, UMTS) 等。此外, 本申请实施例的技术方案还可以应用于侧链路通信。例如, 本申请实施例的技术方案还可以应用于: 设备到设备 (device to device, D2D) 通信, 机器到机器 (machine to machine, M2M) 通信, 机器类型通信 (machine type communication, MTC), 以及车联网系统中的通信。

为便于理解本申请实施例, 首先结合图 1 说明适用于本申请实施例的通信系统。

20 图 1 是适用于本申请实施例的无线通信系统 100 的一示意图。如 1 图所示, 该无线通信系统 100 可以包括至少一个网络设备,

如图 1 所示的网络设备 111, 该无线通信系统 100 还可以包括至少一个终端设备, 例如如图 1 所示的终端设备 121。网络设备和终端设备均可配置多个天线, 网络设备与终端设备可使用多天线技术通信。

25 其中, 网络设备和终端设备通信时, 网络设备可以管理一个或多个小区, 一个小区中可以有整数个终端设备。可选地, 网络设备 111 和终端设备 121 组成一个单小区通信系统, 不失一般性, 将小区记为小区 #1。网络设备 111 可以是小区 #1 中的网络设备, 或者说, 网络设备 111 可以为小区 #1 中的终端设备 (例如终端设备 121) 服务。

需要说明的是, 小区可以理解为网络设备的无线信号覆盖范围内的区域。

30 本申请实施例中提到的发送端设备可以为终端设备, 接收端设备可以为网络设备。例如, 发送端设备为终端设备 121, 接收端设备为网络设备 111。

应理解, 上述图 1 仅是示例性说明, 本申请并未限定于此。例如, 本申请实施例还可以应用于需要重复发送数据 (或者说数据块) 的任何通信场景。

35 还应理解, 该无线通信系统中的网络设备可以是任意一种具有无线收发功能的设备。该设备包括但不限于: 演进型节点 B (evolved Node B, eNB)、无线网络控制器 (radio network controller, RNC)、节点 B (node B, NB)、基站控制器 (base station controller, BSC)、基站收发台 (base transceiver station, BTS)、家庭基站 (例如, home evolved node B, 或 home node B, HNB)、基带单元 (baseband unit, BBU), 无线保真 (wireless fidelity, WIFI) 系统中的接入点 (access point, AP)、无线中继节点、无线回传节点、传输点 (transmission point, TP) 或者发送接收点 (transmission and reception point, TRP) 等,

还可以为 5G, 如, NR, 系统中的 gNB, 或, 传输点 (TRP 或 TP), 5G 系统中的基站的一个或一组 (包括多个天线面板) 天线面板, 或者, 还可以为构成 gNB 或传输点的网络节点, 如基带单元 (BBU), 或, 分布式单元 (distributed unit, DU) 等。

5 在一些部署中, gNB 可以包括集中式单元 (centralized unit, CU) 和 DU。gNB 还可以包括有源天线单元 (active antenna unit, 简称 AAU)。CU 实现 gNB 的部分功能, DU 实现 gNB 的部分功能。比如, CU 负责处理非实时协议和服务, 实现无线资源控制 (radio resource control, RRC), 分组数据汇聚层协议 (packet data convergence protocol, PDCP) 层的功能。DU 负责处理物理层协议和实时服务, 实现无线链路控制 (radio link control, RLC) 层、媒体接入控制 (media access control, MAC) 层和物理 (physical, PHY) 层的
10 功能。AAU 实现部分物理层处理功能、射频处理及有源天线的相关功能。由于 RRC 层的信息最终会变成 PHY 层的信息, 或者, 由 PHY 层的信息转变而来, 因而, 在这种架构下, 高层信令, 如 RRC 层信令, 也可以认为是由 DU 发送的, 或者, 由 DU+AAU 发送的。可以理解的是, 网络设备可以为包括 CU 节点、DU 节点、AAU 节点中一项或多项的设备。此外, 可以将 CU 划分为接入网 (radio access network, RAN) 中的网络设备, 也可以将
15 CU 划分为核心网 (core network, CN) 中的网络设备, 本申请对此不做限定。

还应理解, 该无线通信系统中的终端设备也可以称为用户设备 (user equipment, UE)、接入终端、用户单元、用户站、移动站、移动台、远方站、远程终端、移动设备、用户终端、终端、无线通信设备、用户代理或用户装置。本申请的实施例中的终端设备可以是手机 (mobile phone)、平板电脑 (Pad)、带无线收发功能的电脑、虚拟现实 (virtual reality, VR) 终端设备、增强现实 (augmented reality, AR) 终端设备、工业控制 (industrial control) 中的无线终端、无人驾驶 (self driving) 中的无线终端、远程医疗 (remote medical) 中的无线终端、智能电网 (smart grid) 中的无线终端、运输安全 (transportation safety) 中的无线终端、智慧城市 (smart city) 中的无线终端、智慧家庭 (smart home) 中的无线终端等等。本申请的实施例对应用场景不做限定。

25 为便于理解本申请实施例, 下面首先结合本申请中涉及的几个术语进行简单介绍。

1、时间单元

时间单元, 泛指时间的单位。示例性的, 所述时间单元可以包括但不限于为子帧 (subframe)、时隙 (slot)、符号、物理时隙、可用时隙、时隙的第一符号、物理时隙的第一符号、可用时隙的第一符号等。其中, 所述符号 (例如, 第一符号) 可以是时域符号
30 (例如, 正交频分复用 (orthogonal frequency division multiplexing, OFDM) 符号) 等。时间单元可以根据上下行时隙配比和 TDRA 表共同决定, 可用由 TDRA 表中的起始符号 S 和长度 L 指示。

其中, 时隙 (slot), 一种 slot 的格式可以为包含若干个正交频分复用 (orthogonal frequency division multiplexing, OFDM) 符号。例如, 一个 slot 的格式可以包括 14 个 OFDM 符号, 或者, 一种 slot 的格式可以为包含 12 个 OFDM 符号; 或者, 一种 slot 的格式为包含 7 个 OFDM 符号。一个 slot 中的 OFDM 符号可以全用于上行传输; 可以全用于下行传输; 也可以一部分用于下行传输, 一部分用于上行传输, 一部分灵活时域符号 (可以灵活的配置为用于上行或者下行传输)。应理解, 以上举例仅为示例性说明, 不应对本申请构成任何限定。出于系统前向兼容性考虑, slot 包含的 OFDM 符号的数目以及 slot 用于上行
40 传输和/或下行传输不限于以上示例。本申请中, 时域符号可以为 OFDM 符号, 即时域符

号可以替换为 OFDM 符号。

2、跨多时隙传输块 (transport block over multi-slot, TBoMS)

与 LTE 和长期演进高级 (long term evolution advanced, LTE-A) 无线通信系统相比, NR 无线通信系统部署的频段更高, 以获取更大的通信带宽。然而, 高频段会导致更大的
5 路径损耗和穿透损耗, 使得 NR 的覆盖性能远不如 LTE 和 LTE-A。

为了提升 NR 的覆盖性能, 提出跨多时隙传输块技术。示例性地, 如图 2 所示, 将 TB#1 至 TB#4 聚合成一个大的传输块 (transport block, TB)。该技术将每个时隙上的小数据包聚合成一个大数据包, 并在多个时隙上共同传输聚合后的数据包, 通过减少 TB 切分次数, 从而减少循环冗余码 (cyclic redundancy code, CRC) 的开销, 通过增加传输块
10 大小 (transport block size, TBS), 从而提高编码增益, 并通过降低频率资源, 提高功率谱密度, 提升覆盖性能。

3、类型 A (type A) 和类型 B (type B) 的重复发送

如前所述, 在一些场景下, 如一些深覆盖场景, 如小区边沿, 或者地下室等, 无线信号传播的路径损耗非常严重。为了改善上行传输性能, 一种增强覆盖性能的方法是重复发送数据块。例如, 终端设备重复发送 PUSCH, 网络设备对重复发送的数据块进行合并检测。通过该方式可以提升信道估计性能, 提升数据解调性能, 从而提升小区覆盖能力。
15

以当前的 NR 协议为例, 在当前的 NR 协议中支持对 PUSCH 最大 16 次的重复发送, 支持对 PUCCH 最大 8 次的重复发送。当前 NR 协议对 PUCCH 支持 type A 的重复发送, 对 PUSCH 支持 type A 和 type B 的重复发送。

20 (1) type A 的重复发送

type A 的重复发送, 指的是: N 次重复需要调度连续的 N 个 slot, 配置一次重复发送在一个 slot 中需要占用的时域符号的起始位置和总长度, N 个 slot 中, 满足一次重复发送占据的时域符号的起始位置和总长度与配置的起始位置和总长度相同的 slot, 可以实际用于一次重复发送。其中, N 为大于或等于 1 的整数。

25 (2) type B 的重复发送

type B 的重复发送, 指示是: N 次重复发送, 依据第 1 次重复发送的起始时域符号位置 S, 按照每次重复需要占据的时域符号数目 L, 在连续的多个时域符号上进行重复发送。即从调度的第一个 slot 的第 S 个时域符号开始, 后续的 N*L 个时域符号 (可能会到延伸到其他的 slot 上) 均用于 N 次重复发送。

30 应理解, 关于 type A 和 type B 的重复发送的具体描述可以参考现有的协议, 其对本申请实施例的保护范围不造成限定。

下文为方便描述, 将 type A 的重复发送方式记为重复 type A (repetition type A), 将 type B 的重复发送方式记为重复 type B (repetition type B)。

4、冗余版本 (redundancy version, RV)

35 信息比特串在通过物理天线发出去之前, 一般会经历一些信号处理过程, 如图 3 所示。

信道编码: 通过对信息比特串中引入冗余和校验比特, 使得信号在到达接收端之后, 接收端能够依据收到的多个比特 (包括信息比特和校验比特) 彼此之间的校验关系, 能够较好的恢复出信息比特串。对于数据信道而言, 目前 NR 可以支持低密度奇偶校验码 (low density parity check code, LDPC) 的信道编码。例如, 对于 100 比特 (bit) 的信息比特串,
40 通过 1/5 编码码率的 LDPC 编码, 变成了 500bit 的编码后比特串, 引入了 400bit 的冗余,

信息比特串和编码后比特串长度的比值等于编码码率 1/5。为区分，将编码后的比特串记为编码后比特串。

速率匹配 (rate matching, RM)：速率匹配对每个码块 (code block, CB) 分别进行，主要包括两部分：比特选择 (bit selection) 和比特交织 (bit interleaving)；其中，比特选择根据冗余版本 (redundancy version, RV) 从循环缓存器 (circular buffer) 中顺序读取一部分编码比特，比特交织对所述一部分编码比特进行行列交织。最后，将各个码块经过比特交织后的比特顺序级联，输出码字 (code word, CW)。信息比特串传输经过信道编码得到较长的编码后比特串之后，并不是直接将所有的编码后比特串都发出去。一般地，终端设备可以按照网络设备指示给终端配置的可用的资源元素 (resource element, RE) 个数及调制阶数来确定能够发送多少比特，进而从编码后比特串中进行选择 (当前协议规定了 4 个起点，近似均匀的分布在编码后比特串中，分别标记为 RV0, RV1, RV2, RV3)。

例如，当前 1 个资源块 (resource block, RB) 中的可用 RE 数目=12*12=144，采用正交相移键控 (quadrature phase shift keying, QPSK) 调制，则 1 个物理资源块 (physical resource block, PRB) 中能够承载 144*2=288bit。因此，需要从 500 比特的编码后比特串中选择出 288 比特，作为选择出的比特串，然后对该选择出的比特串进行调制和资源映射等处理。此时，对应的编码码率=信息比特串/选择出的比特串长度=100/288。

更具体地，速率匹配包括比特选择和比特交织。比特选择是指根据 RV 指定的起点从循环缓存中选择编码比特，选择的编码长度由被映射的时频资源大小决定的。比特交织是将选择出的比特进行行列交织，所以比特交织粒度与比特选择粒度一致，可以统称为速率匹配的粒度。

示例性的，如果被映射的时频资源是一个 TO 包括的一个或多个 OFDM 符号，那么选择的编码比特的长度就是一个 TO 上的时频资源 (不考虑码块分割)，也可以理解为，速率匹配的粒度是一个 TO。具体地，如果不考虑码块分割，选择的编码比特的长度就是被映射的时频资源大小，可以用公式表示为： $N_{TO} = N_{RE} \cdot Q_m \cdot N_L$ ，其中， N_{TO} 为一个传输时机的时频资源可以承载的比特数， Q_m 为调制阶数， N_L 为层数。

5、重复 RV 循环机制

下面以重复 type A 为例，介绍目前的重复 RV 循环机制。

假设调度物理上行共享信道 (physical uplink shared channel, PUSCH) 的下行控制信息 (downlink control information, DCI) 指示的 RV 为 rv_{id} 。对于基于重复类型 A，PUSCH 的第 n 个传输时机使用的 RV 如表 1 所示。其中，一个传输时机定义为一个时隙。应理解，RV 更新粒度为一个传输时机，即每一个传输时机都要更新使用的 RV，更新顺序由 RV 序列决定，RV 序列默认为 {0, 2, 3, 1}，即 RV 按照 {0, 2, 3, 1} 顺序进行循环更新，循环的起点由调度 PUSCH 的 DCI 指示的 rv_{id} 确定。另外，如果高层信令配置了 RV 序列指示信息 repK-RV，则按照 repK-RV 指示的顺序更新 RV。对于基于重复类型 B，PUSCH 的第 n 次实际重复使用的 RV 如表 1 所示。其中，实际重复是名义重复被时隙边界、无效符号切分后得到的，名义重复是基站直接配置的，例如起点和长度指示值 (start and length indicator value, SLIV)，重复次数 K 等参数。

表 1 重复的资源映射机制

调度 PUSCH 的 DCI 指示的 rv_{id}	重复 Type A: 第 n 个传输时机使用的 rv_{id} 重复 Type B: 第 n 次实际重复使用的 rv_{id}
------------------------------	--

	$n \bmod 4 = 0$	$n \bmod 4 = 1$	$n \bmod 4 = 2$	$n \bmod 4 = 3$
0	0	2	3	1
2	2	3	1	0
3	3	1	0	2
1	1	0	2	3

- RV 用于指示循环缓存中一个传输时机携带的编码比特。详细地说, 经过低密度奇偶校验 (low density parity check, LDPC) 编码后的数据保存在循环缓存中, 称作编码比特; 每次传输根据 rv_{id} 确定 RV 起点, 记作 k_0 ; 从循环缓存的 k_0 处依次选取编码比特, 映射至一次传输时机的时频资源上, 直到时频资源用完为止。其中, RV 起点共有 4 个, 分别用 $rv_{id}=0$, $rv_{id}=1$, $rv_{id}=2$ 和 $rv_{id}=3$ 指示, 并分别记作 RV0, RV1, RV2 和 RV3, 每个 RV 起点 k_0 与 LDPC 编码使用的基图 (basic graph, BG) 有关, 如表 2 所示, 其中 N_{cb} 表示编码比特长度, Z_c 表示 LDPC 编码的扩展因子。

表 2 RV 起点位置 k_0

rv_{id}	k_0	
	LDPC BG1	LDPC BG2
0	0	0
2	$\left\lfloor \frac{17N_{cb}}{66Z_c} \right\rfloor Z_c$	$\left\lfloor \frac{13N_{cb}}{50Z_c} \right\rfloor Z_c$
3	$\left\lfloor \frac{33N_{cb}}{66Z_c} \right\rfloor Z_c$	$\left\lfloor \frac{25N_{cb}}{50Z_c} \right\rfloor Z_c$
1	$\left\lfloor \frac{56N_{cb}}{66Z_c} \right\rfloor Z_c$	$\left\lfloor \frac{43N_{cb}}{50Z_c} \right\rfloor Z_c$

- 示例性的, 如图 4 中的 (a) 所示, 圆环表示循环缓存, 存储信息比特经过信道编码之后得到的编码后比特串序列, RV0 至 RV3 的位置根据表 2 确定; 如图 4 中的 (b) 所示, 为基于重复类型 A 的 TDRA 的资源映射结果, 按照 RV 序列 {0, 2, 3, 1} 的顺序将编码比特依次映射至 4 次传输时机的时频资源。

- 重复的资源映射机制, 即 RV 循环机制, 一个 TB 只包括一个时隙上的数据包, 而一个 TBoMS 是多个时隙上数据包的聚合结果。在相同数量的时隙上, TBoMS 的 TBS 要远大于重复的 TBS, 很容易出现信息比特无法完整发送的问题, 从而导致解调性能降低。

- 示例性地, 如图 5 所示, 圆环表示循环缓存, 该循环缓存用于存储信息比特经过信道编码之后得到的编码后比特串序列, 圆环中的空白部分为编码后的校验比特, 圆环中的阴影部分为编码后的信息比特。RV0,1,2,3 为该编码后比特序列的近似 4 等分的位置点, 当采用 RV 循环 {0,2,3,1} 的方式进行发送时, 即不同时隙传输的比特从不同 RV 起点进行选择 (选择出一定长度的比特序列) 进行映射和发送。

注意, 全文中循环缓存和循环缓存器意思一致。

6、上行控制信息 (uplink control information, UCI) 复用技术

为了降低终端设备上行发送的交调干扰, 当物理上行控制信道 (physical uplink control channel, PUCCH) 与 PUSCH 在时域上发生重叠时, 支持两种传输方法:

- 1) 当 PUCCH 优先级高于 PUSCH 时, 或者 PUCCH 优先级与 PUSCH 优先级相同但不满足时间线条件时, 丢弃 PUSCH 传输;
- 2) 当 PUCCH 优先级与 PUSCH 优先级相同且满足时间线条件时, 将 PUCCH 上承载

的 UCI 复用在上行 PUSCH 上传输。

针对第二种情况，即 UCI 复用在 PUSCH 上传输，UCI 包括混合自动重传请求确认 (hybrid automatic repeat request-acknowledgement, HARQ-ACK)、信道状态信息 (channel state information, CSI)、配置授权上行控制信息 (configured grant-uplink control information, CG-UCI) 中的至少一种，简述如下：

- 1) 生成 UCI 比特序列，包括 HARQ-ACK、CSI、CG-UCI 中的至少一种；
- 2) 对生成的 UCI 比特序列进行码块切分，生成并添加循环冗余校验 (cyclic redundancy check, CRC) 码；
- 3) 对每个码块进行信道编码，其中 HARQ-ACK 与 CSI 独立编码，CSI part 1 与 CSI part 2 独立编码；对于 CG-UCI，如果没有 HARQ-ACK 复用在 PUSCH 上传输，则 CG-UCI 独立编码；否则，CG-UCI 和 HARQ-ACK 联合编码。
- 4) 对编码后的每个码块进行速率匹配，包括计算 UCI 每层的编码调制符号数以及比特选择；
- 5) 码块级联，将每个码块速率匹配输出的比特序列串联；
- 6) 将经过码块级联的 UCI 比特调制后映射到指定的 PUSCH 传输资源上。
- 7、取消传输；

在通信过程中，可能会有一些信号的发送资源发生冲突，即资源重叠，而终端设备可能不具有同时发送这个信号的能力，就需要定义一些规则，在冲突发送的时候该优先发送哪些信号，从而保证网络设备和终端设备的理解一致，保证通信的正常进行。下面列举一些，现有技术中各种冲突场景下的冲突处理机制。

Case1: 半静态配置的下行时隙或者符号与上行数据信道 PUSCH 的资源发生冲突

网络设备会通过发送高层信令给 UE 配置帧结构，即那些时隙/或者符号为下行，那些时隙/符号为上行，以及那些时隙/符号为灵活的，具体的高层信令可以为 tdd-UL-DL-ConfigurationCommon 和 tdd-UL-DL-ConfigurationDedicated。一般来说配置为下行的时隙或者符号只能传下行信号，配置为上行的时隙或者符号只能传上行信号，配置为灵活的符号有时候可以传输上行或者下行的信号。如果 PUSCH 所占的符号和半静态配置的下行符号有重叠，即使是部分重叠，PUSCH 也会被取消。

Case2: 动态指示的下行符号与半静态配置的上行数据信道 PUSCH 的资源发生冲突

PUSCH 有两种调度方式，具体如下面描述：

第一种：Dynamic grant (DG) 的调度，也就是基于授权的调度。调度网络设备会给用户发送控制信息 PDCCH，该 PDCCH 会调度上行数据 PUSCH 传输，也就是该 PDCCH 会指示 PDSCH 或者 PUSCH 的时域资源；

第二种：配置授权调度。配置授权调度包括 2 种类型，configured grant type 1 和 configured grant type 2，其中 configured grant type 2 的调度方式和第一种动态调度的方式类似，即有一个 PDCCH 激活这个配置调度，之后按照高层配置信息发送，而对于 configured grant type 1，网络设备不会给用户发送控制信息 PDCCH，数据传输所占据的时域资源的位置是通过高层信令配置的。

网络设备可以通过发送一个 PDCCH，承载一个 DCI 格式 2_0，这个 DCI 格式可以指示一个或者多个时隙中的符号为上行符号或者下行符号或者灵活符号。或者 DCI 可以调度一个 PDSCH 或者是 CSI-RS 在一些符号上，如果配置授权调度的 PUSCH 所占的符号和该

DCI 指示的符号有重叠 (这个 DCI 可以是 DCI 格式 2_0, 或者也可以是调度下行数据 PDSCH 或者 CSI-RS 的 DCI), 则按照下面的规则处理:

如果终端设备没有上报部分取消的能力, 或者网络设备没有指示部分取消的能力, 则如果 PUSCH 的开始符号在该 DCI 的最后一个符号之后的 $T'_{proc,2}$ 时间内, 则此时不会取消

5 PUSCH 传输, 否则, 该 PUSCH 传输被取消。

如果终端设备上报了部分取消的能力, 或者网络设备指示部分取消的能力, 则仅取消该 DCI 的最后一个符号之后的 $T'_{proc,2}$ 时间后的 PUSCH 的传输, 之前的不取消。

其中, $T'_{proc,2}$ 通过下面的公式计算:

$$T'_{proc,2} = \max\left(\left(N_2 + d_{2,1} + d_2\right)(2048 + 144) \cdot \kappa 2^{-\mu} \cdot T_c + T_{ext} + T_{switch}, d_{2,2}\right),$$

10 其中, 交叠的 PUCCH(s) 和 PUSCH(s) 中的任意一个与该 PDCCH 对应, 此处的“对应”可以表示该 PDCCH 用于调度 PUSCH 传输, 或者 PDCCH 用于调度 PDSCH, 同时 PDCCH 指示承载 PDSCH 反馈信息的 PUCCH。

其中, N_2 是基于 μ 的根据表 3 和表 4 为 UE 处理能力 1 和 UE 处理能 2 分别确定的 PDSCH 准备时间, 其中 μ 对应 (μ_{DL} , μ_{UL}) 中的一个, μ 的取值满足 $T'_{proc,2}$ 取最大值。其中 μ_{DL} 15 对应调度 PUSCH 的 PDCCH 的子载波间隔, μ_{UL} 对应传输 PUSCH 的上行信道的子载波间隔。对于非共享频谱信道接入的处理, $T_{ext} = 0$ 。如果 PUSCH 的第一个符号只包含 DM-RS, 那么 $d_{2,1} = 0$, 否则 $d_{2,1} = 1$ 。如果调度 DCI 触发 BWP 切换, $d_{2,2}$ 等于切换时间, 否则 $d_{2,2} = 0$ 。如果高优先级的 PUSCH 与低优先级的 PUCCH 重叠, 高优先级 PUSCH 的 d_2 通过 UE 上报值来设置, 否则 $d_2 = 0$ 。

20 表 3 PUSCH 定时能力 1 下的 PUSCH 准备时间

μ	PUSCH 准备时间 N_2 [个符号]
0	10
1	12
2	23
3	36

表 4 PUSCH 定时能力 2 下的 PUSCH 准备时间

μ	PUSCH 准备时间 N_2 [个符号]
0	5
1	5.5
2	11 for frequency range 1

Case3: 重复的 PUCCH 和 PUSCH 的资源发生冲突

如果 PUCCH 将在 N 个 slot 中发送, 而 PUSCH 也会在某个或者某些 slot 中发送, 并且 PUCCH 可能和 PUSCH 在某些 slot 中发送重叠, 这时候, 在重叠的那些 slot 中, 终端 25 设备仅发送 PUCCH, 不发送 PUSCH。

Case4: 高优先级的 PUCCH 和低优先级的 PUSCH 的资源发生冲突

PUCCH 的优先级可能是由调度 PDSCH 的 DCI 中指示的, 也可能是由高层信令配置的, 总之有一种方式能确定某个 PUCCH 的优先级为高优先级还是低优先级。动态调度的 PUSCH 的优先级是 DCI 中指示的, 配置调度的 PUSCH 的优先级是高层信令配置的。

30 一旦高优先级的 PUCCH 的资源和低优先级的 PUSCH 的资源重叠, 则低优先级的 PUSCH 被取消。

Case5: 在载波聚合场景下, 如果参考 cell 上或者其他 cell 上的符号方向与 PUSCH 冲突

如果 UE 配置了多个小区, 并且通过配置 half-duplex-behavior = 'enable', 指示需要进行半双工冲突处理, 且 UE 没有能力在任何一个小区中支持同时收发, 且 UE 上报网络设备能够支持半双工操作, 且没有配置盲检测 DCI 格式 2_0, 则下列符号和 PUSCH 冲突, 则 PUSCH 被取消:

- 5 如果这个符号在任何一个服务小区中被 ssb-PositionsInBurst in SIB1 or ssb-PositionsInBurst in ServingCellConfigCommon, 指示为用于接收同步信号 (synchronization signal, SS) /物理广播信道 (physical broadcast channel, PBCH) 块的符号。

- 10 如果这个符号在参考小区中被 tdd-UL-DL-ConfigurationCommon or tdd-UL-DL-ConfigurationDedicated 参数指示成为下行, 或者被高层信令配置为接收 PDCCH, PDSCH, 或者 CSI-RS。

Case6: 取消指示信息指示取消 PUSCH 的传输

- 15 网络设备可以给终端设备发送一个取消指示信息 Cancellation indication, 承载在 DCI 格式 2_4 中, 通过一个比特位图具体指示在一个参考时频资源内, 哪些资源位置的 PUSCH 要被取消, 具体指示为 1 的位置表示要取消。如果 PUSCH 所占的符号内至少有一个符号被指示为 1, 则从该 PUSCH 中第一个被指示为 1 的符号开始之后的 PUSCH 全部被取消。

比如 PUSCH 占据符号 2-12, 其中符号 3, 4, 被指示为 1, 则符号 2 还保留, 从符号 3-12 的 PUSCH 全部被取消。

- 20 与长期演进 LTE 和 LTE-A 无线通信系统相比, 5G NR 无线通信系统部署的频段更高, 可以获取更大的通信带宽。然而, 高频段会导致更大的路径损耗和穿透损耗, 使得 NR 的覆盖性能远不如 LTE 和 LTE-A。

- 25 为了提升覆盖性能, 提出 TBoMS 技术。该技术将每个时隙上的小数据包聚合成为一个大数据包, 并在多个时隙上共同传输聚合后的数据包, 通过减少 TB 切分次数减少 CRC 的开销, 通过增加 TBS 提高编码增益, 并通过降低频率资源提高功率谱密度, 提升覆盖性能。

目前, TBoMS 速率匹配每个时隙执行一次, 如图 6 所示, 当前速率匹配比特选择起始位置是上一次比特选择结束位置的下一个比特, 即: $k_i = \begin{cases} k_0, i=0 \\ l_{i-1} + 1, i=1, 2, \dots \end{cases}$ 。其中, i 表示

- 30 速率匹配索引, $i=1, 2, \dots$, k_i 表示此次速率匹配比特选择的起始比特在循环缓存器中的位置, l_{i-1} 表示上一次速率匹配比特选择的结束比特在循环缓存器中的位置, k_0 表示第一次速率匹配比特选择的起始比特在循环缓存器中的位置, 由此次 TBoMS 传输的 RV 号 (rv_id=0, 1, 2, 或 3) 和 LDPC 基图决定, 如表 2 所示, 其中 N_{cb} 表示信道编码后的码块大小/循环缓存器的大小 (长度), Z_c 表示 LDPC 编码的扩展因子 (LDPC lifting size)。

在 3GPP RAN1#106-e 次会议上提出, TBoMS PUSCH 传输的 TBS 计算的中间参数——信息比特 N_{info} 的计算公式为:

- 35
$$N_{info} = K \cdot N_{RE} \cdot R \cdot Q_m \cdot N_L$$

其中, N_{RE} 表示一个时隙上的资源单元 (resource element, RE) 数, R 表示编码码率, Q_m 表示调制阶次, N_L 表示层数, K 为 TBS 计算的缩放参数, 表示将 K 个时隙上的 TB 聚合合成一个大 TB。同时, 分配给 TBoMS PUSCH 传输的时隙数 N , 表示将 K 个时隙聚合的

TB 在 N 个时隙上传输, 其中时隙数 N 和缩放参数 K 满足条件: $K \leq N$, 这意味着 TB 聚合的时隙数和 TB 传输的时隙数可以不同。如图 23 所示, 假设时分双工 (time division duplex, TDD) 频谱上下行时隙配比为 DDSUU, $N=8$, $K=4$, 故 TBoMS PUSCH 传输将 4 个时隙聚合的大 TB 在 8 个时隙上进行传输。

- 5 目前, 在终端设备进行速率匹配和网络设备解速率匹配 (可以理解为终端设备的速率匹配的逆处理过程) 的过程中, 都会根据上一次速率匹配比特选择的结束比特在循环缓存器中的位置实时计算本次速率匹配比特选择的起始比特在循环缓存器中的位置, 一旦有一方确定的上一次速率匹配比特选择的结束比特在循环缓存器中的位置错误, 那么从本次速率匹配开始, 所有的速率匹配比特选择的起始比特的位置都会确定错误, 这会造成误差传播, 导致网络设备期望接收到的比特和终端发送的比特无法对齐, 使得网络侧译码错误。应理解, 在计算上一次速率匹配比特选择的结束比特在循环缓存器中的位置之前需要确定上一次速率匹配输出的比特长度。目前, 一般根据可用于传输 TB 的编码比特总数 G' 计算速率匹配输出的比特长度 (G' is the total number of coded bits available for transmission of the transport block)。需要注意的是, 当有 UCI 复用在 PUSCH 上时, UCI 占用了一部分原先要用于传输 TB 的编码比特, 那么在计算 G' 时需要在不复用 UCI 时的编码比特总数中把 UCI 占用的编码比特总数扣除。其中, UCI 可以包括以下至少一项: HARQ-ACK, A-CSI, CSI part1, CSI part2, CG-UCI, HARQ-ACK 和 CG-UCI 联合编码。下面以 UCI 为 A-CSI 为例进行详细说明。

- 20 例如, 网络设备发送 PDCCH 调度终端上报 A-CSI, 且上报的 A-CSI 需要复用在 TBoMS 传输上。网络设备对上行共享控制信道 (uplink shared channel, UL-SCH) 做解速率匹配, 并按照上述方式确定每次速率匹配比特选择的起始比特位置, 如果 UL-SCH 上有 A-CSI 复用, 则在速率匹配计算输出比特长度时, 要扣除 A-CSI 的编码比特总数以得到可用于传输 TB 的编码比特总数 G' 。假设 A-CSI 复用在第二个时隙上, 则第二次速率匹配输出的可用于传输 TB 的比特长度相对于其他时隙而言要偏小, 故第三次速率匹配的起始比特在循环缓存器中的位置要提前。那么, 网络设备在做解速率匹配的时候也会将扣除 A-CSI 的编码比特总数的 G' 作为第二次速率匹配输出的可用于传输 TB 的比特长度。那么网络设备则认为终端设备通过如图 7 中的 (a) 所示的上行时隙向网络设备传输 TB, 然而, 如果终端设备漏检此 PDCCH, 则终端无法获知需要上报 A-CSI, 则终端设备在确定第二速率匹配输出的可用于传输 TB 的比特长度时不会扣除 A-CSI 的编码比特总数, 终端设备在确定第三个时隙内速率匹配比特选择的起始比特位置时, 不会扣除 A-CSI 的编码比特总数以得到可用于传输 TB 的编码比特总数 G' , 那么终端设备向网络设备传输 TB 的上行时隙如图 7 中的 (b) 所示。比较图 7 中的 (a) 和 7 中的 (b) 可见, 此时网络设备和终端设备没有对齐每次速率匹配比特选择的起始比特在循环缓存中的位置, 会导致网络设备译码错误。

- 35 下面对于 G 和 G' 进行举例说明。 $G' = \Phi - \theta - \psi$, 其中 Φ 是分配的时频资源总共可以承载的编码比特数, θ 是被参考信号或其他信道占用的编码比特数, ψ 表示 UCI 占用的编码比特数, 其中 θ 大于等于 0, 在没有其他参考信号或其他信道时 $\theta = 0$, 在没有 UCI 占用时, $\psi = 0$ 。

一种可能的方式 (无 UCI 复用), $G' = N_{RE} \times Q_m \times N_L$, N_{RE} 、 Q_m 、 N_L 分别表示 RE

数、调制阶次、层数, 其中 $N_{RE} = \min(156, N_{RE}') n_{prb}$, n_{prb} 表示分配的总 PRB 个数。其中, $N_{RE}' = N_{sc}^{RB} \cdot N_{ymb}^{sh} - N_{D,urP}^{PRB} - N_{oh}^{PRB}$, 其中 $N_{sc}^{RB} = 12$ 是物理资源块中频域的子载波数, N_{ymb}^{sh} 是 PUSCH 分配的符号数 L , N_{DMRS}^{PRB} 是分配的持续时间内每个 PRB 的 DM-RS 的 RE 数, 包括 DM-RS CDM 组没有数据, N_{oh}^{PRB} 是 PUSCH-ServingCellConfig 中高层参数 xOverhead 配置的开销。

另一种可能的方式 (有 UCI 复用), $G' = N_{RE} \times Q_m \times N_L - \psi$ 。

而本申请实施例提供的上行传输的方法中涉及的 G , 在上述两种方式中, 都不会减去 ψ 。

8、时域资源分配 (TimeDomainResourceallocation, TDRA) 表

PUSCH-TDRA 表用于配置 PDCCH 和 PUSCH 的时域关系。PUSCH-TDRA 表包含一个或多个这样的 PUSCH-TDRA。UE 基于 PUSCH-TDRA 表中的条目数确定 DCI 字段的比特宽度。DCI 字段中的值 0 表示该列表中的第一个元素, DCI 字段中的值 1 表示该列表中的第二个元素, 以此类推。

PUSCH-TimeDomainResourceAllocationList-r16 ::= SEQUENCE (SIZE (1..maxOfUL-Allocations-r16)) OF

PUSCH-TimeDomainResourceAllocation-r16

PUSCH-TimeDomainResourceAllocation-r16 ::= SEQUENCE {

k2-r16 INTEGER (0..32)..... OPTIONAL, -Need -S

puschAllocationList-r16SEQUENCE-(SIZE (1.. maxNrofMultiplePUSCHs-r16))-OF PUSCH-Allocation-r16,

...

20 }

PUSCH-Allocation-r16 ::= SEQUENCE {

mappingType-r16 ENUMERATED {type A, typeB} OPTIONAL, -Cond NotFormat01-02-0r-TypeA

startSymbolAndLength-r16 INTEGER-(0..127).....OPTIONAL, -Cond NotFormat01-02-0r-TypeA

25 startSymipol-r16- INTEGER (0..13) OPTIONAL, -Cond-RepTypeB

length-r16.....INTEGER-(1..14).....OPTIONAL, -Cond -RepTypeB

umberofRepetitions-r16.....ENUMERATED-(n1, n2, n3, n4, n7, n8, n12, n16) OPTIONAL, -Cond-Format01-02

...

具体可以参见协议 TS 38.331。

30 9、填充比特 (filler bit)

在本申请中也被称为空比特 (NULL)。

应理解, LDPC 编码要求编码前码块大小是 Z_c 的整数倍, Z_c 是 LDPC lifting size。然而, 经过 TBS 计算、CRC 添加、基图选择、码块切分后, 每个码块的大小并不一定是 Z_c 的整数倍。为此, 需要在每个码块后面添加填充比特, 使得 LDPC 编码前的码块大小为 Z_c 的整数倍。

关于填充比特的描述具体可以参见协议 TS 38.212 Section 5.2.2 节。

下面结合图 8 至图 14, 对本申请提供的上行传输的方法进行介绍。

下面结合图 8, 对本申请实施例的上行传输的方法 100 进行详细说明。

图 8 是本申请的方法 100 的示意性交互图。方法 100 根据终端设备是否检测到第二指

示信息可以通过两种不同的方式实现。

第一种可能的实现方式，终端设备检测到第二指示信息。

S101，网络设备向终端设备发送第一指示信息，相应地，终端设备接收来自网络设备的第一指示信息，第一指示信息用于指示终端设备在多个时间单元上发送第一传输块。

5 应理解，这里的第一传输块是一个传输块。

需要说明的是，本申请涉及的时间单元可用理解为为可用时隙（available slot），目前 available slot 是根据上下行时隙配比、TDRA 和同步信号和物理广播信道块（synchronization signal and PBCH block, SSB）共同决定的。

10 例如，上下行时隙配比为 DDSUU，S 时隙前 10 个符号为下行符号，中间 2 个符号为灵活符号，后面 2 个符号为上行符号；TDRA 指示：起始时隙为第一个 D，并指示时隙数为 4，一个时隙内的起始符号 S 为第一个符号，符号长度 L 为 10；根据 S 和 L 指示，D 时隙和 S 时隙对应的符号为下行符号，不能用于 TBoMS 传输，从而可以确定 available slot 为 DDSUDDSUU 中的四个 U。

15 再例如，S 指示第 13 个符号，L = 2，则 available slot 可以是 DDSUDDSUU 中的前两个 S 和 U，因为 S 和 L 指示 S 时隙中的上行符号，故 S 时隙也可以用于跨多时隙传输。

S102，网络设备向终端设备发送第二指示信息，相应地，终端设备接收来自网络设备的第二指示信息，该第二指示信息用于指示所述终端设备在第一时间单元上发送上行控制信息 UCI，第一时间单元为多个时间单元中的一个。

20 应理解，第一指示信息已经指示终端设备在多个时间单元上发送第一传输块，第二指示信息指示终端设备在第一时间单元上发送 UCI，终端设备会向网络设备发送复用有 UCI 的第一传输块。其中，复用有 UCI 的第一传输块可以理解为第一时间单元上承载了 UCI 和第一传输块的编码比特。

25 S103，终端设备根据第一时间单元对应的参数 G 确定第一编码比特序列的起始比特在循环缓存中的位置，第一编码比特序列为第二时间单元内经过速率匹配输出的编码比特序列，参数 G 为一个时间单元内可用于传输第一传输块和 UCI 的编码比特总数，G 为正整数，第二时间单元为第一时间单元在多个时间单元中的下一个时间单元。

30 应理解，在多个时间单元中，每个时间单元上进行一次速率匹配，输出一组编码比特序列。作为第一时间单元的下一个时间单元，第二时间单元内经过速率匹配输出的编码比特序列（上述第一编码比特序列）的起始比特在循环缓存中的位置是根据第一时间单元对应的参数 G 确定的。具体地，根据 G 确定第一编码比特序列的起始比特在循环缓存中的位置时，需要先根据 G 确定第一时间单元内经过速率匹配输出的编码比特序列的长度，再根据该长度确定第一编码比特序列的起始比特在循环缓存中的位置。或者，也可以直接根据 G 确定第一编码比特序列的起始比特在循环缓存中的位置。

S104，终端设备在多个时间单元上向网络设备发送 UCI 和第一传输块。

35 应理解，在具体实现中，由于在第一时间单元上部分时域资源用于承载 UCI，因此 S104 中网络设备可能无法完整地发送 S101 中的指示的第一传输块。

S105，网络设备根据第一时间单元对应的参数 G 确定第一编码比特序列的起始比特在循环缓存中的位置。

40 应理解，S105 中网络设备确定第一编码比特序列的起始比特在循环缓存中的位置的方案与 S103 中终端设备确定第一编码比特序列的起始比特在循环缓存中的位置的方案类

似。

具体可以参见方法 200 中第一种可能的实现方式中, S203 中对应的描述。

本申请实施例, 网络设备指示终端设备在第一时间单元上承载 UCI 的指示信息, 在终端设备检测到该指示的情况下, 网络设备和终端设备在确定第二时间单元 (第一时间单元的下一个时间单元) 内经过速率匹配输出的编码比特序列的起始比特在循环缓存中的位置时使用的参数 G 都不扣除 UCI 的编码比特总数, 可以使得从第二时间单元开始网络设备和终端设备可以对齐每次速率匹配中比特选择的起始比特在循环缓存器中的位置, 从而提高译码成功率。

其中, S103 中, 一种可能的实现方式, 终端设备根据第一时间单元对应的参数 G 的 a 倍确定第一编码比特序列的起始比特在循环缓存中的位置, $0 < a \leq 1$ 。其中, a 等于以下任意一个数: $0.1x$, $0 < x < 10$, 或者, $0.2x$, $0 < x < 5$, 或者, $0.01x$, $0 < x < 100$, 其中, x 为整数。例如 a 可以是 $0.1, 0.2, 0.3 \dots \dots, 0.9, 1$ 中的任意一个数。或者, 可以是 $0.2, 0.4, 0.6, 0.8, 1$ 中的任意一个数。

类似地, 终端设备根据 aG 确定第一编码比特序列的起始比特在循环缓存中的位置时, 需要先根据 aG 确定第一时间单元内经过速率匹配输出的编码比特序列的长度, 再根据该长度确定第一编码比特序列的起始比特在循环缓存中的位置。

相应地, 网络设备也用相同的方式确定第一编码比特序列的起始比特在循环缓存中的位置。

进一步地, 方法 100 还包括:

终端设备根据第一时间单元对应参数 G 确定第二编码比特序列的长度。

具体可以参见方法 200 中第二种可能的实现方式中, S202 中步骤二和步骤三部分对应的描述。

本申请实施例, 在终端设备接收到第二指示信息的情况下, 网络设备和终端设备可以使用与终端设备未接收到第二指示信息时相同的方式确定第一编码比特序列的起始比特在循环缓存中的位置和第二编码比特序列的长度, 扩展了本申请提供的上行传输的方法的应用范围。

S103 中, 另一种可能的实现方式, 所述终端设备根据所述第一时间单元对应的参数 G 和 Z_c 确定所述第一编码比特序列的起始比特在循环缓存中的位置, Z_c 为低密度奇偶校验 LDPC 因子。

相应地, 网络设备也用相同的方式确定第一编码比特序列的起始比特在循环缓存中的位置。

具体可以参见方法 200 中第三种可能的实现方式中对应的描述。

应理解, 现有技术中, 终端侧在进行资源映射时, 需要查找映射至上一个时间单元的最后一个比特在循环缓存中的位置, 计算开销大。

本申请实施例, 终端侧在进行资源映射时, 减少了查找映射至上一个时间单元的最后一个比特在第一比特序列中的位置的步骤, 在根据 G 确定第一编码比特序列的起始比特在循环缓存中的位置的基础上结合 Z_c 确定该位置, 简化了映射过程中的计算, 节省了开销。

S103 中, 另一种可能的实现方式, 所述终端设备根据所述第一时间单元对应的参数 G 的 a 倍和 Z_c 确定所述第一编码比特序列的起始比特在循环缓存中的位置, Z_c 为低密度奇偶校验 LDPC 因子。

相应地,网络设备也用相同的方式确定第一编码比特序列的起始比特在循环缓存中的位置。

具体可以参见方法 200 中第二种可能的实现方式中, S202 中步骤二和步骤三部分对应的描述。

- 5 应理解,现有技术中,终端侧在进行资源映射时,需要查找映射至上一个时间单元的最后一个比特在循环缓存中的位置,计算开销大。

本申请实施例,终端侧在进行资源映射时,减少了查找映射至上一个时间单元的最后一个比特在第一比特序列中的位置的步骤,在根据 aG 确定第一编码比特序列的起始比特在循环缓存中的位置的基础上结合 Z₀ 确定该位置,简化了映射过程中的计算,节省了开销。

10

进一步地,方法 100 还包括:

- 终端设备将第二编码比特序列映射至第一时间单元上的时频资源,第二比特序列为第一时间单元内经过速率匹配输出的编码比特序列,第二编码比特序列的比特数等于第一时间单元对应的参数 G;终端设备将 UCI 映射至第一时间单元上的部分时频资源,以替换承载于部分时频资源的第一编码比特序列中的部分比特序列。
- 15

需要说明的是,在具体实现中,映射至时频资源的比特序列不是经过速率匹配的序列,还需要经过码块级联后才可以映射至时频资源。

- 本申请实施例,通过打孔的方式将 UCI 复用在第一时间单元上,可以使得第一时间单元上除 UCI 占用的时频资源外的时频资源上承载的包含于第一传输块的比特序列不改变,从而可以提高网络设备译码的成功率和准确性。
- 20

进一步地,方法 100 还包括:

第一指示信息还用于指示网络设备能够在多个时间单元上发送一个传输块,第一传输块为一个传输块,其中,第一指示信息包括使能跨多时隙传输块传输字段,或者,第一指示信息包括跨多时隙传输块映射类型参数。

- 25 应理解,上述方案给出了第一指示信息的几种可能的信令设计,其中,第一指示信息包括使能跨多时隙传输块传输字段可以通过在第一指示信息中新增 TDRA 表实现,第一指示信息包括跨多时隙传输块映射类型参数可以通过将第一指示信息中的 mappingtype 字段由 1 比特修改为 2 比特实现。

具体可以参见方法 200 中 S201 中对应的描述。

- 30 本申请实施例,通过多种形式向终端设备指示网络设备支持跨多时隙传输功能,丰富信令设计,提高上行传输的效率。

第二种可能的实现方式,终端设备没有检测到第二指示信息。

S101'具体可以参见第一种可能的实现方式中的 S101 中对应的描述。

- 35 S102',网络设备向终端设备发送第二指示信息,该第二指示信息用于指示所述终端设备在第一时间单元上发送上行控制信息 UCI,第一时间单元为多个时间单元中的一个。但是,终端设备并没有检测到来自网络设备的第二指示信息,因此终端设备也不会第一时间单元上发送 UCI。

S103',终端设备根据第一时间单元对应的参数 G 确定第一编码比特序列的起始比特在循环缓存中的位置,参数 G 为一个时间单元内可用于传输第一传输块和 UCI 的编码比特总数, G 为正整数,第二时间单元为第一时间单元在多个时间单元中的下一个时间单元。

40

应理解,此时因为终端设备没有检测到第二指示信息,因此第一时间单元对应的参数G即为该第一时间单元内可用于传输第一传输块的编码比特总数。

还应理解,在多个时间单元中,每个时间单元上一次速率匹配,输出一编码比特序列。作为第一时间单元的下一个时间单元,第二时间单元内经过速率匹配输出的编码比特序列(上述第一编码比特序列)的起始比特在循环缓存中的位置是根据第一时间单元对应的参数G确定的。

可选地,S103'也可以替换为终端设备根据现有技术确定第一编码比特序列的起始比特在循环缓存中的位置。

S104',终端设备在多个时间单元上向网络设备发送第一传输块。

10 应理解,终端设备在多个时间单元上发送的是第一传输块,而网络设备认为终端设备在第一时间单元上发送了UCI。

S105',网络设备根据第一时间单元对应的参数G确定第一编码比特序列的起始比特在循环缓存中的位置,第一编码比特序列为第二时间单元内经过速率匹配输出的编码比特序列,参数G为一个时间单元内能够用于传输第一传输块和UCI的编码比特总数,G为正整数,第二时间单元为第一时间单元在多个时间单元中的下一个时间单元。

15 应理解,虽然网络设备认为终端设备在第一时间单元上发送了UCI,但是网络设备用于确定第一编码比特序列的起始比特在循环缓存中的位置的G并没有减去UCI的编码比特总数,因此无论S103'中终端设备根据与G确定第一编码比特序列的起始比特的在循环缓存中的位置,还是根据现有技术确定第一编码比特序列的起始比特的在循环缓存中的位置,20 终端设备在S103'和网络设备在S105'中计算出的第一编码比特序列的起始比特的在循环缓存中的位置都是一致的。

本申请实施例,网络设备指示终端设备在第一时间单元上承载UCI的指示信息,在终端设备没有检测到该指示的情况下,网络设备和终端设备在确定第二时间单元(第一时间单元的下一个时间单元)内经过速率匹配输出的编码比特序列的起始比特在循环缓存中的位置时使用的参数G都不扣除UCI的编码比特总数,可以使得从第二时间单元开始网络25 设备和终端设备可以对齐每次速率匹配中比特选择的起始比特在循环缓存器中的位置,从而提高译码成功率。

其中,S103'中,一种可能的实现方式,终端设备根据第一时间单元对应的参数G的a倍确定第一编码比特序列的起始比特在循环缓存中的位置, $0 < a \leq 1$ 。其中,a等于以下30 任意一个数: $0.1x$, $0 < x < 10$,或者, $0.2x$, $0 < x < 5$,或者, $0.01x$, $0 < x < 100$,其中,x为整数。例如a可以是0.1,0.2,0.3...,0.9,1中的任意一个数。或者,可以是0.2,0.4,0.6,0.8,1中的任意一个数。

类似地,终端设备根据aG确定第一编码比特序列的起始比特在循环缓存中的位置时,需要先根据aG确定第一时间单元内经过速率匹配输出的编码比特序列的长度,再根据该35 长度确定第一编码比特序列的起始比特在循环缓存中的位置。

相应地,网络设备也用相同的方式确定第一编码比特序列的起始比特在循环缓存中的位置。

应理解,终端设备没有检测到第二指示信息的情况下,网络设备如果按照没有扣除UCI的编码比特总数的参数G计算第一编码比特序列的起始比特的位置,那么该位置与第二编码比特序列的结束比特之间会存在一段间隙,因为在网络设备侧,该间隙是用于承载40

UCI 的 RE 造成的。在系统比特中存在的间隙可能会导致信道质量变差。

本申请实施例,通过 G 乘上大于零且小于一的 a ,使得网络设备计算第一编码比特序列的起始比特在循环缓存中的位置可以提前,从而缩短第二编码比特序列的结束比特与第一编码比特序列的起始比特之间的间隙,提高信道质量。

5 进一步地,方法 100 还包括:

终端设备根据第一时间单元对应参数 G 确定第二编码比特序列的长度,所述第二编码比特序列为所述第一时间单元内经过速率匹配输出的编码比特序列。

本申请实施例,进一步地,仍然根据 G 计算第二编码比特序列的长度,由于 G 大于 aG ,可以进一步确保消除上述间隙。

10 S103 中,另一种可能的实现方式,所述终端设备根据所述第一时间单元对应的参数 G 和 Z_c 确定所述第一编码比特序列的起始比特在循环缓存中的位置, Z_c 为低密度奇偶校验 LDPC 因子。

相应地,网络设备也用相同的方式确定第一编码比特序列的起始比特在循环缓存中的位置。

15 具体可以参见方法 200 中第三种可能的实现方式中对应的描述。

应理解,现有技术中,终端侧在进行资源映射时,需要查找映射至上一个时间单元的最后一个比特在循环缓存中的位置,计算开销大。

20 本申请实施例,终端侧在进行资源映射时,减少了查找映射至上一个时间单元的最后一个比特在第一比特序列中的位置的步骤,在根据 G 确定第一编码比特序列的起始比特在循环缓存中的位置的基础上结合 Z_c 确定该位置,简化了映射过程中的计算,节省了开销。

S103 中,另一种可能的实现方式,所述终端设备根据所述第一时间单元对应的参数 G 的 a 倍和 Z_c 确定所述第一编码比特序列的起始比特在循环缓存中的位置, Z_c 为低密度奇偶校验 LDPC 因子。

25 相应地,网络设备也用相同的方式确定第一编码比特序列的起始比特在循环缓存中的位置。

具体可以参见方法 200 中第二种可能的实现方式中,S202 中步骤二和步骤三部分对应的描述。

应理解,现有技术中,终端侧在进行资源映射时,需要查找映射至上一个时间单元的最后一个比特在循环缓存中的位置,计算开销大。

30 本申请实施例,终端侧在进行资源映射时,减少了查找映射至上一个时间单元的最后一个比特在第一比特序列中的位置的步骤,在根据 aG 确定第一编码比特序列的起始比特在循环缓存中的位置的基础上结合 Z_c 确定该位置,简化了映射过程中的计算,节省了开销。

进一步地,方法 100 还包括:

35 第一指示信息还用于指示网络设备能够在多个时间单元上发送一个传输块,第一传输块为一个传输块,其中,第一指示信息包括使能跨多时隙传输块传输字段,或者,第一指示信息包括跨多时隙传输块映射类型参数。

40 应理解,上述方案给出了第一指示信息的几种可能的信令设计,其中,第一指示信息包括使能跨多时隙传输块传输字段可以通过在第一指示信息中新增 TDRA 表实现,第一指示信息包括跨多时隙传输块映射类型参数可以通过将第一指示信息中的 mappingtype 字段

由 1 比特修改为 2 比特实现。

具体可以参见方法 200 中 S201 中对应的描述。

本申请实施例,通过多种形式向终端设备指示网络设备支持跨多时隙传输功能,丰富信令设计,提高上行传输的效率。

- 5 下面结合图 9,对本申请实施例的上行传输的方法 200 进行详细说明。图 9 是本申请的方法 200 的示意性交互图。

具体地,方法 200 可以有多种不同的实现方式,下面以几种不同的实现方式为例进行介绍。该几种不同的实现方式分别给出了终端设备如何根据时间单元#n-1 对应的 G 计算时间单元#n 内速率匹配输出的比特序列的起始比特的位置。

- 10 第一种可能的实现方式:终端设备根据时间单元#n-1 对应的 G 计算时间单元#n 内速率匹配输出的比特序列的起始比特的位置,根据时间单元#n 对应的 G 或 G' 计算时间单元#n 内速率匹配输出的比特序列的长度。其中,如果每个时间单元上包括的 OFDM 符号的数量不同,则每个时间单元上对应的 G 也会不同,如果每个时间单元上包括的 OFDM 符号的数量相同,则每个时间单元上对应的 G 也相同。

- 15 需要说明的是,在下文的实施例中,为了方便,以一个时间单元等于一个时隙为例进行说明,在具体实现中,一个时间单元还可以理解为本申请术语解释部分设计的符号、物理时隙、可用时隙等,本申请对此不做限定。

- 20 S201,网络设备向终端设备发送调度信息,该调度信息用于指示终端设备在多个时隙上发送 PUSCH,该 PUSCH 承载一个 TB 对应的 UL-SCH,或者,该调度信息指示终端设备发送 TBoMS PUSCH。另外,调度信息还指示终端设备在所述 PUSCH 上承载 UCI。

需要说明的是,终端设备可能检测到调度信息中用于指示终端设备在所述 PUSCH 上承载 UCI 的信令,也可能没有检测到该信令。

应理解,终端设备在以下几种情况下会在 PUSCH 上承载 UCI,下面给出两种常见的示例:

- 25 1) DCI 触发终端设备执行 A-CSI 测量并上报 CSI,且上报 CSI 与 PUSCH 在时域上重叠;

2) DCI 触发终端设备接收 PDSCH,并在 PUCCH 上发送所述 PDSCH 对应的 HARQ-ACK,但所述 PUCCH 与所述 PUSCH 在时域上重叠,且所述 PUCCH 的优先级与所述 PUSCH 的优先级相同。

- 30 可选的,该调度信息还指示以下几项中的至少一项: TBoMS 时隙数、TBoMS 重复次数、TBS 计算缩放参数、RV 更新粒度,使能 TBoMS 传输。

下面介绍几种可能的信令设计:

可能的设计一:

- 35 调度信息指示 TBoMS 时隙数等于 N, N 小于等于 4、6 或 8。具体实现可以是在 RRC 信令 PUSCH-TimeDomainResourceAllocationList 中的 PUSCH-Allocation 中新增 1-3 比特的字段 numberOfSlots 以指示该时隙数 N。

其中, N 的候选值是 {1,2,...,8} 或其子集。示例性的,该子集可以是 {2, 4}, {2,4, 6}, {2, 4, 8}, {2, 4, 6, 8}, {2, 4, 7}, {2, 4, 7, 8}, {2, 3, 4, 7, 8}, {1, 2, 4}, {1, 2, 4, 6}, {1, 2, 4, 8}, {1, 2, 4, 6, 8}, {1, 2, 4, 7}, {1, 2, 4, 7, 8}, {1, 2, 3, 4, 7, 8}。

- 40 应理解,如果不支持 TBoMS 重复, TBoMS 时隙数仅表示分配给一次 TBoMS 传输的

时隙数。TBS 计算缩放参数和 RV 更新粒度由 TBoMS 时隙数隐性指示，其中，TBS 计算缩放参数、RV 更新粒度和 TBoMS 时隙数均等于 K，终端设备可以根据该隐性指示方式确定 $K=N$ 。

对于网络设备是否使能 TBoMS 传输功能，可以向终端设备显性指示，也可以隐性指示。其中，显性指示有三种可能的方法：

(1) 在现有 TDRA 表基础上新增专用于 TBoMS 时域资源分配的 TDRA 表：

PUSCH-TimeDomainResourceAllocationListTBoMS。

应理解，与其它 TDRA 表相比，上述新增的 TDRA 表优先级最高，即如果配置了该新增的 TDRA 表，此 TDRA 表生效，而其它 TDRA 不再生效；因此，如果配置了所述新增的 TDRA 表，则使能 TBoMS 传输；否则，没有使能 TBoMS 传输。其中，其它 TDRA 表包括：

- *pusch-TimeDomainAllocationList*
- *pusch-TimeDomainAllocationListDCI-0-1*
- *pusch-TimeDomainAllocationListDCI-0-2*
- *pusch-TimeDomainAllocationListForMultiPUSCH*

(2) 在 RRC 信令 PUSCH-TimeDomainResourceAllocationList 中 PUSCH-Allocation 中新增 1 比特字段：enableTBoMS，该字段为“0”或者没有配置此字段表示没有使能 TBoMS 传输，否则使能 TBoMS 传输；

(3) 修改 RRC 信令 PUSCH-TimeDomainResourceAllocationList 中 PUSCH-Allocation 中的字段 mappingType，由 1bit 修改为 2bit，分别指示{typeA, typeB, TBoMS}，“10”或“11”指示 TBoMS 传输。

可选地，如果明确使能 TBoMS 传输，但 TBoMS 时隙数 $N=1$ ，则不使能 TBoMS 传输。隐性指示的方法为：当 TBoMS 时隙数 > 1 时，表示使能 TBoMS 传输；如果 TBoMS 时隙数 $= 1$ 或 TBoMS 时隙数没有配置时，表示没有使能 TBoMS 传输。

可能的设计二：

基于可能的设计一，调度信息还指示 TBoMS 重复次数 N_{rep} ，由 RRC 信令 PUSCH-TimeDomainResourceAllocationList 中 PUSCH-Allocation 中的已有字段 numberOfRepetitions 指示。

应理解，此时支持 TBoMS 重复，TBoMS 重复的总时隙数 $= N \times N_{rep}$ 。可选的，总时隙数不期望超过 16，24 或 32。

可能的设计三：

调度信息指示 TBoMS 时隙数，该时隙数小于等于 4、6 或 8，由 RRC 信令 PUSCH-TimeDomainResourceAllocationList 中 PUSCH-Allocation 中已有的字段 numberOfRepetitions 指示，候选值是{1,2,...,8}或其子集，示例性的，该子集可以是{2, 4}，{2, 4, 6}，{2, 4, 8}，{2, 4, 6, 8}，{2, 4, 7}，{2, 4, 7, 8}，{2, 3, 4, 7, 8}，{1, 2, 4}，{1, 2, 4, 6}，{1, 2, 4, 8}，{1, 2, 4, 6, 8}，{1, 2, 4, 7}，{1, 2, 4, 7, 8}，{1, 2, 3, 4, 7, 8}。

应理解，如果不支持 TBoMS 重复，TBoMS 时隙数仅表示分配给一次 TBoMS 传输的时隙数。TBS 计算缩放参数和 RV 更新粒度由 TBoMS 时隙数隐性指示，其中，TBS 计算缩放参数和、RV 更新粒度由和 TBoMS 时隙数均等于 K，终端设备可以根据该隐性指示方式确定 $K=N$ 。

另一方面,由于所述时隙数由已有的字段 `numberOfRepetitions` 指示,现有技术无法区分重复和 TBoMS 两个功能,故需要增加新的信令显性指示是否使能 TBoMS 传输,具体如可能的设计一中的三种显性指示方法所示。

可能的设计四:

- 5 调度信息指示 TBoMS 时隙数,该时隙数小于等于 16, 24 或 32, 由 RRC 信令 `PUSCH-TimeDomainResourceAllocationList` 中 `PUSCH-Allocation` 中已有的字段 `numberOfRepetitions` 指示,该字段有 3-4 比特,候选值是{1,2,...,32}或其子集。示例性的, {1, 2, 3, 4, 7, 8, 12, 16}, {1, 2, 3, 4, 7, 8, 12, 16, 20}, {1, 2, 3, 4, 7, 8, 12, 16, 24}, {1, 2, 3, 4, 7, 8, 12, 16, 28}, {1, 2, 3, 4, 7, 8, 12, 16, 32}, {1, 2, 3, 4, 7, 8, 12, 16, 20, 24}, {1, 2, 3, 4, 7, 8, 12, 16, 20, 28}, {1, 2, 3, 4, 7, 8, 12, 16, 20, 32}, {1, 2, 3, 4, 7, 8, 12, 16, 24, 28}, {1, 2, 3, 4, 7, 8, 12, 16, 24, 32}, {1, 2, 3, 4, 7, 8, 12, 16, 28, 32}, {1, 2, 3, 4, 7, 8, 12, 16, 20, 24, 28}, {1, 2, 3, 4, 7, 8, 12, 16, 20, 24, 32}, {1, 2, 3, 4, 7, 8, 12, 16, 20, 28, 32}, {1, 2, 3, 4, 7, 8, 12, 16, 24, 28, 32}, {1, 2, 3, 4, 7, 8, 12, 16, 20, 24, 28, 32}。

- 15 该调度信息指示 TBS 计算缩放参数,该缩放参数的值等于 K, K 小于等于 4、6 或 8, 在 RRC 信令 `PUSCH-TimeDomainResourceAllocationList` 中 `PUSCH-Allocation` 中增加 1-3bits 新的字段 `scalingFactor` 指示该缩放参数,候选值是{1,2,...,8}或其子集。示例性的, {2, 4}, {2, 4, 6}, {2, 4, 8}, {2, 4, 6, 8}, {2, 4, 7}, {2, 4, 7, 8}, {2, 3, 4, 7, 8}, {1, 2, 4}, {1, 2, 4, 6}, {1, 2, 4, 8}, {1, 2, 4, 6, 8}, {1, 2, 4, 7}, {1, 2, 4, 7, 8}, {1, 2, 3, 4, 7, 8}。

可选的,不期望 N 或 K 不为整数。

- 20 应理解,如果 $K=N$,本质上不是 TBoMS 重复,时隙数仅表示分配给一次 TBoMS 传输的时隙数;如果 $K<N$,本质上是 TBoMS 重复,时隙数表示包括传输 TBoMS 重复在内的总时隙数, K 表示一次 TBoMS 传输的时隙数,且若 N/K 为整数,则 TBoMS 重复次数为 N/K 。RV 更新粒度为 K 个时隙;不期望 $K>N$ 。

- 25 对于是否使能 TBoMS 传输功能,可以显性指示,也可以隐性指示。显性指示如可能的设计一中三种显性指示方法所示,可选地,如果明确使能 TBoMS 传输,但 $K=1$ 或 $K>N$, 则不使能 TBoMS 传输。隐性指示的方法为:如果配置 TBS 计算缩放参数 K, 且 $K \neq 1$, $K \leq N$, 则表示使能 TBoMS 传输;否则(即没有配置缩放参数 K, 或配置了 K 但 $K=1$ 或 $K>N$), 没有使能 TBoMS 传输。

可能的设计五:

- 30 该调度信息指示 TBoMS 时隙数,该时隙数小于等于 16, 24 或 32, 在 RRC 信令 `PUSCH-TimeDomainResourceAllocationList` 中 `PUSCH-Allocation` 中增加 3-4 比特新的字段 `numberOfSlots` 指示所述时隙数 N, 候选值是{1,2,...,32}或其子集,示例性的, {1, 2, 3, 4, 7, 8, 12, 16}, {1, 2, 3, 4, 7, 8, 12, 16, 20}, {1, 2, 3, 4, 7, 8, 12, 16, 24}, {1, 2, 3, 4, 7, 8, 12, 16, 28}, {1, 2, 3, 4, 7, 8, 12, 16, 32}, {1, 2, 3, 4, 7, 8, 12, 16, 20, 24}, {1, 2, 3, 4, 7, 8, 12, 16, 20, 28}, {1, 2, 3, 4, 7, 8, 12, 16, 20, 32}, {1, 2, 3, 4, 7, 8, 12, 16, 24, 28}, {1, 2, 3, 4, 7, 8, 12, 16, 24, 32}, {1, 2, 3, 4, 7, 8, 12, 16, 28, 32}, {1, 2, 3, 4, 7, 8, 12, 16, 20, 24, 28}, {1, 2, 3, 4, 7, 8, 12, 16, 20, 24, 32}, {1, 2, 3, 4, 7, 8, 12, 16, 20, 28, 32}, {1, 2, 3, 4, 7, 8, 12, 16, 24, 28, 32}, {1, 2, 3, 4, 7, 8, 12, 16, 20, 24, 28, 32}。

- 40 该调度信息指示 TBS 计算缩放参数,该缩放参数的值为 K, K 小于等于 4、6 或 8, 由 RRC 信令 `PUSCH-TimeDomainResourceAllocationList` 中 `PUSCH-Allocation` 中 1-3 比特

已有的字段 `numberOfRepetitions` 指示缩放参数, 候选值是 $\{1, 2, \dots, 8\}$ 的子集。示例性的, $\{1, 2, 3, 4, 7, 8, 12, 16\}$, $\{1, 2, 3, 4, 7, 8, 12, 16, 20\}$, $\{1, 2, 3, 4, 7, 8, 12, 16, 24\}$, $\{1, 2, 3, 4, 7, 8, 12, 16, 28\}$, $\{1, 2, 3, 4, 7, 8, 12, 16, 32\}$, $\{1, 2, 3, 4, 7, 8, 12, 16, 20, 24\}$, $\{1, 2, 3, 4, 7, 8, 12, 16, 20, 28\}$, $\{1, 2, 3, 4, 7, 8, 12, 16, 20, 32\}$, $\{1, 2, 3, 4, 7, 8, 12, 16, 24, 28\}$, $\{1, 2, 3, 4, 7, 8, 12, 16, 24, 32\}$, $\{1, 2, 3, 4, 7, 8, 12, 16, 28, 32\}$, $\{1, 2, 3, 4, 7, 8, 12, 16, 20, 24, 28\}$, $\{1, 2, 3, 4, 7, 8, 12, 16, 20, 24, 32\}$, $\{1, 2, 3, 4, 7, 8, 12, 16, 20, 28, 32\}$, $\{1, 2, 3, 4, 7, 8, 12, 16, 24, 28, 32\}$, $\{1, 2, 3, 4, 7, 8, 12, 16, 20, 24, 28, 32\}$ 。

可选的, 不期望 N/K 不为整数。

应理解, 可能的设计四和可能的设计五区别在于具体的信令指示。

10 可能的设计六:

该调度信息指示 TBoMS 时隙数, 该时隙数值为 N , N 小于等于 16, 24 或 32, 在 RRC 信令 `PUSCH-TimeDomainResourceAllocationList` 中 `PUSCH-Allocation` 中增加 3-4bits 新的字段 `numberOfSlots` 指示该时隙数, 候选值是 $\{1, 2, \dots, 32\}$ 或其子集, 示例性的, $\{1, 2, 3, 4, 7, 8, 12, 16\}$, $\{1, 2, 3, 4, 7, 8, 12, 16, 20\}$, $\{1, 2, 3, 4, 7, 8, 12, 16, 24\}$, $\{1, 2, 3, 4, 7, 8, 12, 16, 28\}$, $\{1, 2, 3, 4, 7, 8, 12, 16, 32\}$, $\{1, 2, 3, 4, 7, 8, 12, 16, 20, 24\}$, $\{1, 2, 3, 4, 7, 8, 12, 16, 20, 28\}$, $\{1, 2, 3, 4, 7, 8, 12, 16, 20, 32\}$, $\{1, 2, 3, 4, 7, 8, 12, 16, 24, 28\}$, $\{1, 2, 3, 4, 7, 8, 12, 16, 24, 32\}$, $\{1, 2, 3, 4, 7, 8, 12, 16, 28, 32\}$, $\{1, 2, 3, 4, 7, 8, 12, 16, 20, 24, 28\}$, $\{1, 2, 3, 4, 7, 8, 12, 16, 20, 24, 32\}$, $\{1, 2, 3, 4, 7, 8, 12, 16, 20, 28, 32\}$, $\{1, 2, 3, 4, 7, 8, 12, 16, 24, 28, 32\}$, $\{1, 2, 3, 4, 7, 8, 12, 16, 20, 24, 28, 32\}$ 。

20 该调度信息指示 TBoMS 重复次数 N_{rep} , 由 RRC 信令 `PUSCH-TimeDomainResourceAllocationList` 中 `PUSCH-Allocation` 中的已有字段 `numberOfRepetitions` 指示。

应理解, 此时支持 TBoMS 重复, 重复次数由 N_{rep} 指示, 时隙数表示分配给 TBoMS 重复的所有时隙数。TBS 计算缩放参数和 RV 更新粒度由 TBoMS 重复次数 N_{rep} 和时隙数共同指示, 即 $K=N/N_{\text{rep}}$, 故不期望 N/N_{rep} 不是整数。

25 对于是否使能 TBoMS 传输功能, 可以显性指示, 也可以隐性指示。显性指示如可能的设计一中三种显性指示方法所示, 可选地, 如果明确使能 TBoMS 传输, 但 $N_{\text{rep}} \geq N$ 或 N/N_{rep} 不为整数, 则不使能 TBoMS 传输。隐性指示的方法为: 如果 $N_{\text{rep}} < N$ 且 N/N_{rep} 为整数, 则使能 TBoMS 传输; 否则, 不使能 TBoMS 传输。

30 可选的, 当所述终端设备被调度发送 PUSCH, 所述 PUSCH 没有承载 TB 但承载 CSI 上报, 不期望使用 TBoMS 的时域资源分配发送所述 PUSCH。

以上介绍了调度信息的六种的信令设计。

S202, 根据调度信息生成 PUSCH。

所述终端设备根据所述调度信息执行速率匹配, 生成所述 UL-SCH。

35 应理解, 除了速率匹配之外, 生成所述 UL-SCH 还包括 TBS 计算、码块划分、LDPC 编码, 码块级联、资源映射、调制等过程, 由于这些过程都可以按照现有技术执行, 因此在本申请中不多赘述。

具体地, 步骤一, 所述终端设备根据所述调度信息指示的发送所述 PUSCH 的时隙数, 确定需要执行速率匹配的次数、执行每次速率匹配的时间单元。

40 可选的, 执行速率匹配的次数等于发送所述 PUSCH 的时隙数等于 N 。应理解, 每个

时隙执行一次速率匹配,每次速率匹配输出的比特长度由一个时隙上可用的时频资源确定。

示例性的,如图 10 中的(a)所示,其中上下行时隙配比为 DDDSU,发送所述 PUSCH 的时隙数=4,假设 S 时隙和 U 时隙均可用于发送所述 PUSCH,则速率匹配的次数=4,执行速率匹配 RM#0/RM#2 的时间单元为 S 时隙,执行速率匹配 RM#1/RM#3 的时间单元为 U 时隙。

示例性的,如图 10 中的(b)所示,其中上下行时隙配比为 DDSUU,发送所述 PUSCH 的时隙数=4,假设仅 U 时隙用于发送所述 PUSCH,则速率匹配的次数=4,执行每次速率匹配的时间单元均为 U 时隙。

后续通过步骤二至步骤四详细介绍如何确定速率匹配输出的比特序列。

- 10 其中,在步骤二中,根据本申请提供的上行传输的方法中的参数 G 计算速率匹配输出的比特序列的长度,记作 E_r ,在步骤三中,根据 E_r 确定速率匹配输出的比特序列的起点,在步骤四中,终端设备根据起始比特位置和比特长度 E_r 执行比特选择,确定输出的比特序列。或者,在步骤二中,根据现有技术中的参数 G' 计算速率匹配输出的比特序列的长度 E_r' ,在步骤三中,根据本申请提供的上行传输的方法中的参数 G 计算速率匹配输出的比特序列的长度,记作 E_r ,根据 E_r 确定速率匹配输出的比特序列的起点,在步骤四中,终端设备根据起始比特位置和比特长度 E_r' 执行比特选择,确定输出的比特序列。

其中,对于只有 1 个码块的情况,由于 $E_r=G$,步骤二和步骤三也可以简化为,不进行步骤二,步骤三也可以直接根据 G 直接确定速率匹配输出的比特序列的起点。

- 20 在具体阐述步骤二至步骤四之前,下面先以图 7 中的(b)为例,介绍一下 G 和 G' 的主要区别。假设 RM#2 对应的时隙内需要复用 UCI。那么现有技术的方案计算 RM#3 对应的时隙内输出的比特序列的起始比特在循环缓存中的位置时,使用的编码比特总数 G' 仅为可用于传输 TB 的编码比特总数,即 G' 是不包括 UCI 占用的编码比特总数的。而本申请实施例中计算 RM#3 对应的时隙内输出的比特序列的起始比特在循环缓存中的位置时,编码比特总数 G 包括用于传输 TB 的编码比特总数和 UCI 占用的编码比特总数。

- 25 步骤二,终端设备确定速率匹配输出的比特长度。

应理解,终端设备需要根据编码比特总数 G 或 G' 确定速率匹配输出的比特长度,而 G 或 G' 可以是终端设备确定的也可以是网络设备指示的(例如通过 RRC 或 DCI 指示)。

- 30 还应理解,如果不同的时间单元上的可以用于上行传输的时域资源是不同的,例如特殊时隙和上行时隙上包含的 OFDM 符号的数量是不同的,那么不同时隙上的 G 或 G' 也是不同的,从而终端设备需要分别根据不同的时隙确定不同的 G 或 G' 。

下面在介绍根据 G 或 G' 确定比特长度时,也会根据每个时间单元上包括的 OFDM 符号的数量相同或不同分别进行介绍。

一、根据 G 计算 E_r :

(1) 每个时间单元上包括的 OFDM 符号的数量不同

- 35 终端设备根据在时间单元#n 上传输 TB 的编码比特总数 G_n ,确定时间单元#n 上码块#r 的速率匹配输出的比特长度 $E_{r,n}$, $r=0,1,\dots,C-1$, $n=0,1,\dots,N-1$,其中 C 为码块个数。示例性的,终端设备确定比特长度 $E_{r,n}$ 如算法 1 所示;其中, C' 为被调度的码块个数。

可选的,编码比特总数 G_n 由时间单元#n 内分配给 PUSCH 的 RE 总数 $N_{RE,n}$ 确定,且不排除 UCI 占用的编码比特总数。示例性的, $G_n = N_{RE,n} \times Q_m \times N_L$,其中 Q_m 表示调制阶次、 N_L 表示层数。

应理解,无论终端设备是否将 UCI 与 UL-SCH 复用, G_n 的计算都不扣除 UCI 的载荷大小。具体地,如果终端设备漏检用于指示 UCI 与 UL-SCH 复用的调度信息,那么终端设备自然无法得知 UCI 占用的编码比特大小,如果终端设备检测到用于指示 UCI 与 UL-SCH 复用的调度信息且在 UL-SCH 上复用 UCI,那么 G_n 也是不扣除的。

- 5 示例性的,以图 10 中的 (a) 为例,假设分配的时间单元为 available slot,假设 S 时隙和 U 时隙都被认定为 available slot,其中,时间单元数 =4, S 时隙可用符号数为 4, U 时隙可用符号数为 13, RB 个数为 40,则 $N_{RE,0} = N_{RE,2} = 1920$, $N_{RE,1} = N_{RE,3} = 6240$;同时,假设 $Q_m = 2$, $N_L = 1$, $C' = C = 2$,则 $G_0 = G_2 = 3840$, $G_1 = G_3 = 12480$,故 $E_{0,0} = E_{1,0} = E_{0,2} = E_{1,2} = 960$, $E_{0,1} = E_{1,1} = E_{0,3} = E_{1,3} = 3900$ 。

10

算法 1:

```

Set j=0
for n=0 to N-1
  for r=0 to C-1
    if CBGTI 指示第 r 个码块没有被调度用于传输
       $E_{r,n} = 0$ ;
    else
      If  $j \leq C' - \text{mod}(G_n / (N_L \cdot Q_m), C') - 1$ 
         $E_{r,n} = N_L \cdot Q_m \cdot \left\lfloor \frac{G_n}{N_L \cdot Q_m \cdot C'} \right\rfloor$ ;
      else
         $E_{r,n} = N_L \cdot Q_m \cdot \left\lceil \frac{G_n}{N_L \cdot Q_m \cdot C'} \right\rceil$ ;
      end if
      j = j + 1;
    end if
  end for
end for

```

(2) 每个时间单元上包括的 OFDM 符号的数量相同,根据 G 计算比特长度的方式如下所示:

- 15 终端设备根据在一次速率匹配对应的一个时间单元上传输 TB 的编码比特总数 G,确定时间单元 #n 上码块 #r 的速率匹配输出的比特长度 E_r , $r=0,1,\dots,C-1$, $n=0,1,\dots,N-1$ 。示例性的,终端设备确定比特长度 E_r 如算法 2 所示。

可选的,编码比特总数 G 由一个时间单元内分配给 PUSCH 的 RE 总数 N_{RE} 确定,且不扣除 UCI 的编码比特大小。示例性的, $G = N_{RE} \times Q_m \times N_L$,其中 Q_m 表示调制阶次、 N_L 表示层数。应理解,此时每次速率匹配所在时间单元内分配给 PUSCH 的 RE 总数一致。

- 20 示例性的,以图 10 中的 (b) 为例,假设 $N=4$,每个 U 时隙的可用符号数为 13, RB 个数为 40,则 $N_{RE}=6240$;同时,假设 $Q_m=2$, $N_L=1$, $C'=C=3$,则 $G=12480$,故 $E_0 = E_1 = E_2 = 4160$ 。

算法 2:

```

Set j=0
for r=0 to C-1

```

if CBGTI 指示码块# r 没有被调度用于传输

$E_r = 0$;

else

if $j \leq C' - \text{mod}(G / (N_L \cdot Q_m), C') - 1$

$$E_r = N_L \cdot Q_m \cdot \left\lfloor \frac{G}{N_L \cdot Q_m \cdot C'} \right\rfloor;$$

else

$$E_r = N_L \cdot Q_m \cdot \left\lceil \frac{G}{N_L \cdot Q_m \cdot C'} \right\rceil;$$

end if

$j = j + 1$;

end if

end for

二、根据 G' 计算 E_r'

(1) 每个时间单元# n 上包括的 OFDM 符号的数量不同:

根据 G_n' 计算 $E_{r,n}'$ 的方式与根据 G_n 计算 $E_{r,n}$ 中的 (1) 类似, 区别在于:

- 5 所有 G_n 替换成 G_n' , 所有 $E_{r,n}$ 替换为 $E_{r,n}'$ 。其中, 编码比特总数 G_n' 由时间单元# n 内分配给 PUSCH 的 RE 总数 $N_{RE,n}$ 确定, 并扣除 UCI 占用的编码比特总数的编码比特总数。示例性的, 编码比特总数 G_n 由时间单元# n 内分配给 PUSCH 的 RE 总数 $N_{RE,n}$ 确定 (不扣除 UCI 占用的编码比特总数)。示例性的, $G_n = N_{RE,n} \times Q_m \times N_L$, $G_n' = N_{RE,n} \times Q_m \times N_L - \psi$, 其中 Q_m 表示调制阶次、 N_L 表示层数、 ψ 表示 UCI 占用的编码比特总数。

(2) 每个时间单元# n 上包括的 OFDM 符号的数量相同:

- 10 根据 G' 计算 E_r' 方式与根据 G 计算 E_r 中的 (2) 类似, 区别在于:

所有 G 替换成 G' , 所有 E_r 需要替换为 E_r' 。其中, 编码比特总数 G 由一个时间单元内分配给 PUSCH 的 RE 总数 N_{RE} 确定, 编码比特总数 G 由一个时间单元内分配给 PUSCH 的 RE 总数 N_{RE} 确定, 并扣除 UCI 占用的编码比特总数的编码比特总数。示例性的, $G = N_{RE} \times Q_m \times N_L$, $G' = N_{RE} \times Q_m \times N_L - \psi$, 其中 Q_m 表示调制阶次、 N_L 表示层数、 ψ 表示 UCI 占用的编码比特总数。

- 15

步骤三, 终端设备确定速率匹配比特选择的起始比特位置。

应理解, 终端设备根据速率匹配输出的比特长度 E_r 确定速率匹配比特选择的起始比特位置。

下面给出几种可能的实现方式。

- 20 可能的实现方式一:

所述终端设备根据时间单元# $n-1$ 上的码块# r 比特选择的起始比特位置 $k_{r,n-1}$ 和比特长度 $E_{r,n-1}$, 以及 RV 更新时间间隔 K , 确定时间单元# n 上的码块# r 的速率匹配比特选择的起始比特位置 $k_{r,n}$ 。

- 25 示例性的, 所述终端设备确定时间单元# n 上的码块# r 的速率匹配比特选择的起始比特位置 $k_{r,n}$ 如算法 3 所示, 或基于下式确定:

$$k_{r,n} = \begin{cases} k_0, & n = mK \\ \text{mod} \left(k_0 + \sum_{k=0}^{n-1} E_{r,k}, N_{cb} \right), & \text{otherwise} \end{cases}$$

- 其中, m 是整数, k_0 表示比特选择起始比特的偏置, N_{cb} 表示 LDPC 编码之后最终存储在循环缓存器中的码块大小。应理解, k_0 根据 RV 索引 rv_{id} 和 BG 共同决定, 其中 RV 索引 rv_{id} 会根据起始 RV 和 RV 序列确定; 当 n 是 K 的整数倍时, 时间单元# n 上的码块# r 的速率匹配比特选择的起始比特位置 $k_{r,n}$ 由 k_0 决定, 即由 RV 指定; 当 n 不是 K 的整数倍时, 时间单元# n 上的码块# r 的速率匹配比特选择的起始比特位 $k_{r,n}$ 由时间单元# $n-1$ 上的码块# r 的速率匹配比特选择的起始比特位置 $k_{r,n-1}$ 和比特长度 $E_{r,n-1}$ 确定。

算法 3:

```

for  $n = 0$  to  $N-1$ 
  if  $n = mK$ 
     $k_{r,n} = k_0$ ;
  else
     $k_{r,n} = \text{mod}(k_{r,n-1} + E_{r,n-1}, N_{cb})$ ;
  end if
end for

```

可能的实现方式二:

- 所述终端设备根据码块# r 的一次速率匹配输出的比特长度 E_r 和 RV 更新时间间隔 K , 确定时间单元# n 上的码块# r 的速率匹配比特选择的起始比特位置 $k_{r,n}$ 。

示例性的, 所述终端设备确定时间单元# n 上的码块# r 的速率匹配比特选择的起始比特位置 $k_{r,n}$ 如下式所示, 其中 m 是整数。

$$k_{r,n} = \begin{cases} k_0, & n = mK \\ \text{mod}(k_0 + nE_r, N_{cb}), & \text{otherwise} \end{cases}$$

应理解, 此时每次速率匹配所在时间单元内分配给 PUSCH 的 RE 总数一致。

- 可能的实现方式三:

所述终端设备根据码块# r 速率匹配# $n-1$ 比特选择的起始比特位置 $k_{r,n-1}$ 和比特长度 $E_{r,n-1}$, NULL 比特的个数, 以及 RV 更新时间间隔 K , 确定码块# r 速率匹配# n 比特选择的起始比特位置 $k_{r,n}$ 。

- 示例性的, 所述终端设备确定码块# r 速率匹配# n 比特选择的起始比特位置 $k_{r,n}$ 如算法 4 所示。其中, m 是整数, K_{NULL}' 表示在插入 NULL 比特之前的码块大小, K_{NULL} 表示插入 NULL 比特之后的码块大小, Z_c 表示 LDPC 扩展因子, $K_{NULL}' - 2Z_c$ 表示系统比特的个数, $K_{NULL} - K_{NULL}'$ 表示 NULL 比特的个数, $N_{cb} - (K_{NULL} - K_{NULL}')$ 表示循环缓存器中非 NULL 比特的比特个数。 k_0 、 N_{cb} 、 K_{NULL} 、 K_{NULL}' 、 Z_c 和 BG 的确定方法均为现有技术。

- 应理解, 如果当前速率匹配比特选择经过 NULL 比特时, 下一次速率匹配比特选择的起始位置是当前速率匹配起始位置加上当前速率匹配输出序列长度再加上 NULL 比特数; 否则, 下一次速率匹配比特选择的起始位置是当前速率匹配起始位置加上当前速率匹配输出序列长度。

算法 4:

```

for  $n = 0$  to  $N-1$ 

```

```

if  $n = mK$ 
     $k_{r,0} = k_0$ ;
else
     $k_{r,n} = k_{r,n-1} + \text{mod}(E_{r,n-1}, N_{cb} - (K_{NULL} - K_{NULL}'))$ ;
    if  $(k_{r,n-1} < K_{NULL}' - 2Z_c \text{ and } k_{r,n} \geq K_{NULL}' - 2Z_c)$  or
        $(k_{r,n} \geq K_{NULL}' - 2Z_c + N_{cb})$ 
         $k_{r,n} = k_{r,n} + K_{NULL} - K_{NULL}'$ ;
    end if
     $k_{r,n} = \text{mod}(k_{r,n}, N_{cb})$ ;
end if
end for

```

示例性的，以图 10 中的 (a) 为例，假设起始 RV 为 RV2, $K=4$, $N=4$, $K_{NULL}=6336$, $K_{NULL}'=6056$, $N_{cb}=19008$, $Z_c=288$; 同时，仅考虑第一个码块，则 $E_{0,0}=E_{0,2}=960$, $E_{0,1}=E_{0,3}=3900$, 故 $k_{0,0}=0$, $k_{0,1}=960$, $k_{0,2}=4860$, $k_{0,3}=6100$, 如图 11 中的 (a) 所示。

5 可能的实现方式四:

所述终端设备根据码块#r 速率匹配# n-1 比特选择的起始比特位置 $k_{r,n-1}$ ，一次速率匹配输出的比特长度 E_r ，NULL 比特的个数，以及 RV 更新时间间隔 K ，确定码块#r 速率匹配# n 比特选择的起始比特位置 $k_{r,n}$ 。示例性的，如算法 5 所示。应理解，此时每次速率匹配所在时间单元内分配给 PUSCH 的 RE 总数一致。

算法 5:

```

for  $n = 0$  to  $N-1$ 
    if  $n = mK$ 
         $k_{r,0} = k_0$ ;
    else
         $k_{r,n} = k_{r,n-1} + \text{mod}(E_r, N_{cb} - (K_{NULL} - K_{NULL}'))$ ;
        if  $(k_{r,n-1} < K_{NULL}' - 2Z_c \text{ and } k_{r,n} \geq K_{NULL}' - 2Z_c)$  or
            $(k_{r,n} \geq K_{NULL}' - 2Z_c + N_{cb})$ 
             $k_{r,n} = k_{r,n} + K_{NULL} - K_{NULL}'$ ;
        end if
         $k_{r,n} = \text{mod}(k_{r,n}, N_{cb})$ ;
    end if
end for

```

- 10 示例性的，以图 10 中的 (b) 为例，假设起始 RV 为 RV1, RV 序列为 {0,2,3,1}, $N=4$, $K_{NULL}=6336$, $K_{NULL}'=6176$, $N_{cb}=19008$, $Z_c=288$; 同时，仅考虑第一个码块，则 $E_0=4160$, 故 $k_{0,0}=4896$, $k_{0,1}=9216$, $k_{0,2}=0$, $k_{0,3}=4160$ 。如图 11 中的 (b) 所示。

以上介绍了可能的实现方式四，下面介绍可能的实现方式五至八。

首先，先大致介绍一下可能的实现方式五至八的整体思路和主要步骤。

- 15 分配给一次 TBoMS 传输的时间单元个数为 N ，TBS 计算缩放参数 $K=N$ ，RV 更新周期也为 N 。应理解，如果没有使能 TBoMS 重复，则 TBoMS 传输占用 N 个时间单元，使用一个 RV，该使用的 RV 由高层信令指示；如果使能 TBoMS 重复， M 表示 TBoMS 重复

次数, 每次 TBoMS 重复占用 N 个时间单元, 故 M 次 TBoMS 重复共占用 $M \times N$ 个时间单元, 每次 TBoMS 重复的第一个时间单元更新 RV, 更新顺序由高层信令指示。

那么, 每个时间单元执行的速率匹配的比特选择的起始比特索引 (或起始比特位置) k_0 可以通过以下方式确定:

- 5 (1) 对于每个 TBoMS 重复的 N 个时间单元中的第一个时间单元
 k_0 是根据 RV 和 BG 确定的, 具体确定方式可以参见本申请中表 2 对应的描述。
 可选的, 如果第一个时间单元对应的速率匹配的比特选择的起始比特索引 k_0 指示的起始比特为填充比特 (filler bits), 则需要将 k_0 修正为指向填充比特的最后一位比特的下一个比特的索引。
- 10 应理解, 比特选择不会将填充比特选择出来, 因此对 k_0 进行修正, 可以保证后续方法得到的每个 TBoMS 重复的 N 个时间单元中的除第一个时间单元之外的其它时间单元对应的速率匹配的比特选择的起始比特是该时间单元的上一个时间单元对应的速率匹配的比特选择的结束比特的下一个比特 (注意, 如果该下一比特是填充比特, 则该下一比特也将被修正为最后一位填充比特的下一比特), 从而保证在没有 UCI 复用、丢弃和取消传输的情况下, 不会出现相邻两个时间单元对应的速率匹配输出的编码比特序列有重叠的问题。
- 15 示例性的, 如图 26 所示, 修正前, $K_{NULL}' - 2Z_c \leq k_0 < K_{NULL} - 2Z_c$, 则修正后, $k_0 = K_{NULL} - 2Z_c$, 其中, $K_{NULL}' - 2Z_c$ 表示循环缓存器中填充比特的第一位比特的位置, $K_{NULL} - 2Z_c$ 表示循环缓存器中填充比特的最后一位比特的下一个比特的位置。
- 20 (2) 对于一次 TBoMS 重复的 N 个时间单元中的除第一时间单元以外的其他时间单元
 该其他时间单元对应的速率匹配的比特选择的起始比特索引 k_0 由以下参数中的至少一项确定:
 - 1、上一个时间单元的速率匹配比特选择的起始比特索引 k_0' ;
 - 2、循环缓存器长度 (周期) N_{cb} , 所述循环缓存器用于存储一个经过 LDPC 编码后的
 - 25 一个码块;
 - 3、循环缓存器中填充比特的数量 $K_{NULL} - K_{NULL}'$;
 - 4、循环缓存器中填充比特的起始位置 $K_{NULL}' - 2Z_c$;
 - 5、循环缓存器中最后一位填充比特的下一个比特的位置 $K_{NULL} - 2Z_c$;
 - 6、上一个时间单元对应的 G 值;
 - 30 7、根据步骤二确定的上一个时间单元对应的速率匹配输出的比特长度 E_r (E_r 由上一个时间单元对应的值 G 确定);
 - 8、第一个时间单元对应的 G 值;
 - 9、根据步骤二确定的第一个时间单元对应的速率匹配输出的比特长度 E_r (由第一个时间单元对应的 G 值确定);
 - 35 10、一次 TBoMS 重复的 N 个时间单元中任一时间单元对应的 G 值;
 - 11、根据步骤二确定的一次 TBoMS 重复的 N 个时间单元中任一时间单元对应的速率匹配输出的比特长度 E_r (由该任一时间单元对应的 G 值确定)。

示例性地, 当码块数量 $C=1$ 时, $E = E_r = G$, $r = 0$, 故此时可以直接根据 G 值确定 k_0 ;

当 $C > 1$ 时, $E_r \neq G$, 需根据 E_r 确定 k_0 。其中, $G \equiv N_{RE} \times Q_m \times N_L$, $N_{RE} = \min(156, N_{RE}') n_{prb}$,

n_{prb} 表示分配的总 PRB 个数, 其中, $N_{RE}' = N_{sc}^{RB} \cdot N_{ymb}^{sh} - N_{D,urP}^{PRB} - N_{oh}^{PRB}$, 其中 $N_{sc}^{RB} = 12$ 是物理资源块中频域的子载波数, N_{ymb}^{sh} 是 PUSCH 分配的符号数 L , N_{DMRS}^{PRB} 是分配的持续时间内每个 PRB 的 DM-RS 的 RE 数, 包括 DM-RS CDM 组没有数据, N_{oh}^{PRB} 是 PUSCH-ServingCellConfig 中高层参数 xOverhead 配置的开销, r 为码块的索引, $r=0,1,\dots,C-1$,

5 $n=0,1,\dots,N-1$, 其中 C 为码块个数; 特别地, 层数 N_L 等于 1, 则 $G \equiv N_{RE} \times Q_m$ 。

下面开始分别介绍可能的实现方式五至八。其中, 分别介绍了根据 E_r 或 G 值 (即 $C \neq 1$ 或 $C=1$) 确定 k_0 的方式。

可能的实现方式五:

步骤 (1),

10 确定比特选择的真实周期 N_{cb}' ;

应理解, i) 比特选择过程中, 会将循环缓存器中的数据依次逐个读取。由于循环缓存器是周期性的 (周期为 N_{cb}), 故比特选择过程选择出来的比特也是周期性的, 这里将比特选择过程选择出来的比特的周期称作比特选择的真实周期, N_{cb}' 。ii) 因为填充比特不会被比特选择过程选择出来, 故比特选择的真实周期小于循环缓存器的长度 (周期) N_{cb} 。

15 示例性的, $N_{cb}' = N_{cb} - (K_{NULL} - K_{NULL}')$ 。

步骤 (2),

将 E_r 或 G 值映射到一个比特选择的真实周期 N_{cb}' 内得到的比特序列的长度, 记作 L 。

L 的表示方式包括但不限于以下几种示例:

$L = \text{mod}(E_r, N_{cb} - (K_{NULL} - K_{NULL}'))$, 或者,

20 $L = \text{mod}(E_r, N_{cb}')$, 或者,

$L = \text{mod}(G, N_{cb} - (K_{NULL} - K_{NULL}'))$, 或者,

$L = \text{mod}(G, N_{cb}')$ 。

应理解, 循环缓存器的寻址方式是取模运算, 即当寻址步进是一个比特选择的真实周期 N_{cb}' 的整数倍时, 连续两次寻址的起始地址不变。通过取模运算, 可以理解为将 E_r 或 G 值拆分为两部分, 其中, 第一部分对应的比特序列长度小于 N_{cb}' , 即 $0 \leq L < N_{cb}'$, 另一部分对应的比特序列长度是 N_{cb}' 的 ε 倍, 其中 ε 是整数且 $\varepsilon \geq 0$ 。那么, $E_r = N_{cb}' \times \varepsilon + L$ 或 $G = N_{cb}' \times \varepsilon + L$ 。

步骤 (3),

30 假设没有填充比特, 根据 L 和 k_0' , 确定一次 TBoMS 重复的 N 个时间单元中除第一个时间单元以外的其他时间单元的起始比特索引 $\bar{k}_0$ (中间变量)。

$\bar{k}_0$ 的表示方式包括但不限于以下几种示例:

$\bar{k}_0 = k_0' + L$, 或者,

$\bar{k}_0 = k_0' + \text{mod}(E_r, N_{cb} - (K_{NULL} - K_{NULL}'))$, 或者,

$\bar{k}_0 = k_0' + \text{mod}(E_r, N_{cb}')$, 或者,

35 $\bar{k}_0 = k_0' + \text{mod}(G, N_{cb} - (K_{NULL} - K_{NULL}'))$, 或者,

$\bar{k}_0 = k_0' + \text{mod}(G, N_{cb}')$ 。

步骤 (4),

根据 k_0' , $\bar{k}_0$ 和填充比特在循环缓存器中的位置, 确定时间单元 #n 的起始比特索引 k_0 。

首先, 根据 k_0' , $\bar{k}_0$ 和填充比特在循环缓存器中的位置, 判断比特选择长度为 L 的比特选择过程是否经过填充比特。下面分别介绍几种不同的情况。

- 5 一, 如果比特序列-1 经过填充比特, 根据填充比特数量 $K_{NULL} - K_{NULL}'$ 对 $\bar{k}_0$ 进行修正, 即 $k_0 = \bar{k}_0 + K_{NULL} - K_{NULL}'$ 。

例如, 如图 27 中的 (a) 所示, $k_0' < K_{NULL}' - 2Z_c$ 且 $\bar{k}_0 \geq K_{NULL}' - 2Z_c$ 时, 比特选择过程经过填充比特, 故 $k_0 = \text{mod}(\bar{k}_0 + K_{NULL} - K_{NULL}', N_{cb})$ 。

- 10 再例如, 如图 27 中的 (b) 所示, 当 $k_0' \geq K_{NULL}' - 2Z_c$ 且 $\bar{k}_0 \geq K_{NULL}' - 2Z_c + N_{cb}$ 时 (或者, 此条件可以简化为: 当 $\bar{k}_0 \geq K_{NULL}' - 2Z_c + N_{cb}$ 时), 比特选择过程经过填充比特, $k_0 = \text{mod}(\bar{k}_0 + K_{NULL} - K_{NULL}', N_{cb})$ 。

二, 如果比特序列-1 没有经过填充比特, 则根据 $\bar{k}_0$ 得到所述 k_0 。

例如, 如图 27 中的 (c) 所示, 当 $k_0' < K_{NULL}' - 2Z_c$ 且 $\bar{k}_0 < K_{NULL}' - 2Z_c$, (或者, 此条件可以简化为: 当 $\bar{k}_0 < K_{NULL}' - 2Z_c$ 时), 比特选择过程没有经过填充比特, 故

- 15 $k_0 = \text{mod}(\bar{k}_0, N_{cb})$ 或 $k_0 = \bar{k}_0$ 。

再例如, 例如, 如图 27 中的 (d) 所示, 当 $k_0' \geq K_{NULL}' - 2Z_c$ 且 $\bar{k}_0 < K_{NULL}' - 2Z_c + N_{cb}$, 比特选择过程没有经过填充比特, 故 $k_0 = \text{mod}(\bar{k}_0, N_{cb})$ 或 $k_0 = \bar{k}_0$ 。

可选的, 如果 k_0 指示的起始比特为填充比特, 即 $K_{NULL}' - 2Z_c \leq k_0 < K_{NULL}' - 2Z_c$, 则根据 $K_{NULL}' - 2Z_c$ 和 k_0' 对所述 $\bar{k}_0$ 进行修正, 获得 k_0 。示例性的,

- 20 $k_0 = \text{mod}(\bar{k}_0 + K_{NULL} - 2Z_c - k_0', N_{cb})$ 或 $k_0 = \bar{k}_0 + K_{NULL} - 2Z_c - k_0'$ 。

下面给出可能的实现方式五的几种示例:

算法 7-a:

$$\bar{k}_0 = k_0' + \text{mod}(E_r, N_{cb} - (K_{NULL} - K_{NULL}'));$$

if ($k_0' < K_{NULL}' - 2Z_c$ and $\bar{k}_0 \geq K_{NULL}' - 2Z_c$) or ($k_0' > K_{NULL}' - 2Z_c$ and $\bar{k}_0 \geq K_{NULL}' - 2Z_c + N_{cb}$)

$$\bar{k}_0 = \bar{k}_0 + K_{NULL} - K_{NULL}';$$

end if

$$k_0 = \text{mod}(\bar{k}_0, N_{cb});$$

算法 7-b:

$$\bar{k}_0 = k_0' + \text{mod}(E_r, N_{cb} - (K_{NULL} - K_{NULL}'));$$

if ($k_0' < K_{NULL}' - 2Z_c$ and $\bar{k}_0 \geq K_{NULL}' - 2Z_c$) or ($\bar{k}_0 \geq K_{NULL}' - 2Z_c + N_{cb}$)

$$\bar{k}_0 = \bar{k}_0 + K_{NULL} - K_{NULL}';$$

end if

$$k_0 = \text{mod}(\bar{k}_0, N_{cb});$$

算法 7-c:

$$\bar{k}_0 = k_0' + \text{mod}(E_r, N_{cb} - (K_{NULL} - K_{NULL}'));$$

if ($k_0' < K_{NULL}' - 2Z_c$ and $\bar{k}_0 \geq K_{NULL}' - 2Z_c$) or ($k_0' > K_{NULL}' - 2Z_c$ and $\bar{k}_0 \geq$

$K'_{NULL} - 2Z_c + N_{cb})$
 $k_0 = \text{mod}(\bar{k}_0 + K_{NULL} - K'_{NULL}, N_{cb});$
 else
 $k_0 = \text{mod}(\bar{k}_0, N_{cb});$
 end if

算法 7-d:

$\bar{k}_0 = k'_0 + \text{mod}(E_r, N_{cb} - (K_{NULL} - K'_{NULL}));$
 if $(k'_0 < K'_{NULL} - 2Z_c \text{ and } \bar{k}_0 \geq K'_{NULL} - 2Z_c)$ or $(\bar{k}_0 \geq K'_{NULL} - 2Z_c + N_{cb})$
 $k_0 = \text{mod}(\bar{k}_0 + K_{NULL} - K'_{NULL}, N_{cb});$
 else
 $k_0 = \text{mod}(\bar{k}_0, N_{cb});$
 end if

算法 7-e:

$\bar{k}_0 = k'_0 + \text{mod}(G, N_{cb} - (K_{NULL} - K'_{NULL}));$
 if $(k'_0 < K'_{NULL} - 2Z_c \text{ and } \bar{k}_0 \geq K'_{NULL} - 2Z_c)$ or $(k'_0 > K_{NULL} - 2Z_c \text{ and } \bar{k}_0 \geq K_{NULL}' - 2Z_c + N_{cb})$
 $\bar{k}_0 = \bar{k}_0 + K_{NULL} - K'_{NULL};$
 end if
 $k_0 = \text{mod}(\bar{k}_0, N_{cb});$

算法 7-f:

$\bar{k}_0 = k'_0 + \text{mod}(G, N_{cb} - (K_{NULL} - K'_{NULL}));$
 if $(k'_0 < K'_{NULL} - 2Z_c \text{ and } \bar{k}_0 \geq K'_{NULL} - 2Z_c)$ or $(\bar{k}_0 \geq K'_{NULL} - 2Z_c + N_{cb})$
 $\bar{k}_0 = \bar{k}_0 + K_{NULL} - K'_{NULL};$
 end if
 $k_0 = \text{mod}(\bar{k}_0, N_{cb});$

算法 7-g:

$\bar{k}_0 = k'_0 + \text{mod}(G, N_{cb} - (K_{NULL} - K'_{NULL}));$
 if $(k'_0 < K'_{NULL} - 2Z_c \text{ and } \bar{k}_0 \geq K'_{NULL} - 2Z_c)$ or $(k'_0 > K_{NULL} - 2Z_c \text{ and } \bar{k}_0 \geq K_{NULL}' - 2Z_c + N_{cb})$
 $k_0 = \text{mod}(\bar{k}_0 + K_{NULL} - K'_{NULL}, N_{cb});$
 else
 $k_0 = \text{mod}(\bar{k}_0, N_{cb});$
 end if

5

算法 7-h:

$\bar{k}_0 = k'_0 + \text{mod}(G, N_{cb} - (K_{NULL} - K'_{NULL}));$
 if $(k'_0 < K'_{NULL} - 2Z_c \text{ and } \bar{k}_0 \geq K'_{NULL} - 2Z_c)$ or $(\bar{k}_0 \geq K'_{NULL} - 2Z_c + N_{cb})$
 $k_0 = \text{mod}(\bar{k}_0 + K_{NULL} - K'_{NULL}, N_{cb});$
 else
 $k_0 = \text{mod}(\bar{k}_0, N_{cb});$
 end if

可能的实现方式六:

- 第一计数器 j 和第二计数器 k 置零; 应理解, 第一计数器用于确定一次 TBoMS 重复的 N 个时间单元中的除第一时间单元以外的其他时间单元的起始比特 k_0 , 第二计数器用于计数选择出来的编码比特个数;

根据 E_r 或 G 值, 和当前时间单元的上一个时间单元的起始比特索引 k_0' , 依次逐个遍历循环缓存器中存储的编码比特, 确定当前时间单元的起始比特索引 k_0 。

步骤 1: 从循环缓存器中读取索引为 $\text{mod}(k_0' + j, N_{cb})$ 的比特;

- 步骤 2: 当所读取的比特不为填充比特时, 第二计数器加 1, 即 $k = k + 1$; 否则, 保持第二计数器不变, 跳过此步骤;

步骤 3: 第一计数器加 1, 即 $j = j + 1$;

步骤 4: 判断第二计数器 k 是否小于 E_r 或 G 值; 如果是, 返回步骤 1 继续执行; 否则, 执行步骤 5;

步骤 5: 当前时间单元的起始比特的索引为 $k_0 = \text{mod}(k_0' + j, N_{cb})$ 。

- 下面介绍 2 种可能的算法。

算法 8-a:

```

j = 0 ;
k = 0 ;
while k < Er
    if d(k0' + j) mod Ncb ≠ NULL
        k = k + 1;
    end if
    j = j + 1;
end while
k0 = (k0' + j) mod Ncb;

```

算法 8-b:

```

j = 0 ;
k = 0 ;
while k < G
    if d(k0' + j) mod Ncb ≠ NULL
        k = k + 1;
    end if
    j = j + 1;
end while
k0 = (k0' + j) mod Ncb;

```

其中 d 表示循环缓存器中存储的经过 LDPC 编码后的码块。

可能的实现方式七:

- 根据 E_r 或 G 值, 和上一个时间单元的起始比特索引 k_0' , 确定一次 TBoMS 重复的 N 个时间单元中的除第一时间单元以外的其他时间单元的起始比特索引 k_0 。

示例 1, k_0' 和 k_0 的间隔为 E_r 或 G 值; 示例性的, $k_0 = \text{mod}(k_0' + E_r, N_{cb})$ 或

$$k_0 = \text{mod}(k_0' + G, N_{cb})。$$

- 示例 2, $k_{r,n} = \text{mod}(k_0 + n \times E_r, N_{cb})$ 或 $k_{r,n} = \text{mod}(k_0 + n \times G, N_{cb})$, $n = 1, 2, \dots, N-1$ 。需要注意的是, 示例 2 中的 k_0 表示分配给一次 TBoMS 重复的 N 个时间单元中的第一个时间单元 (时间单元#0) 对应的速率匹配的比特选择的起始比特索引, 是根据 RV 和 BG 确定的, $k_{r,n}$ 表示第 r 个码块时间单元# n (时间单元# n , $n = 1, 2, \dots, N-1$) 对应的速率匹配的比特选择的起始比特索引, r 为码块的索引, $r = 0, 1, \dots, C-1$, $n = 0, 1, \dots, N-1$, 其中 C 为码块个数。

可能的实现方式八:

- 根据 k_0 , 以及, E_r 或 G 值, 确定 $k_{r,n}$, $n = 1, 2, \dots, N-1$, n 表示时间单元的索引。其中, k_0 表示分配给一次 TBoMS 重复的 N 个时间单元中的第一个时间单元 (时间单元#0) 对应的速率匹配的比特选择的起始比特索引, 是根据 RV 和 BG 确定的, $k_{r,n}$ 表示第 r 个码块时间单元# n (时间单元# n , $n = 1, 2, \dots, N-1$) 对应的速率匹配的比特选择的起始比特索引, r 为码块的索引, $r = 0, 1, \dots, C-1$, $n = 0, 1, \dots, N-1$, 其中 C 为码块个数。

步骤 a,

- 计算时间单元# n 对应的时间单元之前的所有时间单元上的比特选择长度之和, 即时间单元#0 至时间单元# $n-1$, 共 n 个时间单元对应的速率匹配的比特选择的长度之和。

示例性的, 假设每个时间单元的比特选择的长度相等, 即 E_r 或 G , 则时间单元#0 至时间单元# $n-1$, 共 n 个时间单元对应的速率匹配的比特选择的长度之和为 $n \times E_r$ 或 $n \times G$;

步骤 b,

- 确定比特选择的真实周期 N_{cb}' 。

示例性的, $N_{cb}' = N_{cb} - (K_{NULL} - K_{NULL}')$ 。

步骤 c,

将 $n \times E_r$ 或 $n \times G$ 映射到一个比特选择的真实周期 N_{cb}' 内, 记作 L_n 。

L_n 的表示方式包括但不限于以下几种示例:

- $L_n = \text{mod}(n \times E_r, N_{cb} - (K_{NULL} - K_{NULL}'))$, 或者,
 $L_n = \text{mod}(n \times E_r, N_{cb}')$, 或者,
 $L_n = \text{mod}(n \times G, N_{cb} - (K_{NULL} - K_{NULL}'))$, 或者,
 $L_n = \text{mod}(n \times G, N_{cb}')$ 。

步骤 d,

- 假设没有填充比特, 根据 L_n 和 k_0 , 确定时间单元# n 的起始比特索引 $\bar{k}_n$ (中间变量)。

$\bar{k}_n$ 的表示方式包括但不限于以下几种示例:

- $\bar{k}_n = k_0 + L_n$, 或者,
 $\bar{k}_n = k_0 + \text{mod}(n \times E_r, N_{cb} - (K_{NULL} - K_{NULL}'))$, 或者,
 $\bar{k}_n = k_0 + \text{mod}(n \times E_r, N_{cb}')$, 或者,
 $\bar{k}_n = k_0 + \text{mod}(n \times G, N_{cb} - (K_{NULL} - K_{NULL}'))$, 或者,
 $\bar{k}_n = k_0 + \text{mod}(n \times G, N_{cb}')$ 。

步骤 e,

根据 k_0' , $\bar{k}_0$ 和填充比特在循环缓存器中的位置, 确定时间单元#n 的起始比特索引 $k_{r,n}$ 。

首先, 根据 k_0' , $\bar{k}_0$ 和填充比特在循环缓存器中的位置, 判断比特选择长度为 L_n 的比特选择过程是否经过填充比特。下面分别介绍几种不同的情况。

一, 如果比特选择长度为 L_n 的比特选择过程经过填充比特, 根据填充比特数量

5 $K_{NULL} - K_{NULL}'$ 对 $\bar{k}_n$ 进行修正, 即 $k_{r,n} = \bar{k}_n + K_{NULL} - K_{NULL}'$ 。

例如, 当 $k_0 < K_{NULL}' - 2Z_c$ 且 $\bar{k}_n \geq K_{NULL}' - 2Z_c$ 时, 比特选择过程经过填充比特, 故 $k_{r,n} = \text{mod}(\bar{k}_n + K_{NULL} - K_{NULL}', N_{cb})$ 。

再例如, 当 $k_0 \geq K_{NULL}' - 2Z_c$ 且 $\bar{k}_n \geq K_{NULL}' - 2Z_c + N_{cb}$ 时 (或者, 此条件可以简化为: 当 $\bar{k}_n \geq K_{NULL}' - 2Z_c + N_{cb}$ 时), 比特选择过程经过填充比特, $k_{r,n} = \text{mod}(\bar{k}_n + K_{NULL} - K_{NULL}', N_{cb})$ 。

10 二, 如果比特选择长度为 L_n 的比特选择过程没有经过填充比特, 则根据 $\bar{k}_n$ 得到 $k_{r,n}$ 。

例如, 当 $k_0 < K_{NULL}' - 2Z_c$ 且 $\bar{k}_n < K_{NULL}' - 2Z_c$, (或者, 此条件可以简化为: 当 $\bar{k}_n < K_{NULL}' - 2Z_c$ 时), 比特选择过程没有经过填充比特, 故 $k_{r,n} = \text{mod}(\bar{k}_n, N_{cb})$ 或 $k_{r,n} = \bar{k}_n$ 。

再例如, 当 $k_0 \geq K_{NULL}' - 2Z_c$ 且 $\bar{k}_n < K_{NULL}' - 2Z_c + N_{cb}$, 比特选择过程没有经过填充比特, 故 $k_{r,n} = \text{mod}(\bar{k}_n, N_{cb})$ 或 $k_{r,n} = \bar{k}_n$ 。

15 可选的, 如果 $k_{r,n}$ 指示的起始比特为填充比特, 即 $K_{NULL}' - 2Z_c \leq k_{r,n} < K_{NULL} - 2Z_c$, 则

根据 $K_{NULL} - 2Z_c$ 和 k_0 对所述 $\bar{k}_n$ 进行修正, 获得 $k_{r,n}$ 。示例性的,

$k_{r,n} = \text{mod}(\bar{k}_n + K_{NULL} - 2Z_c - k_0, N_{cb})$ 或 $k_{r,n} = \bar{k}_n + K_{NULL} - 2Z_c - k_0$ 。

下面给出可能的实现方式八的几种算法示例。

算法 9-a:

$$\bar{k}_n = k_0 + \text{mod}(n \times E_r, N_{cb} - (K_{NULL} - K'_{NULL}));$$

$$\text{if } (k_0 < K'_{NULL} - 2Z_c \text{ and } \bar{k}_n \geq K'_{NULL} - 2Z_c) \text{ or } (k_0 > K_{NULL} - 2Z_c \text{ and } \bar{k}_n \geq K_{NULL}' - 2Z_c + N_{cb})$$

$$\bar{k}_n = \bar{k}_n + K_{NULL} - K'_{NULL};$$

$$\text{end if}$$

$$k_{r,n} = \text{mod}(\bar{k}_n, N_{cb});$$

20

算法 9-b:

$$\bar{k}_n = k_0 + \text{mod}(n \times E_r, N_{cb} - (K_{NULL} - K'_{NULL}));$$

$$\text{if } (k_0 < K'_{NULL} - 2Z_c \text{ and } \bar{k}_n \geq K'_{NULL} - 2Z_c) \text{ or } (\bar{k}_n \geq K'_{NULL} - 2Z_c + N_{cb})$$

$$\bar{k}_n = \bar{k}_n + K_{NULL} - K'_{NULL};$$

$$\text{end if}$$

$$k_{r,n} = \text{mod}(\bar{k}_n, N_{cb});$$

算法 9-c:

$$\bar{k}_n = k_0 + \text{mod}(n \times E_r, N_{cb} - (K_{NULL} - K'_{NULL}));$$

$$\text{if } (k_0 < K'_{NULL} - 2Z_c \text{ and } \bar{k}_n \geq K'_{NULL} - 2Z_c) \text{ or } (k_0 > K_{NULL} - 2Z_c \text{ and } \bar{k}_n \geq K_{NULL}' - 2Z_c + N_{cb})$$

$$k_{r,n} = \text{mod}(\bar{k}_n + K_{NULL} - K'_{NULL}, N_{cb});$$

$$\text{else}$$

$k_{r,n} = \text{mod}(\bar{k}_n, N_{cb});$
end if

算法 9-d:

$\bar{k}_n = k_0 + \text{mod}(n \times E_r, N_{cb} - (K_{NULL} - K'_{NULL}));$
if $(k_0 < K'_{NULL} - 2Z_c \text{ and } \bar{k}_n \geq K'_{NULL} - 2Z_c)$ or $(\bar{k}_n \geq K'_{NULL} - 2Z_c + N_{cb})$
 $k_{r,n} = \text{mod}(\bar{k}_n + K_{NULL} - K'_{NULL}, N_{cb});$
else
 $k_{r,n} = \text{mod}(\bar{k}_n, N_{cb});$
end if

算法 9-e:

$\bar{k}_n = k_0 + \text{mod}(n \times G, N_{cb} - (K_{NULL} - K'_{NULL}));$
if $(k_0 < K'_{NULL} - 2Z_c \text{ and } \bar{k}_n \geq K'_{NULL} - 2Z_c)$ or $(k_0 > K_{NULL} - 2Z_c \text{ and } \bar{k}_n \geq K_{NULL}' - 2Z_c + N_{cb})$
 $\bar{k}_n = \bar{k}_n + K_{NULL} - K'_{NULL};$
end if
 $k_{r,n} = \text{mod}(\bar{k}_n, N_{cb});$

算法 9-f:

$\bar{k}_n = k_0 + \text{mod}(n \times G, N_{cb} - (K_{NULL} - K'_{NULL}));$
if $(k_0 < K'_{NULL} - 2Z_c \text{ and } \bar{k}_n \geq K'_{NULL} - 2Z_c)$ or $(\bar{k}_n \geq K'_{NULL} - 2Z_c + N_{cb})$
 $\bar{k}_n = \bar{k}_n + K_{NULL} - K'_{NULL};$
end if
 $k_{r,n} = \text{mod}(\bar{k}_n, N_{cb});$

算法 9-g:

$\bar{k}_n = k_0 + \text{mod}(n \times G, N_{cb} - (K_{NULL} - K'_{NULL}));$
if $(k_0 < K'_{NULL} - 2Z_c \text{ and } \bar{k}_n \geq K'_{NULL} - 2Z_c)$ or $(k_0 > K_{NULL} - 2Z_c \text{ and } \bar{k}_n \geq K_{NULL}' - 2Z_c + N_{cb})$
 $k_{r,n} = \text{mod}(\bar{k}_n + K_{NULL} - K'_{NULL}, N_{cb});$
else
 $k_{r,n} = \text{mod}(\bar{k}_n, N_{cb});$
end if

5

算法 9-h:

$\bar{k}_n = k_0 + \text{mod}(n \times G, N_{cb} - (K_{NULL} - K'_{NULL}));$
if $(k_0 < K'_{NULL} - 2Z_c \text{ and } \bar{k}_n \geq K'_{NULL} - 2Z_c)$ or $(\bar{k}_n \geq K'_{NULL} - 2Z_c + N_{cb})$
 $k_{r,n} = \text{mod}(\bar{k}_n + K_{NULL} - K'_{NULL}, N_{cb});$
else
 $k_{r,n} = \text{mod}(\bar{k}_n, N_{cb});$
end if

可能的实现方式九:

对可能的实现方式三中的算法 4 进行改进:

算法 4-a:

```

for n = 0 to N-1
  if n = 0
    if  $K'_{NULL} - 2Z_c \leq k_0 < K_{NULL} - 2Z_c$ 
       $k_{r,0} = K_{NULL} - 2Z_c$ ;
    else
       $k_{r,0} = k_0$ ;
    end if
  else
     $k_{r,n} = k_{r,n-1} + \text{mod}(E_{r,n-1}, N_{cb} - (K_{NULL} - K'_{NULL}));$ 
    if ( $k_{r,n-1} < K'_{NULL} - 2Z_c$  and  $k_{r,n} \geq K'_{NULL} - 2Z_c$ ) or
      ( $k_{r,n} \geq K'_{NULL} - 2Z_c + N_{cb}$ )
       $k_{r,n} = k_{r,n} + K_{NULL} - K'_{NULL}$ ;
    end if
     $k_{r,n} = \text{mod}(k_{r,n}, N_{cb})$ ;
  end if
end for

```

应理解，RV 指示的起始比特索引可能落在填充比特上，此时起始比特应当为最后一个填充比特的下一个比特。

可能的实现方式十：

对可能的实现方式四种的算法 5 进行改进

算法 5-a:

```

for n = 0 to N-1
  if n = 0
    if  $K'_{NULL} - 2Z_c \leq k_0 < K_{NULL} - 2Z_c$ 
       $k_{r,0} = K_{NULL} - 2Z_c$ ;
    else
       $k_{r,0} = k_0$ ;
    end if;
  else
     $k_{r,n} = k_{r,n-1} + \text{mod}(E_r, N_{cb} - (K_{NULL} - K'_{NULL}));$ 
    if ( $k_{r,n-1} < K'_{NULL} - 2Z_c$  and  $k_{r,n} \geq K'_{NULL} - 2Z_c$ ) or
      ( $k_{r,n} \geq K'_{NULL} - 2Z_c + N_{cb}$ )
       $k_{r,n} = k_{r,n} + K_{NULL} - K'_{NULL}$ ;
    end if
     $k_{r,n} = \text{mod}(k_{r,n}, N_{cb})$ ;
  end if
end for

```

5 应理解，RV 指示的起始比特索引可能落在填充比特上，此时起始比特应当为最后一个填充比特的下一个比特。

可能的实现方式十一：

算法 10-a 至算法 10-d 的区别在于，对于每个时隙上 G 不同（为 G_n ），可以使用算法

10-a 或 10-c, 对于每个时隙上 G 相同, 可以使用算法 10-b 或 10-d。

算法 10-a:

```

j = 0;
kr,0 = k0;
for n = 1 to N-1
    k = 0;
    while k < Er,n-1
        if d(kr,0+j)modNcb ≠ NULL
            k = k + 1;
        end if
        j = j + 1;
    end while
    kr,n = (kr,0 + j)modNcb;
end for

```

5 应理解, 对于算法 10-a 或 10-c, 时隙#n、码块#r 上的速率匹配比特选择的起始比特 k_{r,n} 是时隙#n-1、码块#r 上的速率匹配以比特选择长度 E_{r,n-1} 的比特选择的结束比特的下一个比特。需要说明的是, 这里的所述的比特选择仅用于确定起始比特, 并不是比特选择输出的比特序列长度, 比特选择输出的比特序列长度由步骤四中使用的比特选择长度为准。

算法 10-b:

```

j = 0;
kr,0 = k0;
for n = 1 to N-1
    k = 0;
    while k < Er
        if d(kr,0+j)modNcb ≠ NULL
            k = k + 1;
        end if
        j = j + 1;
    end while
    kr,n = (kr,0 + j)modNcb;
end for

```

算法 10-c:

```

kr,0 = k0;
for n = 1 to N-1
    j = 0;
    k = 0;
    while k < Er,n-1
        if d(kr,n-1+j)modNcb ≠ NULL
            k = k + 1;
        end if
        j = j + 1;
    end while

```

```

 $k_{r,n} = (k_{r,n-1} + j) \bmod N_{cb};$ 
end for

```

算法 10-d:

```

 $k_{r,0} = k_0;$ 
for n = 1 to N-1
     $j = 0;$ 
     $k = 0;$ 
    while  $k < E_{r,n-1}$ 
        if  $d_{(k_{r,n-1}+j) \bmod N_{cb}} \neq NULL$ 
             $k = k + 1;$ 
        end if
         $j = j + 1;$ 
    end while
     $k_{r,n} = (k_{r,n-1} + j) \bmod N_{cb};$ 
end for

```

本申请的另一个实施例，提供一种上行传输的方法，包括：

- 5 终端设备接收来自网络设备的第一指示信息，所述第一指示信息用于指示所述终端设备在多个时间单元上发送第一传输块；

所述终端设备接收来自所述网络设备的第二指示信息，所述第二指示信息用于指示所述终端设备在第一时间单元上发送上行控制信息 UCI，所述第一时间单元为所述多个时间单元中的一个；

- 10 所述终端设备根据所述第一时间单元对应的参数 G 确定第一编码比特序列的起始比特在循环缓存中的位置，所述第一编码比特序列为第二时间单元内经过速率匹配输出的编码比特序列，所述参数 G 为一个时间单元内可用于传输所述第一传输块和所述 UCI 的编码比特总数，G 为正整数，所述第二时间单元为所述多个时间单元中的一个；

所述终端设备在所述多个时间单元上向所述网络设备发送所述 UCI 和所述第一传输块。

- 15 该实施例与图 8 的实施例的区别在于，所述第二时间单元为所述多个时间单元中的一个。例如，所述第二时间单元为第一时间单元的下一个时间单元，或者所述第二时间单元为所述多个时间单元中的第一个，或者所述第二时间单元为所述多个时间单元中的任意一个。本实施例中的其他相关内容可以参照前面实施例的描述，在此不做赘述。

一种实现方式中，所述多个时间单元对应的参数 G 相等。

- 20 一种实现方式中，终端设备根据所述第一时间单元对应的参数 G 确定第一编码比特序列的起始比特在循环缓存中的位置，包括：

根据所述第一时间单元对应的参数 G 和比特选择的真实周期 N'_{cb} ，确定所述第一编码比特序列的起始比特在循环缓存中的位置。

- 25 例如，根据所述第一时间单元对应的参数 G 和比特选择的真实周期 N'_{cb} ，确定第一起始比特索引 $k'_{r,n}$ ；

根据所述第一起始比特索引 $k'_{r,n}$ 确定所述第一编码比特序列的起始比特在循环缓存中的位置 $k_{r,n}$ 。

下面结合具体例子,描述第一起始比特索引 $k'_{r,n}$ 及第一编码比特序列的起始比特在循环缓存中的位置 $k_{r,n}$ 的确定。

分配给一次 TBoMS 传输的时间单元个数为 N , TBS 计算缩放参数 $K=N$, RV 更新周期也为 N 。应理解,如果没有使能 TBoMS 重复,则 TBoMS 传输占用 N 个时间单元,使用一个 RV,该使用的 RV 由高层信令指示;如果使能 TBoMS 重复, M 表示 TBoMS 重复次数,每次 TBoMS 重复占用 N 个时间单元,故 M 次 TBoMS 重复共占用 $M \times N$ 个时间单元,每次 TBoMS 重复的第一个时间单元更新 RV,更新顺序由高层信令指示。

那么,一次 TBoMS 重复的每个时间单元执行的速率匹配的比特选择的起始比特索引(或起始比特位置) $k_{r,n}$ 可以通过以下方式确定,其中 n 表示时间单元的索引, $n = 0, 1, \dots, N-1$, r 表示码块索引, $r = 0, 1, \dots, C-1$, C 表示码块个数。

(1) 确定比特选择的真实周期 N'_{cb} :

应理解, i) 比特选择过程中,会将循环缓存器中的数据依次逐个读取。由于循环缓存器是周期性的(周期为 N_{cb}),故比特选择过程选择出来的比特也是周期性的,这里将比特选择过程选择出来的比特的周期称作比特选择的真实周期, N'_{cb} 。 ii) 因为填充比特不会被比特选择过程选择出来,故比特选择的真实周期小于循环缓存器的长度(周期) N_{cb} 。

示例性的, $N'_{cb} = N_{cb} - (K_{NULL} - K'_{NULL})$ 。其中, $K_{NULL} - K'_{NULL}$ 表示循环缓存器中填充比特的个数。

(2) 对于一次 TBoMS 重复的 N 个时间单元中的第一个时间单元

根据 RV 和 BG 确定起始比特索引 k_0 ,具体确定方式可以参见本申请中表 2 对应的描述。

将所述起始比特索引 k_0 进行修正,修正后的起始比特索引记作 $k'_{r,0}$,示例性的:

如果起始比特索引 k_0 指示的起始比特为系统比特(systematic bits),即修正前, $0 \leq k_0 < K'_{NULL} - 2Z_c$,则不需要修正,即 $k'_{r,0} = k_0$;其中, $K'_{NULL} - 2Z_c$ 表示循环缓存器中第一个填充比特的索引;

如果起始比特索引 k_0 指示的起始比特为填充比特(filler bits),即修正前, $K'_{NULL} - 2Z_c \leq k_0 < K_{NULL} - 2Z_c$,则修正后, $k'_{r,0} = K'_{NULL} - 2Z_c$;其中, $K_{NULL} - 2Z_c$ 表示循环缓存器中第一个校验比特的索引;

如果起始比特索引 k_0 指示的起始比特为校验比特(parity bits),即修正前, $K_{NULL} - 2Z_c \leq k_0 < N_{cb}$,则修正后, $k'_{r,0} = k_0 - (K_{NULL} - K'_{NULL})$ 。

上述示例也可以写作算法 11-a 的形式。

算法 11-a:

```

if  $0 \leq k_0 < K'_{NULL} - 2Z_c$ 
     $k'_{r,0} = k_0$ ;
elseif  $K'_{NULL} - 2Z_c \leq k_0 < K_{NULL} - 2Z_c$ 
     $k'_{r,0} = K'_{NULL} - 2Z_c$ ;
else
     $k'_{r,0} = k_0 - (K_{NULL} - K'_{NULL})$ ;
end

```

应理解,所述起始比特索引 k_0 指示了编码比特在长度为 N_{cb} 的循环缓存器中的位置,

所述起始比特索引 $k'_{r,0}$ 指示了编码比特在长度为 N'_{cb} 的循环缓存器中的位置,且 k_0 和 $k'_{r,0}$ 指示的是同一个编码比特。其中,长度为 N'_{cb} 的循环缓存器是长度为 N_{cb} 的循环缓存器扣除填充比特序列后得到的。所述长度为 N'_{cb} 的循环缓存器是虚拟的,仅用于理解所述实现方式。

示例性的,长度为 N_{cb} 的循环缓存器中存储的编码比特序列如图 28 (a) 所示,其中系统比特的索引为 0 至 $K'_{NULL} - 2Z_c - 1$,填充比特的索引为 $K'_{NULL} - 2Z_c$ 至 $K_{NULL} - 2Z_c - 1$,校验比特的索引为 $K_{NULL} - 2Z_c$ 至 $N_{cb} - 1$;长度为 N'_{cb} 的循环缓存器中存储的编码比特序列如图 28 (b) 所示,其中系统比特的索引为 0 至 $K'_{NULL} - 2Z_c - 1$,校验比特的索引为 $K'_{NULL} - 2Z_c$ 至 $N'_{cb} - 1$;图 28 (a) 中索引为 0 至 $K'_{NULL} - 2Z_c - 1$ 对应的编码比特序列与图 28 (b) 中索引为 0 至 $K'_{NULL} - 2Z_c - 1$ 对应的编码比特序列相同,图 28 (a) 中索引为 $K_{NULL} - 2Z_c$ 至 $N_{cb} - 1$ 对应的编码比特序列与图 28 (b) 中索引为 $K'_{NULL} - 2Z_c$ 至 $N'_{cb} - 1$ 对应的编码比特序列相同。

(3) 对于一次 TBoMS 重复的 N 个时间单元中的除第一时间单元以外的其他时间单元

该其它时间单元对应的速率匹配的比特选择的起始比特索引 $k'_{r,n}$, $n = 1, 2, \dots, N - 1$, 由以下传输中的至少一项确定:

- 1、第一个时间单元的速率匹配的比特选择的起始比特索引 $k'_{r,0}$;
- 2、比特选择的真实周期 N'_{cb} ;
- 3、上一个时间单元对应的 G 值;
- 4、根据步骤二确定的上一个时间单元对应的速率匹配输出的比特长度 E_r (E_r 由上一个时间单元对应的值 G 确定);
- 5、第一个时间单元对应的 G 值;
- 6、根据步骤二确定的第一个时间单元对应的速率匹配输出的比特长度 E_r (由第一个时间单元对应的 G 值确定);
- 7、一次 TBoMS 重复的 N 个时间单元中任一时间单元对应的 G 值;
- 8、根据步骤二确定的一次 TBoMS 重复的 N 个时间单元中任一时间单元对应的速率匹配输出的比特长度 N_{cb} (由该任一时间单元对应的 G 值确定)。

应理解,所述 $k'_{r,n}$ 是指编码比特在长度为 N'_{cb} 的循环缓存器中的位置。

示例性地,当码块数量 $C = 1$ 时, $E = E_r = G$, $r = 0$,故此时可以直接根据 G 值确定 $k'_{r,n}$;当 $C > 1$ 时, $E_r \neq G$,需根据 E_r 确定 $k'_{r,n}$ 。其中, $G \equiv N_{RE} \times Q_m \times N_L$, $N_{RE} = \min(156, N'_{RE})n_{PRB}$, n_{PRB} 表示分配的总 PRB 个数,其中, $N'_{RE} = N_{sc}^{RB} \cdot N_{symp}^{sh} - N_{DMRS}^{PRB} - N_{oh}^{PRB}$,其中 $N_{sc}^{RB} = 12$ 是物理资源块中频域的子载波数, N_{symp}^{sh} 是 PUSCH 分配的符号数 L , N_{DMRS}^{PRB} 是分配的持续时间内每个 PRB 的 DM-RS 的 RE 数,包括 DM-RS CDM 组没有数据, N_{oh}^{PRB} 是 PUSCH-ServingCellConfig 中高层参数 xOverhead 配置的开销, r 为码块的索引, $r = 0, 1, \dots, C - 1$, $n = 1, 2, \dots, N - 1$,其中 C 为码块个数;特别地,层数 N_L 等于 1, 则 $G \equiv N_{RE} \times Q_m$ 。

具体步骤如下:

步骤 a,

计算时间单元#n 对应的时间单元之前的所有时间单元上的比特选择长度之和, 即时间单元#0 至时间单元#n-1, 共n个时间单元对应的速率匹配的比特选择的长度之和。

示例性的, 假设每个时间单元的比特选择的长度相等, 即 E_r 或 G , 则时间单元#0 至时间单元#n-1, 共 n 个时间单元对应的速率匹配的比特选择的长度之和为 $n \times E_r$ 或 $n \times G$;

步骤 b,

根据 $n \times E_r$ 或 $n \times G$ 和 $k'_{r,0}$, 确定在长度为 N'_{cb} 的循环缓存器中, 时间单元#n 对应的速率匹配的比特选择的起始比特的索引 $k'_{r,n}$ 。

示例性的: $k'_{r,n} = \text{mod}(k'_{r,0} + n \times G, N'_{cb})$, 或者, $k'_{r,n} = \text{mod}(k'_{r,0} + n \times E_r, N'_{cb})$, 其中 mod 表示取模运算。

应理解, 循环缓存器的寻址方式是取模运算。

(4) 根据起始比特索引 $k'_{r,n}$ 确定起始比特索引 $k_{r,n}$, $n = 0, 1, \dots, N-1$, 示例性的:

如果起始比特索引 $k'_{r,n}$ 指示的起始比特为系统比特, 即修正前, $0 \leq k'_{r,n} < K'_{NULL} - 2Z_c$, 则不需要修正, 即, $k_{r,n} = k'_{r,n}$ 。

如果起始比特索引 $k'_{r,n}$ 指示的起始比特为校验比特, 即修正前, $K'_{NULL} - 2Z_c \leq k'_{r,n} < N'_{cb}$, 则修正后, $k_{r,n} = k'_{r,n} + (K_{NULL} - K'_{NULL})$ 。

应理解, 起始比特索引 $k'_{r,n}$ 指示的是编码比特在长度为 N'_{cb} 的循环缓存器中的位置, 故需要将其修正到其所指示的编码比特在长度为 N_{cb} 的循环缓存器中的位置, 即起始比特索引 $k_{r,n}$ 。由于填充比特插入在了系统比特和校验比特之间, 故起始比特索引 $k'_{r,n}$ 指示的起始

比特为校验比特, 需要将填充比特数 $K_{NULL} - K'_{NULL}$ 考虑到起始比特索引 $k_{r,n}$ 的计数中。

上述示例也可以写作算法 11-b 的形式。

算法 11-b: $n = 0, 1, \dots, N-1$

if $0 \leq k'_{r,n} < K'_{NULL} - 2Z_c$

$k_{r,n} = k'_{r,n};$

else

$k_{r,n} = k'_{r,n} + (K_{NULL} - K'_{NULL});$

end

可选的, 对于时间单元#0 对应的速率匹配的比特选择的起始比特索引 $k_{r,0}$, 可以根据 k_0 直接确定: $k_{r,0} = k_0$ 。(其它时间单元对应的速率匹配选择的起始比特索引 $k_{r,n}$, $n = 1, 2, \dots, N-1$, 仍然根据算法 11-b 确定。)

步骤四, 终端设备根据起始比特位置和比特长度执行比特选择, 确定输出的比特序列。

所述终端设备根据所述起始比特位置和所述比特长度执行比特选择, 确定第一比特序列。

可选的,所述终端设备根据时间单元#n上的码块#r的速率匹配比特选择的起始比特位置 $k_{r,n}$ 和时间单元#n上的码块#r的速率匹配输出的比特长度 $E_{r,n}$,确定第一比特序列。示例性的,如算法6所示,其中d表示循环缓存器中存储的码块,e表示所述第一比特序列。

算法 6:

```

k=0;
j=0;
while k < Er,n
    if d(kr,n+j)modNcb ≠ NULL
        ek = d(kr,n+j)modNcb;
        k=k+1;
    end if
    j=j+1;
end if
end while

```

5 或者,上述步骤中的 $E_{r,n}$ 还可以替换为 E_r , $d_{k_{r,n}}$ 可以替换为 d_k ;

或者,上述步骤中的 $E_{r,n}$ 还可以替换为 E_r' , $d_{k_{r,n}}$ 可以替换为 d_k 。

或者,上述步骤中的 $E_{r,n}$ 还可以替换为 $E_{r,n}'$ 。

其中,上述步骤中的 $E_{r,n}$,以及 $E_{r,n}$ 分别替换为 E_r 、 E_r' 、 $E_{r,n}'$ 的情况分别与前述步骤二和步骤三中不同的方案对应,具体如下:

10 在步骤二中,根据上行传输的方法中的参数G计算速率匹配输出的比特序列的长度,记作 E_i ,在步骤三中,根据 E_i 确定速率匹配输出的比特序列的起点,在步骤四中,根据起始比特位置和比特长度 E_i 执行比特选择,确定输出的比特序列;

或者,在步骤二中,根据在时间单元#n上传输TB的编码比特总数 G_n ,确定时间单元#n上码块#r的速率匹配输出的比特长度,记作 $E_{r,n}$,在步骤三中,根据 $E_{r,n}$ 确定时间单元#n上速率匹配输出的比特序列的起点,在步骤四中,根据起始比特位置和比特长度 $E_{r,n}$ 执行比特选择,确定输出的比特序列;

15 或者,在步骤二中,根据参数G'计算速率匹配输出的比特序列的长度 E_i' ,在步骤三中,根据上行传输的方法中的参数G计算速率匹配输出的比特序列的长度,记作 E_i ,根据 E_i 确定速率匹配输出的比特序列的起点,在步骤四中,根据起始比特位置和比特长度 E_i' 执行比特选择,确定输出的比特序列;

20 或者,在步骤二中,根据在时间单元#n上传输TB的编码比特总数 G_n' ,确定时间单元#n上码块#r的速率匹配输出的比特长度,记作 $E_{r,n}'$,在步骤三中,根据上行传输的方法中的参数 G_n 计算速率匹配输出的比特序列的长度,记作 $E_{r,n}$,根据 $E_{r,n}$ 确定时间单元#n上速率匹配输出的比特序列的起点;在步骤四中,根据起始比特位置和比特长度 $E_{r,n}'$ 执行比特选择,确定输出的比特序列。

25 可选的,所述终端设备根据时间单元#n上的码块#r的速率匹配比特选择的起始比特位置 k_0 和时间单元#n上的码块#r的速率匹配输出的比特长度 $E_{r,n}$,确定第一比特序列。示例性的,如算法6-a所示,其中d表示循环缓存器中存储的码块,e表示所述第一比特序

列。

算法 6-a:

```

k=0;
j=0;
while k < E
    if  $d_{(k_0+j)\bmod N_{cb}} \neq \text{NULL}$ 
         $e_k = d_{(k_0+j)\bmod N_{cb}}$ ;
        k=k+1;
    end if
    j=j+1;
end while

```

其中，所述终端设备根据时间单元#n 上的码块#r 的速率匹配比特选择的起始比特位置 k_0 是根据步骤三中可能的实现方式五和六、以及可能的实现方式七中的示例 1 得到的，E 表示时间单元#n 上的码块#r 的速率匹配比特选择的比特选择长度，即 $E_{r,n}$ 、 E_r 、 E_r' 或 $E_{r,n}'$ 。

可选地，对于上述不同的方案中，每个时间单元执行的速率匹配的比特选择的起始比特索引（或起始比特位置） k_0 和比特选择长度 E_r 均由 G 确定的情况。还可以将输出的长度为 E_r 的比特序列进一步处理（例如下面提到的对输出的比特序列进行截断的方案）后进行比特交织等后续处理。

分配给一次 TBoMS 传输的时间单元个数为 N，TBS 计算缩放参数 $K=N$ ，RV 更新周期也为 N。应理解，如果没有使能 TBoMS 重复，则 TBoMS 传输占用 N 个时间单元，使用一个 RV，该使用的 RV 由高层信令指示；如果使能 TBoMS 重复，M 表示 TBoMS 重复次数，每次 TBoMS 重复占用 N 个时间单元，故 M 次 TBoMS 重复共占用 $M \times N$ 个时间单元，每次 TBoMS 重复的第一个时间单元更新 RV，更新顺序由高层信令指示。

步骤 1，

目标时间单元的上一个时间单元的起始比特索引为 k_0' ，比特选择长度为 E_r 或G值。示例性地，当码块数量 $C=1$ 时， $E = E_r = G$ ， $r=0$ ，故此时可以直接根据 G 值确定 k_0 ；当 $C>1$ 时， $E_r \neq G$ ，需根据 E_r 确定 k_0 。其中， $G \equiv N_{RE} \times Q_m \times N_L$ ；特别地，层数 N_L 等于 1，则 $G \equiv N_{RE} \times Q_m$ 。

步骤 2，

根据所述 k_0' 和 E_r 或G值执行比特选择过程，则目标时间单元的起始比特索引 k_0 为上一个时间单元比特选择的最后一位比特的下一位比特；

步骤 3，

根据 E_r' 或G'值，对上一个时间单元选择出来的比特序列进行截断，截取其中前 E_r' 或G'位比特，将截取得到的比特序列进行比特交织等后续处理。

示例性的，经过比特选择输出的比特序列为 $e_0, e_1, \dots, e_{E_r-1}$ （或 $e_0, e_1, \dots, e_{G-1}$ ），截断后的比特序列为 $e_0, e_1, \dots, e_{E_r'-1}$ （或 $e_0, e_1, \dots, e_{G'-1}$ ），然后，将截断后的比特序列 $e_0, e_1, \dots, e_{E_r'-1}$ 进行比特交织等后续操作。

步骤五，所述终端设备根据所述第一比特序列，依次执行比特交织和码块级联，输出所述 UL-SCH。

可选地,终端设备还根据调度信息生成 UCI。

应理解,所述终端设备仅在收到指示所述终端设备在所述 PUSCH 上承载所述 UCI 的信令时,才会执行此步骤。

S203,终端设备至少基于 UL-SCH 生成 PUSCH 并向网络设备发送该 PUSCH。

- 5 可能的情况一,终端设备基于 UL-SCH 生成 PUSCH。应理解,此时终端设备仅检测到发送 PUSCH 的信令,且网络设备没有发送指示终端设备在 PUSCH 上承载 UCI 的信令;或者网络设备发送指示终端设备在 PUSCH 上承载 UCI 的信令,但终端设备漏检信令。

示例性的,如图 12 中的(a)所示,此时循环缓存器中的比特序列按顺序依次由 PUSCH 承载。

- 10 可能的情况二,终端设备将 UCI 复用在 UL-SCH 上,生成 PUSCH。

应理解,此时终端设备检测到网络设备发送的指示终端设备发送 PUSCH 和在 PUSCH 上承载 UCI 的信令。

- 作为一个示例,终端设备在 S202 的步骤二中用 E_r 计算速率匹配输出的比特序列的长度的情况下,假设 UCI 复用在 PUSCH 的第二个时隙上,终端设备在该第二上行时隙内进行速率匹配。由于速率匹配比特选择的起点是根据前一个上行时隙的速率匹配的 E_r 确定,即根据 G 确定,而前一个上行时隙的速率匹配输出的比特长度是根据 G' 确定的 E_r' ,在该情况下,如图 12 中的(b)所示,由于 G' 需要扣除 UCI 的编码比特总数,因此 $G' < G$,因此第三个上行时隙的速率匹配比特选择的起始比特会比第二个上行时隙的比特选择的结束比特在循环缓存中的位置更靠后,两者之间中间会有间隙。显然,该间隙是由 UCI 占用分配给 PUSCH 的 RE 引起的。

- 20 作为另一个示例,终端设备在 S202 的步骤二中用 E_r 计算速率匹配输出的比特序列的长度的情况下,UCI 复用在 PUSCH 的第二个时隙上,终端设备在该第二上行时隙内进行速率匹配。由于速率匹配比特选择的起点是根据前一个上行时隙的速率匹配的 E_r 确定,即根据 G 确定,而前一个上行时隙的速率匹配输出的比特长度也是根据 G 确定的,在该情况下,如图 12 中的(a)所示,第三个上行时隙的速率匹配比特选择的起始比特为第二个上行时隙的比特选择的结束比特在循环缓存中的下一个比特。

应理解,比较图 12 中的(a)和图 12 中的(b),无论 PUSCH 上是否有 UCI 复用,每一次速率匹配比特选择的起点是一致的,即该起点与 UCI 复用无关。

S204,网络设备基于 PUSCH 确定 UL-SCH。

- 30 具体地,网络设备采用与 S202 中终端设备相同的方法确定速率匹配输出的比特序列的长度和起始比特位置,确定 UL-SCH 在循环缓存中的位置。

应理解,网络设备和终端设备使用相同的确定方法,能够保证网络设备和终端设备对 UL-SCH 在 PUSCH 中的位置理解一致,从而保证正确解析 UL-SCH。

- 35 示例性的,如果网络设备通过信令指示 UCI 和 UL-SCH 复用,但终端设备没有检测到信令,终端设备如图 12 中的(a)发送 PUSCH,网络设备如图 12 中的(b)确定 UL-SCH 和 UCI,由于网络设备和终端设备采用相同的方法提前确定 UL-SCH 在循环缓存器的起始比特,且计算 G 时没有将 UCI 排除在外,故从第三个时隙承载的比特开始网络设备都可以完整解析。

- 40 本申请实施例,网络设备指示终端设备在第一时间单元上承载 UCI 的指示信息,在终端设备检测或没有检测到该指示的情况下,网络设备和终端设备在确定第二时间单元(第

一时间单元的下一个时间单元)内经过速率匹配输出的编码比特序列的起始比特在循环缓存中的位置时使用的参数 G 都不扣除 UCI 的编码比特总数, 可以使得从第二时间单元开始网络设备和终端设备可以对齐每次速率匹配中比特选择的起始比特在循环缓存器中的位置, 从而提高译码成功率。

- 5 需要说明的是, 在 S201 中对调度信息的信令设计中提到, 如果缩放参数 K =时隙数 N , 可以理解为网络设备在 N 个时隙内不支持 TBoMS 重复, 时隙数仅表示分配给一次 TBoMS 传输的时隙数; 如果 $K < N$, 时隙数表示包括传输 TBoMS 重复在内的总时隙数, K 表示一次 TBoMS 传输的时隙数, 且若 N/K 为整数, 则 TBoMS 重复次数为 N/K 。而 S202 的步骤三中, 可能的实现方式一、三、四中的算法 3、4、5 中, 都将 $n=mK$ 作为一个判断条件, 其中, n 为速率匹配的次数, m 为一个整数, 当满足该条件时, $k_{r,n} = k_0$, 时间单元 n 上的码块# r 的速率匹配比特选择的起始比特位置 $k_{r,n}$ 由 k_0 决定, 即由 RV 指定。其中, 对于 $K = N$, $n=0$ 可以作为 $n=mK$ 的一种特殊情况, 此时 k_0 由 RV 指示; 对于 $1 < K < N$, 且 N/K 为整数情况, k_m 由 RV 指示, 其中 $m=N/K$ 。

可选地, 方法 200 还包括:

- 15 应理解, 假设网络设备指示终端设备发送 UCI, 并复用在第 b 个时隙上, 但是终端设备没有检测到该 PDCCH, 上述方案能够实现网络设备和终端设备可以对齐从第 b 个时隙往后的所有时隙上速率匹配输出的比特序列的起始比特。进一步地, 方法 200 还可以针对第 b 个时隙上的资源映射的方法进行进一步优化。

- 20 现有技术中, 将 UCI 复元至第 b 个时隙的部分时频资源时, 该部分时频资源原本承载的数据信息的比特会向后顺延。在终端设备漏检 PDCCH 时, 终端设备在第 b 个时隙发送的全是数据信息的比特, 而网络设备则认为上述部分时频资源承载的是 UCI, 第 b 个时隙上的除该部分时频资源以外的时频资源为数据信息的比特。这会导致网络设备在当前时隙无法正确检测 UL-SCH。

- 25 在 S202 中, 在步骤二中根据 G 计算速率匹配输出的比特序列的长度和步骤三中根据 G 计算速率匹配输出的比特序列的起始比特的位置在循环缓存的位置时, 终端设备将第二编码比特序列映射至第一时间单元上的时频资源, 第二比特序列为第一时间单元内经过速率匹配输出的编码比特序列, 第二编码比特序列的比特数等于第一时间单元对应的参数 G ; 终端设备将 UCI 映射至第一时间单元上的部分时频资源, 以替换承载于部分时频资源的第一编码比特序列中的部分比特序列。

- 30 应理解, 在时间单元(本申请实施例以时隙为例)上映射数据比特时, 首先将 UL-SCH 映射至该时间单元上的所有 RE, 然后将 UCI 通过打孔的方式映射至对应的资源, 这样 UCI 替换了该时间单元上部分 RE 承载的资源, 也就是说在该时间单元上除了被 UCI 替换的数据比特之外, 其他的数据比特占用的时频资源没有变。

- 35 本申请实施例, 上述方案, 通过打孔的方式将 UCI 复用在第一时间单元上, 可以使得第一时间单元上除 UCI 占用的时频资源外的时频资源上承载的包含于第一传输块的比特序列不改变, 从而可以提高网络设备译码的成功率和准确性。

- 40 第二种可能的实现方式: 终端设备根据时间单元# $n-1$ 对应的 aG 计算时间单元# n 内速率匹配输出的比特序列的起始比特的位置, 根据 G 或 G' 计算时间单元# n 内速率匹配输出的比特序列的长度。其中, 如果每个时间单元上包括的 OFDM 符号的数量不同, 则每个时间单元上对应的 G 也会不同, 如果每个时间单元上包括的 OFDM 符号的数量相同, 则

每个时间单元上对应的 G 也相同。

S201'具体参见第一种可能的实现方式的 S201 中的描述。

S202'具体参见第一种可能的实现方式的 S202 中的描述，其中区别在于：

- 引入一个参数 a ， $0 < a \leq 1$ ，用 a 和 G 的乘积 aG ，或 $[aG]$ 代替第一种可能的实施方式中的 G 计算速率匹配输出的比特序列的起始比特在循环缓存中的位置，其中， $[]$ 表示向上取整、向下取整、四舍五入取整。示例性地， a 可以由 RRC 信令和 DCI 信令指示，例如，RRC 信令中用于指示 a 的参数可以是 TDRA 表中增加 1 个字段，DCI 中用于指示 a 的参数也可以是新增的一个字段，RRC 或 DCI 中新增的字段宽度比如 3 或 4 或 5bit，分别可以表示 8/16/32 种可能的取值。比如 a 可能的取值可以是 $\{0.1x, 0.2x, 0.01x\}$ 集合中的子集，其中，2bit 表示这个子集大小为 4，3bit 表示这个子集大小为 8，4bit 表示这个子集大小为 16，5bit 表示这个子集大小为 32，6bit 表示这个自己大小为 64。

或者，还可以引入一个参数 G ，在第二种可能的实现方式中用 $G-b$ 替代 aG ，或 $[aG]$ 实现本方法。

下面以 aG 为例对 S202 进行说明。

- 具体地，在步骤二中，根据本申请提供的上行传输的方法中的参数 G 计算速率匹配输出的比特序列的长度，记作 E_r ，在步骤三中，根据 aG 计算出 E_r' ，根据 E_r' 确定速率匹配输出的比特序列的起点；或者，在步骤二中，根据现有技术中的参数 G' 计算速率匹配输出的比特序列的长度 E_r' ，在步骤三中，根据 aG 计算出 E_r' ，根据 E_r' 确定速率匹配输出的比特序列的起点。

- 其中，在根据 aG 计算出 E_r' 时，将第一种可能的实现方式的 S202 中 G 替换成 aG 代入具体的算法获得 E_r' ，并且在后续步骤中将 E_r 替换成 E_r' 进行计算。

或者，也可以用 aG_n 替换上述 aG ，将 E_r 替换成 E_r' 实现上述方案。

S203'，终端设备至少基于 UL-SCH 生成 PUSCH 并向网络设备发送该 PUSCH。

- 可能的情况一，终端设备基于 UL-SCH 生成 PUSCH。应理解，此时终端设备仅检测到发送 PUSCH 的信令，且网络设备没有发送指示终端设备在 PUSCH 上承载 UCI 的信令；或者网络设备发送指示终端设备在 PUSCH 上承载 UCI 的信令，但终端设备漏检信令。

示例性的，如图 13 中的 (a) 所示，此时循环缓存器中的比特序列按顺序依次由 PUSCH 承载。比较图 12 中的 (a) 和图 13 中的 (a) 可见，图 13 中的 (a) 中每一次速率匹配输出比特序列的起始比特的位置相对于图 12 中的 (a) 都提前了。

- 可能的情况二，终端设备将 UCI 复用在 UL-SCH 上，生成 PUSCH。

应理解，此时终端设备检测到网络设备发送的指示终端设备发送 PUSCH 和在 PUSCH 上承载 UCI 的信令。

- 作为一个示例，终端设备在 S202' 的步骤二中用 E_r' 计算速率匹配输出的比特序列的长度的情况下，假设 UCI 复用在 PUSCH 的第二个时隙上，终端设备在该第二时隙内进行速率匹配。由于速率匹配比特选择的起点是根据前一个上行时隙的速率匹配的 E_r' 确定，即根据 aG 确定，而前一个上行时隙的速率匹配输出的比特长度是根据 G' 确定的 E_r' ，在该情况下，由于 G' 需要扣除 UCI 的编码比特总数， G' 可能小于 aG ，也可能大于 aG 。因此，当 $G' < aG$ 时，第三个上行时隙的速率匹配比特选择的起始比特可能仍然会比第二次比特选择的结束比特在循环缓存中的位置更靠后，如图 13 中的 (b) 所示，两者之间仍然会有间隙，但是与图 12 中的 (b) 相比，图 13 中的 (b) 中，每一次速率匹配输出的比特

序列的起始比特的位置都提前了,第三次速率匹配的输出的比特序列的起始比特与第二次速率匹配输出的比特序列的结束比特之间的间隙也更小。或者,当 $G' > aG$ 时,第三次速率匹配比特选择的起始比特可能是第二次比特选择的结束比特在循环缓存中的下一个比特,如图 13 中的 (a) 所示,第三次速率匹配比特选择的起始比特和第二次比特选择的结束比特之间不会有间隙,且,图 13 中的 (a) 中每一次速率匹配输出比特序列的起始比特的位置相对于图 12 中的 (a) 都提前了。

作为另一个示例,终端设备在 S202' 的步骤二中用 E_r 计算速率匹配输出的比特序列的长度的情况下,UCI 复用在 PUSCH 的第二个时隙上,终端设备在该第二上行时隙内进行速率匹配。由于速率匹配比特选择的起点是根据前一个上行时隙的速率匹配的 E_r 确定,即根据 aG 确定,而前一个上行时隙的速率匹配输出的比特长度是根据 G 确定的, $aG < G$,在该情况下,如图 13 中的 (a) 所示,第三上行时隙的速率匹配比特选择的起始比特为第二上行时隙的比特选择的结束比特在循环缓存中的下一个比特,且图 13 中的 (a) 中每一次速率匹配输出比特序列的起始比特的位置相对于图 12 中的 (a) 都提前了。

应理解,比较图 13 中的 (a) 和图 13 中的 (b),无论 PUSCH 上是否有 UCI 复用,每一次速率匹配比特选择的起点是一致的,即该起点与 UCI 复用无关。

本申请实施例,通过 G 乘上大于零且小于一的 a ,使得网络设备计算第一编码比特序列的起始比特在循环缓存中的位置可以提前,从而缩短第二编码比特序列的结束比特与第一编码比特序列的起始比特之间的间隙,提高信道质量。

需要说明的是,在 S201' 中对调度信息的信令设计中提到,如果缩放参数 K =时隙数 N ,可以理解为网络设备不支持 TBoMS 重复,时隙数仅表示分配给一次 TBoMS 传输的时隙数;如果 $K < N$,可以理解为网络设备支持 TBoMS 重复,时隙数表示包括传输 TBoMS 重复在内的总时隙数, K 表示一次 TBoMS 传输的时隙数,且若 N/K 为整数,则 TBoMS 重复次数为 N/K 。而 S202' 的步骤三中,可能的实现方式一、三、四中的算法 3、4、5 中,都将 $n=mK$ 作为一个判断条件,其中, n 为速率匹配的次数, m 为一个整数,当满足该条件时, $k_{r,0}=k_0$,时间单元 $\#n$ 上的码块 $\#r$ 的速率匹配比特选择的起始比特位置 $k_{r,n}$ 由 k_0 决定,即由 RV 指定。其中,对于 $K=N$, $n=0$ 可以作为 $n=mK$ 的一种特殊情况,此时 k_0 由 RV 指示;对于 $1 < K < N$,且 N/K 为整数情况, k_m 由 RV 指示,其中 $m=N/K$ 。

可选地,方法 200 还包括:

应理解,假设网络设备指示终端设备发送 UCI,并复用在第 b 个时隙上,但是终端设备没有检测到该 PDCCH,上述方案能够实现网络设备和终端设备可以对齐从第 b 个时隙往后的所有时隙上速率匹配输出的比特序列的起始比特。进一步地,方法 200 还可以针对第 b 个时隙上的资源映射的方法进行进一步优化。

现有技术中,将 UCI 复元至第 b 个时隙的部分时频资源时,该部分时频资源原本承载的数据信息的比特会向后顺延。在终端设备漏检 PDCCH 时,终端设备在第 b 个时隙发送的全是数据信息的比特,而网络设备则认为上述部分时频资源承载的是 UCI,第 b 个时隙上的除该部分时频资源以外的时频资源为数据信息的比特。这会导致网络设备在当前时隙无法正确检测 UL-SCH。

在 S202' 中,在步骤二中根据 G 或 G_n 或 aG 或 aG_n 计算速率匹配输出的比特序列的长度和步骤三中根据 G 或 G_n 或 aG 或 aG_n 计算速率匹配输出的比特序列的起始比特的位置在循环缓存的位置时,终端设备将第二编码比特序列映射至第一时间单元上的时频资源,第

二比特序列为第一时间单元内经过速率匹配输出的编码比特序列,第二编码比特序列的比特数等于第一时间单元对应的参数 G ; 终端设备将 UCI 映射至第一时间单元上的部分时频资源,以替换承载于部分时频资源的第一编码比特序列中的部分比特序列。

- 5 应理解,在时间单元(本申请实施例以时隙为例)上映射数据比特时,首先将 UL-SCH 映射至该时间单元上的所有 RE,然后将 UCI 通过打孔的方式映射至对应的资源,这样 UCI 替换了该时间单元上部分 RE 承载的资源,也就是说在该时间单元上除了被 UCI 替换的数据比特之外,其他的数据比特占用的时频资源没有变。

- 10 本申请实施例,通过打孔的方式将 UCI 复用在第一时间单元上,可以使得第一时间单元上除 UCI 占用的时频资源外的时频资源上承载的包含于第一传输块的比特序列不改变,从而可以提高网络设备译码的成功率和准确性。

第三种可能的实现方式:终端设备根据时间单元# $n-1$ 对应的 G 和 Z_c 计算时间单元# n 内速率匹配输出的比特序列的起始比特的位置。

S201”具体可以参见第一种可能的实现方式中对应的描述。

S202”具体可以参见第一种可能的实现方式中对应的描述,其区别在于:

- 15 步骤三,终端设备确定速率匹配比特选择的起始比特位置,替换为以下几种可能的方案:

方案一:

当前速率匹配输出的比特序列的起始比特记作 k_n , 上一次速率匹配的结束比特记作 l_{n-1} , 那么根据上一次速率匹配的结束比特确定当前速率匹配的起始比特的起始比特方式例

- 20 如可以是 $k_n = \left\lceil \frac{l_{n-1}}{Z_c} \right\rceil Z_c$ 或 $k_n = \left\lfloor \frac{l_{n-1}}{Z_c} \right\rfloor Z_c$ 或 $k_n = \left\lceil \frac{l_{n-1}}{Z_c} \right\rceil Z_c$, 其中 $\left\lceil \frac{l_{n-1}}{Z_c} \right\rceil$ 、 $\left\lfloor \frac{l_{n-1}}{Z_c} \right\rfloor$ 、 $\left\lceil \frac{l_{n-1}}{Z_c} \right\rceil$ 分别表示对 $\frac{l_{n-1}}{Z_c}$ 四舍五入取整、向下取整、向上取整。其中, l_{n-1} 可以是根据起始位置和长度明确计算出来的,也可以是记录上一次速率匹配最后比特位置得到的,等等。

方案二:

- 25 根据上一次速率匹配输出的比特序列的起始比特的位置确定当前速率匹配输出的比特序列的起始比特的位置。具体地,可以根据上一次速率匹配输出的起始比特的位置和相邻两次速率匹配输出的比特序列的起始比特之间的间隔确定当前速率匹配输出的比特序列的起始比特的位置。

- 30 作为一个示例, $k_n = \text{mod}(k_{n-1} + I_r Z_c, N_{cb})$, 或者, $k_n = \text{mod}(k_{n-1} + I_r Z_c, \left\lceil \frac{N_{cb}}{Z_c} \right\rceil Z_c)$ 其中, 上一次速率匹配输出的比特序列的起始比特记作 k_{n-1} , 当前速率匹配输出的比特序列的起始比特记作 k_n , N_{cb} 表示一个码块的大小, n 表示时间单元, $I_r Z_c$ 为相邻两次速率匹配输出的比特序列的起始比特之间的间隔, Z_c 为低密度奇偶校验 LDPC 因子, $\left\lceil \frac{N_{cb}}{Z_c} \right\rceil$ 表示对于 $\frac{N_{cb}}{Z_c}$ 四舍五入取整、向上取整或者向下取整。

其中,相邻两次速率匹配输出的比特序列的起始比特之间的间隔中的 I_r 可以通过以下

方式计算。 $I_r = \left\lceil \frac{E_r}{Z_c} \right\rceil$ 或 $\left\lfloor \frac{E_r}{Z_c} \right\rfloor$ 或 $\left\lceil \frac{E_r}{Z_c} \right\rceil$, E_r 为 S202 中步骤二中根据 G 计算获得的上一次速率匹配输出的比特序列的长度, $\left\lceil \frac{E_r}{Z_c} \right\rceil$ 、 $\left\lfloor \frac{E_r}{Z_c} \right\rfloor$ 、 $\left\lceil \frac{E_r}{Z_c} \right\rceil$ 分别表示对 $\frac{E_r}{Z_c}$ 四舍五入取整、向下取整、向上取整, r 表示码块。

作为另一个示例, $k_n = \text{mod}(k_{n-1} + I_r, N_{cb})$, 或者, $k_n = \text{mod}(k_{n-1} + I_r, \left\lceil \frac{N_{cb}}{Z_c} \right\rceil Z_c)$ 上一次

- 5 速率匹配输出的比特序列的起始比特记作 k_{n-1} , 当前速率匹配输出的比特序列的起始比特记作 k_n , N_{cb} 表示一个码块的大小, n 表示时间单元, I_r 为相邻两次速率匹配输出的比特序列的起始比特之间的间隔,

其中, 相邻两次速率匹配输出的比特序列的起始比特之间的间隔 I_r 可以通过以下方式

计算。 $I_r = \left\lceil \frac{E_r}{Z_c} \right\rceil Z_c$ 或 $\left\lfloor \frac{E_r}{Z_c} \right\rfloor Z_c$ 或 $\left\lceil \frac{E_r}{Z_c} \right\rceil Z_c$, E_r 为 S202 中步骤二中根据 G 计算获得的上一次

- 10 速率匹配输出的比特序列的长度, $\left\lceil \frac{E_r}{Z_c} \right\rceil$ 、 $\left\lfloor \frac{E_r}{Z_c} \right\rfloor$ 、 $\left\lceil \frac{E_r}{Z_c} \right\rceil$ 分别表示对 $\frac{E_r}{Z_c}$ 四舍五入取整、向

下取整、向上取整, r 表示码块, Z_c 为低密度奇偶校验 LDPC 因子, $\left\lceil \frac{N_{cb}}{Z_c} \right\rceil$ 表示对于 $\frac{N_{cb}}{Z_c}$ 四舍五入取整、向上取整或者向下取整。

需要说明的是, 方案二中的 G 是每个时间单元上包括的 OFDM 符号的数量相同的情况下计算出的。

- 15 示例性地, 这里的 I_r 中的值可以是配置的, 例如在 TDRA 表中增加一列进行配置, 候选值是 $\{1, 2, \dots, 66\}$ 中的子集, 或者 $\{1, 2, \dots, 55\}$ 中的子集。

上述方案还可以通过其他方式进行实现:

将上述 E_r 替换成 $E_{r,n}$, I_r 替换成 $I_{r,n}$; 或者, 将上述 E_r 替换成 $E_{r,n}'$, I_r 替换成 $I_{r,n}$; 或者, 将上述 E_r 替换成 E_r' ; 或者, 将上述 E_r 替换成 $E_{r,n}''$, I_r 替换成 $I_{r,n}$ 。

- 20 方案三:

在方案二的基础上, 如果高层参数 rateMatching 没有设置为 limitedBufferRM, 即参数 $I_{LBRM} = 0$, 则 $N_{cb} = N$, 例如对于 BG1, $N = 66Z_c$, 故方案二中的方法可以简化为:

$k_n = \text{mod}(k_{n-1} + I_r, 66)$, I_r 与方案二中相同。注意 $k_n Z_c$ 表示起始比特位置。

或者, 类似地, 例如对于 BG2, $N = 50Z_c$, $k_n = \text{mod}(k_{n-1} + I_r, 50)$ 。

- 25 需要说明的是, 在 S201 中对调度信息的信令设计中提到, 如果缩放参数 K=时隙数 N, 可以理解为网络设备不支持 TBoMS 重复, 时隙数仅表示分配给一次 TBoMS 传输的时隙数; 如果 $K < N$, 可以理解为网络设备支持 TBoMS 重复, 时隙数表示包括传输 TBoMS 重复在内的总时隙数, K 表示一次 TBoMS 传输的时隙数, 且若 N/K 为整数, 则 TBoMS 重复次数为 N/K 。在方案二或方案三中, 对于 $K = N$ 情况, 当 $n=0$ 时, k_0 由 RV 指示; 对于
- 30 $1 < K < N$, 且 N/K 为整数情况, k_m 由 RV 指示, 其中 $m = N/K$ 。

应理解, 在满足 $N_{cb} = N$ 的条件时使用方案三可以大幅度节省计算开销。

另外, 在方案一至三中, E_r 还可以替换为由 G' 计算获得的 E_r' , 其余与上述具体实

现一致，在此不多赘述。

S203''具体可以参见第一种可能的实现方式中对应的描述。

S204''具体可以参见第一种可能的实现方式中对应的描述。

- 5 本申请实施例，终端侧在进行资源映射时，减少了查找映射至上一个时间单元的最后
一个比特在第一比特序列中的位置的步骤，在根据 G 确定第一编码比特序列的起始比特在
循环缓存中的位置的基础上结合 Z_c 确定该位置，简化了映射过程中的计算，节省了开销。

第四种可能的实现方式：终端设备根据时间单元#n-1 对应的 aG 和 Z_c 计算时间单元#n
内速率匹配输出的比特序列的起始比特的位置。

- 10 将第三种可能的实现方式中的 G 替换成 aG，将 E_r 还可以替换为由 aG 计算获得的 E_r ，
其余与上述具体实现一致，在此不多赘述。

下面结合图 14 至图 15 介绍本申请提供的上行传输的方法 300。

- 15 对于 PUSCH 重复类型 A，在 $K>1$ 的情况下，跨 K 个连续时隙应用相同的符号分配，
并且 PUSCH 仅限于单个传输层。UE 应在 K 个连续时隙中重复 TB，在每个时隙中应用相同
的符号分配。在 TB 的第 n 个传输时机应用的冗余版本，其中 $n=0,1,\dots,K-1$ ，根据前文
中的表 1 确定。示例性地，如图 14 所示，UE 在每一个可用的上行时隙发送一次重复。

- 20 方法 300 中，对于 TBoMSPUSCH 重复，一次 TBoMS 传输的时隙数为 N，一次 TBoMS
传输的重复次数为 M，在 $N>1$ 和 $M>1$ 的情况下，跨 $N \times M$ 个可用时隙应用相同的符号配
置。UE 应在 N 个可用时隙的 M 组（或 M 次重复）中重复 TB，在每个时隙中应用相同的
符号分配。在 TB 的第 n 个传输时机应用的冗余版本可用根据表 5 确定，其中 $n=0,1,\dots,N$
 $\times M-1$ 或 $n=0,N,2N,\dots,(M-1)N$ 。在每个组（或每个重复）内，跨 N 个连续可用时隙（或 N
个传输时机）应用单个 RV。示例性地，如图 15 所示， $N=2$ ， $M=4$ 。

表 5

调度 PUSCH 的 DCI 指示的 r_{vid}	第 n 个传输时机使用的 r_{vid} (TBoMS)			
	$n/N \bmod 4 = 0$	$n/N \bmod 4 = 1$	$n/N \bmod 4 = 2$	$n/N \bmod 4 = 3$
0	0	2	3	1
2	2	3	1	0
3	3	1	0	2
1	1	0	2	3

本申请实施例相对于现有的动态调度单个时隙传输一次重复，提出了动态调度用于一
次 TBoMS 传输的多个时隙的上行传输的方法，完善了 TBoMS 传输的方法。

- 25 下面结合图 16 至图 19 介绍本申请提供的上行传输的方法 400。

- 30 现有技术中，对于 PUSCH 重复类型 A 的 RV 循环配置调度，高层参数 repK-RV 定义
了要应用于重复的冗余版本模式。如果提供了 cg-RetransmissionTimer，则具有配置授权的
上行链路传输的冗余版本由 UE 确定。如果在 configuredGrantConfig 中没有提供参数
repK-RV 并且没有提供 cg-RetransmissionTimer，则具有配置授权的上行链路传输的冗余版
本应设置为 0。如果在 configuredGrantConfig 和 cg- 中提供了参数
repK-RVRetransmissionTimer 不提供，对于 K 次重复中的第 n 个传输时机， $n=1,2,\dots,K$ ，
它与配置的 RV 序列中的第 $(\bmod(n-1,4)+1)$ 个值相关联。如果配置的授权配置了
startingFromRV0 设置为 'off'，则传输块的初始传输可能仅在 K 次重复的第一个传输时机开
始。否则，传输块的初始传输开始于：

- 35 (1) K 次重复的第一个传输时机，对应于配置的 RV 序列为 {0,2,3,1} 的情况；或者，

(2)与RV=0相关联的K个重复的任何传输时机,对应于配置的RV序列为{0,3,0,3};或者,

(3)K次重复的任何传输时机,除了当 $K \geq 8$ 时的最后一次传输时机,对应于配置的RV序列为{0,0,0,0}。

5 方法400中,高层参数repK-RV定义了要应用于重复的冗余版本模式。如果提供了cg-RetransmissionTimer,则具有配置授权的上行链路传输的冗余版本由UE确定。如果在configuredGrantConfig中没有提供参数repK-RV并且没有提供cg-RetransmissionTimer,则对于每次重复(或每组N次传输),具有配置授权的上行链路传输的冗余版本应设置为0。如果configuredGrantConfig中提供参数repK-RV且未提供cg-RetransmissionTimer,则对于
10 $M \times N$ 个传输时机中的第n个传输时机, $n=1, N+1, 2N+1, \dots, (M-1) \times N+1$, 它与配置的RV序列中的第 $(\text{mod}((n-1)/N, 4)+1)$ 个值相关联。如果配置的授权配置中的startingFromRV0设置为'off',则传输块的初始传输可能仅在 $M \times N$ 个传输时机的第一个传输时机开始。否则,传输块的初始传输可能开始于:

(1) $M \times N$ 个传输时机的第一个传输时机,对应于配置的RV序列为{0,2,3,1};或者,
15 (2)与RV=0相关联的 $M \times N$ 个传输时机中的任何一个传输时机,对应于配置的RV序列为{0,3,0,3};或者,

(3)与RV=0相关联的 $M \times N$ 个传输时机中的任何一个传输时机对应于配置的RV序列为{0,0,0,0};或者,

(4)与RV=0相关联的 $M \times N$ 个传输时机中的任何一个传输时机中除了第 $((M-1) \times N+1)$ 个满足条件1时的传输时机以外的传输时机,对应于配置的RV序列为{0,0,0,0}。
20

其中,条件一为: $M \geq \mu$, $N \geq \mu$, $M \times N \geq \mu$, $\mu=2, 4, 8, 16, 32, 64$ 等中的一个值。

在单个TBoMS的每次重复中,跨N个连续可用时隙(或N个传输时机)应用单个RV。

下面以 $N=2$, $M=4$ 为例,分别介绍几种具体的实现方式。其中,图16对应startingFromRV0 set to 'off',图17对应startingFromRV0 set to 'on' & RV sequence {0,2,3,1},图18对应startingFromRV0 set to 'on' & RV sequence {0,3,0,3},图19对应startingFromRV0 set to 'on' & RV sequence {0,0,0,0}。
25

本申请实施例相对于现有的动态调度单个时隙传输一次重复,提出了静态调度用于一次TBoMS传输的多个时隙的上行传输的方法,完善了TBoMS传输的方法。

30 下面结合图20至图22介绍本申请的上行传输的方法500。

对于PUSCH repetition Type A,无论某个传输时机上的PUSCH传输是否被取消,RV依然按照如图14所示方法进行循环,这里假设第一个下行时隙对应RV0。也就是说,跳过被取消的传输时机上的PUSCH传输对应的RV,更准确地说每个传输时机对应的RV是确定不变的,如Table 6.1.2.1-2。某些时域资源被取消传输的原因见上文的case1至case6。

35 方法500针对TBOMS传输中部分时间单元被取消传输的情况,提出了三种上行传输的方案,下面以一个时间单元为一个时隙为例,进行说明。

方案一,对于TBoMS PUSCH传输(无论重复,还是不重复),如果某个传输时机上的PUSCH传输被取消,RV依然按照实施例1/2所述方法进行循环,并跳过被取消的传输时机上的PUSCH传输对应的编码比特。(所述“对应的编码比特”是指经过码块切分、
40 信道编码、速率匹配、码块级联后输出的编码比特)

如图 20 所示, 以一次传输进行 4 次重复, 一次重复占用两个时隙为例, 重复#2 对应的第一个时隙被取消传输, 那么重复#2 对应的第二个时隙继续传输原来要传输的比特序列, 其他重复上传输的比特序列不变。

需要说明的是, 该方法中的一个时隙为一个 available slot, 对应一个传输时机, 对应关系可能是: 一整个时隙, 或部分时隙, 或由 S 和 L 指示的部分时隙, 其中 S 和 L 为 RRC 信令 TDRA 表中的起始符号和长度。Available slot 根据 RRC 配置的上下行时隙配比 (如 tdd-UL-DL-ConfigurationCommon 或 tdd-UL-DL-ConfigurationDedicated)、TDRA 表、SSB 决定, 并不考虑冲突丢弃情况。

方案二, 对于 TBoMS PUSCH 传输 (无论重复, 还是不重复), 如果某个传输时机上的 PUSCH 传输被取消, RV 依然按照方法 300 或 400 所述方法进行循环, 并且:

如果被取消 PUSCH 传输的传输时机的速率匹配的比特选择的起始比特由 RV 指示 (即一次重复的第一个传输时机, 或者 M 组 N 个时隙的第一个时隙), 则被取消 PUSCH 传输的传输时机的下一个传输时机的速率匹配的比特选择的起始比特由所述 RV 指示。

如果被取消 PUSCH 传输的传输时机的速率匹配的起始比特不是由 RV 指示的 (即一次重复的非第一个传输时机, 后者 M 组 N 个时隙的非第一个时隙), 则被取消 PUSCH 传输的传输时机的下一个传输时机的速率匹配的比特选择的起始比特紧接着被取消 PUSCH 传输的传输时机的上一个传输时机的速率匹配的比特选择的结束比特。

需要说明的是, 该方法中的一个时隙为一个 available slot 对应一个传输时机, 对应关系可能是: 一整个时隙, 或部分时隙, 或由 S 和 L 指示的部分时隙, 其中 S 和 L 为 RRC 信令 TDRA 表中的起始符号和长度。Available slot 根据 RRC 配置的上下行时隙配比 (如 tdd-UL-DL-ConfigurationCommon 或 tdd-UL-DL-ConfigurationDedicated)、TDRA 表、SSB 决定, 并不考虑冲突丢弃情况。

方案三: 重新定义 available slot, Available slot 根据 RRC 配置的上下行时隙配比 (如 tdd-UL-DL-ConfigurationCommon 或 tdd-UL-DL-ConfigurationDedicated) 和 TDRA 表决定, 并考虑冲突丢弃情况。一个 available slot 对应一个传输时机, 对应关系可能是: 一整个时隙, 或部分时隙, 或由 S 和 L 指示的部分时隙, 其中 S 和 L 为 RRC 信令 TDRA 表中的起始符号和长度。然后使用实施例 1/2 所述方法进行 RV cycling。

下面介绍本申请提供的上行传输的方法 600。

方法 600 能够实现 N 个时隙一起进行速率匹配。具体地, 在该方法中, 所述参数 G 为一个时间单元内能够用于传输传输块和所述 UCI 的编码比特总数, 其中一个时间单元为 N 个时隙。

通过打孔的方式在该 N 个时隙上进行 UCI 复用, 且该 UCI 仅复用在 1 个时隙上。

因为 N 个时隙仅做一次速率匹配, 故仅有一次比特选择, 起点由 RV 确定, 长度根据 G 确定。

首先, 经过速率匹配和码块级联后得到 UL-SCH 比特。将 UCI 对应的比特通过打孔的方式复用在 UL-SCH 比特。将复用有 UCI 的 UL-SCH 比特映射至 N 个时隙的时频资源上。

下面介绍本申请提供的上行传输的方法 700。

可能的实现方式一, 不期待 A-CSI (非周期的 CSI) 在 TBoMS PUSCH 上复用。

应理解, A-CSI 由 DCI 触发, 并复用在 DCI 调度的 PUSCH 上完成 A-CSI 上报, 如

果 DCI 调度多个 PUSCH, 如 TBoMS, 则 A-CSI 仅复用在第一个 PUSCH 上完成上报。而根据目前的速率匹配方法获得的输出编码比特序列中, 一般第一个时间单元承载的都是系统比特。因此, 通过不期待 A-CSI 在 TBoMS PUSCH 上复用, 可以减少 A-CSI 复用对 TBoMS PUSCH 中系统比特的影响, 提高译码成功率。

- 5 可能的实现方式二, 如果网络设备通过 DCI 触发的 A-CSI 和其他信息指示的配置授权 (configured grant, CG) TBoMS PUSCH 在某个时间单元上重叠, 或者说用于传输 A-CSI 和 CG TBoMS PUSCH 的资源在时域上重叠, 可以采用以下几种传输方式:

方式一, 取消重叠时间单元上的 PUSCH 传输;

方式二, 取消重叠时间单元上的 A-CSI 的传输。

- 10 可能的实现方式三, 不期待 CSI 在 TBoMS PUSCH 上复用。

应理解, CSI 比特数很大, 相比于在 TBoMS PUSCH 上复用 CSI, 或者在 TBoMS PUSCH 上用打孔的方式传输 CSI, 可以减少 CSI 复用对 TBoMS PUSCH 中系统比特的影响, 提高译码成功率。此 CSI 是指所有类型的 CSI。

- 15 这里对本申请实施例中涉及的打孔进行解释, 例如在将 CSI 用打孔的方式复用到 TBoMS PUSCH 上时, 假设 CSI 占用 40bit, 分别为 {a1, a2, ..., a40}, TBoMS PUSCH 占 100bit, 分别为 {b1, b2, ..., b100} 将 CSI 复用到 TBoMS PUSCH 的第 11 至 50 比特上, 那么用 {a1, a2, ..., a40} 替换 {b11, b12, ..., b50}, 其余 {b1, b2, ..., b10} 和 {b51, b52, ..., b100} 位置不变。

- 20 可能的实现方式四, 第一个时间单元起始比特由 RV 确定, 第 2 至第 n 个时间单元起始比特由方法#1 确定, 第 n+1 个时间单元至第 N 个时间单元, 起始比特由方法#2 确定。

其中, 方法#1 即为本申请前述实施例中根据 G' 计算时间单元上速率匹配过程中比特选择的起始比特的索引的方法, 方法#2 即为本申请前述实施例中根据 G 计算时间单元上速率匹配过程中比特选择的起始比特的索引的方法。其中, 对于 G 和 G' 的定义具体参见上文, 在此不多赘述。

- 25 具体地, $k_{n+1} = l_n + 1$, 其中 $l_n = k_n + E_n - 1$, $E_n = N_L Q_m \left\lceil \frac{G_n}{N_L Q_m C'} \right\rceil$, 或 $E_n = N_L Q_m \left\lfloor \frac{G_n}{N_L Q_m C'} \right\rfloor$, k_n 和 l_n 分别表示第 n 个时间单元上速率匹配过程中比特选择的起始比特的索引和结束比特的索引, k_{n+1} 表示下一个时间单元 (即第 n+1 个时间单元) 上速率匹配过程中比特选择的起始比特的索引, N_L 表示层数, Q_m 表示调制阶次, C' 表示调度的码块数, G_n 表示第 n 个时间单元上可用于传输 TB 的可用编码比特数, E_n 为比特序列的比特数。

- 30 其中, 在方法#1 中, 这里的 G_n 为 G', 在方法#2 中, 这里的 G_n 为 G, 在具体实现上可以参考方法 200 中对应的部分, 根据 G 计算每个时间单元内经速率匹配输出比特序列的起点, 根据 G 或 G' 计算该输出比特序列的长度。

n 的取值可以是默认值, 例如 n=2、n=3 或 n=4 等。或者, n 的取值可以为指定值。示例性地, 在 TDRA 表中增加一列, 表示 n 值; n 值集合为 n={2,3,4} 等。

- 35 可能的实现方式五

根据协议版本 Rel-15/16 规定, 会复用在 PUSCH 上传输的 UCI 分为三种: HARQ-ACK, CSI, 和 CG-UCI。

一、对于复用 HARQ-ACK 而言, DCI 调度 UE 接收 PDSCH, UE 接收 PDSCH 之后在 PUCCH 上反馈 HARQ-ACK, 如果 DCI 漏检, 使用方法#1 会出现对不齐的问题, 如果

使用方法#2, 会出现系统比特被打掉的可能, 但考虑到 HARQ-ACK 比特数较少, 通过打孔的方式进行复用对 PUSCH 的影响较小, 故使用方法#2 性能损失不大。

二、对于复用 CSI 而言, 不会出现漏检的问题, 可以使用方法#1; 反之, 如果使用方法#2, 由于 CSI 的比特数较大, 打孔对 PUSCH 的影响较大, 故使用方法#2 性能损失大。

5 CSI 有三种反馈方式: 周期 CSI (period CSI, P-CSI), 半持续 CSI (semi-persistent CSI, SP-CSI), 以及非周期 (aperiodic, A-CSI)。

(1) P-CSI 由 RRC 信令配置和触发, 并承载在 PUCCH 上报, 因此不会出现终端设备漏检 DCI 的情况;

10 (2) SP-CSI 由 RRC 信令配置, 并由 MAC CE 信令触发, 并承载在 PUCCH 或 PUSCH 上报, 因此不会出现终端设备漏检 DCI 的情况;

(3) A-CSI 由 RRC 信令配置, 并由 DCI 信令触发, 并承载在 PUSCH 上报, 其中 PUSCH 同样由此 DCI 调度, 由于 DCI 同时触发和调度 PUSCH, 如果漏检 DCI 的话, 那么 PUSCH 也不会传输, 所以也不会有本申请所要解决的问题。

15 由上可知, 因为 P-CSI 和 SP-CSI 不是由 DCI 触发的, 故不会出现漏检的问题; 由于调度 PUSCH 和触发 A-CSI 上报使用的是同一个 DCI, 如果 DCI 漏检, PUSCH 也不会发送, 因此也不回出现漏检的问题。

三、对于复用 CG-UCI 而言, CG-UCI 由 RRC 信令配置, 由 RRC 或者 DCI 触发。当 CG-UCI 由 RRC 触发时, 则不会出现漏检问题; 当 CG-UCI 由 DCI 触发会出现漏检的问题, 但是 PUSCH 也不会发送, 故同样适用于方法#1。

20 然而, 考虑到 CG-UCI 和 HARQ-ACK 同时复用到 PUSCH 时会进行联合编码, 编码后当做 HARQ-ACK 复用到 PUSCH 上, 如果直接打孔可能影响较大。

综上所述, 只有在复用 HARQ-ACK 时会出现漏检 DCI 的情况。CG-UCI 和 HARQ-ACK 同时复用情况特殊, 还需要进一步讨论。

针对上述情况, 可以考虑通过以下几种方式进行处理。

25 第一种方式

方法#2 中的 G_n , 表示假设没有 HARQ-ACK 复用发生时, 第 n 个时间单元上可用于传输 TB 的可用编码比特的总数。

示例一, 只有 HARQ-ACK 复用到 PUSCH 时, G_n 不扣除第 n 个时间单元上用于传输 HARQ-ACK 的编码比特的总数。

30 示例二, UCI 复用且 UCI 为 CSI 和/或 CG-UCI 到 PUSCH 时, G_n 扣除第 n 个时间单元上用于传输 UCI 的编码比特的总数, 即使用方法#1。

示例三, CSI 和 HARQ-ACK 复用到 PUSCH 时, G_n 扣除第 n 个时间单元上用于传输 CSI 的编码比特的总数, 不扣除第 n 个时间单元上用于传输 HARQ-ACK 的编码比特的总数。

35 以示例一为例, 假设 HARQ-ACK 占用 10 比特, PUSCH 占用 100 比特, 则以打孔的方式在 PUSCH 上复用 HARQ-ACK 时打孔的位置为 PUSCH 占用的 100 比特的最后 10 比特。

第二种方式,

G_n 使用方法#1 中的 G' , 即参照现有技术的方案, 但修改 HARQ-ACK 的复用过程:

40 (1) HARQ-ACK 占用的比特数小于等于 2bit 时, 采用现有技术中的打孔方式, 将

HARQ-ACK 复用到 PUSCH 上;

(2) HARQ-ACK 占用的比特数大于 2bit 时, 仍然对 UL-SCH 编码比特进行打孔, 打孔的位置与现有技术中复用 HARQ-ACK 直接做速率匹配时的位置一致, 现有技术中 UCI 复用的细节可以参见协议 38.212 6.2.7 节。

5 作为一个示例, 复用 CG-UCI 时, TBoMS 不应用于非授权频段条件下, 仅应用与授权频段条件下。

 应理解, CG-UCI 仅是非授权频段下对 CG PUSCH 的增强, 即 CG PUSCH 仅在非授权频段下会上报 CG-UCI, 在授权频段下不会上报 CG-UCI。

10 作为另一个示例, CG-UCI 和 HARQ-ACK 联合编码, 当做 HARQ-ACK 处理。也就是说, 这种情况跟上一个示例中 HARQ-ACK 解决方法一致。

 可能的实现方式六, 再介绍两种方法#1 和方法#2 的切换方式。

 切换方式一, 根据 DCI 的位置进行切换。

 调度 PDSCH (且 HARQ-ACK 在 PUSCH 上复用) 的 DCI 仅出现在调度 PUSCH 的 DCI 之前, 使用方法#1; 否则, 使用方法#2。

15 切换方式二, 根据 TDRA 表的指示进行切换。

 示例性的, TDRA 表中增加一列, 该列中包括 1 比特, 用来指示使用方法#1 或方法#2。

 上述几种可能的实现方式, 都能减少 CSI 复用对 TBoMS PUSCH 中系统比特的影响, 提高译码成功率。

 下面介绍本申请提供的上行传输的方法 800。

20 网络设备向终端设备发送调度信息, 该调度信息指示终端设备通过 PUSCH 发送 TB, 随后网络侧没有接收到该 TB, 或者没有接收到正确的 TB, 网络设备向终端设备再次发送调度信息, 指示终端设备重传该 TB, 终端设备收到该调度信息, 终端设备根据该调度信息向网络设备重传该 TB, 此外, 该调度信息还会指示终端设备重传 TB 时使用的上行传输方式。

25 表 6 示出了四种上行传输方式, 网络侧可以通过 DCI 指示 TDRA 表中不同的行进行动态切换。其中, N 为时隙数, M 为重复次数。

表 6

	单时隙 PUSCH	Type A PUSCH 重复	TBoMS	TBoMS 重复
N=1,M=1	✓	×	×	×
N=1,M>1	×	✓	×	×
N>1,M=1	×	×	✓	×
N>1,M>1	×	×	×	✓

 应理解, TBoMS 限制在一个 CB 上在传输, 而 PUSCH 并没有限制在一个 CB 上传输。在单时隙 PUSCH 或 Type A PUSCH 重复, 以及, TBoMS 或 TBoMS 重复之间进行动态切换时可能出现以下问题: 由没有限制在单个 CB 上进行的 PUSCH 传输切换到限制在单个 CB 上进行的 TBoMS 传输时, 由于现有协议的不支持, 很可能出现无法实现 TBoMS 传输的情况。

30 针对上述问题, 方法 800 提出了不同传输方式之间动态切换的规则。

 规则 1: 不进行动态切换

35 不期望 PUSCH/ Type APUSCH 重复{N=1, M≥1}和 TBoMS/TBoMS 重复{ N> 1, M

≥ 1 之间进行动态切换;即初传为 PUSCH/Type APUSCH 重复 $\{N=1, M \geq 1\}$ 时,重传不能是 TBoMS/TBoMS 重复 $\{N > 1, M \geq 1\}$;初传 TBoMS/TBoMS 重复 $\{N > 1, M \geq 1\}$ 时,重传不能是 PUSCH/Type APUSCH 重复 $\{N=1, M \geq 1\}$ 。

示例性地,TDRA 表所有行仅满足 $N=1$ 或仅满足 $N > 1$,不能同时出现 $N=1$ 和 $N > 1$ 的情况,通过 RRC 重新配置 TDRA 表进行两者之间的半静态切换。

规则 2: 初传 TBoMS 切换到重传 PUSCH

规则 2-1: 仅支持初传 TBoMS/IBoMS 重复 $\{N > 1, M \geq 1\}$ 切换为重传单时隙 PUSCH $\{N=1, M=1\}$ 。

或者,规则 2-2: 仅支持初传 TBoMS/TBoMS 重复 $\{N > 1, M \geq 1\}$ 切换为重传 Type APUSCH 重复 $\{N=1, M > 1\}$ 。

或者,规则 2-3: 仅支持初传 TBoMS/IBoMS 重复 $\{N > 1, M \geq 1\}$ 既可以切换成单时隙的 PUSCH $\{N=1, M=1\}$,又可以切换成 Type APUSCH 重复 $\{N=1, M > 1\}$ 。

规则 3: 初传 PUSCH 切换到重传 TBoMS

仅当单 CB 传输时,支持初传 PUSCH/Type A PUSCH 重复 $\{N=1, M \geq 1\}$ 切换为重传 TBoMS/TBoMS 重复 $\{N > 1, M \geq 1\}$;否则,不支持动态切换。

上述实施例,通过规定不同传输方式之间动态切换的规则,可以减少无法实现 TBoMS 传输的情况。

下面介绍本申请提供的上行传输的方法 900。

与上述方法 200 中 S201 中可能的设计一至可能的设计六对应,可以通过在 PUSCH-TimeDomainAllocationList 配置的 TDRA 表中新增一列来指示 TBoMS 分配的时间单元数 N 。例如在 TDRA 表中配置 PUSCH 重复类型 A 的重复次数配置栏,即 numberofRepetitions,可以用于指示 TBoMS 传输使能时单个 TBoMS 的重复次数 M 。然而,该实现方式仅针对动态授权和配置授权调度的配置授权类型 2 (configured grant type 2) TBoMS 传输提供 TDRA 指示,但未涵盖针对配置授权类型 1 (configured grant type 1) TBoMS 传输的 TDRA 指示。

为了进一步完善本申请提供的上行传输的方法,方法 900 通过在 RRC 信令的信元 (information element, IE) ConfiguredGrantCofig 中引入一个新的字段来指示单个 TBoMS 传输的分配时间单元数 N 来实现针对配置授权类型 1 (configured grant type 1) TBoMS 传输的 TDRA 指示。示例性地,具体的指示方式可以如下所示,其中新字段 numberOfSlots 用于指示 N , N 至少包括 $\{2,4,8\}$,例如 $\{1,2,4,8\}$

示例性地,对于具有配置授权类型 1 或配置授权类型 2 的 TBoMS 传输,如果 N 存在于 TDRA 表中,则 N 由 TDRA 表中的索引行提供;否则, N 由 ConfiguredGrantCofig 中的新增字段提供。进一步地,如果 N 是由 ConfiguredGrantCofig 中的新增字段提供的,同时 $N > 1$,则启用 TBoMS 传输。如果 N 由 ConfiguredGrantCofig 中的新增字段提供,同时 $N=1$,则启用单时隙 PUSCH。在启用 TBoMS 传输时,现有字段 repK 用于指示单个 TBoMS 的重复次数 M 。

上述实施例,通过规定具有配置授权类型 1 的 TBoMS 传输的 TDRA 指示,进一步完善本申请实施例提供的上行传输的方法,进一步提高译码成功率。

以上,结合图 6 至图 23 详细说明了本申请实施例提供的方法。以下,结合图 24 至图 25 详细说明本申请实施例提供的装置。

图 24 是本申请实施例提供的用于上行传输的通信装置的示意性框图。如图 24 所示，该通信装置 10 可以包括收发模块 11 和处理模块 12。

其中，收发模块 11 可以用于接收其他装置发送的信息，还可以用于向其他装置发送信息。比如，接收第一指示信息或发送第二指示信息。处理模块 12 可以用于进行装置的内容处理，比如，根据所述第一时间单元对应的参数 G 确定第一编码比特序列的起始比特在循环缓存中的位置。

在一种可能的设计中，该通信装置 10 可对应于上述方法实施例中的终端设备。

具体地，该通信装置 10 可对应于根据本申请实施例的方法 100 至方法 600 中任一方法中的终端设备或 UE，该通信装置 10 可以包括用于执行相应方法中由终端设备所执行的操作的模块，并且，该通信装置 10 中的各单元分别为了实现相应方法中由终端设备所执行的操作。

示例性的，在该通信装置 10 对应于方法 100 中的终端设备时，收发模块 11 用于执行步骤 S101、S102、S104，处理模块 12 用于执行 S103。

示例性的，在该通信装置 10 对应于方法 200 中的终端设备时，收发模块 11 用于执行步骤 S201、S203，处理模块 12 用于执行 S202。

在另一种可能的设计中，该通信装置 10 可对应于上述方法实施例中的网络设备。

具体地，该通信装置 10 可对应于根据本申请实施例的方法 100 至方法 400 中任一方法中的网络设备，该通信装置 10 可以包括用于执行相应方法中由网络设备所执行的操作的模块，并且，该通信装置 10 中的各单元分别为了实现相应方法中由网络设备所执行的操作。

示例性的，在该通信装置 10 对应于方法 100 中的网络设备时，收发模块 11 用于执行步骤 S101、S102、S104，处理模块 12 用于执行 S105。

示例性的，在该通信装置 10 对应于方法 200 中的网络设备时，收发模块 11 用于执行步骤 S201、S203，处理模块 12 用于执行步骤 S204。

图 25 为本申请实施例提供的通信装置 20 的示意图。

在一种可能的设计中，该装置 20 可以为网络设备，也可以为位于网络设备上的芯片或芯片系统等。

在一种可能的设计中，该装置 20 可以为终端设备包括各种具有无线通信功能的手持设备、车载设备、可穿戴设备、计算设备或连接到无线调制解调器的其它处理设备，以及各种形式的终端，移动台，终端，用户设备，软终端等等，也可以为位于终端设备上的芯片或芯片系统等。

该装置 20 可以包括处理器 21（即，处理模块的一例）和存储器 22。该存储器 22 用于存储指令，该处理器 21 用于执行该存储器 22 存储的指令，以使该装置 20 实现如图 8 至图 23 中对应的方法中上述各种可能的设计中的设备执行的步骤。

进一步地，该装置 20 还可以包括输入口 23（即，收发模块的一例）和输出口 24（即，收发模块的另一例）。进一步地，该处理器 21、存储器 22、输入口 23 和输出口 24 可以通过内部连接通路互相通信，传递控制和/或数据信号。该存储器 22 用于存储计算机程序，该处理器 21 可以用于从该存储器 22 中调用并运行该计算机程序，以控制输入口 23 接收信号，控制输出口 24 发送信号，完成上述方法中终端设备或无线接入网设备或 UE 或基站的步骤。该存储器 22 可以集成在处理器 21 中，也可以与处理器 21 分开设置。

可选地,若该报文传输的装置 20 为通信设备,该输入口 23 为接收器,该输出口 24 为发送器。其中,接收器和发送器可以为相同或者不同的物理实体。为相同的物理实体时,可以统称为收发器。

5 可选地,若该装置 20 为芯片或电路,该输入口 23 为输入接口,该输出口 24 为输出接口。

作为一种实现方式,输入口 23 和输出口 34 的功能可以考虑通过收发电路或者收发的专用芯片实现。处理器 21 可以考虑通过专用处理芯片、处理电路、处理器或者通用芯片实现。

10 作为另一种实现方式,可以考虑使用通用计算机的方式来实现本申请实施例提供的设备。即将实现处理器 21、输入口 23 和输出口 24 功能的程序代码存储在存储器 22 中,通用处理器通过执行存储器 22 中的代码来实现处理器 21、输入口 23 和输出口 24 的功能。

其中,装置 20 中各模块或单元可以用于执行上述方法中进行随机接入的设备(例如,终端设备)所执行的各动作或处理过程,这里,为了避免赘述,省略其详细说明。

15 该装置 20 所涉及的与本申请实施例提供的技术方案相关的概念,解释和详细说明及其他步骤请参见前述方法或其他实施例中关于这些内容的描述,此处不做赘述。

20 应理解,本申请实施例中,该处理器可以为中央处理单元(CPU, central processing unit),该处理器还可以是其他通用处理器、数字信号处理器(DSP, digital signal processor)、专用集成电路(application specific integrated circuit, ASIC)、现成可编程门阵列(field programmable gate array, FPGA)或者其他可编程逻辑器件、分立门或者晶体管逻辑器件、分立硬件组件等。通用处理器可以是微处理器或者该处理器也可以是任何常规的处理器等。

本申请实施例还提供一种计算机可读存储介质,其上存储有用于实现上述方法实施例中由网络设备或终端设备执行的方法的计算机指令。

例如,该计算机程序被计算机执行时,使得该计算机可以实现上述方法实施例中由网络设备或终端设备执行的方法。

25 本申请实施例还提供一种计算机可读存储介质,其上存储有用于实现上述方法实施例中由网络设备或终端设备执行的方法的计算机指令。

例如,该计算机程序被计算机执行时,使得该计算机可以实现上述方法实施例中由网络设备或终端设备执行的方法。

30 还应理解,本申请实施例中的存储器可以是易失性存储器或非易失性存储器,或可包括易失性和非易失性存储器两者。其中,非易失性存储器可以是只读存储器(read-only memory, ROM)、可编程只读存储器(programmable ROM, PROM)、可擦除可编程只读存储器(erasable PROM, EPROM)、电可擦除可编程只读存储器(electrically EPROM, EEPROM)或闪存。易失性存储器可以是随机存取存储器(random access memory, RAM),其用作外部高速缓存。通过示例性但不是限制性说明,许多形式的RAM可用,例如静态随机存取存储器(static RAM, SRAM)、动态随机存取存储器(DRAM)、同步动态随机存取存储器(synchronous DRAM, SDRAM)、双倍数据速率同步动态随机存取存储器(double data rate SDRAM, DDR SDRAM)、增强型同步动态随机存取存储器(enhanced SDRAM, ESDRAM)、同步连接动态随机存取存储器(synchlink DRAM, SLDRAM)和直接内存总线随机存取存储器(direct rambus RAM, DR RAM)。

40 上述实施例,可以全部或部分地通过软件、硬件、固件或其他任意组合来实现。当使

用软件实现时，上述实施例可以全部或部分地以计算机程序产品的形式实现。该计算机程序产品包括一个或多个计算机指令或计算机程序。在计算机上加载或执行所述计算机指令或计算机程序时，全部或部分地产生按照本申请实施例所述的流程或功能。所述计算机可以为通用计算机、专用计算机、计算机网络、或者其他可编程装置。所述计算机指令可以存储在计算机可读存储介质中，或者从一个计算机可读存储介质向另一个计算机可读存储介质传输，例如，所述计算机指令可以从一个网站站点、计算机、服务器或数据中心通过有线（例如红外、无线、微波等）方式向另一个网站站点、计算机、服务器或数据中心进行传输。所述计算机可读存储介质可以是计算机能够存取的任何可用介质或者是包含一个或多个可用介质集合的服务器、数据中心等数据存储设备。所述可用介质可以是磁性介质（例如，软盘、硬盘、磁带）、光介质（例如，DVD）、或者半导体介质。半导体介质可以是固态硬盘。

应理解，本文中术语“和/或”，仅仅是一种描述关联对象的关联关系，表示可以存在三种关系，例如，A 和/或 B，可以表示：单独存在 A，同时存在 A 和 B，单独存在 B 这三种情况。另外，本文中字符“/”，一般表示前后关联对象是一种“或”的关系。

应理解，在本申请的各种实施例中，上述各过程的序号的大小并不意味着执行顺序的先后，各过程的执行顺序应以其功能和内在逻辑确定，而不应对本申请实施例的实施过程构成任何限定。

本领域普通技术人员可以意识到，结合本文中所公开的实施例描述的各示例的单元及算法步骤，能够以电子硬件、或者计算机软件和电子硬件的结合来实现。这些功能究竟以硬件还是软件方式来执行，取决于技术方案的特定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，但是这种实现不应认为超出本申请的范围。

所属领域的技术人员可以清楚地了解到，为描述的方便和简洁，上述描述的系统、装置和单元的具体工作过程，可以参考前述方法实施例中的对应过程，在此不再赘述。

在本申请所提供的几个实施例中，应该理解到，所揭露的系统、装置和方法，可以通过其它的方式实现。例如，以上所描述的装置实施例仅仅是示意性的，例如，所述单元的划分，仅仅为一种逻辑功能划分，实际实现时可以有另外的划分方式，例如多个单元或组件可以结合或者可以集成到另一个系统，或一些特征可以忽略，或不执行。另一点，所显示或讨论的相互之间的耦合或直接耦合或通信连接可以是通过一些接口，装置或单元的间接耦合或通信连接，可以是电性，机械或其它的形式。

所述作为分离部件说明的单元可以是或者也可以不是物理上分开的，作为单元显示的部件可以是或者也可以不是物理单元，即可以位于一个地方，或者也可以分布到多个网络单元上。可以根据实际的需要选择其中的部分或者全部单元来实现本实施例方案的目的。

另外，在本申请各个实施例中的各功能单元可以集成在一个处理单元中，也可以是各个单元单独物理存在，也可以两个或两个以上单元集成在一个单元中。

所述功能如果以软件功能单元的形式实现并作为独立的产品销售或使用时，可以存储在一个计算机可读存储介质中。基于这样的理解，本申请的技术方案本质上或者说对现有技术做出贡献的部分或者该技术方案的部分可以以软件产品的形式体现出来，该计算机软件产品存储在一个存储介质中，包括若干指令用以使得一台计算机设备（可以是个人计算机，服务器，或者网络设备等）执行本申请各个实施例所述方法的全部或部分步骤。而

前述的存储介质包括：U 盘、移动硬盘、只读存储器（read-only memory, ROM）、随机存取存储器（random access memory, RAM）、磁碟或者光盘等各种可以存储程序代码的介质。

- 5 以上所述，仅为本申请的具体实施方式，但本申请的保护范围并不局限于此，任何熟悉本技术领域的技术人员在本申请揭露的技术范围内，可轻易想到变化或替换，都应涵盖在本申请的保护范围之内。因此，本申请的保护范围应以所述权利要求的保护范围为准。

权利要求书

1. 一种上行传输的方法，其特征在于，包括：
- 5 终端设备接收来自网络设备的第一指示信息，所述第一指示信息用于指示所述终端设备在多个时间单元上发送第一传输块；
- 所述终端设备接收来自所述网络设备的第二指示信息，所述第二指示信息用于指示所述终端设备在第一时间单元上发送上行控制信息 UCI，所述第一时间单元为所述多个时间单元中的一个；
- 10 所述终端设备根据所述第一时间单元对应的参数 G 确定第一编码比特序列的比特选择的起始比特在循环缓存中的位置，所述第一编码比特序列为第二时间单元内经过速率匹配输出的编码比特序列，所述参数 G 为一个时间单元内可用于传输所述第一传输块和所述 UCI 的编码比特总数，G 为正整数，所述第二时间单元为所述第一时间单元在所述多个时间单元中的下一个时间单元；
- 所述终端设备在所述多个时间单元上向所述网络设备发送所述 UCI 和所述第一传输块。
- 15 2. 根据权利要求 1 所述的方法，其特征在于，
- 所述终端设备在所述第一时间单元上向所述网络设备发送所述第一 UCI，在所述多个时间单元上向所述网络设备发送所述 UCI 和所述第一传输块。
3. 根据权利要求 1 或 2 所述的方法，其特征在于，所述终端设备还根据以下参数中
- 20 至少一项确定第一编码比特序列的比特选择的起始比特在循环缓存中的位置：
- 在所述第一时间单元的速率匹配的比特选择的起始比特索引；
- 循环缓存长度；
- 循环缓存中填充比特的数量。
4. 根据权利要求 1 或 3 所述的方法，其特征在于，所述方法还包括：
- 25 当所述终端设备未在所述第一时间单元上向所述网络设备发送所述第一 UCI 时，所述终端设备根据所述第一时间单元执行比特选择跳过的填充比特的个数确定在所述第二时间单元上执行比特选择的起始比特索引。
5. 根据权利要求 1 至 4 任一项所述的方法，其特征在于，所述方法还包括：
- 所述终端设备根据所述第二时间单元对应参数 G' 确定在第二时间单元上执行比特选择的比特长度，所述参数 G' 为所述第二时间单元内仅可用于传输所述第一传输块的编码比特总数。
- 30 6. 根据权利要求 1 至 5 任一所述的方法，其特征在于，所述第一指示信息还用于指示所述终端设备能够在所述多个时间单元上发送第一传输块，包括：
- 当所述终端设备收到所述第一指示信息且所述第一指示信息指示的时间单元的数量
- 35 大于 1 时，指示所述终端设备能够在所述多个时间单元上发送所述第一传输块。
7. 根据权利要求 6 所述的方法，其特征在于，所述多个时间单元的个数至少为 1，2，4，8。
8. 一种上行传输的方法，其特征在于，包括：
- 网络设备向终端设备发送第一指示信息，所述第一指示信息用于指示所述终端设备在
- 40 多个时间单元上发送第一传输块；

所述网络设备向所述终端设备发送第二指示信息,所述第二指示信息用于指示所述终端设备在第一时间单元上发送上行控制信息 UCI,所述第一时间单元为所述多个时间单元中的一个;

所述网络设备在所述多个时间单元上接收来自所述终端设备的所述第一传输块;

- 5 所述网络设备根据所述第一时间单元对应的参数 G 确定第一编码比特序列的比特选择的起始比特在循环缓存中的位置,所述第一编码比特序列为第二时间单元内经过速率匹配输出的编码比特序列,所述参数 G 为一个时间单元内能够用于传输所述第一传输块和所述 UCI 的编码比特总数, G 为正整数,所述第二时间单元为所述第一时间单元在所述多个时间单元中的下一个时间单元。

- 10 9. 根据权利要求 8 所述的方法,其特征在于,所述网络设备还根据以下参数中至少一项确定第一编码比特序列的比特选择的起始比特在循环缓存中的位置:

在所述第一时间单元的速率匹配的比特选择的起始比特索引;

循环缓存长度;

循环缓存中填充比特的数量。

- 15 10. 根据权利要求 8 或 9 所述的方法,其特征在于,所述方法还包括:

所述网络设备根据所述第一时间单元执行比特选择跳过的填充比特的个数确定在所述第二时间单元上执行比特选择的起始比特索引。

11. 根据权利要求 8 至 10 任一项所述的方法,其特征在于,所述方法还包括:

- 20 所述网络设备根据所述第二时间单元对应参数 G' 确定在第二时间单元上执行比特选择的比特长度,所述参数 G' 为所述第二时间单元内仅可用于传输所述第一传输块的编码比特总数。

12. 根据权利要求 8 至 11 任一所述的方法,其特征在于,所述第一指示信息还用于指示所述终端设备能够在所述多个时间单元上发送第一传输块,包括:

- 25 当所述终端设备收到所述第一指示信息且所述第一指示信息指示的时间单元的数量大于 1 时,指示所述终端设备能够在所述多个时间单元上发送所述第一传输块。

13. 根据权利要求 12 所述的方法,其特征在于,所述多个时间单元的个数至少为 1, 2, 4, 8。

14. 一种上行传输的装置,其特征在于,包括:

- 30 收发模块,用于接收来自网络设备的第一指示信息,所述第一指示信息用于指示所述终端设备在多个时间单元上发送第一传输块;

所述收发模块,还用于接收来自所述网络设备的第二指示信息,所述第二指示信息用于指示所述终端设备在第一时间单元上发送上行控制信息 UCI,所述第一时间单元为所述多个时间单元中的一个;

- 35 处理模块,用于根据所述第一时间单元对应的参数 G 确定第一编码比特序列的比特选择的起始比特在循环缓存中的位置,所述第一编码比特序列为第二时间单元内经过速率匹配输出的编码比特序列,所述参数 G 为一个时间单元内可用于传输所述第一传输块和所述 UCI 的编码比特总数, G 为正整数,所述第二时间单元为所述第一时间单元在所述多个时间单元中的下一个时间单元;

- 40 所述收发模块,还用于在所述多个时间单元上向所述网络设备发送所述 UCI 和所述第一传输块。

15. 根据权利要求 14 所述的装置, 其特征在于, 所述收发模块, 用于在所述第一时间单元上向所述网络设备发送所述第一 UCI, 在所述多个时间单元上向所述网络设备发送所述 UCI 和所述第一传输块。

16. 根据权利要求 14 或 15 所述的装置, 其特征在于, 所述处理模块, 还用于根据以下参数中至少一项确定第一编码比特序列的比特选择的起始比特在循环缓存中的位置:
5 在所述第一时间单元的速率匹配的比特选择的起始比特索引;
循环缓存长度;
循环缓存中填充比特的数量。

17. 根据权利要求 14 或 16 所述的装置, 其特征在于, 所述方法还包括:
10 所述处理模块, 还用于当所述收发模块未在所述第一时间单元上向所述网络设备发送所述第一 UCI 时, 根据所述第一时间单元执行比特选择跳过的填充比特的个数确定在所述第二时间单元上执行比特选择的起始比特索引。

18. 根据权利要求 14 至 17 任一项所述的方法, 其特征在于, 所述方法还包括:
所述处理模块, 还用于根据所述第二时间单元对应参数 G' 确定在第二时间单元上执行比特选择的比特长度, 所述参数 G' 为所述第二时间单元内仅可用于传输所述第一传输块的编码比特总数。
15

19. 根据权利要求 14 至 18 任一所述的装置, 其特征在于, 所述第一指示信息还用于指示所述终端设备能够在所述多个时间单元上发送第一传输块, 包括:

当所述终端设备收到所述第一指示信息且所述第一指示信息指示的时间单元的数量大于 1 时, 指示所述终端设备能够在所述多个时间单元上发送所述第一传输块。
20

20. 根据权利要求 19 所述的装置, 其特征在于, 所述多个时间单元的个数至少为 1, 2, 4, 8。

21. 一种上行传输的装置, 其特征在于, 包括:

收发模块, 用于向终端设备发送第一指示信息, 所述第一指示信息用于指示所述终端设备在多个时间单元上发送第一传输块;
25

所述收发模块, 还用于向所述终端设备发送第二指示信息, 所述第二指示信息用于指示所述终端设备在第一时间单元上发送上行控制信息 UCI, 所述第一时间单元为所述多个时间单元中的一个;

所述收发模块, 还用于在所述多个时间单元上接收来自所述终端设备的所述第一传输块;
30

处理模块, 用于根据所述第一时间单元对应的参数 G 确定第一编码比特序列的比特选择的起始比特在循环缓存中的位置, 所述第一编码比特序列为第二时间单元内经过速率匹配输出的编码比特序列, 所述参数 G 为一个时间单元内能够用于传输所述第一传输块和所述 UCI 的编码比特总数, G 为正整数, 所述第二时间单元为所述第一时间单元在所述多个时间单元中的下一个时间单元。
35

22. 根据权利要求 21 所述的装置, 其特征在于, 所述处理模块, 还用于根据以下参数中至少一项确定第一编码比特序列的比特选择的起始比特在循环缓存中的位置:

在所述第一时间单元的速率匹配的比特选择的起始比特索引;
循环缓存长度;
40 循环缓存中填充比特的数量。

23. 根据权利要求 21 或 22 所述的装置, 其特征在于, 所述方法还包括:

所述处理模块, 还用于根据所述第一时间单元执行比特选择跳过的填充比特的个数确定在所述第二时间单元上执行比特选择的起始比特索引。

24. 根据权利要求 21 至 23 任一项所述的装置, 其特征在于, 所述方法还包括:

5 所述处理模块, 还用于根据所述第二时间单元对应参数 G' 确定在第二时间单元上执行比特选择的比特长度, 所述参数 G' 为所述第二时间单元内仅可用于传输所述第一传输块的编码比特总数。

25. 根据权利要求 21 至 24 任一所述的装置, 其特征在于, 所述第一指示信息还用于指示所述终端设备能够在所述多个时间单元上发送第一传输块, 包括:

10 当所述终端设备收到所述第一指示信息且所述第一指示信息指示的时间单元的数量大于 1 时, 指示所述终端设备能够在所述多个时间单元上发送所述第一传输块。

26. 根据权利要求 25 所述的装置, 其特征在于, 所述多个时间单元的个数至少为 1, 2, 4, 8。

27. 一种上行传输的方法, 其特征在于, 所述方法包括:

15 终端设备接收来自网络设备的下行控制信息和第一信息, 所述下行控制信息用于指示终端设备在第一时域资源上传输非周期的信道状态信息 A-CSI, 所述第一信息用于指示在第二时域资源上传输物理上行共享信道 PUSCH, 所述第一时域资源与所述第二时域资源重叠;

20 所述终端设备取消传输所述 A-CSI, 或者, 所述终端设备取消所述第一时域资源与所述第二时域资源重叠的时域资源上的 PUSCH 传输。

28. 一种上行传输的方法, 其特征在于, 所述方法包括:

终端设备接收来自网络设备的第一指示信息, 所述第一指示信息用于指示所述终端设备在多个时间单元上发送第一传输块;

25 所述终端设备接收来自所述网络设备的第二指示信息, 所述第二指示信息用于指示所述终端设备在第一时间单元上发送 HARQ-ACK 和第一控制信息, 所述第一控制信息包括信道状态信息 CSI 和/或配置授权上行控制信息 CG-UCI, 所述第一时间单元为所述多个时间单元中的一个;

30 所述终端设备根据所述第一时间单元对应的参数 G 确定第一编码比特序列的起始比特在循环缓存中的位置, 所述第一编码比特序列为第二时间单元内经过速率匹配输出的编码比特序列, 所述参数 G 为一个时间单元内可用于传输所述第一传输块和所述 HARQ-ACK 的编码比特总数, G 为正整数, 所述第二时间单元为所述第一时间单元在所述多个时间单元中的下一个时间单元;

所述终端设备在所述多个时间单元上向所述网络设备发送所述 HARQ-ACK、所述第一信息和所述第一传输块。

35 29. 一种上行传输的方法, 其特征在于, 所述方法包括:

终端设备接收来自网络设备的第一指示信息, 所述第一指示信息用于指示所述终端设备在多个时间单元上发送第一传输块;

40 所述终端设备接收来自所述网络设备的第二指示信息, 所述第二指示信息用于指示所述终端设备在第一时间单元上发送 HARQ-ACK, 所述第一时间单元为所述多个时间单元中的一个;

所述 HARQ-ACK 占用的比特数大于 2 比特时, 所述终端设备根据打孔的方式, 将所述 HARQ-ACK 复用到所述第一传输块;

所述终端设备在所述多个时间单元上向所述网络设备发送复用有所述 HARQ-ACK 的所述第一传输块。

- 5 30. 一种上行传输的方法, 其特征在于, 所述方法包括:

网络设备向终端设备发送第一信息, 所述第一信息用于指示所述终端设备使用第一传输方式发送一个传输块;

所述网络设备未接收到或未正确接收到所述传输块;

- 10 所述网络设备向所述终端设备发送第二信息, 所述第二信息用于指示所述终端设备使用第二传输方式发送所述传输块,

其中, 所述第一传输方式为基于单码块的物理上行共享信道 PUSCH 或者基于单码块的重复类型 A 的 PUSCH 重复, 所述第二传输方式为多时隙传输块或者多时隙传输块重复,

或者, 所述第一传输方式为基于多码块的物理上行共享信道 PUSCH 或者基于多码块的重复类型 A 的 PUSCH 重复, 所述第一传输方式与所述第二传输方式相同。

- 15 31. 一种上行传输的方法, 其特征在于, 包括:

终端设备接收来自网络设备的第一指示信息, 所述第一指示信息用于指示所述终端设备在多个时间单元上发送第一传输块;

所述终端设备接收来自所述网络设备的第二指示信息, 所述第二指示信息用于指示所述终端设备在第一时间单元上发送上行控制信息 UCI, 所述第一时间单元为所述多个时间单元中的一个;

所述终端设备根据所述第一时间单元对应的参数 G 确定第一编码比特序列的起始比特在循环缓存中的位置, 所述第一编码比特序列为第二时间单元内经过速率匹配输出的编码比特序列, 所述参数 G 为一个时间单元内可用于传输所述第一传输块和所述 UCI 的编码比特总数, G 为正整数, 所述第二时间单元为所述多个时间单元中的一个;

- 25 所述终端设备在所述多个时间单元上向所述网络设备发送所述 UCI 和所述第一传输块。

32. 根据权利要求 31 任一所述的方法, 其特征在于, 所述多个时间单元对应的参数 G 相等。

- 30 33. 根据权利要求 31 或 32 所述的方法, 其特征在于, 所述终端设备根据所述第一时间单元对应的参数 G 确定第一编码比特序列的起始比特在循环缓存中的位置, 包括:

根据所述第一时间单元对应的参数 G 和比特选择的真实周期 N'_{cb} , 确定所述第一编码比特序列的起始比特在循环缓存中的位置。

34. 根据权利要求 33 所述的方法, 其特征在于, 根据所述第一时间单元对应的参数 G 和比特选择的真实周期 N'_{cb} , 确定所述第一编码比特序列的起始比特在循环缓存中的位置

- 35 包括:
根据所述第一时间单元对应的参数 G 和比特选择的真实周期 N'_{cb} , 确定第一起始比特索引 $k'_{r,n}$;

根据所述第一起始比特索引 $k'_{r,n}$ 确定所述第一编码比特序列的起始比特在循环缓存中的位置 $k_{r,n}$;

- 40 其中, n 表示所述第二时间单元在所述多个时间单元中的索引, 所述索引是所述多个

时间单元按照时间顺序从 0 开始计数得到的。

35. 根据权利要求 34 所述的方法, 其特征在于, 所述第一起始比特索引 $k'_{r,n}$ 满足:

$$k'_{r,n} = \text{mod}(k'_{r,0} + n \times G, N'_{cb});$$

5 其中, 所述 $k'_{r,0}$ 表示所述多个时间单元中第一个时间单元的第一起始比特索引, 所述 N'_{cb} 表示比特选择的真实周期, n 表示所述第二时间单元在所述多个时间单元中的索引, 所述索引是所述多个时间单元按照时间顺序从 0 开始计数得到的。

36. 根据权利要求 34 或 35 所述的方法, 其特征在于, 所述根据所述第一起始比特索引 $k'_{r,n}$ 确定所述第一编码比特序列的起始比特在循环缓存中的位置 $k_{r,n}$ 包括:

10 若 $0 \leq k'_{r,n} < K'_{NULL} - 2Z_c$, 则 $k_{r,n} = k'_{r,n}$;
若 $K'_{NULL} - 2Z_c \leq k'_{r,n} < N'_{cb}$, 则 $k_{r,n} = k'_{r,n} + (K_{NULL} - K'_{NULL})$;
其中, $K'_{NULL} - 2Z_c$ 表示循环缓存中第一个填充比特的位置, $K_{NULL} - K'_{NULL}$ 表示循环缓存中填充比特的个数。

37. 根据权利要求 33-36 任一所述的方法, 其特征在于, 所述比特选择的真实周期 N'_{cb} 满足:

15 $N'_{cb} = N_{cb} - (K_{NULL} - K'_{NULL})$, 其中, $K_{NULL} - K'_{NULL}$ 表示循环缓存中填充比特的个数, N_{cb} 表示循环缓存的大小。

38. 根据权利要求 34-37 任一所述的方法, 其特征在于, 所述多个时间单元中第一个时间单元的第一起始比特索引 $k'_{r,0}$ 满足:

20 若 $0 \leq k_0 < K'_{NULL} - 2Z_c$, 则 $k'_{r,0} = k_0$; 其中, $K'_{NULL} - 2Z_c$ 表示循环缓存中第一个填充比特的索引;

若 $K'_{NULL} - 2Z_c \leq k_0 < K_{NULL} - 2Z_c$, 则 $k'_{r,0} = K'_{NULL} - 2Z_c$; 其中, $K_{NULL} - 2Z_c$ 表示循环缓存中第一个校验比特的索引;

若 $K_{NULL} - 2Z_c \leq k_0 < N_{cb}$, 则 $k'_{r,0} = k_0 - (K_{NULL} - K'_{NULL})$, 其中, $K_{NULL} - K'_{NULL}$ 表示循环缓存中填充比特的个数;

25 其中, k_0 表示所述多个时间单元中第一个时间单元对应的编码比特的起始比特在循环缓存中的位置, 所述 k_0 由冗余版本 RV 和基图 BG 共同确定。

39. 一种上行传输的方法, 其特征在于, 包括:

终端设备在 N 个可用时隙中向网络设备发送传输块, 所述传输块被配置在所述 N 个时隙中重复 M 次发送;

30 其中,

当冗余版本序列为 $\{0, 2, 3, 1\}$ 时, 所述传输块的初始传输开始于 $M \times N$ 个传输时机的第一个传输时机; 或者,

当冗余版本序列为 $\{0, 3, 0, 3\}$ 时, 所述传输块的初始传输开始于与 $RV=0$ 相关联的 $M \times N$ 个传输时机中的任何一个传输时机; 或者,

35 当冗余版本序列为 $\{0, 0, 0, 0\}$ 时, 所述传输块的初始传输开始于与 $RV=0$ 相关联的 $M \times N$ 个传输时机中的任何一个传输时机;

其中, $M \times N$ 个传输时机中的第 n 个传输时机, 与冗余版本序列中的第 $(\text{mod}((n-1)/N, 4)+1)$ 个值相关联, $n=1, N+1, 2N+1, \dots, (M-1) \times N+1$, 所述冗余版本序列是所述网络设备配置的。

40 40. 一种上行传输的方法, 其特征在于, 包括:

网络设备在 N 个可用时隙中接收来自终端设备的传输块,所述传输块被配置在所述 N 个时隙中重复 M 次发送;

其中,

5 当冗余版本序列为 $\{0,2,3,1\}$ 时,所述传输块的初始传输开始于 $M \times N$ 个传输时机的第一个传输时机;或者,

当冗余版本序列为 $\{0,3,0,3\}$ 时,所述传输块的初始传输开始于与 $RV=0$ 相关联的 $M \times N$ 个传输时机中的任何一个传输时机;或者,

当冗余版本序列为 $\{0,0,0,0\}$ 时,所述传输块的初始传输开始于与 $RV=0$ 相关联的 $M \times N$ 个传输时机中的任何一个传输时机;

10 其中, $M \times N$ 个传输时机中的第 n 个传输时机,与冗余版本序列中的第 $(\text{mod}((n-1)/N,4)+1)$ 个值相关联, $n=1, N+1, 2N+1, \dots, (M-1) \times N+1$, 所述冗余版本序列是所述网络设备配置的。

41. 一种通信装置,其特征在于,包括:

处理器和存储器;

15 所述存储器,用于存储计算机程序;

所述处理器,用于执行所述存储器中存储的计算机程序,以使得所述通信装置执行权利要求 1 至 7 中任一项所述的通信方法,或执行权利要求 8 至 13 中任一项所述的通信方法,或执行权利要求 27 至 40 中任一项所述的通信方法。

20 42. 一种计算机可读存储介质,其特征在于,所述计算机可读存储介质上存储有计算机程序,当所述计算机程序在计算机上运行时,使得所述计算机执行如权利要求 1 至 7 中任一项所述通信方法,或执行如权利要求 8 至 13 中任一项所述的通信方法,或执行权利要求 27 至 40 中任一项所述的通信方法。

25 43. 一种芯片系统,其特征在于,包括:处理器,用于从存储器中调用并运行计算机程序,使得安装有所述芯片系统地通信设备执行如权利要求 1 至 7 中任一项所述的通信方法,或执行如权利要求 8 至 13 中任一项所述的通信方法,或执行权利要求 27 至 40 中任一项所述的通信方法。