

PUSCH 跳频 (PUSCH frequency hopping)

LTE 中，PUSCH 传输通过跳频可以提高分集。在上行资源分配类型中已经介绍过，上行资源分配类型 1 通过将 2 个 RB 集合分布在频域上不同的位置来实现频域分集，且不支持 PUSCH 跳频。而上行资源分配类型 0 分配的是频域上连续的 VRB，需要通过跳频的方式将 VRB 映射到 PRB，从而实现频率分集。

PUSCH 跳频 (PUSCH frequency hopping) 的触发条件为：只有当 DCI format 0 中的“Frequency hopping flag”字段设置为 1，且使用上行资源分配类型 0 (“Resource allocation type”字段设置为 0) 时，UE 才会进行 PUSCH 跳频，否则 UE 不会进行 PUSCH 跳频。

DCI format 4 是不支持 PUSCH 跳频的。

LTE 中定义了 2 种 PUSCH 跳频类型：

- 类型 1 (Type 1)：基于 UL grant 中明确的跳频信息进行跳频；
- 类型 2 (Type 2)：根据小区特定的跳频/镜像样式进行基于 subband 的跳频。

首先，如何确定是使用类型 1 还是使用类型 2 的跳频呢？

DCI format 0 有一个资源分配字段 (“Resource block assignment and hopping resource allocation” 字段)，如果使能跳频的话，该字段最高 1 或 2 bit (MSB，用 $N_{UL_hop}^{UL}$ 表示，称之为 hopping bit) 是用于指示跳频信息，具体选择 1 bit 还是 2 bit 与上行系统带宽相关，如下表所示。（见 36.213 的 Table 8.4-1）

Table 8.4-1: Number of Hopping Bits N_{UL_hop} vs. System Bandwidth

System BW N_{RB}^{UL}	#Hopping bits for 2nd slot RA (N_{UL_hop})
6-49	1
50-110	2

hopping bit 用于指示使用的 PUSCH 跳频类型是类型 1 还是类型 2。如果使用的是类型 1，hopping bit 还决定了 $\tilde{n}_{PRB}(i)$ ，如下表所示。（见 36.213 的 Table 8.4-2）。

Table 8.4-2: PDCCH DCI Format 0 Hopping Bit Definition

System BW N_{RB}^{UL}	Number of Hopping bits	Information in hopping bits	$\tilde{n}_{PRB}(i)$
6 – 49	1	0	$\left(\left\lfloor N_{RB}^{PUSCH} / 2 \right\rfloor + \tilde{n}_{PRB}^{S1}(i) \right) \bmod N_{RB}^{PUSCH}$,
		1	Type 2 PUSCH Hopping
50 – 110	2	00	$\left(\left\lfloor N_{RB}^{PUSCH} / 4 \right\rfloor + \tilde{n}_{PRB}^{S1}(i) \right) \bmod N_{RB}^{PUSCH}$
		01	$\left(-\left\lfloor N_{RB}^{PUSCH} / 4 \right\rfloor + \tilde{n}_{PRB}^{S1}(i) \right) \bmod N_{RB}^{PUSCH}$
		10	$\left(\left\lfloor N_{RB}^{PUSCH} / 2 \right\rfloor + \tilde{n}_{PRB}^{S1}(i) \right) \bmod N_{RB}^{PUSCH}$
		11	Type 2 PUSCH Hopping

从上图可以看出 LTE 中可用的上行系统带宽与 hopping bit 取值以及 PUSCH 跳频类型的对应关系为：

上行系统带宽 (RB)	hopping bit 取值	跳频类型
6、15、25	0	Type 1
	1	Type 2
50、75、100	00	Type 1
	01	Type 1
	10	Type 1
	11	Type 2

DCI format 0 中的资源分配字段除去用于跳频信息的 1 或 2 bit 后，剩余的比特数为 $y = \left\lceil \log_2 (N_{RB}^{UL} (N_{RB}^{UL} + 1) / 2) \right\rceil - N_{UL_hop}$ 。剩余的 bit 携带的是 RIV 值，用于计算 RB_{START} 和 L_{CRBs} 。（详见上行资源分配类型 0 的介绍）

接着，我们来确定是跳频的范围？

既然是 PUSCH 跳频，那跳频的范围只能限定在那些用于 PUSCH 的 RB 上，而不能跳频到那些用于 PUCCH 的 RB 上，否则会把 PUCCH 的数据给破坏了。

在进行上行资源分配时，PUCCH 通常位于上行频域的两端。这里有 2 种可能：偶数个 RB pair 分配给 PUCCH，和奇数个 RB pair 分配给 PUCCH。下图是这 2 种情况的举例：

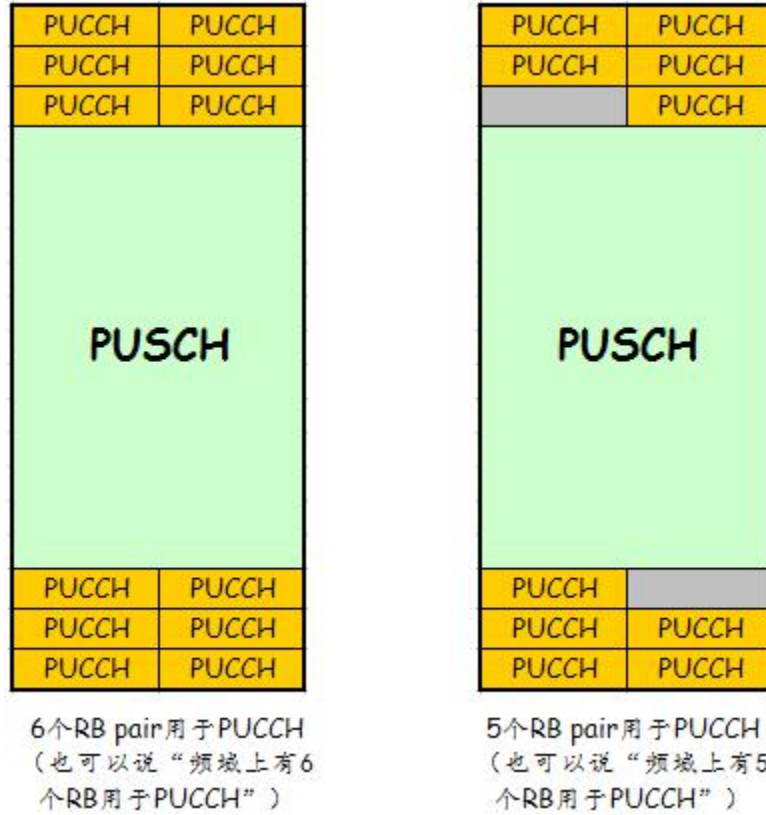


图 1: 上行频域分配举例

如果只分配了奇数个 RB pair 给 PUCCH，则会有额外的一个 RB pair 不能用于 PUSCH 跳频，如图中标记为灰色的部分。

为了计算出那些用于 PUSCH 的 RB，需要把 PUCCH 所占的 RB pair（以及灰色的部分）给剔除掉。

协议中使用 N_{RB}^{HO} 表示 PUCCH 在频域上所占的 RB 数 N_{RB}^{PUCCH} （或者说 RB pair 数），其值是通过 *pusch-hoppingOffset* 配置的。

需要剔除掉的不能用于 PUSCH 传输的 RB 数 \tilde{N}_{RB}^{HO} 的计算公式为：

$$\begin{cases} \tilde{N}_{RB}^{HO} = N_{RB}^{HO} + 1; & N_{RB}^{HO} \text{ 是奇数时} \\ \tilde{N}_{RB}^{HO} = N_{RB}^{HO}; & N_{RB}^{HO} \text{ 是偶数时} \end{cases}$$

可用于 PUSCH 传输的 RB 数 N_{RB}^{PUSCH} （称之为跳频带宽：hopping bandwidth）按如下方式计算：

$$N_{RB}^{PUSCH} = \begin{cases} N_{RB}^{UL} - \tilde{N}_{RB}^{HO} - (N_{RB}^{UL} \bmod 2) & \text{Type 1 PUSCH hopping} \\ N_{RB}^{UL} & \text{Type 2 } N_{sb} = 1 \text{ PUSCH hopping} \\ N_{RB}^{UL} - \tilde{N}_{RB}^{HO} & \text{Type 2 } N_{sb} > 1 \text{ PUSCH hopping} \end{cases}$$

从上面的公式也可以看出，跳频带宽并不能覆盖整个上行系统带宽，这是因为上行频带的两端要预留一部分 RB 给 PUCCH 使用。

进行 PUSCH 跳频的 UE 会从同一 TB 的最近一个 DCI format 0 的资源分配字段（“Resource block assignment and hopping resource allocation” 字段）中，确定用于一个子帧的第一个 slot ($S1$) 的 PUSCH 资源分配 (RA)，该 RA 包含了该子帧的最小索引 PRB (lowest index PRB) 信息 $n_{PRB}^{S1}(n)$ 。

注：“最近一个 DCI format 0” 主要是针对上行非自适应重传而言的。新传或上行自适应重传都有一个对应的 DCI，用于明确地指定如何进行传输。而对于上行非自适应重传，必须依赖对应同一 TB 的最近一个 DCI format 0 来决定如何进行 PUSCH 跳频。

如果针对同一 TB，没有相应的 PDCCH，UE 会基于如下方式确定其跳频类型：

- 如果同一 TB 的初始 PUSCH 传输是半静态调度的，则 UE 会从最近接收到的用于上行半静态调度的 PDCCH 中获取跳频信息；（该 PDCCH 中与跳频相关字段的解析与动态调度中的 DCI format 0 是一致的）
- 如果 PUSCH 传输是由 RAR 中的授权指定的，则 UE 会从用于同一 TB 的 RAR 中的授权中获取跳频信息。（可参见《随机接入过程》的介绍）

eNodeB 还可以通过小区特定的配置参数 *hoppingMode* 来设置上行跳频所支持的跳频模式 (hopping mode)：

- “intra and inter-subframe hopping”：即在“同一个子帧内和不同子帧之间”同时进行跳频。同一个子帧内的跳频是在 2 个 slot 之间进行的，它提供了一个 codeword 内的频率分集；
- “inter-subframe hopping”：在子帧之间进行跳频，同一子帧的 2 个 slot 之间是不进行跳频的。这种模式为同一 TB 的不同 HARQ 重传之间提供了频率分集。

PUSCH 跳频类型 1:

跳频类型 1 使用 DCI format 0 中明确指定的跳变偏移 (hopping offset) 进行跳频。

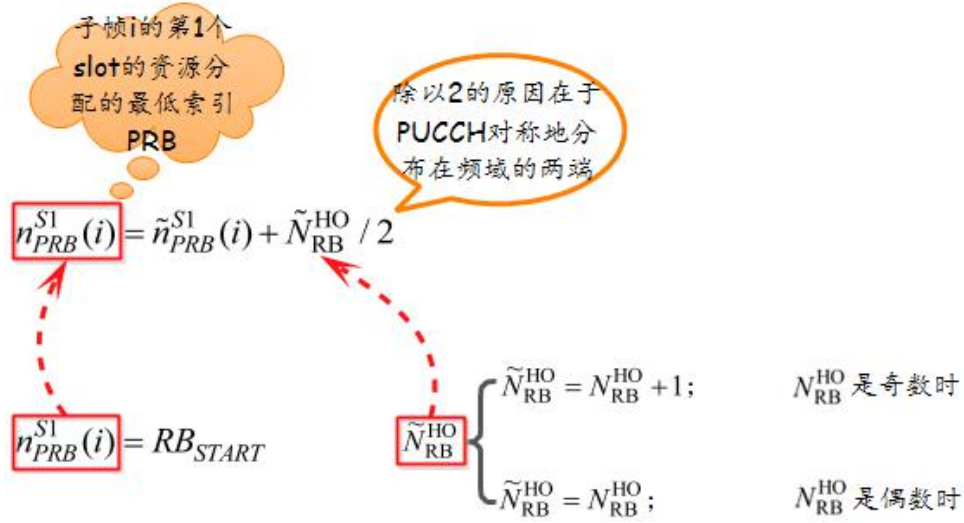


图 2: 第 1 个 slot 中起始 PRB 的计算

$n_{PRB}^{S1}(n)$ 是用于一个子帧的第一个 slot (S1) 的 PUSCH 资源分配的最小索引 PRB (lowest index PRB), 其值等于 RB_{START} 。 \tilde{n}_{PRB}^{S1} 是第一个 slot 中用于发送 PUSCH 的 RB 内 (用于发送 PUCCH 的 RB 不计算在内) 的最小索引 PRB, 其取值范围为 $0 \leq \tilde{n}_{PRB}^{S1} < N_{RB}^{PUSCH}$ (图 4 的例子中, 指出了 $\tilde{n}_{PRB}^{S1} = 0$ 和 $\tilde{n}_{PRB}^{S1} = N_{RB}^{PUSCH} - 1$ 所对应的 PRB)。

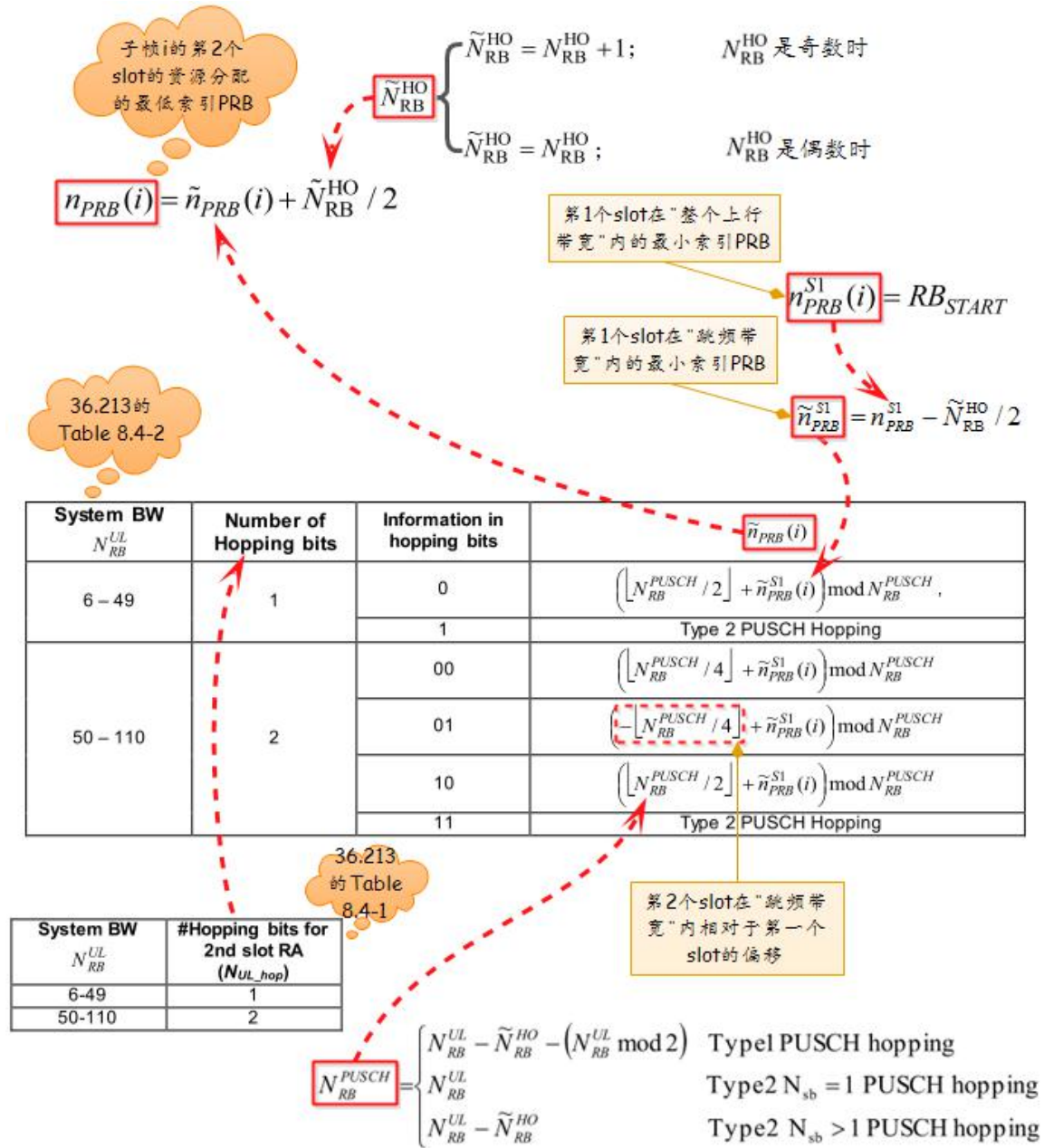


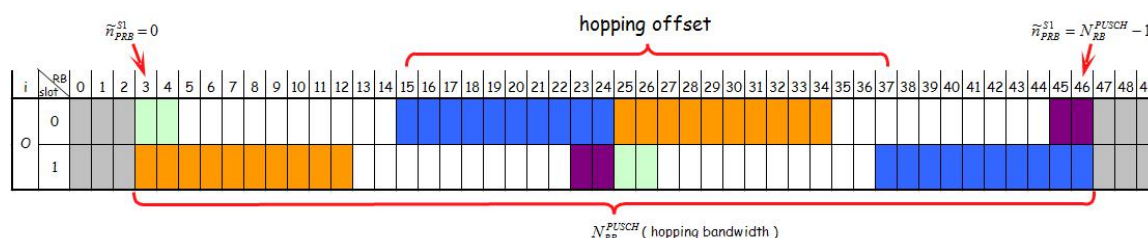
图 3：第 2 个 slot 中起始 PRB 的计算

图 3 是第 2 个 slot 的起始 PRB 的计算方式。可以看出，第二个 slot 是在第一个 slot 的基础上进行大约 1/2、+1/4、-1/4 跳频带宽的偏移。

用于 PUSCH 传输的 PRB 集合由 L_{CRBs} 个连续的 RB 组成，在第一个 slot 是从 PRB 索引 $n_{PRB}^{S1}(i)$ 开始的，在第二个 slot 是从 PRB 索引 $n_{PRB}(i)$ 开始的。

图 4 是在上行系统带宽 $N_{RB}^{UL} = 50$ ，6 个 RB pair 用于 PUCCH 传输，且 hopping bit 的取值为“10”的情况下，使用 PUSCH 跳频类型 1 的一个例子。

Type 1 PUSCH hopping (example)	
Field	Value
N_{RB}^{UL}	= 50
N_{RB}^{HO}	= 6
\tilde{N}_{RB}^{HO}	= $N_{RB}^{HO} = 6$
N_{UL_hop}	= 2 (假设这里取值为“10”)
y	= $\lceil \log_2 (N_{RB}^{UL} (N_{RB}^{UL} + 1) / 2) \rceil - N_{UL_hop} = 9$
N_{RB}^{PUSCH}	= $N_{RB}^{UL} - \tilde{N}_{RB}^{HO} - (N_{RB}^{UL} \bmod 2) = 50 - 6 - (50 \bmod 2) = 44$
可分配的最大连续 RB 数	$\lfloor 2^y / N_{RB}^{UL} \rfloor = \lfloor 512 / 50 \rfloor = 10$
Hopping offset	$\lfloor N_{RB}^{PUSCH} / 2 \rfloor = \lfloor 44 / 2 \rfloor = 22$



grey	PUCCH
orange	UE1
blue	UE2
purple	UE3
white	UE4

UE	RIV	RB_{START}	L_{CRB_1}	$n_{PRB}^{s1}(i)$	$\tilde{n}_{PRB}^{s1}(i)$	$n_{PRB}(i)$
UE1	53	3	2	3	0	25
UE2	465	15	10	15	12	37
UE3	475	25	10	25	22	3
UE4	95	45	2	45	42	23

图 4: Type 1 intra-subframe PUSCH hopping

PUSCH 跳频类型 1 的公式看似复杂，但其要表达的意思很简单：第一个 slot 不跳频。第二个 slot 在第一个 slot 的基础上，在“跳频带宽”范围内发生 $\lfloor N_{RB}^{PUSCH} / 2 \rfloor$ 、或 $\lfloor N_{RB}^{PUSCH} / 4 \rfloor$ 、或 $-\lfloor N_{RB}^{PUSCH} / 4 \rfloor$ 的偏移。

前面介绍的是基于“intra-subframe”的处理。如果 *hoppingMode* 配置成 “inter-subframe”（不同子帧之间），则第一个 slot 的 RA 应用于 CURRENT_TX_NB 为偶数的子帧，第二个 slot 的 RA 应用于 CURRENT_TX_NB 为奇数的子帧。

每个上行 HARQ process 都会维护一个状态变量 CURRENT_TX_NB，该变量表示当前 HARQ buffer 中的某个 MAC PDU 已经发生的传输次数（对应同一个 TB，包括初传和重传）。如果是新传，HARQ process 会将该变量设置为 0；每发生一次重传，该变量会加 1。当达到最大重传次数时，该变量会清零。（见 36.321 的 5.4.2.2 节）

PUSCH 跳频类型 2:

跳频类型 2 是根据小区特定的跳频/镜像样式（hopping/mirroring pattern）进行基于 subband 的跳频。其使用的是一个预先定义的与小区 PCI 相关的伪随机数跳频样式。

对于 PUSCH 跳频类型 2，跳频带宽（hopping bandwidth）会被划分为大小相等的多个 subband，每个 subband 由多个连续的 RB 组成。

subband 的个数 N_{sb} 是通过 *n-SB* 配置的。而每个 subband 包含的 RB 数 N_{RB}^{sb} 通过如下方式计算：

$$N_{RB}^{sb} = \begin{cases} N_{RB}^{UL} & N_{sb} = 1 \\ \left\lfloor (N_{RB}^{UL} - N_{RB}^{HO} - N_{RB}^{HO} \bmod 2) / N_{sb} \right\rfloor & N_{sb} > 1 \end{cases}$$

$(N_{RB}^{UL} - N_{RB}^{HO} - N_{RB}^{HO} \bmod 2)$ 计算的是可用于 PUSCH 传输的 RB 数 N_{RB}^{PUSCH} ，即跳频带宽所包含的 RB 数，其计算结果跟之前介绍的 N_{RB}^{PUSCH} 是一致的。

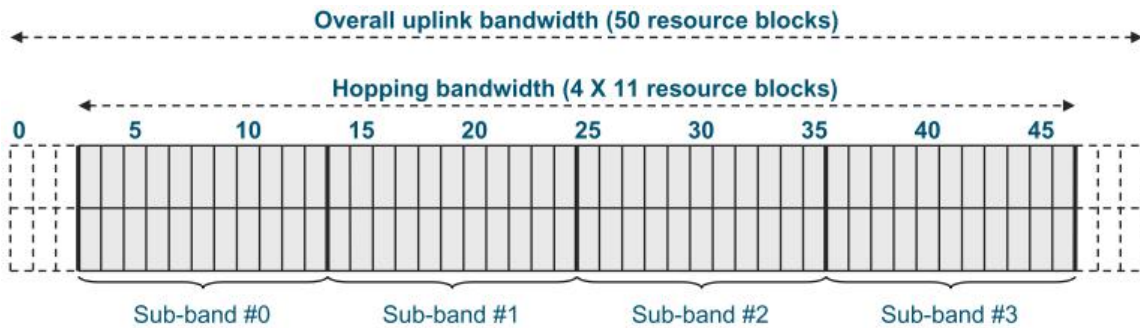


图 5: PUSCH 跳频中的 subband 划分的一个例子

图 5 是 subband 划分的一个例子。整个上行系统带宽 $N_{RB}^{UL} = 50$ ，配置了 4 个 subband，每个 subband 包含 11 个 RB，频域的两端共有 6 个 RB 用于传输 PUCCH。

PUSCH 跳频类型 2 中，用于 PUSCH 传输的 PRB，是通过将 UL grant 中指定的 VRB 偏移多个 subband 得到的，且不同的 slot 可以有不同的偏移量。

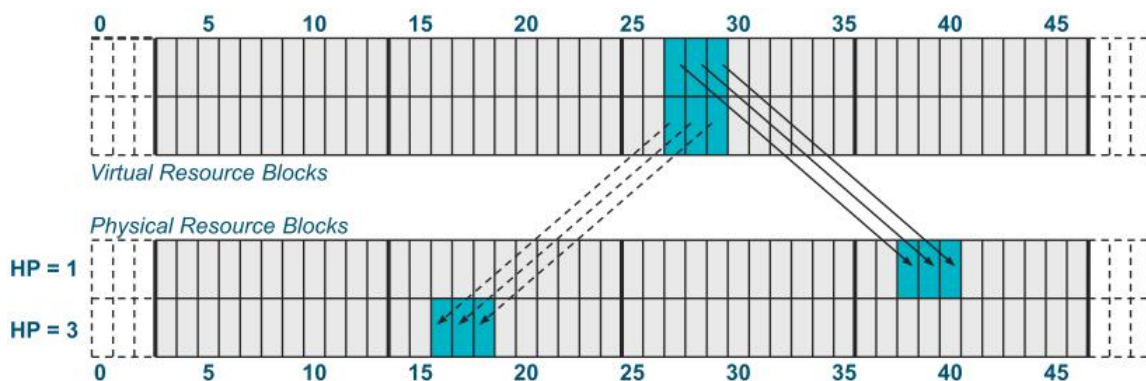


图 6: PUSCH 跳频类型 2 一个例子

图 6 是 PUSCH 跳频类型 2 的一个例子，一个 UE 被分配了 VRB 27、28 和 29。在第 1 个 slot，往右偏移 1 个 subband，得到 PRB 38、39 和 40。在第 2 个 slot，往右偏移 3 个 subband（等价于往左偏移一个 subband，即偏移是“环绕”的），得到 PRB 16、17 和 18。

由于跳频样式是小区特定的，即该小区内的所有 UE 都遵循相同的跳频样式，而不同的 UE 会分配非重叠的 VRB，因此其物理资源 PRB 也不会发生重叠。

除了跳频样式外，小区还定义了一个小区特定的镜像样式。 $f_m(i) \in \{0,1\}$ 决定是否使用镜像。镜像是基于一个 slot 生效的，如果某个 slot 使用镜像的话（ $f_m(i)=1$ ），则该 slot 的每个 subband 内的 RB 编号是“从右往左”依次递增的；如果某个 slot 不使用镜像的话（ $f_m(i)=0$ ），则该 slot 的每个 subband 中的 RB 编号是“从左往右”依次递增编号的。如图 6 所示。

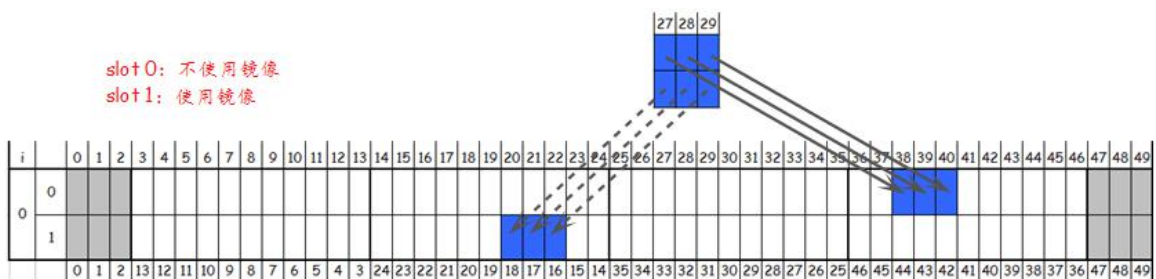


图 7：镜像的一个例子

上图是镜像的一个例子：第 1 个 slot 没有使用镜像，第 2 个 slot 使用了镜像。一个 UE 被分配了 VRB 27、28 和 29。在第 1 个 slot，往右偏移 1 个 subband，得到 PRB 38、39 和 40。在第 2 个 slot，往右偏移 3 个 subband（等价于往左偏移一个 subband，即偏移是“环绕”的），得到 PRB 16、17 和 18，但由于第二个 slot 使用了镜像，因此实际分配的是 PRB 20、21 和 22。

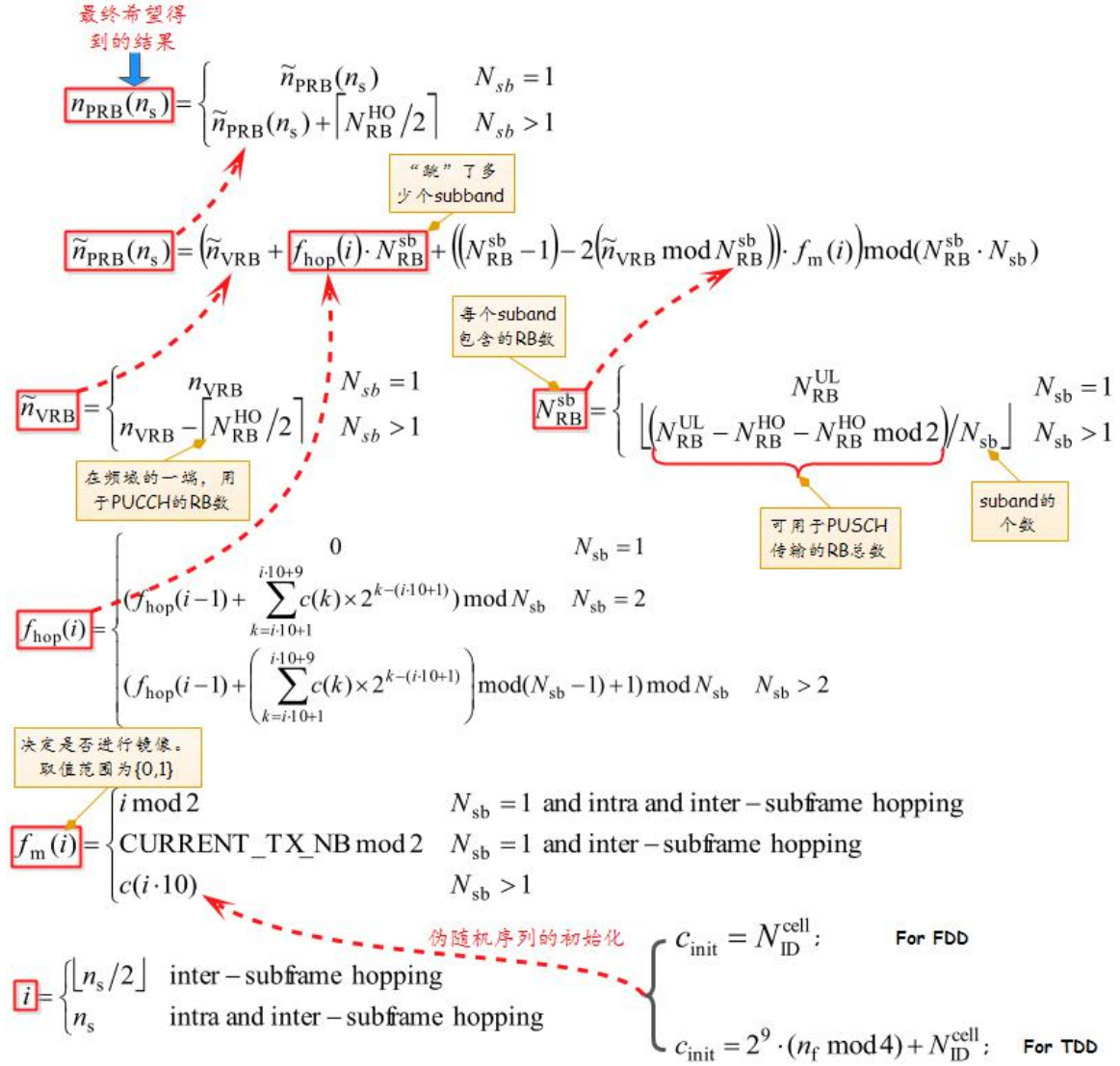


图 8: Type 2 PUSCH hopping

图 8 是 PUSCH 跳频类型 2 的 PRB 的计算公式，可以看出，跳频样式和镜像样式都与 PCI 相关，因此在邻居小区通常是不一样的。而且跳频/镜像样式的周期变化对应一个系统帧（10 ms）。

而且可以看出，如果 $N_{\text{sb}} = 1$ ，那么 $f_{\text{hop}}(i) = 0$ 时，只能进行镜像（当 $f_m(i) = 1$ 时）。

对于 TDD，如果 UE 还没有获取到 SFN，则 UE 不会传输跳频类型为 Type 2 且 $N_{\text{sb}} > 1$ 的 PUSCH。

对于跳频类型 2，同样可以使用 *hoppingMode* 配置了 PUSCH 跳频是“inter-subframe”的还是“intra and inter-subframe”的。

由于有 1 或 2 比特 (hopping bit) 需要被用于指示跳频信息, 因此限制了分配给某个 UE 的连续 RB 数:

- 可以分配给 Type 1 跳频的连续 RB 数不能大于 $\lfloor 2^y / N_{RB}^{UL} \rfloor$ 个。
- 可以分配给 Type 2 跳频的连续 RB 数不能大于 $\min(\lfloor 2^y / N_{RB}^{UL} \rfloor, \lfloor N_{RB}^{PUSCH} / n_{sb} \rfloor)$ 个。

可以看出, 对于 Type 2 跳频, 带宽一定的情况下, 如果 subband 的个数增加, 则可分配给一个 UE 的连续 RB 数将变少; 与此同时, 分配给一个 UE 的所有 RB 只能位于同一个 subband 中, 而不能跨 subband 分配。

类型 1 和类型 2 的比较:

从上面的介绍可以看出, 通过改变 DCI format 0 中的对应字段, 可以在跳频类型 1 和跳频类型 2 之间动态切换。

对于 PUSCH 跳频类型 1, 有 3 个不同的跳频选项 (偏移: 1/2、1/4、-1/4 跳频带宽), 在使用 “intra and inter-subframe hopping” 时, 其周期为 1 ms (即在同一子帧的 2 个 slot 之间进行跳频); 在使用 “inter-subframe hopping” 时, 其周期为 2 ms (即在同一 TB 的奇数次传输和偶数次传输之间进行跳频)。

而 PUSCH 跳频类型 2, 提供了使用不同样式的跳频和镜像, 且周期为 10 ms (即同一系统帧内的不同子帧间是不同的)。因此与 PUSCH 跳频类型 1 相比, 提供了更多的频率分集。

与此同时, PUSCH 跳频类型 2 是小区特定的, 而邻居小区的 PCI 通常是不同的, 这也可以为降低小区间干扰 (inter-cell interference) 提供帮助。

【参考资料】

- [1] TS 36.211 的 5.3.4 节
- [2] TS 36.213 的 8.4 节 UE PUSCH Hopping procedure
- [3] 《4G LTE/LTE-Advanced for Mobile Broadband》的 11.1.3 节
- [4] 《LTE - The UMTS Long Term Evolution, 2nd Edition》的 16.2.1.2 节
- [5] 《Clarification for PUSCH hopping type 1》from Ericsson
- [6] 《[Frequency Hopping in LTE Uplink](#)》
- [7] 《[LTE uplink frequency hopping](#)》