Pdsch

Thursday, August 8, 2024 11:38 AM

5G/NR - PDSCH

PDSCH (Physical Data Shared Channel) in a Nutshell:

- PDSCH is a downlink physical channel that deliver user data from gNB to UE. It has characterstics summarized as below.
- Resource Grid: PDSCH occupies a time-frequency grid of Resource Blocks (RBs) within a slot. Maximum number of PDSCH OFDM symbol within a slot is 14, but in most case it use less than 14 because usually one or more OFDM symbols are used for PDCCH.
- Scheduling: The gNB schedules PDSCH resources for UEs based on their channel quality, data requirements, and fairness considerations.
- Modulation: PDSCH supports various modulation schemes such as QPSK, 16QAM, 64QAM, and 256QAM to accommodate different channel conditions and data rates.
- Channel Coding: PDSCH uses LDPC (Low-Density Parity-Check) coding to provide forward error correction, enhancing the robustness of data transmission over the wireless channel.
- Layer Mapping : In MIMO (Multiple Input Multiple Output) systems, a PDSCH is distributed among multiple layers.
- Rate Matching: PDSCH employs rate matching to adjust the coded data rate to match the allocated resources, accommodating different UE requirements and channel conditions.
- DMRS (Demodulation Reference Signals): PDSCH includes DMRS to assist the UE in channel estimation and demodulation, ensuring accurate data reception

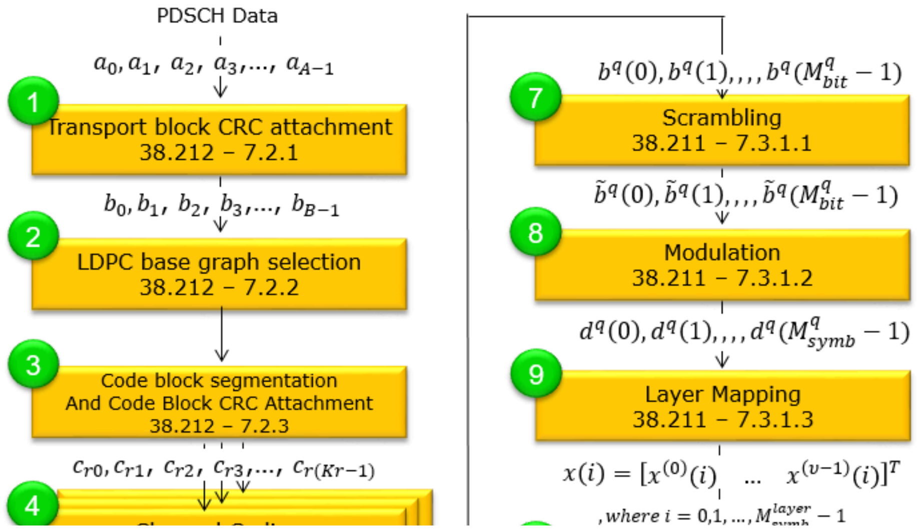
PDSCH (Physical Data Shared Channel) in detail:

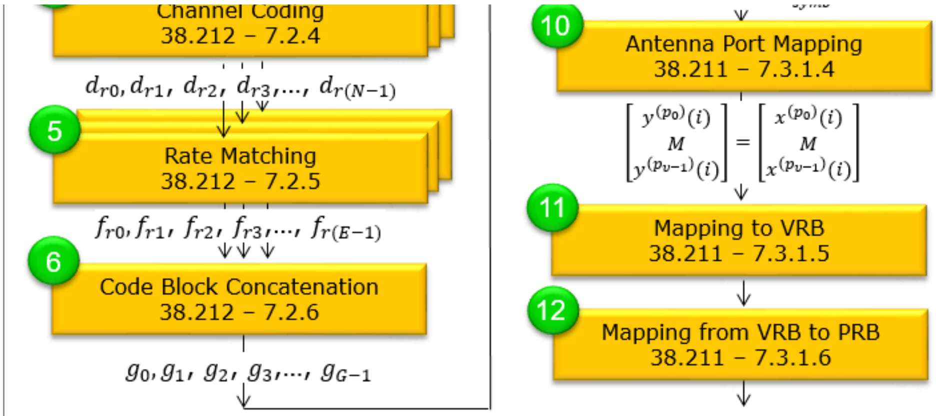
This page is about the process of converting user data into PDSCH data and transmit it through each transmission antenna. This would be one of the most complicated process in NR process and a lot of factors are involved in this process. Followings are those factors getting involved in this process. The critical (core part) is Transport Process and DCI and RRC is to provide (configure) some parameters for the transport process. In LTE, most of the transport parameters are fixed or automatically determined by transport process algorithm and only small numbers of parameters are configured by DCI but RRC message does not influence very much in the process. However, in NR many of the transport process parameters are provided (configure) not only by DCI but also by RRC message, meaning that the process would become more flexible but troubleshoot for the process will become more challenging.

- PDSCH Transport Process
  - (1) Transport block CRC attachment
  - (2) LDPC base graph selection
  - (3) Code block segmentation And Code Block CRC Attachment
  - (4) Channel Coding
  - (5) Rate Matching
  - (6) Code Block Concatenation
  - (7) Scrambling
  - (8) Modulation
  - (9) Layer Mapping
  - (10) Antenna Port Mapping
  - (11) Mapping to VRB
  - (12) Mapping from virtual to physical resource blocks

PDSCH Transport Process:

The PDSCH (Physical Downlink Shared Channel) Transport Process and shows corresponding 3GPP specification for each process.





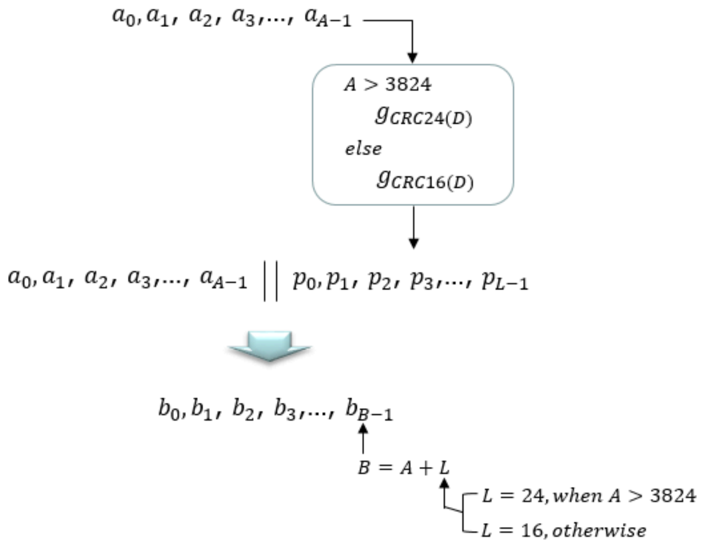
Following is brief summary for each step.

- **Transport Block CRC Attachment**: The PDSCH data $a_0, a_1, a_2, \ldots, a_{A-1}$ undergoes a CRC attachment process to detect errors at the receiver side.
- **LDPC Base Graph Selection**: After CRC attachment, the data is processed through LDPC (Low-Density Parity-Check) encoding for error correction purposes.
- **Code Block Segmentation and CRC Attachment**: The data is then segmented into smaller code blocks, and another CRC is attached to each block.
- **Channel Coding**: The code blocks are channel-coded to protect against errors during transmission.
- **Rate Matching**: The channel-coded data is adjusted to fit the available transmission bandwidth.
- **Code Block Concatenation**: The rate-matched code blocks are concatenated back into a single data stream.
- **Scrambling**: The concatenated data is scrambled to randomize the data pattern for security and interference mitigation.
- **Modulation**: The scrambled bits are then modulated into symbols suitable for transmission.
- **Layer Mapping**: The modulated symbols are mapped across multiple layers for MIMO transmission.
- **Antenna Port Mapping**: The layers are mapped to specific antenna ports.
- **Mapping to VRB (Virtual Resource Block)**: The data is mapped to virtual resource blocks within the frequency domain.
- **Mapping from VRB to PRB (Physical Resource Block)**: The virtual resource blocks are then mapped to physical resource blocks for actual transmission.

Each step in the process is vital for ensuring efficient and reliable communication over the PDSCH.

(1) Transport block CRC attachment

The transport block CRC attachment in 5G PDSCH channel processing is a step that allows the UE to detect errors in the received transport block, ensuring reliable data transmission over the wireless channel. a CRC is calculated for the transport block to enable error detection at the receiver (UE). The CRC is a fixed-size checksum generated by applying a polynomial function to the transport block data. In 5G NR, a 24-bit or 16 bit CRC is attached to the transport block depending on the size of the transport block.



Let's break this down into steps:

- The data from the transport block, represented as a sequence of bits $a_0, a_1, a_2, \ldots, a_{A-1}$, is prepared for CRC attachment to enable error detection at the receiver end.
- If the size of the transport block A is greater than 3824, a 24-bit CRC is attached using the generator polynomial $G_{CRC24}(D)$.
- If the size of the transport block A is less than or equal to 3824, a 16-bit CRC is used instead, with the polynomial $G_{CRC16}(D)$.
- The CRC is computed and appended to the data sequence, resulting in an extended sequence $a_0, a_1, a_2, \ldots, a_{A-1} \mid p_0, p_1, p_2, \ldots, p_{L-1}$.
- The length L of the CRC is set to 24 when A > 3824 and 16 otherwise, to accommodate the CRC bits.
- The resulting sequence after CRC attachment is represented as $b_0, b_1, b_2, \ldots, b_{B-1}$, where B = A + L, indicating the new length of the sequence.

This CRC attachment process is essential for ensuring reliable data transmission over the wireless channel by allowing error detection at the UE.

(2) LDPC base graph selection:

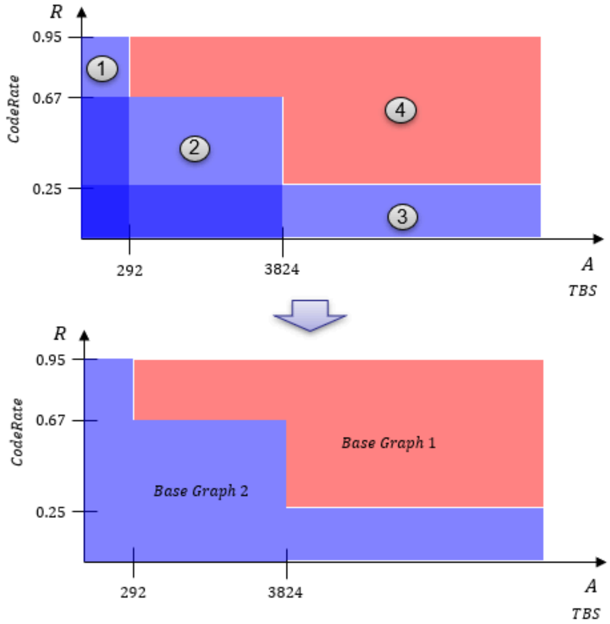
LDPC graph selection is the step that enables efficient channel coding tailored to the transport block size, ensuring reliable data transmission and optimized performance.

5G NR specifies two base graphs for LDPC encoding, known as Base Graph 1 and Base Graph 2. Each base graph has a predefined size, with Base Graph 1 being larger than Base Graph 2.

The selection of a base graph depends on the size of the transport block being transmitted over the PDSCH. If the transport block size is larger than a certain threshold, Base Graph 1 is used; otherwise, Base Graph 2 is employed. The smaller Base Graph 2 is more suitable for smaller transport blocks, as it offers a better trade-off between complexity and performance.
LDPC Base Graph type is determined by Transport Size (A) and Code Rate(R) based on following criteria.

$A \le 292 \longrightarrow LDPC\ base\ graph\ 2$ ①
$A \le 3824\ and\ R \le 0.67 \longrightarrow LDPC\ base\ graph\ 2$ ②
$R \le 0.25 \longrightarrow LDPC\ base\ graph\ 2$ ③
$else \longrightarrow LDPC\ base\ graph\ 1$ ④

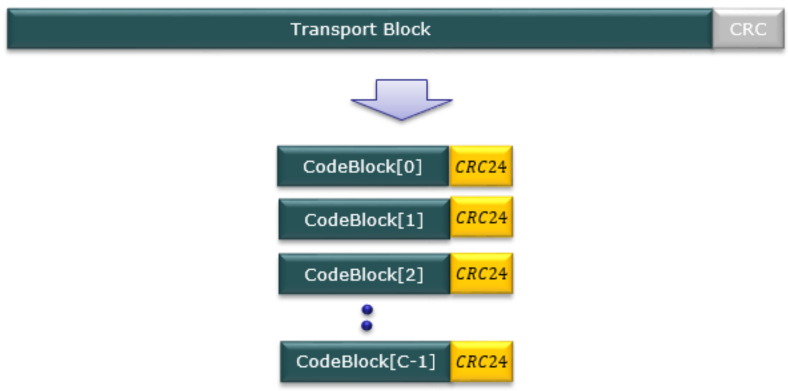
If I represent this as areas in coordinate, it would become as follows.



(3) Code block segmentation And Code Block CRC Attachment:

This step is to ensure efficient and reliable data transmission by dividing large transport blocks into smaller segments and providing error detection capabilities at the code block level. We can think of this with a few different perspectives/steps summarized below.

- **Code Block Segmentation**: If the size of a transport block is too large for efficient LDPC (Low-Density Parity-Check) coding, it is divided into smaller segments, called code blocks. The maximum size of a code block is defined by the 5G NR specifications. Segmentation is performed to ensure efficient channel coding and decoding while maintaining a reasonable complexity.
- **Segmentation Criteria**: The segmentation process is determined by comparing the transport block size with a specified maximum code block size. If the transport block size exceeds the maximum code block size, the transport block is divided into equal-sized code blocks (with the exception of the last code block, which may be smaller). If the transport block size is within the maximum code block size, no segmentation is performed.
- **Code Block CRC Attachment**: After segmentation, a CRC (Cyclic Redundancy Check) is calculated and attached to each code block individually. This 24-bit CRC allows for error detection at the receiver (UE) on a per-code-block basis.



-

The defailed steps for this process is as follows.
  i) Determine the max size of the code block (Kcb)
  : The max size of the code block depends on LDPC base graph type as follows.
  - For LDPC base graph type 1 : Kcb = 8448
  - For LDPC base graph type 2 : Kcb = 3840

  ii) Determine the number of Codeblocks
      if B(Transport block size) < Kcb(Max Codeblock size)
          L = 0
          C (number of codeblocks) = 1
          B' = B   // this mean 'No Segmentation'.
      else
          L = 24
          C = Ceiling(B/(Kcb - L))
          B' = B + C * L
  iii) Determine the number of bits in each code block
      K'(the number of bits in each code block) = B'/C
  iv) Determine Kb

```
        For LDPC base graph type 1
            Kb = 22
        For LDPC base graph type 2
            if B (Transport blocksize) > 640
                Kb = 10
            else if B (Transport blocksize) > 560
                Kb = 9
            else if B (Transport blocksize) > 192
                Kb = 8
            else
                Kb = 6
    v) find the minimum value of  Z in all sets of lifting sizes in [38.212-Table 5.3.2-1: Sets of LDPC lifting size]
    vi) denote Zc such that (Kb * Zc) >= K'
    vii) set K = 22 Zc for LDPC base graph 1
           K = 10 Zc for LDPC base graph 2
    viii) perform segmentation and add CRC bits
    s = 0  // s = bit position in B (transport block)
    for r = 0 to C-1
        for k = 0 to K'-L-1
            crk = bs
            s = s + 1
        end for
        if C > 1  // Do this if the number of the code block is more than 1
            Calculate pr0,pr1,pr2,...,pr(L-1) using the sequence cr0,cr1,cr2,...,cr(K'-L-1) and g_CRC24B(D)
            for k = K'-L to K'-1  // Append CRC bit
                crk = pr(k+L-K')
            end for
        end if
        for k = K' to K-1  // Insertion of filler bis
            crk = < NULL >
        end for
    end for
```

High level summary of the procedure listed above in pseudocode is as follows :

- Determine the maximum size of the code block (Kcb), which varies depending on the LDPC base graph type used.
- Decide on segmentation based on whether the transport block size exceeds Kcb, calculating the number of code blocks (C) and the size of each block (B').
- Calculate the number of bits in each code block by dividing the transport block size by C.
- Determine Kb, which is the number of bits within each code block, based on the LDPC base graph type and the transport block size.
- Find the minimum value of Z from the set of LDPC lifting sizes that satisfies the condition Kb ≤ Zc.
- For each code block, if C > 1, calculate the CRC bits and append them to the code block, inserting filler bits as necessary.

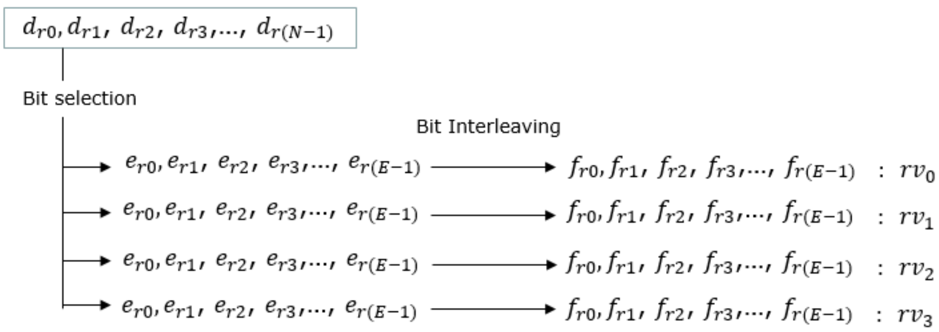
(4) Channel Coding:

Detailed procedure of LDPC as described in 38.212 - 5.3.2 which is out of scope of this section (Beyond my knowledge as well :). Overall process can be summarized as below.

- **Parity Check Matrix**: The LDPC codes are defined by a sparse parity-check matrix that represents the relationship between the data bits and the parity bits. 5G NR specifies two base graphs (Base Graph 1 and Base Graph 2) to construct the parity-check matrix, depending on the transport block size.
- **Encoding**: The LDPC encoding process takes the segmented code blocks (with attached CRC) as input and generates parity bits based on the chosen base graph and lifting factor. These parity bits are then appended to the original data bits, forming a codeword that is transmitted over the PDSCH.

(5) Rate Matching:

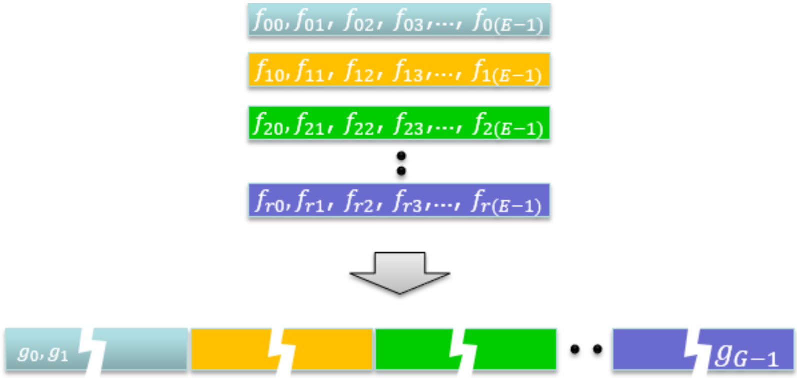
The Purpose of Rate matching is to adapts the output data rate of the channel encoder (LDPC) to match the available resources allocated for transmission in the time-frequency grid of the PDSCH. It can be describe in a few different steps as follows :

- **Bit Collection**: After LDPC coding, the encoded bits (data bits and parity bits) are collected in a circular buffer. The circular buffer is a temporary storage area with a fixed size that can hold bits in a circular manner, allowing for efficient bit selection.
- **Bit Selection**: Depending on the allocated PDSCH resources, a specific number of bits are selected from the circular buffer. The selection process involves three main operations: bit interleaving, bit pruning, and bit puncturing.
  - Bit Interleaving: Rearranges the order of the bits to improve the robustness against burst errors during transmission.
  - Bit Pruning: Removes any extra redundancy bits generated by the LDPC encoder
  - Bit Puncturing: Discards some of the encoded bits (usually parity bits) if the number of encoded bits exceeds the allocated resources.
- 

(6) Code Block Concatenation:

This is the step that combines the multiple code blocks resulting from the previous processing steps into a single data stream for transmission.

After rate matching, the processed code blocks are combined into a single data stream. The concatenation is performed in a specific order to ensure that the receiver (UE) can correctly separate and decode the individual code blocks. Typically, the code blocks are concatenated in the order they were segmented from the original transport block.



(7) Scrambling:

Scrambling process is the step that introduces randomness to the transmitted data, ensuring uniform power distribution, interference management, data privacy, and accurate channel estimation. Scrambling and descrambling operations are performed at the transmitter and receiver, respectively, using the same cell-specific scrambling sequence. Scrambling introduces randomness to the transmitted data by applying a pseudo-random binary sequence (PRBS) to the data stream. This operation ensures that the transmitted signal has a uniform power distribution across different frequency and time resources. Scrambling also aids in mitigating inter-cell interference, improving data privacy, and allowing the receiver (UE) to perform accurate channel estimation. Some of highlights of this process are :

- **Scrambling Sequence**: The scrambling process uses a cell-specific scrambling sequence generated based on the cell ID and a scrambling identity. The scrambling identity can be unique for each user (UE) within a cell, ensuring that the scrambling sequences used by different UEs are orthogonal to each other.
- **Bitwise XOR**: The scrambling process involves a bitwise exclusive-or (XOR) operation between the input data stream (resulting from the code block concatenation step) and the scrambling sequence. The output of this operation is a scrambled data stream.
- **Impact on Data Rate**: Scrambling does not change the data rate, as it only modifies the data stream by introducing randomness. The data rate is determined by other channel processing steps, such as LDPC encoding and rate matching.

CRNTI

$n_{\mathrm{ID}} \in \{0,1,\dots,1023\}$ equals the dataScramblingIdentity PDSCH if configured in PDSCH-Config

$n_{\mathrm{ID}} = N_{\mathrm{ID}}^{\mathrm{cell}}$ otherwise

$$c_{\mathrm{init}} = n_{\mathrm{RNTI}} \cdot 2^{15} + q \cdot 2^{14} + n_{\mathrm{ID}}$$

-

$$x_1(n+31) = (x_1(n+3) + x_1(n)) \bmod 2$$

$$x_2(n+31) = (x_2(n+3) + x_2(n+2) + x_2(n+1) + x_2(n)) \bmod 2$$

$$c(n) = (x_1(n+N_C) + x_2(n+N_C)) \bmod 2$$

$$\tilde{b}^{(q)}(i) = \left(b^{(q)}(i) + c^{(q)}(i)\right) \bmod 2$$

The high leve summary of the above illustration is as follows :

- The scrambling identity n_{ID} is determined based on configuration. If dataScramblingIdentity is configured in PDSCH-Config, n_{ID} takes a value from the set {0,1,...,1023}. Otherwise, n_{ID} is set to the physical cell ID N_{ID}^{cell}.
- The initialization sequence c_{init} is calculated using the formula $c_{init} = n_{RNTI} * 2^{15} + q * 2^{14} + n_{ID}$, where q is a quarter index.
- Two sequences $x_1(n)$ and $x_2(n)$ are generated, with each bit in the sequences updated using the given polynomials and modulo 2 arithmetic.
- The scrambling sequence *c(n)* is then produced by combining $x_1(n + N_c)$ and $x_2(n + N_c)$ also with modulo 2 arithmetic.
- Finally, each bit *b(i)* of the data sequence is scrambled with the corresponding bit *c(i)* from the scrambling sequence to produce the scrambled bit *b~(i)*.

(8) Modulation:

The modulation step is essential for converting the binary data stream into complex symbols suitable for wireless transmission. The choice of modulation scheme affects the data rate, spectral efficiency, and robustness against noise and interference. In 5G, 4 different modulation schemes are supported as of now.

- QPSK: QPSK modulates 2 bits per symbol, offering low data rates with high robustness against noise and interference.
- 16QAM: 16QAM modulates 4 bits per symbol, providing a balance between data rate and robustness.
- 64QAM: 64QAM modulates 6 bits per symbol, enabling higher data rates but with reduced robustness compared to QPSK and 16QAM.
- 256QAM: 256QAM modulates 8 bits per symbol, offering the highest data rates with the lowest robustness among the supported modulation schemes.

The choice of modulation scheme depends on factors such as channel conditions, link adaptation, and UE capabilities.

$\tilde{b}^{(q)}(0),\dots,\tilde{b}^{(q)}(M_{\mathrm{bit}}^{(q)}-1)$: Binary Sequence

38.211-Table 7.3.1.2-1

Modulation scheme	Modulation order Q_m
QPSK	2
16QAM	4
64QAM	6
256QAM	8

: 2 bits → 1 symbol (complex number)
: 4 bits → 1 symbol (complex number)
: 6 bits → 1 symbol (complex number)
: 8 bits → 1 symbol (complex number)

$d^{(q)}(0),...,d^{(q)}(M_{\text{symb}}^{(q)}-1)$: Complex Number Sequence

(9) Layer Mapping:

Layer mapping is the step that distributes the modulated symbols across one or multiple layers for transmission using multiple antennas. It aims to improve the spectral efficiency, reliability, and capacity of the wireless communication system by utilizing advanced antenna techniques such as MIMO and beamforming.
The number of layers depends on the availability of physical antenna, UE capability, and channel conditions. The 5G NR standard supports up to 8 layers per carrier as of now. The number of layers can be dynamically adapted based on the current system requirements and channel conditions.

Codeword index

$d^{(q)}(0),...,d^{(q)}(M_{\text{symb}}^{(q)}-1)$

38.211-Table 7.3.1.3-1

Layer index

$$x(i)=\left[x^{(0)}(i) \quad \ldots \quad x^{(\upsilon-1)}(i)\right]^T \qquad i=0,1,...,M_{\text{symb}}^{\text{layer}}-1$$

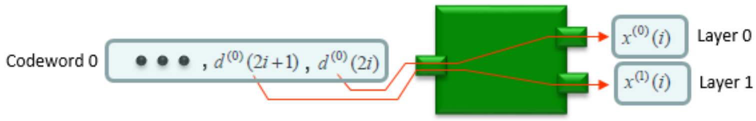
$$=\begin{bmatrix} x^{(0)}(0) & x^{(0)}(1) & \ldots & x^{(0)}(M_{symb}^{layer}-1) \\ x^{(1)}(0) & x^{(1)}(1) & & x^{(1)}(M_{symb}^{layer}-1) \\ & \vdots & \ddots & \vdots \\ x^{(v-1)}(0) & x^{(v-1)}(1) & \cdots & x^{(v-1)}(M_{symb}^{layer}-1) \end{bmatrix}$$

< 38.211 - Table 7.3.1.3-1: Codeword-to-layer mapping for spatial multiplexing. >

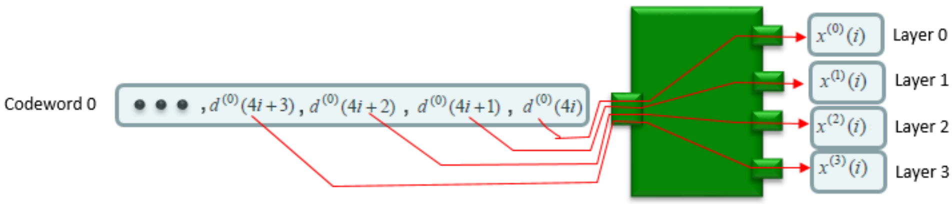
Number of layers	Number of codewords	Codeword-to-layer mapping $i=0,1,...,M_{\text{symb}}^{\text{layer}}-1$	
1	1	$x^{(0)}(i)=d^{(0)}(i)$	$M_{\text{symb}}^{\text{layer}}=M_{\text{symb}}^{(0)}$
2	1	$x^{(0)}(i)=d^{(0)}(2i)$ $x^{(1)}(i)=d^{(0)}(2i+1)$	$M_{\text{symb}}^{\text{layer}}=M_{\text{symb}}^{(0)}/2$
3	1	$x^{(0)}(i)=d^{(0)}(3i)$ $x^{(1)}(i)=d^{(0)}(3i+1)$ $x^{(2)}(i)=d^{(0)}(3i+2)$	$M_{\text{symb}}^{\text{layer}}=M_{\text{symb}}^{(0)}/3$
4	1	$x^{(0)}(i)=d^{(0)}(4i)$ $x^{(1)}(i)=d^{(0)}(4i+1)$ $x^{(2)}(i)=d^{(0)}(4i+2)$ $x^{(3)}(i)=d^{(0)}(4i+3)$	$M_{\text{symb}}^{\text{layer}}=M_{\text{symb}}^{(0)}/4$
5	2	$x^{(0)}(i)=d^{(0)}(2i)$ $x^{(1)}(i)=d^{(0)}(2i+1)$ $x^{(2)}(i)=d^{(1)}(3i)$ $x^{(3)}(i)=d^{(1)}(3i+1)$ $x^{(4)}(i)=d^{(1)}(3i+2)$	$M_{\text{symb}}^{\text{layer}}=M_{\text{symb}}^{(0)}/2=M_{\text{symb}}^{(1)}/3$
6	2	$x^{(0)}(i)=d^{(0)}(3i)$ $x^{(1)}(i)=d^{(0)}(3i+1)$ $x^{(2)}(i)=d^{(0)}(3i+2)$ $x^{(3)}(i)=d^{(1)}(3i)$ $x^{(4)}(i)=d^{(1)}(3i+1)$ $x^{(5)}(i)=d^{(1)}(3i+2)$	$M_{\text{symb}}^{\text{layer}}=M_{\text{symb}}^{(0)}/3=M_{\text{symb}}^{(1)}/3$
7	2	$x^{(0)}(i)=d^{(0)}(3i)$ $x^{(1)}(i)=d^{(0)}(3i+1)$ $x^{(2)}(i)=d^{(0)}(3i+2)$ $x^{(3)}(i)=d^{(1)}(4i)$ $x^{(4)}(i)=d^{(1)}(4i+1)$ $x^{(5)}(i)=d^{(1)}(4i+2)$ $x^{(6)}(i)=d^{(1)}(4i+3)$	$M_{\text{symb}}^{\text{layer}}=M_{\text{symb}}^{(0)}/3=M_{\text{symb}}^{(1)}/4$
8	2	$x^{(0)}(i)=d^{(0)}(4i)$ $x^{(1)}(i)=d^{(0)}(4i+1)$ $x^{(2)}(i)=d^{(0)}(4i+2)$ $x^{(3)}(i)=d^{(0)}(4i+3)$ $x^{(4)}(i)=d^{(1)}(4i)$ $x^{(5)}(i)=d^{(1)}(4i+1)$ $x^{(6)}(i)=d^{(1)}(4i+2)$ $x^{(7)}(i)=d^{(1)}(4i+3)$	$M_{\text{symb}}^{\text{layer}}=M_{\text{symb}}^{(0)}/4=M_{\text{symb}}^{(1)}/4$

Number of layers	Number of codewords	Codeword-to-layer mapping $i=0,1,...,M_{\text{symb}}^{\text{layer}}-1$
[illegible]	[illegible]	$x^{(0)}(i)=d^{(0)}(2i)$ [illegible]

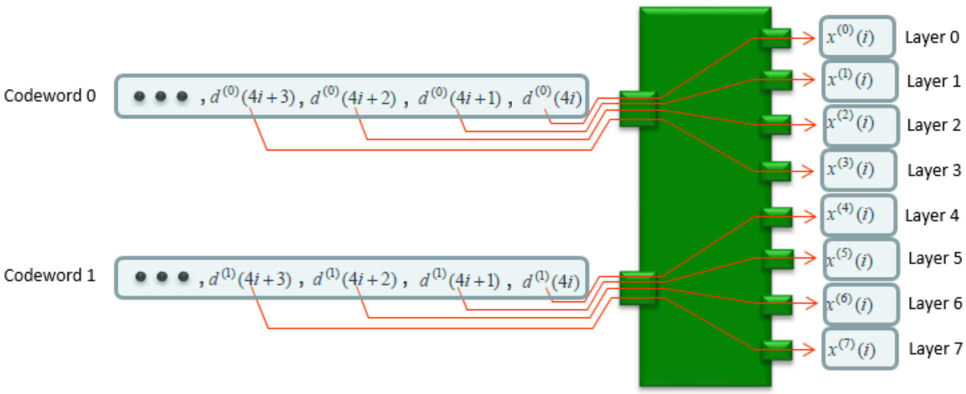
2	1	$x^{(1)}(i) = d^{(0)}(2i+1)$	$M^{layer}_{symb} = M^{(0)}_{symb}/2$



Number of layers	Number of codewords	Codeword-to-layer mapping $i = 0,1,\ldots, M^{layer}_{symb} - 1$	
4	1	$x^{(0)}(i) = d^{(0)}(4i)$ $x^{(1)}(i) = d^{(0)}(4i+1)$ $x^{(2)}(i) = d^{(0)}(4i+2)$ $x^{(3)}(i) = d^{(0)}(4i+3)$	$M^{layer}_{symb} = M^{(0)}_{symb}/4$



Number of layers	Number of codewords	Codeword-to-layer mapping $i = 0,1,\ldots, M^{layer}_{symb} - 1$	
8	2	$x^{(0)}(i) = d^{(0)}(4i)$ $x^{(1)}(i) = d^{(0)}(4i+1)$ $x^{(2)}(i) = d^{(0)}(4i+2)$ $x^{(3)}(i) = d^{(0)}(4i+3)$ $x^{(4)}(i) = d^{(1)}(4i)$ $x^{(5)}(i) = d^{(1)}(4i+1)$ $x^{(6)}(i) = d^{(1)}(4i+2)$ $x^{(7)}(i) = d^{(1)}(4i+3)$	$M^{layer}_{symb} = M^{(0)}_{symb}/4 = M^{(1)}_{symb}/4$



(10) Antenna Port Mapping:

Once the data path through the layer mapping process, the data from each layer are mapped to each Antenna Port. When CSI is not applied, the data maps to physical antenna port as below.

Antenna Port Number   Layer Number (index)

$$\begin{bmatrix} y^{(p_0)}(i) \\ M \\ y^{(p_{v-1})}(i) \end{bmatrix} = \begin{bmatrix} x^{(0)}(i) \\ M \\ x^{(v-1)}(i) \end{bmatrix}$$

When CSI is applied, the data from layer mapper are first mapped to each of CSI antenna port as shown below.

CSI Antenna Port Number   Layer Number (index)

$$\begin{bmatrix} y^{(3000)}(i) \\ M \\ y^{(3000+P-1)}(i) \end{bmatrix} = W(i) \begin{bmatrix} x^{(0)}(i) \\ M \\ x^{(v-1)}(i) \end{bmatrix}$$

Number of CSI-RS Ports   This matrix is determined by reportQuantity in RRC CSI ReportConfig, summarized in the table shown below

$P \in [1,2,4,8,12,16,24,32]$

The determination of W(i) from reportQuantity of CSI-ReportConfig can be summarized as shown below.

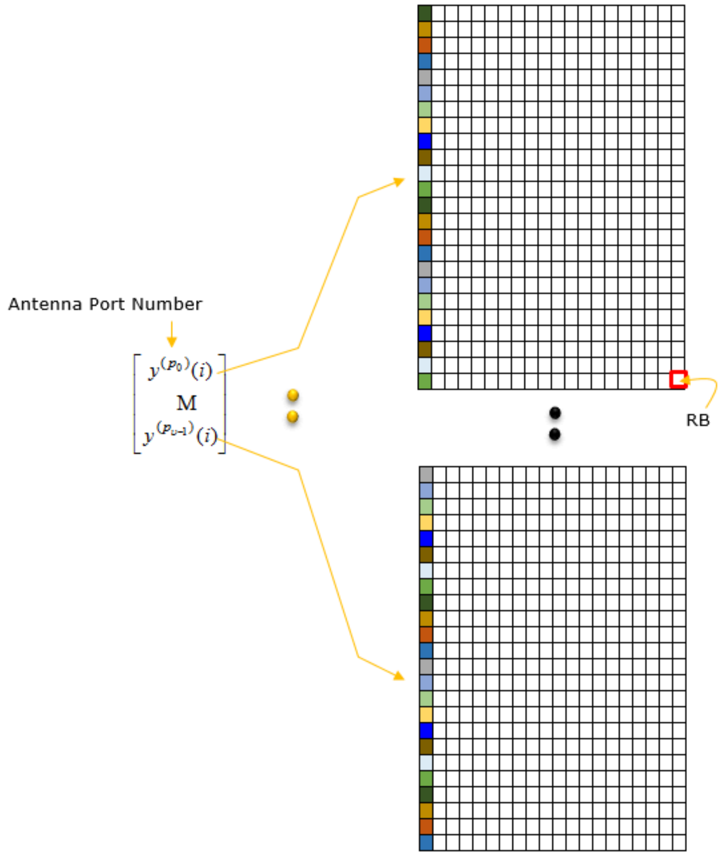
reportQuantity	W(i) determination criterial
cri-RI-PMI-CQI or cri-RI-LI-PMI-CQI	PMI Report from UE
cri-RI-CQI	The process described in 38.214-5.2.1.4.2
cri-RI-i1-CQI	i1 Report from UE

(11) Mapping to VRB:

For each antenna step, a virtual resource grid is created. Within the resource grid, fill out each of the resource elements(RE) with PDSCH data from the RE at the lowest frequency to higher frequency. Once it reaches the RE at the highest frequency of the assigned PDSCH resource block, move to the RE at the lowest frequency of next OFDM symbol.

But you should not use the REs that are assigned for following purpose :

- REs assigned for DMRS associated with the PDSCH to be transmitted
- REs assigned for DMRS intended for other co-scheduled UEs
- REs for non-zero-power CSI-RS, except for non-zero-power CSI-RSs configured by the higher-layer parameter CSI-RS-Resource-Mobility in the MeasObjectNR IE.
- REs for PTRS
- REs declared as 'not available for PDSCH

- 

(12) Mapping from virtual to physical resource blocks:

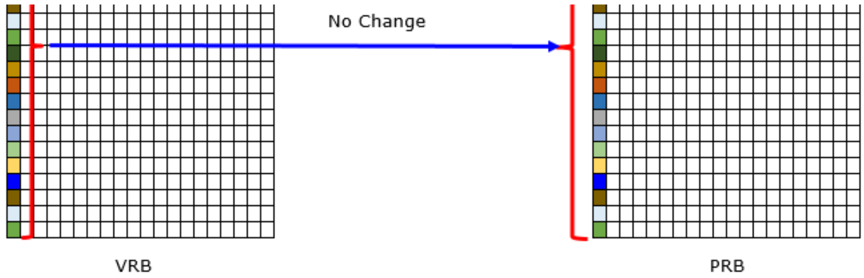
This step is the process that maps (converts) the virtual resource block(resource grid) into physical resource block(resource grid).

Two different types of mappings are supported in 5G as described below.

- **Non-Interleaved Mapping**: In non-interleaved mapping, the modulated and pre coded symbols are allocated to the physical resource blocks in a continuous and sequential manner. This strategy simplifies the mapping process but may result in reduced frequency diversity since adjacent symbols are transmitted on adjacent subcarriers.
    - Advantages: Non-interleaved mapping is simple and straightforward, making it easy to implement and manage.
    - Disadvantages: Due to the continuous allocation of symbols, non-interleaved mapping may be less resilient to frequency-selective fading and interference.
- **Interleaved Mapping**: In interleaved mapping, the modulated and precoded symbols are allocated to the physical resource blocks in a non-sequential and dispersed manner. This strategy increases frequency diversity by spreading the symbols across the available resources, improving resilience against frequency-selective fading and interference.
    - Advantages: Interleaved mapping provides better frequency diversity and is more robust against frequency-selective fading and interference, which can improve the overall system performance.
    - Disadvantages: The increased complexity of interleaved mapping can make it more challenging to implement and manage compared to non-interleaved mapping.

< Non-Interleaved >

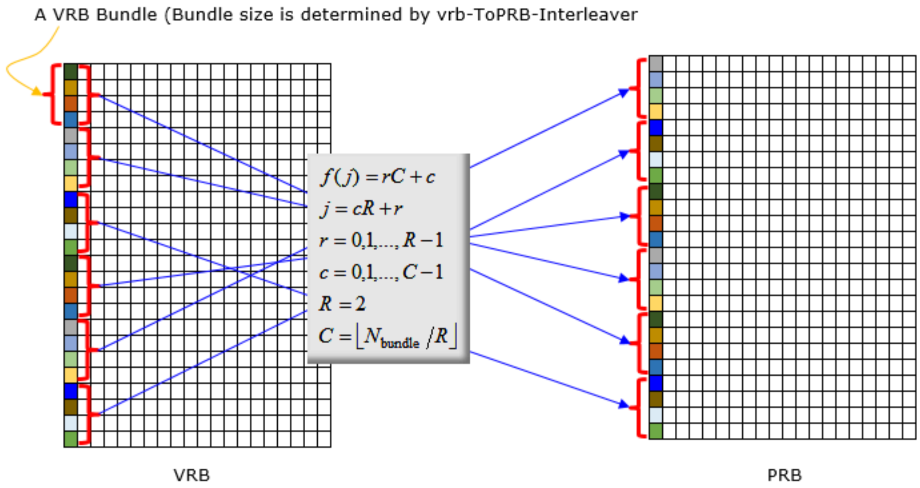
This is illustration of Non-Interleaved mapping from VRB to PRB



- **Definition:** Data or resources are grouped together in a contiguous manner.
- **Purpose:**
  - Simplify data management and access.
  - Optimize for specific processing patterns (e.g., sequential access).

< Interleaved >

This is illustration of Interleaved mapping from VRB to PRB



- **Definition:** Data or resources are distributed across multiple units or locations.
- **Purpose:**
  - Improve performance by reducing access time (e.g., interleaved memory).
  - Enhance error correction or fault tolerance (e.g., RAID).
  - Distribute load evenly (e.g., load balancing).