TB Segmentation

WKT - Maximum TB size = 3,19,874 bits
(for MCS-26, 275 RBs)

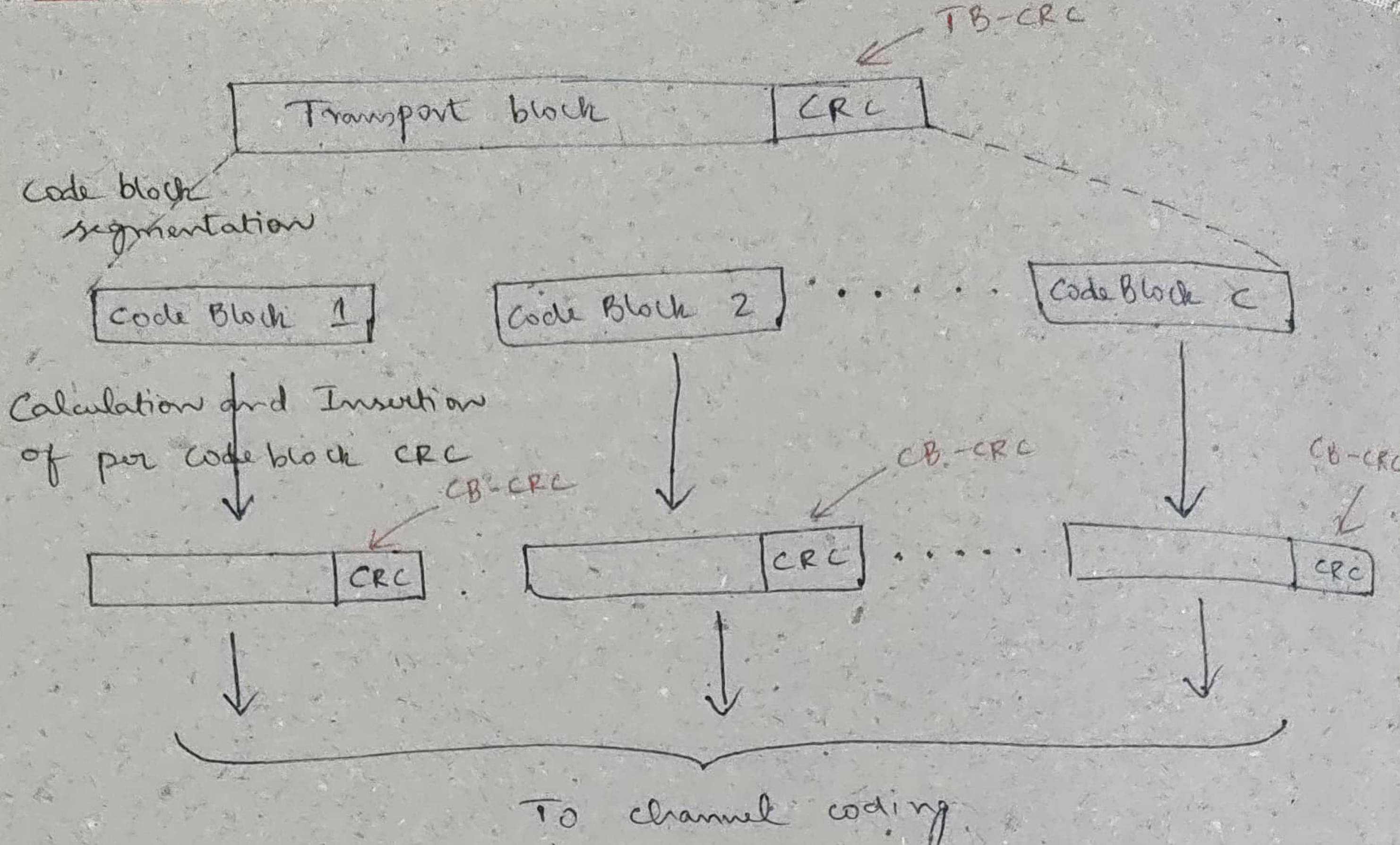
- Maximum input CB size which 5G LDPC encoder can process } K_{cb} = 8448 (or) 3840

- TB should be segmented if
  (TB length + 24-bit TB-CRC) = B > K_{cb}

- Why TB segmentation?
  Why not limit the maximum input CB size of 5G LDPC encoder to 3,19,874 bits?
  → Because, decoding complexity increases with increase in CB length
  → Running multiple 5G LDPC encoders/decoders in parallel, reduces the encoding/decoding time.

- CB-CRC is computed for each segmented code along with TB-CRC.
  → This allows error detection at the segmented CB level and request for their retransmission.

- Why do we need TB-CRC, when each segmented CB has CB-CRC?
  → Different polynomials for TB-CRC and CB-CRC is used.
  → This allows detection of any Residual errors.



- Total number of Segmented CB = C
- If C > 1, then CB-CRC of length (L = 24) is attached to each Code block.
- If C = 1, no CB-CRC is attached to the Code block.

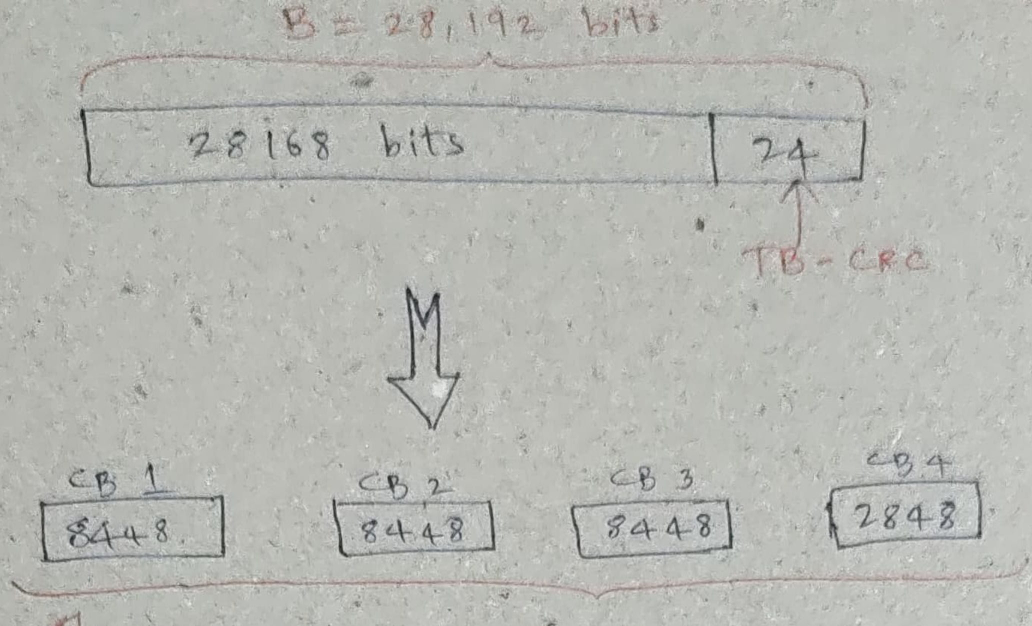
System configuration

→ Assume, a user is allocated 70 RBs over a slot of 14 symbols

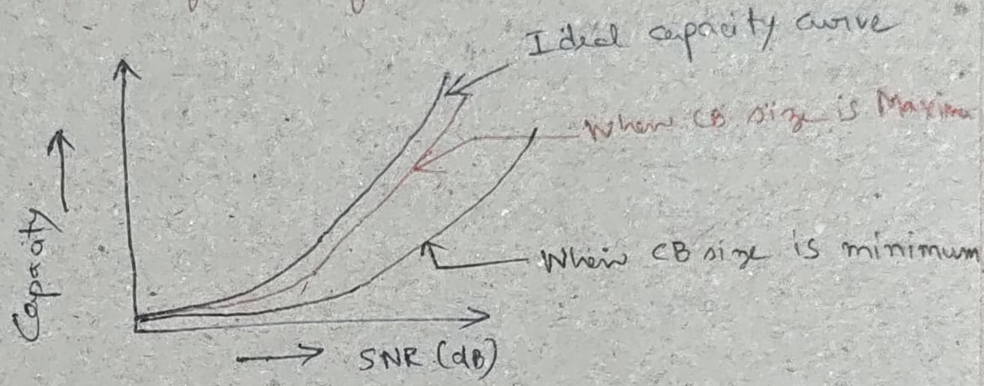
> Note: 1 RB contains 12 Subcarriers
> 70 RBs contain 70 × 12 = 840 Subcarriers

→ MCS-16 (16-QAM), which has a code rate of $\frac{658}{1024} = 0.642$

→ Assume MAC has sent a TB of size 28,168 bits to PHY.



Wrong way of Segmenting a TB.
Why ?



(ii) <u>For smaller CB size, the coding gain is less.</u>

- Hence, the Segmentation ensures that a TB is divided into equal sized CBs
- Also, the BLER performance is limited by the smaller CB size.

Segmentation details – as in the Standard

→ Total number of code blocks C is determined as below.

* if $(B > K_{cb})$, $L = 24$ then

$$C = \left\lceil \frac{B}{K_{cb} - L} \right\rceil$$

$$B' = B + C \times L$$

* if $(B \leq K_{cb})$, $L = 0$ then

$$C = 1$$

$$B' = B$$

where, $B' \to$ Effective payload length
$C \to$ Number of code blocks

→ Our Example,

28,168 bits	24 bits
TB	TB-CRC

$$\underbrace{\text{TB} + \text{TB-CRC}}_{B = 28{,}192 \text{ bits}}$$

Here, $B > K_{cb}$, $L = 24$ then

$$C = \left\lceil \frac{B}{K_{cb} - L} \right\rceil = \left\lceil \frac{28192}{8448 - 24} \right\rceil = 4$$

Not done like this...

$$B' = B + C \times L = 28192 + 4 \times 24 = 28288$$

$\therefore$ Each CB size, $K' = \left\lceil \frac{B'}{C} \right\rceil = \left\lceil \frac{28288}{4} \right\rceil = 7072$

7072	7072	7072	7072

LDPC Encoder details

→ An LDPC code is defined by parity check matrix H

→ Each codeword v is chosen such that $Hv = 0$.

(i.e) the output codeword lies in the NULL space of the parity check matrix H

$$\rightarrow \boxed{\text{Encoder}} \rightarrow \text{codeword}$$

rate, $r < 1$ always.

→ At the receiver side, (i.e) while decoding, a corrupted codeword (non-codeword) will generate a non-zero vector ($Hv \neq 0$), which is called syndrome.

→ 5G NR uses a Base graph matrix u, to construct the parity check matrix H.

<u>Example</u>

2x2 Basegraph Matrix

$$u = \begin{bmatrix} 2 & 3 \\ 0 & 1 \end{bmatrix}$$

$$I = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

3x3 Identity Matrix.

Parity check matrix H is generated, by permuting the column of the Identity Matrix I, according to the Basegraph Matrix u. (i.e).

$$H = \begin{bmatrix} \begin{bmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} & \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \\ \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} & \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 \end{bmatrix}$$

→ The Basegraph Matrix, u needs to be transformed into a Parity Check (PC) matrix, H using a lifting factor Z_c.

Note: Lifting ⇒ Each (integer) entry of basegraph matrix u is replaced by a permuted $Z_c \times Z_c$ identity matrix.

→ To obtain Parity Check (PC) matrix, H

* Start with an Identity matrix I and circularly shift its entries according to the basegraph entry u_{ij}
* We considered an example 2×2 base graph matrix u and lifting factor $Z_c = 3$.

Base Graph Parameters

→ NR data Channel (PUSCH/PDSCH) supports two Basegraphs to ensure good performance.

→ BaseGraph 1

* Optimized for large information block sizes and High code rates
* Fewer number of parity bits are added
* BG1 is used when SNR is good

→ BaseGraph 2

* Optimized for small information block sizes and Lower code rates.
* More number of parity bits are added
* BG2 is used when SNR is not good.

Parameter	Base Graph 1	Base graph 2
1. Minimum code rate, R_{min}	1/3	1/5
2. Base matrix size, u	46 x 68	42 x 52
3. No. of systematic columns, K_b	22	10
4. Maximum information Block size, K_{cb}	8448 (22 x 384)	3840 (10 x 384)
5. No. of non-zero elements	316	197

Table NR LDPC base matrix parameters
(from IEEE magazine)

→ Base Graph 1 is designed for maximum code rate of 8/9 and may be used for code rates upto 0.95

→ Base Graph 2 is for the lowest code rate.

→ Fewer non-zero elements in H indicates lower decoding complexity for a given code rate. So, for designing cheaper/sub-optimal phone, we'll end-up in using BG2.

⊙ BG2 has much lower decoding complexity than BG1.

Segmented Code block sizes

WKT Each CB size, $K' = \left\lceil \frac{B'}{C} \right\rceil = \left\lceil \frac{28288}{4} \right\rceil = 7072$

$\boxed{7072}$ $\boxed{7072}$ $\boxed{7072}$ $\boxed{7072}$

Not suitable for LDPC encoder

Not done like this... Then, how?

→ Standard specifies different lifting sizes to design parity check matrix H.

Set Index i_{LS}	Set of Lifting sizes Z_c
0	{2, 4, 8, 16, 32, 64, 128, 256}
1	{3, 6, 12, 24, 48, 96, 192, 384}
2	{5, 10, 20, 40, 80, 160, 320}
3	{7, 14, 28, 56, 112, 224}
4	{9, 18, 36, 72, 144, 288}
5	{11, 22, 44, 88, 176, 352}
6	{13, 26, 52, 104, 208}
7	{15, 30, 60, 120, 240}

Table 5.3.2-1 : Sets of LDPC Lifting Size Z_c
(38.212-f20.doc)

→ Take minimum value of Z_c from the above table such that,

$$K_b \times Z_c = K \geq K'$$

$$\Rightarrow 22 \times 352 = \underbrace{7744}_{K} \geq \underbrace{7072}_{K'}$$

→ When the segmented TB size is not matched to suitable lifting size, filler bits are added

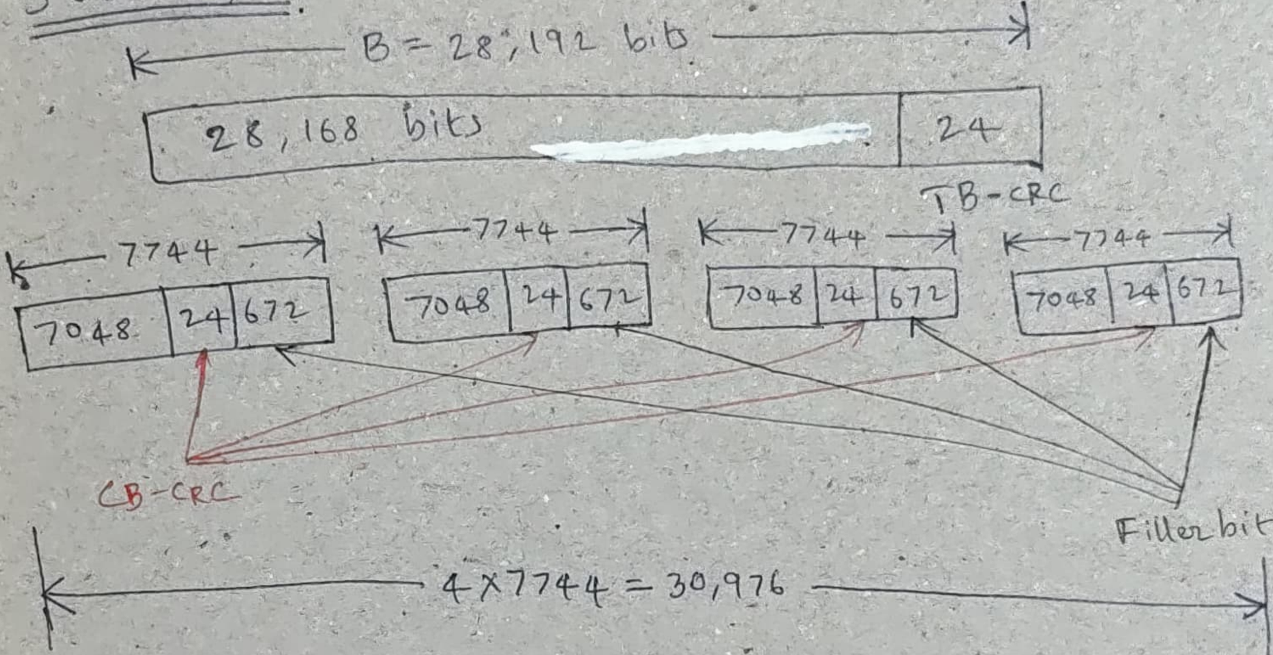
No. of filler bits, $F = K - K'$

$= 7744 - 7072$

$= 672$.

→ Total of 4 code blocks of size $K = 7744$ bits & corresponding set index i_{LS} } → Inputs to LDPC encoder.

SUMMARY



⊙ Input bit sequence to code block segmentation

$b_0, b_1, \ldots b_{B-1}$, where $B > 0$.

⊙ Bits output from code block segmentation:

$c_{r0}, c_{r1}, c_{r2}, c_{r3}, \ldots c_{r(K_r - 1)}$

where, $r \rightarrow$ code block number

$K_r \rightarrow$ No. of bits for r^{th} codeblock

⊙ Filler bits (usually denoted as -1) are added to the end of each code block

Transport block Segmentation as in standard

- No. of code blocks, $C = 4$
- Code block size without filler bits, $K' = 7072$ bits
- code block size with filler bits, $K = 7072 + 672$
  $= 7744$ bits
- CRC size, $L = 24$ bits

The bit sequence c_{rk} is calculated as

```
s = 0;
for r = 0 to C-1
    for k = 0 to K'-L-1
        c_rk = b_s ;
        s = s+1 ;
    end for
    if C > 1
        The sequence c_r0, c_r1, ..., c_r(K'-L-1) is
        used to calculate the CRC parity bits
        p_r0, p_r1, ..., p_r(L-1) according to Clause
        5.1 with the generator polynomial g_CRC24B(D).
        for k = K'-L to K'-1
            c_rk = p_r(k+L-K') ;
        end for
    end if
    for k = K' to K-1        -- Insertion of filler bits
        c_rk = <NULL> ;
    end for
end for
```

Reference : Section 5.2.2 of 38.212