5G PHY Layer Processing – Transport Block Segmentation (2)

Rohit Budhiraja

Simulation-Based Design of 5G Wireless Standards (EE698H)

Agenda for today

- Finish discussing transport block segmentation
  - Section 5.2.2 of 36.218
- Very very briefly discuss LDPC encoding
  - Reference – Chap 4.1.7.3 of 5G NR by SassanA
- Each code block size : $K' = \lceil B'/(C) \rceil = \frac{28288}{4} = 7072$. Not done like this.

LDPC Encoder details (1)

- An LDPC code is defined by parity check matrix H
- Each codeword $\boldsymbol{v}$ is chosen such that $\mathrm{H}\boldsymbol{v} = 0$
- A non-codeword (corrupted codeword) will generate a nonzero vector, which is called syndrome
- 5G NR uses a base graph matrix $\boldsymbol{u}$ to define the parity check matrix H

$$\mathbf{u} = \begin{bmatrix} 2 & 3 \\ 0 & 1 \end{bmatrix}, \quad \mathbf{I} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \rightarrow \mathbf{H} = \begin{bmatrix} \begin{bmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} & \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \\ \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} & \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix} \end{bmatrix} \rightarrow \mathbf{H} = \begin{bmatrix} 0 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 \end{bmatrix}$$

LDPC Encoder details (2)

- $\boldsymbol{u}$ needs to be transformed into a PC matrix H using a lifting factor Z_c
- Lifting – each (integer) entry of base graph $\boldsymbol{u}$ is replaced by a permuted $Z_c \times Z_c$ identity matrix
- To obtain H
  - Start with an identity matrix I and circularly shift its entries according to the base graph entry $\boldsymbol{u}_{ij}$
  - We considered an example 2×2 base graph matrix $\boldsymbol{u}$ and lifting factor $Z_c = 3$

LDPC lifting factor and base graph parameters (1)

- NR data channel supports two base graphs to ensure good performance
- Base graph 1 is optimized for large information block sizes and high code rates.
  - Designed for maximum code rate of 8/9 and may be used for code rates up to 0.95
- Base graph 2 is optimized for small information block sizes and lower code rates
  - Lowest code rate for base graph 2

Parameter	Base Graph 1	Base Graph 2
Minimum code rate R_{min}	1/3	1/5
Base matrix size	46×68	42×52
Number of systematic columns K_b	22	10
Maximum information block size K_{cb}	$8448(=22 \times 384)$	$3840(=10 \times 384)$
Number of non-zero elements	316	197

LDPC lifting factor and base graph parameters (2)

Parameter	Base Graph 1	Base Graph 2
Minimum code rate $R_{\min}$	1/3	1/5
Base matrix size	46×68	42×52
Number of systematic columns K_b	22	10
Maximum information block size K_{cb}	$8448 (= 22 \times 384)$	$3840 (= 10 \times 384)$
Number of non-zero elements	316	197

- Above table is Table 5.3.2-1 of 36.218-f20
- Fewer non-zero elements in H indicate lower decoding complexity for a given code rate
  - Base graph 2 has much lower decoding complexity than base graph 1

Segmented code block sizes (1)

- Each code block size : $K' = \lceil B'/(C) \rceil = \frac{28288}{4} = 7072$. Not done like this.
- 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}

- Minimum value of Z_c from Table 5.3.2-1 such that

$$\begin{aligned} K_b \times Z_c &= K \geq K' \\ \Rightarrow 22 \times 352 &= \underbrace{7744}_{K} \geq \underbrace{7072}_{K'} \end{aligned}$$

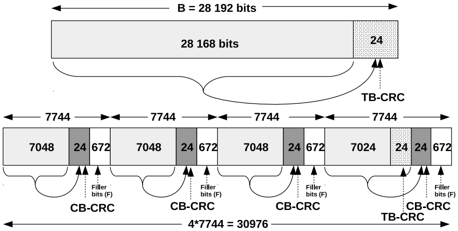
Segmented code block sizes (2)

- When segmented transport block size is not matched to suitable lifting size, filler bits are added
- Number of filler bits

$$F = K - K' = 7744 - 7072 = 672$$

- Total of four code block of size $K = 7744$ bits with filler bits $F = 672$
- Note the corresponding set index i_{LS} also - input to the LDPC encoder

Summary of transport block segmentation



- Input bit sequence to codeblock segmentation is denoted as:
  $b_0, \ldots, b_{B-1}, \quad B > 0$
- Bits output from codeblock segmentation are denoted as:
  $c_{r0}, c_{r1}, c_{r2}, c_{r3}, \ldots, c_{r(K_r-1)}$
  - r is codeblock number, and K_r is number of bits for rth codeblock
- Filler bits (usually denoted as -1) are added to the end of each codeblock

Transport block segmentation as in standard[1]

- Number 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

[1] 38.212 Sec 5.2.2

Transport block segmentation as in standard[2]

if $C > 1$

The sequence $c_{r0}, c_{r1}, c_{r2}, c_{r3}, \ldots, c_{r(K'-L-1)}$ is used to calculate the CRC parity bits $p_{r0}, p_{r1}, p_{r2}, \ldots, p_{r(L-1)}$ according to Subclause 5.1 with the generator polynomial $g_{\mathrm{CRC24B}}(D)$.

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

[2] 38.212 Sec 5.2.2