5G PHY Layer Processing – receive processing (2)

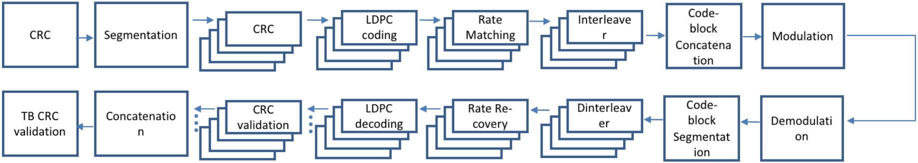
Rohit Budhiraja

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

Agenda for today

- Discuss receiver processing for the transmit chain discussed
- Discuss scrambling
    - References mentioned later in the slides

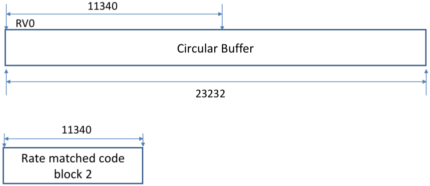
Receiver processing – rate recovery principles



- Rate matching
  - Puncturing/repeating bits to match the allocated resources
- Rate recovery
  - LDPC decoder works only for code rate of 1/3. Code rate should be reverted back to 1/3
  - Zeros are inserted in place of punctured bits
  - Repeated bits are combined

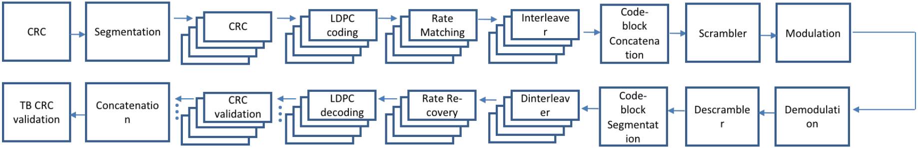
5G receiver Processing - rate recovery

- Rate recovery of code blocks e.g., second code block



- Write bits starting from RV0. Write neutral information for bits punctured at transmitter
- Feed rate recovered data to LDPC decoder. Validate CRC of each code block

5G transmit and receive chain upto scrambler



Scrambling principles

- Consider a concatenated block of G interleaved bits $b(0), \ldots, b(G-1)$
- Scrambled bits $\tilde{b}(0), \ldots, \tilde{b}(G-1)$ are given as

$$\tilde{b}(i) = b(i) \oplus c(i) \qquad i = 0, \ldots, (G-1)$$

- $c(i)$ is pseudo random sequence
- Example: 8-bit coded sequence $b = [0\ 0\ 0\ 0\ 0\ 0\ 0\ 0]$ and $c = [0\ 1\ 1\ 0\ 1\ 0\ 0\ 1]$

$$\tilde{b}(i) = [0\ 1\ 1\ 0\ 1\ 0\ 0\ 1]$$

- Scrambling is done to randomize the output of interleaver
  - both inner and outer signal points in the 16/64/256 QAM constellation to be used

De-scrambling principles

- For block of bits $b(0), \ldots, b(G-1)$, where G is the number of bits in code word
- Received scrambled bits $\tilde{b}(0), \ldots, \tilde{b}(G-1)$ were calculated as

$$\tilde{b}(i) = b(i) \oplus c(i) \qquad i = 0, \ldots, (G-1)$$

- Received descrambled bits $\tilde{b}(0), \ldots, \tilde{b}(G-1)$ can be recovered as

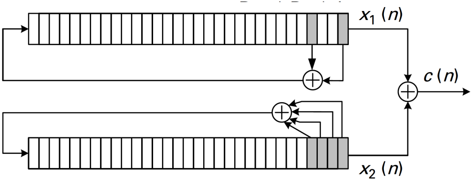
$$b(i) = \tilde{b}(i) \oplus c(i) \qquad i = 0, \ldots, (G-1)$$

- Example: 8-bit coded sequence $b = [0\ 0\ 0\ 0\ 0\ 0\ 0\ 0]$ and $c = [0\ 1\ 1\ 0\ 1\ 0\ 0\ 1]$

$$\begin{aligned} \tilde{b}(i) &= [0\ 1\ 1\ 0\ 1\ 0\ 0\ 1] \\ b(i) &= \tilde{b}(i) \oplus c(i) = [0\ 0\ 0\ 0\ 0\ 0\ 0\ 0] \end{aligned}$$

Pseudo random sequence generation in 5G[1] (1)

- Pseudo random sequences in 5G are defined by a length-31 Gold sequence



- Output sequence $c(n)$ of length G where $n = 0, 1, \ldots, G-1$

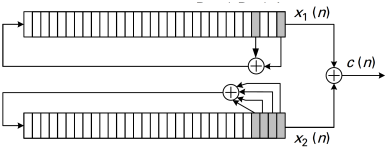
$$\begin{aligned} c(n) &= x_1(n) \oplus x_2(n) \\ x_1(n+31) &= x_1(n+3) \oplus x_1(n) \\ x_2(n+31) &= x_2(n+3) \oplus x_2(n+2) \oplus x_2(n+1) \oplus x_2(n) \end{aligned}$$

- Standard rejects first 1600 samples and uses $c'(n) = c(n+1600)$ instead

[1] Section 5.2 of 38.211

Pseudo random sequence generation in 5G (2)

- Pseudo random sequences in 5G are defined by a length-31 Gold sequence

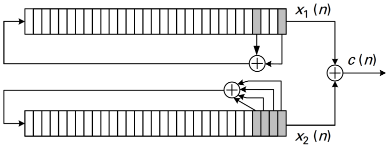


- Output sequence $c(n)$ of length G where $n = 0, 1, \ldots, G-1$
- First sequence $x_1(n)$ is initialized as

$$\begin{aligned} x_1(n) &= 1 \qquad n = 0 \\ &= 0 \qquad 0 < n \leq 30 \end{aligned}$$

Pseudo random sequence generation in 5G (3)

- Pseudo random sequences in 5G are defined by a length-31 Gold sequence



- Output sequence $c(n)$ of length G where $n = 0, 1, \ldots, G-1$
- Second sequence $x_2(n)$ is initialized by writing a constant c_{init} in binary form
- c_{init} is determined based on a cell ID and RNTI e.g, consider $c_{\text{init}} = 255$. Second sequence $x_2(n)$

$$
\begin{aligned}
x_2(n) &= 1 \qquad && n \leq 7 \\
&= 0 \qquad && 7 < n \leq 30
\end{aligned}
$$