УДК 004.728.3

Алгоритмы формирования и приема данных канала PBCH сетей 5G

В. Г. Дроздова[1], Д. В. Завьялова

New Radio – это сети 5-го поколения мобильной связи, активно внедряемые в коммерческую эксплуатацию по всему миру. В рамках данной статьи рассматриваются вопросы разработки программного обеспечения для формирования и приема данных широковещательного PBCH канала этих сетей, а также оценки надежности передачи данных по этому каналу, демонстрируется зависимость вероятности неуспешного декодирования блока PBCH от количества битовых ошибок. Для формирования данных канала используются алгоритмы открытых спецификаций 3GPP 38.212, для приема разрабатываются нестандартизованные алгоритмы.

Ключевые слова: 5G, NR, New Radio, PBCH, BLER, широковещательный канал.

1. Введение

В программе «Цифровая экономика Российской Федерации» [1] утверждена Концепция создания и развития сетей 5G/IMT2020 (5th Generation / Intern. Mobile Telecommunications) в Российской Федерации. В рамках данной концепции предусматривается разработка собственного оборудования и программного обеспечения для сотовых сетей 5G NR (New Radio). Причин для этого множество: отсутствие производителей, готовых выпускать оборудование для выделенного в России диапазона частот 4.8–4.9 ГГц, обеспечение конфиденциальности передаваемых данных, наращивание технологического преимущества и многое другое. В любом случае разработка отечественного программного обеспечения в соответствии с открытыми стандартами 3GPP (3d Generation Partnership Program) [2], а также оценка эффективности описанных спецификациями алгоритмов становится важнейшей задачей для достижения целей, сформулированных в программе [1]. В рамках данной статьи разработаны и реализованы на языке программирования C алгоритмы формирования данных широковещательного физического канала PBCH (Physical Broadcast Channel), такие как перемежение, скремблирование, полярное кодирование, заполнение блока до требуемого размера в соответствии со стандартом 38.212 [3]. Данный код полностью совместим с любым 5G-радиопередатчиком и может быть корректно интерпретирован любым 5G-радиоприемником. Кроме того, были разработаны алгоритмы приема сигнала, а также смоделированы битовые ошибки в блоке PBCH и оценена успешность восстановления блока в зависимости от числа таких ошибок.

Канал PBCH – это первый канал с системными параметрами, который считывает пользовательский терминал UE (User Equipment) после синхронизации с сотой. В нем содержатся важнейшие параметры, позволяющие UE принимать решение о том, можно ли получить доступ в Интернет с соты или нет, например, параметр *cellBarred* [4]. Успешность декодирования данных этого канала определяет задержку на установление соединения.

Актуальность темы оценки эффективности технологий NR вызывает интерес в научно-исследовательских кругах, появляются публикации, посвященные данной тематике. Например, автор работы [5] выполнял исследование зависимостей вероятности битовой ошибки

[1] Исследование выполнено при финансовой поддержке гранта Президента Российской Федерации для государственной поддержки молодых ученых – кандидатов наук МК-1047.2020.9.

BER (Bit Error Rate) и вероятности блоковой ошибки BLER (Block Error Rate) канала PBCH от отношения уровня сигнала к уровню интерференции и шуму SINR (Signal to Noise and Interference Ratio) для канала с АБГШ (аддитивный белый гауссовский шум). Однако результаты, описанные автором, были получены лишь для одного сценария и конкретной конфигурации соты (частота 2 ГГц, нумерология 0, т.е. 15 кГц – разнос между поднесущими и 50 МГц – ширина полосы частот).

В статье [6] авторами демонстрируется, как можно использовать распределенную контрольную сумму CRC (Cyclic Redundancy Check) для увеличения успешности декодирования канала PBCH. В работе [7] авторы акцентируют внимание на возможности снижения BLER в канале PBCH за счет совместной оценки и демодуляции данных с помощью сигналов синхронизации SS (Synchronization Signals) и демодуляционных сигналов DMRS (Demodulation Reference Signals).

Для того чтобы иметь возможность работать с более общими характеристиками надежности широковещательного канала (независимые от сценария), в рамках данной статьи была получена зависимость BLER от количества битовых ошибок в одном блоке, которую не составит труда при необходимости преобразовать в зависимость от SINR или BER для любого требуемого сценария, канала или конфигурации соты.

2. Формирование и прием данных канала PBCH

Физический широковещательный канал PBCH предназначен для передачи данных главного информационного блока MIB (Master Information Block) и передается совместно с сигналами первичной и вторичной синхронизации PSS и SSS (Primary, Secondary Synchronization Signals), образуя блок синхронизации SSB (Synchronization Signal Block), с периодичностью 20 мс, как показано на рис. 1.

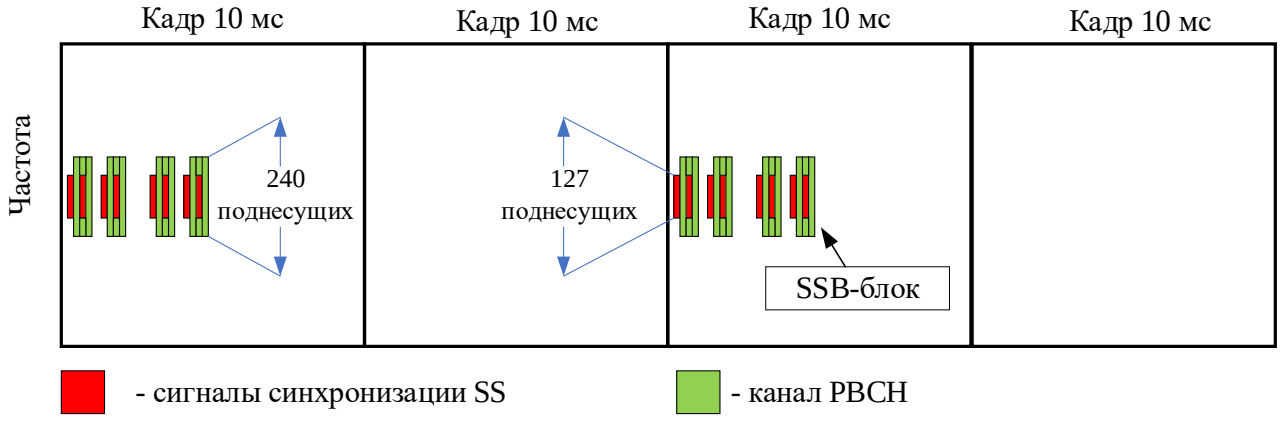

Рис. 1. Пример конфигурации SSB-блоков для 4 лучей

Рассмотрим наполнение PBCH-канала и структуру MIB, представленную в спецификациях [4]. Прежде всего, PBCH – это физический канал, в который передаются данные транспортного широковещательного канала управления BCCH (Broadcast Control Channel), предназначенного, в свою очередь, для передачи информации логического широковещательного канала BCH (Broadcast Channel). На рис. 2 показано, какие именно параметры используются для формирования PBCH-канала. Всего в нем передается 24 бита полезной информации (payload), первый бит из которых определяет тип BCH-сообщения, а оставшиеся 23 бита – системные параметры (*systemFrameNumber* – SFN, номер системного кадра; *ssb-subcarrierOffset* – разнос между поднесущими SSB-блока; *dmrs-TypeA-position* – позиция DMRS-сигналов; *pdcchConfigSib1* – конфигурация канала управления для передачи первого системного информационного блока; *cellBarred* – информирует, закрыта ли сота для доступа; *intraFreqReselection* – разрешен ли перевыбор соты на несущей).

BCCH-BCH-Message

```
-- ASN1START
-- TAG-BCCH-BCH-MESSAGE-START

BCCH-BCH-Message ::=          SEQUENCE {
    message                       BCCH-BCH-MessageType  - 1 бит
}

BCCH-BCH-MessageType ::=      CHOICE {
    mib                           MIB,
    messageClassExtension         SEQUENCE {}
}

-- TAG-BCCH-BCH-MESSAGE-STOP
-- ASN1STOP
```

MIB

```
-- ASN1START
-- TAG-MIB-START

MIB ::=                                SEQUENCE {
    systemFrameNumber                      BIT STRING (SIZE (6)),                  - 6 бит
    subCarrierSpacingCommon                ENUMERATED {scs15or60, scs30or120},     - 1 бит
    ssb-SubcarrierOffset                   INTEGER (0..15),                        - 4 бита
    dmrs-TypeA-Position                    ENUMERATED {pos2, pos3},                - 1 бит
    pdcch-ConfigSIB1                       PDCCH-ConfigSIB1,                       - 8 бит
    cellBarred                             ENUMERATED {barred, notBarred},         - 1 бит
    intraFreqReselection                   ENUMERATED {allowed, notAllowed}        - 1 бит
    spare                                  BIT STRING (SIZE (1))                   - 1 бит
}

                                                                    Всего:23 бита+1=24

-- TAG-MIB-STOP
-- ASN1STOP
```

Рис. 2. Содержимое PBCH-канала [4]

На рис. 3 показана общая схема процедуры формирования и приема данных канала PBCH, вплоть до подачи на OFDM-передатчик (Orthogonal Frequency Division Multiplexing) и распараллеливания последовательного битого потока между модуляторами поднесущих (modulation mappers).

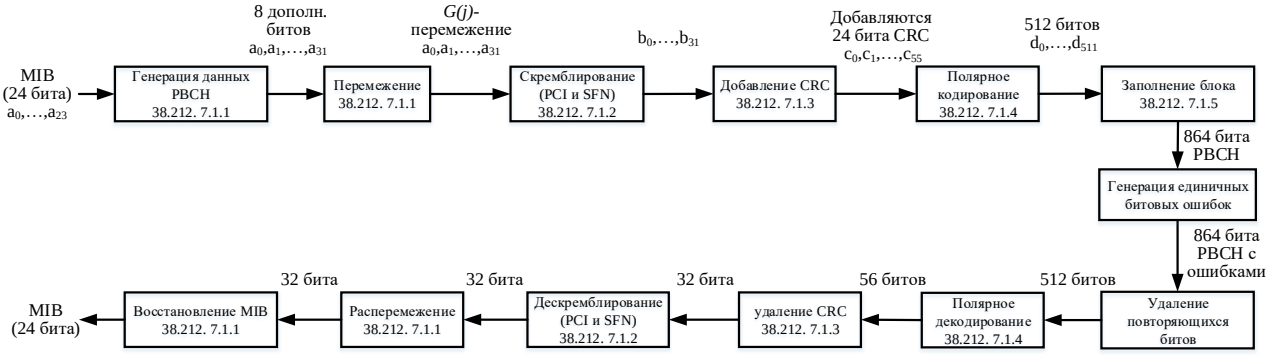

Рис. 3. Формирование и восстановление данных PBCH-канала [3]

К данным MIB-блока добавляются 8 дополнительных битов в соответствии с п. 7.1.1 [3]. Это 4 наименьших значимых бита LSB (Least Significant Bits) от номера кадра SFN, 1 бит индикации половины кадра и оставшиеся биты в зависимости от числа лучей в соте либо индексы лучей, либо наиболее значимые биты частотного сдвига SSB-блока. В результате получается транспортный блок из 32 элементов, обозначенных как $\bar{a}_0, \bar{a}_1, ..., \bar{a}_{31}$, который затем перемежается в соответствии с паттерном, показанным в табл. 1 [3, табл. 7.1.1-1], где j – это

номер бита в массиве $\bar{a}_0, \bar{a}_1, ..., \bar{a}_{31}$, а $G(j)$ – номер позиции, на который этот бит требуется перенести в соответствии с Алгоритмом 1.

Таблица 1. Функция перемежения *G(j)* для канала PBCH

j	*G*(*j*)	*j*	*G*(*j*)	*j*	*G*(*j*)	*j*	*G*(*j*)
0	16	**8**	24	**16**	9	**24**	21
1	23	**9**	7	**17**	11	**25**	22
2	18	**10**	0	**18**	12	**26**	25
3	17	**11**	5	**19**	13	**27**	26
4	8	**12**	3	**20**	14	**28**	27
5	30	**13**	2	**21**	15	**29**	28
6	10	**14**	1	**22**	19	**30**	29
7	6	**15**	4	**23**	20	**31**	31

Для того чтобы осуществить перемежение, необходимо выполнить действия, определяемые Алгоритмом 1.

Алгоритм 1: перемежение массива $\bar{a}_0, \bar{a}_1, ..., \bar{a}_{31}$.

Входные параметры: $\bar{a}_i$: данные канала PBCH, 32 элемента; G_j: вектор перемежения (Таблица 1); $j_{SFN} = 0$; $j_{HFR} = 10$; $j_{SSB} = 11$; $j_{Other} = 14$; $A = 32$.

Выход функции: массив $a = [a_0, a_1, ..., a_{A-1}]$;

For $i = 0$; $i \le A-1$; $i{+}{+}$ // цикл по всем битам массива

| **if** $i = 1 \vee i = 2 \vee i = 3 \vee i = 4 \vee i = 5 \vee i = 6 \vee i = 24 \vee i = 25 \vee i = 26 \vee i = 27$

| | $a_{G_{jSFN}} = \bar{a}_i$; // переставляем биты с *i*-ых позиций на позиции $G(j_{SFN})$

| | $j_{SFN} = j_{SFN} + 1$;

| **if** $i = 28$

| | $a_{G_{j_{HFR}}} = \bar{a}_i$; // переставляем бит с *i*-ой позиции на позицию $G(j_{HFR})$

| **if** $i = 29 \vee i = 30 \vee i = 31$

| | $a_{G_{jSSB}} = \bar{a}_i$; // переставляем биты с *i*-ых позиций на позиции $G(j_{SSB})$

| | $j_{SSB} = j_{SSB} + 1$;

| **else**

| | $a_{G_{jOther}} = \bar{a}_i$; // переставляем биты со всех остальных позиций на позиции $G(j_{Other})$

| | $j_{Other} = j_{Other} + 1$;

Возвращается вектор ***a***.

Выполнив перемежение данных блока PBCH, его требуется скремблировать с помощью SFN и PCI (Physical Cell ID) в соответствии с [3, п. 7.1.2] и [8, п. 5.2.1]. В результате получается массив *b* той же размерности (*A* = 32), элементы которого определены как (1):

$$b_i = \mathrm{mod}(a_i + s_i, 2), i = 0,1,...,A-1, \quad (1)$$

где s_i – это элементы скремблирующей *M*-последовательности, которые равны 0, если *i* – это любой из битов индекса SSB-блока, а для остальных индексов определяется как:

$$s_i = C(j + v \cdot M), i = 0,1,...,A-1, j{+}{+}. \quad (2)$$

В (2) $M = A-3$, если лучей (beams) меньше или равно 8, в противном случае $M = A-6$, а $C(i)$ – это скремблирующая последовательность, для инициализации которой использует-

ся $c_{init} = N_{ID}^{cell}$, где N_{ID}^{cell} – это физический идентификатор соты PCI. Кроме того, параметр v принимает целочисленные значения от 0 до 3 в зависимости от значений LSB номера текущего SFN-кадра ($v = 0$, если 2-й и 3-й LSB-биты SFN = 0; $v = 1$, если 2-й бит равен 1 и 3-й бит равен 0; $v = 2$, если 2-й бит равен 0 и 3-й бит равен 1; $v = 3$, если 2-й и 3-й LSB SFN равны 1). Ниже приводится Алгоритм 2, демонстрирующий генерацию псевдослучайной последовательности Голда длиной M_{NP} с порождающей *M*-последовательностью длиной 31 элемент [9].

Алгоритм 2: генерация скремблирующей последовательности $C = \left[C_0, .., C_{M_{NP}-1} \right]$.

Входные параметры: $c_{init} = N_{ID}^{cell}$: PCI соты; M_{NP}: длина последовательности; $N_C = 1600$;

Выход функции: массив $C = \left[C_0, .., C_{M_{NP}-1} \right]$;

$x1_0 = 1$;

For $i = 0$; $i \le 30$; $i{+}{+}$ // цикл по всем 31 элементам М-последовательности;

| $x2_i = \mathrm{mod}\left(c_{init}, 2\right)$;

| $c_{init} = \left\lfloor \frac{c_{init}}{2} \right\rfloor$;

For $i = 0$; $i \le M_{NP} + N_C - 1$; $i{+}{+}$

| $x1_i = \mathrm{mod}\left(x1_{i+3} + x1_i, 2\right)$;

| $x2_i = \mathrm{mod}\left(x2_{i+3} + x2_{i+2} + x2_{i+1} + x2_i, 2\right)$;

For $i = 0$; $i \le M_{NP} - 1$; $i{+}{+}$

| $C_i = \mathrm{mod}\left(x1_{i+N_C} + x2_{i+N_C}, 2\right)$;

Возвращается вектор ***C***.

После скремблирования последовательности a_i значениями s_i (2), определенными с помощью Алгоритма 2, необходимо вычислить и прикрепить контрольную сумму CRC к полученной последовательности b_i. Вычисление CRC осуществляется в соответствии с п. 7.1.3 и п. 5.1. [3]. Длина CRC обозначается как L и равна 24. Для вычисления CRC используется порождающий полином $g_{CRC24C}(D)$ (3):

$$g_{CRC24C}(D) = D^{24} + D^{23} + D^{21} + D^{20} + D^{17} + D^{15} + D^{13} + D^{12} + D^8 + D^4 + D^2 + D + 1. \quad (3)$$

По сути, CRC является остатком от деления последовательности b_i на порождающий полином. Алгоритм 3 демонстрирует процедуру его вычисления и прикрепления к оригинальному транспортному блоку.

Алгоритм 3: Вычисление CRC и добавление его к исходной последовательности.

Входные параметры: b: скремблированная последовательность (32 элемента);
g_{CRC}: порождающий полином длиной 25 элементов;

Выход функции: массив $c = \left[c_0, .., c_{55} \right]$;

For $i = 32$; $i \le (32 + 24 - 1)$; $i{+}{+}$ // цикл по добавочным 24 элементам к последовательности b

| $c_i = 0$; // Обнуляем элементы с 32-го по 55-ый;

For $k = 0$; $k \le 32 - 1$; $k{+}{+}$

| **if** $c_i = 1$

| | **For** $j = 0$; $j \le 24$; $j{+}{+}$

| | | $c_{j+k} = c_{j+k} \oplus g_{CRC_j}$;

For $k = 0$; $k \le 23$; $k{+}{+}$

| $CRC_k = c_{24+i}$;

Возвращается вектор c и CRC.

Следующий шаг формирования блока данных канала PBCH состоит в том, чтобы определить размер матрицы, которая будет использоваться для полярного кодирования и удлинения до аналогичного размера $N = 2^n$ массива битов данных. Эта процедура описывается в разделе 5.3.1 [3]. Вычисление параметра n показано в Алгоритме 4. Кроме того, перед тем как сформированный ранее PBCH-блок будет закодирован полярным кодом, его биты подвергаются дополнительному перемежению в соответствии с [3, п. 5.3.1.1], также описанному в Алгоритме 4.

Алгоритм 4: Определение размера кодируемого блока *N* и перемежение [3, п. 5.3.1.1].

Входные параметры: c: скремблированная последовательность с CRC (размер $K = 56$); $E = 864$: число битов в итоговом транспортном блоке PBCH; $\Pi_{IL}^{\max}(m)$ – паттерн для перемежения размером $K_{IL}^{\max} = 164$ ([3, табл. 5.3.1.1-1]); $n_{\max} = 9$; $n_{\min} = 5$; $R_{\min} = 1/8$.

Выход функции: массив $\hat{c} = \left[\hat{c}_0, .., \hat{c}_{55}\right]$;

if $E \le \left(9/8\right) \cdot 2^{\left(\lceil \log_2 E \rceil - 1\right)} \wedge K/E < 9/16$

| $n_1 = \lceil \log_2 E \rceil - 1$;

else

| $n_1 = \lceil \log_2 E \rceil$;

$n_2 = \left\lceil \log_2 \left(K / R_{\min} \right) \right\rceil$;

$n = \max\left\{\min\left\{n_1, n_2, n_{\max}\right\}, n_{\min}\right\}$; // для наших исходных данных $n = 9$;

$N = 2^n$;

$k = 0$;

For $m = 0$; $m \le K_{IL}^{\max} - 1$; $m{+}{+}$

| **if** $\Pi_{IL}^{\max}{}_m \ge K_{IL}^{\max} - K$

| | $\Pi_k = \Pi_{IL}^{\max}{}_m - \left(K_{IL}^{\max} - K\right)$;

| | $k{+}{+}$;

For $i = 0$; $i \le K - 1$; $i{+}{+}$

| $\hat{c}_i = c_{\Pi_i}$;

Возвращается вектор $\hat{c} = \left[\hat{c}_0, .., \hat{c}_{55}\right]$ и N – размер блока полярного кода.

После перемежения битов сформированный блок PBCH подается на полярный кодер [10]. В спецификации 3GPP [3, п. 5.3.1.2] для осуществления помехоустойчивого кодирования предлагается использовать заданную таблицей 5.3.1.2-1 полярную последовательность $Q_0^{N_{\max}-1} = \left\{Q_0^{N_{\max}}, Q_1^{N_{\max}}, ..., Q_{N_{\max}-1}^{N_{\max}}\right\}$, где $0 \le Q_i^{N_{\max}} \le N_{\max} - 1$ – битовые индексы до полярного кодирования для $i = 0,1,..., N_{\max} - 1$ и $N_{\max} = 1024$, $W\left(Q_0^N\right) < W\left(Q_1^N\right) < W\left(Q_2^N\right) < ... < W\left(Q_{N-1}^N\right)$ – это надежность бита с индексом $Q_i^{N_{\max}-1}$. Задача состоит в том, чтобы разместить 56 битов

на самые надежные позиции внутри транспортного блока размером N. Остальные элементы (512 – 56 = 456) приравниваются нулю и в результате формируется вектор $u = [u_0, u_1, ..., u_{N-1}]$.

Обозначим через $G_N = (G_2)^{\otimes n}$ n-кратное Кронекеровское произведение матрицы Арикана G_2 (4) с собой:

$$G_2 = \begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix}. \quad (4)$$

Последовательность $d = [d_0, d_1, ..., d_{N-1}]$ на выходе кодера формируется как $d = u \cdot G_N$ в полях Галуа второго порядка GF(2). Сформированную таким образом последовательность $d = [d_0, d_1, ..., d_{N-1}]$ остается лишь дополнить элементами, чтобы наполнить блок размером $E = 864$ в рамках процедуры rate matching, как описано в [3, п. 7.1.5 и п. 5.4], предварительно разбив ее на субблоки и выполнив поблочное перемежение в соответствии с паттерном $P(i)$ [3, табл. 5.4.1.1-1]. Алгоритм 5 демонстрирует пошаговую реализацию полярного кодера, перемежителя и функции rate matching.

Алгоритм 5: Полярный кодер, перемежение [3, п. 5.4.1.1], rate matching.

Входные параметры: $\hat{c}$: скремблированная последовательность с перемежением; $E = 864$: число битов в итоговом транспортном блоке PBCH; $P(i)$ – паттерн для перемежения размером 32 элемента ([3, табл. 5.4.1.1-1]); $Q_0^{N_{\max}-1}$: полярная последовательность, заданная табл. 5.3.1.2-1 [3].

Выход функции: массив закодированных битов $d = [d_0, .., d_{N-1}]$; $e = [e_0, .., e_{E-1}]$ – массив после (rate matching)-выравнивания.

$k = 0$;

While $k \neq K - 1$

| **if** $Q_0^{N_{\max}-1}{}_i \leq N - 1$

| | $u_{Q_0^{N_{\max}-1}{}_i} = \hat{c}_k$; // биты данных PBCH размещаются на самые надежные позиции;

| | $k{+}{+}$;

$G_2 = \begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix}$;

$G_N = \begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix}$;

For $i = 0$; $i \leq n - 2$; $i{+}{+}$

| $G_N = G_N \otimes G_2$;

For $i = 0$; $i \leq N - 1$; $i{+}{+}$

| $d_i = 0$;

| **For** $j = 0$; $j \leq N - 1$; $j{+}{+}$

| $d_i = d_i \oplus (u_j \cdot G_{N\,j,i})$; // полярное кодирование;

For $i = 0$; $i \leq N - 1$; $i{+}{+}$

| $I = \left\lfloor 32 \cdot i/N \right\rfloor$;

| $J_I = P_i \cdot \left(\frac{N}{32}\right) + \mathrm{mod}\left(I, \frac{N}{32}\right)$; // поблочное перемежение;

| $y_I = d_{J_I}$;

For $i=0;\ i\le E-1;\ i++$

| $e_i = y_{\mathrm{mod}(i,N)}$; // rate-matching;

Возвращается вектор $e=\left[e_0,..,e_{E-1}\right]$.

Таким образом, мы получили битовую последовательность данных канала PBCH, с помощью которой затем будут моделироваться поднесущие, сконфигурированные под передачу данных этого канала (modulation mappers). Описанные выше алгоритмы были реализованы на языке программирования C и верифицированы с помощью Mathcad и, по сути, являются функциями, которые могут работать в составе программного обеспечения базовой станции 5G, гарантируя совместимость с мобильными телефонами 5-го поколения любого производителя.

Однако в рамках данного исследования представляет интерес не просто разработка программного обеспечения базовой станции, а также оценка его эффективность с точки зрения надежности передачи данных по данному каналу. Для того чтобы это сделать, необходимо разработать набор алгоритмов, позволяющих восстановить MIB (24 бита) из полученных $E=864$ битов, то есть выполнить все те же функции, но в обратном порядке. Стоит отметить, что алгоритмы приема и восстановления данных не описаны стандартами 3GPP и реализуются вендорами самостоятельно. Авторами данной статьи были разработаны и реализованы все функции, восстанавливающие исходные данные, передаваемые по широковещательному каналу PBCH. В силу того, что эти функции во многом схожи с теми алгоритмами, что уже были представлены выше, в настоящей работе будет приведен лишь один алгоритм (Алгоритм 6), выполняющий дескремблирование данных с помощью перебора *v*. Проблема дескреблирования заключается в том, что скремблер на стороне передатчика использует 2-й и 3-й LSB-биты от номера системного кадра SFN, которые неизвестны UE на приемной стороне. Рассмотрим Алгоритм 6, реализующий метод перебора для дескреблирования битовой последовательности MIB-блока.

Алгоритм 6: Дескремблирование MIB на стороне 5G приемника.

Входные параметры: *c*: биты, полученные после полярного декодирования; *M* = 32-3; *G*(*j*) – функция перемежения (табл. 1); $j_{SFN}=0$; $j_{HFR}=10$; $j_{SSB}=11$; $j_{Other}=14$; $A=32$; *S*(c_{init}, M_{NP}): функция скремблера; M_{NP}: длина скремблирующей последовательности; c_{init}: инициализирующее значение;

Выход функции: массив $\hat{a}=\left[\hat{a}_0,\hat{a}_1,...,\hat{a}_{A-1}\right]$ – биты MIB и 8 биты дополнительных битов;

$c_{init}=PCI$;

For $v=0;\ v\le 3;\ v++$ // перебор, для того, чтобы подобрать верное значение *v*;

| $M_{NP}=32+v\cdot M$;

| **For** $i=0;\ i\le A-1;\ i++$

| | **if** $i=0\vee i=6\vee i=24$

| | | $s_i=0$;

| | **else**

| | | $s_i=S(c_{init},M_{NP})_{j+v\cdot M}$;

| | | $j=j+1$;

| il_{31}=0;

| **For** $i=0;\ i\le A-1;\ i++$

| | $il_i=s_i\oplus c_i$; // дескремблирование;

| **For** $i=0;\ i\le A-1;\ i++$

| | **if** $i=1\vee i=2\vee i=3\vee i=4\vee i=5\vee i=6\vee i=24\vee i=25\vee i=26\vee i=27$

| | | $\hat{a}_i = il_{G_{j_{SFN}}}$; // переставляем биты с $G(j_{SFN})$-ых позиций на позиции i.
| | | $j_{SFN} = j_{SFN} + 1$;
| | **if** $i = 28$
| | | $\hat{a}_i = il_{G_{j_{HFR}}}$; // переставляем бит с $G(j_{HFR})$-й позиции на позицию i.
| | **if** $i = 29 \vee i = 30 \vee i = 31$
| | | $\hat{a}_i = il_{G_{j_{SSB}}}$; // переставляем биты с $G(j_{SSB})$-ых позиций на позиции i.
| | | $j_{SSB} = j_{SSB} + 1$;
| | **else**
| | | $\hat{a}_i = il_{G_{j_{Other}}}$; // переставляем биты с $G(j_{Other})$-ых позиций на позиции i.
| | | $j_{Other} = j_{Other} + 1$;
| **if** $\hat{a}_{25} = 0 \wedge \hat{a}_{26} = 0 \vee v = 0$
| | | *break* ;
| **if** $\hat{a}_{25} = 0 \wedge \hat{a}_{26} = 1 \vee v = 1$
| | | *break* ;
| **if** $\hat{a}_{25} = 1 \wedge \hat{a}_{26} = 0 \vee v = 2$
| | | *break* ;
| **if** $\hat{a}_{25} = 1 \wedge \hat{a}_{26} = 1 \vee v = 3$
| | | *break* ;

Возвращается вектор $\hat{a}$.

3. Численные исследования

Успешность передачи транспортного блока канала PBCH в 5G определяется с помощью CRC. На приемной стороне вычисляется CRC принятого блока данных с помощью того же $g_{CRC24C}(D)$ порождающего полинома и сравнивается с полученным CRC, в результате принимается решение, искажен блок или нет. Вероятность ошибки при передаче блока BLER определяется как (5):

$$BLER = \frac{Количество\_неуспешных\_CRC\_проверок}{Общее\_число\_CRC\_проверок}. \quad (5)$$

Общее число CRC-проверок – это количество переданных по радиоканалу и полученных PBCH блоков. Количество же неуспешных проверок (переданный и вычисленный CRC не совпадают) зависит от числа битовых ошибок в блоке, которое, в свою очередь, определяется качеством канала SINR.

В рамках данной работы не ставилась цель изучать влияние конкретных каналов связи на надежность передачи данных. Для оценки надежности в сформированной на стороне передатчика битовой последовательности данных канала PBCH случайным образом изменялось различное количество битов и фиксировалось количество неуспешных CRC-проверок, затем вычислялось итоговое значение BLER.

Для оценки надежности передачи данных MIB-блока было выполнено имитационное моделирование, в ходе которого генерировался блок с данными, обрабатывался при помощи описанных ранее алгоритмов, затем в нем случайным образом искажались биты и алгоритмы обработки данных на приеме восстанавливали полученную битовую последовательность, вычисляли контрольную сумму CRC. На рис. 4 представлена полученная в ходе имитационного моделирования зависимость блоковой ошибки BLER от количества искаженных битов.

Всего в PBCH передается 864 бита, из которых 24 – это MIB, 8 – дополнительные биты, 24 – CRC, 456 – избыточные биты помехоустойчивого кодирования, а остальные – повторяющиеся биты, добавленные в рамках процедуры rate matching. Для получения каждой точки на графике моделирование выполнялось 10000 раз. PCI соты равен 0.

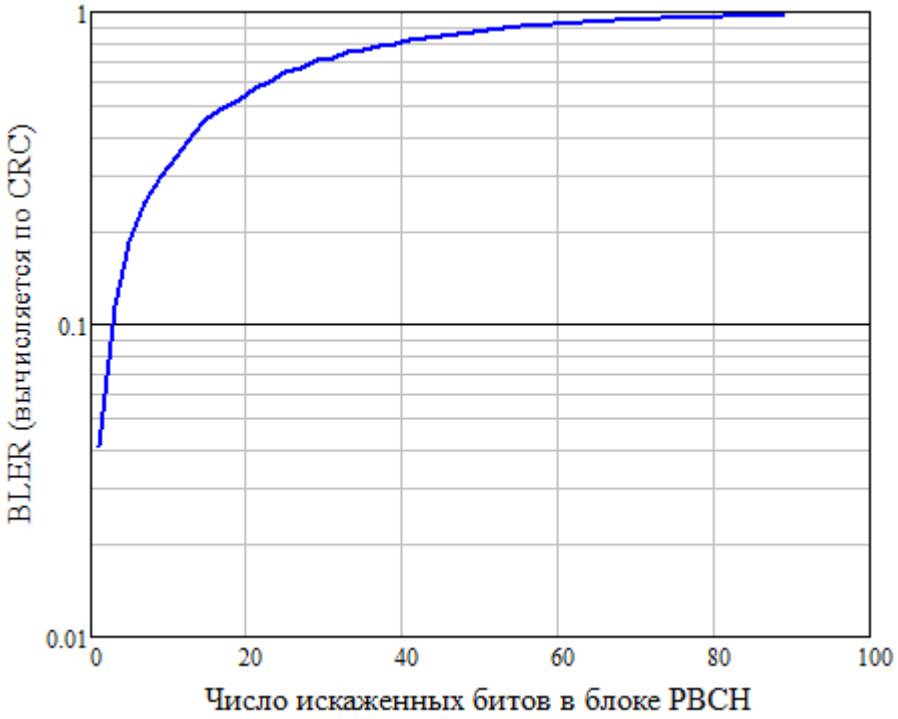

Рис. 4. Зависимость BLER от числа искаженных битов в блоке PBCH

Полученная зависимость наглядно демонстрирует, что даже одна битовая ошибка в 864-битном блоке приводит к невозможности декодирования транспортного блока в 4 % случаев. Таким образом, представленная на рис. 4 зависимость BLER от числа битовых ошибок позволяет оценить надежность передачи данных по PBCH каналу 5G NR.

4. Заключение

Мобильные сети 5-го поколения интенсивно развиваются и внедряются в коммерческую эксплуатацию по всему миру, в связи с чем разработка технологий для их усовершенствования и изучение эффективности уже стандартизованных решений является актуальной задачей. В рамках данной статьи были описаны разработанные авторами алгоритмы формирования и приема данных широковещательного канала PBCH сетей 5G, а также выполнена оценка надежности передачи данных по этому каналу. Продемонстрирована зависимость вероятности ошибочного приема блока PBCH от количества битовых ошибок в блоке.

Работа выполнена при финансовой поддержке гранта Президента Российской Федерации для государственной поддержки молодых ученых – кандидатов наук МК-1047.2020.9.

Литература

1. Паспорт национальной программы «Цифровая экономика Российской Федерации», утв. Президиумом Совета при Президенте Российской Федерации по стратегическому развитию и национальным проектам, протокол № 16 от 24 декабря 2018 г.
2. The 3rd Generation Partnership Project (3GPP), https://www.3gpp.org/about-3gpp.
3. 3GPP TS 38.212: 5G; NR; Multiplexing and channel coding, 3GPP TS 38.212 version 6.2.0 Release 16 (спецификации).
4. 3GPP TS 38.331: 5G; NR; Radio Resource Control (RRC); Protocol specification, version 15.9.0 Release 15, (спецификации).
5. *Bahadkar S.* Receiver Design for Physical Broadcast Channel in 5G NR // A Thesis to Indian Institute of Technology Hyderabad In Partial Fulfillment of The Degree of Master of Technology, 2019.
6. *Pillet C., Bioglio V., Condo C.* On List Decoding of 5G-NR Polar Codes // arXiv:1907.00784 [cs.IT], 2020.
7. *Lin Z., Li J., Zheng Y., Irukulapati N., Wang H., Sahlin H.* SS/PBCH Block Design in 5G New Radio (NR) // Proc. IEEE Globecom Workshop, 2018.
8. 3GPP TS 38.211: 5G; NR; Physical channels and modulation, 3GPP TS 38.212 version 6.2.0 Release 16 (спецификации).
9. *Варакин Л. Е.* Системы связи с шумоподобными сигналами. М.: Радио и связь, 1985. 384 с.
10. *Gazi O.* Polar codes. A non-trivial approach to channel coding. Springer, 2019. 174 p.

Статья поступила в редакцию 06.01.2021.

Дроздова Вера Геннадьевна
к.т.н., доцент кафедры телекоммуникационных сетей и вычислительных средств СибГУТИ (630102, Новосибирск, ул. Кирова, 86), e-mail: drozdova_vera@mail.ru.

Завьялова Дарья Васильевна
старший преподаватель кафедры телекоммуникационных сетей и вычислительных средств СибГУТИ, e-mail: da2215@mail.ru.

Algorithms for generating and receiving data from the 5G networks PBCH channel

V. Drozdova, D. Zavyalova

New radio is a 5G mobile networks that are being rapidly deployed in commercial operation all over the world. In the framework of this paper, the issues related to the software development of 5G broadcast PBCH channel for generating and receiving data are considered. The PBCH data reliability is also estimated demonstrating the dependency between PBCH block error rate and the number of incorrectly received bits. Algorithms of open 3GPP 38.212 specification are used to form the channel, and non-standardized algorithms are developed for receiving data.

Keywords: 5G, NR, New Radio, PBCH, BLER, broadcast channel.