



**Министерство науки и высшего образования
Российской Федерации Федеральное государственное
бюджетное образовательное учреждение высшего
образования «Московский государственный технический
университет имени Н.Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)**

**Факультет «Информатика и системы управления»
Кафедра ИУ5 «Системы обработки информации и управления»**

Курс «Сети и телекоммуникации»

Отчет по домашнему заданию №2

«Разработка и реализация алгоритмов кодирования, декодирования и
определения корректирующей способности кода в линейных протоколах»

Вариант №22

Выполнил:
студент группы ИУ5-53Б
Ювенский Лев

Проверил:
Галкин В. А.

Москва, 2024 г.

Постановка и метод решения задачи для варианта задания

Имеется дискретный канал связи, на вход которого подается закодированная в соответствии с вариантом задания кодовая последовательность. В канале возможны ошибки любой кратности. Вектор ошибки может принимать значения от единицы в младшем разряде до единицы во всех разрядах кодового вектора. Для каждого значения вектора ошибки на выходе канала после декодирования определяется факт наличия ошибки и предпринимается попытка ее исправления.

Таблица 1. Индивидуальный вариант домашнего задания.

№ варианта	Информационный вектор	Код	Способность кода
22	11111010011	Ц [15,11]	C_k

Требуется, используя кодирование циклическим кодом Ц [15,11], определить обнаруживающую способность этого кода C_k :

Обозначения:

- Ц [n, k] – циклический код с образующим полиномом.
- n – число разрядов в закодированной записи. n = 15
- k – число разрядов в информационной части. k = 11
- z – число дополнительных битов для обнаружения ошибки.
 $z = n - k = 4$.
- r – число разрядов в образующем векторе. $r = z + 1 = 5$.
- C_k – корректирующая способность кода.
- g(x) - образующий полином степени z = 4. $x^4 + x + 1$
- V_i - информационный вектор. По заданию 11111010011
- V_c – циклический вектор – результат кодирования V_i .
- I – входящий вектор. Циклический вектор, возможно содержащий ошибки

**Алгоритмы кодирования, декодирования, вычисления
корректирующей способности кода для ошибок всех возможных
кратностей.**

1. Алгоритм кодирования циклическим кодом

- 1.1. Умножить информационный полином $V_i(x)$ на x^z . Т. е. сдвинуть информационный вектор на z разрядов в сторону старших, заполнив освободившиеся нулями.
- 1.2. Разделить полученный полином на порождающий полином $g(x)$, после получить остаток $p(x)$.
- 1.3. Сложить $p(x)$ с $V_i(x)$, чтобы получить $V_c(x)$, то есть объединить остаток $p(x)$ в векторной форме и исходный полином $x^z \times V_i(x)$ для получения кодового слова.

2. Алгоритм декодирования циклическим кодом

- 2.1. Разделить принятый полином $I(x)$ на порождающий полином $g(x)$ и проверить остаток от деления $S(x)$ – он является синдромом ошибки.
- 2.2. Если $S(x) = 0$, то ошибки нет или она не была обнаружена
- 2.3. Если $S(x) \neq 0$, то ошибка есть. По виду вектора синдрома определить место ошибки и исправить ее.
- 2.4. Поделить информационный полином $V_i(x)$ на x^z для получения исходного полинома, то есть сдвинуть информационный вектор на z разрядов в сторону младших, чтобы получить первоначальный вектор.

3. Алгоритм вычисления корректирующей способности кода C_k для ошибок всех возможных кратностей

Корректирующая способность кода C_k определяется как отношение числа исправленных ошибок N_k к общему числу ошибок данной кратности, которое определяется как число сочетаний из n (длина кодовой комбинации) по i (кратность ошибки – число единиц в векторе ошибок) - C_n^i

Корректирующая способность кода вычисляется как

$$C_k = \frac{N_k}{C_n^i}$$

Для подсчета корректирующей способности нужно перебрать все возможные вектора ошибок. Их число

$$s_e = \sum_{i=1}^n C_n^i = 2^n - 1$$
$$e \in [0...1, 1...1]$$

Ошибки нужно сгруппировать по кратности.

3.1. Закодировать данный информационный вектор

3.2. Для каждой группы ошибок по кратности

3.2.1. Ввести счетчик N_{ki} , который по умолчанию будет равен 0.

3.2.2. Для каждой ошибки в группе

3.2.2.1. Наложить на исходный закодированный вектор текущий вектор ошибки e из текущей группы

3.2.2.2. Воспользовавшись алгоритмом коррекции, декодировать получившуюся последовательность.

3.2.2.3. Сравнить декодированную последовательность с заданным информационным вектором.

3.2.2.3.1. Если информационный и декодированный вектора равны, то коррекция ошибки успешна. Увеличить счетчик N_{ki} на 1

3.2.2.3.2. Иначе коррекция ошибки неудачна. Необходимо продолжить выполнение алгоритма, не изменяя счетчик.

3.3. Получим n значений $N_{ki} \in [N_{k1}, N_{kn}]$ для каждой группы. Составить результирующую таблицу корректирующей способности, где номеру группы i будет соответствовать кратность ошибки i и значение $C_{ki} = \frac{N_{ki}}{C_n^i}$, выраженное в процентах.

4. Алгоритм коррекции ошибки

4.1. Составить таблицу соответствия синдрома ошибки S_e ошибке e . Для этого для каждой ошибки $e \in [0...1, 1...1]$ рассчитать ее синдром S_e , сгруппировать синдромы по кратности ошибки.

- 4.2. Найти в таблице синдром-ошибка синдром S_v , полученный при декодировании входной последовательности V , и определить таким образом вектор ошибки.
- 4.3. Инвертировать те разряды во входной последовательности, которые отмечены в векторе ошибки как 1.
- 4.4. Декодировать измененную входную последовательность еще раз.

5. Получение циклического кода по варианту задания

- 5.1. $v_i = 111.1101.0011$. $g(x) = 10011$.
- 5.2. Сдвиг на 4 бита влево: $111.1101.0011.0000$
- 5.3. Деление на образующий полином (см. рисунок 1)

```

111110100110000 | 10011
10011             | -----
-----          | 110011
11000
10011
-----
10111
10011
-----
10000
10011
-----
11110
10011
-----
11010.
10011
-----
10010
10011
-----
0010

```

Рисунок 1 — деление на образующий полином

Остаток: 0010

- 5.4. Конкатенация информационного вектора с остатком:

$111.1101.0011.0010$

6. Программная реализация алгоритмов

Для программной реализации модели канала связи, алгоритмов кодирования, декодирования и вычисления корректирующей способности кода для ошибок всех возможных кратностей выберем язык Go. В нём реализованы необходимые побитовые операции, а так же он позволяет отобразить результаты в удобочитаемом формате с использованием HTML шаблонов.

Исходный код разработанного решения можно скачать по ссылке:
<https://github.com/Yu-Leo/bmstu-networks-corrective-ability>

7. Таблица ошибок всех возможных кратностей для $n = 15$

Для $n = 15$ число возможных ошибок составляет $2^n - 1 = 32767$.

Все ошибки можно разделить на 15 классов: от ошибок кратности 1 до ошибок кратности 15.

Число ошибок в каждом разряде определяется по формуле: C_{15}^i , где i – кратность ошибки.

Таблица, отображающая количество ошибок в каждом классе, приведена на рисунке 2. Суммарное количество ошибок равно 32767. Фрагмент таблицы ошибок, сгруппированных по классам, приведён на рисунке 3.

Реализуем алгоритм операции О. Посчитаем синдром для каждой ошибки. Фрагмент таблицы приведён на рисунке 4.

Вектор ошибки	Синдром
1	1
10	10
100	100
1000	1000
10000	10000
100000	110
1000000	1100
10000000	1011
100000000	101
1000000000	1010
10000000000	111
100000000000	1110
1000000000000	1111
10000000000000	1101
100000000000000	1001
1000000000000001	1000
1000000000000001	1100
1000000000000010	1011
1000000000000011	1010
10000000000001	1110
10000000000010	1111

Рисунок 4 — фрагмент таблицы синдромов всех ошибок

9. Таблица синдромов

Заметим, что у некоторых ошибок совпадают синдромы. Это происходит из-за того, что вектор симптома имеет слишком мало бит, чтобы покрыть все 32767 ошибок, и с образующим полиномом 10011_2 (19) может предоставить только 18 уникальных ненулевых значений.

Сгруппируем ошибки по синдромам. Получим таблицу, приведённую на рисунке 5. Будем использовать её для попыток исправления ошибок в полученной последовательности по синдромам. Конечно, однозначно определить ошибку невозможно. Именно из-за этого корректирующая способность не стопроцентная.

Синдром	Вектор ошибки
1	1
10	10
100	100
1000	1000
10000	10000
1001	1000000000000000
101	100000000
1010	1000000000
1011	100000000
110	1000000
1100	1000000
1101	1000000000000000
111	10000000000
1110	100000000000
1111	10000000000000

Рисунок 5 — ошибки по синдромам для ошибок кратности 1

10. Результирующая таблица

Построим результирующую таблицу. Она приведена на рисунке 6.

Кратность ошибки	Общее число ошибок, шт	Скорректированные ошибки, шт	Корректирующая способность, %
1	15	14	93.33
2	105	15	14.29
3	455	15	3.30
4	1365	15	1.10
5	3003	15	0.50
6	5005	15	0.30
7	6435	15	0.23
8	6435	15	0.23
9	5005	15	0.30
10	3003	15	0.50
11	1365	15	1.10
12	455	15	3.30
13	105	15	14.29
14	15	15	100.00
15	1	0	0.00

Рисунок 6 — результирующая таблица

11. Выводы

В коде Ц[15,11] под вектор симптома выделено 4 дополнительных бита. Их не хватает, чтобы покрыть все 32767 ошибок. С образующим полиномом 10011_2 (19) возможно предоставить только 18 уникальных ненулевых значений. Коллизии неизбежны в тех классах ошибок, в которых

их число превышает 18. Именно из-за этого корректирующая способность не стопроцентная.

Невозможно завести одну таблицу симптомов для ошибок разных кратностей, чтобы иметь возможность распознавать несколько поврежденных бит.

Анализ результатов для ошибок кратности 1. Корректирующая способность: 93,33% — одна ошибка из 15. Число 111.1101.0010.0010 (наложенная на исходный код ошибка $e = 10000_2$) дает остаток 11 при делении на образующий полином (см. рисунок 7). Этого остатка нет в таблице симптомов, т.к. ни одна ошибка с кодами от 1, 10, 100, ..., 100.0000.0000.0000 не дает такого остатка (см. рисунок 8).

```
transferredVector: 111110100100010
syndrome: 11
correctedVector: 111110100100010
```

Рисунок 7 — единственная неисправленная ошибка кратности 1

Синдром	Вектор ошибки
1	1
10	10
100	100
1000	1000
10000	10000
1001	100000000000000
101	100000000
1010	1000000000
1011	100000000
110	100000
1100	1000000
1101	100000000000000
111	10000000000
1110	100000000000
1111	1000000000000

Рисунок 8 — таблица ошибок по синдромам по ошибкам кратности 1

Анализ результатов для ошибок кратности 14. Корректирующая способность: 100%. Достигнута за счет полного совпадения таблицы симптомов с полученными симптомами.

Анализ результатов для ошибок кратности 15. Набор состоит из единственного вектора ошибки. Вектор ошибки 111.1111.1111.1111 имеет симптом 0, т.е. данный вектор ошибки нельзя обнаружить в при использовании циклических кодов.

Итак, циклический код применим каналам связи, у которых статистическая вероятность ошибки кратности 2 и более минимальна.

Список используемой литературы

1. Галкин В.А. Методическое пособие по выполнению домашнего задания по дисциплине «Сети и телекоммуникации», 2024 г.
2. Галкин В.А., Григорьев Ю.А. Телекоммуникации и сети: Учеб. Пособие для вузов.-М.: Изд-во МГТУ им. Н.Э. Баумана, 2003 г.
3. С. М. Рацеев, А. М. Иванцов, П. А. Булдаковский. Об алгоритмах декодирования циклических кодов, 2021, с.87–94
4. Telecommunication technologies - телекоммуникационные технологии (v2.1) [Электронный ресурс] URL:
https://www.opennet.ru/docs/RUS/inet_book/ (дата обращения: 09.10.2024)