

НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО
ФАКУЛЬТЕТ ПРОГРАММНОЙ ИНЖЕНЕРИИ И КОМПЬЮТЕРНОЙ ТЕХНИКИ
НАПРАВЛЕНИЕ ПРОГРАММНОЙ ИНЖЕНЕРИИ

ОТЧЕТ ПО ДОМАШНЕЙ РАБОТЕ
курса «Компьютерные сети»

по теме: «Методы кодирования в компьютерных сетях»

Выполнил студент:
Тюрин Иван Николаевич
группа: Р33102

Преподаватель:
Авксентьева Е. Ю.,
Алиев Т. И.

Санкт-Петербург, 2024 г.

Содержание

Домашняя работа . Методы кодирования в компьютерных сетях	2
1. Часть 1: методы физического и логического кодирования	2
1. Этап 1	2
2. Этап 2	2
3. Этап 3	5
4. Этап 4	5
5. Этап 5	7
2. Часть 2	7
3. Вывод	7

Домашняя работа

Методы кодирования в компьютерных сетях

1. Часть 1: методы физического и логического кодирования

1.1. Этап 1

В работе необходимо использовать свои инициалы для получения варианта согласно таблице кодировки, таким образом мой вариант:

«ТИН» → 0xD2C8CD

и в двоичном виде

$0xD2C8CD = 0b\ 1101'0010'1100'1000'1100'1101.$

Длина полученного сообщения равна 3 байта (24 бит).

1.2. Этап 2

На этом этапе выполнения работы нужно произвести физическое кодирование с использованием манчестерского кодирования и еще двух на выбор. Я выбираю RZ и AMI. Так же по заданию нужно принять, что пропускная способность канала связи равна $C = 100$ Мбит/с. Требуемые характеристики сигнала будем считать в соответствии с приложением в конце описания задания.

Кодирование способом RZ

Результат физического кодирования способом RZ можно видеть на рисунке [1.1](#) временной диаграммы сигнала.

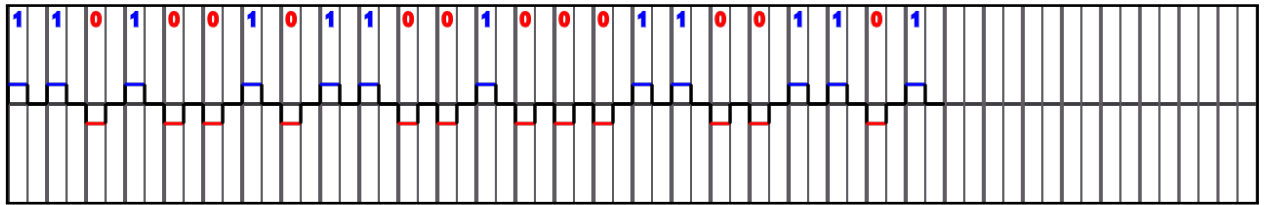


Рис. 1.1: Результат физического кодирования способом RZ (Return-to-Zero)

Характеристики сигнала рассчитанные физического сигнала изображенного на рис. 1.1 для передаваемого сообщения.

- Верхняя граница частоты в передаваемом сообщении считается по самой высокочастотной составляющей сигнала, т.е. при последовательной передаче серии 1 или серии 0 период сигнала будет равен битовому интервалу (биты 1-2, 11-12, 14-16, 17-18, 19-20, 21-22 в сообщении):

$$f_{\text{в}} = C = 100 \text{ МГц.}$$

- Нижняя граница в этом случае получается, если в качестве периода взять два битовых интервала:

$$f_{\text{н}} = \frac{C}{2} = 50 \text{ МГц.}$$

- Средняя частота получается средним арифметическим для всех диапазонов со всеми доступными частотами:

$$f_{\text{ср}} = \frac{33 \cdot f_0 + 3 \cdot \frac{f_0}{2}}{36} = 95 \text{ МГц.}$$

- Полоса пропускания для качественной передачи в соответствии со спектром $S = f_{\text{в}} - f_{\text{н}} = 50 \text{ МГц}$:

$$F = 100 \text{ МГц.}$$

Кодирование способом АМІ

Результат физического кодирования способом АМІ можно видеть на рисунке 1.2 временной диаграммы сигнала.

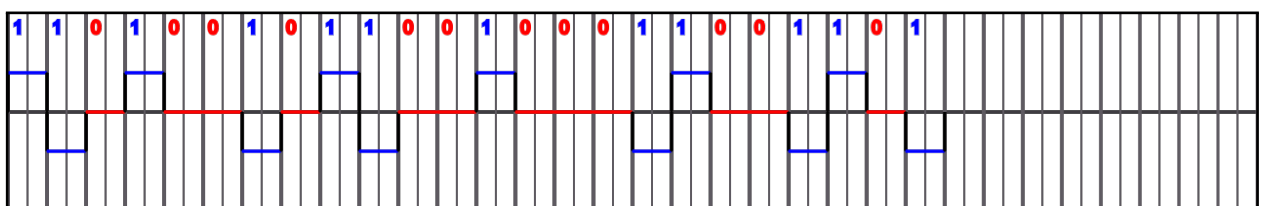


Рис. 1.2: Результат физического кодирования способом АМІ (Bipolar Alternate Mark Inversion)

Характеристики сигнала:

- Верхняя граница частоты в передаваемом сообщении:

$$f_{\text{в}} = \frac{C}{2} = 50 \text{ МГц},$$

- Нижняя граница теоретически нулевая, но в данном сообщении достигается, если рассмотреть передачу трёх 0 подряд (14-16 биты), период в этом случае равен 6 битовым интервалам:

$$f_{\text{н}} = \frac{C}{6} = 16,67 \text{ МГц}.$$

- Средняя частота:

$$f_{\text{ср}} = \frac{15 \cdot f_0 + 6 \cdot \frac{f_0}{2} + 3 \cdot \frac{f_0}{3}}{24} = 39.58 \text{ МГц}.$$

- Полоса пропускания для качественной передачи в соответствии со спектром $S = f_{\text{в}} - f_{\text{н}} = 33,33 \text{ МГц}$:

$$F = 100 \text{ МГц}.$$

Кодирование способом Manchester-2

Результат физического кодирования способом М2 (Manchester-2) можно видеть на рисунке 1.3 временной диаграммы сигнала.

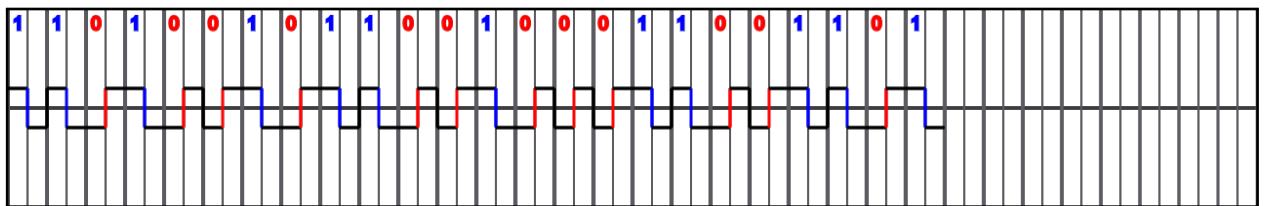


Рис. 1.3: Результат физического кодирования способом М2 (Manchester-2)

Характеристики сигнала:

- Верхняя граница частоты в передаваемом сообщении:

$$f_{\text{в}} = C = 100 \text{ МГц},$$

- Нижняя граница достигается если рассмотреть передачу подряд различных значений: 0 и 1 или 1 и 0 (1-2, 2-3... биты), период в этом случае равен 2 битовым интервалам:

$$f_{\text{н}} = \frac{C}{2} = 50 \text{ МГц}.$$

- Средняя частота:

$$f_{\text{ср}} = \frac{20 \cdot f_0 + 28 \cdot \frac{f_0}{2}}{48} = 70.833 \text{ МГц.}$$

- Полоса пропускания для качественной передачи в соответствии со спектром $S = f_{\text{в}} - f_{\text{н}} = 50 \text{ МГц}$:

$$F = 100 \text{ МГц.}$$

Как можно видеть, все методы кодирования имеют свои преимущества и недостатки: RZ и M2 имеют возможность самосинхронизации и лишены постоянной составляющей сигнала (минимальная частота теоретически не нулевая), но требуют больше спектр частот чем AMI.

1.3. Этап 3

Сообщение закодированное при помощи 4B/5B:

0b 1101'1101'0011'0101'0010'1101'0110'11.

16-ый код сообщения полученного при помощи 4B/5B: 0x374D4B5B. Длина сообщения полученного при этом 3,75 байта (30 бит), чему соответствует избыточность 0,25.

1.4. Этап 4

Исходное сообщение было скремблировано с использованием предложенной формулы

$$B_i = A_i \oplus B_{i-3} \oplus B_{i-5}$$

(см. листинг 1.1) и с помощью собственной формулы

$$B_i = A_i \oplus B_{i-14} \oplus B_{i-16}$$

(см. листинг 1.2). Как оказалось, для предложенной формулы происходит не эффективное кодирование, т.к. с моей формулой в полученном сообщении меньше длина последовательности одинаковых значений (2 против 8 в предложенной).

```

1 msg = 0xD2C8CD # encoded message
2 a = [int(i) for i in bin(msg)[2:]]
3 # [1, 1, 0, 1, 0, 0, 1, 0, 1, 1, 0, 0,
4 #  1, 0, 0, 0, 1, 1, 0, 0, 1, 1, 0, 1]
5
6 b = [None for _ in a] # scrambled msg
7 c = [None for _ in a] # descrambled msg

```

```

8
9 # scramble
10 for i in range(len(a)):
11     if i < 3:
12         b[i] = a[i]
13     elif i < 5:
14         b[i] = a[i] ^ b[i-3]
15     else:
16         b[i] = a[i] ^ b[i-3] ^ b[i-5]
17
18 # [1, 1, 0, 0, 1, 1, 0, 1, 0, 0, 0, 0,
19 #  0, 0, 0, 0, 1, 1, 0, 1, 0, 0, 0, 1]
20
21 print(hex(int("".join(map(str,b)), 2))) # 0xcd00d1
22
23 # descramble
24 for i in range(len(a)):
25     if i < 3:
26         c[i] = b[i]
27     elif i < 5:
28         c[i] = b[i] ^ b[i-3]
29     else:
30         c[i] = b[i] ^ b[i-3] ^ b[i-5]
31
32 print(all([a[i] == c[i] for i in range(len(a))])) # True

```

Листинг 1.1: Скремблирование исходного сообщения с использованием предложенной формулы

```

1 msg = 0xD2C8CD # encoded message
2 a = [int(i) for i in bin(msg)[2:]]
3 # [1, 1, 0, 1, 0, 0, 1, 0, 1, 1, 0, 0,
4 #  1, 0, 0, 0, 1, 1, 0, 0, 1, 1, 0, 1]
5
6 b = [None for _ in a] # scrambled msg
7 c = [None for _ in a] # descrambled msg
8
9 # scramble
10 for i in range(len(a)):
11     if i < 14:
12         b[i] = a[i]
13     elif i < 16:
14         b[i] = a[i] ^ b[i-14]
15     else:
16         b[i] = a[i] ^ b[i-14] ^ b[i-16]
17
18 # [1, 1, 0, 1, 0, 0, 1, 0, 1, 1, 0, 0,
19 #  1, 0, 1, 1, 0, 1, 0, 1, 0, 1, 0, 0]
20
21 print(hex(int("".join(map(str,b)), 2))) # 0xd2cb54
22
23 # descramble
24 for i in range(len(a)):
25     if i < 14:
26         c[i] = b[i]
27     elif i < 16:
28         c[i] = b[i] ^ b[i-14]
29     else:
30         c[i] = b[i] ^ b[i-14] ^ b[i-16]
31

```

```
32 print(all([a[i] == c[i] for i in range(len(a))])) # True
```

Листинг 1.2: Скремблирование исходного сообщения с использованием собственной формулы

1.5. Этап 5

Как можно было видеть, все методы кодирования имеют свои особенности: преимущества и недостатки. Методы физического кодирования накладывают определенные ограничения на канал связи, в то время как кодирование с избыточностью этого не требует, но увеличивает длину передаваемого сообщения. Скремблирование в свою очередь требует подбора подходящей функции, т.к. для разных сообщений они по-разному эффективны. Хорошо выглядит идея использовать эти методы кодирования одновременно для получения преимуществ каждого из них и нивелирования их недостатков, например, скремблирования и физического кодирования АМІ, чтобы увеличить среднюю частоту.

2. Часть 2

Для каждого этапа 6-10 были проведены необходимые симуляции в среде NetworkFurrier 2.1, по результатам которых была составлена таблица 2 представленная на рисунке 1.4. Максимальные номера гармоник высчитывались для подобранных минимальных номеров. При этом, для этапа 9-10 не удалось получить удобоваримые результаты, т.к. во всех случаях при максимальном номере гармоники (255) не удалось добиться передачи сообщения без помех.

3. Вывод

Не удалось определить какой тип кодирования лучше подходит для передачи сообщения по реальному каналу, можно сказать лишь то, что при кодировании со скремблированием (NRZ) ошибки визуально возникают реже. Теоретически манчестерский код дает лучше качество передачи при большем уровне рассинхронизации и граничного напряжения, уровень шума может быть тоже достаточно высок. Скремблирование позволяет повысить качество передачи в сравнении с обычным NRZ, но не так сильно как использование другого кодирования.

Таблица 2

Шестнадцатеричный код сообщения:			Метод кодирования				
			NRZ	RZ	M2	4B/5B	Scramb
Полоса пропускания идеального канала связи	Номера гармоник	min	2	6	28	8	0
		max	16	44	44	32	20
	Частоты, МГц	min	0.4	1.3	5.8	1.0	0.0
		max	3.3	9.2	9.2	4.0	3.1
Минимальная полоса пропускания идеального канала связи			2.9	7.9	3.4	3.0	3.1
Уровень шума		max	1.92	0.66	1.35	0.79	1.04
Уровень рассинхронизации		max	1.0	1.0	1.0	1.0	1.0
Уровень граничного напряж.		max	0.95	0.99	1.0	0.82	0.92
Процент ошибок при max уровнях и минимальной полосе пропускания КС			24 %	30 %	2.5 %	20 %	17 %
Уровень шума		ср.	1,152				
Уровень рассинхронизации		ср.	1				
Уровень граничного напряж.		ср.	0,2304				
Полоса пропускания реального канала связи	Гармоники	min	-	-	-	-	-
		max	-	-	-	-	-
	Частоты, МГц	min	-	-	-	-	-
		max	-	-	-	-	-
Требуемая полоса пропускания реального канала связи			-	-	-	-	-

Рис. 1.4: Результаты симуляции в среде NetworkFurier 2.1 для исходного сообщения