

# Радиосеть: доставка сообщений. Часть 3. Исправленная.



tanyshkabelova, 4 июня 2013г.

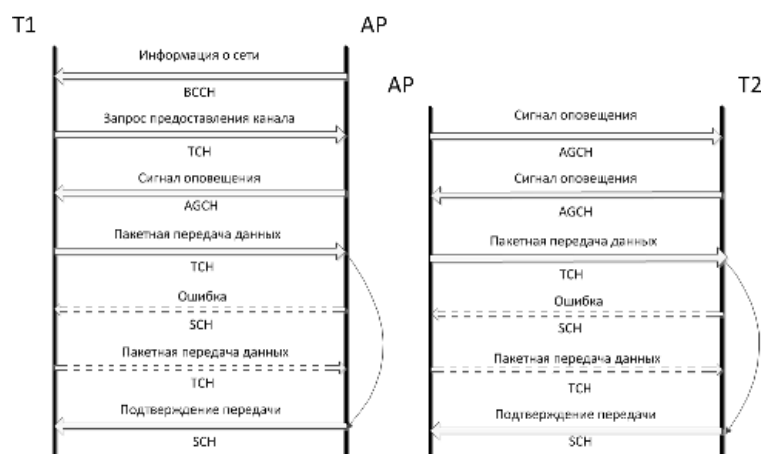
*Самостоятельная работа*

## **Радиосеть: доставка сообщений. Часть 3.**

*Выполнили: ст.гр. 9110 Белова Т.В., Жарко Н.С.*

Разработка и экспериментальное исследование модели.

Разработка и экспериментальное исследование модели сценария взаимодействия терминалов сети с точкой доступа. Сценарий передачи нашего сообщения выглядит следующим образом:

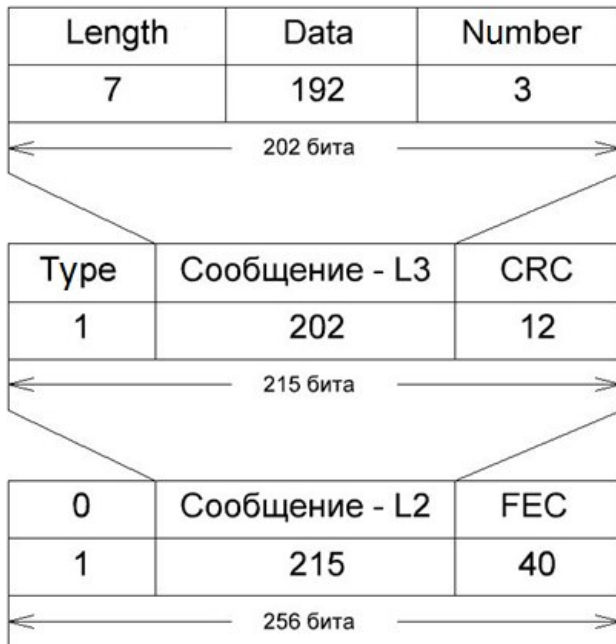


Нам необходимо передать сообщение, содержащееся в текстовом файле. В программной реализации исполнен только процесс передачи сообщения от терминала к базовой станции. Выглядит это примерно так:

- 1) реализован процесс считывания сообщения из файла,
- 2) разбиения его на пакеты определенной длины
- 3) и его передачи

Пакеты, подготовленные к передаче, имеют иерархическую структуру, состоящую из физического, канального уровней и уровня управления.

Используется код (255,215), следовательно, необходимо немного изменить структуру передаваемого пакета. Выглядеть она будет так:



Рассмотрим процесс передачи сообщения:

Программа начинает работу с того, что находит заданный текстовый файл:

```
% Выбор файла для передачи
```

```
set.FileName = 'text.txt';
```

Считывает из него первые 12 символов:

```
% Длина однократно считываемого текстового сообщения
```

```
set.Str_size = 12;
```

Которые переводятся в биты:

```
bin_msg = strtobin(tx_msg)'; %перевод текста в двоичный код (16 бит на символ)
```

– это и есть поле Data L3 уровня. К нему присоединяется служебная информация , содержащая длину и номер пакета:

```
msg_L3 = [bi_length_tx DATA Number];
```

Добавляется CRC – код:

```
% Параметры CRC
```

```
set.poly=[1 1 0 0 0 0 0 0 0 1 1 1 1];
```

И тип передаваемого сообщения:

```
trxIQ = transmit('service',msg_L3);
```

- эту часть пакета уже можно отправлять в передатчик для подготовки к отправлению сообщения.

```
%% Наложение шума на блок символов
```

```
noise_sgn = awgn(trxIQ,7);
```

Далее вся полученная последовательность кодируется избыточным блочным кодом БЧХ (255,215). Он позволяет исправить 5 ошибок, что и необходимо реализовать по заданию.

```
%% Помехоустойчивое кодирование
```

```
code_word = encode(trx,st.n,st.k,'bch/binary');
```

Так же сформированный пакет дополняется нулевым битом для обеспечения четности,

необходимой нам при типа модуляции QPSK.

```
code_word = [0 code_word]';
```

Затем происходит QPSK модуляция и передача битовой последовательности в канал с заданным отношением сигнал-шум.

```
%% Модуляция

h = modem.pskmod(st.mPos);

% h - модем PSK с позиционностью mPos

% Требуется настроить остальные его параметры:

% Параметры модуляции

h.InputType = 'bit';

h.SymbolOrder = 'gray';

h.PhaseOffset = pi/4;

% собственно модуляция

IQ_signal = modulate(h, code_word);
```

На приемной стороне выполняется следующая последовательность действий:

Происходит демодуляция сообщения:

```
%% Параметры демодуляции

h = modem.pskdemod(st.mPos);

h.SymbolOrder = 'gray';

h.PhaseOffset = pi/4;

h.OutputType = 'bit';
```

```
% собственно демодуляция

s = demodulate(h, rxIQ);
```

Декодирование:

```
%% Помехоустойчивое декодирование

received_mes_L1 = decode(s(2:st.n+1), st.n, st.k, 'bch/binary');

%% декодирование L2-сообщения
```

Проверка подлинности CRC:

```
% Подготовка к проверке CRC

[q r]=deconv(received_mes_L1, st.poly);

%проверка остатка

r=mod(abs(r), 2);

if r == zeros(1, length(received_mes_L1))

    detect=0;% ошибок нет
```

```

else

    errs = errs + 1;

    detect=1;% обнаружены ошибки

end;

```

**Принятое сообщение:**

```
% Собственно принятое сообщение
```

```
received_mes_L2 = received_mes_L1(1:st.length_msg_Lev2);
```

**Преобразование сообщения в текстовый формат и извлечение типа сообщения:**

```
% % Преобразование в текстовый формат
```

```
% % тип сообщения
```

```
ТипMSG = my_bi2de(received_mes_L2(1)', 'left_msb');
```

**Извлечение принимаемого сообщения и номера пакета в двоичном виде:**

```
% % Извлечение принимаемого сообщения (в двоичном виде)
```

```
length_data = my_bi2de(received_mes_L2(2:8)', 'left_msb' );
```

```
msgL3 = received_mes_L2(9:8+16*length_data)';
```

```
NUMBER = received_mes_L2(201:203)';
```

**Вывод текстового файла на экран:**

```
txt_msg = bintostr(msg)';
```

```
disp(['Текстовое сообщение: ' txt_msg]);
```

Для примера выполнялась передача сообщения:

*«Привет, друзья! С началом предпоследней сессии) желаем всем удачи!»*

Количество символов текста: 12

Количество бит: 192

Количество пакетов: 6

Поле DATA: 192

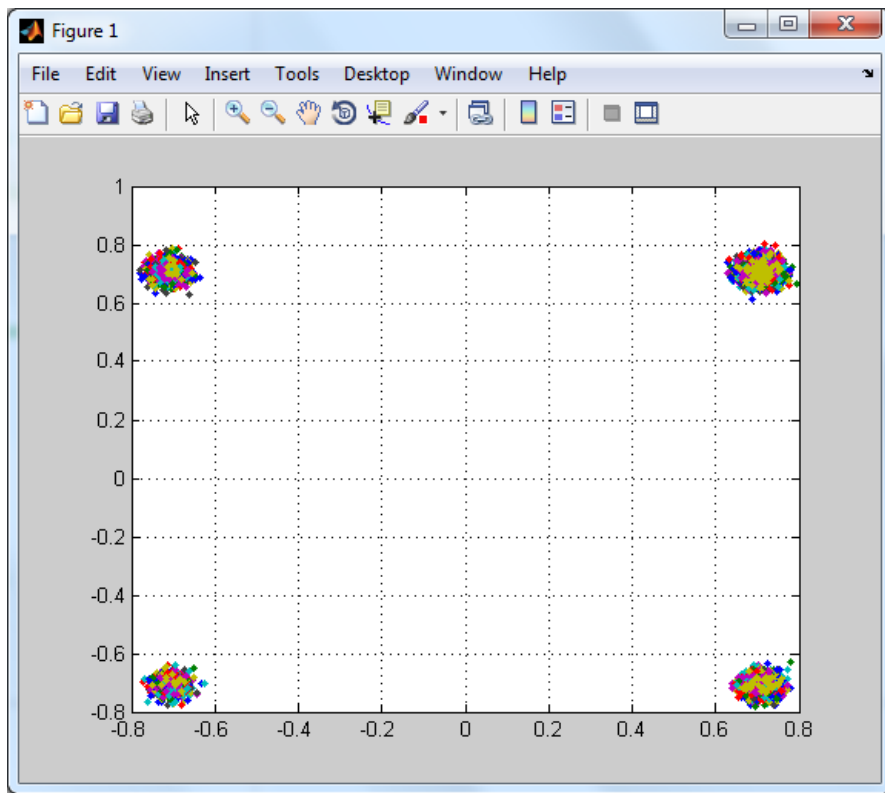
Поле Сообщение\_L3: 202

Поле Сообщение\_L2: 215

Поле "Кодовое слово": 256

**Рассмотрим полученные сообщения и сигнальные созвездия при различных ОСШ:**

ОСШ = 30 дБ

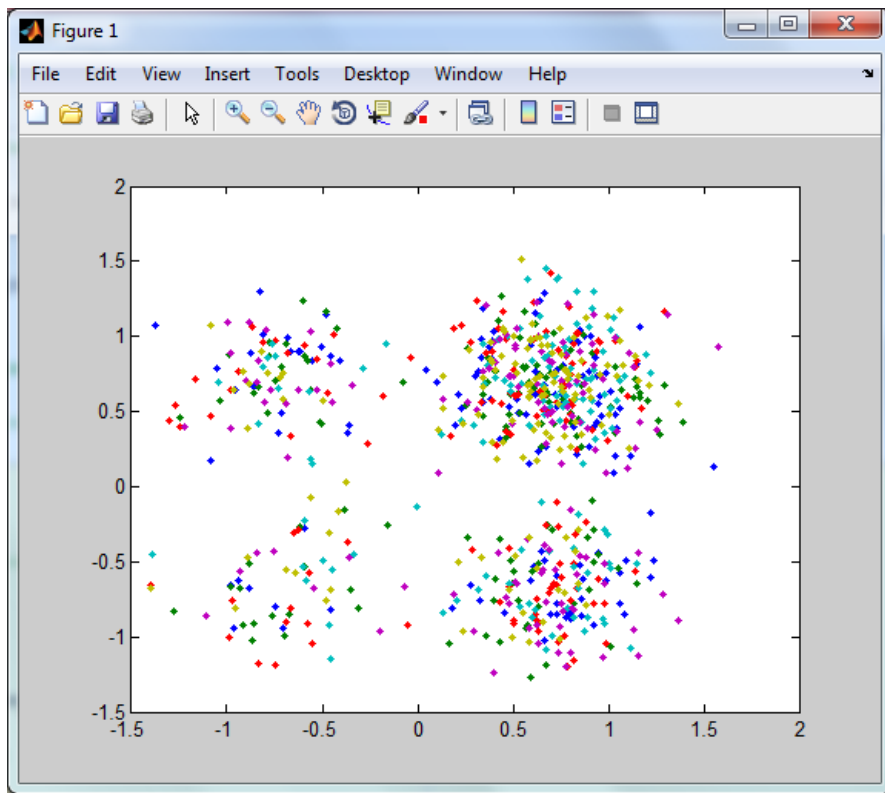


Текст, содержащийся в пакетах:

- 1) Привет, друз
- 2) ья! С начало
- 3) м предпослед
- 4) ней сессии)
- 5) желаем всем
- 6) удачи!

*Передаваемое сообщение принято без ошибки.*

ОСШ = 8 дБ

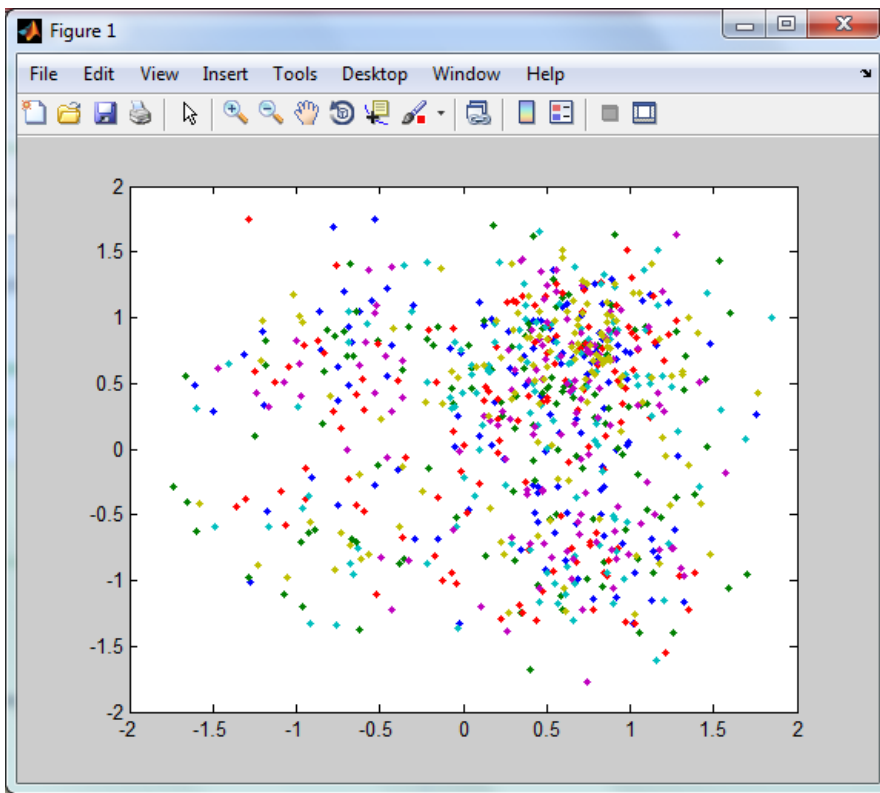


Текст, содержащийся в пакетах:

- 1) Привет, друз
- 2) ья! С начало
- 3) м предпослед
- 4) ней сессии)
- 5) желаем всем
- 6) удачи!

*Передаваемое сообщение принято без ошибки.*

ОСШ = 5 дБ



Текст, содержащийся в пакетах:

- 1) аве
- 2) ы!
- 3) ?напАие
- 4) нй
- 5) желаем всем
- 6) чда!

*Передаваемое сообщение принято с ошибкой.*

В ходе выполнения работы были решены задачи, поставленные в задании –спроектирована система передачи сообщения, использующая модуляцию QPSK, с помехоустойчивым кодом, исправляющим 5 ошибок. В проделанной самостоятельной работе был разработан сценарий передачи данных от терминала 1 к терминалу 2, и часть его реализована программно.

Список использованных источников:

1. <http://omoled.ru/publications/view/386>
2. <http://omoled.ru/publications/view/389>
3. "О построении программной модели сети"
4. Скляр Б. Цифровая связь. Теоретические основы и практическое применение. М.: Вильямс, 2003г.

Выражаем благодарность Воробьеву Алексею за помощь в написании программной части.

