

Тема ТЗ-№6 - "Передача данных с уведомлением о доставке". (Часть 3)



Сергей, 26 февраля 2015г.

Тема ТЗ-№6 - "Передача данных с уведомлением о доставке".

Выполнил: студент группы № 1110 Мелехин С.А.

1.4. Построение итоговой иерархической модели узлов сети (в соответствии с концепцией OSI). В модели должны быть обязательно учтены задачи доставки как служебных, так и информационных сообщений.

2. Экспериментальная часть. Разработка и экспериментальное исследование программной модели канала передачи данных.

2.1. Разработка и описание блок-схемы алгоритма модели.

2.2. Разработка программных модулей передачи и приема сообщений канального уровня

1.4. Построение итоговой иерархической модели узлов сети (в соответствии с концепцией OSI).

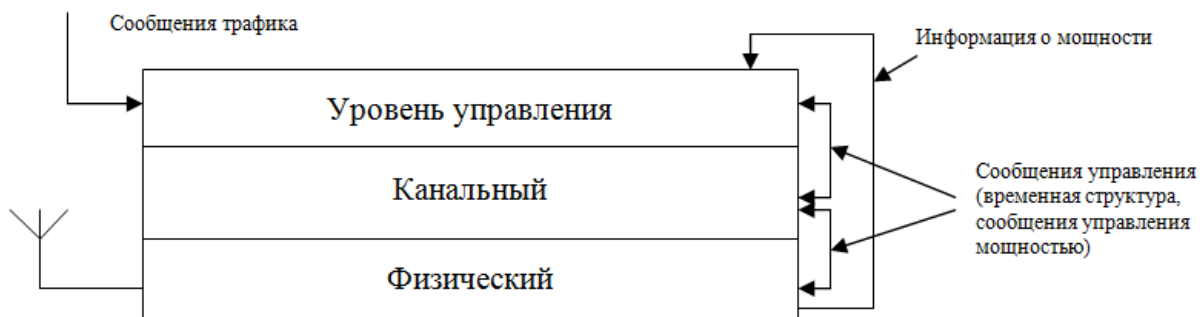


Рис.3.1. Итоговая иерархическая модель узлов сети (в соответствии с концепцией OSI)

Модель OSI описывает схему взаимодействия сетевых объектов, определяет перечень задач и правила передачи данных. В эталонной модели OSI отдельные уровни взаимодействуют друг с другом посредством четко определенных интерфейсов. На каждом уровне реализуется определенное количество базовых функций при помощи набора функций, определенных в эталонной модели OSI нижележащего уровня. Каждый уровень предоставляет набор услуг находящемуся над ним уровню, причем на вышележащем уровне неизвестны детали реализации услуг, предоставляемых нижележащим уровнем.

Рассмотрим данную модель применимо к разрабатываемой радиосистеме и решаемым ею задач (Рис.3.1).

В силу относительной несложности разрабатываемой радиосистемы, три верхних уровня классической модели OSI – прикладной, представительский и сеансовый – объединим в один уровень, являющийся

интеллектуальным центром системы. На данном уровне на основе данных информационной системы и данных, полученных с нижнего уровня, будут приниматься все решения в системе. Поэтому данный уровень назовем уровнем «управления». Задача этого уровня – это анализ полученных с нижних уровней, данных, на его основании принимается решение о дальнейших действиях системы. Решения о функционировании сети принимаются исходя из служебных сообщений, а также на основе сценариев взаимодействия терминалов, прописанных в программном обеспечении терминалов.

Следующий уровень в иерархии - *транспортный*. Он обеспечивает надежный механизм обмена данными, контролирует отсутствие ошибок в принимаемых данных, расположение пакетов в соответствующем порядке, их полноту. В разрабатываемой системе канальный уровень выполняет все перечисленные функции транспортного уровня. Поэтому в наличие отдельного транспортного уровня нет необходимости.

Далее в структуре OSI следует *сетевой* уровень. Но в разрабатываемой модели он отсутствует, поскольку нет соединения с другими сетями.

Моя модель узлов сети будет состоять из трёх уровней: физического, канального и уровня управления.

Физический уровень:

- организация физических каналов;
- радиоизмерения;
- безошибочный приём/передача сообщений;
- синхронизация терминалов.

Канальный уровень:

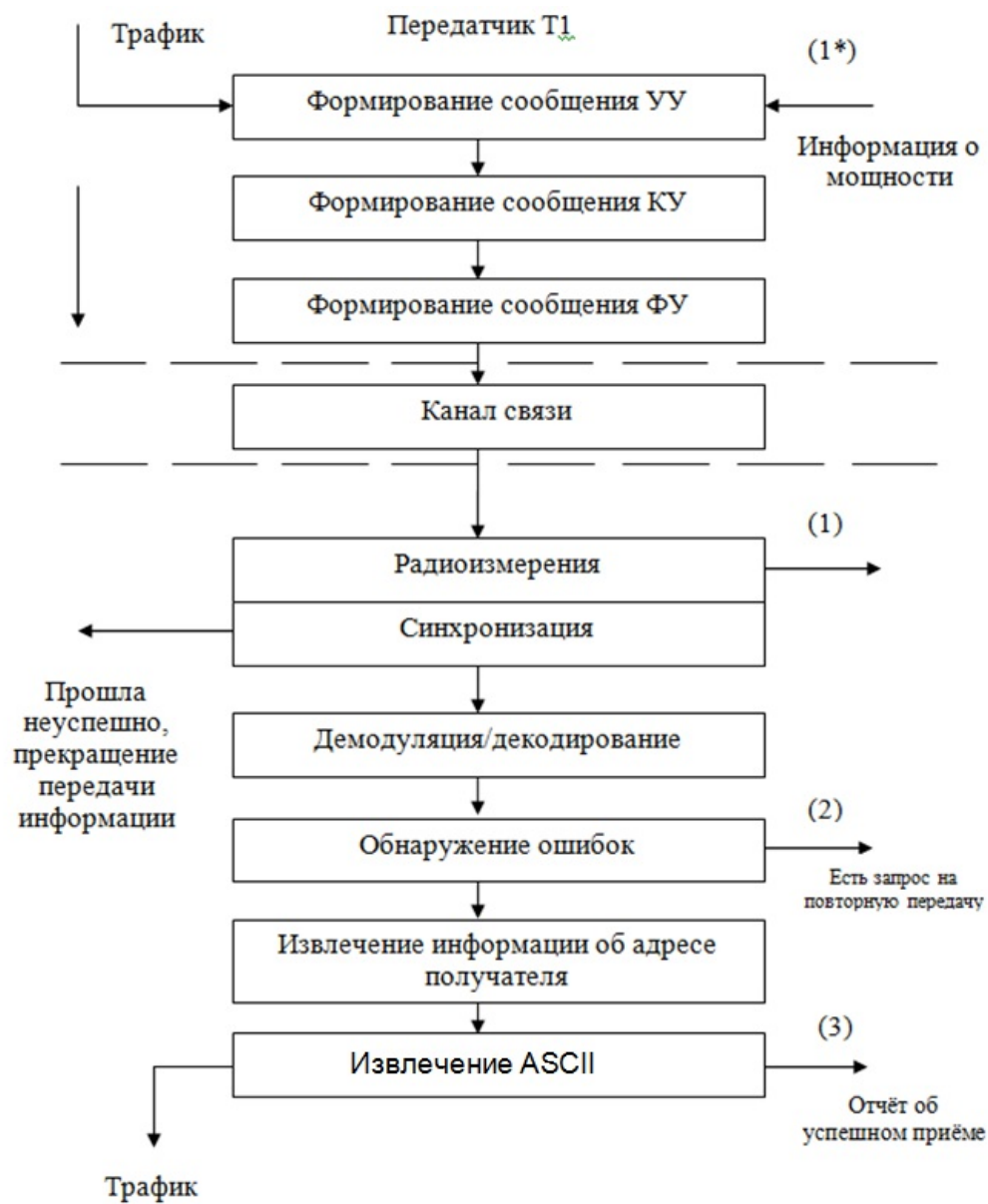
- адресная доставка сообщений;
- контроль за наличием ошибок (ARQ, CRC-8);
- доступ к физическим каналам (FAMA на (TDMA));
- формирование/расформирование сообщений.

Уровень управления:

- управление мощностью передачи;
- формирование временной структуры работы терминалов.

2. Экспериментальная часть. Разработка и экспериментальное исследование программной модели канала передачи данных.

2.1. Разработка и описание блок-схемы алгоритма модели.



Приёмник T2

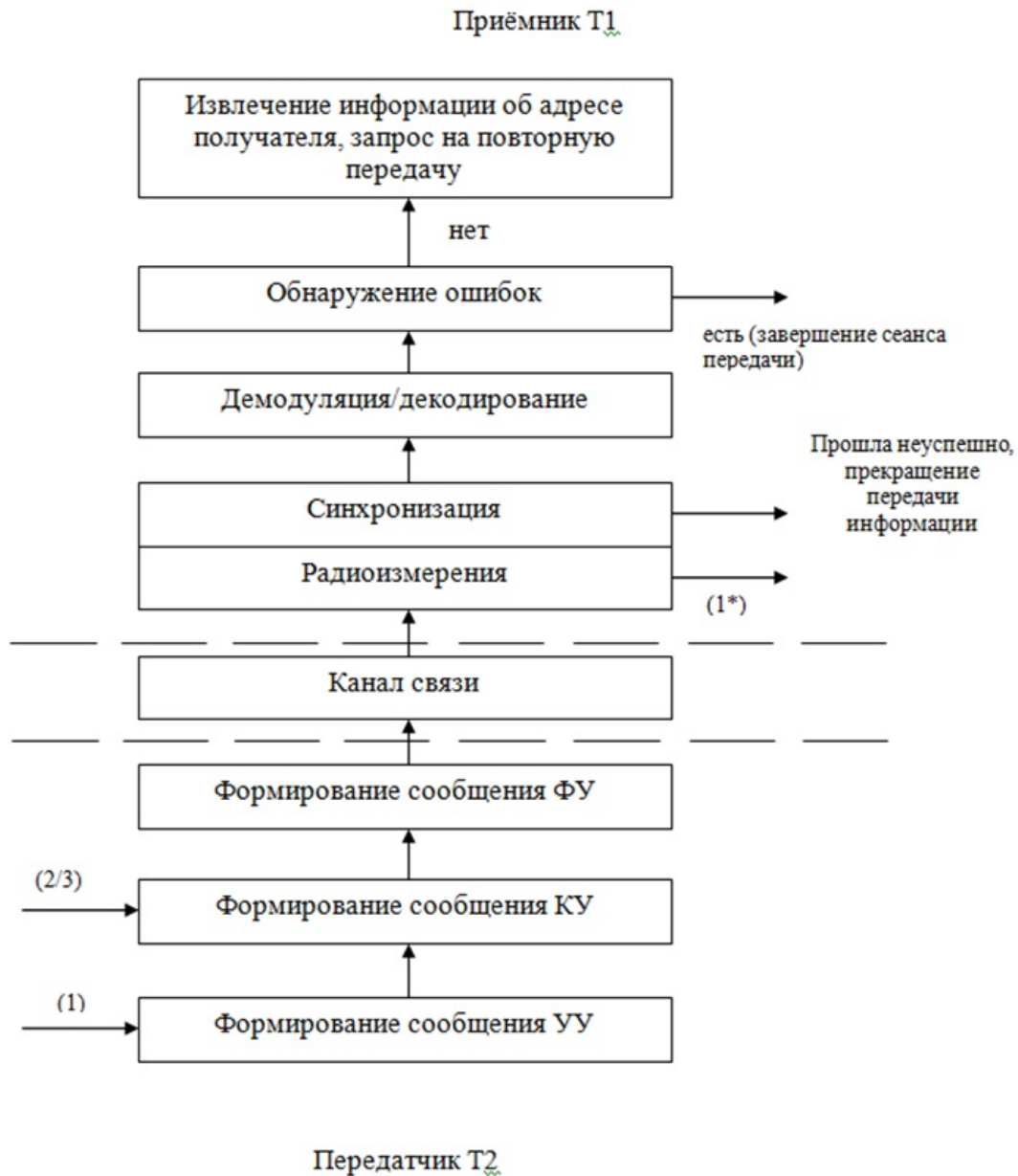


Рис.3.2. Блок-схема алгоритма передачи данных с уведомлением.

Программа начинает работу с того, что передается трафик сообщения.

T1- передатчик , T2 – приемник

На 1-ой части блок-схемы (Рис.3.2) до пунктирной линии описана работа T1 – как передатчика, а после пунктирной линии T2 – как приемника.

На 2-ой части блок-схемы (Рис.3.2) до пунктирной линии описана работа T1 – как приемника, а после пунктирной линии T2 – как передатчика.

- 1) T1 формирует сообщение УУ (здесь содержится поле для управления мощностью);
- 2) T1 формирует сообщение КУ (определяются адреса мастера и слейва, определяется количество передаваемых сообщений);
- 3) T1 формирует сообщение ФУ (здесь содержится частотная и тактовая синхронизация)

- 4) Затем T1 отправляет сообщение по КС.
- 5) T2, получив сообщение от T1, проводит радиоизмерение и передает T2 корректировку мощности (на блок-схеме ссылка (1)).

После радиоизмерений T2 осуществляет синхронизацию.

Если синхронизация прошла не успешно, происходит прекращение передачи информации.

Если синхронизация прошла успешно, T2 производит демодуляцию/декодирование сообщения.

- 6) Затем при помощи ARQ – запроса происходит обнаружение ошибок T2 (слейвом).

Если есть ошибки, то осуществляется запрос на повторную передачу (на блок-схеме ссылка (2)).

Если нет ошибок, то терминалом T2 извлекается информация об адресе получателя,

производится извлечение мультимедиа и формируется сообщение об успешном приеме (на блок-схеме ссылка (3)).

2.2. Разработка программных модулей передачи и приема сообщений канального уровня.

При разработки программных модулей необходимо передать файл, состоящий из символов ASCII. Рассмотрим реализацию процесса передачи сообщений от терминала T1 к терминалу T2. Для этого необходимо:

1. Считать сообщение из файла.
2. Разбить его на пакеты определенной длины.
3. Передать на T2.

Пакеты, подготовленные к передаче, имеют иерархическую структуру, состоящую из физического, канального уровней и уровня управления. Структура передаваемого пакета выглядит следующим образом (Рис.3.3)

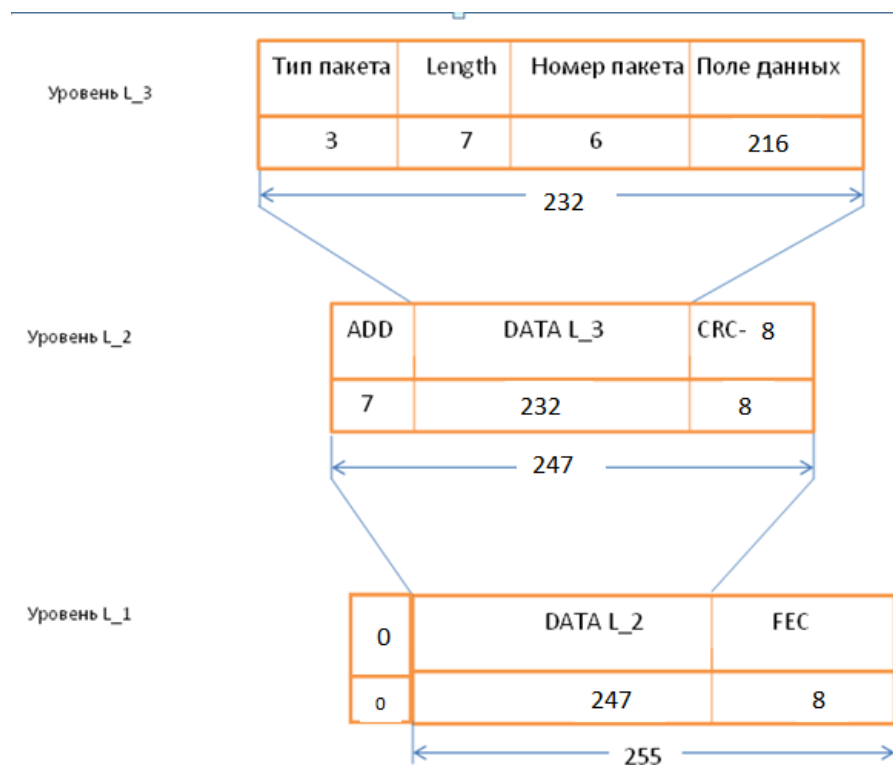


Рис 3.3 Структура передаваемого пакета.

Выбор файла для передачи

```
FileName = 'test.txt';  
  
% Кодировка символа текста: 2 байта на символ  
  
sym_size = 2;  
  
% Длина однократно считываемого текстового сообщения  
  
Str_size = 14;  
  
% Смещение указателя чтения символа из файла на заданную величину  
  
byte_off = 0;
```

Параметры модели

```
позиционность модуляции - BPSK  
  
mPos = 2;  
  
R_message = '';
```

Параметры помехоустойчивого кодека кодека

Для примера используется линейный блочный код Хэмминга:

Длина кодового слова $n = 2^m - 1 = 255$ при $m = 8$

Размер кодируемого блока $k = n - m = 247$

```
m = 8;  
  
n = 2^m - 1; % Codeword length = 255  
  
k = n - m; % Message length = 247
```

инициализация пустых массивов

```
matched = []; % результат согласованной фильтрации
```

% расширяющая кодовая последовательность (для примера - случайный код)

```
ss_code = [1 0 1 1 0 0 1 1 1 0 0 0 1 0 1 1 0 1 0 0 1 1 0 1 1 0 1 1 0 0 0 1];
```

Параметры CRC

```
poly=[1 0 0 0 0 1 1]; %CRC-8  
  
[x y]=size(poly); %размерность полинома
```

% Открытие на чтение и получение идентификатора текстового файла

```
[fid, message] = fopen(FileName, 'r');
```

%Проверка на доступность указанного файла

```
if (fid < 0)
```

```
    disp('Файл с передаваемым сообщением не найден. Завершение работы.');
```

```
    return
```

```
end;
```

Смещение указателя начала чтения из файла на требуемую величину (sym_size*byte_off байтов)

```
fseek(fid, sym_size*byte_off, 'cof');
```

Организуется цикл чтения данных размером Str_size байт

из файла до тех пор, пока не будет достигнут конец файла

```
while ~feof(fid)
```

```
    msg = fscanf(fid, '%c', Str_size);
```

Канальный уровень

Подготовка к передаче сообщения

текстовое сообщение переводится в двоичный вид

```
str_bin = strtobin(msg);
```

получение адреса терминала в двоичном 7-и разрядном виде (Address=5)

```
addr_bin = my_de2bi(5, 'left_msb', 7);
```

сборка части L2-сообщения

```
msg_Lev2 = [addr_bin str_bin];
```

получение CRC кода

```
m=[msg_Lev2 zeros(1,y-1)]; %вставка нулей в поле CRC
```

```
[q r] = deconv(m,poly); % поиск частного и остатка от деления
```

информационной части на порождающий полином CRC кода

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

```
CRC = r(length(msg_Lev2)+1:end);%расчет CRC
```

сборка сообщения

```
msg_Lev1 = [msg_Lev2 CRC];
```

```
trx = msg_Lev1;
```

Физический уровень

Помехоустойчивое кодирование

```
code_word = encode(trx,n,k,'hamming/binary');
```

codeword имеет размер 255 битов

добавление дополнительного бита с целью приведения итогового блока к

величине, кратной степени 2 (256)

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

Здесь выполняется расширение спектра

подготовка массива

```
ss_signal = []; % сигнал после расширения спектра
```

```
for i=1:length(code_word)
```

```
    if code_word(i) == 1
```

```
        ss_signal = [ss_signal ss_code];
```

```
    else ss_signal = [ss_signal ~ss_code]; % "инверсия кода"
```

```
    end;
```

```
end;
```

Модуляция

```
h = modem.pskmod(mPos);
```

h - модем PSK с позиционностью mPos

Требуется настроить остальные его параметры:

Параметры модуляции

```
h.inputtype = 'bit';
```

```
h.SymbolOrder = 'gray';
```

```
h.PhaseOffset = pi/3;
```


Собственно модуляция

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

Наложение шума на сигнал

```
noise_sgn = awgn(IQ_signal,-6);
```

здесь сигнал noise_sgn должен быть записан в файл

Отображение сигнально-кодového созвездия

```
figure(10);  
plot(noise_sgn, '.');
```

Демодуляция

```
% Параметры демодуляции  
h = modem.pskdemod(mPos);  
h.SymbolOrder = 'gray';  
h.PhaseOffset = pi/3;  
h.OutputType = 'bit';
```

Собственно демодуляция

```
s = demodulate(h, noise_sgn);  
rec_mes = [];
```

Согласованная фильтрация (сжатие спекта)

принятая последовательность нарезается на блоки размером расширяющей последовательности (`length(ss_code)`) и перемножается с расширяющим кодом. Учитывая правила перемножения матриц, в результате получается значение скалярного произведения двух векторов:

текущего блока принятого сигнала и расширяющего кода.

таким образом реализуется согласованная фильтрация.

Единственная особенность - принятая последовательность и расширяющий код должны быть переведены в формат (1,-1) из формата (0,1)

```
bin_ss_code = (ss_code .* 2)-1;
```

```
for i=1:length(code_word)
```

```
    x1 = (i-1)*length(ss_code)+1;% нижняя граница извлекаемого блока битов размером  
расширяющей последовательности
```

```
    x2 = i*length(ss_code);% верхняя граница извлекаемого блока битов размером расширяющей  
последовательности
```

```
    bin_s = (s(x1:x2).* 2) - 1; % перевод в формат (1,-1) из формата (0,1)
```

```
    matched(i) = bin_s*bin_ss_code'; % собственно скалярное произведение,
```

```
    в массиве matched имеется оценка принятого сообщения (code_word)
```

```
    if matched(i)>8
```

```
        rec_mes = [rec_mes 1];
```

```
    else
```

```
        rec_mes = [rec_mes 0];
```

```
    end;
```

```
end;
```

Помехоустойчивое декодирование

```
received_mes_L1 = decode(rec_mes(1:n),n,k,'hamming/binary');
```

Канальный уровень

декодирование L2-сообщения

Подготовка к проверке CRC

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

проверка остатка

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

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

```
    detect=0;% ошибок нет
```

```
else
```

```
    err=err+1;
```

```
    detect=1;% обнаружены ошибки
```

```
end;
```

Собственно принятое сообщение

```
received_mes_L2 = received_mes_L1(1:length(msg_Lev2));
```

Преобразование в текстовый формат

адрес получателя

```
addr = my_bi2de(received_mes_L2(2:7),'left_msb');
```

Извлечение принимаемого сообщения (в двоичном виде)

```
txt_message = received_mes_L2(8:length(received_mes_L2));
```

перевод в текстовый формат

```
txt = bintostr(txt_message);
```

накопление частей принятого сообщения в переменной R_message

```
R_message = [R_message txt];
```

% отображение в виде html-ссылки участков текста, принятых с ошибкой

```
if detect==0
```

```
    fprintf('%s', txt);
```

```
else
```

```
    fprintf('%s', [' <a href = ' "' ' "' '> ' txt ' </a> ']);
```

```
end;
```

```
pos = ftell(fid)
```

```
end; % признак завершения цикла
```

Закрытие файла

```
fclose (fid);
```

Для примера передачи сообщения возьмём стихотворение А.С.Пушкина "Буря":

БУРЯ

Ты видел деву на скале

В одежде белой над волнами

Когда, бушуя в бурной мгле,

Играло море с берегами,

Когда луч молний озарял

Ее всечасно блеском алым

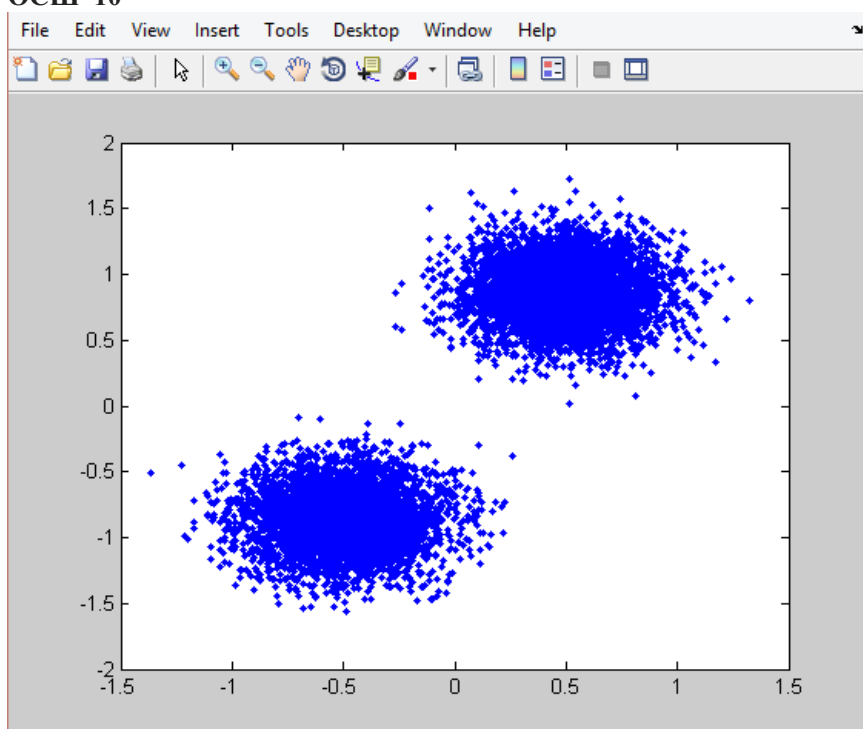
И ветер бился и летал
С ее летучим покрывалом?
Прекрасно море в бурной мгле
И небо в блесках без лазури;
Но верь мне: дева на скале
Прекрасней волн, небес и бури.

По условию нужно передать файл, состоящий из символов ASCII. Для это переведём данное стихотворение через таблицу ASCII символов, данное сообщение будет выглядеть следующим образом:

208 145 208 163 208 160 208 175 32 208 162 209 139 32 208 178 208 184 208 180 208 181 208
187 32 208 180 208 181 208 178 209 131 32 208 189 208 176 32 209 129 208 186 208 176 208
187 208 181 32 208 146 32 208 190 208 180 208 181 208 182 208 180 208 181 32 208 177 208
181 208 187 208 190 208 185 32 208 189 208 176 208 180 32 208 178 208 190 208 187 208 189
208 176 208 188 208 184 32 208 154 208 190 208 179 208 180 208 176 44 32 208 177 209 131
209 136 209 131 209 143 32 208 178 32 208 177 209 131 209 128 208 189 208 190 208 185 32
208 188 208 179 208 187 208 181 44 32 208 152 208 179 209 128 208 176 208 187 208 190 32
208 188 208 190 209 128 208 181 32 209 129 32 208 177 208 181 209 128 208 181 208 179 208
176 208 188 208 184 44 32 208 154 208 190 208 179 208 180 208 176 32 208 187 209 131 209
135 32 208 188 208 190 208 187 208 189 208 184 208 185 32 208 190 208 183 208 176 209 128
209 143 208 187 32 208 149 208 181 32 208 178 209 129 208 181 209 135 208 176 209 129 208
189 208 190 32 208 177 208 187 208 181 209 129 208 186 208 190 208 188 32 208 176 208 187
209 139 208 188 32 208 152 32 208 178 208 181 209 130 208 181 209 128 32 208 177 208 184
208 187 209 129 209 143 32 208 184 32 208 187 208 181 209 130 208 176 208 187 32 208 161 32
208 181 208 181 32 208 187 208 181 209 130 209 131 209 135 208 184 208 188 32 208 191 208
190 208 186 209 128 209 139 208 178 208 176 208 187 208 190 208 188 63 32 208 159 209 128
208 181 208 186 209 128 208 176 209 129 208 189 208 190 32 208 188 208 190 209 128 208 181
32 208 178 32 208 177 209 131 209 128 208 189 208 190 208 185 32 208 188 208 179 208 187
208 181 32 208 152 32 208 189 208 181 208 177 208 190 32 208 178 32 208 177 208 187 208 181
209 129 208 186 208 176 209 133 32 208 177 208 181 208 183 32 208 187 208 176 208 183 209
131 209 128 208 184 59 32 208 157 208 190 32 208 178 208 181 209 128 209 140 32 208 188 208
189 208 181 58 32 208 180 208 181 208 178 208 176 32 208 189 208 176 32 209 129 208 186 208
176 208 187 208 181 32 208 159 209 128 208 181 208 186 209 128 208 176 209 129 208 189 208
181 208 185 32 208 178 208 190 208 187 208 189 44 32 208 189 208 181 208 177 208 181 209
129 32 208 184 32 208 177 209 131 209 128 208 184 46

Рассмотрим полученные сообщения и сигнальные созвездия при различных ОСШ:

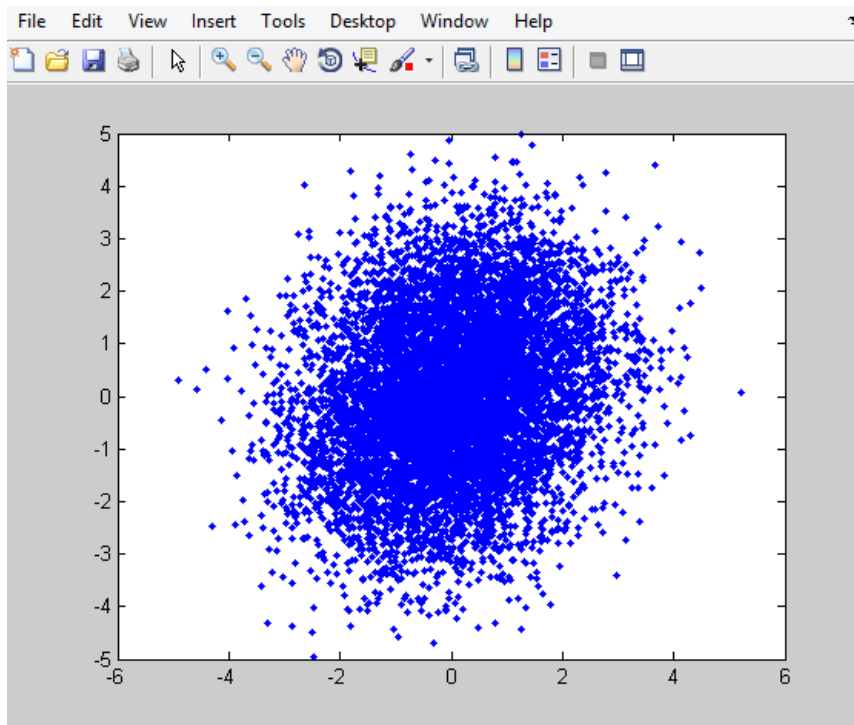
ОСШ=10



208 145 208 163 208 160 208 175 32 208 162 209 139 32 208 178 208 184 208 180 208 181

208 187 32 208 180 208 181 208 178 209 131 32 208 189 208 176 32 209 129 208 186 208 176
208 187 208 181 32 208 146 32 208 190 208 180 208 181 208 182 208 180 208 181 32 208 177
208 181 208 187 208 190 208 185 32 208 189 208 176 208 180 32 208 178 208 190 208 187 208
189 208 176 208 188 208 184 32 208 154 208 190 208 179 208 180 208 176 44 32 208 177 209
131 209 136 209 131 209 143 32 208 178 32 208 177 209 131 209 128 208 189 208 190 208 185
32 208 188 208 179 208 187 208 181 44 32 208 152 208 179 209 128 208 176 208 187 208 190 32
208 188 208 190 209 128 208 181 32 209 129 32 208 177 208 181 209 128 208 181 208 179 208
176 208 188 208 184 44 32 208 154 208 190 208 179 208 180 208 176 32 208 187 209 131 209
135 32 208 188 208 190 208 187 208 189 208 184 208 185 32 208 190 208 183 208 176 209 128
209 143 208 187 32 208 149 208 181 32 208 178 209 129 208 181 209 135 208 176 209 129 208
189 208 190 32 208 177 208 187 208 181 209 129 208 186 208 190 208 188 32 208 176 208 187
209 139 208 188 32 208 152 32 208 178 208 181 209 130 208 181 209 128 32 208 177 208 184
208 187 209 129 209 143 32 208 184 32 208 187 208 181 209 130 208 176 208 187 32 208 161 32
208 181 208 181 32 208 187 208 181 209 130 209 131 209 135 208 184 208 188 32 208 191 208
190 208 186 209 128 209 139 208 178 208 176 208 187 208 190 208 188 63 32 208 159 209 128
208 181 208 186 209 128 208 176 209 129 208 189 208 190 32 208 188 208 190 209 128 208 181
32 208 178 32 208 177 209 131 209 128 208 189 208 190 208 185 32 208 188 208 179 208 187
208 181 32 208 152 32 208 189 208 181 208 177 208 190 32 208 178 32 208 177 208 187 208 181
209 129 208 186 208 176 209 133 32 208 177 208 181 208 183 32 208 187 208 176 208 183 209
131 209 128 208 184 59 32 208 157 208 190 32 208 178 208 181 209 128 209 140 32 208 188 208
189 208 181 58 32 208 180 208 181 208 178 208 176 32 208 189 208 176 32 209 129 208 186 208
176 208 187 208 181 32 208 159 209 128 208 181 208 186 209 128 208 176 209 129 208 189 208
181 208 185 32 208 178 208 190 208 187 208 189 44 32 208 189 208 181 208 177 208 181 209
129 32 208 184 32 208 177 209 131 209 128 208 184 46

ОСШ=-5



Принятое сообщение

208 145 208 16 3 2 160 "08 175 32 208 162 209 139 32 208 178 208 184 208 180 208 18 1 000
17 32 2 08 180 208 18! 208 178 209 131 32 208 189 200 76 32 209 129208 186 2 08 176
208 187 208 181 32 208 146 32 208 190 208 180 208 181 208 182 208 180 208 181 32 208 177 208 181 208
187 008 19 200 18 5 32 208 189 208 176 208 180 32 208 178 208 190 208 187 208 19 208 17 6
208 188 208 14 32 208 15д 608 190 08 0 79 208 180 208 176 44 32 208 177 209 131 209 136 209
131 29 343 12 2 8 178 32 208 1 73209 131 209 128 208 189 208 190 208 185 32 208 188 208 179
208 187 208 181 44 32 08 152 "0817 9 "09 138 208 176 208 187 20 8 180 3r 28 1 88 208 190
209 128 208 181 32 209 129 32 2 08 177 b08 01 209 128 208 181 208 179 208 176 200 1882 08 184
44 32 2 8 150 08 19 208 179 208 1 80 08 166 32 208 187 209 131 209 135 32 208 188 208 190 208
187 208 189 208 184 208 185 32 208 190 208 183 208 166 2081"8 20 9 143 208 187 32 208 149 208
181 32 208 178 209 129 208 181 209 135 208 176 209 129 208 189 208 190 32 208 177 208 187 208 181
209 129 208 186 208 190 208 188 3 2081' 6 208 187 209 139 208 188 32 208 152 32 208 178 208 181

209 130 208 181 209 128 32 208 177 208 184 2 8 17 209 ! "9 r09 0\$3 #2 B08 184 22 208
(187 20181 2 09 130 208 176 208 187 32 20 8 16132 200 1 8 208 !81 32 208 187 208 181 209
130 209 131 209 !35 2 8 184 208 188 20 208 190 208 90 20(186 08 128 "09 128 208 178
208 1 76 28 18 20X 190 208 188 63 32 208 159 209 128 208 181 208 186 209 128 208 176 209 12=
208 1() 2 08 190 32 208 188 200 !y0 20 9 128 208 181 32 208 178 32 200 1'7 29 13 1 209 128
208 189 208 190 208 185 32 208 188 208 179 208 187 208 181 32 208 152 32 208 189 208 181 208 177 208
190 32 208 178 32 208 177 208 17 20 181 2 09 129 208 10 2 8 12 209 133 32 208 177 208 1 208
1(3 20(18 08 172 "09 183 201 121209 ! 28 208 184 59 32 208 157 208 190 32 208 178
208 181 209 128 209 140 32 208 88 208 9 208)81 58 32 208 80208 181 208 178 208 176 32
208 189 208 176 2 209 129 208 186 208 176 208 187 208 181 3 2 2 8 15) 209 128 208 181 20
8086 209 128 208 176 209 129 208 189 208 181 208 185 32 208 178 208 190 208 187 208 189 44 32
208 189 208 181 208 177 208 181 :09 1"9 32 0 8 184 32 200 77 209 131 209 128 208 184 46

Данное сообщения принято с множеством ошибок, из чего следует вывод что сообщения восстановлению не подлежит.

Список используемой литературы:

1. Бакке А.В. «Лекции по курсу ССПО»;
2. Бакке А.В. «Лекции по курсу ОТССПО»;
- 3.<http://omoled.ru/publications/view/565>
- 4.<http://omoled.ru/publications/view/557>
5. Педжман Рошан, Джонатан Лиэри. Основы построения беспроводных локальных сетей стандарта 802.11

