

СПбГУАП

КАФЕДРА № 52

ОТЧЕТ  
ЗАЩИЩЕН С ОЦЕНКОЙ

ПРЕПОДАВАТЕЛЬ

ассистент

\_\_\_\_\_  
должность, уч. степень, звание

\_\_\_\_\_  
дата

А.В. Борисовская

\_\_\_\_\_  
инициалы, фамилия

ОТЧЕТ О ЛАБОРАТОРНОЙ РАБОТЕ № 2

## ПЕРЕДАЧА ДАННЫХ В СИСТЕМАХ С ОБРАТНОЙ СВЯЗЬЮ

по курсу: ОСНОВЫ ПОСТРОЕНИЯ ИНФОКОММУНИКАЦИОННЫХ СИСТЕМ И СЕТЕЙ

РАБОТУ ВЫПОЛНИЛА

СТУДЕНТ ГР. №

5723

\_\_\_\_\_  
дата

Е.А. Иванова

\_\_\_\_\_  
инициалы, фамилия

Санкт-Петербург 2020

# Содержание

<b>1</b>	<b>Цель работы</b>	<b>3</b>
1.1	Задание . . . . .	3
1.2	Дополнительное задание . . . . .	3
<b>2</b>	<b>Теория к работе</b>	<b>3</b>
2.1	Базовая модель систем с обратной связью . . . . .	3
2.2	Описание алгоритма с виртуальными каналами . . . . .	4
<b>3</b>	<b>Результаты выполнения исследований</b>	<b>5</b>
3.1	Графические результаты . . . . .	5
3.2	Примеры работы программы . . . . .	6
<b>4</b>	<b>Выводы</b>	<b>9</b>

# 1 Цель работы

Целью данной работы является исследование типовых алгоритмов передачи данных в системах с решающей обратной связью, оценка коэффициента использования канала в рамках базовой модели для систем с решающей обратной связью с использованием численного расчета и имитационным моделированием.

## 1.1 Задание

Написать моделирующую программу для алгоритма с виртуальными каналами. Размер буфера на приемнике имеет общее ограничение и делится между всеми виртуальными каналами. Сообщения выдаются пользователю только в соответствии с правильным порядком следования, определяемого нумерацией на источнике.

## 1.2 Дополнительное задание

# 2 Теория к работе

## 2.1 Базовая модель систем с обратной связью

Предполагается, что сообщения, передаваемые по прямому каналу, состоят из данных и контрольной суммы. Использование контрольной суммы позволяет на приемной стороне определить наличие ошибок. Канал передачи от источника к получателю называется прямым каналом. Канал передачи от приемника к источнику называется обратным каналом. На рисунке 8 представлена базовая модель передачи данных по каналу с обратной связью, где И – источник сообщений, П – приемник, КС – контрольная сумма для передаваемых данных.



Рис. 1: Схема передачи данных по каналу с обратной связью

Введем следующие допущения:

1. При передаче данных в прямом канале связи с вероятностью  $p$  могут возникать ошибки. Если при передаче произошли ошибки, то приемник всегда их обнаруживает за счет контрольной суммы.
2. Приемник проверяет получение сообщения на наличие ошибок. Если ошибок нет, он отправляет положительную квитанцию по обратному каналу, а данные передает на дальнейшую обработку. В противном случае, получатель отправляет отрицательную квитанцию, а данные стирает.
3. При передаче квитанции по обратному каналу, может произойти ошибка с вероятностью  $p_{обр}$ . Ошибки при передаче квитанции всегда обнаруживаются (в случае

ошибки источник не знает, какая квитанция передавалась, при этом положительная квитанция не может стать отрицательной и наоборот).

4. Все сообщения, которые передает источник, имеют одинаковую длину. Время передачи сообщения принято за единицу времени, а время передачи квитанции считается равным нулю (см. рис. 2). Источник получает квитанцию о результате передачи через  $\tau$  единиц времени после окончания передачи сообщения, где  $\tau$  – целое число.

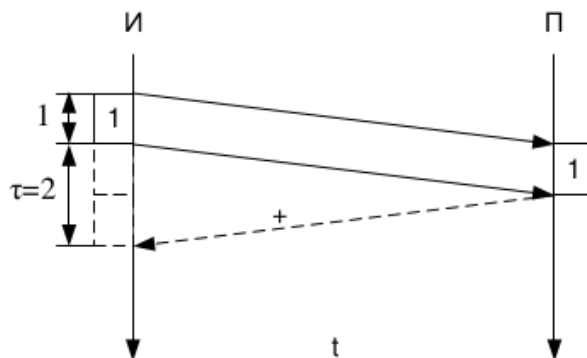


Рис. 2: Пример работы системы в соответствии с допущением 4 при  $\tau = 2$

5. События, связанные с ошибками в прямом и обратном канале, считаются независимыми. События, которые произошли в разные моменты времени в одном канале, так же считаются независимыми.

Передача по такой системе может выполняться с помощью некоторого алгоритма, который описывает последовательность действий источника и получателя. Важнейшей характеристикой такой системы является коэффициент использования канала, который определяется следующим образом:

$$\eta = \lim_{T \rightarrow \infty} \frac{N(T)}{T}$$

где  $\eta$  - коэффициент использования канала,  $T$  - интервал работы системы,  $N(T)$  - число сообщений, переданных за интервал времени  $T$ .

## 2.2 Описание алгоритма с виртуальными каналами

Предполагается, что на источнике есть непрерывный поток сообщений и задержка получения квитанции равна  $\tau$  единиц времени (см. допущение 4). В этом случае система разбивается на  $\tau + 1$  виртуальных каналов. На приемнике необходимо иметь систему из  $\tau + 1$  буферов, в общем случае неограниченного объема, для хранения принятых сообщений. То есть можно перейти от одного канала с задержкой к нескольким виртуальным каналам без задержки. Алгоритм с виртуальными каналами работает следующим образом. По каждому виртуальному каналу источник передает сообщение, до тех пор, пока не будет получена положительная квитанция. При получении положительной квитанции источник начинает передачу следующего сообщения по соответствующему виртуальному каналу. Сообщения, принятые без ошибок, но не соответствующие порядку сохраняются в буфере соответствующему виртуальному каналу, в котором был принят сигнал. Сообщения передаются на вышележащий уровень из буфера в соответствии с требуемым порядком. Пример работы алгоритма с виртуальными каналами при  $\tau = 2$  (при этом виртуальных каналов три) приведен на рисунке 3.

Даже если вероятность ошибки в канале ненулевая, но буфер на приемной стороне не ограничен, то коэффициент использования канала не зависит от времени задержки получения квитанции ( $\tau$ ). Можно доказать, что если на приемной стороне буфер бесконечной длины, то коэффициент использования канала определяется по формуле:  $(1 - p)$ .

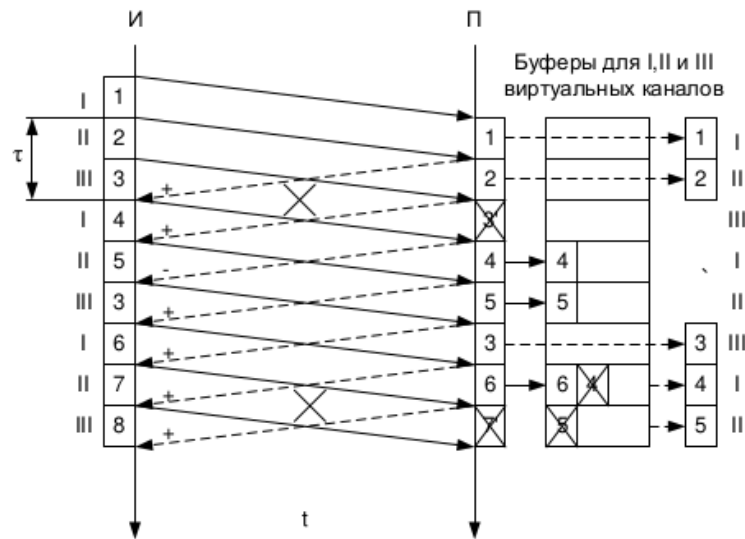


Рис. 3: Пример работы алгоритма с виртуальными каналами при  $\tau = 2$  и тремя виртуальными каналами

### 3 Результаты выполнения исследований

#### 3.1 Графические результаты

По результатам выполнения исследований были построены следующие графики. На Рис. 4 представлен график зависимости среднего числа повторных передач (сверху) и коэффициента использования канала (снизу) от вероятности ошибки в канале.

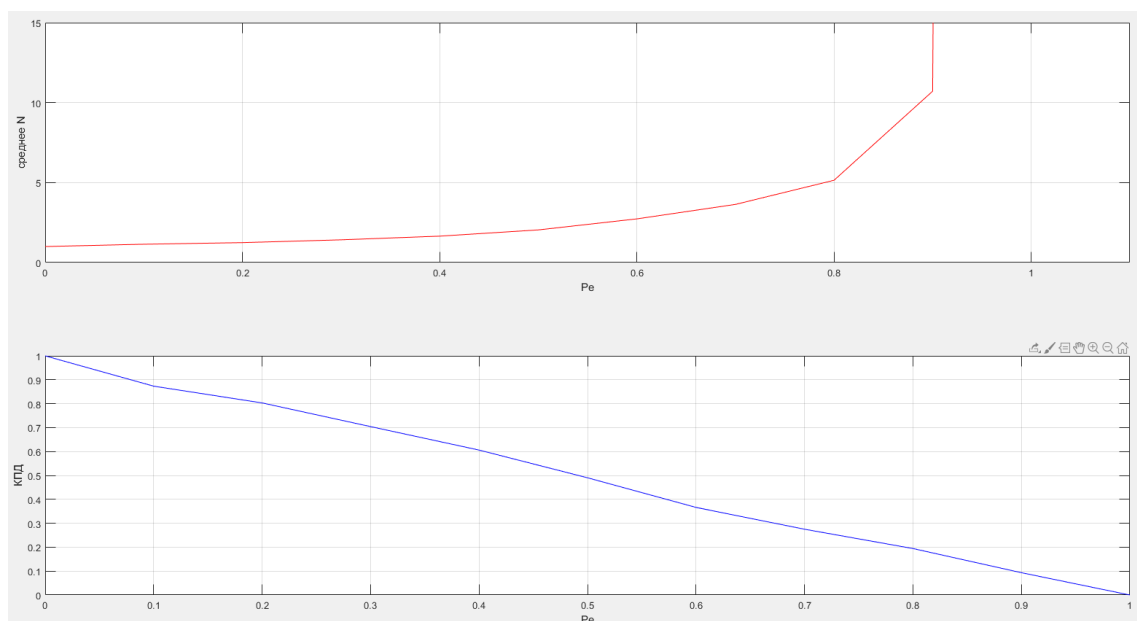


Рис. 4: Зависимости от вероятности ошибки в прямом канале ( $\tau = const$ )

На этих графиках видно, что с увеличением вероятности ошибки увеличивается среднее число повторных передач и уменьшается коэффициент использования канала.

На Рис. 5 показаны графики зависимостей среднего числа повторных передач (сверху) и коэффициента использования канала (снизу) от  $\tau$  - задержки получения квитанции.

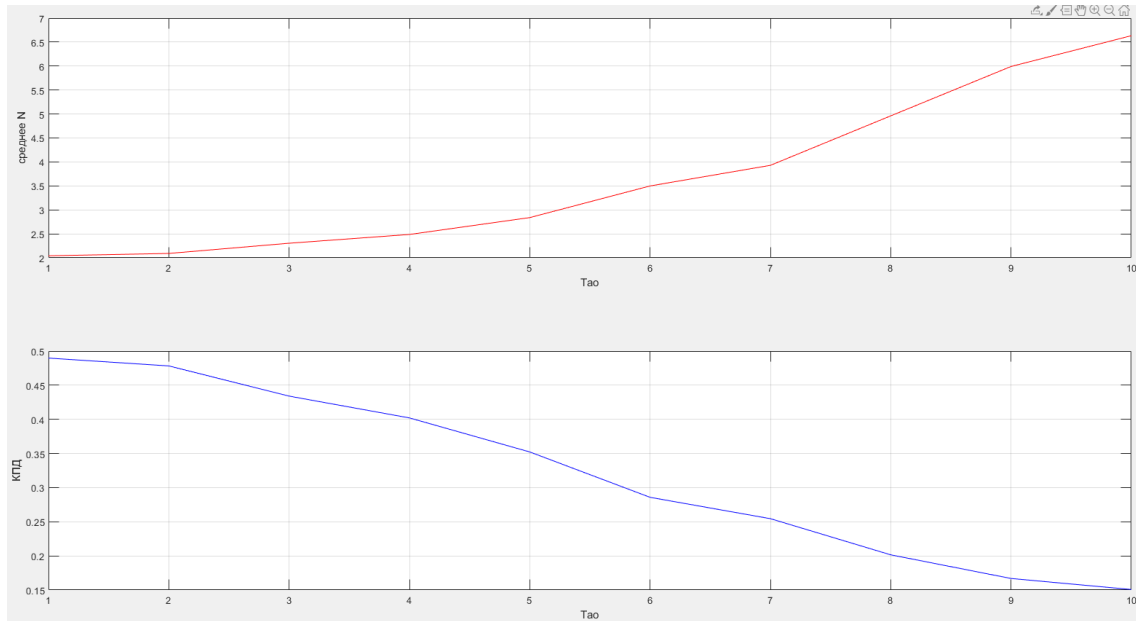


Рис. 5: Зависимости от времени передачи квитанции по обратному каналу ( $p = const$ )

### 3.2 Примеры работы программы

В этом разделе представлены логи-диаграммы работы программы с алгоритмом, выведенные в текстовый файл. В первом примере размер буфера равен 5, он не успевает переполняться при данном значении вероятности ошибки.

Листинг 1: Пример работы №1

```

t = 2
p = 0.5
Num of channels = 3
Buffer size = 5

1 m1                                buffer :
2          m1+ ->
3 m2                                buffer :
4          m2- error in channel
5 m3                                buffer :
6          m3- error in channel
7 m4                                buffer :
8          m4- error in channel
9 m2                                buffer :
10         m2- error in channel
11 m3                                buffer :
12         m3- error in channel
13 m4                                buffer :
14         m4+ to buf

```

15	m2		buffer : 4
16		m2- error in channel	
17	m3		buffer : 4
18		m3+ to buf	
19	m5		buffer : 3 4
20		m5+ to buf	
21	m2		buffer : 5 3 4
22		m2+ ->	
23		m3 from buf ->	
24		m4 from buf ->	
25		m5 from buf ->	
26	m6		buffer :
27		m6+ ->	
28	m7		buffer :
29		m7+ ->	
30	m8		buffer :
31		m8+ ->	
32	m9		buffer :
33		m9+ ->	
34	m10		buffer :
35		m10+ ->	

Во втором примере размер буфера был уменьшен до 2, можно заметить случаи удаления успешно принятых сообщений из-за переполнения буфера.

Листинг 2: Пример работы №2

t = 2			
p = 0.5			
Num of channels = 3			
Buffer size = 2			
1	m1		buffer :
2		m1- error in channel	
3	m2		buffer :
4		m2+ to buf	
5	m3		buffer : 2
6		m3- error in channel	
7	m1		buffer : 2
8		m1+ ->	
9		m2 from buf ->	
10	m4		buffer :
11		m4- error in channel	
12	m3		buffer :
13		m3- error in channel	
14	m5		buffer :
15		m5- error in channel	
16	m4		buffer :
17		m4- error in channel	

18	m3		buffer :
19		m3- error in channel	
20	m5		buffer :
21		m5- error in channel	
22	m4		buffer :
23		m4+ to buf	
24	m3		buffer : 4
25		m3- error in channel	
26	m5		buffer : 4
27		m5- error in channel	
28	m6		buffer : 4
29		m6- error in channel	
30	m3		buffer : 4
31		m3+ ->	
32		m4 from buf ->	
33	m5		buffer :
34		m5+ ->	
35	m6		buffer :
36		m6- error in channel	
37	m7		buffer :
38		m7+ to buf	
39	m8		buffer : 7
40		m8- error in channel	
41	m6		buffer : 7
42		m6- error in channel	
43	m9		buffer : 7
44		m9- error in channel	
45	m8		buffer : 7
46		m8+ to buf	
47	m6		buffer : 8 7
48		m6- error in channel	
49	m9		buffer : 8 7
50		m9- no place in buffer	
51	m10		buffer : 8 7
52		m10- error in channel	
53	m6		buffer : 8 7
54		m6+ ->	
55		m7 from buf ->	
56		m8 from buf ->	
57	m9		buffer :
58		m9- error in channel	
59	m10		buffer :
60		m10- error in channel	
61	empty	subchannel	
62	m9		buffer :
63		m9+ ->	
64	m10		buffer :
65		m10+ ->	



## 4 Выводы

В ходе выполнения работы был смоделирован виртуальный канал с общим буфером. Были построены графики зависимостей  $\eta(p, \tau = \text{const})$ ,  $\eta(p = \text{const}, \tau)$ ,  $\bar{N}(p, \tau = \text{const})$ ,  $\bar{N}(p = \text{const}, \tau)$ , по которым были сделаны следующие заключения:

- Из графиков зависимостей  $\bar{N}(p = \text{const}, \tau)$ ,  $\eta(p = \text{const}, \tau)$  получена обратная зависимость коэффициента использования канала от вероятности ошибки в прямом канале и прямая зависимость среднего числа посторонних отправлений от этой вероятности. При вероятности ошибки, равной нулю, среднее число посторонних отправлений равно 1, то есть, все сообщения передаются с первого раза, коэффициент использования канала в этом случае равен 1, то есть, для передачи используется каждая единица времени и передача проходит успешно. При вероятности ошибки, равной единице, количество повторных отправлений бесконечно, а коэффициент использования канала равен 0, что говорит о том, что сообщения не доходят до получателя.
- Из графиков зависимостей  $\bar{N}(p, \tau = \text{const})$ ,  $\eta(p, \tau = \text{const})$  получено, что с увеличением задержки получения квитанции по обратному каналу, увеличивается среднее число повторных передач и уменьшается коэффициент использования канала. Это объясняется тем, что при увеличении задержки начинает чаще переполняться буфер, а значит, даже успешно принятые сообщения, которые по порядку не подходят для отправления получателю, удаляются. Они не подходят по порядку, потому что в одном из подканалов случается ошибка передачи, и текущее сообщение тормозит отправку получателю всех последующих сообщений, пока не будет успешно получено текущее.

Листинг 3: Построение зависимостей от вероятности ошибки в канале

```

1  clc
2  clear
3  close all
4
5  % 2 лаба 7 Вариант Зависимости от p
6
7  n = 200;
8  p = 0:0.1:1;
9  tao = 2;
10 num_channels = tao+1;
11 buf_size = 5;
12
13 kpd = zeros(1, length(p));
14 N = zeros(1, length(p));
15 N(end) = 1000;
16 kpd(end) = 0;
17 k = 0;
18 for pi=p(1:end-1)
19     k = k + 1;
20     buf = zeros(1, buf_size);
21     buf_size_now = buf_size;
22     messages_in_queue = 1:n;
23     messages_in_queue([1,2,3]) = 0;
24     messages_in_channels = 1:num_channels;
25     last_message_sent = 0;
26     last_message_recieved = 0;
27     channel_now = 0;
28     num_sends = 0;
29     while last_message_sent < n
30         num_sends = num_sends + 1;
31         channel_now = mod(channel_now, num_channels) + 1;
32         message_now = messages_in_channels(channel_now);
33         if message_now == 0
34             continue;
35         end
36         if rand() > pi
37             if last_message_sent == message_now - 1
38                 last_message_sent = message_now;
39             elseif buf_size_now > 0
40                 buf(buf_size_now) = message_now;
41                 buf_size_now = buf_size_now - 1;
42             else
43                 continue;
44             end
45             if last_message_recieved < message_now
46                 last_message_recieved = message_now;

```

```

47         end
48         next_message_index = find(messages_in_queue>0, 1);
49         if isempty(next_message_index)
50             messages_in_channels(channel_now) = 0;
51         else
52             messages_in_channels(channel_now) = messages_in_queue(
                    next_message_index);
53             messages_in_queue(next_message_index) = 0;
54         end
55         while ~isempty(find(buf==last_message_sent+1, 1))
56             buf(find(buf==last_message_sent+1, 1)) = 0;
57             buf_size_now = buf_size_now + 1;
58             last_message_sent = last_message_sent + 1;
59         end
60         put_in_buf = buf(buf~=0);
61         buf = zeros(1, buf_size);
62         if ~isempty(put_in_buf)
63             j=1;
64             for i=buf_size_now+1:buf_size
65                 buf(1, i) = put_in_buf(1, j);
66                 j=j+1;
67             end
68             buf_size_now = buf_size-length(put_in_buf);
69         end
70     end
71     end
72     N(k) = num_sends / n;
73     kpd(k) = n / num_sends;
74 end
75 figure;
76 subplot(2, 1, 1);
77 plot(p, N, 'r');
78 xlabel('Pe');
79 ylabel('среднее N');
80 axis([0, 1.1, 0, 15]);
81 xticks(0:0.2:1);
82 grid('On');
83 subplot(2, 1, 2);
84 plot(p, kpd, 'b');
85 xlabel('Pe');
86 ylabel('КПД');
87 grid('On');

```

Листинг 4: Построение зависимостей от задержки получения квитанции

```

1  clc
2  clear
3  close all
4
5  % 2 лаба 7 Вариант Зависимости от p
6

```

```

7 | n = 1000;
8 | p = 0.5;
9 | tao = 1:10;
10 | buf_size = 5;
11 |
12 | kpd = zeros(1, length(tao));
13 | N = zeros(1, length(tao));
14 | k = 0;
15 | for t=tao
16 |     num_channels = t+1;
17 |     k = k + 1;
18 |     buf = zeros(1, buf_size);
19 |     buf_size_now = buf_size;
20 |     messages_in_queue = 1:n;
21 |     messages_in_queue(1:num_channels) = 0;
22 |     messages_in_channels = 1:num_channels;
23 |     last_message_sent = 0;
24 |     last_message_recieved = 0;
25 |     channel_now = 0;
26 |     num_sends = 0;
27 |     while last_message_sent < n
28 |         num_sends = num_sends + 1;
29 |         channel_now = mod(channel_now, num_channels) + 1;
30 |         message_now = messages_in_channels(channel_now);
31 |         if message_now == 0
32 |             continue;
33 |         end
34 |         if rand() > p
35 |             if last_message_sent == message_now - 1
36 |                 last_message_sent = message_now;
37 |             elseif buf_size_now > 0
38 |                 buf(buf_size_now) = message_now;
39 |                 buf_size_now = buf_size_now - 1;
40 |             else
41 |                 continue;
42 |             end
43 |             if last_message_recieved < message_now
44 |                 last_message_recieved = message_now;
45 |             end
46 |             next_message_index = find(messages_in_queue>0, 1);
47 |             if isempty(next_message_index)
48 |                 messages_in_channels(channel_now) = 0;
49 |             else
50 |                 messages_in_channels(channel_now) = messages_in_queue(
51 |                     next_message_index);
52 |                 messages_in_queue(next_message_index) = 0;
53 |             end
54 |             while ~isempty(find(buf==last_message_sent+1, 1))
55 |                 buf(find(buf==last_message_sent+1, 1)) = 0;
56 |                 buf_size_now = buf_size_now + 1;

```

```

56         last_message_sent = last_message_sent + 1;
57     end
58     put_in_buf = buf(buf~=0);
59     buf = zeros(1, buf_size);
60     if ~isempty(put_in_buf)
61         j=1;
62         for i=buf_size_now+1:buf_size
63             buf(1, i) = put_in_buf(1, j);
64             j=j+1;
65         end
66         buf_size_now = buf_size-length(put_in_buf);
67     end
68 end
69 end
70 N(k) = num_sends / last_message_recieved;
71 kpd(k) = last_message_sent / num_sends;
72 end
73 figure;
74 subplot(2, 1, 1);
75 plot(tao, N, 'r');
76 xlabel('Tao');
77 ylabel('среднее N');
78 grid('On');
79 subplot(2, 1, 2);
80 plot(tao, kpd, 'b');
81 xlabel('Tao');
82 ylabel('КПД');
83 grid('On');

```

Листинг 5: Моделирование с логированием в файл

```

1  clc
2  clear
3  close all
4
5  % 2 лаба 7 Вариант Логи в файл (+вывод в консоль)
6
7  f = fopen('logs.txt', 'w');
8  n = 10;
9  p = 0.5;
10 tao = 2;
11 num_channels = tao+1;
12 buf_size = 5;
13
14 buf = zeros(1, buf_size);
15 buf_size_now = buf_size;
16 fprintf(f, '%s%s\n%s%s\n%s%s\n%s%s\n\n', ...
17         "t = ", int2str(tao), "p = ", num2str(p), ...
18         "Num of channels = ", int2str(num_channels), ...
19         "Buffer size = ", num2str(buf_size));
20 messages_in_queue = 1:n;

```

[illegible]

```

63         continue;
64     end
65     if last_message_recieved < message_now
66         last_message_recieved = message_now;
67     end
68     next_message_index = find(messages_in_queue>0, 1);
69     if isempty(next_message_index)
70         messages_in_channels(channel_now) = 0;
71     else
72         messages_in_channels(channel_now) = messages_in_queue(
73             next_message_index);
74         messages_in_queue(next_message_index) = 0;
75     end
76     % Доотправлять из буфера
77     while ~isempty(find(buf==last_message_sent+1, 1))
78         fprintf(f, '%s\t\t%s%s%s\n', num2str(num_str), "m",
79             int2str(last_message_sent+1), " from buf ->");
80         fprintf('%s\t\t%s%s%s\n', num2str(num_str), "m", int2str(
81             last_message_sent+1), " from buf ->");
82         num_str = num_str + 1;
83         buf(find(buf==last_message_sent+1, 1)) = 0;
84         buf_size_now = buf_size_now + 1;
85         last_message_sent = last_message_sent + 1;
86     end
87     put_in_buf = buf(buf~=0);
88     buf = zeros(1, buf_size);
89     if ~isempty(put_in_buf)
90         j=1;
91         for i=buf_size_now+1:buf_size
92             buf(1, i) = put_in_buf(1, j);
93             j=j+1;
94         end
95         buf_size_now = buf_size-length(put_in_buf);
96     end
97     else
98         fprintf(f, '%s\t\t%s%s%s\n', num2str(num_str), "m", int2str(
99             message_now), "- error in channel");
100         fprintf('%s\t\t%s%s%s\n', num2str(num_str), "m", int2str(
101             message_now), "- error in channel");
102         num_str = num_str + 1;
103     end
104 end

```