МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное образовательное
учреждение высшего образования
САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
АЭРОКОСМИЧЕСКОГО ПРИБОРОСТРОЕНИЯ

А. А. Бурков, А. М. Тюрликов

ИЗУЧЕНИЕ ПРИНЦИПОВ ФУНКЦИОНИРОВАНИЯ КАНАЛЬНОГО УРОВНЯ СЕТЕЙ ПЕРЕДАЧИ ДАННЫХ

Учебно-методическое пособие

ГУАП

Санкт-Петербург
2025

УДК 621.391
ББК 32.968
Б91

Рецензенты:
кандидат технических наук, доцент *О. И. Красильникова*;
кандидат технических наук, доцент *В. С. Коломойцев*

Утверждено
редакционно-издательским советом университета
в качестве учебно-методического пособия

Протокол № 4 от 2 октября 2025 г.

Бурков, А. А.

Б91 Изучение принципов функционирования канального уровня сетей передачи данных: учеб.-метод. пособие / А. А. Бурков, А. М. Тюрликов. – СПб.: ГУАП, 2025. – 70 с.

В пособии рассмотрены базовые принципы функционирования канального уровня сетей передачи данных, на которых основано построение протоколов канального уровня в инфокоммуникационных системах. Издание является дополнением к лекционному курсу по дисциплине «Основы построения инфокоммуникационных систем и сетей». Содержит теоретические сведения по рассматриваемым принципам организации передачи данных по сетям. Для закрепления теоретических знаний и отработки практических навыков разработаны и описаны три лабораторные работы, по одной на каждый рассматриваемый аспект функционирования канального уровня. При выполнении лабораторных работ требуется разработка программ имитационного моделирования. Для реализации этих программ могут быть использованы любые языки программирования.

Предназначено для студентов, обучающихся по направлениям и специальностям 10.03.01 – «Информационная безопасность», 11.03.02 – «Инфокоммуникационные технологии и системы связи», 10.05.03 – «Информационная безопасность автоматизированных систем», 10.05.05 – «Безопасность информационных технологий в правоохранительной сфере», а также студентов других технических направлений и специальностей, изучающих системы передачи данных.

Подготовлено кафедрой инфокоммуникационных технологий и систем связи ГУАП.

УДК 621.391
ББК 32.968

ВВЕДЕНИЕ

Модель взаимодействия открытых систем (модель Open Systems Interconnection или OSI) была разработана в начале 1980-х гг. Данная модель является эталонной и содержит 7 уровней (в порядке от верхнего к нижнему уровню): прикладной уровень, уровень представления, сеансовый уровень, транспортный уровень, сетевой уровень, канальный уровень и физический уровень. Несмотря на то, что семиуровневая модель OSI была предложена давно, любая современная система передачи данных базируется на заложенных в данную модель принципах.

В рамках концепции этой модели используются два важных понятия: протокол и интерфейс. Протокол – это набор правил, определяющих последовательность и формат сообщений, которыми обмениваются сетевые компоненты, лежащие на одном уровне разных узлов сети передачи данных. Интерфейс – это набор правил, определяющих последовательность и формат сообщений, которыми обмениваются сетевые компоненты, лежащие на соседних уровнях в рамках одного узла сети передачи данных.

Бурное развитие технологий передачи данных и, в первую очередь, беспроводных технологий приводит к совершенствованию интерфейсов взаимодействия физического уровня с физической средой и совершенствованию протоколов физического уровня. Что приводит к изменению протоколов канального уровня. Несмотря на развитие протоколов канального уровня, базовые принципы, на которых они основываются, остаются неизменными.

Пособие посвящено рассмотрению базовых принципов функционирования канального уровня сетей передачи данных, на которых основывается построение протоколов канального уровня. Учебно-методическое пособие содержит теоретические сведения и задания лабораторных работ для изучения и практического освоения лекционного курса по дисциплине «Основы построения инфокоммуникационных систем и сетей».

Выполнение лабораторных работ можно условно разбить на 3 этапа: сдача допусков к лабораторной работе, выполнение лабораторной работы, сдача отчета.

Все разделы устроены следующим образом:

• первый подраздел содержит сведения теоретические сведения, посвященные изучаемому принципу построения протоколов канального уровня;

• второй подраздел содержит список вопросов для закрепления теоретического материала и допуска к лабораторной работе, каждый пункт в данном подразделе является отдельным вопросом;

• третий подраздел описывает порядок действий при выполнении лабораторной работы;

• четвертый подраздел содержит список вариантов для выполнения лабораторной работы;

• пятый подраздел описывает требования к оформлению отчета по лабораторной работе.

Дополнительный теоретический материал содержится: для первой лабораторной работы в [1–3,7]; для второй лабораторной работы в [4, 5]; для третьей лабораторной работы в [5, 6]. Материалы курса лекций «Основы построения инфокоммуникационных систем и сетей», относящиеся ко всем лабораторным работам, содержатся в учебном пособии [8].

1. ИСПОЛЬЗОВАНИЕ ЦИКЛИЧЕСКИХ КОДОВ ДЛЯ ОБНАРУЖЕНИЯ ОШИБОК НА КАНАЛЬНОМ УРОВНЕ СЕТЕЙ ПЕРЕДАЧИ ДАННЫХ

Раздел посвящен изучению типового алгоритма формирования контрольной суммы на основе циклических кодов. Рассматриваются методы численного расчета значения вероятности ошибки декодирования (события, что декодер не обнаружит ошибки) и методы оценки вероятности ошибки декодирования с помощью имитационного моделирования.

1.1. Модель системы, алгоритм формирования контрольной суммы и методы анализа вероятности ошибки декодирования

1.1.1. Модель рассматриваемой системы

В большинстве современных систем передачи данных для обнаружения ошибок, возникающих в процессе передачи, применяется следующий подход. К передаваемым данным добавляют дополнительную информацию – контрольную сумму. Контрольная сумма вычисляется на основе данных (является некоторой функцией от данных). По каналу передается сообщение, состоящее из данных и контрольной суммы. Использование контрольной суммы позволяет определить, по принятому сообщению, возникли ли ошибки при передаче данного сообщения по каналу.

На рис. 1 изображена структурная схема рассматриваемой системы передачи данных.

На передающей стороне имеется блок «кодер», а на приемной стороне блок «декодер». Задача кодера состоит в том, что бы к передава-

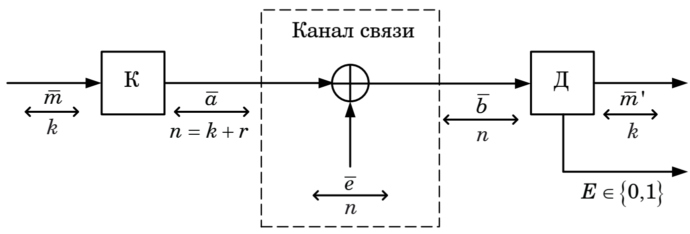

Рис. 1. Структурная схема системы передачи данных: $\overline{m}$ – информационное сообщение, К – блок кодера, $\overline{a}$ – закодированное сообщение, $\overline{e}$ – вектор ошибок, $\overline{b}$ – сообщение на выходе канала, Д – блок декодера, E – принятое решение, $\overline{m}'$ – информационное сообщение на выходе декодера

емому сообщению добавить некоторую *избыточность*. Задача декодера состоит в том, что бы по принятой из канала последовательности за счет добавленной *избыточности* определить содержатся ли в ней ошибки. Если декодер не обнаружил ошибки, то из принятой последовательности извлекается информационная часть и отдается получателю. Последовательность, формируемая на выходе кодера, является кодовым словом. Опишем подробнее схему, представленную на рис. 1:

• Некоторое информационное сообщение, состоящее из нулей и единиц, которое можно представить в виде вектора $\overline{m}$ длины k поступает на вход кодера.

• Кодер по некоторому алгоритму на основе вектора $\overline{m}$ вычисляет контрольную сумму, состоящую из r нулей и единиц, дописывает ее к передаваемому сообщению $\overline{m}$ и таким образом формирует закодированное сообщение (кодовое слово) $\overline{a}$ так же состоящее из нулей и единиц длиной $n=k+r$.

• Далее закодированное сообщение $\overline{a}$ передается по каналу связи. В канале могут произойти ошибки, в результате которых некоторые биты сообщения инвертируются («ноль» становится «единицей» или «единица» становится «нулем»). Для описания процесса возникновения ошибок вводится вектор ошибок $\overline{e}$ длины n, который показывает, на каких позициях произошла ошибка (позиция с ошибкой содержит символ «единица», а на остальных позициях символ «ноль»). При таком подходе сам процесс возникновения ошибки может быть представлен как побитовая операция «исключающее ИЛИ» (операция XOR или для краткости обозначений будем использовать символ $\oplus$) между передаваемым сообщением $\left(\overline{m}\right)$ и вектором ошибок $\left(\overline{e}\right)$.

Рассмотрим пример работы канала. Пусть передавалось закодированное сообщение $\overline{a}=\left(101\right)$, и в канале произошла ошибка в последнем бите, то есть $\overline{e}=\left(001\right)$. Тогда на выходе канала будет сообщение $\overline{b}=\overline{a}\oplus\overline{e}=\left(100\right)$.

• Декодер по некоторому алгоритму с использованием контрольной суммы определяет наличие ошибок в принятом сообщении $\overline{b}$ и принимает одно из следующих решений:

$$E=\begin{cases}1, \text{ если обнаружены ошибки}\\ 0, \text{ иначе}\end{cases}.$$

1.1.2. Работа кодера и декодера

Рассматриваются только двоичные коды. Для описания работы с двоичными кодами используются многочлены с коэффициентами из GF(2) (Galois field, поле Галуа или конечное поле). Подробная информация о полях Галуа содержится в [7].

Кодер хранит порождающий многочлен $g(x)$. Степень порождающего многочлена равна r. С помощью математических обозначений это записывается как $\deg(g(x))=r$. Значение r равно числу бит контрольной суммы в кодовом слове. k – число информационных символов передаваемого сообщения $\overline{m}$.

Передаваемое сообщение рассматривается как двоичный вектор длины k. Для каждого сообщения $\left(\overline{m}\right)$ кодер выполняет следующие действия (шаги):

Шаг 1. На основе двоичного вектора $\overline{m}$ формируется многочлен $m(x)$. Степень многочлена $m(x)$ при этом меньше или равна k – 1;

Шаг 2. Вычисляется многочлен $c(x)=m(x)x^r \bmod g(x)$. Степень многочлена $c(x)$ при этом меньше или равна r – 1. Двоичный вектор $\overline{c}$, соответствующий многочлену $c(x)$, является контрольной суммой для сообщения $\overline{m}$;

Шаг 3. Вычисляется многочлен $a(x)=m(x)x^r+c(x)$;

Шаг 4. На основе многочлена $a(x)$ формируется двоичный вектор $\overline{a}$, длина которого n бит, где $n=k+r$.

Рассмотрим пример работы кодера:

<table>
<tr><td colspan="3">Порождающий многочлен: $g(x)=x^3+x+1 \rightarrow r=\deg(g(x))=3$</td></tr>
<tr><td colspan="3">Передаваемое сообщение: $\overline{m}=\left(1010\right) \rightarrow k=4$</td></tr>
<tr><td>Шаг</td><td colspan="2">Действия</td></tr>
<tr><td>1</td><td colspan="2">$\overline{m}=\left(1010\right) \rightarrow m\left(x\right)=1x^3+0x^2+1x+0x^0=x^3+x$</td></tr>
<tr><td rowspan="2">2</td><td colspan="2">$c(x)=m(x)x^r \bmod g(x)=$
$=(x^3+x)x^3 \bmod (x^3+x+1))=(x^6+x^4) \bmod (x^3+x+1)$;</td></tr>
<tr><td>Вычисление остатка:
$\begin{array}{l|l} -\,x^6+x^4 & x^3+x+1 \\ \quad x^6+x^4+x^3 & x^3+1 \\ \hline \quad -\,x^3 & \\ \quad\quad x^3+x+1 & \\ \hline \quad\quad\quad x+1 & \end{array}$</td><td>$\rightarrow c(x)=x+1$</td></tr>
<tr><td>3</td><td colspan="2">$a(x)=m(x)x^r+c(x)=(x^3+x)x^3+x+1=x^6+x^4+x+1$</td></tr>
<tr><td>4</td><td colspan="2">$a(x)=1x^6+0x^5+1x^4+0x^3+0x^2+1x+1x^0$;
$\rightarrow \overline{a}=\left(1010011\right)$; $n=k+r=4+3=7$</td></tr>
</table>

В курсе лекций доказывается следующее утверждение [8].

Утверждение 1. Многочлен, полученный на третьем шаге описанного алгоритма для любого $m(x)$, будет делиться без остатка на порождающий многочлен $g(x)$, то есть $a(x)\mathrm{mod}g(x)=0$.

В соответствии с данным свойством может быть предложен следующий алгоритм декодирования.

Декодер хранит порождающий многочлен $g(x)$. Для каждой принятой из канала последовательности $\overline{b}$ длины n выполняет следующие действия (шаги):

Шаг 1. Принятое сообщение $\overline{b}$ переводится в многочлен $b(x)$;

Шаг 2. Вычисляется синдром: $s(x)=b(x)\mathrm{mod}g(x)$;

Шаг 3. Если $s(x)\neq 0$, то декодер выносит решение, что произошли ошибки ($E=1$), иначе декодер выносит решение, что ошибки не произошли ($E=0$):

$$E=\begin{cases}1, \ s(x)\neq 0\\ 0, \ s(x)=0\end{cases}.$$

Рассмотрим работу декодера на нескольких примерах.

Пример 1. На вход канала поступило кодовое слово $\overline{a}=(1010011)$. Пусть ошибки произошли на позициях соответствующих вектору $\overline{e}=(1100101)$. Тогда $\overline{b}=\overline{a}\oplus\overline{e}=(1010011)\oplus(1100101)=(0110110)$.

Работа декодера для примера 1:

<table>
<tr><td colspan="3">Порождающий многочлен: $g(x)=x^3+x+1 \to r=\deg(g(x))=3$</td></tr>
<tr><td colspan="3">Принятая из канала последовательность $\overline{b}=(0110110)$.</td></tr>
<tr><td>Шаг</td><td colspan="2">Действие</td></tr>
<tr><td>1</td><td colspan="2">$\overline{b}=(10110110)\to$
$\to b(x)=0x^6+1x^5+1x^4+0x^3+1x^2+1x+0x^0=x^5+x^4+x^2+x$</td></tr>
<tr><td rowspan="2">2</td><td colspan="2">$s(x)=(x^5+x^4+x^2+x)\mathrm{mod}(x^3+x+1)$;</td></tr>
<tr><td>Вычисление остатка:
$$\begin{array}{l|l} -\,x^5+x^4+x^2+x & x^3+x+1 \\ \quad\underline{x^5+x^3+x^2} & x^2+x+1 \\ \qquad -\,x^4+x^3+x & \\ \qquad\quad \underline{x^4+x^2+x} & \\ \qquad\qquad -\,x^3+x^2 & \\ \qquad\qquad\quad \underline{x^3+x+1} & \\ \qquad\qquad\quad x^2+x+1 & \end{array}$$</td><td>$\to s(x)=x^2+x+1$</td></tr>
<tr><td>3</td><td colspan="2">$s(x)\neq 0 \to E=1 \to$ произошли ошибки</td></tr>
</table>

В первом примере декодер обнаружил ошибки, произошедшие в канале, и вынес верное решение.

Пример 2. На вход канала поступило кодовое слово $\overline{a}=(1010011)$. Пусть ошибки произошли на позициях соответствующих вектору $\overline{e}=(0011101)$. Тогда $\overline{b}=\overline{a}\oplus\overline{e}=(1010011)\oplus(0011101)=(1001110)$.

Работа декодера для примера 2:

<table>
<tr><td colspan="3">Порождающий многочлен: $g(x)=x^3+x+1 \to r=\deg(g(x))=3$</td></tr>
<tr><td colspan="3">Принятая из канала последовательность $\overline{b}=(1001110)$.</td></tr>
<tr><td>Шаг</td><td colspan="2">Действие</td></tr>
<tr><td>1</td><td colspan="2">$\overline{b}=(1001110)\to$
$\to b(x)=1x^6+0x^5+0x^4+1x^3+1x^2+1x+0x^0=x^6+x^3+x^2+x$</td></tr>
<tr><td rowspan="2">2</td><td colspan="2">$s(x)=(x^6+x^3+x^2+x)\bmod(x^3+x+1);$</td></tr>
<tr><td>Вычисление остатка:
$$\begin{array}{l|l} -\,x^6+x^3+x^2+x & x^3+x+1 \\ \hline \underline{x^6+x^4+x^3} & x^3+x \\ \quad -\,x^4+x^2+x & \\ \quad \underline{\;\;x^4+x^2+x} & \\ \qquad\qquad 0 & \end{array}$$</td><td>$\to s(x)=0$</td></tr>
<tr><td>3</td><td colspan="2">$s(x)=0 \to E=0 \to$ ошибок не было</td></tr>
</table>

Во втором примере декодер не обнаружил ошибки, произошедшие в канале, и вынес неверное решение. Такое событие будем называть ошибкой декодирования, то есть:

$$\text{Ошибка декодирования }=\begin{cases}\overline{e}\neq 0\\ E=0\end{cases}.$$

где $\overline{e}$ – вектор ошибок, E – решение, принятое декодером.

Кодом называется множество последовательностей, которое может появиться на выходе кодера, при поступлении на вход всех возможных информационных последовательностей. Список всех таких кодовых последовательностей называется множеством кодовых слов (A), а мощность этого множества $|A|$ – это мощность кода. Если длина информационного сообщения равна k, то $|A|=2^k$.

В курсе лекций доказывается следующее утверждение [8].

Утверждение 2. Ошибка декодирования происходит в том случае, если вектор ошибок принадлежит множеству кодовых слов:

$$\overline{e} \in A.$$

Число ошибок, которое гарантированно может обнаружить декодер, зависит от такого параметра кода, как минимальное расстояние d. Минимальным расстоянием кода d называется наименьшее число несовпадающих позиций между двумя любыми кодовыми словами.

В курсе лекций доказывается следующее утверждение [8]:

Утверждение 3: Ранее описанная система кодирования и декодирования позволяет обнаружить ошибки, число которых не превышает $d - 1$, остатка вне зависимости от того, где эти ошибки произошли и того, какое кодовое слово передавалось.

В курсе помехоустойчивого кодирования доказывается, что для рассматриваемого алгоритма построения кодовых слов минимальное расстояние кода может быть определено как минимальный вес кодового слова без учета слова состоящего из всех нулей.

Пусть $\overline{x}$ – двоичный вектор. Под весом вектора $\overline{x}$ подразумевается число единиц в векторе $\overline{x}$ и записывается как $w(\overline{x})$. Например, если $\overline{x} = (1001101)$, то $w(\overline{x}) = w(1001101) = 4$.

Программа расчета минимального расстояния для циклического кода может быть кратко описана в виде следующего псевдокода:

Вход: $g(x)$ – порождающий многочлен; k – длина кодируемой последовательности
Выход: d – минимальное расстояние кода
/*Вычисление длины кодового слова*/
$n \leftarrow k + \deg(g(x))$
/*Инициализация начального значения расстояния кода*/
$d \leftarrow n$
/*Цикл по всем возможным сообщениям $\overline{m}_i$, где $i \in \{0, 2^k - 1\}$ за исключением нулевого кодового слова, состоящего из всех нулей*/
for $i = 1 \ldots 2^k - 1$:
  $\overline{a}_i \leftarrow$ Формирование кодового слова для $\overline{m}_i$ по алгоритму
  if $w(\overline{a}_i) < d$
    $d \leftarrow w(\overline{a}_i)$
  endif
endfor

1.1.3. Вычисление верхней оценки ошибки декодирования

Рассматриваться модель двоично-симметричного канала (ДСК) без памяти, представленная на рис. 2. Как видно из рис. 2, с вероятностью p происходит ошибка («ноль» становится «единицей» или «единица» становится «нулем»). Канал является двоичным, поэтому возможны только два значения сигналов на входе и выходе канала: $\{0,1\}$. Канал называется симметричным ввиду того, что вероятность ошибки для обоих значений битов одинакова. Канал без памяти характеризуется тем, что случайные события, связанные с ошибками в канале независимы для разных моментов времени.

Пусть заданы: порождающий многочлен $g(x)$, длина кодируемой последовательности k, минимальное расстояние кода d, вероятность ошибки в канале p. Необходимо вычислить верхнюю оценку вероятности ошибки декодирования P_e^+.

Рассмотрим два множества векторов ошибок:

$$A = \{\overline{e} : \text{при которых произошла ошибка декодирования}\} = \{\overline{e} : \overline{e} \neq 0 \,\&\, E = 0\},$$

$$B = \{\overline{e} : w(\overline{e}) \geq d\},$$

где $w(\overline{e})$ – вес вектора ошибок.

Обозначим через P_e вероятность ошибки декодирования. Тогда из утверждения 2 следует $P_e = \Pr\{\overline{e} \in A\}$. Далее для краткости обозначений вместо $\Pr\{\overline{e} \in A\}$ будем писать $\Pr\{A\}$. Для множества B будем делать аналогично.

Графическое представление множеств событий A и B изображено на рис. 3.

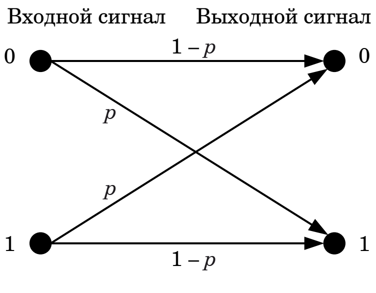

Рис. 2. Модель двоично-симметричного канала

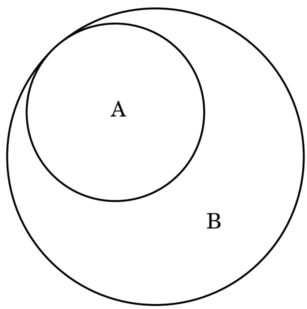

Рис. 3. Схематическое представление соотношения между множествами A и B: A – множество векторов ошибок при которых произошла ошибка декодирования; B – множество векторов ошибок с $w(\overline{e}) \geq d$

Мощность множества B не меньше, чем мощность множества A, где под мощностью множества подразумевается число элементов (векторов ошибок), которое входит во множество. Из этого следует, что $\Pr\{A\}<\Pr\{B\}$. Следовательно, значение для $\Pr\{B\}$ является верхней оценкой для $\Pr\{A\}$. Найдем $\Pr\{B\}$ для ДСК при вероятности ошибки p:

$$\Pr\{B\} = \Pr\{w(\overline{e}) = d \cup w(\overline{e}) = (d+1) \cup \ldots \cup w(\overline{e}) = n\} =$$
$$= \sum_{i=d}^{n} \Pr\{w(\overline{e}) = i\} = \sum_{i=d}^{n} C_n^i p^i (1-p)^{(n-i)} \quad .$$

Тогда верхнюю границу ошибки декодирования P_e^+ можно определить, как вероятность того, что вектор ошибки принадлежит множеству B:

$$P_e^+ = Pr\{B\} = \sum_{i=d}^{n} C_n^i p^i (1-p)^{(n-i)}. \tag{1}$$

Программа для расчета верхней границы ошибки декодирования по формуле (1) может быть кратко описана в виде следующего псевдокода:

Вход: $g(x)$ – порождающий многочлен; k – длина кодируемой последовательности; p – вероятность ошибки в ДСК
Выход: P_e^+ – вероятность ошибки декодирования
/*Вычисление длины кодового слова*/
$n \leftarrow k + \deg(g(x))$
$d\leftarrow$ вычисление минимального расстояния кода
$P_e^+ \leftarrow 0$
/*Цикл по весам векторов от d до n с вычислением элементов суммы из формулы (1)*/
for $i=d \ldots n$:
  $P_e^+ \leftarrow P_e^+ + C_n^i p^i (1-p)^{(n-i)}$
endfor

Стоит отметить, что:

$$\sum_{i=0}^{n} C_n^i p^i (1-p)^{(n-i)} = 1.$$

Так как d обычно много меньше, чем n, то для уменьшения сложности расчета выражение (1) можно переписать следующим образом:

$$P_e^+ = 1 - \sum\nolimits_{i=0}^{d-1} C_n^i p^i (1-p)^{(n-i)}. \tag{2}$$

Программа для расчета верхней границы ошибки декодирования по формуле (2) может быть кратко описана в виде следующего псевдокода:

Вход: $g(x)$ – порождающий многочлен; k – длина кодируемой последовательности; p – вероятность ошибки в ДСК
Выход: P_e^+ – вероятность ошибки декодирования
/*Вычисление длины кодового слова*/
$n \leftarrow k + \deg(g(x))$
$d \leftarrow$ вычисление минимального расстояния кода
$P_e^+ \leftarrow 1$
/*Цикл по весам векторов от 0 до $d - 1$ для расчета по формуле (2)*/
for $i = 0 \ldots d - 1$:
  $P_e^+ \leftarrow P_e^+ - C_n^i p^i (1-p)^{(n-i)}$
endfor

1.1.4. Вычисление точного значения ошибки декодирования

Пусть заданы: порождающий многочлен $g(x)$, длина кодируемой последовательности k, минимальное расстояние кода d, вероятность ошибки в канале p. Необходимо найти точное значение вероятности ошибки декодирования P_e.

Для решения этой задачи рассмотрим, каким образом с помощью кодера множество сообщений $\bar{m}$ может быть отображено на множество кодовых слов $\bar{a}$. Для этого все возможные сообщения $\bar{m}_i$, где $i \in \{0, 2^k - 1\}$ поочередно подаются на вход кодера. На выходе кодера получаем список кодовых слов $\bar{a}_i$ длиной n бит, где $i \in \{0, 2^k - 1\}$. Схематично этот процесс представлен на рис. 4.

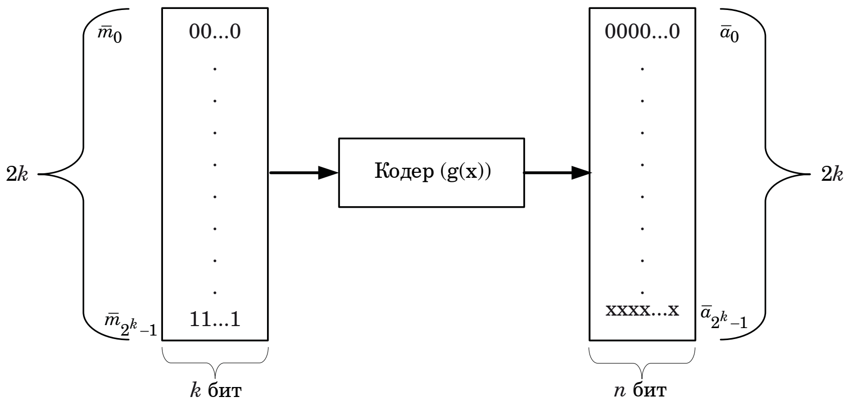

Рис. 4. Схематичное представление формирования множества кодовых слов

$$
\begin{array}{l}
A_0=1\\
A_1=0\\
\dots\\
A_{d-1}=0\\
\left.\begin{array}{l}A_d>0\\ A_{d+1}\\ \dots\\ A_n\end{array}\right\}=?
\end{array}
$$

Рис. 5. Допустимые значения A_i для кода с минимальным расстоянием d

Пусть A – множество кодовых слов, $|A|=2^k$; B – множество всех векторов ошибок, $|B|=2^n$.

Обозначим через A_i число кодовых слов веса i, где i – индекс от 0 до n. На рис. 5 показано, какие допустимые значения могут принимать A_i для кода с минимальным расстоянием d. Список значений A_i также называют весовым спектром кода.

Ошибка декодирования происходит в том случае, если вектор ошибок является кодовым словом (см. утверждение 2). Все слова в коде имеют вес больше либо равный d. Поэтому гарантированно обнаруживаются ошибки, число которых $<d$ (см. утверждение 3). Для точного определения значения вероятности ошибки декодирования следует посчитать вероятность попадания вектора ошибок во множество A, то есть $P_e = \Pr\{A\} = \Pr\{\overline{e} \in A : \overline{e} \neq 0\}$. Теперь мы можем записать следующее выражение для вычисления точного значения вероятности ошибки декодирования:

$$P_e = \sum_{i=d}^{n} A_i p^i (1-p)^{(n-i)}, \tag{3}$$

где P_e – вероятность ошибки декодирования, d – минимальное расстояние кода, n – длина кодового слова, p – вероятность ошибки на бит, A_i – количество кодовых слов с весом i.

Программа для расчета точного значения вероятности ошибки декодирования по формуле (3) может быть кратко описана в виде следующего псевдокода:

Вход: $g(x)$ – порождающий многочлен; k – длина кодируемой последовательности; p – вероятность ошибки в ДСК
Выход: P_e – вероятность ошибки декодирования
/*Вычисление длины кодового слова*/
$n \leftarrow k+\deg(g(x))$
/*Инициализация начального значения расстояния кода*/
$d \leftarrow n$
/*Инициализация начального значения для точного значения вероятности ошибки*/
$P_e \leftarrow 0$
/*Создаем массив счетчиков для весов кодовых слов длиной $n+1$*/
$\overline{A} \leftarrow \{0,0,\dots 0\}$

```
/*Известно, что сообщение из всех нулей дает кодовое слово из всех нулей, поэтому его можно пропустить и сразу обновить массив весов Ā */
A_0 ← A_0 + 1
/*Цикл по всем возможным сообщениям m̄_i, где i ∈ {0,2^k − 1} за исключением нулевого кодового слова, состоящего из всех нулей*/
for i=1 … 2^k − 1:
  ā_i ← Формирование кодового слова для m̄_i по алгоритму
  A_w(ā_i) ← A_w(ā_i) + 1
  if w(ā_i) < d
    /*Обновляем значение для минимального расстояния кода d */
    d ← w(ā_i)
  endif
endfor
/*Цикл для вычисления точной вероятности ошибки декодирования для расчета по формуле (3) с использованием полученных на предыдущих шагах значений:*/
/* d – минимальное расстояние кода*/
/* Ā – весовой спектр кода, Ā = {1,0,...,0,A_d,A_(d+1),...,A_n} */
for i=d … n:
  P_e ← P_e + A^i p^i (1 − p)^(n − i)
endfor
```

Рассмотрим следующий пример. Пусть задано: $g(x)=x^4+x^3+x+1$, $k=2$.

Список всех возможных сообщений на входе кодера состоит из 4 двоичных векторов и имеет следующий вид:

$$\bar{m}_0 = (00)$$
$$\bar{m}_1 = (01)$$
$$\bar{m}_2 = (10)$$
$$\bar{m}_3 = (11)$$

Вычислим длину кодового слова:

$$n = k + \deg(g(x)) = 2 + 4 = 6.$$

Подав сообщения на вход кодера (выполнив процедуру кодирования для каждого сообщения) получим следующий список кодовых слов:

$$\bar{a}_0 = (000000)$$
$$\bar{a}_1 = (011011)$$
$$\bar{a}_2 = (101101)$$
$$\bar{a}_3 = (110110)$$

Определим веса кодовых слов:

$$w(\bar{a}_0) = w(000000) = 0$$
$$w(\bar{a}_1) = w(011011) = 4$$
$$w(\bar{a}_2) = w(101101) = 4$$
$$w(\bar{a}_3) = w(110110) = 4$$

Из списка весов кодовых слов следует, что минимальное расстояние кода d=4. Укажем список значений A_i (весовой спектр кода):

$$A_0 = 1;$$
$$A_1 = 0;$$
$$A_2 = 0;$$
$$A_3 = 0;$$
$$A_4 = 3;$$
$$A_5 = 0;$$
$$A_6 = 0.$$

Отметим, что $\sum_{i=0}^{n} A_i = 2^k$.

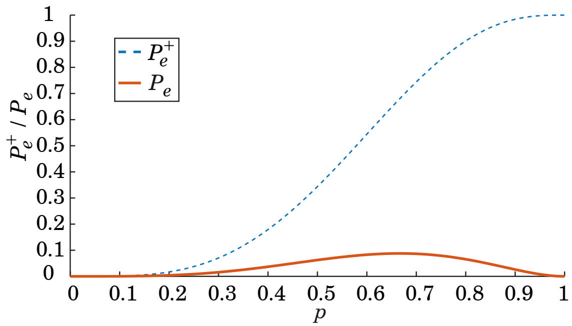

Рис. 6. Верхняя граница ошибки декодирования P_e^+ и точное значение ошибки декодирования P_e для $g(x)=x^4+x^3+x+1$ при k=2

На рис. 6 представлены результаты расчета верхней границы ошибки декодирования P_e^+ и точного значения ошибки декодирования P_e для рассматриваемого примера.

С точки зрения теории информации и помехоустойчивого кодирования вероятность ошибки в канале ДСК не следует рассматривать выше 0.5, так как если в реальном канале вероятность ошибки превышает это значение, то это означает, что выход канала можно инвертировать и тем самым уменьшить вероятность ошибки. Однако, при исследовании значений вероятности ошибки декодирования будем рассматривать все возможное значения вероятности ошибки $p \in [0,1]$.

1.1.5. Оценка вероятности ошибки декодирования с помощью имитационного моделирования

Пусть задан порождающий многочлен $g(x)$, длина кодируемой последовательности k. Необходимо оценить вероятность ошибки декодирования $\widehat{P_e}$ с заданной точностью ε при использовании имитационного моделирования. Для этого:

1. Генерируется случайное сообщение $\overline{m}$.

2. К сообщению $\overline{m}$ добавляется контрольная сумма по алгоритму, описанному в пункте 1.1.2. В результате формируется кодовое слово $\overline{a}$.

3. Генерируется случайный вектор ошибок $\overline{e}$, где для каждой позиции случайно выбирается событие:

а) произошла ошибка с вероятностью p (бит вектора ошибок=1);

b) ошибки не было с вероятностью $(1-p)$ (бит вектора ошибок=0).

4. Вектор ошибки $\overline{e}$ складывается с результатом кодирования $\overline{a}$ (выполняется операция XOR) для получения выхода канала $\overline{b}$.

5. Вычисляется синдром $s(x)$, и если $s(x)=0$, а $\overline{e} \neq 0$, то счетчик ошибок декодирования N_e увеличивается на единицу.

6. Шаги 1) – 4) повторяются N раз. Затем, если в результате моделирования N_e раз произошли ошибки декодирования $(\overline{e} \neq 0, E = 0)$, то оценка вероятности ошибки декодирования вычисляется по следующей формуле $\widehat{P_e} = \dfrac{N_e}{N}$.

Пусть точность полученных результатов моделирования определяется как $\varepsilon = \left|P_e - \widehat{P_e}\right|$, где P_e – теоретическая вероятность декодирования, определяемая по формуле (3).

Тогда для определения примерного количества необходимых итераций моделирования N при заранее заданной требуемой точности полученных результатов ε используется формула:

$$N = \frac{9}{4\varepsilon^2}.$$

Следует отметить, что данный подход для определения числа экспериментов N можно использовать только когда известно, что эксперименты, проводимые при имитационном моделировании, независимы друг от друга. В рамках данной лабораторной работы рассматривается канал без памяти, следовательно, эксперименты независимы друг от друга.

Программа для получения оценки вероятности ошибки декодирования с помощью имитационного моделирования может быть кратко описана в виде следующего псевдокода:

Вход: $g(x)$ – порождающий многочлен; k – длина кодируемой последовательности; p – вероятность ошибки в ДСК, ε – требуемая точность полученных результатов

Выход: $\widehat{P_e}$ – оценка вероятности ошибки декодирования, полученная с помощью имитационного моделирования

$\widehat{P_e} \leftarrow 0$

/*Счетчик произошедших ошибок декодирования*/

$N_e \leftarrow 0$

/*Вычисление длины кодового слова*/

$n \leftarrow k + \deg(g(x))$

/*Вычисление необходимого числа экспериментов N*/

$N \leftarrow \frac{9}{4\varepsilon^2}$

/*Цикл по количеству проводимых экспериментов N*/

for $i = 1 \ldots N$:

/*Генерация случайного сообщения $\overline{m}$, где $\overline{m} \in \{(0...00),(0...01),...,(1...10),(1...11)\}$.Это эквивалентно равновероятному выбору случайного сообщения $\overline{m}$ из множества сообщений*/

$\overline{m} \leftarrow \mathrm{rand}\left(\{(0...00),(0...01),...,(1...10),(1...11)\}\right)$

$\overline{a}$ ← Формирование кодового слова для $\overline{m}$ по алгоритму

$\overline{e} \leftarrow (0,0,\ldots,0)$

/*Цикл для генерации случайного вектора ошибок $\overline{e}$ из нулей и единиц длиной n. Вероятность появления единицы равна p, а вероятность появления нуля $(1-p)$, соответственно.

for $j = 1 \ldots n$:

/*Генерация случайной величины q, имеющей равномерное распределение на интервале $[0,1]$.

```
    q←rand([0,1])
    if q<p:
      e_j←1
    endif
  /*Вычисление выхода канала b̄
  b̄ ← ā ⊕ ē
  /*Вычисление синдрома s(x)
  s(x)←b(x)modg(x)
  if (s(x)=0)&(e(x)≠0)
    /*Наращиваем счетчик ошибок декодирования*/
    N_e← N_e+1
  Endif
endfor
```

$$\widehat{P_e} \leftarrow \frac{N_e}{N}$$

1.2. Вопросы для допуска к лабораторной работе

Каждый студент получает от преподавателя исходные данные, которые необходимы для ответов на вопросы для допуска к лабораторной работе. Исходные данные представлены в 4 вариантах:

Вариант	Исходные данные
1	$g(x)=x^3+x+1$, $d=3$, $k=4$, $n=7$
2	$g(x)=x^3+x^2+1$, $d=3$, $k=4$, $n=7$
3	$g(x)=(x^3+x+1)(x+1)$, $d=4$, $k=3$, $n=7$
4	$g(x)=(x^3+x^2+1)(x+1)$, $d=4$, $k=3$, $n=7$

где $g(x)$ – порождающий многочлен; d – минимальное расстояние кода; k – число информационных символов; n – длина кодового слова.

Получив вариант с исходными данными от преподавателя и информационную последовательность $\overline{m}$ из k символов необходимо выполнить следующие задания:

1. С помощью алгоритма формирования контрольной суммы по заданной последовательности $\overline{m}$ сформировать контрольную сумму и кодовое слово $\overline{a}$, передаваемое в канал. Показать, что на этом кодовом слове декодер работает корректно, то есть синдром $s(x)$ равен 0 (ошибки не обнаружены).

2. Привести пример работы декодера для случая, когда в канале произошло t ошибок при этом $1<t\leq d-1$.

3. Подобрать t ошибок так, чтобы эти ошибки не обнаруживались при этом $t>d-1$.

4. Подобрать t ошибок так, чтобы эти ошибки обнаруживались при этом $t>d-1$.

5. Рассмотреть следующий пример «некорректного» использования системы кодирования. Пусть пользователь для последовательности из l символов (при этом $l>k$) применяет типовой алгоритм формирования контрольной суммы. При этом длина кодового слова будет больше, чем n. Показать на конкретном примере, что возможен случай, когда в канале произойдет меньше, чем d ошибок и такие ошибки не будут обнаружены.

1.3. Порядок выполнения лабораторной работы

1. Необходимо в качестве допусков к лабораторной работе ответить на вопросы в подразделе 1.2 и сдать их преподавателю.
2. После сдачи всех допусков получить от преподавателя вариант задания для лабораторной работы.
3. Написать и отладить программу имитационного моделирования в соответствии с полученным вариантом.
4. Продемонстрировать работу имитационного моделирования программы преподавателю.
5. Получить дополнительное задание для исследования.
6. В соответствии с вариантом и полученным дополнительным заданием оформить и сдать отчет.

1.4. Варианты заданий для лабораторной работы

Вариант 1	
Основное задание	Требуется разработать программу, наглядно демонстрирующую работу кодера и декодера для типового алгоритма формирования циклических кодов. На вход программы подается: • порождающий многочлен $g(x)$ (может, как совпадать с многочленами из допуска, так и быть указан преподавателем при сдаче основного задания); • информационная последовательность из l бит (l может быть как меньше, так и больше значений k из исходных данных допуска); • вектор ошибки $\overline{e}$ (программа должна учитывать размер кодового слова и дополнять вектор ошибок до нужной длины, например, нулями в левой части).

Вариант 1	
	На основе входных данных должно быть сформировано кодовое слово $\overline{a}$. Затем с учетом введенного вектора ошибок $\overline{e}$ формируется последовательность на выходе канала $\overline{b}$. По принятой последовательности $\overline{b}$ вычисляется синдром $s(x)$. По синдрому принимается решение о наличии или отсутствии ошибок в канале. В программе должна быть предусмотрена возможность вывода всех промежуточных значений, которые формируются как при работе кодера, так и декодера
Список дополнительных заданий	a) Исследовать альтернативную реализацию алгоритма декодирования. Последовательность $\overline{b}$ на выходе канала (см. рис. 7) делится на две части. Первая последовательность $\overline{m}_b$ содержит в себе символы, относящиеся к информационной части. Вторая $\overline{c}_b$ содержит в себе символы, относящиеся к контрольной сумме. Последовательность $\overline{m}_b$ вновь подается на вход кодера, в результате чего вычисляется контрольная сумма $\overline{c}_b{}'$. Если $\overline{c}_b \neq \overline{c}_b{}'$, то принимается решение о наличии ошибок при передаче. Выяснить, возможны ли следующие ситуации: • типовой алгоритм декодирования обнаруживает ошибки, а альтернативный не обнаруживает; • типовой алгоритм декодирования не обнаруживает ошибки, а альтернативный обнаруживает. Если такие ситуации существуют, привести примеры, в противном случае обосновать невозможность их возникновения
	b) Исследовать работу кодера/декодера для $g(x)$ из списка в задании на допуск с параметрами n, k и d, когда выбрано «некорректное» число информационных символов l. То есть $l>k$. Для заданного преподавателем значения l, указать все возможные вектора ошибок, для которых $w(\overline{e}) \leq (d-1)$ и ошибки при этом не обнаруживаются.
	c) Пусть $\varphi(x)=x^3+x+1$ или $\varphi(x)=x^3+x^2+1$, а порождающий многочлен $g(x)= \varphi(x)(x+1)$. Привести примеры, когда не обнаруживается нечетное число ошибок (вес вектора ошибок $\overline{e}$ не четный). Если таких примеров нет, то обосновать почему

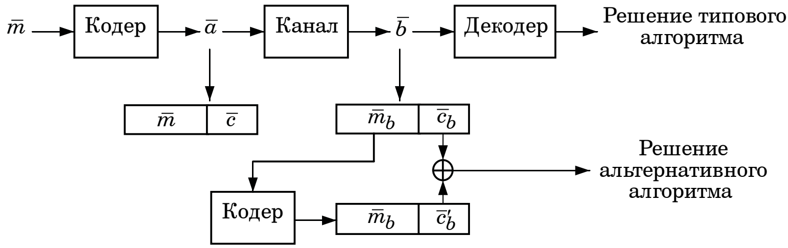

Рис. 7. Схема альтернативного алгоритма декодирования

Вариант 2	
Основное задание	Разработать программу, с помощью которой путем имитационного моделирования оценивается вероятность ошибки декодирования при передаче данных по двоично-симметричному каналу (см. пункт 1.1.5). Исходными данными для работы программы являются: • порождающий многочлен $g(x)$ (может, как совпадать с многочленами из допуска, так и быть указан преподавателем при сдаче основного задания); • значение длины информационной последовательности l (l может быть как меньше, так и больше значений k из исходных данных допуска); • точность ε, с которой программа оценивает вероятность ошибки декодирования. С помощью программы студент должен исследовать зависимость вероятности ошибки декодирования от значения вероятности появления ошибки в канале p при различных значениях l
Список дополнительных заданий	a) Построить графики зависимости оценки вероятности ошибки декодирования $\widehat{P_e}$ от вероятности ошибки в двоично-симметричном канале p при $l<k$, $l=k$ и $l=k$. Обосновать полученные зависимости
	b) В исходном варианте программы моделирования входная последовательность генерируется случайным образом. Необходимо провести моделирование для случая, когда входная последовательность зафиксирована. Сравнить полученную зависимость оценки вероятности ошибки декодирования $\widehat{P_e}$ от вероятности ошибки в двоично-симметричном канале p с исходным вариантом. Повторить моделирование для другой входной последовательности. Выяснить, каким образом влияет изменение входной последовательности на оценку вероятности ошибки декодирования

Вариант 2	
	c) Исследовать, как изменится зависимость оценки вероятности ошибки декодирования $\widehat{P_e}$ от вероятности ошибки в канале p при смене модели канала на модель, показанную на рис. 8. Обосновать полученный результат
	d) Модифицировать программу для возможности применения альтернативного алгоритма декодирования, описанного в дополнительном задании a) для первого варианта (рис. 7). Сравнить работу альтернативного алгоритма декодирования с типовым вариантом декодирования (построить графики зависимости оценки ошибки декодирования $\widehat{P_e}$ от вероятности ошибки в канале p). Выяснить, влияет ли изменение алгоритма декодирования на вероятность ошибки декодирования

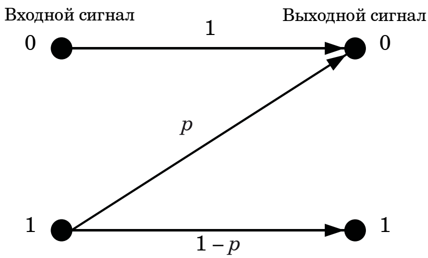

Рис. 8. Измененная модель канала

Вариант 3	
Основное задание	Разработать программу вычисления верхней оценки для вероятности ошибки декодирования (см. пункт 1.1.3) и вычисления точного значения вероятности ошибки декодирования (см. пункт 1.1.4). Исходными данными для работы программы являются: • порождающий многочлен $g(x)$ (может, как совпадать с многочленами из допуска, так и быть указан преподавателем при сдаче основного задания); • значение длины информационной последовательности l (l может быть как меньше, так и больше значений k из исходных данных допуска); С помощью программы нужно исследовать зависимость верхней оценки и точного значения вероятности ошибки декодирования от вероятности появления ошибки в канале p при различных значениях l

Вариант 3	
Список дополнительных заданий	a) Построить зависимости верхней оценки вероятности ошибки декодирования P_e^+ и точной вероятности ошибки декодирования P_e при различных порождающих многочленах $g(x)$. Исследовать, как влияет изменение порождающего многочлена на верхнюю оценку и точное значение вероятности ошибки декодирования
	b) Построить зависимости верхней оценки вероятности ошибки декодирования P_e^+ и точной вероятности ошибки декодирования P_e при $l<k$, $l=k$ и $l=k$. Обосновать полученные зависимости
	c) Построить зависимости верхней оценки вероятности ошибки декодирования P_e^+ и точной вероятности ошибки декодирования P_e при различных значениях l и фиксированной вероятности ошибки в ДСК канале $p=0.1$, $p=0.2$, $p=0.3$. Обосновать полученные зависимости

1.5. Требования к отчету по лабораторной работе

Отчет должен содержать:

1. Титульный лист.
2. Цель и постановку задачи.
3. Описание моделируемой системы.
4. Описание проводимого исследования и рассматриваемого алгоритма.
5. Описание программы имитационного моделирования в виде псевдокода или блок-схемы.
6. Описание результатов проводимых исследований и зависимостей.
7. Выводы по проводимым исследованиям.
8. Листинг программы имитационного моделирования присутствует только в электронной версии отчета.

2. МЕТОДЫ ПЕРЕДАЧИ ДАННЫХ НА КАНАЛЬНОМ УРОВНЕ В СИСТЕМАХ С ОБРАТНОЙ СВЯЗЬЮ

Раздел посвящен исследованию типовых алгоритмов передачи данных в системах с решающей обратной связью, а также использованию численного расчета и имитационного моделирования для оценки коэффициента использования канала в рамках базовой модели для систем с решающей обратной связью.

2.1. Модель системы с обратной связью и алгоритмы передачи

2.1.1. Базовая модель систем с обратной связью

Канал передачи от источника к получателю называется прямым каналом. 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 приемника к источнику называется обратным каналом. По каналу обратной связи передаются квитанции:

- Положительная квитанция (обычно обозначается как ACK, а в рамках курса как «+»), если приемник не обнаружил наличие ошибок в принятом сообщении;
- Отрицательная квитанция (обычно обозначается как NACK, а в рамках курса как «–»), если приемник обнаружил наличие ошибок в принятом сообщении.

На рис. 9 представлена базовая модель передачи данных по каналу с обратной связью, где И – источник сообщений, П – приемник, КС – контрольная сумма для передаваемых данных.

Рис. 9. Схема передачи данных по каналу с обратной связью

Введем следующие допущения:

Допущение 1. При передаче данных в прямом канале связи с вероятностью p могут возникнуть ошибки. Если при передаче произошли ошибки, то приемник всегда их обнаруживает за счет контрольной суммы.

Допущение 2. Приемник проверяет полученное сообщение 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.

Допущение 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).

Допущение 4. Все сообщения, которые передает источник, имеют одинаковую длину. Время передачи сообщения принято за единицу времени, а время передачи квитанции считается равным нулю. Источник получает квитанцию о результате передачи через τ единиц времени после окончания передачи сообщения, где τ – целое число. Пример работы системы в соответствии с допущением представлен на рис. 10.

Допущение 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 систе-

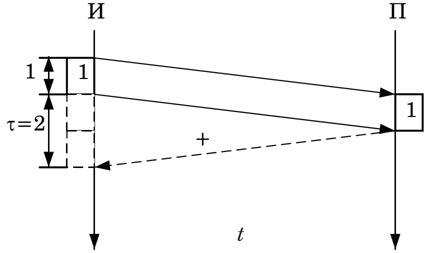

Рис. 10. Пример работы системы в соответствии с допущением 4 при τ=2

мы является коэффициент использования канала, который определяется следующим образом:

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

где π – коэффициент использования канала, T – интервал работы системы, $N(T)$ – число сообщений, переданных за интервал времени T.

2.1.2. Алгоритм с ожиданием

Алгоритм с ожиданием работает следующим образом:

• Источник передает сообщение, затем ждет квитанцию с подтверждением. Предполагается, что время ожидания квитанции (задержка получения квитанции) постоянно и равно τ единицам времени, согласно допущению 4 (см. пункт 2.1.1).

• Если источник принимает отрицательную квитанцию, то он повторяет передачу сообщения.

• Если получена положительная квитанция, то передается следующее сообщение.

Пример работы алгоритма с ожиданием при τ=2 приведен на рис. 11.

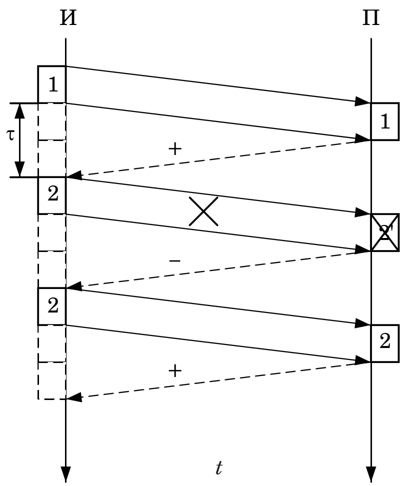

Рис. 11. Пример работы алгоритма с ожиданием при τ=2

Будет рассматриваться два способа работы данного алгоритма:

1. Организация гарантированной доставки сообщения, то есть вероятность доставки сообщения приемнику равна единице. Для этого источник повторяет передачу сообщения до тех пор, пока не получит положительную квитанцию.

2. Передача без гарантированной доставки сообщения. Источник повторяет передачу сообщения до тех пор, пока не получит n отрицательных квитанций. Получив n отрицательных квитанций, источник прекращает работу с данным сообщением и переходит к передаче следующего сообщения.

Алгоритм с ожиданием имеет низкий коэффициент использования канала даже при малых значениях вероятности ошибки и определяется по формуле

$$\eta(\tau, p) = \frac{1-p}{1+\tau}.$$

2.1.3. Алгоритм с возвратом

Алгоритм с возвратом работает следующим образом:

• Источник непрерывно передает сообщения, не дожидаясь квитанции на отправленное ранее сообщение. При задержке получения

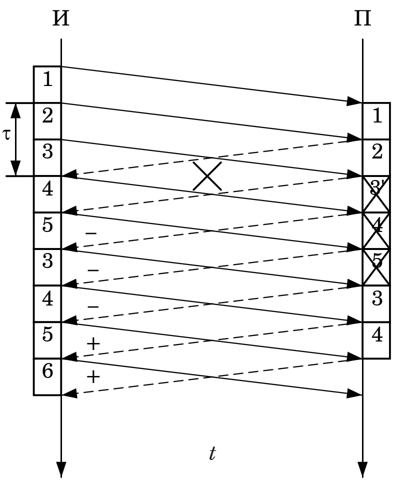

Рис. 12. Пример работы алгоритма с возвратом при τ=2

квитанции равной τ единицам времени, после передачи сообщения, абонент успевает передать еще τ сообщений до того, как получит квитанцию на него.

• При получении отрицательной квитанции на сообщение, источник повторяет передачу этого сообщения и всех следующих за ним сообщений, которые он успел передать за это время. Приемник после отправки отрицательной квитанции удаляет τ пришедших за ним сообщений, даже если они были приняты без ошибок. Предполагается, что источнику и приемнику известна задержка в получении квитанции τ и определено в соответствии с допущением 4.

Пример работы алгоритма с возвратом при $\tau=2$ приведен на рис. 12.

Коэффициент использования канала, при любых значениях вероятности ошибки в канале, для алгоритма с возвратом выше, чем для алгоритма с ожиданием и определяется по формуле:

$$\eta(\tau, p) = \frac{1-p}{1+p\tau}.$$

2.1.4. «Алгоритм» с селективным повторением

Под «алгоритмом» с селективным повторением подразумевается не конкретный алгоритм, а некоторая идея, которая может быть реализована различными способами.

В рассмотренном ранее алгоритме с возвратом, источник в некоторых случаях повторно передавал сообщения, которые были приняты без ошибок. Что бы избавиться от данного недостатка, на приемной стороне можно использовать буфер для хранения верно принятых сообщений и повторно передавать только сообщения, которые были приняты с ошибками или не поместились в буфер. Поэтому алгоритм с селективным повторением работает следующим образом:

• Источник непрерывно передает сообщения, не дожидаясь квитанции на отправленное ранее сообщение. При задержке получения квитанции равной τ единицам времени, после передачи сообщения, абонент успевает передать еще τ сообщений, до того как получит квитанцию на него.

• При получении отрицательной квитанции на сообщение, источник повторяет передачу этого сообщения. Приемник после отправки отрицательной квитанции сохраняет принятые без ошибок последующие сообщения в буфер. При получении цепочки последовательно идущих сообщений на приемнике, все они передаются на вышележащий уровень модели взаимодействия открытых систем (OSI). Если в буфере нет места для принятого верно сообщения, то оно удаляется, а по обратному каналу передается отрицательная квитанция.

Пример реализации этой идеи при $\tau=2$ и длине буфера равной 3 приведен на рис. 13.

При данном методе передачи возможны различные подходы к работе с буфером. Рассмотрим два подхода на примере системы с $\tau=2$ и размером буфера $L=1$:

1. В буфер записывается любое принятое без ошибок сообщение, независимо от его порядкового номера. Сообщения выдаются пользователю только в соответствии с правильным порядком следования, определяемого нумерацией на источнике. Пример работы такой системы приведен на рис. 14.

2. В буфер записывается только сообщение с номером $i+1$, при ожидании сообщения с номером i. Сообщения выдаются пользователю только в соответствии с правильным порядком следования, определяемого нумерацией на источнике. Пример работы такой системы приведен на рис. 15.

Любое из описанных решений ограничено коэффициентом использования канала равным $\pi=(1-p)$, и достигает данного значения при бесконечном размере буфера ($L\to\infty$).

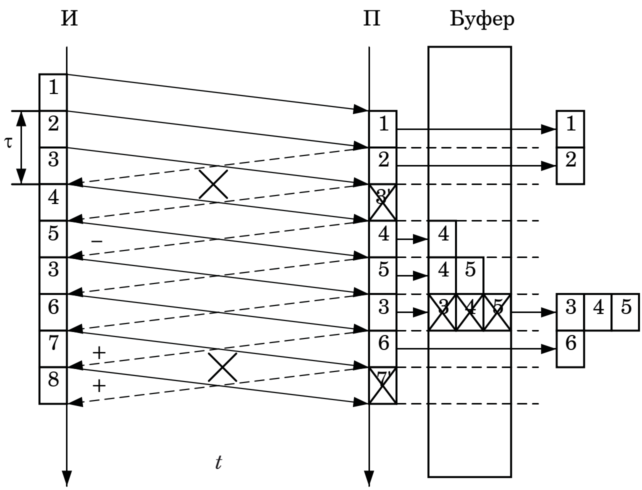

Рис. 13. Пример работы алгоритма с селективным повторением при τ=2

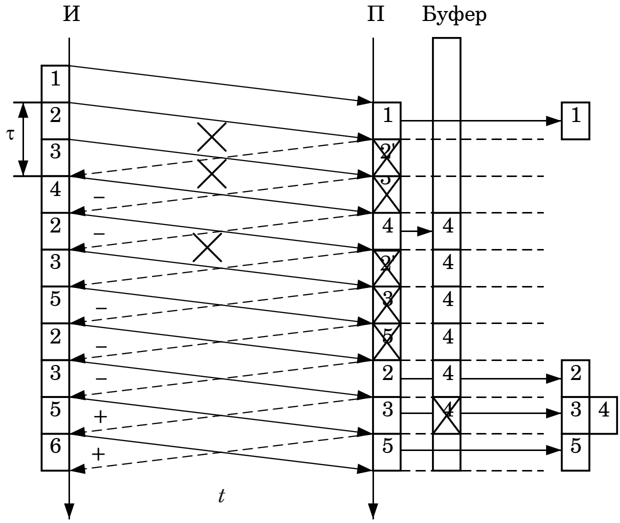

Рис. 14. Работа «алгоритма» с селективным повторением при τ=2 и L=1 для первой стратегии

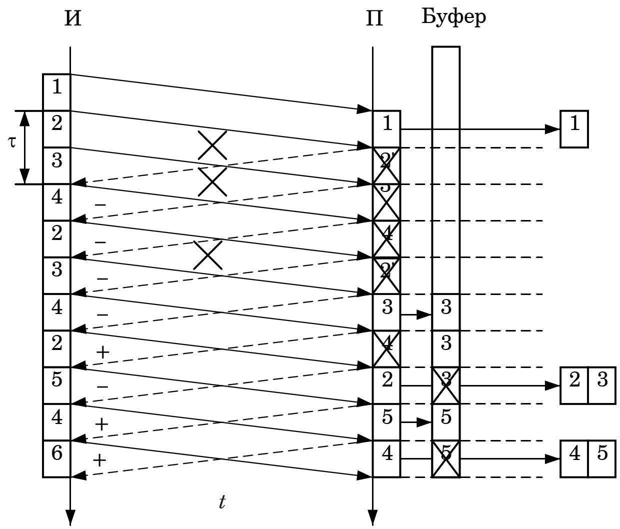

Рис. 15. Работа «алгоритма» с селективным повторением при τ=2 и L=1 для второй стратегии

2.1.5. Алгоритм с виртуальными каналами

Предполагается, что на источнике есть непрерывный поток сообщений, и задержка в получении квитанции равна τ единиц времени (см. допущение 4 из пункта 2.1.1).

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 модели взаимодействия открытых систем (OSI) из буфера в соответствии с требуемым порядком.

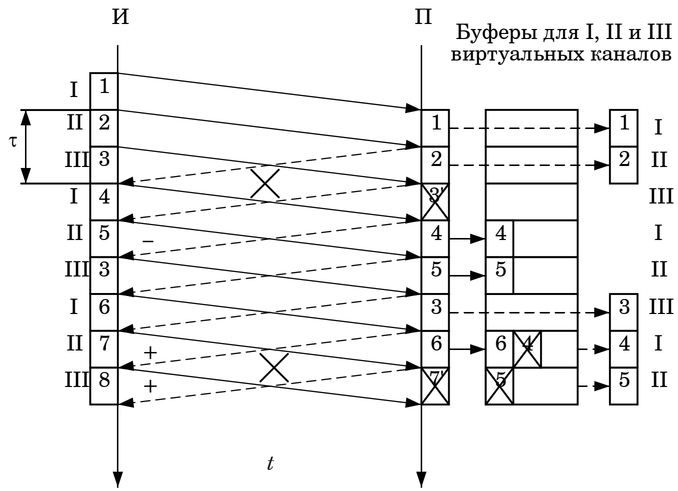

Рис. 16. Пример работы алгоритма с виртуальными каналами при τ=2 и тремя виртуальными каналами

Пример работы алгоритма с виртуальными каналами при $\tau=2$ (при этом виртуальных каналов три) приведен на рис. 16.

Коэффициент использования канала для алгоритма с виртуальными каналами при бесконечной длине буфера ($L \to \infty$) определяется по формуле:

$$\eta(p) = 1 - p.$$

Стоит заметить, что коэффициент использования канала (π) в данном случае не зависит от задержки в получении квитанции (τ).

2.1.6. Алгоритм для передачи по каналу с высокой вероятностью ошибки

Алгоритм для высокой вероятности ошибки в канале работает следующим образом:

• Источник непрерывно передает одно и тоже сообщение, не дожидаясь квитанции на отправленное ранее сообщение до тех пор, пока не получит положительную квитанцию. При задержке получения квитанции равной τ единиц времени, после передачи сообщения, абонент успевает передать еще τ его копий, до того как получит соответствующую квитанцию.

• При получении положительной квитанции источник переходит к следующему сообщению. Дубликаты сообщения, после отправки

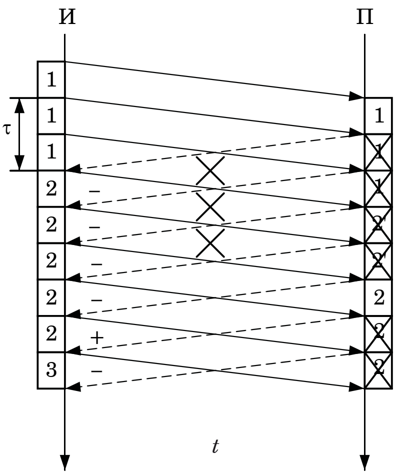

Рис. 17. Пример работы алгоритма для высокой вероятности ошибки при $\tau=2$

положительной квитанции, в количестве τ штук на приемнике удаляются независимо от успешности приема.

Пример работы алгоритма для высокой вероятности ошибки при $\tau=2$ приведен на рис. 17.

Коэффициент использования канала, при больших значениях вероятности ошибки в канале, алгоритма для высокой вероятности ошибки выше, чем для алгоритма с возвратом и определяется по формуле:

$$\eta(\tau, p) = \frac{1-p}{1+(1-p)\tau}.$$

2.2. Вопросы для допуска к лабораторной работе

2.2.1. Нахождение среднего числа передач в алгоритме с ожиданием при неограниченном числе повторных передач

Рассматривается алгоритм с ожиданием для системы с $p_{обр}=0$ и неограниченном числе повторных передач сообщения, до получения положительной квитанции.

Необходимо:

• Написать программу имитационного моделирования для алгоритма с ожиданием, когда $p_{обр}=0$ и число повторных передач сообщения до получения положительной квитанции не ограниченно.

• Самостоятельно вывести формулу для теоретического расчета среднего числа повторных передач до получения положительной квитанции, как функцию от вероятности ошибки в прямом канале – $\bar{N}(p)$. Для вывода формулы использовать следующие рассуждения. Введем в рассмотрение случайную величину N – число передач сообщения до получения положительной квитанции. Для данной системы можно выписать следующий набор вероятностей:

$$\Pr\{N=1\}=1-p;$$
$$\Pr\{N=2\}=p(1-p);$$
$$\ldots$$
$$\Pr\{N=i\}=p^{(i-1)}(1-p);$$
$$\ldots$$

Тогда $\bar{N}(p)$ можно определить следующим образом:

$$\bar{N}(p) = \mathrm{M}[N] = \sum_{i=1}^{\infty} iPr\{N=i\}.$$

Используя свойства бесконечных рядов и математические преобразования, необходимо получить формулу в явном виде (то есть в формуле должно отсутствовать суммирование с бесконечным числом слагаемых).

• Результаты моделирования сравнить с теоретическим расчетом.

2.2.2. Нахождение среднего числа передач в алгоритме с ожиданием при ограниченном числе повторных передач

Рассматривается алгоритм с ожиданием для системы с $p_{обр}=0$ и ограничением на число попыток передачи сообщения до получения положительной квитанции равным n.

Необходимо:

• Написать программу имитационного моделирования алгоритма с ожиданием для системы с $p_{обр}=0$, а число попыток передачи сообщения до получения положительной квитанции ограниченно n.

• Самостоятельно вывести формулу для расчета среднего числа передач до получения положительной квитанции с учетом ограничения, как функцию от вероятности ошибки в прямом канале p и значения n – $\bar{N}(p,n)$. Для вывода формулы использовать следующие рассуждения. Введем в рассмотрение случайную величину N – число передач сообщения до получения положительной квитанции. Пусть число повторных передач ограничивается некоторым числом n, которое является параметром алгоритма. Для данной системы можно выписать следующий набор вероятностей:

$$\Pr\{N=1\}=1-p;$$
$$\Pr\{N=2\}=p(1-p);$$
$$\ldots$$
$$\Pr\{N=n\}=?;$$
$$\Pr\{N\geq n+1\}=0.$$

При самостоятельном выписывании выражений для вероятностей в данной системе, необходимо воспользоваться тем, что:

$$\sum_{i=1}^{n}\Pr\{N=i\}=1.$$

Используя свойства рядов и математические преобразования, получить формулу $\bar{N}(p,n)$ в явном виде (то есть в формуле должно отсутствовать суммирование с числом слагаемых, зависящим от параметра n).

• Результаты моделирования сравнить с теоретическим расчетом.

2.2.3. Нахождение среднего числа передач в алгоритме с ожиданием при наличии ошибок в обратном канале

Рассматривается алгоритм с ожиданием для системы, как без ограничения, так и с ограничением на число попыток передачи сообщения до получения положительной квитанции (n) при условии, что $p_{обр} \geq 0$.

Наличие ошибок в обратном канале говорит о том, что источник может получить искаженную квитанцию. При получении искаженной квитанции, источник воспринимает ее как отрицательную. При такой работе алгоритма возможно некорректное воспроизведение последовательностей сообщений у получателя. Одним из решений данной проблемы может быть следующий метод. Источник вместе с данными посылает номер сообщения. Если сообщение передано успешно, и его номер совпадает с номером предыдущего сообщения, то данное сообщение уничтожается и не передается на дальнейшую обработку, а по обратному каналу отправляется положительная квитанция.

Необходимо:

• Написать программы имитационного моделирования для алгоритма с ожиданием при неограниченном и ограниченном числе повторных передач для системы с $p_{обр} \geq 0$.

• Самостоятельно вывести формулы для среднего числа повторных передач, как в случае с неограниченным числом повторных передач, так и с ограниченным числом передач (аналогично пунктам 2.1 и 2.2).

В случае неограниченного числа передач, необходимо найти среднее число повторных передач, как функцию от вероятности ошибки в прямом канале p и вероятности ошибки в обратном канале $p_{обр}$ – $\bar{N}(p, p_{обр})$.

В случае ограниченного числа передач, необходимо найти среднее число повторных передач, как функцию от вероятности ошибки в прямом канале, вероятности ошибки в обратном канале и параметра n, определяющего ограничение на число повторных передач – $\bar{N}(p, p_{обр}, n)$.

Для рассматриваемой системы, как и раньше можно выписать следующий набор вероятностей (зависящий от наличия или отсутствия ограничений):

$$\Pr\{N=1\}=(1-p)(1-p_{обр});$$

$$\Pr\{N=2\}=?;$$

$$\dots$$

Для вывода формул необходимо выписать вероятности самостоятельно. Используя свойства бесконечных и конечных рядов, а также математические преобразования, необходимо получить формулы $\bar{N}(p, p_{обр})$ и $\bar{N}(p, p_{обр}, n)$ в явном виде (то есть в формулах должно отсутствовать суммирование с бесконечным или числом слагаемых, зависящим от параметра n).

• Результаты моделирования сравнить с теоретическим расчетом.

2.2.4. Моделирование алгоритма с ожиданием для определения коэффициента использования канала при τ>0

Рассматривается алгоритм с ожиданием для системы с $p_{обр}=0$ и неограниченном числе повторных передач сообщения, до получения положительной квитанции.

Необходимо:

• Написать программу имитационного моделирования для алгоритма с ожиданием для системы с $p_{обр}=0$ и неограниченном числе повторных передач сообщения. Программа должна быть написана таким образом, чтобы была возможность задавать любые значения задержки получения квитанции (τ) и вероятности ошибки в прямом канале (p).

• Для предварительной проверки правильности работы программы имитационного моделирования, убедиться, что значения коэффициента использования канала π, полученные моделированием, совпадают с формулой:

$$\eta(\tau, p) = \frac{1 - p}{1 + \tau}.$$

• Продумать и вывести лог работы системы в рамках программы имитационного моделирования, по которому нарисовать временную диаграмму работы алгоритма. Пример временной диаграммы представлен на рис. 11 в пункте 2.1.2.

2.2.5. Моделирование алгоритма с возвратом

Рассматривается алгоритм с возвратом для системы с $p_{обр}=0$ и неограниченном числе повторных передач сообщения, до получения положительной квитанции. Работа алгоритма с возвратом описана в пункте 2.1.3.

Необходимо:

• Написать программу имитационного моделирования для алгоритма с возвратом, (см. пункт 2.1.3) для системы с $p_{обр}=0$. Программа должна быть написана таким образом, чтобы была возможность задавать любые значения задержки получения квитанции (τ) и вероятности ошибки в прямом канале (p).

• Для предварительной проверки правильности работы программы имитационного моделирования, убедиться, что значения коэффициента использования канала π, полученные моделированием, совпадают с формулой:

$$\eta(\tau, p) = \frac{1-p}{1+p\tau}.$$

• Продумать и вывести лог работы системы в рамках имитационного моделирования программы, по которому нарисовать временную диаграмму работы алгоритма. Пример временной диаграммы представлен на рис. 12 в пункте 2.1.3.

$$\eta(\tau, p) = \frac{1-p}{1+p\tau}.$$

2.3. Порядок выполнения лабораторной работы

1. Необходимо в качестве допусков к лабораторной работе ответить на вопросы в подразделе 2.2 и сдать их преподавателю.

2. После сдачи всех допусков получить от преподавателя вариант задания для лабораторной работы.

3. Написать и отладить программу имитационного моделирования для алгоритма передачи данных по каналу с обратной связью (в соответствии с полученным вариантом).

4. Построить графики следующих зависимостей:

a) Коэффициента использования канала от вероятности ошибки в прямом канале при некоторых фиксированных значениях τ: $\pi(p,\tau=\text{const})$;

b) Коэффициента использования канала от величины задержки получения квитанции при некоторых фиксированных значениях p: $\pi(p=\text{const},\tau)$;

c) Среднего числа повторных передач от вероятности ошибки в канале при некоторых фиксированных значениях τ: $\bar{N}(p, \tau = \text{const})$;

d) Среднего числа повторных передач от величины задержки получения квитанции при некоторых фиксированных значениях p: $\bar{N}(p = \text{const}, \tau)$;

5. Продемонстрировать работу программы имитационного моделирования преподавателю.

6. Получить дополнительное задание для исследования.

7. В соответствии с вариантом и полученным дополнительным заданием оформить и сдать отчет.

2.4. Варианты заданий для лабораторной работы

Вариант	Задание
1	Написать программу имитационного моделирования для алгоритма с селективными повторениями, описанного в 2.1.4: • Размер буфера на приемнике равен 1 ($L=1$). • Стратегия работы с буфером: в буфер записывается любое принятое без ошибок сообщение, независимо от его порядкового номера. • Сообщения выдаются пользователю только в соответствии с правильным порядком следования, определяемого нумерацией на источнике. Получить графики зависимостей, указанных в подразделе 2.3
2	Написать программу имитационного моделирования для алгоритма с селективными повторениями, описанного в 2.1.4: • Размер буфера на приемнике равен 1 ($L=1$). • Стратегия работы с буфером: в буфер записывается только сообщение с номером $i+1$, при ожидании сообщения с номером i. • Сообщения выдаются пользователю только в соответствии с правильным порядком следования, определяемого нумерацией на источнике. Получить графики зависимостей, указанных в подразделе 2.3
3	Написать программу имитационного моделирования для алгоритма с селективными повторениями, описанного в 2.1.4: • Размер буфера на приемнике $L>1$. • Стратегия работы с буфером: в буфер записываются любые принятые без ошибок сообщения, независимо от их порядкового номера. • Сообщения выдаются пользователю только в соответствии с правильным порядком следования, определяемого нумерацией на источнике. Получить графики зависимостей, указанных в подразделе 2.3
4	Написать программу имитационного моделирования для алгоритма с селективными повторениями, описанного в 2.1.4: • Размер буфера на приемнике $L>1$. • Стратегия работы с буфером: в буфер записываются только сообщения с номерами, принадлежащими множеству $\{i+1, i+2, \ldots, i+L\}$ при ожидании сообщения с номером i. • Сообщения выдаются пользователю только в соответствии с правильным порядком следования, определяемого нумерацией на источнике. Получить графики зависимостей, указанных в подразделе 2.3

Вариант	Задание
5	Написать программу имитационного моделирования для алгоритма с виртуальными каналами, описанного в 2.1.5: • Размер буфера на приемнике для каждого виртуального канала неограничен. • Сообщения выдаются пользователю только в соответствии с правильным порядком следования, определяемого нумерацией на источнике. Получить графики зависимостей, указанных в подразделе 2.3
6	Написать программу имитационного моделирования для алгоритма с виртуальными каналами, описанного в 2.1.5: • Размер буфера на приемнике для каждого виртуального канала имеет свое ограничение на длину (L_1,L_2,…,$L\tau$). • Сообщения выдаются пользователю только в соответствии с правильным порядком следования, определяемого нумерацией на источнике. Получить графики зависимостей, указанных в подразделе 2.3
7	Написать программу имитационного моделирования для алгоритма с виртуальными каналами, описанного в 2.1.5: • Размер буфера на приемнике имеет общее ограничение L и делится между всеми виртуальными каналами. • Сообщения выдаются пользователю только в соответствии с правильным порядком следования, определяемого нумерацией на источнике. Получить графики зависимостей, указанных в подразделе 2.3
8	Написать программу имитационного моделирования для алгоритма при высокой вероятности ошибки, описанного в 2.1.6. Программа должна быть написана как для случая, когда нет ограничения на количество повторных передач, так и для числа повторных передач ограниченного n. Получить графики зависимостей, указанных в подразделе 2.3. Сравнить алгоритм с возвратом и алгоритм при высокой вероятности ошибки для ограниченного и неограниченного числа повторных передач

2.5. Требования к отчету по лабораторной работе

Отчет должен содержать:

1. Титульный лист
2. Цель и постановку задачи.
3. Описание моделируемой системы.
4. Описание исследуемого алгоритма передачи по каналу с обратной связью (описание можно представить в виде псевдокода или блок-схемы).
5. Графики зависимостей:

a) Коэффициента использования канала от вероятности ошибки в прямом канале при некоторых фиксированных значениях τ: $\pi(p,\tau=\text{const})$;

b) Коэффициента использования канала от величины задержки получения квитанции при некоторых фиксированных значениях p: $\pi(p=\text{const},\tau)$;

c) Среднего числа повторных передач от вероятности ошибки в канале при некоторых фиксированных значениях τ: $\bar{N}(p,\tau=\text{const})$;

d) Среднего числа повторных передач от величины задержки получения квитанции при некоторых фиксированных значениях p: $\bar{N}(p=\text{const},\tau)$;

6. Зависимости, полученные для $\pi(p,\tau=\text{const})$ и $\pi(p=\text{const},\tau)$, сравнить с аналогичными зависимостями для алгоритмов с возвратом и с виртуальными каналами. Использовать формулы, приведенные в 1.3 и 1.5.

7. Необходимо вывести лог работы программы и нарисовать по нему временную диаграмму.

8. Сформулировать выданное преподавателем дополнительное задание и описать результаты выполнения.

9. Выводы по полученным зависимостям и результатам дополнительного задания.

10. Листинг программы имитационного моделирования присутствует только в электронной версии отчета.

3. МЕТОДЫ УПРАВЛЕНИЯ ДОСТУПОМ К СРЕДЕ НА КАНАЛЬНОМ УРОВНЕ В СИТЕМАХ С МНОЖЕСТВОМ АБОНЕНТОВ

Раздел посвящен исследованию типовых методов и алгоритмов разделения общего ресурса канала между абонентами, а также определению характеристик рассматриваемых алгоритмов в рамках базовой модели множественного доступа с использованием численных расчетов и имитационного моделирования.

3.1. Модели систем и методы доступа к общему ресурсу канала

3.1.1. Доступ с разделением времени

Рассматривается синхронная система, то есть у всех абонентов системы имеются синхронизированные часы (единая служба времени). Можно считать, что данная система децентрализована, так как все абоненты равноправны. Абоненты организуют доступ к каналу, используя систему общего времени. Разделение времени относиться к статическому закреплению канала, когда ресурс канала некоторым способом делиться на доли и каждому абоненту выделяется доля ресурса канала.

Предполагается, что все сообщения у абонентов имеют одинаковую длину, и время передачи одного сообщения принято за единицу времени. Все время передачи по каналу разбито на окна. Длина одного окна соответствует времени передачи одного сообщения. Окна закреплены за абонентами, и абонент может передавать сообщения только в начале своего окна. Рассматривается система с M абонентами. В среднем у всех абонентов в одну единицу времени возникает λ сообщений (интенсивность входного потока λ). Интенсивность входного потока у всех абонентов в системе одинакова и у каждого абонента она равна $\frac{\lambda}{M}$.

Рассмотрим пример работы системы, использующей доступ с разделением времени для 4 абонентов ($M=4$). Будем считать, что первое окно закреплено за первым абонентом. Начиная с него, окна закреплены за каждым из четырех абонентов по порядку ($\{1,2,3,4,1,2,3,4,1,\ldots\}$). Опишем события в рассматриваемом примере по окнам:

Первое окно. Окно закреплено за абонентом A_1. У абонента A_1 нет готового к передаче сообщения. В течение данного окна у абонента A_1 появилось первое сообщение, которое будем обозначать как $A_1(1)$.

Абонент A_1 не может передать свое сообщение в текущем окне, так как передача сообщения занимает целое окно. То есть абонент может передать свое сообщение в закрепленном за ним окне, только если к началу окна у абонента уже было сообщение для передачи. Поэтому абонент A_1 будет ждать очередного своего окна.

Второе окно. Окно закреплено за абонентом A_2. У абонента A_2 нет готового к передаче сообщения. В течение данного окна у абонента A_4 появилось первое сообщение, которое будем обозначать как $A_4(1)$. Абонент A_4 ждет свое окно.

Третье окно. Окно закреплено за абонентом A_3. У абонента A_3 нет готового к передаче сообщения. В течение данного окна у абонента A_1 появилось второе сообщение, которое будем обозначать как $A_1(2)$. Теперь у абонента A_1 в буфере храниться очередь из сообщений $A_1(1)$ и $A_1(2)$, которые он будет передавать по очереди (в порядке появления). При этом за одно свое окно абонент сможет передать только одно сообщение.

Четвертое окно. Окно закреплено за абонентом A_4. К началу окна у абонента A_4 имеется готовое к передаче сообщение. К концу данного окна абонент A_4 успешно передаст сообщение $A_4(1)$.

Пятое окно. Окно закреплено за абонентом A_1. К началу окна у абонента A_1 имеется готовое к передаче сообщение. К концу данного окна абонент A_1 успешно передаст сообщение $A_1(1)$. При этом в буфере абонента A_1 осталось сообщение $A_1(2)$, которое он сможет передать только в своем следующем окне. В течение данного окна у абонента A_2 появилось первое сообщение, которое будем обозначать как $A_2(1)$.

Шестое окно. Окно закреплено за абонентом A_2. К началу окна у абонента A_2 имеется готовое к передаче сообщение. К концу данного окна абонент A_2 успешно передаст сообщение $A_2(1)$.

Седьмое окно. Окно закреплено за абонентом A_3. У абонента A_3 нет готового к передаче сообщения.

Восьмое окно. Окно закреплено за абонентом A_4. У абонента A_4 нет готового к передаче сообщения.

Девятое окно. Окно закреплено за абонентом A_1. К началу окна у абонента A_1 имеется готовое к передаче сообщение. К концу данного окна абонент A_1 успешно передаст сообщение $A_1(2)$.

События появления сообщений у абонентов в рассматриваемом примере в рамках лабораторной работы будем представлять по окнам в следующем виде:

1 окно	2 окно	3 окно	4 окно	5 окно	6 окно	7 окно	8 окно	9 окно
$A_1(1)$	$A_4(1)$	$A_1(2)$		$A_2(1)$				

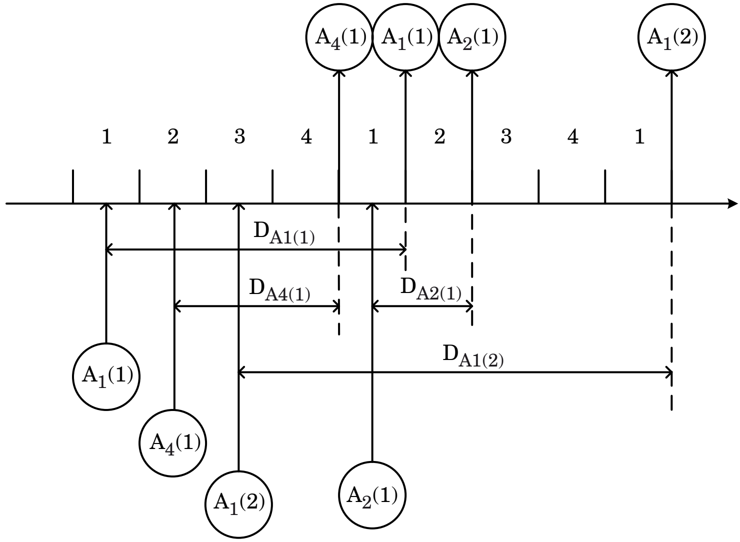

Рис. 18. Пример работы системы
с разделением времени для 4 абонентов

Временная диаграмма для рассматриваемого примера представлена на рис. 18.

3.1.2. Доступ по запросу

Доступ по запросу относится к централизованному методу управления доступом к среде. Рассматривается синхронная система, то есть у всех абонентов системы имеются синхронизированные часы (единая служба времени). Абоненты организуют доступ к каналу, используя систему общего времени. Доступ по запросу относиться к динамическому закреплению канала. При динамическом закреплении некоторым образом определяется потребность абонента в ресурсе канала и после этого весь ресурс канала временно отдается одному абоненту.

Рассматривается система с M абонентами. В среднем у всех абонентов в одну единицу времени возникает λ сообщений (интенсивность входного потока λ). Интенсивность входного потока у всех абонентов в системе одинакова и у каждого абонента она равна $\frac{\lambda}{M}$.

Предполагается, что все сообщения у абонентов имеют одинаковую длину, и время передачи одного сообщения принято за единицу времени. По каналу кроме сообщений передаются запросы к абонентам. Время в течение, которого передается запрос и приходит ответ на этот запрос, равно τ. Абонент принимает запросы, только если у него есть сообщение на передачу. Получив свой запрос, абонент отвечает на него и после передает одно сообщение. Запросы к абонентам чередуются по порядку ($\{З_1, З_2, З_3, З_4, З_1, З_2, \ldots\}$, где $З_i$ – запрос к i абоненту).

Рассмотрим пример работы системы, использующей доступ по запросу для 4 абонентов ($M=4$). Будем считать, что первый запрос относиться к первому абоненту, затем запросы к абонентам идут по порядку. В отличие от системы из 1.1 помимо окон для передачи сообщений в системе присутствуют и окна запросов, которые длятся τ. В рамках примера будем предполагать, что сообщения у абонентов возникают только в окнах запроса, однако в общем случае это не так. Опишем события в рассматриваемом примере по окнам:

Запрос 1. Это окно запроса к A_1. У абонента A_1 нет готового к передаче сообщения, и он не отвечает на запрос. В течение данного окна у абонента A_1 появилось первое сообщение, которое будем обозначать как $A_1(1)$. Абонент A_1 не может передать свое сообщение в текущем окне, так как передача запроса и ответ на него занимает τ. То есть для ответа на запрос и последующей отправки сообщения абонент должен иметь готовое сообщение к началу своего запроса. Поэтому абонент A_1 будет ждать очередного своего запроса.

Запрос 2. Это окно запроса к A_2. У абонента A_2 нет готового к передаче сообщения, и он не отвечает на запрос.

Запрос 3. Это окно запроса к A_3. У абонента A_3 нет готового к передаче сообщения, и он не отвечает на запрос. В течение данного окна у абонента A_4 появилось первое сообщение, которое будем обозначать как $A_4(1)$. Абонент A_4 ждет своего запроса.

Запрос 4. Это окно запроса к A_4. У абонента A_4 есть готовое к передаче сообщение. Он принимает запрос и отправляет ответ на него. После чего система выделяет ему время на передачу одного сообщения.

Окно на передачу сообщения A_4. В течение данного окна абонент A_4 передает свое сообщение $A_4(1)$.

Запрос 5. Это окно запроса к A_1. У абонента A_1 есть готовое к передаче сообщение. Он принимает запрос и отправляет ответ на него. После чего система выделяет ему время на передачу одного сообщения. В течение данного окна у абонента A_2 появилось первое сообщение, которое будем обозначать как $A_2(1)$.

Окно на передачу сообщения A_1. В течение данного окна абонент A_1 передает свое сообщение $A_1(1)$.

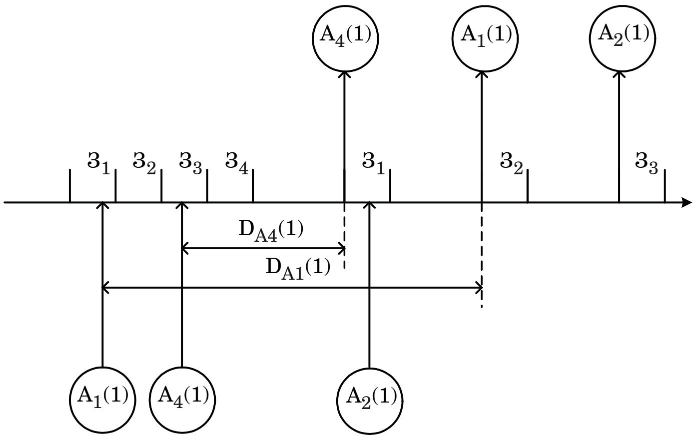

Рис. 19. Пример работы системы с доступом по запросу для 4 абонентов

Запрос 6. Это окно запроса к A_2. У абонента A_2 есть готовое к передаче сообщение. Он принимает запрос и отправляет ответ на него. После чего система выделяет ему время на передачу одного сообщения.

Окно на передачу сообщения A_2. В течение данного окна абонент A_2 передает свое сообщение $A_2(1)$.

Запрос 7. Это окно запроса к A_3. У абонента A_3 нет готового к передаче сообщения, и он не отвечает на запрос.

События появления сообщений у абонентов в рассматриваемом примере в рамках лабораторной работы для упрощения будем представлять только по окнам запросов в следующем виде:

Запрос 1	Запрос 2	Запрос 3	Запрос 4	Запрос 5	Запрос 6	Запрос 7
$A_1(1)$		$A_4(1)$		$A_2(1)$		

Временная диаграмма для рассматриваемого примера представлена на рис. 19.

3.1.3. Случайный доступ

Алгоритмы случайного доступа, рассматриваемые в лабораторной работе, относятся к децентрализованным системам управления доступа к среде. При децентрализованном доступе устройства равноправны и по некоторому алгоритму они организовывают доступ к общему каналу. Предполагается синхронная система, то есть все абоненты имеют синхронизированные часы (единая служба времени). Абоненты используют систему общего времени для доступа к каналу. Случайный доступ относиться к динамическому закреплению ресурса канала. Каждый абонент передает по каналу готовое к передаче сообщение в соответствии с некоторым алгоритмом. Если по каналу передаются сразу несколько сообщений от разных абонентов, то возникает конфликт, который разрешается в соответствии с алгоритмом.

Введем ряд допущений для рассматриваемой модели случайного множественного доступа:

Допущение 1. Предполагается, что все сообщения у всех абонентов имеют одинаковую длину, а время передачи одного сообщения принято за единицу времени. Все время передачи по каналу разбито на окна, длительность окна соответствует времени передачи одного сообщения. Абоненты точно знают моменты разделения и могут начать передачу только в начале окна.

Допущение 2. В окне возможно 3 события:

a) Событие «Конфликт». В окне одновременно передают два абонента или больше. Считается, что из-за наложения сигналов сообщения полностью искажаются и не могут быть приняты правильно.

b) Событие «Успех». В окне передает один абонент, в этом случае считается, что абонент успешно передает сообщение.

c) Событие «Пусто». В окне никто не передает.

Допущение 3. Абоненты наблюдают выход канала в конце окна и достоверно определяют, какое из трех событий произошло.

Допущение 4. В системе имеется M абонентов. В среднем у всех абонентов в одну единицу времени возникает λ сообщений (интенсивность входного потока λ). То есть предполагается Пуассоновский входной поток с параметром λ. Интенсивность входного потока у всех абонентов в системе одинакова и у каждого абонента она равна $\frac{\lambda}{M}$.

Алгоритмом случайного множественного доступа называется правило в соответствии, с которым каждый абонент, имеющий готовое к передаче сообщение, в начале каждого окна решает, передавать сообщение или нет. Существуют различные виды алгоритмов случайного множественного доступа, в данной лабораторной работе рассматриваются следующие алгоритмы:

1. Алгоритм ALOHA:

• Вероятностный вариант алгоритма. Параметр алгоритма: вероятность передачи $p \in (0,1]$. Если к началу с номером окна с номером t у абонента есть готовое для передачи сообщение, то абонент принимает решение передавать с вероятностью p. В случае конфликта абонент продолжает попытки передачи сообщения, принимая решение о передаче с вероятностью p в начале каждого окна.

• Интервальный вариант алгоритма. Параметр алгоритма – интервал выбора окна для передачи W. Если к началу окна с номер t у абонента имеется готовое к передаче сообщение, то абонент случайным образом выбирает для передачи окно из множества окон с номерами $\{t, t+1, \ldots, t+W-1\}$. Если абонент выбрал окно t' из этого множества и произошел конфликт, то абонент случайным образом выбирает новое окно для передачи из множества $\{t'+1, t'+2, \ldots, t'+W\}$.

2. Адаптивный алгоритм ALOHA:

• Вероятностный вариант алгоритма. Параметр алгоритма: начальная вероятность передачи $p_0 \in (0,1]$. Вероятность передачи сообщения абонентами в канал изменяется в зависимости от событий, которые произошли в канале по следующему правилу:

$$p_{t+1} = \begin{cases} \max\left(\dfrac{1}{M}, \dfrac{p_t}{2}\right), & \text{при "конфликте" в канале} \\ p_t, & \text{при "успехе" в канале} \\ \min\left(1, 2p_t\right), & \text{при "пусто" в канале} \end{cases},$$

где p_t – вероятность, с которой все абоненты с готовыми к передаче сообщениями, передают их по каналу в начале t-го окна.

• Интервальный вариант алгоритма. Параметр алгоритма: начальный интервал выбора окна для передачи W_0. Абонент выбирает номер окна, в котором он будет передавать в соответствии со значением W_t. При этом параметр алгоритма W_t изменяется в соответствии с событиями в канале по следующему правилу:

$$W_{t+1} = \begin{cases} \min\left(2M, 2W_t\right), & \text{при "конфликте" в канале} \\ W_t, & \text{при "успехе" в канале} \\ \max\left(1, \left\lceil \dfrac{W_t}{2} \right\rceil\right), & \text{при "пусто" в канале} \end{cases},$$

где W_t – размер интервала, в котором абонент выбирает окно для передачи в t-м окне.

3. Алгоритм двоичной экспоненциальной отсрочки (двоичный экспоненциальный откат). Абоненты узнают о событии в канале только в окне, в котором они передавали (изменяется допущение 3).

• Вероятностный вариант. Параметры алгоритма: $p_{\min}$ – минимальная вероятность передачи $p_{\min} \in (0,1]$, $p_{\max}$ – максимальная вероятность передачи $p_{\max} \in (0,1]$, $p_{\min} < p_{\max}$. Каждый абонент меняет вероятность передачи в соответствии с событием, которое произошло в канале при передаче его сообщения по следующему правилу:

$$p_{i+1} = \begin{cases} \max\left(\dfrac{p_i}{2}, p_{\min}\right), & \text{при "конфликте" в канале} \\ p_{\max}, & \text{при "успехе" в канале} \end{cases},$$

где p_i – вероятность, с которой абонент передавал при i-й попытке передаче; $p_{\min}$ – минимальная вероятность передачи; $p_{\max}$ – максимальная вероятность передачи.

• Интервальный вариант. Параметры алгоритма: $W_{\min}$ – минимальный интервал выбора окна для передачи, $W_{\max}$ – максимальный интервал выбора окна для передачи, $W_{\min} < W_{\max}$. Каждый абонент меняет интервал, в котором он выбирает окно для передачи в соответствии с событием, которое произошло при передаче его сообщения по следующему правилу:

$$W_{i+1} = \begin{cases} \min(2W_i, W_{\max}), & \text{при "конфликте" в канале} \\ W_{\min}, & \text{при "успехе" в канале} \end{cases},$$

где W_i – интервал, в котором выбиралось окно для i-й попытки передачи; $W_{\max}$ – максимальная длина интервала, $W_{\min}$ – минимальная длина интервала.

4. Древовидный алгоритм. При появлении сообщения абонент сразу передает его по каналу в начале следующего окна. Наблюдая выход канала, все абоненты строят дерево разрешения конфликтов и на основе этого дерева выбирают окно для передачи сообщения. В случае события «конфликт» абоненты строят вершину, которая является корнем дерева. У каждой вершины имеются два потомка, верхний и нижний. Абоненты, попавшие в «конфликт» случайным образом, выбирают для повторной передачи одного из потомков. При повторном «конфликте» строятся два новых потомка из данного. Абоненты не знают, сколько передавало в окне, а, следовательно, и кратность конфликта. Конфликт разрешается, когда в последнем нижнем потомке происходят либо событие «успех», либо «пусто». Абоненты, поступающие в систему во время разрешения «конфликта», так же строят дерево и откладывают свою передачу до его разрешения.

Рассмотрим пример работы «неулучшенного» древовидного алгоритма для конфликта кратности 3:

Окно 1. В данном окне передавали свои сообщения три абонента 1, 2 и 3. В соответствии с допущением 2 в данном окне произошло событие «конфликт». Абоненты начинают строить дерево разрешения конфликта. У данной вершины появляется верхний потомок (в окне 2) и нижний (номер окна для данного потомка зависит от событий в верхнем потомке) потомок. Каждый абонент независимо принимает решение о том, в каком потомке повторить попытку передачи сообщения. Так как потомка два, то вероятность выбора любого из них одинакова и равна $\frac{1}{2}$. В рассматриваемом примере:

- Абонент 1 с вероятностью $\frac{1}{2}$ выбрал нижнего потомка;
- Абонент 2 с вероятностью $\frac{1}{2}$ выбрал нижнего потомка;
- Абонент 3 с вероятностью $\frac{1}{2}$ выбрал нижнего потомка.

Совместная вероятность такого исхода равна произведению вероятностей выбора для каждого абонента (так как эти события независимы), то есть $\frac{1}{8}$.

Окно 2. Так как никто не выбрал верхнего потомка, то никто в данном окне не передавал. В соответствии с допущением 2 в данном окне произошло событие «пусто». Данное окно соответствует листу дерева с событием «пусто». Система переходит к нижнему потомку.

Окно 3. Так как все три абонента выбрали нижнего потомка, то в данном окне повторится событие «конфликт». Абоненты продолжают строить дерево разрешения конфликта. У данной вершины появляется верхний потомок и нижний потомок. Каждый абонент независимо принимает решение о том, в каком потомке повторить попытку передачи сообщения. В рассматриваемом примере:

- Абонент 1 с вероятностью $\frac{1}{2}$ выбрал верхнего потомка;
- Абонент 2 с вероятностью $\frac{1}{2}$ выбрал нижнего потомка;
- Абонент 3 с вероятностью $\frac{1}{2}$ выбрал верхнего потомка.

Совместная вероятность такого исхода, как и ранее равна $\frac{1}{8}$.

Окно 4. В данном окне передают абонент 1 и 3, так как они выбрали верхнего потомка. Происходит событие «конфликт». Абоненты продолжают строить дерево разрешения конфликта. У данной вершины появляется верхний потомок и нижний потомок. Абоненты 1 и 3 независимо принимают решение о том, в каком потомке повторить попытку передачи сообщения. В рассматриваемом примере:

- Абонент 1 с вероятностью $\frac{1}{2}$ выбрал верхнего потомка;
- Абонент 3 с вероятностью $\frac{1}{2}$ выбрал нижнего потомка.

Совместная вероятность такого исхода, как и ранее равна $\frac{1}{4}$.

Окно 5. В данном окне передавал только абонент 1. В соответствии с допущением 2 происходит событие «успех». Абонент с номером 1 успешно передает свое сообщение. Данное окно соответствует листу дерева с событием «успех». Система переходит к нижнему потомку.

Окно 6. В данном окне передавал только абонент 3. В соответствии с допущением 2 происходит событие «успех». Абонент с номером 3 успешно передает свое сообщение. Данное окно соответствует листу дерева с событием «успех». Система переходит к следующему нижнему потомку.

Окно 7. В данном окне передавал только абонент 2. В соответствии с допущением 2 происходит событие «успех». Абонент с номером 2 успешно передает свое сообщение. Данное окно соответствует листу дерева с событием «успех». Так как это был последний потомок в дереве, то процедура разрешения конфликта завершается.

Временная диаграмма для рассмотренного примера работы «неулучшенного» древовидного алгоритма для конфликта кратности 3 приведена на рис. 20. Вероятность формирования дерева разрешения конфликта, представленного на рис. 20 определяется как $\frac{1}{8} \cdot \frac{1}{8} \cdot \frac{1}{4} = \frac{1}{256}$.

Данный алгоритм может быть улучшен следующим образом: если в верхнем потомке пусто, значит, все абоненты выбрали нижнего потомка и там снова будет «конфликт», поэтому можно пропустить передачу в этом окне и сразу строить два новых потомка.

Рассмотрим пример работы «улучшенного» древовидного алгоритма для конфликта кратности 3:

Окно 1. В данном окне передавали свои сообщения три абонента 1, 2 и 3. В соответствии с допущением 2 в данном окне произошло событие «конфликт». Абоненты начинают строить дерево разрешения конфликта. У данной вершины появляется верхний потомок

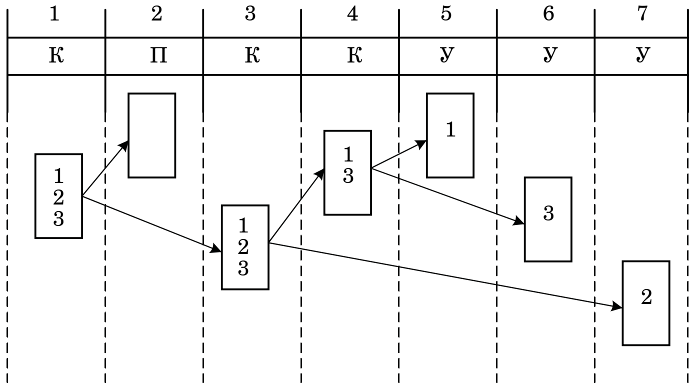

Рис. 20. Пример работы древовидного алгоритма для конфликта кратности 3

(в окне 2) и нижний (номер окна для данного потомка зависит от событий в верхнем потомке) потомок. Каждый абонент независимо принимает решение о том, в каком потомке повторить попытку передачи сообщения. Так как потомка два, то вероятность выбора любого из них одинакова и равна $\frac{1}{2}$. В рассматриваемом примере:

- Абонент 1 с вероятностью $\frac{1}{2}$ выбрал нижнего потомка;
- Абонент 2 с вероятностью $\frac{1}{2}$ выбрал нижнего потомка;
- Абонент 3 с вероятностью $\frac{1}{2}$ выбрал нижнего потомка.

Совместная вероятность такого исхода равна произведению вероятностей выбора для каждого абонента (так как эти события независимы), то есть $\frac{1}{8}$.

Окно 2. Так как никто не выбрал верхнего потомка, то никто в данном окне не передавал. В соответствии с допущением 2 в данном окне произошло событие «пусто». Данное окно соответствует листу дерева с событием «пусто». Так как все абоненты видят событие «пусто» в верхнем потомке, то они понимают, что в нижнем потомке гарантированно произойдет событие «конфликт». Поэтому они не будут передавать сообщения в соответствии выбором, сделанным в окне 1, и сразу стоят верхнего и нижнего потомка. Каждый абонент независимо принимает решение о том, в каком потомке повторить попытку передачи сообщения. В рассматриваемом примере:

- Абонент 1 с вероятностью $\frac{1}{2}$ выбрал верхнего потомка;
- Абонент 2 с вероятностью $\frac{1}{2}$ выбрал нижнего потомка;
- Абонент 3 с вероятностью $\frac{1}{2}$ выбрал верхнего потомка.

Совместная вероятность такого исхода, как и ранее равна $\frac{1}{8}$.

Окно 3. В данном окне передают абонент 1 и 3, так как они выбрали верхнего потомка. Происходит событие «конфликт». Абоненты продолжают строить дерево разрешения конфликта. У данной вершины появляется верхний потомок и нижний потомок. Абоненты 1 и 3 независимо принимают решение о том, в каком потомке повторить попытку передачи сообщения. В рассматриваемом примере:

- Абонент 1 с вероятностью $\frac{1}{2}$ выбрал верхнего потомка;
- Абонент 3 с вероятностью $\frac{1}{2}$ выбрал нижнего потомка.

Совместная вероятность такого исхода, как и ранее равна $\frac{1}{4}$.

Окно 4. В данном окне передавал только абонент 1. В соответствии с допущением 2 происходит событие «успех». Абонент с номером 1 успешно передает свое сообщение. Данное окно соответствует листу дерева с событием «успех». Система переходит к нижнему потомку.

Окно 5. В данном окне передавал только абонент 3. В соответствии с допущением 2 происходит событие «успех». Абонент с номером 3 успешно передает свое сообщение. Данное окно соответствует листу дерева с событием «успех». Система переходит к следующему нижнему потомку.

Окно 6. В данном окне передавал только абонент 2. В соответствии с допущением 2 происходит событие «успех». Абонент с номером 2 успешно передает свое сообщение. Данное окно соответствует листу дерева с событием «успех». Так как это был последний потомок в дереве, то процедура разрешения конфликта завершается.

Временная диаграмма для рассмотренного примера работы «улучшенного» древовидного алгоритма для конфликта кратности 3 приведена на рис. 21. Вероятность формирования дерева разрешения конфликта, представленного на рис. 21, такая же, как и у дерева на

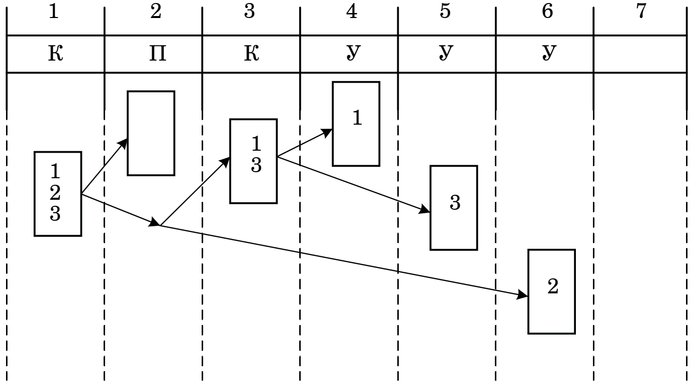

Рис. 21. Пример работы улучшенного древовидного алгоритма для конфликта кратности 3

рис. 20 и равна $\frac{1}{256}$. Можно отметить, что в обоих рассмотренных примерах происхходиили одинаковые ссобытия (решения о выборе потомка в дереве), однако «улучшенный» алгориитм затратил на разрешения конфликта на одно окно меньше.

3.1.4. Точный анализ и анализ на качественном уровне на примере одноканальной системы массового обслуживания

В лабораторной работе будет рассматриваться и анализироваться система массового обслуживания (СМО) $M|D|1$.

Для описания и классификации СМО используется система обозначений, введенная Д. Кендэллом: $A|S|c|K$, где A – время между прибытиями заявок в очередь, S – закон распределение времени обслуживания, c – количество обслуживающих устройств, K – размер очереди (буфера). Так же может быть указан порядок обслуживания очереди.

Для рассматриваемой СМО:

- M – означает, что количество возникающих заявок распределено по закону Пуассона с интенсивностью λ, а интервал между моментами появления заявок по экспоненциальному закону;
- D – детерминированное время обработки заявки;
- 1 – наличие единственного обслуживающего прибора.
- Порядок обслуживания заявок FIFO (First In First Out), то есть «первым вошел, первым вышел».

Так же рассматриваемая система имеет одну очередь, а время обслуживания равняется единице ($\mu=1$). Данная СМО изображена на рис. 22.

Рис. 22. Рассматриваемая система массового обслуживания: λ – интенсивность входного потока; μ – время обслуживания (μ=1)

Введем в рассмотрение случайную величину D – интервал времени от момента появления сообщения до момента, когда сообщение доставлено получателю. Данный интервал является задержкой в обслуживании заявки (передачи сообщения). Среднюю задержку можно вычислить как:

$$d = M[D].$$

Рассматриваются СМО $M|D|1$ двух типов:

• Синхронная. В синхронной СМО время дискретно и разбито на окна единичной длины, сообщения могут быть переданы (обслужены) только в начале следующего окна в порядке очереди.

• Асинхронная. В асинхронной СМО сообщения могут быть переданы (обслужены), как только появились в системе, в порядке очереди.

Будет предполагать, что каждое возникшее в системе сообщение это новый абонент, который после передачи (обслуживания) сообщения покидает систему. Тогда понятия абонент и сообщения будут эквивалентны. Для обоих типов СМО среднее число абонентов (сообщений) в системе $\bar{N}$ определяется в соответствии с выражением:

$$\bar{N}(\lambda) = \mathrm{M}[N(\lambda)] = \frac{\lambda(2-\lambda)}{2(1-\lambda)}.$$

График зависимости среднего числа сообщений $\bar{N}$ от интенсивности входного потока λ представлен на рис. 23.

Средняя задержка d может быть вычислена как:

$$d(\lambda) = \mathrm{M}[D(\lambda)] = \begin{cases} \dfrac{\bar{N}(\lambda)}{\lambda} & \text{, для асинхронной} \\ \dfrac{\bar{N}(\lambda)}{\lambda} + 0.5, & \text{для синхронной} \end{cases}.$$

График зависимости средней задержки d от интенсивности входного потока λ представлен на рис. 24.

Для простых систем (например, рассматриваемые в этом пункте СМО) можно в явном виде получить выражение для $d(\lambda)$. Следует отметить, что данное выражение справедливо при λ, не превышающих некоторое значение $\lambda_{кр}$ (для рассматриваемой системы $M|D|1$

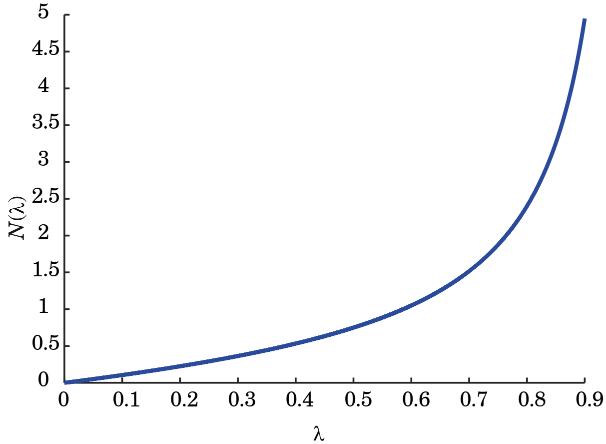

Рис. 23. Среднее число абонентов в системе для СМО

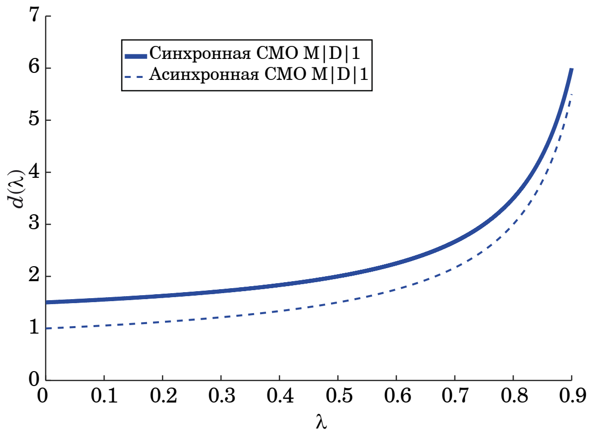

Рис. 24. Средняя задержка для СМО $M|D|1$

$\lambda_{кр}=1$). Получение выражения для вычисления $d(\lambda)$ будем называть *точным анализом системы*. Для более сложных систем найти данное выражение затруднительно, однако почти для любой системы можно найти в явном виде выражение $d_0 \triangleq \lim\limits_{\lambda \to 0}(d(\lambda))$. Нахождение d_0 и $\lambda_{кр}$ без получения явного выражения $d(\lambda)$, будем называть *анализом системы на качественном уровне*.

3.2. Вопросы для допуска к лабораторной работе

3.2.1. Построение временной диаграммы работы систем с разделением времени и доступом по запросу

Доступ с разделением времени

Необходимо нарисовать временную диаграмму работы системы при доступе с разделением времени в соответствии с полученным вариантом. В системе рассматривается 4 абонента ($M=4$). Работа системы начинается с окна, закрепленного за первым абонентом.

Вариант	1 окно	2 окно	3 окно	4 окно	5 окно	6 окно
1	$A_1(1)$		$A_1(2), A_3(1)$	$A_4(1)$		$A_3(2)$
2	$A_4(1)$	$A_3(1), A_2(1)$		$A_2(2)$	$A_1(1)$	
3		$A_4(1)$		$A_1(1), A_3(1)$		$A_3(2), A_2(1)$
4	$A_3(1)$	$A_2(1)$	$A_1(1)$		$A_4(1), A_4(2)$	
5	$A_2(1), A_4(1)$		$A_4(2), A_1(1)$		$A_3(1)$	
6	$A_2(1)$		$A_3(1), A_4(1)$	$A_3(2)$	$A_3(3)$	
7		$A_1(1)$		$A_4(1)$	$A_3(1), A_2(1)$	$A_4(2)$

где $A_i(j)$ обозначает j-е сообщение i-го абонента. Столбец показывает, в каком временном окне появилось сообщение (порядковый номер окна при работе системы, см. пример в пункте 3.1.1).

Доступ по запросу

Необходимо нарисовать временную диаграмму работы системы с доступом по запросу в соответствии с полученным вариантом. В системе рассматривается 4 абонента ($M=4$). Работа системы начинается с запроса для первого абонента.

Вари-ант	Запрос 1	Запрос 2	Запрос 3	Запрос 4	Запрос 5	Запрос 6
1	$A_1(1)$			$A_2(1), A_4(1)$		
2		$A_3(1)$			$A_3(2)$	$A_1(1)$
3		$A_1(1)$		$A_1(2)$	$A_3(1)$	
4	$A_4(1)$		$A_4(1)$			$A_2(1)$
5	$A_2(1)$	$A_1(1), A_4(1)$				
6			$A_1(1), A_4(1)$		$A_2(1)$	
7	$A_2(1)$		$A_2(2)$	$A_3(1)$		

где $A_i(j)$ обозначает j-е сообщение i-го абонента. Столбец показывает, во время какого запроса появилось сообщение (порядковый номер окна запроса при работе системы, см. пример в пункте 3.1.2).

3.2.2. Нахождение средней задержки в системах с разделением времени и доступом по запросу

Доступ с разделением времени

Необходимо найти среднюю задержку d_0 системы, рассмотренной в пункте 3.1.1 d_0 – это значение задержки для абонента в пустой системе (см. пункт 3.1.4). Значение d_0 должно быть представлено в виде функции от M, где M – количество абонентов в системе. Для этого рассматривается ситуация, когда в пустой системе у некоторого абонента возникает сообщение.

Введем следующие обозначения: D_1 – ожидание начала очередного окна, D_1 – ожидание своего окна, D_3 – передача сообщения. Тогда задержка для i-го абонента в пустой системе может быть рассмотрена как сумма трех этих компонент (D_1, D_2 и D_3, см. рис. 25).

Значение d_0 в такой системе вычисляется по формуле:

$$d_0 = M[D_1] + M[D_2] + M[D_3],$$

где $M[D_1] = \dfrac{1}{2}$, $M[D_3]=1$.

Для получения окончательного выражения d_0, необходимо самостоятельно получить выражение для $\mathrm{M}[D_2]$, с учётом того, что:

$$D_2 \in \{0,1,\dots,M-1\} \text{ и } Pr\{D_2 = j\} = \frac{1}{M}, j = \{0,1,\dots,M-1\}.$$

Окончательное выражение для значения d_0 от числа абонентов M ($d_0(M)$) должно быть получено в явном виде замкнутом виде (то есть

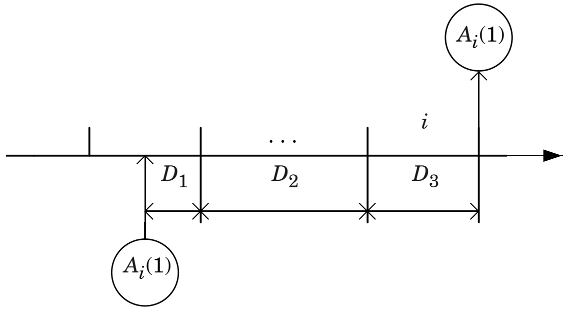

Рис. 25. Задержка для i-го абонента в пустой системе с разделением времени

в формуле должно отсутствовать суммирование с числом слагаемых, зависящим от числа абонентов M).

Доступ по запросу

Необходимо найти значение задержки d_0 для системы, рассмотренной в пункте 3.1.2. Ответ дать в виде функции от двух переменных M и τ, где M – количество абонентов в системе, а τ – время, затрачиваемое на передачу запроса и приём ответа от абонента. Для этого рассматривается ситуация, когда в пустой системе у некоторого абонента возникает сообщение. Когда система пустая, по каналу передаются только запросы.

Введем следующие обозначения: D_1 – ожидание очередного окна, D_2 – ожидание и обработка своего окна запроса, D_3 – передача сообщения. Тогда задержка для i-го абонента в пустой системе может быть рассмотрена как сумма трех этих компонент (D_1, D_2 и D_3, см. рис. 26).

Значение d_0 в такой системе вычисляется по формуле:

$$d_0 = M[D] = M[D_1] + M[D_2] + M[D_3],$$

где $M[D_1] = \frac{\tau}{2}$, $M[D_3]=1$.

Для получения окончательного выражения d_0, необходимо самостоятельно получить выражение для $M[D_2]$, с учётом того, что:

$$D_2 \in \{\tau, 2\tau, \ldots, M\tau\} \text{ и } Pr\{D_2 = j\} = \frac{1}{M}, j = \{1, 2, \ldots, M\}.$$

Окончательное выражение для значения d_0 от числа абонентов M и значения τ ($d_0(M,\tau)$) должно быть получено в явном виде (то есть

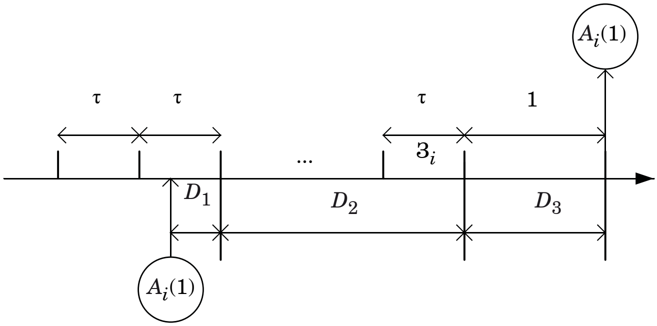

Рис. 26. Задержка для i-того абонента в пустой системе с доступом по запросу

в формуле должно отсутствовать суммирование с числом слагаемых, зависящим от числа абонентов M).

Сравнение доступа с разделения времени и доступ по запросу

Сравнить значение d_0 для доступа с разделением времени и доступа по запросу при фиксированном значении M и установить, начиная с какого значения $\tau_{пор}$, значение d_0 для доступа по запросу будет превышать значение d_0 для доступа с разделением времени. Значение $\tau_{пор}$ представить в виде функции от количества абонентов ($\tau_{пор}(M)$).

3.2.3. Анализ алгоритма ALOHA. Оптимальная вероятность передачи для M абонентов

Рассматривается простейший алгоритм случайного множественного доступа – алгоритм ALOHA. Каждый абонент, имеющий готовое для передачи сообщение, независимо принимает решение о его передаче в начале каждого окна с вероятностью $p<1$.

Необходимо решить следующие задачи:

Задача	Формулировка
1	Пусть в системе n абонентов, имеющих готовое для передачи сообщение. Тогда вероятность события «успех» определяется следующим образом: $$\Pr\{У \mid n\} = C_n^1 p(1-p)^{n-1} = np(1-p)^{n-1} = f(n,p).$$ Необходимо найти при каких значениях p функция $f(n,p)$ принимает максимальное значение, то есть решить оптимизационную задачу: $$\max_{0\le p\le 1} f(n,p).$$ Оптимальное значение p представить в виде функции от n.

Задача	Формулировка
2	При условии, что в системе имеется n абонентов необходимо вывести выражения для: • $\Pr\{П\|n\}$ – вероятность события, что в окне произойдет событие «пусто»; • $\Pr\{К\|n\}$ – вероятность события, что в окне произойдет событие «конфликт». Необходимо убедиться, что $\Pr\{У\|n\}+\Pr\{П\|n\}+\Pr\{К\|n\}=1$.
3	Если зафиксирован алгоритм разрешения конфликта и задана интенсивность входного потока λ, то средняя задержка $M[D]$ зависит от интенсивности входного потока λ и может быть записана как функция $d(\lambda)$. В таком случае скорость алгоритма случайного доступа определяется как: $R \triangleq sup\{\lambda : d(\lambda) < \infty\}.$ В курсе лекций было показано, что скорость алгоритма ALOHA для модели с числом абонентов M определяется как: $R_{ALOHA} = Mp(1-p)^{M-1}$ Чтобы скорость алгоритма была максимальной $p = \frac{1}{M}$. В этом случае: $R_{ALOHA} = \left(1 - \frac{1}{M}\right)^{M-1}.$ Необходимо доказать, что при $M \to \infty$: $R_{ALOHA} \approx e^{-1}$.

3.2.4. Анализ алгоритма ALOHA. Расчет средней задержки для одного абонента в пустой системе

Рассматривается алгоритм ALOHA. В пустой системе появляется один единственный абонент. Абонент принимает решение о передачи сообщения в начале каждого окна с вероятностью p и решает не передавать с вероятностью $(1 - p)$. Задержка D в такой системе равна интервалу времени от момента появления абонента в системе до окончания окна, в котором сообщение абонента будет успешно передано.

Введем следующие обозначения: D_1 – ожидание начала очередного окна, D_2 – количество окон, в которых абонент решил не передавать с вероятностью $(1 - p)$ D_3 – передача сообщения в окне, где абонент решил передавать с вероятностью p.

Тогда задержка для абонента в пустой системе может быть рассмотрена как сумма трех компонент (D_1, D_2 и D_3). Пример для задержки в алгоритме ALOHA при единственном абоненте в пустой системе приведен на рис. 27.

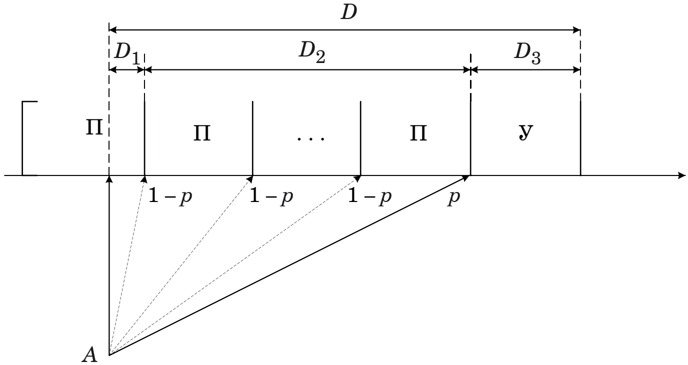

Рис. 27. Задержка для алгоритма ALOHA при единственном абоненте в пустой системе и вероятности передачи p

Значение d_0 (см. пункт 3.1.4) в такой системе вычисляется по формуле:

$$d_0 = M[D] = M[D_1] + M[D_2] + M[D_3],$$

где $M[D_1] = \frac{1}{2}$, $M[D_3] = 1$.

Для получения окончательного выражения для d_0 как функции от p, необходимо самостоятельно получить $M[D_2]$, с учетом того, что:

$$D_2 \in \{0, 1, \dots, \infty\} \text{ и } \quad \begin{aligned} Pr\{D_2 = 0\} &= p, \\ Pr\{D_2 = 1\} &= p(1-p), \\ Pr\{D_2 = 2\} &= ?, \\ &\dots \end{aligned}$$

Окончательное выражение для значения d_0 как функции от вероятности передачи p ($d_0(p)$) должно быть получено в явном виде (то есть в формуле должно отсутствовать суммирование с бесконечным числом слагаемых).

3.2.5. Анализ древовидного алгоритма. Расчет среднего времени разрешения конфликта кратности 2

Найти среднее время разрешения конфликта кратности 2 для «улучшенного» древовидного алгоритма.

3.2.6. Оценка характеристик системы массового обслуживания с помощью имитационного моделирования

Необходимо:

- Написать программу для моделирования синхронной и асинхронной системы массового обслуживания (см. пункт 3.1.4).
- Оценить с помощью программы имитационного моделирования среднюю задержку $M[D]$ и среднее число сообщений в системе $M[N]$ для каждой из систем.
- Получить зависимость данных характеристик от значения интенсивности входного потока λ.
- Сравнить с теоретическими значениями из пункта 3.1.4.

3.3. Порядок выполнения лабораторной работы

1. Необходимо в качестве допусков к лабораторной работе ответить на вопросы в подразделе 3.2 и сдать их преподавателю.

2. После сдачи всех допусков получить от преподавателя вариант задания для лабораторной работы.

3. Написать и отладить программу имитационного моделирования для метода доступа к среде, соответствующего полученному варианту.

4. Построить графики следующих зависимостей:

a) Средней задержки от интенсивности входного потока – $\bar{d}(\lambda)$;

b) Среднего числа абонентов в системе от интенсивности входного потока– $\bar{N}(\lambda)$;

c) Интенсивность выходного потока от интенсивности входного потока – $\lambda_{вых}(\lambda)$. Интенсивность выходного потока соответствует среднему количеству сообщений, успешно переданных за окно.

5. Продемонстрировать работу программы имитационного моделирования преподавателю.

6. Получить дополнительное задание для исследования.

7. В соответствии с вариантом и полученным дополнительным заданием оформить и сдать отчет.

Перед представлением программы имитационного моделирования преподавателю студент должен сам предварительно проверить правильность работы программы следующим образом:

1. Во всех вариантах число абонентов M задается равным единице ($M=1$). Для случайного доступа вероятность передачи задается равной единице ($p=1$). Для доступа по запросу параметр $\tau=0$. Необходимо построить графики $\bar{d}(\lambda)$ и $\bar{N}(\lambda)$, результаты должны совпадать с результатами для синхронной системы, описанной в 2.6. Если результаты не совпадают, то программа имитационного моделирования работает

неверно. Если результаты совпадают, то это не гарантирует правильность работы программы имитационного моделирования при $M>1$.

2. Задается число абонентов в системе больше единицы ($M>1$). Строится график зависимости выходной интенсивности $\lambda_{вых}(\lambda)$ и прямая для входной интенсивности $\lambda_{практ}(\lambda)$. Значения входной и выходной интенсивности должны совпадать при $\lambda < \lambda_{крит}$.

3.4. Варианты заданий для лабораторной работы

Во всех вариантах заданий рассматривается система с M абонентами, у каждого абонента имеется ограниченная или неограниченная очередь. На вход каждой очереди поступает пуассоновский поток сообщений с интенсивностью $\frac{\lambda}{M}$. Все абоненты передают данные по общему каналу. Варианты заданий отличаются только правилом доступа к общему ресурсу канала. Необходимо написать программу имитационного моделирования в соответствии с полученным вариантом. Программа имитационного моделирования должна быть реализована таким образом, чтобы можно было задавать произвольное значение M, в том числе и $M=1$. Должен быть предусмотрен способ сравнения результатов моделирования при $M=1$ с рассмотренной в 2.6 синхронной СМО $M|D|1$.

Вариант	Задание
1	Необходимо: • Написать программу имитационного моделирования для системы с разделением времени. Входными данными является число абонентов в системе M. • Написать программу имитационного моделирования для системы с доступом по запросу. Входными данными является число абонентов в системе M и значение τ. • Проверить правильность работы программы, выполнив указания из подраздела 3.3. • Получить графики зависимостей, указанных в подразделе 3.3. • Сравнить работу системы с разделением времени и системы с доступом по запросу
2	Необходимо: • написать программу имитационного моделирования для вероятностного алгоритма ALOHA. Рассмотреть два случая: a) Передача сообщений каждым абонентом всегда происходит с вероятностью p. b) Первая передача для каждого сообщения абонента происходит с вероятностью равной единице, в случае конфликта последующие попытки передачи сообщения происходят с вероятностью p.

Вариант	Задание
	• Проверить правильность работы программы, выполнив указания из подраздела 3.3. • Получить графики зависимостей, указанных в подразделе 3.3. • Сравнить работу рассматриваемых случаев вероятностного алгоритма ALOHA
3	Необходимо: • Написать программу имитационного моделирования для интервального алгоритма ALOHA. Рассмотреть два случая: a) Абонент для каждого сообщения случайным образом выбирает номер окна для передачи из интервала $\{1,2,\ldots,W\}$. b) Первая передача каждого сообщения абонента происходит в начале следующего окна, в случае конфликта при последующих передачах сообщения абонент случайно выбирает номер окна для передачи из интервала $\{1,2,\ldots,W\}$. • Проверить правильность работы программы, выполнив указания из подраздела 3.3. • Получить графики зависимостей, указанных в подразделе 3.3. • Сравнить работу рассматриваемых случаев интервального алгоритма ALOHA
4	Необходимо: • Написать программу имитационного моделирования для вероятностного адаптивного алгоритма ALOHA. Вероятность передачи определяется по правилу, описанному в 3.1.3. • Проверить правильность работы программы, выполнив указания из подраздела 3.3. • Получить графики зависимостей, указанных в подразделе 3.3. • Сравнить работу системы для разного числа абонентов в системе M
5	Необходимо: • Написать программу имитационного моделирования для интервального адаптивного алгоритма ALOHA. Интервал выбора окна для передачи определяется по правилу, описанному в 3.1.3. • Проверить правильность работы программы, выполнив указания из подраздела 3.3. • Получить графики зависимостей, указанных в подразделе 3.3. • Сравнить работу системы для разного числа абонентов в системе M
6	Необходимо: • Написать программу имитационного моделирования для вероятностного алгоритма двоичной экспоненциальной отсрочки. Вероятность передачи для каждого абонента определяется по правилу, описанному в 3.1.3. • Проверить правильность работы программы, выполнив указания из подраздела 3.3. • Получить графики зависимостей, указанных в подразделе 3.3. • Сравнить работу системы для разного числа абонентов в системе M

Вариант	Задание
7	Необходимо: • Написать программу имитационного моделирования для интервального алгоритма двоичной экспоненциальной отсрочки. Интервал выбора окна передачи для каждого абонента определяется по правилу, описанному в 3.1.3. • Проверить правильность работы программы, выполнив указания из подраздела 3.3. • Получить графики зависимостей, указанных в подразделе 3.3. • Сравнить работу системы для разного числа абонентов в системе M

Все описанные варианты могут рассматриваться как в случае неограниченной очереди, так и с ограниченной очередью. Тип очереди (ограниченная, не ограниченная) задается преподавателем и в случае ограниченной очереди, ее размер задается преподавателем.

3.5. Требования к отчету по лабораторной работе

Отчет должен содержать:

1. Титульный лист.
2. Цель и постановку задачи.
3. Описание моделируемой системы.
4. Описание исследуемого алгоритма доступа к среде (описание можно представить в виде псевдокода или блок-схемы).
5. Для вариантов 1,2 и 3 при неограниченном размере очереди необходимо вычислить значение $\lambda_{крит}$. Для других вариантов получить от преподавателя задание, результат выполнения которого представить в данном пункте отчета.
6. Графики зависимостей:

a) Средней задержки от интенсивности входного потока – $\bar{d}(\lambda)$;

b) Среднего числа абонентов в системе от интенсивности входного потока – $\bar{N}(\lambda)$;

c) Интенсивность выходного потока от интенсивности входного потока – $\lambda_{вых}(\lambda)$. Интенсивность выходного потока соответствует среднему количеству абонентов, выходящих в окне.

7. Выводы по полученным зависимостям;
8. Листинг программы имитационного моделирования присутствует только в электронной версии отчета.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. *Марковский С. Г.* Элементы теории помехоустойчивого кодирования: учеб. пособие / С. Г. Марковский, А. М. Тюрликов. – СПб.: ГУАП, 2014. – 95 с.

2. *Колесник В. Д.* Кодирование при передаче и хранении информации (Алгебраическая теория блоковых кодов): учеб. пособие для студентов высших учебных заведений / В. Д. Колесник. – М.: Высшая школа, 2009 – 549 с.

3. *Кудряшов Б. Д.* Основы теории кодирования: учеб. пособие для студентов высших учебных заведений / Б. Д. Кудряшов. – СПб.: БХВ-Петербург, 2016. – 393 с.

4. *Финк Л. М.* Теория передачи дискретных сообщений / Л. М. Финк. – М.: Советское радио, 1970. – 728 с.

5. *Уолрэнд Д.* Телекоммуникационные и компьютерные сети. Вводный курс. – М.: Постмаркет, 2001.

6. *Бертсекас Д.* Сети передачи данных / Д. Бертсекас, Р. Галлагер. – М.: Мир, 1989. – 544 с.

7. *Крук Е. А.* Основы теории кодирования: учеб. пособие / Е. А. Крук, А. А. Овчинников. – СПб.: Изд-во ГУАП, 2013. – 106 с.

8. *Тюрликов А. М.* Основы построения инфокоммуникационных систем и сетей: учеб. пособие: в 2 ч. Ч. 1 / А. М. Тюрликов, И. А. Пастушок, А. В. Борисовская. – СПб: Изд-во ГУАП, 2019. – 111 с.

СОДЕРЖАНИЕ

Учебное издание

Бурков Артём Андреевич,

Тюрликов Андрей Михайлович

ИЗУЧЕНИЕ ПРИНЦИПОВ ФУНКЦИОНИРОВАНИЯ КАНАЛЬНОГО УРОВНЯ СЕТЕЙ ПЕРЕДАЧИ ДАННЫХ

Учебно-методическое пособие

Публикуется в авторской редакции

Компьютерная верстка *С. Б. Мацапуры*

Подписано к печати 20.10.2025. Формат 60 × 84 1/16.
Усл. печ. л. 4,1. Уч.-изд. л. 4,2.
Тираж 50 экз. Заказ № 334.

Редакционно-издательский центр ГУАП

190000, г. Санкт-Петербург, ул. Большая Морская, д. 67, лит. А