МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное автономное
образовательное учреждение высшего образования

САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
АЭРОКОСМИЧЕСКОГО ПРИБОРОСТРОЕНИЯ

ОСНОВЫ ПОСТРОЕНИЯ ИНФОКОММУНИКАЦИОННЫХ СИСТЕМ И СЕТЕЙ

Лабораторный практикум

ГУАП
Санкт-Петербург
2018

Составители: А. А. Бурков, А. М. Тюрликов

Рецензент – кандидат технических наук *Е. А. Бакин*

Издание содержит три лабораторные работы, предназначенные для закрепления теоретического материала лекционного курса «Основы построения инфокоммуникационных систем и сетей». В тексте приводятся все необходимые сведения для выполнения лабораторных работ. 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 систем.

Публикуется в авторской редакции.
Компьютерная верстка *Н. Н. Караваевой*

Сдано в набор 02.12.18. Подписано к печати 26.12.18. Формат 60×84 1/16.
Усл. печ. л. 1,2. Тираж 50 экз. Заказ № .

Редакционно-издательский центр ГУАП
190000, Санкт-Петербург, Б. Морская ул., 67

ВВЕДЕНИЕ

Выполнение лабораторных работ можно условно разбить на 3 этапа: сдача допусков к лабораторной работе, выполнение лабораторной работы, сдача отчета. Описание всех лабораторных работ построено по общему принципу:

– первый раздел содержит краткие сведения с теоретической информацией, необходимой для выполнения лабораторной работы;

– второй раздел содержит список вопросов для допуска к лабораторной работе, каждый подраздел в данном разделе является отдельным вопросом;

– третий раздел описывает порядок выполнения лабораторной работы;

– четвертый раздел содержит список вариантов для выполнения лабораторной работы;

– пятый раздел описывает требования к оформлению отчета по лабораторной работе.

Дополнительный теоретический материал содержится: для первой лабораторной работы в [1–3]; для второй лабораторной работы в [4, 5]: для третьей лабораторной работы в [5, 6].

ЛАБОРАТОРНАЯ РАБОТА № 1. ИСПОЛЬЗОВАНИЕ ЦИКЛИЧЕСКИХ КОДОВ ДЛЯ ОБНАРУЖЕНИЯ ОШИБОК В СЕТЯХ ПЕРЕДАЧИ ДАННЫХ

Цель работы: исследование типового алгоритма формирования контрольной суммы с использованием циклических кодов, использование численного расчета и имитационного моделирования для оценки вероятности того, что декодер не обнаружит ошибки.

1. Теоретический материал для выполнения лабораторной работы

1.1. Модель системы, рассматриваемой в лабораторной работе

В большинстве современных систем передачи данных для обнаружения ошибок применяется следующий подход. К передаваемым данным добавляют контрольную сумму, которая вычисляется на основе этих же 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 в лабораторной работе системы передачи данных.

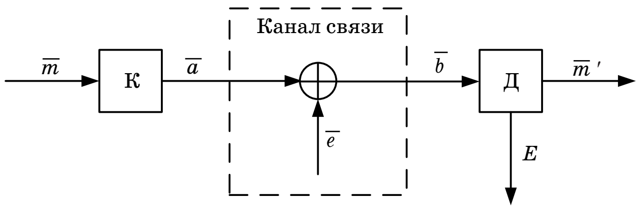

Рис. 1. Структурная схема системы передачи данных: $\overline{m}$ – информационное сообщение, К – блок кодера, $\overline{a}$ –закодированное сообщение, $\overline{e}$ –вектор ошибок, $\overline{b}$ – сообщение на выходе канала, Д – блок декодера, E –принятое решение, $\overline{m}'$ – сообщение на выходе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

На вход кодера поступает некоторое информационное сообщение $\overline{m}$, состоящее из нулей и единиц. Кодер по некоторому алгоритму вычисляет контрольную сумму, дописывает ее к передаваемому сообщению и таким образом формирует закодированное сообщение $\overline{a}$ так же состоящее из 0 и 1. В канале могут произойти ошибки, в результате которых некоторые биты сообщения инвертируются (0 становиться 1 или 1 становиться 0). Вектор ошибок показывает на каких позициях произошла ошибка, при этом канал может быть описан как операция XOR передаваемого сообщения и вектора ошибок. Пример: $\overline{a}=[101], \overline{e}=[001], \overline{b}=[100]$. Декодер по некоторому алгоритму проверяет контрольную сумму в принятом сообщении и принимает одно из следующих решений:

$$E=\begin{cases}1, \text{если были ошибки}\\0, \text{если не было ошибок}\end{cases}.$$

Рассматривается модель двоично-симметричного канала (ДСК) без памяти представленного на рис. 2.

Как видно из рисунка 2 с вероятностью p происходит ошибка (0 становится 1 или 1 становиться 0). Канал является двоичным, поэтому возможны только два значения битов на входе и выходе канала: {0,1}. Канал называется симметричным ввиду того, что вероятность ошибки для обоих значений битов одинакова. Модель ДСК приведена на рис. 2. Канал без памяти характеризуется тем, что случайные события, связанные с ошибками в канале независимы для разных моментов времени.

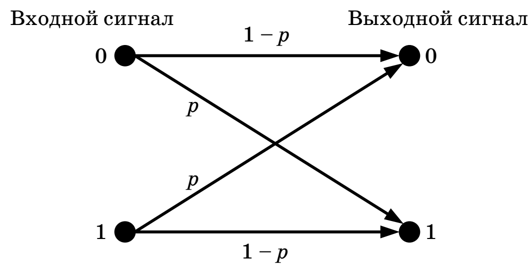

Рис. 2. Модель двоично-симметричного канала

1.2. Работа кодера и декодера

В лабораторной работе рассматриваются только двоичные коды. Для описания работы с двоичными кодами используются многочлены с коэффициентами из GF(2) (Galois field, поле Галуа или конечное поле). Кодер хранит порождающий многочлен $g(x)$. Степень многочлена обозначается как $\deg(g(x)) = r$ и определяет количество бит контрольной суммы в кодовом слове. k – число информационных символов передаваемого сообщения $\overline{m}$.

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

1. На основе вектора $\overline{m}$ формируется многочлен $m(x)$. Степень многочлена $m(x)$ при этом меньше или равна $k-1$;
2. Вычисляется многочлен $c(x) = m(x)x^r \bmod g(x)$. Степень многочлена $c(x)$ при этом меньше или равна $r-1$;
3. Вычисляется многочлен $a(x) = m(x)x^r + c(x)$;
4. На основе многочлена a(x) формируется вектор $\overline{a}$, длина которого n бит, где $n = k + r$.

Пример работы кодера:

0. $g(x) = x^3 + x + 1 \to \deg(g(x)) = 3$;

1. $\overline{m} = 1010 \to k = 4$;

$m(x) = 1x^3 + 0x^2 + 1x + 0x^0 = x^3 + x$;

2. $$\begin{array}{l|l} -\ x^6 + x^4 & x^3 + x + 1 \\ \underline{x^6 + x^4 + x^3} & x^3 + 1 \\ \quad -\ x^3 & \\ \quad \underline{x^3 + x + 1} & \\ \quad\quad x + 1 & \end{array}$$

$\to c(x) = (x^3 + x)x^3 mod(x^3 + x + 1) = x + 1$;

3. $a(x) = (x^3 + x)x^3 + x + 1 = x^6 + x^4 + x + 1$;

4. $a(x) = 1x^6 + 0x^5 + 1x^4 + 0x^3 + 0x^2 + 1x + 1x^0$;
$\to \overline{a} = 1010011,\ \ n = 4 + 3 = 7.$

Стоит отметить, что многочлен, полученный на третьем шаге описанного алгоритма для любого $m(x)$, делиться без остатка на порождающий многочлен $g(x)$, то есть $a(x) mod\, g(x) = 0$.

В соответствии с данным свойством может быть предложен следующий алгоритм декодирования. Декодер хранит $g(x)$ (порождающий многочлен) и n (длина кодового слова). Декодер выполняет следующие действия:

1. Принятое сообщение $\overline{b} = \overline{a} + \overline{e}$ переводится в многочлен $b(x)$;
2. Вычисляется синдром: $s(x) = b(x) mod\, g(x)$;
3. Если $s(x) \neq 0$, то декодер выносит решение, что произошли ошибки ($E = 1$), иначе декодер выносит решение, что ошибки не произошли ($E = 0$):

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

Пример работы декодера:

0. $g(x) = x^3 + x + 1;\ \overline{a} = 1010011;\ \overline{e} = 1100101$
1. $\overline{b} = 1010011 \oplus 1100101 = 0110110$

$b(x) = 0x^6 + 1x^5 + 1x^4 + 0x^3 + 1x^2 + 1x + 0x^0 = x^5 + x^4 + x^2 + x$;

2. $$\begin{array}{l|l} -\ x^5 + x^4 + x^2 + x & x^3 + x + 1 \\ \underline{x^5 + x^3 + x^2} & x^2 + x + 1 \\ \quad -\ x^4 + x^3 + x & \\ \quad \underline{x^4 + x^2 + x} & \\ \quad\quad -\ x^3 + x^2 & \\ \quad\quad \underline{x^3 + x + 1} & \\ \quad\quad x^2 + x + 1 & \end{array}$$

$\to s(x) = (x^5 + x^4 + x^2 + x) mod(x^3 + x + 1) = x^2 + x + 1$;

3. $s(x) \neq 0 \to E = 1 \to$ произошли ошибки;

При этом стоит отметить следующее:

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

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

Это означает следующее: если вектор ошибок принадлежит множеству разрешенных кодовых слов, то на выходе канала будет другое разрешенное кодовое слово (XOR двух кодовых слов дает кодовое слово), и, следовательно, декодер не обнаружит ошибки.

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.

Минимальным расстоянием кода d называется наименьшее число несовпадающих позиций между любыми кодовыми словами. Ранее описанная система кодирования и декодирования позволяет найти количество ошибок, не превышающее $d-1$ вне зависимости от того, где эти ошибки произошли и того, какое кодовое слово передавалось.

1.3. Вычисление верхней оценки ошибки декодирования

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:

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

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

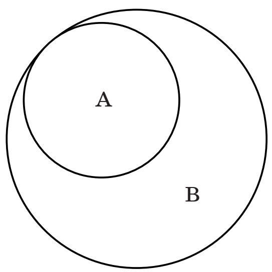

Рис. 3. Схематическое представление соотношения между множествами A и B : A – множество векторов ошибок при которых произошла ошибка декодирования; B – множество векторов ошибок $w(\overline{e}) \geq d$

где $w(\overline{e})$ – вес вектора ошибок. Для случая двоичных векторов ошибок вес равен количеству единиц в векторе. Пример: $\overline{e} = 100101 \to w(\overline{e}) = 3$.

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

Мощность множества В больше, чем мощность множества A, где под мощностью множества подразумевается число элементов (векторов ошибок), которое входит в множество. Предположим, что все вектора ошибок с весом $w(\overline{e}) \geq d$ приводят к ошибке декодирования. Используя данный подход, мы можем получить верхнюю границу для вероятности ошибки декодирования, для этого нужно найти $Pr\{B\}$ при этом $Pr\{B\} > Pr\{A\}$. Найдем $Pr\{B\}$:

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

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

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

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:

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

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 решении этой задачи рассмотрим каким образом с помощью кодера множество сообщений может быть отображено на множество кодовых слов. Схематично этот процесс представлен на рис. 4.

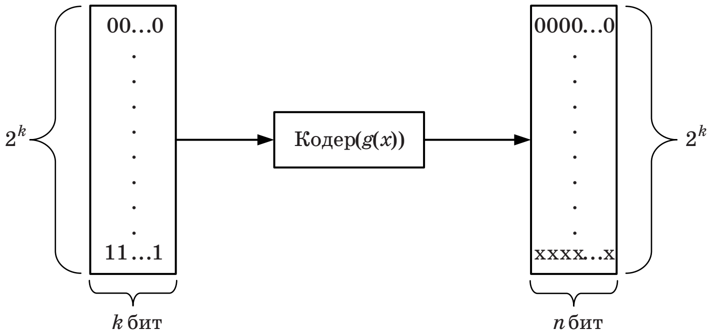

Рис. 4. Схематичное представление формирования множества кодовых слов

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

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

Пусть A – множество кодовых слов, $|A| = 2^k$; B –множество 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 из значения могут принимать A_i для кода с минимальным расстоянием d.

Ошибка декодирования происходит, если вектор ошибок является кодовым словом (доказательство приводится в курсе лекций). Все слова в коде имеют вес больше либо равный d. Поэтому гарантированно обнаруживаются ошибки число которых $< d$. Для точного определения значения вероятности ошибки декодирования следует посчитать вероятность попадания вектора ошибок в 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:

$$P_e = \sum_{i=d}^{n} A_i p^i (1-p)^{(n-i)}, \qquad (2)$$

где P_e – вероятность ошибки декодирования, d – минимальное расстояние кода, n – длинна кодового слова, p – вероятность ошибки на бит, A_i – количество кодовых слов с весом i (w(кодоводого слова) = i).

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, затем к нему добавляется контрольная сумма по алгоритму, описанному в разделе 1.2.

2) генерируется случайный вектор ошибок, где для каждой позиции случайно выбирается событие: 1) произошла ошибка с вероятностью p (бит вектора ошибок = 1); 2) ошибки не было с вероятностью $1-p$ (бит вектора ошибок = 0). Вектор ошибки складывается с результатом кодирования (операция XOR) для получения выхода канала.

3) вычисляется синдром, и определяется, были ошибки или нет.

Данная процедура повторяется N раз. Если в результате моделирования N_e раз произошли ошибки декодирования $(\overline{e} \neq 0, E = 0)$, то вероятность ошибки декодирования вычисляется по следующей формуле $\widehat{P_e} = \frac{N_e}{N}$. Тогда $\varepsilon = \left|P_e - \widehat{P_e}\right|$, где P_e – теоретическая вероятность декодирования, определяемая по формуле (2).

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.

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:

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 – длинна кодового слова.

Получив вариант с исходными данными от преподавателя и информационную последовательность $\overline{m}$ из k символов необходимо ответить на вопросы:

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 равен 0 (ошибки не обнаружены).
2. Привести пример работы декодера для случая, когда в канале произошло t ошибок при этом $1 \le t \le d-1$.
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. Показать на конкретном примере, что возможен случай, когда в канале произойдет меньше чем d ошибок и такие ошибки не будет обнаружены.

3. Порядок выполнения лабораторной работы

1. Необходимо в качестве допусков к лабораторной работе ответить на вопросы в разделе 2 и сдать их преподавателю.
2. После сдачи всех допусков получить от преподавателя вариант задания для лабораторной работы.

3. Написать и отладить моделирующую программу в соответствии с полученным вариантом.
4. Продемонстрировать работу моделирующей программы преподавателю.
5. Получить дополнительное задание для исследования.
6. В соответствии с вариантом и полученным дополнительным заданием оформить и сдать отчет.

4. Варианты заданий для лабораторной работы

1. Требуется разработать программу, наглядно демонстрирующую работу кодера и декодера для типового алгоритма формирования циклических кодов. На вход программы подается порождающий многочлен $g(x)$, число k, вектор ошибки $\overline{e}$ и информационная последовательность из l бит (l может быть как меньше, так и больше k). Программа формирует кодовое слово. На основе него и вектора ошибки формируется последовательность на выходе канала. По принятой последовательности принимается решение о наличии 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 для первого варианта

а) Исследовать альтернативную реализацию алгоритма декодирования. Последовательность $\overline{b}$ на выходе канала (см. рис. 6) делится на две части. Первая последовательность $\overline{m}_b$ содержит в себе символы, относящиеся к информационной части. Вторая $\overline{c}_b$ содер-

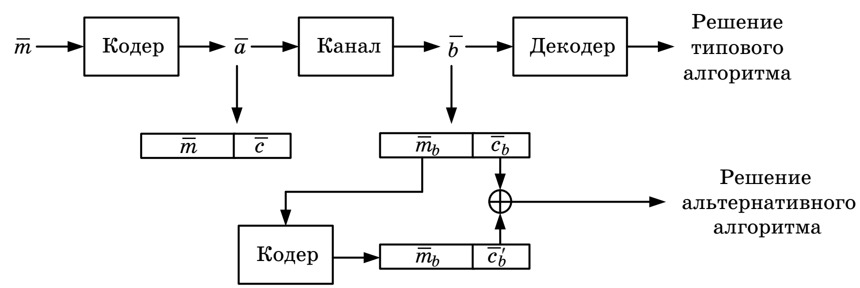

Рис. 6. Схема альтернативного алгоритма декодирования

жит 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, альтернативный не обнаруживает и наоборот. Если такие ситуации существуют, привести примеры, в противном случае обосновать невозможность их возникновения.

b) Исследовать работу кодера/декодера для $g(x)$ с параметрами n, k и d, когда выбрано некорректное число информационных символов $(l+k)$. Для заданной преподавателем 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. Если таких примеров нет, то обосновать почему.

2. Разработать программу, с помощью которой путем имитационного моделирования оценивается вероятность ошибки декодирования при передаче данных по двоично-симметричному каналу. Исходными данными для работы программы являются: порождающий многочлен $g(x)$, длина кодируемой последовательности l (может быть как больше, так и меньше k) и 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 при различных значениях l.

Список дополнительных заданий для второго варианта

a) Построить график зависимости оценки ошибки декодирования $\widehat{P_e}$ от вероятности ошибки в двоично-симметричном канале p при $l<k$, $l>k$, $l=k$. Обосновать полученные зависимости.

b) В исходном варианте моделирования входная последовательность генерируется случайным образом. Провести моделирование для случая, когда входная последовательность зафиксирована. Сравнить зависимость оценки ошибки декодирования $\widehat{P_e}$ от вероятности ошибки в двоично-симметричном канале p с исходным вариантом. Повторить моделирование для другой входной последова-

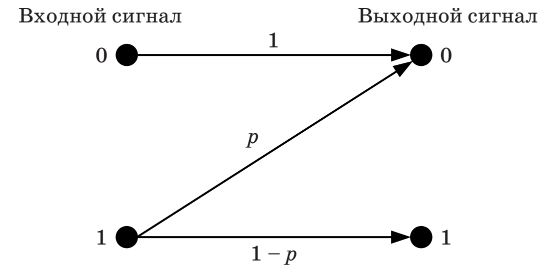

Рис. 7. Измененная модель канала

тельности. Выяснить, каким образом влияет изменение входной последовательности на оценку вероятности ошибки декодирования.

c) Исследовать как изменится зависимость оценки ошибки декодирования $\widehat{P_e}$ от вероятности ошибки в канале p при смене модели канала как показано на рис. 7. Обосновать полученный результат.

d) Разработать программу, реализующую альтернативный алгоритм декодирования, описанный в дополнительном задании a) для первого варианта (рис. 6) и сравнить его с типовым вариантом декодирования (построить графики зависимости оценки ошибки декодирования $\widehat{P_e}$ от вероятности ошибки в канале p). Выяснить, влияет ли изменение алгоритм декодирования на вероятность ошибки декодирования.

3. Разработать программу вычисления верхней оценки для вероятности ошибки декодирования сверху (см. раздел 1.3) и вычисления точного значения вероятности ошибки декодирования (см. раздел 1.4). Исходные данные и исследования такие же, как во втором варианте.

Список дополнительных заданий для третьего варианта

a) Построить зависимости верхней оценки вероятности ошибки декодирования $\widehat{P_e}$ и точной вероятности ошибки декодирования P_e при различных порождающих многочленах $g(x)$. Исследовать, как влияет изменение порождающего многочлена на изменение вероятности ошибки декодирования.

b) Построить зависимости верхней оценки вероятности ошибки декодирования $\widehat{P_e}$ и точной вероятности ошибки декодирования P_e при $l<k$, $l>k$, $l=k$. Обосновать полученные зависимости.

c) Построить зависимости верхней оценки вероятности ошибки декодирования $\widehat{P_e}$ и точной вероятности ошибки декодирования P_e при различных l и фиксированной вероятности ошибки в ДСК канале $p = 0.1, p = 0.2, p = 0.3$. Обосновать полученные зависимости.

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 моделирующей программы в виде псевдокода или блок-схемы.
6. Описание результатов проводимых исследований и зависимостей.
7. Выводы по проводимым исследованиям.
6. Листинг моделирующей программы, присутствует только в электронной версии отчета.

ЛАБОРАТОРНАЯ РАБОТА № 2. ПЕРЕДАЧА ДАННЫХ В СИСТЕМАХ С ОБРАТНОЙ СВЯЗЬЮ

Цель работы: исследование типовых алгоритмов передачи данных в системах с решающей обратной связью, использование численного расчета и имитационного моделирования для оценки коэффициента использования канала в рамках базовой модели для систем с решающей обратной связью.

1. Описание модели системы и алгоритмов передачи рассматриваемых в лабораторной работе

1.1. Базовая модель систем с обратной связью

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

Введем следующие допущения:

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

Рис. 8. Схема передачи данных по каналу с обратной связью

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.

3) При передаче квитанции по обратному каналу, может произойти ошибка с вероятностью $p_{\text{обр}}$. Ошибки при передаче квитанции всегда обнаруживаются (в случае ошибки источник не знает, какая квитанция передавалась, при этом положительная квитанция не может стать отрицательной и наоборот).

4) Все сообщения, которые передает источник, имеют одинаковую длину. Время передачи сообщения принято за единицу времени, а время передачи квитанции считается равным нулю (см. рис. 9). Источник получает квитанцию о результате передачи через τ единиц времени после окончания передачи сообщения, где τ – целое число.

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

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

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

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

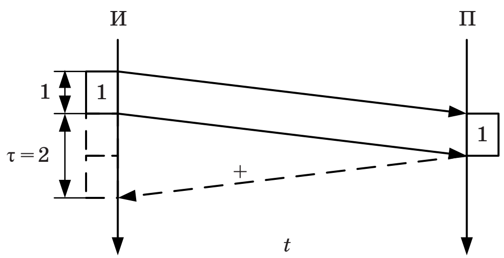

Рис. 9. Пример работы системы в соответствии с допущением 4 при $\tau = 2$

1.2. Алгоритм с ожиданием

Алгоритм с ожиданием работает следующим образом. Источник передает сообщение, затем ждет квитанцию с подтверждением. В лабораторной работе предполагается, что время ожидания квитанции (задержка получения квитанции) постоянно и равно τ единицам времени, согласно допущению 4. Если источник принимает отрицательную квитанцию, то он повторяет передачу сообщения. Если получена положительная квитанция, то передается следующее сообщение. Пример работы алгоритма с ожиданием при $\tau = 2$ приведен на рис. 10.

В данной лабораторной работе рассматриваются 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

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

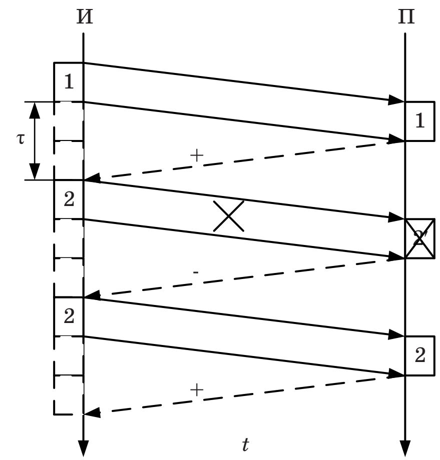

Рис. 10. Пример работы алгоритма с ожиданием при $\tau = 2$

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 квитанции равной τ единицам времен, 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 и τ определено в соответствии с допущением 4. Пример работы алгоритма с возвратом при $\tau = 2$ приведен на рис. 11.

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

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

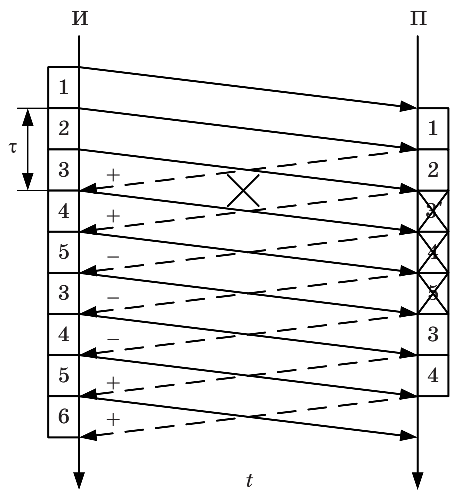

Рис. 11. Пример работы алгоритма с возвратом при $\tau = 2$

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 те сообщения, которые были приняты с ошибками, либо те, которые 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, все они передаются на вышележащий уровень. 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. 12.

При данном методе передачи возможны различные подходы работы 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. 13.

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. 14.

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

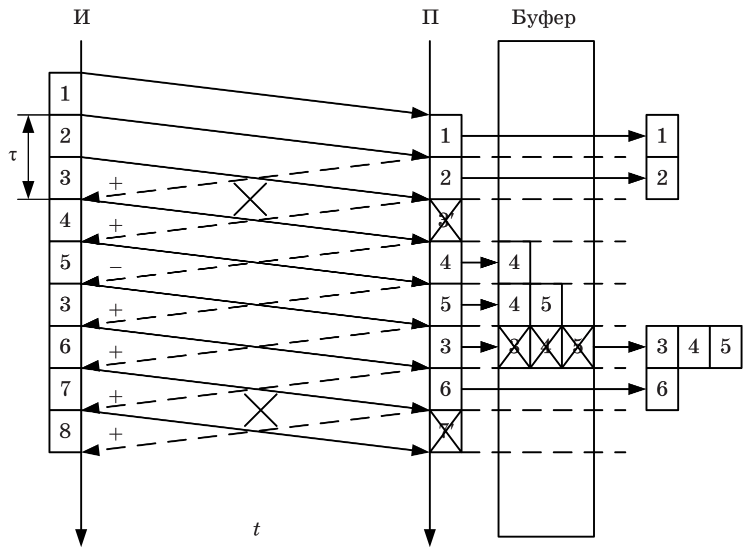

Рис. 12. Пример работы алгоритма с селективным повторением при $\tau = 2$

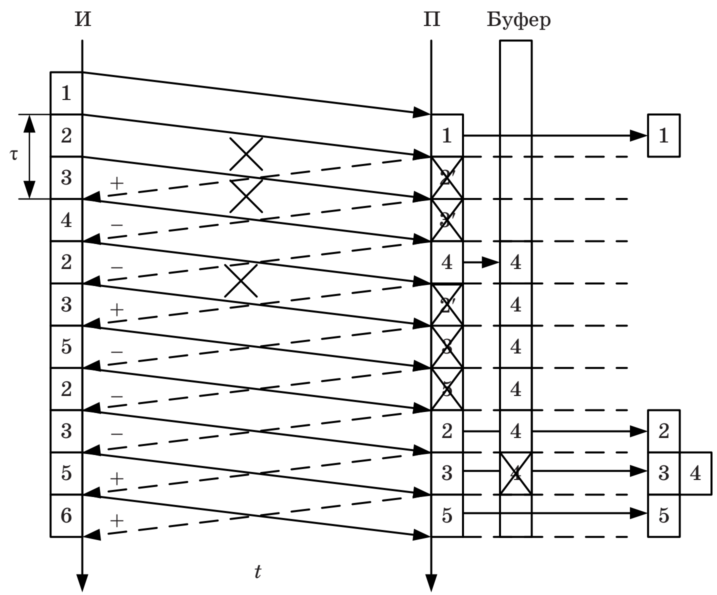

Рис. 13. Работа «алгоритма» с селективным повторением при $\tau = 2$ *и* $L = 1$ *для первой стратегии.*

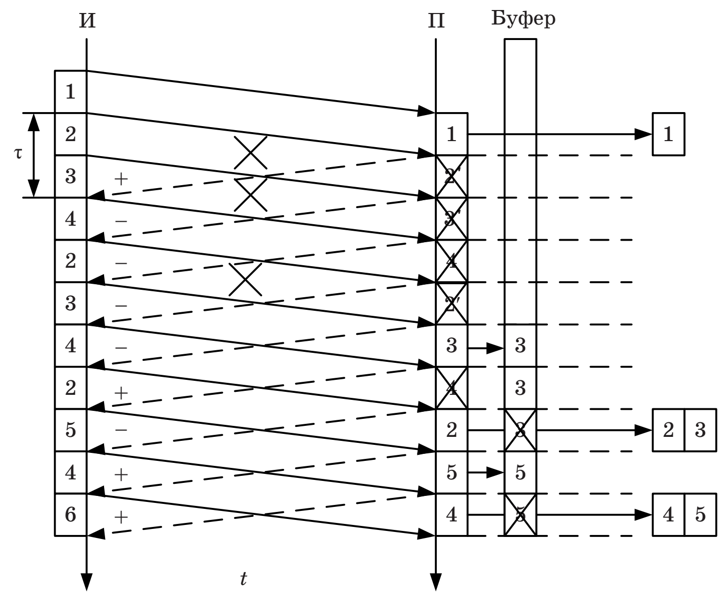

Рис. 14. Работа «алгоритма» с селективным повторением при $\tau = 2$ *и* $L = 1$ *для второй стратегии*

1.5. Алгоритм с виртуальными каналами

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

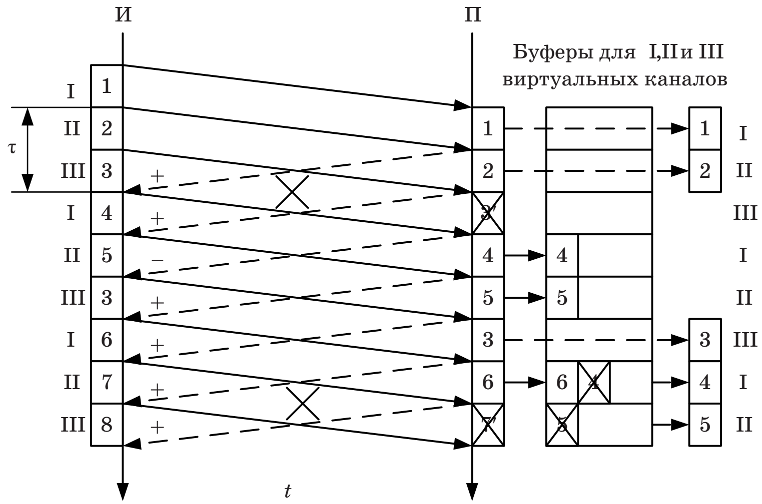

Рис. 15. Пример работы алгоритма с виртуальными каналами при $\tau = 2$ *и тремя виртуальными каналами*

с требуемым порядком. Пример работы алгоритма с виртуальными каналами при $\tau = 2$ (при этом виртуальных каналов три) приведен на рис. 15.

Коэффициент использования канала для алгоритма с виртуальными каналами при бесконечной длине буфера определяется по формуле $(1-p)$. Стоит заметить, что коэффициент использования канала (η) в данном случае не зависит от задержки в получении квитанции (τ).

1.6. Алгоритм для высокой вероятности ошибки в канале

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 единицам времен, 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 положительной квитанции, в количестве τ штук на приемнике удаляются

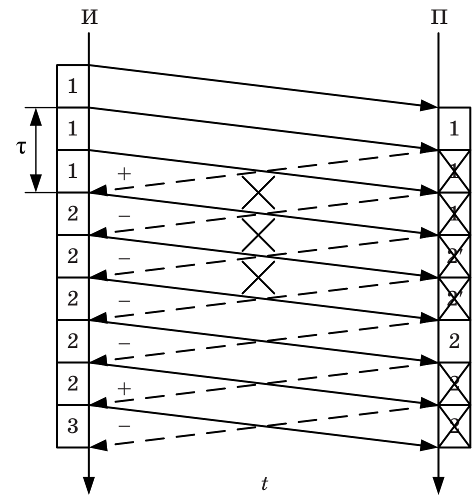

Рис. 16. Пример работы алгоритма для высокой вероятности ошибки при $\tau = 2$

независимо от успешности приема. Пример работы алгоритма для высокой вероятности ошибки при $\tau = 2$ приведен на рис. 16.

Коэффициент использования канала, при больших значениях вероятности ошибки в канале, для 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. Вопросы для допуска к лабораторной работе

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

Рассмотрим алгоритм с ожиданием для системы с $p_{\text{обр}} = 0$ и неограниченном числе повторных передач сообщения, до получения положительной квитанции. Написать моделирующую программу для алгоритма с ожиданием, когда $p_{\text{обр}} = 0$ и число повторных передач сообщения до получения положительной квитанции не ограничено.

Результат моделирования сравнить с расчетом. Для этого самостоятельно вывести формулу для 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:

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

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

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

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

Написать моделирующую программу алгоритма с ожиданием для системы с $p_{\text{обр}}=0,$ где число повторных передач сообщения до получения положительной квитанции ограниченно n.

Результат моделирования сравнить с расчетом. Для этого самостоятельно вывести формулу для расчета среднего числа повторных передач до получения положительной квитанции, как функцию от вероятности ошибки в прямом канале и параметра $n-\bar{N}(p,n)$. 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:

$$Pr\{N=1\}=1-p,$$
$$\ldots,$$
$$Pr\{N=n\}=\ldots,$$
$$Pr\{N\geq n+1\}=0.$$

Самостоятельно выписать выражения вероятностей для данной системы. Для вывода воспользоваться тем, что

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

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

Написать моделирующие программы для алгоритма с ожиданием при неограниченном и ограниченном числе повторных передач для системы с $p_{\text{обр}}\neq 0$.

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.

Результат моделирования сравнить с расчетами. Для этого самостоятельно вывести формулы для среднего числа повторных передач, как в случае с неограниченным числом повторных передач, так и с ограниченным числом передач (аналогично разделам 2.1 и 2.2).

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

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 – $\bar{N}(p,p_{\text{обр}},n)$.

Для данной системы можно выписать следующий набор вероятностей:

$$Pr\{N=1\}=(1-p)(1-p_{\text{обр}}),$$
$$Pr\{N=2\}=?$$
$$\ldots$$

Самостоятельно выписать вероятности и вывести итоговые формулы для среднего числа повторных передач для неограниченного и ограниченного числа повторных передач.

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

Рассматривается алгоритм с ожиданием для системы с $p_{\text{обр}} = 0$. Самостоятельно написать моделирующую программу для алгоритма с ожиданием. Программа должна быть написана таким образом, что бы была возможность задавать любые значения задержки получения квитанции (τ) и вероятности ошибки в прямом канале (p). Необходимо вывести лог программы, по которому нарисовать временную диаграмму работы алгоритма. Для предварительной проверки правильности работы моделирующей программы, необходимо убедиться, что результаты, полученные моделированием, совпадают с формулой

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

2.5. Моделирование алгоритма с возвратом

Необходимо написать моделирующую программу для алгоритма с возвратом, описанного в 1.3 для системы с $p_{\text{обр}} = 0$. Программа должна быть написана таким образом, что бы была возможность задавать любые значения задержки получения квитанции (τ) вероятности ошибки в прямом канале (p). Необходимо вывести лог программы, по которому нарисовать временную диаграмму работы алгоритма. Для предварительной проверки правильности работы моделирующей программы, необходимо убедиться, что результаты, полученные моделированием, совпадают с формулой

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

3. Порядок выполнения лабораторной работы

1) Необходимо в качестве допусков к лабораторной работе ответить на вопросы в разделе 2 и сдать их преподавателю.

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

3) Написать и отладить моделирующую программу 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: $\eta(p, \tau = \text{const})$;

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

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

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

5) Продемонстрировать работу моделирующей программы преподавателю.

6) Получить дополнительное задание для исследования.

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

4. Варианты заданий для лабораторной работы

1. Написать моделирующую программу для алгоритма с селективными повторениями, описанного в 1.4. Размер буфера на приемнике равен 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.

2. Написать моделирующую программу для алгоритма с селективными повторениями, описанного в 1.4. Размер буфера на приемнике равен 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.

3. Написать моделирующую программу для алгоритма с селективными повторениями, описанного в 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.

4. Написать моделирующую программу для алгоритма с селективными повторениями, описанного в 1.4. Размер буфера на приемнике $L>1$. Стратегия работы с буфером: в буфер записываются только сообщения с номерами $i+1,\ i+1,\ \ldots,\ i+L,$ при ожидании сообщения с номером i. Сообщения выдаются пользователю только в соответствии с правильным порядком следования, определяемого нумерацией на источнике.

5. Написать моделирующую программу для алгоритма с виртуальными каналами, описанного в 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.

6. Написать моделирующую программу для алгоритма с виртуальными каналами, описанного в 1.5. Размер буфера на приемнике для каждого виртуального канала имеет свое ограничение на длину $(L_1, L_2, \ldots, L_{\tau+1})$. Сообщения выдаются пользователю только в соответствии с правильным порядком следования, определяемого нумерацией на источнике.

7. Написать моделирующую программу для алгоритма с виртуальными каналами, описанного в 1.5. Размер буфера на приемнике имеет общее ограничение и делится между всеми виртуальными каналами. Сообщения выдаются пользователю только в соответствии с правильным порядком следования, определяемого нумерацией на источнике.

8. Написать моделирующую программу для алгоритма при высокой вероятности ошибки, описанного в 1.6. Программа должна быть написана как для случая, когда нет ограничения на количество повторных передач, так и для числа повторных передач ограниченного n. Сравнить алгоритмы с возвратом и при высокой вероятности ошибки для ограниченного и неограниченного числа повторных передач.

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: $\eta(p, \tau = \mathrm{const})$;

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

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

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

6. Зависимости, полученные для $\eta(p, \tau = const)$ и $\eta(p = const, \tau)$, 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 моделирующей программы, присутствует только в электронной версии отчета.

ЛАБОРАТОРНАЯ РАБОТА № 3.
МЕТОДЫ УПРАВЛЕНИЯ ДОСТУПОМ К СРЕДЕ

Цель работы: исследование типовых методов и алгоритмов разделения общего ресурса канала между абонентами, определение характеристик рассматриваемых алгоритмов в рамках базовой модели множественного доступа с использованием численных расчетов и имитационного моделирования

1. Описание моделей систем изучаемых в лабораторной работе

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). Интенсивность входного потока у всех абонентов в системе одинакова и у каждого абонента она равна $\dfrac{\lambda}{M}$.

Пример работы доступа с разделением времени для 4 абонентов приведен на рис. 17. Как видно из рисунка, у первого абонента A_1 появилось первое сообщение $A_1(1)$ во время своего окна, так как абоненты могут передавать только в начале своего окна, то первый абонент ждет начала следующего своего окна для передачи сообщения. У четвертого абонента A_4 к началу своего окна имеется готовое к передаче сообщение $A_4(1)$, поэтому он передает его. Аналогично A_4 передает свое сообщение A_2.

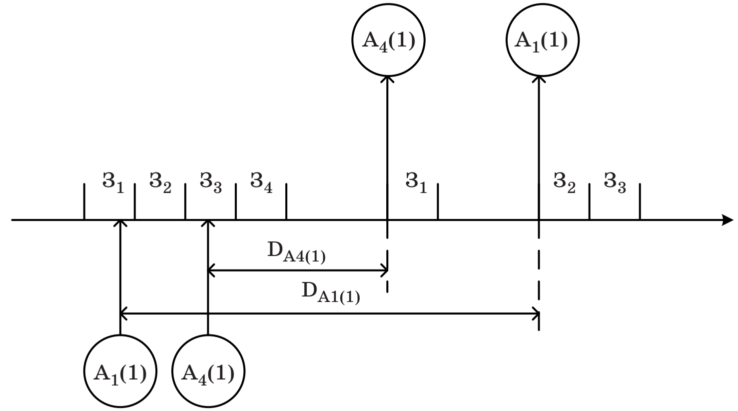

Рис. 18. Пример работы системы с доступом по запросу для 4 абонентов

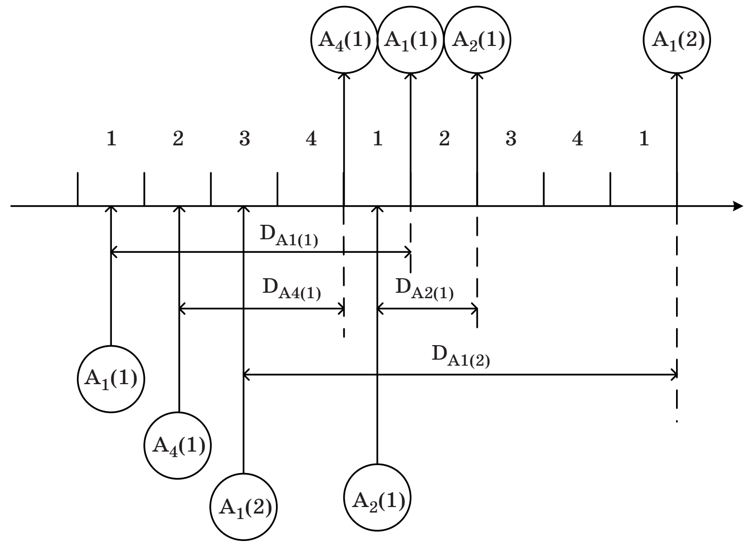

Рис. 17. Пример работы системы с разделением времени для 4 абонентов

1.2. Доступ по запросу

Доступ по запросу относиться 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). Интенсивность входного потока у всех абонентов в системе одинакова и у каждого абонента она равна $\dfrac{\lambda}{M}$.

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 получается 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 передает сообщение. Запросы к абонентам чередуются по порядку.

Пример работы системы с доступом по запросу для 4 абонентов приведен на рис. 18. Как видно из рисунка, у первого абонента A_1 появилось сообщение $A_1(1)$ во время своего окна запроса, согласно допущениям, он не может ответить на запрос и должен ожидать начала следующего своего окна запроса. A_4 к началу своего окна запроса имеет готовое к передаче сообщение $A_4(1)$, поэтому он отвечает на него и ему выделяется окно для передачи сообщения, в котором он передает.

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. Абонент передает сообщение по каналу сообщение готовое к передаче, если по каналу передаются сразу несколько сообщений от разных абонентов, то возникает конфликт, который разрешается по некоторому алгоритму.

Введем ряд допущений для рассматриваемой модели случайного множественного доступа:

1) Предполагается, что все сообщения у всех абонентов имеют одинаковую длину, 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.

2) В окне возможно 3 события:

А) 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.

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

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

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

4) В системе имеется M абонентов. В среднем у всех абонентов в одну единицу времени возникает λ сообщений (интенсивность входного потока) (Пуассоновский входной поток с параметром λ). Интенсивность входного потока у всех абонентов в системе одинакова и у каждого абонента она равна λ / 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. В вероятностном варианте алгоритма абоненты, имеющие к передаче сообщение, всегда передают его в начале очередного окна с вероятностью p. В интервальном варианте алгоритма каждый абонент случайным образом выбирает номер окна, начиная отсчет с текущего, в котором будет предавать из интервала $\{1, 2, \ldots, W\}$.

2) Адаптивный алгоритм ALOHA. Вероятность передачи сообщения абонентами в канал изменяется в зависимости от событий,

которые произошли в канале по следующему правилу для вероятностного варианта:

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

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

При интервальном варианте алгоритма абонент выбирают номер окна, в котором он будет передавать из интервала $\{1, 2, \ldots, W_t\}$, при этом параметр W_t изменяется в соответствии с событиями в канале по следующему правилу:

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

где W_t – размер интервала, в котором абонент выбирает окно дляпередачи в t-ом 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). В вероятностном варианте каждый абонент меняет вероятность передачи в соответствии с событием, которое произошло в канале при его передаче по следующему правилу:

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

где p_t – вероятность, с которой абонент передавал при t-ой передаче; $p_{\min}$ – минимальная вероятность передачи; $p_{\max}$ – максимальная вероятность передачи.

При интервальном варианте каждый абонент меняет интервал, в котором он выбирает окно для передачи в соответствии с событием, которое произошло при его передаче по следующему правилу:

$$W_{t+1} = \begin{cases} \min(2W_t, W_{\max}), \text{ при «конфликте» в канале} \\ W_{\min}, \text{ при «успехе» в канале} \end{cases},$$

где W_t –интервал, в котором выбиралось окно для t-ой 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 дерева 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. 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. Пример работы древовидного алгоритма для конфликта кратности 3 приведен на рис. 19.

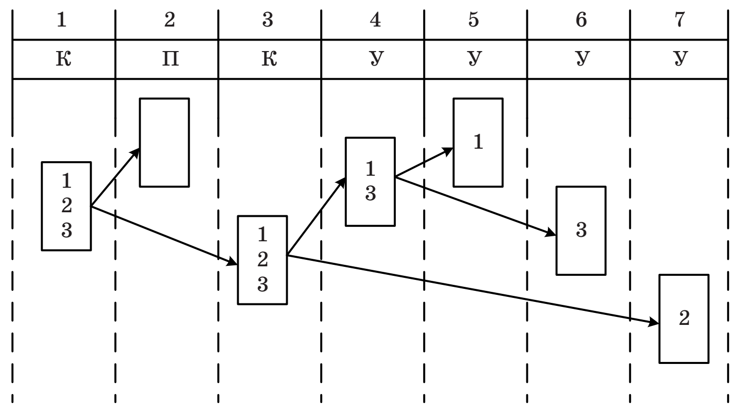

Рис. 19. Пример работы древовидного алгоритма для конфликта кратности 3

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.

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

Рассматривается система массового обслуживания (СМО) $M|D|1$, что можно расшифровать следующим образом: M – система имеет входной пуассоновский поток заявок с интенсивностью λ. D – система имеет детерминированное (фиксированное) время обслуживания. 1 –система имеет одно устройство обслуживания. Так же рассматриваемая система имеет одну очередь, время обслуживания равняется единице. Данная СМО изображена на рис. 20.

Введем в рассмотрение случайную величину D – интервал времени от момента появления сообщения до момента, когда сообщение доставлено получателю. Среднюю задержку можно вычислить как:

$$d = M[D].$$

Рассматриваются СМО $M|D|1$ двух типов: синхронная и асинхронная. В синхронной СМО время дискретно и разбито на окна единичной длинны, сообщения может быть переданы только в начале следующего окна в порядке очереди. В асинхронной СМО сообщения могут быть переданы как только появились в системе, в порядке очереди.

Для обеих систем среднее число абонентов в системе определяются в соответствии с выражением: $\bar{N}(\lambda) = M[N(\lambda)] = \dfrac{\lambda(2-\lambda)}{2(1-\lambda)}$ (рис.21).

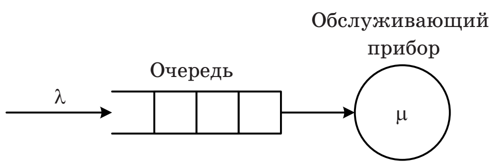

Рис. 20. Рассматриваемая система массового обслуживания: λ – интенсивность входного потока; μ – время обслуживания ($\mu = 1$)

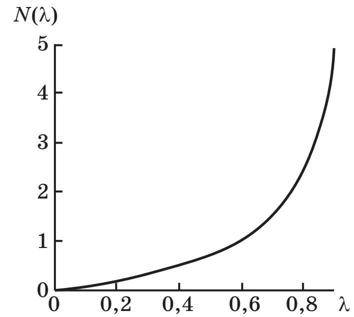

Рис. 21. Среднее число абонентов в системе для СМО $M|D|1$

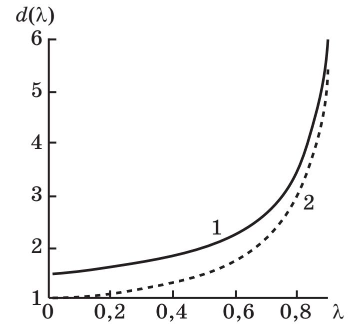

Рис. 22. Средняя задержка для СМО $M|D|1$: 1 – синхронная; 2 – асинхронная

Средняя задержка может быть определена как (рис. 22):

$$d(\lambda) = M[D(\lambda)] = \begin{cases} \dfrac{\bar{N}(\lambda)}{\lambda}, \text{ для асинхронной} \\ \dfrac{\bar{N}(\lambda)}{\lambda} + 0{,}5, \text{ для синхронной} \end{cases}.$$

Для простых систем (например, рассматриваемые в этом разделе 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 $M|D|1$ $\lambda_{кр} = 1$). Получение $d(\lambda)$ будем называть точным анализом системы. Для более сложных систем найти данное выражение затруднительно, однако почти для любой системы можно найти в явном виде выражение $d_0 \triangleq \lim\limits_{\lambda \to 0}\left(d(\lambda)\right)$. Нахождение d_0 и $\lambda_{кр}$ без получения явного выражения $\mathrm{d}(\lambda)$ будем называть анализом системы на качественном уровне.

2. Вопросы для допуска к лабораторной работе

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

2.1.1. 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. 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	$A_{2,1}$		$A_{3,1}, A_{4,2}$	$A_{3,2}$	$A_{3,3}$	
7		$A_{1,1}$		$A_{4,1}$	$A_{3,1}, A_{2,1}$	$A_{4,2}$

где $A_{i,j}$ обозначает j-ое сообщение i-го абонента. Столбец показывает, в каком временном окне появилось сообщение (порядковый номер окна при работе системы).

2.1.2. Доступ по запросу

Необходимо нарисовать временную диаграмму работы такой системы в соответствии с полученным вариантом. В системе рассматривается 4 абонента. Работа системы начинается с запроса для первого абонента.

Вариант	1 запрос	2 запрос	3 запрос	4 запрос	5 запрос	6 запрос
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	$A_{4,1}$		$A_{4,2}$			$A_{2,1}$
5	$A_{2,1}$	$A_{1,1}, A_{4,1}$				
6			$A_{1,1}, A_{4,1}$		$A_{2,1}$	
7	$A_{2,1}$		$A_{2,2}$	$A_{23,1}$		

где $A_{i,j}$ обозначает j-ое сообщение i-го абонента. Столбец показывает во время какого запроса появилось сообщение (порядковый номер окна запроса при работе системы).

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

2.2.1. Доступ с разделением времени

Необходимо найти среднюю задержку (d_0) системы, рассмотренной в 1.1. d_0 –это средняя задержка у абонента в пустой системе (см. раздел 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_2 – ожидание своего окна, D_3 – передача сообщения. Средняя задержка d_0 в такой системе вычисляется по формуле:

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

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

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

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

2.2.2. Доступ по запросу

Необходимо найти среднюю задержку (d_0) для системы, рассмотренной в 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

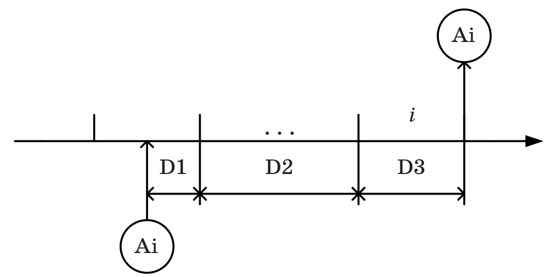

Рис. 23. Задержка для i-го абонента в пустой системе с разделением времени

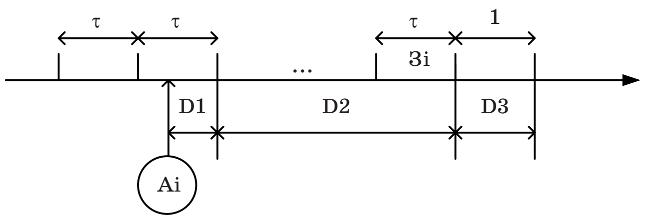

Рис. 24. Задержка для i-го абонента в пустой системе с доступом по запросу

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 своего окна запроса, D_3 – передача сообщения. Средняя задержка 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 $M[D_2]$, с учетом того, что:

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

2.2.3. 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.

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

Рассматривается простейший алгоритм случайного множественного доступа – алгоритм ALOHA. Абонент, имеющий готовое для передачи сообщение, передает его в начале очередного окна с вероятностью $p < 1$.

Пусть в системе n абонентов, имеющих готовое для передачи сообщение. 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:

1) 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.

2) Получить выражение для $Pr\{П \mid n\}$ – вероятность события, что в окне будет пусто, при условии, что n абонентов и для $Pr\{К \mid n\}$ – вероятность события, что в окне будет конфликт, при условии, что n абонентов.

3) Если зафиксирован алгоритм разрешения конфликта, то средняя задержка $(d = M[D])$ зависит от интенсивности входного потока.

Скорость алгоритма определяется как:

$$R \triangleq sup\{\lambda : d(\lambda) < \infty\}.$$

Скорость алгоритма ALOHA для модели с конечным числом абонентов.

$$R_{ALOHA} = Mp(1-p)^{M-1}$$

Чтобы скорость алгоритма была максимальной: $p = \frac{1}{M}$. В этом случае:

$$R_{ALOHA} = (1 - \frac{1}{M})^{M-1}.$$

Доказать, что при $M \to \infty : R_{ALOHA} \approx e^{-1}$.

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 передает сообщение в начале окна с вероятностью p и не передает с вероятностью $1 - p$. Задержка (D) в такой системе равна времени от момента появления абонента в системе до окончания окна, в котором сообщение абонента будет успешно передано.

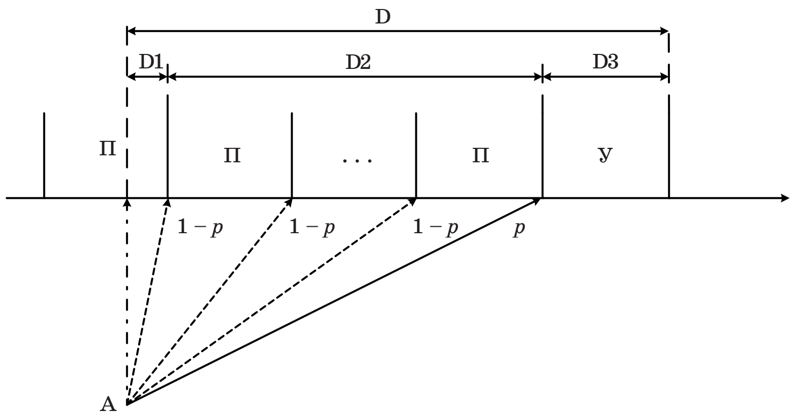

Рис. 25. Задержка алгоритма ALHOHA для единственного в системе абонента, передающего с вероятностью p

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. Средняя задержка d_0 (см. раздел 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:

$$D_2 \in \{0,1,\ldots,\infty\}$$

и

$$Pr\{D_2 = 0\} = p,$$
$$Pr\{D_2 = 1\} = p(1-p),$$
$$Pr\{D_2 = 2\} = ?,$$
$$\ldots$$

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

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

2.6. Практическое задание для допуска к лабораторной работе

Необходимо смоделировать синхронную и асинхронную системы массового обслуживания. Оценить с помощью моделирующей программы $M[D]$ и $M[N]$ для каждой из систем и сравнить с теоретическими значениями из раздела 1.4.

3. Порядок выполнения лабораторной работы

1. Необходимо в качестве допусков к лабораторной работе ответить на вопросы в разделе 2 и сдать их преподавателю.
2. После сдачи всех допусков получить от преподавателя вариант задания лабораторной работы.
3. Написать и отладить моделирующую программу 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 моделирующей программы преподавателю.
6. Получить дополнительное задание для исследования.
7. В соответствии с вариантом и полученным дополнительным заданием оформить и сдать отчет.

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

1) Во всех вариантах число абонентов (M) задается равным единице. Для случайного доступа вероятность передачи задается равной единице. Для доступа по запросу параметр $\tau = 0$. Необходимо построить $\bar{d}(\lambda)$ и $\bar{N}(\lambda)$, результаты должны совпадать с результатами для синхронной системы, описанной в 2.6. Если результаты не совпадают, то моделирующая программа работает неверно. Если результаты совпадают, то это не гарантирует правильность работы моделирующей программы при $M > 1$.

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

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 отличаются только правилом доступа к общему ресурсу канала. Необходимо написать моделирующую программу в соответствии с полученным вариантом. Моделирующая программа 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 системой $M|D|1$.

1. Написать моделирующую программу для систем с разделением времени и доступом по запросу. В моделирующей программе для доступа по запросу также необходима возможность задания τ.

2. Написать моделирующую программу для вероятностного алгоритма ALOHA. Рассмотреть два случая:

a) Передача всегда происходит с вероятностью p.

b) Первая передача сообщения происходит с вероятностью равной единице, последующие передачи сообщения происходят с вероятностью p.

3. Написать моделирующую программу для интервального алгоритма ALOHA. Рассмотреть два случая:

a) Абонент случайным образом выбирает номер окна для передачи из интервала $\{1,2,\ldots,W\}$.

b) Первая передача сообщения происходит в начале следующего окна, при последующих передачах сообщения абонент случайно выбирает номер окна из интервала $\{1,2,\ldots,W\}$.

4. Написать моделирующую программу для вероятностного адаптивного алгоритма ALOHA. Вероятность передачи определяется по следующему правилу, описанному в 1.3.

5. Написать моделирующую программу для интервального адаптивного алгоритма ALOHA. Интервал выбора окна для передачи определяется по правилу, описанному в 1.3.

6. Написать моделирующую программу для вероятностного алгоритма двоичной экспоненциальной отсрочки. Вероятность передачи для каждого абонента определяется по правилу, описанному в 1.3.

7. Написать моделирующую программу для интервального алгоритма двоичной экспоненциальной отсрочки. Интервал выбора окна передачи для каждого абонента определяется по правилу, описанному в 1.3.

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

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. 5. Для вариантов 1,2 и 3 при неограниченном размере очереди необходимо вычислить значение $\lambda_{\text{крит}}$. Для других вариантов получить от преподавателя задание, результат выполнения которого представить в данном пункте отчета.

Для вариантов с неограниченным размером очереди необходимо написать расчет для метода доступа к среде, который соответствует полученному варианту.

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 $-\lambda_{\text{вых}}(\lambda)$. Интенсивность выходного потока соответствует среднему количеству абонентов, выходящих в окне.

5) Выводы по полученным зависимостям;

6) Листинг моделирующей программы, присутствует только в электронной версии отчета.

Библиографический список

1. *Марковский, С. Г.* Элементы теории помехоустойчивого кодирования: учебное 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): учебное пособие для студентов высших учебных заведений / В. Д. Колесник. М.: Высшая школа, 2009. 549 с.

3. *Кудряшов, Б. Д.* Основы теории кодирования: учебное 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.

СОДЕРЖАНИЕ

Введение 3

Лабораторная работа № 1. Использование циклических кодов для обнаружения ошибок в сетях передачи данных 4

1. Теоретический материал для выполнения лабораторной работы 4
2. Вопросы для допуска к лабораторной работе 11
3. Порядок выполнения лабораторной работы 12
4. Варианты заданий для лабораторной работы 13
5. Требования к отчету по лабораторной работе........... 16

Лабораторная работа № 2.Передача данных в системах с обратной связью 17

1. Описание модели системы и алгоритмов передачи рассматриваемых в лабораторной работе 17
2. Вопросы для допуска к лабораторной работе 25
3. Порядок выполнения лабораторной работы 28
4. Варианты заданий для лабораторной работы 29
5. Требования к отчету по лабораторной работе........... 30

Лабораторная работа № 3.Методы управления доступом к среде 32

1. Описание моделей систем изучаемых в лабораторной работе................................ 32
2. Вопросы для допуска к лабораторной работе 40
3. Порядок выполнения лабораторной работы 45
4. Варианты заданий для лабораторной работы 46
5. Требования к отчету по лабораторной работе........... 47

Библиографический список 48

Для заметок