УДК 681.325.3

Д. В. Шиман, кандидат технических наук, доцент,
декан факультета информационных технологий (БГТУ); **Ю. О. Булова**, ассистент (БГТУ)

ПРОГРАММНОЕ СРЕДСТВО ДЛЯ ИССЛЕДОВАНИЯ ХАРАКТЕРИСТИК КОДОВ, НАПРАВЛЕННЫХ НА ИСПРАВЛЕНИЕ ОШИБОК ТИПА «СТИРАНИЕ»

В статье описывается функциональная часть программного средства для исследования корректирующей способности LT- и разработанных авторами ранее LTM- и P-кодов для исправления ошибок типа «стирание». Программное средство рассчитывает избыточность выбранных кодов для испытания и отображает полученные результаты в графической форме. Экспериментально рассчитывается средняя избыточность для моделируемых кодов стираний.

This paper describes the functional part of a software tool for the error-correcting capability investigation of LT- and LTM- and P-codes developed by the authors previously. This codes are applied for error correction of the "erasure". Software tool calculates the redundancy for the selected code and displays the results in graphical form. Experimentally calculated average redundancy for simulated erasure codes.

Введение. Современный человек не представляет своей жизни без мобильных систем и компьютерных сетей. С каждым годом возрастают требования к скорости и надежности передаваемой информации. Такие системы передачи данных подвержены воздействию различных помех и шумов, что приводит к искажению или исчезновению части информации. Для устранения этой проблемы используются различные методы, основанные на помехоустойчивом кодировании информации. В данной статье рассматриваются кодовые методы, используемые в каналах с ошибками типа «стирание», получившими название «фонтанные коды» (Digital Fountain Codes) [1]. Главной особенностью является то, что кодами из этого класса можно закодировать сообщение конечного размера потенциально-неограниченным потоком (фонтаном) независимых пакетов [2].

Эффективность преобразования информации на основе таких кодов не может быть оценена с помощью известных специализированных программных продуктов, таких как MathCAD и MatLAB, которые позволяют реализовывать компьютерные модели только классической теории помехоустойчивого кодирования. Поэтому разработка компьютерного средства для исследования характеристик известных и предлагаемых фонтанных кодов является актуальной задачей.

Основная часть. Основными функциями программного средства, реализованного на языке C#, являются:

1) генерация исходного двоичного сообщения;

2) реализация кодирования согласно известному LT- [3] и предлагаемым LTM- и P-кодам [4];

3) имитация пакетной передачи сообщения с появлением заданного количества стираний;

4) расшифровка сообщения с помощью соответствующих декодеров;

5) сбор статистики и графическое отображение полученных результатов.

Ниже будут рассмотрены способы реализации данных функций.

На рис. 1 представлен внешний вид приложения.

Для задания размера исходного сообщения K используется поле «Количество исходных бит». В соответствии с введенным значением происходит генерация случайных бит с помощью метода Random. Так как предусмотрена пакетная передача данных, на форме указывается количество символов в пакете L.

Процесс кодирования у LT- и LTM-кодов не различается.

1. С использованием робастного распределения вероятностей степеней $p(d)$ (1) выбирается количество исходных символов d_i (степень кодового символа), вовлеченных в операцию кодирования для генерации кодового символа x_i.

$$p(d) = \frac{\mu(d) + \tau(d)}{Z}, \qquad (1)$$

где Z – нормирующий множитель вероятностного распределения; $\mu(d)$ – вероятностное распределение, при котором после каждой итерации декодирования в пакете оставался бы один кодовый символ со степенью 1 (2); $\tau(d)$ – вероятностное распределение, при котором исходный символ входит в большинство кодированных символов (3).

$$\mu(d) = \begin{cases} \frac{1}{K} \text{ при } d = 1, \\ \frac{1}{d(d-1)} \text{ при } d = 2,3,...,K. \end{cases} \qquad (2)$$

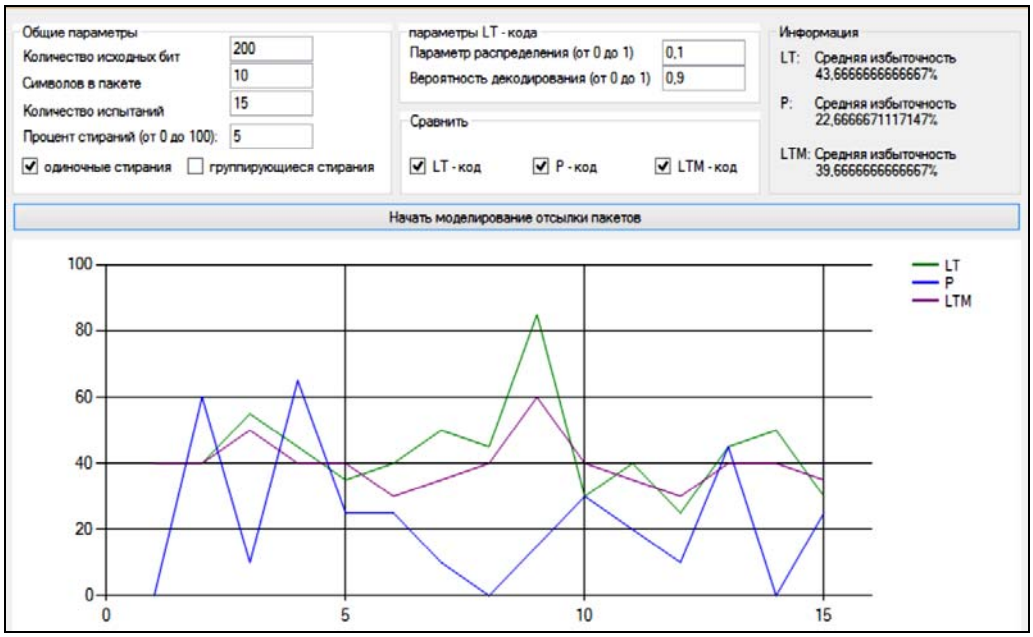

Рис. 1. Внешний вид программного средства

$$\tau(d)=\begin{cases}\dfrac{S}{Kd} \text{ при } d=1,\ 2,\ ...,\ \dfrac{K}{S},\\ \dfrac{S}{K}\cdot\ln\left(\dfrac{S}{\delta}\right) \text{ при } d=\dfrac{K}{S},\\ 0 \text{ при } d>\dfrac{K}{S},\end{cases} \qquad (3)$$

где $S=c\cdot\ln(K/\delta)\cdot\sqrt{K}$ – число исходных символов, хоть один раз входящих в кодовые символы степени $d=K/S$; c – параметр, величина которого $0\le c\le 1$ (задается в поле «Параметр распределения»); δ – вероятность неудачного декодирования (задается в поле «Вероятность декодирования» как $1-\delta$).

2. Из исходного сообщения случайным образом выбираются d_i исходных символов s_j.

3. Сложением бит этих исходных символов по модулю 2 (операция XOR) формируется кодовый символ x_i.

Таким образом формируется необходимое количество (соответствующее размеру пакета) кодовых символов x_i (рис. 2).

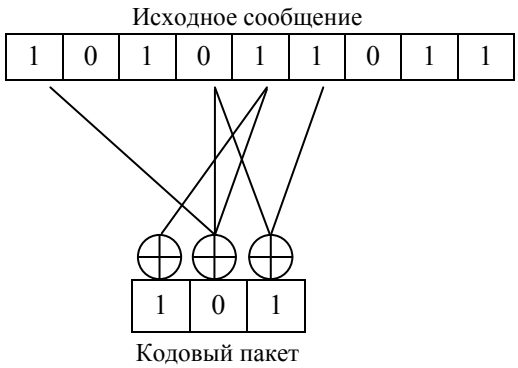

Рис. 2. Пример кодирования LT- и LTM-кодов

Процесс формирования кодовых пакетов продолжается до тех пор, пока не будет получено сообщение от приемника о достоверном приеме сообщения. Поскольку декодеру не важен порядок передачи кодовых символов и время формирования каждого пакета постоянно, кодер такого класса по запросу всегда может добавить «на лету» небольшое число кодовых пакетов [1].

Процесс кодирования P-кода состоит из нескольких этапов.

1. Пока выполняется условие $0\le j\le K-1$, из сообщения последовательно выбираются по 2 исходных символа s_j с шагом $h=2$ и суммируются по модулю 2, образуя новый кодовый символ x_j (4). Причем каждый следующий кодовый символ включает в себя исходные символы, следующие за предыдущими, вовлеченными в операцию кодирования, через 1 бит (рис. 3).

$$x_i=s_j\oplus s_{j+h}. \qquad (4)$$

2. Кодирование на втором этапе происходит согласно формуле (4), однако начинается со смещения на один исходный символ.

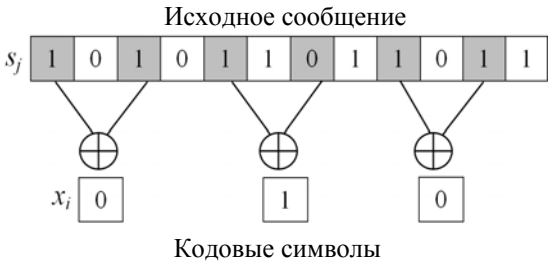

Рис. 3. Схематическое представление первого этапа кодирования P-кода

3. Третий этап кодирования (рис. 4) выполняется при $0 \le j \le K-1$ и заключается в формировании кодовых символов со степенью 3 путем последовательного суммирования трех соседних символов исходного сообщения по модулю 2 (5).

$$x_i = s_j \oplus s_{j+h/2} \oplus s_{j+h}. \quad (5)$$

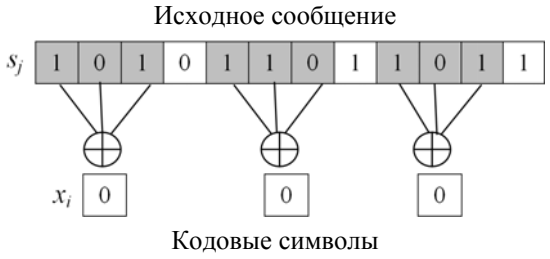

Рис. 4. Схематическое представление третьего этапа кодирования Р-кода

4. Четвертый этап кодирования аналогичен третьему, но происходит со смещением на 1 исходный символ согласно условию $1 \le j \le K-1$.

Сформированные кодовые символы образуют кодовые пакеты из L бит. Для уменьшения степени группирования ошибок типа «стирание» используется W-циклический перемежитель [5]. После перемежения сообщение передается получателю, где осуществляется деперемежение и декодирование полученного сообщения.

Таким образом, формирование кодовых пакетов продолжается до тех пор, пока на принимающей стороне не будет полностью декодировано сообщение. Если сообщение не было расшифровано в течение четырех этапов – процесс кодирования возвращается к первому этапу, при этом шаг h увеличивается в 2 раза.

Результат кодирования каждого из кодов записывается в соответствующую бинарную матрицу, номер строки которой соответствует индексу кодового символа, а позиции единиц в строке – номерам исходных бит, входящих в данный кодовый символ. Пример бинарной матрицы представлен на рис. 5.

После операции кодирования разработанное программное средство имитирует передачу кодовых пакетов по каналу. При этом используются параметры, которые указаны на форме: количество стираний в процентах, а также характер распределения ошибок.

Далее происходит операция декодирования. Для Р-кода сначала необходимо произвести операцию деперемежения полученного кодового пакета.

Декодеры LT-, LTM- и Р-кодов используют такую же бинарную матрицу, как и кодер на стороне отправителя. Таким образом, декодеру известна информация о степени d_i каждого кодового символа x_i и номерах исходных бит, входящих в этот символ, за исключением символов, которые были стерты при передаче сообщения.

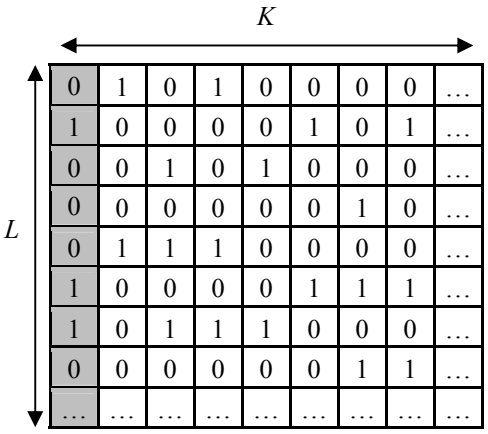

Рис. 5. Бинарная матрица кодирования

Процесс декодирования происходит в несколько этапов.

1. Определяется степень кодового символа d_i.

2. Если степень кодового символа d_i = 1 – устанавливается $s_j = x_i$, тем самым находится j-й символ (рис. 6).

3. Удаляются копии j-го символа (6):

$$x_i = x_i \oplus s_j. \quad (6)$$

Из всех списков номеров кодовых символов удаляется номер i: строка бинарной матрицы суммируется по модулю 2 с бинарной строкой, содержащей единицу в позиции i, что приводит к понижению степеней соответствующих кодовых символов (рис. 7). Если в матрице не осталось кодовых символов со степенью 1 ($d_i \ne 1$) LT-декодер отказывается от декодирования и ожидает нового пакета кодовых символов от LT-кодера. Однако LTM- и Р-декодеры способны сформировать новый кодовый символ из имеющихся, для этого используется четвертый этап декодирования.

Рис. 6. Второй этап декодирования LT-, LTM- и Р-кодов

Бинарная матрица

0	1	0	1	0	0	0	0	…
1	0	0	0	0	1	0	1	…
0	0	1	0	1	0	0	0	…
0	0	0	0	0	0	**0**	0	…
0	1	1	1	0	0	0	0	…
1	0	0	0	0	1	**0**	1	…
1	0	1	1	1	0	0	0	…
0	0	0	0	0	0	**0**	1	…
…	…	…	…	…	…	…	…	…

Рис. 7. Третий этап декодирования LT-, LTM- и P-кодов

Строка бинарной матрицы, соответствующая анализируемому кодовому символу x_i, поочередно суммируется по модулю 2 с бинарными строками всех кодовых символов, в формировании которых участвовал первый исходный символ из x_i. Таким образом получаются новые кодовые символы, и процесс декодирования переходит к этапу 1 (рис. 8).

Бинарная матрица

0	**1**	**0**	**1**	**0**	**0**	**0**	**0**	**…**
0	0	0	0	0	0	0	0	…
0	0	1	0	1	0	0	0	…
0	0	0	0	0	0	0	0	…
0	**1**	**1**	**1**	**0**	**0**	**0**	**0**	**…**
0	0	0	0	0	0	0	0	…
1	0	1	1	1	0	0	0	…
0	0	0	0	0	0	0	0	…
…	…	…	…	…	…	…	…	…

Новое кодовое слово

0	**0**	**1**	**0**	**0**	**0**	**0**	**0**	**…**

Рис. 8. Процесс декодирования

Так как в описанных выше кодерах и декодерах используются случайные вероятностные распределения, сбор статистики необходимо совершать на основе нескольких опытов. Число испытаний можно указать на форме в соответствующем поле. Также реализована возможность выбора кодов, по которым нужно провести сравнительный анализ.

Программное средство рассчитывает избыточность для выбранных кодов на каждом испытании и отображает полученные результаты в виде графика (рис. 1). После всех опытов подсчитывается средняя избыточность, что позволяет оценить эффективность описанных выше кодов.

Заключение. Разработанное программное средство позволяет оценить эффективность преобразования информации на основе фонтанных кодов. В нем реализован алгоритм генерации бинарной последовательности, кодирование (декодирование) согласно известному LT- и предлагаемым LTM- и P-кодам, имитация пакетной передачи сообщения с появлением заданного количества стираний, сбор статистики и графическое отображение полученных результатов.

Литература

1. Варгаузин В. Помехоустойчивое кодирование в пакетных сетях // ТелеМультиМедиа. 2005. № 3. С. 10–16.

2. Булова Ю. О. Кодовые методы нейтрализации ошибок типа «стирание» в каналах передачи двоичной информации // Труды БГТУ. 2012. № 6: Физ.-мат. науки и информатика. С. 142–145.

3. Luby M. LT codes // Proc. 43rd Ann. IEEE Symp. on Foundations of Computer Science, Vancouver, November 16–19, 2002. С. 271–282.

4. Булова Ю. О. Анализ исправления ошибок типа «стирание» П-кодом в каналах передачи данных // Труды БГТУ. 2013. № 6: Физ.-мат. науки и информатика. С. 123–126.

5. Gorbunova Yu., Urbanovich P. W-ciclic method interleaving of data for communication system // NEET'2011: proc. VII International Conference, Zakopane, Poland, June 28 – July 1, 2011. P. 149.

Поступила 19.03.2014