

Заключение

На основе исследований характера построений очередей, когда на сервер подается долговременно зависимый трафик (ДВЗ), в [5] делается вывод о том, что ДВЗ имеет незначительное влияние на емкость буфера и $Q_0 S$ в реальных условиях. Это объясняется тем, что уменьшение емкости буфера снижает память системы, и, следовательно, устраняет последствия воздействия на систему ДВЗ. Эти результаты являются дискуссионными, однако исследования, проведенные авторами, подтверждают сделанные в [5] предположения. Результаты этих исследований опубликованы в [4] для системы массового обслуживания $M/M/m/n$, осуществленные классическим методом, и для трафика-ДВЗ – методом статистических испытаний (Монте-Карло). Проведенные испытания показали, что при моделировании систем связи для достижения положительного эффекта необходи-

мо варьировать не объемом буферов m , а числом каналов n в каждом направлении передачи информации.

Литература

1. Митилино С. А. Фрактальная катастрофа TCP/IP. <http://www.itc.ua>.
2. Назаров Н.А., Симонов М.В. АТМ: технология высокоскоростных сетей. М.: Эко – Трендз, 1999. – 252 с.
3. Тихонов В.И. Выбросы случайных процессов. М.: Наука, 1970. – 359 с.
4. Фомин Л.А., Линец Г.И., Будко П.А., Ватага А.И. Учет влияния спектральных свойств трафика на параметры сети с технологией АТМ // Электросвязь. №11, 2001. – С. 24-26.
5. Grossglauser M., Bolot J-C. On the relevance of long-range dependence in network traffic // SIGCOM-96, August, 1996. – P. 4-7.

УДК 621.391

АЛГОРИТМ ОПРЕДЕЛЕНИЯ КОРРЕКТИРУЮЩЕЙ СПОСОБНОСТИ БЛОЧНЫХ КОДОВ, ПОСТРОЕННЫХ ИЗ СВЕРТОЧНОГО

Малофеев О.П., Манаенко С.С., Мишин Д.В., Потягов Д.А.

Разработан алгоритм определения корректирующей способности блочных кодов, построенных на основе сверточного кодера, при использовании способа синдромного декодирования, учитывающий возможность исправления ошибок большей кратности за счет определения синдромов исправляемых и обнаруживаемых ошибок и представления корректирующей способности в виде нецелого числа.

Известен [1] способ синдромного декодирования сверточных кодов, позволяющий на основе приведения сверточного кода к блочной структуре и применения синдромного декодирования блочных кодов исправлять многократные ошибки. Как правило, для определения корректирующей способности (γ) блочных кодов, построенных из сверточного, используется традиционный способ [2] определения спектра весов и минимального кодового расстояния $d_{\min}$, в соответствии с которым

$$\gamma = [(d_{\min} - 1) / 2]. \quad (1)$$

Однако в [3] рассмотрен пример построения блочного кода, полученного на основе сверточного кодера, при использовании синдромного декодирования, где выявлено, что такой код исправляет ошибки кратности большей, чем определенной согласно выражению (1).

Целью статьи является разработка алгоритма определения корректирующей способности блочных кодов, построенных из сверточного, при использовании способа синдромного декодирования.

Достижение поставленной цели возможно на основе обобщения результатов, полученных в [3], для различных конструкций блочных кодов, построенных на основе сверточного кодера, с использованием предложенного в [4] способа построения квазисовершенного помехоустойчивого кода и представления корректирующей способности (γ) в виде нецелого числа.

Для этого рассмотрим несистематический сверточный код со скоростью $R = 1/2$, тремя ячейками памяти и видами связи между ячейками, описываемых порождающими полиномами $g_1(x) = 1 + x^2$ и $g_2(x) = 1 + x + x^2$. Как показано в [3], один из вариантов проверочной матрицы блочного кода, построенного из сверточного, путем добавления «нулевого хвоста» может иметь вид:

$$H = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}. \quad (2)$$

Столбцы проверочной матрицы представляют собой синдромы, указывающие место пораженного символа в кодовой комбинации при возникновении однократной ошибки ($e_j^{(l)}$, j – значение вектора ошибки в десятичной системе счисления). Для обнаружения и исправления l ошибок в принятой кодовой комбинации $X_i^{(l)}$ (i – значение принятой двоичной кодограммы в десятичной системе счисления) необходимо определить синдром путем умножения ее на транспонированную проверочную матрицу H^T . Максимальное число ненулевых синдромов для кода (10,3) составит $2^7 - 1 = 127$. Если синдром кода перевести в десятичное число (r), то для соответствующего вектора ошибки $e_j^{(l)}$ можно записать синдром в виде $-S_r^{(l)}$, где l – кратность ошибки. Для кода (10,3) и проверочной матрицы (2) стандартная таблица декодирования, составленная по принципу, изложенному в [4], представлена в таблице 1.

Таблица 1. Стандартная таблица декодирования кода (10,3)

$X_i^{(l)}$				$e_j^{(l)}$	$S_r^{(l)}$
$X_0^{(0)}$	$X_{55}^{(0)}$	...	$X_{923}^{(0)}$	$e_0^{(0)}$	$S_0^{(0)}$
$X_1^{(1)}$	$X_{54}^{(1)}$	...	$X_{922}^{(1)}$	$e_1^{(1)}$	$S_{64}^{(1)}$
...	...	...	...	...	...
$X_{512}^{(1)}$	$X_{567}^{(1)}$	...	$X_{411}^{(1)}$	$e_{512}^{(1)}$	$S_{15}^{(1)}$

Представленная часть таблицы декодирования предназначена для исправления однократных ошибок. Следует заметить, что все элементы одной и той же строки имеют один и тот же синдром. Запрещенные кодограммы $X_i^{(l)}$ можно получить путем суммирования по модулю два разрешенной кодограммы $X_i^{(0)}$ и соответствующего данной строке вектора ошибки $e_j^{(l)}$.

Следовательно, возможно некоторое упрощение процедуры декодирования, заключающееся в следующем:

- необходимо вычислить синдром согласно выражению

$$S_r^{(l)} = X_i^{(l)} \times H^T = (X_i^{(0)} \oplus e_j^{(l)}) \times H^T = e_j^{(l)} \times H^T \quad (3)$$

и найти его в правом столбце таблицы декодирования;

- взять из соседнего столбца той же строки вектор ошибки $e_j^{(l)}$ и прибавить его к принятой кодограмме $X_i^{(l)}$.

Таким образом, вместо стандартной таблицы для декодирования достаточно использовать таблицу синдромов и соответствующих им векторов ошибок. Для кода (10, 3) синдромы и соответствующие им векторы однократных ошибок представлены в таблице 2.

Таблица 2. Синдромы и векторы однократных ошибок кода (10,3)

$S_r^{(l)}$	$e_j^{(l)}$	$S_r^{(l)}$	$e_j^{(l)}$
$S_{64}^{(1)}$	$e_1^{(1)}$	$S_8^{(1)}$	$e_{32}^{(1)}$
$S_{108}^{(1)}$	$e_2^{(1)}$	$S_2^{(1)}$	$e_{64}^{(1)}$
$S_{32}^{(1)}$	$e_4^{(1)}$	$S_{54}^{(1)}$	$e_{128}^{(1)}$
$S_{16}^{(1)}$	$e_8^{(1)}$	$S_1^{(1)}$	$e_{256}^{(1)}$
$S_4^{(1)}$	$e_{16}^{(1)}$	$S_{15}^{(1)}$	$e_{512}^{(1)}$

Проведенный анализ синдромов и соответствующих им векторов ошибок показал, что всем однократным (10 векторов ошибок) и всем двукратным (45 векторов ошибок) ошибкам соответствуют свои неповторяющиеся синдромы. Трехкратные ошибки, которым соответствуют не встречавшиеся ранее синдромы, составляют треть часть (39 векторов ошибок) из числа возможных ошибок ($C_{10}^3 = 120$). Данные синдромы позволяют гарантированно исправлять ошибки кратности $l = 1; 2; 3$ и их можно отнести к синдромам исправляемых ошибок (δ_l).

Кроме того, имеются повторяющиеся синдромы трехкратных ошибок, которые не встречаются в ошибках меньшей кратности, причем из них 24 синдрома повторяются по два раза, а синдром $S_{46}^{(3)}$ – три раза. Данные синдромы позволяют однозначно обнаружить трехкратные ошибки и правильно их исправить с вероятностью, зависящей от числа повторений синдромов. В связи с этим их можно отнести к синдромам обнаруживаемых ошибок (t_l). При анализе также получено 28 синдромов трехкратных ошибок, встречающихся в ошибках меньшей степени кратности, а оставшиеся незадействованными 8 синдромов из 127 относятся к четырехкратным ошибкам, которые однозначно не определяются.

Рассматриваемый код (10,3) обладает большой избыточностью, что значительно снижает скорость передачи информации. Для повышения скорости передачи информации необходимо увеличивать скорость кодирования, а это возможно за счет укорочения кода. Укоротить код можно путем удаления соответствующих столбцов по-

рождающей матрицы, что приведет к уменьшению длины кодовой комбинации. Удалять необходимо те столбцы матрицы, которые соответствуют уравнениям кодирования, имеющим наименьшее количество связей с информационными символами. В этом случае можно получить коды с различной длиной кодовой комбинации при неизменном количестве информационных символов. Тогда на основе кода (10,3) можно построить коды с различной скоростью кодирования, а, следовательно, и с различной корректирующей способностью: (9,3), (8,3), (7,3), (6,3).

Таким образом, при определении корректирующей способности блочных кодов, построенных из сверточного, необходимо сначала вычислить максимальную кратность гарантированно исправляемых ошибок ($\gamma_0^{\max}$), рассчитать синдромы всевозможных векторов ошибок кратности $l = 1, 2, \dots, (\gamma_0^{\max} + 1)$ и распределить полученные синдромы на синдромы исправляемых (δ_{γ_0+1}) и обнаруживаемых (t_{γ_0+1}) ошибок. Затем на основе предложенного в [4] уравнения построения квазивершенного кода рассчитать γ . Тогда алгоритм определения корректирующей способности блочных кодов, построенных из сверточного, можно представить в следующем виде.

1. Определить для блочного (n, k) кода максимальную кратность гарантированно исправляемых ошибок ($\gamma_0^{\max}$) согласно выражению

$$\gamma_0^{\max} = \max_{\gamma} \left\{ \sum_{i=0}^{\gamma} C_n^i \leq 2^{n-k} \right\}, \quad (4)$$

где C_n^i – количество сочетаний из n по i .

2. Получить множество десятичных значений векторов ошибок $e_f^{(l)}$ кратности включительно до $l = \gamma_0^{\max} + 1$:

$$J = \bigcup_{l=1}^{\gamma_0^{\max}+1} J^{(l)} = \bigcup_{l=1}^{\gamma_0^{\max}+1} \{j_1^{(l)}, j_2^{(l)}, \dots, j_m^{(l)}\}, \quad (5)$$

где $m = |J^{(l)}| = C_n^l$ – количество векторов ошибок кратности l .

3. Рассчитать синдромы $S_r^{(l)}$ для векторов ошибок, полученных в п.2, согласно выражению (3) и получить множество их значений в десятичной системе счисления

$$R = \bigcup_{l=1}^{\gamma_0^{\max}+1} R^{(l)} = \bigcup_{l=1}^{\gamma_0^{\max}+1} \{r_1^{(l)}, r_2^{(l)}, \dots, r_m^{(l)}\}. \quad (6)$$

4. Определить кратность гарантированно исправляемых ошибок для заданной конструкции кода (γ_0), равной максимальной кратности ошибок, не имеющих ни одного повторяющегося

синдрома, а также встречающегося в ошибках меньшей степени кратности:

$$\gamma_0 = \max_l \left\{ \begin{array}{l} \left(\sum_{b=1}^{C_n^l-1} \sum_{d=2}^{C_n^l} \left\{ \begin{array}{l} 1, r_b^{(l)} = r_d^{(l)} \\ 0, r_b^{(l)} \neq r_d^{(l)} \end{array} \right\} = 0 \vee \right. \\ \left. \vee \sum_{b=1}^{l-1} |R^{(l)} \cap R^{(b)}| = 0 \right\}. \quad (7)$$

5. Получить множество свободных синдромов ошибок кратности $(\gamma_0 + 1)$, не встречающихся в ошибках меньшей степени кратности, и определить их количество:

$$R_{\text{св}}^{(\gamma_0+1)} = R^{(\gamma_0+1)} - \bigcup_{i=1}^{\gamma_0} (R^{(\gamma_0+1)} \cap R^{(i)}), \quad (8)$$

$$m_{\gamma_0+1} = |R_{\text{св}}^{(\gamma_0+1)}|.$$

6. Определить количество синдромов обнаруживаемых ошибок (t_{γ_0+1}) кратности $(\gamma_0 + 1)$, равное числу повторяющихся синдромов из множества свободных данной степени кратности, и число их повторений (η_{γ_0+1}).

7. Определить количество исправляемых ошибок кратности $(\gamma_0 + 1)$ согласно выражению

$$\delta_{\gamma_0+1} = m_{\gamma_0+1} - t_{\gamma_0+1} \eta_{\gamma_0+1}. \quad (9)$$

8. Рассчитать корректирующую способность кода, используя выражения:

$$\gamma = \gamma_0 + P(\bar{E} / e^{(\gamma_0+1)}), \quad (10)$$

$$P(\bar{E} / e^{(\gamma_0+1)}) = \frac{1}{C_n^{\gamma_0+1}} \left(\delta_{\gamma_0+1} + \frac{t_{\gamma_0+1}}{\eta_{\gamma_0+1}} \right). \quad (11)$$

На основе представленного алгоритма в программной среде MatLab разработано программное обеспечение, позволяющее определять корректирующую способность блочного (n, k) кода по заданной проверочной матрице (H). С использованием разработанного программного обеспечения произведен расчет корректирующей способности блочных кодов, построенных из сверточного, результаты которого представлены в таблице 3.

Таблица 3. Результаты расчета блочных кодов, построенных из сверточного

	(6, 3)	(7, 3)	(8, 3)	(9, 3)	(10, 3)
γ_0	1	1	1	1	2
δ_{γ_0+1}	0	0	13	30	39
t_{γ_0+1}	1	7	6	3	24
η_{γ_0+1}	3	3	2	2	3
γ	1,022	1,111	1,583	1,875	2,427
$\gamma^{\text{пот}}$	1,067	1,381	1,821	2,214	2,6
$\Delta\gamma$	0,045	0,27	0,238	0,339	0,173

В предпоследней и последней строках данной таблицы приведены расчеты потенциальной границы качества кодирования и степень достижения корректирующей способности кода данной границы ($\Delta\gamma = \gamma^{\text{пот}} - \gamma$).

Из таблицы видно, что наилучшей корректирующей способностью обладает код (10,3), но по степени достижения потенциальной границы кодирования ($\gamma^{\text{пот}}$) он находится на втором месте после кода (6,3). Причем, следует отметить, что код (6,3) максимально приближается к своей потенциальной границе кодирования, но при этом он обладает наименьшей корректирующей способностью.

Таким образом, разработанные алгоритм (4)-(11) и на его основе программное обеспечение позволяют определять корректирующую способность (γ) блочного кода, построенного из сверточного, и оценивать конструкцию кода по степени достижения им своей потенциальной границы кодирования ($\Delta\gamma$).

Литература

1. Патент на изобретение № 2282307 Способ синдромного декодирования для сверточных кодов / Малофеев О.П., Куликов В.В., Солчатов М.Э., Манаенко С.С., Карпов Д.К., Киселев Н.В. Приор. 30.08.2004, опублик. 20.08.2006.
2. Р. Морелос-Сарагоса. Искусство помехоустойчивого кодирования. Методы, алгоритмы, применение. М.: Техносфера, 2005. – 320 с.
3. Куликов В.В., Баркетов С.В., Солчатов М.Э., Карпов Д.К. Использование синдромного декодирования для сверточных кодов. // Физика волновых процессов и радиотехнические системы. Т.6, №3, 2003. – с. 43-46.
4. Карпов Д.К., Куликов В.В., Манаенко С.С. Определение потенциальных возможностей помехоустойчивых кодов // ИКТ, Т.5, №1, 2007. – С.17-20.

УДК 621.396

РЕКОНСТРУКЦИЯ ВОЛП С ВКЛЮЧЕНИЕМ КОМПЕНСИРУЮЩИХ ВОЛОКОН В МУФТАХ ОПТИЧЕСКОГО КАБЕЛЯ

Волков К.А.

В данной работе рассмотрен один из вариантов реализации плотного режима управления дисперсией, основанного на использовании компенсирующих дисперсию волокон в муфтах оптического кабеля. Методами компьютерного моделирования показана возможность реализации такого подхода.

Введение

Современные сети связи развиваются в условиях непрерывного роста потребностей в увеличении их пропускной способности, что делает крайне актуальной задачу реконструкции волоконно-оптической линии передачи (ВОЛП) для работы на более высоких скоростях. В настоящее время для региональных сетей связи актуален переход к скоростям передачи 10 Гбит/с и выше. Для этой цели перспективным может быть применение солитонов, управляемых за счет изменения дисперсии вдоль линии передачи [1-2]. В частности, так называемого режима управления дисперсией на плотных схемах компенсации – DDMS (Dense Dispersion Management Soliton) [4-6].

Для реализации такого режима требуется компенсация хроматической дисперсии, которая осуществляется включением оптических волокон (ОВ) с дисперсией противоположного знака. При реконструкции ВОЛП это может быть заме-

на строительных длин оптического кабеля (ОК). Однако при разработке проектов реконструкции ВОЛП, необходимо учитывать стоимость отвода земли под строительство и, соответственно, стоимость инфраструктуры в целом. Уже сегодня затраты на землеотводы и согласования составляют большую часть затрат на строительство ВОЛП. По прогнозам в перспективе эта тенденция будет только усугубляться [3]. Как следствие, одним из требований к проектам реконструкции ВОЛП является минимизация объемов работ на линейно-кабельных сооружениях в целом и, по возможности, исключение работ по прокладке и замене ОК. Этим требованиям отвечает вариант реализации DDMS с включением компенсирующих дисперсию ОВ (DCF – dispersion compensation fiber) в муфты ОК.

Исследованию возможности работы ВОЛП в режиме DDMS при включении компенсирующих ОВ в муфты ОК и посвящена данная работа.

Распространение управляемого дисперсией солитона в оптической линии

Реализация DDMS предусматривает чередование с заданным периодом ОВ строительных длин кабеля с положительной и отрицательной хрома-