

УДК 519.8

ПРЕОДОЛЕНИЕ ПАТТЕРН-ЭФФЕКТА
В ОПТОВОЛОКОННЫХ ЛИНИЯХ СВЯЗИ С ПОМОЩЬЮ
АДАПТИВНОГО БЛОЧНОГО КОДИРОВАНИЯ *)

А. С. Скидин, М. П. Федорук, А. В. Шафаренко, С. К. Турицын

Аннотация. Рассмотрены теоретические и практические методы построения блочных кодов с ограничениями, позволяющих снизить число ошибок при передаче сообщения по каналу связи в присутствии паттерн-эффекта — эффекта зависимости частоты ошибок в канале от вида передаваемых данных. Приведена схема построения кода, позволяющего адаптироваться к любой линии связи с наличием паттерн-эффекта. Проанализированы свойства кода и результаты его применения к конкретным линиям.

Ключевые слова: код с ограничениями, теория кодирования, формальный язык.

Введение

Как известно, при передаче данных по оптоволоконной линии связи на больших скоростях порядка 40 Гбит/с и выше на процесс передачи начинают оказывать существенное влияние нелинейные эффекты. Обладая различной природой, они вносят нелинейные помехи в канал связи. Это приводит к тому, что различные виды данных передаются по каналу с различным качеством — одна последовательность символов передаётся практически без ошибок, в то время как на долю другой может приходиться половина всех ошибок в канале. Это явление называется *паттерн-эффектом*, а оценка его влияния на пропускную способность канала может быть дана путём нахождения частот появления ошибок по сочетаниям передаваемых по каналу символов. В оптических линиях паттерн-эффект может быть вызван различными причинами. Например, к нему может привести наличие эффекта четырёхволнового смешения (FWM — four-wave mixing) [1]. Четырёхволновое смешение может

*) Исследование выполнено при финансовой поддержке проекта ОНИТ РАН № 1.1., а также в рамках научно-исследовательской разработки по Гос. контракту № 02.740.11.0396.

привести, как показано в [7], к появлению так называемых импульсов-«призраков». Рассмотрим данное явление более подробно. Пусть в передаваемом сообщении на позициях k , l и m находятся логические «единицы» ($b_k = b_l = b_m = 1$). Импульсом-«призраком» будет появление единичного символа на позиции b_{k+l-m} в случае, когда в исходном сообщении на этой позиции находится «нуль». Несложно показать, например, что битовые сочетания 1101, 1011 и 11011 могут провоцировать появление импульса-«призрака», что может привести к неверной передаче нулевого бита.

Паттерн-эффект также может возникнуть из-за перекрытия соседних импульсов, что создаёт опасность неверного детектирования символа в канале. С учётом постоянного увеличения скоростей передачи данных, сопровождающимся уменьшением длительности импульсов, можно предположить, что паттерн-эффект будет иметь место и в будущих оптоволоконных системах.

1. Методы подавления паттерн-эффекта

Прежде всего, хотя паттерн-эффект имеет место во многих линиях связи, далеко не везде он играет заметную роль. Это связано во многом как с наличием аппаратных средств, позволяющих на физическом уровне предотвратить чрезмерное перекрытие импульсов и уменьшить уровень шума, так и с характером паттерн-эффекта. Но в ряде случаев, как показывают эксперименты, паттерн-эффект очень ярко выражен. Например, в [5] описан канал, в котором вероятность появления ошибки центрального бита при передаче триплета 101 в 10 раз выше, чем при передаче остальных триплетов. Это даёт основание полагать, что в ряде случаев имеет смысл ограничить количество «плохих» триплетов в сообщении, снизив таким образом частоту ошибок при передаче этого сообщения. С этой целью разумно использовать специальные методы кодирования.

Коды, решающие подобные задачи, называются *кодами с ограничениями* (constrained codes). По сути каждый из этих кодов задаёт формальный язык A над алфавитом Ω исходного сообщения. В случае блочного кода с длиной блока m символов задание языка происходит путём перечисления всех его слов ($A \subseteq \Omega^m \subseteq \Omega^*$). Если код потоковый или количество слов блочного кода велико, то задание языка возможно осуществить с помощью некоторой формальной грамматики. Хотя алфавит Ω может быть любым множеством, в практических приложениях, как правило, $\Omega = \{0, 1\}$. Свойства слов, входящих в язык A , определяются задачами, стоящими перед кодом.

Наиболее известным видом ограниченных кодов являются RLL-коды — коды с ограничением длин серий (run-length limited codes). Они применялись (а порой и сейчас применяются) в магнитной записи с целью ограничить количество изменений направления магнитного поля в единицу времени. Грамматику языка, соответствующего данному коду, порождает правило наличия в слове между двумя «единицами» не менее d , но не более k «нулей». Так как обычно в магнитной записи «единица» означает смену направления магнитного поля, а «нуль» направления поля не меняет, то задача ограничения частоты смены поля данным кодом успешно решается. Для построения кодов RLL используется как перечисление всех слов языка, если слова небольшой длины, так и использование комбинаторных методов (например, в [3] код RLL построен на базе рекуррентных соотношений, которые позволяют определить число слов заданной длины, удовлетворяющих ограничению RLL).

В настоящее время предложено несколько ограниченных кодов и для подавления паттерн-эффекта. Как правило, они направлены на решение узкой задачи по удалению отдельной последовательности данных, а именно паттерна 101 и паттернов, содержащих 101. Столь узкая постановка задачи обусловлена прикладной стороной вопроса, поскольку вероятность ошибочной передачи 101, как показали эксперименты [5, 6], во многих случаях гораздо выше вероятности появления ошибки при передаче других триплетов. Теоретический подход на основе марковских цепей, заключающийся в контролируемом уменьшении числа паттернов 101 в передаваемом сообщении, изложен в [4]. В [2, 7] предложены методики кодирования в терминах конечных автоматов, имеющие целью ликвидацию паттернов 1101 и 1011, а также паттерна 10101. Например, в [2] предложен простой код со степенью кодирования $2/3$, работающий по принципу скользящего окна; на выходе получается поток данных, заведомо не содержащий «проблемных» битовых сочетаний.

В статье рассмотрено обобщение предложенных кодов, подавляющих паттерн-эффект. Приведена теоретическая основа построения блочного кода, позволяющего сократить число паттернов, вероятность ошибочной передачи которых максимальна. Сначала в работе рассмотрен код, преследующий цель уменьшения числа определённых последовательностей в блоке до заданного значения. После этого рассмотрен код, направленный на минимизацию ошибки в блоке данных с учётом статистики ошибок в линии связи. Целевое условие для второго кода является с технической точки зрения оптимальным для снижения частоты ошибок в канале, поскольку код де-факто является *адаптивным*, меняя множество

кодовых слов при изменении характера паттерн-эффекта.

В качестве примера применения кодов рассмотрен особый случай блочного кода для линии, в которой преобладают ошибки при передаче триплета 101, поскольку данный триплет является «опасным» как в линиях с перекрытием импульсов, так и при наличии четырёхволнового смещения (данный триплет содержит паттерны 1101, 1011 и 11011, провоцирующие появление импульса-«призрака» на позиции нулевого бита). Приведён пример построения и результат применения адаптивного кода применительно к линии связи, моделирование которой проведено в [6]. Индуктивный характер изложения материала позволяет полнее раскрыть особенности построения кодов, а также обозначить проблемы их построения в плане математического обеспечения.

2. Подавление паттерн-эффекта с помощью блочного кодирования

Выше были приведены методики устранения из потока данных «проблемных» битовых сочетаний. Данные методики отличаются простотой реализации, а комбинаторные свойства методов, на которых они базируются, описаны в литературе. Однако их жёсткость является довольно существенным недостатком, поскольку они не могут, например, удалить лишь половину триплетов 101 из сообщения. При этом гибкость таким кодам просто необходима, поскольку, как показывает теоретико-информационный анализ [4], удаление половины триплетов 101 требует кода с избыточностью 3%, в то время как удаление всех триплетов потребует по крайней мере 18,9% избыточности, что существенно больше (тот факт, что с меньшей избыточностью триплет 101 полностью не удалить, будет доказан ниже).

Рассмотрим теоретические основы блочного кодирования, которое предоставляет возможность управлять степенью удаления паттернов из сообщения, что даёт гибкий инструмент по нахождению компромисса между степенью удаления «плохих» паттернов и избыточностью кода. Как было упомянуто выше, любой из кодов с ограничениями задаёт некоторый язык над алфавитом исходного сообщения. Поэтому рассмотрим блочный код с длиной блока m как формальный язык $B \subseteq \Omega^m$, состоящий из слов длины m над двоичным алфавитом Ω . Хотя алфавит мы считаем двоичным, следующие рассуждения могут быть легко перенесены и на случай недвоичных сообщений. Под паттерном будем понимать подслово слова $a \in B$. Так, в слове длины $a = a_1 a_2 \dots a_m \in B$ ровно $m - q + 1$ паттернов длины $q > 0$. Обозначим через $N_t(a)$ количество пат-

тернов t в кодовом слове a . Например, для кодового слова $a = 0111110111$ $N_{0111}(a) = 2$, а $N_{111} = 4$.

Будем обозначать далее через $T = \{(t_1, k_1), (t_2, k_2), \dots, (t_j, k_j)\}$ множество пар (t_i, k_i) , задающих запреты, которым удовлетворяет язык $B \subseteq \Omega^m$. Если $a \in B$, то $N_{t_i}(a) \leq k_i$ для всякого $1 \leq i \leq j$. Такой язык (и соответствующий ему код) далее будем обозначать через $B(m, T)$. Построенный язык естественно назвать *языком со слабыми запретами*; код, обладающий такими свойствами, в теории кодирования принято называть *слабо ограниченным*.

Назовём *степенью слова* a число $p(a) = \sum_{i=1}^j N_{t_i}(a)$. Очевидно, что если $a \in B$, то $0 \leq p(a) \leq \sum_{i=1}^j k_i = p_{\max}$. Пусть

$$B_r(m, T) = \{a \mid a \in B(m, T), p(a) = r\}.$$

Упорядочим слова языка в последовательность W следующим образом: $W(i) \in B_r(m, T)$, если $\sum_{k=0}^{r-1} |B_k(m, T)| \leq i < \sum_{k=0}^{r-1} |B_k(m, T)| + |B_r(m, T)|$, $r = 0, 1, \dots, p_{\max}$. Следовательно, все кодовые слова построены в порядке возрастания их степеней. Слова же одной степени могут быть упорядочены, например, лексикографически.

Представление информации на языке B ведётся следующим образом: если $s = \sum_{i=0}^{n-1} x_i |\Omega|^i$ — слово длины n над q -ичным алфавитом Ω , то на языке B оно имеет вид $y_0 y_1 \dots y_{i_{\max}}$, где

$$y_i = W\left(\left\lfloor \frac{s}{|B(m, T)|^i} \right\rfloor \bmod |B(m, T)|\right), \quad (1)$$

а i_{\max} такое, что $|B(m, T)|^i \leq s < |B(m, T)|^{i+1}$. Процесс перевода числа из одного языка в другой можно представить как перевод числа из q -ичной записи в запись по основанию $|B(m, T)|$, в которой роль «цифр» играют номера слов из B , задействованные в кодировании фрагмента s . Поскольку список $W(j)$ не содержит двух одинаковых слов, данный код может быть легко декодирован с использованием «обратного словаря» W^{-1} :

$$W^{-1}(y_i) = W\left(\left\lfloor \frac{s}{|B(m, T)|^i} \right\rfloor \bmod |B(m, T)|\right) = p_i.$$

Тогда $s = \sum_{i=0}^{i_{\max}} p_i |B(m, T)|^i$.

В случае длинной входной последовательности относительная избыточность языка (или кода, который ему соответствует) определяется выражением

$$R = 1 - \log_2 |B(m, T)|/m. \quad (2)$$

Плотность паттерна t_i в расчёте на один символ выражения на языке $B(m, k)$ примерно равна

$$\frac{1}{m|B(m, T)|} \sum_{r=0}^{k_i} r \cdot |C(t_i, r)|,$$

где $C(t, k) = \{a \mid a \in B(m, T), N_t(a) = k\}$ — множество таких слов языка $B(m, T)$, что паттерн t входит в слово ровно k раз. Неточный характер данной формулы объясняется тем, что она не учитывает паттерны, которые появляются в последовательности слов языка на границе этих слов; появление таких паттернов зависит от того, какими символами заканчиваются и начинаются слова языка. В кодировании влияние перекрытия блоков называется *краевыми эффектами кода*. Получим формулу частоты появления паттерна $t = b_1 b_2 \dots b_n$. Он появится на границе слов cd , если слово c заканчивается на b_1 , а слово d начинается с $b_2 b_3 \dots b_n$. Продолжая рассуждения, можно прийти к выводу, что всего имеется $n - 1$ различных случаев появления паттерна на границах двух слов. Частота появления каждого случая в расчёте на одно слово языка B задаётся формулой $\varphi_e(b_1 b_2 \dots b_i) \cdot \varphi_b(b_{i+1} b_{i+2} \dots b_n)$, где $\varphi_b(b)$ — частота появления паттерна b в самом начале слова из $B(m, T)$, а $\varphi_e(b)$ — частота появления паттерна в самом конце слова.

Пусть $Q(t, n)$ — частота появления паттерна t длины n в длинной последовательности слов из $B(m, T)$. Тогда

$$Q(t, n) = \frac{1}{m|B(m, T)|} \sum_{r=0}^{k_i} r \cdot |C(t, r)| + \frac{\sum_{i=1}^{n-1} \varphi_e(b_1 b_2 \dots b_i) \cdot \varphi_b(b_{i+1} \dots b_n)}{m}.$$

Отметим, что правомерность использования в приведённой формуле частот появления определённых последовательностей в начале или в конце слов языка обусловлена предположением, что в длинном предложении все слова языка распределены равномерно.

Приведённые рассуждения позволяют задать формальный язык из слов небольшой длины, поскольку табулирование большого количества слов представляет собой серьёзную проблему. Например, для бинарных слов длины 50 необходимо задать 2^{50} слов, занумеровав их способом,

изложенным выше. Разумеется, такие языки построить перечислением не получится, для их построения необходимо использовать другие методы. Зачастую для этого подходят комбинаторные методы, с помощью которых можно установить формальную связь одного множества слов с другим. Однако, если для языков с жёсткими запретами разработаны формулы, позволяющие определить количество слов, не содержащих заданных паттернов, в зависимости от длины этих слов, то для построения языков со слабыми запретами требуется решить более общую комбинаторную задачу на поиск числа слов, содержащих заданное количество определённых паттернов. В общем случае подобные комбинаторные задачи не решены, хотя известны некоторые частные их решения, направленные на подсчёт слов, содержащих заданное количество триплета t . Подход для триплетов данных обусловлен прикладными потребностями и, как мы увидим далее, представляет собой интересный случай для анализа свойств в случае, когда длина слова бесконечно возрастает.

3. Построение и анализ блочного кода, удаляющего 101

Руководствуясь прикладными соображениями, остановимся на аспектах применения кода и его свойствах применительно к частному случаю, связанному с удалением из сообщения триплетов 101. Это будет код $B(m, \{(101, k)\})$; кодовая таблица, следовательно, будет построена по возрастанию количества 101 в кодовых словах. Следовательно, свойства кода определяются двумя параметрами — длиной блока и максимальным количеством 101.

Здесь и далее будем полагать, что незакодированное сообщение обладает равномерным распределением, поэтому $Q_i = 1/8$ для каждого триплета. Тогда избыточность кода для любых конечных m и k можно определить по формуле (2). Особый интерес представляет нахождение энтропии при $m \rightarrow \infty$. Для её нахождения введём параметр ε : $0 \leq \varepsilon \leq 1$, $Q_{101} = (1 - \varepsilon)/8$. Таким образом, значение $\varepsilon = 1$ соответствует некоторому коду, полностью удаляющему из сообщения триплет 101. Напротив, $\varepsilon = 0$ означает, что сообщение остаётся незакодированным. Поставим задачу найти энтропию блочного кода с бесконечно большой длиной блока в зависимости от параметра ε .

Для этого применим теорию марковских процессов и представим процесс кодирования в наглядной форме (рис. 1). На рисунке вершины графа соответствуют состояниям процесса, который содержит три последних передаваемых символа, включая текущий передаваемый символ. Передача следующего символа показана как переход из текущего состояния в следующее со сдвигом последних двух битов и присоединением

к ним нового бита (0 или 1) с вероятностью, которая зависит от текущего состояния. Все переходы на диаграмме в общем случае возникают с вероятностями, отличными от $1/2$, которая соответствует случайному битовому потоку без какого-либо кодирования. Различие между двумя вероятностями, определяющими выход из одного и того же состояния k , равно $\varepsilon_k = T_{k,2k \bmod 8} - T_{k,(2k+1) \bmod 8}$. Таким образом, ε_k является мерой асимметрии кодировщика по данному триплету k .

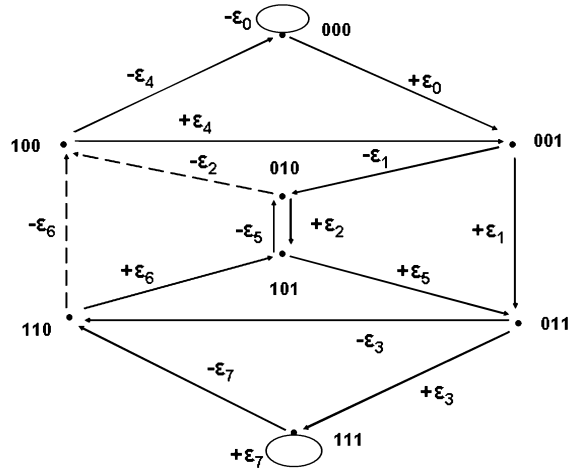


Рис. 1. Модель, иллюстрирующая процесс кодирования

Представленная на рис. 1 схема согласно теории цепей Маркова должна удовлетворять условию: $\hat{T}\vec{Q} = \vec{Q}$, где $\hat{T} = T_{k,l}$ — матрица перехода для марковского процесса, показанного на рисунке. Решение этого уравнения, нормированное равенством $\sum_{k=0}^7 Q_k = 1$, даёт частоту появления триплета k в бесконечно длинной строке битов как функцию параметров ε_k . Зависимости $Q_i = f_i(\vec{\varepsilon})$ известны [4]. Очевидно, что сама цепь Маркова полностью определяется вектором $\vec{\varepsilon} = (\varepsilon_0, \varepsilon_1, \dots, \varepsilon_7)$ и всякий код, меняющий статистику в равномерно распределённом сообщении, статистически определяется данным вектором, что позволяет определить избыточность кода по вектору $\vec{\varepsilon}$.

Найдём вектор $\vec{\varepsilon}$, соответствующий построенному выше блочному коду для удаления триплета 101. Для этого обозначим через $P(x)$ вероятность появления битовой последовательности x в закодированном сообщении. Например, $P(0111)$ — это вероятность того, что четыре последовательных бита в закодированных данных окажутся в точности 0111.

Кроме того, обозначим через $S(m, k)$ набор m -битных строк, в каждой из которых имеется ровно k триплетов 101.

Лемма 1. $P(00000) = P(00100)$ для любых $k \geq 0$ и $m \geq 5$.

ДОКАЗАТЕЛЬСТВО. Поскольку замена паттерна 00000 на 00100 (как и обратная замена) в блочном коде не меняет количества 101 в последовательности $B \in S(m, k)$, эти замены переводят кодовое слово $B \in S(m, k)$ в другое кодовое слово $B' \in S(m, k)$. Лемма 1 доказана.

Аналогично можно доказать, что

$$\begin{aligned} P(00001) &= P(00011), & P(10001) &= P(10011), \\ P(00010) &= P(00110), & P(11101) &= P(11011), \\ P(001100) &= P(001000), & P(000010) &= P(001110). \end{aligned}$$

Теорема 1. Справедливо равенство

$$\vec{\varepsilon} = \varepsilon_0 \left(1, -1, \frac{\varepsilon_0^2 - 2\varepsilon_0 + 5}{(1 + \varepsilon_0)^2}, -1, 1, -1, \frac{\varepsilon_0^2 - 2\varepsilon_0 + 5}{(1 + \varepsilon_0)^2}, -1 \right). \quad (3)$$

ДОКАЗАТЕЛЬСТВО. Так как для каждого значения ε кодовое пространство содержит по крайней мере одно кодовое слово с паттернами 0000, 0001, 1000, 1001, 1110, 1111, 0110, 0111, то справедливы неравенства $|\varepsilon_0| \neq 1$, $|\varepsilon_4| \neq 1$, $|\varepsilon_3| \neq 1$ и $|\varepsilon_7| \neq 1$ (в противном случае переход по марковской цепи в допустимые состояния невозможен). Поэтому для нашего блочного кода выполняются равенства [4]:

$$Q_1 = Q_4 = \frac{1 + \varepsilon_0}{1 - \varepsilon_4} Q_0, \quad Q_3 = Q_6 = \frac{1 - \varepsilon_7}{1 + \varepsilon_3} Q_7,$$

которые имеют смысл для любого ε . Пользуясь леммой и принимая во внимание вероятности перехода в марковской цепи, можно получить, например, что $\varepsilon_1 = -\varepsilon_0$, это следует из равенств $P(00001) = Q_0 P_{0 \rightarrow 0} P_{0 \rightarrow 1}$, $P(00011) = Q_0 P_{0 \rightarrow 1} P_{1 \rightarrow 3}$. Аналогично получаются другие соотношения между ε_i , которые в итоге приводят к равенству (3).

Тем самым мы выразили каждый параметр асимметрии через один параметр — ε_0 . Принимая во внимание выражение для $\vec{\varepsilon}$, а также выражение для вероятности триплетов 101 для однопараметрического кода, получим равенство

$$Q_5 = \frac{\varepsilon_0^4 + 6\varepsilon_0^2 + 8\varepsilon_0 + 1}{8(2\varepsilon_0^2 + \varepsilon_0 + 1)} = \frac{1 - \varepsilon}{8}.$$

Решение данного уравнения четвёртой степени даст нам выражение ε_0 как функцию ε . Само уравнение можно записать в виде

$$\varepsilon_0^4 + 2\varepsilon_0^2(3 - 8Q_5) + 8\varepsilon_0(1 - Q_5) - 8Q_5 + 1 = 0.$$

Стандартными методами показывается, что уравнение имеет ровно один корень из интервала $-1 \leq \varepsilon_0 \leq 1$ для любого $0 \leq Q_5 \leq 1/8$. Поскольку никаких технических проблем с численным решением подобных уравнений нет, ε_0 может быть найден для всякого $\varepsilon \in [0; 1]$.

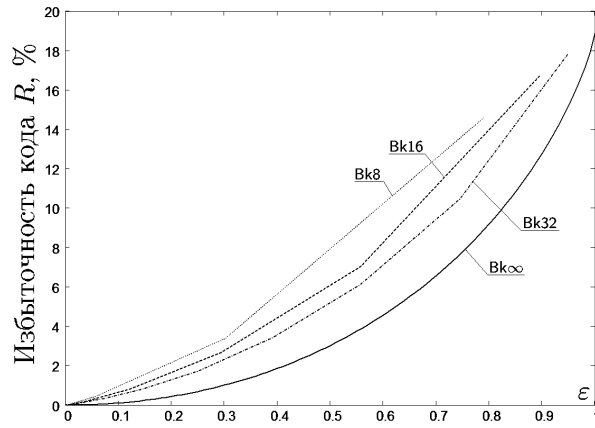


Рис. 2

Выразим энтропию кода через ε_0 . Используя общую формулу из [4] и учитывая, что $\vec{\varepsilon}$ является функцией ε_0 , получим

$$h(\varepsilon_0) = - \sum_{k=0}^7 Q_k(\vec{\varepsilon}(\varepsilon_0)) f(\vec{\varepsilon}(\varepsilon_0)),$$

где

$$f(x) = \frac{1+x}{2} \log_2 \frac{1+x}{2} + \frac{1-x}{2} \log_2 \frac{1-x}{2}.$$

Используя формулы для Q_k из [4] и равенство (3), получаем

$$h(\varepsilon_0) = f(\varepsilon_0)(Q_0(\varepsilon_0) + Q_1(\varepsilon_0) + Q_3(\varepsilon_0) + Q_4(\varepsilon_0) + Q_5(\varepsilon_0) + Q_7(\varepsilon_0)) + f(\varepsilon_2(\varepsilon_0))(Q_2(\varepsilon_0) + Q_6(\varepsilon_0)).$$

С использованием приведённых выше формул построим графики $R(\varepsilon)$ (рис. 2) для различных блочных кодов (BkX обозначает блочный код с

длиной блока X бит; V_{∞} — предельный случай блочного кодирования с бесконечно большим размером блока). Четвёртая кривая показывает, какое *минимальное* значение избыточности будет у кода, снижающего частоту появления триплета 101 до уровня $Q_5 = \frac{1-\varepsilon}{8}$. Любое значение избыточности на данной кривой является теоретическим минимумом потому, что блочный код использует все возможные кодовые слова, что делает невозможным в теории превзойти его характеристики.

Кроме того, на рис. 2 приведены кривые, полученные путём перебора всех возможных кодовых слов на длинах блоков 8, 16 и 32 бита. Анализируя рис. 2, можно заметить, что удаление половины триплетов 101 требует по крайней мере 3% избыточности, а на практике при использовании 32-битного блочного кода такой уровень удаления достигается при коде с 4,5%-ной избыточностью. Эти уровни избыточности небольшие, в отличие от уровня в 18,9%, который требуется для полного удаления 101. Такая нелинейная зависимость позволяет предположить, что в ряде случаев частичное удаление триплета 101 будет целесообразнее его полного удаления.

4. Адаптивный блочный код

Коды, изложенные выше, строились на базе *заранее заданного* набора из паттернов, появление которых в сообщении является нежелательным. Однако зачастую паттерн-эффект имеет сложную структуру, когда ярко выраженных нежелательных паттернов нет, а есть несколько последовательностей, которые передаются значительно хуже других. В этих условиях простое выделение «плохих» паттернов является неэффективным решением.

Рассмотрим на базе теории, введённой в разд. 2, общий случай асимметричной статистики ошибок. Пусть $\vec{R} = (R_{000}, R_{001}, \dots, R_{111})$ — вектор, компонентами которого являются относительные частоты появления ошибки при передаче *центрального бита* каждого триплета. Определим язык $AC(m, \vec{R}, \delta)$ следующим образом:

$$AC(m, \vec{R}, \delta) = \left\{ a \mid a = a_1 a_2 \dots a_m, P_{ne}(a) = \prod_{i=1}^{m-2} (1 - R_{a_i a_{i+1} a_{i+2}}) \geq \delta \right\}.$$

Здесь $P_{ne}(a)$ — вероятность того, что внутренние символы слова a будут переданы без ошибок, $0 \leq \delta \leq 1$ — параметр, показывающий, насколько допустимы ошибки при передаче слов языка AC . Упорядочение слов языка в последовательность W идёт по убыванию вероятностей безошибочной передачи слов. Таким образом, если $P_{ne}(W(i)) \geq P_{ne}(W(j))$,

то $i \leq j$. Относительная избыточность построенного языка определяется выражением, подобным выражению (2):

$$R = \frac{m - \log_2 |\text{AC}(m, \vec{R}, \delta)|}{m}. \quad (4)$$

Частота появления каждого триплета определяется как в разд. 2:

$$Q_{xyz} = \frac{1}{m \cdot 2^m} \sum_{j=0}^{2^m} N_{xyz}(W(j)) + \frac{\varphi_{ex}\varphi_{byz} + \varphi_{exy}\varphi_{bz}}{m}, \quad (5)$$

где $N_{xyz}(W(j))$ — количество триплетов xyz внутри кодового слова $W(j)$, φ_{byz} , φ_{bz} , φ_{ex} и φ_{exy} — частоты появления блоков, которые начинаются с yz и z и заканчиваются на x и xy соответственно.

К преимуществам данного кода следует отнести его простоту, поскольку как кодирование, так и декодирование основаны на простом извлечении кодовых слов из таблиц данных. Среди других кодов, которые имеют целью преодоление паттерн-эффекта, этот код отличается способностью адаптироваться к любой линии связи, обеспечивая оптимальный выбор множества кодовых слов для передачи данных. Среди недостатков следует отметить ограниченность в выборе длин блока данных, вызванную невозможностью строить блоки большой длины, обладающие описанными свойствами.

5. Пример построения кода

Рассмотрим процесс построения кода на следующем примере. В [3] путём прямого численного моделирования передачи данных по волоконно-оптической линии связи со спектральным уплотнением каналов и кодированием информации по разности оптических фаз (RZ-DPSK) была установлена статистика ошибок по каждому из триплетов, т. е. для заданной линии в 5 каналов был определен вектор \vec{R}_j , $1 \leq j \leq 5$. Во время моделирования по каждому каналу было передано 800000 бит, значения битов были распределены равномерно. Таким образом, в силу представительности выборки данных можно полагать, что каждый триплет появился в среднем около 100000 раз. Статистика ошибок по каждому триплету [6] представлена в табл. 1.

Как видно, наибольшее число ошибок наблюдается при передаче триплетов 010 и 101, однако триплеты 001 и 100 тоже весьма ощутимо влияют на частоту появления ошибок. Именно поэтому в данном случае эффект от кода, ликвидирующего только один триплет, будет неполным.

Построим код $AC(16, \vec{R}, \delta)$ для различных значений $\delta \in (0, 1)$ путём перебора 2^{16} кодовых слов и для каждого значения посчитаем вероятность передачи данного кодового слова без ошибок:

$$P_{ne} = \prod_i (1 - R_{T_i}).$$

В данном случае $R_i = E_i/N_i$, где E_i — количество ошибок, полученных при передаче триплета i , а N_i — количество триплетов i , переданных по каналу.

Т а б л и ц а 1

Статистика ошибок по каналам передачи данных

Триплет	Канал 1	Канал 2	Канал 3	Канал 4	Канал 5
000	0	2	4	2	6
001	1216	1353	1471	1559	1604
010	7081	7154	7783	8299	8470
011	111	123	181	190	192
100	1064	1250	1286	1422	1464
101	7650	8029	8590	9298	9354
110	94	122	134	145	168
111	42	53	70	60	93

Для оценки результатов применения кода для каждого δ можно найти частоту битовых ошибок в канале BER (Bit Error Rate):

$$BER = \sum_{i=0}^7 R_i Q_i.$$

Значения BER для каждого канала приведены в табл. 2. Для закодированного сообщения формула нахождения BER остаётся прежней, но расчёт Q_i ведётся по формуле (5). Кроме того, для наглядности результатов применения кода введён параметр $\rho = BER(\delta)/BER(0)$, показывающий во сколько раз уменьшилось число ошибок благодаря адаптивному блочному коду.

Т а б л и ц а 2

Частота битовых ошибок в каналах

	Канал 1	Канал 2	Канал 3	Канал 4	Канал 5
$BER \cdot 10^3$	21,57	22,6	24,4	26,22	26,69

6. Результаты применения кода

По описанной методике на основе статистики ошибок из табл. 1 построен блочный код с длиной блока (длиной кодового слова) 16 бит. На рис. 3 показана зависимость избыточности кода от значения δ . Очевидно, что уже при малых значениях избыточности кода (до 10%) наблюдается существенное увеличение вероятности безошибочной передачи блока данных.

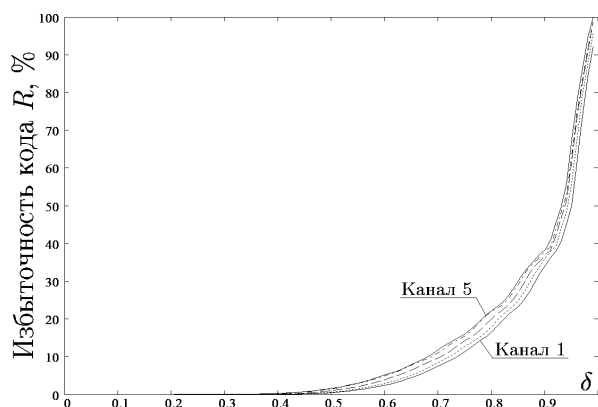


Рис. 3. Зависимость избыточности 16-битного адаптивного блочного кода от δ

Большей частотой ошибок при малых δ обладает канал 5, что естественно, так как число ошибок в этом канале изначально является наибольшим (см. табл. 1).

Рис. 4 иллюстрирует, какую избыточность необходимо добавить в сообщение, чтобы уменьшить BER в заданное число раз. Особо следует отметить экспоненциальный характер падения числа ошибок. Даже при малой избыточности кода эффект от его применения более чем ощутимый: так, избыточность 12% уменьшает BER в 2 раза ($\rho = 0,5$). Дальнейшее же увеличение избыточности уже не приводит к существенному сокращению частоты ошибок. Это позволяет сделать вывод, что описанный код может дополнить коды, корректирующие ошибки, так как, добавляя небольшую избыточность в исходное сообщение, он сделает его значительно более стойким к нелинейным помехам в линии связи, а это, в свою очередь, облегчает работу корректирующей схеме и повышает её эффективность. Кроме того, большинство корректирующих кодов имеют предел в виде некоторого предельного значения BER_l , за которым характеристики корректирующей схемы резко падают. Поэтому искусственное снижение BER, вызванное применением адаптивного кода, может в ряде

случаев сделать возможным работу корректирующей схемы в условиях, когда с естественным уровнем частоты ошибок в канале корректирующая схема не может справиться.

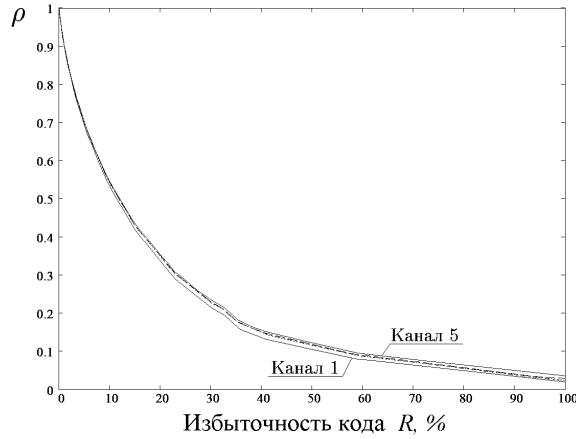


Рис. 4. Отношение вероятности появления ошибок в исходном сообщении к вероятности появления ошибок в закодированном сообщении

7. Эффект от применения кода

Для того чтобы численно выразить эффект от построенного кода, введём в качестве мерил эффективности величину

$$\Gamma = 20 \log_{10} \frac{A(0)}{A(\delta)},$$

где $A(\delta)$ — это среднеквадратичное значение амплитуды аддитивного гауссова шума, действие которого на сигнал эквивалентно действию на сигнал нелинейных эффектов, порождающих неравномерную статистику ошибок. Здесь под эквивалентностью понимается равенство частот появления ошибок в канале. Таким образом, значение Γ показывает, насколько увеличивается отношение «сигнал-шум» при использовании построенного выше адаптивного кода с параметром δ относительно незакодированного сигнала (сигнала, закодированного с $\delta = 0$). В случае гауссова шума справедливо соотношение

$$\text{BER}(\delta) = \frac{1}{2} \cdot \Phi \left(\frac{1}{A(\delta)\sqrt{2}} \right),$$

где $\Phi(x)$ — дополнительная функция ошибок. Пользуясь данным соотношением, можно для всякого $0 < \text{BER}(\delta) < 1/2$ найти амплитуду гауссова

шума, действие которого на сигнал приведёт к частоте ошибок в канале, равной BER.

На рис. 5 показана зависимость эффективности кода от его избыточности. Видно, что технический эффект при применении кода с небольшой избыточностью до 10% весьма резко возрастает, достигая на 10% величины 1 дБ. И хотя при дальнейшем увеличении избыточности рост продолжается, применение чрезмерно избыточных кодов нецелесообразно, поскольку они, в отличие от корректирующих кодов, сами ошибки не правят, а лишь помогают коррекции.

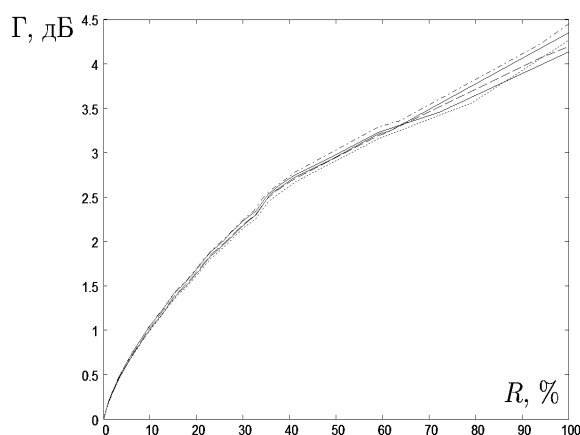


Рис. 5. Зависимость эффективности кода от избыточности, вносимой кодом

8. Заключение

Итак, использование построенных адаптивных блочных кодов с ограничениями позволяет снизить частоту ошибок при передаче сообщения, что делает сообщение значительно более стойким к нелинейностям в линии связи, проявляемым в виде паттерн-эффекта. Построенная общая концепция слабоограниченного блочного кода способна описать любой блочный код с заданным характером ограничений, а также дать формальные методы для расчёта его вероятностных и энтропийных характеристик. Приведённая схема анализа асимптотических свойств кодов со слабыми ограничениями может быть использована для нахождения энтропии кодов в случае различных ограничительных условий.

Снижение частоты ошибок при передаче сообщения позволяет говорить о потенциальной сочетаемости построенных кодов с кодами, корректирующими ошибки. Особенно целесообразно такое сочетание в случаях, когда схема коррекции ошибок работает на грани своих возможностей,

поскольку тогда предложенный способ кодирования позволит существенно усилить возможности коррекции, так как большинство корректирующих кодов имеют свойство размножать ошибки в случаях, когда их становится больше некоторого порогового значения.

ЛИТЕРАТУРА

1. **Agrawal G. P.** Nonlinear fiber optics. — New York: Acad. Press, 2001. — 467 p.
2. **Djordjevic I., Vasic B., Rao V.** Rate 2/3 modulation code for suppression of intrachannel nonlinear effects in high-speed optical transmission // IEEE Proc. Optoelectronics. — 2006. — Vol. 153(2). — P. 87–92.
3. **Immink K. A. S.** A practical method for approaching the channel capacity of constrained channels // IEEE Trans. Inf. Theory. — 1997. — Vol. 43, N 5. — P. 1389–1399.
4. **Shafarenko A., Turitsyn K. S., Turitsyn S. K.** Information-theory analysis of skewed coding for suppression of pattern-dependent errors in digital communications // IEEE Trans. Commun. — 2007. — Vol. 55(2). — P. 237–241.
5. **Shapiro E. G., Fedoruk M. P., Turitsyn S. K., Shafarenko A.** Reduction of nonlinear intrachannel effects by channel asymmetry in transmission lines with strong bit overlapping // IEEE Photonics Technology Lett. — 2003. — Vol. 15(10). — P. 1473–1475.
6. **Turitsyn S. K., Fedoruk M. P., Shtyrina O. V., Yakasov A. V., Shafarenko A., Desbruslais S. R., Reynolds K., Webb R.** Patterning effects in a WDM RZ-DBPSK SMF/DCF optical transmission at 40 Gbit/s channel rate // Optics Communications. — 2007. — Vol. 277(2). — P. 264–268.
7. **Vasic B., Rao V., Djordjevic I., Kostuk R., Gabitov I.** Ghost-pulse reduction in 40-Gb/s systems using line coding // IEEE Photonics Technology Lett. — 2004. — Vol. 16(7). — P. 1784–1786.

Скидин Антон Сергеевич,
e-mail: ask@skidin.org
Федорук Михаил Петрович,
e-mail: mife@ict.nsc.ru
Шафаренко Александр Владимирович,
e-mail: A.Shafarenko@herts.ac.uk
Турицын Сергей Константинович,
e-mail: S.K.Turitsyn@aston.ac.uk

Статья поступила
1 декабря 2009 г.
Переработанный вариант —
29 марта 2010 г.