

ПОМЕХОУСТОЙЧИВОЕ КОДИРОВАНИЕ МУЛЬТИМЕДИА ДАННЫХ В КОМПЬЮТЕРНЫХ СЕТЯХ

К.В. Шинкаренко, А.М. Корикив*

Отдел проблем информатизации ТНЦ СО РАН, г. Томск

*Томский государственный университет систем управления и радиоэлектроники

E-mail: shinkarenko_k@mail.ru, korikov@asu.tusur.ru

Помехоустойчивое кодирование повышает достоверность доставки сообщений, передаваемых по компьютерным сетям, однако классические помехоустойчивые коды не компенсируют потерю пакетов мультимедиа данных в IP сетях. Проанализированы возможности применения стирающих кодов для восстановления пакетов при их потерях в сети. Сделан вывод о том, что применение стирающих кодов для кодирования мультимедиа данных в компьютерных сетях имеет перспективы, но требует усовершенствования. Предложены новые неслучайные стирающие помехоустойчивые коды для уменьшения потерь мультимедиа данных в IP телевидении.

Ключевые слова:

Мультимедиа, помехоустойчивое кодирование, цифровое телевидение, стирающие коды, ЛТ коды, неслучайные стирающие коды.

Введение

Мультимедиа данные, как и любые другие сообщения, передаваемые в компьютерных сетях, представляют собой файлы конечного размера. Файлы пересылаются по сети последовательностью пакетов, размер которых определяется стандартами и существующими архитектурами сети. Желательно, чтобы получатель принял каждый из пакетов без ошибок. Однако часть пакетов может быть доставлена получателю с ошибочными битами, а часть — потеряна в сети. Такие пакеты в теории помехоустойчивого кодирования считают стертными, а в качестве базовой математической модели для описания канала с помехами в компьютерных сетях используется модель канала со стираниями (*erasure channel*) пакетов. Для обнаружения получателем ошибок в пакетах отправитель использует код.

Для любого кода существует вероятность того, что ошибка не будет обнаружена. Ошибка не будет обнаружена в случае, если переданное кодовое слово было преобразовано в канале в другое кодовое слово. Это преобразование принято называть трансформацией кодовых слов. В модели канала со стираниями пренебрегают вероятностью ошибочно принятого пакета, обусловленной трансформацией кодовых слов. Для большинства сетевых приложений (IP-вещание, IP Multicast-сервис, одновременная передача данных с нескольких сайтов в Интернет и т. д.) такое пренебрежение оправдано чрезвычайно малой вероятностью трансформации. Каждое сетевое применение имеет свою специфику, и в данной статье в качестве сетевого приложения рассматривается IP телевидение.

Новые способы доставки данных и архитектуры систем телевидения по сетям IP (далее кратко IPTV, Internet Protocol Television) породили принципиально новые трудности. Одной из наиболее острых и актуальных проблем является проблема потерь пакетов данных в системах IPTV реального времени, или «живого» вещания. Потери данных ведут к деградации качества восприятия телепрограмм: потере звука, рассыпанию изображения на раздражаю-

щие «квадратики», десинхронизации звука и изображения. Суть проблемы в том, что для «живого» телевидения приходится использовать протоколы IP с негарантированной доставкой пакетов, такие как UDP (User Datagram Protocol), RTP (Real Time Protocol), поскольку повторная передача пакетов (так называемая схема ARQ, Automatic Repeat reQuest) невозможна из-за больших задержек на данную операцию, недопустимых при «живом» телевидении. Кроме того, запросы на пересылку, особенно в топологии «один — многим», приведут к дополнительным нагрузкам на канал связи и, возможно, дальнейшей деградации сервиса. Потери пакетов в негарантированных протоколах происходят из-за коллизий, задержек в сетях, «проседания» пропускной способности каналов и т. п.

Отметим, что классические помехоустойчивые коды не способны решить названную проблему. Сверточные коды исправляют отдельные битовые ошибки. Блочные коды, в том числе коды Рида-Соломона, способны исправлять пачки ошибок в отдельно взятом пакете. Но в случае потери пакета целиком они бессильны.

В настоящее время можно говорить о создании нового класса помехоустойчивых кодов для каналов со стираниями — стирающих кодов. Кодами из этого класса можно закодировать сообщение конечного размера потенциально неограниченным потоком независимых пакетов. Это свойство нового класса кодов принципиально отличает его от классических блочных или сверточных, помехоустойчивых кодов с заданной скоростью (*fixed rate*). Для кодов из нового класса появился термин «*rateless*» (нефиксированной скорости). Их называют также фонтанными кодами (Digital Fountain Codes). Исторически первым и наиболее идейно интересным *rateless* кодом считается ЛТ код. Код был создан М. Лаби (Michael Luby) в 2002 г. и наиболее подробно описан в [1]. Свое название он получил от «*Luby Transform*» (Лаби Трансформ, ЛТ).

Важнейшим отличием стирающих кодов от блочных и сверточных кодов является возмож-

ность восстановления целого пакета в случае его потери. В настоящее время стирающие коды уже нашли применение в коммерческих продуктах для компьютерных сетей, поставляемых, к примеру, компанией Digital Fountain.

Анализ имеющихся на данный момент источников показал, что имеются лишь единичные попытки применения стирающих кодов IPTV [2]. Говорить о более-менее целостной технологии IPTV с применением стирающих кодов пока не приходится. Можно выделить лишь стандарт мобильного телевидения DVB-H (Digital Video Broadcasting Handheld) [3], который подразумевает использование стирающих кодов Raptor [4], разработанных на основе кодов ЛТ с учетом специфики мобильного телевидения.

Данная статья посвящена анализу возможности применения стирающих кодов в системах IPTV. В статье предлагаются новые помехоустойчивые стирающие коды для уменьшения потерь мультимедиа данных в IP телевидении.

Принципы построения стирающих кодов. Коды ЛТ

Допустим, что мы имеем исходное сообщение, состоящее из K исходных символов, или пакетов. Длины всех символов эквивалентны и равны L . Кодовые символы в стирающих кодах генерируются как результат операции «исключающее или» (XOR) над d исходными символами. Величину d называют *степенью* кодового символа. Данная величина может принимать значения от 1 до некоторого максимального значения G .

Исходные символы, использованные для генерации некоторого кодового символа i , будем называть *соседями* (*neighbors*) [1].

Введем также распределение вероятностей степеней $p(d)$: $p(d)$ есть вероятность того, что кодовый символ будет иметь степень d . Распределение степеней может быть задано как аналитически, так и таблично в виде набора значений $p(1), p(2), \dots, p(G-1), p(G)$.

Далее приведено описание процесса генерации кодовых символов для кода ЛТ согласно [1] в нашем переводе:

- случайным образом выбрать степень d кодового символа, используя распределение вероятностей степеней $p(d)$;
- выбрать случайным образом d различных исходных символов в качестве соседей кодового символа;
- значение кодового символа задать равным результату операции XOR над d выбранными соседями.

Возникает вопрос, сколько кодовых символов необходимо сгенерировать для восстановления исходного сообщения. Поскольку кодовые символы являются независимыми друг от друга, то потенциально процесс генерации может длиться сколько угодно долго до тех пор, пока все пользователи не

получат достаточного количества кодовых символов для восстановления исходного сообщения полностью. В этом заключается «фонтанное» свойство ЛТ кодов. Пример порождающего графа ЛТ кода приведен на рис. 1.

Теоретически для успешного восстановления всех символов исходного сообщения должно хватить $W=K \cdot \ln(R/\delta)$ кодовых символов, причем $1-\delta$ есть вероятность успешного восстановления всех исходных символов, K – количество исходных символов [4].

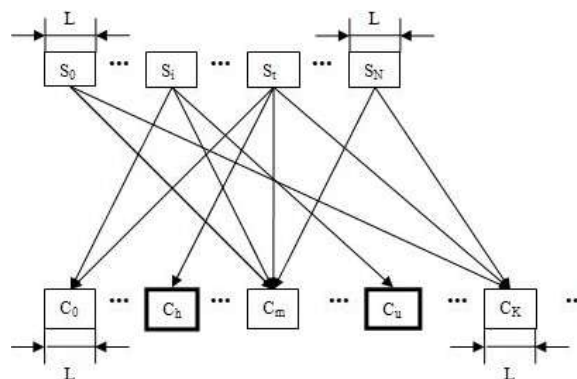


Рис. 1. Пример порождающего графа ЛТ кода

Приведем алгоритм декодирования ЛТ кодов [1] в нашей интерпретации.

Очередь всех полученных приемником кодовых символов обозначим через *ОКС*. Введем понятие *ООС* – очереди обрабатываемых символов, степень которых равна 1 на некоторой итерации декодирования. Очередь декодированных, или, корректнее, восстановленных символов назовем *ОВС*.

Изначально *ОВС* пуста. Среди *ОКС* находим все символы со степенью 1 и перемещаем их в *ООС*. Далее итеративно обрабатываем каждый i -й символ в *ООС* ($ООС_i$), производя следующие действия:

- восстанавливаем исходный символ, который является единственным «соседом» для $ООС_i$ в надлежащий символ в *ОВС*, т. е. копируем содержимое $ОВС_i$ символа в надлежащий исходный символ $ОВС_k$;
- находим среди кодовых символов в *ООС* символы, в которые i -й символ входит в качестве соседа. Производим операцию XOR над содержимым найденного символа в *ОКС* (обозначим его $ОКС_j$) и текущим i -м символом, уменьшаем степень $ОКС_j$ на 1.
- если степень $ОКС_j$ стала равна 1, и такого символа нет ни в *ОВС*, ни в *ООС*, то перемещаем $ОКС_j$ в *ООС*.
- удаляем $ООС_i$ из *ООС*.

Процесс считается завершенным успешно, если все исходные символы восстановлены. Алгоритм завершает свою работу неуспешно, если на каком-то шаге *ООС* оказывается пустой. Тогда некоторые исходные символы окажутся потерянными.

Чтобы предотвратить неудачную ситуацию, необходимо надлежащим образом разработать распределение степеней кодовых символов. Это является важнейшей задачей в проектировании стирающих кодов. Для ЛТ кодов М. Лаби предложил Robust Soliton Distribution – робастное распределение [1, 4].

Робастное распределение $\mu(d)$ задается формулой:

$$\mu(d) = \frac{p(d) + \tau(d)}{\sum_d p(d) + \tau(d)},$$

где

$$p(d) = \begin{cases} p(1) = 1/K, \\ \frac{1}{d(d-1)}, & d = 2, \dots, K, \end{cases}$$

$$\tau(d) = \begin{cases} \frac{S}{K} \frac{1}{d}, & d = 1, 2, \dots, (K/S) - 1 \\ \frac{S}{K} \log(S/\delta), & d = K/S \\ 0, & d > K/S \end{cases}$$

В последней формуле S определяется выражением

$$S(K, \delta) = c \ln(K/\delta) \sqrt{K}.$$

Параметрами робастного распределения являются: K – количество исходных символов; c – параметр распределения; задается как некоторое положительное число, рекомендуется задавать $0 < c < 1$; Процесс декодирования будет успешно завершён с вероятностью $1 - \delta$.

Пример робастного распределения приведен на рис. 2.

Анализ возможности применения ЛТ кодов в IPTV

Сформулируем основные свойства ЛТ кодов.

1. Кодовые символы генерируются независимо друг от друга; это так называемое «фонтовое» свойство ЛТ кодов – для восстановления исходного сообщения приемником могут быть использованы любые полученные кодовые символы.
2. Исходные символы имеют равный приоритет.
3. Длина исходных символов может быть любой, но длины всех исходных символов равны. Длины кодовых символов также равны длине исходных символов.
4. Ввиду того, что в основе теории кодов ЛТ лежит статистическая задача о мячах и корзинах, эффективность ЛТ кодов проявляется при достаточно больших значениях параметра K , т. е. большом числе исходных символов [4]. Изобретателями ЛТ кодов рекомендуются значения K порядка 10000 [1].

Перечисленные свойства ЛТ кодов очень ценны и позволяют использовать ЛТ коды на практике для любых типов передаваемых данных. Важным является свойство фонтовности кодов, в особенности для задачи «многие – многим» [5].

Однако, названных свойств стирающих кодов недостаточно для успешного решения задач передачи цифровых мультимедиа данных, и, в частности, для систем online телевидения.

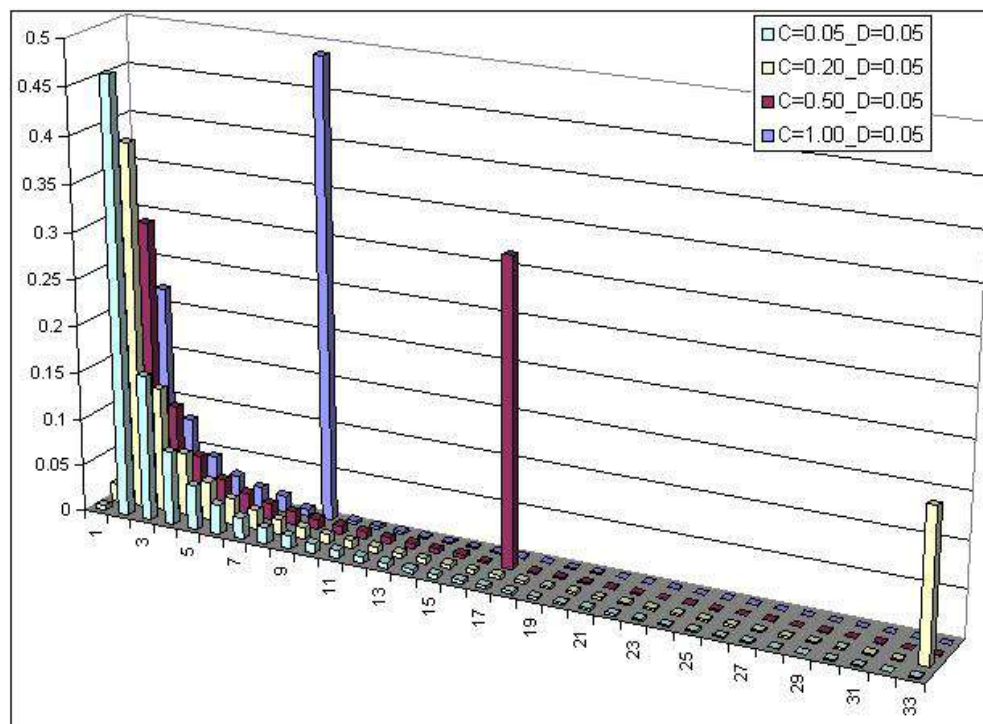


Рис. 2. Робастное распределение степеней кодовых символов $\mu(d)$. Параметры распределения: $K=10000$, $c=0,05\dots1,0$, $\delta=0,05$

Назовем основные проблемы применения кодов ЛТ в системах IPTV.

- Неприемлемо долгая задержка телевизионного сигнала между приемником и передатчиком, вызванная необходимостью буферизации большого числа исходных пакетов, если речь идет о величинах K порядка 10000. В случае загрузки медиафайла в отложенном режиме это не является проблемой, но в режиме реального времени это повлечет задержку в несколько десятков секунд [6].
- Равный приоритет исходных пакетов не является наилучшим решением для систем IPTV, т. к. пакеты, содержащие данные видеок кадров различных типов, аудиоданные, вносят различный вклад в формирование общего качества восприятия телепрограмм. Наибольшую проблему представляют потери звука. Существенную деградацию качества видео влечет потеря данных опорных I -кадров.
- Свойство фонтанности не играет столь существенной роли для систем online IPTV, т. к. вещание в таких системах подчиняется жестким временным требованиям по получению, декодированию и воспроизведению кадров, и отсутствует возможность ожидания очередных «брызг» (кодовых пакетов).

Тем не менее, свойство фонтанности может быть использовано для разгрузки каналов головных серверов вещания, т. к. в случае применения стирающих кодов любой узел в сети вещания может работать одновременно, как в режиме приема кодовых символов, так и их немедленной ретрансляции близлежащим узлам.

Таким образом, в силу основного свойства стирающих кодов – способности исправления пакетов при потере, стирающие коды имеют перспективы применения в системах телевещания по сетям IP, однако требуется найти решение отмеченных выше проблем применения.

Неслучайные стирающие коды

Для решения вышеозначенных проблем были предложены новые стирающие коды, названные «неслучайные стирающие коды» (НССК). Впервые работа, посвященная НССК, была опубликована в [4]. Мотивацией к созданию НССК стало то, что при низких значениях K не работают статистические свойства, составляющие основу теории ЛТ кода. Поэтому необходимо контролировать количество вхождений исходных символов в кодовые пакеты, обеспечивая большую робастность и повышая эффективность кода.

Код НССК использует модифицированное робастное распределение (Robust Soliton Distribution [1, 4, 5]) $\mu(d)$. Робастность модифицированного распределения дополнительно повышена путем увеличения числа пакетов со степенью 3 на 30 % от числа пакетов со степенью 1 (выведено эмпириче-

ски). В текущей реализации НССК количество кодовых символов ограничено конкретной величиной N , определяемой исходя из максимально дозволённого переизбытка кодовых пакетов относительно числа K исходных символов как $N=K\beta$, где β – величина, немногим более 1 (задается исходя из особенностей канала передачи).

Очередь исходных символов обозначим кратко *ОИС*, как и ранее. Очередь кодовых символов обозначим *ОКС*. Будем использовать индексы для обращения к символам в *ОКС* и *ОИС*. Количество вхождений исходного символа f в кодовые символы в качестве «соседа» назовем *мерой защищенности* исходного пакета. Обозначим меру защищенности переменной $R[f]$, $f=0, \dots, K-1$.

Ниже приведен алгоритм кодирования НССК.

Шаг 1. Определить количество кодовых символов, предназначенных для кодирования со степенями d_0, \dots, d_{\max} согласно:

$$n[d_i] = \mu(d_i) \cdot N,$$

где N – целевое количество кодовых символов.

Шаг 2. Выбрать $n[d_0]$ исходных пакетов с наивысшим приоритетом. Закодировать их как опорные символы со степенью 1. Можно записать это действие на языке псевдо-Си как

```
for(i=0; i < n[d_0]; i++)
{
    OKC[i] = OIS[f_i];
}
```

где f_i обозначает индекс исходного символа из *ОИС*.

Шаг 3. Произвести кодирование всех символов, закодированных на шаге 1, согласно следующей процедуре:

```
for(i=0; i < n[d_0]/3; i++)
{
    OKC[i+n[d_0]] = OIS[f_{3i}] XOR OIS[f_{3i+1}] XOR OIS[f_{3i+2}];
    R[f_{3i}] = 2; R[f_{3i+1}] = 2; R[f_{3i+2}] = 2;
}
```

Шаг 4. Обозначим через W_R подмножество всех исходных символов, обладающих на некотором шаге кодирования мерой защищенности R . Среди них подмножество символов с максимальной мерой защищенности обозначим как W_{\max} , с минимальной – W_{\min} .

Задать $i = 2$.

Шаг 4.1. Выполнять действия 4.1.1 и 4.1.2 $n[d_i]$ раз:

Шаг 4.1.1. Сгенерировать очередной k -й кодовый символ как результат операции XOR над одним символом из W_{\max} , одним из $W_{\max-1}, \dots$, одним из W_{L-1} , и одним символом из W_{\min} , где $L = \max-d_i$, т. е. выполнить действие:

$$OKC[k] = W_{\max}[j_0] XOR W_{\max-1}[j_1] XOR \dots XOR W_{L-1}[j_{L-1}] XOR W_{\min}[j_L];$$

Шаг 4.1.2. Увеличить на единицу значения $R[j_i]$ для всех $j_i, i=0...L$ (здесь индексы j_i являются индексами исходных пакетов в ОИС).

Шаг 4.2. Обновить подмножества W_R , а также максимальное и минимальное значения R . Увеличить i на единицу. Если i стало равным $d_{\max}+1$, либо общее число кодовых символов достигло желаемого количества N , выход. Иначе перейти на Шаг 4.1.

Отметим, что чем выше приоритет исходного пакета, выбираемого из W_{\min} , тем ниже должна быть степень кодового пакета, при которой он будет впервые вовлечен в кодирование.

Важнейшим является то, что строки матрицы индексов исходных пакетов J в идеале должны быть линейно независимыми, или, по крайней мере, иметь как можно меньше комбинаций линейно зависимых векторов-строк.

$$J = \begin{bmatrix} j_0^1 & \dots & j_{d_2-1}^1 & 0 & \dots & 0 \\ j_0^2 & \dots & \dots & j_{d_3}^2 & 0 & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots \\ j_0^{N-n[d_0]} & \dots & \dots & \dots & \dots & j_{d_{\max}}^{N-n[d_0]} \end{bmatrix}$$

Порождающим графом кода НССК является граф с систематической структурой. Пример подобного графа приведен на рис. 3. Пример алгоритма выбора индексов «соседей» из подмножеств W_R может быть найден в [6].

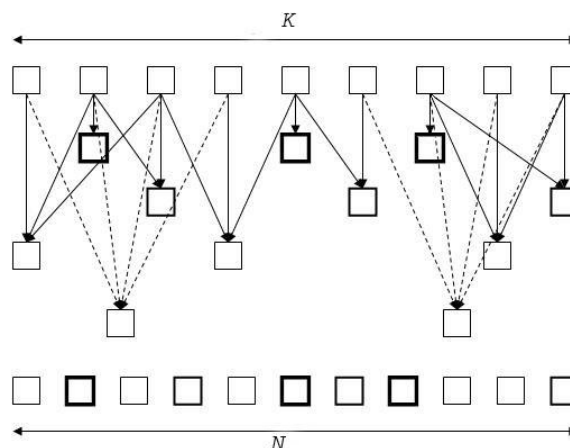


Рис. 3. Пример порождающего графа кода НССК

Заключение

В статье рассмотрены возможности применения стирающих кодов в системах IPTV. Выяснено, что стирающие коды обладают способностью восстанавливать данные при потерях пакетов, однако для применения в системах IPTV необходимы стирающие коды с определенными свойствами. Необходимыми свойствами обладают разработанные нами коды НССК. В работах [6, 7] была теоретически и экспериментально обоснована более высокая эффективность кодов НССК по сравнению с кодами ЛТ при значениях K порядка 100...1000.

Возможности применения НССК в системах IPTV анализируются в нашей статье [8].

СПИСОК ЛИТЕРАТУРЫ

1. Luby M. LT Codes // Proc. of the 43rd Annual IEEE Symp. on Foundations of Computer Science (FOCS). – 2002. – P. 271–282.
2. Argyropoulos S., Tan A.S., Thomos N., Arikani E., Srinivasan M.G. Robust Transmission of Multi-View Video Streams Using Flexible Macroblock Ordering and Systematic LT Codes // Proc. of the 3DTV Intern. Conf.: True-Vision, Capture, Transmission, and Display of 3D Video (3DTV-CON 07). – Kos Island, 2007 – P. 1–4.
3. ETSI TS 102 472: «Digital Video Broadcasting (DVB); IP Datacast over DVB-H: Content Delivery Protocols». V1.2.1 (2006-12).
4. MacKay D.J.C. Fountain codes // IEE Proc.-Commun. – 2005. – V. 152. – № 6 (December). – P. 1062–1068.
5. Варгаузин В. Помехоустойчивое кодирование в пакетных сетях // ТелеМультиМедиа. – 2005. – № 3. – С.10–16.
6. Shinkarenko K.V., Vlcek K. Design of Erasure Codes for Digital Multimedia Transmitting // In: Proc. of Design and Diagnostic of Electronic Circuits and Systems (DDECS08). – Bratislava, 2008. – P. 30–34.
7. Шинкаренко К.В. Применение стирающих кодов в системах цифрового телевидения. // Микроэлектроника и информатика-2008: Сб. науч. докл. 15-й Всеросс. науч.-техн. конф. – М., 2008. – С. 248.
8. Шинкаренко К.В., Кориков А.М. Применение неслучайных стирающих кодов в IP телевидении // Известия Томского политехнического университета. – 2008. – Т. 313. – № 5. – С. 42–46.

Поступила 29.09.2008 г.