УДК 621.391

Жураковський Б. Ю., к.т.н. *(Державний унів-т інформаційно-комунікаційних технологій)*

ДОСЛІДЖЕННЯ ВИКОРИСТАННЯ НОВИХ ЗАВАДОСТІЙКИХ КОДІВ ДЛЯ КАНАЛІВ ЗІ СТИРАННЯМ

Жураковський Б. Ю. Дослідження використання нових завадостійких кодів для каналів зі стиранням. Розглядається новий клас завадостійких кодів – фонтанні коди. Кодами цього класу можливо закодувати будь-яке повідомлення кінцевого розміру потенційно-необмеженим потоком незалежних пакетів. Ці коди мають прості алгоритми декодування і дозволяють на практиці отримувати результати, близькі до граничних можливостей завадостійкого кодування.

Ключові слова: ЗАВАДОСТІЙКИЙ КОД, КОДЕР, ОПТИМАЛЬНИЙ КОД, ФОНТАННИЙ КОД, ПОРОДЖУВАЛЬНИЙ ГРАФ, АНАЛІЗ КОНСТРУКЦІЇ

Жураковский Б. Ю. Исследование использования новых помехоустойчивых кодов для каналов со стиранием. Рассматривается новый класс помехоустойчивых кодов – фонтанные коды. Кодами этого класса можно закодировать любое сообщение конечного размера потенциально-неограниченным потоком независимых пакетов. Эти коды имеют простые алгоритмы декодирования и позволяют на практике получать результаты, близкие к предельным возможностям помехоустойчивого кодирования.

Ключевые слова: ПОМЕХОУСТОЙЧИВЫЙ КОД, КОДЕР, ОПТИМАЛЬНЫЙ КОД, ФОНТАННЫЙ КОД, ПОРОЖДАЮЩИЙ ГРАФ, АНАЛИЗ КОНСТРУКЦИИ

Zhurakovskyi B. Iu. Research of the use of new antijamming codes for channels with elimination. A new class of noiseproof coding – fountain codes. Code of this class can encode any message of finite size of potentially unlimited flow of independent packages. These codes have simple decoding algorithms and allow the practice to obtain results that are close to the limit of opportunities of noiseproof coding.

Key words: ANTIJAMMING CODES, CODER, DECODING ALGORITHM, OPTIMAL CODE, KNOTS OF COUNT, FOUNTAIN CODES, ORIGINATIVE COUNT, ANALYSIS OF CONSTRUCTION.

Сьогодні можна говорити про створення нового класу завадостійких кодів для каналів зі стиранням. Кодами з цього класу можна закодувати повідомлення кінцевого розміру потенційно-необмеженим потоком незалежних пакетів. Ця властивість нового класу кодів принципово відрізняє його від класичних блокових або згорткових, завадостійких кодів із заданою швидкістю. При кодуванні файлу цими кодами отримуємо також файл кодованих даних, а не потік. З цієї причини, для кодів з нового класу з'явився термін "rateless" на противагу класичним кодами, для яких використовується термін "fixedrate".

Новий клас кодів також називають класом фонтанних кодів (DigitalFountainCodes). Кодер такого коду за запитом завжди може додати "на льоту" (on-the-fly, on-line) невелике число кодових символів. При цьому час формування кожного кодового пакету постійний. Цю властивість кодера в зарубіжній літературі іноді називають терміном "localencodability". Незалежність генерування кодових символів забезпечується застосуванням статистичного кодування. Додати "на льоту" трохи перевірочних символів для класичних кодів не завжди вдається. Кодові символи виявляються залежними один від одного. З теорії кодування відомо, що лінійний блоковий (N, K) код може відновити сполучення з K символів при максимальному числі стирань E серед N кодових символів, рівному $M = N\text{-}K$. Це положення є наслідком межі Сінглтона. Код, що задовольняє цій межі, допускає максимальне число стирань. При парному значенні M таким оптимальним кодом є код Ріда-Соломона [1].

Код Торнадо. Головним мотивом створення коду стала необхідність в більш ефективних алгоритмах кодування і декодування у порівнянні з відповідними алгоритмами для кодів Ріда-Соломона. У той же час, код представляє конструкцію з ефективно реалізованою базовою ланкою, що використовують кодер/декодер Ріда-Соломона як підпрограму. Кодер може бути представлений графом типу графа Таннера для кодів LDPC (LowDensityParityCode). Граф має дві групи вузлів: одна група складається з K вихідних вузлів (символів), друга – з $M=N–K$ перевірочних вузлів. Число ребер, що входить в i-й вузол перевірочного вузла, називається ступенем вузла d_i, $i = 1, 2, ..., M$. На відміну від коду Ріда-Соломона, для якого ступінь кожного вузла дорівнює K, для коду Торнадо він є невеликим числом. У матричній термінології це означає, що код має низьку щільність породжувальної

матриці. Тому розглянутий граф логічно назвати породжувальним. Ступінь перевірочного вузла цього графа є не просто малим, але і значенням випадкової величини d з розподілом щільності ймовірності $\rho(d)$. Алгоритм декодування коду представляє собою обмінний алгоритм (message passing decoding) між вузлами породжувального графа коду і багато в чому нагадує алгоритм для LDPC.

Базовою математичної операцією при кодуванні і декодуванні є проста операція додавання за модулем 2 (XOR) над бітами пакетів. Платою за зниження обчислювальної складності є дещо більше у порівнянні з кодом Ріда-Соломона число необхідних для реконструкції повідомлення кодових символів $K'= K(1+\varepsilon)$, де ε – невелике позитивне число (наприклад, 0.05). Величину $\Delta=K'-K$ прийнято іменувати терміном «overhead». При декодуванні повідомлення з K' символів потрібно $N\ln(1/\varepsilon)$ операцій XOR.

Код Торнадо частково дозволив подолати обмеження коду Ріда-Соломона. Середня ступінь перевірочного вузла коду дорівнює 14. В цілому це хороший результат з точки зору обсягу обчислювальних витрат при кодуванні і декодуванні. Тим не менше, час декодування як і раніше визначається величиною N, а не K. Цей же висновок відноситься і до обсягу пам'яті для реалізації алгоритмів.

Випадковий фонтанний код. Код є несистематичним. У кожний з кодових символів з ймовірністю 0.5 входить кожен з K вихідних символів. Складанням біт за модулем 2 (операцією XOR) всіх вхідних вихідних символів обчислюються значення k біт кодового символу. У результаті кодер генерує випадковий код (random fountain), в середньому використовуючи $K/2$ операцій XOR для обчислення одного кодового символу. Ця величина, яка називається вартістю кодування, виявляється порівнянною з K, і тому її слід вважати великою. Породжувальна матриця коду має високу щільність одиниць [2] .

У середньому ступінь кожного перевірочного вузла графа виявляється рівною $K/2$.

Для декодування коду доцільне використання алгоритму максимальної правдоподібності (МП). Аналіз показує, що процес завершується при K' кодових символах з імовірністю $1-\delta$, де $\delta<2^{-\Delta}$, $\Delta=K'-K$ [3]. Наприклад, для K = 104 маємо $\delta<10^{-6}$ при $K'=K+20\cong K$. Це чудовий результат. Однак якою ціною він досягається на практиці? Аналіз показує, що алгоритм МП вимагає порядку K^2 операцій над символами. У результаті, для випадкового фонтанного коду процес завершується повільно.

LT-код. Алгоритм кодування аналогічний алгоритму випадкового фонтанного коду, проте має принципово інший розподіл $\rho(d)$. Суть LT-коду саме в поєднанні цього "хорошого" розподілу з простим і швидким алгоритмом декодування [4].

В основі знаходження $\rho(d)$ і алгоритму декодування лежить нескладне імовірнісне завдання. Є повідомлення (книга) з K вихідних символів (сторінок). З цієї множини з імовірністю $1/K$ проводиться K' випадкових вибірок символів (сторінок). У результаті формується множина з K' кодових символів (випадково висмикнутих сторінок). *Питання*: чому має дорівнювати K' для того, щоб з імовірністю $1-\delta$ кожен з усіх K вихідних символів виявився хоч один раз (All-At-Once) серед K' кодових символів? *Відповідь*: $K'\cong K\ln(K/\delta)$ при досить великому K. Маючи таке число кодових символів (випадково висмикнутих з книги сторінок), можна реконструювати вихідне повідомлення (книгу) з імовірністю $1-\delta$. Алгоритм реконструкції гранично швидкий і використовує лише інформацію про нумерацію символів (сторінок). Кодування і декодування в цьому завданні тривіальне. Кодовий символ (пакет) формується з d_i=1 вихідних символів (сторінок). Будемо вважати, що d_i – це значення випадкової величини d з тривіальним розподілом щільності ймовірності $\rho(1)=1$ (All-At-Once distribution).

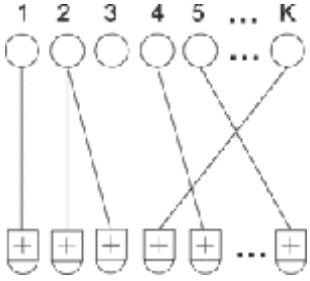

Рис.1. Породжувальний граф фонтанного коду

Породжувальний граф кодового процесу з таким розподілом наведено на рис. 1. Назвемо такий код фонтанним кодом "один в один" (з одиничним ваговим розподілом). Якщо

суму ступенів кодових символів назвати вагою, то отриманий результат можна сформулювати наступним чином. З кінцевої частини процесу ваги $W\cong K\ln(K/\delta)$ можна реконструювати вихідне повідомлення з імовірністю 1-δ. Або: процес завершується з імовірністю 1-δ за умови $W\cong K\ln(K/\delta)$. Розмір K цієї кінцевої частини збігається з її вагою. У результаті, для коду "один в один" процес завершується з високою ймовірністю швидко.

Алгоритм декодування навіть такого простого коду передбачає наявність апріорної інформації від кодера про параметри породжувального графа коду: ступеня кожного кодового символу d_i і списку номерів вихідних символів, підключених на графі до кодових символів. Ця інформація може бути передана декодеру різними способами. Наприклад, якщо відправник і одержувач синхронізовані і використовують ідентичні генератори випадкових чисел для вибору ступенів d_i і номерів вихідних символів, проблема вирішується автоматично. Альтернативним є метод передачі ключа кодового символа в його заголовку. У ключі міститься інформація як про його ступінь d_i, так і список номерів вихідних символів, що входять в кодовий символ [5].

Тепер узагальнимо викладене на довільний ймовірнісний розподіл при обмеженні $\rho(1)> 0$. Воно означає, що ймовірність кодового символу зі ступенем одиниця відмінна від нуля. Скористаємося знову зв'язком процесу з вищезгаданим ймовірнісним завданням і розглянемо кодування книги. Кодову сторінку, сформовану при $d_i>1$ за допомогою операції XOR, назвемо непрочитаною (шифрованою), сторінку із значенням $d_i=1$ – прочитаною (нешифрованою). Тоді можливий наступний алгоритм декодування. Якщо декодер виявив розшифровану кодову сторінку, він виключає як її вміст (тієї ж операцією XOR) з усіх кодових сторінок, які її містять, так і її номер з їх переліку номерів. У результаті знижуються на одиницю ступені цих кодових сторінок, і, тим самим, зростає ймовірність появи розшифрованої сторінки. Якщо декодер знову виявить нешифровану сторінку, він знову виключає ... і т.п. Значить, маючи лише одну нешифровану сторінку, принципово можлива реконструкція всієї книги дуже простим алгоритмом. Ця умова є необхідною умовою старту алгоритму, який відносять до класу ітеративних алгоритмів з поширенням довіри (ПД, BeliefPropagation) [6].

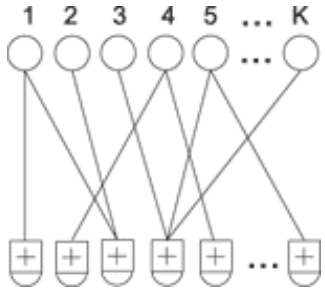

Рис.2. Породжувальний граф LT-коду

Отже, породжувальний граф LT-кода повинен мати вигляд, подібний рис. 2, де умова $\rho(1)>0$ є необхідною умовою старту алгоритму декодування. В іншому випадку декодування по викладеному алгоритму неможливо. Наприклад, якщо $\rho(K)=1$, то декодер на першому кроці завжди буде заявляти про відмову від декодування, оскільки ніколи не знайде кодового символу зі ступенем 1. В результаті повідомлення ніколи не буде декодоване [7].

Процес, як і для випадку "один в один", повинен завершитися з імовірністю 1-δ за умови $W\cong K\cdot\ln(K/\delta)$. Очевидно, що ця вага може бути"набрана" при меншій кількості прийнятих символів K', ніж для процесу "один в один". І платою за цю можливість є вартість кодування, яка у граничному випадку $K'\cong K$ виявляється рівною $\ln(K/\delta)$. У цьому й полягає філософія LT-кода. Залишилося знайти "хороший" розподіл.

Отже, алгоритм з ПД стартує лише за наявності хоча б одного кодового символу зі ступенем, рівним 1. На кожній ітерації потрібно, щоб у графі залишився хоч один символ, який має ступінь 1. Тому ідеальною для алгоритму була б ситуація, при якій після кожної ітерації декодування у графі залишався б лише один кодовий символ зі ступенем 1.

Параметри коду $K=10^4$, $c=0.2$, $\delta=2\cdot10^{-2}$ дають значення $S=244$, $K/S=41$. Розподіл $\tau(d)$ має максимуми в точках $d=1$ і $d=K/S$. Для LT-коду процес завершується досить швидко. Високу ймовірність декодування повідомлення вдається добитися при $K'\cong1.1K$.

Код Raptor. Код представляє собою кодову конструкцію з внутрішнім (по відношенню до каналу) LT-кодом. В якості зовнішнього коду можна використовувати "майже" блоковий код (с фіксованою швидкістю). Розробка та дослідження коду належить A. Shokrollahi [8].

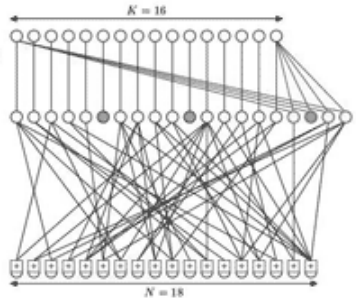

Рис.3. Породжувальний граф Raptor коду

На рис. 3 наведено приклад повідомлення з K=16 символів спочатку кодується систематичним блоковим кодом (K, K_1) = (16,20), що відновлює E=3 стирання (нагадаємо, що згідно границі Сінглтона максимальне число стирань для такого коду дорівнює 4). Потім K_1=20 символів кодуються LT-кодом. Для реконструкції всього повідомлення в цілому декодеру LT-коду достатньо відновити будь-які K_1-E=17 символів блокового коду. На рис. 3 темними кружками виділені не відновлені LT декодером символи. Решта 17 символів зовнішнього коду декодер LT коду відновлює з K'=18 кодових символів.

Аналіз конструкції показує, що вихідне повідомлення може бути реконструйованим з імовірністю 1-δ з $K'=K(1+\varepsilon)$ символів, де ε – невелике позитивне число. Вартість декодування виявляється порядку $\ln(1/\varepsilon)$ операцій XOR. Для декодування повідомлення потрібно близько $K\ln(1/\varepsilon)$ операцій XOR. На сьогоднішній день цей код є, можливо, кращою апроксимацією ідеального фонтанного коду.

Попередній матеріал дозволяє сформулювати вимоги до ідеального кодового “фонтану”.

1. Код повинен представляти потенційно необмежений потік символів x_i, i=1,2, ... Це означає, що і породжувальна матриця, і породжувальний граф коду є напівнескінченними, а кодові слова мають потенційно необмежений розмір. Кодер повинен генерувати “на льоту” нові кодові символи, значення яких не залежать від попередніх. Алгоритм кодування повинен бути швидким. Час кодування одного символу повинно бути малим.

2. Повідомлення з K символів має бути реконструйоване (декодоване) з будь-яких K кодовb[символів. Алгоритм реконструкції повинен бути швидким. Час реконструкції має лінійно залежати від величини K.

Всі ці коди мають прості алгоритми декодування і дозволяють на практиці отримувати результати, близькі до граничних можливостей завадостійкого кодування.

Важливе і зростання ефективності використання кодування зі збільшенням розміру повідомлення. Деяке обмеження ефективності кодування природно спостерігається при невеликих обсягах даних. Це обмеження є платою за використання статистичних методів. Для багатьох практичних застосувань, проте, воно не настільки істотно. У той же час, завдяки статистичному кодуванню можливе вирішення нетривіальних мережевих завдань, як, наприклад, одночасне завантаження файлу великого розміру з кількох сайтів. Відзначимо також, що алгоритми кодування і декодування принципово не залежать від розміру пакета.

Звернемо увагу і на універсальність потокових кодів. Вони можуть бути використані в будь-якому каналі із стиранням, незалежно від статистики стирань.

Література

1. Блейхут Р. Теория и практика кодов, контролирующих ошибки / Р. Блейхут. – М.: Мир, 1986. – 576 с.

2. Шульгин В. И. Основы теории передачи информации. Ч. I. Экономное кодирование : учебн. / В. И. Шульгин. – Харьков: Нац. аэрокосм. ун-т «Харьк. авиац. ин-т», 2002. – 100 с.

3. Потапов В. Н. Теория информации. Кодирование дискретных вероятностных источников : учебник / В. Н. Потапов. – Новосибирск: Новосибирский гос. ун-т, 1999.

4. M. Luby. LT Codes, In Proc. Of the 43-rd Annual IEEE Symposium on Foundations of Computer Science (FOCS). 2002. Pp. 271-282.

5. Берликэмп Э. Алгебраическая теория кодирования / Э. Берликэмп. –М.: Мир, 1971. –238 с.

6. Кодирование информации / [Н. Т. Березюк, А. Г. Андрущенко, С. С. Мощицкий и др.]. – Харьков: Вища школа, 1978. – 252 с.

7. A. Shokrollahi. Raptor Codes, Transactions on Information Theory (IEEE). – 2006. –№ 52(6). – PP. 2551-2567.

8. M. Luby, M. Mitzenmacher, A. Shokrollahi, D. Spielman. Efficient Erasure Correcting Codes", IEEE Transactions on Information Theory, Vol.47, Issue 2, pp. 569-584, February 2001.