УДК 004.032.26

О. А. Турдиев, С. В. Клименко
Петербургский государственный университет путей сообщения Императора Александра I

А. Б. Тухтаходжаев
Ташкентский институт инженеров железнодорожного транспорта

Оценки эффективности обнаружения ошибок контрольного суммирования (CRC) передаваемых данных

Методы обнаружения ошибок предназначены для выявления искажений в сообщениях при их передаче по зашумленным каналам. Основная идея, заложенная в них, это передача избыточной служебной информации, по которой можно судить с некоторой степенью вероятности о достоверности принятых данных. В статье рассматривается применение метода формирования контрольного суммирования (Cyclic redundancy code – CRC) для данных. Для рассматриваемого способа генерации CRC было написано приложение на языке высокого уровня Java, позволяющее получить результаты и собрать статистику обнаружения/необнаружения ошибки передаваемых данных. Собранная статистика показывает вероятность необнаружения ошибки для передаваемого блока данных. Для оценки вероятности необнаружения искаженных битов был сымитирован процесс передачи данных с ошибочными битами. Для написания приложения использовано программное обеспечение Java Development Kit версии 1.8 и среда разработки Net Beans IDE 8.0.2.

CRC (Cyclic redundancy code), генераторный многочлен, циклический избыточный код, пакетные ошибки, ошибочные биты, целостность

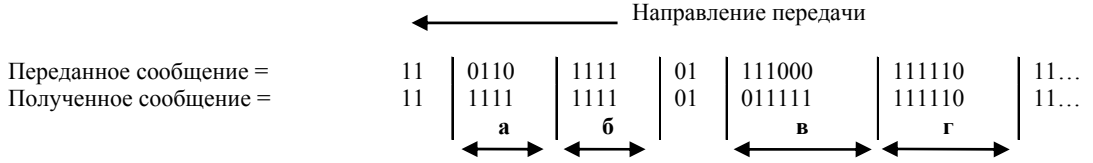
Алгоритм контрольного суммирования CRC расшифровывается как циклический избыточный код (Cyclic redundancy code) и предназначается для контроля целостности данных. Он широко используется в проводных и беспроводных сетях и в устройствах хранения данных для проверки информации на подлинность и защиты от несанкционированного изменения.

Краткое описание алгоритма подсчета CRC. В передаваемом кадре под воздействием помех могут появляться ошибки. Проявляемые ошибки по основным свойствам можно разделить на три класса: одиночные, кратные, пакетные.

Пакетная ошибка определяется как количество битов между двумя последовательными ошибочными битами. Кроме того, при определении длины пакетной ошибки последний ошибочный бит в пакете и первый ошибочный бит в следующем пакете должны быть разделены [1]. Пример пакетной ошибки показан на рис. 1.

Метод контрольного суммирования CRC основывается на свойствах деления с остатком многочлена (двоичное число). По сути, результатом CRC является остаток от деления многочлена, соответствующий исходным данным, на порождающий многочлен фиксированной длины.

Стандартный способ представления генераторного многочлена состоит в том, чтобы показать те позиции, на которых двоичные единицы являются степенями *X*. Примеры порождающих



(**а**) 2-битная пакетная ошибка;
(**б**) 4-битный безошибочный минимум;
(**в**) 4-битная пакетная ошибка;
(**г**) 6-битный безошибочный минимум.

Рис. 1

многочленов, используемых на практике, выглядят следующим образом [2]:

$$\text{CRC16} = x^{16} + x^{15} + x^2 + 1,$$
$$\text{CRC} - \text{CCITT} = x^{16} + x^{15} + x^5 + 1,$$
$$\text{CRC32} = x^{32} + x^{26} + x^{23} + x^{16} + x^{12} + x^{11} + x^{10} + x^8 + x^7 + x^5 + x^4 + x^2 + x + 1.$$

Следовательно, CRC-16 в двоичной форме эквивалентен записи:

11000000000000101.

При таком генераторном многочлене до генерации FCS будет добавлено 16 нулей. Последний будет 16-битным остатком.

CRC-16 и CRC-CCITT широко используются в таких сетях, как ISDN, тогда как CRC-32 применяется в большинстве локальных сетей. Метод CRC можно легко реализовать в аппаратном и программном обеспечении.

Один набор контрольных цифр генерируется (вычисляется) для каждого переданного кадра на основе содержимого фрейма и добавляется передатчиком к хвосту кадра. Затем приемник выполняет аналогичное вычисление по полному фрейму плюс контрольные цифры. Если ошибки не были найдены, всегда должен быть получен известный результат; если получен другой ответ, это указывает на ошибку.

Количество контрольных цифр на кадр выбирается в соответствии с ожидаемым типом ошибок передачи; наиболее часто встречаются 16 и 32 бит. Вычисленные контрольные цифры обозначаются как последовательность проверки кадров FCS (Frame Check Sequence) или циклической избыточности CRC.

По существу, метод использует свойство двоичных чисел. При использовании арифметики по модулю 2 [3]: $M(x)$ – k – разрядное число (сообщение, которое должно быть передано), $G(x)$ – $(n + 1)$ – битное число (делитель или генератор), $R(x)$ – n – разрядное число, такое, что $k \succ n$ (остаток):

$$\frac{M(x)2^n}{G(x)} = Q(x) + \frac{R(x)}{G(x)}, \quad (1)$$

$$\frac{M(x)2^n + R(x)}{G(x)} = Q(x), \quad (2)$$

предполагая арифметику по модулю 2.

Этот результат можно легко подтвердить, подставив выражение для $\frac{M(x)2^n}{G(x)}$ из (1) в (2):

$$\frac{M(x)2^n + R(x)}{G(x)} = Q(x) + \frac{R(x)}{G(x)} + \frac{R(x)}{G(x)},$$

равное $Q(x)$, так как все добавленные к нему числа по модулю 2 будут равны нулю, т. е. остаток будет равен нулю.

Чтобы использовать полное содержимое кадра $M(x)$ вместе с добавленным набором нулей, равным по количеству FCS, они должны быть сгенерированы (т. е. умножены на 2^n, где n – количеств FCS), разделены по модулю 2 на двоичное число ($G(x)$ – генераторный многочлен, содержащий на одну единицу больше, чем FCS). Операция деления эквивалентна выполнению операции «исключающее ИЛИ» по параллельному биту, так как обрабатывается каждый бит в кадре. Тогда остаток $R(x)$ является FCS, который передается в хвосте информационных кадров.

Аналогично при получении принятый поток битов, включающий в себя число CRC, снова делится на один и тот же генераторный многочлен, т. е. $\frac{M(x)2^n}{G(x)}$, и если ошибок нет, остаток – все нули. Однако если присутствует ошибка, остаток не равен нулю.

Выбор генераторного многочлена важен, поскольку он определяет типы обнаруженных ошибок. Предположим, что переданный кадр

$$M(x) = 110101100110,$$

а шаблон ошибки

$$E(x) = 000000001001.$$

Таким образом, 1 в битовой позиции указывает на ошибку. Применяя булеву функцию «сумма по модулю 2»,

$$\text{полученный кадр} = M(x) + E(x),$$
$$\frac{M(x) + E(x)}{G(x)} = \frac{M(x)}{G(x)} + \frac{E(x)}{G(x)}.$$

Поскольку $M(x)/G(x)$ не дает остатка, ошибка присутствует, если $E(x)/G(x)$ дает остаток.

Например, $G(x)$ имеют по крайней мере три ненулевых слагаемых (1 бит), и $E(x)/G(x)$ будет давать остаток для всех однобитовых и всех двухбитовых ошибок с арифметикой по модулю 2, и следовательно, ошибки обнаруживаются. И наоборот, ошибка длиной $G(x)$ не дает остатка и остается незамеченной [4].

Генераторный многочлен из R бит обнаруживает [5]:

– все однобитовые ошибки,

– все двухбитные ошибки,

– все нечетные числа бит-ошибок,

– все пакеты ошибок $\prec R$,

– большинство пакетов ошибок $\succ R$.

Оценка эффективности обнаружения ошибок метода CRC4. Под эффективностью понимается возможность выявления и исправления ошибок. Для рассматриваемого способа генерации CRC было написано приложение на языке высокого уровня Java, позволяющее получить результаты и статистику обнаружения ошибок в передаваемых данных. Статистика показывает обнаружение и необнаружение ошибки. Для написания приложения использованы программное обеспечение Java Development Kit версии 1.8 и среда разработки Net Beans IDE 8.0.2. Эксперимент проводился с помощью следующих аппаратных и программных ресурсов:

– операционная система Windows 10 (64-разрядная);

– двухъядерный процессор Intel Core i3-3217U с тактовой частотой 1.8 ГГц;

– 16 Гбайт оперативной памяти;

– жесткий диск объемом 1 Тбайт.

Программа предназначена для определения целостности передаваемых данных с использованием метода контрольной суммы блока, бита четности, BCC (Block Check Character) – блочный контроль четности.

Программа имитирует процесс передачи данных между источником (рис. 2 – пример отправления случайных битов пакета и CRC4) и приемником (рис. 3 – пример обнаружения ошибок CRC4), а также обеспечивает выполнение следующих функций:

1. Задание исходных данных вручную (кнопка «From Text» на рис. 2) или при помощи генератора псевдослучайных символов (кнопка «Random» на рис. 2) или метода класса Random() (создает генератор чисел, использующий уникальное начальное число).

2. Конвертация исходных данных в шестнадцатеричную, восьмеричную и двоичную системы счисления (поля «Hex», «Oct», «Bin» на рис. 2).

3. По полученным исходным данным проводится подсчет контрольных сумм CRC циклических избыточных кодов (поле «CRC» на рис. 2).

4. Дополнительно на усмотрение пользователя можно вносить одиночные или многократные ошибки (кнопка «Random Error» на рис. 2) в исходные данные (в данном случае были использованы двойные ошибки).

После имитации процесса передачи данных заново производится подсчет контрольных сумм CRC циклических избыточных кодов, которые впоследствии сравниваются с ранее рассчитан-

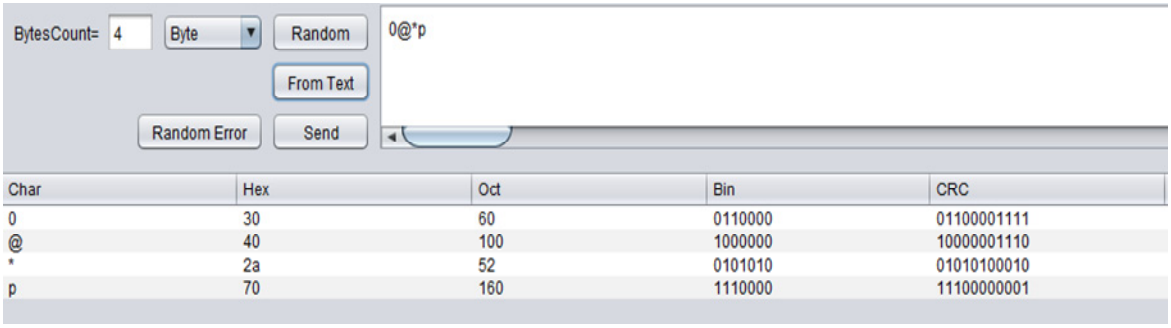


Рис. 2

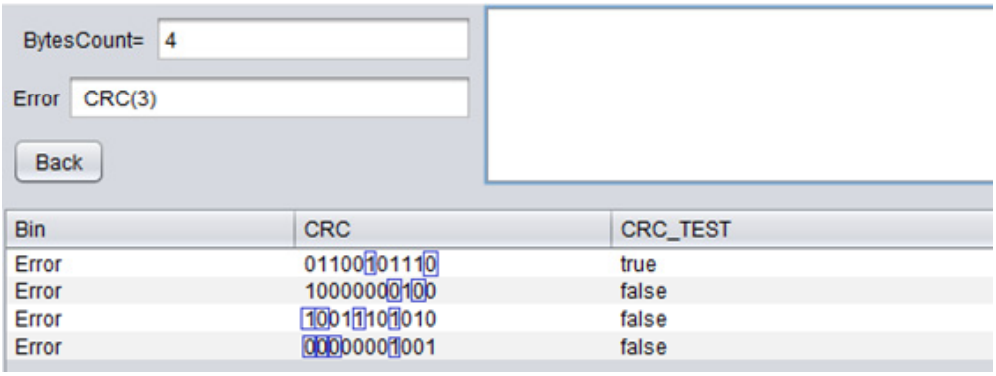


Рис. 3

Параметр	Передаваемый символ, char	Шестнадцатеричное представление символа, hex	Восьмеричное представление символа, oct	Бинарное представление символа, bin
Отправлено	**0**	30	60	0110**0**0 111**1**
Получено	**2**	32	62	0110**1**0 111**0**

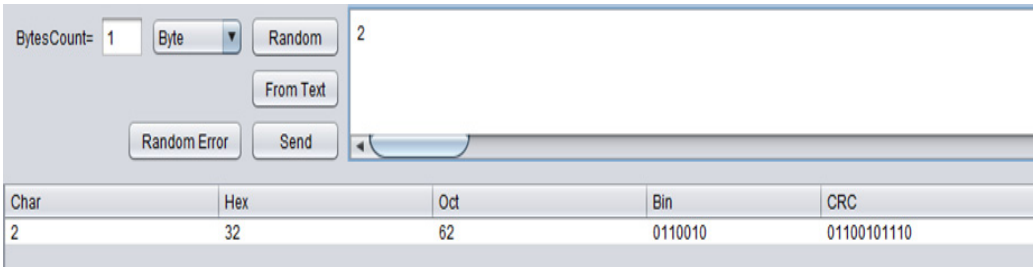


Рис. 4

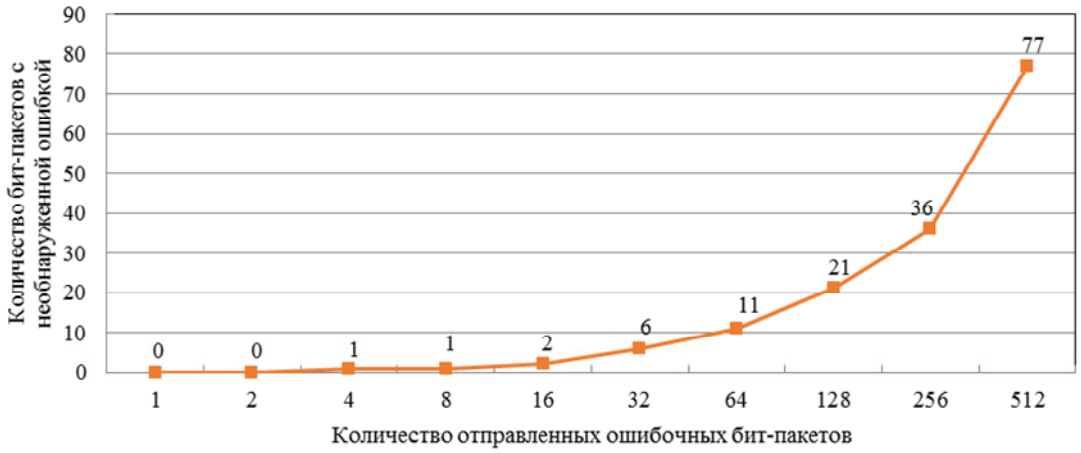


Рис. 5

ными для выявления факта искажения передаваемых данных (рис. 3); поле «CRC_TEST» указывает на обнаруженную ошибку посредством флагов «true» («истина», «1», данные верны) и «false» («ложь», «0», данные содержат ошибку).

Анализ результатов обнаружения ошибок метода CRC4. CRC-коды обладают высокой достоверностью обнаружения искажений, доля P_0 обнаруживаемых искажений не зависит от длины защищаемого массива данных, а определяется только разрядностью R контрольного кода [6]:

$$P_0 = 1 - 2^{-R}.$$

Таким образом, для CRC4 $P_0 = 1 - 2^{-4} = 0.9375$. Исходя из того, что разрядность контрольной суммы CRC4 составляет 4 бит, очень высока вероятность возникновения коллизий, поскольку максимально допустимое число комбинаций контрольной суммы $\text{CRC4} = 2^4 = 16$.

Для оценки вероятности необнаружения искаженных бит и выявления факта коллизий был сымитирован процесс передачи данных с ошибочными битами. Результаты имитации приведены на рис. 4 (пример необнаружения ошибок CRC4).

В таблице показано, что при передаче символа 0 искажение может возникнуть как в передаваемом символе, так и в самой контрольной сумме таким образом, что при получении искаженного символа и искаженной контрольной суммы полученный символ будет восприниматься как корректный.

Поскольку в эксперименте участвует 12 бит (8 бит под исходный символ и 4 бит контрольной суммы CRC4), то вероятность искажения отдельного бита определяется по формуле:

$$p = \frac{1}{R} = 0.08,$$

где R – число бит, представляющих передаваемые символы.

Вероятность безошибочной передачи всех R бит показана на рис. 4 [7].

В статье рассмотрен способ формирования контрольного суммирования циклического избыточного кода CRC4. Основная проблема со всеми проверками CRC4 состоит в возможном появлении коллизий, если в исходные данные были внесены двукратные или многократные ошибки.

Этот факт подтверждается информацией, собранной на рис. 5 (результат определения ошибок алгоритмом CRC4 статистикой). В силу небольшой избыточности алгоритм рекомендуется использовать для значительной экономии памяти и при малых исходных данных, но надо понимать, что вероятность появления коллизии достаточно высока.

Методы обнаружения ошибок в каналах связи предназначены для выявления искажений в сообщениях при передаче в условиях шумов. Авторами статьи исследован метод передачи избыточной служебной информации, по которой можно судить с некоторой степенью вероятности о достоверности принятых данных. Полученные результаты имеют не только научную, но и методическую ценность. Практическую ценность представляет написанное авторами статьи приложения на языке высокого уровня, позволяющее собрать статистику обнаружения ошибки передаваемых данных.

СПИСОК ЛИТЕРАТУРЫ

1. Halsall F. Fifth edition, computer networks and the Internet. Addison-Wesley: Pearson Education, 2005. 803 p.

2. Lin S., Costello D. J. Jr. Error Control Coding: Fundamentals and Applications. New Jersey, Englewood Cliffs: Prentice-Hall, Inc., 1983.

3. Олифер В. Г., Олифер Н. А. Компьютерные сети. Принципы, технологии, протоколы. СПб.: Питер, 2008. 958 с.

4. Halsall F. Data communications, computer networks and open systems. Addison-Wesley: Pearson Education, 1996. 907 p.

5. Ромащенко А., Румянцев А., Шень А. Заметки по теории кодирования. 2-е изд., испр. и доп. М.: МЦНМО, 2017. 88 с.

6. Яковлев В. В., Кушназаров Ф. И. Оценка влияния помех на производительность протоколов канального уровня // Изв. Петерб. гос. ун-та путей сообщения. 2015. Вып. 1 (42). С. 133–138.

7. Исследование формирования блоковой контрольный суммы (ВСС) передаваемых данных / О. А. Турдиев, В. В. Яковлев, С. В. Клименко, А. Х. Болтаев // Изв. СПбГЭТУ «ЛЭТИ». 2019. № 6. С. 72–78.

O. A. Turdiev, S. V. Klimenko
Emperor Alexander I Petersburg State Transport University

A. B. Tuxtaxodjaev
Tashkent Institute of Railway Engineers

THE STUDY OF THE IMPLEMENTATIONS OF THE ALGORITHM METHOD CHECKS SUMMATION (CRC) OF THE TRANSMITTED DATA

Error detection methods are designed to detect distortions in messages when they are transmitted through noisy channels. The main idea embodied in them is the transfer of redundant service information, which can be judged with some degree of probability about the reliability of the received data. The article discusses the use of the control summation method (Cyclic redundancy code, CRC) for data. For the CRC generation method in question, an application was written in a high-level Java language, which allows obtaining results and collecting statistics on the detection / non-detection of data transfer errors. The collected statistics show the probability of not detecting an error for the transmitted data block. To assess the likelihood of undetected distorted bits, the process of transferring data with erroneous bits was simulated. JavaDevelopmentKit software version 1.8 and the NetBeans IDE 8.0.2 development environment ar.

CRC (Cyclic redundancy code), generator polynomial, cyclic redundancy code, packet errors, erroneous bits, integrity