DOI: 10.18137/RNU.V9187.21.04.P.119 УДК 004.056.2

О.А. Турдиев, А.Д. Хомоненко, М.В. Гофман

СРАВНЕНИЕ МОДЕЛЕЙ ВЕРОЯТНОГО КОДА ЧИСЛА PNC И ЦИКЛИЧЕСКОГО ИЗБЫТОЧНОГО КОДА CRC

Постановка задачи: необходимость обеспечения целостности данных, передаваемых в сетях связи, актуализация вопроса формирования контрольных сумм, используемых при анализе целостности. При этом целесообразно снижение вычислительной сложности алгоритмов формирования контрольных сумм. Целью работы является сравнение вычислительной сложности и показателей обнаружения ошибок моделей формирования вероятного кода числа PNC (Probable Number Code) и циклического избыточного кода CRC (Cyclic Redundancy Code). Новизна работы состоит в том, что для сравнения вычислительной сложности и показателей обнаружения ошибок выполнен анализ моделей PNC и CRC. Результат: на основе проведенного анализа моделей подтверждается снижение вычислительной сложности алгоритма формирования контрольных сумм в модели PNC по сравнению с CRC. Практическая значимость: модель вероятного кода числа может применяться в протоколах сетей передачи данных, а также для обоснования перспективности дальнейших исследований в этом направлении.

Ключевые слова: вычислительная сложность, порождающий полином, вероятный код числа, циклический избыточный код, PNC, CRC, сравнение моделей кодирования, пакетные ошибки, ошибочные биты, целостность данных.

O.A. Turdiev, A.D. Khomonenko, M.V. Gofman

COMPARISON OF PROBABLE CODE NUMBER PNC AND CYCLIC REDUNDANCY CODE CRC

Statement of the problem: The need to ensure the integrity of data transmitted in communication networks, actualizes the issue of forming checksums used in the analysis of integrity. In this case, it is advisable to reduce the computational complexity of the algorithms for generating checksums. The aim of the work is to compare the computational complexity and error detection rates of the models for the formation of the probable number code PNC (Probable Number Code) and the cyclic redundancy code CRC (Cyclic Redundancy Code). The novelty of the work lies in the fact that PNC and CRC models were analyzed to compare the computational complexity and error detection rates. Result: on the basis of the analysis of the models, the reduction in the computational complexity of the algorithm for generating checksums in the PNC model in comparison with the CRC is confirmed. Practical significance: the model of the probable code of a number can be used in protocols of data transmission networks, as well as to substantiate the prospects for further research in this direction.

Keywords: computational complexity, generating polynomial, probable number code, cyclic redundancy code, PNC, CRC, coding model comparison, burst errors, bit errors, data integrity.

Введение

В настоящее время существует тенденция развития энергосберегающих технологий. Известно, что большой объем вычислений приводит к существенным затратам энергии. В результате актуальными являются исследования по снижению вычислительной сложности моделирования обработки данных. В сетях передачи данных важным показателем является достоверность и корректность входящих и исходящих потоков данных.

Турдиев Одилжан Акрамович
аспирант Петербургского государственного университета путей сообщения Императора Александра I, Санкт-Петербург. Сфера научных интересов: вычислительная техника и сети в отрасли, моделирование информационных систем, программирование. Автор 10 опубликованных научных работ.
Электронный адрес:odiljan.turdiev@mail.ru

Хомоненко Анатолий Дмитриевич
доктор технических наук, профессор, профессор кафедры информационных и вычислительных систем. Петербургский государственный университет путей сообщения Императора Александра I, Санкт-Петербург. Сфера научных интересов: базы данных, интеллектуальные системы и технологии, информационные технологии в сфере безопасности, моделирование информационных систем, программирование. Автор более 250 опубликованных научных работ.
Электронный адрес: ivs@pgups.ru

Гофман Максим Викторович
кандидат технических наук, доцент, доцент кафедры информатики и информационной безопасности. Петербургский государственный университет путей сообщения Императора Александра I, Санкт-Петербург. Сфера научных интересов: безопасность сетей ЭВМ, вычислительная техника и сети в отрасли, информатика. Автор более 20 опубликованных научных работ.
Электронный адрес: inib@pgups.ru

Существует много моделей и методик кодирования кодов, например, CRC, для контроля и формирования передаваемых данных с помощью добавления контрольных сумм к пакетам данных. Эти модели позволяют обнаружить ошибки, обработать полученные и отправленные данные в целях обеспечения достоверности.

Несмотря на то, что модель CRC широко используется, для нее по-прежнему сложно определить подходящие многочлены. Определение их характеристик становится очень сложным и критически важным для передачи информации в сетях, в которых необходимо обмениваться множеством данных с минимальными остаточными вероятностными ошибками. Эта задача решается с помощью детерминированных и стохастических методов [1].

Для обнаружения и исправления ошибок было разработано несколько методов коммуникации, например, Blahut (2003) [2]; Mac Williams and Sloane (1991) [3]; Суини (1991) [4]. Целью коммуникации, критически важной для безопасности, является обнаружение ошибок и инициирование перехода всего процесса в безопасное состояние, например, МЭК 61508 (2007) [5]. Это состояние не всегда является нормальным, но может быть причиной ограничения функциональности (например, состояние низкой скорости или нулевое напряжение).

Кроме того, в реальных условиях передачи на канал связи могут воздействовать различного рода помехи, проявляющиеся в исследуемом процессе в виде ошибочных бит, которые приводят к нарушению целостности данных [6]. В работе В.В. Яковлева [7] пред-

ложен выбор порождающего полинома для увеличения вероятности распознавание ошибок при формировании контрольных сумм в передаваемых данных.

Создание высококачественных, быстродействующих и достаточно простых алгоритмов формирования контрольных сумм с помощью параллельного генератора случайных чисел является одной из основных задач организации передачи данных с помощью низкочастотных энергосберегающих систем. От решения этой задачи в конечном счете зависит успех построения модели вероятного кода числа PNC (Probable Number Code), так как характеристики параллельного генератора случайных чисел (далее – ПГСЧ) во многом определяют параметры PNC.

Характеристика модели формирования контрольных сумм PNC

Передача данных – важная функция системы сети каналов передачи данных. Система передачи данных отправляет данные на обработку с помощью таких устройств, как программируемые логические контроллеры (далее – ПЛК), а ПЛК отправляют данные получателям. Важным является вопрос обеспечения целостности передаваемых данных, что делает его одним из самых важных функций низкой вычислительной сложности без уменьшения уровня обнаружения ошибки [7–9].

Актуальность вопроса синтеза вероятного кода числа PNC тесно связана с актуальностью задачи реализации принципов вероятностных методов моделирования и вычислений контрольных сумм данных, передаваемых по каналам связи.

На Рисунке 1 представлена модель структуры формирования PNC.

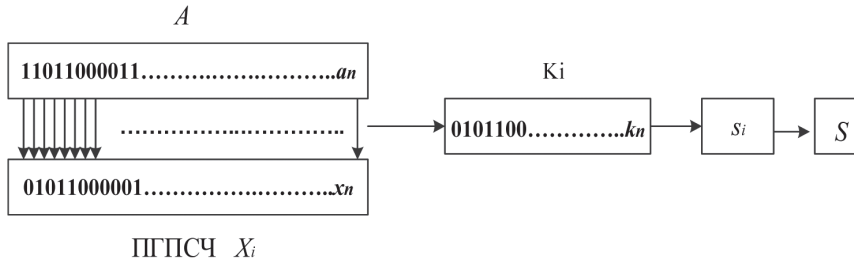

Рисунок 1. Пример структуры формирования PNC

Вычисление контрольной суммы для двоичной последовательности

$$A=(a_1,a_2,a_3,\ldots,a_n),\ \ a_l\in\{0;1\},$$

осуществляется при помощи нескольких параллельных генераторов псевдослучайных чисел (далее – ПГПСЧ). С помощью *i*-го $(i\in\{1,\ldots,M\})$ ПГПСЧ генерируется случайная последовательность

$$X_i=(x_{i,1},x_{i,2},x_{i,3},\ldots,x_{i,n}),\quad x_{i,j}\in\{0;1\},\ \ j\in\{1,\ldots,n\},$$

при этом $x_{1,j}=1$. Элементы последовательностей X_i с элементами последовательности *A* подвергаются логической операция «И» по mod 2 (обозначается символом &); в результате получаются следующие двоичные последовательности:

$$K_i=A\,\&\,X_i=(k_{i,1},k_{i,2},k_{i,3},\ldots,k_{i,n}),$$

где

$$k_{i,j}=(\mathrm{a}_j\,\&\,x_{i,j})\bmod 2.$$

Элемент контрольной суммы получается с помощью K_i :

$$s_i = \left(\sum_{j=1}^{n} k_{i,j} \right) \bmod 2 .$$

Таким образом, контрольная сумма представляет собой двоичную последовательность

$$S = (s_1, s_1, s_1, ..., s_M) ,$$

где M – это число используемых ПГПСЧ.

Все эти действия можно представить в матричной форме следующим образом. Элементы $x_{i,j}$ формируют матритцу

$$x = \begin{pmatrix} x_{1,1} & x_{1,2} & \cdots & x_{1,n} \\ x_{2,1} & x_{2,2} & \cdots & x_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ x_{m,1} & x_{m,2} & \cdots & x_{m,n} \end{pmatrix}.$$

Здесь X – это матрица ПГПСЧ; $x_{i,j}$ – случайное равномерно распределенное целое число, принимающее значения из множества $\{0; 1\}$, однако первая строка этой матрицы состоит только из единиц.

Элементы a_j формируют вектор

$$A = \begin{pmatrix} a_1 \\ a_2 \\ \vdots \\ a_n \end{pmatrix}.$$

На линии передатчика вычисляется следующая сумма: матрица x_{mn} умножаются на a_n – вектор (исходной код), в результате получается сумма по каждой строке, s_m и S – контрольная сумма (PNC),

$$S = \begin{pmatrix} s_1 \\ s_2 \\ \vdots \\ s_m \end{pmatrix} = \begin{pmatrix} a_1 \\ a_2 \\ \vdots \\ a_n \end{pmatrix} \begin{pmatrix} x_{1,1} & x_{1,2} & \cdots & x_{1,n} \\ x_{2,1} & x_{2,2} & \cdots & x_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ x_{m,1} & x_{m,2} & \cdots & x_{m,n} \end{pmatrix}.$$

Если передаваемые A и S получили помехи и сбои, то имеем в приемнике $\tilde{A}$ и $\tilde{S}$,

$$S = \tilde{A} \cdot X .$$

Чтобы обнаружить помехи и сбои, нужно сравнивать S – проверочный код с $\tilde{S}$: если $S \neq \tilde{S}$, не равно есть ошибка, если равно – нет ошибки.

Пример с A = 11010, PNC = 5 показан на Рисунке 2.

1101010000 = A+S

Также это действие можно представить следующей формулой:

$$S[n \cdot 1] = A[m \cdot n] \cdot X[n \cdot 1] .$$

Пример с вероятными помехами показан на Рисунке 3.

01010 10100

01010 00111

$$\begin{matrix} S & & A & & X \end{matrix}$$

$$\begin{pmatrix}1\\0\\0\\0\\0\end{pmatrix} = \begin{pmatrix}1\\1\\0\\1\\0\end{pmatrix} \begin{pmatrix}1&1&1&1&1\\0&1&0&1&1\\1&0&1&1&0\\1&1&1&0&1\\1&0&0&1&1\end{pmatrix}$$

Рисунок 2. Пример вычисления остатка PNC

$$\begin{matrix} \widetilde{S} & & \widetilde{A} & & X \end{matrix}$$

$$\begin{pmatrix}0\\0\\1\\1\\1\end{pmatrix} = \begin{pmatrix}0\\1\\0\\1\\0\end{pmatrix} \begin{pmatrix}1&1&1&1&1\\0&1&0&1&1\\1&0&1&1&0\\1&1&1&0&1\\1&0&0&1&1\end{pmatrix}$$

Рисунок 3. Пример вычисления с ошибками (помехами) PNC

Основные понятия модели формирования контрольных сумм CRC

Метод контрольного суммирования CRC основывается на свойствах деления с остатком многочлена (двоичное число). По сути результатом CRC является остаток деления многочлена, соответствующего 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 многочленов, используемых на практике, выглядят следующим образом [10]:

$$CRC16 = x^{16} + x^{15} + x^2 + 1$$

$$CRC - CCITT = x^{16} + x^{15} + x^5 + 1$$

$$CRC32 = x^{32} + x^{26} + x^{23} + x^{16} + x^{12} + x^{11} + x^{10} + x^8 + x^7 + x^5 + x^4 + x^2 + x + 1\ .$$

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 используется 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 бита. 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 [11]

$M(x)$ $-k$ – 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).

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

где $Q(x)$ является частным;

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

предполагая арифметику по модулю 2.

Этот результат можно легко подтвердить, подставив выражение для $M(x)*2^n / G(x)$ во второе уравнение

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

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

Чтобы использовать полное содержимое кадра $M(x)$ вместе с добавленным набором нулей, равным по количеству FCS, которые должны быть сгенерированы (то есть умноженные 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, то есть $M(x)*2^n + R(x)/G(x)$, 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. С применением булевой функции «сумма по модулю 2» 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, то ошибка присутствует, если $E(x)/G(x)$ дает остаток.

Например, $G(x)$ имеет по крайней мере три не нулевых слагаемых (1 бит), и $E(x)/G(x)$ будет давать остаток для всех однобитовых и всех двубитовых ошибок с арифметикой по модулю 2, следовательно, ошибки обнаруживаются. И наоборот, ошибка длиной $G(x)$ не дает остатка и остается незамеченной [12].

Генераторный многочлен из R бит обнаруживает [13]:

– все однобитовые ошибки;
– все двухбитные ошибки;
– все нечетные числа бит-ошибок;
– все пакеты ошибок $\prec R$;
– большинство пакетов ошибок $\succ R$.

Пример вычисления остатка для построения CRC кодового слова и оценки вычислительной сложности

Вычисление CRC детально показано на Рисунке 4,
где информационный кадр – 1101011011
генераторный полином – 10011
кадр с дополнительными нулями – 11010110110000

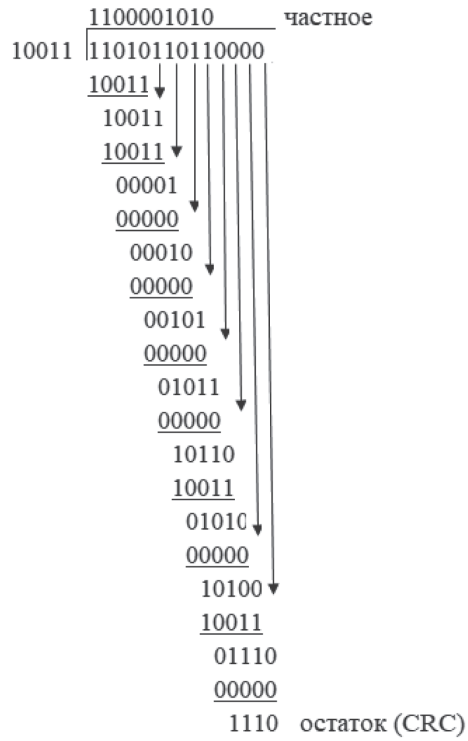

Рисунок 4. Вычисление CRC (CRC кодовое слово (передаваемый кадр) – 11010110111110 [13–15])

Таким образам, пример показывает, что при генераторном полиноме CRC-4-TU (10011) и битовой длине информационного кадра, равной 10 бит, требуется 10 делений

и 50 сложений по модулю 2. В общем случае неочевидна зависимость между вычислительной сложностью (число сложений и делений), размером порождающего полинома и размером информационного полинома [16–18].

Сравнение вычислительной сложности моделей PNC и CRC

Расчет числа вычислений, являющегося одной из важных характеристик алгоритма модели PNC в модели PNC, выполняется по следующей формуле:

$$T_{PNC} = S = X, \tag{1}$$

где T_{PNC} – число тактов операций для вычисления остатка PNC; S – разрядность полинома PNC; X – разрядность полинома, определяющего ПГПСЧ.

Таким образом, число тактов операций зависит от числа использования ПГПСЧ и от разрядности PNC.

Расчет числа вычислений операции & для PNC выполняется по формуле

$$R_{PNC} = (P \cdot X), \tag{2}$$

где R_{PNC} – число вычислений по операции &; P – размер информационного кадра.

Рассмотрим пример оценки вычислительной сложности с полиномом PNC разрядности 8 и информационным кадром размера 48 по формуле (2): $R_{PNC} = (48 \cdot 8) = 384$ двоичных операций.

Результаты оценки вычислительной сложности алгоритма формирования PNC, выраженные в числе операций, показаны на гистограмме (Рисунок 5).

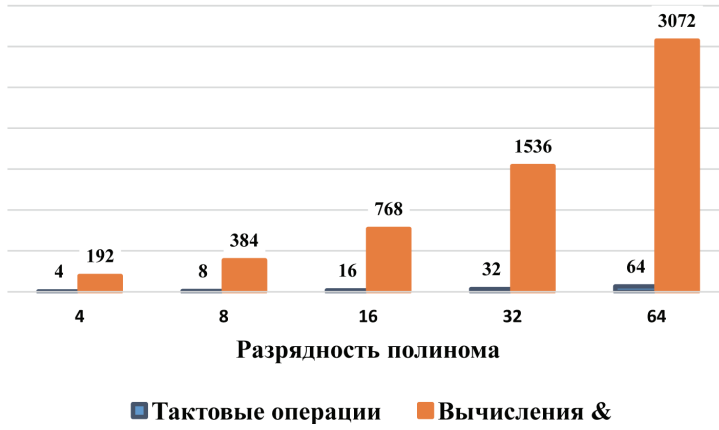

Рисунок 5. Числа операций при формировании PNC для 48-битовых данных

Количество двоичных операций для модели CRC можно определит по формуле

$$R_{CRC} = N \cdot (P + N) \bmod 2, \tag{3}$$

где R_{CRC} – число вычисления по модулю 2; P – размер информационного кадра; N – степень полинома CRC.

Рассмотрим пример оценки вычислительной сложности с полиномом CRC степени 8 и информационным кадром размера 48 по формуле (3): $R_{CRC} = (48 + 8) \cdot 8 = 448$ двоичных операций.

Расчет T_{CRC} – числа тактов операций для вычисления остатка CRC определяется следующим образом:

$$T_{CRC} = \frac{R_{CRC}}{N}. \quad (4)$$

Результаты оценки вычислительной сложности алгоритма формирования CRC, выраженной в числе операций, показаны на гистограмме (Рисунок 6).

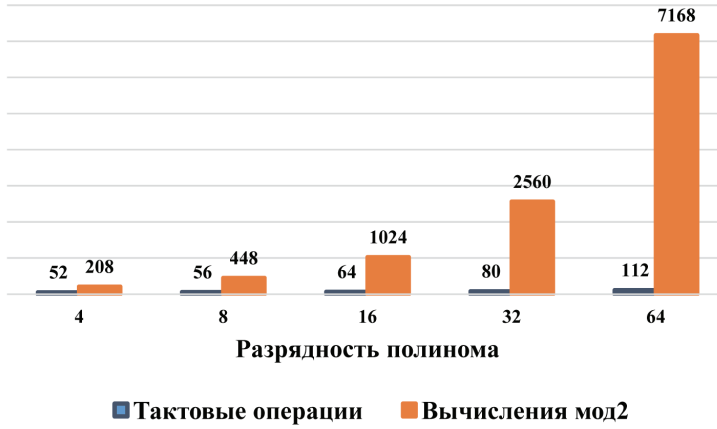

Рисунок 6. Числа операций при формировании CRC для 48-битовых данных

Оценки вычислительной сложности алгоритмов формирования PNC и CRC имеют разные способы вычисления операций логического & и сложения по mod 2, но каждые операции выполняются в соответствующих тактах. При этом число тактов PNC зависит от размера порождающего полинома, в CRC число тактов зависитот размера информационного кадра.

Результаты сравнительного анализа количества тактов (см. формулы (1), (4)), требуемых для вычисления контрольных сумм в моделях PNC и CRC, приведены на Рисунке 7.

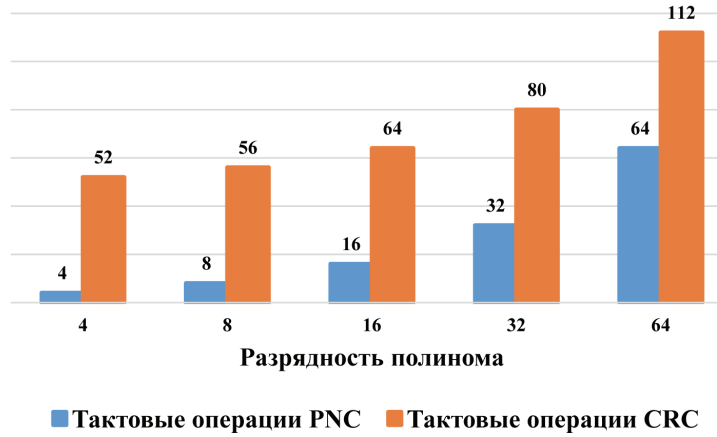

Рисунок 7. Число тактовых операцийв зависимости от разрядности полинома

Таким образом, результаты анализа показали, что по числу тактовых операций модель формирования PNC более эффективна, чем модель CRC.

Сравнение модели PNC и CRC по обнаружению ошибки

Коды PNC и CRC обладают высокой достоверностью обнаружения искажений. Вероятность P_0 обнаруживаемых искажений не зависит от длины защищаемого информационного данных, а определяется только степенью N порождающего полинома [19]:

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

Таким образом, для PNC4 и CRC4 $P_0 = 1 - 2^{-4} = 0.9375$. Исходя из того, что разрядность контрольной суммы PNC4 и CRC4 составляет 4 бита, очень высока вероятность возникновения коллизий, поскольку максимально допустимое число комбинаций контрольной суммы $PNC4\ и\ CRC4 = 2^4 = 16$ [20].

Для оценки вероятности необнаружения искаженных битов и выявления факта коллизий была определена формула для оценки требуемого количества экспериментов по передачи пакетов через канал при фиксированном количестве ошибок в канале. Расчет для сравнения PNC и CRC при n-битовом пакете и при k ошибках реализуется как число сочетаний из n по k:

$$C_n^k = \frac{n!}{k!(n-k)!}. \quad (5)$$

Таким образом, с помощью имитационного моделирования передачи пакетов через канал при фиксированном количестве ошибок в канале получены оценки вероятности необнаружения искажений $P_{не\ об}$ в пакетах:

$$P_{необ} = \frac{Y}{C_n^k}, \quad (6)$$

где Y – количество пакетов с не обнаруженными искажений.

Результаты расчетов, полученные с помощью формулы (6), приведены на Рисунке 8.

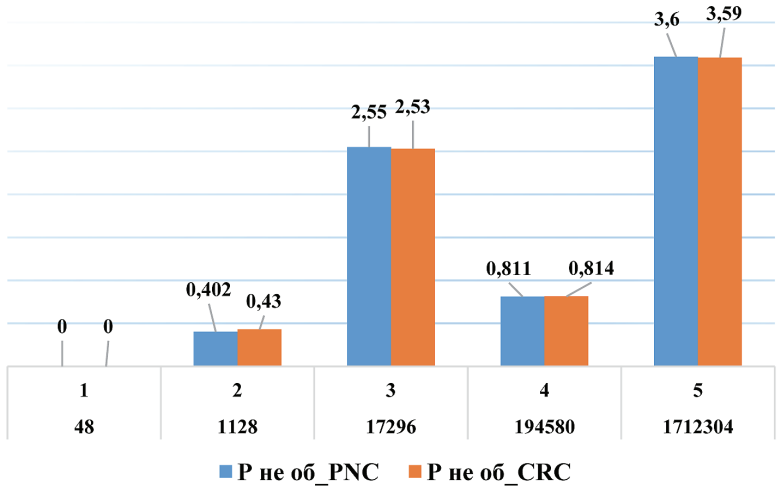

Рисунок 8. Сравнение результатов оценки значении $P_{не\ об}$, представленных в логарифмическом масштабе, для модели PNC и CRC на необнаруженных ошибочных пакетах

В качестве примера рассмотрим следующую задачу, составленную на основании Рисунка 8. Расчет выполнен с помощью формулы (5):

$$\frac{48 \cdot 47}{2!} = 1128.$$

В целях упрощения действий приведем ее к следующему виду в соответствии с формулой (6):

$$\frac{447}{1128} = 0,402.$$

На Рисунке 8 видно, что модель PNC по сравнению с традиционной моделью CRC обеспечивает меньше вероятности необнаружения искажений, когда количество ошибок оказывается нечетным. Следует отметить, что при четных ошибочных битах модель CRC оказывается лучше.

Заключение

В статье приведено сравнение модели PNC с традиционной моделью CRC. Показано, что у CRC увеличение размера пакета данных при фиксированном размере порождающего полинома увеличивает число тактов.

В статье также показано, что количество таков, требуемых для вычисления контрольной суммы пакета данных в модели PNC, не зависит от размера пакета при фиксированном размере порождающего полинома.

Характеристика обнаружения ошибки не снижается в PNC в отличие от CRC, когда количество ошибок пакетов нечетное. Анализ сравнения позволяет сказать, что PNC не на много уступает при четном количестве ошибок, но показывает лучшие результаты, чем CRC, при нечетном количестве ошибок. Результаты анализа открывают путь для дальнейшего исследования по снижению вычислительной сложности операции расчета контрольных сумм.

Литература

1. ГОСТ Р МЭК 61508-4–2007. Группа Т51. Функциональная безопасность систем электрических, электронных, программируемых электронных, связанных с безопасностью.

2. *Макуильям Ф.Дж, Слоан Н.Дж.А.* Псевдослучайные последовательности и таблицы // ТИИЭР. 1976. № 12. С. 80–95.

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

4. *Ромащенко А., Румянцев А., Шень А.* Заметки по теории кодирования. 2-е изд., испр. и доп. М.: МЦНМО, 2017. 88 с. ISBN 978-5-4439-0689-8

5. *Турдиев О.А., Клименко С.В., Тухтаходжаев А.Б.* Оценки эффективности обнаружения ошибок контрольного суммирования (CRC) передаваемых данных // Известия СПбГЭТУ «ЛЭТИ». 2019. № 8.

6. *Турдиев О.А., Яковлев В.В., Клименко С.В., Болтаев А.Х.* Исследование формирования блоковой контрольный суммы (BCC) передаваемых данных // Известия СПбГЭТУ «ЛЭТИ». 2019. № 6.

7. *Яковлев В.В., Кушназаров Ф.И.* Оценка влияния помех на производительность протоколов канального уровня // Известия Петербургского государственного университета путей сообщения. 2015. Вып. 1 (42). С. 133–138.

8. *Яковлев В.В., Федоров Р.Ф.* Стохастические вычислительные машины. Л.: Машиностроение, 1974. 304 с.
9. Eurocontrol – FAQ: Technologies. Available at: http://www.eurocontrol.int/aim/public/faq/chain faq3.html. European Organisation for the Safety of Air Navigation (date of the application: 29 April 2009).
10. *Anachriz* (1999). CRC and how to Reverse it. Retrieved 21 January 2010. Online essay with example x86 assembly code.
11. *Blahut R.E.* (2003) Algebraic Codes for Data Transmission.
12. *Cam-Winget N., Nancy R.H., Russ D.W., David J.W.* (2003). Security Flaws in 802.11 Data Link Protocols. *Communications of the ACM*, no. 46 (5), pp. 35–39.
13. *Halsall F.* (1996) Data communications, computer networks and open systems. Addison-Wesley: Pearson Education, 907 p.
14. *Halsall F.* (2005) Fifth edition, computer networks and the Internet. Addison-Wesley: Pearson Education, 803 p.
15. *Lin S. and Costello D.J.* (1983) J. Error Control Coding: Fundamentals and Applications. Prentice-Hall, Inc., EnglewoodCliffs, N. J.
16. *Peterson W.W. and Brown D.T.* (1961) Cyclic Codes for Error Detection. *Proceedings of the IRE*, 49:228.
17. *Ritter Terry* (1986) The Great CRC Mystery. Dr. Dobb's J. 11 (2): 26–34, 76–83. Available at: http://www.ciphersbyritter.com/ARTS/ CRCMYST HTM (date of the application: 21 May 2009).
18. *Ross N.W.* (1993) An Elementary Guide to CRC Error Detection Algorithms.
19. *Stigge Martin, PlötzHenryk, Müller Wolf, Redlich Jens-Peter* (2006). Reversing CRC – Theory and Practice. Available at: http://sar.informatik.hu-berlin.de/research/publications/SAR-PR-2006-05/SAR-PR-2006-05_.pdf Berlin: Humboldt University Berlin, p. 17.
20. *Suiny W. and Brown D.* (1961) Cyclic codes for error detection. *Proceedings of the IRE*, vol. 49, no. 1, pp. 228–235.

References

1. GOST R IEC 61508-4-2007. Group T51. *Funktsional'naya bezopasnost' sistem elektricheskikh, elektronnykh, programmiruemykh elektronnykh, svyazannykh s bezopasnost'yu* [Functional safety of electrical, electronic, programmable electronic systems related to safety] (in Russian).
2. McWilliams F.J., Sloan J.A. (1976) *Psevdosluchaynye posledovatel'nosti i tablitsy* [Pseudo-random sequences and tables]. *TIIER*, no. 12, pp. 80–95 (in Russian).
3. Olifer V.G., Olifer N.A. (2008) Komp'yuternye *seti. Printsipy, tekhnologii, protokoly* [Computer networks. Principles, technologies, protocols]. St. Petersburg Piter Publishing, 958 p. (in Russian).
4. Romashchenko A., Rumyantsev A., Shen A. (2017) *Zametki po teorii kodirovaniya* [Notes on coding theory]. Moscow, MTsNMO Publishing, 88 p. (in Russian). ISBN 978-5-4439-0689-8
5. Turdiev O.A., Klimenko S.V., Tukhtakhodzhaev A.B. (2019) *Otsenki effektivnosti obnaruzheniya oshibok kontrol'nogo summirovaniya (CRC) peredavaemykh dannykh* [Evaluations

of the efficiency of detecting checksum errors (CRC) of transmitted data]. *Izvestiya SPbGETU "LETI"*, no. 8 (in Russian).

6. Turdiev O.A., Yakovlev V.V., Klimenko S.V., BoltaevA.Kh. (2019) *Issledovanie formirovaniya blokovoy kontrol'nyy summy (BCC) peredavaemykh dannykh* [Investigation of the formation of the block checksum (BCC) of the transmitted data]. *Izvestiya SPbGETU "LETI"*, no. 6 (in Russian).

7. Yakovlev V.V., Kushnazarov F.I. (2015) *Otsenka vliyaniya pomekh na proizvoditel'nost' protokolov kanal'nogo urovnya* [Petersburg. state university of communication lines]. *Izvestiya Peterburgskogo gos. un-ta putey soobshcheniya*, no. 1 (42), pp. 133–138 (in Russian).

8. Yakovlev V.V., Fedorov R.F. (1974) *Stokhasticheskie vychislitel'nye mashiny* [Stochastic computing machines]. Leningrad, Mashinostroenie Publishing, 304 p. (in Russian).

9. Eurocontrol – FAQ: Technologies. Available at: http://www.eurocontrol.int/aim/public/faq/chain faq3.html. European Organisation for the Safety of Air Navigation (date of the application: 29 April 2009).

10. Anachriz (1999). CRC and how to Reverse it. Retrieved 21 January 2010. Online essay with example x86 assembly code.

11. Blahut R.E. (2003) Algebraic Codes for Data Transmission.

12. Cam-Winget N., Nancy R.H., Russ D.W., David J.W. (2003). Security Flaws in 802.11 Data Link Protocols. *Communications of the ACM*, no. 46 (5), pp. 35–39.

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

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

15. Lin S. and Costello D.J. (1983) Jr. Error Control Coding: Fundamentals and Applications. Prentice-Hall, Inc., Englewood Cliffs, N. J.

16. Peterson W.W. and Brown D.T. (1961) Cyclic Codes for Error Detection. *Proceedings of the IRE*, 49:228.

17. Ritter Terry (1986) The Great CRC Mystery. Dr. Dobb's J. 11 (2): 26–34, 76–83. Available at: http://www.ciphersbyritter.com/ARTS/ CRCMYST HTM (date of the application: 21 May 2009).

18. Ross N.W. (1993) An Elementary Guide to CRC Error Detection Algorithms.

19. Stigge Martin, PlötzHenryk, Müller Wolf, Redlich Jens-Peter (2006). Reversing CRC – Theory and Practice. Available at: http://sar.informatik.hu-berlin.de/research/publications/SAR-PR-2006-05/SAR-PR-2006-05_.pdf Berlin: Humboldt University Berlin, p. 17.

20. Suiny W. and Brown D. (1961) Cyclic codes for error detection. *Proceedings of the IRE*, vol. 49, no. 1, pp. 228–235.