Федеральное агентство связи

Сибирский Государственный Университет Телекоммуникаций и

Информатики

**Кафедра ТС и ВС**

**Лабораторная работа № 5**

**По дисциплине:**

**Основы системы мобильной связи**

**Выполнил:** Попов Роман Игоревич

**Группа:** ИА-131

**Вариант:** 19

**Проверила:** Дроздова Вера Генадьевна

Новосибирск, 2023г.

**Занятие №5. Циклический избыточный код. CRC**

**1. Цель работы**

Получить представление о том, как осуществляется проверка на наличие ошибок в пакетах с данными в современных системах связи (Error detection) посредством использования циклического избыточного кода CRC (Cyclic Redundancy Check).

**2. Краткие теоретические сведения**

Псевдослучайные двоичные последовательности CRC — циклический избыточный код, иногда называемый также контрольным кодом или контрольной суммой. CRC – это добавочная порция избыточных бит, вычисляемых по заранее известному алгоритму на основе исходного передаваемого пакета данных (информационной битовой последовательности), которое передаётся вместе с самим пакетом по каналам связи (добавляется после информационных битов) и служит для контроля его безошибочной передачи. Простыми словами, CRC – это остаток от двоичного деления оригинального пакета с данными на какое-то двоичное n-разрядное число (порождающий полином), и его длина будет равна n-1 бит. Рассмотрим пример, где имеется 7 бит данных: 100100 и 4-битный порождающий полином 1101. Требуется определить CRC. Для того, чтобы выполнить деление этих битовых последовательностей нужно в конце последовательности с данными добавить n-1 нулей, как показано ниже, где n=4, для нашего случая. Делитель - 1 1 0 1 | 1 0 0 1 0 0 0 0 0 - Делимое (данные+n-1 нулей). Основной операцией, используемой при делении бинарных чисел, является исключающее ИЛИ (XOR)

**3. Задание**

В рамках данной работы студенты должны научиться вычислять CRCпоследовательности, а также на их основании детектировать ошибки.  
Программа на языке Си:  
  
#include <stdio.h>

#include <time.h>

#include <stdlib.h>

#define NLEN 3000

#define GLEN 8

int main() {

srand(time(NULL));

int k = 0;

int G[GLEN] = {};

int N[NLEN] = {};

int nn[NLEN] = {};

//Заполение пораждающего полинома G = x^7+x^6+x+1 (11000011)

for(int i = 0; i < GLEN; i++) {

if( i == 0 || i == 1 || i == 6 || i == 7)

G[i] = 1;

else

G[i] = 0;

}

//Заполнение случайными 0/1

for(int i = 0; i < NLEN; i++){

N[i] = rand()%2;

nn[i] = N[i];

}

// Добавление нулей в конец

for(int i = 0; i < NLEN - GLEN+1; i++)

N[i] = 0;

//Вычисление XOR и остатка от деления

for(int i = 0; i < NLEN-GLEN+1; i++) {

if(N[i] == 1) {

for(int j = 0; j < GLEN; j++)

N[i+j] ^= G[j];

}

}

//Добавление остатка в конец

for (int i = NLEN-GLEN+1; i > NLEN; i++)

nn[i] = N[i];

for(int i = NLEN-GLEN+1; i < NLEN; i++)

printf("%d", N[i]);

printf(" - CRC\n");

for(int i = 0; i < NLEN-GLEN+1; i++) {

if(nn[i] == 1) {

for(int j = 0; j < GLEN; j++)

nn[i+j] ^= G[j];

}

}

for(int i = NLEN-GLEN+1; i < NLEN; i++)

printf("%d", nn[i]);

printf(" - CRC\n");

for(int i = 0; i<NLEN; i++){

int nnn[NLEN];

for(int i = 0; i<NLEN; i++)

nnn[i] = nn[i];

// Искажение битов по очереди

if (nnn[i] == 1)

nnn[i] = 0;

else

nnn[i] = 1;

// Искажение рандомных битов

if (nnn[rand()%NLEN] == 1)

nnn[rand()%NLEN] = 0;

else

nnn[rand()%NLEN] = 1;

// Вычисление XOR и остатка от деления

for(int j=0; j < NLEN-GLEN+1; j++){

if(nnn[j] == 1){

for(int z=0; z < GLEN; z++) nnn[j+z] ^= G[z];

}

}

// Проверка на ошибки

for(int j = NLEN-GLEN+1; j<NLEN; j++){

if(nnn[j] == 1)

break;

if((j == NLEN-1) & (nnn[j] == 0))

k++;

}

}

printf("Число нераспознования ошибок - %d\n",k);

return 0;

}

**4. Контрольные вопросы**

1) В мобильных сетях CRC-проверки используются для обнаружения ошибок при передаче данных. CRC (циклический избыточный код) - это метод проверки целостности данных, который основывается на вычислении контрольной суммы, которая добавляется к передаваемым данным. При получении данных, получатель также вычисляет контрольную сумму и сравнивает её с переданной контрольной суммой. Если значения не совпадают, это указывает на возможные ошибки в передаче данных.

2) Порождающий полином - это многочлен, который используется в алгоритме CRC для вычисления контрольной суммы. Он задает правила и параметры, по которым вычисляется контрольная сумма. Порождающий полином определяет длину контрольной суммы и определяет, какие значения будут использоваться при вычислении.

3) Чтобы вычислить CRC для пакета с данными, следует использовать алгоритм, который определен в соответствии с заданным порождающим полиномом. Этот алгоритм применяется к каждому биту данных для генерации контрольной суммы. Обычно вычисление CRC включает деление пакета данных на порождающий полином с использованием операции XOR и получение контрольной суммы в результате этого деления. Полученная контрольная сумма включается в пакет данных и передается вместе с ним для последующей проверки целостности приемником.

QR-КОТ :

