Министерство образования Республики Беларусь

Учреждение образования

“Белорусский государственный университет информатики и радиоэлектроники”

Факультет компьютерных систем и сетей  
Кафедра электронных вычислительных машин  
Дисциплина: Основы компьютерных сетей

Отчет по лабораторной работе № 4

на тему

“Программная реализация упрощенного варианта алгоритма CSMA/CD”

Выполнил:  
студент группы 150501 Климович А.Н.

Проверил:  
старший преподаватель Глецевич И.И.

Минск 2023

**1 ТЕОРЕТИЧЕСКАЯ ЧАСТЬ**

**1.1 Алгоритм CSMA/CD на стороне передатчика**



**2 ПРАКТИЧЕСКАЯ ЧАСТЬ**

**2.1 Изменения и дополнения кода программы**

**Файл MainWindow.cpp:**

void MainWindow::sendMessage()

{

.

. (ЛАОРАТОРНАЯ РАБОТА №3)

.

QString collisionStatus, modifiedFrame, fullCollisionStatus = "";

for(size\_t i = 0; i < countOfFrames; ++i)

{

Frame frame(message.mid(position, DATA\_SIZE), port);

modifiedFrame = frame.sendFrame(port, collisionStatus);

fullCollisionStatus += collisionStatus;

position += DATA\_SIZE;

bytesSent += modifiedFrame.size();

}

// Если сообщение по длине меньше DATA\_SIZE, то есть меньше одного кадра

if(message.size() % DATA\_SIZE != 0)

{

Frame frame(message.mid(position).leftJustified(DATA\_SIZE, '?') , port);

modifiedFrame = frame.sendFrame(port, collisionStatus);

fullCollisionStatus += collisionStatus;

bytesSent += modifiedFrame.size();

}

ui->status\_textBrowser->append("Bytes sent: " + QString::number(bytesSent));

ui->status\_textBrowser->append(fullCollisionStatus);

ui->input\_textEdit->clear(); // очищаем строку для ввода сообщений

}

}

**Файл Frame.cpp:**

QString Frame::sendFrame(QSerialPort& port, QString& collisionStatus)

{

.

. ( ЛАБОРАТОРНАЯ РАБОТА №3)

.

collisionStatus = "";

// Передаем кадр посимвольно

for(auto bit : frame)

{

collisionStatus += CSMA\_CD::sendBit(bit);

if(!collisionStatus.contains("Error: bit collision"))

{

port.write(QString(bit).toStdString().c\_str());

CSMA\_CD::waitForCollisionWindow();

}

}

return frame;

}

**Файл CSMA\_CD.h:**

#include <windows.h>

#include <random>

#include <QString>

#include <QSerialPort>

#define MAX\_ATTEMPT\_NUMBER 10

#define COLLISION\_WINDOW\_TIME 20

#define JAM\_SIGNAL\_TIME 5

class CSMA\_CD

{

public:

QString sendBit(QChar bit);

bool isChannelBusy();

bool isCollisionDetected();

void waitForCollisionWindow();

void randomDelay(unsigned int attemptNumber);

void sendJamSignal();

};

**Файл CSMA\_CD.cpp:**

#include "CSMA\_CD.h"

QString CSMA\_CD::sendBit(QChar bit)

{

QString collisionStatus = "";

unsigned int attemptNumber = 0;

while (true)

{

if(!this->isChannelBusy())

{

if(!this->isCollisionDetected())

{

collisionStatus += '-';

break;

}

else

{

collisionStatus += '+';

sendJamSignal();

waitForCollisionWindow();

++attemptNumber;

if(attemptNumber > MAX\_ATTEMPT\_NUMBER)

{

collisionStatus = "Error: bit collision";

break;

} else {

randomDelay(attemptNumber);

}

}

}

}

return collisionStatus + " ";

}

bool CSMA\_CD::isChannelBusy()

{

return rand() % 10 < 7 ? true : false;

}

bool CSMA\_CD::isCollisionDetected()

{

return rand() % 10 < 3 ? true : false;

}

void CSMA\_CD::waitForCollisionWindow()

{

Sleep(COLLISION\_WINDOW\_TIME);

}

void CSMA\_CD::randomDelay(unsigned int attemptNumber)

{

unsigned int power = attemptNumber < MAX\_ATTEMPT\_NUMBER ? attemptNumber : MAX\_ATTEMPT\_NUMBER;

unsigned int delay = rand() % (unsigned int)(pow(2, power) + 1);

Sleep(delay);

}

void CSMA\_CD::sendJamSignal()

{

Sleep(JAM\_SIGNAL\_TIME);

}