БГУИР

Кафедра ЭВМ

Отчет по лабораторной работе

Тема: «Детерминированные методы доступа к моноканалу»

Выполнил:

студент группы 950503 Пастернак А.С.

Проверил:

ассистент Марцинкевич В.А.

Минск

2021

Постановка задачи:

Реализовать алгоритм Token Ring.

Листинг программы:

Функция принятия пакета станцией.

```
void Station::receivePackage(Package package)
{
       this->package = package;
       Sleep(100);
       cout << "Station " << (int)this->id << " received " << package << endl;
       flag = true;
}
```

Функция пересылки пакета станцией.

```
void Station::sendPackage()
{
       cout << "Station " << (int)this->id << " sent " << package << endl;
       nextStation->receivePackage(package);
}
```

Механизм Token Ring:

```
void Station::routine()
{
       while (active) {
              if (flag) {
                     if (package.getToken() == 1)
                     {
                            cout << "Station " << (int)this->id << " received token" <<
endl;
                            if (rand() % 2)
                            {
                                   cout << "Station " << (int)this->id << " passing token"
<< endl;
                                   sendPackage();
                                   flag = false;
                                   continue;
                            }
                            cout << "Station " << (int)this->id << " sending message" <<
endl;
                            package = Package(generateDestination(), id, generateData());
                            sendPackage();
                            flag = false;
                            while (!flag);
                            if (package.getSource() == id)
                            {
                                   if (package.getStatus() == 1)
                                   {
                                          cout << "Station " << (int)this->id << "
successful delivery package" << endl;
                                          flag = false;
                                          package = Package();
                                          cout << "Station " << (int)this->id << " passing
token" << endl;
                                          sendPackage();
```

```
                        continue;
                    }
                    else
                        cout << "Station " << (int)this->id << " 
destination didn't change status" << endl;
                }
                else
                {
                    cout << "Station " << (int)this->id << " get wrong 
package" << endl;
                }
                package = Package();
                cout << "Station " << (int)this->id << " passing token" << 
endl;
                sendPackage();
            }
            else {
                if (package.getDestination() == id) {
                    cout << "\nStation " << (int)this->id << " got message" 
<< endl;
                    package.setStatus(1);
                }
                cout << "Station " << (int)this->id << " passing message" << 
endl;
                sendPackage();
            }
            flag = false;
        }
        Sleep(1000);
    }
}
```

Структура пакета (заполнение начальных данных):

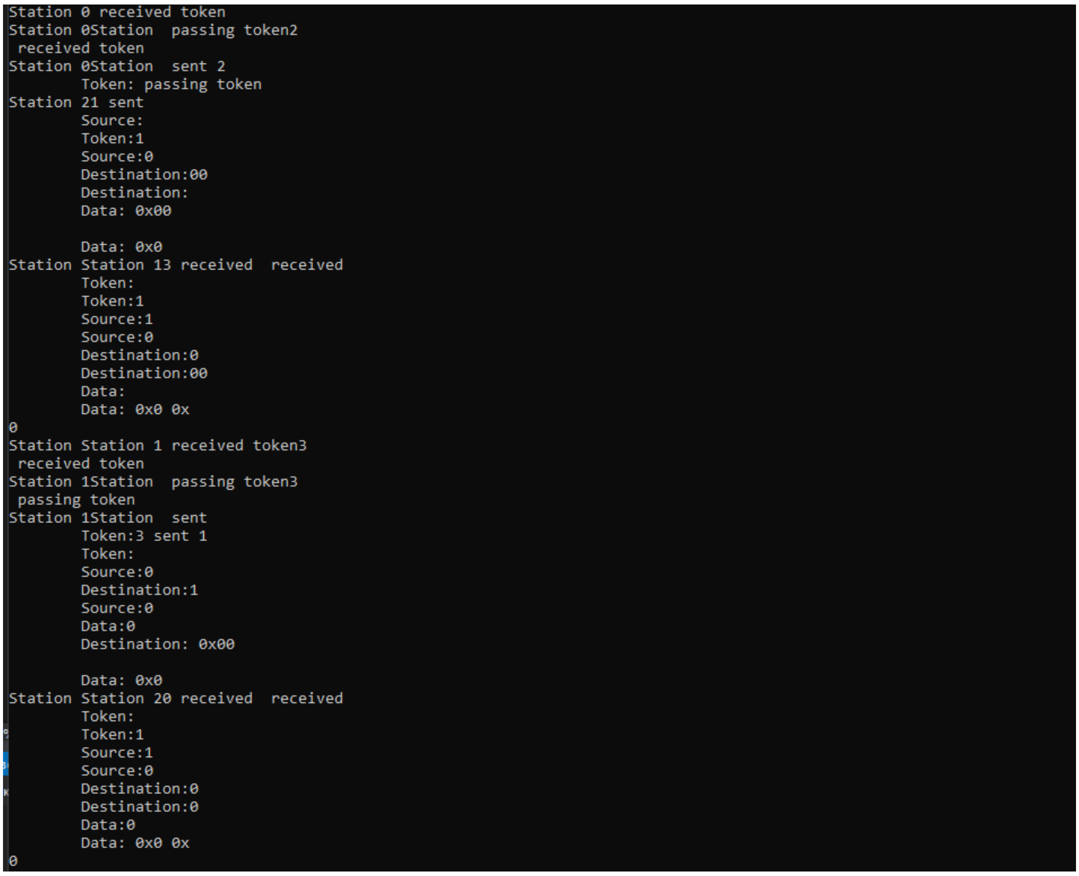
```
Package::Package(byte destination, byte source, unsigned data) {

    this->destinationADR = destination;
    this->sourceADR = source;
    this->data = data;
}
```

Структура пакета (выдача токенов, получение состояния и т.д.):

```
byte Package::getToken() {
    return this->token;
}

void Package::setControl(byte byte) {
    this->token = byte;
}

byte Package::getDestination() {
    return this->destinationADR;
}

void Package::setDestination(byte byte) {
    this->destinationADR = byte;
}

byte Package::getSource() {
    return this->sourceADR;
}
```

```cpp
void Package::setSource(byte byte) {
	this->sourceADR = byte;
}

unsigned Package::getData() {
	return this->data;
}

void Package::setData(unsigned byte) {
	this->data = byte;
}

byte Package::getStatus() {
	return this->status;
}

void Package::setStatus(byte byte) {
	this->status = byte;
}
```

Тестовые примеры:



```
Station 0 received token
Station 0Station  passing token2
 received token
Station 0Station  sent 2
        Token: passing token
Station 21 sent
        Source:
        Token:1
        Source:0
        Destination:00
        Destination:
        Data: 0x00

        Data: 0x0
Station Station 13 received  received
        Token:
        Token:1
        Source:1
        Source:0
        Destination:0
        Destination:00
        Data:
        Data: 0x0 0x
0
Station Station 1 received token3
 received token
Station 1Station  passing token3
 passing token
Station 1Station  sent
        Token:3 sent 1
        Token:
        Source:0
        Destination:1
        Source:0
        Data:0
        Destination: 0x00

        Data: 0x0
Station Station 20 received  received
        Token:
        Token:1
        Source:1
        Source:0
        Destination:0
        Destination:0
        Data:0
        Data: 0x0 0x
0
```

Заключение:

В данной лабораторной работе разработан модуль облегченного алгоритма CSMA/CD.