МИНОБРНАУКИ РОССИИ

федеральное государственное бюджетное образовательное учреждение

высшего образования

«Череповецкий государственный университет»

**Лабораторная работа № 2**

**Построение локальных сетей по стандартам физического и канального уровней**

**Выполнил:**

студент гр. 1ИВТпб-01-31оп

Климов А.Г.  
**Проверил:** преподаватель

Селяничев О.Л.  
Отметка о зачете:

Череповец

2018 год

Цель работы: проверка корректности работы сети Ethernet, состоящей из сегментов различной природы.

**Задание**

Напишите программу, позволяющую:

1. Задавать структуру сети.

2. Рассчитывать удвоенную задержку распространения сигнала между двумя самыми удаленными станциями.

3. Рассчитывать сокращения межкадрового интервала при прохождении последовательности кадров через все повторители.

4. Сравнивать рассчитанные показатели с критическими значениями и выводить результат пользователю.

Соблюдение многочисленных ограничений, установленных для различных стандартов физического уровня сетей Ethernet, гарантирует корректную работу сети (естественно, при исправном состоянии всех элементов физического уровня).

Чтобы сеть Ethernet, состоящая из сегментов различной физической природы, работала корректно, необходимо выполнение четырех основных условий:

* количество станций в сети не более 1024;
* максимальная длина каждого физического сегмента не более величины, определенной в соответствующем стандарте физического уровня;
* время двойного оборота сигнала (Path Delay Value, PDV) между двумя самыми удаленными друг от друга станциями сети не более 575 битовых интервала;
* сокращение межкадрового интервала IPG (Path Variability Value, PVV) при прохождении последовательности кадров через все повторители должно быть не больше, чем 49 битовых интервала.

Соблюдение этих требований обеспечивает корректность работы сети даже в случаях, когда нарушаются простые правила конфигурирования, определяющие максимальное количество повторителей и общую длину сети в 2500 м.

**Расчет PDV**

Для упрощения расчетов обычно используются справочные данные IEEE, содержащие значения задержек распространения сигналов в повторителях, приемопередатчиках и различных физических средах. В табл. 1 приведены данные, необходимые для расчета значения PDV для всех физических стандартов сетей Ethernet. Битовый интервал обозначен как bt.

Комитет 802.3 старался максимально упростить выполнение расчетов, поэтому данные, приведенные в таблице, включают сразу несколько этапов прохождения сигнала. Например, задержки, вносимые повторителем, состоят из задержки входного трансивера, задержки блока повторения и задержки выходного трансивера. Тем не менее, в таблице все эти задержки представлены одной величиной, названной базой сегмента. Чтобы не нужно было два раза складывать задержки, вносимые кабелем, в таблице даются удвоенные величины задержек для каждого типа кабеля.

Таблица 1

Данные для расчета значения PDV

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Тип сегмента** | **База левого сегмента, bt** | **База промежуточного сегмента, bt** | **База правого сегмента, bt** | **Задержка среды на 1 м, bt** | **Максимальная длина**  **сегмента, м** |
| 10Base-5 | 11,8 | 46,5 | 169,5 | 0,0866 | 500 |
| 10Base-2 | 11,8 | 46,5 | 169,5 | 0,1026 | 185 |
| 10Base-T | 15,3 | 42,0 | 165,0 | 0,113 | 100 |
| 10Base-FB | — | 24,0 | — | 0,1 | 2000 |
| 10Base-FL | 12,3 | 33,5 | 156,5 | 0,1 | 2000 |
| FOIRL | 7,8 | 29,0 | 152,0 | 0,1 | 1000 |
| AUI (> 2 м) | 0 | 0 | 0 | 0,1026 | 2+48 |

В таблице используются также такие понятия, как левый сегмент, правый сегмент и промежуточный сегмент. Поясним эти термины на примере сети, приведенной на рис. 10.8. Левым сегментом называется сегмент, в котором начинается путь сигнала от выхода передатчика конечного узла. На примере это сегмент 1*.*Затем сигнал проходит через промежуточные сегменты 2-5и доходит до приемника наиболее удаленного узла наиболее удаленного сегмента 6, который называется правым. Именно здесь в худшем случае происходит столкновение кадров и возникает коллизия, что и подразумевается в таблице.

С каждым сегментом связана постоянная задержка, названная базой, которая зависит только от типа сегмента и от положения сегмента на пути сигнала (левый, промежуточный или правый). База правого сегмента, в котором возникает коллизия, намного превышает базу левого и промежуточных сегментов.

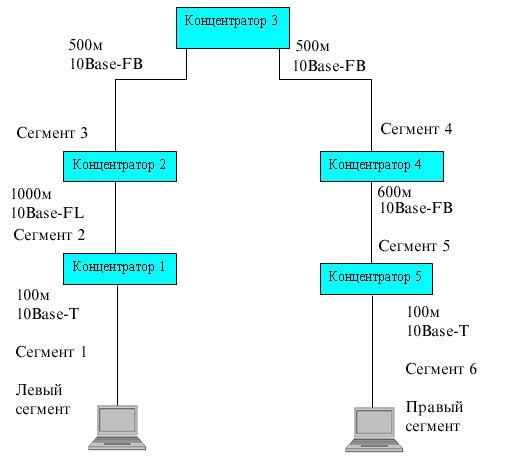


Рис.1. Пример сети Ethernet, состоящей из сегментов различных физических стандартов

Кроме этого, с каждым сегментом связана задержка распространения сигнала вдоль кабеля сегмента, которая зависит от длины сегмента и вычисляется путем умножения времени распространения сигнала по одному метру кабеля (в битовых интервалах) на длину кабеля в метрах.

Расчет заключается в вычислении задержек, вносимых каждым отрезком кабеля (приведенная в таблице задержка сигнала на 1 м кабеля умножается на длину сегмента), а затем суммировании этих задержек с базами левого, промежуточных и правого сегментов. Общее значение PDV не должно превышать 575.

Так как левый и правый сегменты имеют различные величины базовой задержки, то в случае различных типов сегментов на удаленных краях сети необходимо выполнить расчеты дважды: один раз принять в качестве левого сегмента сегмент одного типа, а во второй — сегмент другого типа. Результатом можно считать максимальное значение PDV. В нашем примере крайние сегменты сети принадлежат к одному типу — стандарту 10Base-T, поэтому двойной расчет не требуется, но если бы они были, сегментами разного типа, то в первом случае нужно было бы принять в качестве левого сегмент между станцией и концентратором 1*,*аво втором считать левым сегмент между станцией и концентратором 5.

Приведенная на рисунке сеть в соответствии с правилом 4-х хабов не является корректной — в сети между узлами сегментов 1и 6 имеется 5хабов, хотя не все сегменты являются сегментами 10Base-FB. Кроме того, общая длина сети равна 2800 м, что нарушает правило 2500 м. Рассчитаем значение PDV для нашего примера.

Левый сегмент 1:15,3 (база) + 100 х 0,113 - 26,6.

Промежуточный сегмент 2*:*33,5 + 1000 х 0,1 = 133,5.

Промежуточный сегмент 3*:*24 + 500 х 0,1 = 74,0.

Промежуточный сегмент 4*:*24 + 500 х 0,1 = 74,0.

Промежуточный сегмент 5: 24 + 600 х 0,1 = 84,0.

Правый сегмент 6*:*165 + 100 х 0,113 = 176,3.

Сумма всех составляющих дает значение PDV, равное 568,4.

Так как значение PDV меньше максимально допустимой величины 575, то эта сеть проходит по критерию времени двойного оборота сигнала несмотря на то, что ее общая длина составляет больше 2500 м, а количество повторителей — больше 4-х.

**Расчет PVV**

Чтобы признать конфигурацию сети корректной, нужно рассчитать также уменьшение межкадрового интервала повторителями, то есть величину PVV.

Для расчета PVV также можно воспользоваться значениями максимальных величин уменьшения межкадрового интервала при прохождении повторителей различных физических сред, рекомендованными IEEE и приведенными в табл.2.

Таблица 2

Сокращение межкадрового интервала повторителями

|  |  |  |
| --- | --- | --- |
| **Тип сегмента** | **Передающий сегмент, bt** | **Промежуточный** **сегмент, bt** |
| 10Base-5 или 10Base-2 | 16 | 11 |
| 10Base-FB | — | 2 |
| 10Base-FL | 10,5 | 8 |
| 10Base-T | 10,5 | 8 |

В соответствии с этими данными рассчитаем значение PVV для нашего примера.

Левый сегмент 1: 10Base-T: сокращение в 10,5 bt.

Промежуточный сегмент 2: 10Base-FL: 8.

Промежуточный сегмент 3: 10Base-FB: 2.

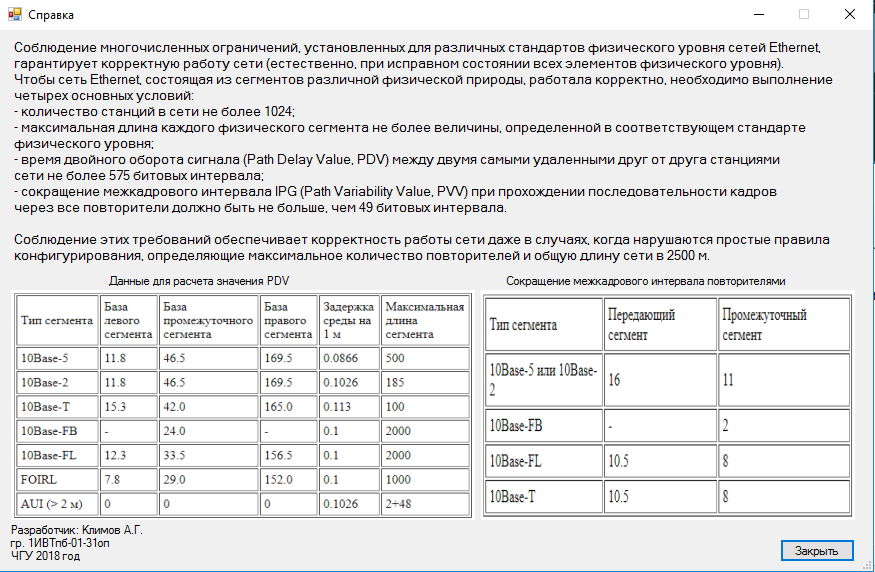
Промежуточный сегмент 4: 10Base-FB: 2.

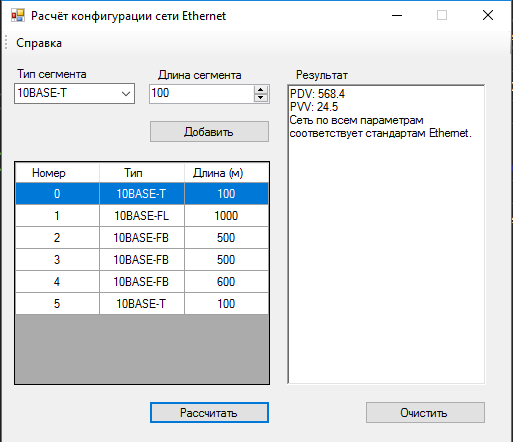
Промежуточный сегмент 5: 10Base-FB: 2.

Сумма этих величин дает значение PVV, равное 24,5, что меньше предельного значения в 49 битовых интервала.

В результате приведенная в примере сеть соответствует стандартам Ethernet по всем параметрам, связанным и с длинами сегментов, и с количеством повторителей.

**Тестирование**





**Текст программы**

**MyForm.cpp**

#include "MyForm.h"

using namespace System;

using namespace System::ComponentModel;

using namespace System::Collections;

using namespace System::Windows::Forms;

using namespace System::Data;

using namespace System::Drawing;

[STAThreadAttribute]

int Main(array<System::String ^> ^args)

{

Application::EnableVisualStyles();

Application::SetCompatibleTextRenderingDefault(false);

Project1::MyForm mainForm;

Application::Run(%mainForm);

return 0;

}

**MyForm.h**

#pragma once

#include <iostream>

#include <vector>

#include "MyForm1.h"

class Segment;

namespace Project1 {

using namespace System;

using namespace System::ComponentModel;

using namespace System::Collections;

using namespace System::Windows::Forms;

using namespace System::Data;

using namespace System::Drawing;

using namespace System::Net::NetworkInformation;

using namespace System::Text;

/// <summary>

/// —водка дл¤ MyForm

/// </summary>

public ref class MyForm : public System::Windows::Forms::Form

{

public:

MyForm(void)

{

InitializeComponent();

//

//TODO: добавьте код конструктора

//

}

protected:

/// <summary>

/// ќсвободить все используемые ресурсы.

/// </summary>

~MyForm()

{

if (components)

{

delete components;

}

}

protected:

private: System::Windows::Forms::Button^ button1;

private: System::Windows::Forms::RichTextBox^ richTextBox1;

private: System::Windows::Forms::ComboBox^ comboBox1;

private: System::Windows::Forms::Button^ button3;

private: System::Windows::Forms::Button^ button4;

private: System::Windows::Forms::DataGridView^ dataGridView1;

private: System::Windows::Forms::DataGridViewTextBoxColumn^ Column1;

private: System::Windows::Forms::DataGridViewTextBoxColumn^ Column2;

private: System::Windows::Forms::DataGridViewTextBoxColumn^ Column3;

private: int count; //Счётчик

private: System::Windows::Forms::NumericUpDown^ numericUpDown2;

private: System::Windows::Forms::ToolStrip^ toolStrip1;

private: System::Windows::Forms::ToolStripButton^ toolStripButton1;

private: System::Windows::Forms::Label^ label2;

private: System::Windows::Forms::Label^ label1;

private: System::Windows::Forms::Label^ label3;

private:

/// <summary>

/// “ребуетс¤ переменна¤ конструктора.

/// </summary>

System::ComponentModel::Container ^components;

#pragma region Windows Form Designer generated code

/// <summary>

/// ќб¤зательный метод дл¤ поддержки конструктора - не измен¤йте

/// содержимое данного метода при помощи редактора кода.

/// </summary>

void InitializeComponent(void)

{

System::Windows::Forms::DataGridViewCellStyle^ dataGridViewCellStyle1 = (gcnew System::Windows::Forms::DataGridViewCellStyle());

this->button1 = (gcnew System::Windows::Forms::Button());

this->richTextBox1 = (gcnew System::Windows::Forms::RichTextBox());

this->comboBox1 = (gcnew System::Windows::Forms::ComboBox());

this->button3 = (gcnew System::Windows::Forms::Button());

this->button4 = (gcnew System::Windows::Forms::Button());

this->dataGridView1 = (gcnew System::Windows::Forms::DataGridView());

this->Column1 = (gcnew System::Windows::Forms::DataGridViewTextBoxColumn());

this->Column2 = (gcnew System::Windows::Forms::DataGridViewTextBoxColumn());

this->Column3 = (gcnew System::Windows::Forms::DataGridViewTextBoxColumn());

this->numericUpDown2 = (gcnew System::Windows::Forms::NumericUpDown());

this->toolStrip1 = (gcnew System::Windows::Forms::ToolStrip());

this->toolStripButton1 = (gcnew System::Windows::Forms::ToolStripButton());

this->label2 = (gcnew System::Windows::Forms::Label());

this->label1 = (gcnew System::Windows::Forms::Label());

this->label3 = (gcnew System::Windows::Forms::Label());

(cli::safe\_cast<System::ComponentModel::ISupportInitialize^>(this->dataGridView1))->BeginInit();

(cli::safe\_cast<System::ComponentModel::ISupportInitialize^>(this->numericUpDown2))->BeginInit();

this->toolStrip1->SuspendLayout();

this->SuspendLayout();

//

// button1

//

this->button1->Location = System::Drawing::Point(147, 89);

this->button1->Name = L"button1";

this->button1->Size = System::Drawing::Size(121, 23);

this->button1->TabIndex = 1;

this->button1->Text = L"Добавить";

this->button1->UseVisualStyleBackColor = true;

this->button1->Click += gcnew System::EventHandler(this, &MyForm::button1\_Click);

//

// richTextBox1

//

this->richTextBox1->Location = System::Drawing::Point(285, 53);

this->richTextBox1->Name = L"richTextBox1";

this->richTextBox1->Size = System::Drawing::Size(199, 301);

this->richTextBox1->TabIndex = 2;

this->richTextBox1->Text = L"";

//

// comboBox1

//

this->comboBox1->FormattingEnabled = true;

this->comboBox1->Items->AddRange(gcnew cli::array< System::Object^ >(7) {

L"10BASE-5", L"10BASE-2", L"10BASE-T", L"10BASE-FB",

L"10BASE-FL", L"FOIRL", L"AUI"

});

this->comboBox1->Location = System::Drawing::Point(12, 52);

this->comboBox1->Name = L"comboBox1";

this->comboBox1->Size = System::Drawing::Size(121, 21);

this->comboBox1->TabIndex = 3;

this->comboBox1->SelectedIndexChanged += gcnew System::EventHandler(this, &MyForm::comboBox1\_SelectedIndexChanged);

//

// button3

//

this->button3->Location = System::Drawing::Point(363, 370);

this->button3->Name = L"button3";

this->button3->Size = System::Drawing::Size(121, 23);

this->button3->TabIndex = 9;

this->button3->Text = L"Очистить";

this->button3->UseVisualStyleBackColor = true;

this->button3->Click += gcnew System::EventHandler(this, &MyForm::button3\_Click);

//

// button4

//

this->button4->Location = System::Drawing::Point(147, 370);

this->button4->Name = L"button4";

this->button4->Size = System::Drawing::Size(121, 23);

this->button4->TabIndex = 10;

this->button4->Text = L"Рассчитать";

this->button4->UseVisualStyleBackColor = true;

this->button4->Click += gcnew System::EventHandler(this, &MyForm::button4\_Click);

//

// dataGridView1

//

this->dataGridView1->AllowUserToAddRows = false;

this->dataGridView1->AllowUserToDeleteRows = false;

this->dataGridView1->AllowUserToResizeColumns = false;

this->dataGridView1->AllowUserToResizeRows = false;

this->dataGridView1->AutoSizeColumnsMode = System::Windows::Forms::DataGridViewAutoSizeColumnsMode::Fill;

dataGridViewCellStyle1->Alignment = System::Windows::Forms::DataGridViewContentAlignment::MiddleCenter;

dataGridViewCellStyle1->BackColor = System::Drawing::SystemColors::Control;

dataGridViewCellStyle1->Font = (gcnew System::Drawing::Font(L"Microsoft Sans Serif", 8.25F, System::Drawing::FontStyle::Regular,

System::Drawing::GraphicsUnit::Point, static\_cast<System::Byte>(204)));

dataGridViewCellStyle1->ForeColor = System::Drawing::SystemColors::WindowText;

dataGridViewCellStyle1->SelectionBackColor = System::Drawing::SystemColors::Highlight;

dataGridViewCellStyle1->SelectionForeColor = System::Drawing::SystemColors::HighlightText;

dataGridViewCellStyle1->WrapMode = System::Windows::Forms::DataGridViewTriState::True;

this->dataGridView1->ColumnHeadersDefaultCellStyle = dataGridViewCellStyle1;

this->dataGridView1->ColumnHeadersHeightSizeMode = System::Windows::Forms::DataGridViewColumnHeadersHeightSizeMode::AutoSize;

this->dataGridView1->Columns->AddRange(gcnew cli::array< System::Windows::Forms::DataGridViewColumn^ >(3) {

this->Column1,

this->Column2, this->Column3

});

this->dataGridView1->Location = System::Drawing::Point(12, 130);

this->dataGridView1->Name = L"dataGridView1";

this->dataGridView1->ReadOnly = true;

this->dataGridView1->Size = System::Drawing::Size(256, 224);

this->dataGridView1->TabIndex = 11;

//

// Column1

//

this->Column1->HeaderText = L"Номер";

this->Column1->Name = L"Column1";

this->Column1->ReadOnly = true;

//

// Column2

//

this->Column2->HeaderText = L"Тип";

this->Column2->Name = L"Column2";

this->Column2->ReadOnly = true;

//

// Column3

//

this->Column3->HeaderText = L"Длина (м)";

this->Column3->Name = L"Column3";

this->Column3->ReadOnly = true;

//

// numericUpDown2

//

this->numericUpDown2->Location = System::Drawing::Point(147, 53);

this->numericUpDown2->Maximum = System::Decimal(gcnew cli::array< System::Int32 >(4) { 2000, 0, 0, 0 });

this->numericUpDown2->Name = L"numericUpDown2";

this->numericUpDown2->Size = System::Drawing::Size(121, 20);

this->numericUpDown2->TabIndex = 13;

//

// toolStrip1

//

this->toolStrip1->Items->AddRange(gcnew cli::array< System::Windows::Forms::ToolStripItem^ >(1) { this->toolStripButton1 });

this->toolStrip1->Location = System::Drawing::Point(0, 0);

this->toolStrip1->Name = L"toolStrip1";

this->toolStrip1->Size = System::Drawing::Size(509, 25);

this->toolStrip1->TabIndex = 16;

this->toolStrip1->Text = L"toolStrip1";

//

// toolStripButton1

//

this->toolStripButton1->DisplayStyle = System::Windows::Forms::ToolStripItemDisplayStyle::Text;

this->toolStripButton1->Name = L"toolStripButton1";

this->toolStripButton1->Size = System::Drawing::Size(57, 22);

this->toolStripButton1->Text = L"Справка";

this->toolStripButton1->Click += gcnew System::EventHandler(this, &MyForm::toolStripButton1\_Click);

//

// label2

//

this->label2->AutoSize = true;

this->label2->Location = System::Drawing::Point(153, 37);

this->label2->Name = L"label2";

this->label2->Size = System::Drawing::Size(91, 13);

this->label2->TabIndex = 5;

this->label2->Text = L"Длина сегмента";

//

// label1

//

this->label1->AutoSize = true;

this->label1->Location = System::Drawing::Point(12, 36);

this->label1->Name = L"label1";

this->label1->Size = System::Drawing::Size(77, 13);

this->label1->TabIndex = 4;

this->label1->Text = L"Тип сегмента";

//

// label3

//

this->label3->AutoSize = true;

this->label3->Location = System::Drawing::Point(291, 37);

this->label3->Name = L"label3";

this->label3->Size = System::Drawing::Size(59, 13);

this->label3->TabIndex = 7;

this->label3->Text = L"Результат";

//

// MyForm

//

this->AutoScaleDimensions = System::Drawing::SizeF(6, 13);

this->AutoScaleMode = System::Windows::Forms::AutoScaleMode::Font;

this->ClientSize = System::Drawing::Size(509, 411);

this->Controls->Add(this->toolStrip1);

this->Controls->Add(this->numericUpDown2);

this->Controls->Add(this->dataGridView1);

this->Controls->Add(this->button4);

this->Controls->Add(this->button3);

this->Controls->Add(this->label3);

this->Controls->Add(this->label2);

this->Controls->Add(this->label1);

this->Controls->Add(this->comboBox1);

this->Controls->Add(this->richTextBox1);

this->Controls->Add(this->button1);

this->MaximizeBox = false;

this->MaximumSize = System::Drawing::Size(525, 450);

this->MinimumSize = System::Drawing::Size(525, 450);

this->Name = L"MyForm";

this->StartPosition = System::Windows::Forms::FormStartPosition::CenterScreen;

this->Text = L"Расчёт конфигурации сети Ethernet";

this->Load += gcnew System::EventHandler(this, &MyForm::MyForm\_Load);

(cli::safe\_cast<System::ComponentModel::ISupportInitialize^>(this->dataGridView1))->EndInit();

(cli::safe\_cast<System::ComponentModel::ISupportInitialize^>(this->numericUpDown2))->EndInit();

this->toolStrip1->ResumeLayout(false);

this->toolStrip1->PerformLayout();

this->ResumeLayout(false);

this->PerformLayout();

}

#pragma endregion

double left\_base;

double intermediate\_base;

double right\_base ;

double delay;

double max\_length;

double transmitting\_segment;

double intermediate\_segment;

String ^ CurrentStr;

int RowsCount;

int CurrentRow;

int CurrentLength;

double sum\_pdv;

double sum\_pvv;

bool check = false;

double pdv(String ^ currentStr, int currentCounter, int rowsCount, int length)

{

if (currentStr == "10BASE-5") {

left\_base = 11.8;

intermediate\_base = 46.5;

right\_base = 169.5;

delay = 0.0866;

max\_length = 500;

}

if (currentStr == "10BASE-2") {

left\_base = 11.8;

intermediate\_base = 46.5;

right\_base = 169.5;

delay = 0.1026;

max\_length = 185;

}

if (currentStr == "10BASE-T") {

left\_base = 15.3;

intermediate\_base = 42.0;

right\_base = 165.0;

delay = 0.113;

max\_length = 100;

}

if (currentStr == "10BASE-FB") {

//left\_base = 0;

intermediate\_base = 24.0;

//right\_base = 169.5;

delay = 0.1;

max\_length = 2000;

}

if (currentStr == "10BASE-FL") {

left\_base = 12.3;

intermediate\_base = 33.5;

right\_base = 156.5;

delay = 0.1;

max\_length = 2000;

}

if (currentStr == "FOIRL") {

left\_base = 7.8;

intermediate\_base = 29.0;

right\_base = 152.0;

delay = 0.1;

max\_length = 1000;

}

if (currentStr == "AUI") {

left\_base = 0;

intermediate\_base = 0;

right\_base = 0;

delay = 0.1026;

max\_length = 2 + 48;

}

if (currentCounter == 0) {

if (check == false) {

sum\_pdv += left\_base + length \* delay;

}

else {

sum\_pdv += right\_base + length \* delay;

}

return sum\_pdv;

}

if (currentCounter == rowsCount) {

if (check == false) {

sum\_pdv += right\_base + length \* delay;

}

else {

sum\_pdv += left\_base + length \* delay;

}

return sum\_pdv;

}

if (currentCounter > 0 && currentCounter < rowsCount) {

sum\_pdv += intermediate\_base + length \* delay;

return sum\_pdv;

}

return 0;

}

double pvv(String ^ currentStr, int currentCounter, int rowsCount) {

if (currentStr == "10BASE-5") {

transmitting\_segment = 16;

intermediate\_segment = 11;

}

if (currentStr == "10BASE-2") {

transmitting\_segment = 16;

intermediate\_segment = 11;

}

if (currentStr == "10BASE-T") {

transmitting\_segment = 10.5;

intermediate\_segment = 8;

}

if (currentStr == "10BASE-FB") {

//transmitting\_segment = 0;

intermediate\_segment = 2;

}

if (currentStr == "10BASE-FL") {

transmitting\_segment = 10.5;

intermediate\_segment = 8;

}

if (currentStr == "FOIRL") {

//transmitting\_segment = 0;

//intermediate\_segment = 0;

}

if (currentStr == "AUI") {

//transmitting\_segment = 0;

//intermediate\_segment = 0;

}

if (currentCounter == 0) {

if (check == false) {

sum\_pvv += transmitting\_segment;

}

return sum\_pvv;

}

if (currentCounter == rowsCount) {

if (check == false) {

return sum\_pvv;

}

else {

sum\_pvv += transmitting\_segment;

}

}

if (currentCounter > 0) {

sum\_pvv += intermediate\_segment;

return sum\_pvv;

}

return 0;

}

private: System::Void MyForm\_Load(System::Object^ sender, System::EventArgs^ e) {

this->comboBox1->SelectedIndex = 0;

dataGridView1->RowHeadersVisible = false;

//dataGridView1->ColumnHeadersVisible = false;

this->dataGridView1->MultiSelect = false;

this->dataGridView1->SelectionMode = DataGridViewSelectionMode::FullRowSelect;

this->dataGridView1->AllowUserToAddRows = false;

this->dataGridView1->RowsDefaultCellStyle->Alignment = DataGridViewContentAlignment::MiddleCenter;

count = 0;

}

private: System::Void button1\_Click(System::Object^ sender, System::EventArgs^ e) {

this->dataGridView1->Rows->Add(count, comboBox1->Text, numericUpDown2->Text);

count++;

}

private: System::Void button3\_Click(System::Object^ sender, System::EventArgs^ e) {

this->dataGridView1->RowCount = 0;

count = 0;

this->richTextBox1->Clear();

}

private: System::Void button4\_Click(System::Object^ sender, System::EventArgs^ e) {

RowsCount = dataGridView1->Rows->Count - 1;

sum\_pdv = 0;

sum\_pvv = 0;

check = false;

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

CurrentRow = Convert::ToInt32(dataGridView1[0, i]->Value);

CurrentStr = dataGridView1[1, i]->Value->ToString();

CurrentLength = Convert::ToInt32(dataGridView1[2, i]->Value);

sum\_pdv + pdv(CurrentStr, CurrentRow, RowsCount, CurrentLength);

sum\_pvv + pvv(CurrentStr, CurrentRow, RowsCount);

}

if (sum\_pdv < 576 && sum\_pvv < 50) {

this->richTextBox1->Text += "PDV: " + sum\_pdv + "\n";

this->richTextBox1->Text += "PVV: " + sum\_pvv + "\n";

this->richTextBox1->Text += "Сеть по всем параметрам соответствует стандартам Ethernet.\n";

}

else

{

this->richTextBox1->Text += "PDV: " + sum\_pdv + "\n";

this->richTextBox1->Text += "PVV: " + sum\_pvv + "\n";

this->richTextBox1->Text += "Сеть не работоспособна!\n";

}

RowsCount = dataGridView1->Rows->Count - 1;

sum\_pdv = 0;

sum\_pvv = 0;

check = true;

if (dataGridView1[1, 0]->Value->ToString() != dataGridView1[1, RowsCount]->Value->ToString())

{

for (int i = RowsCount; i >= 0; i--) {

CurrentRow = Convert::ToInt32(dataGridView1[0, i]->Value);

CurrentStr = dataGridView1[1, i]->Value->ToString();

CurrentLength = Convert::ToInt32(dataGridView1[2, i]->Value);

sum\_pdv + pdv(CurrentStr, CurrentRow, RowsCount, CurrentLength);

sum\_pvv + pvv(CurrentStr, CurrentRow, RowsCount);

}

if (sum\_pdv < 576 && sum\_pvv < 50) {

this->richTextBox1->Text += "PDV2: " + sum\_pdv + "\n";

this->richTextBox1->Text += "PVV2: " + sum\_pvv + "\n";

this->richTextBox1->Text += "Сеть по всем параметрам соответствует стандартам Ethernet.\n";

}

else

{

this->richTextBox1->Text += "PDV2: " + sum\_pdv + "\n";

this->richTextBox1->Text += "PVV2: " + sum\_pvv + "\n";

this->richTextBox1->Text += "Сеть не работоспособна!\n";

}

}

//this->richTextBox1->Text += "PDV: " + pvd( + "\n";

//this->richTextBox1->Text += "PDV: " + dataGridView1[0, i]->Value + "\n";

//this->richTextBox1->Text += "PDV: " + dataGridView1[1, i]->Value + "\n";

//pdv(comboBox1->SelectedIndex)

}

private: System::Void comboBox1\_SelectedIndexChanged(System::Object^ sender, System::EventArgs^ e) {

if (comboBox1->SelectedIndex == 0) { this->numericUpDown2->Maximum = 500; }

if (comboBox1->SelectedIndex == 1) { this->numericUpDown2->Maximum = 185; }

if (comboBox1->SelectedIndex == 2) { this->numericUpDown2->Maximum = 100; }

if (comboBox1->SelectedIndex == 3) { this->numericUpDown2->Maximum = 2000; }

if (comboBox1->SelectedIndex == 4) { this->numericUpDown2->Maximum = 2000; }

if (comboBox1->SelectedIndex == 5) { this->numericUpDown2->Maximum = 1000; }

if (comboBox1->SelectedIndex == 6) { this->numericUpDown2->Maximum = 2 + 48; }

}

private: System::Void toolStripButton1\_Click(System::Object^ sender, System::EventArgs^ e) {

MyForm1 ^form = gcnew MyForm1();

form->ShowDialog();

}

};

}

**MyForm1.h**

#pragma once

namespace Project1 {

using namespace System;

using namespace System::ComponentModel;

using namespace System::Collections;

using namespace System::Windows::Forms;

using namespace System::Data;

using namespace System::Drawing;

using namespace System::Net::NetworkInformation;

using namespace System::Text;

/// <summary>

/// Summary for MyForm1

/// </summary>

public ref class MyForm1 : public System::Windows::Forms::Form

{

public:

MyForm1(void)

{

InitializeComponent();

//

//TODO: Add the constructor code here

//

}

protected:

/// <summary>

/// Clean up any resources being used.

/// </summary>

~MyForm1()

{

if (components)

{

delete components;

}

}

private: System::Windows::Forms::Button^ button1;

private: System::Windows::Forms::Label^ label1;

private: System::Windows::Forms::Label^ label2;

private: System::Windows::Forms::PictureBox^ pictureBox2;

private: System::Windows::Forms::PictureBox^ pictureBox1;

private: System::Windows::Forms::Label^ label3;

private: System::Windows::Forms::Label^ label4;

protected:

private:

/// <summary>

/// Required designer variable.

/// </summary>

System::ComponentModel::Container ^components;

#pragma region Windows Form Designer generated code

/// <summary>

/// Required method for Designer support - do not modify

/// the contents of this method with the code editor.

/// </summary>

void InitializeComponent(void)

{

System::ComponentModel::ComponentResourceManager^ resources = (gcnew System::ComponentModel::ComponentResourceManager(MyForm1::typeid));

this->button1 = (gcnew System::Windows::Forms::Button());

this->label1 = (gcnew System::Windows::Forms::Label());

this->label2 = (gcnew System::Windows::Forms::Label());

this->pictureBox2 = (gcnew System::Windows::Forms::PictureBox());

this->pictureBox1 = (gcnew System::Windows::Forms::PictureBox());

this->label3 = (gcnew System::Windows::Forms::Label());

this->label4 = (gcnew System::Windows::Forms::Label());

(cli::safe\_cast<System::ComponentModel::ISupportInitialize^>(this->pictureBox2))->BeginInit();

(cli::safe\_cast<System::ComponentModel::ISupportInitialize^>(this->pictureBox1))->BeginInit();

this->SuspendLayout();

//

// button1

//

this->button1->Location = System::Drawing::Point(781, 509);

this->button1->Name = L"button1";

this->button1->Size = System::Drawing::Size(75, 23);

this->button1->TabIndex = 0;

this->button1->Text = L"Закрыть";

this->button1->UseVisualStyleBackColor = true;

this->button1->Click += gcnew System::EventHandler(this, &MyForm1::button1\_Click);

//

// label1

//

this->label1->AutoSize = true;

this->label1->Font = (gcnew System::Drawing::Font(L"Microsoft Sans Serif", 9.75F, System::Drawing::FontStyle::Regular, System::Drawing::GraphicsUnit::Point,

static\_cast<System::Byte>(204)));

this->label1->Location = System::Drawing::Point(12, 9);

this->label1->Name = L"label1";

this->label1->Size = System::Drawing::Size(830, 224);

this->label1->TabIndex = 1;

this->label1->Text = resources->GetString(L"label1.Text");

//

// label2

//

this->label2->AutoSize = true;

this->label2->Location = System::Drawing::Point(9, 493);

this->label2->Name = L"label2";

this->label2->Size = System::Drawing::Size(142, 39);

this->label2->TabIndex = 2;

this->label2->Text = L"Разработчик: Климов А.Г. \r\nгр. 1ИВТпб-01-31оп\r\nЧГУ 2018 год";

//

// pictureBox2

//

this->pictureBox2->Location = System::Drawing::Point(12, 260);

this->pictureBox2->Name = L"pictureBox2";

this->pictureBox2->Size = System::Drawing::Size(464, 230);

this->pictureBox2->TabIndex = 16;

this->pictureBox2->TabStop = false;

//

// pictureBox1

//

this->pictureBox1->Location = System::Drawing::Point(482, 260);

this->pictureBox1->Name = L"pictureBox1";

this->pictureBox1->Size = System::Drawing::Size(374, 230);

this->pictureBox1->TabIndex = 17;

this->pictureBox1->TabStop = false;

//

// label3

//

this->label3->AutoSize = true;

this->label3->Location = System::Drawing::Point(107, 244);

this->label3->Name = L"label3";

this->label3->Size = System::Drawing::Size(187, 13);

this->label3->TabIndex = 18;

this->label3->Text = L"Данные для расчета значения PDV";

//

// label4

//

this->label4->AutoSize = true;

this->label4->Location = System::Drawing::Point(505, 244);

this->label4->Name = L"label4";

this->label4->Size = System::Drawing::Size(286, 13);

this->label4->TabIndex = 19;

this->label4->Text = L"Сокращение межкадрового интервала повторителями";

//

// MyForm1

//

this->AutoScaleDimensions = System::Drawing::SizeF(6, 13);

this->AutoScaleMode = System::Windows::Forms::AutoScaleMode::Font;

this->ClientSize = System::Drawing::Size(874, 541);

this->Controls->Add(this->label4);

this->Controls->Add(this->label3);

this->Controls->Add(this->pictureBox1);

this->Controls->Add(this->pictureBox2);

this->Controls->Add(this->label2);

this->Controls->Add(this->label1);

this->Controls->Add(this->button1);

this->MaximizeBox = false;

this->MaximumSize = System::Drawing::Size(890, 580);

this->MinimumSize = System::Drawing::Size(890, 580);

this->Name = L"MyForm1";

this->StartPosition = System::Windows::Forms::FormStartPosition::CenterScreen;

this->Text = L"Справка";

this->Load += gcnew System::EventHandler(this, &MyForm1::MyForm1\_Load);

(cli::safe\_cast<System::ComponentModel::ISupportInitialize^>(this->pictureBox2))->EndInit();

(cli::safe\_cast<System::ComponentModel::ISupportInitialize^>(this->pictureBox1))->EndInit();

this->ResumeLayout(false);

this->PerformLayout();

}

#pragma endregion

private: System::Void MyForm1\_Load(System::Object^ sender, System::EventArgs^ e) {

this->pictureBox1->SizeMode = PictureBoxSizeMode::StretchImage;

this->pictureBox2->SizeMode = PictureBoxSizeMode::StretchImage;

pictureBox1->Load("tb2.png");

pictureBox2->Load("tb1.png");

}

private: System::Void button1\_Click(System::Object^ sender, System::EventArgs^ e) {

this->Close();

}

};

}