TracerouteV2

Software Documentation

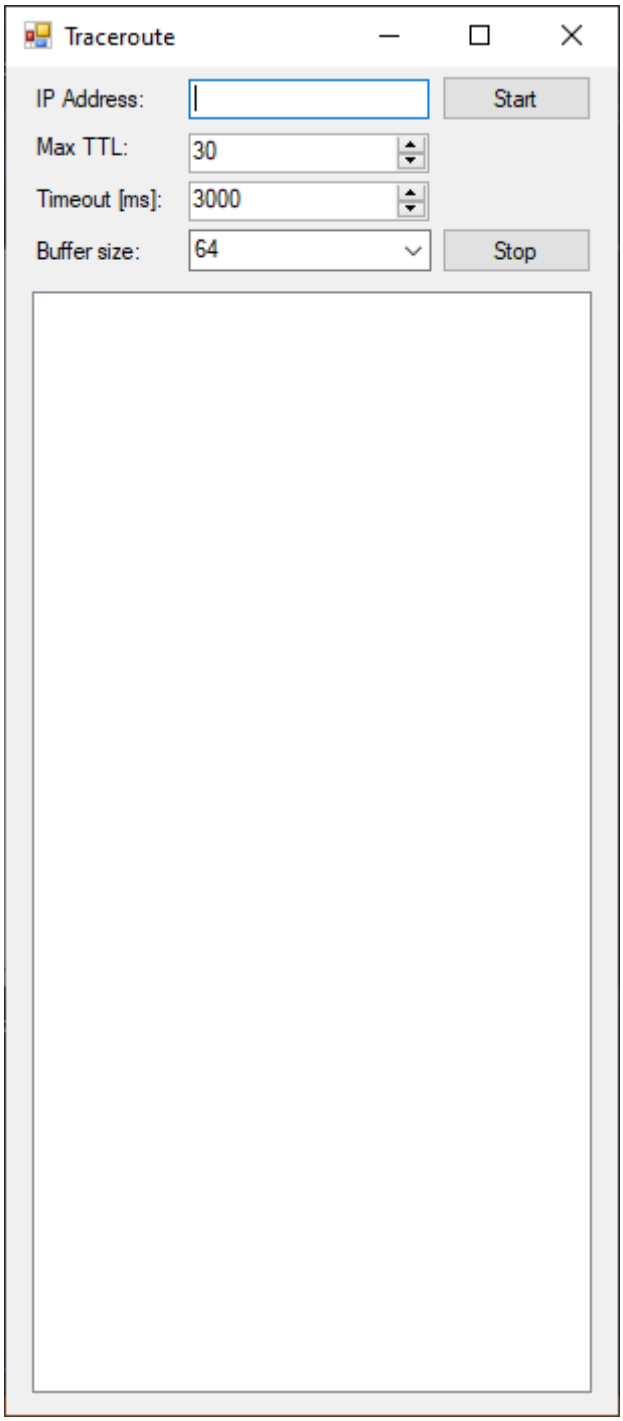
Author: matiwa

Table of contents

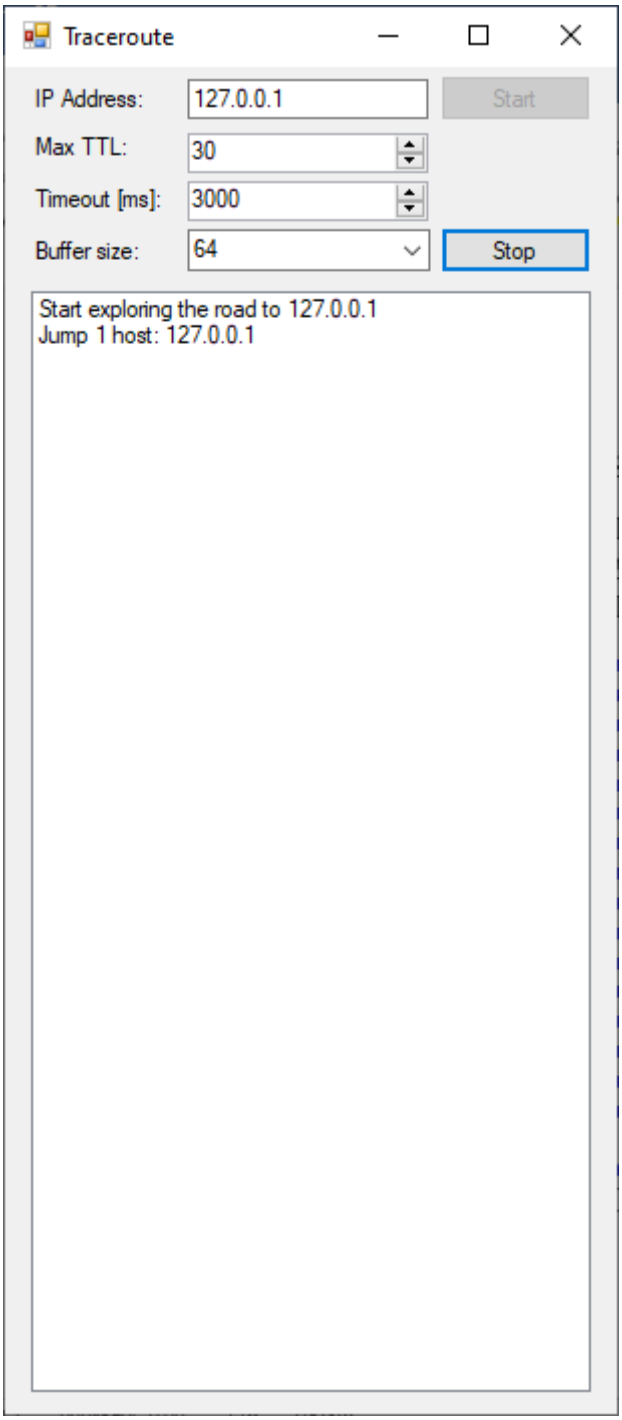
Introduction

This software documentation includes: description of the application's operation, what is needed for use, algorithms used, interface description and source code description. This application is used to study the route of packets in an IP network.

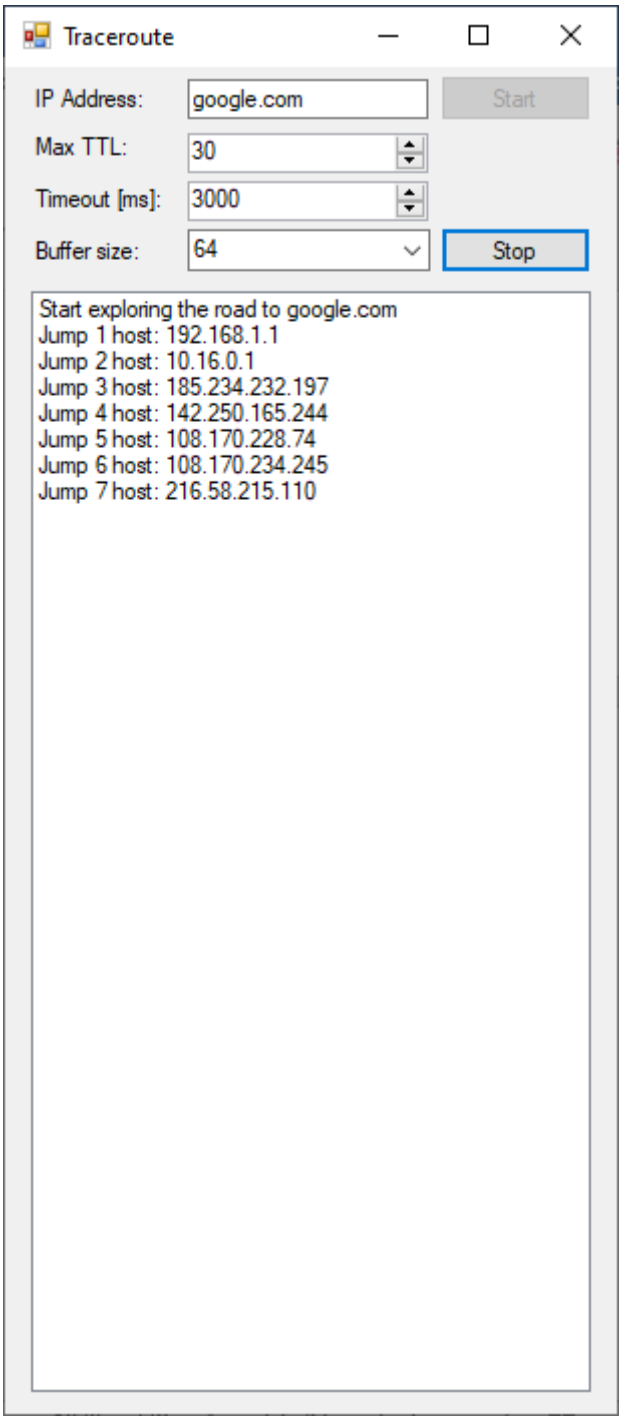
Describing of the application’s operation



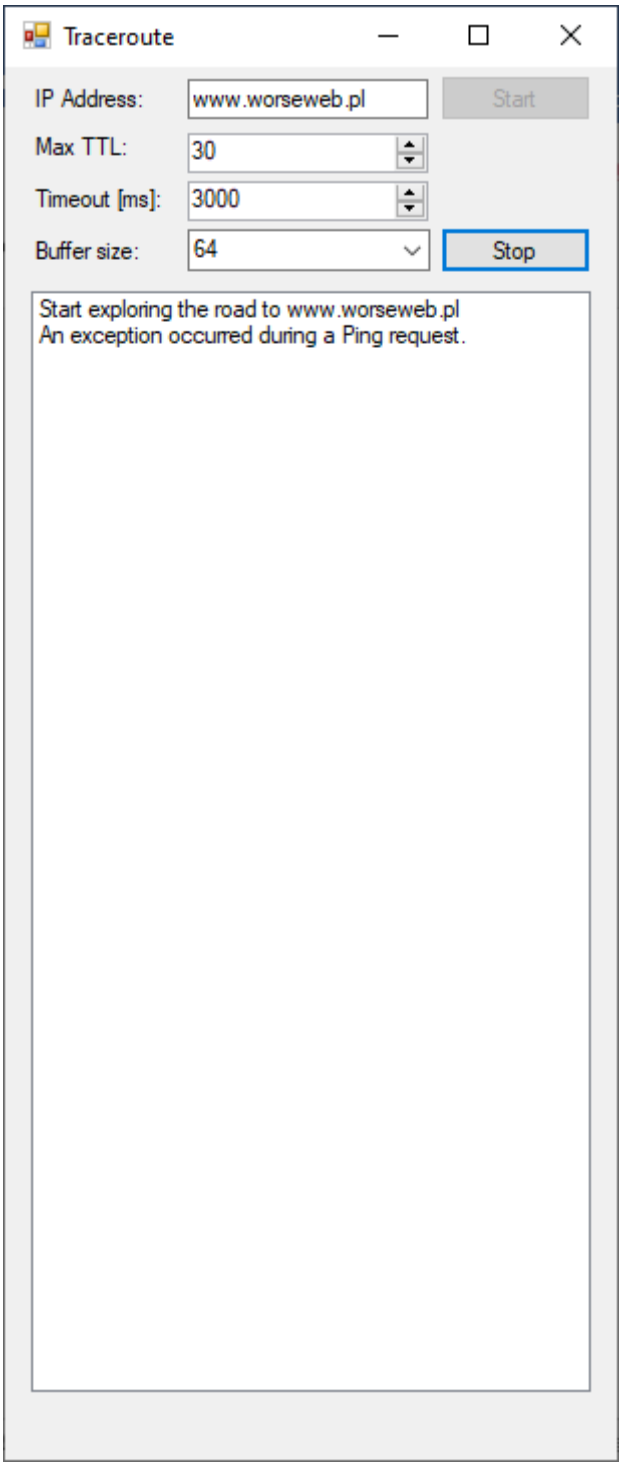
Drawing 1: The beginning of the application's operation [own study]



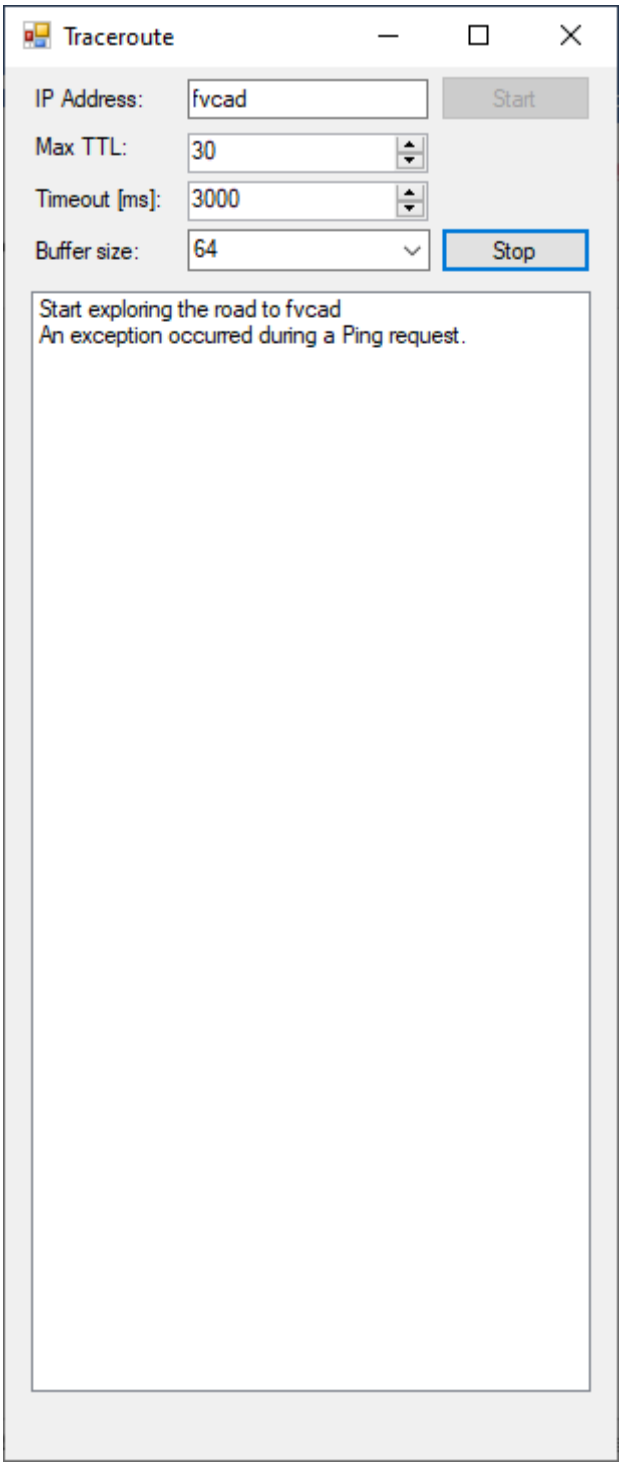
Drawing 2: The effect after entering the IPv4 address of the localhost [own study]



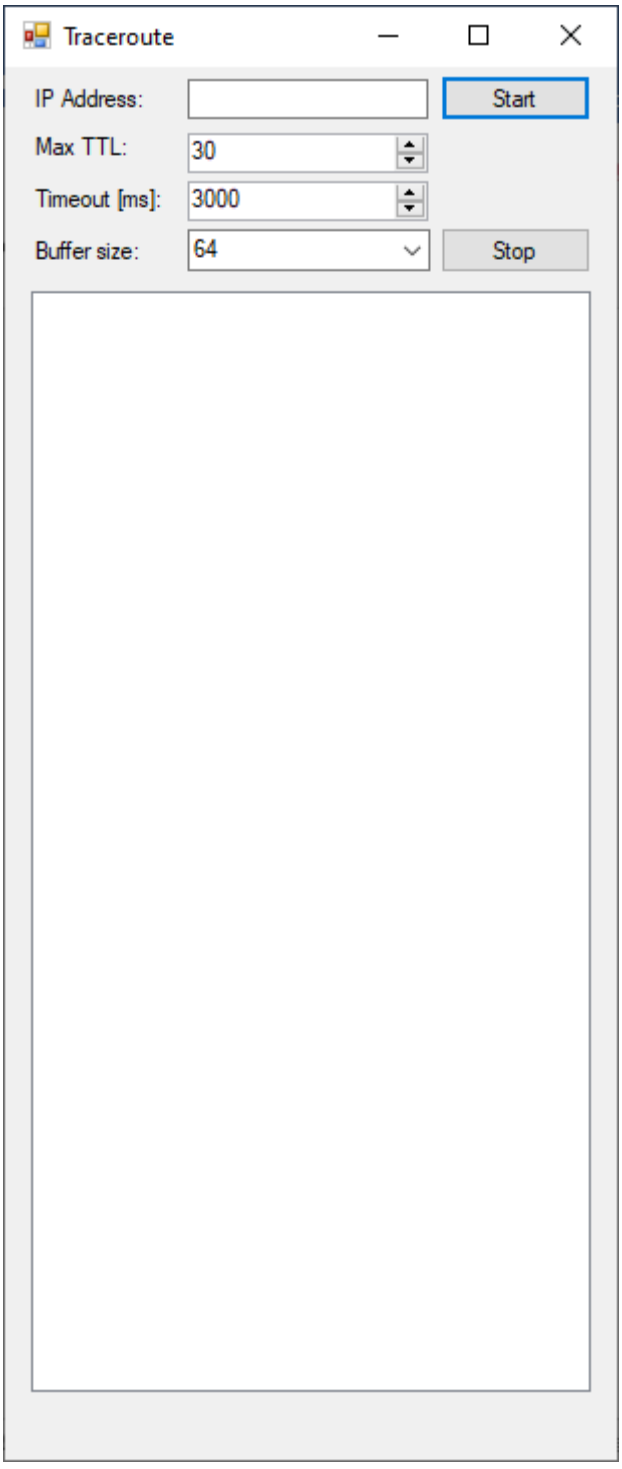
Drawing 3: The effect after entering the website [own study]



Drawing 4: Effect after entering a wrong address [own study]



Drawing 5: Effect after entering random characters [own study]



Drawing 6: Effect after typing nothing [own study]

After starting the program, the user enters the IP address or website into the text field. Can set the maximum number of packet hops on the way to the specified host, which is 30 by default (the maximum value can be 100), the maximum waiting time for a response [ms], which is 3000 by default (the maximum value), and also specify the size of the sent ICMP packet, the default is 64 (it can also be 32 or 128). Clicking the Start button starts the program procedure for examining the packet route in the IP network, based on the UDP and ICMP protocols. If

there is a network connection (when connecting to a host other than the local host) and the IP or WWW address is correct, you get feedback. Otherwise "An exception occured during a Ping request." it will be displayed in the case of an invalid address, random characters (not IP and WWW address) or no connection. It is possible that the user is not staring at anything and typing nothing, then the procedure will not start. There are certainly some bugs that the developer didn't discover while working on the app.

What is needed for use?

The application does not require installation. It only needs the Windows operating system.

Algorithm used

The basic form of the algorithm can be deduced from the previous section. It only needs the Windows operating system. In summary, the application is the command line equivalent of Tracert on Windows systems, and Traceroute on Unix systems.

The traceroute operation relies on the UDP and ICMP communication protocols.

- The packet is sent first with the TTL (Time To Live) field set to 1. This value is decreased as it traverses subsequent routers in the route. If the TTL field reaches the value 0, the packet is dropped by the router which then sends back an ICMP "Time Exceeded" message. This is how the source computer obtains the IP address of the first router in the route.

- In the next packet sent from the source computer, the TTL field is set to 2. The first router will reduce this value to 1 and pass it to the second router in the route. The second router will behave like the first one before, ie it will reduce TTL to 0 and discard the packet sending "Time Exceeded" message to the source computer.

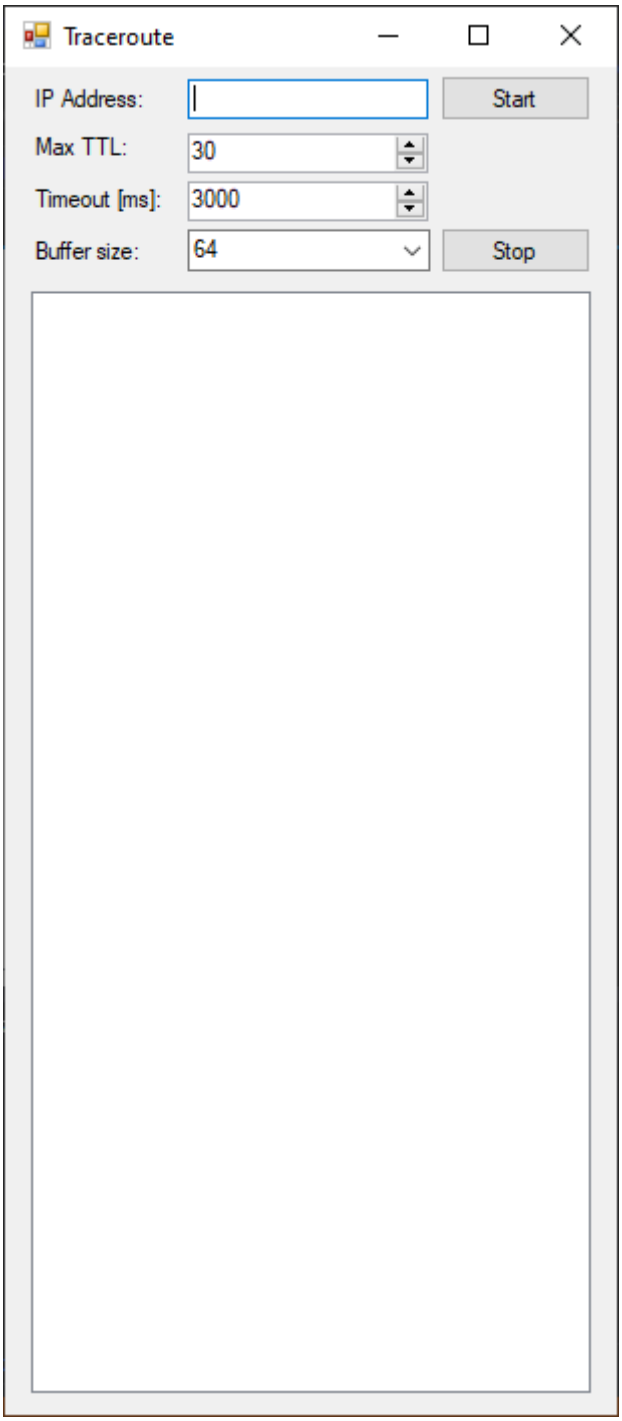
- Information about the next routers on the route is obtained in the same way as in the above two points. You just incrementally increment the TTL field value by 1 in the outgoing packets.

- If the packet eventually reaches the target host, an ICMP "Port Unreachable" message will most likely be returned. This is because Unix traceroute implementations are deliberately sending UDP packets with a port number greater than 30000. Typically, no services are running on a port with this high number, so no process on the target computer will try to handle the incoming packet. In this case, the route study is terminated.

Similar tools as mtr and the tracert program in the Microsoft Windows family are implemented slightly differently. The main principle of operation, consisting in gradually increasing the TTL field in the sent packets, remains the same. The difference is that the packets sent are not UDP datagrams, but ICMP "Echo Request" messages. If such a message

reaches its destination, an "Echo Reply" will always be returned. This saves you from having to rely on the assumption of high port numbers for UDP datagrams, and you can bypass some firewalls. [1]

Interface description



Drawing 7: Graphical interface [own study]

The interface is typical for a Windows Forms Application. There are essential components: listbox, textbox, labels, combobox, numericupdowns and two buttons.

Source code description

The project was made in the C# programming language, in the Visual Studio Community 2017 programming environment. All work was done on the Windows 10 operating system. The application's source code looks like this.

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;
using System.Net.NetworkInformation;

namespace TracerouteV2
{
    public partial class Form1 : Form
    {
        private Ping ping = new Ping();
        private byte[] bufor;
        private int i = 1;

        public Form1()
        {
            InitializeComponent();
            ping.PingCompleted += new PingCompletedEventHandler(Ping_PingCompleted);
        }

        private void BStart_Click(object sender, EventArgs e)
        {
            listBox1.Items.Clear();
            i = 1;
            PingOptions opcjePing = new PingOptions(i, true);
            string s = null;
            if (textBox1.Text != "" && textBox1.Text.Trim() != "")
            {
                int wielkoscBuforu = 32;
                try
                {
                    wielkoscBuforu = System.Convert.ToInt16(comboBox1.Text);
                }
                catch
                {
                    wielkoscBuforu = 32;
                    comboBox1.Text = "32";
                }
                for (int j = 0; j < wielkoscBuforu; j++)
                {
                    s += "a";
                }
                bufor = Encoding.ASCII.GetBytes(s);
                ping.SendAsync(textBox1.Text, (int)numericUpDown2.Value, bufor,
```

```
                opcjePing, null);
                listBox1.Items.Add("Start exploring the road to " +
                 textBox1.Text);
                BStart.Enabled = false;
            }
        }

        private void BStop_Click(object sender, EventArgs e)
        {
            ping.SendAsyncCancel();
            BStart.Enabled = true;
        }

        public void Ping_PingCompleted(object sender, PingCompletedEventArgs e)
        {
            if (e.Error != null)
            {
                listBox1.Items.Add(e.Error.Message);
                return;
            }
            if (e.Cancelled)
            {
                listBox1.Items.Add("The operation was aborted at the request of the
user.");
            }
            else
            {
                if (e.Reply.Status == IPStatus.TtlExpired)
                    listBox1.Items.Add("Jump " + i.ToString() + " host: " +
                    e.Reply.Address.ToString());
                if (e.Reply.Status == IPStatus.TimedOut)
                    listBox1.Items.Add("Jump " + i.ToString() + " host: * The request
timed out.");
                if (e.Reply.Status == IPStatus.Success)
                {
                    listBox1.Items.Add("Jump " + i.ToString() + " host: " +
e.Reply.Address.ToString());
                    return;
                }
                if (i++ < (int)numericUpDown1.Value)
                {
                    PingOptions opcjePing = new PingOptions(i, true);
                    ping.SendAsync(textBox1.Text, (int)numericUpDown2.Value, bufor,
                    opcjePing, null);
                }
                else
                {
                    listBox1.Items.Add("The maximum number of hops (parameter TTL =" +
numericUpDown1.Value.ToString() + ")");
                    BStart.Enabled = true;
                }
            }
        }
    }
                                            }
```

Listing 1: Source code [own study]

List of drawings

List of listings

Bibliography

[1] https://pl.wikipedia.org/wiki/Traceroute