**Министерство науки и высшего образования Российской Федерации**

**Федеральное государственное автономное образовательное**

**учреждение высшего образования**

**«Национальный исследовательский университет ИТМО»**

**Мегафакультет:** Компьютерных технологий и управления

**Факультет:** Безопасности информационных технологий

**Направление (специальность):** 10.03.01 «Информационная безопасность»

**Лабораторная работа №1**

**на тему**

**«Обработка и тарификация CDR (Call Detail Record)»**

Выполнила:

студентка группы N3353

Чубаркина И.Р.



Проверил:

Федоров Иван Романович

Санкт-Петербург

2020 г.

**Цель работы**

* данной работе предполагается обработка трафика NetFlow v5 из файла. Работа включает в себя 4 этапа:
  1. Приведение файла в читаемый вид
  2. Формирование собственного файла для последующей обработки
  3. Построение графика зависимости объема трафика от времени
  4. Тарификация пользователя исходя из варианта работы

**Задание**

Вариант №11.

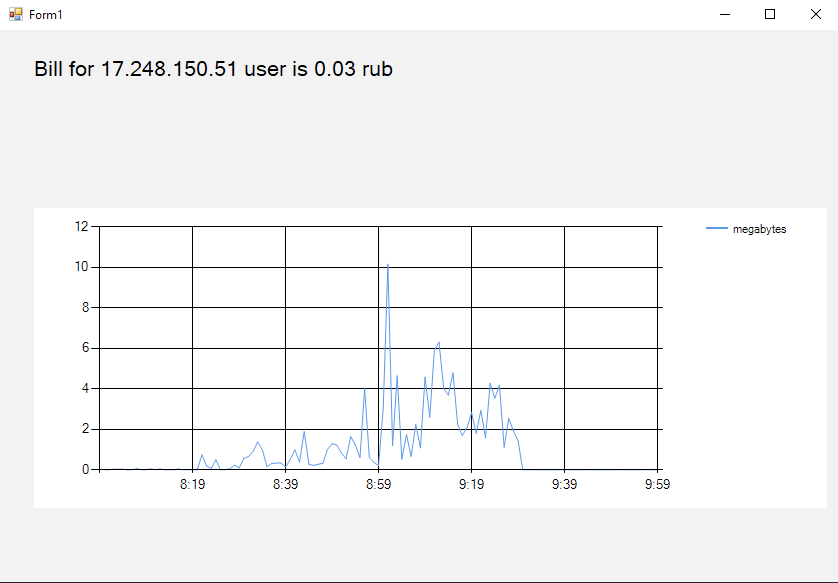
Протарифицировать абонента с IP-адресом 17.248.150.51 с коэффициентом k: 0.5 руб за каждый мегабайт.

**Описание выбранных средств реализации и обоснования выбора**

Мной был выбран .NET(С#) в силу того, что это популярная платформа-фреймворк с поддержкой практически любых утилит, которые могли бы потребоваться в ходе разработки, а также с многофункциональной и понятной средой разработки.

**Программа**

Программа использует IP-адрес, который необходимо протарифицировать в соответствии с вариантом задания.

Lab2.exe 

**Вывод**

* результате выполнения работы была реализована программа на .Net(C#), осуществляющая тарификацию по заданным правилам.

**Приложение**

Form1.cs

using CsvHelper;

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Globalization;

using System.IO;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace lab2

{

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

}

private void Form1\_Load(object sender, EventArgs e)

{

var targetIp = "17.248.150.51:443";

long totalBytes = 0;

List<TrafficRecord> records = new List<TrafficRecord>();

using (var reader = new StreamReader("traffic.csv"))

using (var csv = new CsvReader(reader, CultureInfo.InvariantCulture))

{

records = csv.GetRecords<TrafficRecord>().Where(el => el.SourceIp != "UDP").ToList();

var targetNumberRows = records.Where(el => el.SourceIp == targetIp || el.DestinationIp == targetIp).ToList();

totalBytes = targetNumberRows.Sum(el => el.InByte + el.OutByte);

}

var totalMb = (totalBytes / 1024f) / 1024f;

label1.Text = $"Bill for 17.248.150.51 user is {totalMb \* 0.5:F2} rub";

for (var hour = 0; hour < 2; hour++)

for (var minute = 0; minute < 60; minute++)

{

var currHour = hour > 0 ? 9 : 8;

var hourBytes = records.Where(el => el.Date.Hour == currHour && el.Date.Minute == minute).Sum(el => el.InByte + el.OutByte);

chart1.Series["megabytes"].Points.AddXY($"{currHour}:{minute}", (hourBytes / 1024f) / 1024f);

}

}

}

}