**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РФ**

Федеральное государственное автономное

образовательное учреждение высшего образования

«Национальный исследовательский университет ИТМО»

**ФАКУЛЬТЕТ БЕЗОПАСНОСТИ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ**

Лабораторная работа № 2  
по теме «Обработка и тарификация трафика NetFlow»

по дисциплине «Управление мобильными устройствами»

Работу выполнил

студент группы №3351

очного отделения

Меренков Даниил

Проверил

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Федоров Иван Романович

 Дата сдачи: 11.04.2020

Санкт-Петербург

2020



**Цель работы(задача):**

Цель данный работы – изучение протокола NetFlow, его выходных файлов данных, парсинг этих файлов, составление диаграммы зависимости трафика от времени, а так же тарификация абонента, в соответствии с заданными условиями.

**Описание выбранных средств реализации и обоснование выбора:**

Исходные данные: Протарифицировать абонента с IP-адресом 192.168.250.62 с коэффициентом k: 0,5руб/Мб первые 100Мб, далее 1руб/Мб

Для реализации программного модуля в данной лабораторной работе мною был выбран язык Python3 из-за его легкости чтения, гибкости и практичности в задачах, касающихся парсинга.

При реализации данной программы была задействована библиотека matplotlib для рисования графика по переданным параметрам и сохранения его в файл .png, а так же утилита nfdump для составления нормализованного вида исходных данных.

Пример использования утилиты nfdump.



Где ключ -о позволяет задать параметр вывода в нужном формате fmt

**%sap** Source Address:Port

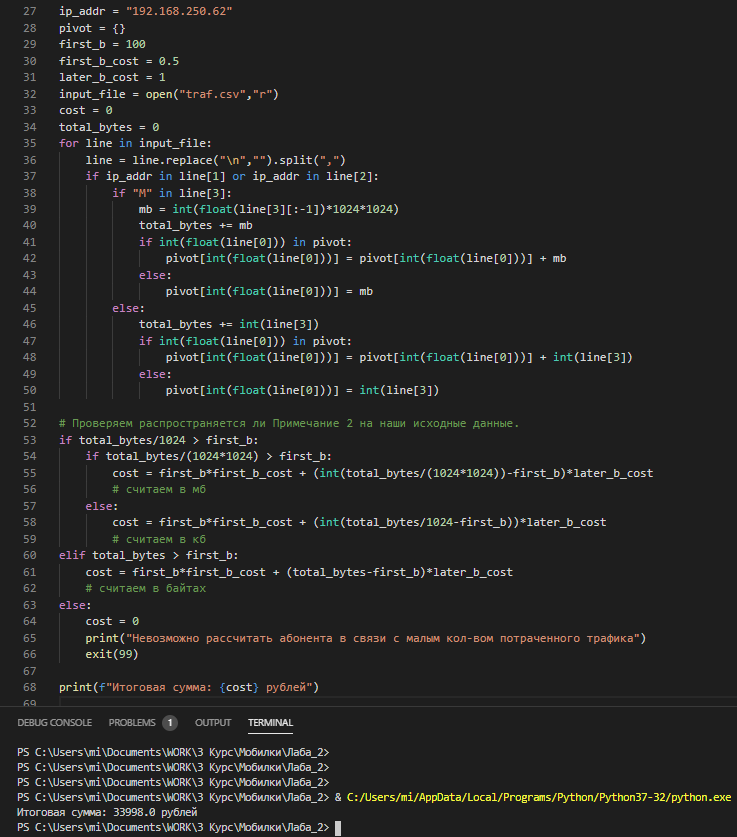
**%dap** Destination Address:Port

**%byt** Bytes - default input

**%tsr** Start Time, but in fractional seconds since the epoch (1970-01-01)

На выход получаем csv файл с исключительно необходимой информацией – time,srcIP,dstIP и суммарное кол-во байт при данном обращении.

Вывод программы:

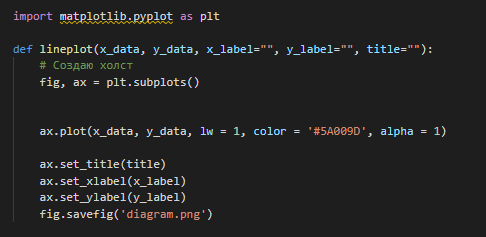


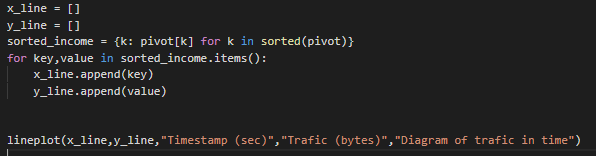
Согласно Примечанию 2, так как в наших исходных данных кол-во трафика абонента слишком мало, мы уменьшаем единицу учета до байт -> исходное задание меняется на

«Протарифицировать абонента с IP-адресом 192.168.250.62 с коэффициентом k: 0,5руб/байт первые 100 байт, далее 1руб/байт».

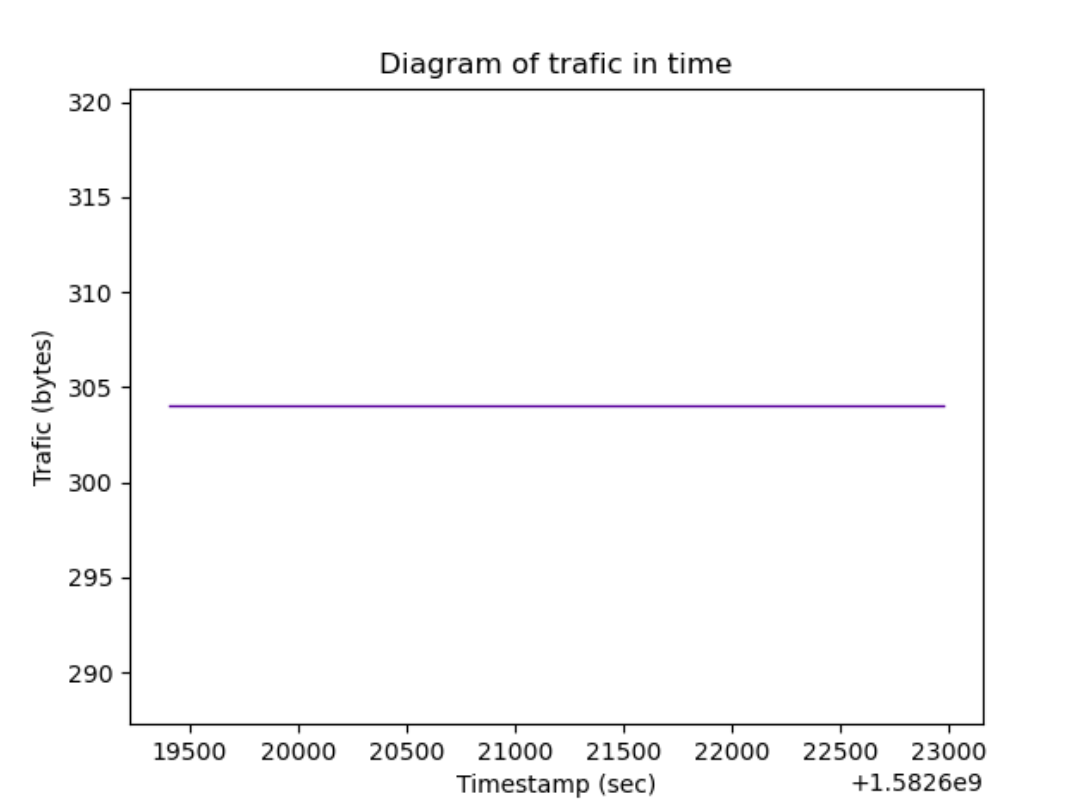
Ответом на данные условия является: 33998 рублей

Так же, собираем статистику о зависимости трафика от времени для построения графика и визуализации данных. Для этого суммируем общий объем трафика для нужного абонента на каждую зарегистрированную секунду, согласно UNIX-времени(%tsr) и с помощью библиотеки matplotlib строим график, где по оси x – время , а по оси y – трафик. Сохраняем данную статистику в виде графика в файле.





В итоге, получаю следующую зависимость для моих исходных данных:



**Исходный код:**

import matplotlib.pyplot as plt

import os

def lineplot(x\_data, y\_data, x\_label="", y\_label="", title=""):

    # Создаю холст

    fig, ax = plt.subplots()

    ax.plot(x\_data, y\_data, lw = 1, color = '#5A009D', alpha = 1)

    ax.set\_title(title)

    ax.set\_xlabel(x\_label)

    ax.set\_ylabel(y\_label)

    fig.savefig('diagram.png')

def main\_2():

    ip\_addr = "192.168.250.62"

    pivot = {}

    first\_b = 100

    first\_b\_cost = 0.5

    later\_b\_cost = 1

    input\_file = open("traf.csv","r")

    cost = 0

    total\_bytes = 0

    for line in input\_file:

        line = line.replace("\n","").split(",")

        if ip\_addr in line[1] or ip\_addr in line[2]:

            if "M" in line[3]:

                mb = int(float(line[3][:-1])\*1024\*1024)

                total\_bytes += mb

                if int(float(line[0])) in pivot:

                    pivot[int(float(line[0]))] = pivot[int(float(line[0]))] + mb

                else:

                    pivot[int(float(line[0]))] = mb

            else:

                total\_bytes += int(line[3])

                if int(float(line[0])) in pivot:

                    pivot[int(float(line[0]))] = pivot[int(float(line[0]))] + int(line[3])

                else:

                    pivot[int(float(line[0]))] = int(line[3])

    # Проверяем распространяется ли Примечание 2 на наши исходные данные.

    if total\_bytes/1024 > first\_b:

        if total\_bytes/(1024\*1024) > first\_b:

            cost = first\_b\*first\_b\_cost + (int(total\_bytes/(1024\*1024))-first\_b)\*later\_b\_cost

            # считаем в мб

        else:

            cost = first\_b\*first\_b\_cost + (int(total\_bytes/1024-first\_b))\*later\_b\_cost

            # считаем в кб

    elif total\_bytes > first\_b:

        cost = first\_b\*first\_b\_cost + (total\_bytes-first\_b)\*later\_b\_cost

        # считаем в байтах

    else:

        cost = 0

        print("Невозможно рассчитать абонента в связи с малым кол-вом потраченного трафика")

        exit(99)

    x\_line = []

    y\_line = []

    sorted\_income = {k: pivot[k] for k in sorted(pivot)}

    for key,value in sorted\_income.items():

        x\_line.append(key)

        y\_line.append(value)

    lineplot(x\_line,y\_line,"Timestamp (sec)","Trafic (bytes)","Diagram of trafic in time")

    return cost

if \_\_name\_\_ == '\_\_main\_\_':

    os.chdir('/home/Mobile/Lab\_2')

    cost = main\_2()

    print(f"Итоговая стоимость: {cost} рублей")

    print("Диаграмма сформированна, она находится в директории Lab\_2 под именем diagram.png")

**Выводы:**

В рамках данной лабораторной работы, я научился обрабатывать файлы, порождаемые протоколом NetFlow, научился работать с библиотекой matplotlib и генерировать с ее помощью графики зависимостей с выводом в заданный файл. Более того, мной были закреплены такие практические навыки, как парсинг файлов, тарификация заданного абонента и создание диаграмм для данных этого абонента.