**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 №2**

**на тему**

**«Обработка и тарификация трафика NetFlow»**

**Вариант №3**

Выполнил:

студент группы N3353

Вишняков М. Д.

Проверил:

Федоров Иван Романович

Санкт-Петербург

2020 г.

**Цели работы:**

1. Привести данный файл в читабельный вид, сформировать собственный файл для тарификации любого формата, с которым удобно работать (в соответствии с вариантом работы)
2. Построить график зависимости объема трафика от времени (любым удобным образом)
3. Протарифицировать абонента с IP-адресом 192.168.250.27 с коэффициентом k: 1руб/Мб

**Описание выбранных средств реализации и обоснования выбора**

Python - высокоуровневый язык программирования общего назначения, ориентированный на повышение производительности разработчика и читаемости кода. Стандартная библиотека включает большой объём полезных функций.

Причины, по которым выбран Python:

* кроссплатформенность
* обширная стандартная библиотека
* простота написания кода

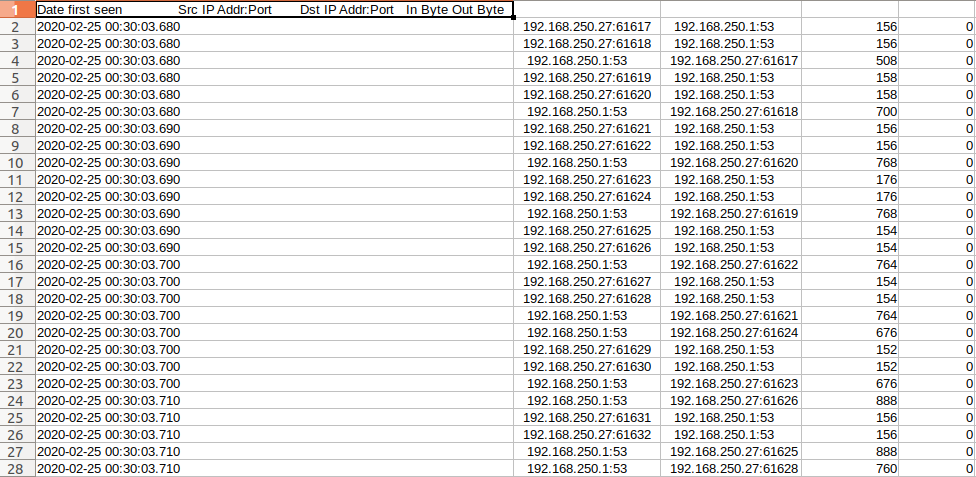
Microsoft Excel для построения графика.

**Выполнение работы**

1. Выгружаю данные при помощи утилиты nfdump:



traffic.csv



1. Построю график зависимости объема трафика от времени для абонента по выгруженным данным:
2. Протарифицирую абонента с IP-адресом 192.168.250.27 при помощи написанной программы.

IP-адрес абонента передаётся как аргумент командной строки.



Таким образом, за заданный промежуток времени абонент должен 3 рубля 25 копеек.

**Вывод:** в ходе выполнения работы был построен график зависимости объема трафика от времени и написана программа для тарификации абонента.

**Листинг**

**lab2.py**

import csv

import sys

def calculateCost(stats,ip):

    traffic = 0

    for item in stats:

        if len(item) == 5:

            if item[1].find(ip) != -1:

                trafficString = item[3]

            elif item[2].find(ip) != -1:

                trafficString = item[4]

            if trafficString.find('M') == -1:

                traffic += int(trafficString)

            else:

                traffic += float(piceOfTraffic[:piceOfTraffic.find('M')]) \* 1024 \* 1024

    return '%.2f' % (traffic / 1024 / 1024)

def main():

    with open('traffic.csv','r') as csvTrafic:

        cost = calculateCost(csv.reader(csvTrafic),sys.argv[1])

        print(cost)

if \_\_name\_\_ == '\_\_main\_\_':

    main()