**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ**

**«Национальный исследовательский университет ИТМО»**

**Факультет безопасности информационных технологий**

**Дисциплина:**

**«Управление мобильными устройствами»**

**Лабораторная работа № 2 на тему**

**«Обработка и тарификация трафика NetFlow»**

**Выполнил:**

Студент гр. N3349

Шишков В. Е.

**Проверил:**

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

Федоров И. Р.

Санкт-Петербург

2020г.

**Цель работы:** создать программное обеспечение для автоматизации процесса тарификации услуг «Интернет» путем обработки файла с данными статистики NetFlow версии 5.

**Теория:**

NetFlow — это протокол, разработанный компанией Cisco и предназначенный для сбора информации об IP-трафике внутри сети. Маршрутизаторы Cisco анализируют проходящий через интерфейс трафик, суммируют данные и отправляют статистику в формате NetFlow на специальный узел, называемый NetFlow Сollector. NetFlow часто используется для ведения биллинга или для анализа трафика сети. Протокол существует в нескольких версиях, последняя версия 9 предназначена для учёта трафика между АС (Автономная Система) и в импортируемых данных имеет несколько дополнительных полей таких как АС источника, АС назначения и пр., но обычно, для биллинга в несложной сети внутри одной АС достаточно информации, содержащейся в данных NetFlow версии 5.

В данной работе предполагается обработка трафика NetFlow v5 из файла nfcapd.202002251200

Правила тарификации услуг “Интернет”:

X = Q \* k, где

• X – итоговая стоимость,

• Q – общий объем трафика NetFlow за отчетный период,

• k – множитель тарифного плана (у каждого варианта свой).

**Задание (5 вариант):** Протарифицировать абонента с IP-адресом 192.168.250.59 с коэффициентом k: 1 руб / Мб, первая 1000 Кб бесплатно.

**Исходный файл**: «nfcapd.202002251200».

**Ход работы**:

Для обработки исходного файла используется команда   
«nfdump -r nfcapd.202002251200 -o csv> dump.csv», сохраняющая csv вид данного файла в файл «dump.csv».

Для написания ПО для тарификации был выбран язык программирования python (версия 3.6.9). Используемые библиотеки:csv, matplotlib, datetime, itertools.

Основой выбора данного языка программирования является: простота понимания программного кода и мгновенная коррекция в случае изменения условий тарификации; наличие встроенных библиотек для обработки файлов csv формата и построения графиков.

**Входные данные ПО**: «dump.csv»

**Выходные данные ПО**: список услуг, за которые взымается плата с абонента, итоговая сумма за все услуги и график зависимости объема трафика от времени (сохраняется в виде картинки "plot.png»).

**Исходный код:**

#! /usr/bin/env python3

import csv

import matplotlib

import matplotlib.pyplot as plt

import datetime

import itertools

def tariff(data, ip\_addr, traffic\_cost, free\_traffic):

total\_bytes = 0

stats = []

for row in data:

if row['sa'] == ip\_addr or row['da'] == ip\_addr:

# Обработка входящих и исходящих пакетов

bytes\_len = int(row['ibyt'])

total\_bytes += bytes\_len

stats.append(('Incoming' if row['sa'] == ip\_addr else 'Outcoming') + \

f' traffic, bytes length = {bytes\_len} bytes')

total\_mb = total\_bytes / 2 \*\* 20

total = (total\_mb - free\_traffic) \* traffic\_cost if total\_mb > free\_traffic else 0

return stats, total

def make\_plot(data, ip\_addr, filename):

dates\_lens = []

for row in data:

if row['sa'] == ip\_addr or row['da'] == ip\_addr:

dates\_lens.append((datetime.datetime.strptime(row['ts'], "%Y-%m-%d %H:%M:%S"), int(row['ibyt']) / 1024))

# group traffic by minutes

def group\_filter(x):

return datetime.datetime.strptime(x[0].strftime('%Y-%m-%d %H:%M:0'), '%Y-%m-%d %H:%M:%S')

dates\_lens = itertools.groupby(sorted(dates\_lens, key=group\_filter), group\_filter)

dates\_lens = [(key, sum([x[1] for x in group])) for key, group in dates\_lens]

dates = [x[0] for x in dates\_lens]

kb\_len = [x[1] for x in dates\_lens]

dates = matplotlib.dates.date2num(dates)

ax = plt.subplot(111)

ax.plot(dates, kb\_len)

ax.xaxis\_date()

plt.xlabel('Time')

xformatter = matplotlib.dates.DateFormatter('%H:%M')

plt.gcf().axes[0].xaxis.set\_major\_formatter(xformatter)

plt.ylabel('Bytes transferred (kB)')

plt.savefig(filename)

print(f'plot saved as {filename}')

if \_\_name\_\_ == '\_\_main\_\_':

with open('dump.csv', 'r') as csvfile:

file = list(csv.DictReader(csvfile))

result = tariff(file, '192.168.250.59', 1, 1)

for row in result[0]:

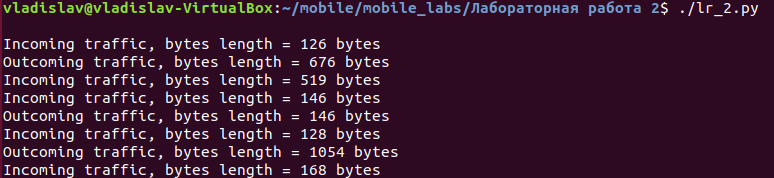
print(row)

print(f'total cost is {result[1]} rubles')

make\_plot(file, '192.168.250.59', 'plot.png')

**Вывод программы**:

Первые и последние строки вывода программы:



…

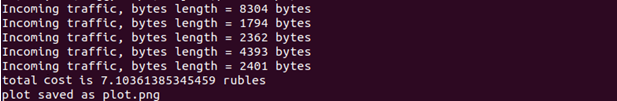
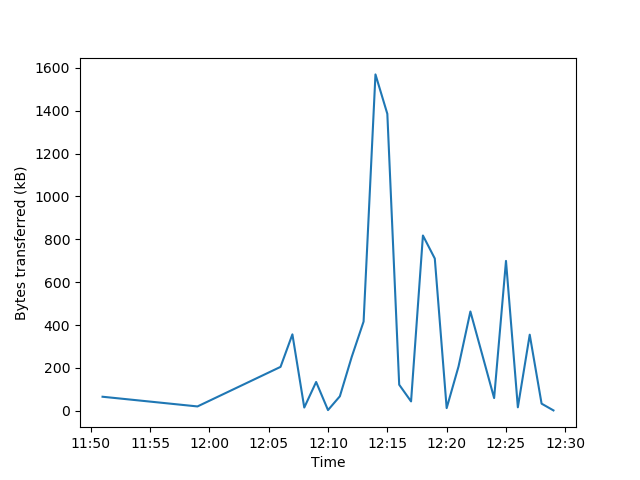


График (файл «plot.png»):



**Вывод**: в результате работы была написана программа, позволяющая протарифицировать абонента и подсчитать итоговую стоимость использования услуги «Интернет» с учётом определённого тарифа, также позволяющую пострить график зависимости объема трафика от времени.