**Министерство науки и высшего образования РФ**

Федеральное государственное автономное

образовательное учреждение высшего образования

«Национальный исследовательский университет ИТМО»

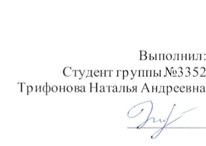
Факультет безопасности информационных технологий

Управление мобильными устройствами

Отчет по лабораторной работе №2

«Обработка и тарификация трафика NetFlow»

Вариант №5



Дата сдачи: 12.05.2020

Проверил:

Федоров Иван Романович

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

Санкт-Петербург 2020 г.

**Цель работы:**

Научиться обрабатывать трафик NetFlow и реализовать тарификацию услуг для конкретного абонента.

**Описание работы:**

Для программной реализации данной задачи мною был выбран язык программирования «python3» по следующим причинам: я имею достаточный опыт работы на нем; данный язык довольно удобен для реализации подобных задач.

Данная работа включает в себя следующие этапы:

1. Привести данный файл в читабельный вид (проще всего это сделать с помощью утилиты nfdump)   
   nfdump -r nfcapd.202002251200
2. Сформировать собственный файл для тарификации любого формата, с которым удобно работать (в соответствии с вариантом работы)
3. Построить график зависимости объема трафика от времени (любым удобным образом)
4. Протарифицировать трафик в соответствии с вариантом задания

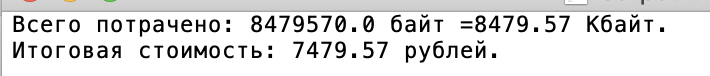
Правила тарификации услуг “Интернет”:  
                 X = Q \* k,   
 где X - итоговая стоимость, Q - общий объем трафика NetFlow за отчетный период, k - множитель тарифного плана.

Моему варианту (№5) соответствует следующее задание:

Протарифицировать абонента с IP-адресом 192.168.250.59   
с коэффициентом k: 1руб/Мб, первая 1000Мб бесплатно.

Общий объем трафика по абоненту оказался меньше заявленного в варианте работы, поэтому я уменьшила единицу учета (k = 1 руб/Кб, первые 1000 Кб бесплатно).

**Результат работы:**



**Изображение выглядит как снимок экрана

Автоматически созданное описание**

**Листинг программы:**

import subprocess

import matplotlib.pyplot as plt

file = open("file.txt", "w")

data = subprocess.run(["nfdump", "-o", "fmt:%te|%sa|%da|%byt", "-r", "nfcapd.202002251200"], stdout=subprocess.PIPE, text=True)

file.write(data.stdout)

file.close()

file = open("file.txt", "r")

outputfile = open("output.txt", "w")

def parsing(file):

'''Представим нашу таблицу в виде словаря списков.'''

mydict = {}

mydict["date\_first\_seen"] = []

mydict["src\_add"] = []

mydict["dst\_add"] = []

mydict["bytes"] = []

line = file.readline() #(просто считываем первую строку, она нам не нужна)

#Столбцы таблицы представим в виде списков - значений для каждого ключа нашего словаря:

word = ""

key\_number = 1

for char in file.read():

if (char == "|") or (char == "\n"):

if key\_number == 1:

mydict["date\_first\_seen"].append(word)

elif key\_number == 2:

word = word.replace(' ', '')

mydict["src\_add"].append(word)

elif key\_number == 3:

word = word.replace(' ', '')

mydict["dst\_add"].append(word)

elif key\_number == 4:

word = word.replace(' ', '')

#если встречается запись "... M", то переводим в байты:

if word[-1] == "M":

word = word[:-1]

word = str(float(word)\*1000000)

mydict["bytes"].append(word)

word = ""

if key\_number < 4:

key\_number += 1

else:

key\_number = 1

else:

word += char

mydict["date\_first\_seen"].pop()

mydict["src\_add"].pop()

mydict["dst\_add"].pop()

mydict["bytes"].pop()

return mydict

def tariffication(mydict, ip, k):

'''Тарификация услуг и сохранение нужных данных.'''

cost = 0

x = [] #абсцисса для графика

y = [] #ордината графика

num = 0 #счетчик

cur\_time = 0

for i in mydict["src\_add"]:

if (i == ip) or (mydict["dst\_add"][num] == ip):

cost += float(mydict["bytes"][num]) \* k/1000

if mydict["date\_first\_seen"][num] != cur\_time:

x.append(mydict["date\_first\_seen"][num])

y.append(int(float(mydict["bytes"][num])))

cur\_time = mydict["date\_first\_seen"][num]

else:

y[-1] += int(float(mydict["bytes"][num]))

num += 1

answer = "Всего потрачено: " + str(round(cost\*1000/k, 2)) + " байт =" + str(round(cost, 2)) + " Кбайт.\n"

return cost - free\_cost, x, y, answer

def graph(x, y):

'''Функция создает график.'''

plt.plot(x, y, lw=1)

plt.ylabel('Объем трафика')

plt.xlabel('Время')

plt.title('Зависимость объема трафика от времени')

plt.show()

#Множитель тарифного плана:

k = 1 #руб/Кб

#Количество бесплатных Кбайт:

free\_cost = 1000

#IP-адрес абонента:

ip = "192.168.250.59"

mydict = parsing(file)

cost, x, y, answer = tariffication(mydict, ip, k)

answer += "Итоговая стоимость: " + str(round(cost, 2)) + " рублей."

outputfile.write(answer)

graph(x, y)

file.close()

outputfile.close()

**Вывод:**

В ходе выполнения данной работы я ознакомилась с трафиком NetFlow, научилась его обрабатывать, а также получила интересный опыт программной реализации правила тарификации.