## «Обработка и тарификация трафика NetFlow»

**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

Данный файл был сформирован при помощи утилиты nfcapd, установленной на коллекторе.

В рамках работы требуется:

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 (у каждого варианта свой).

В качестве результата работы необходимо представить программный модуль для обработки, просмотра статистики (график) и тарификации трафика NetFlow. Средства реализации выбираются студентом самостоятельно.

Вариант работы:

5. Протарифицировать абонента с IP-адресом 192.168.250.59   
с коэффициентом k:

* 1руб/Мб
* первые 1000Мб бесплатно

Обоснование выбранного средства реализации:

В качестве средства реализации мной был выбран язык программирования Python. Передо мной стоял выбор между Python и С. Не смотря на надежность языка программирования си, он не обладает такой обширной библиотекой готовых инструментов, как Python. Требования к работе и ее объем позволили мне использовать Python.

Ход работы

**Исходный код:**

import csv

import matplotlib.pyplot as plt

total\_traffic = 0

total\_occurences = 0

i = 0

filename = "D:\DEV\mobile\lab2\data.csv"

#file = input("Enter path to CDR file: ")

#number = input("Enter IP address: ")

number = "192.168.250.59"

#first 1000#b for free

tariff = 1

# initializing the titles and rows

fields = []

rows = []

rowlist = []

traflist = []

# reading csv file

with open(filename, 'r') as csvfile:

# creating a csv reader object

csvreader = csv.reader(csvfile)

# extracting field names through first row

fields = next(csvreader)

print("fields:", fields)

# extracting each data row one by one

for row in csvreader:

rows.append(row)

# get total number of rows

#print("Total no. of lines in a CDR file: %d"%(csvreader.line\_num))

#

# calculating incoming calls tariffication

for row in rows[1:17450]:

if row[3] == number:

total\_traffic = total\_traffic + int(row[12])

total\_occurences = total\_occurences + 1

#print(int(row[12]))

traflist += {(row[1], row[12])}

if row[4] == number:

total\_traffic = total\_traffic + int(row[12])

total\_occurences = total\_occurences + 1

#print(int(row[12]))

traflist += {(row[1], row[12])}

c = ""

# total price

if total\_traffic < 1000:

c = "bytes"

total\_cost = (total\_traffic\*1000-1000)\*1/1000000000

if total\_cost < 0:

total\_cost = 0

if 1000000 > total\_traffic > 1000:

total\_traffic = total\_traffic/1000

c = "kilobytes"

total\_cost = (total\_traffic\*1000-1000)\*1/1000000

if total\_cost < 0:

total\_cost = 0

if 1000000000 > total\_traffic > 1000000:

total\_traffic = total\_traffic/1000000

c = "megabytes"

total\_cost = (total\_traffic-1)\*1

if total\_cost < 0:

total\_cost = 0

print("total traffic:", total\_traffic, c)

#print("total occurences:", total\_occurences)

print("total cost: ", round(total\_cost, 2), "rubles")

#print(traflist)

with open('out.csv', 'w', newline='') as resultFile:

wr = csv.writer(resultFile, dialect='excel')

for traflist in traflist:

wr.writerow(traflist)

resultFile.close

############

### plot ###

############

i = 0

globalrow = ""

globaltraf = 0

rows = []

temptraf = 0

newdate = []

newtraf = []

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

csvreader = csv.reader(csvfile)

for row in csvreader:

rows.append(row)

csvfile.close

print(rows)

for row in rows:

if row[0] == globalrow:

#print("Plus", row[1])

temptraf = temptraf + int(row[1])

else:

if temptraf != 0:

newtraf += {temptraf}

#print("total: ", temptraf)

#print("new iteration: ", row[0])

temptraf = int(row[1])

newdate += {row[0]}

globalrow = row[0]

#print(globalrow)

newtraf += {int(row[1])}

#print(len(newdate), newdate)

#print(len(newtraf), newtraf)

x = newdate

y = newtraf

plt.plot(x,y, marker='o')

plt.title('Data from the CSV File: Traffic during the time period')

plt.xlabel('time')

plt.ylabel('traffic, bytes')

plt.show()

Результаты работы:

На рисунке 1 представлен результат выполнения программного модуля:

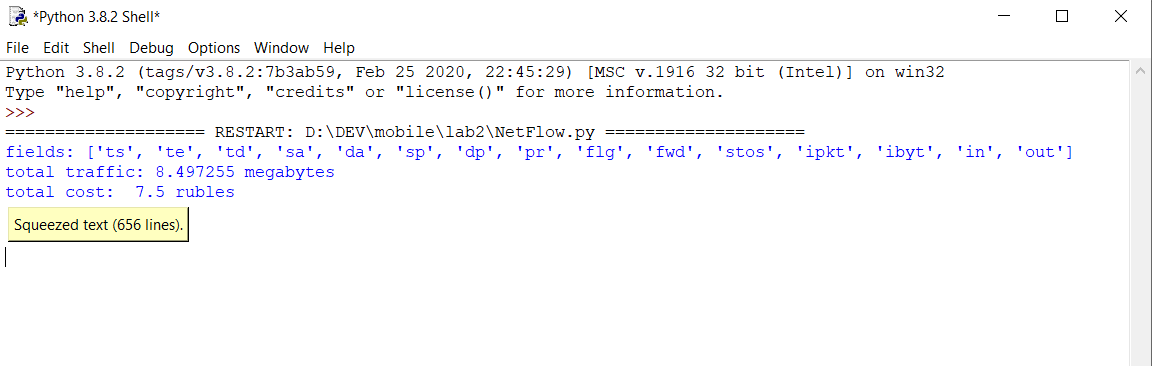


Рисунок 1 – результат выполнения программного модуля

На рисунке 2 представлен график, создаваемый программой по исходным данным:

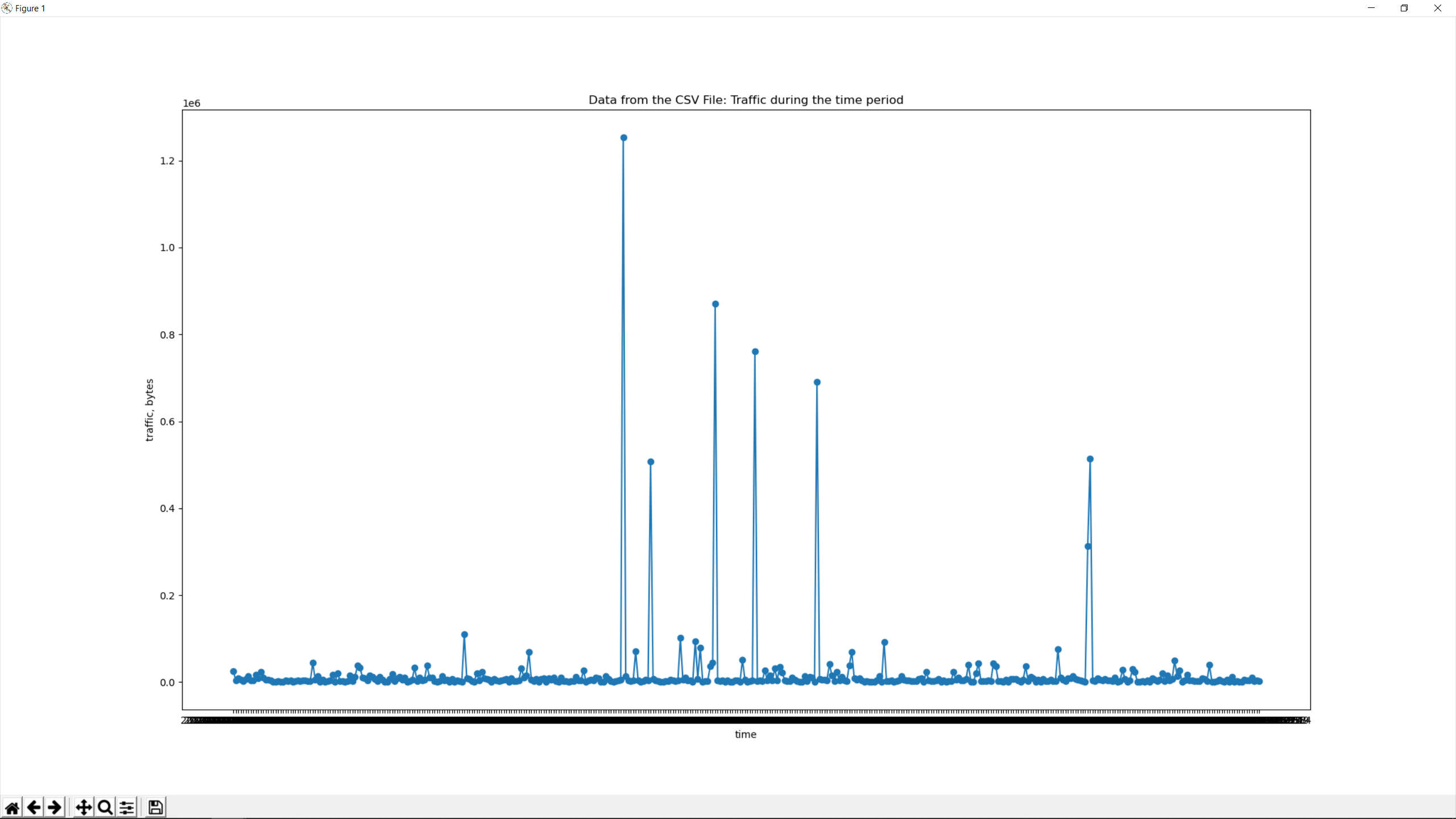


Рисунок 2 – График зависимости объема трафика от времени

Для тарификации любого другого адреса необходимо изменить значение переменной «number» в исходном коде программы.

Выводы:

Цель работы достигнута, получены знания о тарификации услуг типа «Интернет». Создан программный модуль обработки исходных данных и тарификации абонента.