

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 2

Работу выполнил
студент группы N3351
Сысуева Валерия



Проверил: Федоров И.Р.

Санкт-Петербург 2020

Цель работы: написать программный модуль для обработки и тарификации трафика, полученного в формате NetFlow.

Задачи в рамках лабораторной работы:

- а) сформировать собственный файл для тарификации,
- б) построить график зависимости объема трафика от времени,
- в) протарифицировать трафик в соответствии с вариантом задания

Исходные данные по варианту:

- а) IP-address: 217.15.20.194,
- б) коэффициент для тарификации $k = 1$ руб/Мб,
- в) количество бесплатных Мб – первые 1000 Мб

Теоретическая часть

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 Collector**. 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.

Правила тарификации услуг “Интернет”:

$X = Q * k$, где:

X – итоговая стоимость всех звонков абонента,

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.

В случае, если общий объем трафика по абоненту, меньше заявленного во варианте работы (например, требуется тарификация с увеличенным коэффициентом после 1000Мб, а у абонента всего 100), то тогда уменьшается единица учета (т.е. вместо Мб считаются Кб и т д).

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

Практическая часть

Для программной реализации был выбран язык программирования Python 3.7. Также, для работы с файлом, в котором содержатся все данные для тарификации и выполнения лабораторной работы, была использована утилита nfdump. Данные записали в файл traffic.csv с помощью команды:

nfdump -r nfcapd.202002251200 -o csv > traffic.csv

Данные по тарификации были записаны в settings.py, где 'price' – коэффициент для тарификации, а 'free' – количество бесплатного трафика в байтах.

```
settings.py > ...  
1 settings = {'price':1, 'free':1024000}
```

Рисунок 1 Содержание settings.py

```
1 from settings import settings  
2 input_filename = "traffic.csv"  
3 input_file = open(input_filename, "r")  
4 data = []  
5 IP = ''  
6 free = settings["free"]  
7 k = settings["price"]  
8 traffic, bill = 0, 0  
9  
10 with open(input_filename, "r") as file:  
11     for line in file:  
12         data.append(line.split(","))  
13 IP = '217.15.20.194'  
14 #IP = input('введите IP-адрес для тарификации: ') #217.15.20.194 по варианту  
15 for i in range(len(data)-3):  
16     if data[i][3] == IP or data[i][4] == IP:  
17         traffic += int(data[i][12])  
18         traffic += int(data[i][14])  
19 if traffic > free:  
20     traffic -= free  
21     traffic /= 1024*1024  
22     bill = traffic * k  
23 else:  
24     bill = 0  
25 print('общая стоимость:', bill)
```

Рисунок 2 Программный код

Также был разработан программный модуль, позволяющий построить график зависимости объема трафика от времени.

```
1  import sys
2  import matplotlib
3  import matplotlib.pyplot as plot
4  import matplotlib.dates as dates
5  import datetime
6  import matplotlib.patches as patches
7
8  def lineplot(x_data, y_data, x_label="", y_label="", title=""):
9
10     fig, ax = plot.subplots()
11
12
13     ax.plot(x_data, y_data, lw = 1, color = '#5A009D', alpha = 1)
14
15     ax.set_title(title)
16     ax.set_xlabel(x_label)
17     ax.set_ylabel(y_label)
18     fig.savefig('traffic_diagram.pdf')
19
20 def draw_graphic(data, IP):
21     lrg = 23
22     medium = 17
23     small = 13
24
25     params = {
26         "figure.titlesize": medium,
27         "figure.figsize": (16,10),
28         "axes.labelsize": medium,
29         "axes.titlesize":medium,
30         "legend.fontsize":medium,
31         "xtick.labelsize":small,
32         "ytick.labelsize":medium
33     }
```

Рисунок 3 Программный код для графика (часть 1)

```
35     plot.rcParams.update(params)
36     coef = 0.0009765625
37     # x = [datetime.datetime.strptime(s[0], "%Y-%m-%d %H:%M:%S") for s in data[1:len(data)-3]]
38     x = []
39     y = []
40     datamas = {}
41
42     for i in range(1, len(data)-3):
43         if data[i][3] == IP or data[i][4] == IP:
44             delta = int(data[i][12])
45             time = data[i][0]
46             if time in datamas:
47                 prev_t = datamas[time]
48                 datamas[time] = prev_t + delta
49             else:
50                 datamas[time] = delta
51
52     print(data[15904][0])
53     datamas_ks = list(datamas.keys())
54     datamas_ks.sort()
55
56     # print(len(datamas))
57     # print(datamas)
58
59     prev_i = 0
60     prev_prev_i = 0
61     traffic = 0
62     for i in datamas_ks:
63         traffic += datamas[i]
```

Рисунок 4 Программный код для графика (часть 2)

```

65     prev_prev_i = prev_i
66     prev_i = i
67     print(f"{i}:{traffic}")
68     x.append(datetime.datetime.strptime(i,"%Y-%m-%d %H:%M:%S"))
69     y.append(traffic)
70
71     lineplot(x,y,"Timestamp (sec)","Trafic (bytes)","Diagram of trafic in time")
72
73     datenums = dates.date2num(x)
74
75     plot.figure(figsize=(16,10), dpi= 80, facecolor='w', edgecolor='k')
76     plot.xticks(rotation = 14)
77     ax = plot.gca()
78     ax.set(xlabel = "Time", ylabel = "Traffic (Kb)")
79     fmt = dates.DateFormatter('%Y-%m-%d %H:%M:%S')
80     ax.xaxis.set_major_formatter(fmt)
81     ax.xaxis.set_major_locator(dates.MinuteLocator(interval=10))
82     plot.scatter(datenums, y, s=5,c='tab:green', label="Traffic, Kb")
83     plot.title("Traffic over time", fontsize = 22)
84     plot.legend(fontsize=13)
85     plot.savefig("traffic_scatter.pdf", bbox_inches="tight")
86
87     data = []
88
89     with open("traffic.csv", "r") as file:
90         for line in file:
91             data.append(line.split(","))
92
93     IP = "217.15.20.194"

```

Рисунок 5 Программный код для графика (часть 3)

```

94     k = 2
95     traffic = 0
96     trafic_range = []
97
98     for i in range(len(data)-3):
99         line = data[i]
100         sa = line[3]
101         da = line[4]
102         ibyt = line[12]
103         if sa == IP or da == IP:
104             traffic += int(ibyt)
105
106     draw_graphic(data, IP)
107
108     traffic /= 1024*1024
109
110     print(round(traffic * k, 2))
111
112     # for i in range(len(mas[0])):
113     #     if mas[0][i] == "sa":
114     #         print(f"sa: {i}", end = " ")
115     #     elif mas[0][i] == "da":
116     #         print(f"da: {i}", end = " ")
117     #     elif mas[0][i] == "ibyt":
118     #         print(f"ibyt: {i}", end = " ")
119     #     elif mas[0][i] == "obyt":
120     #         print(f"obyt: {i}", end = " ")
121     #     elif mas[0][i] == "ts":
122     #         print(f"ts: {i}", end = " ")
123     #     elif mas[0][i] == "te":
124     #         print(f"te: {i}", end = " ")
125     # print()

```

Рисунок 6 Программный код для графика (часть 4)

Разработанный программный модуль строит два графика – traffic_diagram.pdf и traffic_scatter.pdf, первый из которых соединяет точки друг между другом для создания диаграммы.

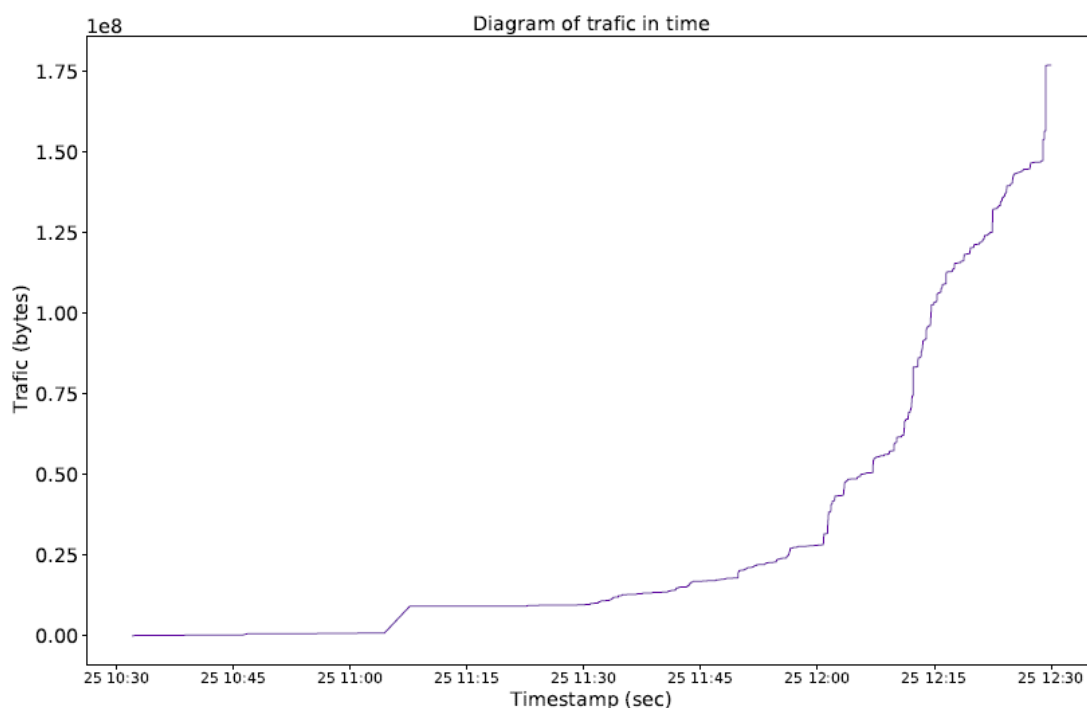


Рисунок 7 Traffic_diagram.pdf

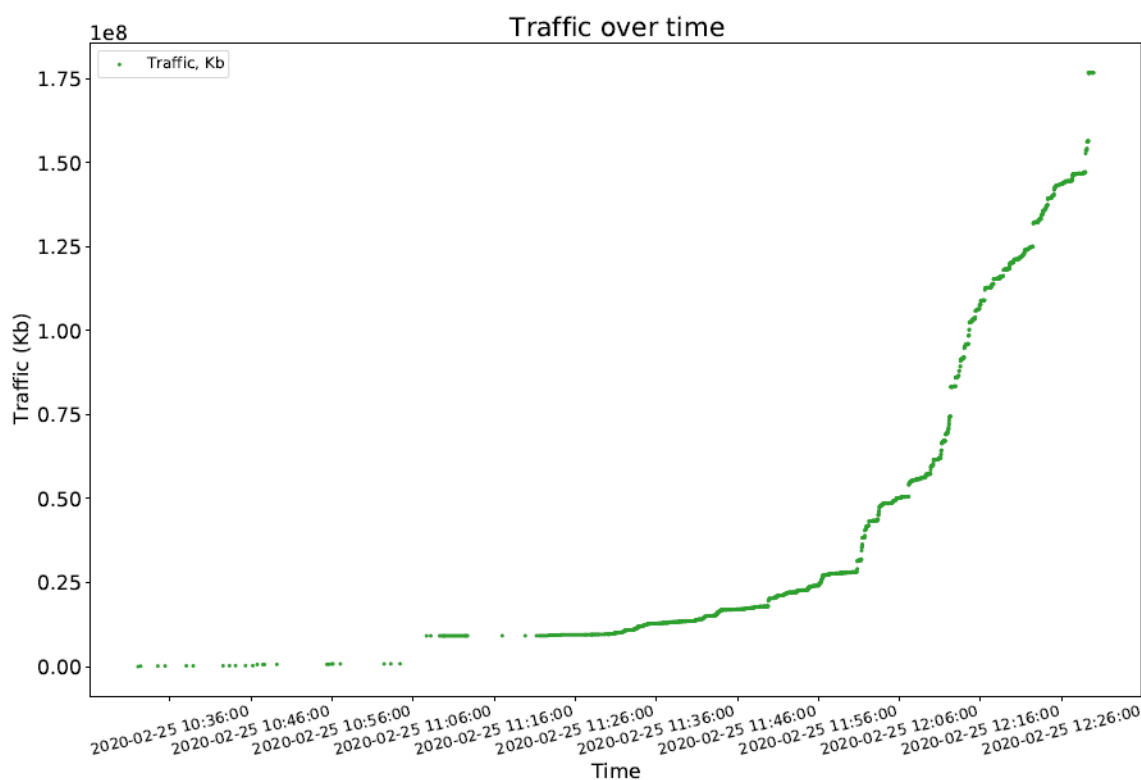


Рисунок 8 Traffic_scatter.pdf

Выводы

В ходе данной лабораторной работы был изучен принцип работы протокола NetFlow, предназначенного для сбора информации об IP – трафике внутри сети, применяющегося в биллинговых системах.

Был написан программный модуль, позволяющий выполнить тарификацию абонента по услуге «Интернет». По полученным данным об использованном трафике и о периоде, за который этот трафик был использован, был построен график зависимости объема трафика от времени, где объем трафика постоянно увеличивается с течением времени.