

Министерство науки и высшего образования Российской Федерации
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО»


Факультет безопасности информационных технологий

Дисциплина:
«Управление мобильными устройствами»

ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №2
«Обработка и тарификация трафика NetFlow»

Выполнила:

Студентка гр. N3349
Серова Ольга Евгеньевна

14.04.2020 

Проверил:

Федоров Иван Романович

Санкт-Петербург
2020 г.

Цель работы: обработка трафика NetFlow v5 из данного файла и реализация простейшего правила тарификации этого трафика.

Задачи:

1. Привести данный файл в читабельный вид (например 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;
4. Протарифицировать трафик в соответствии с вариантом задания.

Ход работы

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).

Для того, чтобы привести данные из файла в читабельный вид необходимо выполнить команду:

```
nfdump -r nfcapd.202002251200 -s ip
```

данная команда предназначена для анализа трафика и выдачи статистики. В нашем случае статистика будет составлена для ip адресов (рисунок 1). В статистику входят такие показатели, как Date first seen, Duration, Proto, IP Addr, Flows(%), Packets(%), Bytes(%), pps, bps, bpp.

MacBook-Pro-MacBook:Downloads macbook\$ nfdump -r nfcapd.202002251200 -s ip										
Top 10 IP Addr ordered by flows:										
Date first seen	Duration	Proto	IP Addr	Flows(%)	Packets(%)	Bytes(%)	pps	bps	bpp	
2020-02-25 10:32:54.720	7020.460	any	192.168.250.1	7053(40.4)	26812(6.4)	6.2 M(2.8)	3	7070	231	
2020-02-25 10:32:04.060	7072.860	any	217.15.20.194	4823(27.6)	210004(50.3)	176.7 M(80.3)	29	199823	841	
2020-02-25 12:11:00.360	1125.200	any	192.168.250.39	2494(14.3)	31101(7.5)	4.2 M(1.9)	27	29831	134	
2020-02-25 10:39:39.360	6617.560	any	192.168.250.3	2177(12.5)	50836(12.2)	14.0 M(6.4)	7	16964	276	
2020-02-25 11:50:56.470	2340.300	any	192.168.250.41	1696(9.7)	30679(7.4)	5.5 M(2.5)	13	18843	179	
2020-02-25 11:51:04.070	2330.740	any	192.168.250.59	1617(9.3)	33582(8.0)	8.5 M(3.9)	14	29165	253	
2020-02-25 10:32:39.070	7033.320	any	192.168.250.27	1537(8.8)	21203(5.1)	3.8 M(1.7)	3	4319	179	
2020-02-25 10:34:30.650	6903.190	any	192.168.250.50	1427(8.2)	20551(4.9)	4.3 M(2.0)	2	4995	209	
2020-02-25 11:53:13.150	30.670	any	34.201.100.14	858(4.9)	5458(1.3)	962930(0.4)	177	251171	176	
2020-02-25 10:32:23.840	7023.120	any	192.168.250.57	435(2.5)	4660(1.1)	968045(0.4)	0	1102	207	
Summary: total flows: 17449, total bytes: 220.0 M, total packets: 417314, avg bps: 248857, avg pps: 59, avg bpp: 527										
Time window: 2020-02-25 10:32:04 - 2020-02-25 12:29:56										
Total flows processed: 17449, Blocks skipped: 0, Bytes read: 1117124										
Svs: 0.022s flows/second: 770681.5 Wall: 0.010s flows/second: 1633189.8										

Рис. 1 «Результат выполнения команды nfdump -r nfcapd.202002251200 -s ip»

Также стоит отметить то, что данная статистика собирает весь объем трафика по ip адресам, не зависимо от того, был ли рассматриваемый ip адрес адресом приемной для трафика стороны или же адресом отправителя этого трафика.

Полученные данные переносим из терминала в таблицу **DATA.xlsx** (рисунок 2) вручную, так как команда `nfdump -r nfcapd.202002251200 -s ip` с ключом `-o cvs` выполняет некорректную конвертацию данных. Стоит также привести все значения столбца Bytes к единому формату представления данные, иными словами в записи каждого отдельного значения не должны присутствовать буквы, их следует заменить умножением на 10 в соответствующей этой буквенной приставке степени.

	A	B	C	D	E	F	G	H	I	J
1	Date first seen	Duration	Proto	IP_Addr	Flows(%)	Packets(%)	BytesMb	pps	bps	bpp
2	2020-02-25 10:32:54.720	7020.460	any	192.168.250.1	7053(40.4)	26812(6.4)	6.2	3	7070	231
3	2020-02-25 10:32:04.060	7072.860	any	217.15.20.194	4823(27.6)	210004(50.3)	176.7	29	199823	841
4	2020-02-25 12:11:00.360	1125.200	any	192.168.250.39	2494(14.3)	31101(7.5)	4.2	27	29831	134
5	2020-02-25 10:39:39.360	6617.560	any	192.168.250.3	2177(12.5)	50836(12.2)	14.0	7	16964	276
6	2020-02-25 11:50:56.470	2340.300	any	192.168.250.41	1696(9.7)	30679(7.4)	5.5	13	18843	179
7	2020-02-25 11:51:04.070	2330.740	any	192.168.250.59	1617(9.3)	33582(8.0)	8.5	14	29165	253
8	2020-02-25 10:32:39.070	7033.320	any	192.168.250.27	1537(8.8)	21203(5.1)	3.8	3	4319	179
9	2020-02-25 10:34:30.650	6903.190	any	192.168.250.50	1427(8.2)	20551(4.9)	4.3	2	4995	209
10	2020-02-25 11:53:13.150	30.670	any	34.201.100.14	858(4.9)	5458(1.3)	0.962930	177	251171	176
11	2020-02-25 10:32:23.840	7023.120	any	192.168.250.57	435(2.5)	4660(1.1)	0.968045	0	1102	207

Рис.2 «Перенесенные в таблицу данные»

После того, как файл для тарификации в виде таблицы формата xlsx был сформирован, он был обработан программным модулем листинг которого приведен на рисунке 3:

```
import pandas as pd

data=pd.read_excel('/Users/macbook/Documents/DATA.xlsx')

print("Введите множитель тарифного плана k руб/Mb-",end='')
k=int(input())

print("Введите ip-адрес клиента-",end='')
ip=input()

X=k*data[data.IP_Addr==ip].BytesMb.values[0]-k*1
print("Результат тарификации-",X,"руб.")
```

Рис.3 «Листинг программного модуля **mob2.py**»

Как видно из кода программы для ее реализации так же, как и в первой лабораторной работе был выбран высокоуровневый язык программирования Python3 и библиотека Pandas, предназначенная для анализа и обработки данных. Такой выбор был сделан мной ввиду простоты работы как с самим Python, так и с библиотекой Pandas.

Вариант 2

Протарифицировать абонента с IP-адресом 217.15.20.194

с коэффициентом k: 1руб/Мб, первая 1000Мб бесплатно

Результат тарификации приведен на рисунке 4:

```
===== RESTART: /Users/macbook/Desktop/mob2.py
Введите множитель тарифного плана k руб/Мб-1
Введите ip-адрес клиента-217.15.20.194
Результат тарификации- 175.7 руб.
```

Рис. 4 «Результата выполнения тарификации клиента с IP-адресом 217.15.20.194 по условиям тарификации»

Для построения графика зависимости объема трафика от времени необходимо отфильтровать данные NetFlow из файла. Для этого необходимо выполнить команды:

```
nfdump -r nfcapd.202002251200 >mobile2.txt
```

```
grep -bn 217.15.20.194 mobile2.txt >graph
```

Первая команда просто сохраняет полученные с помощью утилиты nfdump данные в файл **mobile2.txt**, а вторая команда находит все строки, в которых встречается заданный условием варианта ip адрес, и сохраняет их в бинарный файл без расширения **graph**.

Далее для извлечения нужных данных, а именно объема трафика(Bytes) и времени (Date first seen), был написан код, листинг которого представлен на рисунке 5. На выходе данной программы имеем файл **grep** наполовину состоящий из величин объема трафика и на половину из значений времени, соответствующих объемам трафика. Также еще одним результатом выполнения данного кода является сумма всех объемов трафика (в байтах) (рисунок 6). Сравнивая данный результат (=176,81 МБ) я тем, который мы получили из статистики (рис.1, =176,7 МБ), можно отметить, что в статистике результаты посчитаны менее точно, однако погрешность, в переводе на денежный эквивалент согласно варианту задания, незначительная.

Получив необходимые для построения графика значения объема трафика и времени, перенесем их в таблицу excel и средствами MS-Excel построим требуемый график (рисунок 7).

Выводы: на основе проделанной лабораторной работы можно сделать вывод о том, что автоматизация процесса тарификации абонентов за услуги типа «Интернет» неизбежна, так же, как и за услуги типа «Телефония», поскольку количество активностей пользователей сети Интернет (манипулирование трафиком) значительно превышает само количество пользователей, что приводит к необходимости постоянной фильтрации и отбора данных, что сделать намного проще с помощью автоматизированных программных модулей. Таким образом непосредственно процесс тарификации не является сложным, поскольку не содержит в себе методов из сложных разделов математики, однако главной трудностью является именно количество обрабатываемой информации.

```

file=open('/Users/macbook/Downloads/graph','r+')
data=file.read()
file.close()
j=0
ch=0
leng=-1;
suma=0
file=open('/Users/macbook/Downloads/grep','r+')

for i in range(len(data)):
    if data[i]==" " and data[i+1]!=" ":
        j+=1
    if j==8 and data[i-1]==" ":
        k=i
        while not data[k]==" ":
            leng+=1
            k+=1
        k=i
        while not data[k]==" ":
            ch+=int(data[k])*pow(10,leng)
            k+=1
            leng-=1
        if ch!=0:
            file.write(str(ch))
            file.write("\n")
            suma+=ch
        ch=0
    if j==9:
        j=0

print(suma)

buf=[]
j=0
for i in range(len(data)):
    if data[i]==" " and data[i+1]!=" ":
        j+=1
    if j==1 and data[i-4]=="-":
        k=i
        m=0
        while not data[k]==" ":
            buf+=data[k]
            k+=1
        stroka=''.join(buf)
        if len(buf)==8:
            file.write(stroka)
            file.write("\n")
            buf=[]
    if j==9:
        j=0

file.close()

```

Рис.5 «Листинг программы **graph-points.py**, выделяющей значения точек для построения графика»

```

===== RESTART: /Users/macbook/Desktop/graph-points.py
176808151

```

Рис.6 «Суммарный объем трафика пользователя с IP-адресом 217.15.20.194 »

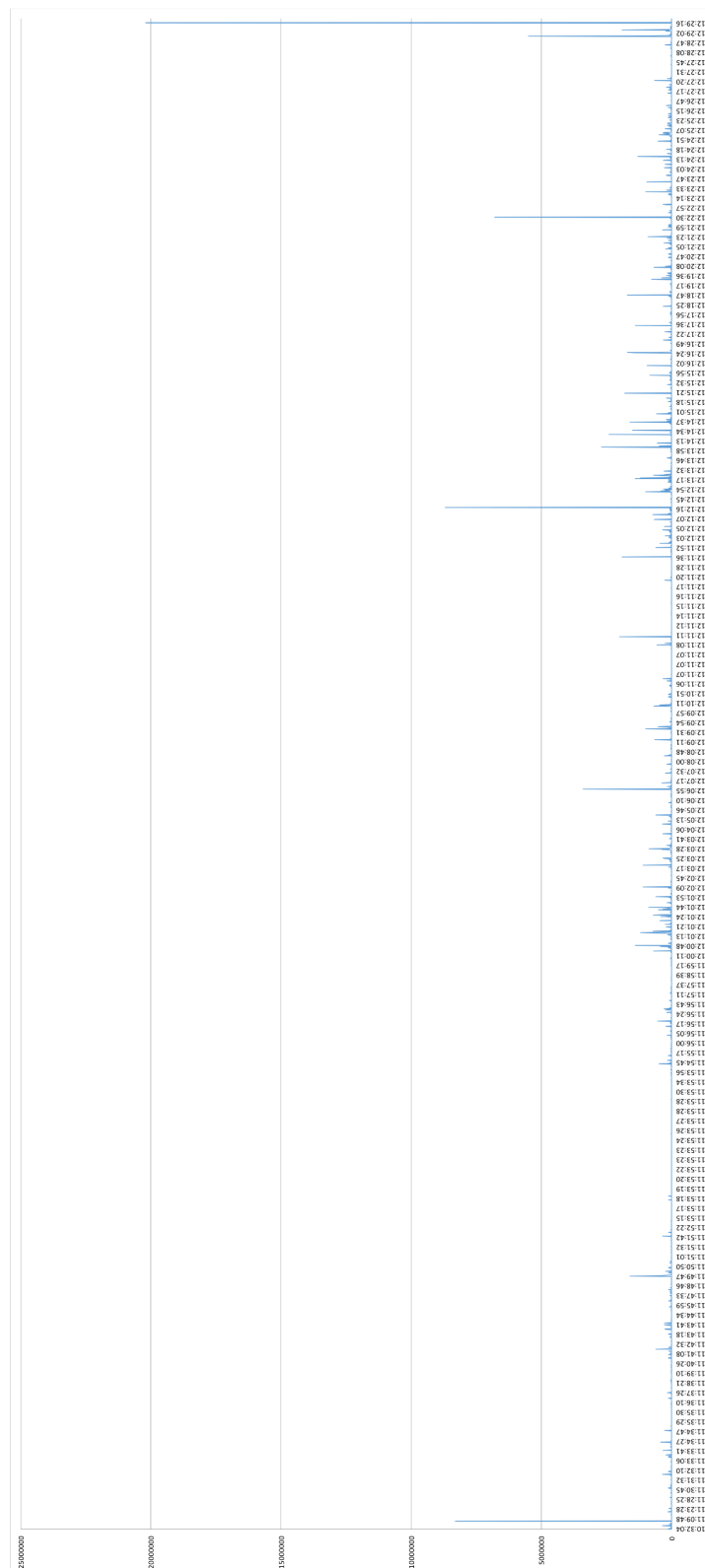


Рис.7 «График зависимости объема трафика от времени» (перевернут на 90° влево)