

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РФ

Федеральное государственное автономное
образовательное учреждение высшего образования
«Национальный исследовательский университет ИТМО»

ФАКУЛЬТЕТ БЕЗОПАСНОСТИ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ

Лабораторная работа №2
по теме:
Обработка и тарификация трафика NetFlow
Вариант 11

Работу выполнил
студент группы № N3348
очного отделения:

Ниценкова Д. В.



Проверил

Федоров И. Р.

Цель работы: реализовать простейшее правило тарификации для услуг типа «Интернет»

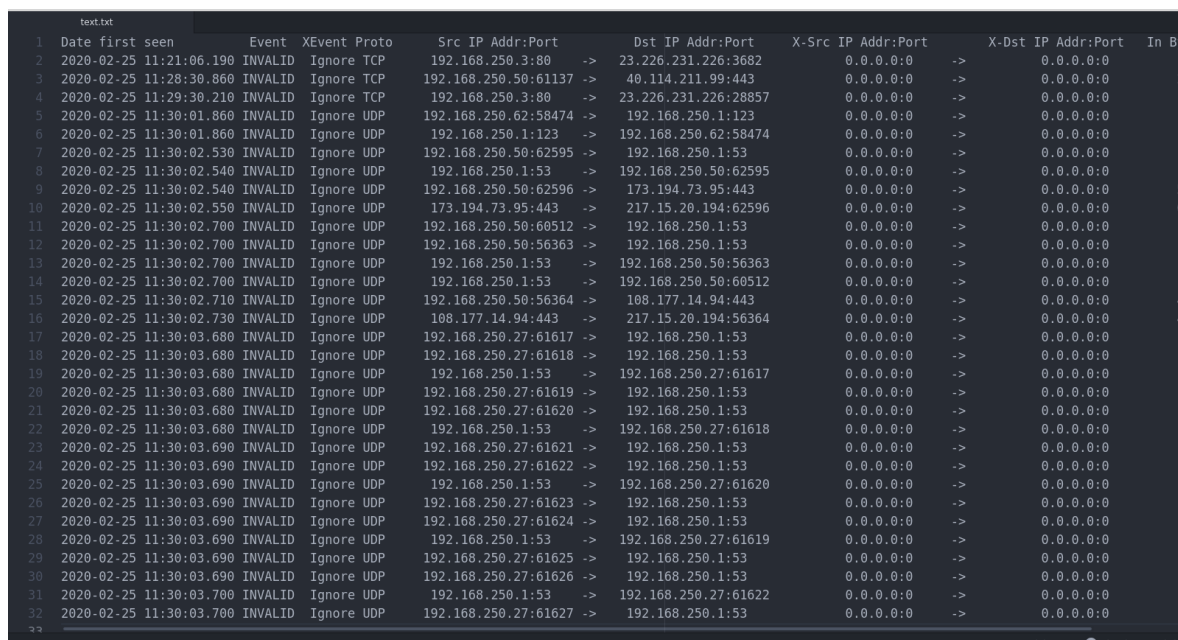
Выбранные средства: Язык программирования C

Ссылка на код: <https://github.com/darianic/MobileDevelopmentClass>

Ход работы:

1) Привести файл `nfcapd.202002251200` в читабельный вид (с помощью утилиты `nfdump`)

С помощью команды `nfdump -r nfcapd.202002251200 > text.txt` сформируем собственный файл для тарификации.



1	Date first seen	Event	XEvent	Proto	Src IP Addr:Port	Dst IP Addr:Port	X-Src IP Addr:Port	X-Dst IP Addr:Port	In	By
2	2020-02-25 11:21:06.190	INVALID	Ignore	TCP	192.168.250.3:80	-> 23.226.231.226:3682	0.0.0.0:0	-> 0.0.0.0:0		
3	2020-02-25 11:28:30.860	INVALID	Ignore	TCP	192.168.250.50:61137	-> 40.114.211.99:443	0.0.0.0:0	-> 0.0.0.0:0	2	
4	2020-02-25 11:29:30.210	INVALID	Ignore	TCP	192.168.250.3:80	-> 23.226.231.226:28857	0.0.0.0:0	-> 0.0.0.0:0		
5	2020-02-25 11:30:01.860	INVALID	Ignore	UDP	192.168.250.62:58474	-> 192.168.250.1:123	0.0.0.0:0	-> 0.0.0.0:0		
6	2020-02-25 11:30:01.860	INVALID	Ignore	UDP	192.168.250.1:123	-> 192.168.250.62:58474	0.0.0.0:0	-> 0.0.0.0:0		
7	2020-02-25 11:30:02.530	INVALID	Ignore	UDP	192.168.250.50:62595	-> 192.168.250.1:53	0.0.0.0:0	-> 0.0.0.0:0		
8	2020-02-25 11:30:02.540	INVALID	Ignore	UDP	192.168.250.1:53	-> 192.168.250.50:62595	0.0.0.0:0	-> 0.0.0.0:0		
9	2020-02-25 11:30:02.540	INVALID	Ignore	UDP	192.168.250.50:62596	-> 173.194.73.95:443	0.0.0.0:0	-> 0.0.0.0:0	5	
10	2020-02-25 11:30:02.550	INVALID	Ignore	UDP	173.194.73.95:443	-> 217.15.20.194:62596	0.0.0.0:0	-> 0.0.0.0:0	6	
11	2020-02-25 11:30:02.700	INVALID	Ignore	UDP	192.168.250.50:60512	-> 192.168.250.1:53	0.0.0.0:0	-> 0.0.0.0:0		
12	2020-02-25 11:30:02.700	INVALID	Ignore	UDP	192.168.250.50:56363	-> 192.168.250.1:53	0.0.0.0:0	-> 0.0.0.0:0		
13	2020-02-25 11:30:02.700	INVALID	Ignore	UDP	192.168.250.1:53	-> 192.168.250.50:56363	0.0.0.0:0	-> 0.0.0.0:0		
14	2020-02-25 11:30:02.700	INVALID	Ignore	UDP	192.168.250.1:53	-> 192.168.250.50:60512	0.0.0.0:0	-> 0.0.0.0:0		
15	2020-02-25 11:30:02.710	INVALID	Ignore	UDP	192.168.250.50:56364	-> 108.177.14.94:443	0.0.0.0:0	-> 0.0.0.0:0	4	
16	2020-02-25 11:30:02.730	INVALID	Ignore	UDP	108.177.14.94:443	-> 217.15.20.194:56364	0.0.0.0:0	-> 0.0.0.0:0	4	
17	2020-02-25 11:30:03.680	INVALID	Ignore	UDP	192.168.250.27:61617	-> 192.168.250.1:53	0.0.0.0:0	-> 0.0.0.0:0		
18	2020-02-25 11:30:03.680	INVALID	Ignore	UDP	192.168.250.27:61618	-> 192.168.250.1:53	0.0.0.0:0	-> 0.0.0.0:0		
19	2020-02-25 11:30:03.680	INVALID	Ignore	UDP	192.168.250.1:53	-> 192.168.250.27:61617	0.0.0.0:0	-> 0.0.0.0:0		
20	2020-02-25 11:30:03.680	INVALID	Ignore	UDP	192.168.250.27:61619	-> 192.168.250.1:53	0.0.0.0:0	-> 0.0.0.0:0		
21	2020-02-25 11:30:03.680	INVALID	Ignore	UDP	192.168.250.27:61620	-> 192.168.250.1:53	0.0.0.0:0	-> 0.0.0.0:0		
22	2020-02-25 11:30:03.680	INVALID	Ignore	UDP	192.168.250.1:53	-> 192.168.250.27:61618	0.0.0.0:0	-> 0.0.0.0:0		
23	2020-02-25 11:30:03.690	INVALID	Ignore	UDP	192.168.250.27:61621	-> 192.168.250.1:53	0.0.0.0:0	-> 0.0.0.0:0		
24	2020-02-25 11:30:03.690	INVALID	Ignore	UDP	192.168.250.27:61622	-> 192.168.250.1:53	0.0.0.0:0	-> 0.0.0.0:0		
25	2020-02-25 11:30:03.690	INVALID	Ignore	UDP	192.168.250.1:53	-> 192.168.250.27:61620	0.0.0.0:0	-> 0.0.0.0:0		
26	2020-02-25 11:30:03.690	INVALID	Ignore	UDP	192.168.250.27:61623	-> 192.168.250.1:53	0.0.0.0:0	-> 0.0.0.0:0		
27	2020-02-25 11:30:03.690	INVALID	Ignore	UDP	192.168.250.27:61624	-> 192.168.250.1:53	0.0.0.0:0	-> 0.0.0.0:0		
28	2020-02-25 11:30:03.690	INVALID	Ignore	UDP	192.168.250.1:53	-> 192.168.250.27:61619	0.0.0.0:0	-> 0.0.0.0:0		
29	2020-02-25 11:30:03.690	INVALID	Ignore	UDP	192.168.250.27:61625	-> 192.168.250.1:53	0.0.0.0:0	-> 0.0.0.0:0		
30	2020-02-25 11:30:03.690	INVALID	Ignore	UDP	192.168.250.27:61626	-> 192.168.250.1:53	0.0.0.0:0	-> 0.0.0.0:0		
31	2020-02-25 11:30:03.700	INVALID	Ignore	UDP	192.168.250.1:53	-> 192.168.250.27:61622	0.0.0.0:0	-> 0.0.0.0:0		
32	2020-02-25 11:30:03.700	INVALID	Ignore	UDP	192.168.250.27:61627	-> 192.168.250.1:53	0.0.0.0:0	-> 0.0.0.0:0		

Рисунок 1. Результат выполнения команды

2) Построить график зависимости объема трафика от времени

Воспользуемся программой `gnuplot` для создания графиков. Из строк, в которых IP назначения или IP источника равен 17.248.150.51, выберем время и объем трафика и поместим в новый файл `graph.plt`. Для удобства возьмем часы и минуты (код программы приведен ниже).

```

root@kali:~/Desktop/mobile# cat graph.plt
set terminal jpeg
set output "res.jpeg"
plot '-' using 1:2 w points ps 3 pt 4
1224 6358
1224 3092
1229 3966
1229 8613
1229 4264
1229 26162
e
root@kali:~/Desktop/mobile#

```

Рисунок 2. Файл graph.plt

Создадим график командой **gnuplot graph.plt**.

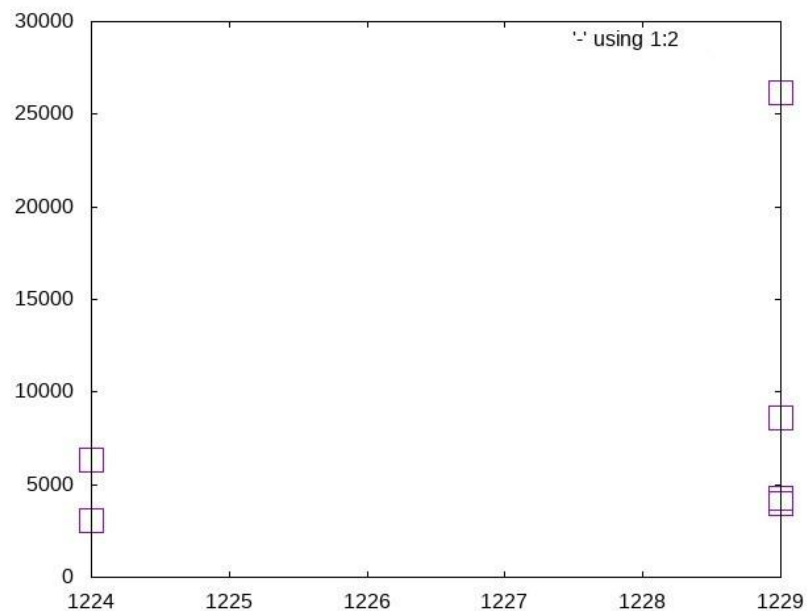


Рисунок 3. График зависимости объема трафика от времени

По данному графику можно понять, что в 12:29 объем трафика был больше, чем в 12:24.

- 3) Протарифицировать трафик в соответствии с вариантом задания

Исходный код:

```

#include <stdio.h>
#include <stdlib.h>
#include <string.h>

void Internet(float bytes) {
    //X = Q * k

```

```

//Q - общий объем трафика NetFlow за отчетный период
//k - множитель тарифного плана, k = 0,5руб/Мб
float k = 0.5;
float x = bytes * k / 1048576;

printf("Итоговая стоимость трафика для IP 17.248.150.51: %f\n", x);

}

int main() {

    FILE*f1 = NULL;
    f1=fopen("./text.txt","r");

    FILE*f2 = NULL;
    f2=fopen("./graph.plt","w");

    fprintf(f2,"set terminal jpeg\nset output \"res.jpeg\"\nplot '-' using 1:2 w points ps 3 pt
4\n");

    char str[300];

    float bytes = 0;
    char * hi;

    while (fgets(str,300,f1)) {
        //puts(str);
        char m[100][100];
        int i=0;
        char *p;
        p = strtok(str, " ");
        while(p!=NULL){
            strcpy(m[i],p);
            i++;
            p=strtok(NULL, " ");
        }
        int j=0;
        char *istr1;
        char *istr2;
        char *istr3;

        for (int j=0; j<i; j++) {

            char * value1 = m[j];
            char * value2 = m[j+1];
            char * value3 = m[j+2];

            char * str = "217.15.20.194";
            char * str1 = "M";

            istr1 = strstr(value1,str);
            istr2 = strstr(value2,str);

```

```

        istr3 = strstr(value3, str1);

        if (istr1 != NULL || istr2 != NULL) {
            if (istr3 != NULL) {
                bytes = bytes + atof(m[11])/i*1048576;
            }
            else {
                bytes = bytes + atoi(m[11])/i;
            }
        }
    }

    if (istr1 != NULL || istr2 != NULL) {
        char * time = m[1];
        int k=0;
        char *t;
        int chisla[4] = {0};
        t = strtok(time, ".");
        while(t!=NULL){
            chisla[k] = atoi(t);
            k++;
            t=strtok(NULL, ".");
        }

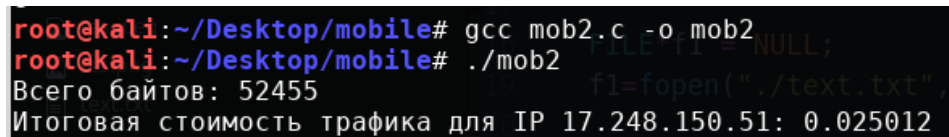
        if (istr3 != NULL)
            fprintf(f2, "%d%d %f\n", chisla[0], chisla[1], atof(m[11])*1048576);
        else
            fprintf(f2, "%d%d %d\n", chisla[0], chisla[1], atoi(m[11]));
    }
}
fclose(f1);
fprintf(f2, "e\n");
fclose(f2);
printf("Всего байтов: %.0f\n", bytes);
printf("\n");

Internet(bytes);

return 0;
}

```

Пример работы программы:



```

root@kali:~/Desktop/mobile# gcc mob2.c -o mob2
root@kali:~/Desktop/mobile# ./mob2
Всего байтов: 52455
Итоговая стоимость трафика для IP 17.248.150.51: 0.025012

```

Рисунок 4. Результат выполнения программы

Вывод:

В ходе выполнения лабораторной работы был обработан трафик NetFlow v5 и было программно реализовано простейшее правило тарификации для услуг типа «Интернет».