Семинарска по Мултимедиски мрежи

iptv udp multicast proxy streaming

https://github.com/pecop2/iptv_multicast_udp_proxy_streaming

Идеја

Како корисник на интернет телевизија, произлезе и идејата за овој проект, односно лимитот од провајдерот за глеадање исклучиво еден видео стрим за истиот корисник во даден момент. Целта на семинарската е да го надмине проблемот, односно да се овозможи гледање на повеќе од еден стрим (на истиот или различни уреди) во ист момент. Јазик за имплементација: Python

Краток опис

Од веќе постоечката m3u листа со стримови, се генерира нова на начин што се прави мапа која клучеви ги содржи новите мултикаст адреси на кои што истиот тој стрим ќе се праќа, а како вредност е оригиналниот линк до стримот. Мултикаст адресите стартуваат од 239.123.1.1 до таму колку што е доволно во зависност од бројот на стримови во листата (0 и 255 не се користат поради можноста да се резервирани)

239.123.X.X; 1<=X<=254 → вкупно 254*254 = 64516 адреси (максимум број на стримови).

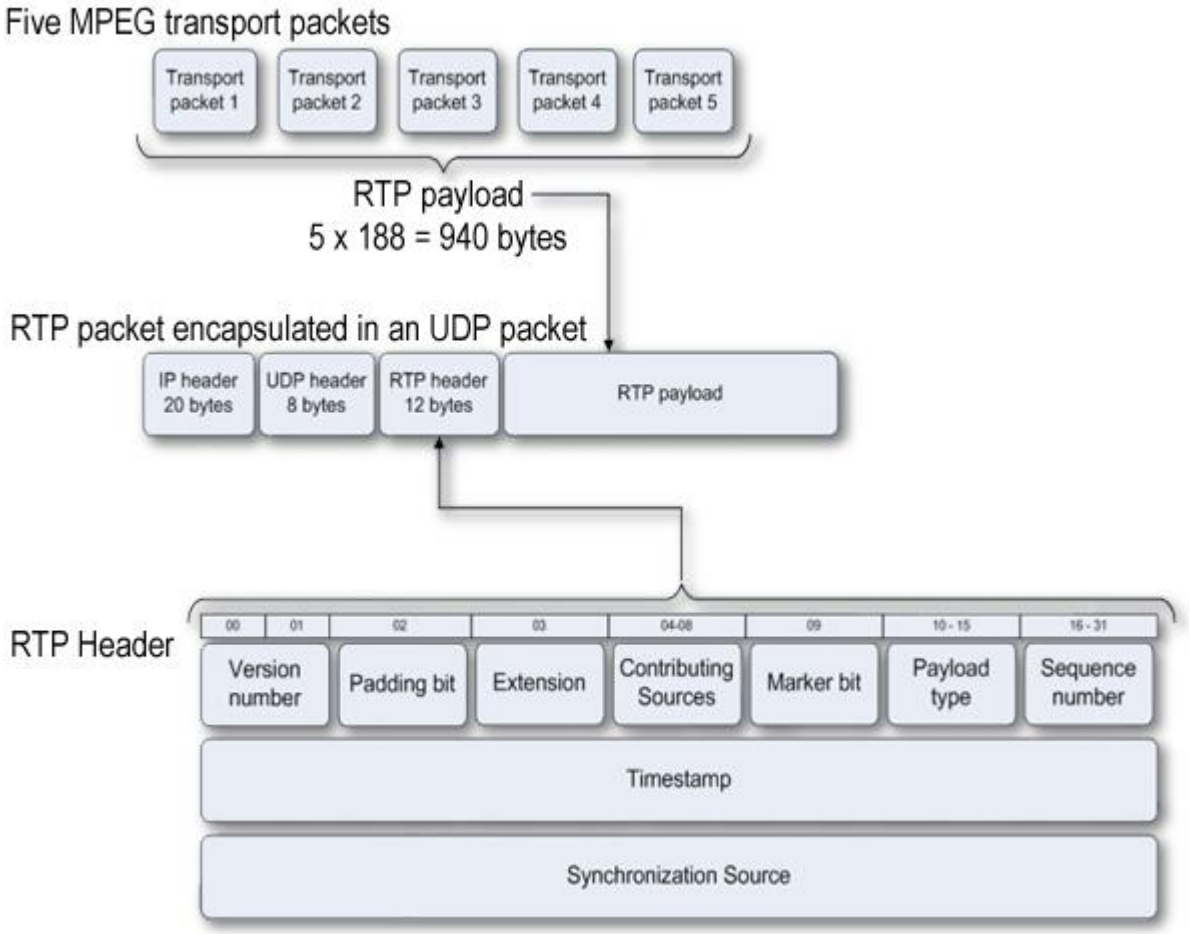
Сепак ако се исполни, се преминува на 239.124.X.X.

Истата листа потоа се користи наместо оригиналната за пристап до стримовите. Листата се опслужува со локален фајл сервер.

Се користи python библиотеката за VLC: https://pypi.org/project/python-vlc/

Кога клиент ќе направи барање до прокси серверот, се стартува VLC player (доколку веќе нема стартувано од корисник кој го глеада истиот стрим), кој како извор го користи оригиналниот стрим, а како излез е rtp стрим на мултикаст адресата која соодветствува според гореспоменатата мапа. *Во кодот класа **MulticastStream** која наследува од **Thread**.*

Откако стартувањето на VLC player-от е завршено, се креира мултикаст сокет кој го прима истиот стрим кој претходно беше стартуван. Откако ќе се прими стримот, се тргаат првите 12 бајти (мноооогу изгубено време дури сфатам дека ова е проблемот што го расипува стримот, 12те бајти соодветствуваат на 12 бајтниот rtp header), а остатокот се препраќа на корисникот. Доколку корисникот заврши со гледање, а не постои друг корисник кој го гледа истиот стрим, VLC player-от се стопира.



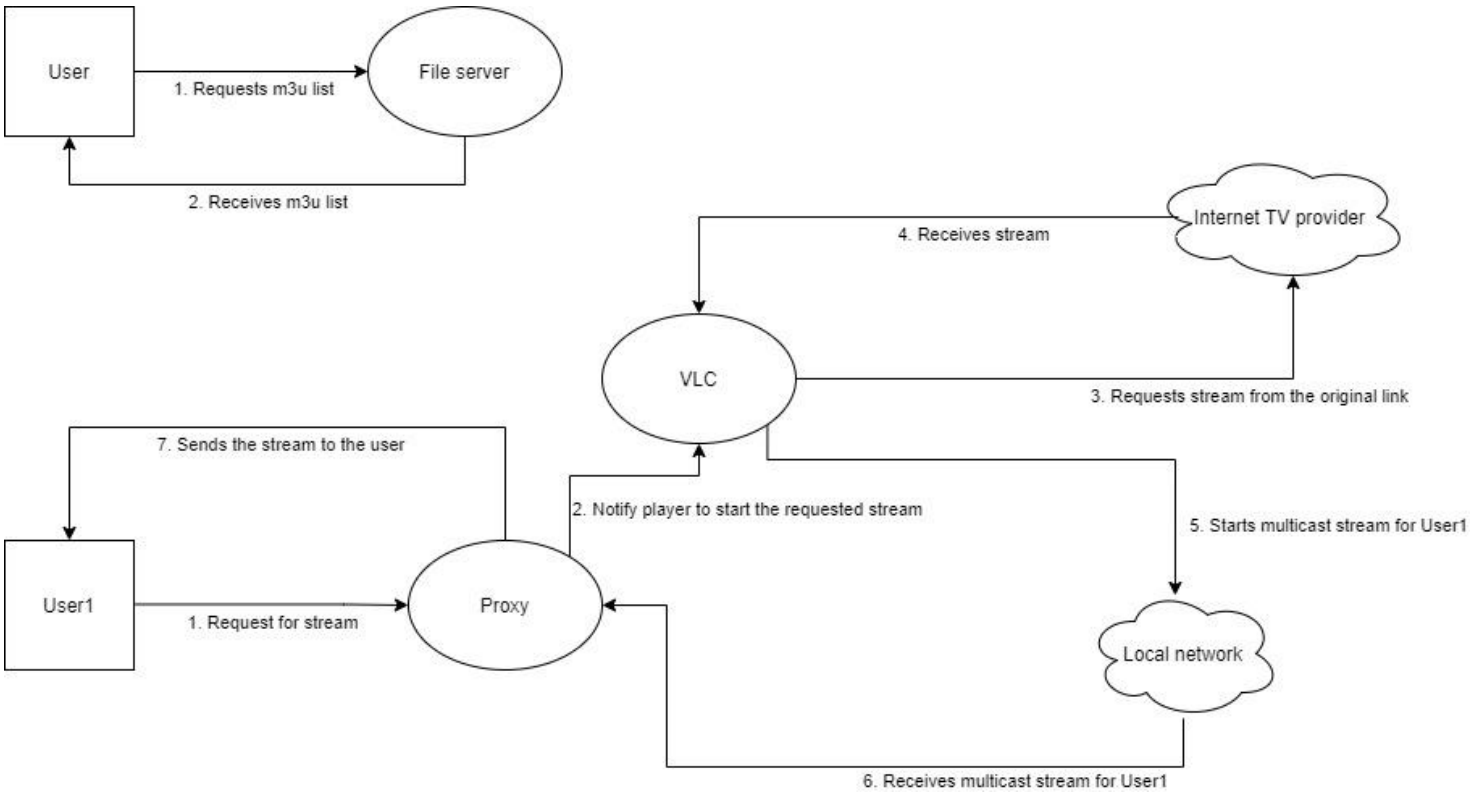
1. RTP packet fields

Колку за полесно тестирање и/или стартување направен е и Dockerfile, преку кој може да се избилда image и да се стартува сервисот. Се користат environment variables за конфигурација и тоа:

- **ORIGINAL_M3U_URL**=your_m3u_url_here # ex: http://your-provider.com/list.m3u → Се користи за да се симне оригиналната m3u листа

- **NUMBER_OF_CLIENTS**=3 →Бројот на очекувани клиенти, се инстанцираат толку VLC players

- **HOST_IP**=local_ip_of_the_host_running_this_service # ex: 192.168.1.2 → за да се конструира правилно m3u листата, не успеав да најдам начин од docker container-от да ја откријам ИП адресата на хостот.



2. Flow chart

Резултат

Очекувани беа мали прекини при повторно враќање на стримот на првиот корисник. Во пракса се случува некогаш да се забележат мали прекини, или навраќање на стримот одредено мало време наназад. Честотата на овие прекини ќе се зголемува со зголемување на бројот на корисници, но исто така зависи и од квалитетот на првобитниот стрим.

Овој проблем пробував да го решам на различни начини како на пример:

- намалување на buf_size при читање од мултикаст стримот

- баферирање со едноставно споредување на претходно испратените податоци со новодојдените податоци (не се полни баферот ако веќе има испратено ист податок)

но овие обиди беа неуспешни.

Така што, како понатамошна работа останува решавање на овој проблем.

Изработил: Петар Петров

Индекс: 161052