



РЕАЛИЗАЦИЈА ТРОСЛОЈНОГ СИСТЕМА ЗА ВИДЕО ПРЕНОС ПО RTSP ПРОТОКОЛУ

REALIZATION OF THREE-LAYER RTSP VIDEO STREAMING SYSTEM

Јован Славујевић, Предраг Теодоровић, *Факултет техничких наука, Нови Сад*

Област – ЕЛЕКТРОТЕХНИКА И РАЧУНАРСТВО

Кратак садржај – У овом чланку описана је реализација и функционисање трослојног система за видео пренос по RTSP протоколу. Омогућено је вишезначно видео преношење уживо са једне или више камера, као и једнозначно приказивање по захтеву раније снимљених видео записа у временским размацама од 15 минута. Систем је имплементиран на личној машини - i7@3GHz са 16GB RAM и Windows 10 OS. Систем је подржан и на Linux OS, како би се омогућила имплементација на популарним ембедед платформама као што је нпр. Raspberry Pi.

Кључне речи: RTSP, Qt, GUI, SQL, криптографија

Abstract – This article describes realization and functioning of three-layer video streaming system according to RTSP protocol. Here is provided live multicast video streaming from one or more cameras and unicast On Demand video streaming of earlier recorded video files in 15 minutes time slots. System was implemented on personal computer - i7@3GHz with 16GB RAM and Windows 10 OS. System has supported Linux OS, in order to provide implementation on popular embedded platforms such as Raspberry Pi.

Keywords: RTSP, Qt, GUI, SQL, cryptography

1. УВОД

Видео пренос представља незаобилазну појаву данашњице, а у највећој мери предњачи интернет видео пренос. Није изненађујуће што је моћ директности коју са собом носи телевизија пронашла свој пут до интернета. Online видео постаје императив, па је тако све више различитих платформи које омогућавају живи видео пренос.

Постоје два различита типа живог преноса. У првом начину, видео садржај је доступан директно на сајту и користи софтвер који је већ интегрисан у сам сајт. Код другог начина, гледаоци могу да прате садржај само кроз самостални (енг. *Standalone*) програм – рачунарску или мобилну апликацију, где софтвер омогућава приказивање садржаја.

У оквиру овог рада реализован је софтверски пакет који обухвата видео пренос – серверску страну, али и његово приказивање – клијентску страну. Ток видео

НАПОМЕНА:

Овај рад проистекао је из мастер рада чији ментор је био др. Предраг Теодоровић, доцент.

преноса диктиран је по специјалном стандарду видео преноса какав је RTSP [1]. Поред тога, приступ видео преносима лимитиран је евидентирањем корисника у бази података [2] и приступањем уз помоћ јединствених корисничких креденцијала.

Видео пренос представља пренос директних снимака са камере. Број камера је конфигурабилан, па их тако може бити једна или више. Камере морају бити физички повезане на уређају на којем је покренут сервер. Систем подржава и својеврсно „враћање у назад“ по захтеву у виду 4 могућа преноса која константно пружају снимак од последњих сат времена у размацама од по 15 минута.

Корисничка апликација поседује графички кориснички интерфејс – GUI, са менијем усмереним ка кориснику (енг. *User-friendly*). Цео систем је имплементиран и тестиран на локалној мрежи, али би уз извесне измене могао да функционише и на глобалној мрежи.

2. Концепт решења

У овом поглављу описан је и илустрован концепт решења, идеја пројекта, као и архитектура и функционалност појединачних модула система.

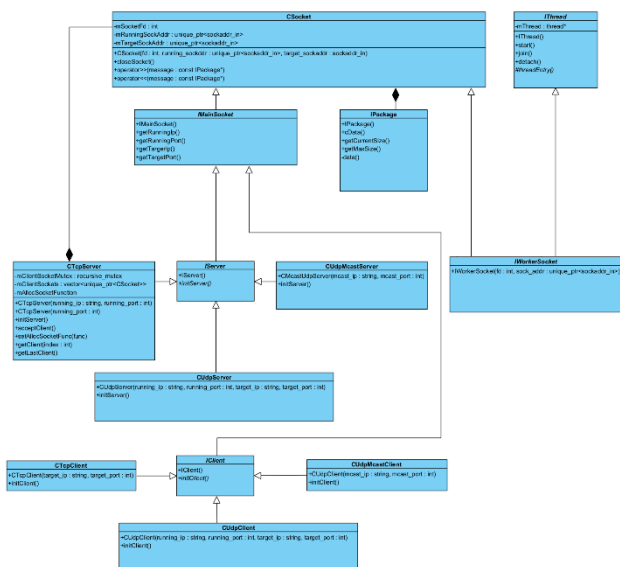
2.1 Библиотека мрежне утичнице у објектно оријентисаној форми

Интерфејс за програмирање апликације - API мрежне утичнице (енг. *Socket*) представља срце комуникације међу процесима или уређајима, старајући се о ниским нивоима комуникације као што су интернет протокол или ниже од тога, остављајући кориснику могућност да се стара и размишља само о комуникацији на апликативном нивоу.

Иако веома неопходан и готово свеприсутан, овај API написан је у програмском језику C, те га је, као таквог, некомфорно користити. Идеја је била прилагодити библиотеку објектно-оријентисаној парадигми паковањем њених функција и структура у класе познатије као *Wrapper* класе. На тај начин, поред сажетог и читког кода, добија се заједнички интерфејс за оба оперативна система, који се лако може користити у оквиру било које апликације.

Интерфејса садржи 3 пара класа које је могуће користити (инстанцирати). У питању су парови – клијент и сервер, а 3 пара покривају TCP, затим једнозначни и вишезначни UDP IP [3]. Поред наведених постоје и апстрактне класе које је могуће наследити. Једна од њих је класа *IPackage* која се користи за слање и пријем порука. У оквиру изведене класе потребно је

дефинисати (*override*) чисте (*pure*) виртуалне метода које заправо представљају тзв. *Getter* методе за максималну и искоришћену величину, као и стартну адресу поруке. На следећој слици (*слика 1.*) приказан је *UML* дијаграм класа.



Слика 1. UML класни дијаграм Wrapper библиотеке

2.2 Архитектура, подела и улога модула система

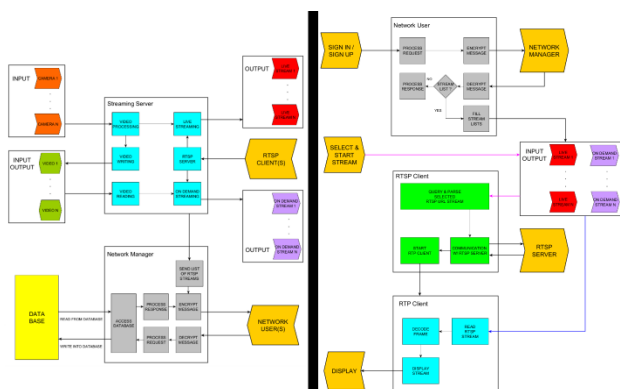
Серверска апликација садржи два велика модула:

1. Сервер за видео пренос - модул задужен за сву видео обраду, интеракцију и дистрибуцију преноса.
2. Мрежни менаџер - модул задужен за евиденцију корисника, одржавање базе података и заштиту личних података корисника.

Клијентска апликација садржи 3 мања модула:

1. *RTSP* клијент - модул задужен за потраживање, прекид или пребацивање на други видео пренос.
2. *RTP* клијент - модул задужен за пријем, обраду и приказивање видео преноса по налогу *RTSP* клијента.
3. Мрежни корисник - модул задужен за приступ мрежи система и набавку листе могућих преноса.

На следећој слици (*слика 2.*) приказане су идејне блок шеме за обе апликације у оквиру система.



Слика 2. Идејне блок шеме серверске и клијентске апликације у оквиру трослојног система

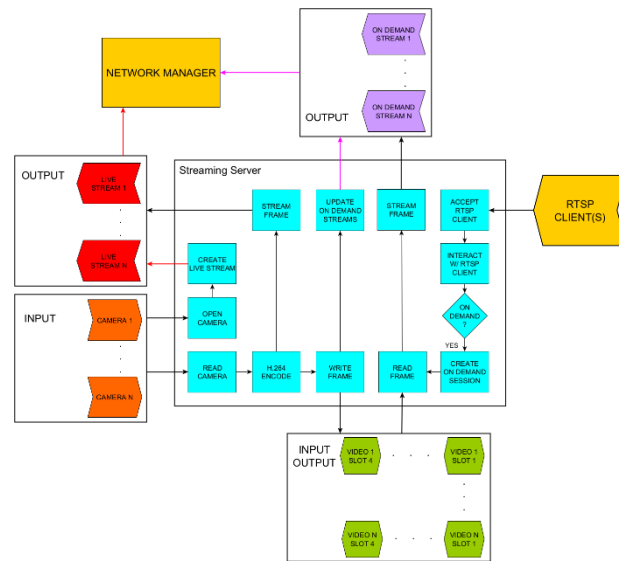
3. ПРОГРАМСКО РЕШЕЊЕ

У овом поглављу описано је програмско решење пројекта - рад појединачних модула.

3.1 Програмско решење серверске апликације

3.1.1 Сервер за видео пренос

Главна улога овог модула јесте видео пренос. Међутим његове активности превазилазе те оквире, па је тако задужен за целокупну видео обраду, интеракцију са клијентима преноса и услуживање видео преноса – живо и по захтеву. На следећој слици (*слика 3.*) приказана је детаљнија блок шема сервера за видео пренос.



Слика 3. Детаљна блок шема сервера за видео пренос

Читање са камере и видео записивање

Серверска апликација при покретању зна којим бројем камера ће управљати. Да би управљала камером, апликација користи објекте из *OpenCV* библиотеке. Након што је камера успешно повезана, креира се и покреће посебна нит у којој се извршава петља читања све до краја рада серверске апликације или физичког прекида везе са камером.

Слика са камере се преузима у сировом формату, где као таква досеже огромне величине. За слику резолуције 640x480 добија се 307.200 пиксела, што множењем са бројем основних боја из *RGB* система резултује у 921.600 бајта, односно 900 *kB*. Очигледно је да слика у оваквом формату није практична за манипулисање или пренос путем мреже.

Због тога се након читања слика компресује по *H.264* [4] кодеку. Овај механизам реализује се уз помоћ библиотеке *Ffmpeg*. Након што је сирова слика камере компресована добија се *frame* који је знатно мањи, а његова величина варира у зависности од промена које је ухватила камера, али највећа величина до које досеже је око 70 *kB*.

Новонастали *frame* се прослеђује на видео записивање. Генерисање видео функционише тако да сваки фајл не треба бити дужи од 15 минута. У моменту када то време истекне, ажурира се листа видео преноса по захтеву. Евиденција о протеклом

времену од 15 минута извршава се у другој нити која мери протекло време и обавештава када оно истекне, да би се видео записивач пребацио на нови видео.

Креирање новог видео записа функционише по принципу кружног попуњавања постојећих фајлова када постану застарели. Како би се избегла ситуација да видео записивач покуша да пресними видео који се користи, уведена су два фајла која се наизменично користе за уписивање. На тај начин постоји 8 видео записа од по 15 минута за 4 могућа видео преноса по захтеву који се циклично попуњавају. Уписани видео *frame* се даље прослеђује на пренос уживо.

RTSP интеракција са клијентима

Сервер за видео пренос пружа пренос искључиво по *RTSP* протоколу. Протокол захтева од корисника да, поред утичнице за добављање видео садржаја (*RTP*), поседује и једну за интеракцију са *RTSP* сервером.

RTSP сервер се може илустровано представити као телевизор, а клијент као даљински управљач, где се притиском бројева на управљачу добија комбинација на основу које телевизор зна на који канал треба усмерити гледаоца. Комбинацију, у *RTSP* протоколу, представља *URL* линк који клијент шаље приликом интеракције, док усмеравање гледаоца представља прослеђивање *IP* адресе, порта и протокола видео преноса, односно *RTP* сервера, али и друге важне информације као што је кодек који се користи.

RTP вишезначни видео пренос

Раније је утврђено да *RTSP* и *RTP* представљају уређени пар серверских утичница за видео пренос, где један без другог не могу. *RTSP* сервер мора да има пар који ће да пласира видео садржај, док *RTP* сервер у већини случаја не постоји или није започео пренос све док овај други није сигнализирао.

Вишезначни видео пренос је, у оквиру овог система, намењен искључиво за пренос уживо, што има смисла, јер живи пренос пласира идентичан садржај ка свим корисницима истовремено. *RTP* вишезначни сервер *H.264 frame* шаље у *RTP* формату. *RTP* пакет поседује максималну величину од 1460 бајта за *payload*, па се велики *frame*-ови шаљу у сегментима. На почетку сваког пакета умиће се заглавље - *RTP Header*, које има 12 *byte*-а у оквиру којих су садржани подаци неопходни за композицију и коришћење.

RTP једнозначни видео пренос

Видео пренос по захтеву је тип преноса који ради у „1 на 1“ односу сервера и клијента. За сваког пристиглог клијента репродукује се раније снимљени видео садржај од почетка. На пример, један корисник може гледати пети минут видео снимка, док други гледа почетак.

Да би тако нешто било изводљиво сервер за видео пренос, након *RTSP* интеракције, мора инстанцирати посебан једнозначни *RTP* сервер, као и читач видео записа који ће видео запис пласирати ка клијенту у оквиру засебне сесије (нити). Важно је напоменути да сваки *RTP* сервер мора поседовати јединствени порт на којем извршава видео пренос по захтеву. Овај део

је веома осетљив, јер овакав подухват приликом повезивања више клијената резултује већом потрошњом *CPU* и *RAM* серверског уређаја.

3.1.2 Мрежни менаџер

Мрежни менаџер је модул који је, у оквиру овог система, задужен за интеракцију са мрежним корисницима, одржавање и приступање бази података, као и заштиту личних података корисника.

Интеракција са мрежним корисницима

Мрежни корисници, попут *RTSP* клијената у интеракцији са сервером, морају испоштовати стриктан протокол односно начин на који комуницирају са менаџером. Протокол не представља неки од познатих стандарда, већ специфичан вид паковања и формирања захтева и одговора који је посебно осмишљен за потребе овог модула.

Комуникација се може поделити у два режима:

1. Иницијални – одвија се приликом корисничког приступања мрежи.
2. Устаљени – одвија се од тренутка када је иницијални део комуникације завршен све до изласка мрежног корисника.

Током иницијалног режима, мрежни корисник може послати захтев за пријаву и/или регистрацију. Уколико је корисник већ регистрован, тај корак прескаче и шаље захтев за пријаву. Мрежни менаџер комуникацију са корисником држи у иницијалном режиму све док захтев за пријаву није успешно обрађен, а потом се прелази у устаљени режим.

У устаљеном режиму комуникације, мрежни менаџер благовремено ажурира корисника о листи постојећих *RTSP* видео преноса. Поред тога, уколико корисник жели да користи неки од понуђених видео преноса по захтеву мрежни менаџера врши проверу заузетости таквих преноса. Број активних корисника видео преноса по захтеву је ограничен на 5.

За евиденцију корисника користи се база података помоћу *MySQL* сервиса. За регистрацију нових корисника ради се унос нових података у базу, а за пријаву читање са исте. Током акција за регистрацију и пријаву корисника кроз мрежу се у форми поруке преносе приватни подаци корисника – корисничко име, *e-mail* адреса и лозинка. У интернет мрежи је могуће те поруке пресрести и видети њихов садржај, чиме се нарушава приватност.

Мрежни менаџер чува корисничке податке преко криптографије – енкрипције података. У питању је *AES* симетрична енкрипција, у оквиру које и мрежни менаџер и корисник знају јавни кључ и вектор иницијализације – *IV*. За реализацију енкрипције података користи се библиотека *Crypto++*. Принцип рада овог подмодула је једноставан – пре слања поруке, корисник или менаџер ураде енкрипцију података, а након пријема исте ураде декрипцију примљеног садржаја.

3.2 Програмско решење клијентске апликације

3.2.1 RTSP клијент

Након што је корисник успешно евидентиран у мрежи, добија листу активних видео преноса, где одабиром неког од њих *RTSP* започиње комуникацију са сервером.

Приликом комуникације са сервером – клијент шаље следеће *RTSP* захтеве: *Options*, *Describe*, *Setup* и на крају *Play*. У оквиру ових порука налазе се, између осталог, и информације као што су *IP* адреса, порт и протокол *RTP* сервера жељеног видео преноса, али и порт који сервер очекује од *RTP* клијента, у случају преноса по захтеву. Након успешно обрађеног *Setup* захтева, могуће је иницијализовати *RTP* клијента, а након успешно обрађеног *Play* захтева, могуће је започети његов рад. Уколико корисник жели прекид преноса шаље се *Teardown RTSP* захтев.

3.2.2 RTP клијент

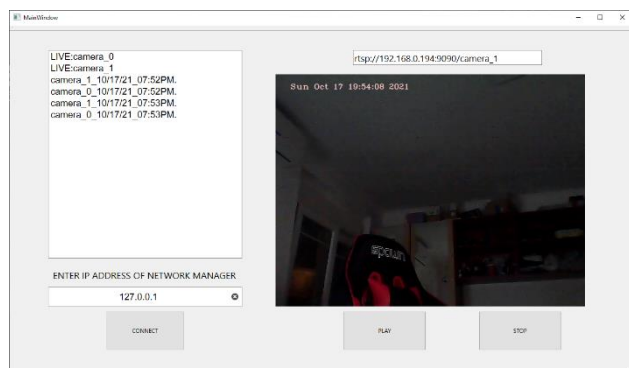
Након што је *RTSP* клијент успешно одрадио свој део посла, овај подмодул започиње примање и колекцију *RTP* пакета, *H.264* декомпресију и приказивање преноса на главном *UI* прозору апликације.

Постоји могућност да декодер у некој итерацији не успе да добије слику. То је честа појава код живог видео преноса јер не може успешно да одради предвиђања. За неуспелу декомпресију се примљени материјал брише и наставља се колекција следећег. Ова појава се дешава при самом почетку репродуковања, где је потребно да се декодер „уигра“, а након првог успешног декодовања видео пренос даље тече у најбољем реду.

3.2.3 Мрежни корисник

Да би корисник могао да гледа било који од видео преноса у оквиру овог система неопходно је да буде евидентиран у мрежи са следећим подацима: *e-mail* адреса, корисничко име и лозинка. При првом приступању у мрежи корисник се региструје, а након тога само пријављује.

Притиском на дугме за повезивање покреће се процес конекције са мрежним менаџером. Уколико је конекција успешна отвара се додатни прозор за пријаву/регистрацију корисника, слика 4.



Слика 4. Успешно приказивање живог видео преноса

Након успешне пријаве, мрежни корисник попуњава табелу на менију главног прозора на основу поруке коју благовремено добија од мрежног менаџера. Интеракција са мрежним менаџером одиграва се у главној нити све до краја иницијалног дела комуникације, док се устаљени режим врши у позадинској нити.

4. ЗАКЉУЧАК

Коначно тестирање показало је да су сви елементи система успешно интегрисани у синхронизовану целину која успешно обавља тражени задатак и обезбеђен је поуздан рад свих модула система.

Предлози за побољшање:

- Омогућити подршку за паузирање и пуштање насумичних делова за видео пренос по захтеву
- Раздвојити модуле серверске апликације у више апликација, тако да се рад подели у више уређаја у систему:
 - Апликација која пласира живи видео пренос са камере и врши записивање видео снимака на самба путањи.
 - Апликација која врши услуживање видео преноса по захтеву.
 - Апликација која врши евиденцију корисника система, управља базом података и преусмерава на видео пренос.
- Имплементирати *Watchdog* сервис, који би поновно покретао неактивне делове система.

5. ЛИТЕРАТУРА

- [1] Henning Schulzrinne, Anup Rao, and Robert Lanphier. *Real time streaming protocol (RTSP)*. (1998).
- [2] Алемпије Вељовић, и Мирољуб Захорјански. *Увод у базе података*. СЕТ, 2014.
- [3] Милојко Јевтовић. *Комуникациони протоколи интернета*. Академска мисао, 2015.
- [4] Richardson, Iain E. *The H. 264 advanced video compression standard*. John Wiley & Sons, 2011.

Кратка биографија:



Јован Славујевић рођен је у Београду 1996. год. Дипломски рад на Факултету техничких наука из области Електротехника и рачунарство – Примењена електроника одбранио је 2020.год.

Области интересовања су му развој софтвера у програмским језицима C++, C# и Java.
контакт: slavujevic.jovan.96@gmail.com