|  |  |
| --- | --- |
|  | Зборник радова Факултета техничких наука, Нови Сад |

**UDK: (Upisuje redakcija, ostaviti ovaj red)**

**DOI: (Upisuje redakcija, ostaviti ovaj red)**

**Реализација трослојног система за видео пренос по *RTSP* протоколу**

**Realization of three-layer *RTSP* video streaming system**

Јован Славујевић, *Факултет техничких наука, Нови Сад*

**Област – ЕЛЕКТРОТЕХНИКА И РАЧУНАРСТВО**

**Кратак садржај –** *У овом чланку је описана 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 размацима од 15 минута. Систем је имплементиран на личној Windows машини са равноправном подршком за Linux оперативни систем.*

**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 Windows personal computer with equally supported Linux OS.*

**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. [1]

У оквиру овог рада реализован је софтверски пакет који обухвата видео пренос – серверску страну, али и његово приказивање – клијентску страну. Ток видео преноса је диктиран по специјалном стандарду видео преноса какав је *RTSP*. Поред тога, приступ видео преносима лимитиран је евидентирањем корисника у бази података и приступањем уз помоћ јединствених корисничких креденцијала.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**НАПОМЕНА:**

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

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 размацима 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. Теоријске основе**

У овом поглављу укратко су описане теоријске основе коришћених технологија у оквиру овог рада.

**2.1 Етернет**

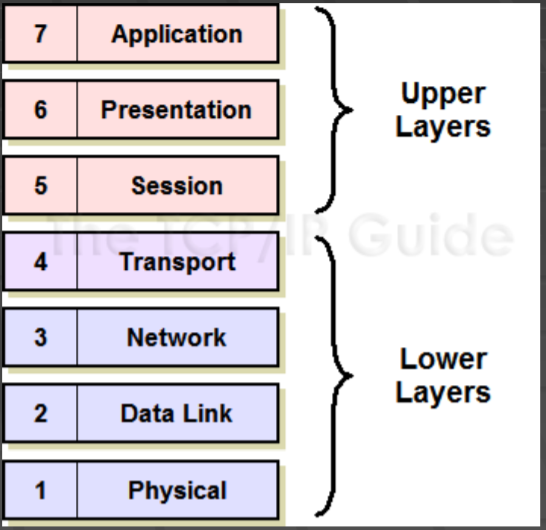
Етернет је најраспрострањенија мрежна технологија која се користи код локалних рачунарских мрежа - *LAN*. Уобичајена битска брзина код преноса података на овој мрежи је 10 *Mbps*, новији стандарди дозвољавају брзине преноса од 100 *Mbps*. У последњим годинама те брзине досежу и величине реда *Gbps*. [2]



Слика 1. Етернет каблови повезани на комутатору [3]

**2.2 *OSI* референтни модел**

*Open System Interconnection* (*OSI*) референтни модел је најкоришћенији апстрактни опис архитектуре мреже. Описује интеракцију уређаја, програма, сервиса и протокола при мрежним комуникацијама. [4] Референтни модел представља све процесе потребне за успешну комуникацију и дели све процесе у логичке групе – слојеве. Овако дизајниран модел се назива слојевита архитектура. [5]



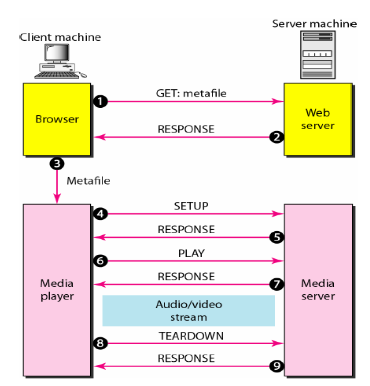
Слика . Илустрација OSI нивоа [6]

**2.3 *IP* протокол**

Интернет протокол - *IP* је протокол трећег слоја *OSI* референтног модела (слоја мреже). Садржи информације о адресирању, чиме се постиже да сваки мрежни уређај који је повезан на интернет има јединствену адресу и може се лако идентификовати у целој интернет мрежи. Такође, садржи контролне информације које омогућавају пакетима да буду рутирани на основу познатих *IP* адреса. [7]

**2.4 *RTSP* протокол**

*Real-Time Streaming Protocol* – *RTSP* је протокол апликативног нивоа намењен контроли доставе података са *real-time* својствима као што су аудио и видео. *RTSP* остварује и контролише један или више временски синхронизованих токова континуираних података, не захтевајући конекцију. Сервер одржава сесију која није никако зависна од веза преносних слојева као што је, на пример, *TCP* веза. Клијент зато може да користи и протокол без везе (енг. *Connectionless*) као што је *UDP*. Токови података контролисаних од стране *RTSP*-а могу користити *RTP* али функције које обавља *RTSP* не зависе од транспортног механизма који се користи за пренос података. [8]



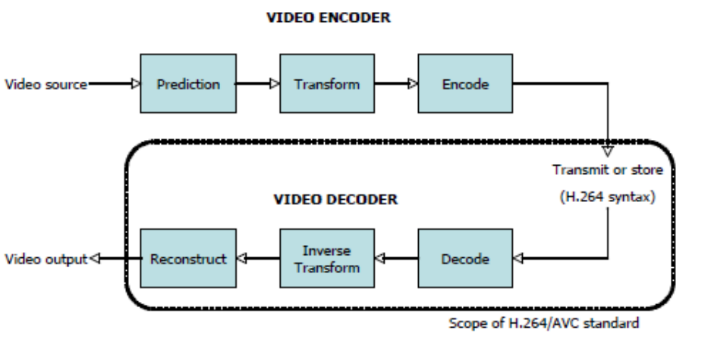
Слика . Блок шема RTSP протокола [9]

**2.5 Кодек**

Кодек је алат за кодирање који обрађује видео и складишти га у низу бајтова. Овај термин води порекло од скраћенице за кодер – декодер, односно компресор – декомпресор. Добро је познато да се приликом чувања било код аудио или видео садржаја захтева значајна количина меморије рачунара па се јавља потреба да такви типови садржаја буду на неки начин смањени, како би заузели што мању количину простора. Управо у те сврхе користи се кодек. [9]

**2.5.1 *H.264***

*H.264* је индустријски стандард за видео компресију, процес претварања дигиталног видеа у формат који заузима мањи капацитет када се складишти или преноси. Представља најпопуларнији избор кодека, јер може користити компресију и са губитком и без губитака у зависности од поставки које су одабране приликом кодирања, као што су брзина кадрова, квалитет и циљна величина датотеке.



Слика . Процес H.264 кодирања и декодирања [10]

**2.6 База података**

База података је скуп међусобно повезаних података, похрањених у спољњој меморији рачунара. Подаци су истовремено доступни разним корисницима и апликационим програмима. Убацивање, промена, брисање и читање података обавља се посредством заједничког софтвера. [11]

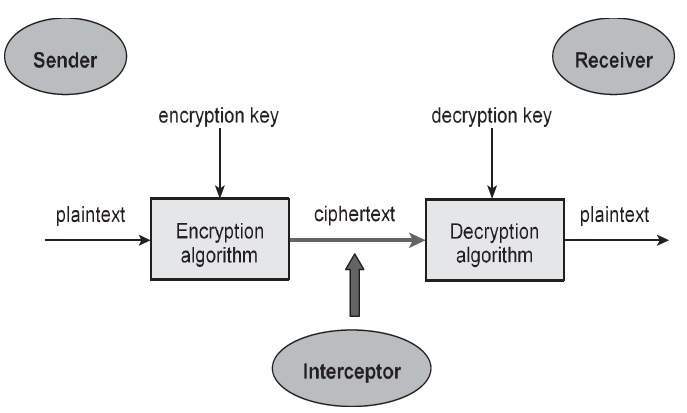
Систем за управљање базом података – *DBMS* је сервер базе података. То је софтверски систем који омогућава креирање, дефинисање, коришћење, одржавање и контролу приступа бази података. *DBMS* омогућава крајњим корисницима или програмерима да деле податке, односно пружа могућност да се подаци истовремено користе од стране више апликација и лишава нас потребе да свака апликација има своју копију података. [12]

**2.6.1 *SQL***

*Structure Query Language (SQL)* је релациони упитни језик. Релације се креирају једном наредбом и одмах су доступне, што га чини једноставним за коришћење. Униформан је, јер се сви подаци и резултати операције приказују у виду табела и омогућавају интерактивно и класично програмирање. У основи, то је језик који омогућава комуникацију са базама података и управљањем свих података које оне садрже. [13]

**2.7 Криптографија**

Криптографија је наука која се бави методама очувања тајности информација. Када се личне, финансијске, војне или информације државне безбедности преносе са места на место, оне постају рањиве на прислушкивачке тактике. Овакви проблеми се могу избећи криптовањем информација које их чине недоступним нежељеној страни. Шифра и дигитални потпис су криптографске технике које се користе да би се имплементирали безбедносни сервиси. Основни елемент који се користи назива се шифарски систем или алгоритам шифровања. Сваки шифарски систем обухвата пар трансформација података, које се називају шифровање и дешифровање. [14]



Слика . Блок шема криптографије [15]

**3. Концепт решења**

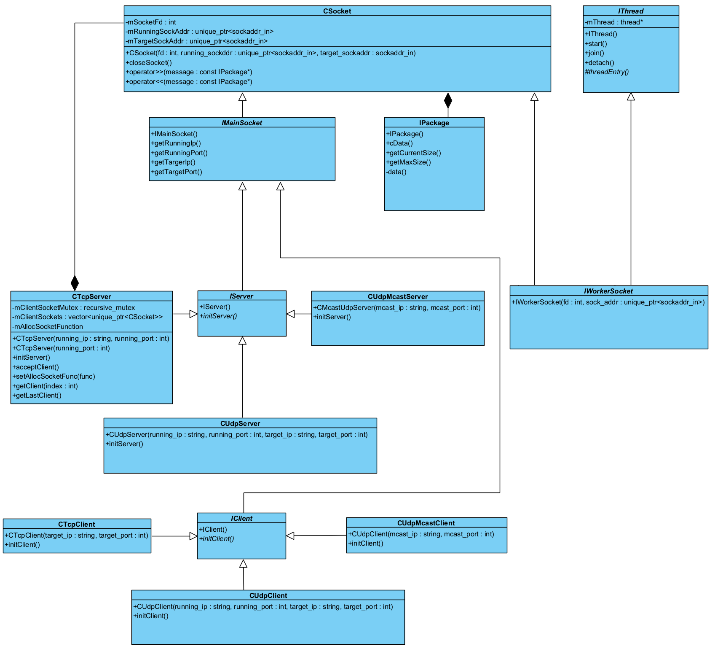
У овом поглављу описан је и илустрован концепт решења, идеја пројекта, као и архитектура и функционалност појединачних модула система.

**3.1 Сокет библиотека у објектно оријентисаној форми**

Сокет интерфејс за програмирање апликације (енг. *Application Programming Interface*) - *API* 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, веома робустно 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.

У оквиру интерфејса постоје 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*.



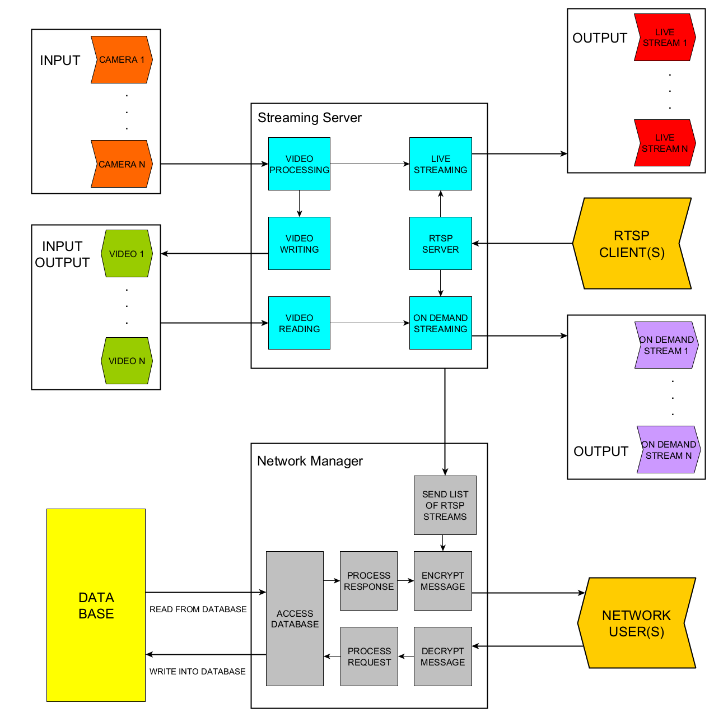
Слика . UML класни дијаграм Socket Wrapper библиотеке

Поред наведених класа 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*) чистих (*pure*) виртуалних метода које заправо представљају тзв. *Getter* методе за максималну и искоришћену величину, као и стартну адресу поруке.

**3.2 Архитектура и улога серверске апликације**

Серверска апликација обавља велики број задатака, али се може груписати у 2 велика целине - модула:

* Сервер за видео пренос – модул који је задужен за целокупну видео обраду, интеракцију са клијентима и услуживање видео преноса.
* Мрежни менаџер – модул који је задужен за комуникацију са мрежним корисницима, одржавање базе података, као и заштиту личних података корисника.

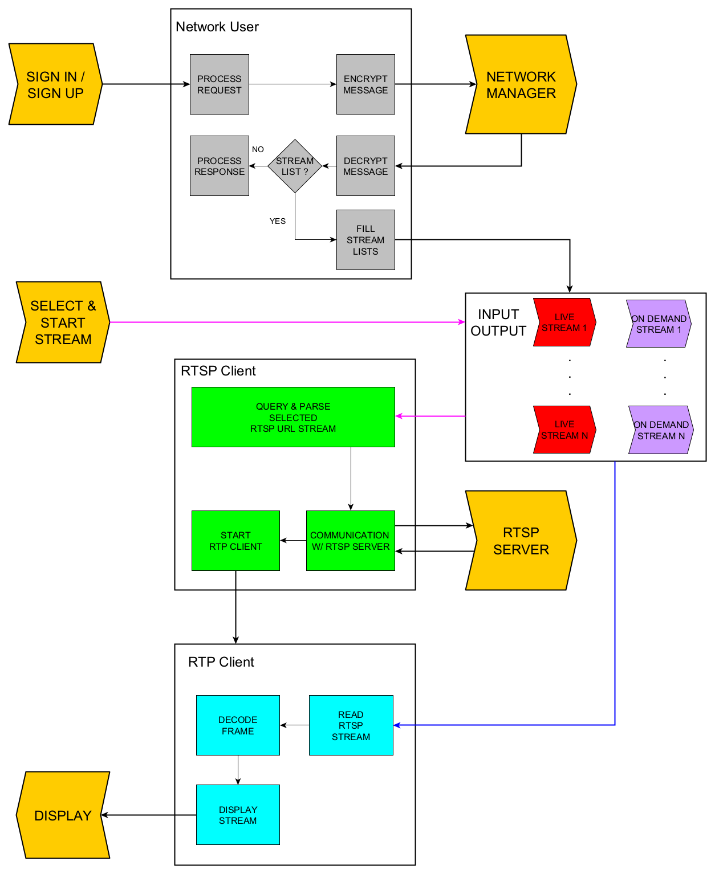


Слика . Идејна блок шема серверске апликације

**3.2 Архитектура и улога клијентске апликације**

У оквиру овог пројекта, у функционалном смислу, клијентска апликације може се поделити у 3 веће целине - модула:

* *RTSP* клијент – модул који комуницира са *RTSP* сервером, потражује живи или пренос по захтеву, те га прекида или омогућава прелазак на други.
* *RTP* клијент – модул који се повезује на видео пренос, врши пријем, обраду и приказивање видео преноса.
* Мрежни корисник – модул који се повезује са мрежним менаџером, омогућава евидентирање у систему и добавља листу активних видео преноса.



Слика . Идејна блок шема клијентске апликације

**4. Програмско решење**

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

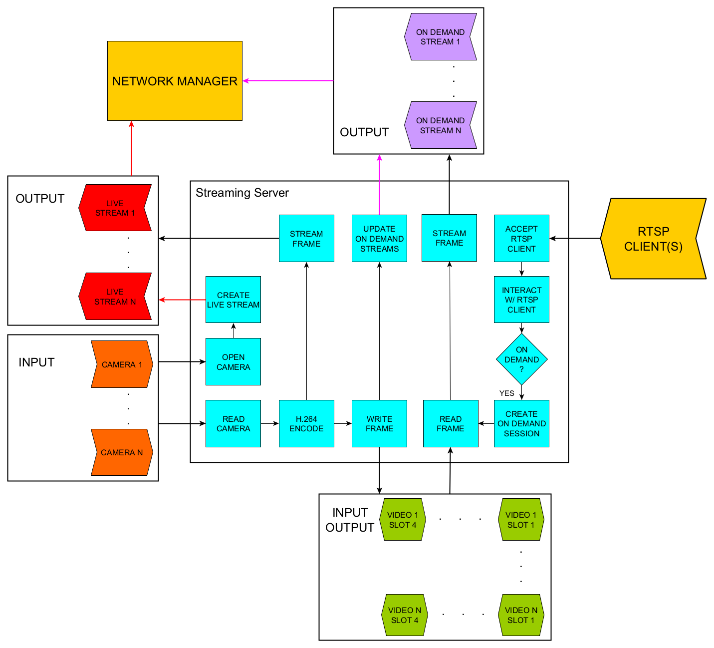
**4.1 Програмско решење серверске апликације**

На основу концепта решења са прошлог поглавља, серверска апликација је у оквиру овог система подељена у два главна модула:

* Сервер за видео пренос
* Мрежни менаџер

**4.1.1 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 приказана је детаљнија блок шема сервера за видео пренос.



Слика . Детаљна блок шема сервера за видео пренос

**Читање са камере и видео записивање**

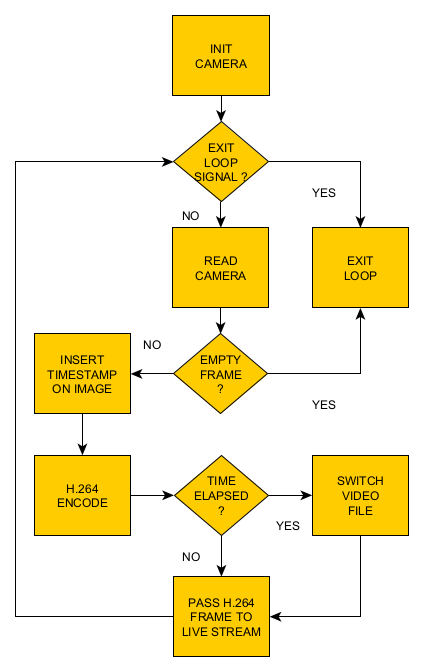
Серверска апликација при покретању зна којим бројем камера ће управљати. Да би управљала камером, апликација користи објекте из *OpenCV* библиотеке. Након што је камера успешно повезана, креира се и покреће посебна нит у којој се извршава петља читања све до краја рада серверске апликације или физичког 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 основих 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* 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 видеа је конципирано 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 другој нити која има за улогу да одбројава протекло време и обавести другу нит да је време истекло, те да је потребно пребацити се 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 префлека видео фајл 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.



Слика . Алгоритам рада подмодула за читање и видео записивање

***RTSP* интеракција са клијентима**

Сервер за видео пренос пружа пренос искључиво по *RTSP* протоколу. Протокол захтева од корисника да, поред сокета за добављање видео садржаја (*RTP*), поседује и сокет 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 сокета 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 умеће се заглавље - *RTP Header*, које има 12 *byte*-a 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.

**4.1.2 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:

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

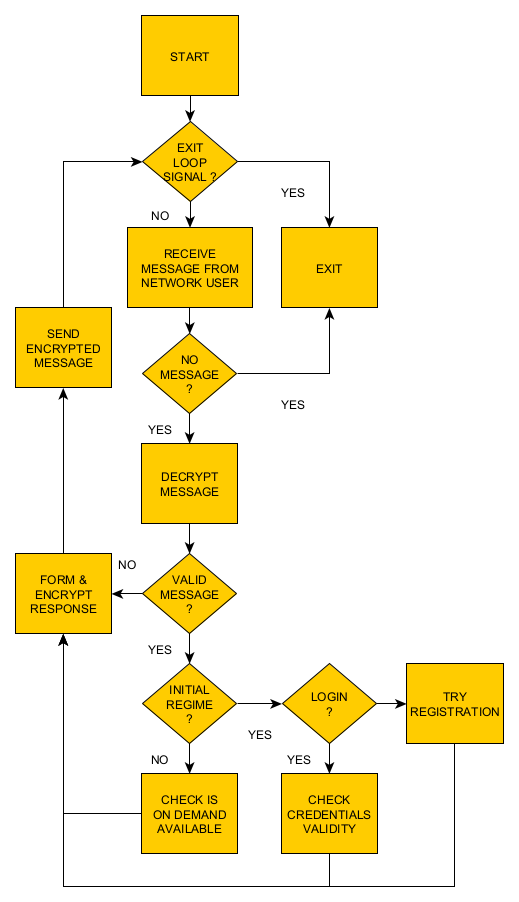
Током иницијалног режима, мрежни корисник може послати захтев за пријаву и/или регистрацију. Уколико је корисник већ регистрован, тај корак прескаче и шаље захтев за пријаву. Мрежни менаџер комуникацију са корисником држи у иницијалном режиму све док захтев за пријаву није успешно обрађен, а потом се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

**Читање и писање у бази података**

За сервер базе података користи се *MySQL*, а за приступање бази користи се њен *API*, који захтеве мрежног менаџера шаље у форми која је идентична командама које администратор може унети путем терминала (*Command Prompt* или други).

За пријаву корисника потребно је извршити упит да ли се за прослеђено корисничко име или имејл поклапа послата у бази, док је за регистрацију потребно уписати нове податке у базу, али се пре тога проверава да ли у бази већ постоји идентично корисничко име и/или *e-mail* адреса.



Слика . Алгоритам рада једне инстанце послужитеља мрежног корисника

**Криптографија порука**

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.

**4.2 Програмско решење клијентске апликације**

На основу концепта решења са прошлог поглавља, клијентска апликација је подељена у три модула:

* *RTSP* клијент
* *RTP* клијент
* Мрежни корисник

Клијентска апликација је много једноставније реализована у односу на серверску, што је и логично, јер је њен систем знатно мање комплекснији.

**4.2.1 *RTSP* клијент**

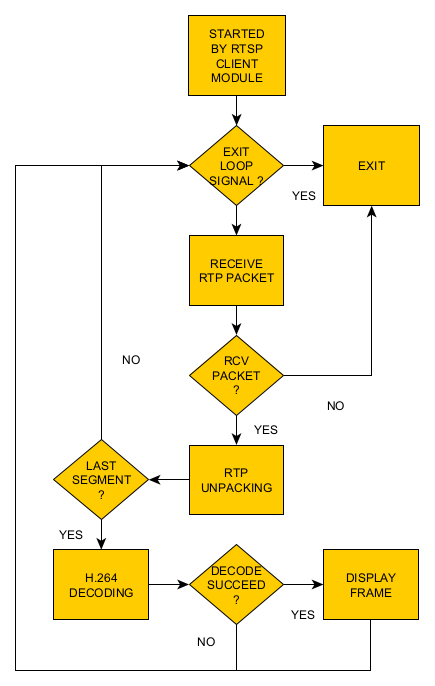
Након што је корисник успешно евидентиран у мрежи, добија листу активних видео преноса, где одабиром неког од њих *RTSP* започиње свој комуникацију са сервером.

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.

**4.2.2 *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.



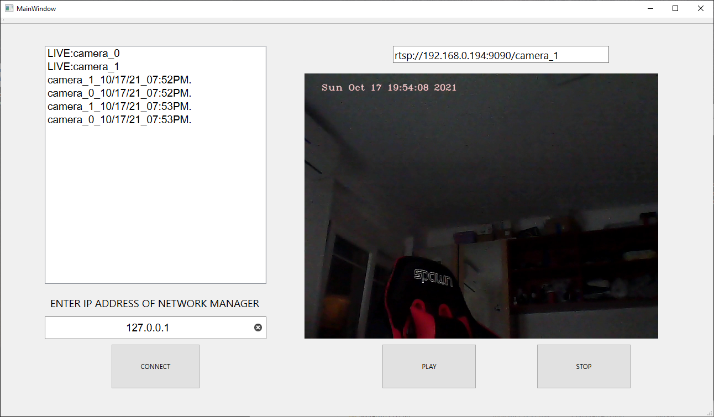
Слика . Алгоритам рада RTP клијента

**4.2.3 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.

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.



Слика 13. Успешно приказивање живог видео преноса

**5. 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:

* Проширити решење за *Web* подршку, односно имплементирати *Web* верзију *RTSP* клијента.
* Омогућити подршку за паузирање и пуштање насумичних делова за видео пренос по захтеву
* Имплементирати сервис за дијагностику, који би проверавао потрошњу коју изазива серверска апликација и благовремено зауставио неки од модула или ограничио његове ресурсе.
* Раздвојити модуле серверске апликације у више апликација, тако да се рад подели у више уређаја у систему:
  + Aпликација која пласира живи видео пренос са камере и врши записивање видео снимака на самба путањи.
  + Aпликација која врши услуживање видео преноса по захтеву.
  + Aпликација која врши евиденцију корисника система, управља базом података и преусмерава на видео пренос.
* Имплементирати *Watchdog* сервис, који би поновно покретао „мртве“ делове система.
* Лепши изглед корисничког интерфејса код клијентске апликације.

**6. Литература**

[1] [https://sites.google.com/site/izradasajtovatippingpoint/livestream  
ing---direktni-internet-video-prenosi](https://sites.google.com/site/izradasajtovatippingpoint/livestreaming---direktni-internet-video-prenosi) , *Livestreaming*, страница *izradasajtova.rs*, приступљено: октобар 2021.

[2] <http://es.elfak.ni.ac.rs/rmif/Prenos-podatak-februar-2011/Pre.-pod-%202010/Pdf-2010/Pogl-09-Ethernet%20%28210-225%29.pdf> , Етернет, Истраживачки центар за ембедед системе, Електронски факултет, Универзитет у Нишу, приступљено: октобар 2021.

[3] <https://sr.wikipedia.org/sr-ec/Етернет> , Етернет, *Wikipedia* чланак, приступљено: октобар 2021.

[4] <https://sr.wikipedia.org/sr-ec/ОСИ_модел> , ОСИ модел, *Wikipedia* чланак, приступљено: октобар 2021.

[5] [http://etsntesla.edu.rs/SAJT%20EC3/racmreze/predavanja/Modeli\_  
umrezavanja.pdf](http://etsntesla.edu.rs/SAJT%20EC3/racmreze/predavanja/Modeli_umrezavanja.pdf) , Модели умрежавања, ЕТШ „Н. Тесла“, Панчево, приступљено: октобар 2021.

[6] [https://www.bpa.edu.rs/FileDownload?filename=5f7860cc-8f2f-47ca-badd-cbf59eb5d92c.pdf&originalName=9.%20RM\_Mrezni%  
20standardi.pdf](https://www.bpa.edu.rs/FileDownload?filename=5f7860cc-8f2f-47ca-badd-cbf59eb5d92c.pdf&originalName=9.%20RM_Mrezni%20standardi.pdf) , Рачунарске мреже, др. Душан Љубичић, Београдска академија пословних и уметничких струковних студија, приступљено: октобар 2021.

[7] <https://sr.wikipedia.org/wiki/Интернет_протокол> , Интернет протокол, *Wikipedia* чланак, приступљено: октобар 2021.

[8] <https://sr.wikipedia.org/sr-ec/Интернет_телефонија> , Интернет телефонија, *Wikipedia* чланак, приступљено: октобар 2021.

[9] <https://pdfcoffee.com/rtp-rtcp-rtsp-3-pdf-free.html> , *RTP-RTCP-RTSP* презентација, Факултет електротехнике у Тузли, С. Дудић, А. Хукић и Ј. Бешић, приступљено: октобар 2021.

[10] <https://www.vcodex.com/an-overview-of-h264-advanced-video-coding/> , интернет чланак о *H.264* видео кодеку, страница *VCodex*, приступљено: октобар 2021.

[11] [http://msts-travnik.net/wp-content/uploads/2013/10/baze\_  
podataka.pdf](http://msts-travnik.net/wp-content/uploads/2013/10/baze_podataka.pdf) , БАЗЕ ПОДАТАКА, МСТШ Травник, приступљено: октобар 2021.

[12] <https://kompjuteras.com/baze-podataka/> , интернет чланак о бази података, страница ИТ блог Компјутераш, приступљено: октобар 2021.

[13] <https://sr.wikipedia.org/sr/SQL> , *SQL*, *Wikipedia* чланак, приступљено: октобар 2021.

[14] <https://sh.wikipedia.org/wiki/Kriptografija> , Криптографија, *Wikipedia* чланак, приступљено: октобар 2021.

[15] <https://www.tutorialspoint.com/cryptography/cryptosystems.htm> , интернет чланак о крипто системима, страница *Tutorials point*, приступљено: октобар 2021.

**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 |