



УНИВЕРЗИТЕТ У НОВОМ САДУ
ФАКУЛТЕТ ТЕХНИЧКИХ НАУКА У
НОВОМ САДУ



Небојша Раблов

Имплементација РТСП протокола за дигиталне телевизијске пријемнике

ДИПЛОМСКИ РАД
- Основне академске студије -

Нови Сад, 2018.



КЉУЧНА ДОКУМЕНТАЦИЈСКА ИНФОРМАЦИЈА

Редни број, РБР:			
Идентификациони број, ИБР:			
Тип документације, ТД:	Монографска документација		
Тип записа, ТЗ:	Текстуални штампани материјал		
Врста рада, ВР:	Завршни (Bachelor) рад		
Аутор, АУ:	Небојша Раблов		
Ментор, МН:	Проф. Др Илија Башичевић		
Наслов рада, НР:	Имплементација РТСП протокола за дигиталне телевизијске пријемнике		
Језик публикације, ЈП:	Српски / латиница		
Језик извода, ЈИ:	Српски		
Земља публиковања, ЗП:	Република Србија		
Уже географско подручје, УГП:	Војводина		
Година, ГО:	2018		
Издавач, ИЗ:	Ауторски репринт		
Место и адреса, МА:	Нови Сад; трг Доситеја Обрадовића 6		
Физички опис рада, ФО: (поглавља/страна/цитата/табела/ слика/графика/прилога)	7/27/0/3/8/0/0		
Научна област, НО:	Електротехника и рачунарство		
Научна дисциплина, НД:	Рачунарска техника		
Предметна одредница/Кључне речи, ПО:	Дигитална телевизија, ИПТВ, РТСП протокол		
УДК			
Чува се, ЧУ:	У библиотеци Факултета техничких наука, Нови Сад		
Важна напомена, ВН:			
Извод, ИЗ:	У овом раду описано је једно решење имплементације РТСП протокола за дигиталне телевизијске уређаје. Задатак је био написати библиотеку за управљање РТСП захтевима, а затим њена интеграција у систем дигиталног ТВ пријемника заснованог на Андроид платформи. Овим решењем проширује се ДТВ систем новим могућностима за пријем ТВ садржаја. Тестирања и верификација исправности решења рађена су коришћењем Андроид СТБ уређаја и САТ>ИП сервера за дистрибуцију ДВБ сигнала кроз ИП мрежу.		
Датум прихватања теме, ДП:			
Датум одбране, ДО:			
Чланови комисије, КО:	Председник:	Проф. Др Небојша Пјевалица	
	Члан:	Проф. Др Иштван Пап	Потпис ментора
	Члан, ментор:	Проф. Др Илија Башичевић	



KEY WORDS DOCUMENTATION

Accession number, ANO:		
Identification number, INO:		
Document type, DT:	Monographic publication	
Type of record, TR:	Textual printed material	
Contents code, CC:	Bachelor Thesis	
Author, AU:	Nebojša Rablov	
Mentor, MN:	Ilija Bašičević PhD	
Title, TI:	Implementation of RTSP protocol for set top box devices	
Language of text, LT:	Serbian	
Language of abstract, LA:	Serbian	
Country of publication, CP:	Republic of Serbia	
Locality of publication, LP:	Vojvodina	
Publication year, PY:	2018	
Publisher, PB:	Author's reprint	
Publication place, PP:	Novi Sad, Dositeja Obradovica sq. 6	
Physical description, PD: (chapters/pages/ref./tables/pictures/graphs/appendices)	7/27/0/3/8/0/0	
Scientific field, SF:	Electrical Engineering	
Scientific discipline, SD:	Computer Engineering, Engineering of Computer Based Systems	
Subject/Key words, S/KW:	Digital television, IPTV, RTSP protocol	
UC		
Holding data, HD:	The Library of Faculty of Technical Sciences, Novi Sad, Serbia	
Note, N:		
Abstract, AB:	In this thesis work, one implementation of RTSP protocol for set top box devices is described. The aim was to write a library which manages RTS requests, and then to integrate it into an Android-based digital TV system. This solution extends the DTV system with new possibilities for receiving TV content. Verification of the solution has been done using Android STB device and SAT>IP server.	
Accepted by the Scientific Board on, ASB:		
Defended on, DE:		
Defended Board, DB:	President:	Nebojša Pjevalica PhD
	Member:	Ištván Pap PhD
	Member, Mentor:	Ilija Bašičević PhD
		Menthor's sign

Zahvalnost

Ovom prilikom bih se zahvalio mentoru dr Iliji Bašičević na savetima i stručnoj pomoći.

Zahvaljujem se RT-RK institutu na pruženoj mogućnosti za realizaciju ovog rada, kao i kolegama iz tima.

Posebno se zahvaljujem Ačanski Milanu i Marković Radovanu na stručnoj pomoći, korisnim savetima, izdvojenom vremenu i strpljenju.

Na kraju, želim da se zahvalim porodici i prijateljima na motivaciji i neiscrpnoj podršci.

SADRŽAJ

1.	Uvod.....	1
2.	Teorijske osnove	3
2.1	Digitalna televizija	3
2.2	IPTV	4
2.3	RTSP	5
2.3.1	RTP i RTCP	5
2.3.2	RTSP operacije	6
2.3.3	Primer razmene RTSP poruka	7
3.	Opis rešenja.....	10
3.1	Opis ciljne platforme.....	10
3.1.1	SAT>IP server	10
3.1.2	DTV STB	12
3.1.3	Srednji sloj programske podrške ciljne platforme (iWedia Comedia)	13
3.2	RTSP biblioteka	14
3.3	Integracija u HAL nivo srednjeg sloja	15
4.	Programsko rešenje.....	17
4.1	RTSP biblioteka (libRtsp)	17
4.1.1	Modul za rukovanje RTSP funkcionalnostima.....	17
4.1.1.1	Funkcije RTSP modula	19
4.1.2	Modul za rukovanje TCP konekcijom.....	21
5.	Testiranje i verifikacija	22
5.1	Ispitivanje RTSP biblioteke	22
5.2	Funkcionalno testiranje na ciljnoj platformi	23

6. Zaključak	26
7. Literatura.....	27

SPISAK SLIKA

Slika 2-1 Pregled rasprostranjenosti DTV standarda	3
Slika 2-2 Primer razmene RTSP poruka sa prenosom RTP i RTCP podataka	7
Slika 3-1 SAT>IP server	11
Slika 3-2 Android Set-Top Box	12
Slika 3-3 Pozicija srednjeg sloja u okviru programskog steak sistema	13
Slika 3-4 Arhitektura Comedia srednjeg sloja	14
Slika 3-5 Prikaz organizacije podmodula u okviru RTSP biblioteke	15
Slika 3-6 Prikaz integracije realizovane RTSP biblioteke	16

SPISAK TABELA

Tabela 1 Pregled testova za verifikaciju ispravnosti RTSP biblioteke	22
Tabela 2 Pregled testova za funkcionalno ispitivanje na ciljanoj platformi	23
Tabela 3 Dobijeni rezultati testiranja na ciljnoj platformi	24

SKRAĆENICE

DTV	- <i>Digital television</i> , Digitalna televizija
IPTV	- <i>Internet protocol television</i> , Televizija preko interneta
RTSP	- <i>Real-Time Streaming Protocol</i> , Protokol za prenos u realnom vremenu
TCP	- <i>Transmission Control Protocol</i> , Protokol za kontrolu transmisije
UDP	- <i>User Datagram Protocol</i> , protokol korisničkih datagrama
HLS	- <i>HTTP Live Streaming</i> , Protokol za prenos uživo preko HTTP
DASH	- <i>Dynamic Adaptive Streaming over HTTP</i> , Protokol za dinamički adaptivni prenos uživo preko HTTP
SDP	- <i>Session Description Protocol</i> , Protokol za opisivanje multimedijalnih tokova
MHEG	- <i>Multimedia and Hypermedia Expert Group</i> , Standard za prezentaciju multimedijalnih informacija

1. Uvod

Danas je televizija jedna od tehnologija koja se najbrže razvija. Kao i svaka tehnologija i televizija je prošla kroz svoj proces razvoja i unapređenja. Razvoj televizije kroz istoriju se može podeliti u nekoliko etapa i faza, kao što su: način emitovanja, tehnike emitovanja slike, prenosnog puta, pa sve do uske povezanosti sa razvojem televizijskih prijemnika koji su se prilagođavali zahtevima tržišta.

Najveću ekspanziju u dosadašnjem razvoju televizije doživeo je način prenosa signala. Te promene se ogledaju u tranziciji sa prvobitnih, analognih na digitalne signale. Kao i svaka nova napredna tehnologija i digitalna televizija je sa sobom donela niz prednosti i otvorila put novim mogućnostima za dalji razvoj. Smenom analogne televizije javljaju se problemi na tržištu, prvenstveno kod korisnika čiji prijemnik nije mogao da obradi digitalni signal. Zbog tog problema televizijska industrija je plasirala STB (set-top-box) uređaje na tržište koji su datim korisnicima omogućili pogodnosti koje digitalna televizija pruža[1].

Ono što digitalna televizija omogućava, a predstavlja ujedno i najveći zahtev tržišta, je mogućnost gledanja televizije visoke rezolucije. Ova tehnologija obezbeđuje bolji kvalitet slike, a samim tim i bolje iskustvo pri korišćenju televizijskih usluga, što najviše doprinosi popularnosti ove tehnologije.

Razvoj višenamenskih uređaja (kao što su tableti, pametni telefoni, pametni televizori) je postavio nove zahteve na tržištu, ukazujući na potrebu za prenosom televizijskog sadržaja na više različitih uređaja u okviru kuće, zgrade, hotela, aerodroma, bolnice itd. Instalacija klasične mreže za prenos multimedijalnog sadržaja zahteva postavku koaksijalnih kablova do svakog prijemnika, a u slučaju prenosnih uređaja u potpunosti onemogućuje prijem satelitskog, kablovskog ili zemaljskog signala. Jedno rešenje ovog problema jeste premeštanje birača kanala i demodulatora iz prijemnika televizijskog sadržaja u servere čija je uloga u sistemu da

televizijski signal pretvori u oblik pogodan za prenos preko IP mreže. Prenos digitalnog televizijskog sadržaja obavlja se preko postojeće IP mreže. Upravljanje biračem kanala i demodulatorima je omogućeno preko RTSP protokola.

U ovom radu predstavljena je realizacija RTSP za digitalne televizijske uređaje i sastoji se od dva dela. Prvi deo odnosi se na pisanje statičke biblioteke koja implementira generičke funkcionalnosti RTSP protokola i obavlja uspostavu veze i komunikaciju između klijentske i serverske strane. Drugi deo zadatka obuhvata integraciju RTSP biblioteke u okviru Comedia srednjeg sloja čije se funkcionalnosti izvršavaju u okviru android aplikacije Live Channels.

Android aplikacija preko srednjeg sloja dobavlja parametre za reprodukciju video i audio toka za traženi kanal. Srednji sloj komunikaciju s IPTV serverom obavlja pomoću RTSP protokola, koristeći parametre zadate u listi kanala. Nakon što IPTV server obavi zaključavanje na zadatu frekvenciju kanala, vraća sistemu port na kojem počinje emitovanje televizijskog programa kroz IP mrežu.

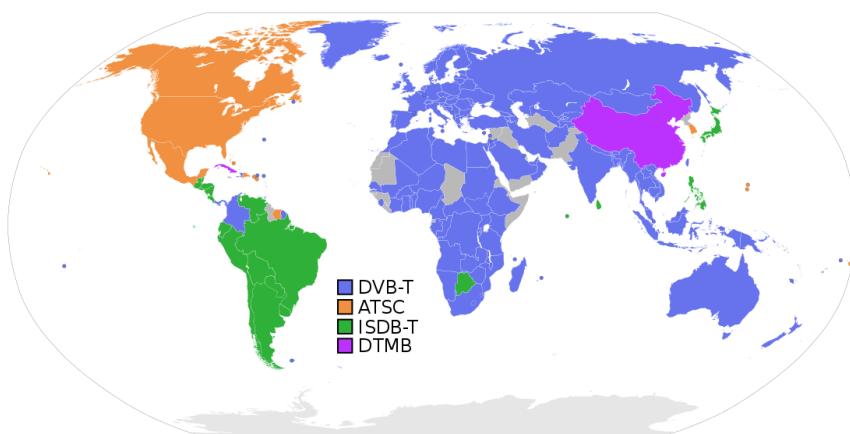
Rad se sastoji od sedam poglavlja, gde je u prvom dat kratak pregled sadržaja samog rada, datog problema i njegovog rešenja. U drugom poglavlju date su opšte teorijske osnove o pojmovima i tehnologijama korišćenim tokom realizacije rada, a sve u svrhu boljeg i lakšeg razumevanja ovog rešenja. U trećem poglavlju predstavljen je koncept idejnog rešenja. Opis programskog rešenja dat je u četvrtom poglavlju. Peto poglavlje daje uvid u rezultate testiranja i verifikacije rešenja. U šestom poglavlju je ukratko sumirano sta je urađeno u okviru rada kao i predlog za dalje unapređivanje i proširivanje. Korišćena literatura je data u okviru sedmog poglavlja.

2. Teorijske osnove

2.1 Digitalna televizija

Danas je televizija jedna od tehnologija koja se najbrže razvija. Kao i svaka tehnologija i televizija je prošla kroz svoj proces razvoja i unapređenja. Razvoj televizije kroz istoriju se može podeliti u nekoliko etapa i faza, kao što su: način emitovanja, tehnike emitovanja slike, prenosnog puta, pa sve do uske povezanosti sa razvojem televizijskih prijemnika koji su se prilagođavali zahtevima tržišta.

Digitalna televizija (DTV) predstavlja oblast potrošačke elektronike koja se bavi uređajima čija je primarna namena prijem TV signala i prezentacija video i audio sadržaja, koja treba da zadovolji mnoge tradicionalne i istorijske zahteve, kako samog tržišta tako i propisanih standarda. I pored grubih okvira ove delatnosti, u ovoj oblasti postoji dosta prostora za inovacije i uvođenje novih tehnologija u navike korisnika.



Slika 2-1 Pregled rasprostranjenosti DTV standarda

Prelazak sa analognog na digitalni TV signal je posledica potrebe za poboljšanjem kvaliteta TV slike. Danas u svetu postoje četiri različita standarda za emitovanje digitalnog TV signala DVB (Digital Video Broadcasting), ATSC (Advanced Television Systems Committee), ISDB (Integrated Services Digital Broadcasting) i DTMB (Digital Terrestrial Multimedia Broadcast).

Prednosti prelaska sa analognog na digitalni TV signal su kvalitetnija slika i zvuk, video unazad, video na zahtev, itd. Najznačajnija prednost prelaska jeste da prenos digitalnih kanala zahteva manji propusni opseg u frekventnom spektru, tako da se količina prenesenih podataka višestruko povećava. Prenos veće količine podataka je bitan za emitovanje multimedijalnog sadržaja visokog kvaliteta (HD, Full HD, Ultra HD), a takođe omogućava i emitovanje više digitalnih kanala na istom frekvencijskom opsegu.

2.2 IPTV

Tradicionalni način emitovanja televizijskog sadržaja podrazumeva korišćenje jednosmernog prenosa od distributera ka korisnicima, korišćenjem antena (satelitskih ili zemaljskih) ili koaksijalnim kablovima u slučaju kablovnih operatera. Rasprostranjenost dvosmerne komunikacije u vidu Internet veze je dovela do postavljanja novih i složenijih zahteva krajinjih korisnika koji se ne mogu adekvatno ispratiti tradicionalnim prenosom televizijskog sadržaja.

Digitalni multimedijalni sadržaj je precizno, vremenski određen, kontinualni tok podataka koji se obično prenosi putem mreža, kod kojih se svaki signal prenosi putem kanala koji je namenjen za prenos audio i video sadržaja. IP mreže služe za prenos širokog skupa podataka sa velikog broja izvora putem jednog kanala, što znači da je Internet u najboljem slučaju slabo vremenski određen sistem koji je podeljen u diskretne pakete, što dovodi do zaključka da IP ne predstavlja idealno rešenje za prenos televizijskog signal[4].

Uprkos činjenici da IP i prenos video sadržaja nisu kompatibilni, tržište za prenos televizijskog sadržaja je u ekspanziji iz sledećih razloga [2]:

1. Internet je široko rasprostranjen, što omogućava operaterima da pružaju usluge bez potrebe da projektuju i instaliraju vlastitu, namensku mrežnu infrastrukturu
2. IP pojednostavljuje unapredavanje i uvođenje novih servisa u ponudu kao što su: Video na Zahtev (eng. VoD – Video on Demand) i reklame koje se prikazuju određenim ciljnim grupama korisnika.

3. Cena formiranja i održavanja IP mreža je u konstantnom padu zbog dostupnosti opreme velikog broja proizvođača i njihove međusobne konkurentnosti i postojanja standarda koji važi na svetskom nivou
4. Broj korisnika IP se ubrzano povećava
5. IP je tehnologija koja ima širok spektar primene kao što su elektronska pošta, rad sa bankama, pristup internet stranicama, deljenje sadržaja.

Za razliku od klasične televizije, IPTV nudi korisnicima širok spektar servisa:

1. elektronski programski vodič (eng. Electronic Program Guide – EPG),
2. digitalnu televiziju
3. “Plati pa gledaj” (eng. Pay Per View- PPV),
4. video na zahtev (eng. Video on Demand - VoD),
5. integrisane TV servise (portali, pauziranje TV programa uživo, digitalna muzika, roditeljska kontrola)
6. mrežni personalni video snimač sadržaja (eng. Network Personal Video Recorder – N-PVR).

2.3 RTSP

Protokol za slanje bitskog toka (RTSP) je kontrolni protokol za dostavljanje multimedijalnog sadržaja preko IP mreže. Klijentska aplikacija koristi RTSP da bi dostavila serveru informacije o multimedijalnim datotekama koje potražuje, tipu aplikacije koju klijent koristi, mehanizme za dostavljanje datoteka (unicast, multicast) i važnih kontrolnih komandi kao što su DESCRIBE, SETUP i PLAY. RTSP protokol je baziran na TCP protokolu u pogledu pouzdanosti prenosa, a takođe ima veoma slične funkcionalnosti i sintaksu kao HTTP protokol (Hypertext Transfer Protocol).

RTSP koristi kombinovani sistem prenosa. Za kontrolu koristi TCP, a za dostavljanje sadržaja se koristi UDP. Pod ovim se smatra da prenos datoteka i prikazivanje audio i video sadržaja na klijentskoj strani može da počne pre nego što je kompletna datoteka pristigla.

2.3.1 RTP i RTCP

Protokol za transport u realnom vremenu RTP je protokol koji se koristi za prenos audio i video sadržaja u realnom vremenu. Kako je dostavljanje audio i video sadržaja osetljivo na kašnjenje, koristi se jednostavniji UDP protokol kao mehanizam za transport paketa na transportnom nivou OSI modela, dok se TCP koristiti u okruženjima koja su podložna velikim gubicima paketa.

Tok RTP protokola za dostavljanje sadržaja je jednosmeran od servera ka klijentu. Protokolom je definisano da je port koji server koristi za slanje podataka uvek paran broj, iako se

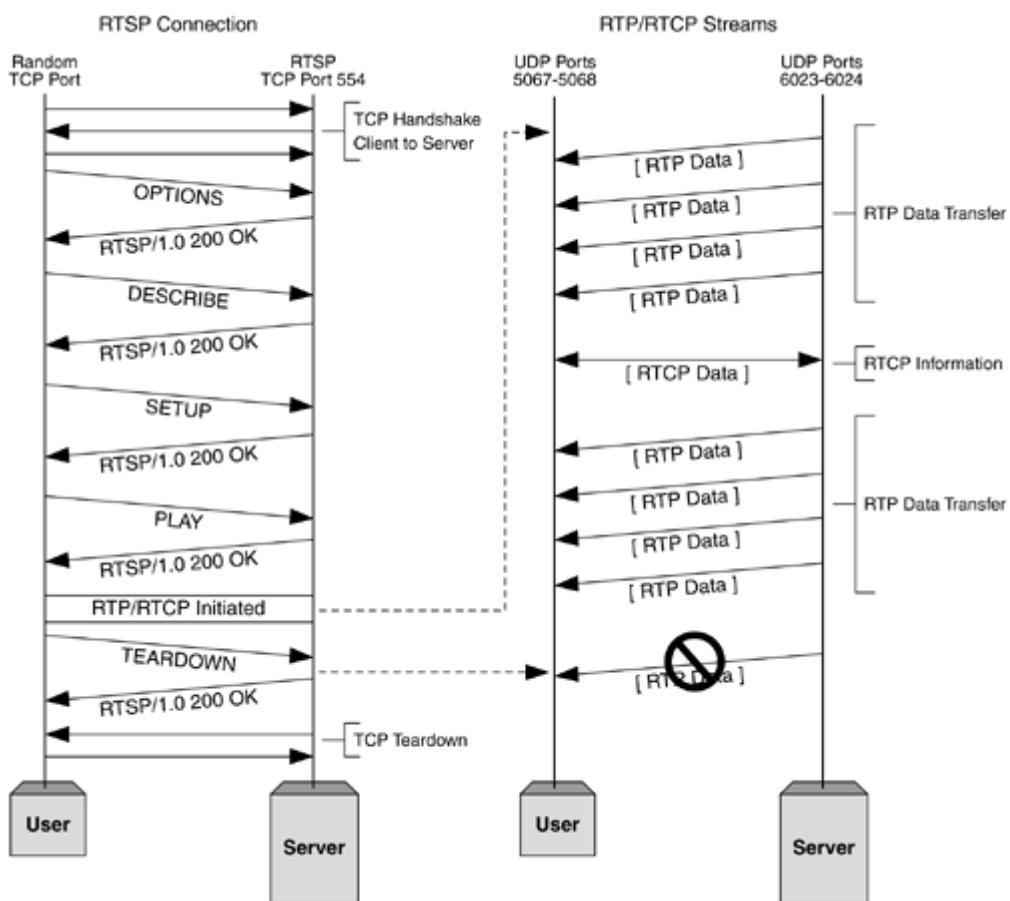
dinamički dodeljuje. Port na prijemnoj strani (port na kojem klijent očekuje UDP pakete) određuje klijent, a razmena podataka se vrši preko RTSP kontrolne veze.

Svaki RTP bitski tok (unikast ili multikast) je praćen RTCP dvosmernim bitskim tokom koji u sebi nosi informacije koje opisuju RTP bitski tok. Bitski tok RTCP-a nosi informacije o kvalitetu signala i parametrima koji su korišćeni prilikom zaključavanja birača kanala. RTCP uvek koristi port koji je dodeljen za RTP uvećan za jedan i uvek je neparan broj[3].

2.3.2 RTSP operacije

RTSP protokol je veoma sličan po strukturi, a naročito sintaksno sa HTTP. Oba koriste istu strukturu URL za opis objekta. RTSP uvodi brojna dodatna zaglavla i takođe omogućava prenos podataka različitim tokovima preko različitih protokola, kao preko već opisanog RTP protokola. Osnovni koraci u procesu korišćenja RTSP protokola su:

1. Klijent uspostavlja TCP vezu sa serverom, u većini slučajeva na podrazumevanom RTSP portu 554
2. Klijet zatim šalje seriju RTSP komandi sličnog formata HTTP protokola. U okviru tih komandi klijent detaljno opisuje serveru zahteve sesije, kao što su verzija RTSP koja je podržana, transportni protokol koje će biti korišćen za prenos podatka, portove za UDP i TCP povezivanje. Te informacije se šalju slanjem DESCRIBE i SETUP komandi. U odgovoru server šalje identifikator sesije koji klijent koristi za identifikovanje prenosnog toka u narednim razmenama poruka.
3. Kada se završi razmena transportnih parametara, klijent tada šalje PLAY komandu da obavesti server da započne dostavljanje preko RTP toka podataka.
4. Kada klijent odluči da prekine tok podataka, šalje serveru TEARDOWN komandu zajedno sa identifikatorom sesije da zaustavi slanje preko tog toka.



Slika 2-2 Primer razmene RTSP poruka sa prenosom RTP i RTCP podataka

2.3.3 Primer razmene RTSP poruka

U prvom koraku klijent uspostavlja TCP konekciju sa serverom i šalje OPTIONS komandu kako bi pokazao serveru koju verziju protokola koristi. Server odgovara 200 OK porukom.

C->S OPTIONS rtsp://video.foocorp.com:554 RTSP/1.0

Cseq: 1

S->C RTSP/1.0 200 OK

Cseq: 1

Nakon toga klijent šalje DESCRIBE komandu sa URL multimedijalne datoteke koju zahteva. Server odgovara 200 OK porukom koja sadrži i pun opis multimedijalnog sadržaja, predstavljen u SDP ili MHEG format.

C->S	DESCRIBE rtsp://video.foocorp.com:554/streams/example.rm RTSP/1.0 Cseq:2
------	---

S->C	RTSP/1.0 200 OK Cseq: 2 Content-Type: application/sdp Content-Length: 210 <SDP Data...>
------	---

U trećoj fazi klijent šalje SETUP komandu kako bi serveru naznačio koje transportne mehanizme želi da koristi. Server potvrđuje mehanizme za transport i dostavlja port za transport sadržaja i jedinstveni identifikator sesije.

C->S	SETUP rtsp://video.foocorp.com:554/streams/example.rm RTSP/1.0 Cseq: 3 Transport: rtp/udp;unicast;client_port=5067-5068
S->C	RTSP/1.0 200 OK Cseq: 3 Session: 12345678 Transport: rtp/udp;client_port=5067-5068;server_port=6023-6024

Sada je klijent spremjan da započne prijem toka podataka i slanjem komande PLAY daje serveru signal da započne slanje multimedijalnog sadržaja. Poruka sadrži URL i identifikator sesije koje je server prethodno dostavio. Server nakon slanja potvrdnog odgovora započinje RTP slanje.

C->S	PLAY rtsp://video.foocorp.com:554/streams/example.rm RTSP/1.0 Cseq: 4 Session: 12345678
S->C	RTSP/1.0 200 OK Cseq: 4

Kada klijent odluči da zaustavi prenos, šalje TEARDOWN komandu da bi prekinuo RTSP sesiju označenu identifikatorom. Server nakon slanja potvrdnog odgovora zaustavlja RTP prenos podataka.

C->S TEARDOWN rtsp://video.foocorp.com:554/streams/example.rm RTSP/1.0
Cseq: 5
Session: 12345678

S->C RTSP/1.0 200 OK
Cseq: 5

3. Opis rešenja

Primenom ovog rešenja omogućuje se proširivanje skupa uređaja koji se mogu koristiti za prenos TV programa, jer se omogućuje upravljanje uređajima koji su osposobljeni za kontrolu preko RTSP protokola. Pre početka prikazivanje TV programa potrebno je izvršiti pretragu dostupnih uređaja za prijem TV signala. Ovo se radi koristeći neki od već postojećih rešenja (npr. UPnP). Ovim postupkom dobija se lista servisa dinamički, umesto čitanjem iz unapred zadate liste servisa.

Prilikom projektovanja rešenja za zadati problem, neohodno je voditi računa o modularnosti rešenja, kako bi ono bilo jednostavno za nadogradnju i ugradnju u postojeća rešenja ukoliko za tim postoji potreba.

Ideja je da se rešenje podeli u više različitih modula koji izvršavaju i kontrolisu različite funkcije sistema. Time bi se obezbedila veća modularnost rešenja i omogućlo lakše testiranje i pronalaženje eventualnih grešaka.

3.1 Opis ciljne platforme

Za razvoj rešenja korišćeni su uređaji za dostavljanje televizijskog programa uživo i STB za prikaz digitalno TV sadržaja. Za distribuciju DVB signala preko IP mreže do klijentskih uređaja korišćen je RT3040-00 SAT>IP server. Za prihvatanje i prikazivanje TV sadržaja koristen je Android 4k hybrid PVR STB uređaj.

3.1.1 SAT>IP server

RK3040-00 server predstavlja standradizovan način za IP klijente da pristupe emisiji TV sadržaja uživo. Ovim putem gledaoci mogu pratiti sadržaj uživo na širokom spektru uređaja kao što su pored STB uređaja i tablet računari, pametni telefoni, desktop i laptop računari.

U zavisnosti od konfiguracije klijentske sesije, upotreba RK3040-00 može se podeliti na dve grupe:

1. Statički režim rada za primenu u hotelima, aerodromima, bolnicama.
2. Dinamički režim rada za primenu u kućama, stanovima, spavaonama i slično.

U statičkom modu za svaki frontend otvara se po jedna multicast grupa što omogućava povezivanje više od jednog klijenta po grupi.

U dinamičkom režimu za svaki frontend kreira se po jedna unicast grupa od strane klijenta i pristup grupi nije moguća za više klijenata. Broj klijenata koji mogu pristupiti serveru je ograničen brojem frontenda dostupnih u okviru samog servera.

Hardverske komponente ovog uređaja su:

- Abilis TB101 SoC, 128MB, opcionalno 256MB DDR2, 256MB SPI
- 4 x DVB-S/S2 RF ulaza
- 1 x Ethernet 10/100/1000BaseT, IEEE 802.3
- 1 x SD Card slot
- 1 x 12VDC konektor za napajanje
- 2 x dvobojne LED (1 LED za indikaciju napajanja, 1 LED za indikaciju mrežne aktivnosti)



Slika 3-1 SAT>IP server

3.1.2 DTV STB

DTV prijenik koji je korišćen pri izradi ovog rada je Android platforma bazirana na Broadcom 7252S čipsetu, sa 2 video dekodera koji mogu da dekoduju i 4K video sadržaj.

Komponente koja ova platforma poseduje su:

- BCM7252S (12k DMIPS, dual ARM core, 1MB L2 cache)
- 2D GPU: 2x 800Mpix/s
- 3D GPU: 2,7Gpix/s OpenGL 2.0/3.0
- Flash memorija 8GB eMMC, 4MB NOR
- 4K UHD HEVC 2048p60/MP4/MPEG2/VP9/HDR video dekoder
- MPEF-1 layer 1,2, AAC LC, HE-AACv1/2, MP3 Dolby Digital, Dolby Digital Plus audio dekoder
- NorDig compliant DVB-C quad tuner (BCM31584)
- 802.11ac 3x3 2.4GHz/5GHz Wi-Fi i BlueTooth

Konektori i kontrole:

- RF In
- S/PDIF (optički)
- HDMI 2.0a Tx (HDCP 2.2 i CEC podržani)
- RJ-45 GbE
- USB 3.0
- 12V DC priključak za napajanje
- BT/IR hibridni daljinski upravljač by Stofa



Slika 3-2 Android Set-Top Box

3.1.3 Srednji sloj programske podrške ciljne platforme (iWedia Comedia)

Srednji sloj (Middleware) je softverski paket koji služi kao sprega između operativnog sistema ili baza podataka i aplikativnog nivoa. On omogućava programerima da lakše razvijaju aplikacije za specijalnu namenu tako što apstrahuje funkcionalnosti i hardver platforme kao i funkcije operativnog sistema.



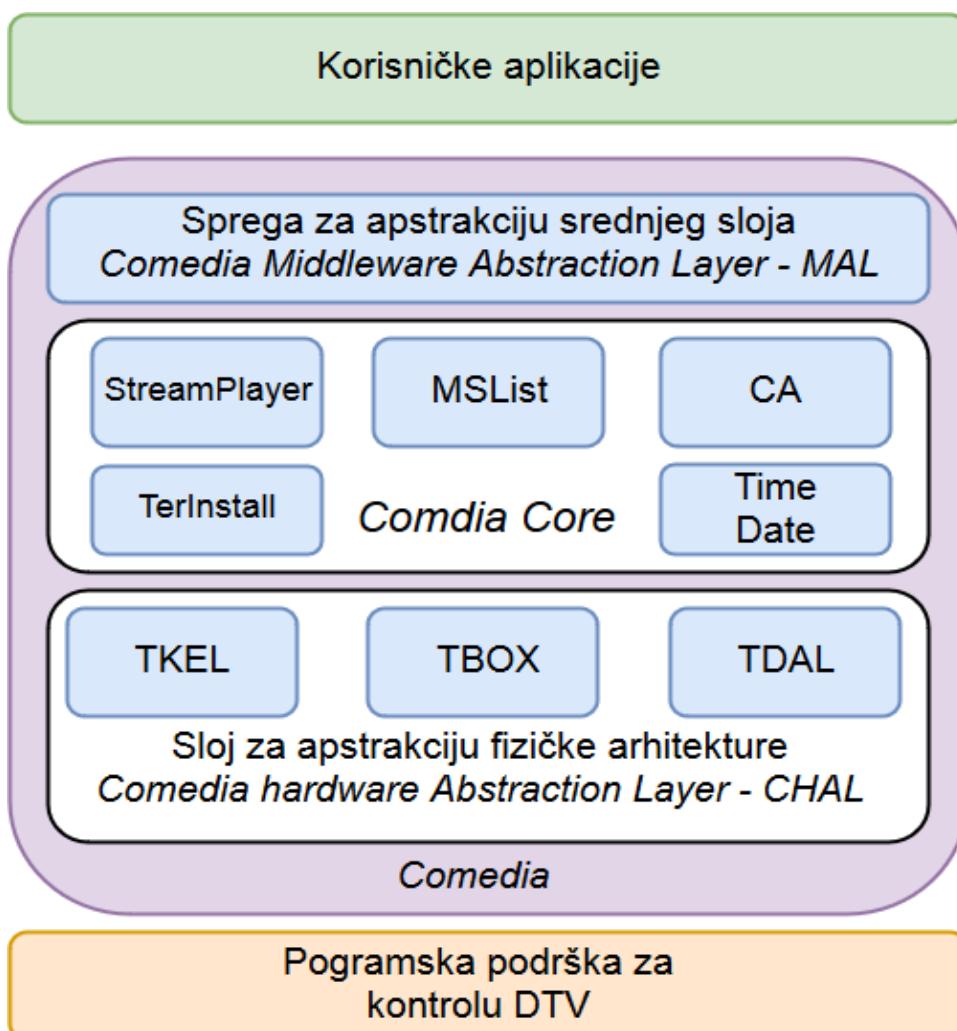
Slika 3-3 Pozicija srednjeg sloja u okviru programskog sistema

U sklopu programske podrške za ovu DTV platformu koristi se srednji sloj Comedia, rešenje kompanije iWedia.

CHAL (Comedia Hardware Abstraction Layer) je sloj koji implementira apstrakciju fizičke arhitekture ciljane platforme i on je zavisan od fizičke arhitekture DTV prijemnika. Da bi se omogućile funkcionalnosti Comedia srednjeg sloja, CHAL je taj koji treba prilagoditi datom hardveru. CHAL se sastoji iz tri nezavisne celine:

1. TKEL sloj za apstrakciju funkcionalnosti operativnog sistema
2. TBOX je modul koji deklariše pomoćne funkcije i makroje i služi za debagovanje programske podrške
3. TDAL je sloj za apstrakciju rukovalaca (drivers) i sastoji se od više modula, gde svaki blok fizičke arhitekture DTV prijemnika poseduje odgovarajući TDAL modul. TDAL je specifičan za uređaje koji podržavaju DVB/MPEG, odnosno za digitalne TV prijemnike.

Za izradu ovog rešenja važan je TDAL_DMD podmodul TDAL-a. Uloga ovog modula je upravlja fizičkim mrežnim sprežnim modulom (tuner i demodulator) tako što podešava parametre za demodulaciju zavisno od korišćenog standarda (DVB-T/T2, DVB-S, DVB-C). Takođe obavlja postavljanje prijema na određenu frekvenciju i obavlja pretraživanje zadatog frekvencijskog opsega (scanning).

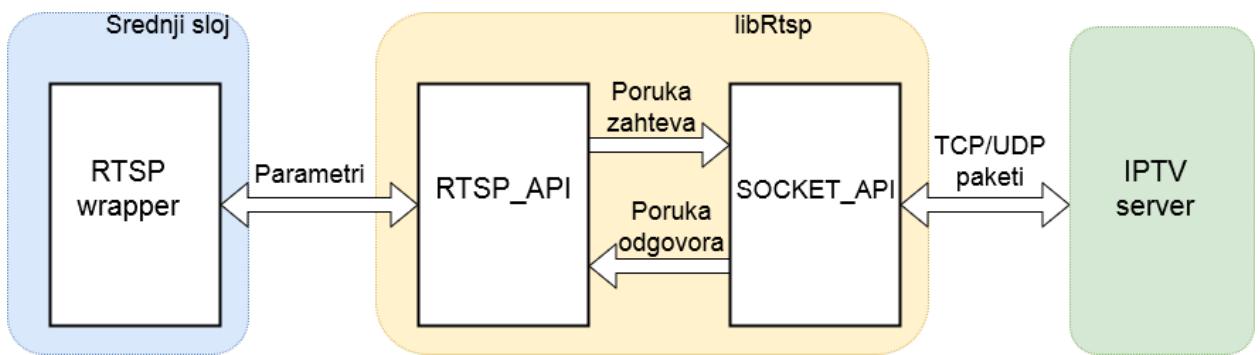


Slika 3-4 Arhitektura Comedia srednjeg sloja

3.2 RTSP biblioteka

Ovaj modul realizovan je kao statička biblioteka *libRtsp* koja sadrži genericke funkcionalnosti RTSP protokola. Generička u smislu da je primenljiva u rezličitim implementacijama od strane različitih korisnika, tj od korisnika zavisi koji podaci će biti slati i u kom formatu.

Sastoji se od dva podmodula. Prvi podmodul obezbeđuje RTSP funkcionalnosti, dok drugi podmodul obavlja funkcionalnosti TCP komunikacije.



Slika 3-5 Prikaz organizacije podmodula u okviru RTSP biblioteke

U RTSP_API podmodul pristižu podaci u formi stringova koji označavaju komandu RTSP zahteva, putanju (path) i RTSP zaglavlja (headers). Ovi podaci koriste se da bi se formirala kompletна poruka RTSP zahteva koja se šalje serveru.

Ovaj podmodul obezbeđuje neophodne funkcije za parsiranje poruke RTSP odgovora. Odgovor se razlaže na statusni kod, zaglavlja odgovora i tela odgovora ako ga ima. Vrednosti zaglavlja koji nose korisne informacije za upravljanje bitskim tokovima uvezuju se u liste kako bi bili čitljivi i dostupni korisniku tih informacija. Takođe, postoje i složena zaglavlja, ona koja se sastoje od više različitih informacija formirana po određenom formatu. Dostupne su i funkcije za parsiranje takvih zaglavlja, kao i strukture za njihovo čuvanje.

Neke pomoćne, statičke funkcije su realizovane u ovom podmodulu. Takve funkcije su one koje vrše konverziju tipova podataka kako bi bili pogodni za dalje korišćenje ili čuvanje.

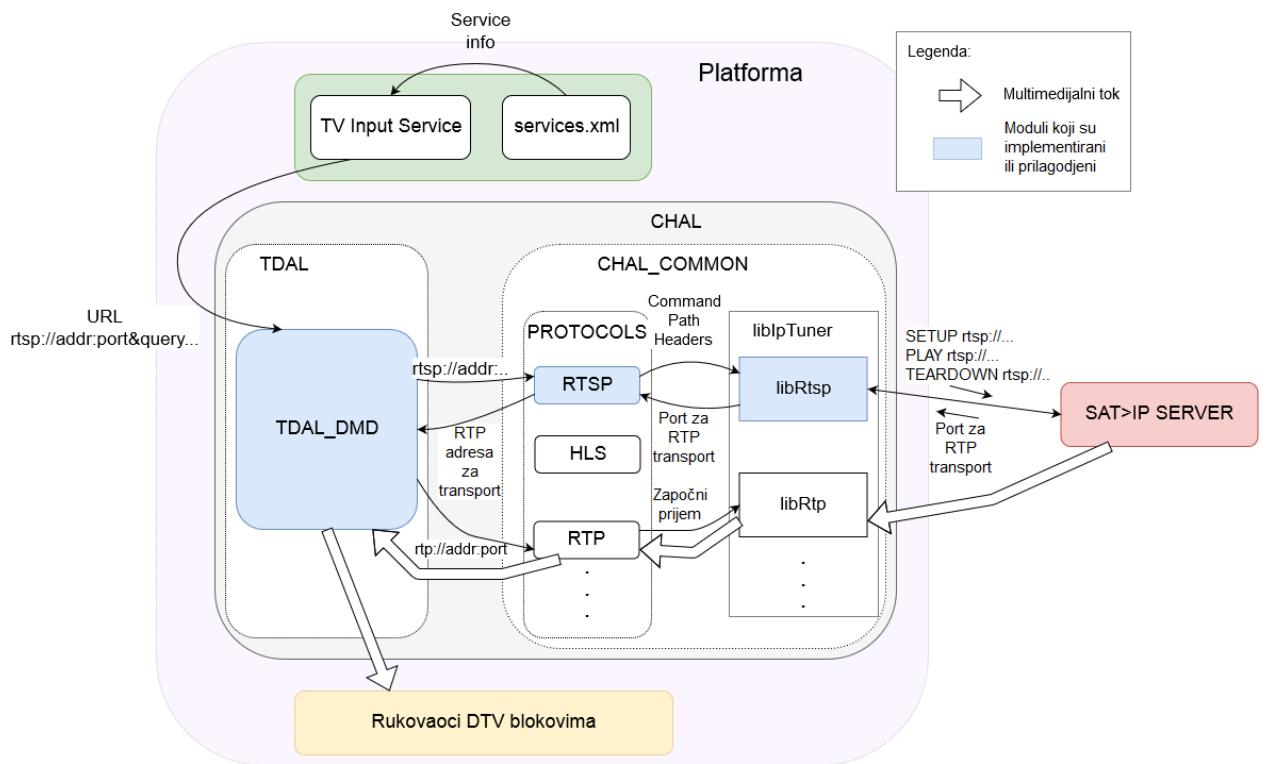
U sklopu biblioteke za RTSP mehanizme, uključen je modul socket_api koji rukuje TCP konekcijom i komunikacijom sa RTSP serverskom aplikacijom na IPTV serveru. U tom modulu koriste se funkcije TCP/UDP socket api-a za povezivanje i raskidanje veze kao i slanje i prijem TCP poruka.

3.3 Integracija u HAL nivo srednjeg sloja

Prilikom startovanja prikazivanja TV sadržaja ili pri promeni kanala, Android TV input servis čitanjem iz liste servisa *services.xml* dobavlja podatke potrebne za početak emitovanja. Ti podaci su u obliku URL-a koji su adresa do traženog servisa. U RTSP slučaju URL sadrži i parameter potrebne tjuneru da se pozicionira na željenu frekvenciju. Parametri se dostavljanju srednjem sloju, koji dospevaju u sloju za apstrakciju hardvera.

U okviru CHAL-a, modul namenjen upravljanju tjunerom i demodulatorom je modul TDAL_DMD. U ovom modulu radi se prepoznavanje zadatog protokola za dostavljane televizijskog sadržaja koristeći funkcije iz modula CHAL_COMMON u kome su implementirane funkcije za rad sa protokolima, PROTOCOLS. Kada je prepoznat RTSP protokol, TDAL_DMD prosleđuje URL do modula za upravljanje RTSP zahtevima.

RTSP podmodul modula PROTOCOLS koristi funkcije definisane u biblioteci *libRtsp* koja je u sklopu *libIpTuner* biblioteke kako bi obavljao komunikaciju sa IPTV serverom. Kroz ovu komunikaciju obavlja se razmena RTSP komandi sa serverom čime se javlja serveru da se pozicionira na zadatu frekvenciju i uspostavlja sesiju za kontrolu servera. Takođe, dobavlja se port na koji klijent otvara RTP konekciju za prijem miltimedijalnog toka. Zatim daje znak serveru da započne slanje TV sadržaja preko RTP transportnog protokola. U isto vreme obavlja se inicijalizacija RTP modula koje potom obavlja prenos multimedijalnog toka od servera ka rukovaocima za prikaz sadržaja.



Slika 3-6 Prikaz integracije realizovane RTSP biblioteke

Izmene u okviru srednjeg sloja odnose se na prilagođavanje određenih modula za rad sa RTSP protokolom. TDAL_DMD modul proširen je funkcijama koje mu omogućavaju prepoznavanje ulaznih parametara koji su namenjeni za prenos posredstvom RTSP protokola.

Izmene u CHAL_COMMON modulu vezane za podmodul koji upravlja protokolima odnosile su se na prilagođavanje generičkih funkcionalnosti *libRtsp* potrebama za rad na ciljanoj platformi. To se postiže implementiranjem wrapper funkcijama („omotač“ funkcija) koje izvršavaju RTSP funkcionalnosti za izvršavanje na ciljnoj platformi. Za rad SAT>IP servera neophodno je slanje keep-alive signala koji se implementira pokretanjem nezavisnog procesa koji periodično šalje OPTIONS RTSP komandu, kako je definisano Sat2Ip specifikacijom.

4. Programsко rešenje

U ovom poglavlju biće detaljnije opisano programsko rešenje zadatka. Programsko rešenje pisano je u programskom jeziku C i razvijano u Linux okruženju. Podeljeno je u dve celine, gde je u prvoj celini opisana implementacija funcionisanja RTSP protokola, formiranje poruka u Rtsp sintaksi i komunikacija između servera i klijenta. U okviru druge celine opisana je integracija RTSP funkcija u Comedia srednji sloj, tačnije nivo za apstrakciju hardvera (CHAL).

4.1 RTSP biblioteka (libRtsp)

Ovaj modul deli se na dva podmodula. Prvi modul sadrži strukture i funkcije koje omogućavaju realizaciju RTSP zahteva. U drugom podmodulu realizovani su mehanizmi TCP konekcije i komunikacije između servera i klijenta.

4.1.1 Modul za rukovanje RTSP funkcionalnostima

U datoteci *libRtsp.h* definisane su enumeracije i strukture koje predstavljaju podatke neophodne za funkcionisanje RTSP modula.

Enumeracija *RtspErrorCode* definise kodove gresaka koje se mogu javiti tokom rada RTSP funkcija.

```
typedef enum _RtspErrorCode{
    RTSP_OK          = 0,
    RTSP_CONNECT_SOCKET_ERROR,
    RTSP_SEND_REQUEST_ERROR,
    RTSP_RECV_REQUEST_ERROR,
    RTSP_ALLOC_ERROR,
    RTSP_RESPONSE_PARSING_ERROR,
    RTSP_SERVER_CONNECT_ERROR,
    RTSP_SERVER_DISCONNECT_ERROR
}
```

```
    } RtspErrorCode;
```

Enumeracija Method definise oznake tipova komandi koje se salju kao zahtevi serveru.

```
typedef enum _Method {
    OPTIONS = 0,
    DESCRIBE,
    SETUP,
    PLAY,
    TEARDOWN
} Method;
```

Struktura RtspHandler predstavlja atribute i stanja jedne RTSP sesije.

```
typedef struct _RtspHandler {
    int      socket;           // ID soketa koji je otvoren za komunikaciju sa IPTV serverom
    int      cseq;            // Uvećava se svakom transmisijom
    char*   serverIp;        // IP adresa servera
    char*   serverPort;       // Port servera
    char*   sessionId;        // ID aktivne sesije
    char*   streamId;         // ID bitskog toka u okviru sesije, nakon PLAY metode
    int      isSession;        // Označava da li je sesija aktivna u okviru handlera
    pthread_t keepAliveThread; // ID programske niti koja salje RTSP poruke za održavanje sesije aktivnom
} RtspHandler;
```

HeaderItem struktura definise jednog člana liste koja predstavlja RTSP zaglavlj u okviru RTSP poruke, čiji je format headerTag: headerValue.

```
typedef struct _HeaderItem {
    char*   header_tag;
    char*   header_value;
    struct _HeaderItem* next;
} HeaderItem;
```

SdpBodyItem struktura definiše jednog člana liste koja predstavlja elemente tela RTSP poruke, čiji je format definisan SDP protokolom.

```
typedef struct _SdpBodyItem {
    char*   body_tag;
    char*   body_value;
    struct _SdpBodyItem* next;
} SdpBodyItem;
```

Struktura ParsedResponse sadrži elemente RTSP odgovora od servera formatiran u oblik pogodan za pretraživanje.

```
typedef struct _ParsedResponse {
```

```

char*      status_code;
char*      status_desc;
HeaderItem*   headerList;
SdpBodyItem*  bodyList;
} ParsedResponse;

```

4.1.1.1 Funkcije RTSP modula

U nastavku je dat pregled funkcija implementiranih u svrhu funkcionisanja RTSP protokola.

RtspErrorCode **connectToServer**(RtspHandler* handler)

Pozivom ove funkcije obavlja se konekcija sa RTSP serverom. Definiše se polje socket parametra handler, IP adresu servera i port su postavljeni u toku inicijalizacije. Te vrednosti se koriste za poziv funkcije za TCP konekciju iz modula sock_comm.c. Podrazumevana vrednost RTSP porta je 554, ukoliko drugačije nije zadato.

RtspErrorCode **initializeRtsp**(RtspHandler* handler)

Funkcija obavlja inicijalizaciju RTSP handlera postavljanjem vrednosti elemenata strukture RtspHandler na početne vrednosti. Ovo je potrebno uraditi na početku rada, kao i nakon ponovnog pokretanja rtsp sesije kao reinicijalizaciju modula.

RtspErrorCode **makeRtspRequest**(Method methodType, **char*** path,
char* headers, RtspHandler* handler, **char**** returnString)

Ova funkcija obavlja sastavljanje RTSP zahteva, njegovo slanje serveru, prihvatanje RTSP odgovora od servera i vraća dobijenu poruku odgovora. Koristeći parametre methodType, path i headers formira se poruka RTSP zahteva u format definisan specifikacijom protokola. Poruka se zatim salje ka serveru pozivom funkcije za slanje socketSend modula sock_comm.c, zatim se čeka odgovor od servera koji se prihvata funkcijom socketRecieve. Odgovor se vraća korisniku preko izlaznog parametra returnString.

Povratna vrednost je kod greske.

RtspErrorCode **parseResponse**(**const char*** msg, ParsedResponse* response)

Korišćenjem ove funkcije omogućeno je parsiranje poruke odgovora koji pristiže u formi stringa veće dužine kao parametar **char*** msg. Pre početka parsiranja, neophodno je napraviti kopiju poruke kako bi se sačuvao izvorni oblik. Poruka je formatirana tako da svaka linija nosi posebno informaciju, odvojenu novim redom (**\r\n**). Prva linija nosi informacije o statusu. Zatim

svaka naredna linija predstavlja header u formi *ime: vrednost*. Ukoliko postoji opcionalno telo odgovora, ono je odvojeno praznim redom (`\r\n`) i svaki linija tela se parsira posebnom funkcijom.

Parsirane vrednosti se čuvaju u obliku spregnute liste čiji su elementi instance strukture `ParsedResponse`, a korisniku vraća preko izlaznog parametra `ParsedResponse* response`.

Povratna vrednost funkcije je kod greske.

char* getHeaderValue(const ParsedResponse* response, const char* headerTag)

Nakon parsiranja poruke odgovora podaci su sačuvani u listi tipa `ParsedResponse`. Da bi se dobavili podaci iz liste koristi se metoda `getHeaderValue`. Tražena vrednost podataka se dobija prosleđivanjem imena zaglavlja koje se traži parametrom `char* headerTag`.

Ulagani parametar je struktura poja sadrži parsiranu poruku odgovora RTSP metode.

Povratna vrednost funkcije je string koji sadrži traženu vrednost zaglavlja.

char* getBodyValue(const ParsedResponse* response, const char* bodyTag);

Ova funkcija radi slično kao i prethodna, samo sto pretražuje listu vrednosti tela odgovora.

Ulagani parametar je struktura koja sadrži parsiranu poruku odgovora RTSP metode.

Povratna vrednost funkcije je string koji sadrži vrednost traženog parametra iz tela odgovora.

void printParsedResponse(const ParsedResponse* response)

Ovo je pomoćna funkcija koja se koristi za stampanje parsirane poruke odgovora. Ova funkcija je korisna tokom razvijanja same biblioteke, ali i pri njenoj kasnijoj primeni.

Ulagani parametar funkcije je adresa strukture sa parsiranom porukom.

RtspErrorCode freeParsedResponse(ParsedResponse* response);

Tokom parsiranja poruke odgovora, alocira se dinamička memorija za smeštanje vrednosti zaglavlja i vrednosti parametara tela poruke. Tu memoriju je nepodobno osloboditi nakon završetka rada sa vrednostima parsirane poruke

Ulagani parametar funkcije je struktura sa vrednostima parsirane poruke.

Povratna vrednost funkcije je kod greške.

4.1.2 Modul za rukovanje TCP konekcijom

U ovom fajlu implementirane su funkcije neophodne za TCP komunikaciju sa serverom. Ova komunikacija se obavlja korišćenjem TCP socket api-a. Podmodul je namenjen za uspostavljanje i raskidanje veze sa serverom kao i slanje i prihvatanje TCP paketa.

U nastavku je dat pregled funkcija i njihov detaljniji opis, kao i opis njihove namene.

int socketConnect(int* clientSocketID, char* serverAddress, char* serverPort)

Ova funkcija obavlja TCP konekciju sa serverom. To se radi pozivanjem funkcije *connect* Socket api-a. Parametar *clientSocketID* predstavlja identifikator TCP soketa. Parametri *serverAddress* i *serverPort* su IP adresa i port servera sa kojim se ostvaruje konekcija. Preko ove veze šalju se RTSP komande za upravljanje sesijom.

int socketSend(int clientSocketID, char* buffer, int bufferSize)

Nakon uspostavljenе veze sa serverskom stranom, pozivanjem prethodno opisane funkcije, omogućeno je slanje TCP paketa. To se izvršava funkcijom *socketSend* gde *clientSocketID* označava identifikator prethodno uspostavljenе veze. Parametar *buffer* sadrži tekst RTSP poruke koju treba poslati, a *bufferSize* predstavlja dužinu poruke.

int socketReceive(int clientSocketID, char* buffer, int bufferSize)

Ova funkcija se koristi za prihvatanje TCP paketa sa servera. Prihvatanje dolaznih podataka omogućeno je pozivom funkcije *recv()* socket api-a. Ne blokirajući rad ove funkcije omogućen je ograničenjem vremena čekanja na odgovor koje bi moglo biti beskonačno ukoliko dođe do prekida veze sa serverskom stranom. Ovo se postiže prosleđivanjem *clientSocketID* funkciji *poll* koja omogućuje čekanje na dolazne podatke. Program će čekati na podatke dok oni ne stignu ili do isteka zadatog vremena čekanja.

Parametar *buffer* predstavlja izlazni parametar funkcije i to je memorijski bafer u kojem će biti smešteni podaci pristigli od serverske strane. Parametar *bufferSize* predstavlja dužinu izlaznog bafera.

int socketClose(int clientSocketID)

Ovom funkcijom prekida se TCP konekcija sa serverom. Prekid veze obavlja se pozivom funkcije socket api-a *close()* prosleđivanjem identifikatora soketa na kom je veza uspostavljena.

5. Testiranje i verifikacija

Ispravnost rada ispitana je u dve faze testiranja. Prva se odnosi na pisanje testne aplikacije na linux-u gde unit testovi ispituju ispravnost funkcionisanja realizovanih funkcija RTSP biblioteke. Druga faza predstavlja funkcionalno testiranje integracije biblioteke u srednji sloj programske podrške ciljne platforme.

5.1 Ispitivanje RTSP biblioteke

Ispitivanje ispravnosti biblioteke izvođeno je u više faza. Jedna od faza bila je postavljanje različitih stresnih scenarija za sistem kako bi se ispitala izdržljivost sistema.

Data tabela pokazuje opise scenarija testiranja kao i očekivane i dobijene rezultate testiranja:

Tabela 1 Pregled testova za verifikaciju ispravnosti RTSP biblioteke

Scenario	Očekivani rezultati	Dobijeni rezultati
Naizmenično slanje SETUP, PLAY, TEARDOWN zahteva	Dobijanje odgovora na svaki zahtev, bez curenja memorije	Očekivan
Slanje OPTIONS zahteva u kratkim vremenskim intervalima, bez aktivne sesije	Primljen odgovor na svaki zahtev	Očekivan
Slanje neispravnih I nepotpunih zahteva	Poruka sa statusnim kodom 400 Bad Request	Očekivan
Provera ispravnosti funkcije za parsiranje odgovora. Ovde je bitno da nema curenja memorije jer se u ovoj funkciji radi	Ispravno parsirana poruka. Bez curenja memorije	Očekivan

dinamička alokacija memorije		
Provera ispravnosti funkcije za parsiranje Transport zaglavlja RTSP odgovora na SETUP zahtev	Ispravno parsirana poruka, Bez curenja memrije	Očekivan
Proverava funkcije za pronalaženje i dobavljanje vrednosti zaglavlja i vrednosti parametara tela RTSP odgovora	Dostavljanje odgovarajuce vrednosti za tražene parametre	Očekivan

5.2 Funkcionalno testiranje na ciljnoj platformi

U svrhu funkcionalnog testiranja napisana shell skripta koja izvršava input komande pomoću kojih se vrši simulacija fizičkih događaja na ploči. Ovim putem izvršavane su operacije promene kanala, zaustavljanje i ponovno pokretanje rada platforme i slično.

Takođe su vršena ispitivanja reakcije sistema na prekid konekcije sa IP mrežom, prekid dotoka TV sadržaja do SAT>IP servera, gubitak napajanja i drugi. Za testiranje korišteni su video tokovi koji se transportuju pomoću tri različita protokola, HLS, DASH i RTSP.

U narednoj tabeli prikazani su testni slučajevi sa opisom i očekivanim rezultatima:

Tabela 2 Pregled testova za funkcionalno ispitivanje na ciljanoj platformi

Opis testa	Očekivani rezultat
Jedan kanal tokom 8 sati	Sadržaj pristiže bez prekida tokom celog perioda
Promena kanala na svakih 10 sekundi tokom 10 minuta	Uspešno pokretanje sesije svaki put pri promeni na kanal koji se prenosi preko RTSP protokola. Pri prelasku na kanal sa drugim protokolom, prekida se sesija. Ne sme doći do prekida emitovanja.
Promena kanala na svakih 10 sekundi tokom sat vremena	Uspešno pokretanje sesije svaki put pri promeni na kanal koji se prenosi preko RTSP protokola. Pri prelasku na kanal sa drugim protokolom, prekida se sesija. Ne sme doći do prekida emitovanja.
Prekid IP konekcije na STB strani na 30 sekundi i ponovna uspostava veze	Prikaz video sadržaja se nastavlja nakon ponovnog uspostavljanja IP konkcije. Ukoliko je veza prekinuta u trenutku slanja keep-alive signala server, RTSP sesija se prekida, što je definisano SAT2IP specifikacijom.

Prethodni test ponovljen 20 puta	Prikaz video sadržaja se nastavlja nakon ponovnog uspostavljanja IP konkcije za svaku iteraciju. Ukoliko je veza prekinuta u trenutku slanja keep-alive signala server, RTSP sesija se prekida, što je definisano SAT2IP specifikacijom.
Prekid veze preko koaksijalnog kabla na SAT>IP serveru na 30 sekundi i ponovno uspostavljanje veze	Dolazi do prekida prikaza video sadržaja. Prikaz se nastavlja nakon ponovnog povezivanja. Sesija ostaje aktivna sve vreme.
Prethodni test ponovljen 20 puta	Dolazi do prekida prikaza video sadržaja za svaku iteraciju. Prikaz se nastavlja nakon ponovnog povezivanja za svaku iteraciju. Sesija ostaje aktivna sve vreme.

Ovim testiranjem pokazana je robustnost sistema i mogućnost rada RTSP biblioteke u stresnim slučajevima. Takođe se vidi mogućnost oporavka sistema posle različitih grešaka ili problema na hardveru ili infrastrukturi. Dobijeni rezultati testova prikazanih u tabeli 2 dati su u sledećoj tabeli:

Tabela 3 Dobijeni rezultati testiranja na ciljnoj platformi

Opis testa	Dobijeni rezultati
Jedan kanal tokom 8 sati	Očekivani
Promena kanala na svakih 10 sekundi tokom 10 minuta	Očekivani
Promena kanala na svakih 10 sekundi tokom sat vremena	Očekivani
Prekid IP konekcije na STB strani na 30 sekundi i ponovna uspostava veze	Očekivani
Prethodni test ponovljen 20 puta	Očekivani
Prekid veze preko koaksijalnog kabla na SAT>IP serveru na 30 sekundi i ponovno uspostavljanje veze	Očekivani

Prethodni test ponovljen 20 puta	Očekivani
----------------------------------	-----------

6. Zaključak

Prikazano je jedno rešenje implementacije RTSP protokola za digitalne televizijske uređaje. Primenom ovog rešenja omogućena je podrška za komunikaciju TV prijemnika sa serverima koji podržavaju RTSP protokol. Pažnja je posvećena projektovanju modularnog sistema kako bi se omogućila jednostavnija integracija sa postojećim rešenjima.

Implementacijom RTSP biblioteke omogućene su generičke funkcionalnosti protokola definisane specifikacijom. U okviru biblioteke obezbeđeni su mehanizmi za TCP i UDP konekciju, formiranje poruka RTSP zahteva, parsiranje RTSP odgovora i čuvanje podataka.

Izmenama u okviru srednjeg sloja omogućena je primena RTSP biblioteke za specifičnu namenu upravljanja i dostavljanja televizijskog sadržaja.

Za razvoj i verifikaciju ispravnosti korišćen je satelitski IP server RK-3040, kompanije RT-RK i DTV Set-Top Box zasnovan na Broadcom 7252S čipsetu i Android operativnom sistemu. Testiranja su pokazala robustnost sistema pri korišćenju RTSP protokola za prenos televizijskog sadržaja kao i mogućnost uspešnog oporavka od prekida IP veze i dotoka TV sadržaja do SAT>IP servera.

Ono što ovo rešenje doprinosi je mogućnost dinamičke kontrole udaljenog IPTV servera i tjunera koristeći RTSP protokol. Prethodno dato rešenje je za IP transport koristilo statičku konfiguraciju sa fiksiranim adresama i portovima za prihvatanje RTP tokova.

Jedna od smernica za proširivanje funkcionalnosti ovog rešenja bila bi implementacija mehanizama za otkrivanje prisutnih IPTV servera na mreži i dobavljanje njihovih IP adresa. U tu svrhu neophodno je implementirati protokol sa klijentske (napr. UPnP) strane koji bi periodično proveravao prisustvo IPTV servera na mreži. Zbog obima posla potrebnog za realizaciju ovog mehanizma, ono je izostavljeno iz ovog rešenja.

7. Literatura

- [1] R. Marković, D. Dejanović, S. Tanacković, M. Ačanski, I. Bašičević, "Implementacija Android 4K hibridnog OTT STB uređaja", ETRAN, 2017.
- [2] W. Simpson, H. Freenfield, *IPTV and Internet Video: Expanding the Reach of Television Broadcasting*, Second edition, 2009.
- [3] M. Szme, Ph. Goldie, *Optimizing Network Performance with Content Switching: Server, Firewall and Cache Load Balancing: Server, Firewall, and Cache Load Balancing*, 2003.
- [4] M. Ačanski, "Prilog rešenju komunikacionog protokola za prenos televizijskog sadržaja preko IP mreže", Magistarski rad, Fakultet Tehničkih Nauka, Novi Sad, 2015.
- [5] H. Schulzrinne, A. Rao, R. Lanphier, *Real Time Streaming Protocol (RTSP)*, RFC 2326, IETF, 1998 <https://tools.ietf.org/html/rfc2326>
- [6] SAT>IP protocol specification, SES S.A., 2015, version 1.2.2, http://www.satip.info/sites/satip/files/resource/satip_specification_version_1_2_2.pdf
- [7] R. Fielding, J. Gettys, J. Mogul, H. Frystyk, L. Masinter, P. Leach, T. Berners-Lee, *Hypertext transfre protocol – HTTP-1.1*, 1999 <https://tools.ietf.org/html/rfc2616>
- [8] R. Pantos, W. Maz, *HTTP Live Streaming*, RFC 8216, IETF, 2017, <https://tools.ietf.org/html/rfc8216>
- [9] Dynamic Adaptive Streaming over HTTP, <http://dashif.org/>
- [10] I. Bašičević, M. Ačanski, N. Teslić, M. Popović, M. Nikolić, *The concept of remote tuner in digital television software*, TELFOR, 2015