



УНИВЕРЗИТЕТ У НОВОМ САДУ
ФАКУЛТЕТ ТЕХНИЧКИХ НАУКА У
НОВОМ САДУ



Milan Ačanski

**PRILOG REŠENJU KOMUNIKACIONOG
PROTOKOLA ZA PRENOS
TELEVIZIJSKOG SADRŽAJA PREKO IP
MREŽE**

MAGISTARSKI RAD

Mentor:
Nikola Teslić

Novi Sad, 2016



UNIVERZITET U NOVOM SADU • FAKULTET TEHNIČKIH NAUKA
21000 NOVI SAD, Trg Dositeja Obradovića 6

KLJUČNA DOKUMENTACIJSKA INFORMACIJA

Redni broj, RBR :	
Identifikacioni broj, IBR :	
Tip dokumentacije, TD :	Monografska dokumentacija
Tip zapisa, TZ :	Tekstualni štampani materijal
Vrsta rada, VR :	Magistarski rad
Autor, AU :	Milan Ačanski, dipl. Inž.
Mentor, MN :	Prof. Dr. Nikola Teslić
Naslov rada, NR :	Prilog rešenju komunikacionog protokola za prenos televizijskog signala preko IP mreža
Jezik publikacije, JP :	Srpski / latinica
Jezik izvoda, Jl :	Srpski
Zemlja publikovanja, ZP :	Republika Srbija
Uže geografsko područje, UGP :	Vojvodina
Godina, GO :	2016
Izdavač, IZ :	Autorski reprint
Mesto i adresa, MA :	Novi Sad; trg Dositeja Obradovića 6
Fizički opis rada, FO : (poglavlja/strana/ citata/tabela/slika/grafika/priloga)	9 poglavlja/87 strana/28 Citata/19 tabela/20 slika
Naučna oblast, NO :	Elektrotehnika i računarstvo
Naučna disciplina, ND :	Računarska tehnika
Predmetna odrednica/Ključne reči, PO :	IP mreže, satelitska televizija, DLNA, UPNP, RTP, HTTP, RTSP, RTCP
UDK	
Čuva se, ČU :	U biblioteci Fakulteta tehničkih nauka, Novi Sad
Važna napomena, VN :	
Izvod, IZ :	Ova magistarska teza se bavi istraživanjem u oblasti prenosa televizijskog sadržaja preko IP mreža. Ciljevi su implementacija SAT>IP protokola I definisanje okruženja u kojem će se implementirana tehnologija upoređivati sa postojećim tehnologijama. Tokom izrade teze implementirani su SAT>IP klijent I SAT>IP server koji je verifikovan od strane kompanije SES SA. U tezi se analiziraju parametri implementirane tehnologije i postojećih tehnologija i na osnovu uočenih nedostataka predlažu unapređenja protokola.
Datum prihvatanja teme, DP :	25.11.2015.
Datum odbrane, DO :	23.03.2016.
Članovi komisije, KO :	
Predsednik:	Dr. Miodrag Temerinac, red. profesor
Član:	Dr. Miroslav Popović, red.profesor
Član:	Dr. Jovan Đorđević, red. profesor
Član:	Dr. Ilija Bašičević, vanredni profesor
Član, mentor:	Dr. Nikola Teslić, red. profesor
	Potpis mentora



KEY WORDS DOCUMENTATION

Accession number, ANO :	
Identification number, INO :	
Document type, DT :	Monographic publication
Type of record, TR :	Textual printed material
Contents code, CC :	Master Thesis
Author, AU :	Milan Ačanski
Mentor, MN :	Prof. Dr. Nikola Teslić
Title, TI :	Addition to solution for television signal distribution over IP networks
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 :	2016
Publisher, PB :	Author's reprint
Publication place, PP :	Novi Sad, Dositeja Obradovica sq. 6
Physical description, PD : (chapters/pages/ref./tables/pictures/graphs/appendixes)	9 chapters /87 pages /28 references /19 tables /20 pictures
Scientific field, SF :	Electrical Engineering
Scientific discipline, SD :	Computer Engineering, Engineering of Computer Based Systems
Subject/Key words, S/KW :	IP networks, Sattelite Television, DLNA, UPNP, RTP, HTTP, RTSP, RTCP
UC	
Holding data, HD :	The Library of Faculty of Technical Sciences, Novi Sad, Serbia
Note, N :	
Abstract, AB :	This thesis address area of television signal distribution over IP networks. The goal is SAT>IP client and server implementation and setting up environment for protocol evaluation against protocol that are already present on market. During thesis implemented SAT>IP server is certified by the company SES SA. Thesis is analyzing protocol parmeters and based on that analysis defining limitations and proposing protocol upgrades.
Accepted by the Scientific Board on, ASB :	25.11.2015.
Defended on, DE :	23.03.2016.
Defended Board, DB :	President: Dr. Miodrag Temerinac, red. profesor
	Member: Dr. Miroslav Popović, red.profesor
	Member: Dr. Jovan Đorđević, red. profesor
	Member: Dr. Ilija Bašičević, vanredni profesor
	Member, Mentor: Dr. Nikola Teslić, red. profesor
	Mentor's sign

Zahvalnost

Najiskrenije se zahvaljujem prof. dr. Nikoli Tesliću na podršci u toku realizacije rada. Takođe se zahvaljujem timu kolega sa kojima sam imao sreću da sarađujem. Ipak najveću zahvalnost dugujem svojoj porodici.

SADRŽAJ

1. Uvod.....	1
2. Prikaz postojećih tehnologija.....	3
2.1 DLNA.....	3
2.1.1 Arhitektura DLNA kućne mreže	4
2.1.2 Podela DLNA uređaja.....	5
2.1.2.1 Kućni mrežni uređaji	6
2.1.2.2 Mobilni ručni uređaji	6
2.1.2.3 Infrastrukturni uređaji.....	7
2.1.3 DLNA birač kanala.....	7
2.1.3.1 Predstavljanje DLNA birača kanala drugim uređajima.....	8
2.1.3.2 Pristupanje sadržaju deljenom preko objekta koji predstavlja birač kanala..	8
2.1.3.3 Primer opisa DLNA birača kanala	9
2.2 IPTV	11
2.2.1 Arhitektura IPTV mreže	12
2.2.2 Prenos sadržaja putem IPTV sistema	14
2.3 OTT	15
2.3.1 Arhitektura OTT sistema	15
2.3.2 Prenos video sadržaja putem OTT sistema.....	16
2.4 Poređenje postojećih tehnologija	16
2.4.1 Poređenje IPTV i OTT sistema.....	17
2.4.2 Pozicija DLNA sistema	18
3. Opis predloženog rešenja.....	19
3.1 Opis SAT>IP protokola.....	19

3.1.1	Adresiranje.....	20
3.1.2	Otkrivanje	20
3.1.3	Opisivanje SAT>IP uređaja u mreži.....	24
3.1.4	Kontrola	25
3.1.4.1	RTSP	25
3.1.4.1.1	Formiranje sesije (RTSP SETUP).....	25
3.1.4.1.2	Pokretanje slanja bitskog toka (RTSP PLAY).....	28
3.1.4.1.3	Održavanje sesije na mreži (RTSP OPTIONS)	29
3.1.4.1.4	Operacije nad bitskim tokom	29
3.1.4.1.5	Dobijanje liste bitskih tokova (RTSP DESCRIBE).....	30
3.1.4.1.6	Zatvaranje sesije (RTSP TEARDOWN).....	32
3.1.4.1.7	Format RTSP_URI niza karaktera	33
3.1.4.2	RTCP	34
3.1.4.3	HTTP	35
3.1.5	Slanje bitskog toka.....	35
3.2	Upotreba SAT>IP protokola	35
3.3	Opis realizacije.....	37
3.3.1	Opis realizacije SAT>IP klijenta.....	37
3.3.2	Opis realizacije SAT>IP server	39
3.3.2.1	Arhitektura SAT>IP servera.....	39
3.3.2.2	Adresiranje, otkrivanje i opisivanje.....	40
3.3.2.3	Glavna programska nit SAT>IP servera	42
3.3.2.4	Obrada RTSP zahteva	43
3.3.2.5	Obrada HTTP zahteva	44
3.3.2.6	Programska nit za slanje bitskog toka RTP protokolom	45
3.3.2.7	Programska nit za slanje bitskog toka HTTP protokolom	46
3.3.2.8	Memorijski zahtevi.....	46
4.	Rezultati ispitivanja	47
4.1	Verifikacija servera	47
4.1.1	Testovi izdržljivosti	48
4.1.2	Funkcionalni testovi.....	49
4.2	Verifikacija SAT>IP klijentske biblioteke.....	49
5.	Pravci za dalji razvoj.....	51
5.1	Otključavanje zaštićenog sadržaja na serveru.....	51
5.2	Dodavanje birača kanala jednom ili grupi klijenata.....	52

6.	Poređenje SAT>IP i postojećih tehnologija.....	54
6.1	Metrike za evaluaciju tehnologije	54
6.1.1	Mrežna infrastruktura	54
6.1.2	Korisničko iskustvo	55
6.1.2.1	Garancija kvaliteta servisa.....	56
6.1.2.2	Vreme potrebno za menjanje kanala	56
6.2	Rezultati evaluacije	58
7.	Zaključak	60
8.	Literatura.....	62
9.	Dodatak A – Test program za verifikaciju SAT>IP klijentske biblioteke.....	63

SPISAK SLIKA

Slika 1 – Funkcionalne komponente	4
Slika 2 – Pristup sadržaju deljenom preko DLNA birača kanala.....	9
Slika 3 – Arhitektura IPTV sistema	13
Slika 4 – Arhitektura OTT sistema	15
Slika 5 - Format RTSP poruke	33
Slika 6 – Format RTCP APP paketa	34
Slika 7 – Struktura jednog datagrama	35
Slika 8 – Primer upotrebe dinamičke konfiguracije.....	36
Slika 9 – Primer statičke konfiguracije	36
Slika 10 – Primer integracije SAT>IP klijentske biblioteke u sistem.....	37
Slika 11 – Arhitektura SAT>IP klijentske biblioteke	38
Slika 12 – Arhitektura SAT>IP servera.	39
Slika 13 – Inicijalizacija UPnP biblioteke	41
Slika 14 – Glavna programska nit servera	42
Slika 15 – Obrada RTSP zahteva	43
Slika 16 – Obrada HTTP zahteva.....	44
Slika 17 – Programska nit za slanje bitskog toka RTSP klijentima.....	45
Slika 18 - Programska nit za slanje bitskog toka HTTP klijentima	46
Slika 19 – Predlog proširenja SAT>IP standarda	53
Slika 20 – Okruženje za merenje brzine promene kanala	58

SPISAK TABELA

Tabela 1 – Podela DLNA uređaja	5
Tabela 2 - Poređenje OTT i IPTV sistema.....	17
Tabela 3 - Format poruke za obaveštavanje uređaja o prisustvu na mreži	21
Tabela 4 – Format klijentske M-SEARCH poruke	23
Tabela 5 – Format RTSP SETUP poruke	26
Tabela 6 – Primeri formata Transport polja RTSP SETUP poruke.....	27
Tabela 7 – Format RTSP PLAY poruke	28
Tabela 8 – Format RTSP OPTIONS poruke.....	29
Tabela 9 – Format RTSP DESCRIBE poruke	30
Tabela 10 – Primer SDP listinga u okviru odgovora na DESCRIBE zahtev.....	31
Tabela 11 – Format RTSP TEARDOWN poruke.....	33
Tabela 12 – Format RTSP_URI niza karaktera	34
Tabela 13 – Opis polja RTCP APP paketa	34
Tabela 14 – Opis programskih niti SAT>IP servera.....	40
Tabela 15 – Memorijski zahtevi SAT>IP servera.....	46
Tabela 16 – Testovi izdržljivosti.....	48
Tabela 17 – Oficijalni testovi.....	49
Tabela 18 – Vreme potrebno za promenu kanala.....	58
Tabela 19 – Poređenje SAT>IP sa postojećim tehnologijama.....	59

SKRAĆENICE

DLNA	- <i>Digital Living Network Alliance</i>
IP	- <i>Internet Protocol</i> , Internet Protokol
RTP	- <i>Real-time Transport Protocol</i> , Protokol za transport u realnom vremenu
RTSP	- <i>Real Time Streaming Protocol</i> , Protokol za slanje bitskog toka
RTCP	- <i>RTP Control Protocol</i> , RTP kontrolni protokol
HTTP	- <i>Hypertext Transfer Protocol</i> , Protokol za prenos teksta
QoS	- <i>Quality of Service</i> , Kvalitet servisa
UPnP	- <i>Universal Plug and Play</i>
VoD	- <i>Video on Demand</i> , Video na zahtev
EPG	- <i>Electronic program guide</i> , Elektronski programski vodič
IPTV	- <i>Internet Protocol Television</i> , Protokol za prenos televizijskog sadržaja preko Interneta
OTT	- <i>over-the-top content</i> , Protokol za prenos sadržaja putem mreža otvorenog tipa
DRM	- <i>Digital rights management</i> , Digitlna zaštita autorskih prava
PVR	- <i>Personal Video Recorder</i> , Lični video snimač sadržaja
BER	- <i>Bit Error Rate</i> , Bitska Greška

1. Uvod

Najnoviji trendovi prenosa multimedijalnog sadržaja putem IP mreža usloveli su razvoj SAT>IP standarda, kojim je definisan prenos satelitskog, kablovskog i zemaljskog sadržaja između SAT>IP servera i prijemnika.

Razvoj višenamenskih uređaja (kao što su tableti, pametni telefoni, pametni televizori...) postavio je nove zahteve na tržištu ukazujući na potrebu za prenosom televizijskog sadržaja na više različitih uređaja u okviru kuće, zgrade, hotela, aerodroma, bolnice ...

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.

SAT>IP standard, za razliku od klasičnog prenosa sadržaja, ne podrazumeva postojanje birača kanala i demodulatora signala na prijemnicima televizijskog sadržaja. Birači kanala i demodulatori su koncentrisani na SAT>IP serveru, čija je uloga u sistemu da televizijski signal pretvori u oblik pogodan za prenos putem IP mreža. Konvertovan signal se dalje prosleđuje SAT>IP prijemnicima.

Za prenos sadržaja se koriste već postojeći standardi RTP i HTTP. SAT>IP standard u najvećoj meri predstavlja unapređenje postojećih komandnih protokola (RTSP i RTCP), čime je SAT>IP prijemnicima obezbeđeno upravljanje biračem kanala i demodulatorima na serveru na isti način kao i kod klasičnih prijemnika – umesto I2C magistrale za upravljanje se koristi RTSP protokol. SAT>IP prijemnici bez birača kanala i demodulatora imaju identičnu funkcionalnost kao u slučaju klasičnih prijemnika (programski vodič, prevod, izbor zvučne prezentacije na različitim jezicima).

Prenos sadržaja putem IP mreže pojednostavljuje projektovanje mreže za prenos televizijskog signala u okviru kuće, zgrade, hotela, bolnice, aerodroma itd. jer ne podrazumeva postavljanje dodatnih koaksijalnih kablova do svakog prijemnika već daje širok izbor prenosnih medija korišćenjem bežičnog prenosa, prenosa putem postojećih električnih instalacija, lokalnih mreža, optičkih kablova, itd.

Rad se sastoji od devet poglavlja. Prvo poglavlje je uvodno i opisuje zadatak i cilj rada.

U drugom poglavlju dat je prikaz postojećih tehnologija za prenos multimedijalnih sadržaja putem IP mreža.

Treće poglavlje daje opis SAT>IP standarda, realizacije servera na platformi sa ograničenim resursima i klijentske biblioteke.

Rezultati ispitivanja kvaliteta servera i klijenta je opisano u okviru četvrtog poglavlja.

U petom poglavlju su dati predlozi za dalji razvoj SAT>IP tehnologije.

Definisanje metrika za evaluaciju tehnologija kao u poređenje sa postojećim tehnologijama je dato u okviru šestom poglavlja.

Sedmo poglavlje sadrži zaključak rada, dok je osmo spisak korišćene literature.

Deveto poglavlje predstavlja dodatak radu koji sadrži listing izvornog koda programa koji je korišćen za funkcionalnu verifikaciju SAT>IP klijentske biblioteke.

2. Prikaz postojećih tehnologija

2.1 DLNA

Razvoj tehnologije i pojava pametnih telefona i televizora, kamera, računara i računarske tehnike u znatnoj meri je unapredio način stvaranja, skladištenja i reprodukcije multimedijalnih sadržaja. Kvalitet i količina slika, muzike i video zapisa koje jedan prosečan korisnik napravi je porastao, što je rezultovalo otežanim upravljanjem podacima. Ovaj trend je praćen unapređivanjem uređaja za reprodukciju multimedijalnih sadržaja i načinom za njihovo međusobno povezivanje koji su podržali vodeći svetski proizvođači iz tri različite grupe:

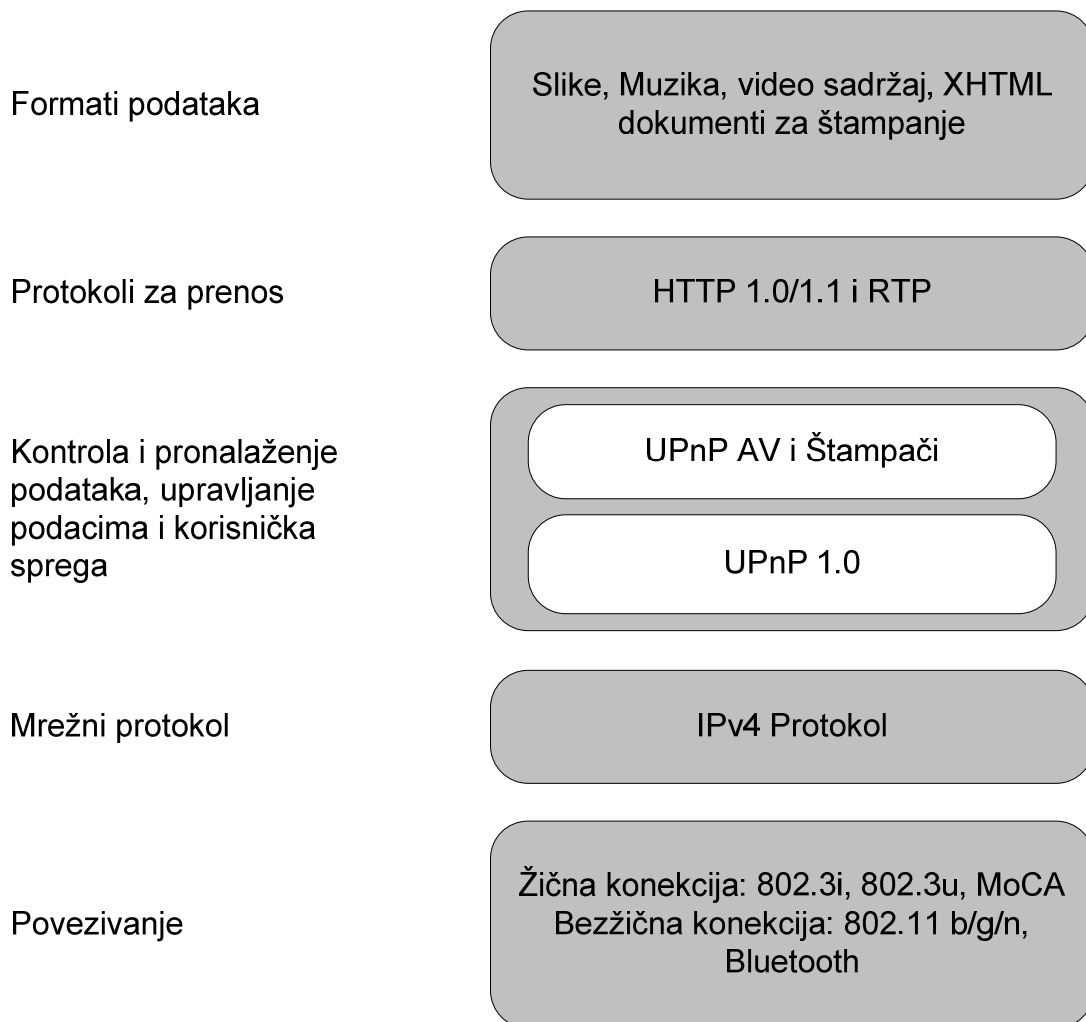
1. Proizvođači računara i računarske opreme
2. Proizvođači televizora, prijemnika satelitske, kablovske i zemaljske televizije,...
3. Proizvođači prenosnih uređaja

Porastom broja uređaja iz sve tri grupe pojavio se problem međusobne komunikacije uređaja i deljenja sadržaja između njih. Uglavnom su se proizvođači iz svake od navedene tri grupe trudili da u svom domenu obezbede efikasnu komunikaciju između uređaja. Sa druge strane korisnici su želeli da se obezbedi i pojednostavi komunikacija između uređaja iz sve tri grupe čime bi se pojednostavilo upravljanje i reprodukcija multimedijalnih sadržaja.

Jedno rešenje ovog problema je ponudila kompanija Sony koja je 2003 godine osnovala neprofitnu organizaciju DLNA (eng. Digital Living Network Alliance) koja je definisala pravila za razmenu multimedijalnih sadržaja.

2.1.1 Arhitektura DLNA kućne mreže

Da bi se postigla komunikacija između uređaja povezanih na mrežu definisan je skup funkcionalnih komponenti prikazan na Slici 1 [3].



Slika 1 – Funkcionalne komponente

IPv4 protokol predstavlja bazu za povezivanje DLNA uređaja i izabran je zbog sledećih osobina:

1. Dozvoljava nesmetan rad uređaja bez obzira na topologiju mreže
2. Omogućava vezu bilo kojeg uređaja na mreži sa Internetom
3. Široko je rasprostranjen i jeftin. Najpoznatiji standardi koji se koriste su Bluetooth, 802.3i, 802.3u, 802.11a, 802.11b, 802.11g, 802.11n.

Korišćenjem informacije o kvalitetu servisa (eng. Quality Of Service) optimizovano je deljenje mrežnih resursa između različitih uređaja na mreži. Aplikacije za prenos zvučnog i video zapisa su veoma osetljive na promene propusnog opsega i vreme odziva. Ukoliko bi svi paketi imali pravo da na isti način koriste mrežne resurse broj količina multimedijalnih sadržaja

koji se mogu preneti u okviru mreže bi bio znatno smanjen. Postavljanjem indeksa prioriteta paketa na osnovu sadržaja koji prenose i kvaliteta servisa se određuje način na koji će paketi koristiti mrežne resurse i u isto vreme povećava ukupan propusni opseg mreže.

DLNA je zasnovan na UPnP arhitekturi koja predstavlja skup protokola koji definiše pravila kojim uređaj na mreži može da otkrije druge uređaje na mreži, sakupi informacije o njihovim sposobnostima i da sa njima razmenjuje podatke koji u slučaju DLNA mogu biti spiskovi multimedijalnih sadržaja na svakom uređaju u mreži, samu multimedijalni uređaji, itd. Za razmenu podataka se koristi HTTP protokol koji je obavezan i RTP koji je opcioni.

2.1.2 Podela DLNA uređaja

DLNA uređaji se dele na 12 klasa koje su organizovane u 3 kategorije, kao što je prikazano u Tabeli 1 [2].

Kategorije	Klase
Kućni mrežni uređaji	Server multimedijalnog sadržaja
	Primalac multimedijalnog sadržaja
	Prikazivač multimedijalnog sadržaja
	Kontroler multimedijalnog sadržaja
	Štampač multimedijalnog sadržaja
Mobilni ručni uređaji	Prenosni server multimedijalnog sadržaja
	Prenosni primalac multimedijalnog sadržaja
	Prenosni kontroler multimedijalnog sadržaja
	Prenosni pošiljalac multimedijalnog sadržaja
	Prenosni prikazivač multimedijalnog sadržaja
Infrastrukturni uređaji	Funkcije za povezivanje prenosnih uređaja
	Jedinica za obezbeđivanje komunikacije

Tabela 1 – Podela DLNA uređaja

Ključne osobine kategorija DLNA uređaja su:

1. Svaka je optimizovana prema zahtevima radnog okruženja
2. Pravila za svaki od DLNA uređaja su fokusirana na međusobnu komunikaciju unutar jedne kategorije
3. Data su i pravila za komunikaciju između kategorija
4. Uređaj može istovremeno da bude član više kategorija

2.1.2.1 Kućni mrežni uređaji

Kućni mrežni uređaji (eng. Home Network Devices – HND) se koriste u kućnoj mreži i oslanjaju se na iste formate multimedijalnog sadržaja i zahteve za mrežnom vezom.

Server multimedijalnog sadržaja (eng. Digital Media Server – DMS) ima ulogu da prikuplja, snima, čuva i čini dostupnim digitalne multimedijalne sadržaje. U zavisnosti od izvora multimedijalnog sadržaja i zahteva za njegovo distribuiranje u računarsku mrežu DMS može da sprovodi zaštitu sadržaja. Neki od uređaja koji mogu pripadati ovoj klasi su:

1. Personalni računari
2. Uređaji za snimanje multimedijalnog sadržaja (eng. Personal Video Recorder – PVR)
3. Napredni dekoderi
4. Digitalni fotoaparati i kamere
5. Pametni telefoni

Primalac multimedijalnog sadržaja (eng. Digital Media Player – DMP) pronalazi sadržaje koje nudi DMS i omogućava njihovu reprodukciju. Ova klasa uređaja nije vidljiva drugim klasama u okviru kućne mreže. U ovi klasu mogu da spadaju:

1. Pametni televizori
2. Pametni telefoni
3. Kućni bioskop
4. Konzole za igranje

Prikazivač multimedijalnog sadržaja (eng. Digital Media Renderer – DMR) omogućavaju reprodukciju multimedijalnog sadržaja slično kao i DMP uređaji. Osnovna razlika u odnosu na DMP je što DMP uređaji ne mogu sami da pronađu sadržaje za reprodukciju na mreži.

Kontroler multimedijalnog sadržaja (eng. Digital Media Controller – DMC) pronalazi sadržaje sa DMS-a i uklapa ih sa mogućnostima reprodukcije DMR uređaja.

Štampač multimedijalnog sadržaja (eng. Digital Media Printer- DMP) pruža usluge štampanja DLNA mreži. Primarna namena ove klase uređaja je štampanje fotografija.

2.1.2.2 Mobilni ručni uređaji

Mobilni ručni uređaji (eng. Mobile Handheld Devices – MHD) u poređenju sa HND kategorijom imaju drugačije zahteve za formate multimedijalnog sadržaja i mrežnu povezanost. Pojedine klase kao što su prenosni server multimedijalnog sadržaja (eng. Mobile Digital Media

Server – MDMS), prenosni primalac multimedijalnog sadržaja (eng. Mobile Digital Media Player – MDMP) i prenosni kontroler multimedijalnog sadržaja (eng. Mobile Digital Media Controller - MDMC) imaju direktne duplikate u HND kategoriji. Takođe ova kategorija uređaja definiše i dve nove klase:

1. Prenosni pošiljalac multimedijalnog sadržaja (eng. Mobile Digital Media Uploader – MDMU) – šalje multimedijalni sadržaj na MDMS koji podržava ovu funkcionalnost
2. Prenosni preuzimač multimedijalnog sadržaja (eng. Mobile Digital Media Downloader – MDMD) – Preuzima multimedijalni sadržaj sa MDMS.

2.1.2.3 Infrastrukturni uređaji

Suštinska razlika dve prethodno navedene kategorije je format multimedijalnog sadržaja i načini njihovog povezivanja. Ipak, u okviru kućne mreže se mogu nalaziti uređaji iz obe kategorije. Kako bi se omogućilo njihovo povezivanje definisana je kategorije infrastrukturnih uređaja (eng. Home Infrastructure Device – HID) koja definiše međusobnu komunikaciju uređaja HND i MHD kategorija kroz sledeće klase:

1. Funkcije za povezivanje prenosnih uređaja (eng. Mobile Network Connectivity Function - MNCF) koje definišu vezu između HND i MHD mrežnog povezivanja
2. Jedinica za obezbeđivanje međusobne komunikacije uređaja (engl. Media Interoperability Unit - MIU) omogućava transformaciju multimedijalnog sadržaja između zahtevanih formata HND i MHD kategorija uređaja.

2.1.3 DLNA birač kanala

Digitalizacija televizijskog signala za ovaj vid prenosa multimedijalnog sadržaja je dala mogućnost obrade ovakvog signala korišćenjem šireg spektra uređaja i povećala njegovu upotrebljivost u okviru kućnih multimedijalnih sistema. Sa druge strane, u okviru kućnih mreža, korisnici i dalje poseduju dva odvojena uređaja, mrežni DLNA uređaj za deljenje multimedijalnog sadržaja u okviru kućne mreže i dekodirer digitalnog televizijskog sadržaja.

Uloga DLNA birača kanala jeste da centralizuje sve multimedijalne sadržaje i unificira način pristupa i reprodukcije tako što će se televizijskim sadržajima pristupati na način kao i u slučaju ostalih multimedijalnih sadržaja.

DLNA birač kanala je komponenta servera koja omogućuje prenos televizijskog sadržaja od servera ka prikazivačima multimedijalnog sadržaja.

2.1.3.1 Predstavljanje DLNA birača kanala drugim uređajima

DLNA birač kanala je predstavljen kao direktorijum (eng. Content Directory Servis – CDS) koji sadrži elemente od kojih svaki element predstavlja jedan servis. Ukoliko server poseduje više birača kanala svaki od njih može biti predstavljen kao zaseban objekat ili kao jedan objekat. Svaki objekat mora da ima naziv koji korisniku mora omogućiti da lako prepozna birač kanala, na primer na osnovu njegovog tipa.

Elementi koji su sadržani u objektu koji predstavlja birač kanala predstavljaju jedan servis koji treba da sadrži informacije u zvuku i slici, tako da jedinstveno predstave servis klijentima i omoguće pristup i reprodukciju multimedijalnog sadržaja koji predstavljaju.

Redosled elemenata treba da bude takav da reprezentuje redosled kanala koji se tipično prikazuju korisnicima. Na ovaj način se daje mogućnost DLNA kontrolnim tačkama da se kreću kroz listu na način isti kao i kod navigacije gore/dole u slučaju klasičnih dekodera. U zavisnosti od tipa birača kanala redosled može da prati frekvenciju, logički broj kanala dodeljen na primer od strane kablovskog ili satelitskog operatera, ...

Kada god je moguće, server treba svakom elementu objekta da dodeli broj kanala. Ovim se omogućuje direktan izbor kanala unošenjem njegovog broja.

Na kraju, elementi objekta moraju biti opisani imenom kanala.

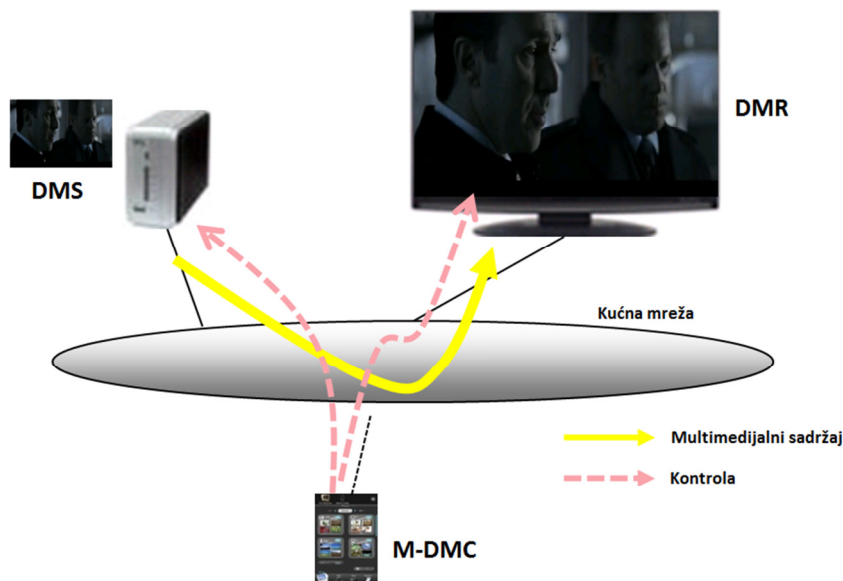
2.1.3.2 Pristupanje sadržaju deljenom preko objekta koji predstavlja birač kanala

Odabirom elementa objekta klijent na osnovu jedinstvenog identifikatora resursa (eng. Uniform Resource Identifier – URI) pristupa jednom od kanala deljenih na serveru. Ukoliko server prihvati vezu, birač kanala se konfigurira i traženi sadržaj se putem IP mreže šalje prikazivaču multimedijalnog sadržaja.

Tipičan scenario pristupa sadržaju koji se deli pomoću DLNA birača kanala je dat na slici 2. Kontrolna tačka (M-DMC) prikazuje dostupne kanale korisniku na način definisan objektom koji predstavlja birač kanala. Kada korisnik odabere određeni servis, prikazivač kanala korišćenjem URI niza karaktera šalje HTTP Get komandu serveru i time zahteva početak slanja televizijskog sadržaja preko IP mreže. Kada korisnik želi da promeni servis prikazivač kanala zatvara postojeću HTTP vezu i šalje novi HTTP Get zahtev, čime se otpočinje slanje novog servisa.

Ovde treba napomenuti da ne postoji povratna informacija koja bi trebala da obavesti kontrolnu tačku da je neka druga kontrolna tačka promenila kanal. Jednom kada prikazivač multimedijalnih sadržaja uspostavi HTTP vezu sa serverom, i server kasnije po nalogu druge kontrolne tačke promeni servis, server je dužan da zaustavi slanje podataka i prekine HTTP vezu

i na taj način objavi prikazivaču da je došlo do promene servisa. Prikazivač, sa druge strane, može u bilo kom trenutku da prekine HTTP vezu.



Slika 2 – Pristup sadržaju deljenom preko DLNA birača kanala

2.1.3.3 Primer opisa DLNA birača kanala

Primer opisa servera koji sadrži dva birača kanala, jedan televizijski i drugi za FM radio je dat kroz sledeći XML listing [\[1\]](#):

```
<DIDL-Lite
  xmlns="urn:schemas-upnp-org:metadata-1-0/DIDL-Lite/"
  xmlns:dc="http://purl.org/dc/elements/1.1/"
  xmlns:upnp="urn:schemas-upnp-org:metadata-1-0/upnp/"
  xmlns:dlna="urn:schemas-dlna-org:metadata-1-0/">
  <!-- Root Container -->
  <!-- (NOTE: XML Comments prohibited per 7.2.29 and are only included for clarity) -->
  <container id="0" parentID="-1" restricted="1" childCount="2">
    <dc:title>DLNA Device</dc:title>
    <upnp:class>object.container</upnp:class>
    <!-- TV Tuner Container -->
    <container id="1" parentID="0" restricted="1" childCount="2">
      <dc:title>TV Tuner</dc:title>
      <upnp:class>object.container</upnp:class>
      <dlna:containerType>Tuner_1_0</dlna:containerType>
      <!-- TV Channels -->
      <item id="1-1" parentID="1" restricted="1">
        <!-- Full Description -->
        <dc:title>Cartoons, Cartoons, Cartoons</dc:title>
        <upnp:class>object.item.videoItem.videoBroadcast</upnp:class>
        <upnp:genre>Movie</upnp:genre>
        <upnp:channelNr>2</upnp:channelNr>
        <upnp:channelName>PBS</upnp:channelName>
        <res protocolInfo="http-get:*:video/mpeg:DLNA_PN=MPEG_PS">
```

```
    http://192.168.0.20:58849/Tuner1/ch2.mpg
  </res>
</item>
<item id="1-2" parentID="1" restricted="1">
  <!-- Minimal Description -->
  <dc:title>Channel 4</dc:title>
  <upnp:class>object.item.videoItem.videoBroadcast</upnp:class>
  <upnp:channelNr>4</ upnp:channelNr>
  <res protocolInfo="http-get:*:video/mpeg:DLNA_PN=MPEG_PS">
    http://192.168.0.20:58849/Tuner1/ch4.mpg
  </res>
</item>
</container>
<!-- FM Radio Tuner Container -->
<container id="2" parentID="0" restricted="1" childCount="3">
  <dc:title>FM Radio Tuner</dc:title>
  <upnp:class>object.container</upnp:class>
  <dlna:containerType>Tuner_1_0</dlna:containerType>
  <!-- FM Radio Channels -->
  <item id="2-1" parentID="2" restricted="1">
    <!-- Preset #1 -->
    <dc:title>FM 89.9</dc:title>
    <upnp:class>object.item.audioItem.audioBroadcast</upnp:class>
    <upnp:channelNr>1</upnp:channelNr>
    <upnp:channelName>FM 89.9</upnp:channelName>
    <res protocolInfo="http-get:*:audio/L16:DLNA_PN=LPCM">
      http://192.168.0.20:58849/Tuner2/ch1.L16
    </res>
  </item>
  <item id="2-2" parentID="2" restricted="1">
    <!-- Preset #2 -->
    <dc:title>FM 101.9</dc:title>
    <upnp:class>object.item.audioItem.audioBroadcast</upnp:class>
    <upnp:channelNr>2</upnp:channelNr>
    <res protocolInfo="http-get:*:audio/L16:DLNA_PN=LPCM">
      http://192.168.0.20:58849/Tuner2/ch2.L16
    </res>
  </item>
  <item id="2-3" parentID="2" restricted="1">
    <!-- Preset #3 -->
    <dc:title>FM 95.5</dc:title>
    <upnp:class>object.item.audioItem.audioBroadcast</upnp:class>
    <upnp:channelNr>3</upnp:channelNr>
    <res protocolInfo="http-get:*:audio/L16:DLNA_PN=LPCM">
      http://192.168.0.20:58849/Tuner2/ch3.L16
    </res>
  </item>
</container>
</container>
</DIDL-Lite>
```

2.2 IPTV

Tradicionalni način emitovanja televizijskih sadržaja podrazumeva korišćenje jednosmernog prenosa od distributera ka korisnicima, korišćenjem antena (satelitskih ili zemaljskih) ili koaksijalnim kablovima u slučaju kablovskih operatera. Rasprostranjenost dvosmerne komunikacije u vidu Internet veze je dovela do postavljanja novih i složenijih zahteva krajnjih 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 namenski namenjen za prenos audio i video sadržaja. Sa druge strane IP mreže služe za prenos širokog skupa podataka sa velikog broja izvora putem jednog kanala kao što su na primer elektronska pošta, internet stranice, poruke, prenos glasa... Sa svim dodatnim informacijama koje se prenose putem jednog kanala Internet je 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 signala.

Uprkos činjenici da IP i prenos video sadržaja nisu kompatibilni tržište za prenos televizijskog sadržaja je u ekspanziji iz sledećih razloga [4]:

- Internet je postao široko rasprostranjen i nalazi se u velikom broju domaćinstava, što omogućava operatorima da pružaju usluge bez potrebe da projektuju i instaliraju svoju, namensku mrežnu infrastrukturu
- IP pojednostavljuje unapređivanje 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.
- 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
- Broj korisnika IP se ubrzano povećava
- IP je tehnologija koja ima širok spektar primene kao što su na primer elektronska pošta, rad sa bankama, pristup internet stranicama, deljenje sadržaja, ...

Pored navedenih činjenica koje idu u prilog prenosu televizijskog sadržaja putem IP mreža postoje i nedostaci ove tehnologije koje je za njeno bolje razumevanje važno navesti:

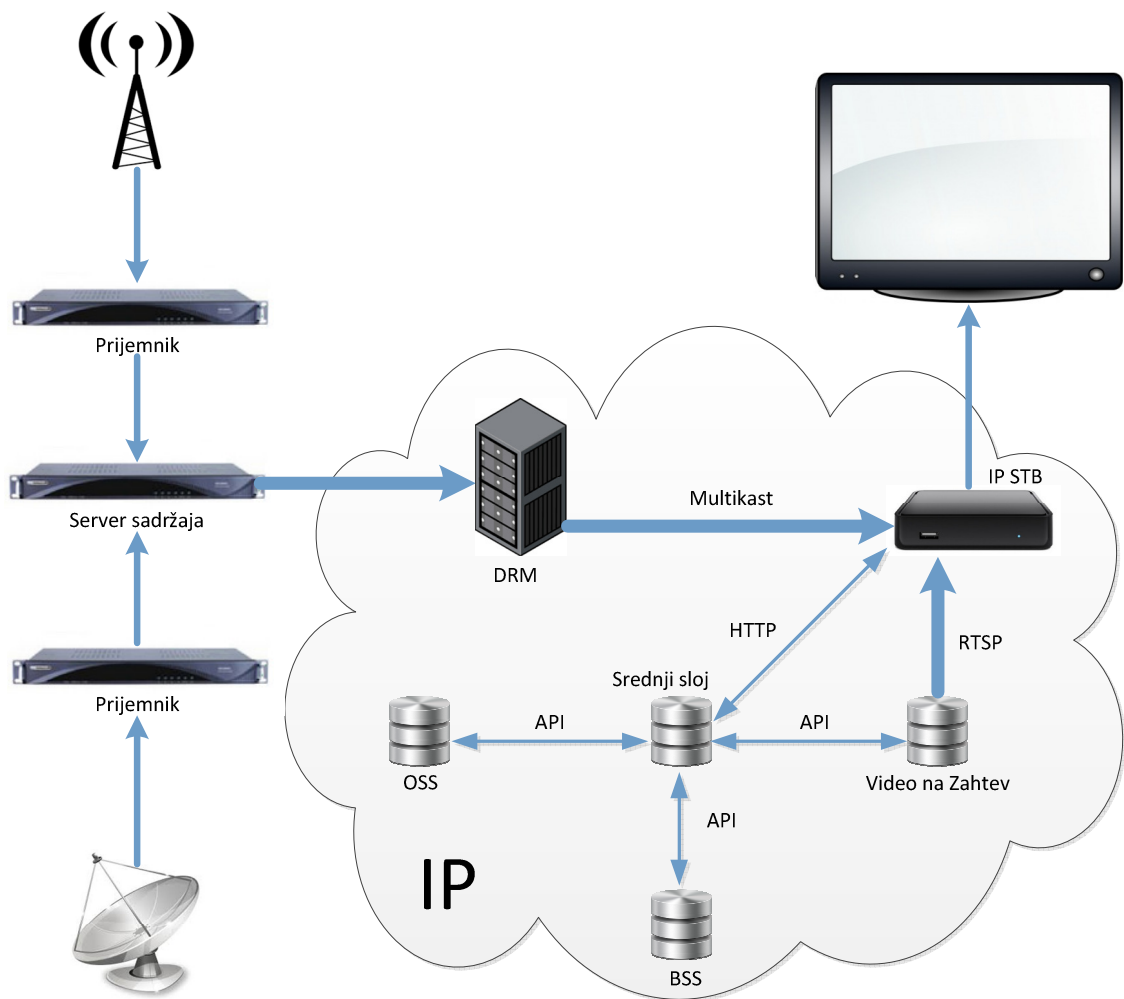
- prvi nedostatak je ekonomske i istorijske prirode i povezan je sa osnovnom idejom Interneta – da je sve na njemu besplatno, što otežava distributerima opstanak na tržištu pored ilegalnog deljenja muzike i video sadržaja,
- drugi nedostatak je tehničke prirode i ogleda se u potrebi za prilagođavanjem multimedijalnog sadržaja promenljivom propusnom opsegu IP mreže koja može da varira od korisnika do korisnika,
- treći nedostatak je kombinovanje multimedijalnog sadržaja koji ima visoke zahteve za propusnim opsegom sa ostalim paketima koji se razmenjuju putem istog prenosnog kanala i određivanje prioriteta prilikom njihovog slanja.

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

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

2.2.1 Arhitektura IPTV mreže

Primer arhitekture IPTV sistema je prikazan na slici 3. Televizijski sadržaji se prikupljaju preko prijemnika i potom prosleđuju dalje ka serveru sadržaja. Nakon toga se obavlja zaštita bitskog toka nekom od metoda za upravljanje digitalnim pravima (eng. Digital Right Management – DRM). Prenos tako zaštićenog sadržaja do krajnjih korisnika kontroliše deo sistema koji se zove srednji sloj (eng. Middleware). Od centralnih lokacija televizijski sadržaji se distribuiraju korišćenjem multikast protokola ka IPTV prijemnicima koji primljeni bitski tok otključavaju, dekoduju i reprodukuju na TV prijemniku.



Slika 3 – Arhitektura IPTV sistema

Video sadržaj može doći iz više izvora:

- Satelitska televizija
- Zemaljska televizija
- Video sadržaj koji se nalazi na lokalnom serveru kojem se pristupa putem servisa video na zahtev
- Sadržaj koji se uživo emituje iz lokalnih javnih, obrazovnih ili vladinih ustanova
- Sadržaj emitovan sa drugih IPTV sistema (na primer od regionalnog ka lokalnom)

Server sadržaja prima signale iz različitih izvora, koduje ih i prenosi ka odlaznoj transportnoj mreži. Ovaj server može da opsluži server za video na zahtev, servere srednjeg sloja, servere za distribuciju elektronskog programskog vodiča i da podrži operative sisteme distributera servisa.

Server za zaštitu video sadržaja koduje signal i samim tim daje mogućnost distributeru video sadržaja da kontroliše njegov prijem i upotrebu.

Srednji sloj predstavlja distribuirani sistem čija je uloga da poveže sve komponente IPTV sistema i nalazi se na IPTV serverima i prijemnicima obezbeđujući [5]:

- Integraciju sa izvorima video sadržaja, DRM serverima, IPTV prijemnicima i mrežom
- Interaktivne usluge
- Upravljanje različitim nivoima pretplatničkih paketa i usluga
- Vezu između elektronskog programskog vodiča i samog video sadržaja
- Ažuriranje programske podrške IPTV prijemnika

2.2.2 Prenos sadržaja putem IPTV sistema

IPTV sistemi za prenos video sadržaja od distributera do krajnjih korisnika koriste IP multikast. Korišćenjem ovog protokola u velikoj meri se smanjuje potreba za širokim propusnim opsegom kanala za prenos video sadržaja, tako što se svaki servis dostavlja do svakog korisnika u isto vreme čime se:

- sprečava redundantnosti u prenosu
- obezbeđuje efikasniju kontrolu mrežnog saobraćaja
- smanjuje opterećenost IPTV servera

Distribucija visoko kvalitetnog video sadržaja kroz pouzdanu mrežu predstavlja osnovnu ideju IPTV sistema. U okviru mreže za prenos, na prenos podataka mogu da utiču razni faktori:

- gubitak paketa
- varijacije u vremenskom razmaku primljenih paketa u odnosu na konstantni vremenski razmak između paketa na predajnoj strani – JITTER, koje mogu da nastanu kao posledica preopterećenosti mreže, nepravilnog raspoređivanja paketa ili greške u konfiguraciji mreže
- unošenje grešaka u paket tokom prenosa

Uticaj navedenih faktora se minimizuje procenom kvaliteta servisa (eng. Quality of Service – QoS), čija je uloga da obezbedi uslove za prenos video signala visokog kvaliteta. Za razliku od interneta, koji je distribuirani sistem na kojem ne postoji centralizovana kontrola i gde je procena kvaliteta servisa otežana, IPTV je mreža zatvorenog tipa u okviru koje distributeri pored IPTV sistema kontrolišu i pristup internetu svakog korisnika. Pravilnom konfiguracijom IPTV mreže distributeri mogu da obezbede i garantuju korisnicima nesmetan prijem visoko kvalitetnog video sadržaja na sledeći način:

- obezbeđujući da maksimalni propusni opseg prenosnog kanala ka svakom domaćinstvu bude veći od broja bita po sekundi potrebnih za prenos jednog IPTV servisa

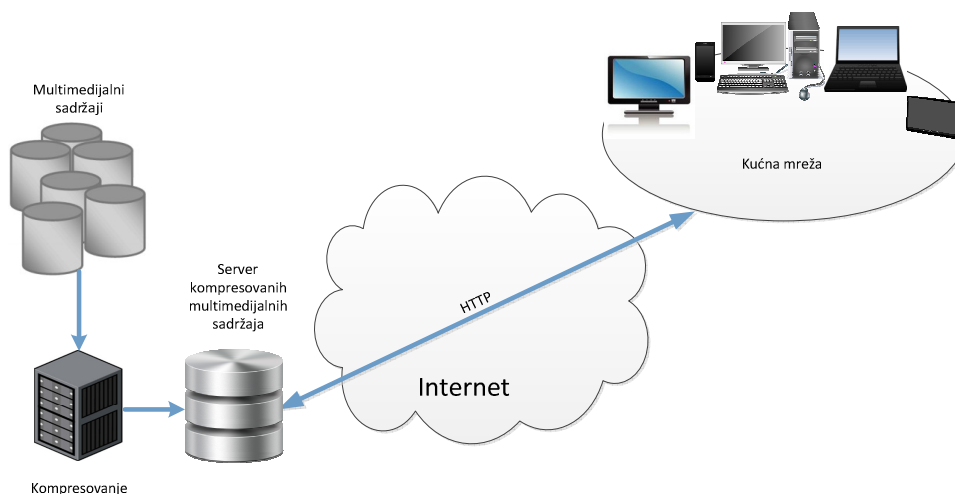
- davanje većeg prioriteta IPTV paketima u odnosu na druge pakete u mreži, čime se osigurava da će IPTV paketi biti nesmetano prosleđeni ka prijemniku u okviru kućne mreže.

2.3 OTT

Za razliku od IPTV sistema gde se prenos video sadržaja obavlja putem mreža zatvorenog tipa, OTT (eng. Over The Top – OTT) sistemi za prenos video sadržaja koriste internet. Kako je internet distribuirani sistem na kojem ne postoji centralizovana kontrola, glavni problem OTT sistema predstavlja nesmetan prenos sadržaja do korisnika i garancija konstantnog kvaliteta reprodukcije. Za razliku od IPTV distributera, gde izgradnja i održavanje zatvorene mreže predstavljaju najskuplje procese, kod OTT distributera akcentat je stavljen na povećanje stepena zadovoljstva korisnika korišćenjem dva ključna bloka: korišćenjem bitskog toka prilagodivog propusnom opsegu kanala za prenos i izgradnjom mreža za distribuciju sadržaja.

2.3.1 Arhitektura OTT sistema

OTT sistemi, čiji je jedan primer prikazan na slici 4 [6], multimedijalne sadržaje preuzimaju iz biblioteka koje se nalaze na serveru i nakon kompresije ih putem interneta prosleđuju ka korisničkim uređajima koji predstavljaju klijente sposobne da preuzmu i obrade sadržaje kod kojih se bitski tok od strane servera prilagođava trenutnom propusnom opsegu prenosnog kanala. Kao protokol za prenos podataka koristi se HTTP (eng. Hypertext Transfer Protocol – HTTP). Gledano sa poslovne strane, distributeri OTT sadržaja nemaju potrebu da vode računa o troškovima izgradnje i održavanja mreža zatvorenog tipa iz razloga što se prenos sadržaja obavlja preko interneta. Sa druge strane, glavni problem predstavlja prenos digitalnih multimedijalnih sadržaja koji predstavljaju precizno vremenski određen kontinualan tok podataka putem interneta, koji je slabo vremenski određen sistem.



Slika 4 – Arhitektura OTT sistema

2.3.2 Prenos video sadržaja putem OTT sistema

Unapređivanje korisničkog iskustva i prevazilaženje problema prouzrokovanih Internetom se rešava korišćenjem prenosa sa prilagodivim bitskim tokom i izgradnjom mreža za distribuciju sadržaja.

Prilikom kompresovanja video sadržaj se deli na delove koji se u isto vreme kompresuje različitim parametrima kompresije, čime se obezbeđuju višestruke kopije sadržaja koje variraju u nivou kvaliteta, a samim tim i u zahtevu za propusnim opsegom kanala za prenos. Ovako dobijeni delovi se zatim grupišu u segmente čija je dužina obično između dve i deset sekundi. Segmenti se čuvaju u zasebnim datotekama ili u okviru jedne datoteke sa logički odvojenim segmentima tako da je omogućen pristup segmentima po stepenu kompresije. Kada klijent započne prenos sadržaja, on prvo od servera dobija datoteku koja u sebi sadrži listu podržanih kvaliteta za traženi sadržaj.

Tokom prenosa, na osnovu trenutnih uslova i propusnog opsega kanala za prenos, klijent može da izabere element iz liste i time tokom prijema i obrade sadržaja optimizuje kvalitet reprodukcije. Klijent je odgovoran za nadgledanje trenutnog stanja mreže i propusnog opsega kanala. Na osnovu tako dobijenih podataka klijent obavlja procenu vremena potrebnog sa prenos narednog segmenta sadržaja koje varira u zavisnosti od stepena kompresije na serverskoj strani i trenutnih uslova za prenos. Na osnovu tako dobijenog procenjenog vremena klijent bira odgovarajući segment i time minimizuje verovatnoću da će naredni segment zakasniti sa jedne strane, a sa druge strane održava kvalitet reprodukcije na najboljem mogućem nivou.

U zavisnosti od geografske lokacije klijenta, vreme pristupa segmentu istog kvaliteta može da varira. Ukoliko se na primer server nalazi u Evropi korisnici iz Evrope će imati brži pristup istom sadržaju nego klijenti iz Azije. Vreme pristupa, a samim tim i isti kvalitet reprodukcije, se obezbeđuje izgradnjom mreža za distribuciju sadržaja [6] (eng. Content Delivery Network – CDN), koje predstavljaju velike distribuirane sisteme servera koji se nalaze na različitim geografskim lokacijama. Cilj ovih sistema je da krajnjim korisnicima obezbede brži pristup video sadržajima.

2.4 Poređenje postojećih tehnologija

OTT i IPTV sistemi predstavljaju dve tehnologije koje su u konstantnom porastu na tržištu koje se bavi distribucijom multimedijalnih sadržaja. Oba sistema koriste IP tehnologije za prenos sadržaja od operatera do pretplatnika. Za razliku od njih DLNA sistemi nisu striktno vezani za operatera i dostupni su svakom korisniku, dajući mu mogućnost da u okviru svog doma napravi

lokalnu mrežu za deljenje audio, video sadržaja, ličnih fotografija,... Sve tri tehnologije zajedno imaju dosta prednosti i u znatnoj meri olakšavaju organizaciju, pristup i upotrebu informacija čiji kvalitet, broj i dostupnost eksponencijalno rastu poslednjih godina.

2.4.1 Poređenje IPTV i OTT sistema

Osnovna razlika između OTT i IPTV sistema je da iako oba koriste internet za prenos podataka, OTT se obavlja putem otvorene mreže dok se za IPTV koristi mreža zatvorenog tipa .

IPTV sistemi krajnjim korisnicima nude garantovan kvalitet što je posledica korišćenja zatvorene mreže što rezultuje činjenicom da je pristup plaćenim sadržajima ograničen. Sa druge strane, OTT sistemi omogućuju pristup plaćenim sadržajima iz drugim mreža. Otvorenost OTT sistema za sobom vuče posledicu koja se ogleda u činjenici da ne postoje garancije kvaliteta prenosa.

U poslednje vreme OTT doživljava popularnost zbog porasta propusnog opsega kanala za prenos na internetu i dostupnosti relativno jeftinih servisa kao što su na primer Netflix i Hulu.

Detaljno poređenje OTT i IPTV sistema dato je u tabeli 2 [\[7\]](#).

	OTT	IPTV
Distribucija sadržaja	Koristi mreže otvorenog tipa	Koristi mreže zatvorenog tipa
Vlasništvo nad mrežom	Ne postoji potreba distributera za izgradnjom sopstvene mreže	Zatvorena mreža kojoj mogu da pristupe samo pretplatnici
Kvalitet servisa (QoS)	Ne postoje garancije na kvalitet.	Garantovan kvalitet servisa kao posledica zatvorene mreže.
Primeri	YouTube, Netflix, Hulu, Amazon, BBC, iPlayer,...	IPTV Telekoma Srbije
Protokol	Koristi HTTP protokol. Najnoviji trendovi su korišćenje protokola koji prilagođavaju bitski tok propusnom opsegu prenosnog kanala.	Koristi UDP multikast protokol
Nedostaci/izazovi	Nema garancije kvaliteta prenosa sadržaja koji se emituje uživo	IPTV je dosta skuplji, ima veliku konkurenciju u kablovskim operaterima
Prednosti	Jeftin, fleksibilan u smislu dostupnosti sadržaja na raznim uređajima	Interaktivni servisi, garantovan kvalitet servisa

Tabela 2 - Poređenje OTT i IPTV sistema

Iako je trenutno IPTV rasprostranjeniji nego OTT, analize tržišta i tehnologija predviđaju da će u narednih par godina OTT preuzeti glavnu ulogu u prenosu video sadržaja putem IP mreža.

2.4.2 Pozicija DLNA sistema

Za razliku od IPTV i OTT sistema gde se video sadržaj distribuira od operatera ka pretplatnicima, DLNA tehnologija dozvoljava svakom korisniku da u okviru svog doma formira sopstvene mreže i omogući pristup multimedijalnim sadržajima svim uređajima koji imaju mogućnost pristupa mreži i mogu da reprodukuju deljene sadržaje. Kombinujući DLNA uređaje korisnik sam konfigurise DLNA sistem i tako na osnovu svojih potreba formira sistem za deljenje sadržaja u okviru kućne mreže bez potrebe za pristupom internetu.

3. Opis predloženog rešenja

3.1 Opis SAT>IP protokola

SAT>IP predstavlja protokol kojim se definiše način prenosa televizijskog sadržaja koji se emituje uživo ka IP klijentima putem IP mreža. Po svojoj definiciji on ne predstavlja specifikaciju uređaja, već samo definiše komunikacioni protokol, dok je tržištu ostavljena sloboda da po svojim potrebama određuje njegovu primenu. Osnovna ideja ovog protokola je da u vremenu u kojem popularnost IPTV i OTT usluga raste pruži mogućnost krajnjem korisniku da televizijske sadržaje primljene sa satelita, kablovske ili zemaljske digitalne televizije prati ne samo na TV aparatima već na svim uređajima koji se mogu povezati na IP mrežu i koji mogu da dekoduju i prikažu video sadržaje.

Protokol, u svojoj osnovi, uređaje deli na dva dela [\[9\]](#):

- SAT>IP klijenti. Klijenti imaju mogućnost izbora i prijema televizijskih programa. Primeri uređaja koji mogu imati SAT>IP klijentsku funkcionalnost su: dekoderi sa IP korisničkom spregom; aplikacije za pametne telefone, tablete, personalne računare...
- SAT>IP serveri. Serveri imaju ulogu da odgovaraju na zahteve klijenata i prosleđuju im tražene servise. Serveri se mogu implementirati u obliku kućnih IP adaptera ili mogu biti integrisani u standardnu opremu za prijem kao što je na primer LNB (eng. Low Noise block downconverter).

Po svojoj arhitekturi SAT>IP dozvoljava prijem televizijskog signala na uređajima koji u sebi nemaju birač kanala koji su, ako se posmatra standardni prijemnik, izmešteni u zaseban uređaj gde se kontrola umesto I2C magistralom obavlja korišćenjem IP protokola.

SAT>IP protokol se bazira na postojećim IP protokolima [\[9\]](#):

- UPnP za adresiranje, otkrivanje i opis uređaja na mreži
- RTSP i HTTP za kontrolu
- RTP i HTTP za prenos televizijskih sadržaja.

3.1.1 Adresiranje

Adresiranje je proces u kojem SAT>IP uređaj dobavlja IP adresu što predstavlja prvi korak pre bilo koje druge komunikacije. Standard prati UPnP specifikaciju nudeći dve opcije za pribavljanje adrese:

- DHCP – Svaki SAT>IP uređaj mora biti DHCP (eng. Dynamic Host Configuration - DHCP) klijent koji prilikom povezivanja na IP mrežu šalje zahtev za dobijanje IP adrese DHCP serveru.
- Automatsko IP adresiranje – ukoliko na mreži ne postoji DHCP server SAT>IP uređaju moraju sami konfigurisati IP adrese koje treba da budu u opsegu 169.254/16.

3.1.2 Otkrivanje

Tokom faze otkrivanja SAT>IP server objavljuje svoje prisustvo na mreži drugim serverima i klijentima, dok klijenti prilikom pristupanja mreži obavljaju pretragu postojećih servera.

Otkrivanje uređaja je bazirano na SSDP protokolu (eng. Simple Service Device) koji je specificiran UPnP 1.1 (eng. Universal Plug and Play) arhitekturom. Svaki SAT>IP server predstavlja UPnP uređaj ili UPnP kontrolnu tačku, dok su klijenti kontrolne tačke.

S obzirom na činjenicu da se za otkrivanje SAT>IP uređaja koristi SSDP kao deo UPnP protokola, da bi se razlikovali od ostalih uređaja u mreži koji su saglasni sa UPnP protokolom, koristi se jedinstveni identifikator tipa uređaja URN koji ima sledeći oblik:

Urn:ses-com:device:SatIPServer:1

gde je:

- “ses-com” ime domena
- “SatIPServer” ime tipa uređaja
- “1” verzija tipa uređaja

Svaku instancu SAT>IP uređaja u okviru mreže jedinstveno određuje UUID (eng. Universally Unique Identifier). Format ovog identifikatora je:

4B-2B-2B-2B-6B

gde B predstavlja bajt upisan kao dve heksadecimalne cifre.

Prilikom pristupa mreži svaki server šalje tri poruke (eng. NOTIFY) kojima ostale uređaje obaveštava o svom prisustvu na SSDP adresu 239.255.255.250 i port 1900. Format poruke je dat u tabeli 3 [9]:

Metod		NOTIFY
Zahtev	Protokol	MUDP (multikast UDP)
	Zaglavlje	Poruka treba da sadrži "DEVICEID.SES.COM" polje u svom zaglavlju sa identifikatorom servera
Odgovor		Ne zahteva slanje odgovora
Primer	Zahtev	NOTIFY * HTTP /1.1 HOST: 239.255.255.250:1900 CACHE-CONTROL: max-age=1800 LOCATION: http://<satip_server_addr>/<description>.xml NT: upnp:rootdevice ili uuid:<UUID> ili urn:ses-com:device:SatIPServer:1 NTS: ssdp:alive SERVER: OS/version UPnP/1.1 product/version USN: <jedinstveno ime servisa> BOOTID.UPNP.ORG: <redni broj paljenja uređaja> CONFIGID.UPNP.ORG: <konfiguracioni indeks> SEARCHPORT.UPNP.ORG: <port za pretragu> DEVICEID.SES.COM: <identifikator uređaja> <CRLF>

Tabela 3 - Format poruke za obaveštavanje uređaja o prisustvu na mreži

NTS polje paketa treba da sadrži ssdp:alive.

Kao što je rečeno, prilikom obaveštavanja ostalih uređaja pri pristupanju mreži šalju se tri NOTIFY poruke i u svakoj od njih se menja vrednost NT polja tako da se uređaj prijavi kao rootdevice, kao uređaj sa jedinstvenom UUID vrednošću i kao SAT>IP uređaj respektivno.

USN (eng. Unique Service Name) se takođe menja u svakoj od tri NOTIFY poruke i predstavlja identifikator čija vrednost zavisi od polja NT (detaljan proces generisanja vrednosti USN polja je dat u UPnP 1.1 specifikaciji).

Polje LOCATION sadrži putanju do do XML datoteke koja sadrži opis uređaja.

Svaka poslata NOTIFY poruka ima svoj period važenja i nakon isteka tog perioda moraju se ponovo poslati. Ovaj period je određen poljem CACHE-CONTROL i za SAT>IP uređaje iznosi 1800 sekundi. Vreme kada će se NOTIFY poruke ponovo poslati uzima se kao nasumično

izabran broj iz intervala od 900 do 1800 sekundi, čime se smanjuje verovatnoća opterećenosti mreže NOTIFY poruka i garantuje da će poruke uvek biti poslate pre isteka perioda važenja.

Vrednost polja BOOTID.UPONP.ORG se uvećava za jedan svaki put kada se uređaj prijavljuje na mrežu i mora biti sačuvan u memoriji čiji se sadržaj ne gubi ako što se uređaj ugasi.

CONFIGID.UPNP.ORG polje predstavlja konfiguracioni broj uređaja. Ovaj broj mora da bude identičan polju “configID” koje se nalazi u XML datoteci koja opsuje uređaj.

Opciono polje SEARCHPORT.UPNP.ORG nije preporučljivo koristiti iz razloga što SAT>IP uređaji treba da slušaju unicast M-SEARCH poruke samo na portu 1900. Ovo polje treba koristiti samo ukoliko port 1900 nije dostupan.

U okviru jedne mreže može da postoji više SAT>IP uređaja. Kako bi se oni međusobno razlikovali, svakom serveru je dodeljen jedinstveni identifikator uređaja koji se prenosi poljem DEVICEID:SES:COM.

Osnovna pravila kojih se uređaji moraju pridržavati prilikom pristupa ili prilikom odlaska sa mreže su:

- SAT>IP server prilikom pristupa mreži treba da objavi svoj dolazak slanjem tri NOTIFY poruke
- SAT>IP server koji se nalazi na mreži svoje prisustvo na mreži mora objavljivati slanjem NOTIFY poruka u vremenu određenom vrednošću polja CACHE-CONTROL.
- SAT>IP server koji napušta mrežu mora objaviti svoj odlazak slanjem tri NOTIFY poruke kod kojih je vrednost NTS polja “ssdp:byebye”. U ovom slučaju SSDP poruka ne sme sadržati polja CACHE-CONTROL, LOCATION, SERVER i DEVICEID.SES.COM
- SAT>IP server koji menja mrežu treba da objavi svoj odlazak sa stare mreže i prijavi svoj dolazak na novu po već objašnjenim pravilima
- SAT>IP klijenti ne objavljuju svoje prisustvo na mreži tako da server njihov odlazak ne može da detektuje. Stoga su serveri dužni da detektuju odsustvo klijenta i prekinu slanje bitskog toka kako bi rasteretili prenosni kanal. Mehanizmi koje serveri koriste su period isteka RTSP sesije i IGMP upiti za prisutne članove multikast grupe.

SAT>IP klijenti prilikom pristupa mreži šalju tri multikast M-SEARCH poruke sa razmakom od 100 ms i time otkrivaju prisustvo SAT>IP servera. Format ove poruke je dat u tabeli 4 [9].

Metod		M-SEARCH
Zahtev	Protokol	MUDP (Multicast UDP=
	Zaglavlje	Specificirano u UPnP 1.1 specifikaciji
Odgovor	Linija	HTTP/1.1 200 OK
	Protokol	UDP
	Zaglavlje	Specificirano u UPnP 1.1 specifikaciji
Primer	Zahtev	M-SEARCH * HTTP/1.1 HOST: 239.255.255.250:1900 MAN: "ssdp:discover" MX: 2 ST: urn:ses-com:device:SatIPServer:1 USER-AGENT: OS/version UPnP/1.1 product/version <CRLF>
	Odgovor	HTTP/1.1 200 OK CACHE-CONTROL: max-age=1800 DATE: <date> EXT: LOCATION: http://<SatIPServer_IP_Address>/<description>.xml SERVER: OS/version UPnP/1.1 product/version ST: urn:ses-com:device:SatIPServer:1 USN:uuid:01234567-0123-0123-0123-0123456789ab::urn:ses-com:device:SatIPServer:1 BOOTID.UPNP.ORG: <bootID> CONFIGID.UPNP.ORG: <configID> SEARCHPORT.UPNP.ORG: <searchPort> <CRLF>

Tabela 4 – Format klijentske M-SEARCH poruke

Polje MAN mora imati vrednost "ssdp:discover".

Tip uređaja koji se traži na mreži je određen poljem ST i u slučaju SAT>IP uređaja mora da ima vrednost istu kao URN polje servera.

Da bi se ravnomerno rasporedio broj zahteva koje mora da obradi SAT>IP uređaj kao UPnP kontrolna tačka koristi se polje MX koje određuje vreme za koje će odgovor biti odložen i koje se računa kao slučajan broj u intervalu od 0 do 2 sekunde.

3.1.3 Opisivanje SAT>IP uređaja u mreži

Tokom faze opisivanja SAT>IP uređaji daju detaljan opis UPnP kontrolnim tačkama na mreži putem XML datoteke čija adresa se nalazi u polju LOCATION svake NOTIFY poruke čije NTS polje ima vrednost "ssdp:alive". Primer XML datoteke je [9]:

```
<?xml version="1.0"?>
<root xmlns="urn:schemas-upnp-org:device-1-0" configId="konfiguracioni broj">
  <specVersion>
    <major>1</major>
    <minor>1</minor>
  </specVersion>
  <device>
    <deviceType>urn:ses-com:device:SatIPServer:1</deviceType>
    <friendlyName>Ime uređaja</friendlyName>
    <manufacturer>Proizvođač</manufacturer>
    <manufacturerURL>Link ka sajtu proizvođača</manufacturerURL>
    <modelDescription>Opis modela uređaja</modelDescription>
    <modelName>Ime modela uređaja</modelName>
    <modelName>Broj modela uređaja</modelName>
    <modelURL>Link ka modelu uređaja </modelURL>
    <serialNumber>Serijski broj proizvođača</serialNumber>
    <UDN>uuid:UUID</UDN>
    <UPC>Univerzalni broj proizvoda</UPC>
    <iconList>
      <icon>
        <mimetype>slika/format</mimetype>
        <width>širina u pikselima</width>
        <height>dužina u pikselima</height>
        <depth>broj boja</depth>
        <url>Link ka ikoni</url>
      </icon>
    </iconList>
    <presentationURL>Link ka korisničkoj sprezi uređaja</presentationURL>
  </device>
</root>
```

Polje configID treba da bude jednako sa vrednošću polja CONFIGID.UPNP.ORG koje je korišćeno tokom faze otkrivanja uređaja.

SAT>IP server treba svim klijentima da obezbedi ukupno četiri ikone:

- JPEG ikonu dimenzija 48x48 piksela
- JPEG ikonu dimenzija 120x120 piksela
- PNG ikonu dimenzija 48x48 piksela

- PNG ikonu dimenzija 120x120 piksela

Link ka ikonama mora biti relativan u odnosu na link ka xml datoteci tako što se umesto imena xml datoteke postavi jedna od četiri vrednosti:

- “icon/small.jpg”
- “icon/large.jpg”
- “icon/small.png”
- “icon/large.png”

Ukoliko SAT>IP server podržava HTTP korisničku spregu, link ka njoj se klijentima prosleđuje putem presentationURL polja.

3.1.4 Kontrola

Korišćenjem RTSP ili HTTP protokola SAT>IP klijentima je omogućena kontrola televizijskog sadržaja.

SAT>IP serveri moraju da implementiraju sve mehanizme navedene u specifikaciji, dok je klijentima ostavljen izbor u skladu sa njihovom specifikacijom i željenim načinom rada.

3.1.4.1 RTSP

SAT>IP klijenti koriste RTSP preko TCP protokola da uspostave RTSP sesiju sa serverom. RTSP sesija započinje RTSP SETUP porukom, a završava se RTSP TEARDOWN porukom. Svaka sesija je jednoznačno određena brojem sesije koju dodeljuje server. Prenos sadržaja počinje nakon slanja PLAY poruke, a tokom prenosa se sesija održava periodičnim slanjem OPTIONS poruka.

Klijent i serveri moraju da podržavaju RTSP 1.0 specifikaciju i sve poruke se šalju preko standardnog RTSP porta 554.

3.1.4.1.1 Formiranje sesije (RTSP SETUP)

Formiranje sesije inicira klijent slanjem RTSP SETUP poruke koja sadrži:

- RTSP URI niz karaktera koji sadrži parametre potrebne biraču kanala da se zaključa na zadatu frekvenciju
- Specifikaciju tipa transporta (multikast/unikast, port,...)

Ukoliko je sesija uspešno uspostavljena i birač kanala je zaključan na zadatu frekvenciju server vraća 200 OK odgovor na SETUP poruku koji sadrži:

- Nasumično generisani jedinstveni broj sesije
- Stvarne parametre tipa transporta
- Identifikator bitskog toka (streamID) koji klijenti koriste za identifikaciju konfigurisanog bitskog toka.

Format RTSP SETUP poruke je dat u tabeli 5 [9].

RTSP	SETUP	
RTSP_URI		rtsp://<ip_adresa>/?src=<srcID>&freq=<frekvencija>&... rtsp://<ip_adresa>/stream=<streamID>?src=<srcID>&... rtsp://<ip_adresa>/stream=<streamID>
Zahtev	Zaglavlje	CSeq <i>Session</i> ⁽¹⁾ Transport
	Telo	Nije dostupno
Odgovor	Zaglavlje	CSeq <i>Session</i> ⁽²⁾ Transport Com.ses.streamID
	Telo	Nije dostupno
	Status	400, 403, 404, 414, 454, 461, 500, 503, 505, 551
Napomene		¹ Ovo polje se koristi samo ukoliko sesija između klijenta i servera postoji ² Vreme trajanja sesije je specificirano "timeout" parametrom u Session polju odgovora. Podrazumevana vrednost je 60 sekundi

Tabela 5 – Format RTSP SETUP poruke

Polje CSeq se nalazi u zaglavlju zahteva i odgovora svake RTSP poruke. Ovaj broj se uvećava nakon svakog poslatog zahteva dok u okviru jednog para zahtev-odgovor ostaje isti.

Polje Session identifikuje svaku sesiju iniciranu od strane servera prilikom slanja RTSP odgovora na SETUP zahtev i važi sve dok se sesija ne okonča slanjem TREATDOWN poruke. Ovaj identifikator generiše server i njegova dužina je osam okteta. Kada klijent dobije odgovor na SETUP i samim tim broj sesije, svaka naredna komunikacija u okviru te sesije mora da sadrži u sebi ovaj identifikator.

Ukoliko polje Session pored identifikatora sesije sadrži i vrednost koja određuje vreme isteka sesije server je obavestio klijenta koliko će vremena čekati na narednu RTSP poruku pre nego prekine sesiju. Podrazumevana vrednost je 60 sekundi i ona se koristi ako server ne javi klijentu koliko je vreme isteka sesije. Vreme isteka sesije se koristi za detekciju nestanka

klijenata sa mreže. U zavisnosti od tipa transporta preporučena je sledeća konfiguracije vremena isteka sesije:

- RTSP Unikast – vreme isteka sesije ne bi trebalo da bude kraće od 30 sekundi, dok je preporučeno da bude 60 sekundi
- RTSP Multikast – preporučeno je da vreme trajanja bude podešeno na 0. Ova specijalna vrednost znači da sesija nikada neće isteći, osim ukoliko klijent ne pošalje TEARDOWN poruku. Ovaj režim rada je preporučljiv u postavkama servera gde se sve sesije kontrolišu preko terminala ili WEB korisničke sprege.

Polje com.ses.streamID jedinstveno identifikuje bitski tok. StreamID je 16-bitna numerička vrednost.

Tip transporta se određuje poljem Transport gde se specificiraju parametri poput adrese klijenta, port, multikast ttl. Neki primeri Transport polja su dati u tabeli 6 [9].

Polje

Transport RTP/AVP;unicast;client_port=<client RTP port>-<client RTCP port>
 RTP/AVP;multicast;destination=<IP multicast address>;port=<RTP port>-<RTCP port>;ttl=<ttl>

RTP/AVP;multicast

RTP/AVP;multicast;ttl=<ttl>

RTP/AVP;multicast;port=<RTP port>-<RTCP port>

RTP/AVP;multicast;port=<RTP port>-<RTCP port>;ttl=<ttl>

RTP/AVP;multicast;destination=<IP multicast address>

RTP/AVP;multicast;destination=<IP multicast address>;ttl=<ttl>

RTP/AVP;multicast;destination=<IP multicast address>;port=<RTP port>-<RTCP port>

RTP/AVP;multicast;destination=<IP multicast address>;port=<RTP port>-<RTCP port>;ttl=<ttl>

Tabela 6 – Primeri formata Transport polja RTSP SETUP poruke

Opis konfiguracionih parametara Transport polja datih u tabeli 6 je:

- RTP/AVP – predstavlja format bitskog toka. SAT>IP standard isključivo podržava RTP/AVP
- unicast ili multikast – su međusobno isključivi i određuju tip transporta
- destination – koristi se isključivo za multikast tip prenosa i određuje adresu multikast grupa na koju će biti poslat bitski tok

- client_port – predstavlja par RTP/RTCP portova koji se koriste isključivo u slučaju unicast tipa prenosa
- port - predstavlja par RTP/RTCP portova koji se koriste isključivo u slučaju multikast tipa prenosa
- ttl – specifičan isključivo za multikast i predstavlja period egzistencije paketa na mreži koje se umanjuje prilikom svakog prolaska kroz ruter i kada ova vrednost postane nula paket se uklanja sa mreže

3.1.4.1.2 Pokretanje slanja bitskog toka (RTSP PLAY)

Nakon uspostavljanja sesije i uspešnog zaključavanja birača kanala na serveru, klijent je u mogućnosti da pokrene slanje bitskog toka koji sadrži televizijski sadržaj u TS formatu. U zavisnosti od tipa transporta prenos se započinje:

- U slučaju unicast tipa transporta slanjem RTSP PLAY poruke koja sadrži identifikator bitskog toka (streamID) i opciono RTSP URI niz karaktera koji sadrži parametre potrebne biraču kanala da se zaključa na zadatu frekvenciju i listu PID-ova kojima se na serveru filtrira željeni servis
- U slučaju multikast tipa transporta slanjem:
 - RTSP PLAY poruke formata istog kao u kod unicast-a,
 - ili slanjem IGMPv3 poruke multikast grupi

RTSP PLAY poruka, čiji je format dat u tabeli 7 [9], ne može biti poslata bez prethodno uspostavljene sesije. Zbog mogućnosti slanja RTSP_URI niza karaktera može se koristiti i za menjanje parametara bitskog toka, tj. za menjanje kanala.

RTSP		PLAY
RTSP_URI		rtsp://<ip_address>/stream=<streamID> rtsp://<ip_address>/stream=<streamID>?src=<srcID>&...
Zahtev	Zaglavlje	CSeq Session
	Telo	Nije dostupno
Odgovor	Zaglavlje	CSeq Session RTP-Info
	Telo	Nije dostupno
	Status	400, 403, 404, 408, 453, 454, 500, 505, 551

Tabela 7 – Format RTSP PLAY poruke

3.1.4.1.3 Održavanje sesije na mreži (RTSP OPTIONS)

Nakon što se uspostavi RTSP sesija klijent mora da je održava prisutnom na mreži periodičnim slanjem RTSP OPTIONS poruka. Perioda slanja je određena prilikom uspostavljanja sesije u odgovoru koji server šalje na RTSP SETUP poruku. Nakon svake primljene poruke server zna da je klijent prisutan na mreži u time održava sesiju. Ukoliko se desi da OPTIONS poruka ne stigne nakon isteka vremena definisanog prilikom uspostavljanja sesije ista će biti prekinuta na isti način kao i kod slanja RTSP TEARDOWN poruke.

Pored održavanja sesije OPTIONS poruka može biti poslata u bilo koje vreme sa ciljem da klijent sazna koje RTSP metode server podržava.

Format RTSP OPTIONS poruke je dat u tabeli 8 [\[9\]](#).

RTSP		OPTIONS
RTSP_URI		rtsp://<ip_address>/
Zahtev	Zaglavlje	CSeq Session ⁽¹⁾
	Telo	Nije dostupno
Odgovor	Zaglavlje	CSeq Session ⁽¹⁾ Public
	Telo	Nije dostupno
	Status	400, 403, 454, 500, 505, 551
Napomene		Ova metoda može biti poslata u bilo koje vreme ¹ Ovo polje je potrebno samo ukoliko postoji sesija između klijenta koji šalje poruku i servera koji je prima

Tabela 8 – Format RTSP OPTIONS poruke

3.1.4.1.4 Operacije nad bitskim tokom

Bitski tok može biti promenjen samo od strane klijenta koji je uspostavio sesiju sa serverom. Novi parametri bitskog toka se šalju putem RTSP_URI niza karaktera.

Klijent se, pored formiranja sesije i iniciranja bitskog toka, može priključiti na bitski tok formiran od strane nekog drugog klijenta na sledeći način:

- formiranjem sesije koja će umesto parametara poslatih preko RTSP_URI polja imati samo sessionID bitskog toka koji postoji u okviru neke druge sesije ili
- pridruživanjem multikast grupi slanjem IGMPv3 poruke

Klijent koji se pridružio multikast grupi slanjem IGMPv3 poruke neće dobiti vlasništvo nad bitskim tokom i neće biti u mogućnosti da modifikuje njegove parameter niti da ga zaustavi.

3.1.4.1.5 Dobijanje liste bitskih tokova (RTSP DESCRIBE)

SAT>IP standard klijentima daje mogućnost da šalju upite serverima kako bi dobili informacije o trenutno uspostavljenim bitskim tokovima. Da bi došao do ove informacije, klijent treba ka serveru da pošalje RTSP DESCRIBE poruku koja može biti poslata u toku sesije koju je klijent uspostavio sa serverom ili u trenutku kada između servera i klijenta ne postoji sesija.

Kao odgovor na poslatu DESCRIBE poruku, ukoliko na serveru postoji bar jedan konfigurisani bitski tok, server šalje opis u okviru SDP poruke (eng. Session Description Protocol). Ukoliko na serveru ne postoji otvorena sesija ni uspostavljen bitski tok server će ka klijentu poslati poruku greške 404.

Format RTSP DESCRIBE zahteva i odgovora dat je u tabeli 9 [\[9\]](#).

RTSP		DESCRIBE
RTSP_URI		rtsp://<ip_address>/ rtsp://<ip_address>/stream=<streamID>
Zahtev	Zaglavlje	CSeq Session ⁽¹⁾ Accept
	Telo	Nije dostupno
Odgovor	Zaglavlje	CSeq Session ⁽¹⁾ Content-Type: application/sdp Content-Base Content-Length
	Telo	application/sdp
	Status	400, 404, 406, 500, 505, 551
Napomene		Ova metoda može biti poslata u bilo koje vreme ¹ Ovo polje je potrebno samo ukoliko postoji sesija između klijenta koji šalje poruku i servera koji je prima

Tabela 9 – Format RTSP DESCRIBE poruke

Polje Accept određuje tip sadržaja koji je prihvatljiv u okviru odgovora. U okviru SAT>IP standarda ovo polje mora biti postavljeno na application/sdp. Na istu vrednost mora biti postavljeno i polje Content-Type.

Content-Base polje sadrži link ka serveru, dok Content-Length nosi informaciju o dužini sadržaja metode.

Primer SDP listinga dat je u tabeli 10 [9].

Zahtev	<pre>DESCRIBE rtsp://192.168.128.5/ CSeq: 5 Accept: application/sdp <CRLF></pre>
Odgovor	<pre>RTSP/1.0 200 OK CSeq: 5 Content-Type: application/sdp Content-Base: rtsp://192.168.128.5/ Content-Length: 724 <CRLF> v=0 o=- 5678901234 7890123456 IN IP4 192.168.128.5 s=SatIPServer:1 4 t=0 0 m=video 5004 RTP/AVP 33 c=IN IP4 239.0.0.8/5 a=control:stream=0 a=fmtp:33 ver=1.0;src=1;tuner=1,240,1,7,12402,v,dvbs,,,,27500,34;pids=0,16,56,112,168,1709 a=inactive m=video 5006 RTP/AVP 33 c=IN IP4 239.0.0.9/5 a=control:stream=1 a=fmtp:33 ver=1.0;src=1;tuner=1,240,1,7,12402,v,dvbs,,,,27500,34;pids=0,16,50,104,166,1707 a=sendonly m=video 0 RTP/AVP 33 c=IN IP4 0.0.0.0 a=control:stream=2 a=fmtp:33 ver=1.0;src=1;tuner=1,240,1,7,12402,v,dvbs,,,,27500,34;pids=all a=sendonly m=video 5010 RTP/AVP 33 c=IN IP4 239.0.0.11/5 a=control:stream=3 a=fmtp:33 ver=1.0;src=2;tuner=2,221,1,6,11758,h,dvbs2,8psk,off,25,27500,56;pids=all a=sendonly</pre>

Tabela 10 – Primer SDP listinga u okviru odgovora na DESCRIBE zahtev

Opis polja SDP odgovora je:

- “v=” – daje verziju SDP protokola (mora biti 0 za SAT>IP)
- “o=” – opisuje vlasnika sesije: “-” <identifikator sesije> <verzija sesije> IN (za internet) i IP adresa servera.
- “m=” – označava početak sekcije kola opisuje bitski tok i specificira tip sadržaja, port (za unicast ime vrednost 0), protokol: RTP/AVP i format opisa koji za SAT>IP ima vrednost 33.
- “c=” – opisuje vezu gde je za unicast vrednost 0.0.0.0, a za multikast predstavlja IP multikast adresu koju prati ttl vrednost.
- “c=” – daje trenutno stanje bitskog toka na sledeći način
 - inactive za sesije kod kojih bitski tok nije pokrenut
 - sendonly za sesije gde je bitski tok pokrenut

Pored standardnih polja SDP listing je proširen poljima koja su specifična za SAT>IP protokol:

- s=SatIPServer:1 <Broj birača kanala>
- a=control:stream=<identifikator bitskog toka>
- a=fmtp:33
 ver=<major>.<minor>;src=<srcID>;tuner=<feID>,<level>,<lock>,<quality>,<frequency>,<polarisation>,<system>,<type>,<pilots>,<roll_off>,<symbol_rate>,<fec_inner>;pids=<pid0>,...,<pidn> gde:
 - major i minor opisuju verziju a samim tim i sintaksu fmtp niza karaktera
 - level – jačina signala na biraču kanala u opsegu [0, 255]
 - lock – koji daje informaciju o tome da li je birač kanala zaključan (1) ili ne (0) na zadatu frekvenciju
 - quality – daje vrednost bitske greške (Eng. Bit Error Rate - BER) na biraču kanala

3.1.4.1.6 Zatvaranje sesije (RTSP TEARDOWN)

Zatvaranje sesije klijent obavlja slanjem RTSP TEARDOWN poruke server. Prilikom zatvaranja sesije stopira se bitski tok bez obzira na tip prenosa. U slučaju unicast tipa transporta klijent, bio vlasnik sesije ili ne, može da prekine sesiju, dok je u slučaju multikast tipa transporta samo vlasnicima sesije dozvoljeno da prekidaju sesiju.

Format RTSP TEARDOWN poruke je dat u tabeli 11 [\[9\]](#).

RTSP		TEARDOWN
RTSP_URI		rtsp://<ip_address>/stream=<streamID>
Zahtev	Zaglavlje	CSeq Session
	Telo	Nije dostupno
Odgovor	Zaglavlje	CSeq Session
	Telo	Nije dostupno
	Status	400, 403, 404, 408, 454, 500, 505, 551

Tabela 11 – Format RTSP TEARDOWN poruke

3.1.4.1.7 Format RTSP_URI niza karaktera

Svaki RTSP zahtev se sastoji od imena metode koja je praćena URI nizom karaktera. URI se dalje sastoji od IP adrese i od upita koji se šalje serveru kao (slike 5).

PLAY rtsp://<ip_adresa>:554/?src=1&freq=10744&pol=h&...&sr=22000&fec=56&pids=0,200,201,202,203

Metoda URI Upit

Slika 5 - Format RTSP poruke

Upit specificira parametre za zaključavanje birača kanala i za filtriranje željenih PID-ova što omogućava klijentu da od servera dobije samo jedan, željeni servis. Format upita dat je u tabeli 12 [9].

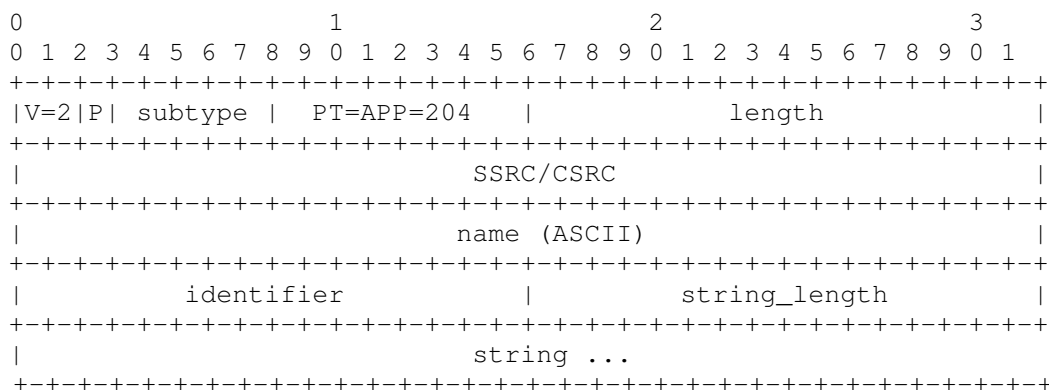
Atribut	Ime	Vrednost
fe	Redni broj birača kanala	[1, 65535]
src	DiSEqC pozicija	[1, 255]
pol	polarizacija	h - horizontalna linearna; v – vertikalna linearna; l – kružna ulevo; r – kružna udesno
ro	Roll_off	Koristi se samo za DVBS2 i može da ima jednu od sledeće tri vrednosti: 0.35, 0.25, 0.20
msys	Tip sistema	“dvbs” ili “dvbs2”
mtype	Tip modulacije	Koristi se samo za DVBS2 i može da ima jednu od sledećih vrednosti: "qpsk", "8psk".
plts	Pilot tonovi	Koristi se samo za DVBS2 i može da ima jednu od sledećih vrednosti: "on", "off".

sr	Broj simbola u jedinici vremena	Broj u kSymb/s.
fec	Fec_inner	"12", "23", "34", "56", "78", "89", "35", "45", "910".
pids	Lista pidova za jedan servis	CSV lista čiji elementi imaju vrednosti u opsegu [0, 8191]. Npr. pids=0,16,201,302

Tabela 12 – Format RTSP_URI niza karaktera

3.1.4.2 RTCP

Svaki RTP bitski tok (unikast ili multikast) je praćen RTCP bitskim tokom koji u sebi nosi informacije koje opisuju RTP bitski tok. RTCP je baziran na standardu RFC 3550 i klijentima se dostavlja na istu IP adresu kao i RTP bitski tok i na port koji je jednak RTP portu uvećanim za jedan. RTCP bitski tok nosi informacije o kvalitetu signala i parametrima koji su korišćeni prilikom zaključavanja birača kanala. Informacije o bitskom toku se nose u okviru RTCP APP paketa čiji je format dat na slici 5.



Slika 6 – Format RTCP APP paketa

Polja APP paketa su definisana u tabeli 13 [9]. Server je u obavezi da pošalje pet RTCP paketa u sekundi.

Polje	Vrednost
name	Uvek postavljeno na "SES1"
identifier	16-bitna verzija postavljena uvek na 0000
string_length	16-bitna vrednost koja predstavlja veličinu polja string na primer 0048
string	Format identičan formatu "a=fmtp:33" polja definisanog u okviru SDP-a. Na primer: ver=1.0;src=1;tuner=1,240,1,7,12402,v,dvbs,,,,,27500,34;pids=0,16,56,112,168,1709

Tabela 13 – Opis polja RTCP APP paketa

3.1.4.3 HTTP

SAT>IP standard klijentima dozvoljava kontrolu servera korišćenjem HTTP 1.1 protokola. Prenos televizijskog sadržaja se započinje slanjem HTTP GET metode od klijenta ka serveru koja mora biti u sledećem formatu:

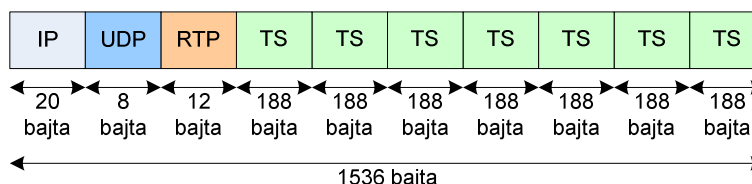
HTTP_URI = "http://" ip_adresa [":"port] "/"? upit

Format upita je identičan formatu RTSP_URI niza karaktera.

Prekidanje slanja bitskog toka se obavlja prekidanjem TCP veze od strane klijenta.

3.1.5 Slanje bitskog toka

Prenos televizijskog sadržaja se u okviru SAV>IP standarda bazira na standardu RFC 2250. U okviru jednog IP paketa se, kao što je prikazano na slici 7, se prenosi maksimalno sedam TS paketa. RTP enkapsulacija je obavezna



Slika 7 – Struktura jednog datagrama

Ukoliko se birač kanala nije uspešno zaključao na zadatu frekvenciju umesto TS paketa koji sadrže željeni servis, server je u obavezi da na svakih 100ms pošalje jedan prazan RTP paket.

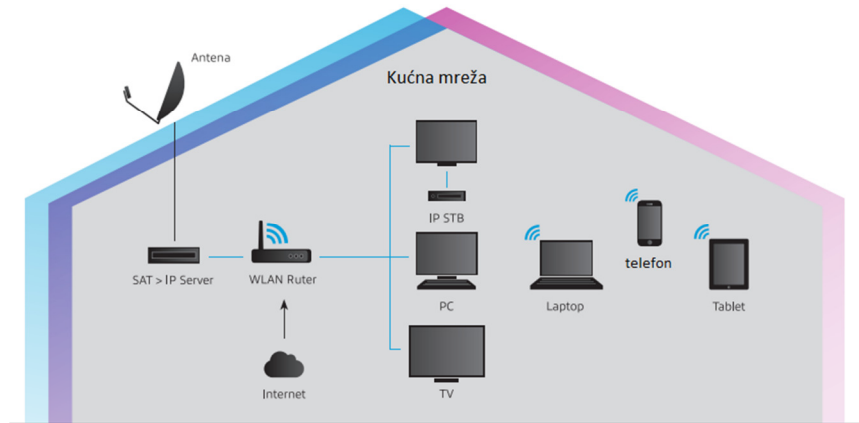
U slučaju da je klijent odabrao HTTP protokol za prenos, HTTP zaglavlje mora kao MIME tip da ima "video/MP2T".

3.2 Upotreba SAT>IP protokola

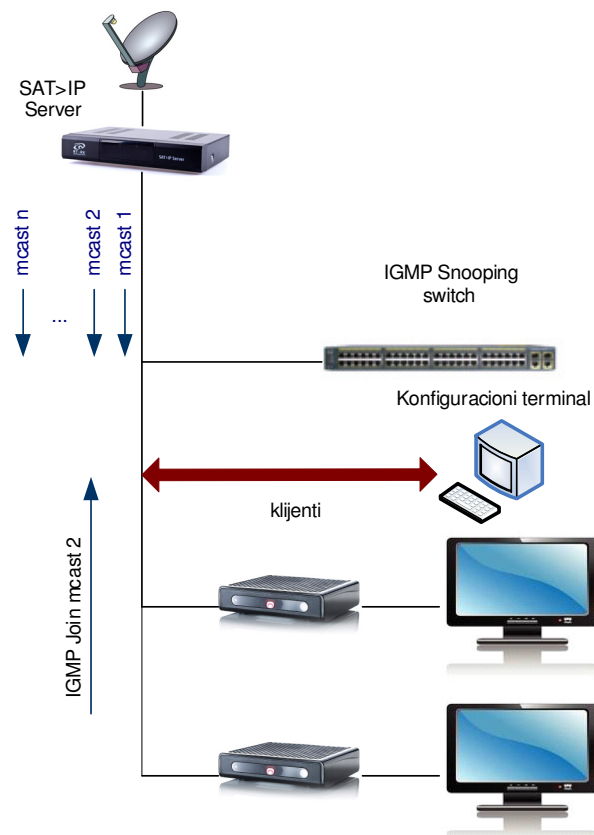
Osnovna ideja SAT>IP protokola je da klijentima omogući prenos televizijskog sadržaja putem IP mreža, što u znatnoj meri olakšava instalaciju satelitske opreme i omogućava prijem i prikaz televizijskog sadržaja na uređajima koji na sebi nemaju birač kanala (tableti, pametni telefoni, personalni računari,...). Uzimajući u obzir tip transporta upotreba SAT>IP standarda se može podeliti u tri grupe:

- Server koji je baziran na dinamičkoj konfiguraciji (RTSP Unikast)
- Server koji je baziran na statičkoj konfiguraciji (RTSP multikast)
- Server koji je baziran na mešovitoj konfiguraciji

Dinamička konfiguracija je podrževani režim rada gde klijenti konfiguriraju bitski tok i definišu parametar potrebne za zaključavanje birača kanala. Svaki put kada klijent konfigurira birač kanala na serveru, server formira sesiju u okviru koje klijent ima mogućnost promene njenih parametara. Tipičan primer upotrebe dinamičke konfiguracije, prikazan na slici 8, server je u okviru kućne mreže.



Slika 8 – Primer upotrebe dinamičke konfiguracije



Slika 9 – Primer statičke konfiguracije

Konfigurisanje servera da radi u statičkom režimu onemogućava klijente da modifikuju parametar sesije, već mogu samo da se pridruže jednoj od njih. Server se konfigurira preko

korisničke sprege koja najčešće predstavlja WEB server koji se nalazi na SAT>IP serveru. Korisnička sprega predstavlja virtuelnog vlasnika svih sesija koje postoje na serveru. Statički konfigurisani serveri, čiji je jedan primer prikazan na slici 9, se najčešće primenjuju u slučaju mreža koje podrazumevaju desetine ili stotine ili više klijenata kao što su na primer bolnice, hoteli, javne ustanove, aerodromi, stambene zgrade,...

3.3 Opis realizacije

3.3.1 Opis realizacije SAT>IP klijenta

Uzimajući u obzir da po SAT>IP specifikaciji klijent nema birač kanala i da su mu za prijem i prikaz televizijskog sadržaja potrebni jedino pristup IP mrežama i mogućnost dekodovanje i prikazivanja televizijskog sadržaja, arhitektura SAT>IP klijenta je osmišljena tako da olakša i ubrza integraciju u već postojeća rešenja. Sam SAT>IP standard se ne bavi načinom na koji se sadržaj u okviru klijenta obrađuje već je fokus stavljen na upravljanje resursima na serveru i dobavljanje bitskog toka. Stoga je SAT>IP klijent realizovan kao biblioteka čija je aplikativna sprega osmišljena tako da bude što približnija aplikativnoj sprezi koju imaju birači kanala i filteri bitskog toka. Na ovaj način, kao što je prikazano na slici 10, integracija SAT>IP biblioteke u već postojeće uređaje je u velikoj meri olakšana. Ovakvim pristupom integraciji se minimizuju promene u postojećim aplikacijama, tako što srednji sloj koji upravlja resursima SAT>IP biblioteku vidi kao još jedan birač kanala i nad njim obavlja operacije na isti način kao sa fizički prisutnim biračem kanala.

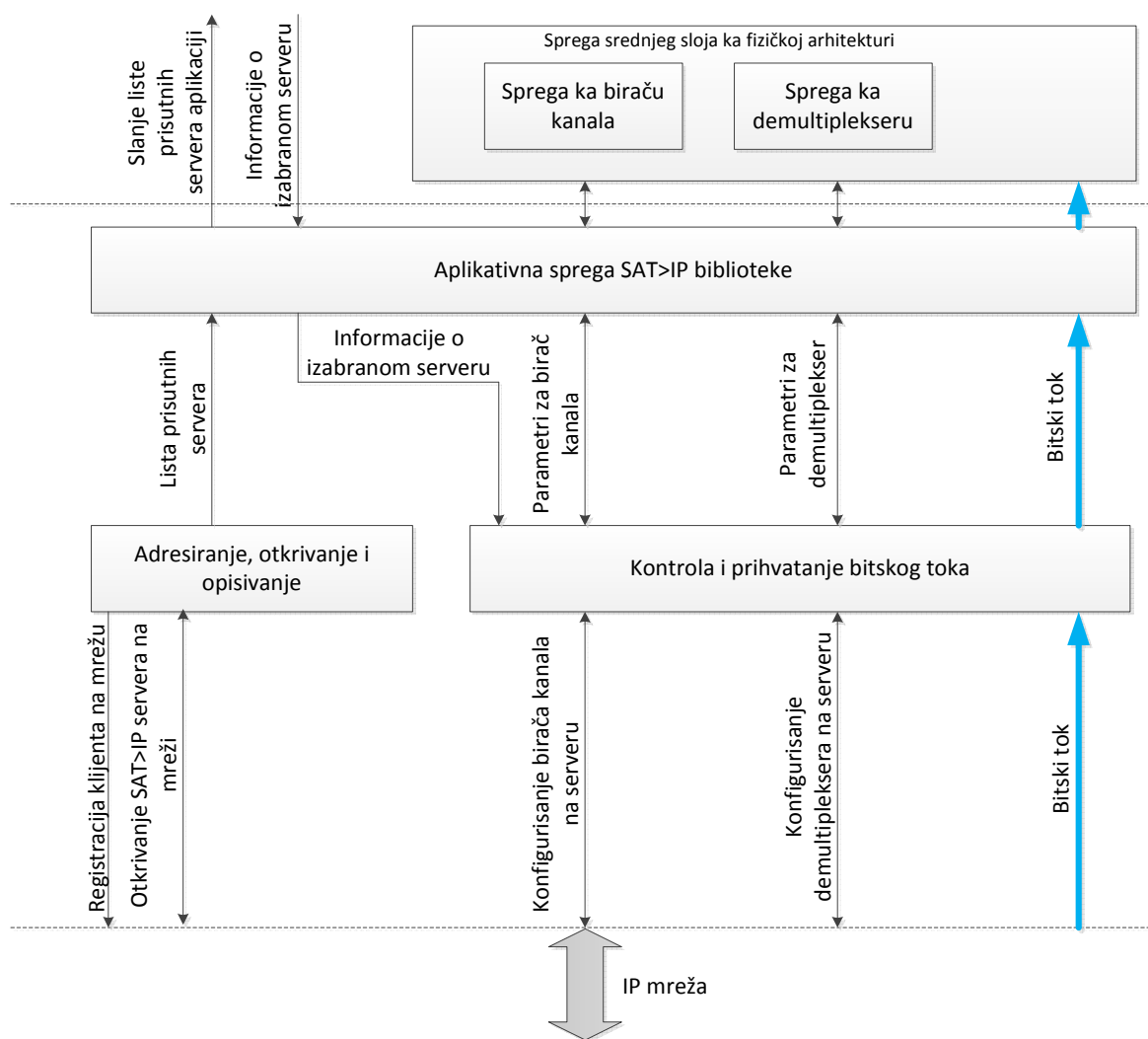


Slika 10 – Primer integracije SAT>IP klijentske biblioteke u sistem

Arhitektura SAT>IP biblioteke je data na slici 11.

Blok za adresiranje, otkrivanje i opisivanje je baziran na biblioteci otvorenog koda libupnp koja zadovoljava sve zahteve SAT>IP standarda i čija je funkcionalnost proširena zahtevima definisanim u okviru SAT>IP standarda.

Blok za kontrolu i prihvatanje bitskog toka je baziran na biblioteci otvorenog koda live555 gde su RTSP, RTCP, HTTP i RTP protokoli prilagodjeni zahtevima postavljenim u okviru SAT>IP speifikacije.



Slika 11 – Arhitektura SAT>IP klijentske biblioteke

Aplikativna sprega predstavlja skup funkcija koje su dostupne prilikom integracije. Podeljene su u tri logički odvojene grupe:

- Funkcije za rukovanje sesijom koje se nalaze u datoteci `sat2ip_session_api.h`

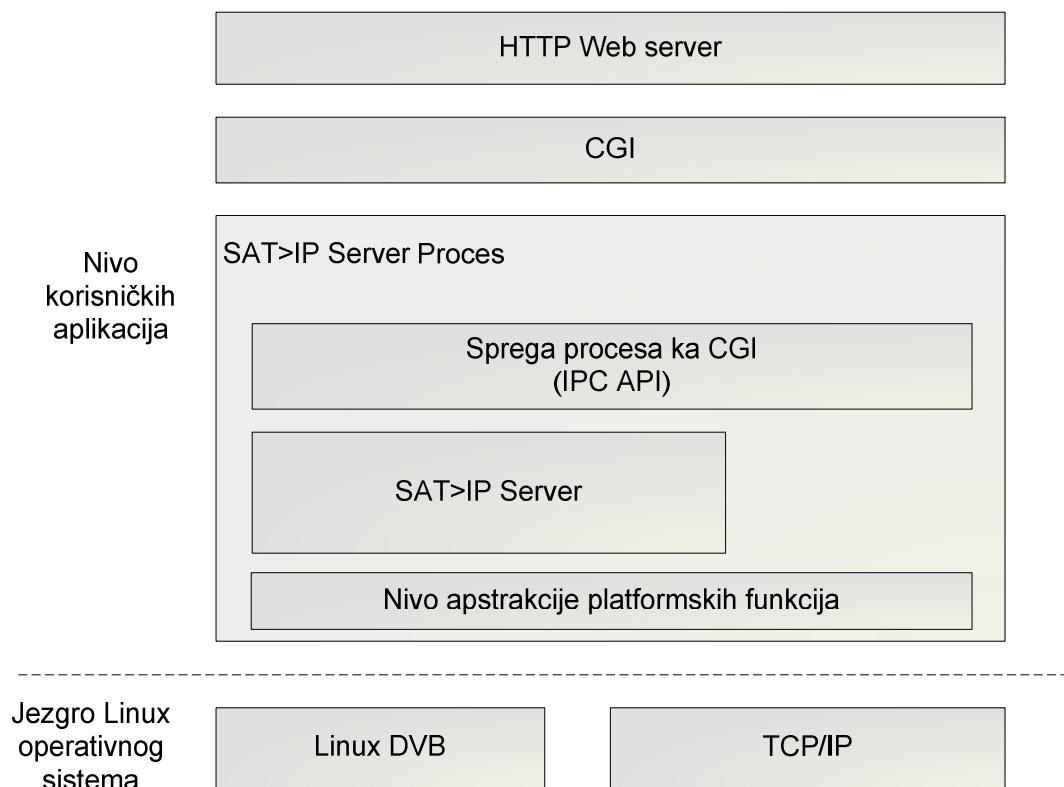
- Funkcije za slanje RTSP i HTTP komandi koje se nalaze u okviru datoteke sat2ip_command.h
- Funkcije kod kojih implementacija može da varira od platforme do platforme i koje su zarad portabilnosti postavljene u datoteku sat2ip_system.h sa ciljem da se olakša proces integracije.

3.3.2 Opis realizacije SAT>IP server

SAT>IP server, po svojoj specifikaciji, nije namenjen da obrađuje i modifikuje televizijski program. Njegova uloga je isključivo da traženi sadržaj prosledi klijentima. Ukoliko je televizijski sadržaj zaštićen nekim od algoritama za zaštitu digitalnog sadržaja, podrazumeva se da su klijenti sposobni da otključaju takav bitski tok, tako da i u ovom slučaju server ne obrađuje televizijski sadržaj.

3.3.2.1 Arhitektura SAT>IP servera

SAT>IP server je implementiran na platformi sa ograničenim resursima koja je namenski dizajnirana za uređaje koji služe za slanje podataka u mrežu u realnom vremenu na kojoj se nalazi operativni sistem Linux. Arhitektura SAT>IP servera je data na slici 12.



Slika 12 – Arhitektura SAT>IP servera.

SAT>IP server je organizovan u četiri nivoa:

- HTTP Web server koji predstavlja korisničku spregu servera u vidu Web stranice koja se nalazi na samom uređaju
- CGI koji predstavlja program koji obrađuje prenos informacija između HTTP Web servera i SAT>IP server procesa.
- SAT>IP server proces koji predstavlja je podeljen na tri dela
 - Spregu procesa ka CGI programu
 - SAT>IP server koji obuhvata sve funkcionalnosti definisane SAT>IP specifikacijom
 - Nivo apstrakcije platformskih funkcija, koji predstavlja spregu ka ciljnim platformama koje mogu biti različite strukture i ima cilj da olakša i ubrza prelazak sa jedne na drugu platformu.
- Jezgro Linux operativnog sistema koje uključuje sve drajvere potrebne za rad servera (DVB Linux, TCP/IP...).

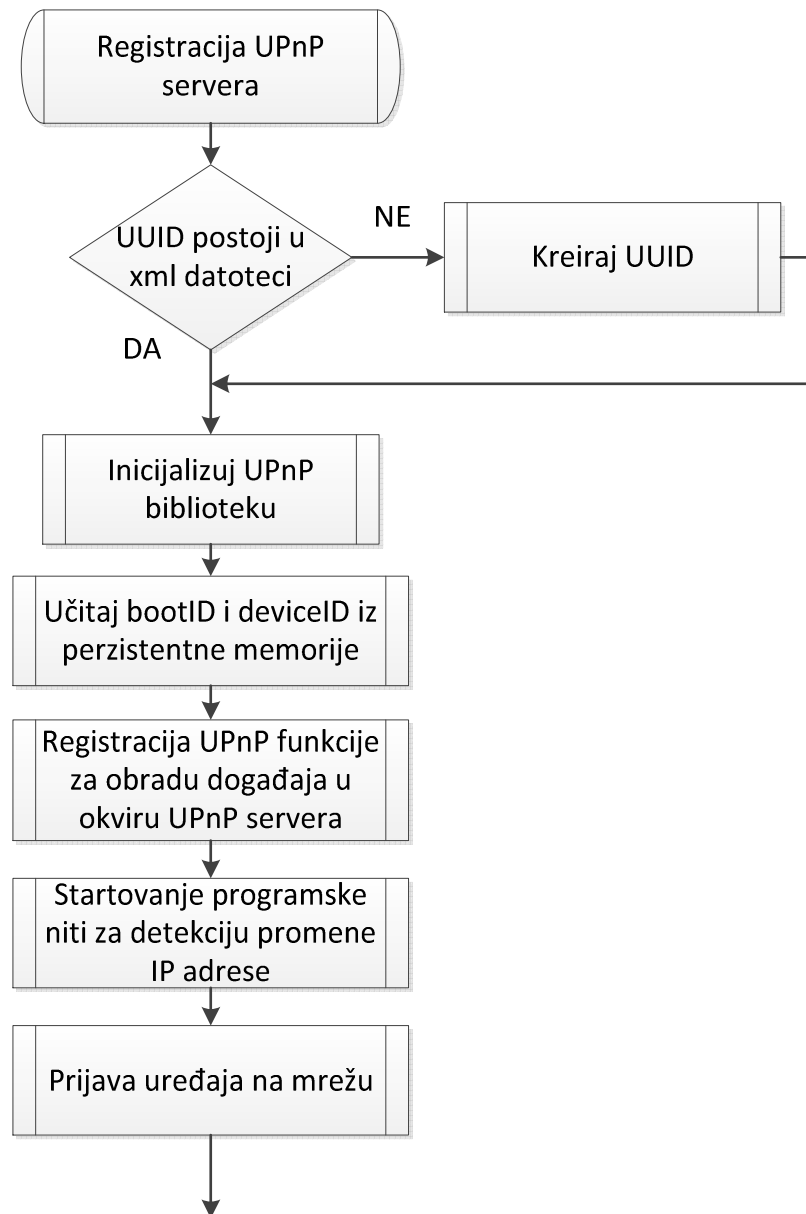
SAT>IP server je sačinjen od programskih niti, čiji je opis dat u tabeli 14.

Programska nit	Životni vek	Broj niti	Opis
main	Uvek prisutna	1	Programska nit zadužena da prihvata klijentske veze
Nivo apstrakcije platformskih funkcija	Uvek prisutne	2	Predstavljaju spregu između servera i drajvera na ciljnoj platformi
UPnP	Uvek prisutne	6	Programske niti koje koristi biblioteka otvorenog koda libupnp
CGI_API	Uvek prisutna	1	Programska nit koja obrađuje zahteve primljene od strane HTTP Web servera
LED_API	Uvek prisutna	1	Programska nit koja upravlja radom LED dioda na prednjoj strani uređaja
Za slanje bitskog toka	Prisutna	1 po otvorenoj sesiji	Programska nit koja šalje bitski tok ka klijentima (RTP ili TCP)

Tabela 14 – Opis programskih niti SAT>IP servera

3.3.2.2 Adresiranje, otkrivanje i opisivanje

Funkcionalnosti vezane za adresiranje, otkrivanje i opisivanje su bazirane na standardnoj biblioteci otvorenog koda libupnp-1.6.16, koja je prilagođena zahtevima postavljenim u okviru SAT>IP specifikacije. Blok dijagram koji prikazuje inicijalizaciju UPnP biblioteke je dat na slici 13.



Slika 13 – Inicijalizacija UPnP biblioteke

UUID se generiše pozivom funkcije *format_uuid_v1()* koja se nalazi u okviru biblioteke libupnp.

BootID i deviceID se čuvaju u datoteci *data-upnp_bootid.dat* koja se nalazi u perzistentnoj memoriji. Svaka promena navedene dve promenljive se upisuje u navedenu datoteku.

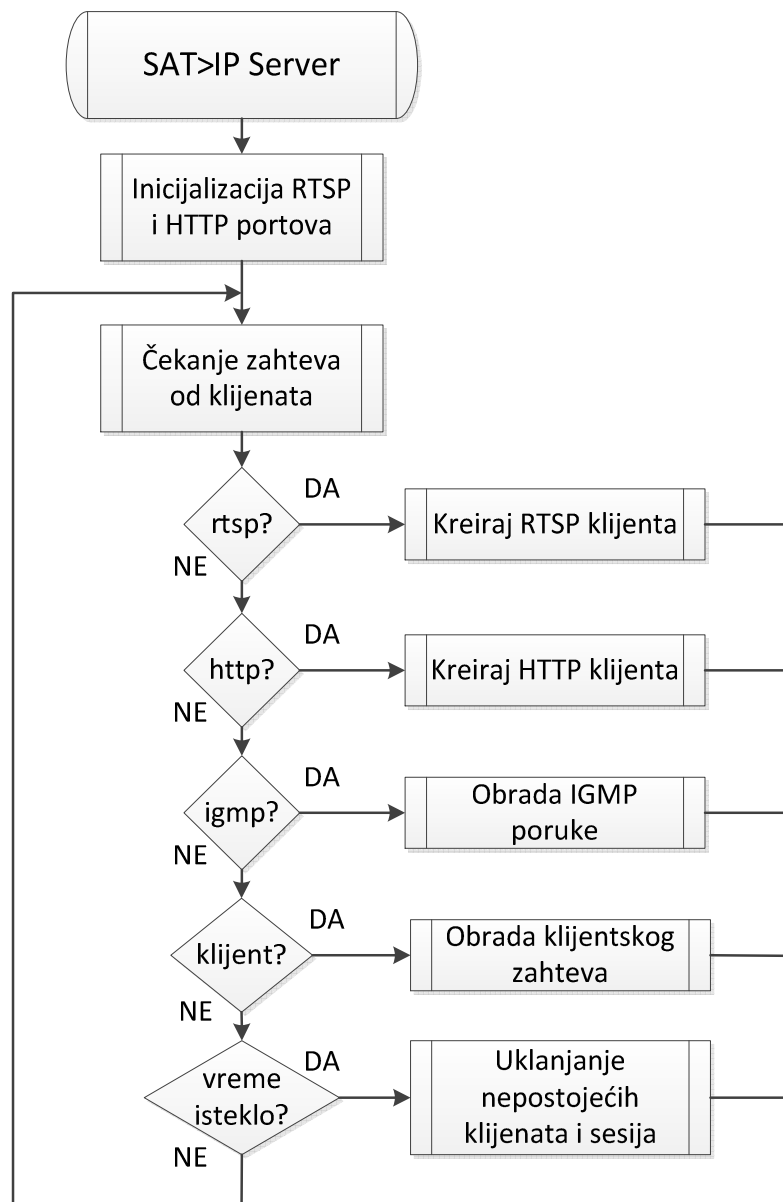
Registracijom funkcije za obradu događaja se obezbeđuje da server odreaguje na svaku poruku primljenu od drugih uređaja i da ukoliko je potrebno na događaj odgovori slanjem odgovarajuće M-SEARCH poruke.

Programska nit za detekciju promene adrese periodično proverava trenutnu IP adresu i na detekciju promene šalje NOTIFY byebye poruku na mrežu sa koje se odjavljuje i NOTIFY alive poruku na mrežu na koji se prijavljuje.

Blok za prijavu uređaja na mrežu šalje tri NOTIFY poruke na način opisan u poglavlju [3.1.2](#) ovog dokumenta.

3.3.2.3 Glavna programska nit SAT>IP servera

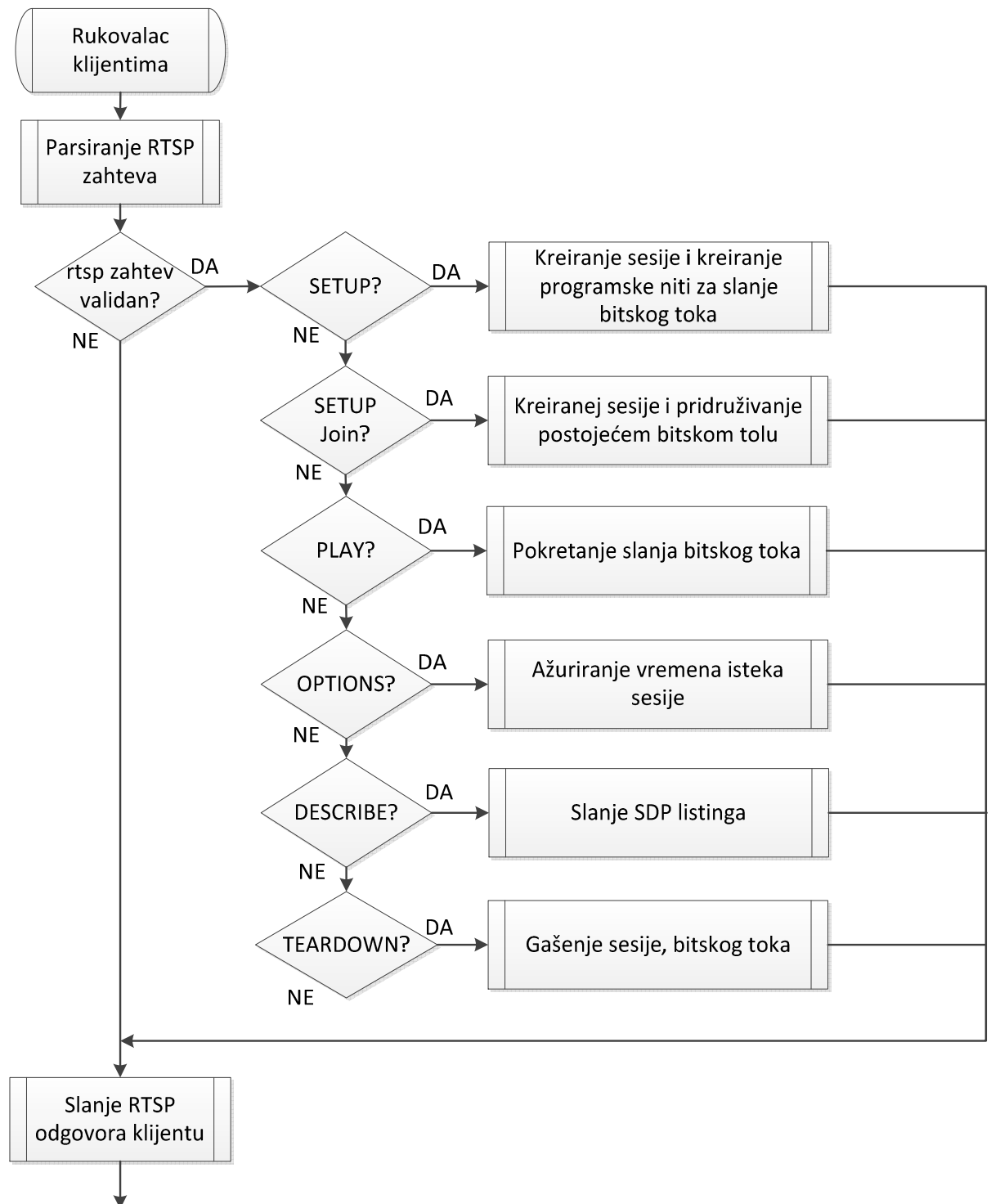
U okviru glavne programske niti servera, čiji je blok dijagram dat na slici 14, inicijalizuju se portovi i server osluškuje poruke od klijenata. Ukoliko zahtev pristigne na HTTP ili RTSP portove, server stvara nove klijente i dodaje ih u listu klijenata čije poruke obrađuje. Poruke pristigle od IGMP grupe ili od klijenata koji se nalaze u listi se obrađuju u skladu sa primljenim zahtevima u okviru tih poruka. Ukoliko ne postoje poruke server svake sekunde proverava stanje postojećih klijenata i uklanja one za koje detektuje da nisu prisutni.



Slika 14 – Glavna programska nit servera

3.3.2.4 Obrada RTSP zahteva

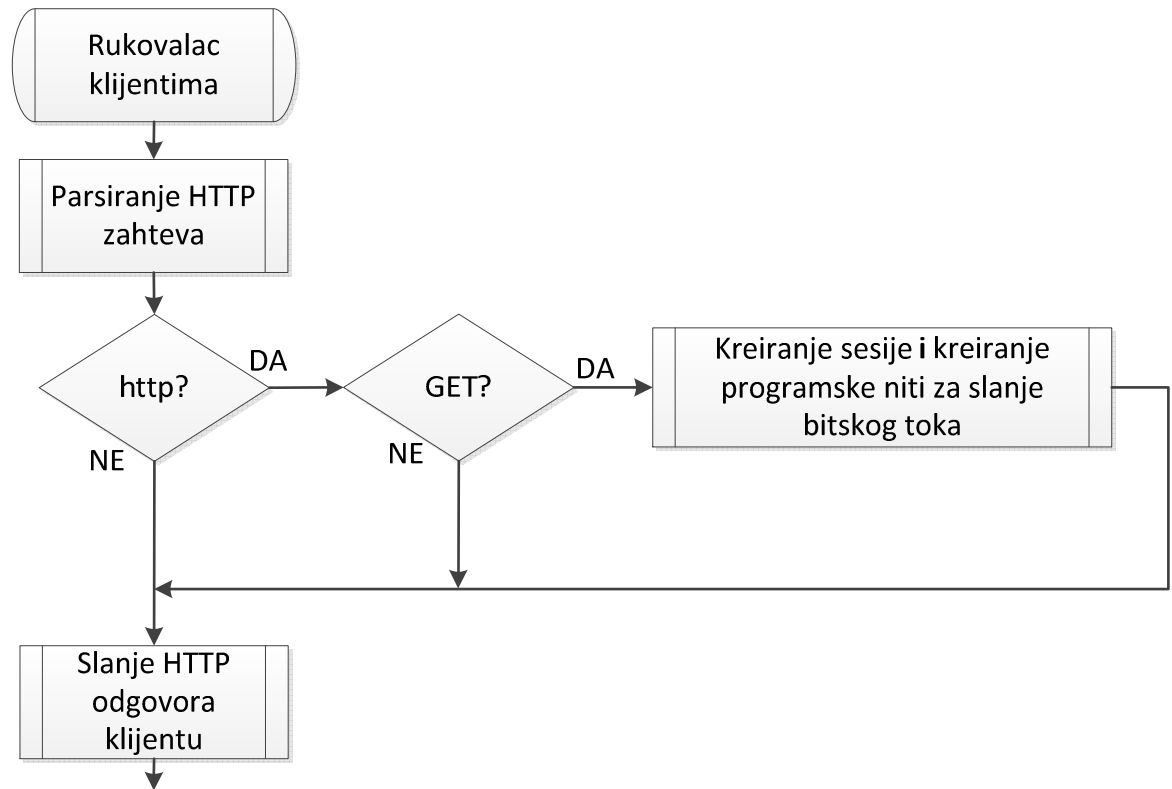
Blok dijagram obrade RTSP zahteva na način koji je opisan u poglavlju [3.1.4.1](#) ovog dokumenta je prikazana na slici 15.



Slika 15 – Obrada RTSP zahteva

3.3.2.5 Obrada HTTP zahteva

Blok dijagram obrade HTTP zahteva na način koji je opisan u poglavlju [3.1.4.3](#) ovog dokumenta je prikazana na slici 16.

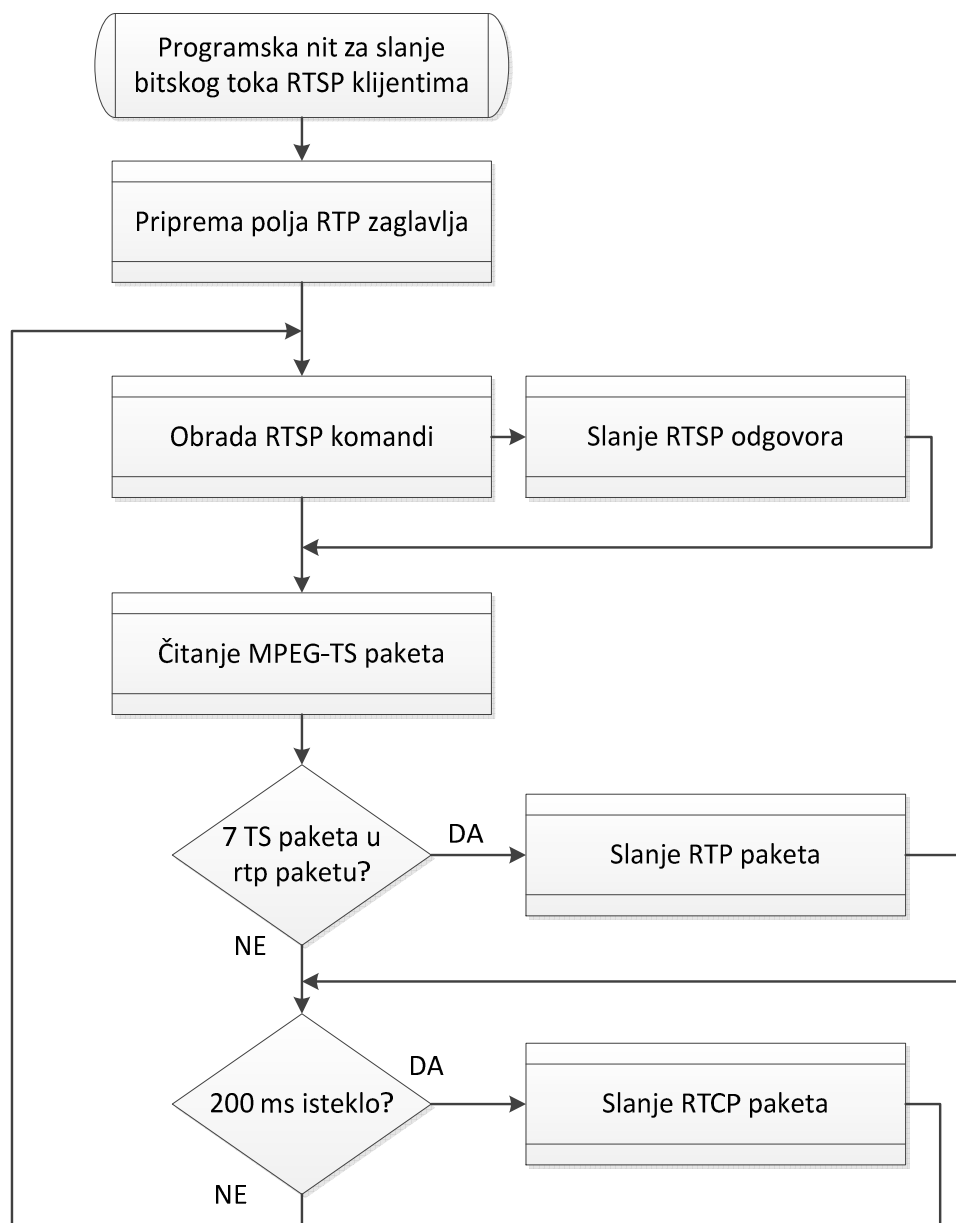


Slika 16 – Obrada HTTP zahteva

3.3.2.6 Programska nit za slanje bitskog toka RTP protokolom

Programska nit zadužena za slanje bitskog toka RTSP klijentima, prikazana na slici 17, obavlja sledeće operacije:

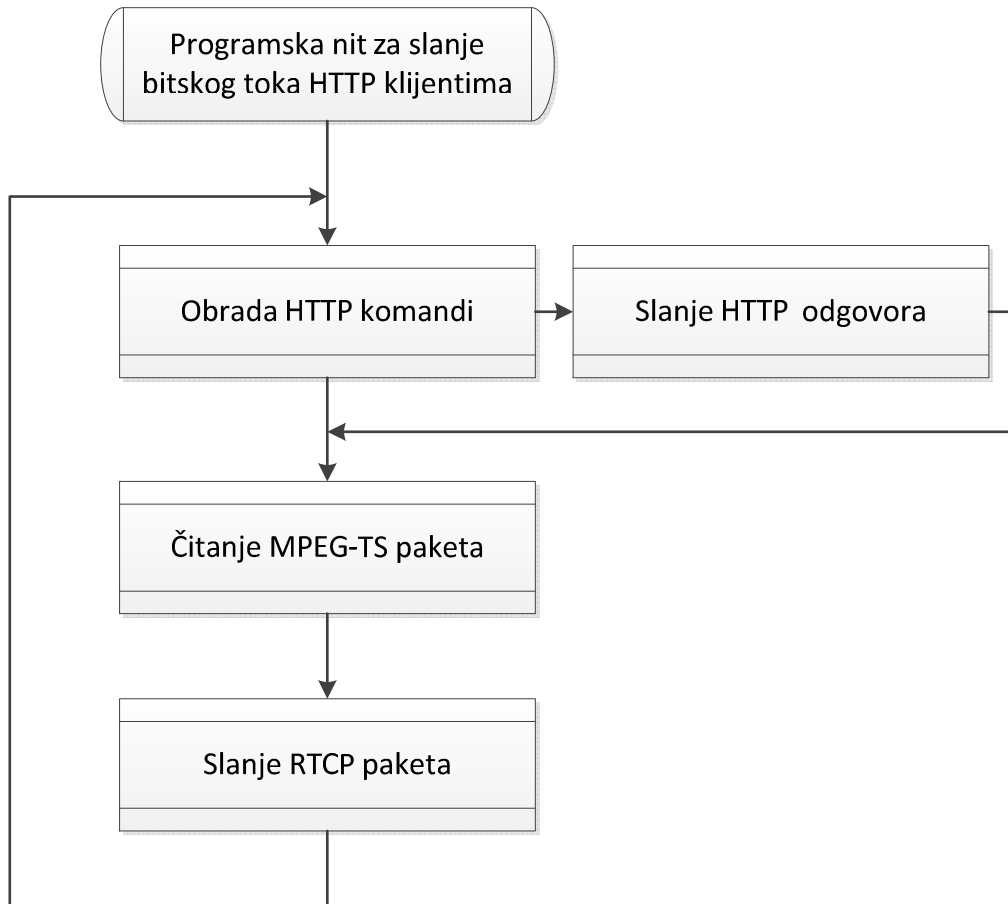
- formira RTP zaglavlje,
- dodaje u RTP paket 7 TS-MPEG paketa
- šalje RTP paket klijentu na RTP port
- svakih 200 ms formira i šalje klijentu RTSP paket na RTCP port



Slika 17 – Programska nit za slanje bitskog toka RTSP klijentima

3.3.2.7 Programska nit za slanje bitskog toka HTTP protokolom

Programska nit zadužena za slanje bitskog toka HTTP klijentima, prikazana na slici 18, ima ulogu da čita MPEG-TS bitski tok i prosleđuje ga HTTP klijentima.



Slika 18 - Programska nit za slanje bitskog toka HTTP klijentima

3.3.2.8 Memorijski zahtevi

Memorijski zahtevi SAT>IP servera izmereni na ciljnoj platformi baziranoj na Linux operativnom sistemu su dati u tabeli 15.

	Flash	RAM
Operativni sistem i drajveri	8 MB	19.9MB
SAT>IP server	2Mb	2.8MB po klijentu (ukupno 2.8MB*N gde je N maksimalni broj podržanih klijenata)

Tabela 15 – Memorijski zahtevi SAT>IP servera

4. Rezultati ispitivanja

U ovom poglavlju dati su rezultati testova koji su korišćeni za funkcionalnu verifikaciju SAT>IP servera i klijenta kao i za potvrdu njihove izdržljivosti. Za verifikaciju servera korišćena su dva skupa testova, jedan oficijalni skup testova dobijen od strane kreatora same SAT>IP specifikacije i drugi sastavljen na početku samog projekta koji je imao za cilj da dodatno potvrdi funkcionalnost i izdržljivost servera. Za SAT>IP klijenta zvanični set testova nije postojao, već je klijentska biblioteka testirana u dve faze: pisanjem test programa koji je služio za verifikaciju interoperabilnosti sa verifikovanim serverima i korišćenjem testova izdržljivosti korišćenih za verifikaciju dekodera u koji je biblioteka integrisana.

4.1 Verifikacija servera

Proces verifikacije SAT>IP server je podeljen u dve faze:

- prva faza predstavlja testiranje korišćenjem oficijalnog skupa testova koji su napisani na početku projekta
- druga faza predstavlja slanje ciljne platforme na kojoj se nalazi programska podrška koja sadrži SAT>IP server u laboratorije kompanije SES gde su se izvršavali testovi koji nisu dostupni javnosti. Ova faza je predstavljala iterativnu fazu koja je zahtevala više ciklusa slanja unapredene programske podrške u laboratorije SES-a, što je na kraju rezultovalo dobijanjem sertifikata o kompatibilnosti servera sa specifikacijom i dozvolom za njegovu distribuciju pod zvaničnim logom.

Prva faza testiranja podrazumevala je unošenje oficijalnih testova i dodatno napisanih testova u sistem za testiranje INTENT+. Ovako dobijeni skup testova je korišćen tokom razvoja servera za periodično proveravanje stanja programske podrške pre slanja na verifikaciju u SES

laboratorije za finalnu proveru server. Uslov za otpočinjanje finalne verifikacije je bio da svi oficijalni testovi imaju pozitivan status.

Skup testova koji su uneti na Intent + je podeljen na:

- testove izdržljivosti i
- funkcionalne testove

4.1.1 Testovi izdržljivosti

Svrha ovih testova je da dokažu da server može da radi pod punim opterećenjem u ekstremnim uslovima, čime se obezbeđuje sigurnost u radu kod krajnjih korisnika. Ova grupa testova je od velike važnosti zbog same prirode uređaja koji predstavlja server koji nije namenjen da se gasi i često je u kućnoj mreži lociran na nepristupačnim mestima kao što je, na primer, krov. Detaljan opis testova koji pripadaju ovoj grupi je dat u tabeli 16,

Test definicija	Testni slučaj	Opis
Slanje bitskog toka na maksimalni broj kanala tokom dugog perioda	RTP SD kanali	Slanje RTP SD bitskog toka u periodu od 9 sati
	RTP HD kanali	Slanje RTP HD bitskog toka u periodu od 9 sati
	HTTP HD kanali	Slanje HTTP HD bitskog toka u periodu od 9 sati
Otpornost na gubitak signala	Isti kao i test definicija	Tokom slanja bitskog toka svakih 10 sekundi izvaditi koaksijalni kabl iz servera i vratiti ga nazad
Merenje količine bita koje server u sekundi može da pošalje u mrežu	Isti kao i test definicija	Korišćenjem modulatora signala Alitronika pronaći maksimalni broj bita u sekundi koji server može da pošalje u mrežu
Otpornost na pogrešne komande	Isti kao i test definicija	Korišćenjem test programa poslati poruke sa namerno unetim greškama serveru
Nezavisnost sesija	Isti kao i test definicija	Pokrenuti bitski tok na jednom klijentu, zatim pokrenuti ostale klijente da menjaju kanal na 2 sekunde i utvrditi da promene kanala ne utiču na bitski tok prvog klijenta
Menjanje kanala	Tokom 9 sati	Menjanje kanala na svakom klijentu na 3 minuta tokom 9 sati
	2000 puta	Menjanje kanala na svakom klijentu na 2 sekunde 2000 puta
	30 minuta	Menjanje kanala na svakom klijentu na 10 sekundi tokom 30 minuta

Tabela 16 – Testovi izdržljivosti

Pre slanja servera na sertifikaciju u SES, svi testovi izdržljivosti su prošli i utvrđeno je da je server u stanju da pošalje u mrežu sa svakog birača kanala maksimalni broj bita po jednom HD kanalu koji iznosu 20 Mbps.

4.1.2 Funkcionalni testovi

Funkcionalni testovi predstavljaju skup testova napisanih od strane kreatora SAT>IP standarda koji se izvršavaju u okviru posebnog programa pisanog za operativni sistem Windows. Ovi testovi oponašaju SAT>IP klijente i testiraju funkcionalnost svih protokola definisanih u okviru specifikacije. Kako su ovi testovi zatvorenog tipa u okviru ovog poglavlja biće prikazan samo njihov osnovni opis dat u tabeli 17.

Grupa testova	Broj testova	Kratak opis
RTSP	255	Testovi koji testiraju RTSP protokol SAT>IP servera
RTCP	9	Testovi koji testiraju RTCP protokol SAT>IP servera
IGMP	24	Testovi koji testiraju IGMP protokol SAT>IP servera
RTP	10	Testovi koji testiraju RTP protokol SAT>IP servera
HTTP	1	Testovi koji testiraju HTTP protokol SAT>IP servera
Testovi sa više klijenata	27	Testovi koji verifikuju pristup dva klijenta istom bitskom toku gde je jedan klijent vlasnik bitskog toka
UPnP	79	Testovi koji testiraju adresirenja, prijavljivanje i opisivanje
Ukupno:	405	

Tabela 17 – Oficijalni testovi

4.2 Verifikacija SAT>IP klijentske biblioteke

SAT>IP klijenti po svojoj specifikaciji ne moraju da podržavaju sve protokole date specifikacijom, već samo one koji su potrebni za normalno funkcionisanje klijenta u skladu sa njegovom proizvodnom specifikacijom.

Zvaničan skup testova za verifikaciju klijenta nije bio dostupan tokom razvoja biblioteke te je biblioteka verifikovana korišćenjem servera koji je prethodno sertifikovan od strane SES-a tako što je dokazano da je ona u stanju da pronade, konfiguriše server i od njega preuzme bitski tok.

Kako je SAT>IP klijent realizovan kao biblioteka koja oponaša fizički birač kanala i demultiplekser i nema funkcionalnost da preuzeti saržaj dekoduje i prikaže proces verifikacije je podeljen u dva dela:

- verifikacija SAT>IP protokola pomoću konzolnog test programa koji je dizajniran da funkcionalno verifikuje sve protokole definisane specifikacijom. Listing test programa je dat u okviru Dodatka A

- verifikacija biblioteke integrisane u postojeći dekodirer baziran na Android operativnom sistemu, gde je verifikovano da biblioteka može da radi tokom dugog perioda bez smetnji.

5. Pravci za dalji razvoj

SAT>IP tehnologija uvedena je sa ciljem da se distributerima satelitskog, zemaljskog i kablovskog televizijskog sadržaja da šansa da se takmiče sa novim tehnologijama kao što su IPTV i OTT koje velikom brzinom osvajaju tržište.

Klasičan prenos televizijskog signala predstavlja jednosmeran način komunikacije između distributera i krajnjih korisnika gde je praćenje televizijskog sadržaja moguće isključivo preko TV prijemnika sa ili bez dekodera. Pojavom SAT>IP standarda se ovaj vid prenosa stavlja u nešto ravnopravniji položaj u odnosu na IPTV i OTT, iz razloga što omogućuje praćenje televizijskog sadržaja u okviru kućne mreže na uređajima poput pametnih telefona, tableta, personalnih računara,...

Prilikom razvoja servera i klijenta uočeni su neki nedostaci koji bi dali veće šanse SAT>IP i klasičnom prenosu televizijskog sadržaja u odnosu na IPTV i OTT tehnologije:

- Rešavanje pitanja otključavanja sadržaja na serveru
- Implementacija mehanizma za trajno dodeljivanje birača kanala jednom ili grupi klijenata

5.1 Otključavanje zaštićenog sadržaja na serveru

Prenos zaštićenog sadržaja u neizmenjenom obliku ka klijentima u velikoj meri ograničava gledanje takvog sadržaja TV aparate i dekodere koji su opremljeni CAS sistemima i onemogućava dekodovanje na prenosnim uređajima. Neki od problema koje treba rešiti su:

- Obezbeđivanje modela plaćanja usluga koji je primeljiv na prenosnim uređajima

- Efikasna zaštita otključanog sadržaja koji se distribuira u kućnu mrežu - ponovno zaključavanje sadržaja sa algoritmom i ključem, prilikom slanja bitskog toka od servera ka klijentu.

Rešavanje ovog problema je kompleksna stvar koja pored tehničkih ima i pravne poteškoće, te je stoga, u Aprilu 2015 formirana grupa SAT>IP Alliance koja je okupila neke od vodećih satelitskih operatera i proizvođače opreme i programske podrške sa ciljem da se unapredi pozicija SAT>IP tehnologije na tržištu.

5.2 Dodavanje birača kanala jednom ili grupi klijenata

Jedan od načina na koji se SAT>IP tehnologija može efikasno upotrebiti jeste njena instalacija u okviru stambenih zgrada, gde bi umesto postavljanja velikog broja satelitskih antena na krovu i postavljanja koaksijalnih kablova, instalacija bila dosta jednostavnija i unapredila bi korisničko iskustvo omogućavajući:

- korišćenje jedne satelitske antene,
- postavljanje više SAT>IP servera na krovu ili u potkrovlju zgrade
- Prenos TV signala do svakog stana putem naponskih parica korišćenjem PLC adaptera bez ikakve potrebe za građevinskim radovima potrebnim za instalaciju koaksijalnih kablova
- gledanje TV kanala na prenosnim uređajima.

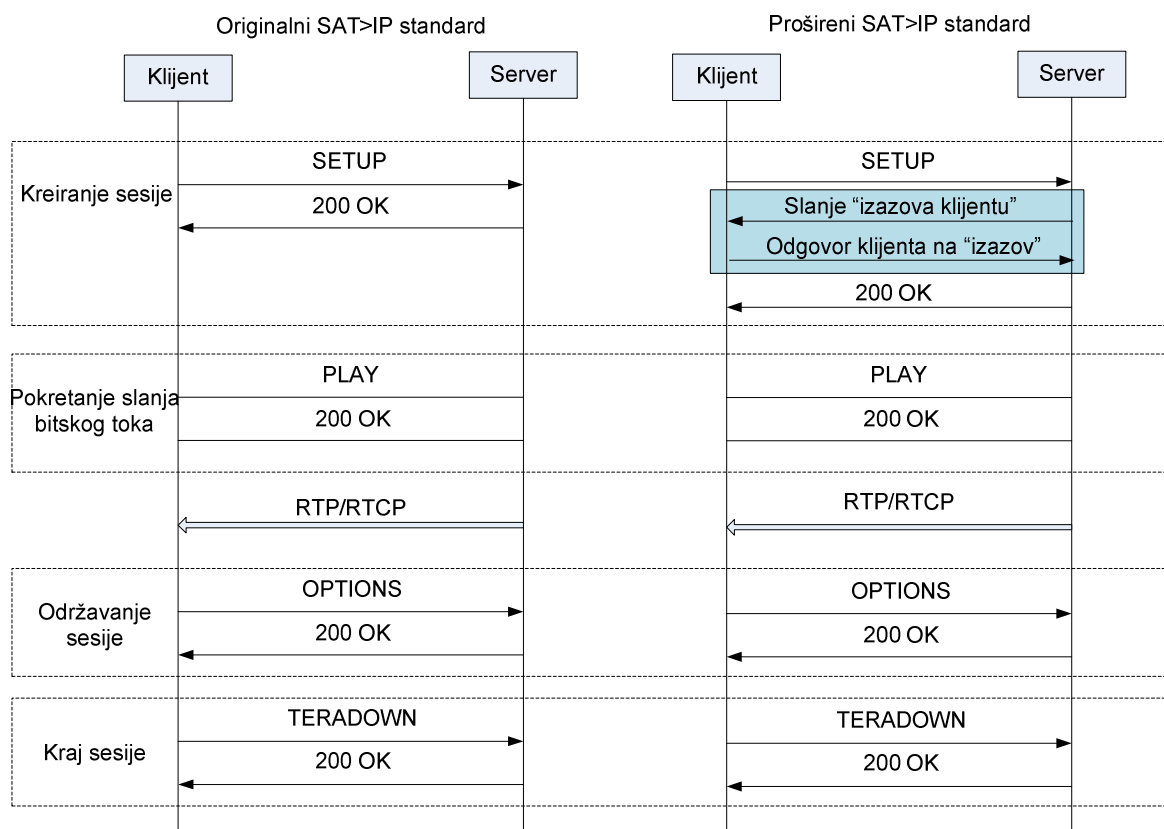
Problem koji ograničava upotrebu SAT>IP standarada u ovakvom tipu instalacije je što sam standard ne daje mogućnost dodeljivanja birača kanala na server jednom ili grupi klijenata što u realnim uslovima može da dovede to trke za resursima.

Korišćenje servera koji rade u statičkom režimu, koje je podržano standardom, je primenljivo ali postavlja ograničenje na broj kanala koji bi mogli da se distribuira na mrežu. Sa porastom broja kanala povećavaju se zahtevi za propusnom moći mreže što je u velikoj meru usložnjava i poskupljuje.

Predlog za prevazilaženje ovog problema je proširivanje SAT>IP protokola i dodavanje mogućnosti da se birač kanala dodeli jednom ili grupi klijenata [8]. Ovakvim pristupom bi se svakom korisniku omogućilo gledanje svih TV sadržaja koji su dostupni na satelitu. Kompleksnost mreže u ovom slučaju ne zavisi od broja kanala nego od ukupnog broja korisnika, što omogućava relativno laku i jeftinu instalaciju u manjim stambenim zgradama.

Predloženo rešenje, prikazano na slici 19, podzumeva dodeljivanje birača kanala na server na osnovu jedinstvenog ključa za svakog klijenta čija se verifikacija obavlja tokom formiranja sesije u sledećim koracima:

- SAT>IP klijent šalje SETUP zahtev serveru
- Server klijentu šalje slučajno generisan niz karaktera zaštićen jedinstvenim ključem
- Klijent dekoduje primljeni niz karaktera na osnovu ključa i šalje ga nazad serveru
- Server upoređuje niz karaktera koji je dobio od klijenta sa onim koje je generisan posle primanja SETUP poruke
- Ukoliko su nizovi karaktera identični server vraća klijentu 200 OK poruku i od te tačke se dalja komunikacija između klijenta i servera obavlja na isti način kao i kod originalnog SAT>IP standarda
- Ukoliko su nizovi karaktera različiti, server vraća kod greške i onemogućuje klijentu pristup biraču kanala.



Slika 19 – Predlog proširenja SAT>IP standarda

6. Poređenje SAT>IP i postojećih tehnologija

6.1 Metrike za evaluaciju tehnologije

SAT>IP predstavlja podskup UPnP-DLNA arhitekture i može se proširiti da postane server sadržaja. Jedan od načina da se ovo postigne je da se SAT>IP server, u neizmenjenoj formi, koristi kao deo DLNA birača kanala koji bi bio zadužen za slanje HTTP bitskog toka prikazivačima multimedijalnog sadržaja. U ovom slučaju DLNA birač kanala bi imao ulogu da klijentima dostavi listu kanala putem CDS objekta, gde bi se u svakom elementu, koji predstavlja jedan kanal, URI niz karaktera promenio tako da sadrži HTTP zahtev saglasan sa SAT>IP standardom koji je opisan u poglavlju [3.1.4.3](#). Na ovaj način, SAT>IP tehnologijom se korisniku daje mogućnost boljeg organizovanja deljenog sadržaja u kućnoj mreži i daje se mogućnost pristupu televizije koja se emituje uživo uređajima koji nisu opremljeni SAT>IP klijentom.

Uzimajući u obzir specifičnosti svake navedene tehnologije, metrike za njihovu evaluaciju su podeljene u dve grupe:

1. mrežna infrastruktura i
2. korisničko iskustvo.

6.1.1 Mrežna infrastruktura

IPTV i OTT tehnologije podrazumevaju prenos sadržaja preko IP mreža od distributera do korisnika, gde su distributeri geografski udaljeni od krajnjeg korisnika i gde se za prenos u slučaju UTT-a koriste mreže otvorenog tipa, a u slučaju IPTV-ja mreže zatvorenog tipa. Obe tehnologije zahtevaju pretplatnički odnos i plaćanje usluga operaterima.

SAT>IP i DLNA birač kanala sa druge strane su prvenstveno namenjeni za rad u okviru kućnih mreža koje su zatvorenog tipa i kod kojih prenos sadržaja ne podrazumeva postojanje interent veze i postojanje pretplatničkog odnosa.

Potreban propusni opseg IP mreže koji je potreban za pravilno funkcionisanje svake od navedenih tehnologija zavisi od njenog tipa i za svaku od njih važi:

- Propusni opseg potreban za funkcionisanje SAT>IP tehnologije je u direktno proporcionalan broju birača kanala na serveru i maksimalnoj količini bita u sekundi koje server može da pošalje u mrežu. Prilikom testiranja i merenja u realnim uslovima utvrđeno je da je broj bita po sekundi potrebnih za prenos jednog HD kanala približno 15Mbps. Prilikom projektovanja kućne mreže i instalacije servera mora se voditi računa da se izborom adekvatne opreme obezbedi da je:

$$\text{Broj birača kanala} * 15\text{Mbps} + X < \text{od propusne moći deklarisanе mrežnom opremom}$$

Gde je X dodatan broj bita u sekundi koji treba ostaviti za saobraćaj ka interentu i ostalim servisima koje je korisnik koristio pre instalacije SAT>IP sistema. Preporuka je da se mreža konstruiše gigabitnom opremom i bežičniom opremom koja minimalno zadovoljava standard 802.11n.

- U slučaju DLNA birača kanala propusni opseg potreban za njegovo pravilno funkcionisanje se može izraziti na isti način kao i u slučaju SAT>IP tehnologije.
- Za razliku od DLNA birača kanala i SAT>IP tehnologija, propusni opseg koji koristi IPTV system za prenos sadržaja je konfigurisan od strane operatora i njegova kontrola nije dostupna krajnjem korisniku.
- OTT sistemi koriste mreže otvorenog tipa koje video sadržaje prenose putem Interneta. Zbog same prirode mreže, kod ovih sistema ne postoji garancija propusnog opsega te se kod njih koriste algoritmi za prilagođavanje bitskog toka trenutnoj propusnoj moći kanala za prenos kao što su HLS ili MPEG DASH.

6.1.2 Korisničko iskustvo

Korisničko iskustvo za sve nevedene tehnologije predstavlja najvažniji parametar koji u najvećoj meri određuje njihovu poziciju na tržištu. Parametri koji su u okviru ovog rada uzeti u razmatranje, a koji su u direktnoj korelaciji sa stepenom zadovoljstva krajnjih korisnika su:

1. Garancija kvaliteta servisa – QoS (eng. Quality of Service)
2. Vreme potrebno za menjanje kanala

3. Broj podržanih kanala u isto vreme
4. Postojanje naprednih servisa

6.1.2.1 Garancija kvaliteta servisa

Garancija kvaliteta servisa zavisi od tipa IP mreža koje se koriste za prenos.

IPTV koriste mreže zatvorenog tipa kojima se upravlja od strane IPTV distributera. Nesmetan prenos visoko kvalitetnog video sadržaja, IPTV operateri obezbeđuju davanjem većeg prioriteta IPTV paketima u odnosu na ostali saobraćaj i obezbeđujući propusni opseg ka svakom korisniku dovoljno širok da može da prenese u isto vreme visokokvalitetni IPTV sadržaj i internet saobraćaj.

OTT sistemi za prenos video sadržaja koriste internet koji je mreža otvorenog tipa i kod koje ne postoji mehanizam za garanciju kvaliteta servisa. OTT sistemi ovaj problem rešavaju prilagođavanjem bitskog toka trenutnoj propusnoj moći kanala za prenos ka svakom korisniku. Na ovaj način se omogućava prenos u nepredvidivim uslovima na račun smanjenog kvaliteta prenesenog sadržaja.

DLNA birač kanala i SAT>IP sistemi su prvenstveno namenjeni za korišćenje u okviru kućnih mreža koje su zatvorenog tipa. Ipak, za razliku od IPTV tehnologije, kućne mreže konfiguriše i njima upravlja krajnji korisnik, te garancija kvaliteta servisa zavisi od trenutnog zauzeća propusnog opsega kućne mreže kao i njene konfiguracije.

6.1.2.2 Vreme potrebno za menjanje kanala

Vreme potrebno za menjanje kanala predstavlja vreme koje se meri od momenta kada korisnik promeni kanal korišćenjem daljinskog upravljača pa sve dok se ne prikaže slika na ekranu. Ovo vreme postoji na svim televizijskim uređajima, a naročito je izraženo kod digitalne televizije koja se prenosi putem IP mreža. Faktori koji utiču na vreme potrebno za menjanje kanala su:

1. Karakteristike mreže i protokola - gde su neki od glavnih faktora vreme potrebno za obradu IGMP Leave i IGMP Join poruka na serveru i mehanizmi multikast rutiranja. Ovi faktori uglavnom predstavljaju mali deo celokupnog vremena potrebnog za prenos kanala i njihovo trajanje je između 50 ms i 200 ms [\[10\]](#).
2. Struktura MPEG bitskog toka - gde su glavni faktori vreme potrebno za pronalaženje PAT i PMT tabela, dobavljanje ključeva za dekrpciju koji iznose između 100 ms i 500 ms [\[10\]](#) i dobavljanje ključnog okvira za početak prikaza slike (eng. I-Frame) gde je frekvencija slanja obično oko 500ms [\[10\]](#)

3. Karakteristike dekodera od kojih na menjanje kanala utiču veličine memorijskih struktura za privremeno skladištenje bitskog toka (tipično između 750 ms i 2 s [10]) i kašnjenje dekodera koje iznosi oko 50 ms [10].

S obzirom da je za merenje vremena promene kanala OTT i IPTV servisa potrebno obezbediti identičan bitski tok na oba servera, što nije moguće bez saglasnosti distributera, merenje brzine promene kanala je obavljeno nad DLNA biračem kanala i SAT>IP serverom.

Kako su DLNA i SAT>IP tehnologije namenjene za instalaciju u malim mrežama zatvorenog tipa, merenje i upoređivanje je moguće obaviti korišćenjem identičnog bitskog toka. Okruženje za laboratorijka ispitivanja i merenja je prikazano na slici 20 i sastoji se od:

1. Personalnog računara koji zajedno sa Alitronika modulatorom predstavlja izvor sateliteskog sadržaja i u potpunosti zamenjuje satelitsku antenu i ostalu opremu potrebnu za pristup sadržajima koji se emituju uživo. Ovako konfigurisan izvor omogućava korišćenje identičnog bitskog toka, čime se osigurava tačnost merenja.
2. RK3040-00 SAT>IP server koji je realizovan u sklopu ovog rada i nadogradjen DLNA biračem kanala. Korišćenjem jedne platforme za serviranje uređaja anulirane su vremenske razlike koje mogu nastati korišćenjem birača kanala različitih proizvođača.
3. Ruter koji obezbeđuje funkcionisanje oba protokola na istoj mreži i
4. Personalnog računara koji predstavlja klijenta. Personalni računar radi pod Windows 7 operativnim sistemom i za merenje koristi:
 - a. VLC klijenta koji je saglasan sa DLNA i SAT>IP protokolom
 - b. Wireshark koji predstavlja program za snimanje mrežnog saobraćaja

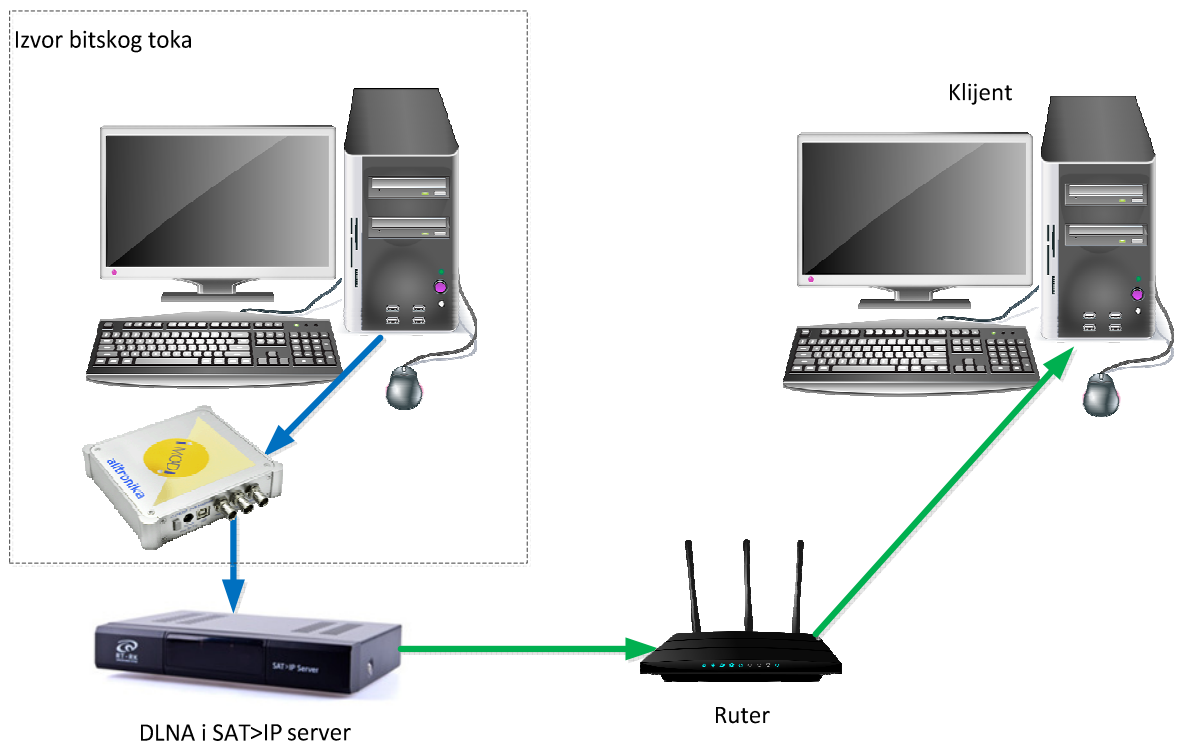
VLC klijent je prilikom merenja brzine promene kanala korišćen kao:

1. DLNA klijent koji prilikom izbora kanala iz liste koju daje DLNA birač kanala server šalje sledeći URL:

```
http://ip/?src=1&freq=10744&msys=dvbs&plts=off&fec=&pol=h&ro=0.35&sr=22000
&mtype=qpsk&pids=0,18,400,401,403,404
```

2. SAT>IP klijent koji šalje sledeći URL-a serveru

```
rtsp://ip:554/?src=1&freq=10744&msys=dvbs&plts=off&fec=&pol=h&ro=0.35&sr=2
2000&mtype=qpsk&pids=0,18,400,401,403,404
```



Slika 20 – Okruženje za merenje brzine promene kanala

Uzimajući u obzir da je okruženje za merenje napravljeno tako da i serveri i klijenti koriste iste resurse, na brzinu promene kanala u slučaju DLNA birača kanala i SAT>IP tehnologije utiču vreme potrebno da se paketi prenesu putem IP mreže i veličina memorijske strukture potrebne za skladištenje HTTP paketa na klijentskoj strani pre nego što se počne dekodovanje. Izmerene vrednosti su date u tabeli 18.

	VLC SAT>IP	VLC DLNA
Request [packet/time]	RTSP SETUP – 17 / 1.4129 s	HTTP GET – 43 / 4.8624 s
PAT [packet/time]	71 / 2.4024 s	1577 / 7.1759
PMT [packet/time]	90 / 2.4267 s	2025 / 7.6077
HTTP buffer	0	1s
iFrame	0.5 s	0.5s
Ukupno	$2.4267s-1.4129s+0.5s=1.5138s$	$7.6077s-4.8624s+1s+0.5s=4.2453 s$

Tabela 18 – Vreme potrebno za promenu kanala

6.2 Rezultati evaluacije

Rezultati evaluacije tehnologija na osnovu definisanih metrika su dati u tabeli 19.

	OTT	IPTV	DLNA birač	SAT>IP
--	-----	------	------------	--------

			kanala	
Tip mreže	Internet (mreže otvorenog tipa)	Mreže zatvorenog tipa	Mreže zatvorenog tipa	Mreže zatvorenog tipa
Garancija kvaliteta servisa (QoS)	Ne postoji. Koriste se adaptivni algoritmi (HLS, MPEG'DASH)	Da	Da (zavisi i od trenutnog opterećenja kućne mreže)	Da (zavisi i od trenutnog opterećenja kućne mreže)
Primeri	YouTube, Netflix, Hulu, Amazon, BBC, iPlayer,...	IPTV Telekoma Srbije	Instalacija DLNA birača kanala u kućnu mrežu	Instalacija SAT>IP servera u kućnu mrežu
Protokol	Koristi HTTP protokol. Najnoviji trendovi su korišćenje protokola koji prilagođavaju bitski tok propusnom opsegu prenosnog kanala.	Koristi UDP multikast protokol	HTTP	RTSP (unikast ili multikast) i HTTP
Vreme menjanja kanala	-	-	1,5138 s	4,2453 s
Broj podržanih klijenata u isto vreme	Definisan u zavisnosti od operatera, najčešće između 1 i 3	Definisan u zavisnosti od operatera, najčešće između 1 i 3	Zavisi od broja birača kanala na serveru	Zavisi od broja birača kanala na serveru
EPG	Ne	Da	Ne	Da
Plati pa gledaj	Da	Da	Ne	Ne
Mrežni personalni video snimač	Da	Ne	Ne	Ne
Video klub	Da	Da	Ne	Ne
Nedostaci/i zazovi	Nema garancije kvaliteta prenosa sadržaja koji se emituje uživo	IPTV je dosta skuplji, ima veliku konkurenciju u kablovskim operaterima	Ne postoje napredni servisi kao što je EPG	Ne postoji nivo interaktivnosti i naprednih servisa kao kod IPTV ili OTT
Prednosti	Jeftin, fleksibilan u smislu dostupnosti sadržaja na raznim uređajima	Interaktivni servisi, garantovan kvalitet servisa	Kompatibilan sa većinom postojećih dekodera i televizora. Lako se integriše u kućni multimedijalni sistem	Ne zahteva pristup Internetu, pretplatničke ugovore, jeftin za instalaciju jer nakon toga ne zahteva dodatne troškove u vidu mesečnog plaćanja usluga. Širok spektar scenarija gde se može upotrebiti

Tabela 19 – Poređenje SAT>IP sa postojećim tehnologijama

7. Zaključak

U okviru ovog rada realizovani su SAT>IP server na platformi sa ograničenim resursima, baziranoj na operativnom sistemu Linux, i SAT>IP klijentska biblioteka koja je radi testiranja integrisana u postojeći dekodirer RK2010.

SAT>IP server je uspešno prošao oficijalni proces setifikacije u laboratorijama kompanije SES i kao rezultat je dobijena potvrda o saglasnosti servera sa specifikacijom protokola i dozvola za njegovu distribuciju.

SAT>IP klijentska biblioteka je, u nedostatku oficijalnog procesa, verifikovana pomoću testova definisanih na samom startu implementacije i tim testovima je utvrđeno da je biblioteka saglasna sa protokolom i da je, kao što je definisano kao cilj na samom početku, moguće uspešno integrisati u postojeće uređaje.

Postojeće tehnologije za prenos multimedijalnog sadržaja putem IP mreža i njihov ubrzan razvoj, svojim naprednim funkcijama i servisima prete da u potpunosti potisnu klasičan način prenosa televizijskog sadržaja. Pojava SAT>IP standarda je unapredila korisničko iskustvo dajući mogućnost praćenja televizijskog sadržaja na prenosivim uređajima.

Uzimajući u obzir podržane protokole i načine prenosa u okviru mreže SAT>IP standardom, prenos televizijskog sadržaja preko antenskih ili satelitskih sistema postao je uporediv sa trenutno postojećim tehnologijama.

Pravci daljeg razvoja mogu biti usmereni ka unapređivanju protokola sa ciljem bolje kontrole resursa na serveru čime bi se unapredila primena SAT>IP uređaja a samim tim i njihova pozicija na tržištu.

Definisanjem pravila za otključavanje sadržaja na serveru i ponovnom zaštitom sadržaja, algoritmom i ključevima poznatim serveru i klijentima, omogućilo bi se praćenje zaštićenih sadržaja na prenosnim uređajima.

8. Literatura

- [1] *Home networked device Interoperability guidelines V1.0*, DLNA, 2004
- [2] *Networked digital media standards (A UpnP / DLNA overview)*, Allegro Software, Development Corporation , 1740 Massachusetts Avelue, Boxporough, MA 01719, 2006
- [3] IEC 62481-1, *Architecture and Protocols Part-1*, 2011
- [4] Wes Simpson, Howard Freenfield, *IPTV and Interent Video, Expanding the reach of Television Broadcasting*, Second eddition, 2009
- [5] Andreja Samčović, Goran Marković, *Transportne mrežne tehnologije za distribuciju IPTV sadržaja*, Univerzitet u Beogradu – Saobraćajni fakultet, 2010
- [6] Alistair Blair, Gerard Parr, Philip Morrow, Bryan Scotney, Steve Appleby and Mike Nilsson, *A Unified Architecture for Video Delivery Over the Internet*, Faculty of Computing and Engineering, University of Ulster, Coleraine, Northern Ireland and Video Delivery Research, BT Innovate & Design, Adastral Park, Ipswich, England, 2011
- [7] Wikipedia, the free encyclopedia, *Comparison between OTT and IPTV*, 2014
- [8] Milan Acanski, Ilija Basiccevic, Jelena Kovacevic, Milan Savic, *Addition to protocol for satellite, cable and terrestrial signal distribution over IP networks*, RT-RK Institute for Computer Based systems and Faculty of Technical Sciences, University of Novi Sad, 2015
- [9] *SAT>IP protocol specification*, SES S.A., 2015, version 1.2.2
- [10] Wikipedia, the free encyclopedia, *Zap time*, 2015

9. Dodatak A – Test program za verifikaciju SAT>IP klijentske biblioteke

```

/*****
*                                     *
*          Copyright 2013 IWEDIA TECHNOLOGIES          *
*                                     *
*****
*
* MODULE NAME: SAT>IP Client library test application
*
* DESCRIPTION
*
* Command line library test application. Stream is recorded to a file named
* zapp_xxxx
*
*****/

/*****/
/*          Includes          */
/*****/
#include <stdio.h>
#include <pthread.h>
#include <string.h>
#include <stdlib.h>
#include <unistd.h>
#include "sat2ip_command_api.h"
#include "sat2ip_session_api.h"
#include "sat2ip_system.h"

/*****/
/*          Defines          */
/*****/
#define STRNG_LENGTH_256 256
#define NUM_CHANNELS_IN_PLAYLIST 4
// #define USE_HARD_CODED_SERVER
// #define USE_KEY_FOR_SELECTION

```

```

/*****
/*          Typedefs          */
/*****
// SAT>IP client
typedef struct client_session {
    pSATIP_Instance clientInstance; // pointer to client instance
    int zapp;
    int read_thread_end;
    int callback_event_code;      // event code passed by libsipcl
    char key[STRNG_LNGTH_256];    // Last played server
    int channel;                  // Last played channel
    pthread_t reader_tread_id;    // Reading thread
}tClientSession;
typedef tClientSession *pClientSession;

/*****
/*          Global Variables          */
/*****
//Channel list
char ChannelNames[NUM_CHANNELS_IN_PLAYLIST][100] = {
    "arte",
    "Eins Festival",
    "Phoenix",
    "Tagesschau 24"};
char ChannelList[NUM_CHANNELS_IN_PLAYLIST][5][100] = {
    { "?src=1&freq=10744&pol=h&msys=dvbs&mtype=qpsk&sr=22000&fec=56",
      "pids=0,1,2,16,17,18,20", "pids=0,1,2,16,17,18,20,400",
      "pids=0,1,2,16,17,18,20,400,401",
      "pids=0,1,2,16,17,18,20,400,401,402"
    },
    { "?src=1&freq=10744&pol=h&msys=dvbs&mtype=qpsk&sr=22000&fec=56",
      "pids=0,1,2,16,17,18,20",
      "pids=0,1,2,16,17,18,20,200",
      "pids=0,1,2,16,17,18,20,200,201",
      "pids=0,1,2,16,17,18,20,200,201,202"},
    { "?src=1&freq=10744&pol=h&msys=dvbs&mtype=qpsk&sr=22000&fec=56",
      "pids=0,1,2,16,17,18,20",
      "pids=0,1,2,16,17,18,20,500",
      "pids=0,1,2,16,17,18,20,500,501",
      "pids=0,1,2,16,17,18,20,500,501,502"
    },
    { "?src=1&freq=10744&pol=h&msys=dvbs&mtype=qpsk&sr=22000&fec=56",
      "pids=0,1,2,16,17,18,20",
      "pids=0,1,2,16,17,18,20,100",
      "pids=0,1,2,16,17,18,20,100,101",
      "pids=0,1,2,16,17,18,20,100,101,102"
    },
    },
};
/*****
/*          Function prototypes          */
/*****
void MainMenu();
void GetInputFromUser(char *output, int size);

```

```

void *cmdLineMainThread(void* args);
void SPICLCallback(int callback_event_code, void *cb_arg);
void *streamReaderThread(void* args);

/*****
/*****
/**          MAIN Function          **/
/*****
/*****

int main()
{
    //pSATIP_Instance clientInstance;
    pthread_t cmdLine_tread_id;
    tClientSession clientSession;

    // By default enable only integration and basic SIPCL logs
    SATIP_SESSION_EnableLogs (SATIP_DBG_INTEGRATION|SATIP_DBG_BASIC);

    // Initialize client
    if (SATIP_SESSION_InitClient (&clientSession.clientInstance,
        SPICLCallback,
        (void*) (&clientSession),
        eSATIP_SESSION_DISABLE_DTCPIP) !=eSATIP_SESSION_OK)
        return -1;

    // Initialize thread arguments
    clientSession.zapp = 0;
    clientSession.read_thread_end = -1;
    strcpy(clientSession.key, "none");
    clientSession.channel = -1;

    // Create command line thread
    pthread_create(&cmdLine_tread_id, NULL, cmdLineMainThread, &clientSession);
    pthread_join(cmdLine_tread_id, NULL);

    SATIP_LOG(SATIP_DBG_INTEGRATION, LOG_NORMAL, "Client closed.\n");
    return 0;
}

/*****
* Function Name      : cmdLineMainThread
*
* Description        : Command line thread
*
* Side effects       :
*
* Comment            : application thread
*
* Parameters         : args - pointer to the pMainThreadArgs
*
* Returns            : nothing
*
*****/

```

```

void *cmdLineMainThread(void* args)
{
    pSATIP_ServerListNode pServerlist;
    pClientSession clientSession;
    char inputBuffer[STRNG_LNGTH_256];
    pSATIP_Instance clientInstance;
    int i;

    SATIP_LOG(SATIP_DBG_INTEGRATION, LOG_NORMAL, "[cmdLineMain] thread started.\n");

    // Get thread parametars
    if (args==NULL) {
        SATIP_LOG(SATIP_DBG_INTEGRATION, LOG_ERROR,
            "[cmdLineMain] - thread arguments missing. <%s:%d>\n", __FILE__, __LINE__);
        return NULL;
    }
    clientSession = (pClientSession) args;
    clientInstance = (pSATIP_Instance) clientSession->clientInstance;

    while(1)
    {
        MainMenu();
        GetInputFromUser(inputBuffer, STRNG_LNGTH_256);
        //-----
        // 1. List servers
        //-----
        if (strcmp(inputBuffer, "1") == 0)
        {
            int i = 0;
            pServerlist = SPICL_SESSION_GetListOfServers(&clientInstance);
            if (pServerlist==NULL) {
                SATIP_LOG(SATIP_DBG_INTEGRATION, LOG_NORMAL, "List of servers is empty!\n");
            } else {
                SATIP_LOG(SATIP_DBG_INTEGRATION, LOG_NORMAL, "List of servers:\n");
                while (pServerlist) {
                    SATIP_LOG(SATIP_DBG_INTEGRATION, LOG_NORMAL,
                        "\t%d - <Name = %s> <UDN = %s>\n", i++,
                        pServerlist->acFriendlyName, pServerlist->acUDN);
                    pServerlist = pServerlist->next;
                }
            }
        }
        //-----
        // 2. Select Server
        //-----
        if (strcmp(inputBuffer, "2") == 0)
        {
#ifdef USE_HARD_CODED_SERVER
            // Hardcode server used by this client. The upnp phase will
            // be performed, all servers discovered but
            // we are hardcoding search key.
            strcpy(inputBuffer, "TB100-SAT2IP-Server (mlh-miniFitPC)");
#endif
        }
    }
}

```

```

#elif USE_KEY_FOR_SELECTION
    // Selection based on server name
    SATIP_LOG(SATIP_DBG_INTEGRATION, LOG_NORMAL, "Enter server name: ");
    GetInputFromUser(inputBuffer, STRNG_LNGTH_256);
#else
    // Selection based on server name (server name given through index in list)
    int i = 0;
    int j = 0;
    SATIP_LOG(SATIP_DBG_INTEGRATION, LOG_NORMAL, "Select server:");
    GetInputFromUser(inputBuffer, STRNG_LNGTH_256);
    j = atoi(inputBuffer);

    pServerlist = SPICL_SESSION_GetListOfServers(&clientInstance);

    if (pServerlist==NULL) {
        SATIP_LOG(SATIP_DBG_INTEGRATION, LOG_NORMAL,
            "List of servers is empty!\n");
        strcpy(clientSession->key, "none");
    } else {
        while (pServerlist) {
            SATIP_LOG(SATIP_DBG_INTEGRATION, LOG_NORMAL,
                "\t%d - <Name = %s> <UDN = %s>\n",
                i, pServerlist->acFriendlyName, pServerlist->acUDN);
            if(i>=j) break;
            i++;
            pServerlist = pServerlist->next;
        }

        strcpy(clientSession->key,pServerlist->acFriendlyName);
    }
#endif

if (SATIP_SESSION_SelectServer(&clientInstance,
    clientSession->key, 0)==eSATIP_SESSION_OK){
    int err = SATIP_SendCmdTEARDOWN(&clientSession->clientInstance));
    if((err !=eSATIP_SESSION_OK){
        SATIP_LOG(SATIP_DBG_INTEGRATION, LOG_WARNING,
            "Sending TEARDOWN failed. <%s:%d>\n", __FILE__, __LINE__);
    }
    //server found check if there was active playback
    if (clientSession->channel>=0){
        // reader thread is not started yet.
        clientSession->read_thread_end = 1;

        char cMediaTransportType;
        if(SATIP_SESSION_GetMediaTransportType(&clientSession->clientInstance,
            &cMediaTransportType) !=eSATIP_SESSION_OK){
            continue;
        }

        switch(cMediaTransportType){
            case eSATIP_SESSION_RTSP_UNICAST:
            case eSATIP_SESSION_RTSP_MULTICAST_CREATE:
            case eSATIP_SESSION_RTSP_MULTICAST_JOIN:

```

```

        usleep(500);
        // Send tuning parameters
        if (SATIP_SendCmdSETUP(&clientSession->clientInstance,
            ChannelList[clientSession->channel][0] != eSATIP_SESSION_OK) {
            continue;
        }

        //Send list of PIDS to the server
        if (SATIP_SendCmdPLAY(&clientSession->clientInstance,
            ChannelList[clientSession->channel][4],
            /*cPIDOperator*/eSATIP_PIDSLIST) != eSATIP_SESSION_OK) {
            continue;
        }
        break;
    case eSATIP_SESSION_HTTP_8080:
    case eSATIP_SESSION_HTTP_80:{
        //Send HTTP GET to the server
        char query[1024];
        strcpy(query, ChannelList[clientSession->channel][0]);
        strcat(query, "&");
        strcat(query, ChannelList[clientSession->channel][4]);
        if (SATIP_HTTP_SendCmdGET(&clientSession->clientInstance,
            query) != eSATIP_SESSION_OK) {
            continue;
        }
    }
    break;
    default:
        SATIP_LOG(SATIP_DBG_BASIC, LOG_ERROR,
            "Wrong media transport type. <%=:d>\n", __FILE__, __LINE__);
        continue;
    }

    // Increase zapp index
    clientSession->zapp++;
}

} else {
    SATIP_LOG(SATIP_DBG_INTEGRATION, LOG_ERROR, "Server not found! \n");
}
}

//-----
// 3. Zapp all pids at once (VLC like)
//-----
if (strcmp(inputBuffer, "3") == 0) {

    SATIP_LOG(SATIP_DBG_INTEGRATION, LOG_NORMAL, "Available channels: \n");

    for (i = 0; i < NUM_CHANNELS_IN_PLAYLIST; i++) {
        SATIP_LOG(SATIP_DBG_INTEGRATION, LOG_NORMAL, "\t%d. [%s]:\t\t%s&%=s\n",
            i, ChannelNames[i], ChannelList[i][0], ChannelList[0][4]);
    }
}

```

```

}

GetInputFromUser(inputBuffer, STRNG_LNGTH_256);
if ((atoi(inputBuffer)>=0)&&(atoi(inputBuffer)<NUM_CHANNELS_IN_PLAYLIST)) {

    clientSession->channel = atoi(inputBuffer);
    // reader thread is not started yet.
    clientSession->read_thread_end = 1;

    char cMediaTransportType;
    if(SATIP_SESSION_GetMediaTransportType(&clientInstance,
        &cMediaTransportType)!=eSATIP_SESSION_OK){
        continue;
    }
    switch(cMediaTransportType){
    case eSATIP_SESSION_RTSP_UNICAST:
    case eSATIP_SESSION_RTSP_MULTICAST_CREATE:
    case eSATIP_SESSION_RTSP_MULTICAST_JOIN:
        // Send tuning parameters
        if(SATIP_SendCmdSETUP(&clientInstance,
            ChannelList[atoi(inputBuffer)][0])!=eSATIP_SESSION_OK){
            continue;
        }

        //Send list of PIDS to the server
        if(SATIP_SendCmdPLAY(&clientInstance, ChannelList[atoi(inputBuffer)][4],
            eSATIP_PIDSLIST)!=eSATIP_SESSION_OK){
            continue;
        }
        break;
    case eSATIP_SESSION_HTTP_8080:
    case eSATIP_SESSION_HTTP_80:{
        //Send HTTP GET to the server
        char query[1024];
        strcpy(query, ChannelList[atoi(inputBuffer)][0]);
        strcat(query, "&");
        strcat(query, ChannelList[atoi(inputBuffer)][4]);
        if(SATIP_HTTP_SendCmdGET(&clientInstance, query)!=eSATIP_SESSION_OK){
            continue;
        }
    }
    }
    break;
    default:
        SATIP_LOG(SATIP_DBG_BASIC, LOG_ERROR,
            "Wrong media transport type. <%s:%d>\n", __FILE__, __LINE__);
        return NULL;
    }

    // Create reader thread
    clientSession->read_thread_end = 0;
    pthread_create( &clientSession->reader_tread_id, NULL,
        streamReaderThread, args);
}

```

```

        // Increase zapp index
        clientSession->zapp++;
    } else
        SATIP_LOG(SATIP_DBG_INTEGRATION, LOG_ERROR,
            "Channel does not exist. <%s:%d>\n", __FILE__, __LINE__);
}

//-----
// 4. Zapp with adding PIDs (Like STB)
//-----
if (strcmp(inputBuffer, "4") == 0) {
    for (i = 0; i < NUM_CHANNELS_IN_PLAYLIST; i++) {
        SATIP_LOG(SATIP_DBG_INTEGRATION, LOG_NORMAL, "\t%d. [%s]:\t\t%s&%s\n", i,
            ChannelNames[i], ChannelList[i][0], ChannelList[0][4]);
    }
    GetInputFromUser(inputBuffer, STRNG_LNGTH_256);
    if ((atoi(inputBuffer) >= 0) && (atoi(inputBuffer) < NUM_CHANNELS_IN_PLAYLIST)) {
        clientSession->channel = atoi(inputBuffer);
        // reader thread is not started yet.
        clientSession->read_thread_end = 1;

        // Send tuning parameters
        if (SATIP_SendCmdSETUP(&clientInstance,
            ChannelList[atoi(inputBuffer)][0]) != eSATIP_SESSION_OK) {
            continue;
        }

        for (i=1; i<4; i++) {
            //Send list of PIDs to the server
            if (SATIP_SendCmdPLAY(&clientInstance, ChannelList[atoi(inputBuffer)][i],
                /*cPIDOperator*/eSATIP_PIDSLIST) != eSATIP_SESSION_OK) {
                continue;
            }
        }

        // Create reading thread only on first PLAY msg
        if (i==1){
            // Create reader thread
            clientSession->read_thread_end = 0;
            pthread_create( &clientSession->reader_tread_id,
                NULL, streamReaderThread, args);
        }
        usleep(500*1000); // sleep for 500ms
    }

    // Increase zapp index
    clientSession->zapp++;
} else
    SATIP_LOG(SATIP_DBG_INTEGRATION, LOG_ERROR,
        "Channel does not exist. <%s:%d>\n", __FILE__, __LINE__);
}

```

```

//-----
// 5. Send Options
//-----
if (strcmp(inputBuffer, "5") == 0) {
    if (SATIP_UNICAST_SendCmdOPTIONS(&clientInstance) == NULL) {
        SATIP_LOG(SATIP_DBG_INTEGRATION, LOG_ERROR,
            "Sending OPTIONS failed. <%s:%d>\n", __FILE__, __LINE__);
        continue;
    }
}

//-----
// 6. Stop Playback
//-----
if (strcmp(inputBuffer, "6") == 0) {
    int err;
    if (clientSession->read_thread_end == 0) {
        clientSession->read_thread_end = 1;
        pthread_join(clientSession->reader_tread_id, NULL);
    }
    if ((err = SATIP_SendCmdTEARDOWN(&clientInstance)) != eSATIP_SESSION_OK) {
        SATIP_LOG(SATIP_DBG_INTEGRATION, LOG_WARNING,
            "Sending TEARDOWN failed. <%s:%d>\n", __FILE__, __LINE__);
        continue;
    }
}

//-----
// 7. Send Describe
//-----
if (strcmp(inputBuffer, "7") == 0) {
    if (SATIP_UNICAST_SendCmdDESCRIBE(&clientInstance) != eSATIP_SESSION_OK) {
        SATIP_LOG(SATIP_DBG_INTEGRATION, LOG_WARNING,
            "Sending DESCRIBE failed. <%s:%d>\n", __FILE__, __LINE__);
        continue;
    }
    pSATIP_SESSION_ServerDescribe serverDescribe;
    serverDescribe
SATIP_SESSION_GetSelectServerInfo((pSATIP_Instance*) (&clientInstance));
    pSATIP_ActiveStreams pStream = serverDescribe->pSATIP_ActiveStreamsHead;
    char buf [128];
    char buf2[128];
    FILE* dat = NULL;
    char file_name[128];
    strcpy( file_name, "multicast_streams");
    dat = fopen(file_name, "wb");
    while (pStream != NULL) {
        if (pStream->pcConnectionAddress) {
            if (pStream->pcConnectionAddress[0] != '0') {
                int i;
                for (i = 0; i < strlen(pStream->pcConnectionAddress); i++) {
                    if (pStream->pcConnectionAddress[i] == '/')
                        break;
                }
            }
        }
    }
}

```

```

        buf[i] = pStream->pcConnectionAddress[i];
    }
    buf[i++] = 0;
    sprintf( buf2, "%s %d\n", buf, pStream->uiRTPPort);
    fwrite(buf2, strlen(buf2), 1, dat);
}
}
pStream = pStream->next;
}
fclose(dat);

}

//-----
// 8. Set media transport type
//-----
if (strcmp(inputBuffer, "8") == 0) {
    SATIP_LOG(SATIP_DBG_INTEGRATION, LOG_NORMAL, "Available stream types: \n");
    SATIP_LOG(SATIP_DBG_INTEGRATION, LOG_NORMAL, "\t0. unicast \n");
    SATIP_LOG(SATIP_DBG_INTEGRATION, LOG_NORMAL, "\t1. multicast create \n");
    SATIP_LOG(SATIP_DBG_INTEGRATION, LOG_NORMAL, "\t2. multicast join \n");
    SATIP_LOG(SATIP_DBG_INTEGRATION, LOG_NORMAL, "\t3. HTTP on port 8080 \n");
    SATIP_LOG(SATIP_DBG_INTEGRATION, LOG_NORMAL, "\t4. HTTP on port 80 \n");
    SATIP_LOG(SATIP_DBG_INTEGRATION, LOG_NORMAL, "Select stream type:");
    GetInputFromUser(inputBuffer, STRNG_LNGTH_256);
    if (atoi(inputBuffer)==0) {
        SATIP_SESSION_SetMediaTransportType(&clientInstance,
            eSATIP_SESSION_RTSP_UNICAST);
    }else if (atoi(inputBuffer)==1) {
        SATIP_SESSION_SetMediaTransportType(&clientInstance,
            eSATIP_SESSION_RTSP_MULTICAST_CREATE);
    }else if (atoi(inputBuffer)==2) {
        SATIP_SESSION_SetMediaTransportType(&clientInstance,
            eSATIP_SESSION_RTSP_MULTICAST_JOIN);
    }else if (atoi(inputBuffer)==3) {
        SATIP_SESSION_SetMediaTransportType(&clientInstance, eSATIP_SESSION_HTTP_8080);
    }else if (atoi(inputBuffer)==4) {
        SATIP_SESSION_SetMediaTransportType(&clientInstance, eSATIP_SESSION_HTTP_80);
    }else
        SATIP_LOG(SATIP_DBG_INTEGRATION, LOG_ERROR,
            "Stream type does not exist. <%s:%d>\n", __FILE__, __LINE__);
}

if (strcmp(inputBuffer, "9") == 0) {
    while(1){
        int err;
        strcpy(inputBuffer, "1");
        clientSession->channel = atoi(inputBuffer);
        // reader thread is not started yet.
        clientSession->read_thread_end = 1;

        char cMediaTransportType;
        if(SATIP_SESSION_GetMediaTransportType(&clientInstance,

```

```

        &cMediaTransportType) !=eSATIP_SESSION_OK) {
            continue;
        }
        switch(cMediaTransportType) {
        case eSATIP_SESSION_RTSP_UNICAST:
        case eSATIP_SESSION_RTSP_MULTICAST_CREATE:
        case eSATIP_SESSION_RTSP_MULTICAST_JOIN:
            // Send tuning parameters
            if(SATIP_SendCmdSETUP(&clientInstance,
                ChannelList[atoi(inputBuffer)][0]) !=eSATIP_SESSION_OK) {
                continue;
            }

            //Send list of PIDS to the server
            if(SATIP_SendCmdPLAY(&clientInstance,
                ChannelList[atoi(inputBuffer)][4],
                eSATIP_PIDSLIST) !=eSATIP_SESSION_OK) {
                continue;
            }
            break;
        case eSATIP_SESSION_HTTP_8080:
        case eSATIP_SESSION_HTTP_80:{
            //Send HTTP GET to the server
            char query[1024];
            strcpy(query, ChannelList[atoi(inputBuffer)][0]);
            strcat(query, "&");
            strcat(query, ChannelList[atoi(inputBuffer)][4]);
            if(SATIP_HTTP_SendCmdGET(&clientInstance, query) !=eSATIP_SESSION_OK) {
                continue;
            }
        }
        break;
        default:
            SATIP_LOG(SATIP_DBG_BASIC, LOG_ERROR,
                "Wrong media transport type. <%s:%d>\n", __FILE__, __LINE__);
            return NULL;
        }

        // Create reader thread
        clientSession->read_thread_end = 0;
        pthread_create( &clientSession->reader_tread_id,
            NULL, streamReaderThread, args);

        // Increase zapp index
        clientSession->zapp++;
        sleep(2);
    }
}

```

```

//-----
// q. Exit
//-----
if (strcmp(inputBuffer, "q") == 0) {
    if(SATIP_SendCmdTEARDOWN(&clientInstance)){
        // There is no need to handle Teardown response
    }
    SATIP_SESSION_CloseClient (&clientInstance);
    break;
}
}
return NULL;
}

/*****
* Function Name      : streamReaderThread
*
* Description        : stream reader thread
*
* Side effects       :
*
* Comment            :
*
* Parameters         :
*
* Returns            :
*
*****/
void *streamReaderThread(void* args)
{
    FILE *dat = NULL;
    pClientSession ReaderArgs = NULL;
    unsigned char buff[4096];
    int nBytesRead = 0;
    char file_name[128];
    int zapp = 0;

    if (args == NULL) {
        SATIP_LOG(SATIP_DBG_INTEGRATION, LOG_ERROR,
            "[streamReaderThread] - invalid thread argument. <%s:%d>\n",
            __FILE__, __LINE__);
        return NULL;
    }
    SATIP_LOG(SATIP_DBG_INTEGRATION, LOG_NORMAL,
        "[streamReaderThread] - thread started.\n");

    ReaderArgs = (pClientSession)args;
    zapp = ReaderArgs->zapp;
#ifdef ANDROID
    sprintf(file_name, "/data/data/zapp_%.4d", zapp);
#else
    sprintf(file_name, "zapp_%.4d", zapp);
#endif
#endif

```

```

dat = fopen(file_name, "wb");
while (ReaderArgs->read_thread_end==0){
    nBytesRead = SATIP_SESSION_Read(&ReaderArgs->clientInstance, buff, 500);
    if (nBytesRead>0) {
        fwrite(buff, nBytesRead, 1, dat);
    }else{
        usleep(100);
    }
}
fclose(dat);
SATIP_LOG(SATIP_DBG_INTEGRATION, LOG_NORMAL,
    "[streamReaderThread] - thread stopped.\n");
return NULL;
}

void MainMenu()
{
    SATIP_LOG(SATIP_DBG_INTEGRATION, LOG_NORMAL, "\n");
    SATIP_LOG(SATIP_DBG_INTEGRATION, LOG_NORMAL, "=====\n");
    SATIP_LOG(SATIP_DBG_INTEGRATION, LOG_NORMAL, "          Management Console          \n");
    SATIP_LOG(SATIP_DBG_INTEGRATION, LOG_NORMAL, "=====\n");
    SATIP_LOG(SATIP_DBG_INTEGRATION, LOG_NORMAL, "\t1. List servers.\n");
    SATIP_LOG(SATIP_DBG_INTEGRATION, LOG_NORMAL, "\t2. Select Server.\n");
    SATIP_LOG(SATIP_DBG_INTEGRATION, LOG_NORMAL, "\t3. Zapp all pids at once.\n");
    SATIP_LOG(SATIP_DBG_INTEGRATION, LOG_NORMAL, "\t4. Zapp with adding pids one by one.\n");
    SATIP_LOG(SATIP_DBG_INTEGRATION, LOG_NORMAL, "\t5. Send Options.\n");
    SATIP_LOG(SATIP_DBG_INTEGRATION, LOG_NORMAL, "\t6. Stop Playback.\n");
    SATIP_LOG(SATIP_DBG_INTEGRATION, LOG_NORMAL, "\t7. Send Describe.\n");
    SATIP_LOG(SATIP_DBG_INTEGRATION, LOG_NORMAL, "\t8. Set media transport type.\n");
    SATIP_LOG(SATIP_DBG_INTEGRATION, LOG_NORMAL, "\tq. Exit.\n\n");
}

void GetInputFromUser(char *output, int size)
{
    memset(output, 0, size);
    int i = 0;
    char a = 0;

    while (a != '\n')
    {
        while ((a = getchar()) == -1) {}
        if (a != '\n')
        {
            output[i] = a;
            i++;
        }
    }
}

void SPICLcallback(int callback_event_code, void *cb_arg)
{
    pClientSession clientSession;
    char *server;

```

```

if (cb_arg==NULL) {
    SATIP_LOG(SATIP_DBG_INTEGRATION, LOG_ERROR, "Callback argument is NULL.\n");
}
clientSession = (pClientSession)cb_arg;
server = clientSession->key;
switch (callback_event_code ) {
case eSATIP_LIST_OF_SERVERS_UPDATED:
    SATIP_LOG(SATIP_DBG_INTEGRATION, LOG_NORMAL, "[CALLBACK] List of servers updated\n");

    if (strcmp(server, "none")!=0) {
        if (SATIP_SESSION_SelectServer(&clientSession->clientInstance, server,
            0)==eSATIP_SESSION_OK) {
            //server found check if there was active playback
            if (clientSession->channel>=0){
                // reader thread is not started yet.
                clientSession->read_thread_end = 1;

                char cMediaTransportType;
                if(SATIP_SESSION_GetMediaTransportType(&clientSession->clientInstance,
                    &cMediaTransportType)!=eSATIP_SESSION_OK) {
                    return;
                }

                switch(cMediaTransportType){
                case eSATIP_SESSION_RTSP_UNICAST:
                case eSATIP_SESSION_RTSP_MULTICAST_CREATE:
                case eSATIP_SESSION_RTSP_MULTICAST_JOIN:
                    // Send tuning parameters
                    if(SATIP_SendCmdSETUP(&clientSession->clientInstance,
                        ChannelList[clientSession->channel][0])!=eSATIP_SESSION_OK) {
                        return;
                    }

                    //Send list of PIDS to the server
                    if(SATIP_SendCmdPLAY(&clientSession->clientInstance,
                        ChannelList[clientSession->channel][4],
                        /*cPIDOperator*/eSATIP_PIDSLIST)!=eSATIP_SESSION_OK) {
                        return;
                    }
                    break;
                case eSATIP_SESSION_HTTP_8080:
                case eSATIP_SESSION_HTTP_80: {
                    //Send HTTP GET to the server
                    char query[1024];
                    strcpy(query, ChannelList[clientSession->channel][0]);
                    strcat(query, "&");
                    strcat(query, ChannelList[clientSession->channel][4]);
                    if(SATIP_HTTP_SendCmdGET(&clientSession->clientInstance,
                        query)!=eSATIP_SESSION_OK) {
                        return;
                    }
                }
            }
        }
    }
}

```

```

        break;
    default:
        SATIP_LOG(SATIP_DBG_BASIC, LOG_ERROR,
            "Wrong media transport type. <%=s:%d>\n", __FILE__, __LINE__);
        return;
    }

    // Increase zapp index
    clientSession->zapp++;
}

} else {
    SATIP_LOG(SATIP_DBG_INTEGRATION, LOG_ERROR, "Server not found! \n");
}
}

break;
case eSATIP_SIGNAL_RETURNED_TO_NORMAL:
    SATIP_LOG(SATIP_DBG_INTEGRATION, LOG_NORMAL,
        "[CALLBACK] Signal level returned to non zero value\n");
    break;
case eSATIP_NO_SIGNAL:
    SATIP_LOG(SATIP_DBG_INTEGRATION, LOG_NORMAL,
        "[CALLBACK] Signal level on tuner is zero\n");
    break;
case eSATIP_TUNING_FAILED:
    SATIP_LOG(SATIP_DBG_INTEGRATION, LOG_NORMAL,
        "[CALLBACK] TUNING Failed\n");
    break;
case eSATIP_TUNER_LOCKED:
    SATIP_LOG(SATIP_DBG_INTEGRATION, LOG_NORMAL,
        "[CALLBACK] TUNING OK.\n");
    break;
case eSATIP_SERVER_INFORMATION_RETRIEVED:
    SATIP_LOG(SATIP_DBG_INTEGRATION, LOG_NORMAL,
        "[CALLBACK] Server informations received\n");
    break;
case eSATIP_SERVER_INFORMATION_RETRIEVING_FAILED:
    SATIP_LOG(SATIP_DBG_INTEGRATION, LOG_NORMAL,
        "[CALLBACK] There are no streams running on server so server describe is empty\n");
    break;
case eSATIP_SERVER_DROPPED:
    SATIP_LOG(SATIP_DBG_INTEGRATION, LOG_NORMAL,
        "[CALLBACK] Server Dropped\n");
    break;
default:
    SATIP_LOG(SATIP_DBG_INTEGRATION, LOG_ERROR,
        "Unknown callback event.\n");
    break;
}
}
}

```