



УНИВЕРЗИТЕТ У НОВОМ САДУ
ФАКУЛТЕТ ТЕХНИЧКИХ НАУКА У
НОВОМ САДУ



Ђорђе Ковачевић

**Реализација подршке за концепт
„другог екрана“ у дигиталној
телевизији**

ДИПЛОМСКИ РАД
- Основне академске студије -

Нови Сад, 2014.



КЉУЧНА ДОКУМЕНТАЦИЈСКА ИНФОРМАЦИЈА

Редни број, РБР:			
Идентификациони број, ИБР:			
Тип документације, ТД:	Монографска документација		
Тип записа, ТЗ:	Текстуални штампани материјал		
Врста рада, ВР:	Завршни (Bachelor) рад		
Аутор, АУ:	Ђорђе Ковачевић		
Ментор, МН:	др Иштван Пап, доцент		
Наслов рада, НР:	Реализација подршке за концепт „другог екрана“ у дигиталној телевизији		
Језик публикације, ЈП:	Српски / латиница		
Језик извода, ЈИ:	Српски		
Земља публиковања, ЗП:	Република Србија		
Уже географско подручје, УГП:	Војводина		
Година, ГО:	2014		
Издавач, ИЗ:	Ауторски репринт		
Место и адреса, МА:	Нови Сад; трг Доситеја Обрадовића 6		
Физички опис рада, ФО: (поглавља/страница/цитата/табела/слика/графика/прилога)	7/27/0/4/9/0/0		
Научна област, НО:	Електротехника и рачунарство		
Научна дисциплина, НД:	Рачунарска техника		
Предметна одредница/Кључне речи, ПО:	Други екран, UPnP протокол, Android		
УДК			
Чува се, ЧУ:	У библиотеци Факултета техничких наука, Нови Сад		
Важна напомена, ВН:			
Извод, ИЗ:	У овом раду је представљено неколико расположивих комуникационих протокола, међу којима је и UPnP, који је изабран за основу овог програмског решења. Описане су компоненте и кораци у раду UPnP протокола као и реализација подршке за концепт „другог екрана“ засноване на овом протоколу.		
Датум прихватања теме, ДП:			
Датум одбране, ДО:			
Чланови комисије, КО:	Председник:	др Јелена Ковачевић, доцент	
	Члан:	др Милан Ђелица, доцент	Потпис ментора
	Члан, ментор:	др Иштван Пап, доцент	



KEY WORDS DOCUMENTATION

Accession number, ANO:		
Identification number, INO:		
Document type, DT:	Monographic publication	
Type of record, TR:	Textual printed material	
Contents code, CC:	Bachelor Thesis	
Author, AU:	Đorđe Kovačević	
Mentor, MN:	Ištván Pap, PhD	
Title, TI:	Implementation of support for the second screen concept in digital television	
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:		
Publisher, PB:	Author's reprint	
Publication place, PP:	Novi Sad, Dositeja Obradovica sq. 6	
Physical description, PD: <small>(chapters/pages/ref./tables/pictures/graphs/appendices)</small>	7/27/0/4/9/0/0	
Scientific field, SF:	Electrical Engineering	
Scientific discipline, SD:	Computer Engineering, Engineering of Computer Based Systems	
Subject/Key words, S/KW:	Second screen, UPnP protocol, Android	
UC		
Holding data, HD:	The Library of Faculty of Technical Sciences, Novi Sad, Serbia	
Note, N:		
Abstract, AB:	This paper briefly describes several communication protocols. UPnP protocol, which is chosen as base for this software, is described in more details. Paper also explains implementation of software that represents support for second screen.	
Accepted by the Scientific Board on, ASB:		
Defended on, DE:		
Defended Board, DB:	President:	Jelena Kovačević, PhD
	Member:	Milan Bjelica, PhD
	Member, Mentor:	Ištván Pap, PhD
		Menthor's sign

SADRŽAJ

1.	Uvod	1
2.	Teorijske osnove.....	3
2.1	Analiza raspoloživih komunikacionih protokola	3
2.1.1	UPnP.....	4
2.1.2	DLNA.....	4
2.1.3	DIAL.....	5
2.1.4	Izabrani protokol	5
2.2	UPnP – detalji protokola.....	6
2.2.1	Arhitektura UPnP protokola.....	6
2.2.2	Komponente UPnP mreže	7
2.2.2.1	Uredaj kojim se upravlja	8
2.2.2.2	Usluge koje pruža uređaj.....	8
2.2.2.3	Kontrolna tačka	9
2.2.3	Koraci UPnP protokola.....	9
2.2.3.1	Postupak pribavljanja adrese	9
2.2.3.2	Otkrivanje uređaja	9
2.2.3.3	Opisivanje uređaja.....	10
2.2.3.4	Upravljanje uređajem	10
2.2.3.5	Objavljivanje događaja.....	10
2.2.3.6	Predstavljanje uređaja	10
3.	Koncept rešenja.....	11
3.1	Moduli programske podrške	12
4.	Programsko rešenje.....	13
4.1	Sprega ka korisniku.....	13
4.1.1	Stvaranje instance kontrolne tačke i njena inicijalizacija	14
4.1.2	Deinicijalizacija i uništavanje date instance kontrolne tačke	14
4.1.3	Stvaranje instance upravljanog uređaja i njena inicijalizacija	14
4.1.4	Deinicijalizacija i uništavanje date instance upravljanog uređaja	15

4.1.5	Prihvati liste upravljenih uređaja	15
4.1.6	Oslabađanje liste upravljenih uređaja	15
4.1.7	Izazivanje akcije na upravljanom uređaju	16
4.2	Upravljeni uređaj	16
4.2.1	Inicijalizovanje tabele stanja uređaja.....	17
4.2.2	Povratna funkcija.....	18
4.2.3	Rukovanje zahtevom za izvršavanje akcije	18
4.2.4	Funkcija za prenos sadržaja.....	18
4.3	Kontrolna tačka	19
4.3.1	Povratna funkcija.....	19
4.3.2	Dodavanje novootkrivenog uređaja u listu.....	21
4.3.3	Prihvati informaciju o uređaju	21
4.3.4	Uklanjanje iz liste uređaja	21
4.3.5	Prozivanje akcije određenog uređaja.....	21
4.4	Parser XML dokumenata	22
4.5	Lista upravljenih uređaja.....	22
5.	Ispitivanje i rezultati	23
5.1	Ispitivanje programske podrške	23
5.2	Performanse programske podrške	24
6.	Zaključak	26
7.	Literatura	27

SPISAK SLIKA

Slika 1.1 Prenos sadržaja sa „drugog ekrana“ na centralni uređaj.....	2
Slika 2.1 UPnP stek protokola	6
Slika 2.2 UPnP uređaji, kontrolna tačka i usluge	7
Slika 2.3 Uredaj kojim se upravlja.....	8
Slika 2.4 Komunikacija kontrolne tačke sa uređajima	9
Slika 3.1 Slojevita struktura uređaja koji međusobno komuniciraju i položaj ciljne biblioteke u toj strukturi	11
Slika 4.1 Arhitektura programske podrške	13
Slika 4.2 Funtcionisanje uređaja	17
Slika 4.3 Funtcionisanje kontrolne tačke	20

SPISAK TABELA

Tabela 5.1 Testni slučajevi	24
Tabela 5.2 Memorija koju zauzima aplikacija pokrenuta u režimu uređaja na platformi	24
Tabela 5.3 Memorija koju zauzima aplikacija pokrenuta u režimu uređaja na platformi na kojoj su pokrenute aplikacije u oba režima	25
Tabela 5.4 Memorija koju zauzima aplikacija pokrenuta u režimu kontrolne tačke na platformi na kojoj su pokrenute aplikacije u oba režima	25

SKRAĆENICE

UPnP	- <i>Universal Plug and Play</i> , Skup protokola za uspostavljanje funkcionalnih veza između multimedijalnih uređaja
IP	- <i>Internet Protocol</i> , Internet protokol
TCP	- <i>Transmission Control Protocol</i> , Protokol kontrole toka
UDP	- <i>User Datagram Protocol</i> , Protokol za razmenu datagrama
HTTP	- <i>HyperText Transfer Protocol</i> , Protokol za prenos hiper teksta
XML	- <i>Extensible Markup Language</i> , Proširivi meta jezik za označavanje
DLNA	- <i>Digital Living Network Alliance</i> , Protokol za povezivanje uređaja u kućnoj mreži
DIAL	- <i>Discovery And Launch</i> , Protokol koji omogućava pratećim uređajima da otkriju i pokrenu aplikacije na centralnim uređajima
SSDP	- <i>Simple Service Discovery Protocol</i> , Protokol za otkrivanje usluga
GENA	- <i>General Event Notification Architecture</i> , Arhitektura za objavljivanje događaja
SOAP	- <i>Simple Object Access Protocol</i> , Protokol za pristup objektima
DDD	- <i>Device Description Document</i> , Dokument koji opisuje uređaj
DHCP	- <i>Dynamic Host Configuration Protocol</i> , Protokol za dinamičko konfigurisanje računara
UUID	- <i>Universally Unique Identifier</i> , Univerzalni jedinstveni identifikator
URL	- <i>Uniform Resource Locator</i> , Jednoobrazni lokator resursa
UDN	- <i>Unique Device Name</i> , Jedinstveno ime uređaja
API	- <i>Application Programming Interface</i> , Aplikativna programska sprega
SDK	- <i>Software Development Kit</i> , Skup alata za razvoj programske podrške

1. Uvod

U današnje vreme sve je više različitih uređaja sa ekranom (pametni telefoni, tablet računari, televizori). Bitni faktori koji su doveli do toga su nove tehnologije izrade ekrana, porast procesne moći, snažne prenosive baterije, kao i širok spektar veličina ekrana i formata slike itd. Ono što prati ekspanziju ekrana u našim svakodnevnim životima (kako brojno tako i po raznovrsnosti) jesu novi načini za njihovo međusobno povezivanje kao i naša sve veća sposobnost interakcije sa sadržajem i aplikacijama posredstvom ekrana.

Termin „drugi ekran“ (eng. second screen device, companion screen device) odnosi se na korišćenje mobilnih uređaja, kao što su tablet računari i pametni telefoni, s namenom da unaprede doživljaj gledanja televizije, što je obično omogućeno putem interaktivnih funkcija dostupnih u okviru posebne aplikacije. Pojava “drugog ekrana” dovila je do toga da televizor više nije u centru pažnje. Došlo je do širenja navika korisnika sa gledanja same televizije na istovremeno praćenje više medijskih platformi (kombinacija televizije i mobilnih uređaja). Iako mnogi i dalje gledaju program korišćenjem isključivo standardnih TV prijemnika, korisnici koji povezuju gledanje televizije sa sadržajem na mobilnim uređajima i internetom predstavljaju važan i rastući segment populacije. Najnovija istraživanja su pokazala da u Americi čak 77% gledalaca televizije u isto vreme koristi neki drugi uređaj [1]. Drugi uređaji se najčešće koriste u svrhu traženja onoga što je viđeno na TV ekranu. Takođe je zabeležen porast statusa i komentara na društvenim mrežama koji su u vezi sa trenutno gledanim emisijama na televiziji.

Zbog sve veće popularnosti ovih uređaja i njihovog korišćenja u kombinaciji sa TV prijemnikom, došlo je do potrebe za pronalaženjem što boljeg načina da se ovi uređaji međusobno sinhronizuju i povežu da bi se sadržaj između njih razmenjivao na što jednostavniji i neprimetniji (eng. seamless) način.

U ovom radu prikazana je jedna realizacija programske podrške za komunikaciju između centralnog uređaja i „drugog ekrana“ odnosno pratećeg uređaja. Programska podrška će se koristiti za prenos (eng. cast) posmatranog sadržaja sa pratećeg na centralni uređaj (Slika 1.1). Ciljna platforma za koju je razvijana ova programska podrška je Android.



Slika 1.1 Prenos sadržaja sa „drugog ekrana“ na centralni uređaj

Rad se sastoji od sedam poglavlja.

U drugom poglavlju je izloženo nekoliko raspoloživih komunikacionih protokola i njihova uporedna analiza. U okviru istog poglavlja je detaljno opisan komunikacioni protokol koji je izabran i na kome je zasnovano ovo rešenje.

U trećem poglavlju izloženi su zahtevi koje je bilo neophodno ispuniti prilikom realizacije rešenja kao i koncept samog rešenja.

U četvrtom poglavlju je opisano programsko rešenje sa detaljima realizacije modula (sa odgovarajućom programskom spregom).

Testiranje i rezultati rada su opisani u petom poglavlju. U šestom poglavlju je dat kratak osvrt na rad sa predlozima i mogućnostima za dalji razvoj i unapređivanje rešenja. Sedmo poglavlje predstavlja spisak literature korišćene prilikom izrade ovog rada.

2. Teorijske osnove

Pod komunikacionim protokolom se podrazumeva skup dogovorom utvrđenih pravila po kojima se obavlja komunikacija porukama (tekst, grafika, govor, audio, video i drugi podaci) između određenih entiteta različitih sistema u okviru telekomunikacionih mreža. U telekomunikacionim mrežama (digitalne telefonske, multiservisne, računarske, Internet itd.) primenjuju se različite vrste protokola.

Usluge na više ekrana (eng. multi-screen services) predstavljaju interaktivne usluge koje uključuju specifično realizovane aplikacije na različitim uređajima sa ekranom (eng. display device). Uređaji koji koriste usluge na više ekrana razvrstavaju se u dve grupe: centralni uređaji i „drugi ekran“ ili prateći uređaji.

Centralni uređaj (eng. main screen device) obično predstavlja TV uređaj ili prijemnik digitalnog TV signala (eng. set-top box) kojim upravlja prateći uređaj, mada bilo koji uređaj sa ekranom može biti u ulozi centralnog uređaja, u zavisnosti od scenarija korišćenja.

„Drugi ekran“ ili prateći uređaj je obično mobilni uređaj poput pametnog telefona ili tablet računara koji upravlja centralnim uređajem, iako u zavisnosti od scenarija korišćenja u ovoj ulozi mogu da se nađu i TV uređaj i prijemnik digitalnog TV signala.

2.1 Analiza raspoloživih komunikacionih protokola

Broj rešenja za podršku „drugog ekrana“ raste. Ipak, većina njih su vlasnička rešenja vezana za određene proizvođače. Najpoznatiji protokoli za komunikaciju i sinhronizaciju korisničkih uređaja su UPnP, DLNA i DIAL protokol.

2.1.1 UPnP

UPnP (eng. Universal Plug and Play) je skup mrežnih protokola koji omogućavaju automatsko pronalaženje, priključivanje i korišćenje uređaja koji su priključeni na računarsku mrežu [2]. Uređaji mogu biti računari, štampači, pretvarači protokola, pristupne tačke bežične mreže, mobilni uređaji itd. UPnP omogućava lako međusobno otkrivanje uređaja na mreži i uspostavljanje funkcionalnog mrežnog servisa za razmenu podataka, komunikaciju, i zabavu. Namjenjen je prvenstveno za kućne mreže bez uređaja poslovne klase. UPnP je konceptualni produžetak „plug-and-play“ protokola, tehnologije za dinamičko povezivanje uređaja direktno na računar, mada UPnP nije direktno srođan sa ranijom „plug-and-play“ tehnologijom. UPnP uređaji su „plug-and-play“ u smislu da kad su povezani na mrežu oni automatski uspostavljaju radne konfiguracije sa drugim uređajima. UPnP tehnologiju promoviše UPnP Forum. To je inicijativa računarske industrije namenjena omogućavanju jednostavne i robusne povezanosti samostalnih uređaja i ličnih računara različitih proizvođača.

UPnP tehnologija je nezavisna od operativnog sistema i programskog jezika i izgrađena je na Internet baziranim tehnologijama kao što su IP, TCP, UDP, HTTP i XML. Samim tim što koristi standardne TCP/IP protokole, omogućeno joj je da se neprimetno uklopi u postojeće mreže. Zbog toga što je distribuirana, otvorena mrežna arhitektura definisana protokolima koje koristi, UPnP je nezavisan od operativnog sistema, programskog jezika ili fizičkog/prenosnog medijuma. Osnovni nedostatak UPnP tehnologije je nedostatak autentifikacije, odnosno svaki UPnP uređaj mora da realizuje sopstveni mehanizam autentifikacije.

2.1.2 DLNA

Sa velikim brojem protokola i tehnologija na raspolaganju, najveća prepreka razmeni sadržaja između uređaja je njihova međusobna nekompatibilnost. DLNA (Digital Living Network Alliance) organizacija je osnovana od strane kompanija *Sony* i *Intel* (a u koju je sad uključen veliki broj firmi iz oblasti informacionih tehnologija i mobilne telefonije) s ciljem da omoguće međusobnu kompatibilnost njihovih proizvoda. DLNA standard obuhvata skup pravila koja omogućavaju različitim uređajima (pametni telefoni, tableti, računari, televizori itd.) da pristupe sadržajima koji su skladišteni na raznim mestima unutar lokalne mreže i da ih prikazuju. Po ovom standardu, povezivanje i međusobna komunikacija su mogući isključivo između DLNA sertifikovanih uređaja. Dakle, osnovni preduslovi da bi uređaji mogli međusobno da komuniciraju i razmenjuju sadržaje su da moraju biti DLNA sertifikovani i da moraju da se nalaze unutar iste mreže [3].

DLNA u osnovi koristi UPnP protokol. UPnP je taj koji definiše koji tip uređaja podržava DLNA i mehanizme za pristup medijima preko mreže. DLNA razlikuje uređaje za deljenje, kontrolu i reprodukciju multimedijalnog sadržaja, kako bi drugi uređaji koji su uključeni u mrežu mogli da zatraže odgovarajuću uslugu. Takođe, definiše ograničenja nad vrstama formata medijske datoteke, kodiranja i rezolucijama koje uređaj mora da podržava. Drugi standardi koje koristi su HTTP, IPv4, Ethernet, Wi-Fi itd. Ono što razlikuje DLNA od UPnP standarda je što DLNA mora da podržava određene audio/video standarde da bi uređaji mogli da reprodukuju sadržaj koji im je poslat. DLNA je donekle ograničen time što ne podržava mnogo codec-a i radi sa sadržajem koji nije zaštićen.

2.1.3 DIAL

DIAL (DIIscovery And Launch) je otvoreni standard iniciran od strane *Netflix*-a i *YouTube*-a (jedni od najvećih dobavljača online videa) s ciljem da olakša reprodukciju online sadržaja na TV uređaju. DIAL je jednostavan protokol koji omogućava “drugim ekranima” da otkriju centralne uređaje u mreži i obezbede pristup njihovim uslugama koje omogućavaju upravljanje aplikacijama na centralnom uređaju [4]. Prednost DIAL-a je u tome što eliminiše potrebu za posebnim (ručnim) pokretanjem aplikacije na centralnom uređaju da bi njom moglo da se upravlja sa pratećeg uređaja. Kao osnova DIAL protokola koriste se UPnP protokol za otkrivanje uređaja (tačnije koristi se SSDP definisan u okviru UPnP) i HTTP za pristup uslugama. Ono što je karakteristično za DIAL je da imena svih aplikacija koje koriste DIAL protokol moraju biti navedena u DIAL registru. Takođe, sadržaj na nekom uređaju se može pustiti samo u okviru aplikacije koja je registrovana za puštanje sadržaja tog tipa.

Osnovna prednost DIAL-a u odnosu na DLNA je što omogućava da, umesto samog slanja toka podataka sa uređaja na TV, uređaj preko DIAL-a zahteva od TV ili prijemnika digitalnog TV signala da pokrene odgovarajuću aplikaciju i reprodukuje sadržaj direktno sa izvora, izbegavajući na taj način potrebu da se isti sadržaj istovremeno prikazuje na oba uređaja tj. omogućavajući prikaz različitih ali koordinisanih sadržaja.

2.1.4 Izabrani protokol

Za protokol na kojem će se zasnovati realizacija programske podrške za komunikaciju između centralnog uređaja i “drugog ekrana” izabran je UPnP. UPnP ne specificira programsku spregu koju će aplikacije koristiti, omogućavajući proizvođačima da kreiraju programsku spregu koja zadovoljava njihove potrebe, što je velika prednost nad ostalim protokolima i jedan od razloga zbog kog je izabran. O tome koliko je UPnP zastupljen protokol u prilog govori i činjenica da su i DLNA i DIAL zasnovani upravo na njemu. S obzirom na potrebe zadatka sama UPnP biblioteka će biti dovoljna za realizaciju programskog

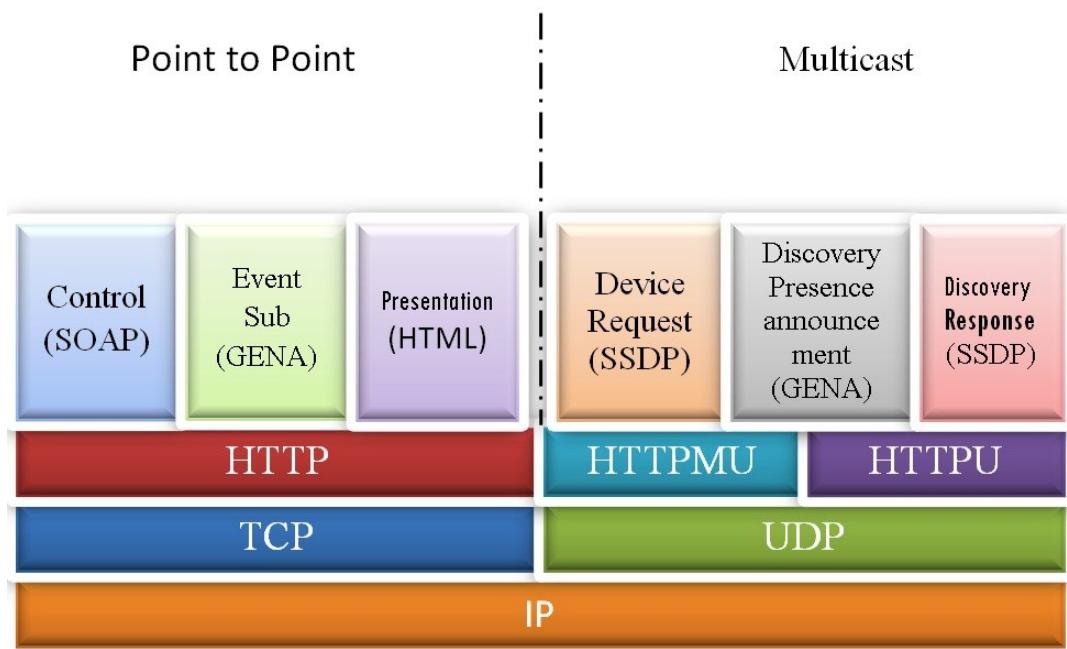
rešenja (naprednije funkcionalnosti koje pružaju DLNA i DIAL nisu neophodne za ovu realizaciju). Nedostatak DLNA protokola je taj što preko njega mogu komunicirati samo DLNA sertifikovani uređaji. S druge strane, DIAL zahteva da aplikacija bude navedena u njihovom registru i omogućava samo prozivanje odgovarajuće aplikacije na centralnom uređaju, ali ne i da se, na primer, prozove neka funkcija u okviru postojeće TV aplikacije. Iz tih razloga je za osnovu ove programske podrške izabran UPnP protokol.

2.2 UPnP – detalji protokola

Koristeći UPnP, uređaj može da se dinamički uključi u mrežu, prenese (objavi) svoje sposobnosti i da otkrije druge uređaje u mreži i njihove sposobnosti. Uređaji koji međusobno komuniciraju preko UPnP protokola mogu da se ponašaju kao da su u mreži „od učesnika do učesnika“ (eng peer to peer).

2.2.1 Arhitektura UPnP protokola

Komunikacija između UPnP uređaja zasnovana je na više postojećih komunikacionih protokola. Na slici 2.1 predstavljen je UPnP stek protokola.



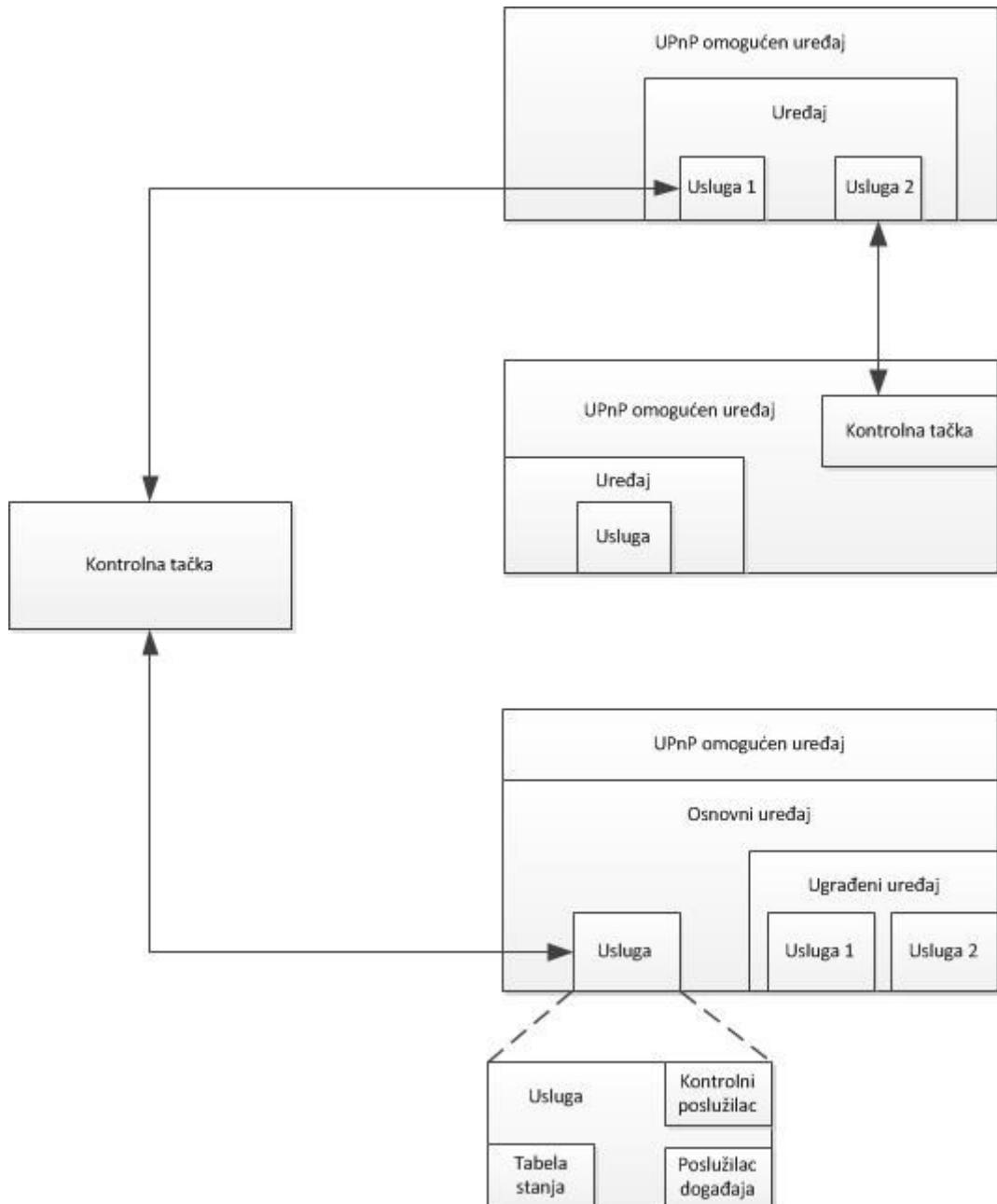
Slika 2.1 UPnP stek protokola

SSDP (eng. Simple Service Discovery Protocol) je protokol koji UPnP koristi za otkrivanje i pretragu uređaja na mreži. GENA (eng. General Event Notification Architecture) protokol se koristi za objavljivanje događaja i za „preplaćivanje“ na praćenje događaja. SOAP (eng. Simple Object Access Protocol) se koristi za upravljanje uređajem, tj. slanje

zahteva za izvršavanje akcije. Ovi protokoli se oslanjaju na HTTP (eng. Hypertext Transfer Protocol) i njegove verzije HTTPMU (HTTP over multicast UDP) i HTTPU (HTTP over unicast UDP). U sloju ispod nalaze se TCP i UDP. Na kraju sve poruke se prenose korišćenjem IP protokola [5].

2.2.2 Komponente UPnP mreže

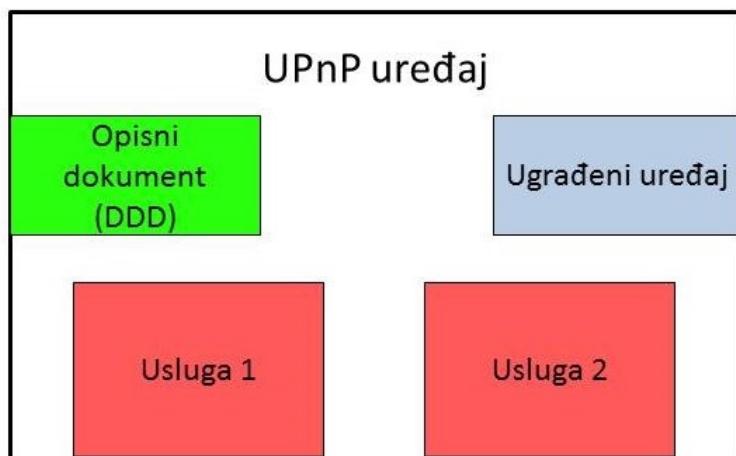
Osnovne komponente UPnP su uređaji i usluge (eng. service). Uređaji su podeljeni u dve kategorije: uređaji kojim se upravlja (eng. devices) i kontrolne tačke (eng. control points)) (Slika 2.2).



Slika 2.2 UPnP uređaji, kontrolna tačka i usluge

2.2.2.1 Uređaj kojim se upravlja

Uređaj kojim se upravlja obuhvata proizvoljan broj usluga (eng. services), a može da obuhvata i druge upravljane uređaje koji su ugrađeni u njega (eng. nested devices) (na primer: mobilni telefon je upravljeni uređaj, a kamera je upravljeni uređaj u okviru njega) (Slika 2.3). Različite usluge su vezane za različite kategorije uređaja (na primer: usluge koje pruža video-rekorder se razlikuju od onih koje pruža štampač). Upravljeni uređaj sadrži i XML dokument koji ga opisuje (eng. Device Description Document - DDD). U njemu su sadržane sve neophodne informacije o uređaju kao što su ime uređaja i lista usluga koje pruža. Dokument je stavljen na raspolaganje drugim uređajima u mreži (kontrolnim tačkama) kako bi mogli da se informišu o njegovim osobinama.



Slika 2.3 Uređaj kojim se upravlja

2.2.2.2 Usluge koje pruža uređaj

Usluga ili servis je najmanja jedinica kojom se može upravljati u UPnP mreži i predstavlja skup akcija koje pruža dati upravljeni uređaj. Informacije o uslugama su takođe uključene u XML dokument koji opisuje uređaj. Uređaji mogu imati više usluga. Svaku uslugu karakterišu stanje i njen skup akcija. Ako posmatramo primer sata, akcije bi bile „podesi tačno vreme“ i „očitaj vreme“, dok bi promenljiva „vreme“ predstavljala njegovo stanje.

Usluge se sastoje od tabele stanja (eng. state table), kontrolnog poslužioca (eng. control server) i poslužioca događaja (eng. event server) (Slika 2.2). Trenutno stanje usluge se beleži kao vrednost u tabeli stanja. Prilikom promene stanja, tabela stanja se ažurira.

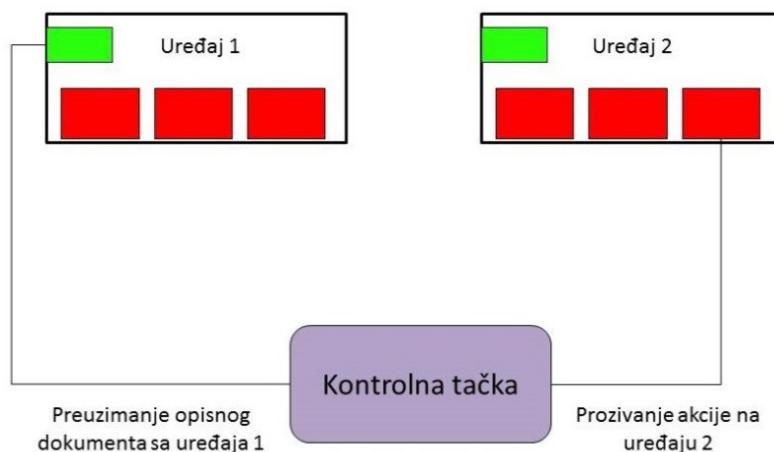
Kontrolni poslužilac prima zahteve za akcijama, izvršava ih, ažurira tabelu stanja i vraća odgovor ukoliko postoji. Poslužilac događaja je zadužen za obaveštavanje o izmeni stanja date usluge one uređaje koji su zainteresovani da primaju takve informacije (one koji su se „preplatili“ na praćenje tih događaja).

2.2.2.3 Kontrolna tačka

Kontrolna tačka je zadužena za otkrivanje drugih uređaja i upravljanje njima. Nakon otkrivanja uređaja, kontrolna tačka može da:

- dobavi opis uređaja sa listom pripadajućih usluga
- dobavi opise odgovarajućih usluga
- pošalje zahtev za izvršavanje akcije
- pretplati se na praćenje događaja (promena stanja) na uređaju.

Poželjno je da svaki uređaj u sebi sadrži kontrolnu tačku i obrnuto, da bi se omogućilo umrežavanje “od korisnika do korisnika”.



Slika 2.4 Komunikacija kontrolne tačke sa uređajima

2.2.3 Koraci UPnP protokola

2.2.3.1 Postupak pribavljanja adrese

Uredaj prvo pokušava da pribavi adresu od DHCP poslužioca (Dynamic Host Configuration Protocol). Ukoliko je poslužilac dostupan, uređaj će koristiti IP adresu koju mu je dodelio, u suprotnom adresu će dobiti autokonfiguracijom u opsegu od 169.254.1.0 do 169.254.254.255 (auto IP). Uredaj koji je autokonfiguracijom dobio IP adresu i dalje periodično proverava dostupnost DHCP poslužioca, i ako isti postane dostupan pribavlja novu adresu korišćenjem DHCP.

2.2.3.2 Otkrivanje uređaja

Za otkrivanje uređaja koristi se SSDP protokol. On omogućuje upravljanom uređaju da, kada se uključuje u mrežu, objavi svoje prisustvo kontrolnim tačkama. Sa druge strane omogućuje kontrolnoj tački koja se uključuje u mrežu da pretražuje mrežu i pronađe uređaje

od interesa. U oba slučaja vrši se razmena poruka za otkrivanje koje nose tip uređaja, univerzalni jednoznačni identifikator, URL stranice na kojoj se nalaze detaljnije informacije (XML dokument), kao i opcione podatke o stanju uređaja.

2.2.3.3 Opisivanje uređaja

Nakon što je kontrolna tačka otkrila upravljeni uređaj, i dalje vrlo malo zna o njemu (UPnP tip uređaja, identifikator i URL opisa uređaja). Da bi saznala više o uređaju i da bi mogla da upravlja uređajem, kontrolna tačka mora da preuzme opis uređaja sa URL-a dobijenog tokom otkrivanja uređaja. XML dokument koji predstavlja opis samog uređaja (DDD) sadrži ime modela, serijski broj, ime proizvođača itd. Takođe sadrži listu usluga dostupnih kontrolnim tačkama kao i listu ugrađenih uređaja ukoliko postoje.

2.2.3.4 Upravljanje uređajem

U okviru dokumenta koji opisuje upravljeni uređaj nalazi se i lista komandi ili akcija na koje će data usluga reagovati kao i argumente neophodne za odgovarajuću akciju. Takođe sadrži i listu promenljivih koje modeluju stanje usluga u datom trenutku.

Da bi upravljala uređajem, kontrolna tačka šalje zahtev za izvršavanje akcije usluzi odgovarajućeg uređaja. Ovo radi slanjem odgovarajuće kontrolne poruke (koja je takođe opisane XML struktrom) na URL date usluge.

2.2.3.5 Objavljivanje događaja

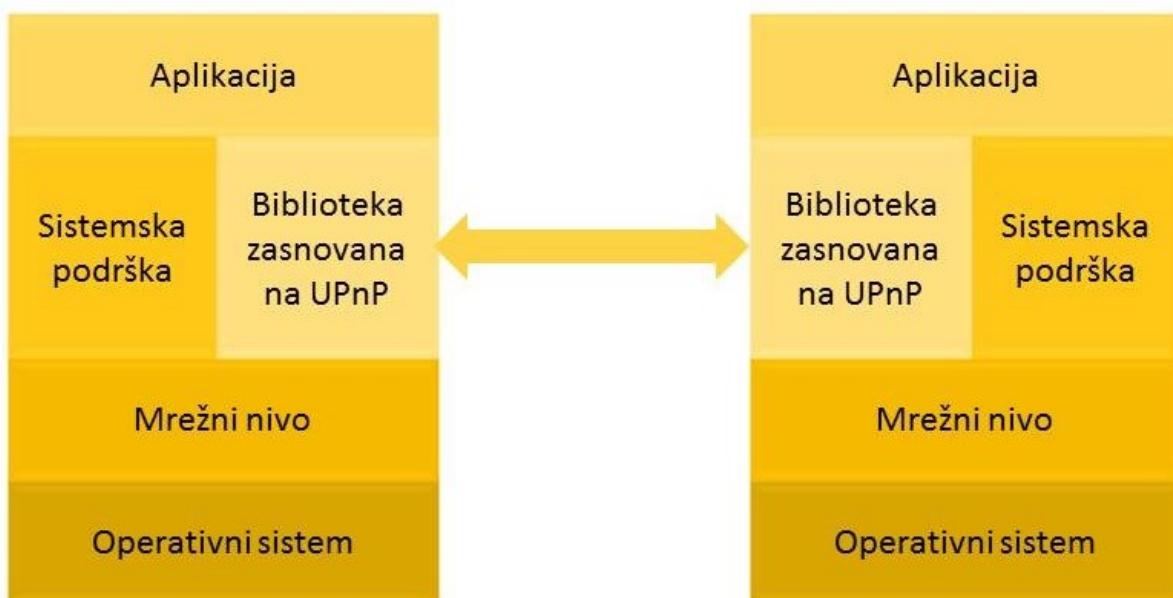
Kada se promenljive koje modeluju stanje usluge ažuriraju, usluga to objavljuje, a kontrolna tačka može da se pretplati kako bi dobijala ta obaveštenja o promeni stanja usluge. Usluga objavljuje ažuriranja slanjem posebnih poruka koje sadrže imena i trenutne vrednosti promenljivih. Ove poruke se šalju kontrolnoj tački nakon što se ona pretplati. Šalju se svim kontrolnim tačkama koje su pretplaćene svaki put kada se desi neka promena stanja bez obzira na to koji je njen uzrok.

2.2.3.6 Predstavljanje uređaja

Ukoliko uređaj poseduje prezentacioni URL, kontrolna tačka može da pribavi stranicu sa datog URL-a, otvoriti je u pretraživaču i omogući kontrolnoj tački da upravlja uređajem i/ili vidi stanje uređaja. Šta će od ovog biti realizovano, zavisi od specifičnih mogućnosti prezentacione stranice i samog uređaja.

3. Koncept rešenja

Programska podrška koju je neophodno realizovati zasnivaće se na postojećoj UPnP biblioteci (*libUPnP 1.6.12*) [6]. Rezultat će biti dinamička biblioteka sa programskom spregom (eng. API) koja će omogućiti korišćenje osnovnih funkcionalnosti vezanih za prenos sadržaja sa jednog uređaja na drugi (Slika 3.1).



Slika 3.1 Slojevita struktura uređaja koji međusobno komuniciraju i položaj ciljne biblioteke u toj strukturi

Osnovni zahtev koji treba ispuniti prilikom realizacije programske podrške je da se omogući njeno izvršavanje na Android platformi. Takođe je neophodno obezbediti multiklijentsku podršku tj. omogućiti da se ista biblioteka može istovremeno koristiti od strane više klijentskih aplikacija.

3.1 Moduli programske podrške

Programska podrška se sastoji od više modula. Modularizacija je izvršena tako da rešenje bude logički korektno, u skladu sa načinom funkcionisanja UPnP-a, i da omogući jednostavnu nadogradnju tj. dodavanje novih funkcionalnosti.

S obzirom na to da svaka kontrolna tačka vodi evidenciju o upravljanim uređajima koji su od interesa za nju, jedan od modula ove programske podrške predstavlja listu upravljenih uređaja. U ovom modulu su realizovane osnovne funkcionalnosti jedne liste (dodavanje u listu, brisanje iz liste, očitavanje vrednosti elementa sa date pozicije itd.). Prilikom njegove realizacije korišćen je podmodul koji predstavlja generičku jednostruku spregnutu listu. Kako je jedan od zahteva bio da se obezbedi multiklijentska podrška, lista uređaja je realizovana tako da je onemogućeno da joj istovremeno pristupa više programskih niti.

Tokom otkrivanja uređaja u mreži kojima može da upravlja, kontrolna tačka pribavlja njihove opise koji su dati u obliku XML dokumenta prema standardnom *UPnP Device Template* šablonu. Da bi iz tog dokumenta kontrolna tačka zapravo i dobila neke informacije, neophodno je obaviti parsiranje tog dokumenta. Stoga je realizovan poseban modul koji se time bavi.

Jedan modul predstavlja apstrakciju upravljanog uređaja. Ovaj modul realizuje osnovne funkcionalnosti koje upravljeni uređaj treba da obezbedi kao što su izvršavanje akcije prozvane od strane kontrolne tačke ili slanje obaveštenja o promeni stanja. Programska sprega ka spoljnim modulima omogućava korišćenje ovih funkcionalnosti.

Postoji i modul koji predstavlja apstrakciju kontrolne tačke, sa realizovanim funkcionalnostima koje obuhvataju otkrivanje uređaja od interesa (uređaja kojima može da upravlja) i slanje zahteva za izvršavanjem akcija na tim uređajima. Ovaj modul takođe obuhvata i rukovanje listom dostupnih uređaja od interesa, za šta se koristi gore pomenuti modul koji predstavlja listu upravljenih uređaja. Pružena programska sprega omogućava korišćenje tih funkcionalnosti.

Poslednji modul objedinjuje funkcionalnosti kontrolne tačke i upravljanog uređaja i pruža jedinstvenu programsku spregu koja omogućava aplikaciji da kreiranu biblioteku koristi u jednom od ova dva režima ili u oba režima istovremeno.

4. Programsko rešenje

Programsko rešenje je razvijano na PC računaru i pisano u programskom jeziku C. Za realizaciju ovog programskog rešenja korišćena je UPnP biblioteka (*libUPnP-1.6.12*) koja omogućava komunikaciju između više različitih uređaja. Programska podrška se sastoji od više modula. Njihova zavisnost je prikazana na slici 4.1.



Slika 4.1 Arhitektura programske podrške

4.1 Sprega ka korisniku

Ono što ovo programsko rešenje omogućuje korisniku je mogućnost korišćenja u jednom od dva režima (kao kontrolna tačka ili kao upravljeni uređaj) ili u oba režima istovremeno. Rad u režimu kontrolne tačke pruža mogućnost iniciranja akcije određene

usluge na nekom od dostupnih upravljanih uređaja koji tu akciju podržavaju i mogu da je izvrše. S druge strane, rad u režimu upravljanog uređaja omogućava prijem upravljačke komande, izvršavanje odgovarajuće akcije na osnovu nje i slanje informacije o uspešnosti izvršavanja date akcije kontrolnoj tački koja je to izvršavanje inicirala.

4.1.1 Stvaranje instance kontrolne tačke i njena inicijalizacija

```
COM_ErrorCode CP_Create(ControlPoint* ctrlPoint, cp_notify notifyFunc);
```

Parametri:

- `ctrlPoint` – instanca kontrolne tačke koja se inicijalizuje
- `notifyFunc` – funkcija koju korisnik registruje da se pozove iz povratne funkcije kontrolne tačke

Pre kreiranja i inicijalizacije kontrolne tačke neophodno je obaviti inicijalizaciju Linux SDK za UPnP uređaje pozivom odgovarajuće funkcije UPnP biblioteke [7]. Nakon toga se inicijalizuju sva polja *ControlPoint* instance. Aplikacija koja koristi datu instancu se preko nje, a putem funkcije UPnP sprege, registruje na SDK kako bi mogla koristiti ostale funkcionalnosti UPnP biblioteke. Posle uspešnog registrovanja, započinje se pretraga za upravljanim uređajima koji su tipa definisanog u okviru *ControlPoint* instance. O rezultatima pretrage kontrolna tačka biva obaveštена preko povratne funkcije koja je prosleđena kao parametar prilikom registrovanja i koja će biti pozivana od strane UPnP biblioteke. Na kraju je potrebno inicijalizovati i pokrenuti merač vremena (eng. timer) koji će svaki put po isteku zadatog intervala u odvojenoj programsкоj niti pokrenuti izvršavanje funkcije koja proverava da li je isteklo vreme objavljivanja dostupnih uređaja.

4.1.2 Deinicijalizacija i uništavanje date instance kontrolne tačke

```
COM_ErrorCode CP_Destroy(ControlPoint ctrlPoint);
```

Parametri:

- `ctrlPoint` – instanca kontrolne tačke koja se deinicijalizuje

Omogućava pozivanje svih funkcija UPnP sprege neophodnih za završavanje rada sa bibliotekom, oslobađanje prostora alociranog za čuvanje liste uređaja i zaustavljanje odgovarajućeg merača vremena.

4.1.3 Stvaranje instance upravljanog uređaja i njena inicijalizacija

```
COM_ErrorCode DEV_Create(Device* device, dev_notify notifyFunc);
```

Parametri:

- `device` – instanca uređaja koja se inicijalizuje
- `notifyFunc` – funkcija koju korisnik registruje da se pozove iz povratne funkcije uređaja

Slično kao i u slučaju kontrolne tačke, obavi se inicijalizacija Linux SDK za UPnP uređaje, inicijalizuju se polja odgovarajuće *Device* instance i aplikacija se, preko nje, registruje na korišćenje SDK-a. Ono što je karakteristično za upravljeni uređaj je da je potrebno postaviti putanju za mali HTTP poslužilac koji se koristi za isporučivanje dokumenta koji opisuje uređaj kontrolnim tačkama koje ga zatraže. Potrebno je i inicijalizovati tabelu stanja. Na kraju se započinje objavljivanje prisustva uređaja svim kontrolnim tačkama u mreži.

4.1.4 Deinicijalizacija i uništavanje date instance upravljanog uređaja

```
COM_ErrorCode DEV_Destroy(Device* device);
```

Parametri:

- `device` – instanca uređaja koja se deinicijalizuje

Omogućava pozivanje svih funkcija UPnP sprege neophodnih za završavanje rada sa bibliotekom.

4.1.5 Prihvatanje liste upravljenih uređaja

```
COM_ErrorCode
```

```
CP_GetDeviceList(ControlPoint ctrlPoint, DevArray* devList, int* size);
```

Parametri:

- `ctrlPoint` – instanca uređaja čija lista uređaja se dobavlja
- `devList` – lista uređaja
- `size` – broj elemenata u listi

Ova funkcija omogućava klijentu da dobavi listu uređaja koji su dostupni datoj kontrolnoj tački. U okviru te liste klijent dobija imena dostupnih uređaja i UDN (jedinstveni identifikator). Na osnovu toga može da izabere na kom od ponuđenih uređaja će zatražiti izvršavanje akcije.

4.1.6 Oslobođanje liste upravljenih uređaja

```
COM_ErrorCode CP_FreeDeviceList(DevArray* devList, uint32_t size);
```

Parametri:

- `devList` – lista uređaja koja se uništava
- `size` – broj elemenata u listi

Ova funkcija omogućava korisniku da uništi listu uređaja kada više nema potrebu za njenim korišćenjem i samim tim oslobodi memorijski prostor koji je ta lista zauzimala.

4.1.7 Izazivanje akcije na upravljanom uređaju

```
COM_ErrorCode CP_Cast(ControlPoint ctrlPoint, char* UDN, char** message);
```

Parametri:

- `ctrlPoint` – instanca uređaja koji izaziva akciju
- `UDN` – UDN uređaja (jedinstveno ime)
- `message` – niz podataka koji se šalje

Pozivanjem ove funkcije data kontrolna tačka šalje zahtev za izvršavanjem akcije na uređaju sa datim UDN.

4.2 Upravljeni uređaj

Upravljeni uređaj je predstavljen strukturu *Device*. Ova struktura, u okviru svojih polja, nosi informacije koje definišu dati upravljeni uređaj. U ove informacije spadaju:

- IP adresa i port preko kog dati uređaj komunicira sa ostalim uređajima u mreži, pribavlja se preko odgovarajuće funkcije UPnP programske sprege
- identifikator uređaja
- podržane tipove usluga
- tabelu usluga.

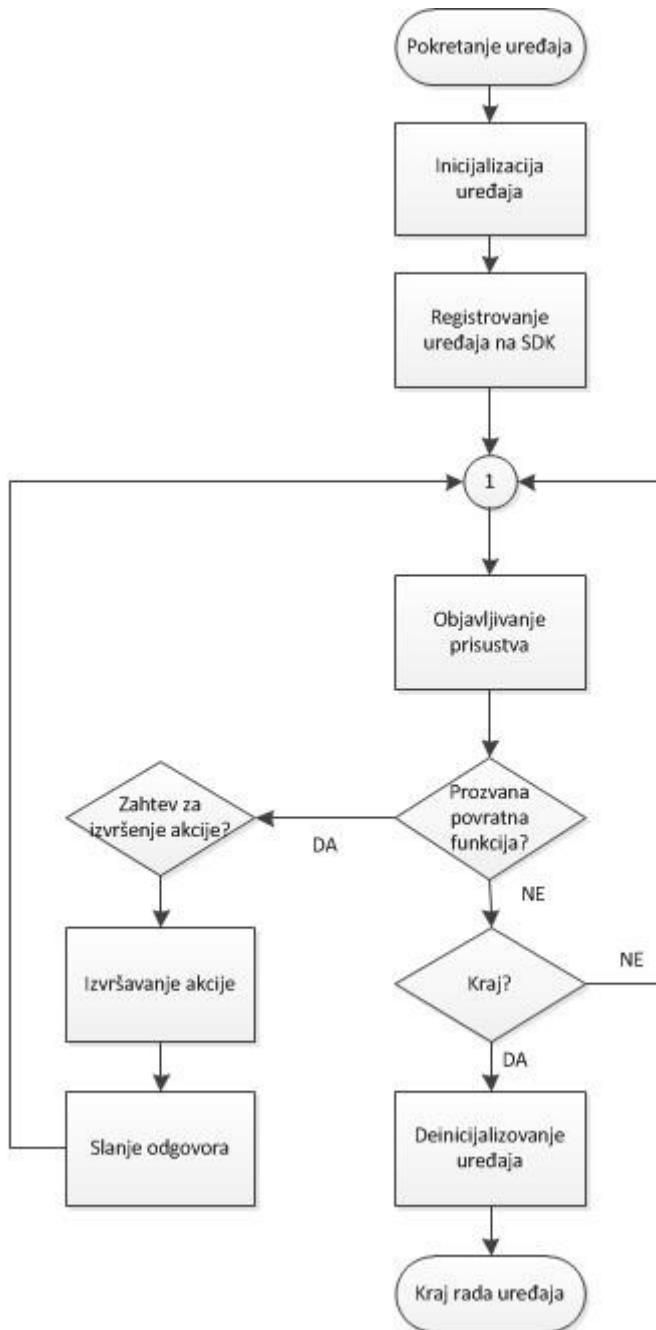
Uz to, u strukturi se nalaze i podaci neophodni za realizaciju određenih funkcionalnosti uređaja, kao što su:

- muteks
- vreme isteka objave uređaja
- URL do dokumenta koji opisuje uređaj
- pokazivač na funkciju koju je klijent registrovao da se poziva prilikom izvršavanja povratne funkcije

Usluge su predstavljene strukturu *Service* koja sadrži neophodne informacije za rukovanje uslugama, a koje su izdvojene iz dokumenta koji opisuje uređaj:

- identifikator usluge
- UDN – identifikator uređaja
- tip usluge
- nazivi akcija
- pokazivači na funkcije koje predstavljaju akcije
- imena i vrednosti promenljivih koje oslikavaju trenutno stanje usluge

Na slici 4.2 prikazan je algoritam kojim je opisan rad uređaja.



Slika 4.2 Funkcionisanje uređaja

4.2.1 Inicijalizovanje tabele stanja uređaja

Tabela stanja je predstavljena kao niz *Service* elemenata (jer se u okviru *Service* strukture nalaze imena i vrednosti promenljivih koje oslikavaju trenutno stanje usluge). Njena inicijalizacija podrazumeva popunjavanje polja *Service* struktura odgovarajućim vrednostima koje se iščitaju iz dokumenta koji opisuje uređaj (DDD) i postavljanjem promenljivih na inicijalne vrednosti. Stoga inicijalizovanje tabele stanja uređaja započinje parsiranjem dokumenta koji opisuje uređaj. Parsiranje se vrši pomoću funkcija posebnog modula za

parsiranje XML dokumenta. Nakon parsiranja, dobijeni podaci se koriste za popunjavanje tabele stanja tj. *Service* struktura.

4.2.2 Povratna funkcija

Povratnu funkciju proziva UPnP biblioteka i time signalizira uređaju kada se desi neki događaj od značaja. Prototip povratne funkcije je definisan u okviru *libUPnP* biblioteke. Kao parametre funkcija dobija tip događaja koji je izazvao njen poziv (EventType) i strukturu Event u okviru koje su joj prosleđene informacije od značaja. Tipovi događaja koji dovode do pozivanja povratne funkcije su:

- pristizanje zahteva za pretplatu na obaveštenja o događajima od strane kontrolne tačke
- pristizanje zahteva za izvršavanjem neke akcije
- pristizanje zahteva za slanje vrednosti određene promenljive stanja kontrolnoj tački

Po pristizanju tih zahteva kreće njihova obrada.

4.2.3 Rukovanje zahtevom za izvršavanje akcije

Kontrolna tačka može da pošalje zahtev upravljanom uređaju za izvršavanjem neke akcije. Odgovarajući uređaj primi zahtev i posredstvom UPnP biblioteke se pozove povratna funkcija. U okviru strukture koja se pošalje kao parametar te funkcije nalazi se:

- ime akcije koja treba da se izvrši
- pokazivač na XML dokument koji sadrži podatke prosleđene od stane kontrolne tačke, a koji je generisala UPnP biblioteka
- pokazivač na XML dokument u koji će se smestiti odgovor na datu akciju. Taj dokument generiše UPnP biblioteka pozivanjem odgovarajuće funkcije i potom se šalje kontrolnoj tački.

U tabeli usluga se na osnovu imena akcije pronađe pokazivač na funkciju koje predstavlja datu akciju, a iz ulaznog XML dokumenta se parsiranjem izdvoje podaci prosleđeni od strane kontrolne tačke. Nakon što je pronađena odgovarajuća funkcija i nakon što su izdvojeni neophodni podaci (argumenti date funkcije) može da se započne sa izvršavanjem akcije.

4.2.4 Funkcija za prenos sadržaja

Svaka od funkcija koja predstavlja neku akciju ima isti prototip koji je unapred definisan. Ove funkcije kao ulazne parametre imaju tip usluge i XML dokument koji sadrži

podatke poslate od strane kontrolne tačke. Kao izlazne parametre sadrži XML dokument koji sadrži odgovor koji se šalje kontrolnoj tački i poruku greške.

4.3 Kontrolna tačka

Kontrolna tačka je predstavljena struktrom *ControlPoint*. Polja ove strukture sadrže podatke koji karakterišu kontrolnu tačku ili su neophodni za realizaciju određenih funkcionalnosti kontrolne tačke:

- IP adresa i port preko kog data kontrolna tačka komunicira sa ostalim uređajima u mreži, pribavlja se preko odgovarajuće funkcije UPnP programske sprege
- identifikator kontrolne tačke
- tip upravljanog uređaja koji podržava
- lista dostupnih upravljenih uređaja
- pokazivač na funkciju koju je klijent registrovao da se poziva prilikom izvršavanja povratne funkcije

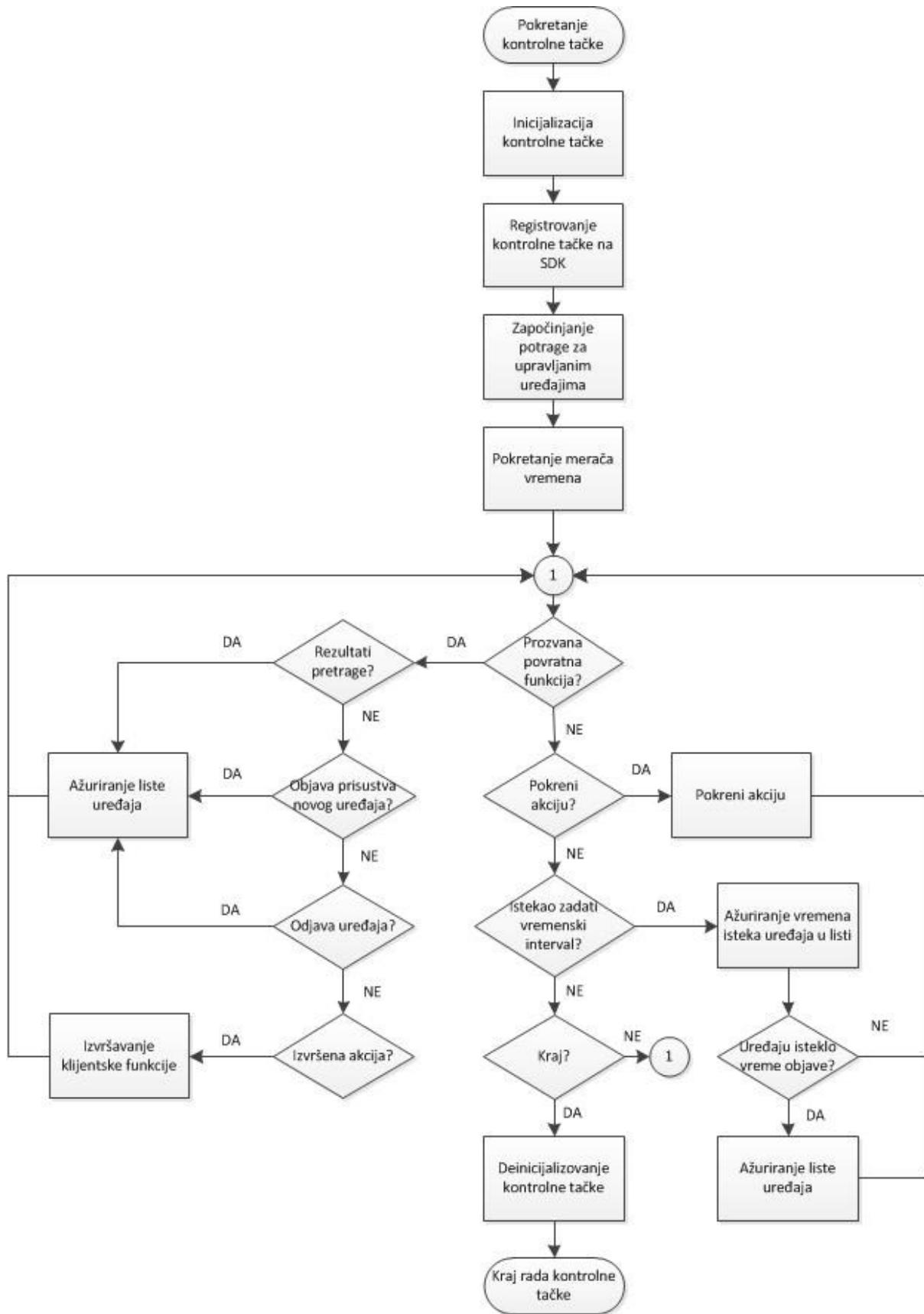
Za stvaranje i inicijalizaciju, kao i za deinicijalizaciju i uništavanje primerka kontrolne tačke, koristi se programska sprega modula koji obezbeđuje funkcionalnosti kontrolne tačke, u koje spadaju rukovanje listom uređaja od interesa i slanje zahteva za izvršavanje akcija. Na slici 4.3 prikazan je algoritam funkcionisanja kontrolne tačke.

4.3.1 Povratna funkcija

Slično kao i kod upravljanog uređaja, UPnP biblioteka prozivanjem povratne funkcije signalizira kontrolnoj tački kada se desi neki događaj od značaja. Prototip povratne funkcije je definisan UPnP standardom. Kao parametre funkcija dobija tip događaja koji je izazvao njen pozivanje (*EventType*) i strukturu *Event* u okviru koje su joj prosleđene informacije od značaja. Tipovi događaja koji dovode do pozivanja povratne funkcije su:

- signaliziranje kontrolnoj tački o tome da je novi uređaj u mreži i da je objavio svoje prisustvo.
- obaveštenje o uspešnosti pretrage koju je inicirala kontrolna tačka i o pronađenim uređajima
- obaveštenje kada upravljeni uređaj odjavi svoje prisustvo
- obaveštenje o isteku vremena pretrage za upravljenim uređajima
- obaveštenje o uspešnosti izvršavanja akcije koju je kontrolna tačka inicirala na nekom od upravljenih uređaja

Po pristizanju tih zahteva kreće njihiva obrada.



Slika 4.3 Funkcionisanje kontrolne tačke

4.3.2 Dodavanje novootkrivenog uređaja u listu

Nakon što je biblioteka signalizirala kontrolnoj tački da se novi uređaj uključio u mrežu i da je objavio svoje prisustvo, kontrolna tačka kreće u pribavljanje dodatnih informacija o tom uređaju, preuzimanjem dokumenta koji je opisuje (DDD). Dokument je potrebno parsirati. Za parsiranje se koriste funkcije modula koji je realizovan za tu namenu. Na osnovu informacija dobijenih parsiranjem, kontrolna tačka proverava da li je uređaj onog tipa koji ona podržava. Ukoliko jeste, taj uređaj se dodaje u listu uređaja. Za dodavanje uređaja u listu koristi se odgovarajuća funkcija modula koji realizuje operacije nad listom uređaja.

4.3.3 Prihvata informacija o uređaju

Informacije o uređaju u listi je moguće dobaviti na osnovu pozicije u listi ili na osnovu UDN tj. jedinstvenog identifikatora uređaja. U oba slučaja je potrebno proslediti kao parametar funkciji adresu memorije u koju će da se kopira sadržaj koji predstavlja odgovarajući uređaj. Izvršava se pozivom odgovarajuće funkcije modula koji predstavlja listu uređaja.

4.3.4 Uklanjanje iz liste uređaja

Uklanjanje uređaja iz liste se vrši kada uređaj prestane da objavljuje svoje prisustvo tj. kada istekne vreme objavljivanja ili kada se eksplicitno odjavi. Uklanjanje svih uređaja iz liste vrši se pri završavanju sa radom kontrolne tačke. Uklanjanje vrši odgovarajuća funkcija modula koji predstavlja listu uređaja.

4.3.5 Prozivanje akcije određenog uređaja

Pre nego što se prozove neka akcija, potrebno je obezbediti neophodne informacije da bi se prozivanje izvršilo. Ove informacije se prosleđuju kao parametri funkcije. U njih spadaju:

- identifikator uređaja čija akcija se proziva (UDN)
- ime akcije
- broj parametara koji se prosleđuju akciji
- niz imena parametara
- niz njihovih vrednosti.

Prvo se na osnovu UDN iz liste uređaja pribave ostale informacije o datom uređaju. Od naročitog značaja je upravljački URL (eng. Control URL) na osnovu kojeg UPnP biblioteka zna gde da uputi zahtev. Zahtev za akcijom se šalje pozivanjem odgovarajuće funkcije UPnP biblioteke. O uspešnosti izvršavanja akcije kontrolna tačka bude obaveštена putem povratne funkcije.

4.4 Parser XML dokumenata

Ovaj pomoćni modul se koristi od strane kontrolne tačke kada joj je potrebno da izvuče informacije o nekom upravljanom uređaju iz XML dokumenta koji ga opisuje. Sadrži funkcije:

- za izdvajanje elemenata na osnovu njihovih oznaka
- za izdvajanje liste usluga iz XML dokumenta koji opisuje uređaj
- za izdvajanje svih informacija vezanih za konkretnu uslugu.

Oslanja se na programsku spregu *ixml* modula u okviru UPnP biblioteke.

4.5 Lista upravljenih uređaja

Lista upravljenih uređaja je predstavljena kao *DeviceList* struktura koja sadrži pokazivač na listu i muteks. Ovaj pomoćni modul se takođe koristi od strane kontrolne tačke s ciljem formiranja i rukovanja listom upravljenih uređaja unutar mreže koji ispunjavaju njene kriterijume. Sve funkcije u okviru ovog modula su zaštićene datim muteksom tako da onemoguće istovremeni pristup listi od strane više programske niti. Inicijalizacija muteksa se vrši pozivom odgovarajuće funkcije *ithread* modula u okviru UPnP biblioteke.

Uredaji koji se smeštaju u listu su predstavljeni *DeviceNode* strukturom. Ona sadrži sve informacije o uređaju koje su od značaja za kontrolnu tačku: UDN (eng. Unique Device Name), ime uređaja, vreme isteka objave, lista zastupljenih usluga itd.

Lista uređaja pruža sledeće funkcionalnosti:

- kreiranje i uništavanje liste
- dodavanje novootkrivenog uređaja u listu
- brisanje uređaja iz liste
- dobavljanje informacije o konkretnom uređaju
- ažuriranje informacija o konkretnom uređaju
- pribavljanje liste dostupnih uređaja
- pribavljanje broja uređaja u listi

5. Ispitivanje i rezultati

5.1 Ispitivanje programske podrške

Sam proces ispitivanja počeo je i tekao uporedno sa fazom razvoja programskog rešenja. Greške i nedostaci koji su uočeni tokom realizacije su uklanjeni. Za potrebe ispitivanja i verifikacije funkcionalnosti programskog rešenja kreirane su posebne Android aplikacije koje su implementirale funkcionalnosti realizovane biblioteke. Jedna aplikacija je radila u režimu kontrolne tačke, a druga u režimu upravljanog uređaja. Praćenjem njihovih ispisa proveravana je realizacija algoritma tj. da li se program ponaša u skladu sa očekivanim ponašanjem. U tabeli 5.1 izloženi su ključni testni slučajevi kojima je proveravana ispravnost realizacije programskog rešenja.

Testni slučaj	Očekivano ponašanje	Uspešnost
Uključivanje kontrolne tačke u mrežu u kojoj postoji aktivan uređaj za upravljanje.	Po uključivanju u mrežu, kontrolna tačka pretražuje mrežu da pronađe upravljljane uređaje od interesa. Kada pronađe uređaj, dodaje ga u svoju listu uređaja.	Uspešan
Uključivanje uređaja u mrežu u kojoj postoji aktivna kontrolna tačka	Po uključivanju u mrežu, uređaj objavljuje svoje prisustvo kontrolnim tačkama. Kontrolna tačka prima objavu i dodaje uređaj u svoju listu ukoliko je od interesa za nju.	Uspešan
Isključivanje upravljanog uređaja iz mreže	Prilikom isključivanja sa mreže, uređaj šalje poruku kojom poništava prethodne objave. Kontrolna tačka će po priјemu ove poruke ukloniti dati uređaj iz svoje liste jer više nije raspoloživ.	Uspešan

Ažuriranje vremena do isteka objave za uređaj u listi uređaja kontrolne tačke	Kada kontrolnoj tački stigne objava datog uređaja, ona ažurira broj sekundi do isteka objave tog uređaja na primljenu vrednost. U periodu između objava, vreme objave ističe. Ukoliko vreme istekne, uređaj se uklanja iz liste, jer se smatra da nije više na raspolaganju.	Uspešan
Slanje zahteva za akcijom uređaju od strane kontrolne tačke	Uređaj primi poruku koja sadrži ime akcije i parametre. Nakon toga kreće u izvršavanje funkcije koja predstavlja datu akciju i šalje odgovor kontrolnoj tački.	Uspešan

Tabela 5.1 Testni slučajevi

Za potrebe ispitivanja tokom realizacije korišćen je i programski alat *Valgrind*, koji služi za profilisanje memorije i pronalaženje i uklanjanje problema vezanih za rukovanje memorijom, što uključuje i otkrivanje “curenja memorije” (eng. memory leak). Ovim alatom je utvrđeno da je rešenje korektno urađeno i da ne postoji mogućnost da dođe do “curenja memorije”.

5.2 Performanse programske podrške

Performanse programske podrške su praćene korišćenjem *top* komande. *Top* komanda pruža pogled na aktivnosti procesora u realnom vremenu. Prikazuje spisak procesa i za svaki od njih prikaže koliki procenat procesora koriste, kao i koliko virtuelne i fizičke memorije zauzimaju. Postoji mogućnost sortiranja procesa po procentu iskorišćenosti procesora, veličini virtuelne ili fizičke memorije. Većina funkcija može da se bira interaktivnom komandom ili njihovim navođenjem u konfiguracionom fajlu.

Performanse su praćene pokretanjem testnih aplikacija koje koriste realizovanu biblioteku na dve Android platforme. Jedna platforma je predstavljala upravljeni uređaj, a druga je radila u oba režima, i kao upravljeni uređaj i kao kontrolna tačka. Merenja su pokazala da se iskorišćenost procesora na obe platforme kreće između 0% i 1%. Što se tiče zauzeća memorije, rezultati zabeleženih merenja prikazani su u sledećim tabelama:

VSS(KB)	7016	7016	7016	10088	7016	7016	9064	7016	7016
RSS(KB)	528	528	528	548	528	528	544	528	528

Tabela 5.2 Memorija koju zauzima aplikacija pokrenuta u režimu uređaja na platformi

VSS(KB)	6984	7004	7004	9052	7004	8028	7004	7004	7004
RSS(KB)	500	512	512	524	512	520	512	512	512

Tabela 5.3 Memorija koju zauzima aplikacija pokrenuta u režimu uređaja na platformi na kojoj su pokrenute aplikacije u oba režima

VSS(KB)	8060	9084	8060	8060	8060	10108	8060	8060	8060
RSS(KB)	536	548	536	536	536	560	536	536	536

Tabela 5.4 Memorija koju zauzima aplikacija pokrenuta u režimu kontrolne tačke na platformi na kojoj su pokrenute aplikacije u oba režima

RSS (eng. Resident Set Size) pokazuje koliko proces zauzima fizičke memorije (RAM). VSS (eng. Virtual Set Size) pokazuje koliko virtuelne memorije zauzima proces. U trenucima kada upravljeni uređaj prima zahtev za izvršenje akcije, zabeležene su nešto veće vrednosti zauzeća memorije. Razlog tome je to što tada upravljeni uređaj parsira XML dokument u okviru kog mu je stiglo ime akcije koju je potrebno pokrenuti sa odgovarajućim parametrima. U slučaju kontrolne tačke, nešto veće vrednosti su zabeležene u trenucima slanja zahteva za izvršavanje akcije, jer se tada formira XML dokument koji sadrži ime akcije za izvršavanje i odgovarajuće parametre.

Prilikom praćenja performansi, zabeleženo je trajanje razmene podataka između kontrolne tačke i uređaja. Mereno je vreme na kontrolnoj tački od trenutka kada je poslat zahtev za izvršavanjem akcije do trenutka kada je putem povratne funkcije kontrolnoj tački stigao odgovor o uspešnosti izvršenja akcije. Za poslatih sto uzastopnih zahteva, vreme koje je proteklo do završetka poslednje akcije iznosilo je oko 0,15 sekundi. Za izvršavanje hiljadu uzastopnih zahteva bilo je neophodno oko 1,5 sekundi.

6. Zaključak

U ovom radu je realizovana programska podrška za komunikaciju između centralnog uređaja i „drugog ekrana” u digitalnoj televiziji. Pre same realizacije programske podrške bilo je neophodno izvršiti analizu raspoloživih komunikacionih protokola i izabrati jedan na kojem bi se zasnivalo rešenje. Nakon analize pronađenih protokola izabran je UPnP protokol. Korišćena je biblioteka *libUPnP-1.6.12* koja predstavlja jednu od realizacija datog protokola. Programska podrška je realizovana za Android platformu korišćenjem programskog jezika C. Rezultat je dinamička biblioteka bazirana na funkcionalnostima *libUPnP-1.6.12* biblioteke.

Programska podrška je projektovana i realizovana tako da se moduli koji je čine po potrebi lako i brzo mogu izmeniti ili nadograditi. Prostora za unapređivanje i nadogradnju rešenja ima mnogo. Na jednostavan način se mogu dodavati nove usluge i akcije koji bi uređaj mogao da podržava, što bi dovelo i do proširivanja aplikativne programske sprege funkcijama koje bi omogućile korišćenje tih novih usluga. Ispitivanja pomoću testne aplikacije su pokazala da se program ponaša u skladu sa očekivanjima.

U bliskoj budućnosti je planiran nastavak na realizaciji programske podrške. Primarni fokus je integrisanje njenih funkcionalnosti u postojeću Android aplikaciju za TV. Postavljanjem iste aplikacije na prateći uređaj, biće omogućena razmena sadržaja između njega i televizora. Takođe je u planu dodavanje novog modula koji će biti zadužen za generisanje opisnog dokumenta uređaja (DDD).

7. Literatura

- [1] *The New Multi-Screen World: Understanding Cross-Platform Consumer Behaviour*, Google Study, 2012.
- [2] *Understanding Universal Plug and Play, white paper*, www.upnp.org, učitano 23.06.2014.
- [3] *DLNA for HD Video Streaming in Home Networking Environments*, www.dlna.org, učitano 23.06.2014.
- [4] *Discovery And Launch protocol specification, version 1.7*, www.dial-multiscreen.org, 2014.
- [5] M. Jeronimo, J. Weast: *UPnP Design by Example, A Software Developer's Guide to Universal Plug and Play*, Intel Press, 2003.
- [6] <http://sourceforge.net/projects/pupnp/files/pupnp/libUPnP%201.6.12/>, učitano 23.06.2014.
- [7] *Linux SDK for UPnP devices, v1.4*