

# Jedno rješenje poslužioca televizijskog sadržaja sa podrškom za IPv4 i IPv6 protokole

Radenko Banović, Ilija Bašičević

**Apstrakt** — U ovom radu je prikazano jedno rješenje poslužioca televizijskog sadržaja sa podrškom za IPv6 (Internet Protokol verzije 6) kao i IPv4 (Internet Protokol verzije 4) prenosom podataka višeznačnim i jednoznačnim upućivanjem i korištenjem RTP (eng. Real-Time Transport Protocol) / RTSP (eng. Real-Time Streaming Protocol) protokola. Zbog masovnog prelaska na IPv6 protokol pojavili su se prijemnici multimedijalnog toka podataka u kojima je implementiran isključivo IPv6 protokol iako je IPv4 (internet protokol verzije 4) i dalje najdominantniji internet protokol na svijetu. Krajnji cilj je da IPv6 protokol (čije se učešće u svjetskom internetu svakodnevno povećava) u konačnici u potpunosti zamijeni IPv4 protokol. Stoga, pojavila se potreba za poslužiocem koji podržava oba navedena protokola. Rješenje je testirano korištenjem nekoliko klijenata.

**Ključne reči** — IPv4, IPv6, RTP, RTSP, IP

## I. UVOD

Dostavljanje televizijskog sadržaja putem interneta poslednjih godina doživljava veliki procvat zbog sve bolje infrastrukture računarskih mreža koja omogućuje brži protok podataka, što je prikazano u radovima [1] i [2]. Njegova prednost je u tome što omogućuje dvosmjernu klijent – server komunikaciju, te može da se koristi već postojeća mrežna infrastruktura, bez potrebe za pravljenjem nove fizičke infrastrukture.

Korištenje mrežne infrastrukture podrazumijeva implementaciju protokola koji služe za komunikaciju u računarskim mrežama. Da bi se TV (televizijski) sadržaj efikasno dostavljao putem interneta koristi se protokol za prenos u realnom vremenu RTP koji je enkapsuliran u UDP (eng. User Datagram protocol) i protokol za prenos toka u realnom vremenu RTSP. Sve dok traje tranzicijski period sa IPv4 na IPv6 protokol potrebno je imati rješenja koja podržavaju rad sa oba navedena protokola čime se omogućuje većem broju klijenata pristup željenom sadržaju.

Ovaj rad je sačinjen od 5 poglavlja. U prvom poglavlju su dat je kratak opis rada. U drugom dijelu su opisani protokoli korišteni u implementaciji. Opis rješenja je prikazan u trećem poglavlju u kom su opisane i ključne komponente sistema. U četvrtom poglavlju je opisan način funkcionalne provjere i opis testnih slučajeva. Peto poglavlje sadrži kratak pregled rada i potencijalna unaprjeđenja realizovanog rješenja.

Radenko Banović – Fakultet Tehničkih Nauka, Trg Dositeja Obradovića 6, 21000 Novi Sad, Srbija (e-mail: Radenko.Banovic@rt-rk.com).

Ilija Bašičević – Istraživačko-razvojni Institut RT-RK, Novi Sad, Srbija (e-mail: Ilija.Basicevic@rt-rk.com).

## II. PROTOKOLI KORIŠTENI U IMPLEMENTACIJI

Protokoli prikazani u ovom radu se najlakše vizuelno predstavljaju OSI referentnim modelom koji dijeli arhitekturu mreže u sedam logičkih nivoa, daje spisak servisa i protokola koji funkcionišu na svakom od nivoa (Sl. 1.).

Slojevi	Jedinica	Protokoli
Aplikacija	Paket	HTTP, FTP, RTP, RTSP, POP, SSH, DHCP, ...
Prezentacija	Paket	MIME, SSL, TLS, XDR, ...
Sesija	Paket	SSH, SIP, PAP, CHAP, ...
Transport	Segment	TCP, UDP, DCCP, SCTP
Mreža	Datagram	IP (IPv4, IPv6), ICMP, ARP...
Sloj veze	Okvir	PPP, Frame Relay, HDLC, ...
Fizički sloj	Bit	Ethernet, USB, Bluetooth, ...

Sl. 1. Slojevi OSI modela

Tri gornja sloja (aplikacija, prezentacija i sesija) imaju ulogu da opišu proces komunikacije aplikacija među sobom kao krajnjim tačkama, dok donja četiri sloja (transport, mreža, sloj veze i fizički sloj) imaju ulogu definisanja prenosa informacije sa jednog na drugi kraj mreže.

### A. RTP protokol

Protokol za prenos u realnom vremenu – RTP je mrežni protokol za dostavljanje audio i video sadržaja preko IP mreže. Najčešće se izvršava korišćenjem UDP protokola (Sl. 2.), kome je brzina prenosa podataka bitnija od kvaliteta primljenog signala.

Zbog grešaka u prenosu paketi mogu da se izgube, zakasne ili prenesu bez poštovanja redoslijeda. Protokol za prenos u realnom vremenu ne obezbjeđuje isporuku podataka na vrijeme i ne uključuje vremenska odstupanja. Međutim on u svom zaglavlju nosi podatke koji omogućavaju da se utvrdi koje su poruke izgubljene u prenosu i na osnovu toga ispravlja redoslijed paketa na prijemu. Sam RTP protokol ne može da rezerviše resurse na mreži, ili da obezbijedi kvalitet prenosa. RTP protokol je specificiran u dokumentu [3].

IP zaglavlje	UDP zaglavlje	RTP zaglavlje	RTP sadržaj
--------------	---------------	---------------	-------------

Sl. 2. Struktura IP paketa

Digitalizovan audio i video signal (u bitima) se smješta u RTP sadržaj ispred koga se dodaje RTP zaglavlje (Sl. 3.) koje nosi informacije o korisničkom sadržaju, načinu kodovanja, itd.

Redni broj bita	0-1	2	3	4-7	8	9-15	16-31
0	Verzija	P	X	CC	M	PT	Redni broj poruke
32	Vrijeme nastanka						
64	Identifikator izvora						
96	Lista učesnika						
96+32+CC	Definicija učesnika				Dužina zaglavljia proširenja		
128+32+CC	Zaglavlje proširenja						

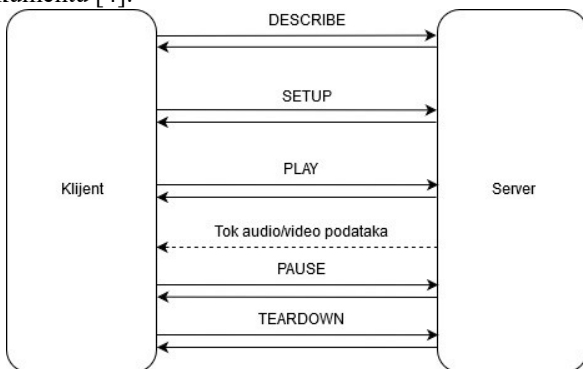
Sl. 3. Zaglavlje RTP paketa

### B. RTSP protokol

Protokol za kontrolu prenosa toka u realnom vremenu (eng. Real-Time Streaming Protocol) je mrežni protokol dizajniran za kontrolu nad dostavljanjem audio/video sadržaja u realnom vremenu, na aplikativnom nivou. Ovaj protokol omogućava odabir kanala isporuke (kao i protokol dostavljanja sadržaja), dakle protokol za dostavljanje audio/video sadržaja ne mora biti RTP, već bilo koji drugi protokol sa istom namjenom (kao što je HLS (eng. Http Live Streaming)).

Protokol se koristi za uspostavljanje i kontrolu veze između klijenta i servera, kao i slanje klijentskih zahtjeva kao što su puštaj (eng. Play), snimaj (eng. Record) i pauziraj (eng. Pause), što je prikazano u Sl. 4.

RTSP je baziran na RTSP zahtjevima koji se šalju od klijenta ka serveru, na koje server treba da odgovori i izvrši akciju koja se od njega očekuje. RTSP protokol je specifikiran u dokumentu [4].



Sl. 4. Dijagram RTSP komunikacije

Svaki od RTSP zahtjeva i odgovora ima svoj format, primjer RTSP options zahtjeva je prikazan u Sl 5.

Zahtjev od strane klijenta	<pre> OPTIONS rtsp://example.com/media.mp4RTSP/1.0 CSeq: 1 Require: implicit-play Proxy-Require: gzipped-messages           </pre>
Odgovor od strane poslužioca	<pre> RTSP/1.0 200 OK CSeq: 1 Public: DESCRIBE, SETUP, TEARDOWN, PLAY, PAUSE           </pre>

Sl. 5. RTSP options zahtjev

### C. IP protokol

Internet protokol je protokol trećeg sloja OSI referentnog modela – sloja mreže. Sadrži informacije o adresiranju, čime se postiže da svaki mrežni uređaj (računar, server, radna stanica, ...) koji je povezan na internet ima jedinstvenu adresu i može se lako identifikovati u cijeloj internet mreži.

IP ima tri osnove funkcije:

- Adresiranje – definiše način dodjele internet adresa
- Rutiranje – određivanje putanje za prenos podataka sa jednog računara na drugi bez prethodne uspostave veze
- Fragmentaciju – i ponovno sastavljanje paketa kada je to potrebno

### D. IPv4

Internet protokol verzija 4 je najrašireniji internet protokol na internetu, pojavljuje se 1983. godine, a definisan je u dokumentu RFC 791 septembra 1981. godine. IPv4 koristi 32-bitni adresni prostor što ga limitira na  $2^{32}$  jedinstvenih adresa, ali veliki blokovi su rezervisani za specijalne mrežne metode (privatne mreže i adrese za višeznačno upućivanje).

IPv4 adresa može biti predstavljena bilo kojom notacijom koja predstavlja 32-bitnu cjelobrojnu vrijednost, ali je najčešće predstavljena sa po četiri 8-bitna bloka predstavljena decimalno, i odvojena tačkama [5].

### E. IPv6

Pošto su zalihe IPv4 adresa pri samom kraju, pojavila se potreba za prelaskom na adrese sa većom adresnom širinom. IPv6 je novi, ali još ne široko korišćen, standardni internet protokol, čije su adrese 128 bita široke, što znači da postoji tačno  $2^{128}$  unikatnih adresa, a to bi trebalo da zadovolji blisku budućnost. IPv6 adresa je predstavljena sa osam četvorocifrenih heksadecimalnih brojeva (8 puta po 16 bita) odvojenih dvotačkama. IPv6 ne uključuje mehanizam kontrolne sume [6]. Zaglavlje IPv6 paketa je prikazano u Sl. 6.

Verzija (4 bita)	Klasa saobraćaja (8 bita)	Oznaka toka (20 bita)	
Dužina podataka (16 bita)		Sledeće zaglavlje (8 bita)	Broj skokova (8 bita)
Adresa izvora (128 bita)			
Adresa odredišta (128 bita)			
Podaci dodatnog zaglavljia (promjenljiva dužina)			

Sl. 6. Zaglavlje IPv6 paketa

## III. OPIS RJEŠENJA

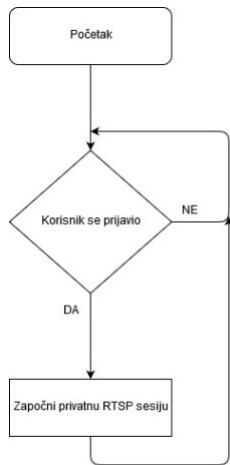
Ovaj poslužilac je realizovan tako da omogući i video na zahtjev poslužnje podataka, kao i prenos podataka višeznačnim upućivanjem za živi prenos multimedijalnog sadržaja. Podešavanjem utičnica pruža se podrška za IPv4 I IPv6 protokole.

Izvršava se u komandnoj liniji (bez grafičkog interfejsa), osnovne informacije (port, putanja do fajla za prenos višeznačnim upućivanjem, itd.) se podešavaju prosljeđivanjem parametara prilikom pokretanja softvera, dok klijent može uticati na vrstu sadržaja koja mu se dostavlja korišćenjem RTSP protokola.

### A. RTP/RTSP poslužilac jednoznačnim upućivanjem

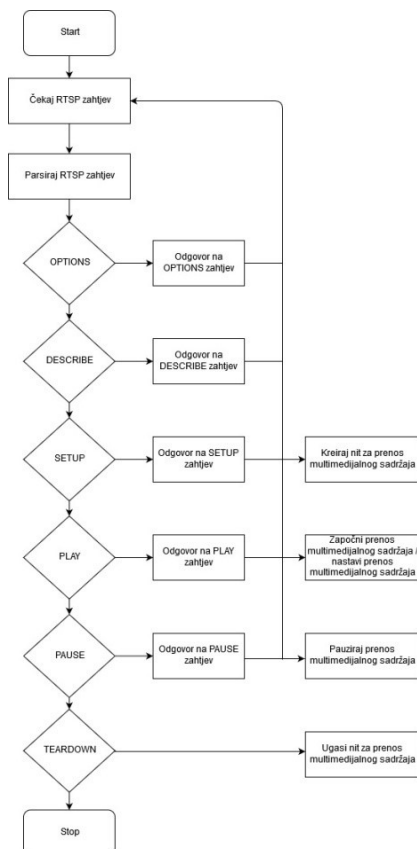
Mora da implementira protokol koji se koristi za uspostavljanje u kontrolu veze između klijenta i poslužioca, kao i slanje klijentskih zahtjeva. Pošto je protokol za kontrolu

prenosa toka u realnom – RTSP protokol, najkorišćeniji protokol za takvu namjenu, on će biti korišćen za kontrolu veze i slanje klijentskih zahtjeva između klijenta i poslužioca. Naredna slika prikazuje arhitekturu osnovne niti poslužioca.



Sl. 7. Osnovna arhitektura poslužioca jednoznačnim upućivanjem

Osnovna nit čeka prijavljivanje korisnika i za svakog od prijavljenih korisnika pokreće posebnu nit koja vrši RTSP komunikaciju sa klijentom individualno koja je prikazana na na Sl. 8, bez ometanja drugih klijenata koji su istovremeno opsluživani od strane istog poslužioca.

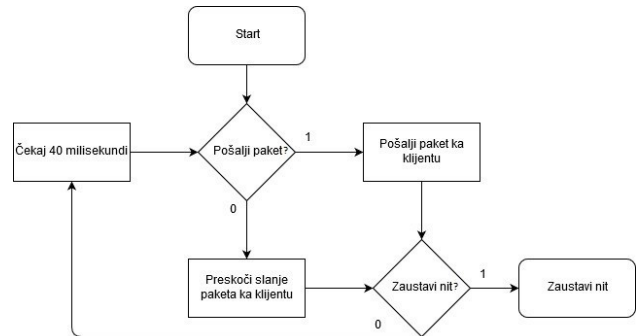


Sl. 8. Klijentska nit za RTSP komunikaciju

Prilikom obrade klijentskog RTSP zahtjeva OPTIONS potrebno je kreirati novu nit koja čita bitske vrijednosti iz

MPEG-TS fajla koji će se korišćenjem RTP protokola prenositi do klijenta. Prilikom kreiranja niti kao parametar se prenose klijentski RTP i RTSP port, kao i identifikacioni broj koji jednoznačno određuje svakog klijenta.

U toku obrade RTSP zahtjeva PLAY i PAUSE mijenja se zastavica koja ima vrijednost 1 kada treba da se prenosi multimedijalni tok ka klijentu, odnosno 0 kada prenos treba da bude pauziran. Prilikom obrade RTSP zahtjeva TEARDOWN zastavica koja označava gašenje niti u kojoj se prenosi multimedijalni tok ka klijentu se postavlja na 1, što označava gašenje niti. Dijagram niti za prenos podataka je prikazan u Sl. 9.

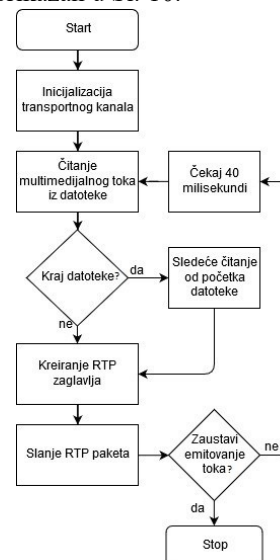


Sl. 9. Dijagram niti za prenos podataka

### B. RTP poslužilac višeznačnim upućivanjem

Višeznačno upućivanje je napogodniji način prenosa multimedijalnih tokova koji se u jednom trenutku trebaju poslati ka nekoliko krajnjih korisnika, jer se u tom slučaju šalje samo jedan IP paket do grupe klijenata, čime se štedi propusni opseg a krajnjim korisnicima omogućava jednak kvalitet dostavljenog sadržaja.

Za ovakav vid prenosa podataka klijent treba da zna samo IP adresu poslužioca, port na koji će se emitovati multimedijalni tok podataka i protokol po kom se podaci šalju, te je njegova arhitektura dosta jednostavnija od arhitekture RTP/RTSP poslužioca jednoznačnim upućivanjem. Dijagram RTP poslužioca višeznačnim upućivanjem je prikazan u Sl. 10.

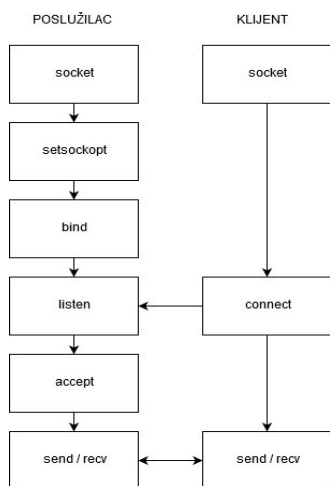


Sl. 10. Dijagram RTP poslužioca višeznačnim upućivanjem

### C. Podešavanje mrežnih utičnica

Mrežna utičnica (eng. Socket) je krajnja tačka dvosmjernog međuprocenog komunikacionog toka preko računarske mreže bazirane na internet protokolu, kao što je internet. Adresa mrežne utičnice je spoj IP adrese i porta u jedinstven entitet. Mrežna utičnica je određena jedinstvenom kombinacijom protokol (TCP ili UDP), lokalne adrese mrežne utičnice (adresa i port) i udaljene adrese mrežne utičnice (samo za uspostavljene TCP mrežne utičnice).

Da bi mogli raditi sa mrežnim utičnicama u C++ programskom jeziku na Linux operativnom sistemu potrebno je uključiti datoteke `<sys/socket.h>`, `<netinet/in.h>` i `<arpa/inet.h>`. Dijagram komunikacije i kreiranja mrežnih utičnica je prikazan u Sl. 11.



Sl. 11. Dijagram komunikacije i kreiranja mrežnih utičnica

Prvi korak pri kreiranju mrežne utičnice je poziv funkcije `socket(int domain, int type, int protocol)` čija je povratna vrijednost file descriptor integer tipa koji jednoznačno određuje mrežnu utičnicu. Parametar `domain` određuje porodicu protokola koja će biti korištena za komunikaciju. Vrijednost koja se prosljeđuje za IPv6 protokol u parametru `domain` je `AF_INET6`, dok se za IPv4 protokol prosljeđuje vrijednost `AF_INET`. Parametar `type` određuje tip veze na transportnom nivou OSI model (TCP ili UDP). Prvi korak u ostvarivanju komunikacije sa klijentom je zasnivanje veze bazirane na RTSP protokolu, potrebno je otvoriti TCP vezu (koja je dvosmjerna i pouzdana), tako da se prosljeđuje vrijednost `SOCK_STREAM`. Posljednji parametar određuje koji protokol će biti korišten sa prosljeđenom porodicom protokola, tako da se najčešće prosljeđuje vrijednost 0.

Pozivom funkcije `setsockopt(int socket, int level, int option_name, const void *option_value, socklen_t option_len)` moguće je podesiti specifičnosti, ali nije obavezno. Kada je mrežna utičnica kreirana pozivom funkcije `socket()` ona još uvijek nema dodjeljenu adresu. Pozivom funkcije `bind(int sockfd, const struct sockaddr *addr, socklen_t addrlen)` dodjeljuje se adresa određena parametrom `addr` mrežnoj utičnici čiji je file descriptor prosljeđen kao parametar `sockfd`. Struktura koja se popunjava za IPv6 mrežu je `sockaddr_in6`, u polje `sin6_family` se postavlja vrijednost `AF_INET6`, u polje `sin6_port` se postavlja vrijednost koja je prosljeđena kao

parametar komandne linije prilikom pokretanja aplikacije, dok se u polje `sin6_addr` postavlja vrijednost `in6addr_any`. Struktura koja se popunjava za IPv4 mrežu je `sockaddr_in` sa poljima `sin_family`, `sin_port`, `sin_addr`.

Pozivom funkcije `listen(int sockfd, int backlog)` označavamo mrežnu utičnicu određenu vrijednošću `sockfd` kao pasivnu mrežnu utičnicu, te će ona biti korištena za primanje zahtjeva za konekciju. Vrijednost parametra `backlog` određuje maksimalan broj neobrađenih zahtjeva za konekciju koji mogu biti prihvaćeni u jednom trenutku, svi koji pošalju zahtjev ako je broj neobrađenih zahtjeva popunjen dobiće poruku o grešci.

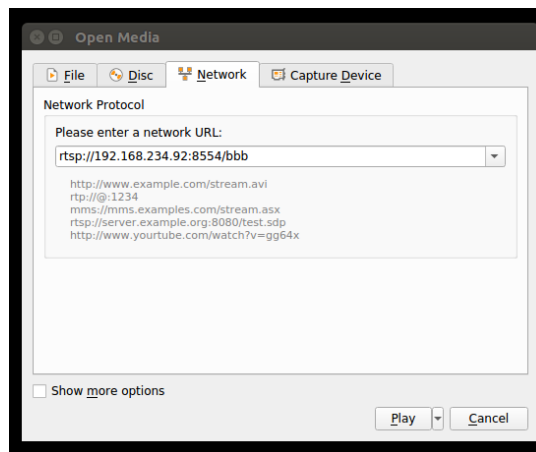
Funkcija `accept(int sockfd, struct sockaddr *addr, socklen_t *addrlen)` je blokirajuća funkcija. Ona iz reda zahtjeva za konekciju izuzima prvi zahtjev u redu po FIFO (eng. First In First Out) redoslijedu, te kreira novu mrežnu utičnicu. Povratna vrijednost ove funkcije je file descriptor novokreirane mrežne utičnice koja predstavlja klijenta. Nakon ovoga poslužilac i klijent mogu da razmjenjuju poruke po IPv6 protokolu pozivanjem funkcija `send()` i `recv()`.

## IV. TESTIRANJE

Za testiranje je korišteno nekoliko klijenata: AVPlay, VLC, komercijalna aplikacija koja može da se izvršava na privatnom računaru i set-top box uređaju, kao i klijent napravljen u C programskom jeziku koji simulira rad klijenta i upisuje dobijene podatke u datoteku (Sl. 12 i Sl. 13).

### A. Testiranje korišćenjem AVPlay i VLC klijenta

AVPlay i VLC klijenti su korišteni za testiranje RTP/RTSP poslužioca jednoznačnim upućivanjem.



Sl. 12. URL za uspostavljanje veze sa poslužiocem

Testiranja su vršena na način da se u klijentskoj aplikaciji unese URL koji je sadržan od protokola koji se koristi, IP adrese računara na kom se pokreće poslužilac (internet protokol verzije 4 ili verzije 6), porta, dok poslednji dio predstavlja sadržaj koji klijent želi da mu poslužilac dostavi.

### B. Testiranje korišćenjem komercijalnog STB uređaja

Korišćenjem komercijalnog STB uređaja testirani su i poslužioc višeznačnim upućivanjem kao i poslužioc jednoznačnim upućivanjem.



Sl. 13. Prikazan sadržaj na komercijalnom STB uređaju

Klijent je prilagođen tako da mu je na jednom IPTV servisu promijenjena izvorna adresa poslužioca, dok je poslužilac slao podatke na adresu za višeznačno usmjeravanje koju klijent očekuje. Testiranje na klijentu je vršeno audio-vizuelno, i sve datoteke su se uspješno reprodukovale na klijentskoj strani.

### C. Testiranje korišćenjem klijentske testne aplikacije

Razvijena je testna aplikacija koja u potpunosti simulira rad klijenta, tako da podržava i prenos podataka jednoznačnim i višeznačnim upućivanjem, RTP i RTSP protokol za IPv4 i IPv6 protokol, ali tako da umjesto prikaza audio i video sadržaja korisniku isti upisuje u izlaznu datoteku [7]. Prilikom testiranja poslužioca jednoznačnim upućivanjem zaustavljan je prenos podataka slanjem klijentske RTSP poruke Pause, i nastavljan je prenos podataka slanjem klijentske poruke Play. Izvorne datoteke sadrže multimedijalni sadržaj u TS (eng. Transport Stream) formatu. Izlazna datoteka bi po završetku prenosa podataka trebala da ima isti sadržaj kao izvorna datoteka da bi test bio uspješan. Poređenja su vršena korišćenjem cmp komande Linux operativnog sistema koja poredi svaki bajt ulazne datoteke sa bajtom izlazne datoteke na istoj poziciji i prijavljuje prvu poziciju na kojoj se nalazi razlika u bajtima. Ukoliko su dve datoteke identične, komanda se završava bez ispisa o razlikama. Sva poređenja ulaznih i izlaznih datoteka su bila bez ispisa o greškama čime se potvrđuje da su testovi uspješni (Tabela I).

TABELA I

REZULTATI TESTOVA POSLUŽIOCA JEDNOZNAČNIM I VIŠEZNAČNIM UPUĆIVANJEM

Izvorna datoteka	Vrsta upućivanja/ Protokol	Različih bajta	Rezultat
izvorna 1.ts	Jednoznačno/ IPv4	0	Uspješan
izvorna 1.ts	Jednoznačno/ IPv6	0	Uspješan
izvorna 2.ts	Jednoznačno/ IPv4	0	Uspješan
izvorna 2.ts	Jednoznačno/ IPv6	0	Uspješan
izvorna.ts	Jednoznačno/ IPv4	0	Uspješan
izvorna.ts	Jednoznačno/ IPv6	0	Uspješan
izvorna 1.ts	Višeznačno/ IPv4	0	Uspješan
izvorna 1.ts	Višeznačno/ IPv6	0	Uspješan
izvorna 2.ts	Višeznačno/ IPv4	0	Uspješan
izvorna 2.ts	Višeznačno/ IPv6	0	Uspješan
izvorna.ts	Višeznačno/ IPv4	0	Uspješan

## V. ZAKLJUČAK

U ovom radu je realizovan poslužilac multimedijalnog sadržaja sa podrškom za prenos podataka jednoznačnim i višeznačnim upućivanjem, i oba IP protokola. Opisana je arhitektura kompletne aplikacije, te je dat spisak funkcionalnosti. Do detalja je opisano kreiranje mrežnih utičnica tako da podržavaju rad sa oba IP protokola.

Rješenje je verifikovano na nekoliko klijentskih aplikacija, i krajnje rješenje je uspješno izvršilo sve testove. Dato rješenje je moguće nadograditi na nekoliko načina. Ova aplikacija podržava rad isključivo sa RTP protokolom, dok bi se mogla proširiti tako da podržava i neke druge protokole, poput HLS protokola. Time bi se proširio broj klijenata kojima bi multimedijalni sadržaj bilo moguće dostaviti.

Moguće je proširiti podršku i za ostale RTSP zahtjeve, kao što su GET\_PARAMETER, SET\_PARAMETER, REDIRECT, ANNOUNCE, itd.

## ZAHVALNICA

Ovaj rad je delimično finansiran od strane Ministarstva za prosvetu, nauku i tehnološki razvoj Republike Srbije.

## LITERATURA

- [1] C.H. Lee, H.C. Chiu, "Telco IPTV growth strategy in Taiwan," in *13<sup>th</sup> International Conference on Advanced Communication Tehchnology*, pp. 1469-1474, Feb. 2011.
- [2] E. H. Khabbiza, R. E. Alami, H. Qjidaa, "A new solution to optimize the time shift TV bandwidth," in *International Conference on Intelligent Systems and Computer Vision (ISCV)*, Fez, Morocco, April, 2018.
- [3] RTP: A Transport Protocol for Real-Time Applications, Jul 2003. [online]. Available: <https://tools.ietf.org/html/rfc3550>
- [4] Real Time Streaming Protocol (RTSP), April 1998. [online]. Available: <https://tools.ietf.org/html/rfc2326>
- [5] Internet Protocol, Septembar 1981. [online]. Available: <https://tools.ietf.org/html/rfc791>
- [6] Internet Protocol Version 6 (IPv6), Decembar 1998. [online]. Available: <https://tools.ietf.org/html/rfc2460>
- [7] Apparatus and method for dtv closed-captioning processing in broadcasting and communication system, by JeHo Nam. (2011, Jun 23). Accessed on Sept. 2. 2020. [Online]. Available: <https://patents.google.com/patent/US20110149153A1/en>

## ABSTRACT

This paper presents a solution of television content streamer with support for IPv6 (Internet Protocol version 6) as well as IPv4 (Internet Protocol version 4). Unicast and multicast transmission is used and rtp/rtsp protocol. Due to the mass transition to IPv6 protocol, some media players have implemented only IPv6 protocol eventhough IPv4 is still the most dominant Internet protocol in the world. The ultimate goal is that the IPv6 protocol (whose participation in the global Internet is increasing daily) will eventually completely replace the IPv4 protocol. Therefore, there is a need for streamer that supports both of these protocols. The solution was tested using several clients.

## One TV Content Server Solution with Support for IPv4 and IPv6 Protocols

Radenko Banović, Ilija Bašičević