Rješavanje NAT *traversal* problema upotrebom TURN protokola i *Media Latching* mehanizma

Dragan Botić, mr Siniša Vujčić
Direkcija za tehniku
Mtel a.d. – Telekom Srpske
Banja Luka, RS, BiH
dragan.botic@mtel.ba, sinisa.vujcic@mtel.ba

mr Dejan Nemec
Departman za energetiku, elektroniku i telekomunikacije
Fakultet tehničkih nauka, Univerzitet u Novom Sadu
Novi Sad, Srbija
denem@uns.ac.rs

Sažetak **— U ovom radu opisan je NAT *traversal* problem, izložena su njegova moguća rješenja upotrebom TURN protokola i *Media Latching* mehanizma. Urađena je uporedna analiza rada TURN protokola i *Media Latching* mehanizma, te na osnovu rezultata izveden zaključak i date preporuke u smislu izbora optimalnog rješenja.**

Ključne riječi – NAT; TURN; Media Latching; SIP, SBC

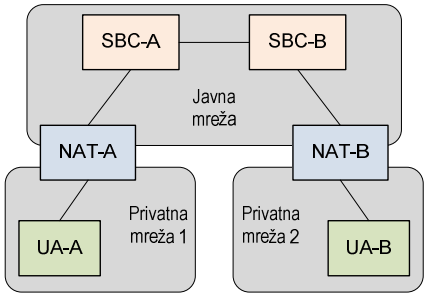
I. UVOD

Intenzivan razvoj Interneta u proteklih deset do petnaest godina doveo je do konvergencije telekomunikacionih mreža u jednu jedinstvenu mrežu za prenos različitih vrsta servisa. U okviru ove mreže klasičan prenos govora komutacijom kola postepeno je zamjenjen komutacijom paketa, pri čemu se koristi IP protokol (*Internet Protocol*). Najčešće korišćen naziv za ovaj način prenosa govora je VoIP (*Voice over Internet Protocol*). IP protokol, prvobitno namijenjen za prenos podataka, zahvaljujući razvoju dodatnih protokola, počinje intenzivnije da se koristi i za prenos servisa u realnom vremenu: govor, video, multimedijalni sadržaji i drugo. Jedan od ključnih dodatnih protokola razvijenih za uspostavljanje, modifikaciju i raskidanje multimedijalnih sesija u paketskim mrežama je SIP (*Session Initiation Protocol*) protokol [1]. U poslednjih desetak godina SIP je prerastao istraživačku i akademsku fazu i postao praktično standard za multimedijalne servise u mobilnoj i fiksnoj mreži. Jedan od krupnijih nedostataka SIP protokola, čija priroda je u *end-to-end* komunukaciji, jeste činjenica da se tokom razvoja računalo na brži tempo iscrpljivanja IPv4 adresnog prostora i ranije uvođenje u upotrebu IPv6 adresnog prostora, čime bi bio prevaziđen problem konektivnosti u prisustvu NAT (*Network Address Translation*). Ipak, kako se IPv4 adresni prostor zadržao duže od predviđenog, trebalo je riješiti taj problem. Rješenja su najčešće realizovana upotrebom TURN (*Traversal Using Relay around NAT*) protokola ili *Media Laching* mehanizma u okviru SBC-a (*Session Border Controller*).

Cilj ovog rada jeste da prikaže NAT *traversal* problem (konektivnost u prisustvu NAT-a), njegova moguća rješenja upotrebom TURN protokola i *Media Latching* mehanizma, uporedi ova rješenja, i na osnovu toga izvede zaključak i da preporuke u pogledu izbora optimalnog rješenja.

II. PROBLEM NAT FUNKCIONALNOSTI

NAT je široko rasprostranjena funkcionalnost u Internet mreži. Spektar funkcionalnosti NAT uređaja vezan je za mapiranje IP adresa i portova između različitih adresnih područja. Ovdje će se NAT funkcionalnosti ograničiti na mapiranje IP adresa i portova iz IPv4 privatnog adresnog prostora u IPv4 javni adresni prostor i obrnuto, pri čemu će se podrazumijevati da se na transportnom nivou koristi UDP (*User Datagram Protocol*) protokol. SIP protokol najčešće podrazumeva različite putanje za signalizaciju i za prenos medije. Dok signalizacioni tokovi zavise od signalizacionih čvorova u mreži, medija tokovi se uspostavljaju najčešće direktno, najkraćim putem između korisnika koji su u komunikaciji. Ovaj protokol koristi IP adresu i broj porta, koji se prenose u tijelu SIP poruka korišćenjem SDP (*Session Description Protocol*) protokola [2], [3]. Tu postoji problem da adrese i portovi nisu iskoristivi ako se svi *peer* entiteti ne nalaze sa iste strane NAT-a, odnosno u istom adresnom prostoru. Mehanizmi kao što su STUN (*Session Traversal Utilities for NAT*) [4], TURN (*Traversal Using Relays around NAT*) i ICE (*Interactive Connectivity Establishment*) ne mogu egzistirati u prisustvu SIP protokola, a da ne naruše njegovu *end-to-end* prirodu. SBC je jedan od uređaja, čiji skup funkcionalnosti omogućava SIP uređajima pozicioniranim iza NAT-a da ostvare komunikaciju kroz NAT [5]-[7]. Ovi mehanizmi su dizajnirani da rade u *end-to-end* komunikaciji i oslanjaju se na tehniku dinamičkog otkrivanja adresa i portova zvanu "*latching*". Termin koji se često koristi za ovaj mehanizam jeste HNT (*Hosted NAT Traversal*). HNT funkcionalnost je najčešće realizovana u SBC-u. U primjenjenom scenariju, medija između dva učesnika u komunikaciji ne ide najkraćim putem, nego preko SBC-a kako bi se osigurala kontrola toka. Ovo unosi neke očigledne probleme kao što su smanjenje skalabilnosti sistema i povećanje kašnjenja. Međutim, postoje brojni razlozi za usmjeravanje medija tokova preko SBC-a (sl. 1). U scenariju u kom administrator mora biti siguran da se medija isporučuje po unaprijed dogovorenim uslovima (SLA – *Service Level Agreement*), on može uticati na optimizaciju toka. Takođe, postoje i brojni sigurnosni aspekti zbog kojih se medija tokovi usmjeravaju preko SBC-a, a ne najkraćim putem između dva učesnika u komunikaciji.



Slika 1. Medija i signalizacioni tok u prisustvu SBC-a

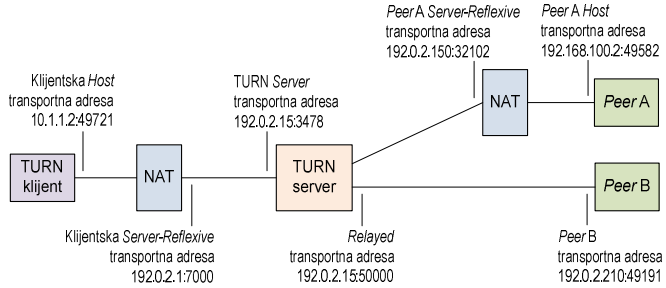
III. PREGLED OPERACIJA I OSNOVNE KARAKTERISTIKE TURN PROTOKOLA

U tipičnoj mrežnoj konfiguraciji TURN klijent je konektovan na strani privatne mreže (u domenu privatnih IPv4 adresa), a preko NAT-a (jednog ili više) i na javnu, Internet mrežu (domen javnih IPv4 adresa). Na strani javne mreže nalazi se TURN server, a dalje iza njega su locirani drugi *peer* entiteti sa kojima klijent želi uspostaviti vezu. Klijent koristi TURN server za prosleđivanje i primanje paketa ka drugim *peer* entitetima i od njih. Sl. 2 prikazuje tipičan raspored mrežnih elemenata u TURN server okruženju [8]. Klijent komunicira sa serverom preko svoje (host) adrese (IP adrese i porta). Ova adresa se naziva transportna adresa. Klijent šalje TURN poruke sa svoje transportne adrese na transportnu adresu TURN servera. Tokom razmjene poruka sa TURN serverom klijent će naučiti njegovu transportnu adresu. Ovu adresu TURN servera može da koristi veći broj klijenata istovremeno. Pošto se klijent nalazi iza NAT-a, TURN server pakete primljene od klijenta vidi kao da su poslati od NAT-a, jer stižu sa NAT transportnom adresom. Ova adresa je poznata kao *server-reflexive* transportna adresa. Paketi koje server šalje na klijentsku *server-reflexive* transportnu adresu će NAT proslijediti na host transportnu adresu klijenta. Klijent koristi TURN komande za alokaciju servera. Alokacija se odnosi na strukturu podataka na serveru. Ova struktura podataka sadrži, između ostalog, *relayed* transportnu adresu servera. *Relayed* transportna adresa servera je transportna adresa servera koju *peer* entiteti koriste za slanje paketa koje će server proslijediti ka klijentu. Transportna adresa jednoznačno određuje lokaciju. Po izvršenoj alokaciji servera, klijenti mogu slati serveru aplikacione podatke zajedno sa podacima koje određuju kom *peer* entitetu se oni šalju i server će ih proslijediti odgovarajućem *peer* entitetu. Klijent šalje serveru aplikacione podatke u okviru TURN poruka. Na serveru se podaci izdvajaju i šalju klijentu u formi UDP datagrama. U suprotnom pravcu *peer* entitet će slati aplikacione podatke ka *relayed* transportnoj adresi servera u formi UDP datagrama. Server će izvršiti enkapsulaciju primljenog UDP datagrama u TURN poruku i poslati ga ka klijentu zajedno sa podacima o *peer* entitetu koji je podatke izvorno poslao. TURN poruka sadrži podatke sa kojih *peer* klijenata je stigla. Klijent može koristiti jednu alokaciju za komunikaciju sa više *peer* entiteta. Kada je *peer* iza NAT-a, tada klijent mora identifikovati *peer* entitet koristeći *server-reflexive* transportnu adresu, a ne direktno host transportnu adresu. Npr. za slanje aplikacionih podataka ka *Peer* A entitetu u primjeru na sl. 2, klijent mora specificirati 192.0.2.150:32102 (*Peer* A *server-reflexive* transportna adresa), a ne 192.168.100.2:49582 (host transportna adresa *Peer* A). Svaka alokacija na serveru pripada jednom klijentu i ima jednu transportnu adresu koju koristi samo ta alokacija. Tako kad paketi stignu na *relayed* transportnu adresu na serveru, server zna kom klijentu su podaci namijenjeni. Jedan klijent može istovremeno imati više alokacija na serveru.

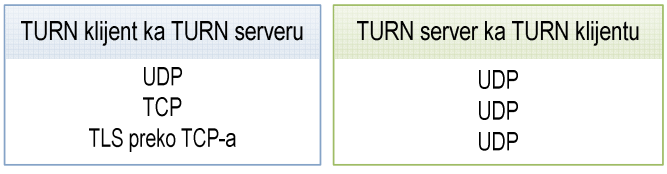
TURN server između *peer* entiteta i servera najčešće koristi UDP kao transportni protokol [8], međutim postoji mogućnost korišćenja i TCP (*Transmission Control Protocol*) ili TLS (*Transport Layer Security*) protokola preko TCP-a za prenos TURN poruka između klijenta i servera (Sl. 3).

Ako se koristi TCP ili TLS preko TCP-a između klijenta i servera, onda će server vršiti konverziju tih transportnih protokola u UDP prilikom prosleđivanja ka i od *peer* entiteta. TURN podržava TCP između klijenta i servera, jer su neki *firewall*-ovi konfigurisani tako da u potpunosti blokiraju UDP prenos, ali ne i TCP.

Nedostatak ovog koncepta je činjenica da TURN server zahtijeva značajan propusni opseg prema javnoj mreži, što sa sobom nosi i odgovarajuće troškove. Optimalno rješenje bi bilo korišćenje TURN servera samo u situacijama u kojim nije moguće pronaći direktan put između odgovarajućih *peer* entiteta koji pokušavaju uspostaviti vezu i u slučajevima kada postoji potreba za kontrolom toka medije i njenom optimizacijom (SLA). Aplikacije koje na transportnom nivou koriste UDP protokol, i uz to idu preko TURN servera, moraju imati u sebi rješenje za izbjegavanje fragmentacije IP paketa, gdje god je to moguće. Aplikacije koje koriste TCP protokol na transportnom nivou mogu ignorisati ovaj problem jer je izbjegavanje fragmentacije dio TCP standarda. Aplikacije pokrenute na TURN klijentu i *peer* entitetu moraju raspolagati jednom od ove dve navedene mogućnosti izbjegavanja fragmentacije.



Slika 2. Raspored mrežnih elemenata u TURN server okruženju



Slika 3. Transportni protokoli kod TURN servera

Prvi pristup je izbjegavanje slanja velike količine aplikacionih podataka u TURN porukama (UDP datagramima) razmijenjenim između klijenta i *peer* entiteta. Ovaj pristup se često koristi kod VoIP aplikacija i zasniva se na preporuci da IP paketi manji od 576 bajta ne bi trebali biti fragmentirani. Količina aplikacionih podataka koja će se prenositi preko TURN servera zavisi i od toga koji protokoli se koriste na transportnom nivou (TCP, UDP ili TLS), da li se koriste *ChannelData* poruke ili *Send/Data* indikatori, koji dodatni atributi su uključeni (npr. *do not-fragment* bit), da li je MTU (*Maximum Transmit Unit*) redukovan negdje ranije tokom prenosa, da li je tunelovanje korišteno tokom prenosa i slično. Preporučuje se slanje maksimalno 500 bajta podataka po pojedinačnoj TURN poruci na relaciji klijent-server ili po UDP datagramu na relaciji *peer* entitet-server. Na ovim relacijama se generalno izbjegava IP fragmentacija. Da bi se dodatno smanjile šanse za fragmentaciju preporučuje se da klijent koristi *ChannelData* poruke, kada se prenose veće količine podataka, jer je *overhed ChannelData* poruka manji nego kod *Send* i *Data* indikatora.

Drugi način da klijent i *peer* entitet izbjegnu fragmentaciju je korišćenje *Path MTU discovery* algoritma za određivanje maksimalne količine aplikacionih podataka bez fragmentacije.

IV. PRINCIP FUNKCIONISANJA I OSNOVNE KARAKTERISTIKE *MEDIA LATCHING* MEHANIZMA

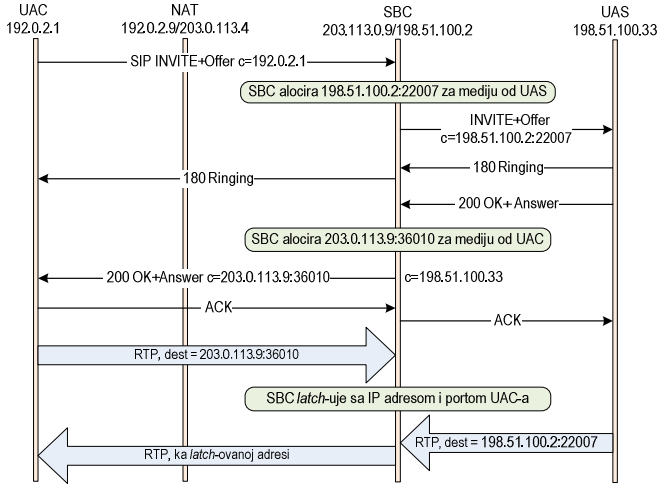
UA (*User Agent*) pozicioniran iza NAT-a šalje medija pakete iz privatnog adresnog prostora. Da bi paket prešao u javnu mrežu, adresa mora biti mapirana u javnu. UA, tipično nije svjestan ove promjene, tako da on u okviru signalizacionih procedura objavljuje svoju stvarnu privatnu adresu i port. U ovoj situaciji u kojoj SBC obavlja funkciju HNT-a, on prima medija paket sa privatne IP adrese i porta i ne zna na koju adresu i port da ga proslijedi, jer se on nalazi u privatnom adresnom prostoru. Zbog toga će se za prenos medije koristiti mehanizam zvani *latching*. Istorijski gledano *latching* se odnosi samo na proces kojim SBC pridružuje izvorišnu adresu i port IP paketa, koje je poslao udaljeni UA, u svrhu rezervisanja IP adrese i porta za slanje medije u suprotnom pravcu. *Symetric-latching* je mehanizam u kom se rezervisana adresa i port koriste za slanje medije u suprotnom pravcu ka UA. *Media Latching* (RTP *latching,* RTP – *Real-time Transport Protocol*) se događa u procesu razmjene SIP signalizacionih poruka, tačnije u procesu slanja ponude i prijema odgovora na dostavljenu ponudu, pri čemu se ignoriše SDP sadržaj SIP poruke.

Latching mehanizam funkcioniše na sljedeći način [9]:

- Po prijemu ponude od NAT-ovanog UA (klijentski UA koji se nalazi iza NAT-a, UAC), SBC lociran u javnom domenu (Internet) izdvojiće par IP adresa:port koji će služiti za prosleđivanje medije (*media relay*). Par adresa:port će potom biti objavljen drugom UA-u (serverski UA, UAS) i biće korišten za sva slanja medije ka UAC.
- Nakon prijema odgovora od UAS-a, na dostavljenu ponudu, SBC će dodijeliti drugi par adresa:port za *media relaying*. U odgovoru UAC-u, SBC će zamijeniti adresu i port sa ovim drugim parom adresa:port. Na ovaj način će UAC poslati mediju ka SBC-u.
- *Media relay* prima medija pakete na izdvojeni port i koristi njihovu izvorišnu adresu i port kao odredište za prosleđivanje medija paketa u suprotnom smijeru. Drugim riječima SBC je izvorišni UA par IP adresa:port "zaključao/zabravio" (*latch*) i na njih prosleđuje sve medija pakete namijenjene tom UA.
- Na ovaj način, kada UAC šalje medija tok ka SBC-u, medija paketi će biti primljeni na drugi par IP adresa:port. Par izvorišna IP adresa:port će pripadati javnom interfejsu NAT-a, i sav medija saobraćaj koji se bude prosleđivao nazad ka UAC će ići ovim putem.
- Slično prethodnom primjeru i UAS će hvatati izvorišnu IP adresu:port dolaznog medija paketa i na tu adresu slati medija saobraćaj u suprotnom smijeru.

Latching se obično radi jedanput po *peer* entitetu i nije dozvoljeno mijenjanje (*re-latching*) prije nego što se razmijene nova ponuda i primi odgovor na nju, recimo u narednom pozivu. Razlozi za ovo su uglavnom sigurnosni, jer kad krene jedna sesija ne očekuje se striming novih sesija sa drugih portova bez prethodno izvršenog dogovaranja kroz slanje ponuda i primanje odgovora za svaku sesiju. Promjena može ukazati na pokušaj napada u toku ove sesije. U nekim slučajevima promjena porta može biti uzrokovana ponovnim mapiranjem porta u NAT-u, koji se nalazi između SBC-a i UA. Naprednije verzije SBC-a mogu dozvoliti neki stepen *re-latching*-a pri čemu razmatraju sigurnosne aspekte i po potrebi reaguju na njih.

Sl. 4 daje prikaz *latching*-a gdje je HNT funkcionalno obezbijeđen kroz SBC koji je vezan na dve mreže: 203.0.113/24 prema UAC-u i 198.51.100/24 prema UAS-u.



Slika 4. *Latching* u uslovima u kojim je HNT funkcionalnost realizovana kroz SBC [9]

Opšti problemi koji se javljaju kod NAT *traversal* funkcionalnosti u radu sa SIP protokolom jesu:

- Adresa i broj porta kodovani u SDP tijelu SIP poruke, koji su reprezent NAT-ovanih UA-a nisu dostižni kroz Internet, jer se nalaze u privatnom adresnom prostoru.
- NAT primjenjuje takav sistem filtriranja na dolazne pakete da UA može primiti paket sa neke vanjske adrese i porta samo ako je UA prethodno poslao paket na tu adresu.

Obično se prosleđivanje medije i signalizacije preko SBC-a vrši preko iste IP adrese, pri čemu se brojevi portova dodjeljuju dinamički. Strategija dodjele adresa zavisi od vrste servisa. *Latching* nije isključivo serverski posao. Klijenti takođe mogu koristiti *latching* kada su konfigurisani sa javnom IP adresom i kada su kontaktirani od NAT-ovanog klijenta. Da bi *latching* mogao ispravno da funkcioniše, UA iza NAT-a treba da podržava simetričan RTP, tj. treba da koristi iste portove za slanje i prijem paketa. UA-i treba da osiguraju da je moguće slanje paketa nezavisno i bez čekanja da se prvo prime paketi. Teoretski je moguće da neki UAS neće prvi poslati pakete, npr. ako SIP sesija počinje iz neaktivnog SDP moda, inicirana od UA pozicioniranog iza NAT-a. U praksi se međutim ovo nikada ne događa iz očiglednih razloga, taj medija pravac bi bio problematičan ako bi SBC imao indikaciju neaktivnog SDP moda pri slanju SDP sadržaja ka UA. SBC sa HNT funkcionalnošću će uvijek biti konfigurisan da izbjegne ovu situaciju.

Da bi *latching* radio ispravno, *media relay*-i treba da počnu prijem medije prije slanja. U tim uslovima moguće je da se dogode „mrtve petlje". Ovo se može dogoditi kada su UAC i UAS koji učestvuju u sesiji, konektovani na različite SBC-ove koji podržavaju HNT. U ovom slučaju bi svaki od *media relay*-a pod kontrolom svog signalizacionog servera mogao ostati u čekanju onog drugog da inicira striming. Da bi se ovo izbjeglo *media relay*-i povremeno iniciraju striming prema parovima adresa:port obezbijeđenim u procesu razmjene ponuda i odgovora, pre prijema bilo kakvog dolaznog saobraćaja. Ukoliko je entitet prema kom se uspostavlja striming drugi HNT server koji obezbjeđuje prosleđivanje medije preko javne adrese i porta, ovaj rani striming će uspješno startovati. Većina SBC-ova podržava samo UDP *latching*, a neki podržavaju i TCP *latching*. TCP *Media Latching* je znatno komplikovaniji. HNT i *latching*, svaki za sebe i pojedinačno, rade pouzdano, ali postoje i neka ograničenja i upozorenja u radu sa njima. Jedno od njih je da UA nije svjestan procesa NAT-ovanja niti *latching*-a. To onemogućava zajednički rad sa protokolima kao što je npr. ICE [10]. Promjena adresa u procesu ponude i odgovora u potpunosti onemogućava rad sa ICE-om, jer je u suprotnosti s osnovnim pravilima na kojima je ICE zasnovan.

V. SIGURNOSNI ASPEKTI TURN PROTOKOLA I *MEDIA LATHCING* MEHANIZMA

Po pitanju sigurnosnih prijetnji postoje određene razlike između TURN servera i *Media Latching* mehanizma [8], [9]. Ove razlike se prvenstveno odnose na slabosti u pojedinim fazama procedura i pogodnostima za eventualne napade. Međutim, u oba slučaja se kao preventiva za većinu mogućih napada koristi mehanizam autentifikacije. TURN server, odnosno SBC, ima mogućnost da odredi identitet pošiljaoca poruke i da na osnovu identiteta utvrdi da li pošiljalac poruke ima potrebnu dozvolu. Svi pristigli paketi sa određene lokacije, odnosno sa određenim identitetom koji nije prisutan u listi dozvola će biti odbačeni. Zaštitu tajnosti podataka je najefikasnije vršiti na aplikacionom sloju. Ako zaštita tajnosti podataka predstavlja prioritet onda treba koristiti protokole koji vrše kriptovanje podataka. Za osiguravanje povjerljivosti prenosa medije u realnom vremenu se koristi SRTP (*Secure RTP*) protokol. Mehanizam razmjene ključeva bi zaštitio mediju u fazi prenosa. Na transportnom nivou je moguće koristiti TLS protokol. Kombinacija zaštite na aplikativnom i transportnom sloju bi bila efikasnija, pri čemu treba imati u vidu da će to dovesti do dodatne degradacije performansi u prenosu, odnosno unijet će dodatno kašnjenje.

VI. ZAKLJUČAK

Praveći uporednu analizu rada TURN protokola i *Media Latching* mehanizma u SIP mrežnom okruženju može se zaključiti da je TURN server značajno složeniji i zahtjevniji sistem. TURN server zahtijeva instalaciju TURN klijenata na krajnjim korisničkim uređajima, pored već postojećih SIP klijenata, za razliku od *Media Latching* mehanizma koji to ne traži. Ova činjenica je izraženija ukoliko je broj NAT-ovanih korisnika veći. TURN server u svom radu sadrži veći broj procedura potrebnih za rezervaciju resursa za prosleđivanje medije, po pojedinačnoj uspostavljenoj vezi, u odnosu na *Media Latching* mehanizam. Upotreba *ChannelData* poruka, odnosno *Send* i *Data* indikatora unosi dodatni *overhead*, odnosno generiše dodatne količine saobraćaja, zbog čega je potreban i dodatni propusni opseg. Pri prenosu veće količine podataka ova pojava je izraženija. Kod *Media Latching* mehanizma procedure "kačenja" para IP adrese i porta iz dolaznog medija strima i njegova rezervacija za slanje medije u suprotnom pravcu na istu adresu i port se obavlja u okviru SIP procedura, za šta su dovoljni postojeći SIP klijenti. Tokom ovog postupka se SDP sadržaj SIP poruka ignoriše. U uslovima funkcionalne integracije TURN servera ili *Media Latching* mehanizma u SBC mrežnom elementu ovo znači da će opterećenje procesora pri istom broju proslijeđenih sesija biti veće ako prosleđivanje vrši TURN server nego ako se to radi *Media Latching* mehanizmom.

Preporuka je da se zbog jednostavnosti implementacije i efikasnosti samog mehanizma, kao i uticaja na performanse sistema i prenosne karakteristike signala (unosi manja dodatna kašnjenja) u praksi daje prednost *Media Latching* mehanizmu nad TURN protokolom.

LITERATURA

[1] D. Nemec, D. Vukobratović, V. Crnojević, Č. Stefanović, "Tehnologija VoIP sistema", Katedra za telekomunikacije i obradu signala, Fakultet tehničkih nauka, Novi Sad, 2006.

[2] J. Rosenberg, H. Schulzrinne, G. Camarillo, A. Johnston, J. Peterson, R. Sparks, M. Handley, E. Schooler, "SIP: Session Initiation Protocol", RFC 3261, IETF, 2002.

[3] M. Handley, V. Jacobson, C. Perkins, "SDP: Session Description Protocol", RFC 4566, IETF, 2006.

[4] J. Rosenberg, J. Weinberger, C. Huitema, R. Mahy, "STUN – Simple Traversal of User Datagram Protocol (UDP) Through Network Address Translators (NATs)", RFC 3489, IETF, 2003.

[5] Pat Hurley, "Session Border Controllers for Dummies", 2nd Sonus Special Edition, John Wiley & Sons, Inc. 2013.

[6] Al Sultani Dawood Akdas, "Session Border Controller Roles in IP Multimedia Subsystem", AV Akademikerverlag, 2015.

[7] J. Hautakorpi, G. Camarillo, R. Penfield, A. Hawrylyshen, M. Bhatia, "Requirements from Session Initiation Protocol (SIP) Session Border Control (SBC) Deployments", RFC 5853, IETF, 2010.

[8] R. Mahy, P. Matthews, J. Rosenberg, "Traversal Using Relays around NAT (TURN): Relay Extensions to Session Traversal Utilities for NAT (STUN)", RFC 5766, IETF, 2010.

[9] E. Ivov, H. Kaplan, D. Wing, "Latching: Hosted NAT Traversal (HNT) for Media in Real-Time Communication", RFC 7362, IETF, 2014.

[10] J. Rosenberg, "Interactive Connectivity Establishment (ICE): A Protocol for Network Address Translator (NAT) Traversal for Offer/Answer Protocols", RFC 5245, IETF, 2010.

ABSTRACT

This paper describes NAT traversal problem and its possible solutions using TURN protocols and Media Latching mechanism. A comparative analysis of the TURN protocols and Media Latching mechanism was done, and based on the results, conclusions and recommendations in terms of choice of optimal solutions are derived.

SOLVING NAT TRAVERSAL PROBLEM USING TURN PROTOCOL AND MEDIA LATCHING MECHANISM

Dragan Botić, Dejan Nemec, Siniša Vujčić