SVEUČILIŠTE JOSIPA JURJA STROSSMAYERA U OSIJEKU
FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA I
INFORMACIJSKIH TEHNOLOGIJA OSIJEK

mr.sc. Dražen Tomić

DINAMIČKI ODABIR POSLUŽITELJA VIŠESTRUKE MREŽNE USLUGE PRIMJENOM KOMPOZITNE DNS-METRIKE

DOKTORSKA DISERTACIJA

Osijek, srpanj 2018.

Posvećeno mojim najdražima: obitelji u kojoj sam odrastao i obitelji koju sam zasnovao.

Doktorska disertacija izrađena je na:	Sveučilište Josipa Jurja Strossmayera u Osijeku Fakultet elektrotehnike, računarstva i informacijskih tehnologija Osijek
Mentor:	prof.dr.sc. Drago Žagar
Doktorska disertacija ima:	154 stranice
Disertacija broj:	65

Povjerenstvo za ocjenu doktorske disertacije:

1. Prof.dr.sc. Goran Martinović, Fakultet elektrotehnike, računarstva i informacijskih tehnologija Osijek, predsjednik Povjerenstva

2. Prof.dr.sc. Drago Žagar, Fakultet elektrotehnike, računarstva i informacijskih tehnologija Osijek, mentor

3. Prof.dr.sc. Gordan Ježić, član, Fakultet elektrotehnike i računarstva Sveučilišta u Zagrebu

Povjerenstvo za obranu doktorske disertacije:

1. Prof.dr.sc. Goran Martinović, Fakultet elektrotehnike, računarstva i informacijskih tehnologija Osijek, predsjednik Povjerenstva

2. Prof.dr.sc. Drago Žagar, Fakultet elektrotehnike, računarstva i informacijskih tehnologija Osijek, mentor

3. Prof.dr.sc. Gordan Ježić, član, Fakultet elektrotehnike i računarstva Sveučilišta u Zagrebu

4. Prof.dr.sc. Radoslav Galić, Fakultet elektrotehnike, računarstva i informacijskih tehnologija Osijek, član

5. Izv.prof.dr.sc. Krešimir Grgić, Fakultet elektrotehnike, računarstva i informacijskih tehnologija Osijek, član

Datum obrane doktorske disertacije: 19. srpnja 2018. godine

Sadržaj:

1. UVOD

Zbog sve veće važnosti stalne dostupnosti poslužitelja, odnosno mrežnih usluga koje poslužitelji pružaju svojim korisnicima, i njihovog sve većeg opterećenja, uobičajena je praksa da se za obavljanje funkcije visoko dostupne mrežne usluge koristi dva ili više poslužitelja smještenih na istoj ili različitim lokacijama, spojenih na internet jednostrukim ili višestrukim komunikacijskim linkom [1][2]. Uvođenjem višestrukih poslužitelja za obavljanje iste mrežne funkcije moguće je [3][4]:

- raspoređivati opterećenja između poslužitelja s obzirom na trenutno opterećenje pojedinog poslužitelja i s obzirom na trenutno opterećenje pojedinog pristupnog komunikacijskog linka poslužitelja (eng. *server and link load balancing*), u slučaju da se koriste višestruki komunikacijski linkovi
- omogućiti dostupnost mrežnih usluga u slučaju nedostupnosti pojedinog poslužitelja uzrokovanog greškom na poslužitelju ili mrežnoj putanji između klijenta i poslužitelja (eng. *network and server failure failover*).

Pri tome se smještajem poslužitelja na različite fizičke lokacije i/ili različite pristupne komunikacijske/internetske linkove, osim raspoređivanja opterećenja i omogućavanja dostupnosti mrežnih usluga u slučaju nedostupnosti pojedinog komunikacijskog linka, istovremeno omogućuje optimizacija mrežnog prometa između klijenta i poslužitelja u smislu odabira optimalnog poslužitelja za pojedinog klijenta [5][6]. Ako u takvim slučajevima redundantni poslužitelji imaju različito vrijeme mrežnog odziva (eng. *network response time*) zbog različitog kapaciteta i/ili opterećenja poslužitelja i/ili komunikacijskog linka, klijent može pristupiti poslužitelju koji mu je mrežno najbliži i/ili najmanje opterećen, odnosno koji ima brže vrijeme odgovora na upit.

Za određivanje poslužitelja koji će najbrže odgovoriti na korisnikov zahtjev, uz vrijeme mrežnog odziva, potrebno je uzeti u obzir i još jedan parametar - opterećenje poslužitelja, čime se definiraju dva osnovna parametra koji utječu na vrijeme odgovora mrežne usluge:

- vrijeme mrežnog odziva, kao parametar računalne mreže
- opterećenje poslužitelja, kao sistemski poslužiteljski parametar mrežne usluge

Kako se radi o dinamičkim, vremenski promjenjivim veličinama, za navedene parametre odabira optimalnog poslužitelja višestruko dostupne mrežne usluge potrebno je imati njihove što točnije tj. novije vrijednosti, te algoritam koji će, temeljem dostupnih podataka vremena mrežnog odziva i opterećenja poslužitelja, odrediti koji je poslužitelj optimalan za isporuku mrežne usluge za svakog pojedinog klijenta, uvažavajući pri tome:

- vremenski trenutak u kojem je klijent zatražio pristup mrežnoj usluzi
- činjenicu da svaki klijent koji je zatražio mrežnu uslugu može imati vlastitu, jedinstvenu mrežnu udaljenost prema mrežnoj usluzi.

Algoritam dinamičkog odabira optimalnog poslužitelja višestruko dostupne mrežne usluge pri tome mora biti:

1. otvoren: da omogući primjenu na svim klijentskim i poslužiteljskim operacijskim sustavima
2. univerzalan: da podržava IPv4 i IPv6 protokol
3. jednostavan: da omogući jednostavnu i brzu implementaciju uz minimalne izmjene na postojećoj klijentskoj i sistemskoj infrastrukturi, a bez zahtjeva za izmjenama na mrežnoj infrastrukturi
4. skalabilan: da omogući daljnji razvoj sukladno zahtjevima tehničko-tehnološkog napretka
5. modularan: da se veže na neku od postojećih mrežnih sistemskih usluga kao proširenje postojeće funkcionalnosti
6. siguran: da ne unosi nove sigurnosne probleme u mrežnoj komunikaciji i dostupnosti mrežnih usluga te sigurnosti poslužitelja i klijenata
7. dinamički: da omogućuje dinamičko prikupljanje i promjenu parametara odabira optimalnog poslužitelja višestruko dostupne mrežne usluge
8. klijentsko-poslužiteljski: da mehanizam odabira uključuje i klijentske i poslužiteljske parametre
9. upravljiv: da omogućuje upravljanje rezultatima odabira
10. prilagodljiv: da omogućuje prilagodbu različitim zahtjevima sustava u koji se implementira
11. konfigurabilan: da omogućuje prilagodbu korištenjem konfigurabilnih, a ne zadanih (predefiniranih) parametara
12. robustan: da može ponuditi odgovor u svakoj predvidivoj situaciji

13. opcijski: da ga poslužitelji i mrežni klijenti mogu ali i ne moraju implementirati
14. kompatibilan: da ne isključuje druge metode za optimizaciju pristupu mrežnim resursima
15. jeftin: da ne zahtijeva skupa ulaganja u hardversku i softversku infrastrukturu

1.1. MOTIVACIJA I CILJEVI RADA

Problem upravljanja odabirom poslužitelja višestruko dostupne mrežne usluge uočen je procesu izgradnje informacijskog sustava zasnovanog na primjeni informatičkih tehnologija u Zavodu za informatiku Osijek (ZIO), računskom centru Osječko-baranjske županije. Definiranjem zahtjeva prilikom odabira poslužitelja višestruko dostupne mrežne usluge i usporedbom s postojećim metodama za odabir poslužitelja mrežnih usluga uočen je nedostatak metode koja bi u cijelosti zadovoljila postavljene zahtjeve, a iz čega proizlaze ciljevi ovog istraživanja:

- analiza postojećih metoda za odabir poslužitelja višestruko dostupne mrežne usluge i rješavanja problema nedostupnosti mrežne usluge uslijed nedostupnosti poslužitelja mrežne usluge ili pripadajuće mrežne putanje od klijenta do poslužitelja
- razvoj nove metode za odabir optimalnog poslužitelja višestruko dostupne mrežne usluge, a koja ispunjava sve ranije navedene zahtjeve. Metoda se zasniva na proširenju postojećih DNS [7][8] zapisa o resursima (eng. *resource records* - *RR*) novim RR zapisom. U novom RR zapisu se definiraju parametri opterećenja svakog poslužitelja te mehanizmi za određivanje vremena odziva mrežne usluge pojedinog poslužitelja. Temeljem dobivenih parametara algoritmom za izračun optimalnog mrežnog poslužitelja klijent dobiva listu internetskih (IP) adresa za traženu mrežnu uslugu optimiziranih upravo za tog klijenta, a prema izračunatoj kompozitnoj DNS-metrici. Poslužitelj s najmanjom kompozitnom DNS-metrikom treba omogućiti klijentu najbrži odgovor na postavljeni zahtjev
- analitički izračun očekivanog vremena odgovora poslužitelja mrežne usluge na postavljeni zahtjev i povezivanje vremena odgovora sa izračunatom kompozitnom DNS-metrikom kod primjene TCP protokola

- primjena nove metode za:
    - dinamički odabir poslužitelja višestruko dostupne mrežne usluge kompozitnom DNS-metrikom s ciljem odabira poslužitelja mrežne usluge koji će najbrže odgovoriti na upit klijenta
    - brzo utvrđivanje nedostupnosti poslužitelja višestruko dostupne mrežne usluge.

Cilj razvoja metode za dinamički odabir poslužitelja višestruko dostupne mrežne usluge primjenom kompozitne DNS-metrike je omogućiti:

- klijentu: brži odgovor mrežne usluge i brži prijenos podataka, a time i veće zadovoljstvo klijenta uslugom, pogotovo u situacijama gdje je usluga mrežno heterogena, kao npr. http usluga gdje web stanica u sebi sadržava prosječno 5-10 vanjskih poveznica (poveznice na oglase, društvene mreže, praćenje web lokacija, ostale vanjske sadržaje) koje se dinamički generiraju u trenutku učitavanja stranice na klijentskoj strani, te u situacijama gdje se prenosi veća količina podataka
- davatelju mrežnih usluga: poboljšano rješavanje problema nedostupnosti poslužitelja, upravljanje opterećenjem poslužitelja i internetskih linkova
- temeljnoj mrežnoj infrastrukturi / ISP-ovima: brži protok podataka i smanjenje mogućnosti pojave zagušenja na mreži.

1.2. ORGANIZACIJA RADA

Rad je organiziran kako slijedi: u 2. poglavlju prikazani su pregled i analiza područja istraživanja kroz kronološki pregled znanstvenih radova iz područja istraživanja i pregled postojećih rješenja s analizom. U 3. poglavlju opisana je nova metoda dinamičkog odabira poslužitelja višestruko dostupne mrežne usluge – DSS (eng. *Dynamic Server Selection*) kroz prikaz ideje metode, postavljenih zahtjeva za metodu te opis formata i funkcionalnosti DSS metode. Poglavlje 4 opisuje analitički izračun vremena odgovora poslužitelja i njegovog povezivanja s kompozitnom DNS-metrikom, a poglavlje 5 implementaciju metode i analizu učinkovitosti metode u realnom okruženju u postupcima određivanja optimalnog poslužitelja za pristup višestruko dostupnoj mrežnoj usluzi te brzom utvrđivanju nedostupnosti poslužitelja. Poglavlje 6 sadrži zaključak rada.

2. DINAMIČKI ODABIR POSLUŽITELJA VIŠESTRUKO DOSTUPNE MREŽNE USLUGE

Sve veća potreba za osiguranjem visoke dostupnosti mrežnih usluga, stalno povećanje opterećenosti mrežnih poslužitelja te potreba za topologijskim približavanjem mrežne usluge korisniku, dovode do sve češćeg postavljanja višestrukih (redundantnih) poslužitelja za pružanje istovrsne mrežne usluge korisnicima. Kako u takvom sustavu istu mrežnu uslugu korisniku može ponuditi dva ili više poslužitelja potrebno je ustanoviti mehanizam odabira poslužitelja koji će istovremeno osigurati raspodjelu opterećenja resursa i komunikacijskih linkova na poslužiteljskoj strani, a korisniku najbrže vrijeme odgovora mrežne usluge. Statičke metode, kao što su broj skokova ili geografska udaljenost između poslužitelja i klijenta, u početku značajno korištene zbog svoje jednostavnosti, ubrzo su zamijenjene kompleksnijim, dinamičkim metodama (kao što je npr. vrijeme odziva poslužitelja mrežne usluge), koje omogućuju bolje ispunjenje postavljenih ciljeva ravnomjernog opterećenja resursa poslužitelja i skraćivanja vremena odgovora mrežne usluge.

2.1. TEORIJSKA RAZMATRANJA O STATIČKOM I DINAMIČKOM ODABIRU POSLUŽITELJA

Dinamički odabir poslužitelja višestruko dostupne mrežne usluge obrađen je u literaturi, znanstvenim i stručnim radovima korištenjem vrlo različitih pristupa problematici i raznolikim predloženim rješenjima.

Autori u [9] po prvi puta uvode dinamički odabir najbližeg poslužitelja, za razliku od prethodnih radova gdje je odabir poslužitelja bio određen isključivo statički, pri čemu nije potrebno poznavati lokaciju poslužitelja i mrežnu topologiju od klijenta do poslužitelja. Definiraju se osnovne metrike za mjerenje mrežne udaljenosti: broj skokova (statička metrika) i vrijeme kružnog putovanja paketa (dinamička metrika). U istraživanju je potvrđeno da su dinamičke metode odabira poslužitelja dominantne u odnosu na statičke, pri čemu su korištene 4 metode odabira poslužitelja:

- Statička, temeljena na geografskoj udaljenosti (eng. *Geographical*)
- Statička, temeljena na broju skokova (eng. *Hops*)

- Dinamička, temeljena na slučajnom odabiru poslužitelja (eng. *Random*)
- Dinamička, temeljena na mjerenju vremena kružnog putovanja (eng. *Round Trip Time* - RTT)

U zaključku se navodi da daljnji razvoj dinamičkih metoda odabira poslužitelja zahtijeva uključivanje informacija o opterećenju poslužitelja kao i mjerenja raspoložive propusnosti prema poslužiteljima.

U [10] autori definiraju cilj dinamičkog odabira poslužitelja: isporuku usluge u najkraćem vremenu, a opterećenje poslužitelja i kašnjenje zbog korištenja sporih putova kao osnovne razloge za uvođenje repliciranih mrežnih usluga. Utvrđuju da dinamički odabir poslužitelja distribuirane, višestruko dostupne mrežne usluge smanjuje vrijeme odgovora do 50% u odnosu na statički odabir. Predlažu protokol za dinamički odabir poslužitelja koji ne povećava mrežni promet više od 1%, pri čemu veći dokumenti mogu imati preciznija mjerenja mrežnih uvjeta. Osnovni problem metode je jedini postavljeni uvjet metode, a to je da klijent ima listu IP adresa poslužitelja koji pružaju određenu mrežnu uslugu. Definira se raspoloživa propusnost koja je uvjetovana propusnošću najsporijeg linka (eng. *bottleneck*) i postojanjem zagušenja na mreži (eng. *congestion*). Autori zaključuju da dodatno vrijeme potrebno za mjerenje mrežnih uvjeta često rezultira boljim performansama, da je RTT izrazito dinamički parametar i da najviše ovisi o zagušenju računalne mreže (opterećenje poslužitelja se ne razmatra) te da više RTT mjerenja povećava korelaciju sa raspoloživom propusnošću i može zamijeniti dodatna mjerenja propusnosti.

Autori u [11] utvrđuju da kod odabira najbližeg poslužitelja njegova geografska blizina ne znači nužno i njegovu mrežnu blizinu kao i najmanje opterećeni poslužitelj, te da korištenje DNS metode kružnog dodjeljivanja (eng. *round-robin*[1]*)* nije dovoljno precizno. Definiraju da su performanse poslužitelja funkcija opterećenja poslužitelja i putanje između poslužitelja i klijenta. Predlaže se korištenje najbliže zajedničke adrese (eng. *anycast*[2]*)* na aplikacijskoj razini (prvenstveno za HTTP uslugu) temeljem baze metrike razlučitelja najbliže zajedničke adrese (eng. *anycast resolvera)* koji sadrži podatke o poslužiteljima i njihovim linkovima pri čemu metriku čine:

[1] *Round-robin* - algoritam za vremensko raspoređivanje procesa/resursa pri čemu je svakom procesu/resursu dodijeljen jednak dio vremena po kružnom principu dodjele, bez davanja prioriteta

[2] *Anycast* - metodologija mrežnog adresiranja i usmjeravanja u kojoj su datagrami jednog pošiljatelja usmjereni topologijski najbližoj točki u grupi potencijalnih primatelja

- praćenje opterećenja poslužitelja i slanje razlučitelju „*Server Push*“ metodom
- procjena performansi koje klijent može očekivati periodičkim upitima „*Client Probe*“ metodom

Za efikasan rad metoda zahtjeva postavljanje što više razlučitelja koji će biti što je moguće bliže klijentima.

U [12] se pokazuje da prikupljanje dodatnih podataka o poslužiteljima i mrežnim putanjama prije odabira poslužitelja značajno unapređuje vrijeme odgovora poslužitelja, odnosno da se isplati ulagati u dodatna mjerenja za postizanje bržeg odziva mrežne usluge. Metrika koja se koristi je višestruko mjerenje RTT-a, predlaže se uvođenje opterećenja poslužitelja kao dodatne metrike. Nakon testiranja učitavanja nepostojeće stranice, kao metode za određivanje opterećenja, zaključak je da je bolje da poslužitelj sam održava varijablu opterećenja koja bi se mogla lagano očitati. Također je zaključeno kako se povećava veličina mrežnih usluga na internetu i potrebno je razvijati nove metode za smanjenje kašnjenja i povećanje propusnosti na mreži.

Za rješavanje problema distribucije zahtjeva između redundantnih poslužitelja u [13] se predlaže uvođenje sustava pred-procesiranja DNS upita (eng. *query preprocessor*) vezanog za DNS koji radi kao DNS *proxy*[3] i može raditi i na klijentskoj i na poslužiteljskoj strani. Za rad na klijentskoj strani klijent mora koristi DNS *proxy* umjesto dotadašnjeg DNS poslužitelja i svi hostovi koji koriste predmetnu metodu moraju biti upisani u DNS *proxy*. Kriteriji za odabir poslužitelja na klijentskoj strani je RTT koji se periodički mjeri za definirane hostove korištenjem testnih i uslužnih paketa (ICMP je često filtriran na uređajima mrežne putanje). Za određivanje vrste usluge koristi se naziv hosta (npr. za *www.example.com* „www“ definira HTTP uslugu) jer nema općenite metode za određivanje vrste usluge temeljem DNS-a. Utjecaj uvođenja *proxy* DNS-a je dodatno prosječno kašnjenje od 9,1 ms.

Nastavno na prethodno istraživanje, u [14] se za korištenje DNS-a kao metode za odabir poslužitelja definiraju zahtjevi mehanizma odabira poslužitelja: transparentnost prema korisnicima, neovisnost o usluzi, neovisnost implementacije, skalabilnost, fleksibilnost

[3] DNS *proxy* – prima DNS upite od klijenata i prosljeđuje ih DNS poslužitelju, pri čemu može privremeno pohranjivati DNS zapise

metode odabira poslužitelja i neovisnost o davatelju usluge. Promatraju se postojeće metode: korisnički odabir i DNS kružno dodjeljivanje. Ukazuje se na nužnost uvođenja metode odabira poslužitelja na klijentskoj strani, pri čemu je problem prikupljanje poslužiteljskih parametara, kao što je opterećenje CPU-a. Definiraju se moguće politike odabira poslužitelja: najkraći RTT između klijenta i poslužitelja, poslužitelj s najvećom propusnošću, poslužitelj s najmanjim CPU opterećenjem, najkraća putanja usmjeravanja između poslužitelja i klijenta i odluka administratora.

U [15] autori zaključuju da je DNS jednostavna metoda za usmjeravanje klijenata prema najbližem poslužitelju, pri čemu se ne zahtijeva promjena postojećih mrežnih protokola. Navode da su druge metode (*„Application layer redirection“, „Application-specific communication protocols“, „Routing protocol modification“*) često vrlo kompleksne ili ograničene u funkcioniranju, i da se na poslužiteljskoj strani često koriste komercijalne DNS-bazirane tehnike za raspoređivanje opterećenja (*„Cisco Distributed Director“, „F53/DNS“, „Nortel/Alteon WebOS“*). Mnogi klijenti i njihovi DNS poslužitelji su topološki udaljeni (prosječno 8 skokova) pa je mjerenje kašnjenja između poslužitelja usluge i DNS poslužitelja loš indikator za procjenu kašnjenja kod klijenta. Navode da podešavanje TTL vrijednosti DNS RR-ova može značajno smanjiti kašnjenje kod klijenata pri čemu se ne povećava značajno dodatni UDP DNS promet. Mnoge web stranice u sebi imaju ugrađene objekte koji mogu zahtijevati dodatne DNS upite, prosječno 14 (median 5), a za popularne indeksne stranice i 35 (median 25) objekata po stranici. Dodatni problem je što neki stari BIND DNS poslužitelji ne poštuju postavljene male TTL vrijednosti. Prijedlog rješenja za rješavanje topološke udaljenosti između DNS klijenta i poslužitelja za određivanje optimalnog DNS odgovora autoritativnog DNS-a je modifikacija DNS protokola zapisom koji će nositi dodatne informacije kojim će se identificirati klijent koji je postavio upit. Predlaže se modifikacija DNS-a u formi novog DNS RR-a „*CA*“ (eng. *Client Address*) u dodatnoj sekciji poruke, koji se može inkrementalno primijeniti, pri čemu DNS poslužitelji koji ne poznaju CA RR ga jednostavno ignoriraju.

U [16] se razrađuje ideja iz [14] DNS-baziranog mehanizma, *proxy* DNS-a. Kao mrežna informacija između klijenta i poslužitelja se koristi RTT upotrebom ICMP protokola. Navodi se da upotreba metode koja koristi mrežne informacije, u usporedbi sa jednostavnim metodama kao što su korisnički odabir i DNS kružno dodjeljivanje, doprinosi boljem odabiru

poslužitelja te da se geografska i mrežna udaljenost ne moraju podudarati ako se radi o različitim autonomnim sustavima.

Autori u [17] uvode DNS-baziranu shemu odabira poslužitelja u CDN[4]-u pri čemu su osnovni zahtjevi za shemu opstojnost i jednostavnost. Da bi se izbjegao dodatni mrežni promet TCP-a za DNS se koristi samo UDP. Uvodi se pojam "od klika do prikaza" (eng. *Click-to-display*) kašnjenja kao korisnička metrika za ocjenu kvalitete weba i definiraju se metode za povećanje kvalitete weba: privremena pohrana web stranica (eng. *web caching*) (klijentski orijentirano) i distribucija sadržaja (eng. *content distribution*) (orijentirano prema davatelju usluge), pri čemu je odabir poslužitelja koji će klijent kontaktirati na poslužiteljskoj strani. Analiziraju se metode odabira poslužitelja: DNS metoda kružnog dodjeljivanja ne balansira efikasno zahtjeve, geografska distribucija replika ne jamči odabir najboljeg poslužitelja. Predlaže se korištenje DNS infrastrukture postavljanjem lokalnih DNS poslužitelja što bliže klijentima pri čemu je prednost DNS-baziranog odabira poslužitelja što koristi postojeću DNS infrastrukturu, a što ju čini odmah primjenjivom na internetu. Nedostatak DNS-baziranog odabira poslužitelja je što neke aplikacije tretiraju dobivene IP adrese različito, npr. ne poštuju TTL IP adrese te što nema oznake protokola koji se koristi pa se teško mogu razlikovati različiti tipovi usluga. Dodatni problem je što samo oko 64% svih korisnika koristi lokalne DNS poslužitelje (unutar istog autonomnog sustava) a samo oko 16% koristi lokalne DNS poslužitelje u istom mrežnom klasteru. CDN radi selekciju poslužitelja mrežne usluge na strani CDN DNS poslužitelja, idealno bi bilo da se selekcija poslužitelja radi na klijentskoj strani jer samo klijent može reći koji poslužitelj je najbolji temeljem njegovih vlastitih kriterija.

Dinamička klijentska metoda odabira poslužitelja temeljem QoS-a prikazana je u [18]. Navodi se da odabir na klijentskoj strani ima problem određivanja opterećenja poslužitelja ali se najbolji poslužitelj iz klijentske perspektive može odabrati korištenjem QoS-a. Autori rade podjelu metoda odabira poslužitelja na statičke (*hops*), dinamičke (RTT) i statističke (QoS statistički podaci) koje mogu biti aktivne i pasivne. Predložena metoda zahtijeva postavljanje QoS agenata na mrežu i selektira fiksni broj poslužiteljskih kandidata baziranih na QoS povijesnim podacima prema danu u tjednu i vremenskom periodu.

[4] CDN – eng. *Content Delivery Network* ili eng. *Content Distribution Network* je veliki distribuirani sustav poslužitelja u višestrukim podatkovnim centrima diljem interneta. Cilj CDN-a je pružanje usluge isporuke sadržaja korisnicima po principima visoke dostupnosti i visokih performansi.

Autori u [19] navode da bi odgovarajući poslužitelj trebao biti odabran uzimajući u obzir procijenjenu lokaciju, izmjeren RTT i oglašeno opterećenje poslužitelja. Kako se u praksi teško mogu dobiti navedeni parametri predlaže se selekcija poslužitelja temeljem parcijalnog RTT-a i povijesnih informacija o opterećenju poslužitelja, a što predstavlja poslužiteljsku metodu koja pokušava aproksimirati korisnički pogled. Navodi se da nije jasno kako sintetizirati opterećenje poslužitelja i mrežnu povezanost koji se uobičajeno mjere odvojeno sa različitim metrikama. Definiraju postojeće CDN sheme odabira poslužitelja: najbliži, najmanje opterećen, najbolja mrežna povezanost. Uočava se da za mnoge velike ISP-ove i organizacije jedan blok IP adresa može biti raširen širom interneta. Mrežna konekcija se mjeri raspoloživom propusnošću, brojem skokova i RTT-om, a status poslužitelja se mjeri raspoloživim CPU-om, memorijom i I/O kapacitetom. Navodi se da ta dva skupa metrika nisu potpuno kompatibilne.

U [20] autori razvijaju „*MyXDN*“ alat kojim istraživači mogu istraživati vlastite metrike za određivanje mrežne blizine i algoritme za usmjeravanje upita. Navode da je DNS infrastruktura osnova za usmjeravanje web zahtjeva u CDN-u, za upravljanje opterećenjem na farmama poslužitelja i za zahtjeve za usmjeravanjem u geografski distribuiranim podatkovnim centrima temeljem lokacije. U *MyXDN*-u se definiraju IP regije koje čine grupe IP adresa koje se jednako tretiraju u procesu usmjeravanja zahtjeva temeljem *GeoIP*[5] usluge.

Autori u [21] prave empirijsku procjena pet klijentski baziranih politika za odabir web usluge („*Random selection*“, „*Parallel invocation*“, „*HTTPing*“, „*Best last*“ i „*Best median*“) pri čemu navode da karakteristike lokalnog klijentskog okruženja mogu imati značajan utjecaj na neke od politika.

U [22] autori analiziraju rješavanje DNS upita u CDN-u pri čemu definiraju sljedeće „*CDN DNS Request Routing*“ strategije: DNS bazirano usmjeravanje zahtjeva, „*Global Server Load Balancing*“, HTTP preusmjeravanje, *Anycasting*, URL prepisivanje i „*CDN peering*“.

Autori u [23] predlažu mrežno infrastrukturno rješenje za odabir poslužitelja u kojemu usmjerivači mogu odabrati alternativne putove za pristup klijenata poslužiteljima.

[5] *GeoIP* usluga – omogućuje pružanje informacije o geografskom položaju temeljem IP adrese

Prema autorima u [24] korištenjem višedomnih (eng. *multihomed*[6]) mreža internetska usluga može biti dostupna višestrukim mrežnim putovima, pa je moguće unaprijediti osjetljivost poslužitelja na mrežne ispade i povećati performanse sustava balansirajući promet između višestrukih linkova. Metoda zahtijeva višestruke lokalne DNS poslužitelje i omogućuje primjenu samo za autoritativne domene. Temelji se na indirektnom mjerenju kašnjenja između klijenta i poslužitelja. Prema predloženoj metodi idealno bi bilo mjeriti kašnjenje između klijenta i poslužitelja prije nego što započne glavni prijenos podataka i dopustiti da klijent odabere najpovoljniji poslužitelj. Navedeno zahtijeva suradnju sa klijentskom stranom što znači da je prilagodba klijentske strane neizbježna čime se izvedba komplicira, zahtijeva se puno prilagodbi na sustavu i unosi se veliko dodatno kašnjenje.

U [25] i [26] se navodi da CDN-ovi usmjeravaju klijente na svoje poslužitelje temeljem nepouzdanih informacija o lokaciji klijenata i da CDN-ovi imaju ograničena saznanja o mrežnim uvjetima između klijenata i svojih poslužitelja. Navodi se da oko 50% internet prometa čini HTTP usluga. Definiraju se dvije osnovne metode za usmjeravanje klijenata: DNS-bazirano (koje se najčešće koristi) i HTTP usmjeravanje (koje unosi više dodatnog prometa i zahtijeva prilagodbu aplikacije). Za primjenu predložene metode klijent mora koristiti ISP-ov DNS poslužitelj pri čemu ISP-ov DNS sustav prosljeđuje autoritativne odgovore „*PaDIS*“ poslužitelju koji ih rangira temeljem mrežnih podataka koje prikuplja od ISP-a. Stanje mreže ISP-a sadrži topologijske informacije (mrežna topologija, opterećenja linkova, opterećenja usmjerivača, promjene u topologiji) i informacije o konekcijama (usmjerivačke informacije) temeljem kojih se izrađuje „*Network Map Database*“.

Autori u [27] analiziraju utjecaj DNS TTL vrijednosti na opterećenje DNS poslužitelja. Navode da je najveći utjecaj korištenja DNS međuspremnika na vršne DNS poslužitelje jer ih koriste svi DNS upiti i da se sa smanjenjem DNS TTL vrijednosti povećava broj upita prema autoritativnom DNS poslužitelju. Iako je tipična preporučena DNS TTL vrijednost 1-5 dana, korištenje dinamičkog DNS-a i upotreba TTL-a u CDN-u za korištenje DNS-a za raspoređivanje opterećenja utječe na skraćenje TTL-a. Utvrđeno je da se postavljanjem TTL na trajanje od 1 dana smanjuje broj DNS upita za 91,8% (jer se samo do 9% upita odnosi na jedinstvene upite a ostali se mogu odgovoriti iz međuspremnika). Daljnjim skraćivanjem

[6] *Multihomed* – označava računalo ili mrežni uređaj koji je spojen na više od jedne računalne mreže

TTL-a na 15 minuta prosječni broj DNS upita je smanjen za 74,4% u odnosu na broj upita koji se postavi bez korištenja međuspremnika.

U [28] je napravljena analiza DNS sustava, te se navodi da se dnevno napravi do 20 milijardi DNS translacija. Predviđa se daljnji razvoj DNS sustava prema obliku točka-točka gdje bi korisnik administrirao svoje DNS zapise bez centralne uprave. U [29] se predlaže proširenje funkcionalnosti DNS sustava i njegovog korištenja kao sustava za određivanje položaja u senzorskim mrežama. U [30] autori predlažu uvođenje *proxy* DNS-a na klijentskoj strani koji će za CNAME DNS odgovore za pristup CDN poslužiteljima direktno slati upite na autoritativne DNS poslužitelje CDN-ova kako bi CDN znao lokaciju klijenta i prilagodio DNS odgovor.

U [31] se predlaže pristup raspoređivanju dijelova video zapisa sa višestrukih poslužitelja u paralelnom načinu rada za HTTP protokol koji se zasniva na vjerojatnosti, uzimajući u obzir heterogenost i vremensku promjenjivost mrežne propusnosti višestrukih poslužitelja. U [32] je dizajniran novi protokol s ciljem poboljšanja korisničkog doživljaja video strujanja koji omogućuje veću pravednost, efikasnost i stabilnost kod prilagodljivog HTTP video strujanja sa višestrukih poslužitelja. Autori u [33] predlažu strategiju dinamičkog odabira poslužitelja koja omogućuje klijentu odabir optimalnog poslužitelja za video strujanje i omogućuje dodatni priključak proizvoljnih algoritama za prilagodbu.

Pregled predloženih rješenja za dinamički odabir poslužitelja višestruko dostupne mrežne usluge upućuje da je potrebno razviti metodu koja nije specifična za jedan tip mrežne usluge te koja uzima u obzir trenutno opterećenje poslužitelja i mrežnu, a ne geografsku, udaljenost klijenta i poslužitelja. Nadalje, nova metoda ne bi trebala aproksimirati mrežnu udaljenost klijenta i poslužitelja temeljem IP adrese klijenta, zahtijevati razvoj specijalne infrastrukture ili korištenje specijaliziranih poslužitelja, zahtijevati stalni nadzor mrežnih usluga, imati kompleksne mrežne zahtjeve za implementaciju nego koristiti temeljnu mrežnu infrastrukturu i postojeći DNS sustav.

2.2. PREGLED I ANALIZA POSTOJEĆIH RJEŠENJA

Zbog velike važnosti stalne dostupnosti i što veće brzine odgovora modernih mrežnih usluga, postoji više rješenja, u različitim fazama implementacije – od prijedloga do funkcionalnih rješenja, kojima je cilj:

1. omogućiti i optimizirati raspoređivanje opterećenja između poslužitelja,
2. skratiti vrijeme preusmjeravanja klijentskih zahtjeva u slučaju ispada komunikacijskog linka ili poslužitelja
3. smanjiti vrijeme odgovora poslužitelja i ubrzati komunikaciju s klijentom.

Postojeće metode za raspoređivanje opterećenja, preusmjeravanje klijentskih zahtjeva i upravljanje vremenom odgovora poslužitelja su po metodologiji rada i načinu implementacije vrlo raznolike i kombiniraju neke ili sve od tri navedene funkcionalnosti te se mogu svrstati u tri osnovne kategorije: DNS metode (metode uključene u DNS protokol), metode koje se oslanjaju na DNS i ostale metode.

2.2.1. DNS metode (metode uključene u DNS protokol)

1) *Mail Exchanger* (MX) RR [34]: omogućuje višestruke DNS zapise za poslužitelje elektroničke pošte kojima se mogu dodijeliti različiti prioriteti. Na taj način moguće je raspoređivati opterećenje te imati osnovni mehanizam za redundanciju u slučaju nedostupnosti poslužitelja. Nedostatak ove metode je u tome što je specifična za određenu vrstu mrežne usluge (u kojem je definirano da se moraju koristiti višestruki MX zapisi ako su raspoloživi), ima dugo vrijeme oporavka i ne uzima u obzir mrežnu udaljenost klijenta

2) *Time To Live* (TTL) parametar RR-a [35]: TTL omogućuje upravljanje vremenom čuvanja RR u privremenoj bazi DNS poslužitelja i DNS klijenta. Skraćenjem TTL parametra povećava se brzina kojom se DNS podaci osvježavaju tako da se za određeni A/AAAA zapis može brzo promijeniti IP adresa u slučaju nedostupnosti ili preopterećenosti mrežnog poslužitelja. Nedostatak je što jako kratak TTL može opteretiti vršne i autoritativne DNS poslužitelje, a i neki DNS poslužitelji ne poštuju minimalno postavljene TTL vrijednosti. Kratak TTL u slučaju gubitka mrežne povezanosti sa rekurzivnim poslužiteljem uzrokuje gubitak funkcionalnosti jer nema

privremeno pohranjene uobičajene DNS odgovore u svojoj privremenoj bazi. Osim toga, potrebu za promjenom DNS zapisa potrebno je detektirati i unijeti u autoritativni DNS poslužitelj.

3) Višestruki A/AAAA zapisi i „*rrset-order*“ opcija [36]: moguće je koristiti višestruke A/AAAA zapise koji pokazuju na isto domensko ime kao jednostavnu i jeftinu tehniku raspoređivanja opterećenja korištenjem opcija „*random*“ i „*cyclic*“ („*round-robin*”) za slučajnu i cikličku rotaciju redoslijeda A/AAAA zapisa u DNS odgovorima. Neke mrežne aplikacije mogu automatski zatražiti pristup sljedećem A/AAAA zapisu u slučaju nedostupnosti trenutnog poslužitelja. Nedostaci ove metode su da je rotiranje zapisa neovisno o opterećenju poslužitelja i njihovih mrežnih veza, da nije optimizirano u odnosu na klijenta te da ju DNS klijenti neautoritativni DNS poslužitelji nisu u obavezi poštivati. Pri tome se na klijentskoj strani opcijom „*sortlist*“ RR zapisi mogu presložiti prema IP adresi klijenta koji je zatražio DNS upit. Kod ove metode je nedostatak i što je vrijeme potrebno za prelazak na drugu IP adresu u slučaju nedostupnosti relativno dugo.

4) GPOS (*Geographical Position*) RR [37], LOC (*Location Information*) RR [38]: ovim DNS zapisima moguće je geografski locirati pojedini poslužitelj, a što se može upotrijebiti za određivanje (aproksimiranje) mrežne udaljenosti poslužitelja. Nedostatak je što u sebi ne sadrži nikakve dodatne informacije, a i klijent bi morao imati informaciju o svom geografskom položaju da bi mogao koristiti podatak o geografskoj lokaciji pojedinog poslužitelja. Kada bi se i koristila u praksi, ova metoda bi bila samo aproksimativna.

5) SRV (*Server Selection*) RR [39] i *DNS-Based Service Discovery* (DNS-SD) [40], (kombinacija SRV RR i TXT RR za dodatne podatke): iako je osnovna namjena SRV RR-a da klijenti mogu pitati za specifičnu uslugu/protokol za određenu domenu i dobiti listu raspoloživih poslužitelja bez poznavanja točnog DNS naziva poslužitelja na kojima se tražena usluga/protokol nalazi, ovim RR-om moguće je upravljati raspodjelom opterećenja poslužitelja jer u sebi sadrži informacije „*Priority*“, o prioritetu pojedinog poslužitelja, i „*Weight*“ o relativnoj raspoloživosti poslužitelja istog prioriteta. Osnovni nedostatak ovog mehanizma, kao i njegove nadogradnje -

DNS-SD, je što ne uzima u obzir mrežnu udaljenost klijenta te što predstavlja statički a ne dinamički parametar o raspoloživosti poslužitelja za odgovor na zahtjev klijenta.

6) *Split Horizon DNS* [41]: neki autoritativni DNS poslužitelji lokaliziraju odgovore temeljem IP adrese s koje je došao upit. Osnovni problemi ove metode su što je takva lokalizacija neprecizna i što se u slučaju rekurzivnih upita odgovor lokalizira u odnosu na rekurzivni DNS poslužitelj koji je upit proslijedio, a ne u odnosu na IP adresu klijenta koji ga je inicijalno generirao.

7) *Edns-client-subnet EDNS0* opcija [42]: omogućuje DNS upitu prosljeđivanje informacije o mrežnoj adresi klijenta koji je postavio DNS upit i na taj način autoritativnom DNS poslužitelju lokalizaciju DNS odgovora u odnosu na mrežnu adresu klijenta. Kako neki autoritativni DNS poslužitelji formiraju odgovore temeljem pretpostavljene topološke lokacije upita ovom metodom se izbjegava problem kreiranja odgovora temeljem adrese rekurzivnog DNS poslužitelja, a ne IP adrese klijenta, u slučajevima da su rekurzivni DNS poslužitelj i klijent značajno mrežno udaljeni. Osnovni nedostaci ove metode su što geografski bliski hostovi mogu biti mrežno značajno udaljeni i što se javlja problem ili velike količine privremene pohrane podataka (eng. *cache*) i njihova pretraživanja u rekurzivnim poslužiteljima za duže mrežne maske, ili povećanog broja upita od strane rekurzivnog poslužitelja ako se RR sa *Edns-client-subnet EDNS0* opcijom privremeno ne pohranjuju u previđenom TTL vremenu.

2.2.2. Metode koje se oslanjaju na DNS

8) *Proxy DNS* [30]: klijentski DNS upiti se prosljeđuju *proxy* DNS poslužiteljima, na klijentima ili u lokalnim mrežama, koji preusmjeravaju rekurzivne DNS upite na autoritativne DNS poslužitelje sa ciljem pružanja autoritativnim DNS poslužiteljima neposredne informacije o klijentu u cilju poboljšanja odgovora DNS poslužitelja. Nedostatak je što se konačna odluka o DNS odgovoru ipak formira na poslužiteljskoj strani ne uvažavajući sve specifičnosti klijenta i što se, u nekim slučajevima, primjenjuje samo za pristup CDN-ovima.

9) *Global Server (geo-based) Load Balancing* [43][44][45]: kod ove metode poslužitelji se postavljaju na različite geografske lokacije i klijenti pristupaju poslužiteljima koji

su im geografski najbliži (npr. Europljani pristupaju europskom poslužitelju, Australci australskom poslužitelju, …). Iako ova metoda predstavlja jednostavan i brz način segmentiranja i usmjeravanja klijenata osnovni nedostatak je što nema optimiziranosti u odnosu na klijenta, na istoj geografskoj lokaciji može biti više ISP-ova koji prema nekom poslužitelju imaju različite brzine (različita mrežna udaljenost), neke geografske lokacije mogu biti bez fizički bliskih ili sa više fizički bliskih poslužitelja, a moguće je i da su poslužitelji ili komunikacijski linkovi na nekoj lokaciji preopterećeni.

10) *Managed ('Intelligent') DNS services* [46][47]: prostorno distribuirane DNS usluge koji nadziru rad mrežnih usluga na pojedinim distribuiranim poslužiteljima i malim TTL vrijednostima omogućuju upravljanje DNS zapisima u slučajevima velikog opterećenja i nedostupnosti poslužitelja. Nedostatak je što zahtijevaju posebnu infrastrukturu i stalni aktivni nadzor usluga, što se oslanjaju na brzinu propagacije RR-a i na geografski smještaj klijenta/rekurzivnog DNS poslužitelja, a ne na njegovu mrežnu udaljenost.

11) *DNS based Request-Routing Mechanisms* [48][49]: specijalizirani DNS poslužitelji se ubacuju u proces slanja DNS odgovora, imaju mogućnost slanja različitih RR zapisa u ovisnosti o korisnički definiranim pravilima, metrici ili njihovoj kombinaciji. Osnovni nedostatak je što zahtijeva upotrebu specijaliziranih DNS poslužitelja, ovisi o TTL vrijednosti i može dovesti do preopterećenja autoritativnih DNS poslužitelja. Osnovna namjena je raspoređivanje opterećenja u CDN-ovima.

2.2.3. Ostale metode

12) *Anycast* [50] – *BGP* [51] *based failover system*: omogućuje visoku dostupnost distribuiranih usluga korištenjem jedinstvene IP adrese i BGP protokola. Ovisi o brzini konvergencije BGP protokola, ima složene mrežne zahtjeve te stoga ima i skupu implementaciju

13) *Relative network proximity and latency estimation* [52]: različite metode za određivanje mrežne blizine i kašnjenja u mreži, vezane su za mrežne komunikacijske protokole i temeljnu mrežnu infrastrukturu

14) *Available bandwidth estimation* [53]: različite metode za određivanje raspoložive propusnosti linkova i putanja, vezane su za temeljnu mrežnu infrastrukturu

15) *Hardware load balancers* [54][55]: rade na principu usmjeravanja opterećenja na mrežnoj (IP) razini, ne uzimaju u obzir mrežnu udaljenost klijenta. Uobičajene metode/strategije za upravljanje opterećenjem su: metoda kružnog dodjeljivanja (eng. *round-robin*), najmanji broj konekcija, opterećenje poslužitelja, najbrže vrijeme odgovora poslužitelja i podaci prikupljeni od poslužiteljskog agenta. Uobičajene metode provjere dostupnosti poslužitelja su ping, TCP konekcija, jednostavni i puni HTTP GET[7]

16) *HTTP Load Balancing* [56][57]: omogućuje raspodjelu HTTP zahtjeva za postizanje raspoređivanja opterećenja između poslužitelja HTTP sadržaja te postizanja visoke raspoloživosti i skalabilnosti HTTP usluge. Radi na aplikacijskoj razini OSI referentnog modela[8] korištenjem aplikacijskog usmjeravanja i specifičan je za HTTP protokol

17) *Content Delivery Network* (CDN) [58][59][60]: mreža za isporuku sadržaja određuje na kojoj točki će klijentu isporučiti sadržaj temeljem performansi sustava (mrežna udaljenost, performanse poslužitelja) i cijene isporuke (linka) na pojedinoj lokaciji. Usmjeravanje klijentskih zahtjeva temelji se na različitim algoritmima: *„Global Server Load Balancing“, „DNS-based request routing“, „HTTP Load Balancing (HTML rewriting)“, „Anycasting“, „Proximity“* te CDN specifičnim protokolima (*„Dynamic Metafile Generation“, „Internet Content Adaptation Protocol“, „Open Pluggable Edge Services“ i „Edge Side Includes“*). Zahtijeva korištenje posebne mrežne infrastrukture i umnožavanje sadržaja na različitim geografskim lokacijama.

2.2.4. Analiza postojećih metoda

U Tablici 1. prikazana su najznačajnija obilježja postojećih metoda uz sažeti opis njihovih nedostataka u odnosu na uvodno postavljene zahtjeve za odabir poslužitelja višestruko dostupne mrežne usluge:

[7] HTTP GET – metoda za traženje podatka od HTTP usluge, jednostavni HTTP GET provjerava samo zaglavlje odgovora (npr. 200 OK) dok puni HTTP GET provjerava sadržaj tijela odgovora

[8] OSI referentni model - najkorišteniji apstraktni opis arhitekture računalne mreže, sastoji se od 7 slojeva

R.br.	Metoda	Kategorija metode	Primjena metode[9]	Značajke / Prednosti	Nedostaci
1.	*Mail Exchanger* (MX) RR [34]	DNS metoda	Poslužitelj	Raspoređivanje opterećenja, redundancija poslužitelja	Specifična za jednu vrstu mrežne usluge, bez mrežne udaljenosti klijenta
2.	*Time To Live* (TTL) parametar RR-a [35]	DNS metoda	Poslužitelj	Mogućnost raspoređivanja opterećenja, redundancija poslužitelja	Opterećenje vršnih DNS-ova, nepoštivanje TTL polja, ovisnost o rekurzivnom poslužitelju, bez mrežne udaljenosti klijenta
3.	Višestruki A/AAAA zapisi i „*rrset-order*“ opcija [36]	DNS metoda	Poslužitelj, klijent	Raspoređivanje opterećenja, ograničena redundancija poslužitelja	Rotiranje RR nije vezano za opterećenje poslužitelja i linkova, dugo vrijeme oporavka, sortiranje na klijentskoj strani samo po IP adresi klijenta (*sortlist*)
4.	GPOS (*Geographical Position*) RR, LOC (*Location Information*) RR [37][38]	DNS metoda	Poslužitelj	Ubrzanje vremena odgovora usluge aproksimacijom mrežne udaljenosti geografskom udaljenošću	Potreba određivanja geografske udaljenosti poslužitelja i klijenta, bez mrežne udaljenosti
5.	SRV (*Server Selection*) RR i *DNS-Based Service Discovery* (DNS-SD) [39][40]	DNS metoda	Poslužitelj	Raspoređivanje opterećenja	Ne uzima u obzir mrežnu udaljenost klijenta, opterećenje je statički parametar
6.	*Split Horizon DNS* [41]	DNS metoda	Poslužitelj	Ubrzanje odziva usluge lokaliziranjem DNS odgovora temeljem IP adrese DNS upita	Neprecizna, posebno kod rekurzivnih upita, bez mrežne udaljenosti klijenta
7.	*Edns-client-subnet EDNS0* opcija [42]	DNS metoda	Poslužitelj	Ubrzanje odziva usluge lokaliziranjem DNS odgovora temeljem IP adrese klijenta	Mrežna blizina se aproksimira geografskom, opterećenje DNS poslužitelja
8.	*Proxy DNS* [30]	DNS-bazirana metoda	Klijent	Ubrzanje odziva usluge lokaliziranjem DNS odgovora temeljem IP adrese DNS upita (klijenta)	Konačnu odluku o DNS zapisu donosi poslužitelj temeljem IP adrese klijenta
9.	*Global Server (geo-based) Load Balancing* [43][44][45]	DNS-bazirana metoda	Poslužitelj	Ubrzanje odziva usluge fizičkim približavanjem višestruko dostupnih mrežnih usluga klijentu	Nema optimiziranosti u odnosu na mrežnu udaljenost klijenta i opterećenje poslužitelja

[9] Pojam „Poslužitelj“ označava stvarni poslužitelj, mrežnu i(ili) sistemsku infrastrukturu koja nije unutar lokalne mreže klijenta

R.br.	Metoda	Kategorija metode	Primjena metode[9]	Značajke / Prednosti	Nedostaci
10.	*Managed ('Intelligent') DNS services* [46][47]	DNS-bazirana metoda	Poslužitelj	Raspoređivanje opterećenja, redundancija poslužitelja	Zahtijeva posebnu infrastrukturu i stalni nadzor usluga, oslanja se na brzinu propagacije RR, bez mrežne udaljenosti klijenta
11.	*DNS based Request-Routing Mechanisms* [48][49]	DNS-bazirana metoda	Poslužitelj	Raspoređivanje opterećenja, redundancija poslužitelja	Upotreba specijaliziranih DNS poslužitelja, aproksimacija mrežne udaljenosti klijenta
12.	*Anycast – BGP based failover system* [50][51]	Ostale metode	Poslužitelj	Redundancija poslužitelja	Ovisi o konvergenciji BGP protokola, složeni mrežni zahtjevi
13.	*Relative network proximity and latency estimation* [52]	Ostale metode	Poslužitelj	Ubrzavanje odziva usluge određivanjem mrežne blizine i kašnjenja u mreži	Vezane za mrežne komunikacijske protokole i temeljnu infrastrukturu
14.	*Available bandwidth estimation* [53]	Ostale metode	Poslužitelj	Ubrzavanje odziva usluge određivanjem raspoložive propusnosti linkova i putanja	Vezane za temeljenu mrežnu infrastrukturu
15.	*Hardware load balancers* [54][55]	Ostale metode	Poslužitelj	Raspoređivanje opterećenja, redundancija poslužitelja	Orijentirani prema poslužiteljima a ne i prema klijentima
16.	*HTTP Load Balancing* [56][57]	Ostale metode	Poslužitelj	Raspoređivanje opterećenja, redundancija poslužitelja	Radi na aplikacijskoj razini, specifičan za HTTP protokol
17.	*Content Delivery Network* [58][59][60]	Ostale metode	Poslužitelj	Raspoređivanje opterećenja, redundancija poslužitelja	Korištenje posebne mrežne infrastrukture, aproksimacija mrežne udaljenosti klijenta

Tablica 1. *Pregled postojećih metoda*

Zajedničko svim navedenim metodama je da pitanja raspoređivanja opterećenja, preusmjeravanja u slučaju ispada i upravljanja vremenom odgovora poslužitelja rješavaju na poslužiteljskoj strani, pokušavajući u rješenje unijeti i parametre klijenta (najčešće temeljem geografske lokacije klijenta) ali opet gledajući klijenta sa poslužiteljske strane, a ne uslugu sa klijentske strane. Problemi i nedostaci postojećih metoda:

- specifične za jednu vrstu mrežne usluge
- ne uzimaju u obzir mrežnu udaljenost klijenta
- ne uzimaju u obzir opterećenje poslužitelja i komunikacijskih linkova
- ovise o procjeni geografske udaljenosti poslužitelja i klijenta

- aproksimiraju mrežnu udaljenost klijenta IP adresom klijenta
- zahtijevaju izradu posebne infrastrukture i/ili upotrebu specijaliziranih poslužitelja
- zahtijevaju stalni nadzor mrežnih usluga
- imaju složene mrežne zahtjeve, vezani su za temeljnu mrežnu infrastrukturu.

Zbog postojanja nedostataka niti jedna od analiziranih metoda ne zadovoljava u cijelosti uvodno postavljene zahtjeve za odabir poslužitelja višestruko dostupne mrežne usluge te se iz tog razloga u ovom radu predlaže nova metoda za dinamički odabir poslužitelja višestruko dostupne mrežne usluge koja u cijelosti zadovoljava uvodno postavljene zahtjeve. Nova metoda ne isključuje bilo koju od postojećih metoda odabira poslužitelja i može biti korištena kao samostalna ili dodatna metoda za odabir optimalnog poslužitelja na klijentskoj strani, naročito kao proširenje poslužiteljskih metoda uključivanjem klijentskog pogleda u odluku o konačnom odabiru optimalnog poslužitelja. Metodom se rješava problem uvođenja odabira poslužitelja na klijentskoj strani kao i postojeći problemi takvog odabira: prikupljanje, kombiniranje i dostava potrebnih poslužiteljskih parametara.

3. DSS (*DYNAMIC SERVER SELECTION*) – PRIJEDLOG METODE ZA DINAMIČKI ODABIR POSLUŽITELJA VIŠESTRUKO DOSTUPNE MREŽNE USLUGE

Da bi se napravio kvalitativni napredak u odnosu na postojeće metode odabira poslužitelja višestruko dostupne mrežne usluge potrebno je omogućiti klijentu da samostalno, temeljem dostavljenih DNS informacija o dostupnim IP adresama za traženu uslugu, provjeri koji su mu poslužitelji trenutno dostupni, koji su im faktori opterećenja te koja im je mrežna udaljenost i na taj način, prema unaprijed utvrđenim pravilima, odabere za sebe najpogodniji poslužitelj. Također je potrebno omogućiti, u slučaju nedostupnosti poslužitelja - bilo zbog nedostupnosti samog poslužitelja ili komunikacijske putanje između klijenta i poslužitelja, što je moguće brži prelazak na drugi dostupni poslužitelj.

Prijedlog rješenja za odabir poslužitelja višestruko dostupnih mrežnih usluga je uvođenje dinamičkih poslužiteljskih i klijentskih parametara korištenjem DNS sustava, temeljem kojih će se posebnim algoritmom izraditi metrika DNS A/AAAA zapisa. Metrikom će se definirati prioritet IP adresa mrežnih poslužitelja dostavljenih kao odgovor na postavljeni DNS upit za određenu mrežnu uslugu. Na taj način se dva dinamička parametra, jedan sa strane klijenta (mrežna udaljenost predstavljena RTT vrijednošću, prema [9][11]) i jedan sa poslužiteljske strane (opterećenje poslužitelja, prema [12]) mogu međusobno kombinirati. Time se rješava problem uvođenja metode odabira poslužitelja na strani klijenta pri čemu su dosadašnji problemi bili prikupljanje, kombiniranje i isporuka potrebnih poslužiteljskih parametara [14][18].

Metodom se vodi nova vrsta DNS zapisa: *Resource Record (RR) TYPE*: DSS u DNS klasi *CLASS (IN)* [61]

3.1. DINAMIČKI ODABIR MREŽNE USLUGE PRIMJENOM DSS METODE

Osnovna ideja metode je omogućiti klijentu dinamički odabir najpogodnijeg poslužitelja višestruko dostupne mrežne usluge korištenjem DNS sustava kojim će se optimizirati:

- korištenje/raspodjela poslužiteljskih resursa (eng. *load balancing*)
- korištenje optimalne (najbrže) mrežne putanje do poslužitelja (eng. *network response time*)
- omogućiti brzo rješavanje problema nedostupnosti mrežnih usluga (eng. *failover*) koji mogu biti uzrokovani:
    - nedostupnošću poslužitelja
    - nedostupnošću mrežne putanje od klijenta do poslužitelja.

Pri tome su parametri odabira najpogodnijeg poslužitelja:

- opterećenje poslužitelja
- vrijeme mrežnog odziva poslužitelja.

3.2. FORMAT DNS PORUKE

DNS poruka ima generički oblik i sastoji se od 5 sekcija:

Sekcija	Značenje/Upotreba
Sekcija 1	Zaglavlje poruke
Sekcija 2	DNS upit
Sekcija 3	Zapis koji odgovara na upit
Sekcija 4	Zapis koji pokazuje na autoritativnu domenu
Sekcija 5	Zapis koji može sadržavati dodatne informacije

Zaglavlje poruke je fiksne veličine i prisutno je u svim porukama. Sadrži različite zastavice i vrijednosti koje kontroliraju DNS transakciju (upit/odgovor):

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Message ID															
QR	OPCODE				AA	TC	RD	RA	res1	res2	res3	RCODE			
QDCOUNT															
ANCOUNT															
NSCOUNT															
ARCOUNT															

DSS metoda će u ARCOUNT polje (polje u koje se upisuje broj zapisa u sekciji 5 - dodatne informacije) upisati broj DSS zapisa koji se prosljeđuju u odgovoru.

Sekcija DNS upita je uvijek prisutna u DNS poruci i sastoji se od 3 polja, a broj DNS sekcija upita se upisuje u QDCOUNT polje:

Naziv polja	Značenje/Upotreba
QNAME	Domensko ime koje se pita
QTYPE	Tip zapisa koje se traži
QCLASS	Klasa zapisa koji se traži

Sekcije odgovora, autoritativne domene i dodatnih informacija mogu, ali i ne moraju, biti u DNS poruci i sve imaju isti oblik:

Naziv polja	Značenje/Upotreba
NAME	Naziv koji se vraća u odgovoru
TYPE	Tip zapisa
CLASS	Klasa zapisa
TTL	TTL u sekundama
RLENGTH	Dužina RDATA polja u bajtovima
RDATA	Specifični podaci za tip zapisa

3.3. ZAHTJEVI NA IMPLEMENTACIJU DSS METODE

Zahtjevi metode definiraju odnos metode sa postojećim DNS sustavom, uključujući i poslužiteljsku i klijentsku stranu, i definirani su kako slijedi:

- potpuna kompatibilnost sa postojećim:
    - DNS poslužiteljima (autoritativnim i neautoritativnim)
    - DNS klijentima (eng. *local resolver*)
    - mrežnim aplikacijama kao krajnjim korisnicima DNS podataka

  Kompatibilnost znači da u slučaju da DNS poslužitelj koji podržava DSS metodu pošalje DSS RR-ove klijentu ili drugom DNS poslužitelju koji ih ne prepoznaje taj klijent ili DNS poslužitelj ih treba ignorirati bez obavijesti o greški [62]

- omogućiti funkcioniranje metode i za rekurzivne i za iterativne DNS upite
- za korištenje DSS zapisa klijent ne mora u zahtjevu slati informaciju da je DSS-kompatibilan već DSS-kompatibilni DNS poslužitelji mogu uvijek slati DSS zapise koje će DSS-nekompatibilni klijenti odbacivati/ignorirati bez poruke o greški. To omogućuje da se DNS zapisi spremaju u privremene baze neautoritativnih DNS poslužitelja i klijenata (što je definirano osnovnim dizajnom DNS sustava – eng. *local caching to improve performance*) i da se koriste i za odgovore klijentima koji podržavaju i za klijente koji ne podržavaju DSS zapis
- metoda ne smije unositi nove sigurnosne propuste u postojeći DNS sustav
- metoda mora omogućiti veći broj DSS odgovora:
    - za višestruke zahtjeve u jednom upitu (kada se u jednom upitu traže podaci za više poslužitelja)
    - za višestruke protokole usluga za provjeru mrežne udaljenosti (kada se na istoj IP adresi poslužitelja nalazi više različitih usluga)
- DNS administratorima autoritativnih DNS poslužitelja se omogućuje da samostalno odlučuju da li će koristiti DSS ili ne, dok administratori neautoritativnih DNS poslužitelja trebaju pohranjivati DSS zapise u privremenu bazu i prosljeđivati ih kao i sve ostale DNS RR-ove
- DSS metoda ne unosi promjene u postojećem DNS sustavu, ali zahtijeva da DNS odgovori budu A/AAAA, a ne CNAME zapisi te preporuča da ukupna dužina odgovora, uključujući *Answer*, *Authority* i *Additional* sekcije ne prelazi 512 okteta kao ne bi bili filtrirani na usmjerivačima, vatrozidovima i ostalim mrežnim uređajima koji uobičajeno smatraju UDP DNS paket legitimnim samo ako ne prelazi veličinu od 512 okteta.
- DSS metoda ne isključuje niti jednu do sada uvedenu strategiju rješavanja pitanja načina pristupa mrežnim resursima korištenjem DNS upita
- Metoda ne smije unositi zahtjeve za promjenu mrežnih ograničenja i kontrole prometa, stoga se pri provjeri vremena mrežnog odziva usluge omogućuje komunikacija sa predefiniranim protokolom/portom (slanjem mrežnih paketa predefiniranoj mrežnoj usluzi), a ne isključivo slanjem standardnih testnih paketa (*ICMP Echo request*) koji mogu biti filtrirani na mrežnoj putanji

- za punu funkcionalnost metode potrebno je modificirati postojeću funkcionalnost DNS poslužitelja, DNS klijenata i mrežnih aplikacija, a minimalno je potrebno modificirati DNS poslužitelje i DNS klijente.

3.4. PRIJEDLOG NOVOG FORMATA DNS ZAPISA DSS METODE

DSS RR je opcionalni zapis u DNS bazi autoritativnog poslužitelja i upisuje se u sekciju 5 DNS poruke. Omogućuje opisivanje parametara opterećenosti pojedinog poslužitelja (definiranog A/AAAA zapisom), parametre za provjeru mrežne udaljenosti i parametre za izračun kompozitne metrike A/AAAA zapisa. DSS RR se koristi kao zapis u dodatnoj sekciji DNS odgovora (eng. *Additional Section*) i ima standardnu strukturu zapisa dodatne sekcije:

Name Type Class TTL A/AAAA Priority Load Impact Request Interval Protocol Port Time Refresh Timeout Flags

Naziv polja:	Tip polja:	Opis polja:
NAME:	u_int16_t	domensko ime RR-a, vrijednost QNAME (DNS naziv za koji se postavlja upit) iz DNS upita (*DNS Question*) u formi 16-bitnog pokazivača (zbog ograničenja i potencijalnog skraćenja dužine polja)
TYPE:	u_int16_t	tip RR zapisa u formi 16-bitnog polja, za DSS RR je potrebno u IANA bazi tipova zapisa dodati novi tip zapisa: DSS
CLASS:	u_int16_t	klasa RR zapisa u formi 16-bitnog polja, za DSS RR ima standardnu vrijednost 0x0001 (1) IN
TTL:	u_int32_t	vrijeme čuvanja zapisa u privremenoj bazi u formi 32-bitnog polja. Za DSS metodu TTL vrijednost treba osigurati ažurne vrijednosti DSS parametara, posebno LOAD parametra koji može biti izrazito vremenski promjenjiv. TTL vrijednost također treba uzeti u obzir povećanje broja DNS upita i DNS mrežnog prometa za njegove male vrijednosti ili vrijednost 0
RDLENGHT:	u_int16_t	16-bitno polje koje definira dužinu RDATA zapisa u oktetima
RDATA:	sadrži podatke DSS zapisa i ima format:	

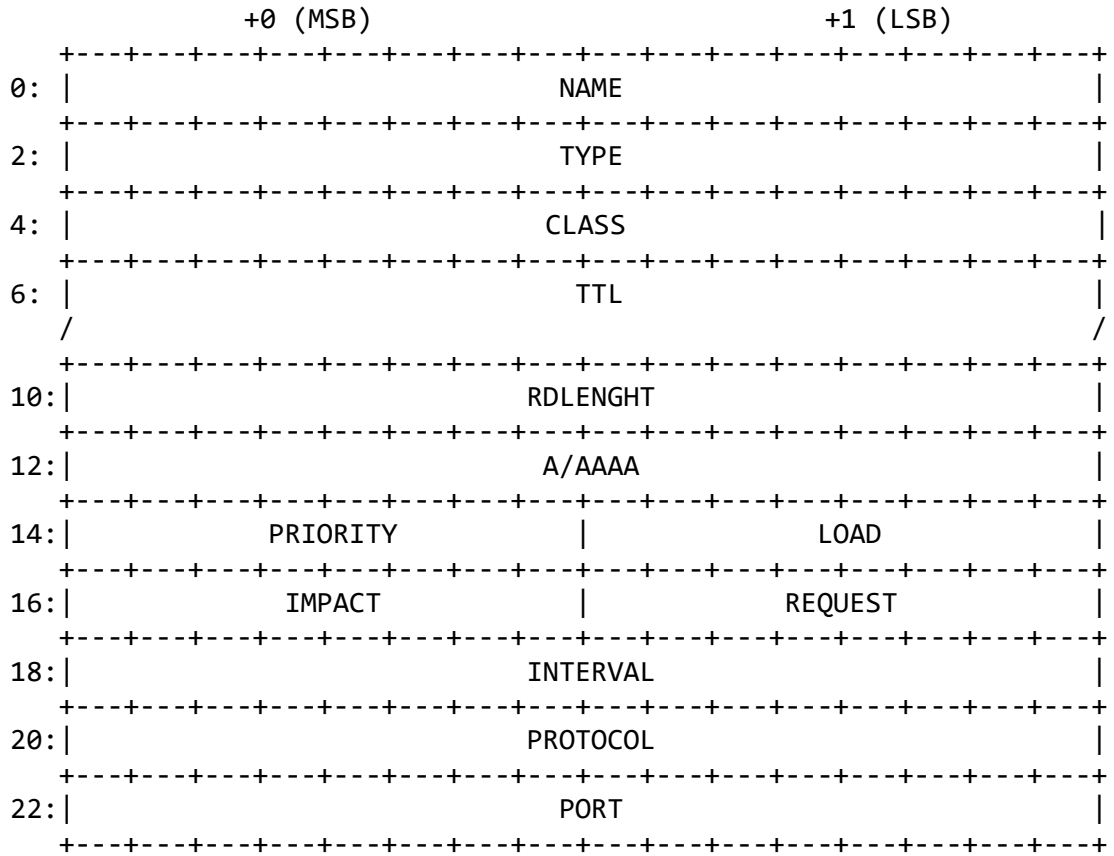
Naziv polja:	Tip polja:	Opis polja:

A/AAAA:	u_int16_t	vrijednost A/AAAA zapisa poslužitelja za koji je DSS zapis definiran. U bazi autoritativnog poslužitelja upisuje se 32-bitna IP adresa za IPv4, odnosno 128-bitna IP adresa za IPv6 protokol. Tijekom formiranja odgovora na DNS upit zamjenjuje se 16-bitnim pokazivačem (zbog ograničenja i skraćenja dužine polja) u formi pokazivača NAME polja
PRIORITY	u_int8_t	faktor prioriteta poslužitelja u formi 8-bitnog polja, omogućuje razlikovanje primarnih, sekundarnih,… poslužitelja. Vrijednost polja 0 označava najveći prioritet a vrijednost 255 najmanji prioritet. Poslužitelji najvećeg prioriteta se prvi kontaktiraju u postupku izračuna kompozitne DNS-metrike
LOAD	u_int8_t	faktor opterećena poslužitelja u formi 8-bitnog polja, omogućuje opisivanje opterećenja poslužitelja unutar klase prioriteta. Vrijednost polja 0 označava najmanje, a vrijednost 255 najveće opterećenje. Parametar LOAD definira administrator poslužitelja i ovisi o tipu mrežne usluge. Može se temeljiti na opterećenju CPU-a, IO opterećenju diska, raspoloživoj memoriji, mrežnom opterećenju i slično, ili bilo kojoj kombinaciji poslužiteljskih parametara koji opisuju njegove performanse. Predstavlja trenutno opterećenje sustava koje uzrokuje kašnjenje obrade zahtjeva korisnika u poslužitelju, a preporuka je da u sebi sadržava informacije o opterećenju ključnih komponenti poslužitelja za funkcioniranje mrežne usluge te da se održava ažurnim. Ovim parametrom

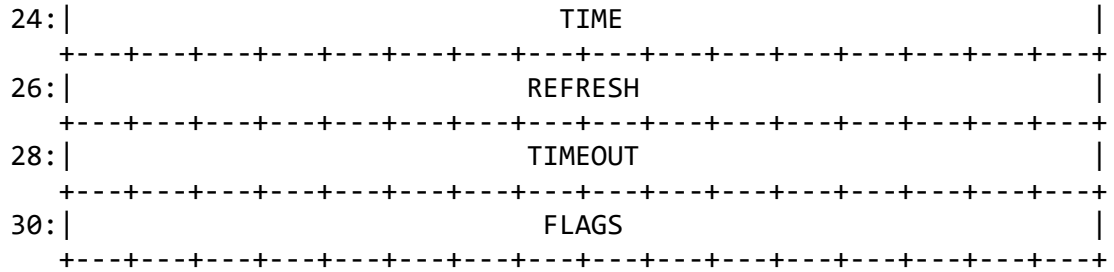
		administrator poslužitelja predlaže klijentu na koji poslužitelj bi trebao slati zahtjev
IMPACT	u_int8_t	faktor kompozitne metrike DSS zapisa u formi 8-bitnog polja, koristi se u postupku izračuna kompozitne metrike za određivanje utjecaja mrežne udaljenosti poslužitelja na izračun kompozitne metrike. Vrijednost 0 označava najmanji utjecaj (mrežna udaljenost se ne uzima u obzir) a vrijednost 255 označava najveći utjecaj. Može se koristiti i za davanje prioriteta pojedinim mrežnim linkovima poslužitelja
REQUEST	u_int8_t	broj upita za testiranje mrežne udaljenosti u formi 8-bitnog polja, definira broj upita koji klijent šalje prema poslužitelju u svrhu određivanja utjecaja mrežne udaljenosti poslužitelja, veći broj upita povećava povezanost između RTT-a i propusnosti
INTERVAL	u_int16_t	vrijeme u milisekundama između slanja upita za testiranje mrežne udaljenosti u formi 16-bitnog polja
PROTOCOL	u_int16_t	oznaka IP protokola u formi 16-bitnog polja, ima standardnu vrijednost 1 za ICMP, 6 za TCP a 17 za UDP
PORT	u_int16_t	oznaka PROTOCOL porta u formi 16-bitnog polja, na kojem se nalazi usluga za testiranje mrežne udaljenosti
TIME	u_int16_t	vrijeme u milisekundama od slanja prvog paketa za testiranje mrežne udaljenosti do pokretanja DSS procedure za izračun kompozitne DNS-metrike, u formi 16-bitnog polja

REFRESH	u_int16_t	vrijeme u milisekundama od pokretanja izračuna DNS-metrike do ponovnog pokretanja DSS izračuna za osvježavanje DNS-metrike, u formi 16-bitnog polja
TIMEOUT	u_int16_t	vrijeme u milisekundama od slanja prvog paketa za testiranje mrežne udaljenosti do proglašavanja poslužitelja nedostupnim, u formi 16-bitnog polja
FLAGS	u_int16_t	16-bitno polje, 8 bitova (MSB) za razvoj (prvi bit vrijednošću 0 označava *stateful*[10] a vrijednošću 1 *stateless*[11] mrežnu uslugu, ostali bitovi slobodni), 8 bitova (LSB) za oznaku DSS verzije, inicijalna verzija je 0.

Binarni prikaz DSS zapisa, koji je nepromjenjive dužine i sastoji se od 32 okteta:



[10] Stateful – prati se stanje konekcije/interakcije
[11] Stateless – ne prati se stanje konekcije/interakcije



3.5. OPIS FUNKCIONALNOSTI DSS ZAPISA

3.5.1. Autoritativni DNS poslužitelj

Pravila implementacije DSS zapisa u autoritativnim DNS poslužiteljima:

- Autoritativni DNS poslužitelji implementiraju DSS zapise isključivo za A/AAAA zapise i ne koriste ih za CNAME zapise. Svaki A/AAAA zapis može imati jedan ili više DSS zapisa ali samo jedan zapis za istu vrijednost PROTOCOL/PORT. Ako je za isto domensko ime definirano više DSS zapisa primjenjuje se posljednji definirani zapis, osim ako se eksplicitno ne zatraži određeni DSS zapis definiranjem vrijednosti PROTOCOL/PORT. Ako za neko domensko ime postoji barem jedan DSS zapis vrijednost zadnjeg definiranog DSS zapisa se primjenjuje na sve A/AAAA zapise tog mrežnog imena, a za koje DSS zapis nije eksplicitno definiran. Ako za domensko ime nije definiran niti jedan DSS zapis podrazumijeva se da za to domensko ime DSS nije implementiran
- Za DSS zapise vrijede sva pravila kao i za ostale RR zapise, za vrijednosti koje u zapisu nisu navedene podrazumijeva se vrijednost definirana u prvom prethodno definiranom DSS NAME zapisu. Prvi DSS zapis za svako domensko ime mora imati definirana sva DSS polja, odnosno nema predefiniranih vrijednosti DSS parametara (polja)
- Za vrijednosti polja PROTOCOL i PORT moguće je upisivati ili brojčanu oznaku protokola i porta ili njihov simbolički naziv ali se u odgovor upisuje samo brojčana oznaka
- Autoritativni DNS poslužitelj će u odgovoru na traženi A/AAAA zapis u dodatnu sekciju upisati sve definirane DSS zapise, popunjavajući pri tome sva definirana polja, redoslijedom upisa u bazi, pri čemu je redoslijed unosa zapisa u bazi DNS

zone relevantan ako se koristi skraćeni način pisanja DSS RR zapisa. DSS algoritam DNS klijenta će primijeniti sva pravila za višestruke zapise i zapise koji nisu definirani

Preporuka je da TTL vrijednosti DSS zapisa odgovaraju TTL vrijednostima pripadajućih A/AAA zapisa zbog omogućavanja istovremenog uklanjanja iz priručnih memorija DNS poslužitelja i klijenata [63][64].

Primjer implementacije DSS zapisa u autoritativnom poslužitelju:

```
$ORIGIN example.com.
@                       SOA server.example.com. postmaster.example.com. (
                        2013010101 3600 3600 604800 3600 )
                        NS  ns1.example.com.
                        NS  ns2.example.com.
; A zapisi
server  2h      IN      A       161.53.201.130
                        A       193.198.68.135
                        A       213.191.152.140
                        A       85.114.46.150
; DSS zapisi
server  2h      IN      DSS     161.53.201.130 0 0 0 4 5 TCP HTTP 2000 5 5000 0
                        DSS     213.191.152.140 0 127
                        DSS     85.114.46.150 0 100 255
                        DSS     161.53.201.130 0 0 0 4 10 TCP SMTP 3000 10 5000 0
```

Objašnjenje zapisa autoritativnog DNS poslužitelja:

Domensko ime „server.example.com.“ ima definirane 4 IPv4 adrese mrežnih poslužitelja: 161.53.201.130, 193.198.68.135, 213.191.152.140 i 85.114.46.150. Iako je u SOA zapisu definiran minimalni TTL od 1 sata (3600 sekundi) administrator je za A zapise mrežnog imena „server“ postavio TTL na 2 sata (7200 sekundi).

Za domensko ime „server.example.com.“ definirana su 4 DSS zapisa, 3 za TCP port 80 (HTTP usluga) i 1 za TCP port 25 (SMTP usluga) sa TTL vrijednostima koje su identične pripadajućem A zapisu. U prvom zapisu su definirane vrijednosti svih DSS polja. Za IP adresu 193.198.68.135 nije definiran niti jedan DSS zapis, a kako se ta IP adresa nalazi u A zapisu mrežnog imena „server“ za tu IP adresu primjenjuje se posljednji definirani DSS zapis: „DSS 161.53.201.130 0 0 0 4 10 TCP SMTP 3000 10 5000 0“. Za DSS zapise za IP adrese 213.191.152.140 i 85.114.46.150, a koji nemaju definirane sve DSS parametre, primjenjuju se parametri prvog DSS zapisa, odnosno za IP adresu 213.191.152.140

vrijednosti „0 4 5 TCP HTTP 2000 5 5000 0“ a za IP adresu 85.114.46.150 vrijednosti „4 5 TCP HTTP 2000 5 5000 0“.

Odgovor na DNS upit koji sadrži DSS zapis sastoji se od četiri sekcije koje imaju sljedeći format:

- Zaglavlje: 6 x 2 bajta = 12 bajtova (fiksno)
- DNS upit: varijabilno domensko ime (max. 256 znakova) + 2 x 2 bajta = varijabilno + 4 bajta
  za server.example.com. = 20 + 4 = 24 bajta
- DNS odgovor: varijabilno domensko ime (max. 256 znakova, min. 2 bajta) + 2 x 2 bajta + 4 bajta + 2 bajta + varijabilna RR vrijednost = 10 + 2 x varijabilno, min. 12 bajtova + varijabilno
  za komprimirani A zapis: 2 bajta + 10 bajtova + 4 bajta = 16 bajtova (fiksno)
  za server.example.com. sa 4 A zapisa = 16 bajtova x 4 = 64 bajta
- Dodatni zapis: za DSS zapis 32 bajta (fiksno)
  za server.example.com. sa 4 DSS zapisa = 32 bajtova x 4 = 128 bajtova

Izračun veličine UDP datagrama odgovora za upit A zapisa za domensko ime „server.exampe.com.“:

Ukupna veličina DNS odgovora za A upit „server.example.com.“, a koji u odgovoru ima 4 A zapisa i 4 DSS zapisa, je 12 + 24 + 64 + 128 = 228 bajtova. Kako je preferirani način slanja DNS upita i odgovora UDP protokol, ukupna veličina UDP datagrama je: 8 bajtova UDP zaglavlje + 228 bajtova DNS odgovor = 236 bajtova što je značajno manje od preporučene maksimalne veličine DNS UDP datagrama od 512 bajtova (bez IP i UDP zaglavlja). Kada se u A/AAAA polju ne bi koristio 16 bitni pokazivač nego 32 bitna IP adresa ukupna veličina DNS odgovora iz primjera bi bila 228 + 4 x 2 = 236 bajtova (veličina jednog DSS zapisa se u tom slučaju povećava za 2 bajta). Kada se računa ukupna veličina DNS odgovora mora se uzeti u obzir veličina svih DNS RR zapisa uključenih u odgovor, kao što su *Name Server* zapisi u sekciji autoritativne domene i njihovi A ili AAAA zapisi u dodatnoj sekciji.

3.5.2. Neautoritativni DNS poslužitelj

Neautoritativni DNS poslužitelji trebaju DSS zapise prosljeđivati u istom obliku kako su ih zaprimili: za A/AAAA zapise trebaju proslijediti sve pripadajuće DSS zapise nepromijenjenog sadržaja (osim TTL polja) istim redoslijedom kojim su ih u DNS odgovoru zaprimili od autoritativnog ili nekog drugog neautoritativnog poslužitelja.

Neautoritativni poslužitelji koji DSS zapise dobiju od neautoritativnog poslužitelja nemaju potrebu zatražiti autoritativni odgovor.

Neautoritativni DNS poslužitelji mogu DSS zapise pohranjivati u privremenu bazu poštujući TTL vrijednost DSS zapisa i zapisujući ih istim redoslijedom kojim su ih zaprimili.

3.5.3. DNS klijent (*resolver*)

Iako implementacija DSS metode omogućuje odabir optimalnog poslužitelja za neku mrežnu uslugu DNS klijent može birati želi li koristiti DSS metodu ili ne, te može u cijelosti ignorirati DSS RR zapise. DNS klijent koji dobije DNS odgovor koji sadrži DSS zapis, a podržava DSS metodu i želi ju implementirati, izračunava DNS-metriku na sljedeći način:

- Provjerava da li svi A/AAAA zapisi imaju pripadajući DSS zapis. Ako neki A/AAAA zapis nema pripadajući A/AAAA zapis na njega primjenjuje podatke posljednjeg DSS zapisa dobivenog u DNS odgovoru
- Provjerava da li neki A/AAAA zapis ima višestruki DSS zapis. Ako postoje višestruki DSS zapisi uzima se posljednji DSS zapis osim u slučaju da DNS klijent ima podatak za koji PROTOCOL/PORT se traži A/AAAA zapis, a u kojem slučaju se odabire odgovarajući DSS zapis. Za punu funkcionalnost DSS zapisa, koji podržava višestruke DSS zapise za isti A/AAAA zapis, potrebno je omogućiti DSS klijentu da u zahtjevu za rješavanje DNS upita ima informaciju na koju uslugu se upit odnosi
- Provjerava da li polje PRIORITY u svim DSS zapisima ima istu vrijednost. Ako polje PRIORITY ima različite vrijednosti procesira samo DSS zapise sa najvećim prioritetom (najmanjom vrijednosti polja PRIORITY)

- (a) Pokreće mjerenje vremena i na sve A/AAAA IP adrese iz prethodnog koraka svakih INTERVAL milisekundi šalje upite, a čiji je broj definiran REQUEST parametrom, o dostupnosti mrežne usluge na definirani PROTOCOL/PORT
- Nakon isteka TIME perioda od slanja upita provjerava pristigle odgovore (RTT vrijednosti):
    - Ako je pristigao odgovor od minimalno jedne A/AAAA IP adrese DNS klijent pokreće postupak izračuna DNS-metrike koristeći srednju RTT vrijednost kao RESPONSE parametar (u milisekundama) u izračunu metrike. Nakon izračuna DNS-metrike IP adrese se sortiraju redoslijedom od IP adrese s najmanjom metrikom do IP adrese s najvećom metrikom. S IP adresama dobivenim nakon izračuna metrike DNS klijent nastavlja raditi kao s klasičnim DNS odgovorima poštujući dobiveni redoslijed IP adresa
    - Ako nije pristigao niti jedan odgovor od A/AAAA IP adresa na koje je poslan upit, nakon isteka REFRESH perioda algoritam se vraća na točku (a) povećavajući TIME period na dvostruku vrijednost prethodnog TIME perioda (TIME = TIME * 2, eng. *binary exponential backoff*). Maksimalni broj iteracija je 2, nakon čega se procesiraju DSS zapisi sa sljedećim manjim prioritetom (sljedećom većom vrijednosti polja PRIORITY). Ako niti nakon provjere zapisa sa najmanjim prioritetom nema pristiglih odgovora DNS klijent s DNS upitom nastavlja raditi kao s klasičnim DNS upitom bez podrške za DSS. Za poslužitelje koji ne odgovore na upit u TIMEOUT periodu izbacuju se pripadajući DSS zapisi do isteka TTL vremena
- Rezultati DSS algoritma mogu se pohranjivati u privremenu memoriju DNS klijenta uvažavajući TTL vrijednost A/AAAA odnosno DSS zapisa (preporuka je da imaju istu TTL vrijednost) te navodeći PROTOCOL/PORT za koji je napravljen izračun.

Izračun kompozitne DNS-metrike:

Nakon primitka DSS RR zapisa i završetka mjerenja RESPONSE parametra klijent započinje s računanjem kompozitne DNS-metrike za svaku poslužiteljsku IP adresu koja je odgovorila na klijentski upit, a temeljem DSS podataka koje je klijentu posao DNS poslužitelj i izmjerenih RESPONSE parametra. Vrijeme analitičkog izračuna kompozitne DNS-metrike ne utječe u značajnoj mjeri na trajanje implementacije metode na klijentskoj

strani, a koje je u najvećoj mjeri određeno trajanjem faze prikupljanja DSS podataka. Analitički izračun DSS metrike nije računalno zahtjevan i ne doprinosi u značajnoj mjeri dodatnom opterećenju računala klijenta. Metrika (METRIC) za i-ti DSS zapis od ukupno N DSS zapisa za koji je pristigao odgovor na upit računa se:

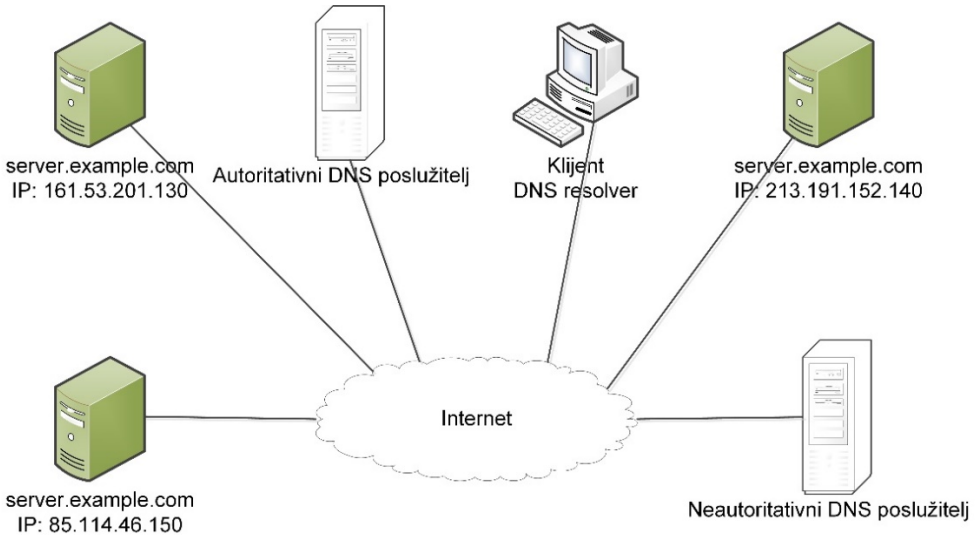
$$METRIC_{(i)} = \begin{cases} \frac{LOAD_{(i)}}{\max\limits_{j\in N} LOAD_{(j)}} + \frac{IMPACT_{(i)}}{255} \times \frac{RESPONSE_{(i)}}{\max\limits_{j\in N} RESPONSE_{(j)}}, \forall\, i \in \{1, \dots, N\}\ \forall \max\limits_{j\in N} LOAD_{(j)} > 0 \\ \frac{IMPACT_{(i)}}{255} \times \frac{RESPONSE_{(i)}}{\max\limits_{j\in N} RESPONSE_{(j)}}, \forall\, i \in \{1, \dots, N\}\ \forall \max\limits_{j\in N} LOAD_{(j)} = 0 \end{cases} \quad (1)$$

Metrika za svaki DSS RR se računa uzimajući u obzir maksimalnu vrijednost parametra LOAD (maxLOAD) primljenom u DSS RR zapisima za određeno domensko ime, i maksimalnim iznosom parametra RESPONSE (maxRESPONSE) dobivenim u postupku mjerenja RTT-a. IMPACT faktor se koristi za određivanje utjecaja mrežne udaljenosti poslužitelja na izračun kompozitne DNS-metrike, pri čemu vrijednost 0 označava najmanji utjecaj (mrežna udaljenost se ne uzima u obzir), a vrijednost 255 označava najveći utjecaj.

Maksimalni mogući iznos DNS-metrike za neku IP adresu ($METRIC_{(i)}$) je 2, koji je moguće dobiti kao zbroj maksimalnog iznosa metrike opterećenja poslužitelja ($LOAD_{(i)} = \max\limits_{j\in N} LOAD_{(j)}$) i maksimalnog iznosa vremena odgovora poslužitelja ($RESPONSE_{(i)} = \max\limits_{j\in N} RESPONSE_{(j)}$) uz maksimalni iznos utjecaja vremena odgovora na izračun metrike ($IMPACT_{(i)} = 255$). Minimalni iznos DNS-metrike je 0, za minimalni iznos metrike opterećenja ($LOAD_{(i)} = 0$) i minimalni iznos utjecaja vremena odgovora na izračun metrike ($IMPACT_{(i)} = 0$). Ideja za uvođenje kompozitne DNS-metrike nastala je promatranjem funkcionalnosti EIGRP[12] usmjerivačkog protokola [65] kod kojeg se četiri parametra koji utječu na izračun kompozitne metrike EIGRP protokola za neku mrežu: propusnost, opterećenje, kašnjenje i pouzdanost, po potrebi koeficijentima uključuju u izračun metrike.

[12] EIGRP – eng. *Enhanced Interior Gateway Routing Protocol*, usmjerivački protokol tvrtke Cisco Systems, kod kojeg se kompozitna metrika za udaljenu mrežu računa kao: *metric = ([K1 * bandwidth + (K2 * bandwidth) / (256 - load) + K3 * delay] * [K5 / (reliability + K4)]) * 256*, pri čemu je predefinirano *K1=K3=1*, *K2=K4=k5=0*

3.5.4. Primjeri izračuna kompozitne DNS-metrike DSS metodom



Slika 1. *Mrežna topologija primjera izračuna kompozitne DNS-metrike DSS metodom*

Na Slici 1. prikazana je mrežna topologija primjera izračuna kompozitne DNS-metrike DSS metodom. DNS klijent koji želi pristupiti mrežnoj usluzi dostupnoj na poslužiteljima sa IP adresama 161.53.201.130, 213.191.152.140 i 85.114.46.150, od autoritativnog ili neautoritativnog DNS poslužitelja uz A zapise dobiva i pripadajuće DSS zapise te pokreće postupak izračuna kompozitne DNS metrike.

Izračun kompozitne DNS-metrike primjenom DSS metode prikazan je na dva primjera sa različitim vrijednostima pojedinih parametara kompozitne DNS-metrike.

Primjer 1:

```
                    DSS parametri                                    RESPONSE METRIC
DSS  161.53.201.130   0   63   63   4 5 TCP HTTP 50 100 1000 0 20       0,411765
DSS  213.191.152.140  0   127  127                                 10       0,664052
DSS  85.114.46.150    0   255  127                                 30       1,498039
```

U Primjeru 1 parametar za izračun kompozitne DNS-metrike max*LOAD* ima vrijednost 255, a parametar max*RESPONSE* ima vrijednost 30. IP adresa 161.53.201.130 ima najmanju metriku iako nema najmanju RESPONSE vrijednost jer je administrator odredio da taj poslužitelj ima najmanje opterećenje (LOAD = 63) i da je utjecaj brzine mrežnog odziva za taj poslužitelj (IMPACT = 63) upola manji od utjecaja brzine mrežnog odziva na izračun DNS-metrike za ostala dva poslužitelja. Parametar TIME ima vrijednost 50 ms što znači da

su svi poslužitelji ušli u postupak izračuna DNS-metrike. TIMEOUT parametar ima vrijednost 1 sekundu (1000 ms) što je značajno kraće od predefiniranog inicijalnog *timeout* parametra za TCP konekciju koji iznosi 21 sekundu. Administrator DNS poslužitelja je odredio da se izračun RESPONSE parametra temelji na slanju 4 upita za testiranje (REQUEST = 4) u razmaku od 5 ms (INTERVAL = 5).

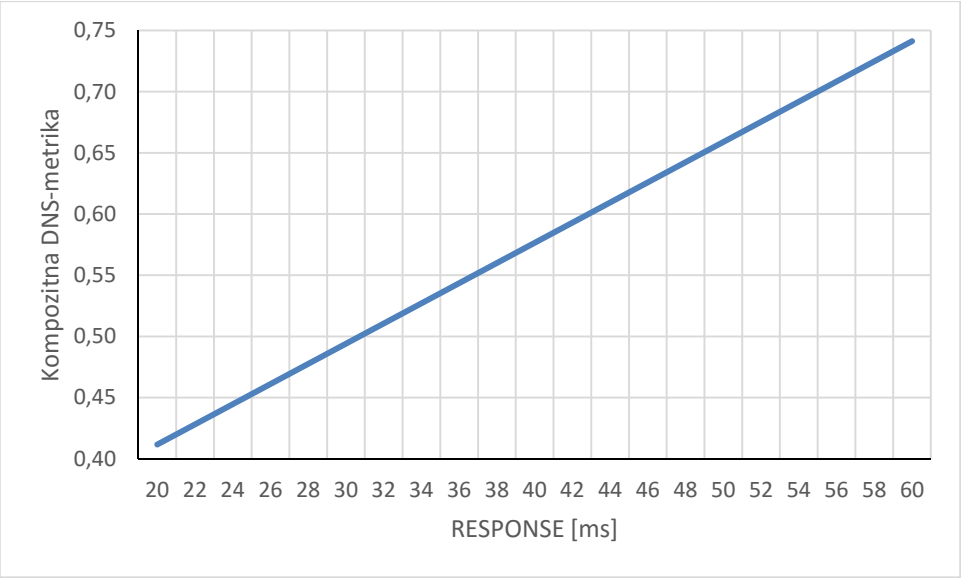
Primjer 2:

```
                    DSS parametri                                    RESPONSE METRIC
DSS  161.53.201.130   0   65   255  4 5 TCP HTTP 50 100 1000 0 15       0,759403
DSS  213.191.152.140  0   127  127                                15       0,749989
DSS  85.114.46.150    0   130  127                                10       0,685471
```

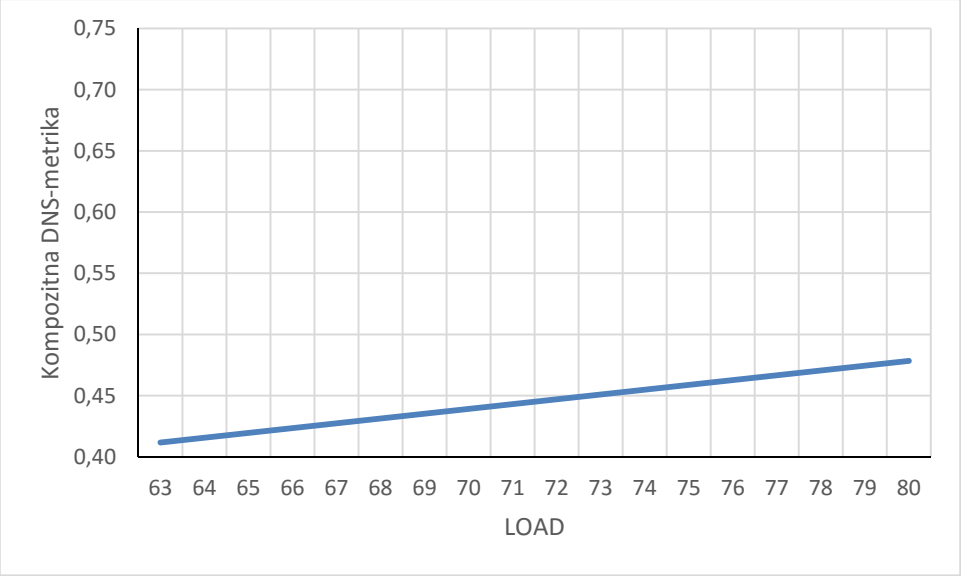
U Primjeru 2 parametar za izračun kompozitne DNS-metrike $\max LOAD$ ima vrijednost 130, a parametar $\max RESPONSE$ ima vrijednost 15. IP adresa 85.114.46.150 ima najmanju DNS-metriku iako ima najveće opterećenje, ali ima najmanju RESPONSE vrijednost te duplo manji parametar IMPACT od poslužitelja 161.53.201.130 koji ima značajno manje opterećenje (50% manje opterećenje) ali za 1/3 veći parametar RESPONSE čime je administrator tom poslužitelju kao primarni parametar za izračun metrike odredio brzinu mrežnog odgovora.

3.5.5. Utjecaj promjene parametara RESPONSE, LOAD i IMPACT na izračun kompozitne DNS-metrike

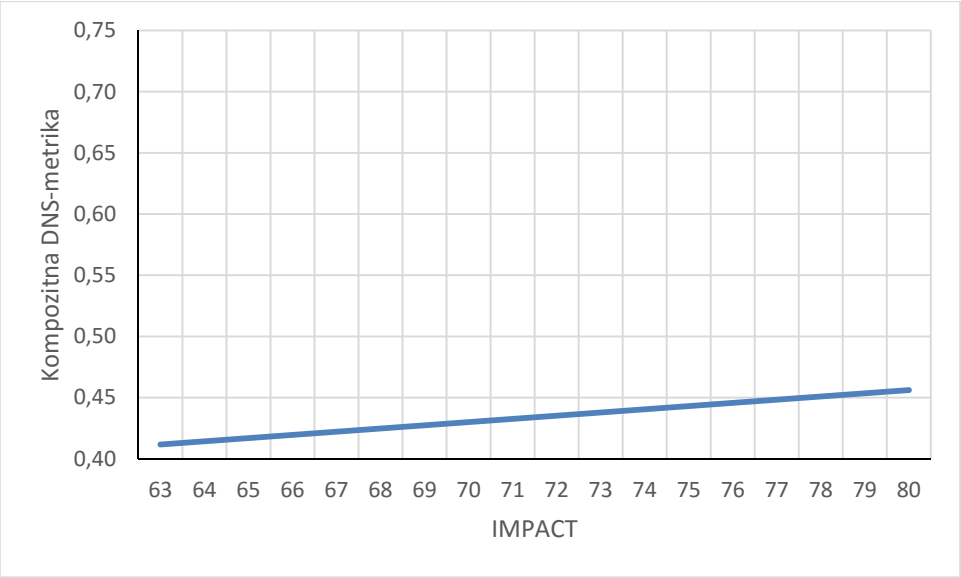
Utjecaj promjene parametara RESPONSE, LOAD i IMPACT na izračun kompozitne DNS-metrike prikazan je pojedinačnom promjenom svakog od promatranih parametara uz konstantan iznos ostala dva parametra kao i istovremenom promjenom parametara RESPONSE i LOAD te RESPONSE i IMPACT uz konstantan treći parametar.



Grafikon 1. *Utjecaj promjene parametra RESPONSE na izračun kompozitne DNS-metrike*



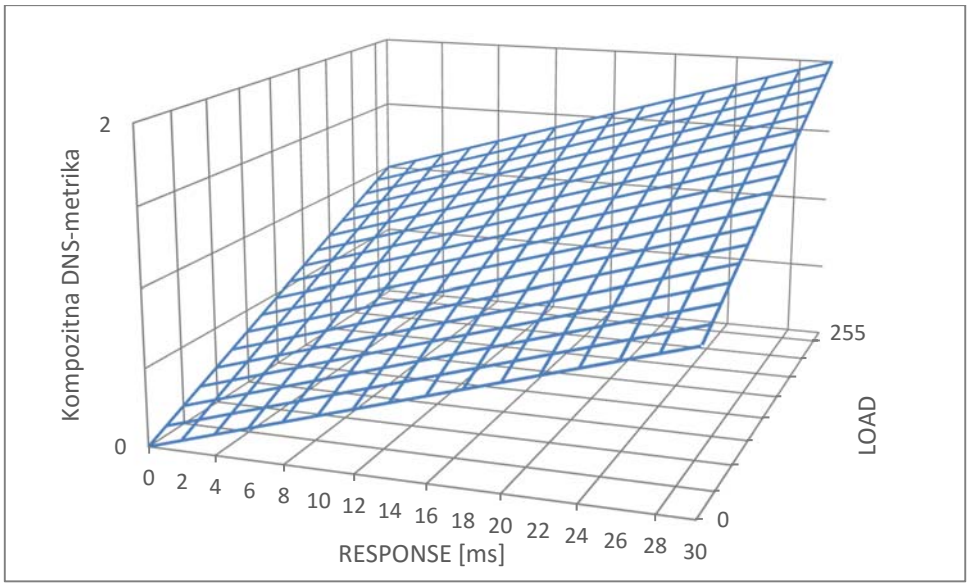
Grafikon 2. *Utjecaj promjene parametra LOAD na izračun kompozitne DNS-metrike*



Grafikon 3. *Utjecaj promjene parametra IMPACT na izračun kompozitne DNS-metrike*

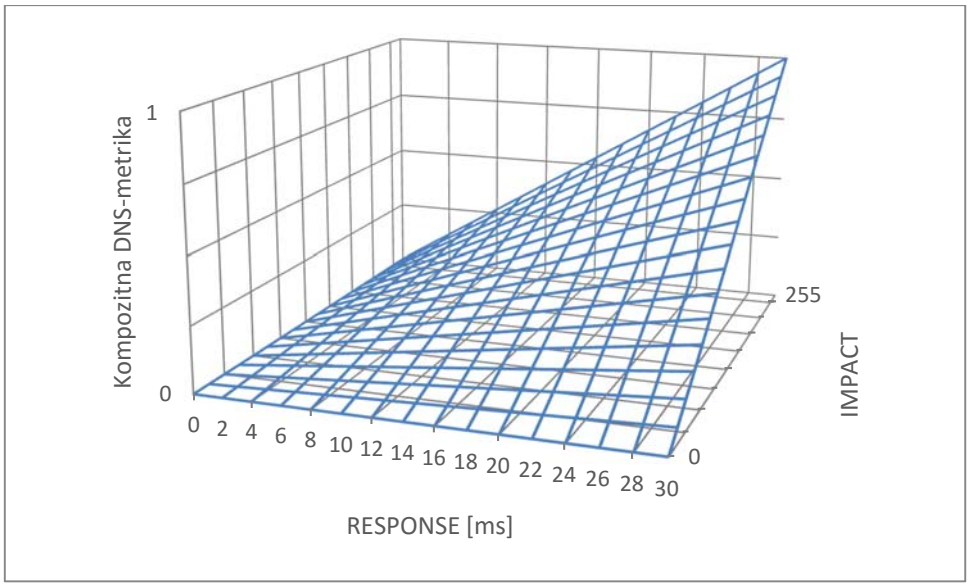
Na Grafikonima 1., 2. i 3. je prikazan utjecaj promjene parametara RESPONSE, LOAD i IMPACT na izračun kompozitne DNS-metrike za poslužitelj 161.53.201.130 iz Primjera 1.

Prvo je promatrana promjena kompozitne DNS-metrike pri promjeni parametra RESPONSE od početnog iznosa 20 ms do završnog iznosa 60 ms uz konstantne iznose parametara LOAD = 63 i IMPACT = 63, a pri čemu se kompozitna DNS-metrika promijenila od početnog iznosa 0,41 do završnog iznosa 0,74. Zatim je, u sljedećem koraku, promatrana promjena parametra LOAD od početne vrijednosti 63 do završne vrijednosti 80, uz konstantne iznose parametara RESPONSE = 20 i IMPACT = 63, a pri čemu se kompozitna DNS-metrika promijenila od početnog iznosa 0,41 do završnog iznosa 0,48. U sljedećem koraku je promatrana promjena parametra IMPACT od početne vrijednosti 63 do završne vrijednosti 80, uz konstantne iznose parametara RESPONSE = 20 i LOAD = 63, a pri čemu se kompozitna DNS-metrika promijenila od početnog iznosa 0,41 do završnog iznosa 0,46.



Grafikon 4. *Utjecaj promjene parametara RESPONSE i LOAD na izračun kompozitne DNS-metrike*

Grafikon 4. prikazuje utjecaj promjene parametara RESPONSE, koji se mijenja od početne vrijednosti 0 ms do završne vrijednosti 30 ms, i parametra LOAD, koji se mijenja od svoje minimalno moguće vrijednosti 0 do maksimalno moguće vrijednosti 255, na izračun kompozitne DNS-metrike. Pri tome parametar max*LOAD* ima vrijednost 255, a parametar max*RESPONSE* ima vrijednost 30, kako je definirano u Primjeru 1 za poslužitelj 161.53.201.130, dok parametar IMPACT ima vrijednost 255. Vrijednost kompozitne DNS-metrike se mijenja od minimalne početne vrijednosti 0, za najmanje vrijeme odgovora mrežne usluge RESPONSE i najmanje opterećenje poslužitelja LOAD, do završne vrijednosti 2, za njihove maksimalne vrijednosti. Parametar IMPACT je, u odnosu na Primjer 1, povećan sa vrijednosti 30 na vrijednost 255 kao bi se u prikazu mogao dobiti maksimalni iznos kompozitne DNS-metrike jer bi sa vrijednošću 30 maksimalni iznos kompozitne DNS-metrike iznosio 1,247058824.

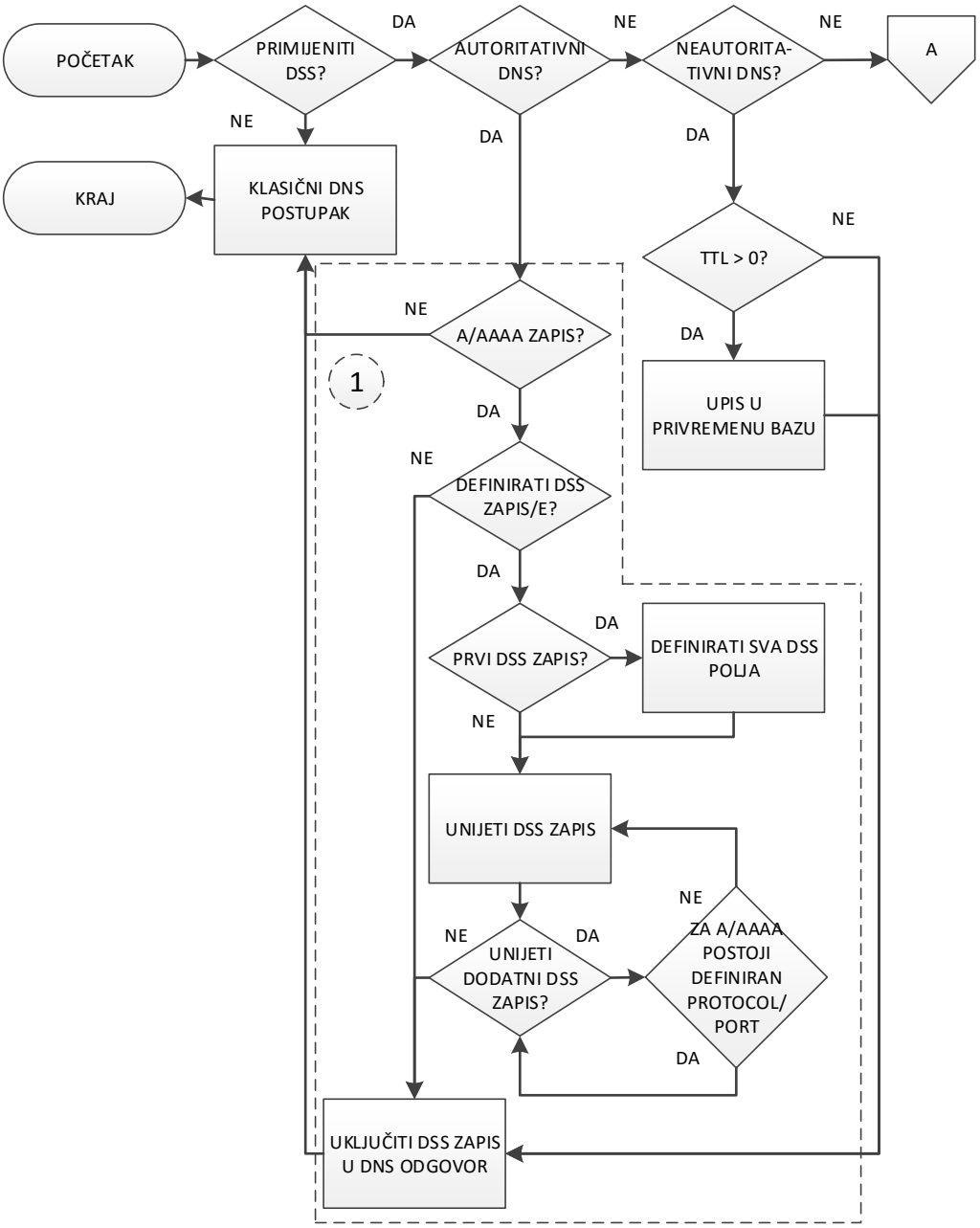


Grafikon 5. *Utjecaj promjene parametara RESPONSE i IMPACT na izračun kompozitne DNS-metrike*

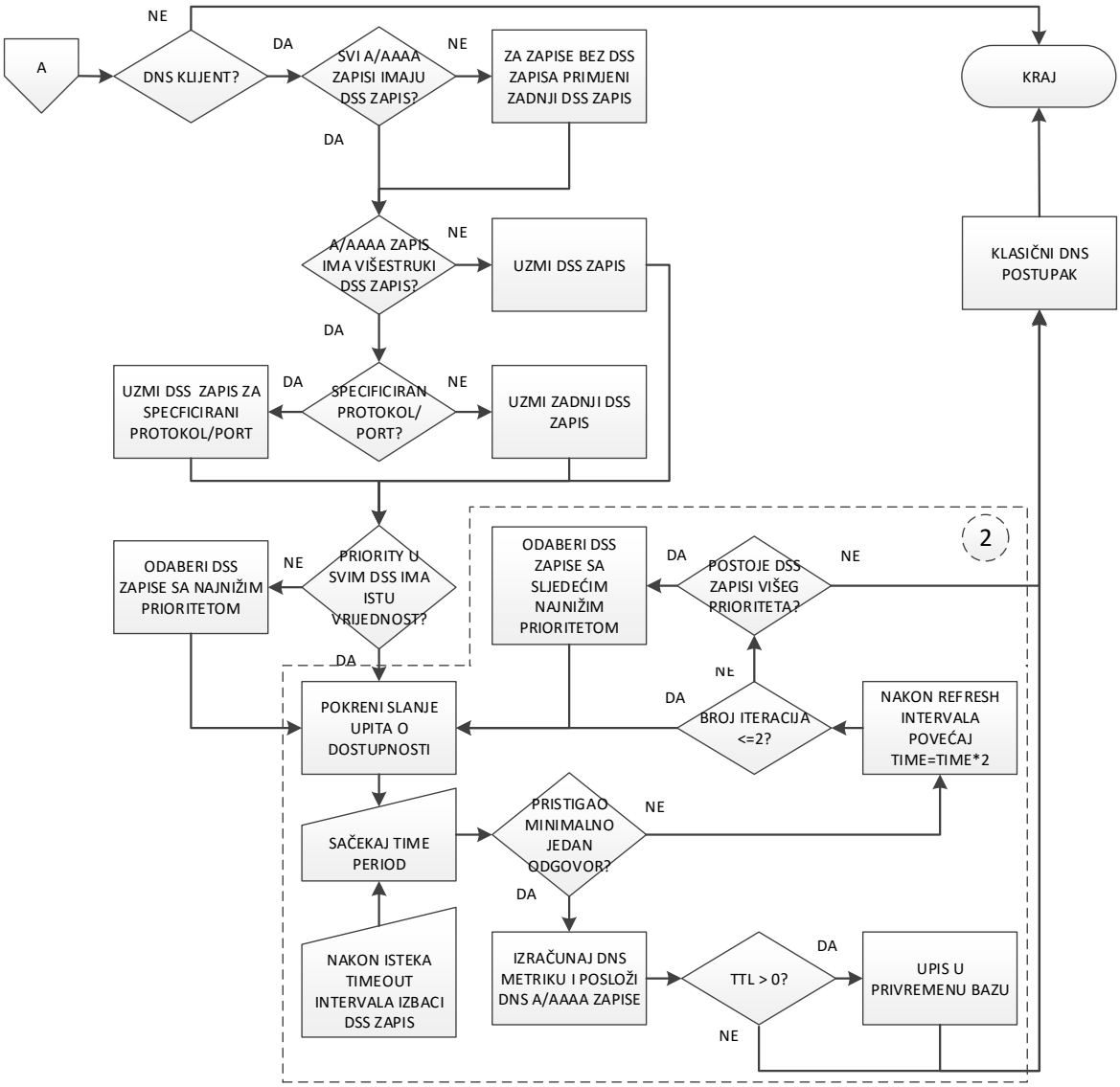
Grafikon 5. prikazuje utjecaj promjene parametara RESPONSE na izračun kompozitne DNS-metrike, koji se mijenja od početne vrijednosti 0 ms do završne vrijednosti 30 ms, i parametra IMPACT koji se mijenja od svoje minimalno moguće vrijednosti 0 do maksimalno moguće vrijednosti 255. Pri tome parametar max*RESPONSE* ima vrijednost 30, kako je definirano u Primjeru 1 za poslužitelj 161.53.201.130. Vrijednost kompozitne DNS-metrike se mijenja od minimalne početne vrijednosti 0, za najmanje vrijeme odgovora mrežne usluge RESPONSE i najmanju vrijednost parametra IMPACT, do završne vrijednosti 1, za njihove maksimalne vrijednosti. Parametru max*LOAD* je, u odnosu na Primjer 1, smanjena vrijednosti sa 255 na vrijednost 0 kao bi se u prikazu isključio utjecaj opterećenja poslužitelja na izračun te prikazao samo utjecaj promatranih parametara na izračun iznosa kompozitne DNS-metrike. Uključivanjem parametra LOAD u prikaz ukupna kompozitna DNS-metrika bi se povećala za fiksni iznos vrijednosti koji može iznositi od 0 do 1, ovisno o iznosu LOAD i maxLOAD parametara.

3.5.6. Dijagram toka DSS metode

Dijagramima toka DSS metode grafički je opisana funkcionalnost i pravila implementacije DSS metode i DSS zapisa u autoritativnim i neautoritativnim DNS poslužiteljima (Dijagram 1.) i DNS klijentima (Dijagram 2.) iz prethodnih poglavlja.



Dijagram 1. *Dijagram toka DSS metode – autoritativni i neautoritativni DNS poslužitelji*



Dijagram 2. *Dijagram toka DSS metode – DNS klijenti*

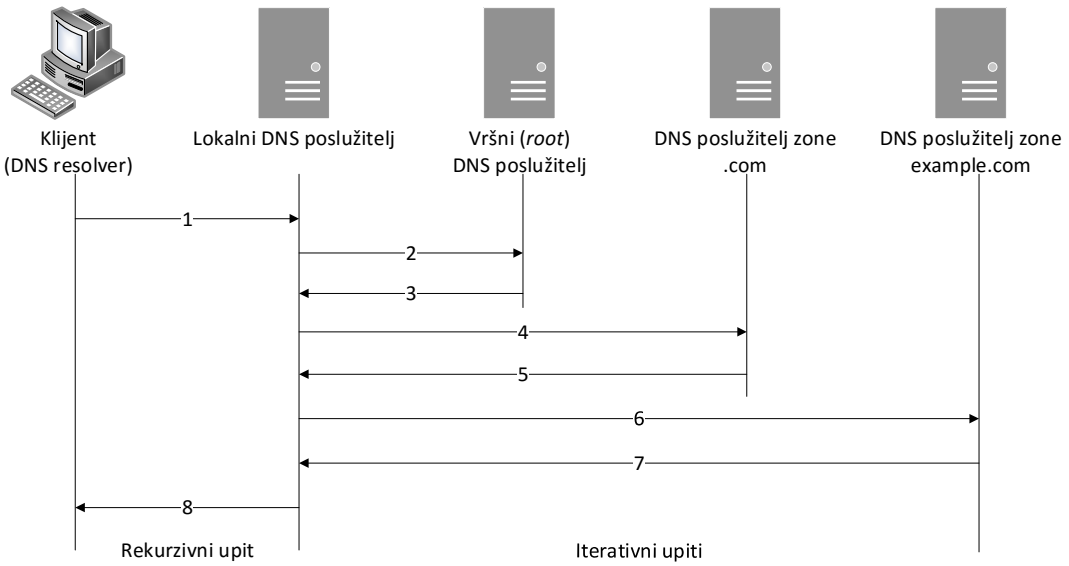
Funkcionalna grupa 1 u Dijagramu 1. predstavlja pravila implementacije DSS metode i kreiranja DSS zapisa u autoritativnom DNS poslužitelju, a što predstavlja prvi korak u primjeni metode za dinamički odabir poslužitelja višestruko dostupne mrežne usluge.

Funkcionalna grupa 2 u Dijagramu 2. predstavlja aktivni dio DSS metode koji se odnosi na prikupljanje RESPONSE parametara za svaki pojedini poslužitelj i izračun kompozitne DNS-metrike temeljem izmjerenih RESPONSE vrijednosti te DSS metodom dostavljenih parametara LOAD i IMPACT. Nakon izračuna kompozitne DNS-metrike će se sortirati IP adrese poslužitelja u rastućem redoslijedu DNS-metrike.

3.5.7. Vremenski slijed komunikacije kod primjene DSS metode

Da bi klijent uspostavio komunikaciju s poslužiteljem mrežne usluge u prvom koraku je potrebno da klijent dobije informaciju o IP adresi/adresama poslužitelja u obliku A/AAAA DNS zapisa, a temeljem domenskog imena poslužitelja. Pri tome se za slanje DNS upita klijenta lokalnom DNS poslužitelju uobičajeno koristi rekurzivni upit dok lokalni DNS poslužitelj za rješavanje postavljenog upita koristi iterativne upite [66]:

- *rekurzivni upiti* se uobičajeno koriste od strane DNS klijenata prema DNS poslužiteljima ili od strane DNS poslužitelja koji su konfigurirani da prosljeđuju sve DNS upite koje ne mogu samostalno riješiti drugom definiranom DNS poslužitelju. Karakteristika rekurzivnog upita je da se kao odgovor očekuje ili traženi RR ili poruka o greški da traženi zapis ili domena ne postoje. Ne dopušta se da DNS poslužitelj kao odgovor ponudi neki drugi DNS poslužitelj koji bi mogao odgovoriti na postavljeni upit nego, ukoliko DNS poslužitelj nema traženu informaciju, mora pitati ostale DNS poslužitelje sve dok ne dobije informaciju o traženom zapisu ili poruku o greški
- *iterativni upit* je oblik DNS upita kod kojeg DNS klijent (to može biti i neki DNS poslužitelj) dopušta upitanome DNS poslužitelju da vrati najbolji mogući odgovor koji ima temeljem podataka svojih zona ili sadržaja DNS međuspremnika. Ako upitani DNS poslužitelj nema točan odgovor na postavljeni upit može odgovoriti upućivanjem na DNS poslužitelj koji ima specifičnije, tj. točnije podatke. Pri tome se postupak slanja upita nastavlja sve do dobivanja odgovora ili poruke o greški

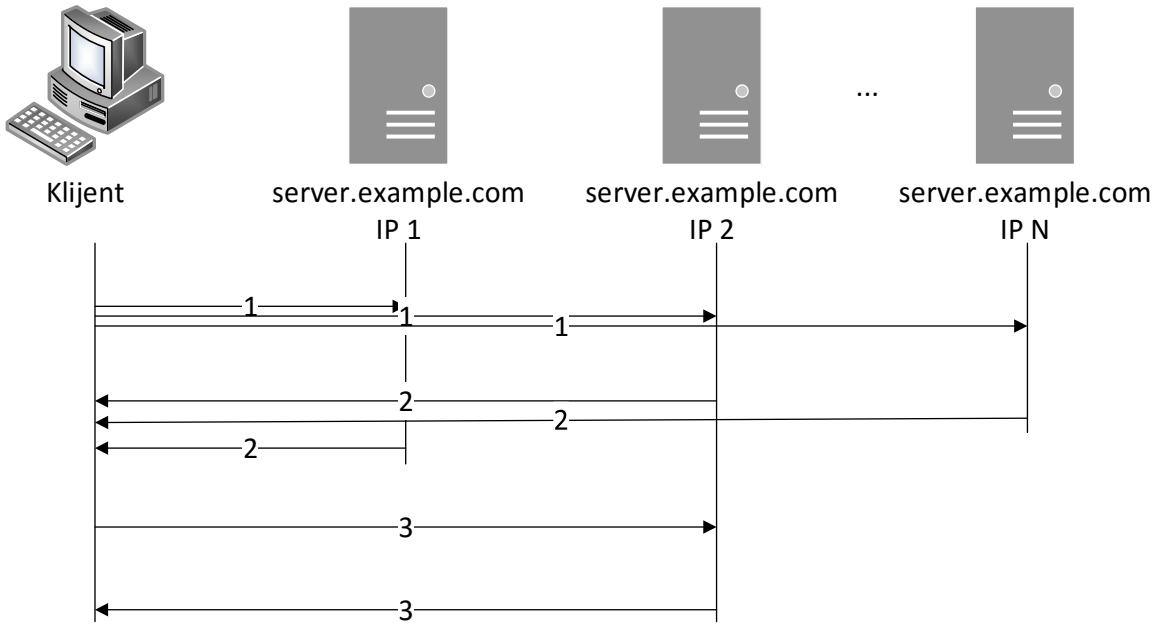


Slika 2. *Primjer iterativnog i rekurzivnog DNS upita*

Slika 2. prikazuje primjer iterativnog i rekurzivnog DNS upita pri čemu se podrazumijeva da niti jedan od promatranih DNS poslužitelja nema odgovor na postavljeni zahtjev u svojem DNS međuspremniku. U primjeru klijent želi uspostaviti komunikaciju s poslužiteljem server.example.com. Postupak dobivanja IP adrese se odvija u 8 faza:

1. Klijent kontaktira svoj lokalno definirani DNS poslužitelj rekurzivnim upitom za A zapis poslužitelja server.example.com. pri čemu DNS poslužitelj mora odgovoriti ili traženim zapisom ili porukom o greški
2. Lokalni DNS poslužitelj provjerava svoje zone i međuspremnik tražeći odgovor i kako ga ne pronalazi, upućuje upit poslužitelju autoritativnom za Internet (vršni poslužitelj) sa iterativnim upitom za server.example.com.
3. Vršni DNS poslužitelj ne zna odgovor na postavljeni upit pa odgovara sa informacijom o DNS poslužitelju autoritativnom za .com zonu (domenu)
4. Lokalni DNS poslužitelj upućuje iterativni upit za server.example.com. DNS poslužitelju autoritativnom za zonu .com
5. DNS poslužitelj autoritativan za zonu .com ne zna potpuni odgovor pa odgovora sa informacijom o DNS poslužitelju autoritativnom za example.com domenu
6. Lokalni DNS poslužitelj upućuje iterativni upit za server.example.com. DNS poslužitelju autoritativnom za zonu example.com
7. Autoritativni DNS poslužitelj domene example.com zna točan odgovor i odgovora sa IP adresom/adresama
8. Lokalni DNS poslužitelj odgovara na klijentski upit sa IP adresom/adresama za server.example.com.

Kod primjene DSS metode podrazumijeva se da će kao odgovor na postavljeni upit pristići minimalno dvije IP adrese i pripadajući DSS zapisi u dodatnoj sekciji odgovora na traženi A/AAAA zapis. Prije nego klijent započne komunikaciju sa jednim od poslužitelja, a čiju je IP adresu dobio kao odgovor na postavljeni upit za pristup mrežnoj usluzi na adresi server.example.com., klijent temeljem parametara iz DSS zapisa uspostavlja testnu komunikaciju sa poslužiteljima na svim dostavljenim IP adresama.



Slika 3. *Komunikacija klijenta sa poslužiteljima višestruke mrežne usluge uz primjenu DSS metode*

Na Slici 3. prikazana je komunikacija klijenta sa poslužiteljima višestruke mrežne usluge uz primjenu DSS metode. Komunikacija se odvija u tri faze:

1. Temeljem IP adresa i DSS zapisa koje je dobio kao odgovor na postavljeni DNS upit, klijent svakih INTERVAL milisekundi šalje upite, a čiji je broj definiran REQUEST parametrom, na definirani PROTOCOL/PORT IP adresa poslužitelja koji imaju najmanju vrijednost polja PRIORITY tj. imaju najveći prioritet
2. Nakon isteka TIME parametra temeljem pristiglih odgovora računa se RESPONSE parametar (u milisekundama) svakog od N poslužitelja korištenjem njegove srednje RTT vrijednosti, a nakon toga i kompozitna DNS metrika
3. Podatkovnu komunikaciju klijent ostvaruje s poslužiteljem koji ima najmanju kompozitnu DNS metriku.

3.5.8. Podrška za IPv6 protokol

DSS metoda svojim osnovnim dizajnom podržava IPv6 protokol jer informaciju o IP adresi, neovisno o tome da li se radi o IPv4 ili IPv6 adresi, u DSS zapisu dostavlja kao pokazivač na odgovarajući A/AAAA zapis u sekciji odgovora.

Veličina UDP datagrama koji sadrži DNS odgovor za AAAA zapis se povećava za 12 bajtova po AAAA zapisu, kolika je razlika u veličini između IPv4 (32 bita) i IPv6 (128 bita) adrese. Pri tome veličina DSS zapisa ostaje nepromijenjena.

Ukupna veličina DNS odgovora za upit „server.example.com.“ iz točke 3.4.1, ako bi se umjesto A tražio AAAA zapis koji bi u odgovoru imao 4 AAAA zapisa i 4 DSS zapisa, bila bi 12 + 24 + (64 + 12 x 4) + 128 = 276 bajtova što je još uvijek značajno manje od preporučene maksimalne veličine DNS UDP datagrama od 512 bajtova. Kada se u A/AAAA polju ne bi koristio 16 bitni pokazivač nego 128 bitna IP adresa ukupna veličina DNS odgovora iz primjera bi bila 276 + 4 x 14 = 332 bajtova (veličina jednog DSS zapisa se u tom slučaju povećava za 14 bajtova).

3.5.9. Podrška za višedomne klijentske sustave

DSS metoda podržava višedomne (eng. *multihomed*) sustave na klijentskoj strani na način da, ako je DNS klijent svjestan višestrukih izlaza (eng. *gateway*[13]) na lokalnoj mreži, može slati upite o dostupnosti mrežne usluge preko višestrukih izlaza paralelno te sve dobivene rezultate uključiti u izračun kompozitne DNS-metrike. U tome slučaju DNS klijent bi, uz informaciju o IP adresama, kao parametar morao proslijediti i informaciju o *gateway*-u s kojeg je odgovor došao.

3.5.10. Zahtjevi za resursima kod implementacije DSS metode

DSS metoda predstavlja proširenje i nadopunu postojećeg DNS sustava i nema specifičnih zahtijeva za resursima na DNS poslužiteljima, klijentima, poslužiteljima višestruke mrežne usluge i temeljenoj mrežnoj infrastrukturi u procesima implementacije i upotrebe metode.

Uobičajena primjena DSS metode klijentima omogućuje odabir optimalnog poslužitelja između od minimalno dva do preporučeno ne više od desetak poslužitelja višestruko dostupne mrežne usluge. Promjene koje se unose u postojeći DNS sustav, a odnose se na proširenje zahtjeva za resursima, su minimalne i sadrže:

[13] *Gateway* – točka izlaska računalne mreže u drugu računalnu mrežu

- DNS poslužitelj: dodavanje dodatna 32 bajta u dodatnu sekciju DNS odgovora za svaki poslužitelj višestruke mrežne usluge uključen u DSS metodu
- klijenti: inicijalno testiranje svakog poslužitelja upitima veličine do preporučeno 64 bajta, a čiji je ukupan broj definiran REQUEST parametrom, svakih INTERVAL milisekundi, te numerički izračun kompozitne DNS metrike prema jednadžbi (1)
- poslužitelji: generiranje odgovora na inicijalne upite klijenata u svrhu utvrđivanja RESPONSE parametra
- temeljena mrežna infrastruktura: prijenos dodatna 32 bajta u DNS odgovoru za svaki poslužitelj uključen u DSS metodu i prijenos inicijalnih upita klijenata i odgovora poslužitelja u svrhu utvrđivanja RESPONSE parametra

3.6. ZAKLJUČAK

Uvođenjem predložene DSS metode za dinamički odabir poslužitelja višestruke mrežne usluge omogućuje se klijentu, temeljem informacija dobivenih kroz DNS sustav, samostalno određivanje poslužitelja koji će mu pružiti najbrži odgovor na postavljeni upit. U izračun kompozitne DNS-metrike, kao mjere za određivanje najpogodnijeg poslužitelja, pri tome mogu biti uključeni opterećenje i vrijeme mrežnog odziva poslužitelja. Za implementaciju metode potrebno je napraviti prilagodbe u DNS poslužiteljima i klijentima.

4. MODEL POVEZIVANJA KOMPOZITNE DNS-METRIKE S ANALITIČKIM IZRAČUNOM VREMENA ODGOVORA POSLUŽITELJA

Cilj izrade modela povezivanja kompozitne DNS-metrike mrežne usluge s analitičkim izračunom vremena odgovora poslužitelja je za kompozitnu DNS-metriku poslužitelja mrežne usluge prikazanu jednadžbom (1) napraviti analitički izračun vremena potrebnog za odgovor poslužitelja na postavljeni upit, pri čemu se za komunikaciju koristi TCP protokol kao uobičajeni i dominantni mrežni protokol u suvremenim računalnim mrežama. Pri tome je parametar povezivanja vrijeme odziva mrežne usluge (RTT) kao zajednički parametar izračuna kompozitne DNS metrike i analitičkog izračuna vremena odziva poslužitelja. Povezivanjem kompozitne DNS-metrike s analitičkim izračunom vremena odgovora poslužitelja omogućuje se administratorima višestruke mrežne usluge predviđanje trajanja vremena odgovora poslužitelja mrežne usluge za izračunatu kompozitnu DNS-metriku jer je cilj DSS metode omogućiti klijentu što kraće vrijeme odgovora mrežne usluge na način da poslužitelju koji će to omogućiti dodijeli najmanji iznos kompozitne DNS-metrike.

4.1. PROPUSNOST I KAŠNJENE U RAČUNALNIM MREŽAMA

Brzina mrežne konekcije kako je vidi klijent, a očituje se u vremenu proteklom od slanja zahtjeva za pristup nekoj mrežnoj usluzi do trenutka primitka odgovora mrežne usluge na postavljeni zahtjev, ovisi o kombinaciji dva osnovna parametra [67][68]:

- propusnosti mrežne konekcije (*bandwidth*)
- kašnjenja (*latency*)

Propusnost u računalnim mrežama označava koliko se podataka može prenijeti u jedinici vremena, odnosno koliki je kapacitet prijenosa podataka mrežne konekcije ili mrežnog sučelja. Mjeri se bitovima po sekundi (*bit/s*) i uobičajeno predstavlja primarnu mjeru brzine računalne mreže. Što je veća propusnost veća je i vjerojatnost da će i performanse mrežne konekcije biti bolje.

Kašnjenje je vrijeme potrebno za slanje podataka od izvorišta do odredišta, uobičajeno se mjeri u milisekundama (*ms*), a tijekom mjerenja brzine često se naziva i *ping rate* te označava vrijeme kružnog putovanja (RTT) podatkovnih paketa.

Što je propusnost veća, a kašnjenje manje, korisnik će dobiti brži odgovor, npr. ostvariti brže učitavanje web stranice ili ostvariti brži prijenos datoteke. Važnost smanjenja kašnjenja, a time i smanjenja vremena odgovora mrežne usluge, povezana je s navikama korisnika koji toleriraju sve manja odnosno kraća kašnjenja, tako da npr. korisnici počinju odbacivati zahtjeve za pokretanjem videa već dvije sekunde od slanja zahtjeva, a svaka dodatna sekunda kašnjenja u početku njegove reprodukcije povećava odbacivanje za 5,8% [69]. I dva nova trenda koja zahtijevaju komunikaciju s manjim kašnjenjima – podatkovni centri i aplikacije sa iznimno malim kašnjenjem, nameću kašnjenje kao primarnu metriku za računalne mreže nove generacije [70]. Dodatni razlog za potrebu ubrzavanja učitavanja web stranica, kao dominantne mreže usluge, je i činjenica da brže učitavanje sadržaja web stranice može dovesti do boljeg rangiranja stranice na tražilicama [71].

Osnovni uzroci kašnjenja, gledajući sa klijentske strane, su:

- *tip konekcije* odnosno mrežna tehnologija: npr. uobičajeni *ping rate* za satelitske konekcije je nekoliko stotina ms, dok je za uobičajene kablovske ili ADSL konekcije nekoliko desetaka ms [72]:

Ethernet	0.3ms
DS1/T1	2-5ms
DSL/Cable	10-20ms
ISDN	15-30ms
Analogni Modem	100-200ms
Stacionarni sateliti	>500ms

- *udaljenost*: što je veća udaljenost između izvorišta i odredišta potrebno je duže vrijeme za prijenos podataka komunikacijskim linkom. Vrijeme potrebno da svjetlost obiđe Zemlju je približno 134 ms. Kada se u obzir uzmu refrakcije u optičkom kabelu i druge neefikasnosti, RTT i u najboljim mrežama može doseći 300 ms [73].
- *zagušenje*: vezano za propusnost, što je manja raspoloživa propusnost to će podaci duže čekati na prijenos komunikacijskim medijem

Analiza vrijednosti RTT-a za pristup web stranicama pokazuje da su osnovi uzroci povećanog kašnjenja [74]:

- neadekvatni algoritmi dodjele poslužitelja klijentima
- korištenje drugih kriterija (npr. opterećenja), a ne RTT-a za dodjelu poslužitelja klijentima od strane davatelja usluga
- komponente uključene u posluživanje klijentskih zahtjeva nemaju adekvatne resurse ili imaju neoptimiziranu konfiguraciju
- manji davatelji usluga nemaju poslužitelje na višestrukim lokacijama
- povremene greške u konfiguraciji.

Postoji uzročno-posljedična povezanost propusnosti i kašnjenja, odnosno jedan parametar utječe na funkcioniranje drugoga. Što je veća propusnost veća je vjerojatnost da će kašnjenje biti manje, uobičajeno zbog primijenjene tehnologije. Ako je propusnost u cijelosti iskorištena, to će dovesti do zagušenja, a time i povećanja kašnjenja ali istovremeno postojanje neiskorištenog dijela propusnosti ne mora značiti i da će se kašnjenje smanjiti.

Postojanje značajnih kašnjenja na mreži može dovesti do nemogućnosti punog iskorištenja propusnosti pri čemu utjecaj na mrežnu propusnost može biti privremen, u trajanju do nekoliko sekundi, ili konstantan. Različite vrste konekcija mogu imati različite vrijednosti propusnosti i kašnjenja, tako npr. satelitske internetske konekcije imaju i veliku propusnost ali i veliko kašnjenje. Propusnost se uvijek može povećati ali se kašnjenje ne može uvijek smanjiti, pogotovo kašnjenje koje je vezano za fizikalnu granicu brzine rasprostiranja signala kroz komunikacijski medij, a koje zajedno sa kašnjenjem obrade u aktivnoj mrežnoj opremi čini najveći dio ukupnog kašnjenja.

Propusnost računalne mreže se povećava značajno brže nego što se u njoj smanjuje kašnjenje, iskustveno je pokazano da se u vremenskom intervalu u kojem se propusnost poveća za dvostruku vrijednost kašnjenje smanji samo za faktor 1.2-1.4, tj. poboljšanje propusnosti se minimalno kvadratno povećava u odnosu na poboljšanje u kašnjenju, pri čemu navedeno pravilo vrijedi i za mikroprocesore, tvrde diskove i memorije [75][76].

LAN	Ethernet	Fast Ethernet	Gigabit Ethernet	10 Gigabit Ethernet
IEEE standard	802.3	802.3u	802.3ab	802.3ae
Godina	1978	1995	1999	2003
Propusnost [Mbit/s]	10	100	1000	10000
Kašnjenje [μs]	3000	500	340	190

Tablica 2. *Promjene propusnosti i kašnjenja u računalnim mrežama*

U periodu od pojave IEEE 802.3 standarda do pojave IEEE 802.ae standarda propusnost LAN-a je povećana 1000 puta dok se kašnjenje smanjilo samo 15 puta. Dok se problemi propusnosti mogu riješiti većim financijskim ulaganjima većina problema vezanih za kašnjenje je limitirana granicom brzine svjetlosti pa moguća poboljšanja u smanjivanju kašnjenja trebaju uključivati i metode upotrebe međuspremnika, repliciranja i predviđanja. Upravo je DSS metoda usmjerena na smanjenje kašnjenja prilikom pristupa višestruko dostupnim (repliciranim) mrežnim uslugama odabirom poslužitelja mrežne usluge za koji klijent ima najmanje kašnjenje.

Kako se u ovom radu konekcija prema mrežnoj usluzi promatra sa stanovišta klijenta, onda je u određivanju brzine mrežne konekcije propusnost klijentske internetske veze konstantna, a kašnjenje koje se ostvaruje prema svakom od poslužitelja višestruko dostupnih mrežnih usluga predstavlja parametar koji utječe na brzinu odziva mrežne usluge.
Ukupno kašnjenje, od izvorišta do odredišta i od odredišta do izvorišta, a koje zbog prirode IP protokola i karakteristika mrežne putanje ne mora biti identično u oba smjera, sastoji se od četiri vrste kašnjenja:

- kašnjenja obrade (*processing delays*, D_{Pr}) – vremena potrebna za analizu zaglavlja paketa i donošenja odluke o postupanju s paketom
- kašnjenja u međuspremnicima (*buffer delays*, D_B) – vremena koja paket provede u međuspremnicima prije slanja
- kašnjenja prijenosa (*transmission delays*) – vremena potrebna za stavljanje svih bitova paketa na komunikacijski medij. Za N bitova koji se prenose i brzinu transmisije R kašnjenje prijenosa D_T je:

$$D_T = N/R \tag{2}$$

- kašnjenja rasprostiranja (*propagation delays*) – vremena propagacije signala kroz fizički medij, ovisi o dužini fizičkog medija i limitiran je brzinom svijetlosti. Za udaljenost *d* i brzinu propagacije *s* kašnjenje rasprostiranja D_P je

$$D_P = d/s \tag{3}$$

Ukupno kašnjenje u mreži je zbroj svih kašnjenja:

$$D = D_{Pr} + D_B + D_T + D_P \tag{4}$$

odnosno:

$$D = D_{Pr} + D_B + N/R + d/s \tag{5}$$

U ovome radu ukupna kašnjenja u mreži *D*, a odnose se na ukupna kašnjenja u oba smjera, izražena su kroz jedinstvenu vrijednost RTT-a. Kako bi se što je moguće više izbjegao utjecaj zagušenja, odnosno kašnjenja u međuspremnicima (D_B), praksa je da se prilikom mjerenja RTT-a šalju mali podatkovni paketa, uobičajeno 32 bajta, a što je primijenjeno i u DSS metodi.

4.2. MATEMATIČKI MODEL PROPUSNOSTI TCP PROTOKOLA

Kod korištenja TCP protokola za prijenos podataka između dva hosta pomoću vrijednosti kašnjenja (RTT) i veličine TCP prozora (*TCP window size*)[14], može se izračunati maksimalna moguća propusnost prijenosa podataka, neovisno o propusnosti komunikacijske mreže koja je na raspolaganju [77]:

$$B = \frac{W}{RTT} \tag{6}$$

gdje je:

B – maksimalna propusnost TCP protokola [bit/s]

W – TCP veličina prozora [bit]

RTT – kašnjenje [s]

[14] *TcpWindowSize* parametar: Za Windows operativni sustav kod Ethernet mreža predefinirana vrijednost je 17.520 bajtova ili 12 segmenata po 1.460 bajtova svaki. Za sve ostale mreže predefinirana vrijednost je 65.535 bajtova (64 KB), moguća su odstupanja u nekim verzijama operacijskih sustava

Iz navedenoga je vidljivo da je jedini način da se ubrza prijenos podataka između hostova, uz pretpostavku da je raspoloživa propusnost komunikacijskog linka između hostova dovoljno velika, povećanje veličine prozora i(li) smanjenje kašnjenja. To znači da veličina prozora izravno utječe na propusnost komunikacije i da se za veće vrijednosti W mogu postići veće vrijednosti propusnosti, odnosno da se s povećanjem RTT-a propusnost smanjuje.

Da bi se iskoristila maksimalna moguća propusnost TCP protokola potrebno je imati međuspremnike na strani klijenta i poslužitelja koji mogu privremeno pohraniti cijeli prozor nepotvrđenih podataka, izraženih kao vrijednost umnoška propusnosti i kašnjenja (BDP – *Bandwidth-Delay Product*), a pri čemu se treba uzeti najsporiji link u mrežnoj putanji [78]. U slučaju da je BDP veći od maksimalno dopuštene veličine TCP prozora TCP konekcija neće biti u mogućnosti u cijelosti popuniti link i postići maksimalnu propusnost [79]. Da bi se riješio ovaj problem, pogotovo kod konekcija koje imaju veliki RTT, TCP uvodi opciju skaliranja prozora kako bi se povećala njegova veličina. Skaliranje se dogovara na početku konekcije, u paketima sa podignutom SYN zastavicom, iskorištavanjem svojstva da je veličina prozora u paketu 16 bitova, a u hostu 32 bita [80].

Uz opciju skaliranja prozora, još dvije opcije TCP protokola utječu na brzinu prijenosa podataka:

- selektivno potvrđivanje [81][82]: omogućuje potvrdu pojedinačnih paketa i prijenos samo onih paketa iz prozora koji nisu došli na odredište. S tim je povezana i vjerojatnost gubitka paketa koja ovisi o tehnologiji i trenutnim uvjetima na mreži (BER - *Bit Error Rate*, PER - *Packet Error Rate*)
- eksplicitna kontrola zagušenja (ECN - *Explicit Congestion Notification*): omogućuje razlikovanje gubitka paketa zbog zagušenja u odnosu na gubitak/oštećenje paketa u transportu pri čemu se kod zagušenja prepolovljuje brzina slanja paketa (promjenom veličine prozora) za razliku od gubitka/oštećenja gdje se zadržava brzina slanja paketa odnosno veličina prozora [83].

Općenito promatrano, da bi se povećala propusnost računalne mreže, uz zadovoljen BDP uvjet, potrebno je smanjiti RTT, smanjiti vjerojatnost gubitka paketa i povećati maksimalnu veličinu segmenta (MSS[15] – *Maximum Segment Size*).

4.2.1. Standardni matematički modeli za izračun parametara TCP protokola

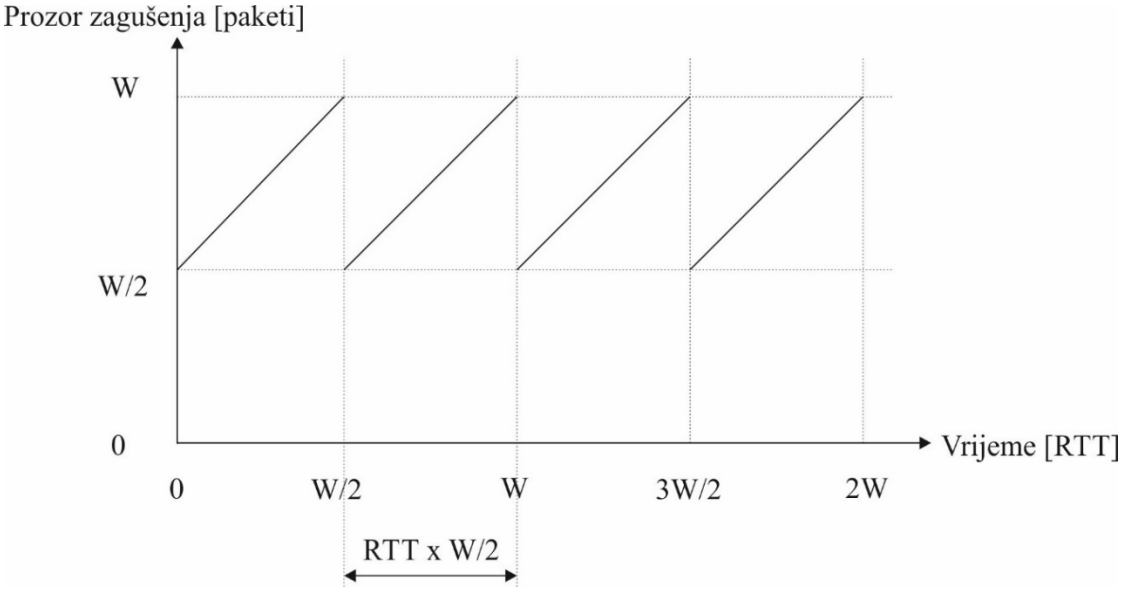
Analizom standardnih matematičkih modela koji se uobičajeno koriste za modeliranje TCP konekcija velike ili proizvoljne dužine trajanja [85], a to su Mathisov, Padhyev i Cardwellov model, te razmatranjem njihove primjene za modeliranje TCP konekcije u ovom radu utvrđeno je:

1. **Mathisov model** [86] se u praksi često koristi zbog svoje jednostavnosti. Propusnost Mathisovim modelom, za definiranu maksimalnu veličinu segmenta *MSS* i vrijeme kružnog putovanja *RTT*, te uz konstantnu vjerojatnost slučajnog gubitka paketa p i konstantu proporcionalnosti za periodičke gubitke i potvrdu svakog paketa $C = \sqrt{3/2}$, izračunava se:

$$B = \frac{MSS}{RTT} \times \frac{C}{\sqrt{p}} \tag{7}$$

   Na Slici 4. prikazana je promjena (smanjenje) veličine TCP prozora W kod pojave periodičkih gubitaka. U svakom ciklusu koji traje vremenski interval $\frac{W}{2}$ RTT-a isporučuje se $\left(\frac{W}{2}\right)^2 + \frac{1}{2}\left(\frac{W}{2}\right)^2 = \frac{1}{p}$ paketa.

[15] MSS: maksimalna veličina segmenta, fiksna za svaku mrežnu putanju, uobičajeno 1.460 bajtova (1.500 bajtova MTU (*Maximum Transmission Unit*) umanjeno za 40 bajtova IP i TCP zaglavlja) [84]



Slika 4. *Mathisov TCP prozor pri periodičkim gubicima*

Mathisov model nije dobar za modeliranje TCP konekcija koje imaju gubitke veće od 2% [87], jer u takvim slučajevima predimenzionira propusnost (promatra samo fazu izbjegavanja zagušenja i TD (*Triple Duplicate*) periodičke gubitke). Kod primjene DSS metode promatraju se mrežne konekcije klijenata koji se spajaju na mrežu različitim tehnologijama, pa i onim s većim BER vrijednostima, kao i tehnologijama koje vrlo često imaju dijeljene, a sukladno tome i potencijalno zagušene komunikacijske medije, tako da model mora biti primjenjiv i za gubitke veće od 2% što kod Mathisovog modela nije slučaj.

2. **Padhyev model** [88][89] za ustaljenje TCP konekcije, baziran na TCP Reno implementaciji, u cijelosti odgovara zahtjevima TCP konekcije promatrane ovim radom, a to su mogućnost primjene za ustaljene TCP konekcije i primjenjivost modela za veće gubitke. U obzir uzima i TO (*Time-Out*) i TD gubitke u CA (*Collision Avoidance*) fazi kao varijabilne (*bursty*) gubitke tijekom prijenosa te je u mogućnosti predvidjeti propusnost u širem rasponu gubitaka. Komunikacija između klijenta i poslužitelja se promatra u rundama gdje je trajanje runde jednako trajanju RTT-a.
Aproksimativna, pojednostavljena propusnost Padhyevog modela, izražena u paketima u jedinici vremena je:

(8)

$$B(p) \approx \min\left(\frac{W_{max}}{RTT}, \frac{1}{RTT\sqrt{\frac{2bp}{3}} + T_0 min\left(1{,}3\sqrt{\frac{2bp}{8}}\right)p(1+32p^2)}\right)$$

Pri tome je W_{max} maksimalna veličina prozora TCP konekcije, *RTT* vrijeme kružnog putovanja paketa (trajanje runde), *b* broj paketa potvrđen sa primljenim ACK paketom (uobičajeno 2), *p* vjerojatnost da je paket izgubljen, a T_0 RTO vrijednost, uobičajeno 3 sekunde.

3. **Cardwellov model** [90], kao najpotpuniji od promatranih modela omogućuje promatranje TCP konekcija različite dužine. Razvijen je kao proširenje Padhyevog modela i uvodi u proračun, osim faze ustaljenog stanja opisane Padhyevim modelom, i faze uspostave konekcije (CE – *Connection Establishment*) i sporog starta (SS – *Slow Start*) te pojavu zakašnjele potvrde.
   Izračun kašnjenja Cardwellovim modelom:

   Kašnjenje prijenosa podataka *D[p]*, odnosno vrijeme potrebno za dovršetak prijenosa segmenata veličine *N* se računa kao suma četiri komponente:

   $$D[p] = E[D_{ss}] + E[D_{loss}] + E[D_{ca}] + E[D_{delack}] \quad (9)$$

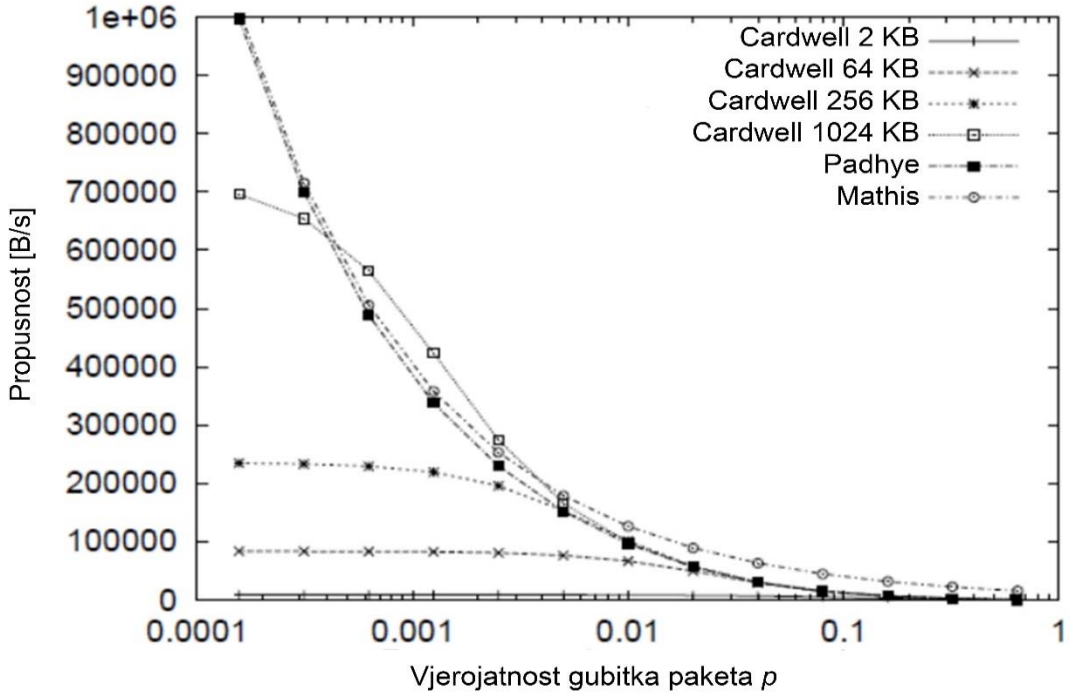
   Pri tome je D_{ss} kašnjenje inicijalnog sporog starta, D_{loss} je očekivan utjecaj TO gubitaka ili brzih oporavaka koji se pojavljuju na kraju inicijalnog sporog starta, D_{ca} je očekivano vrijeme potrebno za prijenos preostalih (N - $E[d_{ss}]$) segmenata, a D_{delack} je utjecaj prve odgođene potvrde, ako je inicijalna veličina prozora jednaka jedan. $E[D_{ss}]$ i $E[D_{loss}]$ su funkcije *RTT, b,* W_m*, p, N,* $E[t_{TO}]$ i inicijalne veličine prozora $cwnd_0$. D_{ca} iznosi:

   $$E[D_{ca}] = (N - E[d_{ss}])/B \quad (10)$$

   gdje je *B* propusnost. $E[d_{ss}]$ je očekivani broj poslanih segmenata prije pojave gubitaka u inicijalnom sporom startu.

Kako se DSS metoda primjenjuje za ustaljene TCP konekcije, pri čemu ostale Cardwellovim modelom opisane karakteristike TCP konekcije nemaju značajan utjecaj na izračun propusnosti, ovaj model bi se u primjeni za DSS metodu sveo na Padhyev model.

Na Slici 5. je grafički prikazan odnos Mathisovog (označenog sa "Mathis"), Padhyevog (označenog sa "Padhye") i četiri verzije Cardwellovog (označenih sa "Cardwell 2 KB", "Cardwell 64 KB", "Cardwell 256 KB" i "Cardwell 1024 KB") matematičkog modela TCP konekcije [90] iz kojeg je vidljivo da se s povećanjem količine prenesenih podataka Cardwellov model približava Padhyevom modelu, a da s povećanjem gubitaka Mathisov model povećava odstupanja u odnosu na ostala dva promatrana modela (Padhye i Cardwell 1024 KB), odnosno da povećava (predimenzionira) propusnosti TCP konekcije.



Slika 5. *Usporedba Mathisovog, Padhyevog i Cardwellovog modela [90]*

4.2.2. Padhyev analitički model propusnosti

Padhyev model je analitički opis ustaljene i zasićene TCP konekcije kao funkcije prosječnog RTT-a i gubitaka, a koji u obzir uzima utjecaj ograničenja veličine prozora kao i TCP mehanizam brze retransmisije i TCP mehanizam isteka vremena potvrde. Pri tome istek vremena potvrde predstavlja većinu događaja gubitaka u odnosu na brzu retransmisiju u prosječnoj TCP konekciji.

Kontrola zagušenja u Padhyevom modelu temelji se na *TCP Reno* algoritmu. U fazi sporog starta prijenos započinje eksponencijalnim rastom veličine prozora zagušenja W koji se

povećava za svaku uspješnu primljenu potvrdu sve dok ne dođe do gubitka paketa ili dostizanja najveće dopuštene veličine prozora. U fazi izbjegavanja zagušenja veličina prozora se povećava linearno za svaku uspješno primljenu potvrdu. *TCP Reno* implementacija podrazumijeva da je razlog gubitaka paketa u komunikaciji zagušenje, a gubitak paketa se detektira istekom vremena potvrde ili pojavom najmanje tri duple potvrde. U slučaju isteka vremena potvrde udvostručuje se vrijeme isteka potvrde najviše do dostizanja maksimalno definiranog broja sekundi ili maksimalnog broja dopuštenih retransmisija. Kako je vrijeme isteka potvrde relativno dugo postupkom brze retransmisije se odmah nakon primitka tri duple potvrde ponovno šalju paketi, bez čekanja na istek vremena potvrde, te se, u općem slučaju, ponovo ulazi u fazu sporog starta uz smanjivanje veličine prozora. Postupkom brzog oporavka, karakterističnim za *TCP Reno*, sprječava se pražnjenje komunikacijskog kanala nakon brze retransmisije tako što se pretpostavlja da svaki paket duple potvrde predstavlja paket koji je napustio komunikacijski kanal, smanjuje se na pola veličina prozora i prelazi se direktno u fazu izbjegavanja zagušenja, preskačući pri tome fazu sporog starta.

Padhyev model se fokusira na mehanizam kontrole zagušenja *TCP Reno* gdje se veličina prozora W povećava za $1/W$ nakon svakog primljenog ACK paketa ($W' = W + 1/W$). Jednako tako se veličina prozora smanjuje u slučaju pojave gubitaka paketa p, pri čemu veličina smanjenja ovisi o tome radi li se od TD ili TO gubitcima. Komunikacija se promatra u rundama pri čemu komunikacija započinje sa veličinom prozora zagušenja W. Do primitka prvog ACK paketa, za bilo koji paket iz prozora od W paketa, ne šalju se paketi sljedeće runde, odnosno primitak ACK paketa označava kraj trenutne i početak sljedeće runde pri čemu je trajanje runde jednako RTT-u. Podrazumijeva se da je vrijeme potrebno za slanje svih paketa u prozoru W manje od RTT-a. Ako je u trenutnoj rundi poslano W paketa i ako su svi ispravno potvrđeni onda je ukupno primljeno W/b potvrda, pri čemu je b broj paketa koje potvrđuje jedan ACK i ima tipičnu vrijednost dva (jer puno implementacija TCP primatelja šalje jedan kumulativni ACK za dva uzastopno primljena paketa). U sljedećoj rundi šalje se W' paketa, a kako svaki ACK povećava veličinu prozora za $1/W$ W' će se linearno povećati za iznos od $1/b$ paketa za svaki RTT [89]:

$$W' = W + 1/b \tag{11}$$

Model promatra mehanizam TCP kontrole zagušenja u tri područja rada:

- kada su indikatori gubitaka paketa isključivo TD ACK paketi pri čemu je

(12)

$$B(p) = \frac{\frac{1-p}{p} + \frac{2+b}{3b} + \sqrt{\frac{8(1-p)}{3bp} + \left(\frac{2+b}{3b}\right)^2}}{RTT\left(\frac{2+b}{6} + \sqrt{\frac{2b(1-p)}{3p} + \left(\frac{2+b}{6}\right)^2} + 1\right)}$$

- kada su indikatori gubitka paketa TD ACK paketi i TO pri čemu je

$$B(p) \approx \frac{1}{RTT\sqrt{\frac{2bp}{3}} + T_0 min\left(1{,}3\sqrt{\frac{2bp}{8}}\right)p(1+32p^2)} \tag{13}$$

- kada je veličina prozora zagušenja ograničena primateljevom oglašenom veličinom prozora

$$B(p) = \frac{\frac{1-p}{p} + W_{max} + Q(W_{max})\frac{1}{1-p}}{RTT(\frac{b}{8}W_{max} + \frac{1-p}{pW_{max}} + 2) + Q(W_{max})T_0\frac{f(p)}{1-p}} \tag{14}$$

Kompletna[16] karakterizacija TCP propusnosti *B(p)*, u modelu prikazana u paketima u jedinici vremena nasuprot uobičajenog prikaza u bitovima u jedinici vremena je:

$$B(p) = \begin{cases} \frac{\frac{1-p}{p}+E[W]+Q(E[W])\frac{1}{1-p}}{RTT(\frac{b}{2}E[W_u]+1)+Q(E[W])T_0\frac{f(p)}{1-p}} & \forall\, E[W_u] < W_{max} \\ \frac{\frac{1-p}{p}+W_{max}+Q(W_{max})\frac{1}{1-p}}{RTT(\frac{b}{8}W_{max}+\frac{1-p}{pW_{max}}+2)+Q(W_{max})T_0\frac{f(p)}{1-p}} & \forall\, E[W_u] \geq W_{max} \end{cases} \tag{15}$$

gdje je:

$$E[W_u] = \frac{2+b}{3b} + \sqrt{\frac{8(1-p)}{3bp} + \left(\frac{2+b}{3b}\right)^2} \tag{16}$$

$$Q_w = \min\left(1, \frac{(1-(1-p)^3)(1+(1-p)^3)(1-(1-p)^{w-3})}{1-(1-p)^w}\right) \tag{17}$$

$$f(p) = 1 + p + 2p^2 + 4p^3 + 8p^4 + 16p^5 + 32p^6 \tag{18}$$

[16] Aproksimativna, pojednostavljena propusnost Padhyevog modela prikazana je u jednadžbi (8)

W – veličina prozora TCP konekcije

W_u – neograničena veličina prozora (eng. *unconstrained windows size*)

$E[W_u]$ – funkcija stanja (eng. *state-space*) neograničene veličine prozora

Q_w – vjerojatnost da je gubitak u prozoru veličine w uzrokovao *TO* (*time-out*)

b – broj paketa potvrđen sa primljenim ACK paketom (uobičajeno 2)

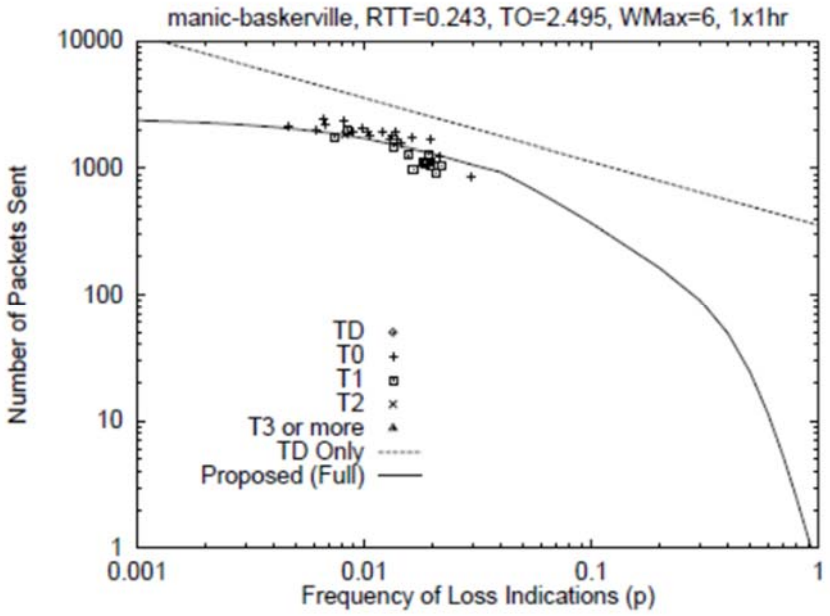
p – vjerojatnost da je paket izgubljen

T_0 – RTO vrijednost, uobičajeno 3 sekunde

TCP definira maksimalnu veličinu prozora $W_{max} = min(cwnd, rwnd)$[17] kao maksimalnu veličinu do koje prozor može rasti u periodu bez indikacije gubitaka. Model uzima u obzir ograničenje veličine prozora tako što za W_u, koji predstavlja neograničenu veličinu prozora, $E[W]$ aproksimira sa $E[W_u]$ ako je $E[W_u]<W_{max}$, a što znači da u tom slučaju *rwnd* ima zanemariv utjecaj na TCP propusnost. Ako je $E[W_u]\geq W_{max}$, $E[W]$ se aproksimira sa W_{max}.

Usporedba Padhyevog matematičkog modela sa rezultatima jednog od provedenih eksperimentalnih mjerenja prikazana je na Slici 6 [89]. Trajanje mjerenja je jedan sat, a podaci o prosječnom RTT-u, prosječnom isteku vremena konekcije i maksimalnoj veličini prozora oglašenoj od strane primatelja, izraženoj u broju paketa, prikazani su u zaglavlju slike. Kod iscrtavanja grafa ukupni vremenski interval je podijeljen u 36 slijednih 100 sekundnih intervala i svaka točka predstavlja broj poslanih paketa u odnosu na frekvenciju indikacija gubitaka paketa u promatranom intervalu. Svaki od 100 sekundnih intervala je klasificiran u jednu od četiri kategorije: sa "TD" označeni su rezultati mjerenja za intervale mjerenja u kojima nije bilo isteka vremena konekcije već samo pojave TD potvrda, sa "T0" rezultati mjerenja koji su imali najmanje jedan jednostruki istek vremena konekcije ali ne i eksponencijalni *backoff*, sa "T1" rezultati mjerenja za intervale kod kojih se pojavio minimalno jedan jednostruki eksponencijalni *backoff* (dvostruki istek vremena konekcije), sa "T2" intervali sa minimalno dva, a sa "T3 or more" intervali sa tri ili više eksponencijalnih *backoff*-a. Sa "TD only" označena je propusnost matematičkog modela koji u obzir uzima samo TD gubitke, a sa "Proposed (Full)" je označena propusnost primjenom potpunog Padhyevog matematičkog modela:

[17] cwnd – *sender's congestion window,* rwnd – *receiver's congestion window*



Slika 6. *Usporedba Padhyevog modela s eksperimentalnim mjerenjima [89]*

4.3. ANALITIČKI IZRAČUN VREMENA ODGOVORA POSLUŽITELJA

Analitički izračun vremena odgovora poslužitelja, izveden iz Padhyevog modela TCP konekcije, omogućuje administratoru sustava povezivanje informacije o kompozitnoj DNS-metrici sa procjenom vremena trajanja odgovora poslužitelja na poslani upit. Osnovni cilj DSS metode je omogućiti klijentu što kraće vrijeme odgovora mrežne usluge pa je stoga vrijeme odgovora mrežne usluge ključan podatak u primjeni DSS metode.

Vrijeme potrebno da se *N* bajtova prenese kroz komunikacijski link propusnosti *B* iznosi

$$t = \frac{N}{B} \tag{19}$$

Kako jednadžba (15) daje propusnost u paketima potrebno je propusnost izraziti u bajtovima. Broj bajtova u paketu je *MSS* pa iz jednadžbi (15) i (19) vrijeme *t* (s) potrebno za prijenos *N* bajtova kroz komunikacijski link propusnosti *B*, odnosno vrijeme odgovora mrežne usluge, iznosi:

$$t = \begin{cases} \dfrac{N(RTT(\frac{b}{2}E[W_u]+1)+Q(E[W])T_0\frac{f(p)}{1-p})}{MSS(\frac{1-p}{p}+E[W]+Q(E[W])\frac{1}{1-p})} & \forall\, E[W_u] < W_{max} \\ \dfrac{N(RTT(\frac{b}{8}W_{max}+\frac{1-p}{pW_{max}}+2)+Q(W_{max}))T_0\frac{f(p)}{1-p})}{MSS(\frac{1-p}{p}+W_{max}+Q(W_{max})\frac{1}{1-p})} & \forall\, E[W_u] \geq W_{max} \end{cases} \quad (20)$$

a za pripadajuću kompozitnu DNS-metriku mrežne usluge opisanu u jednadžbi (1), pri čemu je zajednički parametar za (1) i (20) vrijeme odziva poslužitelja (definirano RTT-om).

Kako DSS metoda u sebi sadrži i informaciju o utjecaju obrade zahtjeva u poslužitelju, u formi parametra LOAD, u izračun vremena odgovora mrežne usluge potrebno je, u slučajevima kada vrijeme tj. kašnjenje procesiranja zahtjeva u poslužitelju nije zanemarivo u odnosu na vrijeme prijenosa podataka kroz komunikacijski link, uračunati i vrijeme procesiranja zahtjeva u poslužitelju D_S (eng. *Server processing delay*) tako da u tim slučajevima ukupno vrijeme odgovora mrežne usluge iznosi:

$$t = \begin{cases} \dfrac{N(RTT(\frac{b}{2}E[W_u]+1)+Q(E[W])T_0\frac{f(p)}{1-p})}{MSS(\frac{1-p}{p}+E[W]+Q(E[W])\frac{1}{1-p})} + D_S & \forall\, E[W_u] < W_{max} \\ \dfrac{N(RTT(\frac{b}{8}W_{max}+\frac{1-p}{pW_{max}}+2)+Q(W_{max}))T_0\frac{f(p)}{1-p})}{MSS(\frac{1-p}{p}+W_{max}+Q(W_{max})\frac{1}{1-p})} + D_S & \forall\, E[W_u] \geq W_{max} \end{cases} \quad (21)$$

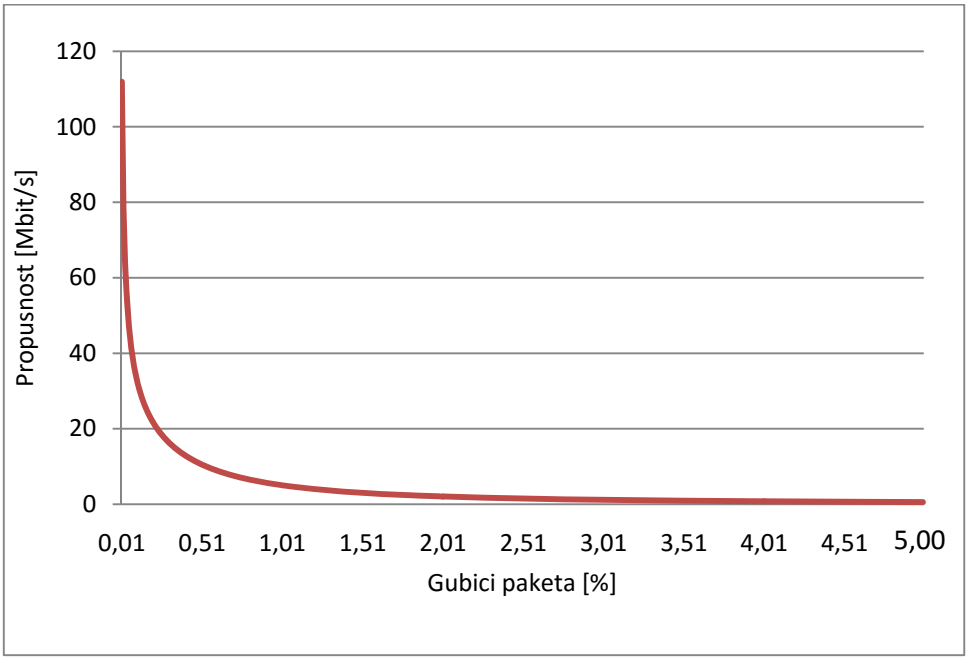
gdje je D_S kašnjenje koje unosi obrada zahtjeva u poslužitelju kao funkcija opterećenja poslužitelja opisanog DSS parametrom LOAD.

Analitičkim izračunom moguće je, za izmjereni RTT (odziv mrežne usluge) te uz poznavanje parametara TCP konekcije definirane modelom (*RTT, W_{max}, MSS, p, RTO, b*), a u pojedinim slučajevima i uz poznavanje kašnjenja obrade zahtjeva u poslužitelju D_S, izračunati vrijeme odgovora mrežne usluge za izračunatu kompozitnu DNS-metriku mrežne usluge. Povezivanje kompozitne DNS-metrike s analitičkim izračunom vremena odgovora poslužitelja omogućuje administratorima predviđanje trajanja vremena odgovora mrežne usluge za izračunatu kompozitnu DNS-metriku jer je cilj DSS metode omogućiti klijentu što kraće vrijeme odgovora mrežne usluge na način da poslužitelju koji će to omogućiti dodijeli najmanji iznos kompozitne DNS-metrike.

Računalni program modela povezivanja kompozitne DNS-metrike mrežne usluge s analitičkim izračunom vremena odgovora poslužitelja, napisan u programskom jeziku C++, zajedno sa primjerom ispisa rezultata izračuna prikazan je u Prilogu 1.

U Grafikonu 6. prikazan je odnos gubitaka i propusnosti Padhyevog analitičkog modela propusnosti izračunat korištenjem računalnog programa za slijedeći primjer:

- Maksimalna veličina segmenta: 1.460 bajtova
- Gubitak paketa: od 0,0001 (0,01%) do 0,05 (5 %)
- Maksimalna veličina prozora: 65.535 bajtova
- RTO: 3 sekunde
- Broj paketa potvrđenih sa jednim ACK paketom: 2
- Broj prenesenih bajtova: 1.000.000



Grafikon 6. *Grafički prikaz odnosa gubitaka i propusnosti Padhyevog analitičkog modela*

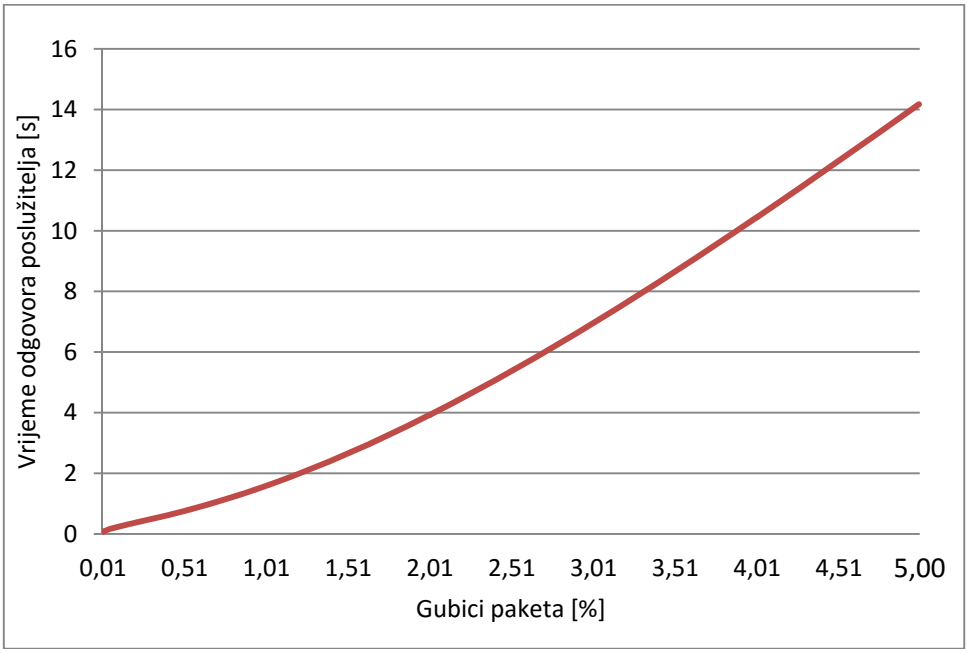
U Tablici 3. prikazan je numerički oblik podatka iz Grafikona 6. za korak promjene gubitaka od 0,5%.

Gubitci paketa [%]	Propusnost [Mbit/s]
0,010000	111,884855
0,500000	10,681598
1,000000	5,082553

Gubitci paketa [%]	Propusnost [Mbit/s]
1,500000	3,034113
2,000000	2,048711
2,500000	1,496678
3,000000	1,154850
3,500000	0,927607
4,000000	0,768342
4,500000	0,652084
5,000000	0,564450

Tablica 3. *Numerički prikaz odnosa gubitaka i propusnosti Padhyevog analitičkog modela*

Iz Grafikona 6. i Tablice 3. vidljivo je da se s povećanjem gubitaka značajno smanjuje propusnost, pogotovo u prvom dijelu krivulje gdje se za povećanje gubitaka od približno 0,5% propusnost smanjuje za približno 11 puta. Daljnjim povećanjem gubitaka smanjuje se njihov utjecaj na smanjenje propusnosti.



Grafikon 7. *Grafički prikaz odnosa gubitaka paketa i vremena odgovora poslužitelja primjenom modela analitičkog izračuna*

U Grafikonu 7. je prikazan odnos gubitaka paketa i vremena odgovora poslužitelja višestruko dostupne mrežne usluge primjenom modela analitičkog izračuna vremena odgovora poslužitelja, a pri čemu su korišteni identični parametri kao i za izračun vrijednosti za Grafikon 6.

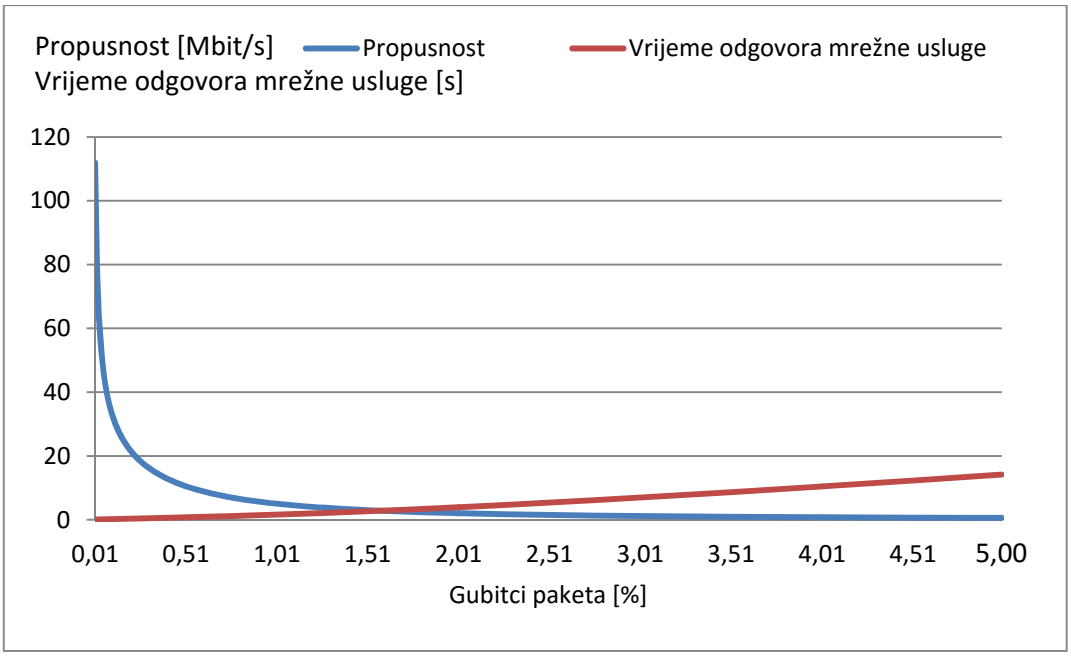
U Tablici 4. je prikazan numerički oblik podataka iz Grafikona 7. za korak promjene gubitaka od 0,5%:

Gubitci paketa [%]	Vrijeme odgovora mrežne usluge poslužitelja[s]	Vrijeme odgovora mrežne usluge [s]/Gubitci paketa [%]
0,010000	0,071502	7,150200
0,500000	0,748952	1,497904
1,000000	1,574012	1,574012
1,500000	2,636685	1,757790
2,000000	3,904894	1,952447
2,500000	5,345173	2,138069
3,000000	6,927309	2,309103
3,500000	8,62434	2,464097
4,000000	10,41203	2,603007
4,500000	12,26835	2,726300
5,000000	14,17309	2,834619

Tablica 4. *Numerički prikaz odnosa gubitaka paketa i vremena odgovora mrežne usluge primjenom modela analitičkog izračuna*

Iz Grafikona 7. i Tablice 4. vidljivo je da se s povećanjem gubitaka povećava vrijeme odgovora mrežne usluge.

U Grafikonu 8. prikazan je usporedni odnos smanjenja propusnosti komunikacijskog kanala i pripadajućeg povećanja vremena odgovora mrežne usluge pri čemu je uzrok smanjenja propusnosti i povećanja vremena odgovora mrežne usluge modela analitičkog izračuna uvjetovan porastom gubitaka prema parametrima korištenim za Grafikon 6.



Grafikon 8. *Grafički prikaz odnosa propusnosti i vremena odgovora mrežne usluge primjenom modela analitičkog izračuna*

U Tablici 5. i Grafikonu 9. prikazani su podaci izračuna kompozitne DNS-metrike i vremena odgovora poslužitelja mrežne usluge analitičkim izračunom pri promjeni odziva poslužitelja od 5 ms do 500 ms. Za Tablicu 5. korak promjene je 50 ms. U računalnom programu za simulaciju korišteni su parametri:

a) Parametri TCP propusnosti (eng. *TCP Bandwidth parameters*):

- Maksimalna veličina segmenta (eng. *Maximum Segment Size*): 1.460 B
- Gubitak paketa (eng. *Packet lost rate*): 0,005000 (0,500000%)
- Maksimalna veličina prozora (*Maximum window size*): 65.535 B
- Inicijalni istek retransmisije (eng. *Initial Retransmission Timeout* (RTO)): 3 s
- Broj paketa potvrđenih sa primljenim ACK (eng. *Number of packets acknowledged by a received ACK*): 2
- Količina prenesenih bajtova (eng. *Number of transmitted bytes*): 1.000.000 B

b) Kašnjenje obrade zahtjeva u poslužitelju (eng. *Server processing delay*): 0 s

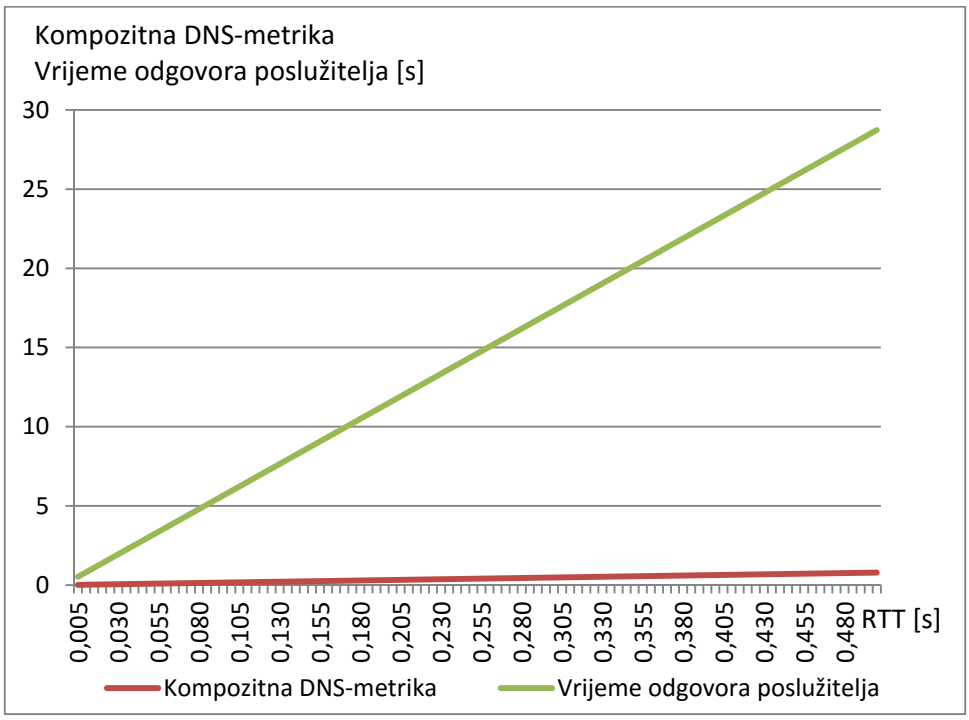
c) DSS parametri (eng. *DSS parameters*):

- DSS faktor opterećenja poslužitelja (eng. *DSS server load factor*) (min 0, max 255): 10
- DSS maksimalni faktor opterećenja poslužitelja (eng. *Max DSS server load factor*) (min 0, max 255): 30

- DSS faktor utjecaja vremena mrežnog odgovora (eng. *DSS network response time impact factor*) (min 0, max 255): 200
- Maksimalno DSS vrijeme mrežnog odgovora (eng. *Max DSS network response time*) / Max RTT: 0,500000 s

RTT [s]	Kompozitna DNS-metrika	Vrijeme odgovora poslužitelja
0,005	0,007843	0,521023
0,050	0,078431	3,085215
0,100	0,156863	5,934317
0,150	0,235294	8,783418
0,200	0,313725	11,632520
0,250	0,392157	14,481621
0,300	0,470588	17,330723
0,350	0,549020	20,179825
0,400	0,627451	23,028926
0,450	0,705882	25,878028
0,500	0,784314	28,727129

Tablica 5. *Numerički prikaz kompozitne DNS-metrike i vremena odgovora poslužitelja analitičkim izračunom u odnosu na RTT*



Grafikon 9. *Grafički prikaz kompozitne DNS-metrike i vremena odgovora poslužitelja analitičkim izračunom u odnosu na RTT*

Iz Tablice 5. i Grafikona 9. vidljiva je povezanost vrijednosti kompozitne DNS-metrike i vremena odgovora poslužitelja analitičkim izračunom u odnosu na RTT. Povećanjem vremena odziva mrežne usluge (RTT) povećava se i kompozitna DNS-metrika i vrijeme odgovora mrežne usluge analitičkim izračunom.

Uspostavom veze između kompozitne DNS-metrike i vremena odgovora poslužitelja analitičkim izračunom omogućeno je administratorima neposrednije sagledavanje utjecaja primjene DSS metode jer je cilj metode smanjiti vrijeme odgovora poslužitelja, a uspostava povezanosti omogućuje da administratori uz DNS-metriku promatraju i analiziraju i pripadajuće procijenjeno vrijeme odgovora poslužitelja.

4.4. INDEKS EFIKASNOSTI I UVJET OPRAVDANOSTI UVOĐENJA DSS METODE

Temeljem analitičkog izračuna vremena odgovora poslužitelja višestruko dostupne mrežne usluge iz jednadžbe (21) moguće je odrediti indeks efikasnosti DSS metode za klijenta koji ju koristi i to kao omjer vremena odgovora mrežne usluge bez korištenja DSS metode i minimalnog vremena odgovora mrežne usluge kao posljedice mogućnosti odabira poslužitelja i(li) komunikacijskog linka za pristup mrežnoj usluzi. Pri tome se minimalno vrijeme odgovora uvećava za vrijeme potrebno za provedbu DSS metode na klijentskoj strani, a vrijeme kada se koristi samo odabir poslužitelja na poslužiteljskoj strani (označeno sa *srv*) predstavlja vrijeme odgovora poslužitelja bez primjene DSS metode:

$$I_E = \frac{t_{srv}}{t_{min} + t_{DSS}} \tag{22}$$

Pri čemu je:

- I_E – indeks efikasnosti DSS metode
- t_{srv} – vrijeme odgovora poslužitelja bez primjene DSS metode, kada se koristi samo odabir poslužitelja na poslužiteljskoj strani
- t_{min} – minimalno vrijeme odgovora poslužitelja uz primjenu DSS metode
- t_{DSS} – vrijeme potrebno za provedbu DSS metode na klijentskoj strani

Kako se za odabir poslužitelja na poslužiteljskoj strani mogu koristiti razne metode, od kojih neke, npr. *Random*, ovise o trenutku u kojem se donosi odluka o odabiru i koje mogu dati

različite odgovore za istog klijenta, moguće je u takvim slučajevima uvesti aproksimaciju t_{srv} parametra u obliku prosječnog vremena odgovora poslužitelja bez primjene DSS metode:

$$\bar{t}_{srv} = \frac{\sum_{i=1}^{N} t_{srv}(i)}{N} \tag{23}$$

Pri čemu je:

$\bar{t}_{srv}$ – prosječno vrijeme odgovora poslužitelja bez primjene DSS metode, kada se koristi samo odabir poslužitelja na poslužiteljskoj strani

$t_{srv}(i)$ – vrijeme odgovora pojedinog poslužitelja bez primjene DSS metode, kada se koristi samo odabir poslužitelja na poslužiteljskoj strani

N – ukupan broj različitih poslužitelja i njihovih komunikacijskih linkova omogućenih klijentu sa poslužiteljske strane

Vrijeme potrebno za provedbu DSS metode na klijentskoj strani se računa temeljem definiranih parametara DSS metode:

- TIME: vrijeme u milisekundama od slanja prvog paketa za testiranje mrežne udaljenosti (početak DSS metode) do pokretanja DSS procedure za izračun kompozitne DNS-metrike
- REFRESH: vrijeme u milisekundama od pokretanja izračuna DNS-metrike do ponovnog pokretanja DSS izračuna za osvježavanje DNS-metrike ako u prethodnom koraku nije pristigao niti jedan odgovor od poslužitelja višestruko dostupne mrežne usluge

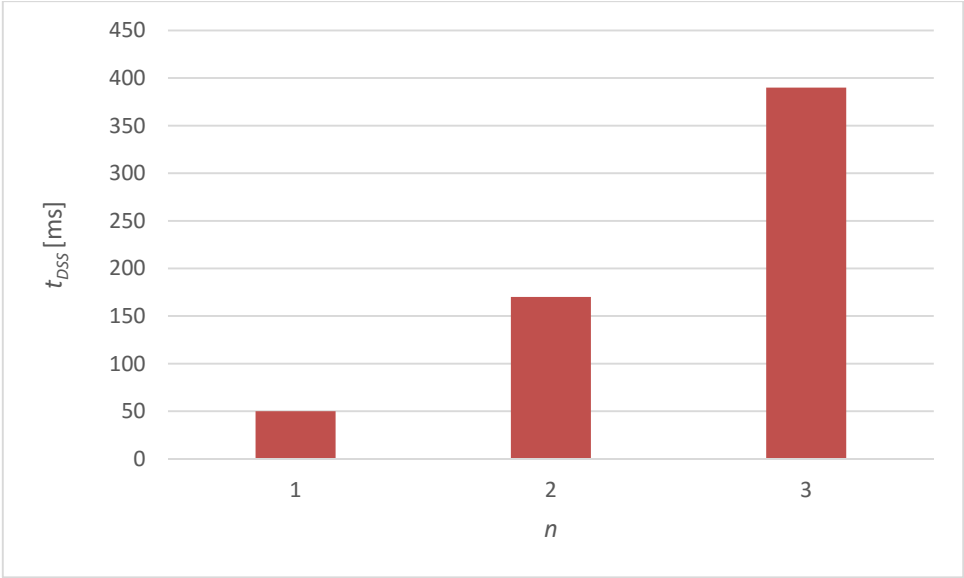
Prema opisu funkcionalnosti DSS metode i Dijagramu 2. vrijeme potrebno za provedbu DSS metode se izračunava kao:

$$t_{DSS} = (2^n - 1)\, TIME + (n - 1)\, REFRESH \tag{24}$$

Pri čemu je:

n – broj ciklusa izračuna kompozitne DNS-metrike

Primjer: za TIME = 50 ms i REFRESH = 20 ms, ako se DSS metoda završi u prvom ciklusu (n = 1) t_{DSS} će imati vrijednost 50 ms, za n = 2 je t_{DSS} = 170 ms, a za n = 3 je t_{DS} = 390 ms.



Grafikon 10. *Grafički prikaz primjera izračuna vremena za provedbu DSS metode*

Da bi DSS metoda bila efikasna prilikom primjene njezin indeks efikasnosti mora biti veći od 1, odnosno:

$$I_E > 1 \tag{25}$$

iz čega proizlazi da minimalno vrijeme odgovora poslužitelja, kako bi primjena metode bila efikasna, mora biti:

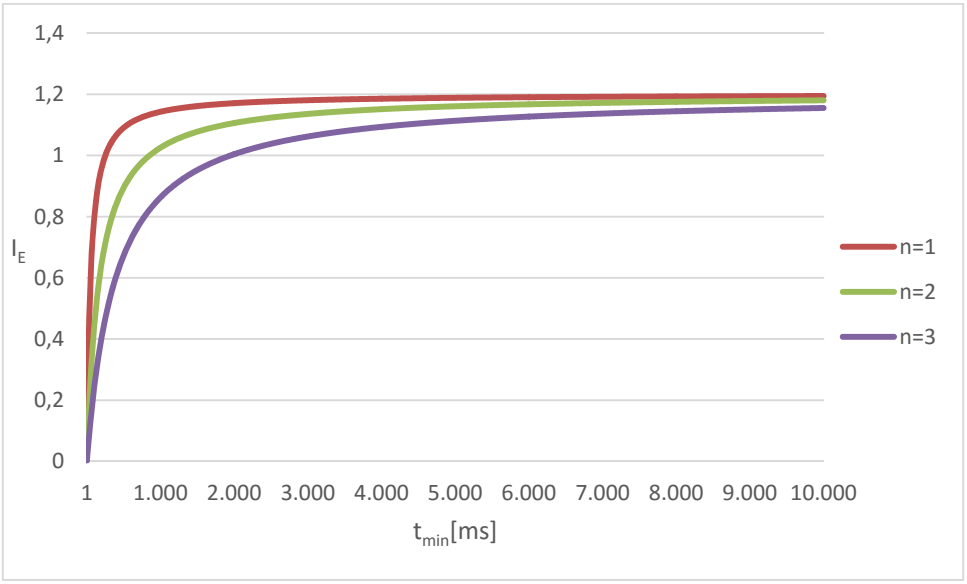
$$t_{min} < t_{srv} - (2^n - 1)\, TIME - (n - 1)\, REFRESH \tag{26}$$

Iz navedenoga je razvidno da je uvjet za opravdanost uvođenja DSS metode da zbroj najkraćeg mogućeg vremena odgovora mrežne usluge t_{min}, kojeg klijent može ostvariti korištenjem DSS metode, i vremena potrebnog za samu provedbu DSS metode t_{DSS} na klijentskoj strani, bude manji od vremena odgovora mrežne usluge kojeg bi klijent dobio ako ne bi koristio DSS metodu.

Bez ispunjenja uvjeta (26) vrijeme utrošeno na primjenu DSS metode bi produžilo vrijeme odgovora mrežne usluge u odnosu na vrijeme koje bi bilo potrebno da klijent dobije odgovor mrežne usluge bez primjene DSS metode.

Primjer: minimalno vrijeme odgovora t_{min} mijenja se od 1 ms do 10.000 ms pri čemu je $t_{srv} = 1{,}2\, t_{min}$, odnosno maksimalno vrijeme koje bi klijent ostvario bez korištenja DSS metode je 20% veće od vremena koje ostvaruje korištenjem DSS metode. Parametri DSS metode su: TIME = 50 ms, REFRESH = 20 ms, a DSS metoda se završi u prvom ciklusu (*n*

= 1). Grafički prikaz promjene indeksa efikasnosti DSS metode I_E za promatrani primjer, uz dodatno prikazan I_E za broj iteracija n = 2 i n = 3, prikazan je u Grafikonu 11:



Grafikon 11. *Grafički prikaz promjene indeksa efikasnosti DSS metode I_E*

Za najmanje promatrano vrijeme odgovora poslužitelja t_{min} = 1 ms indeks efikasnosti iznosi I_E = 0,023529412, dok za najveće promatrano vrijeme odgovora t_{min} = 10.000 ms indeks efikasnosti iznosi I_E = 1,194029851.

Iz (26) moguće je izračunati granično vrijeme opravdanosti uvođenja DSS metode koje za promatrani primjer ($t_{srv} = 1{,}2\ t_{min}$, *TIME* = 50 ms, *REFRESH* = 20 ms, n = 1) iznosi:

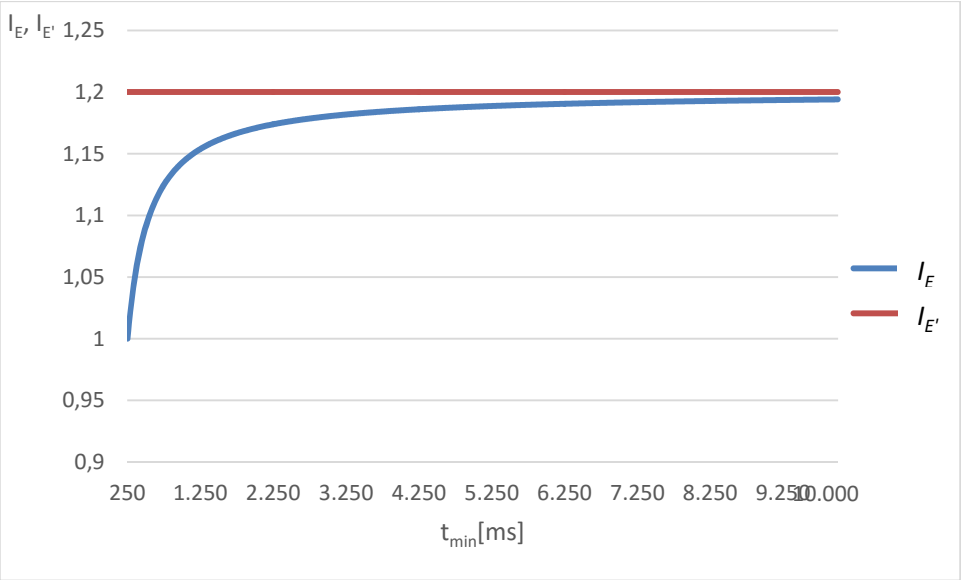
$$t_{min} < 1{,}2\ t_{min} - (2^1 - 1)\ TIME - (1 - 1)\ REFRESH$$

odnosno

$$t_{min} > \frac{TIME}{0{,}2}$$

što znači da za promatrani primjer minimalno vrijeme odgovora mrežne usluge, a za koje je opravdano uvoditi DSS metodu, mora biti veće od 250 ms.

Utjecaj vremena potrebnog za provedbu DSS metode na klijentskoj strani t_{DSS} na indeks efikasnosti DSS metode I_E prikazan je na prethodnom primjeru pri čemu je promatrano vrijeme u kojem je $I_E > 1$, odnosno onaj period za koji je opravdano uvođenje DSS metode.



Grafikon 12. *Grafički prikaz utjecaja t_{DSS} vremena na indeks efikasnosti DSS metode I_E*

Na Grafikonu 12. I_E predstavlja indeks efikasnosti sa izraženim utjecajem t_{DSS} vremena, kako je prikazano u jednadžbi (22). $I_{E'}$ predstavlja indeks efikasnosti koji bi se dobio sa zanemarivanjem utjecaja t_{DSS} vremena na izračun indeksa efikasnosti, odnosno uz t_{DSS} = 0 ms. Vidljivo je da je utjecaj vremena potrebnog za provedbu DSS metode na klijentskoj strani na indeks efikasnosti tim veći što je vrijeme odgovora poslužitelja kraće, a sa povećanjem vremena odgovora poslužitelja njegov utjecaj opada i indeks efikasnosti se približava svojoj maksimalnoj vrijednosti koja, za promatrani primjer, iznosi I_E = 1,2.

4.5. ZAKLJUČAK

Osnovni parametri brzine mrežne konekcije koji utječu na vrijeme kružnog putovanja paketa od izvorišta do odredišta i nazad su propusnost komunikacijskog kanala i ukupna kašnjenja u računalnoj mreži. Kako RTT i veličina TCP prozora omogućuju izračun maksimalne moguće propusnosti prijenosa podataka, analizom uobičajenih matematičkih modela za TCP konekcije velike ili proizvoljne dužine trajanja, a koji se oslanjaju na RTT i veličinu TCP prozora, utvrđeno ja da Padhyev model za ustaljenje TCP konekcije u cijelosti odgovara zahtjevima ovog rada.

Temeljem Padhyevog modela propusnosti TCP konekcije napravljen je analitički izračun vremena odgovora poslužitelja višestruko dostupne mrežne usluge. Predloženim modelom povezivanja kompozitne DNS-metrike s analitičkim izračunom vremena odgovora

poslužitelja moguće je za izmjereni RTT, te uz poznavanje parametara TCP konekcije definirane Padhyevim modelom, analitički izračunati očekivano vrijeme odgovora poslužitelja mrežne usluge za izračunatu kompozitnu DNS-metriku poslužitelja. Pri tome sa povećanjem gubitaka paketa dolazi do značajnog smanjenja propusnosti i produženja vremena odgovora mrežne usluge. S povećanjem kompozitne DNS-metrike poslužitelja povećava se i vrijeme odgovora poslužitelja mrežne usluge.

Indeks efikasnosti DSS metode je omjer maksimalnog i minimalnog vremena odgovora mrežne usluge, kao posljedice mogućnosti odabira poslužitelja i(li) komunikacijskog linka za pristup mrežnoj usluzi, uvećanog za vrijeme potrebno za provedbu DSS metode na klijentskoj strani.

5. IMPLEMENTACIJA I ANALIZA UČINKOVITOSTI DSS METODE U REALNOM OKRUŽENJU

5.1. PRIMJENA I IZVEDBA DSS METODE

Primarne primjene DSS metode su:

1) Određivanje optimalnog poslužitelja (poslužitelja koji će najbrže odgovoriti na korisnički zahtjev) za pristup višestruko dostupnoj mrežnoj usluzi temeljem parametara opterećenja poslužitelja i vremena mrežnog odziva u obliku kompozitne DNS-metrike (PRIORITY, LOAD, IMPACT i RESPONSE DSS parametri)
2) Brzo utvrđivanje nedostupnosti poslužitelja pomoću DSS TIMEOUT parametra
    a. Primarno: utvrđivanje inicijalne nedostupnosti poslužitelja u fazi izračuna kompozitne DNS-metrike kao zamjena za „*TCP timeout*" parametar (za Windows OS 21 sekunda, za Linux ovisno o distribuciji i kernelu 21-189 sekundi) i određivanje sljedećeg najpovoljnijeg dostupnog poslužitelja
    b. Sekundarno: mogućnost upravljanja „*Retransmission Timeout"* (RTO) parametrom za već uspostavljene konekcije, a koji se dinamički prilagođava temeljem povijesnih *Smoothed Round Trip Time* vrijednosti pojedine TCP konekcije. Moguće povezivanje sa *stateful/stateless* informacijom stanja protokola iz FLAGS polja za određivanje mogućnosti prelaska na drugi poslužitelj u određenoj komunikacijskoj fazi (*SYN*, *ESTABLISHED*,…)
3) Optimiziranje raspoređivanja opterećenja između poslužitelja temeljem PRIORITY, LOAD i IMPACT parametara.

Sekundarne primjene DSS metode su:

1. Definiranje vrste usluga koje se nalaze na poslužitelju pomoću DNS-a
2. DSS metoda se može primijeniti i za CDN sadržaje uvažavajući specifičnosti CDN-a (velik broj poslužitelja, velik broj ulazno-izlaznih točaka sustava,…).

Kako je DSS metoda visoko konfigurabilna i prilagodljiva te kako njezina primjena ne isključuje niti jednu drugu metodu, primjena metode može biti prilagođena svakom sustavu i selektivno primijenjena na pojedinu uslugu. Za implementaciju je potrebno da autoritativni DNS poslužitelj višestruko dostupne mrežne usluge i DNS razlučitelj na strani klijenta

podržavaju DSS metodu. Pri izvedbi je potrebno paziti da ukupna veličina DNS zapisa ne prijeđe 512 bajtova. U slučaju da se iz sigurnosnih razloga ne želi oglašavati stvarno opterećenje poslužitelja onda se svim poslužiteljima može staviti identičan iznos opterećenja nula što dovodi do zanemarivanja utjecaja faktora opterećenja poslužitelja pri izračunu kompozitne DNS metrike.

5.1.1. Potvrda učinkovitosti DSS metode

DSS metoda se zasniva na proširenju funkcionalnosti DNS protokola koji je uobičajeno prva faza u ostvarenju mrežne komunikacije između primatelja i davatelja mrežne usluge. Kako se radi o novoj metodi, a koja ima funkcionalno drukčiji princip rada od postojećih metoda, može se mjeriti efekt primjene metode i napraviti usporedba rezultata primjene različitih metoda odabira poslužitelja sa rezultatima primjene DSS metode. Efikasnost metode se može mjeriti za dva temeljna doprinosa metode:

- Određivanje optimalnog poslužitelja za pristup višestruko dostupnoj mrežnoj usluzi, tj. poslužitelja koji će klijentu omogućiti nakraće vrijeme odgovora mrežne usluge temeljem parametara opterećenja poslužitelja i vremena mrežnog odziva (DSS PRIORITY, LOAD, IMPACT i RESPONSE parametri)
- Brzo utvrđivanje nedostupnosti poslužitelja pomoću DSS TIMEOUT parametra, a u odnosu na inicijalni *TCP timeout* parametar

5.1.2. Temeljne pretpostavke metode

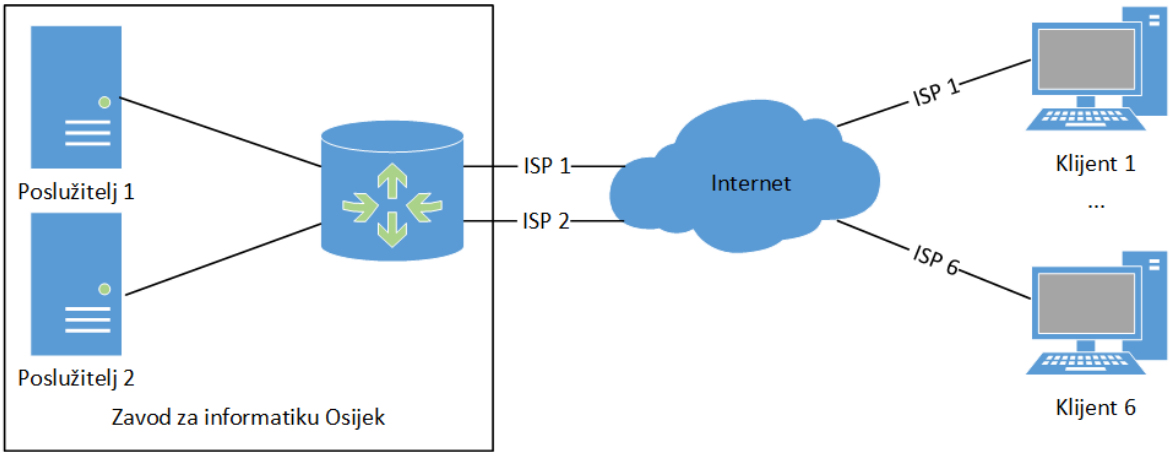
Najznačajnija pretpostavka metode je da su ključni parametri za određivanje optimalnog poslužitelja za pristup višestruko dostupnoj mrežnoj usluzi opterećenje poslužitelja (LOAD) i vrijeme mrežnog odziva (RESPONSE). Kako svaka mrežna usluga može imati različite zahtjeve na navedene parametre potrebno je omogućiti da se kod određivanja optimalnog poslužitelja definira utjecaj svakog od parametara u izračunu kompozitne DNS-metrike, a što je u DSS metodi omogućeno korištenjem IMPACT parametra.

Moguće je da višestruko dostupne mrežne usluge imaju različite zahtjeve za definiranjem parametra opterećenja (LOAD). Kako se u DSS metodi parametar LOAD definira na razini pojedine mrežne usluge, a ne poslužitelja kako cjeline, sistemski administratori mrežne usluge/poslužitelja mogu samostalno definirati što čini parametar opterećenja za pojedinu

uslugu (npr. opterećenje procesora, raspoloživa memorija, opterećenje diskovnog podsustava,…) te na taj način prilagoditi navedeni parametar u ovisnosti o karakteristikama usluge (npr. HTTP usluga koja poslužuje web stranice za matematičke proračune može imati parametar LOAD temeljen na opterećenju procesora, HTTP usluga koja poslužuje korisnicima datoteke kao parametar opterećenja može imati opterećenje diskovnog podsustava, dok će neka treća HTTP usluga imati kompozitni LOAD parametar sastavljen od sva tri ranije navedena parametra opterećenja). Uobičajeno je da se za standardne mrežne usluge (HTTP, HTTPS, FTP,…) parametar LOAD određuje mjerenjem vremena odgovora na razini mrežne usluge dok se za ostale usluge koriste generički parametri na razini operativnog sustava poslužitelja [91].

5.2. IMPLEMENTACIJA I TESTIRANJE

Za potvrdu temeljnih pretpostavki metode koriste se testna mjerenja u realnim uvjetima, korištenjem šest ISP-ova koji čine vlastite autonomne sustave na klijentskoj strani te klasterskih poslužitelja Zavoda za informatiku Osijek na poslužiteljskoj strani.



Slika 7. *Testno okruženje za verifikaciju predloženih rješenja*

Poslužitelj programi.zio.hr, virtualni poslužitelj na klasteru Zavoda za informatiku Osijek, je javno dostupan preko dva neovisna internetska linka (dva ISP-a koji pripadaju različitim autonomnim sustavima) na kojima ima po jednu javnu IP adresu:

- Iskon Internet optički stalni link 8/8 Mbit/s, IP: 213.191.152.142/29
- Optima telekom optički stalni link 10/10 Mbit/s, IP: 85.114.46.152/28

Napomena: odabir dolaznog internetskog linka se obavlja na strani primatelja mrežne usluge odabirom odredišne IP adrese poslužitelja (Iskon Internet ili Optima telekom IP adresa).

Odabir odlaznog internetskog linka obavlja se na strani usmjerivača poslužitelja korištenjem *Policy-Based Routinga*[18] temeljem izvorišne adrese poslužitelja (Iskon Internet ili Optima telekom IP adresa). Rezervacija resursa za poslužitelj programi.zio.hr na internetskim linkovima napravljena je korištenjem QoS mehanizma na centralnom internetskom usmjerivaču.

Poslužitelj programi.zio.hr (OS Windows Server 2008 R2 x64) ima aktivnu HTTP uslugu na TCP portu 80, FTP uslugu na TCP portovima 20 i 21, Telnet uslugu na TCP portu 23 i omogućen ICMP protokol. Za poslužitelj programi.zio.hr su u DNS-u zone zio.hr definirana 2 A zapisa na IP adresama: 213.191.152.142 (Optima telekom) i 85.114.46.152 (Iskon Internet) za puno kvalificirano domensko ime (eng. *fully qualified domain name* - FQDN) programi1.zio.hr, te dva A zapisa za dodatni testni poslužitelj programi2.zio.hr, klonirani virtualni poslužitelj poslužitelja programi1.zio.hr, na IP adresama: 213.191.152.139 i 85.114.46.149.

Planom rada mjerenja definirana su dva osnovna područja provedbe mjerenja:

- analiza parametara mjerenja: cilj ovih mjerenja je utvrditi utjecaj mrežne infrastrukture na RTT, a koji čini osnovicu DSS metode. Mjeri se i istražuje utjecaj veličine ICMP paketa na mjerenje RTT-a, veza između RTT-a i broja skokova između klijenta i poslužitelja, utjecaj međusobne povezanosti mrežnih infrastruktura ISP-ova na broj skokova a time i na RTT, utjecaj propusnosti klijentskog linka na RTT te utjecaj geopozicioniranja na RTT i broj skokova
- mjerenje i analiza utjecaja osnovnih parametara kompozitne DNS-metrike, a to su utjecaj opterećenja poslužitelja i utjecaj vremena mrežnog odgovora poslužitelja, na vrijeme izvršavanja korisničkog zahtjeva odnosno na vrijeme odgovora poslužitelja mrežne usluge.

Napravljena su mjerenja pristupu poslužiteljima programi.zio.hr preko oba internetska linka za hostove koji dolaze iz šest različitih autonomnih sustava (šest ISP-ova: HT, Optima telekom, B-net, Iskon Internet, Metronet telekomunikacije i CARNet).

[18] *Policy-Based Routing* – tehnika usmjeravanja mrežnog prometa temeljena na politici usmjeravanja postavljenoj od strane administratora usmjerivača

Mjerenjima na klijentima (Windows 7 x86/x64) su prikupljeni sljedeći podaci:

- Broj skokova od hosta do poslužitelja (korištenjem alata *traceroute*)
- RTT kroz 3 mjerenja sa po 8 ICMP paketa (alatima *Karat Packet Builder* i *Wireshark*) i veličinom paketa od 16 bajta (prvo mjerenje), 32 bajta (drugo mjerenje) i 64 bajta (treće mjerenje), sa intervalom slanja paketa 10 ms
- Vrijeme prijenosa datoteke veličine 256 kB, 1MB i 4MB te vrijeme odgovora na pozivanje stranice s CPU izračunom korištenjem oba internetska linka, s neopterećenim i s opterećenim poslužiteljem (korištenjem alata *wget*), gubitci TCP paketa (*Wireshark*)
- Ostalih parametara hosta (javna IP adresa, OS, opterećenje procesora, memorije, mrežne kartice,… a sve prikupljeno alatom *HAKOMetar*) od kojih se u rezultatima prikazuje samo propusnost linka hosta.

Za postupak mjerenja korišteni su alati i programi za Microsoft Windows operativni sustav: Windows Ping, Windows Tracert, Windows Consume[19], HAKOMetar[20], Karat Packet Bulder[21], Wireshark[22], Wget[23], Windows Telnet, Putty[24], Windows FTP Client, FileZilla[25], Internet Explorer, Chrome[26], Firefox[27].

Za obradu podataka i izradu tablica i grafikona korišten je tablični kalkulator Microsoft Excel.

5.3. REZULTATI MJERENJA

Temeljem rezultata provedenih mjerenja za pristup klijenta poslužitelju višestruko dostupne mrežne usluge i izračunate kompozitne DNS-metrike može se napraviti analiza učinkovitosti

[19] Consume.exe - Microsoftov komandno-linijski alat za simulaciju opterećenja resursa računala/poslužitelja u svrhu testiranja (fizička memorija, virtualna memorija, diskovni prostor, opterećenje procesora i kernela)
[20] HAKOMetar - Java aplikacija kojom korisnici mogu ispitati kakvoću usluge širokopojasnog pristupa internetu odnosno izmjeriti brzinu prijenosa korisnih podataka do svojega računala, http://www.hakom.hr/default.aspx?id=1144
[21] Karat Packet Bulder - generator IPv4 i drugih paketa, https://sites.google.com/site/catkaratpacketbuilder
[22] Wireshark – analizator mrežnih protokola, http://www.wireshark.org/
[23] Wget - komandno-linijski softverski paket za preuzimanje datoteka HTTP, HTTPS i FTP protokolima, https://www.gnu.org/software/wget/
[24] Putty - SSH i telnet klijent, http://www.putty.org/
[25] FileZilla - FTP, FTP preko SSL/TLS (FTPS) i SSH FTP (SFTP) klijent, https://filezilla-project.org/
[26] Chrome - web preglednik tvrtke Google, http://www.google.com/intl/hr/chrome/browser/
[27] Firefox - web preglednik čiji razvoj koordiniraju Mozilla Foundation i Mozilla Corporation, http://www.mozilla.org/hr/firefox/new/

DSS metode u realnom okruženju i njezina usporedba sa rezultatima ostalih uobičajenih metoda.

5.3.1. Analiza parametara mjerenja

Mjerenja RTT-a u ovom radu rađena su korištenjem ICMP protokola kao dominantnog alata za aktivno mjerenje performansi putanje između dva hosta [92], pri čemu se koriste mali paketi kako bi se izbjegao efekt zagušenja na mreži. Mogućnost da se povećavanjem i variranjem veličine paketa u mjerenjima može aproksimirati brzina mrežne konekcije [93] nije korištena u ovom radu nego je u tu svrhu korišten *HAKOMetar.* Također nije korištena ni mogućnost pasivnog mjerenja TCP RTT-a [94], korištenjem *TCP timestamp* opcijskog polja u TCP zaglavlju, u slučajevima kada višestruko dostupna mrežna usluga koristi TCP protokol.

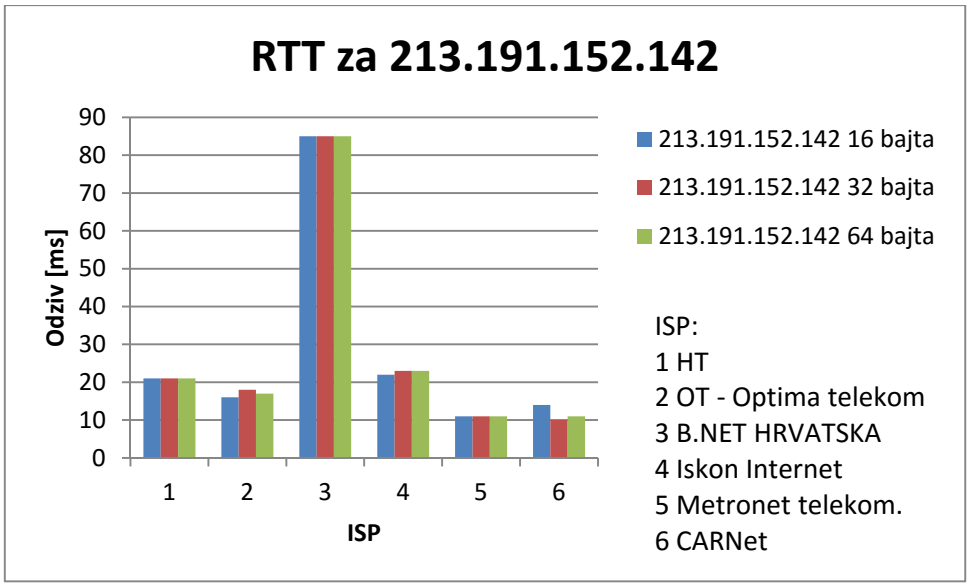
Cilj analize parametara mjerenja je utvrditi utjecaj veličine ICMP paketa na mjerenje RTT-a, utvrditi povezanost RTT-a i broja skokova, utvrditi utjecaj međusobne povezanosti mrežnih infrastruktura ISP-a na broj skokova, a time i na RTT, utvrditi utjecaj propusnosti linka klijenta na RTT te odrediti odnos geopozicioniranja s RTT-om i brojem skokova, a kako bi se mogla provesti mjerenja RESPONSE parametra DSS metode koji se temelji na RTT-u.

5.3.1.1. Utjecaj veličine ICMP paketa na mjerenje RTT-a

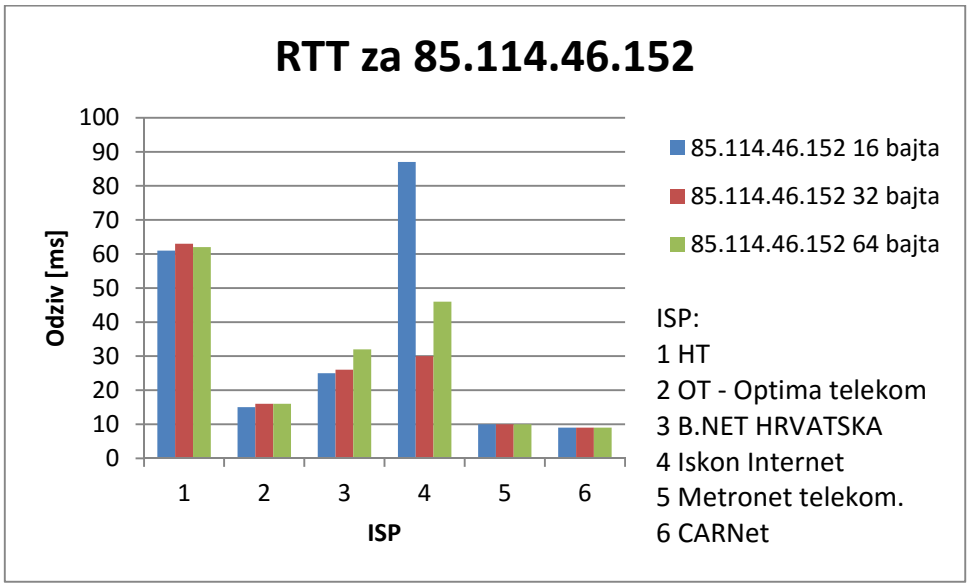
Napravljena su mjerenja RTT-a na oba internetska linka za ICMP pakete veličine 16, 32 (predefinirana veličina za Windows OS) i 64 bajta sa neopterećenim poslužiteljem. Klijentska računala imaju Windows 7 x86/x64 operativni sustav.

			1 HT	**2 OT - Optima telekom**	**3 B.NET HRVATSKA**	**4 Iskon Internet**	**5 Metronet telekomunikacije**	**6 CARNet**
RTT 213.191.152.142 [ms]	Mjerenje 1 (16 bajta)	Out/In	8	8	8	8	8	8
		Min	20	15	76	21	11	10
		Max	26	23	99	24	15	31
		Avr	**21**	**16**	**85**	**22**	**11**	**14**
	Mjerenje 2 (32 bajta)	Out/In	8	8	8	8	8	8
		Min	21	15	77	22	11	10
		Max	22	31	106	25	15	14
		Avr	**21**	**18**	**85**	**23**	**11**	**10**
	Mjerenje 3 (64 bajta)	Out/In	8	8	8	8	8	8
		Min	21	15	78	23	11	10
		Max	22	24	115	24	15	10
		Avr	**21**	**16**	**86**	**23**	**11**	**10**
	Mjerenje 1, 2 i 3 – prosjek	Out/In	8	8	8	8	8	8
		Min	20	15	76	21	11	10
		Max	26	31	115	25	15	31
		Avr	**21**	**17**	**85**	**23**	**11**	**11**
		St. dev.	**0,00**	**0,94**	**0,47**	**0,47**	**0,00**	**1,89**
RTT 85.114.46.152 [ms]	Mjerenje 1 (16 bajta)	Out/In	8	8	8	8	8	8
		Min	60	15	18	28	10	9
		Max	62	22	42	381	12	9
		Avr	**61**	**15**	**25**	**87**	**10**	**9**
	Mjerenje 2 (32 bajta)	Out/In	8	8	8	8	8	8
		Min	62	15	19	30	10	9
		Max	67	24	36	31	10	9
		Avr	**63**	**16**	**26**	**30**	**10**	**9**
	Mjerenje 3 (64 bajta)	Out/In	8	8	8	8	8	8
		Min	62	15	20	30	10	9
		Max	63	24	48	116	11	10
		Avr	**62**	**16**	**32**	**46**	**10**	**9**
	Mjerenje 1, 2 i 3 – prosjek	Out/In	8	8	8	8	8	8
		Min	60	15	18	28	10	9
		Max	67	24	48	381	12	10
		Avr	**62**	**16**	**28**	**54**	**10**	**9**
		St. dev.	**0,82**	**0,47**	**3,09**	**24,00**	**0,00**	**0,00**

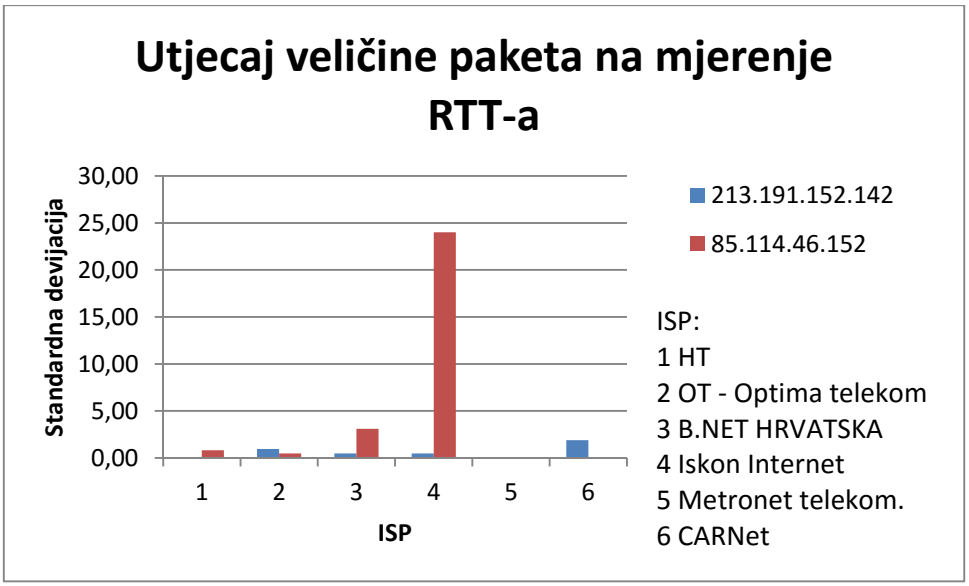
Tablica 6. *Utjecaj veličine paketa na mjerenje RTT-a*



Grafikon 13. *Utjecaj veličine paketa na mjerenje RTT-a za 213.191.152.142*



Grafikon 14. *Utjecaj veličine paketa na mjerenje RTT-a za 85.114.46.152*



Grafikon 15. *Utjecaj veličine paketa na mjerenje RTT-a – standardna devijacija*

Rezultati mjerenja pokazuju da različite veličine ICMP paketa koji su korišteni u mjerenjima nemaju utjecaj na dobivene rezultate, jedino značajno odstupanje je kod mjerenja RTT-a prema IP adresi 85.114.46.152 korištenjem ISP-a Iskon Internet (ISP 4), a zbog velikog pojedinačnog odstupanja u prvom i trećem mjerenju. Kako su klijentska i poslužiteljska strana tijekom mjerenja bili u kontroliranim uvjetima pretpostavka je da su odstupanja uzrokovana trenutnim vršnim opterećenjima ISP-ova linka u trenutku mjerenja (zagušenje), a što je i potvrđeno ponovljenim mjerenjima.

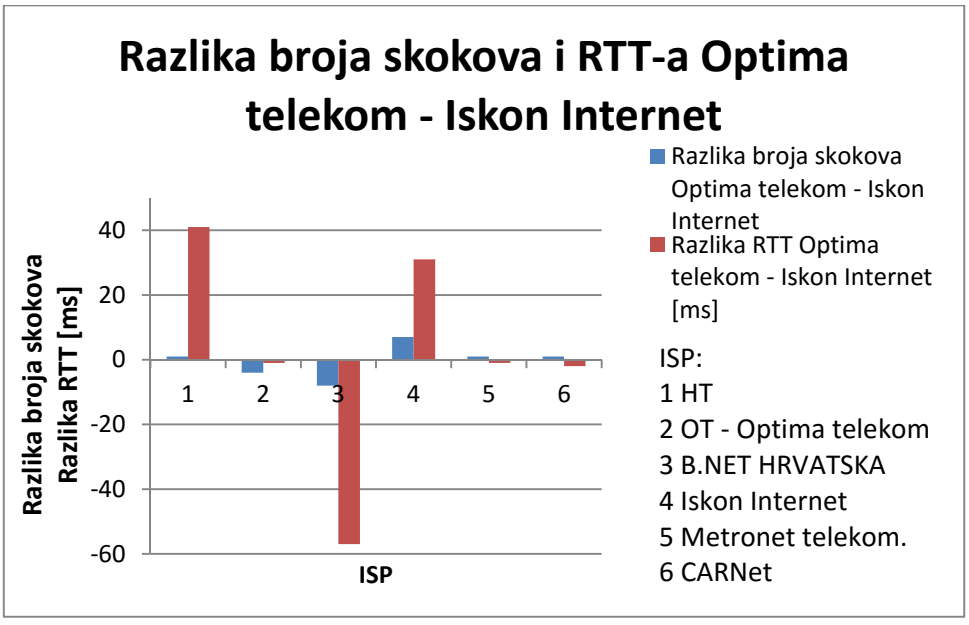
DSS metoda omogućuje da se za mjerenje RESPONSE parametra ne koristi samo ICMP promet, koji na nekim računalnim mrežama može biti filtriran, nego da se definira protokol kojim će se mjeriti vrijeme mrežnog odziva. Kako se u pravilu radi o paketima okvirno veličine paketa kojima se mjerio utjecaj veličine ICMP paketa na mjerenje RTT-a, navedena razmatranja se mogu koristiti kao pokazatelj za korištenje drugih protokola za mjerenje RTT-a (npr. veličina SYN i SYN+ACK paketa je 60 odnosno 56 bajtova (mjereno Wiresharkom), a veličina ICMP ECHO *request/reply* paketa sa predefiniranim 32-bajtnim sadržajem je 60 bajtova pa se stoga ICMP ECHO paketi u mjerenjima RESPONSE parametra mogu zamijeniti SYN/SYN+ACK paketima).

5.3.1.2. Povezanost RTT-a i broja skokova

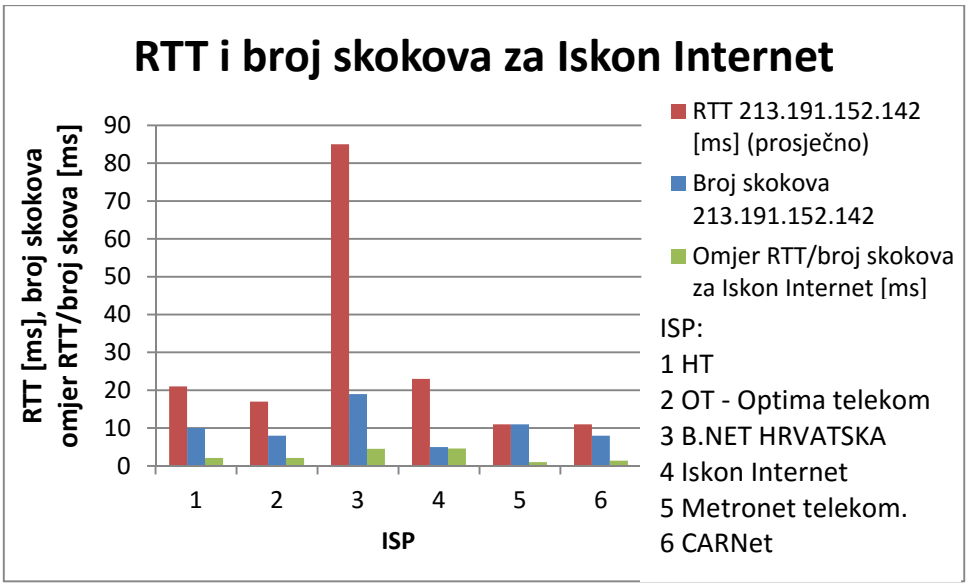
Cilj usporedbe RTT-a i broja skokova je utvrditi da li je moguće RTT, kao dinamički parametar, aproksimirati vrijednošću broja skokova, kao statičkog parametra.

	1 HT	2 OT - Optima telekom d.d.	3 B.NET HRVATSKA d.o.o.	4 Iskon Internet	5 Metronet telekomunikacije d.d.	6 CARNet
Broj skokova 213.191.152.142	10	8	19	5	11	8
Broj skokova 85.114.46.152	11	4	11	12	12	9
RTT 213.191.152.142 [ms] (prosječno)	21	17	85	23	11	11
RTT 85.114.46.152 [ms] (prosječno)	62	16	28	54	10	9
Omjer RTT/broj skokova za Iskon Internet [ms]	2,1	2,1	4,5	4,6	1	1,4
Omjer RTT/broj skokova za Optima telekom [ms]	5,6	4	2,5	4,5	0,8	1
Razlika omjera RTT/broj skokova Optima - Iskon [%]	166,7	90,5	-44,4	-2,2	-20	-28,6
Razlika broja skokova Optima telekom - Iskon Internet	1	-4	-8	7	1	1
Razlika RTT Optima telekom - Iskon Internet [ms]	41	-1	-57	31	-1	-2

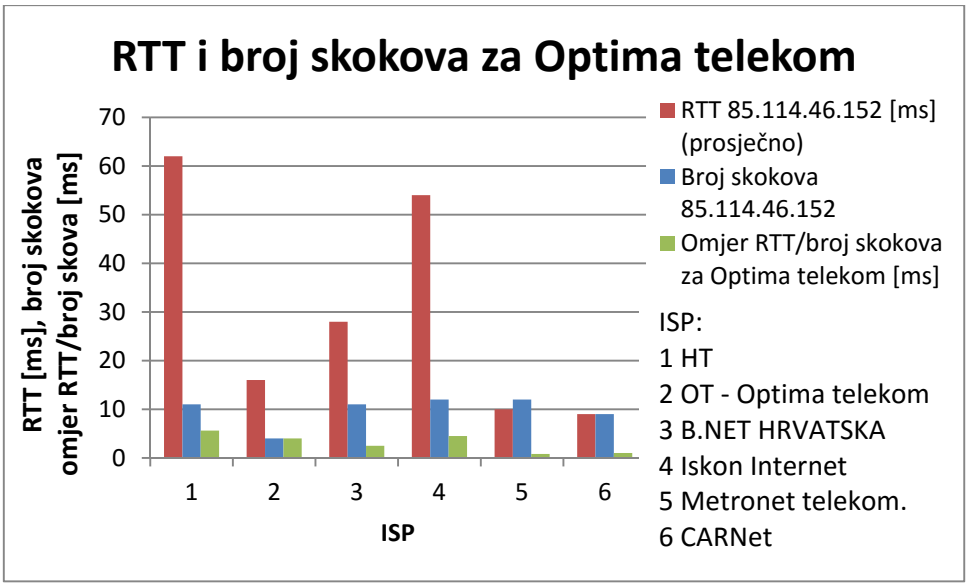
Tablica 7. *Povezanost RTT-a i broja skokova*



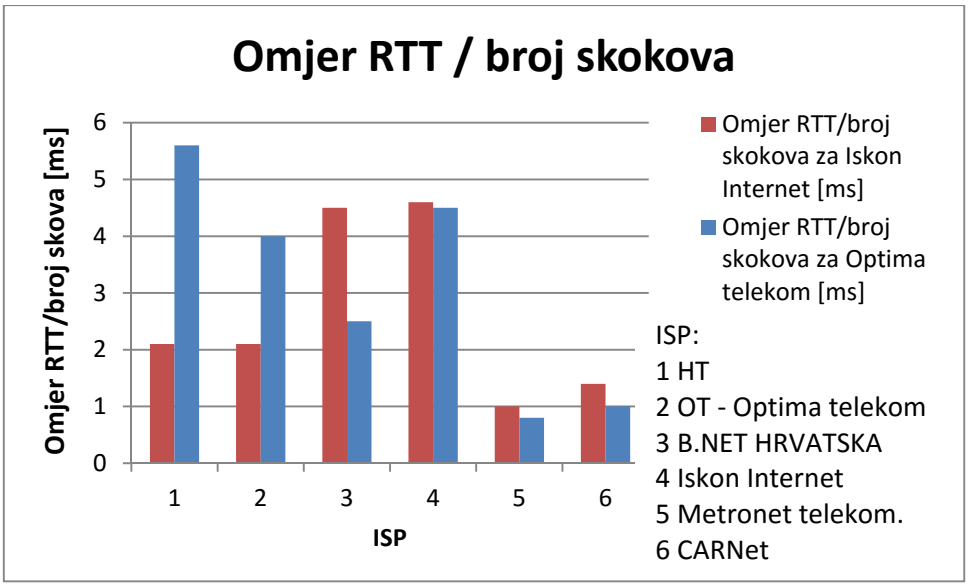
Grafikon 16. *Razlika broja skokova i RTT-a Optima telekom – Iskon Internet*



Grafikon 17. *Odnos RTT-a i broja skokova za Iskon Internet*



Grafikon 18. *Odnos RTT-a i broja skokova za Optima telekom*



Grafikon 19. *Omjer RTT/broj skokova za Iskon Internet i Optima telekom*

Mjerenja pokazuju da broj skokova ne može zamijeniti RTT parametar niti u jednom od dva moguća oblika aproksimacije:

- odnosom broja skokova i RTT-a
- omjerom RTT-a i broja skokova.

Odnos broja skokova i RTT-a kod pristupa istog klijenata poslužitelju programi.zio.hr preko dvaju neovisnih internetskih linkova Iskon Internet i Optima telekom pokazuje da u 2/3 slučajeva veći broj skokova znači i veći RTT ali da se u 1/3 slučajeva za veći broj skokova dobivaju manje vrijednosti RTT-a zbog čega takva metoda aproksimacije RTT parametra nije prihvatljiva.

Omjer RTT-a i broja skokova, koji predstavlja prosječno vrijeme potrebno za jedan skok, pokazuje velike oscilacije u iznosu, od minimalnih 2,2% do maksimalnih 166,7% (prosječno 19,2%), kod pristupa istog klijenta poslužitelju programi.zio.hr preko Iskon Internet i Optima telekom linkova. Zbog toga se broj skokova, u promatranim mjerenjima, ne može uzeti kao parametar za aproksimaciju RTT-a kod kojeg bi se ukupan RTT aproksimirao umnoškom broja skokova i prosječnog vremena potrebnog za jedan skok.

5.3.1.3. Utjecaj međusobne povezanosti mrežnih infrastruktura ISP-ova na broj skokova i RTT

Za broj skokova, a time i RTT, između klijenta i poslužitelja bitan je i način međusobne mrežne povezanosti ISP-ova (njihovih autonomnih sustava) u situacijama kada se klijent i poslužitelj nalaze u različitim ISP-ovima. *Croatian Internet eXchange* (CIX) je hrvatsko nacionalno središte za razmjenu internetskog prometa koje udomljava Sveučilišni računski centar (Srce). Uspostavom izravnih komunikacijskih kanala među hrvatskim ISP-ovima postiže se ušteda na razmjeni podataka među hrvatskim internet korisnicima izravnim međusobnim povezivanjem ISP-ova u cilju smanjenja nepotrebnog prometa kroz treće mreže. U Tablici 8. prikazana je matrica međusobne mrežne povezanosti (eng. *peering*) za 6 ISP-ova na kojima se nalaze klijenti, prema dva ISP-a na kojima se nalaze poslužitelji (Optima telekom i Iskon Internet) [95]. Znak „+“ označava uspostavljen *peering*, znak „-„ da *peering* nije uspostavljen a znak „*“ nepoznato stanje.

	1 HT	2 OT - Optima telekom d.d.	3 B.NET HRVATSKA d.o.o.	4 Iskon Internet	5 Metronet telekomunikacije d.d.	6 CARNet
OT – Optima telekom d.d.	-		+	+	*	+
Iskon Internet	*	+	*		*	+

Tablica 8. *Peering matrica CIX-a za IPv4 protokol za 6 ISP-ova*

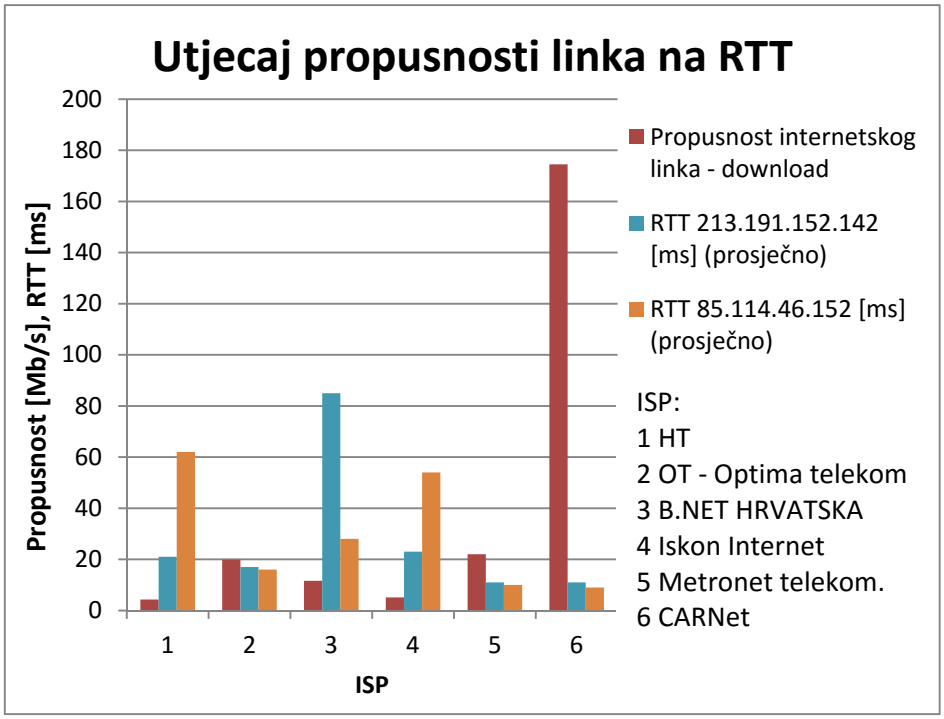
Usporedbom *peering* matrice iz Tablice 8. i rezultata iz Tablice 7. za davatelja usluge B.NET HRVATSKA d.o.o., a koji ima najveću razliku broja skokova, vidljivo je da se broj skokova prema Iskon Internet linku, a koji iznosi 19, zbog upotrebe CIX-a smanjio na samo 11 prema linku Optima telekoma. Detaljan ispis *tracert* naredbe za ISP B.NET HRVATSKA d.o.o. prikazan je u Prilogu 2 gdje je vidljivo da se interkonekcija između ISP-ova BNET HRVATSKA d.o.o i Iskon Internet odvija izvan Republike Hrvatske. Kako CIX trenutno ima 28 članica, a odnosi između članica se bilateralno dogovaraju i promjenjivi su, vidljivo je da je utjecaj koji CIX, kao i globalno bilo koji eng. *Internet Exchange Point*-a (IXP) ima u određivanju mrežne udaljenosti značajan, a promjenjivost u njegovoj implementaciji potvrđuje potrebu dinamičkog klijentskog mjerenja mrežne udaljenosti.

5.3.1.4. Utjecaj propusnosti linka klijenta na RTT

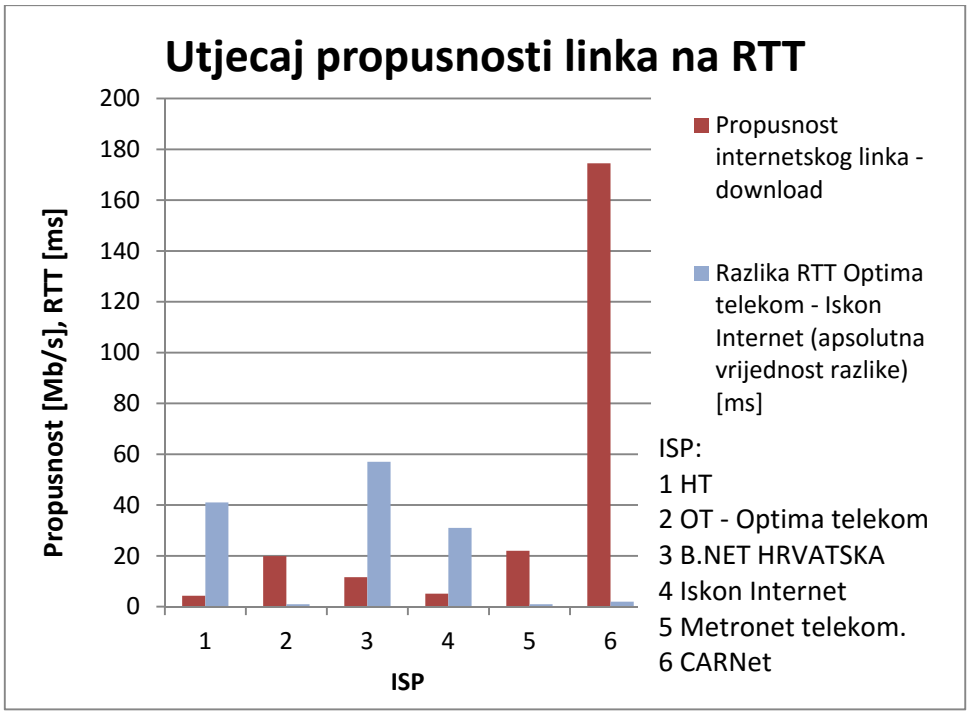
Usporedbom rezultata mjerenja RTT-a i propusnosti linkova klijenata može se prikazati njihova međusobna ovisnost.

	1 HT	2 OT - Optima telekom d.d.	3 B.NET HRVATSKA d.o.o.	4 Iskon Internet	5 Metronet telekomunikacije d.d.	6 CARNet
Propusnost internetskog linka klijenta download [Mb/s]	4,3	19,9	11,6	5,1	22	174,5
Propusnost internetskog linka klijenta upload [Mb/s]	0,5	0,8	0,8	0,4	15,8	16
RTT 213.191.152.142 [ms] (prosječno)	21	17	85	23	11	11
RTT 85.114.46.152 [ms] (prosječno)	62	16	28	54	10	9
Razlika RTT Optima telekom - Iskon Internet (apsolutna vrijednost razlike) [ms]	41	1	57	31	1	2
Tehnologija pristupa	ADSL	ADLS	Kabelski Internet	ADSL	Stalni link	Stalni link
Komunikacijski medij tehnologije	bakrena parica	optički kabel	koaksijalni kabel	bakrena parica	optički kabel	optički kabel

Tablica 9. *Utjecaj propusnosti linka klijenta na RTT*



Grafikon 20. *Utjecaj propusnosti linka RTT za Iskon Internet i Optima telekom*



Grafikon 21. *Utjecaj propusnosti linka RTT – razlika RTT Optima telekom – Iskon internet*

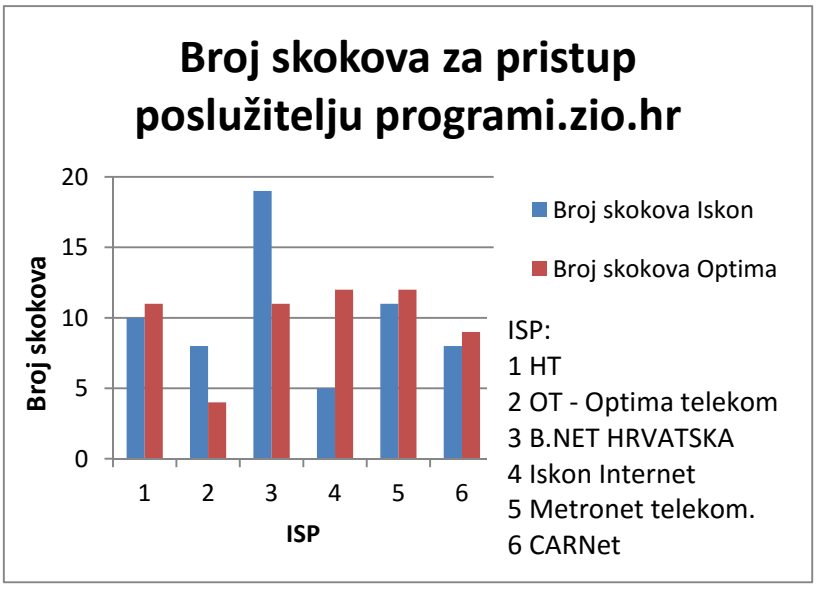
Uočljivo je da klijenti koji dolaze s internetskih linkova većih brzina (Optima telekom, Metronet telekomunikacije i CARNet) imaju značajno manju apsolutnu vrijednost razlike RTT Optima telekom - Iskon Internet od klijenata koji pristupaju poslužitelju programi.zio.hr sa internetskih linkova manjih brzina (HT, B-NET HRVATSKA, Iskon Internet). Važno je napomenuti da je grupa internetskih linkova većih brzina spojena optičkim vezama dok je grupa internetskih linkova manjih brzina spojena bakrenim kablovima (ADSL ili kablovska tehnologija) tako da se potvrđuje utjecaj mrežne tehnologije na kašnjenje u mreži.

5.3.1.5. Geopozicioniranje i odnos s RTT-om i brojem skokova

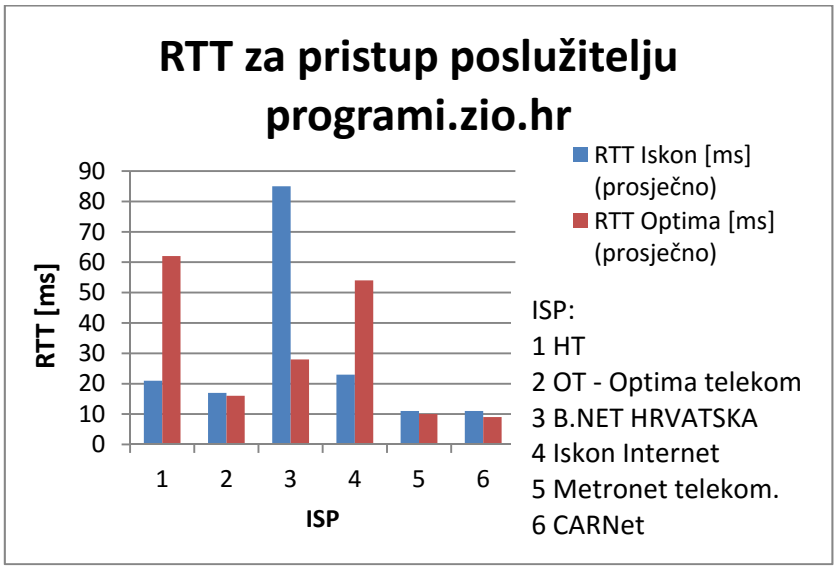
Da bi se prikazao nedostatak gepozicioniranja kod određivanja optimalnog poslužitelja višestruko dostupne mrežne usluge, poslužitelj programi.zio.hr (dostupan preko dva neovisna internetska linka koji pripadaju vlastitim autonomnim sustavima) kao i klijenti spojeni na Internet preko šest neovisnih ISP-ova postavljeni su u isti grad (Osijek) i geografski su smješteni u polumjeru od 3 kilometra od poslužitelja programi.zio.hr. Na taj način isključen je utjecaj geopozicioniranja koji se u pravilu definira na razini naselja.

	1 HT	2 OT - Optima telekom d.d.	3 B.NET HRVATSKA d.o.o.	4 Iskon Internet	5 Metronet telekomunikacije d.d.	6 CARNet
Broj skokova Iskon	10	8	19	5	11	8
Broj skokova Optima	11	4	11	12	12	9
Razlika broja skokova Optima - Iskon	1	-4	-8	7	1	1
Razlika broja skokova Optima - Iskon [%]	10	-50	-42,1	140	9,1	12,5
RTT Iskon [ms] (prosječno)	21	17	85	23	11	11
RTT Optima [ms] (prosječno)	62	16	28	54	10	9
Razlika RTT Optima - Iskon [ms]	41	-1	-57	31	-1	-2
Razlika RTT Optima - Iskon [%]	195,2	-5,9	-67,1	134,8	-9,1	-18,2

Tablica 10. *Geopozicioniranje i odnos sa RTT-om i brojem skokova*



Grafikon 22. *Broj skokova za pristup poslužitelju programi.zio.hr*



Grafikon 23. *RTT za pristup poslužitelju programi.zio.hr*

Iako bi postupkom geopozicioniranja i poslužitelj i klijenti bili smješteni na istu lokaciju vidljive su značajne razlike u broju skokova i RTT-u za pristup poslužitelju preko različitih internetskih linkova. Tako je najveća apsolutna razlika u broju skokova 8, za ISP B.NET HRVATSKA d.o.o.. Najveća razlika u RTT-u je za ISP B.NET HRVATSKA d.o.o. i iznosi -57 ms.

Iz Priloga 2, gdje je prikazan ispis *tracert* naredbe za ISP B.NET HRVATSKA d.o.o., vidljivo je da je tako velika razlika u broju skokova nastala zbog toga što ISP-ovi B.NET HRVATSKA d.o.o. i Iskon Internet nemaju direktnu konekciju između svojih autonomnih sustava.

Iz navedenih razmatranja je potvrđeno da geopozicioniranje nije dovoljno kvalitetan parametar za određivanje optimalnog poslužitelja višestruko dostupne mrežne usluge jer se geografska i mrežna udaljenost mogu značajno razlikovati, pogotovo kod komunikacije između različitih autonomnih sustava.

5.3.2. Utjecaj opterećenja poslužitelja (LOAD) na vrijeme izvršavanja zahtjeva

Kako opterećenje poslužitelja može biti parametar u izračunu kompozitne DNS-metrike korištenjem parametra LOAD, razmatra se utjecaj opterećenja poslužitelja na vrijeme izvršavanja zahtjeva tj. vremena odgovora poslužitelja. Mjeri se vrijeme trajanja izvršavanja zahtjeva prema HTTP usluzi poslužitelja programi.zio.hr, dostupnom na IP adresama 213.191.152.142 i 85.114.46.152, pri neopterećenom i opterećenom poslužitelju, pri čemu se parametar opterećenja poslužitelja LOAD zasniva na opterećenju procesora, ostali sistemski resursi su u svim mjerenjima neopterećeni. Opterećenje procesora je u svim mjerenjima kontrolirano.

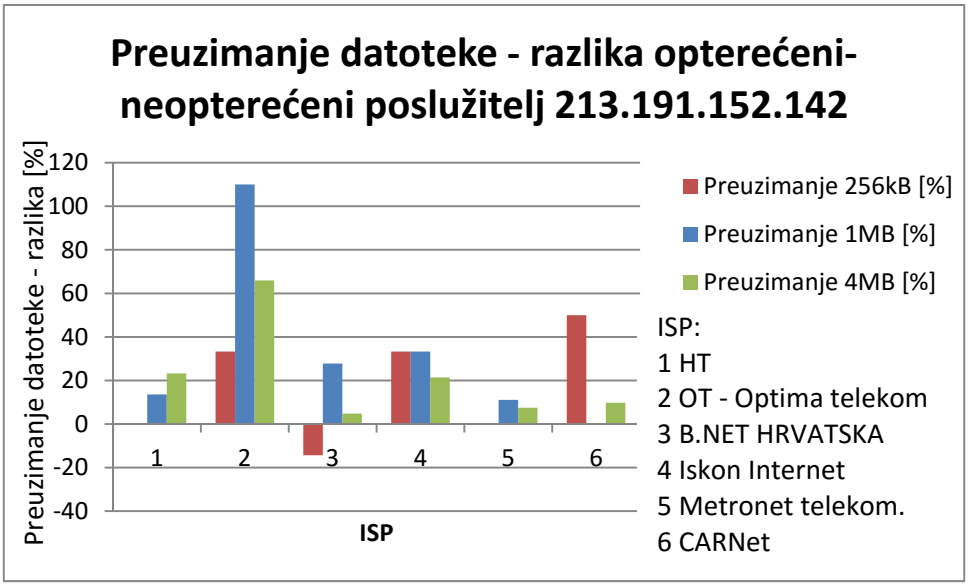
Mjeri se razlika u vremenu izvršavanja zahtjeva prema HTTP usluzi u dva slučaja:

- Preuzimanje datoteke veličine 256 kB, 1 MB i 4 MB HTTP uslugom s poslužitelja programi.zio.hr pri neopterećenom procesoru poslužitelja (prosječno opterećenje procesora 1%) i pri opterećenom procesoru (prosječno opterećenje procesora 100%)

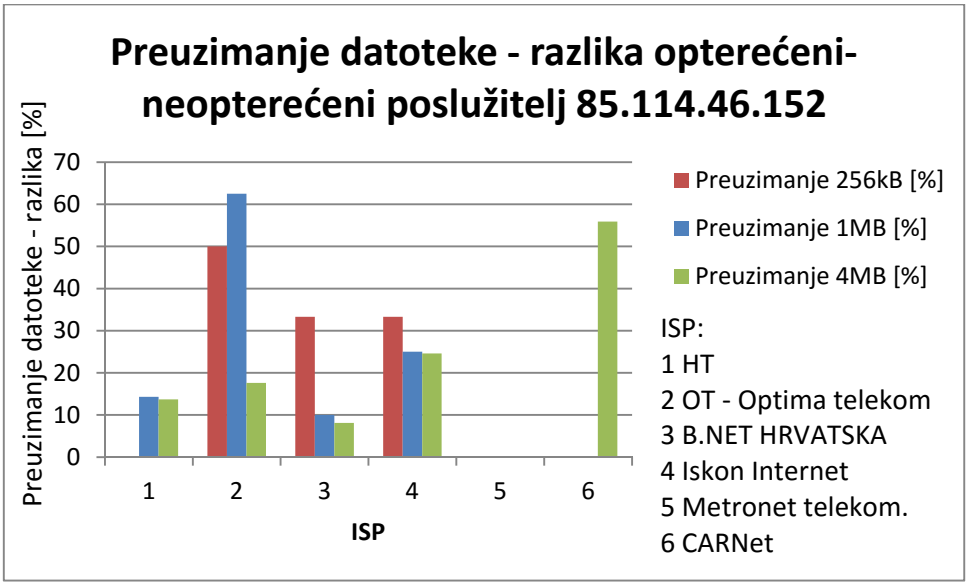
- Pozivanje web stranice sa matematičkim izračunom HTTP uslugom s poslužitelja programi.zio.hr pri neopterećenom procesoru poslužitelja (prosječno opterećenje procesora 1%) i pri opterećenom procesoru (prosječno opterećenje procesora 100%)

	1 HT	2 OT - Optima telekom d.d.	3 B.NET HRVATSKA d.o.o.	4 Iskon Internet	5 Metronet telekomunikacije d.d.	6 CARNet
Razlika opterećeno-neopterećeno 213.191.152.142:						
Preuzimanje 256kB [s]	0	0,1	-0,1	0,1	0	0,1
Preuzimanje 256kB [%]	0	33,3	-14,3	33,3	0	50
Preuzimanje 1MB [s]	0,3	1,1	0,5	0,5	0,1	0
Preuzimanje 1MB [%]	13,6	110	27,8	33,3	11,1	0
Preuzimanje 4MB [s]	1,9	2,9	0,3	1,5	0,3	0,4
Preuzimanje 4MB [%]	23,2	65,9	4,8	21,4	7,5	9,8
Pozivanje stranice CPU izračun [s]	1.049,4	277,9	777,6	500,4	456,1	244,8
Pozivanje stranice CPU izračun [%]	97,5	26	72,9	46,5	42,7	22,8
Razlika opterećeno-neopterećeno 85.114.46.152:						
Preuzimanje 256kB [s]	0	0,1	0,1	0,1	0	0
Preuzimanje 256kB [%]	0	50	33,3	33,3	0	0
Preuzimanje 1MB [s]	0,5	0,5	0,1	0,4	0	0
Preuzimanje 1MB [%]	14,3	62,5	10	25	0	0
Preuzimanje 4MB [s]	1,4	0,6	0,3	1,7	0	1,9
Preuzimanje 4MB [%]	13,7	17,6	8,1	24,6	0	55,9
Pozivanje stranice CPU izračun [s]	328,9	224,2	415,9	429,2	292,5	353,2
Pozivanje stranice CPU izračun [%]	30,9	21	38,9	39,7	27,2	33,4

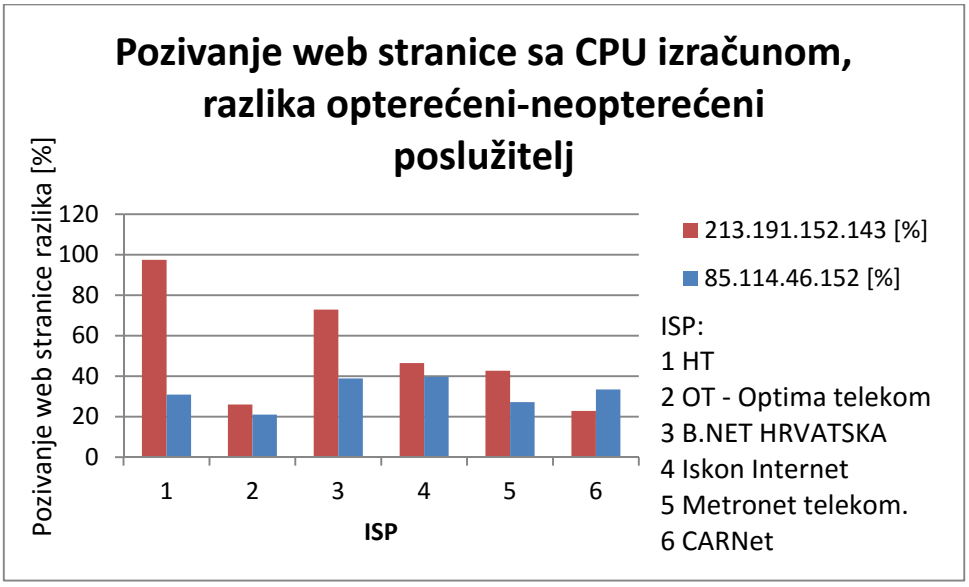
Tablica 11. *Utjecaj opterećenja poslužitelja (LOAD) na vrijeme izvršavanja zahtjeva*



Grafikon 24. *Preuzimanje datoteke - razlika opterećeni-neopterećeni poslužitelj za IP 213.191.152.142*



Grafikon 25. *Preuzimanje datoteke - razlika opterećeni-neopterećeni poslužitelj za IP 85.114.46.152*



Grafikon 26. *Pozivanje web stranice sa CPU izračunom, razlika opterećeni-neopterećeni poslužitelj*

Mjerenja pokazuju značaj utjecaj opterećenja procesora (parametra LOAD) na vrijeme izvršavanja zahtjeva. Od 48 mjerenja u 38 mjerenja opterećenje procesora je uzrokovalo dulje vrijeme izvršavanja zahtjeva, u 9 slučajeva nije bilo razlike a u jednom slučaju je bilo skraćenje vremena izvršavanja što je uvjetovano stanjem sustava u trenutku mjerenja (kako su klijentska i poslužiteljska mrežna infrastruktura u kontroliranom okruženju pretpostavka je da je došlo do trenutnog zagušenja kod ISP-a).

Prosječno povećanje izvršenja zahtjeva za preuzimanje datoteke sa poslužitelja 213.191.152.142 iznosi 23,9% sa rasponom vrijednosti od 0% do 110% (ako se izuzme jedini negativni rezultata od -14,3%), dok prosječno povećanje izvršenja zahtjeva za preuzimanje datoteke sa poslužitelja 85.114.46.142 iznosi 19,4% sa rasponom vrijednosti od 0% do 62,5%.

Izvršavanje zahtjeva pozivanja web stranice s matematičkim izračunom, koji se temelji na CPU izračunu, pokazuje značajnije prosječno povećanje vremena trajanja izvršenja zahtjeva u odnosu na povećanje izvršenja zahtjeva za preuzimanje datoteke. Prosječno povećanje vremena izvršenja zahtjeva pozivanja web stranice sa matematičkim izračunom sa poslužitelja 213.191.152.142 iznosi 52,4% sa rasponom vrijednosti od 22,8% do 97,5%, dok prosječno povećanje vremena izvršenja zahtjeva za pozivanje web stranice sa matematičkim izračunom sa poslužitelja 85.114.46.142 iznosi 31,9% sa rasponom vrijednosti od 21% do 39,7%.

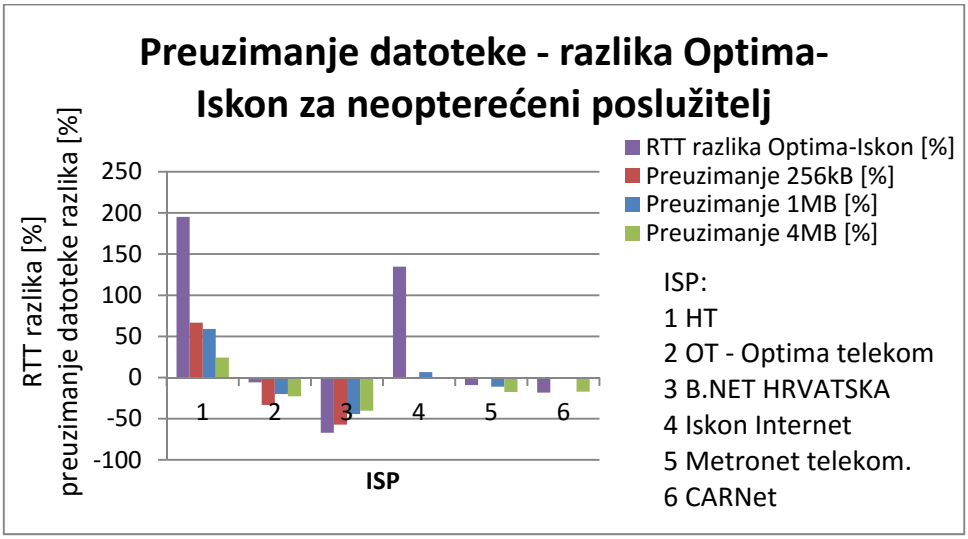
U oba slučaja je potvrđen utjecaj opterećenja poslužitelja (u konkretnom slučaju opterećenja procesora) na vrijeme izvršavanja zahtjeva te opravdanost parametra LOAD u izračunu kompozitne DNS-metrike.

5.3.3. Utjecaj vremena mrežnog odziva (RESPONSE) na vrijeme izvršavanja zahtjeva

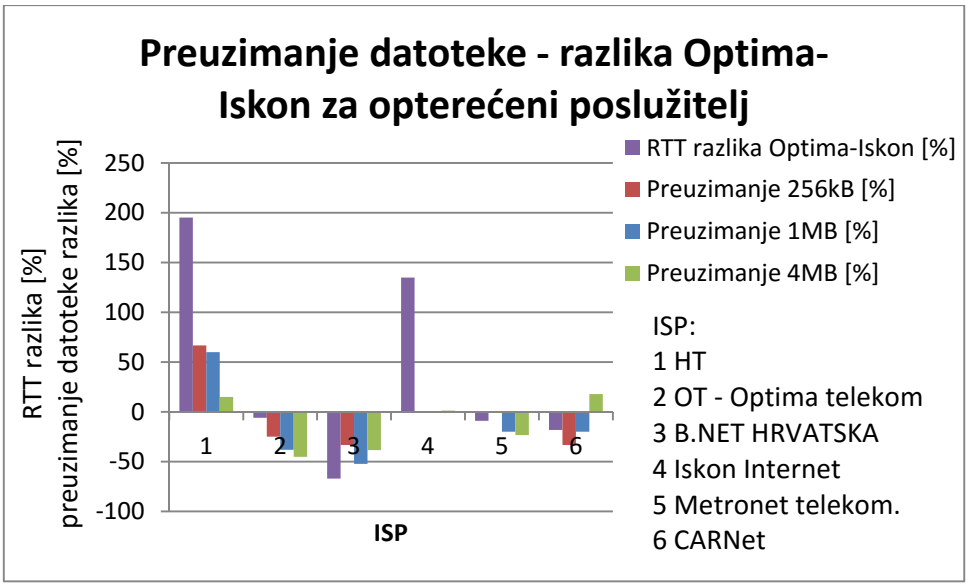
Za određivanje utjecaja vremena mrežnog odziva (RESPONSE) na vrijeme izvršavanja zahtjeva promatra se odnos razlike RTT-a prema poslužitelju programi.zio.hr kada mu se pristupa preko internetskih linkova Optima telekom i Iskon internet, i razlike vremena izvršenja zahtjeva za preuzimanje datoteke veličine 256 kB, 1MB i 4 Mb preko navedenih internetskih linkova za neopterećeni (prosječno opterećenje procesora 1%) i opterećeni (prosječno opterećenje procesora 100%) poslužitelj.

	1 HT	2 OT - Optima telekom d.d.	3 B.NET HRVATSKA d.o.o.	4 Iskon Internet	5 Metronet telekomunikacije d.d.	6 CARNet
RTT Iskon [s] (prosječno)	21	17	85	23	11	11
RTT Optima [s] (prosječno)	62	16	28	54	10	9
RTT razlika Optima-Iskon [s]	41	-1	-57	31	-1	-2
RTT razlika Optima-Iskon [%]	195,2	-5,9	-67,1	134,8	-9,1	-18,2
Neopterećeno razlika Optima-Iskon:						
Preuzimanje 256kB [s]	0,4	-0,1	-0,4	0	0	0
Preuzimanje 256kB [%]	66,7	-33,3	-57,1	0	0	0
Preuzimanje 1MB [s]	1,3	-0,2	-0,8	0,1	-0,1	-0,2
Preuzimanje 1MB [%]	59,1	-20	-44,4	6,7	-11,1	-20
Preuzimanje 4MB [s]	2	-1	-2,5	-0,1	-0,7	-0,7
Preuzimanje 4MB [%]	24,4	-22,7	-40,3	-1,4	-17,5	-17,1
Opterećeno razlika Optima-Iskon:						
Preuzimanje 256kB [s]	0,4	-0,1	-0,2	0	0	-0,1
Preuzimanje 256kB [%]	66,7	-25	-33,3	0	0	-33,3
Preuzimanje 1MB [s]	1,5	-0,8	-1,2	0	-0,2	-0,2
Preuzimanje 1MB [%]	60	-38,1	-52,2	0	-20	-20
Preuzimanje 4MB [s]	1,5	-3,3	-2,5	0,1	-1	0,8
Preuzimanje 4MB [%]	14,9	-45,2	-38,5	1,2	-23,3	17,8

Tablica 12. U*tjecaj vremena mrežnog odziva (RESPONSE) na vrijeme izvršavanja zahtjeva*



Grafikon 27. *Preuzimanje datoteke - razlika Optima-Iskon za neopterećeni poslužitelj*



Grafikon 28. *Preuzimanje datoteke - razlika Optima-Iskon za opterećeni poslužitelj*

Iz rezultata mjerenja je jasno vidljiva povezanost vremena mrežnog odziva na vrijeme izvršavanja zahtjeva poslužitelja u smislu da poslužitelj koji ima manji RTT brže izvrši zahtjev, tj. klijent brže preuzme datoteku sa poslužitelja prema kojemu ima manji RTT. Rezultati pokazuju isti trend utjecaja RTT-a na izvršavanje zahtjeva i kod pristupa neopterećenom i kod pristupa opterećenom poslužitelju, uz jedan izuzetak u mjerenju uzrokovan trenutnom promjenom stanja sustava (kako su klijentska i poslužiteljska mrežna infrastruktura u kontroliranom okruženju pretpostavka je da je došlo do trenutnog zagušenja kod ISP-a). Mjerenjima je potvrđen utjecaj vremena mrežnog odziva na vrijeme odgovora poslužitelja te opravdanost parametra RESPONSE u izračunu kompozitne DNS-metrike.

5.4. ODREĐIVANJE OPTIMALNOG POSLUŽITELJA ZA PRISTUP VIŠESTRUKO DOSTUPNOJ MREŽNOJ USLUZI DSS METODOM

Za određivanje optimalnog poslužitelja za pristup višestrukoj mrežnoj usluzi DSS metodom napravljena su tri scenarija mjerenja u kojima su uspoređeni rezultati odabira poslužitelja za DSS metodu sa četiri uobičajene metode odabira poslužitelja. Dobiveni rezultati su analizirani kako bi se usporedila efikasnost DSS metode u odnosu na ostale metode.

Opći scenarij sa n poslužitelja na m internetskih linkova ($n \in N, m \in N$) moguće je razdvojiti u tri osnovna scenarija:

1. Jedan poslužitelj na dva ili više internetskih linkova
2. Dva ili više poslužitelja na jednom internetskom linku
3. Dva ili više poslužitelja na dva ili više internetskih linkova

Za sva tri navedena scenarija moguće je napraviti mjerenja i usporediti ih sa rezultatima ostalih uobičajenih metoda:

1. Statička, temeljena na geografskoj udaljenosti (*Geographical*)
2. Statička, temeljena na broju skokova (*Hops*)
3. Dinamička, temeljena na slučajnom odabiru poslužitelja (*Random*)
4. Dinamička, temeljena na mjerenju vremena kružnog putovanja (RTT)

Primjer 1 - jedan poslužitelj na dva internetska linka:

Potrebno je napraviti mjerenja za pristup poslužitelju programi.zio.hr preko oba linka za klijente koji dolaze iz 6 različitih autonomnih sustava (šest ISP-ova: HT, Optima telekom, B-net, Iskon Internet, Metronet telekomunikacije i CARNet). Pri tome se poslužitelj i svi klijenti nalaze u istom gradu (Osijek) tako da se isključuje mogući utjecaj geopozicioniranja, a korištenjem jednog poslužitelja isključuje se parametar opterećenja poslužitelja tako da jedini parametar ostaje RESPONSE – najjednostavniji primjer koji DSS metodu transformira u metodu temeljenu na RTT-u. Za poslužitelj programi.zio.hr definirana su dva DNS A zapisa: 213.191.152.142 (Iskon Internet link) i 85.114.46.152 (Optima telekom link).

	1 HT	2 OT - Optima telekom d.d.	3 B.NET HRVATSKA d.o.o.	4 Iskon Internet	5 Metronet telekomunikacije d.d.	6 CARNet
Broj skokova 213.191.152.142	10	8	19	5	11	8
Broj skokova 85.114.46.152	11	4	11	12	12	9
RTT 213.191.152.142 [ms] (prosječno)	21	17	85	23	11	11
RTT 85.114.46.152 [ms] (prosječno)	62	16	28	54	10	9
Preuzimanje datoteke 1MB sa 213.191.152.142 [s] (neopterećen CPU)	2,2	1	1,8	1,5	0,9	1
Preuzimanje datoteke 1MB sa 85.114.46.152 [s] (neopterećen CPU)	3,5	0,8	1	1,6	0,8	0,8
Razlika Optima-Iskon za preuzimanje 1MB [s] (neopterećen CPU)	1,3	-0,2	-0,8	0,1	-0,1	-0,2
Razlika Optima-Iskon za preuzimanje 1MB [%](neopterećen CPU)	59,1	-20	-44,4	6,7	-11,1	-20
Kompozitna DNS-metrika za 213.191.152.142	0,3387097	1	1	0,4259259	1	1
Kompozitna DNS-metrika za 85.114.46.152	1	0,9411765	0,3294118	1	0,9090909	0,8181818
Optimalni poslužitelj (prijenos 1MB podataka)	Iskon	Optima	Optima	Iskon	Optima	Optima
Geographical (kao *Random*)	Iskon	Optima	Iskon	Optima	Iskon	Optima
Hops	Iskon	Optima	Optima	Iskon	Iskon	Iskon
Random	Iskon	Optima	Iskon	Optima	Iskon	Optima
RTT	Iskon	Optima	Optima	Iskon	Optima	Optima
DSS (isto kao RTT)	Iskon	Optima	Optima	Iskon	Optima	Optima

Tablica 13. *Primjer 1 - jedan poslužitelj na dva internetska linka*

Iz Tablice 13. je vidljivo da se DSS metoda pretvorila u RTT metodu jer se, zbog korištenja jednog poslužitelja dostupnog preko dva internetska linka, nije koristio LOAD parametar (napomena: da se i definirao LOAD parametar ne bi imao utjecaj na redoslijed IP adresa poslužitelja temeljem DNS-metrike, samo bi se iznos DNS-metrike povećao za iznos 1), a parametar IMPACT je u oba slučaja postavljen na maksimalnu vrijednost 255 tako da se ne preferira niti jedan od linkova poslužitelja. Također se geografska metoda (temeljena na geopozicioniranju) pretvorila u slučajnu (*Random*) metodu jer se svi klijenti i poslužitelj nalaze na istoj geolokaciji. Slučajna metoda je u DNS poslužiteljima definirana kao kružna (eng. *round-robin*) metoda prema čemu je napravljen ciklički odabir IP adrese poslužitelja (Iskon/Optima) redoslijedom slanja zahtjeva. U tablici su optimalno odabrani poslužitelji prikazani zelenom bojom a neoptimalno odabrani poslužitelji crvenom bojom.

Temeljem vremena potrebnog za prijenos 1 Mb podataka definiran je optimalni poslužitelj (internetski link za pristup poslužitelju) za svakog ISP-a. *Random* i geografska metoda su u 3 od ukupno 6 slučajeva kao rezultat dale optimalni pristup poslužitelju a u 3 slučaja pristup poslužitelju koji nije optimalan tj. s kojeg klijenti imaju duže vrijeme prijenosa podataka te su navedene metode najnepovoljniji slučaj. Metoda broja skokova je od 6 slučajeva u 4 dala optimalni pristup, a u 2 pristup koji nije optimalan. RTT i DSS metoda su u svih 6 slučajeva dale optimalni pristup poslužitelju programi.zio.hr.

Primjer 2 - dva poslužitelja na jednom internetskom linku:

Potrebno je napraviti mjerenja za pristup poslužiteljima programi1.zio.hr i programi2.zio.hr preko istog internetskog linka za klijente koji dolaze iz 6 različitih autonomnih sustava (šest ISP-ova: HT, Optima telekom, B-net, Iskon Internet, Metronet telekomunikacije i CARNet). Pri tome se poslužitelj i svi klijenti nalaze u istom gradu (Osijek) tako da se isključuje mogući utjecaj geopozicioniranja. Poslužitelji programi1.zio.hr i programi2.zio.hr su identični (poslužitelj programi1.zio.hr je poslužitelj programi.zio.hr iz Primjera 1, a poslužitelj programi2.zio.hr je klonirani virtualni poslužitelj programi.zio.hr), jedina razlika je u opterećenju procesora koje za poslužitelj programi1.zio.hr prosječno iznosi 1% a za poslužitelj programi2.zio.hr prosječno iznosi 100%. Zbog toga parametar LOAD za poslužitelj programi1.zio.hr ima minimalnu vrijednost 0 a za programi2.zio.hr ima maksimalnu vrijednost 255. Parametar IMPACT je postavljen na maksimalnu vrijednost 255, iako u ovom konkretnom slučaju nema utjecaj na stvaranje razlike u DSS metrici između poslužitelja, jer zbog korištenja istog internetskog linka imaju identičan RTT, i utječe isključivo na ukupni iznos iskazane metrike.

Poslužitelji programi1.zio.hr i programi2.zio.hr su dostupni preko dva A DNS zapisa hosta programi.zio.hr, od kojih prvi A zapis ima IP adresu poslužitelja programi1.zio.hr (213.191.152.142) a drugi A zapis IP adresu poslužitelja programi2.zio.hr (213.191.152.139), gdje su obje IP adrese na internetskom linku davatelja usluge Iskon Internet.

	1 HT	2 OT - Optima telekom d.d.	3 B.NET HRVATSKA d.o.o.	4 Iskon Internet	5 Metronet telekomunikacije d.d.	6 CARNet
Broj skokova za programi1.zio.hr i programi2.zio.hr	10	8	19	5	11	8
RTT za programi1.zio.hr i programi2.zio.hr [ms] (prosječno)	21	17	85	23	11	11
Preuzimanje datoteke 1MB sa programi1.zio.hr [s] (neopterećen CPU)	2,2	1	1,8	1,5	0,9	1
Preuzimanje datoteke 1MB sa programi2.zio.hr [s] (opterećen CPU)	2,5	2,1	2,3	2	1	1
Razlika preuzimanje 1MB programi2.zio.hr-programi1.zio.hr [s]	0,3	1,1	0,5	0,5	0,1	0
Razlika preuzimanje 1MB programi2.zio.hr-programi1.zio.hr [%]	13,6	110	27,8	33,3	11,1	0
Pozivanje stranice sa CPU izračunom za programi1.zio.hr [ms] (neopterećen CPU)	1.076	1.070,1	1.066,3	1.075	1.067,9	1.071,5
Pozivanje stranice sa CPU izračunom za programi2.zio.hr [ms](opterećen CPU)	2.125,4	1.348	1.843,9	1.575,4	1.524	1.316,3
Razlika pozivanje CPU programi2.zio.hr-programi1.zio.hr [s]	1.049,4	277,9	777,6	500,4	456,1	244,8
Razlika pozivanje CPU programi2.zio.hr-programi1.zio.hr [%]	97,5	26	72,9	46,5	42,7	22,8
Kompozitna DNS-metrika za programi1.zio.hr	1	1	1	1	1	1
Kompozitna DNS-metrika za programi2.zio.hr	2	2	2	2	2	2
Optimalni poslužitelj (prijenos 1MB podataka i pozivanje CPU stranice)	programi1	programi1	programi1	programi1	programi1	programi1
Geographical (kao *Random*)	programi1	programi2	programi1	programi2	programi1	programi2
Hops (kao *Random*)	programi1	programi2	programi1	programi2	programi1	programi2
Random	programi1	programi2	programi1	programi2	programi1	programi2
RTT (kao *Random*)	programi1	programi2	programi1	programi2	programi1	programi2
DSS	programi1	programi1	programi1	programi1	programi1	programi1

Tablica 14. *Primjer 2 - dva poslužitelja na jednom internetskom linku*

Kako su oba poslužitelja smještena na istom internetskom linku klijent koji dolazi iz jednog ISP-a prema njima ima isti RTT (parametar RESPONSE) i isti broj skokova pa se *Hops* i RTT metoda, kao i *Geographical* (zbog smještaja na istu geografsku lokaciju), kod odabira poslužitelja ponašaju kao *Random* metoda, odnosno kružnim mehanizmom odabiru IP adrese poslužitelja programi1.zio.hr i programi2.zio.hr.

Iz Tablice 14. je vidljivo da *Random*, a time i sve ostale metode koje su se pretvorile u *Random* metodu, imaju 50% optimalnih odabira poslužitelja dok DSS metoda ima 100% odabranih optimalnih poslužitelja. Optimalni poslužitelj je definiran manjim vremenom preuzimanja datoteke veličine 1MB, odnosno kraćim vremenom odgovora na pozivanje stranice sa CPU izračunom.

Primjer 3 - dva poslužitelja na dva internetska linka:

U primjeru 3 poslužitelji programi1.zio.hr i programi2.zio.hr iz Primjera 2 su, osim internetskim linkom davatelja usluge Iskon Internet, dostupni i internetskim linkom davatelja usluge Optima telekom:

programi1.zio.hr: IP_1: 213.191.152.142, IP_2: 85.114.46.152

programi2.zio.hr: IP_1: 213.191.152.139, IP_2: 85.114.46.149

Pristup poslužiteljima programi1.zio.hr i programi2.zio.hr, kao jedinstvenoj višestruko dostupnoj mrežnoj usluzi programi.zio.hr koristeći oba poslužitelja i oba internetska linka, omogućen je s 4 A DNS zapisa za host programi.zio.hr:

```
programi.zio.hr     A     213.191.152.142
                    A     85.114.46.152
                    A     213.191.152.139
                    A     85.114.46.149
```

DSS metoda je implementirana na način da je parametar LOAD za poslužitelj programi1.zio.hr postavljen na vrijednost 0 (neopterećeni poslužitelj), dok je za poslužitelj programi2.zio.hr parametar LOAD postavljen na vrijednost 255 (maksimalno opterećeni poslužitelj). Parametar IMPACT je postavljen na maksimalnu vrijednost 255 (maksimalan utjecaj parametra RESPONSE).

	1 HT	2 OT - Optima telekom d.d.	3 B.NET HRVATSKA d.o.o.	4 Iskon Internet	5 Metronet telekomunikacije d.d.	6 CARNet
Broj skokova za 213.191.152.142 i 213.191.152.139 (Iskon internet)	**10**	8	19	**5**	**11**	**8**
Broj skokova za 85.114.46.152 i 85.114.46.149 (Optima telekom)	11	**4**	**11**	12	12	9
RTT za 213.191.152.142 i 213.191.152.139 [ms] (prosječno)	**21**	17	85	**23**	11	11
RTT za 85.114.46.152 i 85.114.46.149 [ms] (prosječno)	62	**16**	**28**	54	**10**	**9**

	1 HT	2 OT - Optima telekom d.d.	3 B.NET HRVATSKA d.o.o.	4 Iskon Internet	5 Metronet telekomunikacije d.d.	6 CARNet
Preuzimanje datoteke 1MB sa 213.191.152.142 (neopterećen CPU) [s]	**2,2**	1	1,8	**1,5**	0,9	1
Preuzimanje datoteke 1MB sa 85.114.46.152 (neopterećen CPU) [s]	3,5	**0,8**	**1**	1,6	**0,8**	**0,8**
Preuzimanje datoteke 1MB sa 213.191.152.139 (opterećen CPU) [s]	2,5	2,1	2,3	2	1	1
Preuzimanje datoteke 1MB sa 85.114.46.149 (opterećen CPU) [s]	4	1,3	1,1	2	**0,8**	**0,8**
Najkraće vrijeme prijenosa 1MB podataka [s]	2,2	0,8	1	1,5	0,8	0,8
Optimalni poslužitelj (najkraći prijenos 1MB podataka)	**213.191.152.142**	**85.114.46.152**	**85.114.46.152**	**213.191.152.142**	**85.114.46.152**	**85.114.46.152**
Kompozitna DNS-metrika za 213.191.152.142	**0,55**	0,47619	0,78261	**0,75**	0,9	1
Kompozitna DNS-metrika za 85.114.46.152	0,875	**0,38095**	**0,43478**	0,8	**0,8**	**0,8**
Kompozitna DNS-metrika za 213.191.152.139	1,625	2	2	2	2	2
Kompozitna DNS-metrika za 85.114.46.149	2	1,61905	1,47826	2	1,8	1,8
Najmanja kompozitna DNS-metrika	0,55	0,38095	0,43478	0,75	0,8	0,8
DSS	213.191.152.142	85.114.46.152	85.114.46.152	213.191.152.142	85.114.46.152	85.114.46.152
Geographical (kao *Random*)	213.191.152.142	85.114.46.152	213.191.152.139	85.114.46.149	213.191.152.142	85.114.46.152
Hops	213.191.152.142	85.114.46.152	85.114.46.149	213.191.152.139	213.191.152.142	213.191.152.139
Random	213.191.152.142	85.114.46.152	213.191.152.139	85.114.46.149	213.191.152.142	85.114.46.152
RTT	213.191.152.142	85.114.46.152	85.114.46.149	213.191.152.139	85.114.46.152	85.114.46.149

Tablica 15. *Primjer 3 - dva poslužitelja na dva internetska linka*

Iz rezultata mjerenja i prikazanih izračuna za svih 5 metoda prikazanih u Tablici 15. vidljivo je da se u slučaju pristupa višestruko dostupnoj mrežnoj usluzi, a koja se sastoji od dva poslužitelja dostupna svaki na po dva internetska linka, metoda broja skokova (*Hops*) pokazala najnepovoljnijom jer je samo u 1/3 slučajeva (u dva od ukupno šest slučajeva) kao rezultat dala optimalnu IP adresu. Slučajna (*Random*) metoda, a time i geografska (*Geographical*) metoda, te RTT metoda kao rezultat su u 50% promatranih slučajeva dale optimalne IP adrese za pristup višestruko dostupnoj mrežnoj usluzi. DSS metoda je u svih 6 slučajeva, dakle sa 100% točnošću, kao rezultat dala optimalnu IP adresu. Kod određivanja

IP adrese za višestruke A zapise u metodama *Hops*, *Random* i RTT pretpostavljen je uobičajeni kružni (eng. *round-robin)* DNS mehanizam sortiranja A zapisa te slijedni tijek DNS upita od ISP-a 1 do ISP-a 6.

Metoda	Ukupno vrijeme preuzimanja datoteke veličine 1MB za 6 ISP-ova [s]:	Povećanje ukupnog vremena preuzimanja datoteke 1MB za 6 ISP-ova u odnosu na DSS [%]:
DSS	7,1	0
Geographical (kao *Random*)	9,0	26,8
Hops	8,0	12,7
Random	9,0	26,8
RTT	7,7	8,5

Tablica 16. *Primjer 3 - ukupno vrijeme preuzimanja datoteke 1MB za 6 ISP-ova*

U Tablici 16. prikazano je ukupno vrijeme preuzimanja datoteke veličine 1MB za svih 6 ISP-ova i za svaku od 5 promatranih metoda. Vidljivo je da je RTT metoda ima najmanje povećanje vremena preuzimanja datoteke veličine 1MB za 6 promatranih ISP-ova u odnosu na DSS metodu, a koje iznosi 8,5%. *Hops* metoda produžava ukupno vrijeme prijenosa datoteke za 12,7%, a *Geographical/Random* metoda za 26,8% u odnosu na DSS metodu.

Izračun indeksa efikasnosti DSS metode za Primjer 3

Iz podataka prikazanih u Tablici 15. moguće je izračunati $\bar{t}_{srv}$ prema jednadžbi (23) za svih šest klijenata. Uz pretpostavku da su parametri DSS metode TIME = 80 ms (najveći prosječni RTT je 62 ms) i REFRESH = 20 ms te ako se DSS metoda završi u prvom ciklusu (n = 1) t_{DSS} će imati vrijednost 80 ms. Parametar t_{min} ima vrijednost najkraćeg vremena odgovora poslužitelja za svakog klijenta. Slijedom navedenoga moguće je izračunati indeks efikasnosti DSS metode prema jednadžbi (22) za svih šest klijenata pojedinačno te prosječni indeks efikasnosti za svih šest klijenata:

	1 HT	2 OT - Optima telekom d.d.	3 B.NET HRVATSKA d.o.o.	4 Iskon Internet	5 Metronet telekomunikacije d.d.	6 CARNet	Prosječno
$\bar{t}_{srv}$ [s]	3,05	1,3	1,55	1,775	0,875	0,9	1,575
t_{min} [s]	2,2	0,8	1	1,5	0,8	0,8	1,2
t_{DSS} [s]	0,08	0,08	0,08	0,08	0,08	0,08	0,08
I_E	**1,34**	**1,48**	**1,44**	**1,12**	**0,99**	**1,02**	**1,23**

Tablica 17. *Indeks efikasnosti DSS metode za Primjer 3*

Iz Tablice 17. je vidljivo da je DSS metoda postigla vrlo visok prosječan indeks efikasnosti od 1,23. To znači da bi prosječno vrijeme preuzimanja datoteke bez primjene DSS metode bilo 23% duže. Najveći indeks efikasnosti je 1,48 za klijenta koji dolazi iz mreže Optima telekoma, dok je za najmanji indeks efikasnosti 0,99 za klijenta iz mreže Metronet telekomunikacije d.d. vidljivo da nije ispunjen uvjet opravdanosti uvođenja DSS metode prema jednadžbi (25) zbog visoko postavljene vrijednosti TIME parametra.

5.5. BRZO UTVRĐIVANJE NEDOSTUPNOSTI POSLUŽITELJA POMOĆU DSS TIMEOUT PARAMETRA

Utvrđivanje nedostupnosti poslužitelja i/ili mrežne usluge DSS metodom provodi se temeljem parametra TIMEOUT koji se definira za svaki PROTOCOL/PORT, a može se postaviti na administratorski zadani iznos (npr. 100 ms).

Na ovaj način se može smanjiti vrijeme utvrđivanja inicijalne nedostupnosti poslužitelja koje kod upotrebe klasičnog pristupa korištenjem inicijalnog *TCP timeout* parametra uobičajeno, ovisno o operativnom sustavu, iznosi 21-189 sekundi[28] (primjeri za Windows OS u Prilogu 3), na vrijednost definiranu od strane administratora poslužitelja u parametru TIMEOUT.

Primjer 1:

Poslužitelj programi.zio.hr je dostupan preko dva internetska linka, na svakom linku ima po jednu javnu IP adresu:

Iskon Internet stalni link 8/8 Mbit/s, IP: 213.191.152.142/29

Optima telekom stalni link 10/10 Mbit/s, IP: 85.114.46.152/28

Poslužitelj ima aktivnu HTTP uslugu na TCP portu 80.
Za poslužitelj programi.zio.hr su u DNS-u zone zio.hr definirana dva A zapisa:
213.191.152.142 i 85.114.46.152

Moguća su dva scenarija:

1) Ispad Iskon Internet linka bez primjene DSS metode:

[28] Windows OS: 3 s + 2*3 s + 4*3 s = 21 s (dvije retransmisije), Linux OS: 3 s + 2*3 s + 4*3 s + 8*3 s + 16*3 s + 32*3 s = 189 s (pet retransmisija) – vrijednosti mogu varirati u odnosu na verziju OS-a

Host koji pristupa poslužitelju programi.zio.hr dobiva od svog DNS klijenta obje IP adrese. Kada pokuša pristupiti IP adresi 213.191.152.142 mora čekati 21 sekundu (Windows OS) za inicijalni *TCP timeout* kod pristupa HTTP usluzi, a što se u praksi može promijeniti na razini mrežne aplikacije.

2) Ispad Iskon Internet linka uz primjenu DSS metode:

Sa primijenjenom DSS metodom i administratorski definiranim TIMEOUT parametrom od 100 ms DSS metoda će IP adresu 213.191.152.142 nakon isteka TIMEOUT parametra proglasiti nedostupnom i DNS klijent će hostu ponuditi samo dostupnu IP adresu 85.114.46.152 čime se značajno skratilo vrijeme u kojem host određuje inicijalnu nedostupnost poslužitelja.

Potreba za ubrzanjem RTO parametra TCP-a vidljiva je i u [96] gdje se inicijalni RTO parametar, koji u [97] iznosi 3 sekunde, smanjuje na 1 sekundu.

Primjer 2:

U Primjeru 2 analizirane su inicijalne RTO vrijednosti za tri vrste TCP prometa. Za Telnet protokol aplikacije Windows Telnet client i Putty imaju RTO ciklus 21 sekundu prije konačnog odustajanja od uspostave komunikacije. Za FTP protokol Windows FTP klijent ima konačni RTO nakon 21 sekunde dok FileZilla nakon prvog RTO ciklusa (nakon 21 sekunde) javlja korisniku nedostupnost usluge, čeka 4 sekunde te ponovno pokušava uspostaviti konekciju i konačno odustaje nakon isteka drugog RTO ciklusa od 21 sekunde, što ukupno iznosi 46 sekundi. Kod svih navedenih slučajeva aplikacije su slale po jedan TCP SYN paket za svaku fazu pokušaja uspostave konekcije, sve u okviru istog soketa. Za HTTP uslugu Internet Explorer i Firefox imaju konačno odustajanje nakon 42 sekunde, odnosno nakon završena dva RTO ciklusa (2 x 21 sekundu), a Chrome nakon 21 sekunde. Pri tome je Internet Explorer slao samo po jedan TCP SYN paket za svaku fazu pokušaja uspostave konekcije dok su Firefox i Chrome odmah slali dva ili više SYN paketa s različitih portova, dakle odmah su kreirani višestruki soketi. Pri tome je vidljivo da i Firefox i Chrome, kako je prikazano u Prilogu 4, imaju interni mehanizam kojim pokušavaju ubrzati uspostavu inicijalne konekcije s poslužiteljem. Chrome odmah, u razmaku od 1 ms otvara dva soketa prema HTTP poslužitelju, a nastavlja komunicirati soketom koji prvi da odgovor. Ako u

roku od 250 ms Chrome i Firefox ne dobiju odgovor otvaraju dodatni soket prema usluzi. Ovime se dodatno potvrđuje potreba za brzim određivanjem nedostupnosti poslužitelja na sistemskoj razini.

U Prilogu 5 analiziran je prelazak na drugi DNS A zapis, u situaciji kada je usluga na prvom A zapisu nedostupna, za različite TCP usluge. Za FQDN testiranje.zio.hr pridružena su dva DNS A zapisa: 172.16.0.60 i 172.16.0.56 na kojima se nalaze konfigurirane višestruko dostupne Telnet, FTP i HTTP usluge. DNS je konfiguriran na način da uvijek daje A zapise istim redoslijedom, pri čemu je prvi A zapis IP adresa 172.16.0.60, a drugi A zapis IP adresa 172.16.0.56. Simulirana je nedostupnost poslužitelja 172.16.0.60 za pristup svih promatranih mrežnih aplikacija (Telnet: Windows Telnet client i Putty; FTP: Windows FTP i FileZilla; HTTP: Internet Explorer, Chrome i Firefox). Sve mrežne aplikacije, osim FileZilla FTP klijenta, nakon inicijalnog RTO ciklusa pokrenule su uspostavu konekcije prema drugoj IP adresi (172.16.0.56) definiranoj u drugom A zapisu.

Rezultati analize potvrđuju potrebu testiranja inicijalne dostupnosti poslužitelja jer neke aplikacije ne koriste mogućnost prelaska na ostale dostupne A zapise, a aplikacije koje koriste tu mogućnost čekaju završetak inicijalnog RTO ciklusa za prelazak na slijedeći A zapis.

Analiza rada promatranih mrežnih aplikacija, kod inicijalnog utvrđivanja nedostupnosti mrežnih usluga i korištenja ostalih dostupnih A zapisa, potvrđuje činjenicu da i TCP, kao protokol namijenjen komunikaciji u vrlo širokom spektru primjene, kao i mrežne aplikacije namijenjene općoj primjeni, slijede pravila koja omogućuju rad u najrazličitijim mrežnim situacijama te stoga imaju parametre rada koji omogućuju zadovoljavajuće funkcioniranje u općim situacijama.

Uvođenjem DSS TIEMOUT parametra moguće je, korištenjem DNS mehanizma, napraviti prilagodbu načina utvrđivanja inicijalne nedostupnosti poslužitelja višestruko dostupne mrežne usluge za specifične primjene definiranjem vrijednosti TIMEOUT parametra za pojedinu višestruko dostupnu mrežnu uslugu. Slična ideja uvođenja korisničke *TCP timeout* opcije definirana je u [98], trenutno u fazi predloženog standarda, koja se odnosi na

međusobni dogovor klijenta i poslužitelja o *TCP timeout* parametrima za pojedinu uspostavljenu sesiju, a koja bi se definirala u fazi uspostave TCP konekcije.

5.6. ZAKLJUČNA ANALIZA REZULTATA

Temeljem provedenih mjerenja utvrđeno je sljedeće:

- Mjerenja pokazuju zanemarivu razliku u rezultatima kada se za mjerenja koriste ICMP paketi veličina 16 bajta, 32 bajta i 64 bajta, odnosno testirane veličine ICMP paketa nemaju utjecaj na RTT
- Mjerenja potvrđuju da se u svih 6 slučajeva broj skokova od hosta do poslužitelja programi.zio.hr razlikuje u ovisnosti o tome da li se za pristup koristi Iskon Internet ili Optima telekom stalni link. Najmanja razlika je jedan, a najveća razlika osam skokova. Najmanji broj skokova za pristup hosta poslužitelju je očekivano za pristup hosta poslužitelju unutar istog autonomnog sustava (Optima-Optima 4 skoka i Iskon-Iskon 5 skokova). Također je potvrđeno da veći broj skokova ne znači uvijek i veći RTT te da se ne može napraviti jasna poveznica između broja skokova i trajanja RTT-a [9]
- Hostovi koji dolaze sa internetskih linkova većih brzina imaju značajno manju apsolutnu vrijednost razlike RTT za različite pristupne internetske linkove od klijenata koji pristupaju poslužitelju sa internetskih linkova manjih brzina, pri čemu je grupa internetskih linkova većih brzina spojena optičkim vezama dok je grupa internetskih linkova manjih brzina spojena bakrenim kablovima (potvrđen utjecaj tehnologije)
- Geopozicioniranje nije dovoljno kvalitetan parametar za određivanje optimalnog poslužitelja višestruko dostupne mrežne usluge jer se geografska i mrežna udaljenost mogu značajno razlikovati, pogotovo kod komunikacije između različitih autonomnih sustava
- Mjerenja pokazuju značaj utjecaj opterećenja procesora (parametra LOAD) na vrijeme izvršavanja zahtjeva, veće opterećenje procesora donosi i duže trajanje izvršavanja zahtjeva
- Vidljiva je povezanost vremena mrežnog odziva (parametar RESPONSE) na vrijeme izvršavanja zahtjeva poslužitelja u smislu da poslužitelj koji ima manji RTT brže izvrši zahtjev, tj. klijent brže preuzme datoteku sa poslužitelja prema kojemu ima

manji RTT. Rezultati pokazuju isti trend utjecaja RTT-a na izvršavanje zahtjeva i kod pristupa neopterećenom i kod pristupa opterećenom poslužitelju

- Potvrđeno je da DSS metoda za određivanje optimalnog poslužitelja za pristup višestruko dostupnoj mrežnoj usluzi temeljem parametara opterećenja poslužitelja i vremena mrežnog odziva daje značajno bolje rezultate u odnosu na četiri ostale promatrane metode: *Geographical, Hops, Random* i RTT
- Potvrđeno je da se DSS metodom može značajno skratiti vrijeme utvrđivanja nedostupnosti poslužitelja mrežne usluge korištenjem TIMEOUT parametra.

5.7. USPOREDBA REZULTATA ANALITIČKOG IZRAČUNA VREMENA ODGOVORA POSLUŽITELJA I MJERENJA

Usporedbom rezultata analitičkog izračuna vremena odgovora poslužitelja i provedenih mjerenja moguće je numerički izraziti odstupanja analitičkih i eksperimentalnih rezultata i potvrditi mogućnost primjene analitičkog modela za predviđanje vremena odgovora poslužitelja mrežne usluge.

Detaljna usporedba rezultata analitičkog izračuna vremena odgovora poslužitelja i mjerenja napravljena je za slučaj pristupa klijenta s CARNet mreže prema poslužitelju na linku Optima telekoma (IP 85.114.46.152) čija je propusnost 10 Mbit/s. Kako klijent ima veću propusnost od poslužitelja ograničenje propusnosti u ovom slučaju je na poslužiteljskoj strani, a koja je kontrolirana u procesu mjerenja. Izmjereno vrijeme preuzimanja datoteke veličine 1 MB sa poslužitelja s neopterećenim CPU-om je 0,8 sekundi što daje nazivnu propusnost od 10 Mbit/s. Analitičkim izračunom, čiji su parametri prikazani u Prilogu 1 (Model povezivanja kompozitne DNS-metrike mrežne usluge s analitičkim izračunom vremena odgovora poslužitelja) za gubitke 0,53% izračunata je propusnost od 10,107951 Mbit/s i vrijeme odgovora mrežne usluge od 0,791456 s. Kompozitna DNS-metrika za promatrani slučaj je 0,141176 uz DSS parametre:

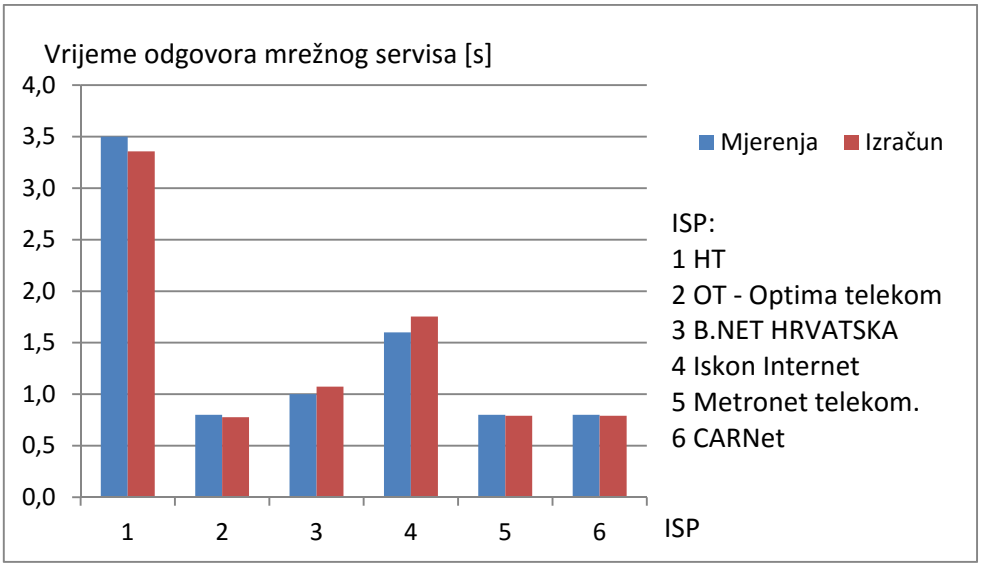
- DSS server load factor (min 0, max 255): 10
- Max DSS server load factor (min 0, max 255): 30
- DSS network response time impact factor (min 0, max 255): 200

- Max DSS network response time / Max RTT: 0.05

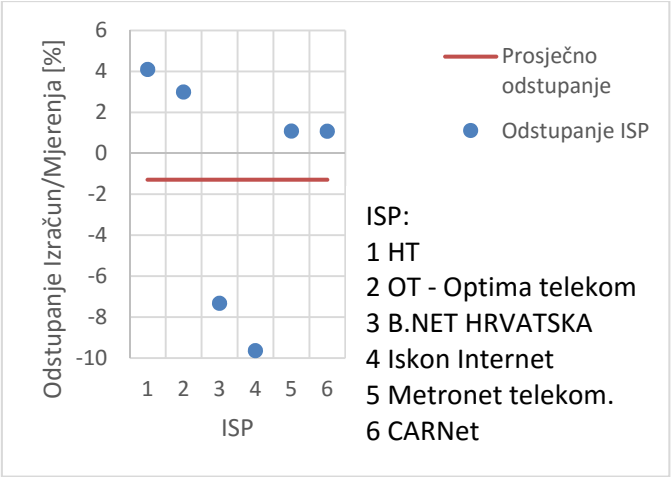
U Tablici 18. i Grafikonu 29. prikazana je usporedba rezultata provedenih mjerenja i analitičkog izračuna za slučaj pristupa klijenta sa svih 6 ISP-ova prema poslužitelju na linku Optima telekoma kod preuzimanja datoteke veličine 1 MB.

ISP	RTT [s]	Gubitci [%]	Vrijeme odgovora mrežne usluge [s]		Odstupanje Mjerenja/Izračun [%]
			Izračun	Mjerenja	
1 Hrvatski telekom	0,062	0,41	3,356793	3,5	4,09
2 Optima telekom	0,016	0,29	0,776124	0,8	2,98
3 B.NET	0,028	0,21	1,073351	1,0	-7,34
4 Iskon internet	0,054	0,16	1,754366	1,6	-9,65
5 Metronet telekomunikacije	0,010	0,49	0,791374	0,8	1,08
6 CARNet	0,009	0,53	0,791456	0,8	1,07
Prosječna vrijednost	*0,030*	*0,348*	*1,424*	*1,417*	*-1,293*

Tablica 18. *Usporedba rezultata mjerenja i analitičkog izračuna*



Grafikon 29. *Grafički prikaz usporedbe rezultata mjerenja i analitičkog izračuna*



Grafikon 30. *Grafički prikaz disperzije odstupanja analitičkog izračuna i rezultata mjerenja*

Usporedba rezultata analitičkog izračuna i mjerenja pokazuje visok stupanj podudarnosti i potvrdu analitičkog izračuna baziranog na Padhyevom matematičkom modelu TCP konekcije. Rezultati se više poklapaju za ISP-ove većih brzina koji koriste optičke linkove (Optima telekom, Metronet telekomunikacije, CARNet) i gdje je ograničenje propusnosti na strani poslužitelja u odnosu na ISP-ove koje koriste ADSL (Hrvatski telekom, Iskon Internet) ili kabelsku tehnologiju (B.NET) i gdje je propusnost limitirana na strani klijenta. Iz provedene usporedbe vidljivo je da se vrijeme odgovora poslužitelja mrežne usluge dobiveno analitičkim modelom može koristiti za predviđanje vremena odgovora poslužitelja mrežne usluge. Pri tome je vrijeme odgovora poslužitelja vidljivi rezultat, tj. posljedica primjene izračunate vrijednosti kompozitne DNS-metrike mrežne usluge, a poveznica između izračuna je RTT.

5.8. MOGUĆNOSTI PRIMJENE DSS METODE

DSS metoda je opcionalna, osnovna ili dodatna metoda za odabir optimalnog poslužitelja višestruko dostupne mrežne usluge koja ne isključuje niti jednu od postojećih metoda statičkog ili dinamičkog odabira poslužitelja, uključujući pri tome i poslužiteljske hardverske raspoređivače opterećenja ili alate za softversko raspoređivanje opterećenja, u slučajevima kada je više od jedne IP adrese izloženo i dostupno klijentu. Metoda može dodatno poboljšati proces odabira optimalnog poslužitelja na klijentskoj strani temeljem trenutnog opterećenja poslužitelja i mrežne topološke udaljenosti između poslužitelja i

pojedinog klijenta. Pri tome se sve potrebne aktivnosti odvijaju na strani klijenta tako da nema dodatnog opterećenja poslužitelja mrežne usluge zbog korištenja DSS metode.
DSS metoda je razvijena primarno za male i srednje velike mrežne usluge sa dinamičkim, specifičnim sadržajem generiranim temeljem specifičnih korisničkih upita. Metoda koristi postojeću DNS infrastrukturu za razmjenu potrebnih informacija između poslužitelja i klijenta te nije primarno namijenjena za inicijalni odabir DNS poslužitelja od strane klijenta ili raspoređivanje opterećenja između DNS poslužitelja gdje neautoritativni DNS poslužitelji uobičajeno odabiru autoritativni DNS poslužitelj temeljem *smoothed* RTT vrijednosti i neke od selekcijskih shema [99][100]. U situacijama gdje klijent može izabrati pojedinačni poslužitelj ili poslužiteljski grozd (predstavljen i dostupan klijentu kao jedan poslužitelj s jednom IP adresom) DSS metoda dostavlja klijentu dvije ili više, ali ograničen broj *unicast* ili *anycast* IP adresa poslužitelja koji imaju traženu mrežnu uslugu kako bi klijent odabrao optimalni poslužitelj za sebe. Pri tome poslužitelji i poslužiteljski grozdovi uobičajeno imaju različite topološke mrežne udaljenosti u odnosu na klijenta i/ili različita opterećenja pa je moguće da se u nekim implementacijama DSS metode dostupnost poslužitelja proširi na veći broj IP adresa zbog njihove bolje granulacije. Uz ovu primarnu svrhu, DSS metoda može biti korištena u različitim implementacijama kao samostalna ili dodatna metoda za odabir optimalnog poslužitelja, uključujući i CDN-ove i distribuciju statičkog sadržaja.

U implementacijama DSS metode koje uključuju parametar opterećenja poslužitelja u izračunu kompozitne DNS-metrike vrlo je važno održavati parametar LOAD ažurnim. Kako DNS poslužitelji omogućuju dinamičko osvježavanje RR zapisa [101], to može biti realizirano korištenjem postojećih DNS alata za dinamičko ažuriranje DNS zapisa i dovoljno kratkom TTL vrijednošću A/AAAA i DSS RR-a. Pri tome se mora uzeti u obzir da česta osvježavanja autoritativnih DNS poslužitelja ažurnim DSS parametrima mogu povećati vrijeme ažuriranja DNS zone, pogotovo za DNSSEC (*Domain Name System Security Extensions*) potpisane zone. Nula ili jako male TTL vrijednosti mogu povećati broj DNS upita u DNS razlučitelju i povećati opterećenje autoritativnih DNS poslužitelja. U takvim slučajevima postignuti indeks efikasnosti DSS metode treba biti dovoljno velik da bi opravdao povećanje DNS mrežnog prometa i opterećenja DNS poslužitelja. U slučajevima kada se oglašavanje opterećenja poslužitelja ne preporuča (najčešće iz sigurnosnih razloga) ono se može izostaviti, iako se DSS metoda može primijeniti i kao alat za efikasno sprječavanje napada uskraćivanjem usluge usmjeravanjem klijenata na neopterećene poslužitelje.

6. ZAKLJUČAK

Zahtjevi korisnika za što većom dostupnošću i što bržim odgovorom mrežnih usluga uvjetuju postavljanje više poslužitelja za pristup mrežnoj usluzi, pri čemu su poslužitelji vrlo često i prostorno distribuirani. Pri tome se pristup pojedinom poslužitelju može raspoređivati s obzirom na trenutno opterećenje pojedinog poslužitelja i komunikacijskog linka te omogućiti korisniku pristup poslužitelju mrežne usluge od kojeg će korisnik dobiti najbrži odgovor. Višestruki poslužitelji, a vrlo često i višestruki komunikacijski linkovi poslužitelja, omogućuju redundantnost mrežne usluge pri čemu je značajno što je moguće brže utvrditi nedostupnost pojedinog poslužitelja ili mrežne putanje prema poslužitelju i odabrati sljedeći dostupni poslužitelj, a koji će za korisnika u smislu brzine odgovora biti najpovoljniji.

DSS metoda za odabir optimalnog poslužitelja višestruko dostupne mrežne usluge omogućuje dinamički odabir poslužitelja temeljem informacija o opterećenosti poslužitelja i njegovoj mrežnoj udaljenosti u odnosu na klijenta. DSS metoda se zasniva na dodavanju novog DNS zapisa o resursima veličine 32 bajta za svaki poslužitelj u dodatnu sekciju DNS poruke. Odabir poslužitelja zasniva se na izračunu kompozitne DNS-metrike kojom se poslužitelji rangiraju koristeći parametre odziva mrežne usluge, koje klijent mjeri za svaki poslužitelj samostalno, te parametar opterećenja poslužitelja koji određuje administrator poslužitelja/usluge i prosljeđuje klijentu, zajedno sa pravilima za izračun kompozitne DNS-metrike.

Radom je potvrđena potreba za novom metodom kojom će se riješiti problem dinamičkog odabira poslužitelja višestruko dostupne mrežne usluge i brzog inicijalnog određivanja nedostupnosti poslužitelja mrežne usluge. Rezultatima mjerenja potvrđeni su utjecaji vremena mrežnog odziva i opterećenja poslužitelja na vrijeme odgovora poslužitelja na postavljeni korisnički zahtjev.

DSS metoda je pokazala značajno ubrzanje vremena odgovora mrežne usluge u odnosu na ostale četiri promatrane metode, koje je, u općem promatranom slučaju s dva poslužitelja na dva internetska linka, iznosilo od 8,5% do 26,8%. Korištenjem parametra za određivanje maksimalnog vremena odgovora mrežne usluge prije njegovog proglašavanja nedostupnim omogućeno je da administratori poslužitelja samostalno definiraju vrijeme za proglašavanje

nekog poslužitelja nedostupnim i da se ne ovisi o predefiniranoj općoj vrijednosti TCP parametra za utvrđivanje inicijalne nedostupnosti poslužitelja. Na taj način se vrijeme za utvrđivanje inicijalne nedostupnosti poslužitelja mrežne usluge može smanjiti s uobičajenih 21 sekunde na administratorski zadanu vrijednost, a koja može biti manja i od 1 sekunde. Razvijeni analitički izračun procjene vremena odgovora poslužitelja mrežne usluge, baziran na Padhye-ovom modelu TCP konekcije, omogućuje izračun očekivanog vremena izvršavanja zahtjeva za izračunatu kompozitnu DNS-metriku kada se u komunikaciji između klijenta i poslužitelja koristi TCP protokol.

6.1. ZNANSTVENI DOPRINOSI ISTRAŽIVANJA

Tri su znanstvena doprinosa ovog istraživanja:

1. Metoda dinamičkog odabira poslužitelja višestruko dostupne mrežne usluge na strani klijenta temeljem parametara vremena odziva mrežne usluge i opterećenja poslužitelja (*Dynamic Server Selection*, DSS)
2. Model povezivanja kompozitne DNS-metrike mrežne usluge s analitičkim izračunom vremena odgovora poslužitelja uz primjenu DSS metode
3. Prijedlog područja primjene i izvedbe DSS-metode te verifikacija predloženih rješenja u testnom okruženju

Znanstvenim doprinosima su, nakon analize područja istraživanja, predložena rješenja za poboljšanja dinamičkog odabira poslužitelja višestruko dostupne mrežne usluge u obliku nove metode i njezinih primjena, a za rješavanje uočenih nedostataka u postojećim metodama. Omogućeno je modeliranje povezivanja kompozitne DNS-metrike s analitičkim izračunom vremena odgovora poslužitelja višestruke mrežne usluge te je predloženo moguće područje primjene i izvedbe DSS metode uz verifikaciju predloženih rješenja provedenim mjerenjima u testnom okruženju.

6.2. DALJNJI RAZVOJ DSS METODE

Temeljem provedenih mjerenja i analize dobivenih rezultata, kao i daljnjeg razmatranja problematike odabira poslužitelja višestruko dostupne mrežne usluge DSS metodom, mogu se utvrditi sljedeći mogući pravci razvoja DSS metode:

1. Trenutno je mrežni parametar odziv mrežne usluge (RTT), kao parametar se može dodati raspoloživa propusnost komunikacijskog linka između klijenta i mrežne usluge, utvrditi utjecaj broja mjerenja i količine podataka na kvalitetnije utvrđivanje RTT-a i korelacije s propusnošću mreže
2. Omogućiti DNS klijentu što ažurniju informaciju o opterećenju poslužitelja, osvježavanje LOAD parametra trenutno ovisi o TTL parametru
3. Razraditi međusobni utjecaj LOAD, IMPACT i RESPONSE parametara na odabir poslužitelja višestruko dostupne mrežne usluge, razrada metode za optimiziranje raspoređivanja opterećenja između poslužitelja i komunikacijskih linkova na poslužiteljskoj strani, optimiziranje metode sa poslužiteljske strane korištenjem informacija o klijentima (korisnicima usluga) na poslužiteljskoj strani, utjecaj DSS metode na opterećenje sustava (s povećanim brojem klijenata)
4. Aplikacije DNS klijentima trebaju prosljeđivati informaciju o protokolu/portu za koji traže DNS upit kako bi se DSS odgovor optimizirao za traženu uslugu
5. Optimizacija metode za odlazne konekcije (npr. *upload*, *P2P file sharing*,…), primjena metode za specifične usluge (HTTP, P2P,…).

POPIS SLIKA, DIJAGRAMA, TABLICA, GRAFIKONA I PRILOGA:

Popis slika:

Popis dijagrama:

Popis tablica:

Popis grafikona:

Popis priloga:

POPIS POJMOVA

visoko dostupna mrežna usluga	sadrži mehanizme za održavanje usluge dostupnom bez obzira na prestanak rada pojedinih dijelova sustava. Zahtijeva oblikovanje sustava na način da se omogući detektiranje greške u sustavu i da se definiraju mehanizmi za oporavak (ponovnu dostupnost) usluge
eng. *server and link load balancing*	distribucija opterećenja poslužitelja i komunikacijskog linka na višestruke poslužitelje i komunikacijske linkove s ciljem optimiziranja korištenja resursa i ubrzanja vremena odziva
eng. *network and server failure failover*	prelazak na redundantni (sekundarni) mrežni segment ili poslužitelj u slučaju prestanka rada primarnog mrežnog segmenta ili poslužitelja
eng. *network response time*	vrijeme proteklo od slanja upita prema mrežnoj usluzi do trenutka primitka odgovora na postavljeni upit
eng. *resource records*	definiraju tipove podataka koje podržava *Domain Name System* (DNS)
eng. *round-robin*	algoritam za vremensko raspoređivanje procesa/resursa pri čemu je svakom procesu/resursu dodijeljen jednak dio vremena po kružnom principu dodjele, bez prioretiziranja
eng. *anycast*	metodologija mrežnog adresiranja i usmjeravanja u kojoj su datagrami jednog pošiljatelja usmjereni topologijski najbližoj točki u grupi potencijalnih primatelja
eng. *DNS proxy*	prima DNS upite od klijenata i prosljeđuje ih DNS poslužitelju, pri čemu može privremeno pohranjivati DNS zapise
CDN	eng. *Content Delivery Network* ili eng. *Content Distribution Network* je veliki distribuirani sustav poslužitelja u višestrukim podatkovnim centrima diljem interneta. Cilj CDN-a je pružanje

	usluge isporuke sadržaja korisnicima po principima visoke dostupnosti i visokih performansi
GeoIP usluga	omogućuje pružanje informacije o geografskom položaju temeljem IP adrese
Eng. *multihomed*	označava računalo ili mrežni uređaj koji je spojen na više od jedne računalne mreže
HTTP GET	metoda za traženje podatka od HTTP usluge, jednostavni HTTP GET provjerava samo zaglavlje odgovora (npr. 200 OK) dok puni HTTP GET provjerava sadržaj tijela odgovora
eng. *Gateway*	točka izlaska računalne mreže u drugu računalnu mrežu
eng. *Policy-Based Routing*	tehnika usmjeravanja mrežnog prometa temeljena na politici usmjeravanja postavljenoj od strane administratora usmjerivača
OSI referentni model	najkorišteniji apstraktni opis arhitekture računalne mreže, sastoji se od 7 slojeva
EIGRP	eng. *Enhanced Interior Gateway Routing Protocol*, usmjerivački protokol tvrtke Cisco Systems
MSS	maksimalna veličina segmenta, fiksna za svaku mrežnu putanju, uobičajeno 1.460 bajtova (1.500 bajtova MTU (*Maximum Transmission Unit*) umanjeno za 40 bajtova IP i TCP zaglavlja)

POPIS KRATICA

A/AAAA	IPv4/IPv6 mrežna adresa
ACK	eng. *Acknowledge*
ADSL	eng. *Asymmetric Digital Subscriber Line*
BDP	eng. *Bandwidth-Delay Product*
BER	eng. *Bit Error Rate*
BGP	eng. *Border Gateway Protocol*
CA	eng. *Collision Avoidance*
CARNet	*Hrvatska akademska i istraživačka mreža*
CDN	eng. *Content Delivery Network* ili eng. *Content Distribution Network*
CE	eng. *Connection Establishment*
CIX	eng. *Croatian Internet eXchange*
CNAME	eng. *Canonical Name* RR
CPU	eng. *Central Processing Unit*
cwnd	eng. *Congestion Window Size*
DNS	eng. *Domain Name System*
DNS-SD	eng. *DNS Service Discovery*
DNSSEC	eng. DNSSEC *Domain Name System Security Extensions*
DSS	eng. *Dynamic Server Selection*
EDNS0	eng. *Extension mechanisms for DNS*
ECN	eng. *Explicit Congestion Notification*
EIGRP	eng. *Enhanced Interior Gateway Routing Protocol*
FQDN	eng. *Fully Qualified Domain Name*
FTP	eng. *File Transfer Protocol*
FTPS	eng. *File Transfer Protocol Secure*
GPOS	eng. *Geographical Location*
HT	Hrvatski telekom d.d.
HTTP	eng. *HyperText Transfer Protocol*
HTTPS	eng. *HyperText Transfer Protocol Secure*
IANA	eng. *Internet Assigned Numbers Authority*
ICMP	eng. *Internet Control Message Protocol*
IN	eng. *Internet*
IP	eng. *Internet Protocol*

ISP	eng. *Internet Service Provider*
IXP	eng. *Internet Exchange Point*
I/O	eng. *Input/Output*
LOC	eng. *Location* RR
LSB	eng. *Least Significant Bit*
MSS	eng. *Maximum Segment Size*
MSB	eng. *Most Significant Bit*
MTU	eng. *Maximum Transmission Unit*
MX	eng. *Mail Exchanger* RR
NS	eng. *Name Server*
OSI	eng. *Open Systems Interconnection*
PER	eng. *Packet Error Rate*
QNAME	eng. *Query Domain Name*
QoS	eng. *Quality of Service*
RR	eng. *Resource Records*
RTO	eng. *Retransmission Timeout*
RTT	eng. *Round-Trip Time*
rwnd	eng. *Receiver's Advertised Window Size*
SFTP	eng. *SSH File Transfer Protocol* ili eng. *Secure File Transfer Protocol*
SMTP	eng. *Simple Mail Transfer Protocol*
SOA	eng. *Start of Authority* RR
SRV	eng. *Service* RR
SS	eng. *Slow Start*
SSL	eng. *Secure Sockets Layer*
SYN	eng. *Synchronize*
TCP	eng. *Transmission Control Protocol*
TD	eng. *Triple Duplicate*
TLS	eng. *Transport Layer Security*
TO	eng. *Time-out*
TTL	eng. *Time To Live*
TXT	eng. *Text* RR
UDP	eng. *User Datagram Protocol*
URL	eng. *Uniform Resource Locator*
ZIO	Zavod za informatiku Osijek

LITERATURA

[1] James Sonderegger, Orin Blomberg, Kieran Milne, Senad Palislamovic: *JUNOS High Availability*, O'Reilly Media, Inc. 2009.

[2] Hewlett-Packard: *WAN Design Guide The Lower Layers*; ProCurve Networking by HP, August 2005

[3] Cisco Systems: *CCIE Fundamentals: Network Design and Case Studies*; Second Edition, Cisco Press. 1999

[4] Cisco Systems: *CCNA Routing and Switching curriculum*; Cisco Networking Academy Program 2013

[5] Ya Wen: *Enterprise IP LAN/WAN Design version 1.1*; TAOS 2001

[6] Robert S. Cahn: *Wide Area Network Design: Concepts and Tools for Optimization*; Kaufmann Publishers, May 1998

[7] RFC 1034: *Domain Names - Concepts and Facilities*

[8] RFC 1035: *Domain Names - Domain Names - Implementation And Specification*

[9] Mark E. Crovella, Robert L. Carter: *Dynamic Server Selection in the Internet*, Technical Report, 1995 Boston University

[10] Robert L. Carter, Mark E. Crovella: *Server Selection using Dynamic Path Characterization in Wide-Area Networks*, 1997 In Proceedings of IEEE Infocom

[11] Zongming Fei, Samrat Bhattacharjee, Ellen W. Zegura, Mostafa H. Ammar: *A novel server selection technique for improving the response time of a replicated service*, INFOCOM 1998 Seventeenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE Volume:2, stranice 783 – 791

[12] Robert L. Carter, Mark E. Crovella: *On the network impact of dynamic server selection*, 1999 Elsevier Science B.V., Computer Networks 31, str. 2529-2558

[13] Toshihiko Shimokawa, Norihiko Yoshida, Kazuo Ushijima: *Flexible server selection using DNS*, 2000 Proceedings of the 2000 ICDCS Workshops

[14] Toshihiko Shimokawa, Norihiko Yoshida Kazuo Ushijima: *DNS-based Mechanism for Policy-added Server Selection*, International Conference on Advances in Infrastructure for Electronic Business, Science, and Education on the Internet, SSGRR 2000

[15] Anees Shaikh, Renu Tewari, Mukesh Agrawal: *On the Effectiveness of DNS-based Server Selection*, INFOCOM 2001, Twentieth Annual Joint Conference of the IEEE Computer and Communications Societies, Proceedings, IEEE (Volume: 3)

[16] Toshihiko Shimokawa, Norihiko Yoshida Kazuo Ushijima: *Server Selection Mechanism with Pluggable Selection Policies*, 2006 Electronics and Communications in Japan, Part 3, Vol. 89, No. 8, Translated from Denshi Joho Tsushin Gakkai Ronbunshi, Vol. J84-D-I, No. 9, September 2001, stranice 1396–1403

[17] Jianping Pan, Y. Thomas Hou, Bo Li: *An overview of DNS-based server selections in content distribution networks*, 2003 Elsevier, Computer Networks 43 (2003), stranice 695–711

[18] Kenichi Mase, Takayuki Kuribayashi, Akihiko Tsuno: *A Dynamic Server Selection Method Using QoS Statistics*, 2004 Electronics and Communications in Japan, Part 1, Vol. 87, No. 7, Translated from Denshi Joho Tsushin Gakkai Ronbunshi, Vol. J86-B, No. 3, 2003, stranice 499–510

[19] Lin Cai, Jun Ye, Jianping Pan, Xuemin (Sherman) Shen, Jon W. Mark: *Dynamic server selection using fuzzy inference in content distribution networks*, 2006 Elsevier, Computer Communications 29, stranice 1026–1038

[20] Hussein A. Alzoubi, Michael Rabinovich, Oliver Spatscheck: *MyXDNS: A Request Routing DNS Server With Decoupled Server Selection*, 2007 Proceedings of the 16th international conference on World Wide Web, stranice 351-360

[21] Nabor C. Mendonc, Jose Airton F. Silva, Ricardo O. Anido: *Client-side selection of replicated web services*: An empirical assessment, 2008 Elsevier, The Journal of Systems and Software 81 (2008), stranice 1346–1363

[22] Sushil Kumar Bhardwaj, Jagjit Singh Malhotra: *Simulation and comparison of hashing techniques in CDN DNS*, 2011 International Journal of Engineering Science & Technology, Vol. 3 Issue 4

[23] Martin O. Nicholes, Chen-Nee Chuah, S. Felix Wub, Biswanath Mukherjee: *Inter-domain collaborative routing (IDCR): Server selection for optimal client performance*, 2011 Elsevier, Computer Communications 34 (2011), stranice 1798–1809

[24] Yong Jin, Nariyoshi Yamai, Kiyohiko Okayama, Motonori Nakamura: *An Adaptive Route Selection Mechanism per Connection Based on Multipath DNS Round Trip*

Time on Multihomed Networks, 2012 Journal of Information Processing Vol. 20, No. 2, stranice 386-395

[25] Ingmar Poese, Benjamin Frank, Bernhard Ager, Georgios Smaragdakis, Steve Uhlig, Anja Feldmann: *Improving Content Delivery with PaDIS*, 2012 Internet Computing, IEEE (Volume: 16, Issue: 3), stranice 46-52

[26] Benjamin Frank: *Dynamic content delivery infrastructure deployment using network cloud resources*, Technische Universität Berlin, Fakultät IV - Elektrotechnik und Informatik, Doctoral Thesis, 2014

[27] T. Mackus, T. Simonaitis, D. Tamulioniene: *Adaptive TTL based Approach to Balance DNS Server Load*, 2012 Electronics and Electrical Engineering ISSN 1392 – 1215. No. 1 (117)

[28] Michael Brian Pope, Merrill Warkentin, Leigh A. Mutchler, Xin (Robert) Luo: *The Domain Name System—Past, Present, and Future*, 2012 Communications of the Association for Information Systems, Volume 30, Article 21, stranice 329-346

[29] Andreas Kamilaris, Andreas Pitsillides: *Using DNS for Global Discovery of Environmental Services*, 2012 8th International Conference on Web Information Systems and Technologies (WEBIST), stranice 280-284

[30] John S. Otto, Mario A. Sánchez, John P. Rula, Ted Stein, Fabián E. Bustamante: *namehelp: Intelligent Client-Side DNS Resolution*, ACM SIGCOMM Computer Communication Review Special Issue 42(4), 287-288. October 2012.

[31] L. Liu, C. Zhou, X. Zhang, Z. Guo, C. Li: *Probabilistic chunk scheduling approach in parallel multiple-server dash*, 2014 IEEE Visual Communications and Image Processing Conference, stranice 5–8

[32] S. Zhang, B. Li, B. Li, Presto *Towards fair and efficient http adaptive streaming from multiple servers*, 2015 IEEE International Conference on Communications (ICC), June 2015, stranice 6849–6854

[33] N. Bouten, M. Claeys, B. Van Poecke, S. Latrey, F. De Turck, *Dynamic Server Selection Strategy for Multi-server HTTP Adaptive Streaming Services*, 12th International Conference on Network and Service Management, Montreal, Quebec, Canada, Oct. 31 - Nov. 4, 2016, stranice 201-209

[34] RFC 1033: *Domain Administrators Operations Guide*

[35] ZYTRAX, Inc. : *Minimum TTLs & The Process of a Name Lookup* http://www.zytrax.com/books/dns/info/minimum-ttl.html, 08.01.2014.

[36] ZYTRAX, Inc. : *DNS BIND9 Query Statements* http://www.zytrax.com/books/dns/ch7/queries.html#rrset-order, 16.03.2014.

[37] RFC 1712: *DNS Encoding of Geographical Location*

[38] RFC 1876: *A Means for Expressing Location Information in the Domain Name System*

[39] RFC 2782: *A DNS RR for specifying the location of services (DNS SRV)*

[40] RFC 6763: *DNS-Based Service Discovery*

[41] ZYTRAX, Inc. : *DNS Sample BIND Configurations* http://www.zytrax.com/books/dns/ch6/#split-view, prosinac 2013.

[42] RFC 6891: *Extension Mechanisms for DNS (EDNS(0))*

[43] Array Networks, Inc.: *Global Server Load Balancing*, svibanj 2011.

[44] Cloud Leverage, Inc.: *Global Cloud Load Balancing: Direct visitors based on performance or geography*, http://cloudleverage.com/global-load-balancing/, 2014

[45] A10, Inc.: *Global Server Load Balancing (GSLB)*, http://www.a10networks.com/products/axseries-gslb.php, 2014.

[46] LoadDNS Inc., http://www.loaddns.com/Solutions/site_failover.aspx, 2014.

[47] http://edgedirector.com/, 2014.

[48] *Content Distribution Internetworking*, Proceedings of the Fifty-First Internet Engineering Task Force, London, August 6. 2001.

[49] RFC 3568: *Known Content Network (CN) Request-Routing Mechanisms*

[50] RFC 1546: *Host Anycasting Service*

[51] RFC 4271: *A Border Gateway Protocol 4 (BGP-4)*

[52] Puneet Sharma, Zhichen Xuy, Sujata Banerjee, SungJu Lee: *Estimating Network Proximity and Latency*, ACM SIGCOMM Computer Communication Review Volume 36 Issue 3 Pages 39-50, July 2006

[53] Manish Jain: *Available Bandwidth Estimation*, Networking and Telecom Group CoC, Georgia Tech, 8803 Class Presentation, 23.09.2003.

[54] jetNEXUS Inc.: http://www.hardwareloadbalancer.com/#load_balancer, 2014.

[55] Cisco Systems, Inc.: *Cisco IOS IP Configuration Guide Release 12.2 - Configuring Server Load Balancing*, 2006

[56] Microsoft Inc.: *HTTP Load Balancing using Application Request Routing*, http://www.iis.net/learn/extensions/configuring-application-request-routing-%28arr%29/http-load-balancing-using-application-request-routing, 2.6.2008.

[57] The Apache Software Foundation: *Apache Module mod_proxy_balancer*, http://httpd.apache.org/docs/2.2/mod/mod_proxy_balancer.html, 2014.

[58] Erik Nygren, Ramesh K. Sitaraman, Jennifer Sun: *The Akamai Network: A Platform for High-Performance Internet Applications*, ACM SIGOPS Operating Systems Review, vol. 44, no. 3, July 2010.

[59] Microsoft Inc.: *What is Azure?*, http://azure.microsoft.com/en-us/overview/what-is-azure/, 2014.

[60] Amazon Web Services Inc.: Amazon CloudFront, http://aws.amazon.com/cloudfront/, 2014

[61] RFC 6895: *Domain Name System (DNS) IANA Considerations*

[62] RFC 3597: *Handling of Unknown DNS Resource Record (RR) Types*

[63] RFC 1123: *Requirements for Internet Hosts -- Application and Support*

[64] RFC 2181: *Clarifications to the DNS Specification*

[65] Enhanced Interior Gateway Routing Protocol, http://www.cisco.com/c/en/us/support/docs/ip/enhanced-interior-gateway-routing-protocol-eigrp/16406-eigrp-toc.html, 2015

[66] Recursive and Iterative Queries, https://technet.microsoft.com/en-us/library/cc961401.aspx, 2015.

[67] Cisco Systems: *CCNA Accessing the WAN curriculum*; Cisco Networking Academy Program 2013

[68] Plug Things In: What is Latency?, http://www.plugthingsin.com/internet/speed/latency/, 2014.

[69] S. Shunmuga Krishnan, Ramesh K. Sitaraman: *Video stream quality impacts viewer behavior: inferring causality using quasi-experimental designs*, Proceedings of the 2012 ACM conference on Internet measurement conference, stranice 211-224

[70] Mohammad Alizadeh, Abdul Kabbani, Tom Edsall, Balaji Prabhakar, Amin Vahdat, Masato Yasuda: *Less is More: Trading a little Bandwidth for Ultra-Low Latency in the Data Center*, NSDI'12 Proceedings of the 9th USENIX conference on Networked Systems Design and Implementation

[71] The Most Misleading Measure of Response Time: Average, http://blog.optimizely.com/2013/12/11/why-cdn-balancing/, 2014

[72] Latency versus Bandwidth - What is it?, http://www.dslreports.com/faq/694, 2014.

[73] Latency and Bandwidth: Keys to User Experience, http://blog.equinix.com/2011/11/latency-bandwidth-keys-to-user-experience/, 2014

[74] Salil Banerjee: *IPV4 And IPV6 Latency Analysis*, ISOC Latency Workshop, 2013

[75] David A. Patterson: *Why Latency Lags Bandwidth, and What It Means to Computing*, High Performance Embedded Computing Proceedings, 2004

[76] David A. Patterson: *Latency Lags Bandwith*, Communications Of The ACM, October 2004/Vol. 47, No. 10

[77] How to Calculate TCP throughput for long distance WAN links, http://bradhedlund.com/2008/12/19/how-to-calculate-tcp-throughput-for-long-distance-links/, 2014

[78] Glen Turner: *TCP performance*, LinuxSA Adelaide, 2003

[79] Debessay Fesehaye Kassa: *Analytic Models of TCP Performance*, Master of Science Thesis, University of Stellenbosch, 2005

[80] RFC 1323: *TCP Extensions for High Performance*

[81] RFC 2018: *TCP Selective Acknowledgment Options*

[82] RFC 5681: *TCP Congestion Control*

[83] RFC 3168: *The Addition of Explicit Congestion Notification (ECN) to IP*

[84] RFC 6691: *TCP Options and Maximum Segment Size (MSS)*

[85] Inas Khalifa, Ljiljana Trajkovic: *An overview and comparison of analytical TCP models*, Proceedings of the 2004 International Symposium on Circuits and Systems ISCAS '04. Vol. 5, 2004.

[86] Matthew Mathis, Jeffrey Semke, Jamshid Mahdavi, Teunis Ott: *The Macroscopic Behavior of the TCP Congestion Avoidance Algorithm*, ACM SIGCOMM Computer Communication Review Homepage archive, Volume 27 Issue 3, July 1997, stranice 67-82

[87] Domenico Giustiniano, Eduard Goma, Alberto Lopez Toledo, George Athanasiou: *Optimizing TCP Performance in Multi-AP Residential Broadband Connections via Minislot Access*, Journal of Computer Networks and Communications Volume 2013, Article ID 752363

[88] Jitendra Padhye, Victor Firoiu, Don Towsley, Jim Kurose: *Modeling TCP Throughput: A Simple Model and its Empirical Validation*, SIGCOMM '98, stranice 303-314

[89] Jitendra Padhye: Model based approach to TCP friendly congestion control, Doctor dissertation, 2000

[90] Neal Cardwell, Stefan Savage, Thomas Anderson: *Modeling TCP Latency*, IEEE INFOCOM 2000, stranice 1724-1751

[91] Akamai Technologies, Inc.: Systems and methods for traffic management using load metrics reflective of a requested service, U.S. patent application, Publication Number: 20140156839, Publication Date: 05.06.2014

[92] Geoff Huston: *Measuring IP Network Performance*, The Internet Protocol Journal - Volume 6, Number 1, 2003

[93] ICMP Bandwith Test, https://www.mikrotik.com/documentation/manual_2.7/Tools/Speed.html, 2014

[94] Stephen D. Strowes: *Passively Measuring TCP Round-trip Times*, Queue - High-frequency Trading, ACM New York, Volume 11, Issue 8, 2013

[95] Croatia Internet eXchange, Peering matrica: http://www.cix.hr/index.php?id=1559, 2014

[96] RFC 6298: *Computing TCP's Retransmission Timer*

[97] RFC 1122: *Requirements for Internet Hosts -- Communication Layers*

[98] RFC 5482: *TCP User Timeout Option*

[99] Y. Yu, D. Wessels, M.Larson, L. Zhang, Authority server selection in DNS caching resolvers, SIGCOMM Computer Communication Review, Volume: 42, Number: 2, March 2012, pp. 80-86.

[100] Z. Wang, X. Wang, X. Lee. Analyzing BIND DNS Server Selection Algorithm, International Journal of Innovative Computing, Information and Control, Volume: 6, Number: 11, November 2010, pp. 5131–5142

[101] RFC 2136: Dynamic Updates in the Domain Name System (DNS UPDATE)

SAŽETAK

Zahtjevi korisnika za visokom dostupnošću i što bržim odgovorom mrežnih usluga uvjetuju postavljanje više poslužitelja, često korištenjem više komunikacijskih linkova, za pristup mrežnoj usluzi. Pri tome se odabir poslužitelja može temeljiti na trenutnom opterećenju pojedinog poslužitelja i karakteristikama mrežne putanje između korisnika i poslužitelja te omogućiti korisniku pristup poslužitelju mrežne usluge od kojeg će korisnik dobiti najbrži odgovor. Visoko dostupne mrežne usluge ugrađenom redundancijom omogućuju neprekinutost isporuke usluge u slučaju greške u sustavu. Pri tome je potrebno što je moguće brže utvrditi nedostupnost pojedinog poslužitelja ili mrežne putanje prema poslužitelju i odbrati sljedeći dostupni, a za korisnika po brzini odgovora na upit i najpovoljniji, poslužitelj mrežne usluge.

Dynamic Server Selection (DSS) je nova metoda za odabir optimalnog poslužitelja višestruko dostupne mrežne usluge. Metoda omogućuje dinamički odabir poslužitelja temeljem informacija o opterećenosti poslužitelja i njegovoj mrežnoj udaljenosti u odnosu na korisnika usluge. Odabir poslužitelja zasniva se na izračunu kompozitne DNS-metrike kojom se poslužitelji rangiraju od najpogodnijeg prema najnepogodnijem koristeći pri tome parametar odziva mrežne usluge, koje klijent mjeri za svaki poslužitelj samostalno, i parametar opterećenja poslužitelja koji određuje administrator poslužitelja i prosljeđuje klijentu, zajedno sa pravilima za izračun kompozitne DNS-metrike.

DSS metoda pokazuje značajno skraćenje vremena odgovora mrežne usluge u odnosu na ostale četiri promatrane metode (*Geographical, Hops, Random, RTT*), koje u općem promatranom slučaju sa dva poslužitelja na dva internetska linka, iznosi od 8,5% do 26,8%. Korištenjem parametra za određivanje maksimalnog vremena odgovora mrežne usluge prije njegovog proglašavanja nedostupnim omogućeno je da administratori poslužitelja samostalno definiraju vrijeme za proglašavanje nekog poslužitelja nedostupnim i da ne ovise o predefiniranoj vrijednosti TCP parametra za utvrđivanje inicijalne nedostupnosti poslužitelja. Na taj način se vrijeme za utvrđivanje inicijalne nedostupnosti poslužitelja mrežne usluge može smanjiti s uobičajenih 21 sekunde na administratorski zadanu vrijednost, a koja može biti manja i od 1 sekunde. Razvijeni model povezivanja kompozitne DNS-metrike s analitičkim izračunom vremena odgovora poslužitelja, baziran na Padhye-ovom modelu TCP konekcije, omogućuje izračun očekivanog vremena izvršavanja zahtjeva

za izračunatu kompozitnu DNS-metriku kada se u komunikaciji između klijenta i poslužitelja koristi TCP protokol.

Ključne riječi: mrežne usluge, visoka dostupnost, odabir poslužitelja, DNS, metrika

ABSTRACT

User requirements for high availability and faster response time of network services causes placement multiple servers, often using multiple communication links, for accessing network service. In doing so, the server selection can be based on the current load of each server and the characteristics of the network path between the user and the server and therefore enable user to access a network service from server which will give the quickest response. High available network services built-in redundancy enables continuity of service delivery in the event of a fault in the system. It is necessary for user, as quickly as possible, to determine the unavailability of a particular server or network path to the server and select next available and the most convenient, according to the speed of response to the request, server for network service.

Dynamic Server Selection (DSS) is a new method for selecting optimal server of multiple available network services. The method allows dynamic server selection based on information about the server load and its network distance in relation to the network service client. Server selection is based on the calculation of the composite DNS metric which ranks servers from most suitable to the worst suitable, using as parameter network service response, which user measure independently for each server, and the server load parameter, which specifies server administrator and forwarded it to the client, along with the rules for calculating composite DNS metrics.

DSS method demonstrates significant network services response times shortening in relation to other four observed methods (Geographical, Hops, Random, RTT), ranging from 8.5% to 26.8%, in the general case with two servers on two Internet links. Using defined parameter to determine the maximum response time of network service prior to its designation inaccessible, allows server administrators to independently define amount of time for declaring a server unavailable and therefore does not depend on predefined values of TCP parameters for determining the initial unavailability of the server. In this way, the time for

determining the initial unavailability of the network service's server can be reduced from usual value of 21 seconds to value defined by administrator, which can be less than 1 second. Developed model of relationship between composite DNS-metric and analytic calculation of server response time, allows calculation of expected execution time of client request for calculated composite DNS metrics, when communication between client and server uses TCP protocol.

Keyword: network service, high availability, server selection, DNS, metric

ŽIVOTOPIS

Dražen Tomić rođen je 19. listopada 1970. godine u Osijeku. Nakon završene osnovne škole u mjestu Dalj upisuje se u CUO 'Braća Ribar' u Osijeku gdje 1989. maturira i stječe zvanje Prirodoslovno-matematički tehničar. Iste godine odlazi na jednogodišnje služenje vojnog roka. Elektrotehnički fakultet u Osijeku upisuje 1990. godine a diplomira 1997. godine. Za vrijeme studiranja dvije godine aktivno sudjeluje u Domovinskom ratu. Na Elektrotehničkom fakultetu u Osijeku 2008. godine stječe titulu magistra znanosti s temom magistarskog rada: „Optimizacija modela gradske računalne mreže“.

Od 1998. do 2004. godine zaposlen je u Tajništvu Osječko-baranjske županije na mjestu stručnog suradnika a potom i savjetnika za informacijski sustav. Od 2003. do 2014. godine djelatnik je Elektrotehničkog fakulteta u Osijeku gdje obavlja poslove CARNet sistem inženjera, asistenta, voditelja Odsjeka za računalnu podršku te voditelja i predavača Cisco akademije mrežnih tehnologija. Od veljače 2005. godine zaposlen je i u Zavodu za informatiku Osijek, računskom centru Osječko-baranjske županije, na mjestu ravnatelja, a gdje ujedno obavlja i poslove sistemskog i mrežnog inženjera.

Područje interesa su mu računalne mreže, operacijski sustavi, projektiranje i sigurnost informatičkih sustava.

Oženjen je suprugom Viktorijom i otac kćeri Ljupke i Arete.

PRILOZI

Prilog 1: Model povezivanja kompozitne DNS-metrike mrežne usluge s analitičkim izračunom vremena odgovora poslužitelja (C++ program)

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>

double f(double p)  // Padhye formula 28
{
	return (1.0 + p + 2.0*pow(p,2) + 4.0*pow(p,3) + 8.0*pow(p,4) + 16.0*pow(p,5)
+ 32.0*pow(p,6));
}

double Q(double p, double w) // Padhye formula 23
{
	return __min(1.0,((1.0-pow(1.0-p,3))*(1.0 + pow(1.0-p,3))*(1.0-pow(1.0-p,w-
3))/(1.0-pow(1.0-p,w))));
}

double E(double p, double b)  // Padhye formula 13
{
	return ((2.0+b)/(3.0*b)) + sqrt(8.0*(1.0-p)/(3*b*p) + pow((2.0+b)/(3*b),2));
}

double bw_tcp_Padhye(int mss, double rtt, double loss_rate, int wmax, double
initial_rto, int b)  // Padhye formula 31
{
	double rate;
	double p = loss_rate/100.0;
	if (E(p,b)<wmax)
		rate = mss * ((1.0-p)/p +  E(p,b) + Q(p,E(p,b))/(1.0-p)) / (rtt *
(b/2.0*E(p,b)+1.0) + Q(p,E(p,b))*initial_rto*f(p)/(1.0-p));
	else
		rate = (mss * ((1.0-p)/p + wmax + Q(p,wmax)/(1.0-p))) / (rtt *
(b/8.0*wmax + (1.0-p)/(p*wmax) + 2.0) + Q(p,wmax)*initial_rto*f(p)/(1.0-p));
	return rate;
}

double bw_tcp_Padhye_DSS(int N, int mss, double rtt, double loss_rate, int wmax,
double initial_rto, int b)  // DSS formula (5)
{
	double rate;
	double p = loss_rate/100.0;
	if (E(p,b)<wmax)
		rate = N * (rtt * (b/2.0*E(p,b)+1.0) +
Q(p,E(p,b))*initial_rto*f(p)/(1.0-p)) / (mss * ((1.0-p)/p +  E(p,b) +
Q(p,E(p,b))/(1.0-p)));
	else
		rate = N * (rtt * (b/8.0*wmax + (1.0-p)/(p*wmax) + 2.0) +
Q(p,wmax)*initial_rto*f(p)/(1.0-p)) / (mss * ((1.0-p)/p + wmax + Q(p,wmax)/(1.0-
p)));
	return rate;
}

double dss_metric (int load, int max_load, int impact, double max_response, double
rtt)
{
```

```
	double dss_metric;
	if (max_load>0)
		dss_metric = load/max_load + impact*rtt/(255*max_response);
	else
		dss_metric = impact*rtt/(255*max_response);
	return dss_metric;
}

void main(void)
{
//	printf("\nw: %f ", E(0.016842272,2.0)),
//	printf("\nQ: %f ", Q(0.016842272,E(0.016842272,2.0))),
//	printf("\nG: %f ", f(0.016842272)),
    double rtt = 0.009; // Round Trip Time (RTT) / DSS response parameter (seconds)
	int mss = 1460; // Maximum Segment Size (bytes)
    double loss_rate = 0.53; // Packet lost rate %
	int wmax = 65535; // Maximum window size (bytes)
    double initial_rto = 3.0; // Initial Retransmission Timeout (RTO) (seconds)
	int b = 2; // Number of packets acknowledged by a received ACK
	int N = 1000000; // Number of transmitted bytes
	int load = 10; // DSS server load factor (min 0, max 255)
	int max_load = 30; // Max DSS server load factor (min 0, max 255)
	int impact = 200; // DSS network response time impact factor (min 0, max 255)
	double max_response = 0.05; // Max DSS network response time / Max RTT
(seconds)
	double Ds = 0.0; // Server processing delay (seconds)
	printf("\nRound Trip Time (RTT) / DSS response parameter: %f s\n", rtt);
	printf("\nTCP Bandwidth parameters:");
	printf("\n  Maximum Segment Size: %i B", mss);
	printf("\n  Packet lost rate: %f (%f%%)", loss_rate/100, loss_rate);
	printf("\n  Maximum window size: %i B", wmax);
	printf("\n  Initial Retransmission Timeout (RTO): %f s", initial_rto);
	printf("\n  Number of packets acknowledged by a received ACK: %i", b);
	printf("\n  Number of transmitted bytes: %i \n", N);
	printf("\nServer processing delay: %f s\n", Ds);
	printf("\nDSS parameters:");
	printf("\n  DSS server load factor (min 0, max 255): %i", load);
	printf("\n  Max DSS server load factor (min 0, max 255): %i", max_load);
	printf("\n  DSS network response time impact factor (min 0, max 255): %i",
impact);
	printf("\n  Max DSS network response time / Max RTT: %f\n", max_response);
	printf("\nMax TCP Bandwidth (Padhye): %f Mbps\n", bw_tcp_Padhye(mss, rtt,
loss_rate, wmax, initial_rto, b)*8/1000/1000);
//	printf("\nTranssmision time (Padhye): %f s\n", N/(bw_tcp_Padhye(mss, rtt,
loss_rate, wmax, initial_rto, b)));
	printf("\nTranssmision time (Padhye DSS): %f s\n", bw_tcp_Padhye_DSS(N, mss,
rtt, loss_rate, wmax, initial_rto, b)+Ds);
	printf("\nComposite DNS (DSS) metric: %f \n", dss_metric( load, max_load,
impact, max_response, rtt));
	getchar();
}
```

Primjer ispisa rezultata matematičkog modela:

Round Trip Time (RTT) / DSS response parameter: 0.009000 s

TCP Bandwidth parameters:
 Maximum Segment Size: 1460 B
 Packet lost rate: 0.005300 (0.530000%)

Maximum window size: 65535 B
Initial Retransmission Timeout (RTO): 3.000000 s
Number of packets acknowledged by a received ACK: 2
Number of transmitted bytes: 1000000

Server processing delay: 0.000000 s

DSS parameters:
DSS server load factor (min 0, max 255): 10
Max DSS server load factor (min 0, max 255): 30
DSS network response time impact factor (min 0, max 255): 200
Max DSS network response time / Max RTT: 0.050000

Max TCP Bandwidth (Padhye): 10.107951 Mbps

Transsmision time (Padhye DSS): 0.791456 s

Composite DNS (DSS) metric: 0.141176

Prilog 2: Tracert za ISP B.NET HRVATSKA d.o.o.:

Tracing route to programi1.zio.hr [213.191.152.142] over a maximum of 30 hops:

1	<1 ms	<1 ms	<1 ms	192.168.1.4
2	36 ms	13 ms	25 ms	88.207.124.1
3	21 ms	15 ms	23 ms	dh121-189.xnet.hr [83.139.121.189]
4	15 ms	19 ms	15 ms	dh121-189.xnet.hr [83.139.121.189]
5	26 ms	35 ms	32 ms	te3-8.ccr01.zag01.atlas.cogentco.com [149.6.30.73]
6	30 ms	33 ms	28 ms	te0-1-0-4.ccr21.vie01.atlas.cogentco.com [154.54.62.33]
7	31 ms	31 ms	28 ms	te0-0-0-6.ccr22.muc01.atlas.cogentco.com [130.117.3.21]
8	58 ms	48 ms	40 ms	te0-3-0-2.ccr22.fra03.atlas.cogentco.com [154.54.39.29]
9	59 ms	47 ms	33 ms	be2027.mag21.fra03.atlas.cogentco.com [154.54.74.142]
10	34 ms	37 ms	74 ms	ffm-b12-link.telia.net [213.248.92.141]
11	41 ms	54 ms	57 ms	ffm-bb1-link.telia.net [213.155.136.196]
12	88 ms	87 ms	81 ms	prag-bb1-link.telia.net [80.91.246.137]
13	97 ms	71 ms	93 ms	win-b4-link.telia.net [80.91.245.233]
14	79 ms	105 ms	79 ms	iskon-ic-123023-win-b4.c.telia.net [213.248.77.202]
15	106 ms	103 ms	80 ms	cat02.net.iskon.hr [89.164.64.214]
16	85 ms	147 ms	86 ms	cat03-te1-1-1.net.iskon.hr [89.164.64.7]
17	80 ms	80 ms	93 ms	89.164.86.161
18	75 ms	119 ms	94 ms	cs-os-access01-gi0-1.net.iskon.hr [89.164.64.91]
19	78 ms	88 ms	77 ms	213.191.152.142

Trace complete.

Tracing route to programi2.zio.hr [85.114.46.152] over a maximum of 30 hops:

1	<1 ms	<1 ms	<1 ms	192.168.1.4
2	21 ms	23 ms	31 ms	88.207.124.1
3	35 ms	35 ms	18 ms	dh121-189.xnet.hr [83.139.121.189]
4	21 ms	15 ms	15 ms	dh121-189.xnet.hr [83.139.121.189]
5	14 ms	19 ms	35 ms	dh121-193.xnet.hr [83.139.121.193]
6	21 ms	14 ms	35 ms	83.139.123.2
7	23 ms	12 ms	24 ms	193.192.15.76
8	25 ms	15 ms	13 ms	85.114.32.130
9	15 ms	15 ms	15 ms	85.114.32.138
10	29 ms	68 ms	42 ms	85.114.47.50
11	25 ms	17 ms	29 ms	85.114.46.152

Trace complete.

Prilog 3: RTO za Windows OS:

1. Telnet

a) Windows Telnet client – timeout nakon 21 sekunde:

Time	Source	Source port	Destination	Dest port	Protocol	Length	Info
REF	172.16.4.10	58934	172.16.0.60	21	TCP	74	58934 > ftp [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=4 SACK_PERM=1 TSval=77657644 TSecr=0
2.999862000	172.16.4.10	58934	172.16.0.60	21	TCP	74	58934 > ftp [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=4 SACK_PERM=1 TSval=77657944 TSecr=0
9.000706000	172.16.4.10	58934	172.16.0.60	21	TCP	70	58934 > ftp [SYN] Seq=0 Win=8192 Len=0 MSS=1460 SACK_PERM=1 TSval=77658544 TSecr=0

b) Putty – timeout nakon 21 sekunde:

Time	Source	Source port	Destination	Dest port	Protocol	Length	Info
REF	172.16.4.10	58957	172.16.0.60	23	TCP	74	58957 > telnet [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=4 SACK_PERM=1 TSval=77664778 TSecr=0
2.997420000	172.16.4.10	58957	172.16.0.60	23	TCP	74	58957 > telnet [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=4 SACK_PERM=1 TSval=77665078 TSecr=0
8.997176000	172.16.4.10	58957	172.16.0.60	23	TCP	70	58957 > telnet [SYN] Seq=0 Win=8192 Len=0 MSS=1460 SACK_PERM=1 TSval=77665678 TSecr=0

2. FTP

a) Windows FTP client – timeout nakon 21 sekunde:

Time	Source	Source port	Destination	Dest port	Protocol	Length	Info
REF	172.16.4.10	58977	172.16.0.60	21	TCP	74	58977 > ftp [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=4 SACK_PERM=1 TSval=77671523 TSecr=0
3.002882000	172.16.4.10	58977	172.16.0.60	21	TCP	74	58977 > ftp [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=4 SACK_PERM=1 TSval=77671823 TSecr=0
9.003680000	172.16.4.10	58977	172.16.0.60	21	TCP	70	58977 > ftp [SYN] Seq=0 Win=8192 Len=0 MSS=1460 SACK_PERM=1 TSval=77672423 TSecr=0

b) FileZilla – prvi timeout nakon 21 sekunde, drugi nakon 46 sekundi (21+4+21):

Time	Source	Source port	Destination	Dest port	Protocol	Length	Info
REF	172.16.4.10	59096	172.16.0.60	21	TCP	74	59096 > ftp [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=4 SACK_PERM=1 TSval=77705331 TSecr=0
3.005241000	172.16.4.10	59096	172.16.0.60	21	TCP	74	59096 > ftp [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=4 SACK_PERM=1 TSval=77705631 TSecr=0

9.006020000 172.16.4.10 59096 172.16.0.60 21 TCP 70 59096 > ftp [SYN] Seq=0 Win=8192 Len=0 MSS=1460 SACK_PERM=1 TSval=77706231 TSecr=0
25.278886000 172.16.4.10 59106 172.16.0.60 21 TCP 74 59106 > ftp [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=4 SACK_PERM=1 TSval=77707858 TSecr=0
28.278260000 172.16.4.10 59106 172.16.0.60 21 TCP 74 59106 > ftp [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=4 SACK_PERM=1 TSval=77708158 TSecr=0
34.278031000 172.16.4.10 59106 172.16.0.60 21 TCP 70 59106 > ftp [SYN] Seq=0 Win=8192 Len=0 MSS=1460 SACK_PERM=1 TSval=77708758 TSecr=0

3. HTTP

a) Internet Explorer – timeout nakon 42 sekunde (21+21):

Time Source Source port Destination Dest port Protocol Length Info
REF 172.16.4.10 58560 172.16.0.60 80 TCP 74 58560 > http [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=4 SACK_PERM=1 TSval=77566651 TSecr=0
2.999310000 172.16.4.10 58560 172.16.0.60 80 TCP 74 58560 > http [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=4 SACK_PERM=1 TSval=77566951 TSecr=0
9.000097000 172.16.4.10 58560 172.16.0.60 80 TCP 70 58560 > http [SYN] Seq=0 Win=8192 Len=0 MSS=1460 SACK_PERM=1 TSval=77567551 TSecr=0
21.001956000 172.16.4.10 58569 172.16.0.60 80 TCP 74 58569 > http [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=4 SACK_PERM=1 TSval=77568751 TSecr=0
24.001431000 172.16.4.10 58569 172.16.0.60 80 TCP 74 58569 > http [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=4 SACK_PERM=1 TSval=77569051 TSecr=0
30.002249000 172.16.4.10 58569 172.16.0.60 80 TCP 70 58569 > http [SYN] Seq=0 Win=8192 Len=0 MSS=1460 SACK_PERM=1 TSval=77569651 TSecr=0

b) Chrome – timeout nakon 21 sekunde:

Time Source Source port Destination Dest port Protocol Length Info
REF 172.16.4.10 58813 172.16.0.60 80 TCP 74 58813 > http [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=4 SACK_PERM=1 TSval=77629077 TSecr=0
0.000823000 172.16.4.10 58814 172.16.0.60 80 TCP 74 58814 > http [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=4 SACK_PERM=1 TSval=77629077 TSecr=0
0.250464000 172.16.4.10 58819 172.16.0.60 80 TCP 74 58819 > http [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=4 SACK_PERM=1 TSval=77629102 TSecr=0

3.000061000 172.16.4.10 58814 172.16.0.60 80 TCP 74 58814 > http [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=4 SACK_PERM=1 TSval=77629377 TSecr=0
3.000094000 172.16.4.10 58813 172.16.0.60 80 TCP 74 58813 > http [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=4 SACK_PERM=1 TSval=77629377 TSecr=0
3.250025000 172.16.4.10 58819 172.16.0.60 80 TCP 74 58819 > http [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=4 SACK_PERM=1 TSval=77629402 TSecr=0
9.000166000 172.16.4.10 58814 172.16.0.60 80 TCP 70 58814 > http [SYN] Seq=0 Win=8192 Len=0 MSS=1460 SACK_PERM=1 TSval=77629977 TSecr=0
9.000195000 172.16.4.10 58813 172.16.0.60 80 TCP 70 58813 > http [SYN] Seq=0 Win=8192 Len=0 MSS=1460 SACK_PERM=1 TSval=77629977 TSecr=0
9.256093000 172.16.4.10 58819 172.16.0.60 80 TCP 70 58819 > http [SYN] Seq=0 Win=8192 Len=0 MSS=1460 SACK_PERM=1 TSval=77630002 TSecr=0

c) Firefox – timeout nakon 42 sekunde:

Time Source Source port Destination Dest port Protocol Length Info
REF 172.16.4.10 58630 172.16.0.60 80 TCP 74 58630 > http [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=4 SACK_PERM=1 TSval=77581793 TSecr=0
0.258189000 172.16.4.10 58631 172.16.0.60 80 TCP 74 58631 > http [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=4 SACK_PERM=1 TSval=77581819 TSecr=0
2.998497000 172.16.4.10 58630 172.16.0.60 80 TCP 74 58630 > http [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=4 SACK_PERM=1 TSval=77582093 TSecr=0
3.258483000 172.16.4.10 58631 172.16.0.60 80 TCP 74 58631 > http [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=4 SACK_PERM=1 TSval=77582119 TSecr=0
8.998254000 172.16.4.10 58630 172.16.0.60 80 TCP 70 58630 > http [SYN] Seq=0 Win=8192 Len=0 MSS=1460 SACK_PERM=1 TSval=77582693 TSecr=0
9.258249000 172.16.4.10 58631 172.16.0.60 80 TCP 70 58631 > http [SYN] Seq=0 Win=8192 Len=0 MSS=1460 SACK_PERM=1 TSval=77582719 TSecr=0
21.000588000 172.16.4.10 58638 172.16.0.60 80 TCP 74 58638 > http [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=4 SACK_PERM=1 TSval=77583893 TSecr=0
21.260421000 172.16.4.10 58639 172.16.0.60 80 TCP 74 58639 > http [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=4 SACK_PERM=1 TSval=77583919 TSecr=0
21.260639000 172.16.4.10 58640 172.16.0.60 80 TCP 74 58640 > http [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=4 SACK_PERM=1 TSval=77583919 TSecr=0

```
23.999636000  172.16.4.10   58638  172.16.0.60   80  TCP  74
      58638 > http [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=4 SACK_PERM=1
      TSval=77584193 TSecr=0
24.259649000  172.16.4.10   58639  172.16.0.60   80  TCP  74
      58639 > http [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=4 SACK_PERM=1
      TSval=77584219 TSecr=0
24.259664000  172.16.4.10   58640  172.16.0.60   80  TCP  74
      58640 > http [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=4 SACK_PERM=1
      TSval=77584219 TSecr=0
30.000395000  172.16.4.10   58638  172.16.0.60   80  TCP  70
      58638 > http [SYN] Seq=0 Win=8192 Len=0 MSS=1460 SACK_PERM=1
      TSval=77584793 TSecr=0
30.260407000  172.16.4.10   58639  172.16.0.60   80  TCP  70
      58639 > http [SYN] Seq=0 Win=8192 Len=0 MSS=1460 SACK_PERM=1
      TSval=77584819 TSecr=0
30.260424000  172.16.4.10   58640  172.16.0.60   80  TCP  70
      58640 > http [SYN] Seq=0 Win=8192 Len=0 MSS=1460 SACK_PERM=1
      TSval=77584819 TSecr=0
42.262752000  172.16.4.10   58657  172.16.0.60   80  TCP  74
      58657 > http [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=4 SACK_PERM=1
      TSval=77586019 TSecr=0
45.261852000  172.16.4.10   58657  172.16.0.60   80  TCP  74
      58657 > http [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=4 SACK_PERM=1
      TSval=77586319 TSecr=0
51.262628000  172.16.4.10   58657  172.16.0.60   80  TCP  70
      58657 > http [SYN] Seq=0 Win=8192 Len=0 MSS=1460 SACK_PERM=1
      TSval=77586919 TSecr=0
```

Prilog 4: Otvaranje dodatnog soketa:

Firefox: ako u roku od 250 ms ne stigne odgovor otvara se dodatni soket

a) Odgovor ne stiže u prvih 250 ms, otvara se dodatni soket:

Time Source Source port Destination Dest port Protocol Length Info
REF 172.16.4.10 60947 172.16.0.60 80 TCP 74 60947 > http [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=4 SACK_PERM=1 TSval=8597599 TSecr=0
0.254241000 172.16.4.10 60948 172.16.0.60 80 TCP 74 60948 > http [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=4 SACK_PERM=1 TSval=8597624 TSecr=0
3.003616000 172.16.4.10 60947 172.16.0.60 80 TCP 74 60947 > http [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=4 SACK_PERM=1 TSval=8597899 TSecr=0
3.253617000 172.16.4.10 60948 172.16.0.60 80 TCP 74 60948 > http [SYN]
…

b) Odgovor stiže u prvih 250 ms, ne otvara se dodatni soket:

Time Source Source port Destination Dest port Protocol Length Info
REF 172.16.4.10 63469 172.16.0.56 80 TCP 74 63469 > http [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=4 SACK_PERM=1 TSval=9271249 TSecr=0
0.000923000 172.16.0.56 80 172.16.4.10 63469 TCP 70 http > 63469 [SYN, ACK] Seq=0 Ack=1 Win=8192 Len=0 MSS=1460 SACK_PERM=1 TSval=796529180 TSecr=9271249
0.001002000 172.16.4.10 63469 172.16.0.56 80 TCP 66 63469 > http [ACK] Seq=1 Ack=1 Win=65160 Len=0 TSval=9271249 TSecr=796529180
0.001179000 172.16.4.10 63469 172.16.0.56 80 HTTP 372 GET / HTTP/1.1
0.189947000 172.16.0.56 80 172.16.4.10 63469 TCP 66 http > 63469 [ACK] Seq=1 Ack=307 Win=65160 Len=0 TSval=796529199 TSecr=9271249
…

Chrome: u roku od 1 ms otvara se dodatni soket, ako u roku od 250 ms ne stigne odgovor otvara se još jedan dodatni soket

a) Nakon 1 ms otvara se dodatni soket, odgovor ne stiže u prvih 250 ms, otvara se još jedan dodatni soket:

Time Source Source port Destination Dest port Protocol Length Info
REF 172.16.4.10 58813 172.16.0.60 80 TCP 74 58813 > http [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=4 SACK_PERM=1 TSval=77629077 TSecr=0
0.000823000 172.16.4.10 58814 172.16.0.60 80 TCP 74 58814 > http [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=4 SACK_PERM=1 TSval=77629077 TSecr=0

0.250464000 172.16.4.10 58819 172.16.0.60 80 TCP 74 58819 > http [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=4 SACK_PERM=1 TSval=77629102 TSecr=0
3.000061000 172.16.4.10 58814 172.16.0.60 80 TCP 74 58814 > http [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=4 SACK_PERM=1 TSval=77629377 TSecr=0
3.000094000 172.16.4.10 58813 172.16.0.60 80 TCP 74 58813 > http [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=4 SACK_PERM=1 TSval=77629377 TSecr=0
3.250025000 172.16.4.10 58819 172.16.0.60 80 TCP 74 58819 > http [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=4 SACK_PERM=1 TSval=77629402 TSecr=0

b) Odgovor stiže u prvih 250 ms, otvara se samo novi soket nakon 1 ms, komunikacija se nastavlja soketom za koji je prvi stigao odgovor od poslužitelja:

Time Source Source port Destination Dest port Protocol Length Info
REF 172.16.4.10 63566 172.16.0.56 80 TCP 74 63566 > http [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=4 SACK_PERM=1 TSval=9284081 TSecr=0
0.000895000 172.16.4.10 63567 172.16.0.56 80 TCP 74 63567 > http [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=4 SACK_PERM=1 TSval=9284081 TSecr=0
0.001921000 172.16.0.56 80 172.16.4.10 63567 TCP 70 http > 63567 [SYN, ACK] Seq=0 Ack=1 Win=8192 Len=0 MSS=1460 SACK_PERM=1 TSval=796542016 TSecr=9284081
0.001922000 172.16.0.56 80 172.16.4.10 63566 TCP 70 http > 63566 [SYN, ACK] Seq=0 Ack=1 Win=8192 Len=0 MSS=1460 SACK_PERM=1 TSval=796542016 TSecr=9284081
0.001989000 172.16.4.10 63567 172.16.0.56 80 TCP 66 63567 > http [ACK] Seq=1 Ack=1 Win=65160 Len=0 TSval=9284081 TSecr=796542016
0.001990000 172.16.4.10 63566 172.16.0.56 80 TCP 66 63566 > http [ACK] Seq=1 Ack=1 Win=65160 Len=0 TSval=9284081 TSecr=796542016
0.042135000 172.16.4.10 63567 172.16.0.56 80 HTTP 412 GET / HTTP/1.1
0.046673000 172.16.0.56 80 172.16.4.10 63567 TCP 1514 [TCP segment of a reassembled PDU]
0.047914000 172.16.0.56 80 172.16.4.10 63567 TCP 1514 [TCP segment of a reassembled PDU]
0.047927000 172.16.4.10 63567 172.16.0.56 80 TCP 66 63567 > http [ACK] Seq=347 Ack=2897 Win=65160 Len=0 TSval=9284086 TSecr=796542021
0.049743000 172.16.0.56 80 172.16.4.10 63567 TCP 1514 [TCP segment of a reassembled PDU]

Prilog 5: Prelazak na drugi DNS zapis nakon RTO za Windows OS:

C:\Users\dtomic>nslookup testiranje.zio.hr
Server: obz1.obz.hr
Address: 172.16.0.2
Name: testiranje.zio.hr
Addresses: 172.16.0.60
172.16.0.56
C:\Users\dtomic>

1. Telnet

a) Windows Telnet clinet:

Time Source Source port Destination Dest port Protocol Length Info
REF 172.16.4.10 59772 172.16.0.60 23 TCP 74 59772 > telnet [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=4 SACK_PERM=1 TSval=8305955 TSecr=0
2.999937000 172.16.4.10 59772 172.16.0.60 23 TCP 74 59772 > telnet [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=4 SACK_PERM=1 TSval=8306255 TSecr=0
8.999808000 172.16.4.10 59772 172.16.0.60 23 TCP 70 59772 > telnet [SYN] Seq=0 Win=8192 Len=0 MSS=1460 SACK_PERM=1 TSval=8306855 TSecr=0
21.001816000 172.16.4.10 59778 172.16.0.56 23 TCP 74 59778 > telnet [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=4 SACK_PERM=1 TSval=8308055 TSecr=0
21.002721000 172.16.0.56 23 172.16.4.10 59778 TCP 70 telnet > 59778 [SYN, ACK] Seq=0 Ack=1 Win=8192 Len=0 MSS=1460 SACK_PERM=1 TSval=795565727 TSecr=8308055
21.002833000 172.16.4.10 59778 172.16.0.56 23 TCP 66 59778 > telnet [ACK] Seq=1 Ack=1 Win=65160 Len=0 TSval=8308056 TSecr=795565727
25.525768000 172.16.0.56 23 172.16.4.10 59778 TELNET 87 Telnet Data ...

b) Putty:

Time Source Source port Destination Dest port Protocol Length Info
REF 172.16.4.10 59813 172.16.0.60 23 TCP 74 59813 > telnet [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=4 SACK_PERM=1 TSval=8318427 TSecr=0
3.001303000 172.16.4.10 59813 172.16.0.60 23 TCP 74 59813 > telnet [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=4 SACK_PERM=1 TSval=8318727 TSecr=0
9.002160000 172.16.4.10 59813 172.16.0.60 23 TCP 70 59813 > telnet [SYN] Seq=0 Win=8192 Len=0 MSS=1460 SACK_PERM=1 TSval=8319328 TSecr=0

21.003517000 172.16.4.10 59821 172.16.0.56 23 TCP 74 59821 > telnet [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=4 SACK_PERM=1 TSval=8320528 TSecr=0
21.004413000 172.16.0.56 23 172.16.4.10 59821 TCP 70 telnet > 59821 [SYN, ACK] Seq=0 Ack=1 Win=8192 Len=0 MSS=1460 SACK_PERM=1 TSval=795578203 TSecr=8320528
21.004521000 172.16.4.10 59821 172.16.0.56 23 TCP 66 59821 > telnet [ACK] Seq=1 Ack=1 Win=65160 Len=0 TSval=8320528 TSecr=795578203
21.004704000 172.16.4.10 59821 172.16.0.56 23 TELNET 87 Telnet Data ...
21.201419000 172.16.0.56 23 172.16.4.10 59821 TCP 66 telnet > 59821 [ACK] Seq=1 Ack=22 Win=65160 Len=0 TSval=795578222 TSecr=8320528

2. FTP

a) Windows FTP client:

Time Source Source port Destination Dest port Protocol Length Info
REF 172.16.4.10 63945 172.16.0.60 23 TCP 74 63945 > telnet [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=4 SACK_PERM=1 TSval=41964124 TSecr=0
2.992985000 172.16.4.10 63945 172.16.0.60 23 TCP 74 63945 > telnet [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=4 SACK_PERM=1 TSval=41964424 TSecr=0
8.993857000 172.16.4.10 63945 172.16.0.60 23 TCP 70 63945 > telnet [SYN] Seq=0 Win=8192 Len=0 MSS=1460 SACK_PERM=1 TSval=41965024 TSecr=0
20.994837000 172.16.4.10 63952 172.16.0.56 23 TCP 74 63952 > telnet [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=4 SACK_PERM=1 TSval=41966224 TSecr=0
20.995928000 172.16.0.56 23 172.16.4.10 63952 TCP 70 telnet > 63952 [SYN, ACK] Seq=0 Ack=1 Win=8192 Len=0 MSS=1460 SACK_PERM=1 TSval=829242520 TSecr=41966224
20.996040000 172.16.4.10 63952 172.16.0.56 23 TCP 66 63952 > telnet [ACK] Seq=1 Ack=1 Win=65160 Len=0 TSval=41966224 TSecr=829242520
25.520089000 172.16.0.56 23 172.16.4.10 63952 TELNET 87 Telnet Data ...

b) FileZilla:

Time Source Source port Destination Dest port Protocol Length Info
REF 172.16.4.10 59679 172.16.0.60 21 TCP 74 59679 > ftp [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=4 SACK_PERM=1 TSval=8278541 TSecr=0
2.999655000 172.16.4.10 59679 172.16.0.60 21 TCP 74 59679 > ftp [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=4 SACK_PERM=1 TSval=8278841 TSecr=0

9.000526000 172.16.4.10 59679 172.16.0.60 21 TCP 70 59679 > ftp [SYN] Seq=0 Win=8192 Len=0 MSS=1460 SACK_PERM=1 TSval=8279441 TSecr=0
25.264634000 172.16.4.10 59688 172.16.0.60 21 TCP 74 59688 > ftp [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=4 SACK_PERM=1 TSval=8281068 TSecr=0
28.272022000 172.16.4.10 59688 172.16.0.60 21 TCP 74 59688 > ftp [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=4 SACK_PERM=1 TSval=8281368 TSecr=0
34.271956000 172.16.4.10 59688 172.16.0.60 21 TCP 70 59688 > ftp [SYN] Seq=0 Win=8192 Len=0 MSS=1460 SACK_PERM=1 TSval=8281968 TSecr=0

3. HTTP

a) Internet Explorer:

Time Source Source port Destination Dest port Protocol Length Info
REF 172.16.4.10 60519 172.16.0.60 80 TCP 74 60519 > http [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=4 SACK_PERM=1 TSval=8498455 TSecr=0
3.003947000 172.16.4.10 60519 172.16.0.60 80 TCP 74 60519 > http [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=4 SACK_PERM=1 TSval=8498755 TSecr=0
9.004811000 172.16.4.10 60519 172.16.0.60 80 TCP 70 60519 > http [SYN] Seq=0 Win=8192 Len=0 MSS=1460 SACK_PERM=1 TSval=8499355 TSecr=0
21.005924000 172.16.4.10 60528 172.16.0.56 80 TCP 74 60528 > http [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=4 SACK_PERM=1 TSval=8500555 TSecr=0
21.006812000 172.16.0.56 80 172.16.4.10 60528 TCP 70 http > 60528 [SYN, ACK] Seq=0 Ack=1 Win=8192 Len=0 MSS=1460 SACK_PERM=1 TSval=795758279 TSecr=8500555
21.006896000 172.16.4.10 60528 172.16.0.56 80 TCP 66 60528 > http [ACK] Seq=1 Ack=1 Win=65160 Len=0 TSval=8500555 TSecr=795758279

b) Chrome:

Time Source Source port Destination Dest port Protocol Length Info
REF 172.16.4.10 61011 172.16.0.60 80 TCP 74 61011 > http [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=4 SACK_PERM=1 TSval=8609008 TSecr=0
0.000220000 172.16.4.10 61012 172.16.0.60 80 TCP 74 61012 > http [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=4 SACK_PERM=1 TSval=8609008 TSecr=0
0.246725000 172.16.4.10 61013 172.16.0.60 80 TCP 74 61013 > http [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=4 SACK_PERM=1 TSval=8609032 TSecr=0

3.009503000 172.16.4.10 61012 172.16.0.60 80 TCP 74 61012 > http [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=4 SACK_PERM=1 TSval=8609308 TSecr=0
3.009517000 172.16.4.10 61011 172.16.0.60 80 TCP 74 61011 > http [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=4 SACK_PERM=1 TSval=8609308 TSecr=0
3.246583000 172.16.4.10 61013 172.16.0.60 80 TCP 74 61013 > http [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=4 SACK_PERM=1 TSval=8609332 TSecr=0
9.009760000 172.16.4.10 61012 172.16.0.60 80 TCP 70 61012 > http [SYN] Seq=0 Win=8192 Len=0 MSS=1460 SACK_PERM=1 TSval=8609908 TSecr=0
9.009764000 172.16.4.10 61011 172.16.0.60 80 TCP 70 61011 > http [SYN] Seq=0 Win=8192 Len=0 MSS=1460 SACK_PERM=1 TSval=8609908 TSecr=0
9.246805000 172.16.4.10 61013 172.16.0.60 80 TCP 70 61013 > http [SYN] Seq=0 Win=8192 Len=0 MSS=1460 SACK_PERM=1 TSval=8609932 TSecr=0
21.002709000 172.16.4.10 61022 172.16.0.56 80 TCP 74 61022 > http [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=4 SACK_PERM=1 TSval=8611108 TSecr=0
21.004006000 172.16.0.56 80 172.16.4.10 61022 TCP 70 http > 61022 [SYN, ACK] Seq=0 Ack=1 Win=8192 Len=0 MSS=1460 SACK_PERM=1 TSval=795868861 TSecr=8611108
21.004079000 172.16.4.10 61022 172.16.0.56 80 TCP 66 61022 > http [ACK] Seq=1 Ack=1 Win=65160 Len=0 TSval=8611108 TSecr=795868861

c) Firefox:

Time Source Source port Destination Dest port Protocol Length Info
REF 172.16.4.10 60947 172.16.0.60 80 TCP 74 60947 > http [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=4 SACK_PERM=1 TSval=8597599 TSecr=0
0.254241000 172.16.4.10 60948 172.16.0.60 80 TCP 74 60948 > http [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=4 SACK_PERM=1 TSval=8597624 TSecr=0
3.003616000 172.16.4.10 60947 172.16.0.60 80 TCP 74 60947 > http [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=4 SACK_PERM=1 TSval=8597899 TSecr=0
3.253617000 172.16.4.10 60948 172.16.0.60 80 TCP 74 60948 > http [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=4 SACK_PERM=1 TSval=8597924 TSecr=0
9.004475000 172.16.4.10 60947 172.16.0.60 80 TCP 70 60947 > http [SYN] Seq=0 Win=8192 Len=0 MSS=1460 SACK_PERM=1 TSval=8598499 TSecr=0
9.254476000 172.16.4.10 60948 172.16.0.60 80 TCP 70 60948 > http [SYN] Seq=0 Win=8192 Len=0 MSS=1460 SACK_PERM=1 TSval=8598524 TSecr=0

```
21.005707000 172.16.4.10 60955 172.16.0.56 80 TCP 74
    60955 > http [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=4 SACK_PERM=1
    TSval=8599699 TSecr=0
21.013619000 172.16.0.56 80 172.16.4.10 60955 TCP 70 http
    > 60955 [SYN, ACK] Seq=0 Ack=1 Win=8192 Len=0 MSS=1460
    SACK_PERM=1 TSval=795857449 TSecr=8599699
21.013700000 172.16.4.10 60955 172.16.0.56 80 TCP 66
    60955 > http [ACK] Seq=1 Ack=1 Win=65160 Len=0 TSval=8599700
    TSecr=795857449
```