

**ELEKTROTEHNIČKI FAKULTET UNIVERZITETA U BEOGRADU**



**ANALIZA MULTIKAST KONTROLERA ZA RAZLIČITE PARAMETRE  
MULTIKAST SAOBRAĆAJA**

– Diplomski rad –

Kandidat:

Marija U. Vidović 2009/403

Mentor:

dr Zoran Čiča, doc.

Beograd, Oktobar 2015.

# SADRŽAJ

<b>SADRŽAJ .....</b>	<b>2</b>
<b>1. UVOD .....</b>	<b>3</b>
<b>2. IP MULTIKAST.....</b>	<b>4</b>
2.1. FUNKCIJA I ZNAČAJ MULTIKASTA .....	4
2.2. PROBLEMI KOJU MOGU NASTATI PRIMENOM MULTIKASTA.....	5
<b>3. OPIS APLIKACIJE .....</b>	<b>7</b>
<b>4. REZULTATI SIMULACIJE.....</b>	<b>8</b>
4.1. GRAFIČKO PREDSTAVLJANJE REZULTATA SIMULACIJE.....	8
4.2. ANALIZA REZULTATA SIMULACIJE.....	13
<b>5. ZAKLJUČAK.....</b>	<b>14</b>
<b>LITERATURA .....</b>	<b>15</b>
<b>A. KOD SIMULACIJE REALIZOVAN U MATLAB-U.....</b>	<b>16</b>

# 1. UVOD

Prenos od jednog izvorišta do pojedinih odredišta (*one to many*) kao vrsta multikasta, predstavlja jedan od prvih servisa za poboljšanje komunikacije u Internet protokolu (IP). Multikast pruža višestruke mogućnosti slanja jednog paketa za pristup milionima potencijalnih korisnika. Bez multikasta, aplikacije kao što su *streaming Internet radio*, *video-on-demand (VoD)*, *Internet-based video jukeboxes* i video konferencije je gotovo nemoguće realizovati [1]. Međutim, još uvek nije uobičajena primena na Internetu, s obzirom da je komplikovano realizovati IP multikast na globalnom nivou [2].

Iz tog razloga, u ovom radu je izvršena analiza rutiranja paketa korišćenjem upravo jedne takve multikast komunikacije. S obzirom da primena svakog pristupa zavisi od slučaja do slučaja, ispitivana su dva pristupa – VOQ (*Virtual Output Queing*) i FIFO (*First In - First Out*) pristup.

U ovom radu najpre će biti reči o samom IP multikastu i njegovim karakteristikama, zatim o prednostima i problemima koji mogu nastati primenom multikasta. Zatim će biti ukratko opisane vrste protokola za rešavanje takvih problema i predstavljen opis aplikacije koja je korišćena u ovom radu.

Za analizu je korišćena simulacija u programskom jeziku *MATLAB*. Na osnovu dobijenih rezultata simulacije može se zaključiti o prednostima i manama svakog od korišćenih pristupa. Dobijeni rezultati analize značajni su za sagledavanje efikasnosti korišćenja ispitanog algoritma za raspoređivanje multikast saobraćaja unutar svičeva i rutera..

## 2. IP MULTIKAST

Komunikacija između dva pojedinačna korisnika (*unicast*) ili od jednog korisnika ka svim ostalim (*broadcast*) predstavlja tradicionalni način komunikacije u računarskim mrežama. IP multikast, sa druge strane, predstavlja slanje podataka od jednog korisnika ka podskupu svih korisnika kao grupni koncept.

Multikast uključuje sledeće tipove saobraćaja [4]:

- Prenos od jednog izvorišta do pojedinih odredišta (*one to many*) – što podrazumeva slanje obaveštenja grupi pretplatnika poput najnovijih vesti, onlajn predavanja, onlajn prezentacije, stanja na berzi, vremenske prognoze, distribucija mrežnog vremena uređajima u mreži i dr.
- Prenos od pojedinih izvorišta do pojedinih odredišta (*many to many*) – gde se podrazumevaju sinhronizacija baza podataka, multimedijalne konferencijske sesije, onlajn predavanja sa mogućnošću aktivnog učestvovanja učesnika pa i postavljanja pitanja predavaču (učenje na daljinu), onlajn video igrice sa više učesnika (*multiplayer*) itd.
- Prenos od pojedinih izvorišta do jednog odredišta (*many to one*) – je tip multikasta koji još uvek nije dovoljno dokumentovan i standardizovan. Pod ovim tipom podrazumevaju se aukcije gde ponuđač skuplja ponude, prikupljanje podataka, itd. Karakteristično za ovaj tip multikasta kao najproblematičnijeg sa stanovišta skalabilnosti, je mogućnost zagušenja prijemnika usled prevelike količine saobraćaja.

Multikast komunikacija bi se mogla sprovesti u vidu većeg broja simultanih unicast komunikacija ali takva organizacija može izazvati niz problema. Jedan takav primer, kada predajnik šalje saobraćaj grupi prijemnika, on bi mogao da uspostavi unicast vezu sa svakim od prijemnika i da svakom prijemniku pojedinačno šalje primerak paketa. Prvi razlog zašto ovakav primer nije dobro rešenje jeste neefikasno korišćenje mrežnih resursa. Takva primena unicast sistema brzo bi prouzrokovala zagušenje mreže suvišnim kopijama paketa, a posebno u slučaju u kojima učestvuje velik broj prijemnika sa velikim brojem takvih komunikacija. Drugi problem je ugrožena skalabilnost takve komunikacije sa stanovišta veličine grupe prijemnika usled preopterećenja samog predajnika i njegovog mrežnog interfejsa prevelikim brojem unicast komunikacija i prevelikim brojem paketa koje predajnik mora generisati. Upravo iz ovih razloga, radi sprečavanja navedenih problema koji mogu nastati, koristi se multikast komunikacija.

Iako se u istraživanju multikasta poslednjih godina nije daleko odmaklo, postoji veliko interesovanje za ovu vrstu komunikacije od strane telefonskih kompanija u svrhu pružanja što različitijih sadržaja novih generacija, a koje se zasnivaju na IP umrežavanju [3].

### 2.1. Funkcija i značaj multikasta

IP multikast je tehnologija kojom se omogućava efikasniji saobraćaj prilikom slanja informacija hiljadama korporacijskih i kućnih primalaca. Multikast omogućava ostvarivanje izvornog saobraćaja ka mnogostrukim primaocima sa mnogo manjim opterećenjem izvora i

primaoca, čime je saobraćaj znatno efikasniji u poređenju sa drugim tehnologijama, prilikom slanja informacija hiljadama korporacijskih i kućnih primalaca.

Za funkcionisanje multikasta neophodna su dva mehanizma – mehanizam prijavljivanja/odjavljivanja prijemnika na multikast sesiju i mehanizam komunikacije rutera u mreži kako bi se kreiralo multikast stablo prosleđivanja. Mehanizam prijavljivanja/odjavljivanja treba da omogući članovima multikast grupe dinamički pristup (prijavu) a isto tako i napuštanje multikast grupe. Pri tome je za neke primene vrlo značajna brzina prijave i odjave, pa se i o tome mora voditi računa. Multikast stablo prosleđivanja predstavlja stablo sačinjeno od mrežnih čvorova i linkova u mreži, kao i predajnika i prijemnika u čijem korenu se nalazi predajnik, a listovi stabla su prijemnici. Paketi koje šalje predajnik putuju kroz mrežu prolazeći multikast stablo prosleđivanja formirano za određenu multikast sesiju. U čvorovima gde se stablo grana formiraju se dodatne kopije paketa koji stižu do svih listova (prijemnika) u stablu. Nakon prijavljivanja svakog novog prijemnika u grupu, multikast stablo se ažurira, što isto važi i za njegovu odjavu.

Značaj multikasta ogleda se kroz prenos multikast paketa do određinih multikast adresa (tzv. multikast grupa) sa istom pouzdanošću kao i regularni unicast paketi; prenos više tokova do iste grupe; omogućavanje članovima multikast grupe dinamičko (po potrebi) pristupanje i napuštanje grupi; podržavanje više grupe hostova bez obzira na lokaciju ili broj članova kao i da pojedinačni hostovi budu članovi jedne ili više grupa; korišćenje jedne grupe adrese za više aplikacija.

## **2.2. Problemi koju mogu nastati primenom multikasta**

Multikast ima i određene probleme prilikom implementacije. Jedan od problema je konstruisanje optimalnog multikast stabla prosleđivanja. Postoje dve varijante multikast stabla. Kod prve varijante koristi se najkraća putanja do svakog od prijemnika, a kod druge varijante se optimalnije koriste linkovi u mreži, odnosno, manji broj linkova je opterećen paketima koje šalje izvor. U slučaju da je cilj smanjivanje opterećenja mreže, koristi se druga varijanta. Pored optimalnog iskorišćenja propusnog opsega mreže, i kašnjenje se može koristiti kao drugi kriterijum prilikom konstruisanja multikast stabla. Konstrukcija optimalnog multikast stabla je problem koji se aktivno istražuje, u cilju pronalaženje što efikasnijeg i kvalitetnijeg algoritma za optimalno kreiranje multikast stabla sa velikim brojem listova.

Drugi značajan problem je multikast prosleđivanje u samim mrežnim čvorovima tj. ruterima u slučaju IP mreža. Kada paket stigne u ruter koji mora da umnoži paket u više primeraka i prosledi ga na odgovarajuće izlazne portove (u skladu sa multikast stablom kome ruter pripada), ovi ruteri formiraju odgovarajući broj primeraka paketa i prosleđuju, odnosno komutiraju ih kao unicast pakete jer je veliki broj komutatora predviđen za unicast komunikaciju i ne podržava multikast komunikaciju. Usled toga značajno se povećava opterećenje komutatora što može da dovede do zagušenja rutera pa i time do nepotrebnog odbacivanja paketa. Čak i ruteri čiji komutatori podržavaju multikast pakete i njihovo komutiranje imaju često problem u efikasnom raspoređivanju multikast paketa za komutaciju (na primer, algoritmi raspoređivanja u komutatorima sa baferima na ulazu). Pošto su multikast paketi u generalnom slučaju namenjeni većem broju izlaznih portova, dolazi i do problema u strukturi tabela usmeravanja čiji zapisi postaju veći jer moraju da imaju mogućnost mapiranja većeg broja izlaznih portova. Svi navedeni problemi su i dalje aktuelni i veliki broj istraživanja je posvećen njihovom rešavanju [4].

Postoji i niz drugih problema multikast saobraćaja kao što je kreiranje multikast stabala koja se protežu na više autonomnih sistema, pouzdanog prenosa, kvaliteta servisa i sl. U tom slučaju neophodno je uspostaviti mehanizme dogovora između provajdera koji će omogućiti kreiranje takvih multikast stabala koji su potrebni za dobijanje globalne pokrivenosti, ali to još uvek nije postignuto. U cilju rešavanja ovakvih problema kreiran je protokol Multikast proširenje BGP protokola (MBGP). IP multikast koristi UDP za transport a posledice njegove primene su: nedostatak pouzdanog prenosa, nedostatak kontrole zagušenja, opasnost od dobijanja paketa u pogrešnom redosledu kao i opasnost od dobijanja duplikata. Razvijeno je nekoliko protokola kojima je moguće rešiti problem pouzdanog multikast prenosa u vidu omogućavanja mehanizama retransmisije paketa. Protokoli koji su predloženi su PGM (*Pragmatic General Multicast*) definisan u RFC 3208, NORM (*NACK Oriented Reliable Multicast*) definisan u RFC 5401 i RFC 5748, FLUTE (*File Delivery over Unidirectional Transport*) definisan u RFC 6726. U slučaju implementacije i kvaliteta servisa mogu se koristiti protokoli koji omogućavaju podršku za kvalitet servisa poput RSVP protokola [4].

### 3. OPIS APLIKACIJE

U okviru rada, simuliran je multikast saobraćaj u programskom jeziku *MATLAB*. Simulira se ruter sa ulazno-izlaznim portovima, dolazak paketa na ulaze njegovih portova, njihovo baferovanje i rutiranje kroz svič ka izlazima portova. Ulazne parametre simulacije čine broj ulazno-izlaznih portova ( $N$ ), maksimalan broj portova na čije izlaze je upućen paket ( $N_{max}$ ), verovatnoća pojavljivanja paketa na ulazu portova ( $P$ ) i ubrzanje simulacije ( $U$ ). Izlazi simulacije grafički su predstavljani preko ponuđenog, ostvarenog i ulaznog protoka po portovima sviča. Svaka pojedinačna simulacija po zasebnom sviču (16x16, 32x32 i 64x64) ispituje se za 100 000 vremenskih slotova.

U svakom slotu, upoređivanjem slučajne vrednosti sa vrednošću  $P$ , simulira se pristizanje novih paketa, a nakon toga se određuju portovi na koje je paket upućen. Ovo se obavlja za sve portove sviča, a nakon toga se baferuju svi paketi koji pristignu u slot. Posle toga, podaci u baferu se obrađuju i odabiraju paketi koji će se poslati ka odgovarajućim izlazima. Paketi se brišu iz bafera, a ustanovljena statistika ostaje sačuvana.

Paket predstavlja vektor veličine  $N$ , čije kolone sačinjavaju svaki od  $N$  portova svičeva. Ukoliko je paket upućen na izlaz nekog porta, tada će u koloni koja predstavlja taj port biti upisan broj 1, a na svim ostalim pozicijama biće upisana 0. Svi paketi jednog slota nalaze se u matrici „matrica portova“ dimenzija  $N*N$ , čiji red predstavlja port na čiji je ulaz pristigao paket, a kolone predstavljaju portove za čije je izlaze taj paket namenjen.

Struktura bafer je trodimenzionalna matrica veličine  $N*N*(U*broj\ slotova)$ , kojom su predstavljeni baferi na portovima sviča. U svakom slotu se u bafer upisuje celokupna matrica „matrica portova“ na novu poziciju treće dimenzije bafera.

Vektor „vektor slobodnih izlaza“ je dužine  $N$ , i u njemu se beleže izlazi onih portova koji su rutirani u paketu u trenutnom slotu. Ako je već određen paket koji će biti rutiran na izlaz nekog porta, u koloni koja predstavlja taj port biće upisan broj 1. U kolonama portova, čiji su izlazi još slobodni, biće i dalje upisan broj 0. Vektori „matrica ponuđen protok“, „matrica ulazni protok“ i „matrica ostvaren protok“ su veličine  $N$  i sadrže statističke vrednosti protoka podeljenih po portovima.

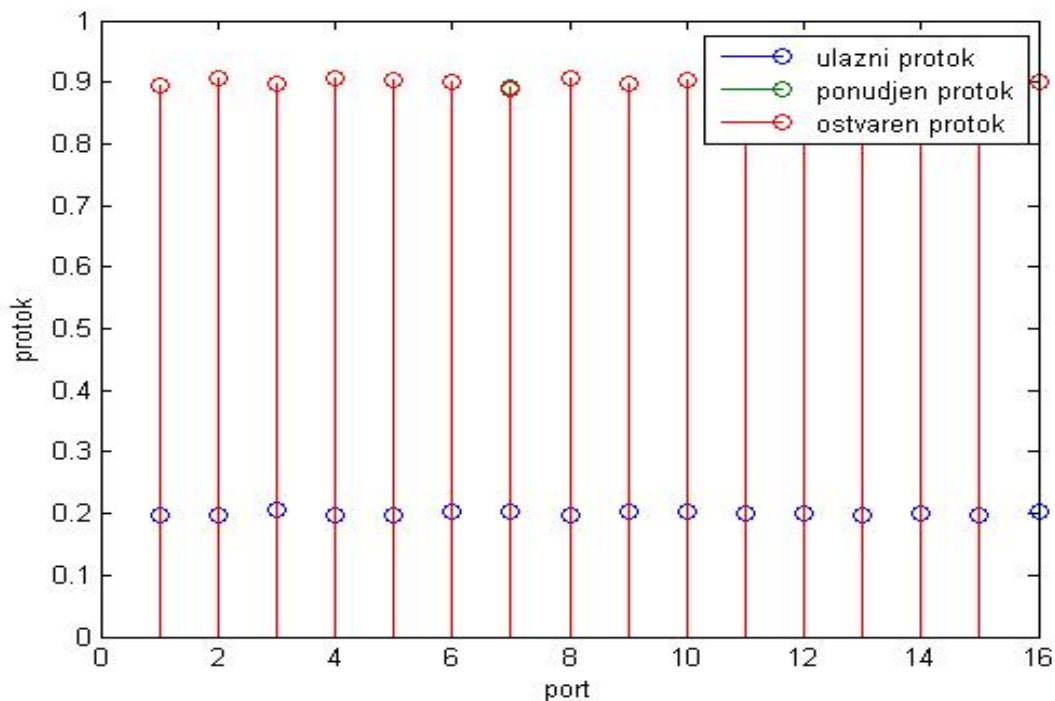
Nakon generisanja i baferisanja paketa, u svakom slotu se pristupa rutiranju. Za svaki od portova, redom se prolazi kroz pozicije u baferu koje predstavljaju pakete baferovane na ulazu tog porta. Pronalazi se prvi paket čije su sve zahtevane destinacije slobodne. Taj paket se briše iz bafera tako što se na svim njegovim pozicijama upisuje broj 0. U slučaju nenalaženja takvog paketa, bira se paket čiji je najveći broj zahtevanih portova slobodan, i u baferu se postavljaju nule na mesta portova na koje je paket uspešno poslat, dok jedinice ostaju na mestima portova na koje paket nije poslat, kako bi bio poslat u nekom od narednih slotova. Pošto se obrade baferi svih portova, i „vektor slobodnih izlaza“ postane pun jedinica, unutar „matrica ostvaren protok“ inkrementiraju se brojevi na pozicijama na kojima se u „vektor slobodnih izlaza“ nalaze jedinice, tj. na koje su rutirani paketi.

Ubrzanje ( $U$ ) predstavlja broj koji pokazuje koliko puta će biti obavljena funkcija rutiranja tokom tokom jednom slotu.

# 4. REZULTATI SIMULACIJE

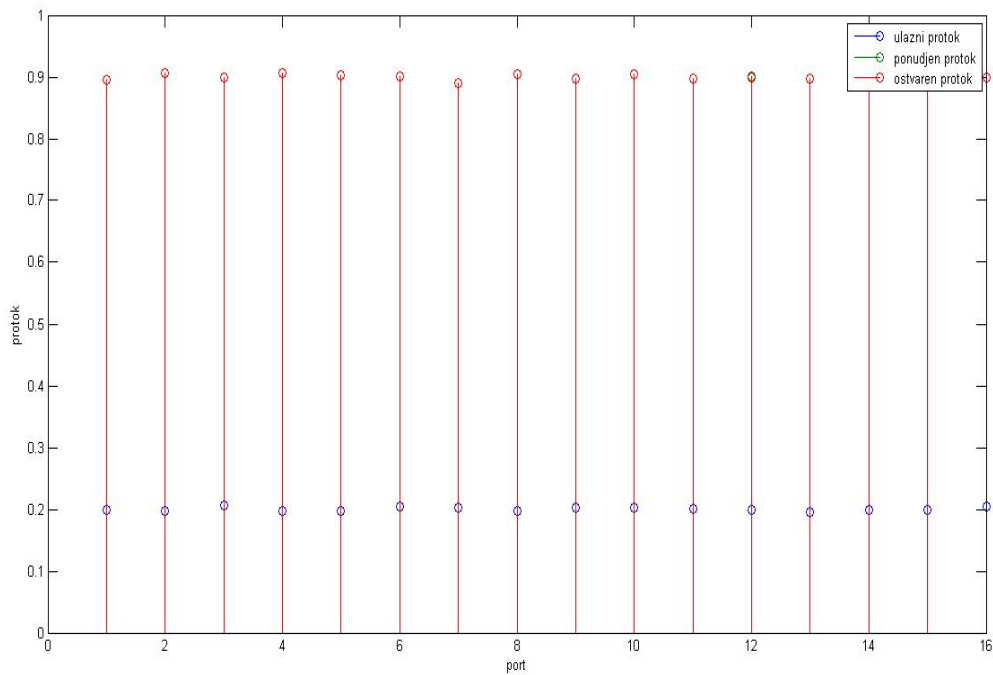
## 4.1. Grafičko predstavljanje rezultata simulacije

U ovom radu su prikazani rezultati dobijeni uporednom analizom rezultata simulacije koja je bazirana na FIFO (*First In - First Out*) baferu i simulacije bazirane na VOQ (*Virtual Output Queing*) baferu. Simulacije su rađene za parametre koji obezbeđuju ponuđen protok za vrednosti manje ili približne vrednosti 1. Znači, za svaki izlazni port, u skoro svakom vremenskom slotu, na neki ulazni port je stigao paket namenjen tom izlaznom portu, i prebačen je u bafer.

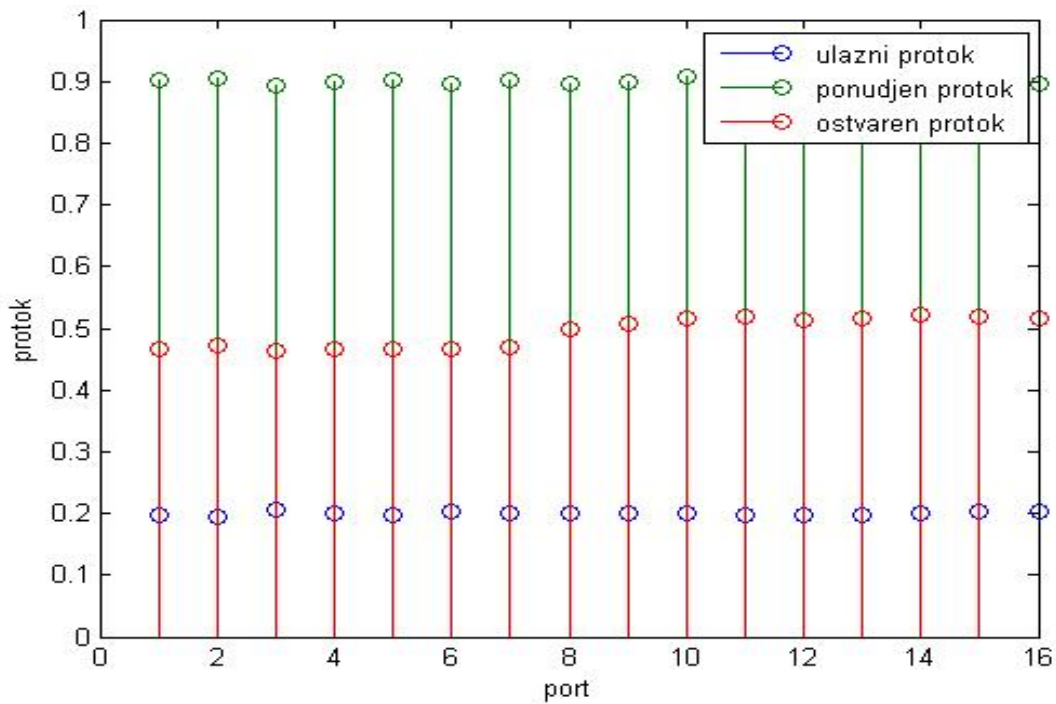


**Slika 4.1.1. 16 ulazno-izlaznih portova, maksimalno 8 zahtevanih izlaznih portova po paketu i opterećenje 0,2 na ulaznim portovima (za  $U = 1$ ), VOQ**

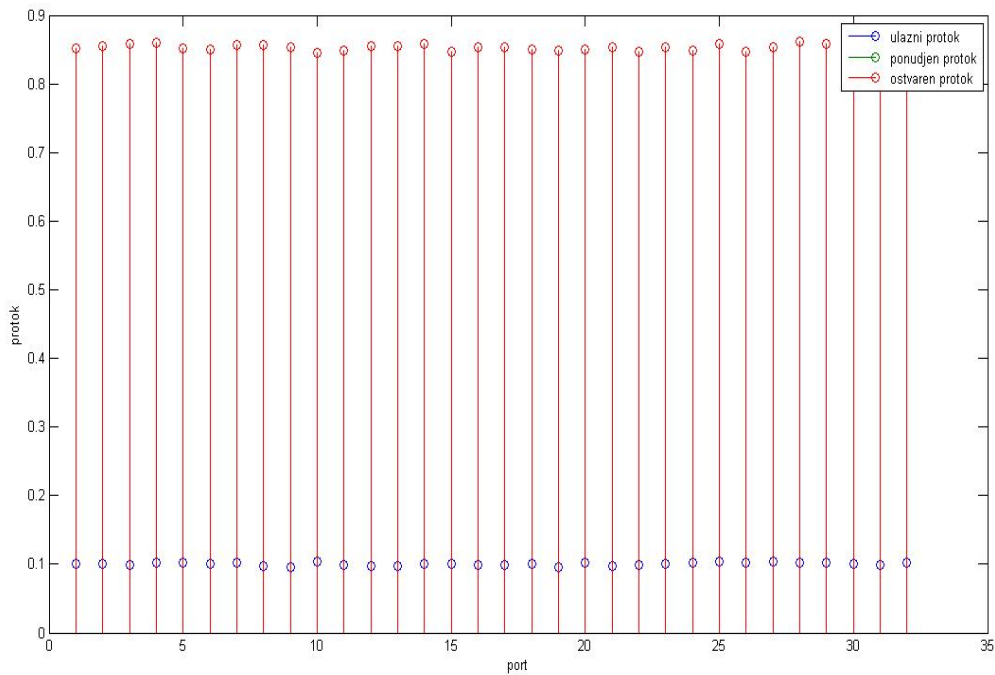




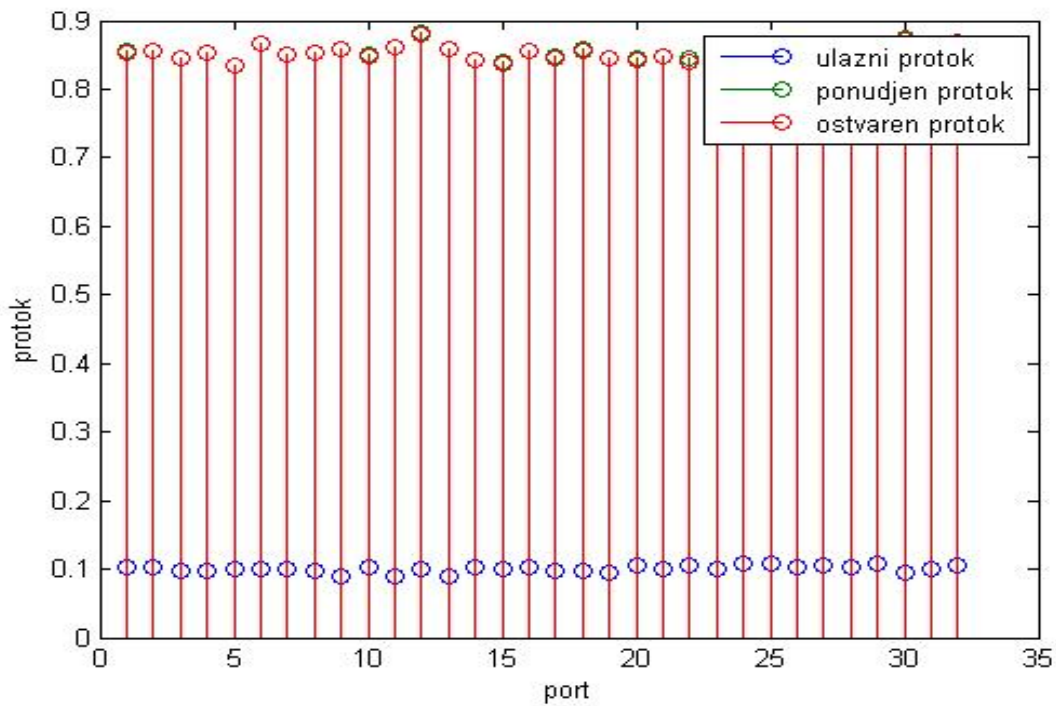
**Slika 4.1.2. 16 ulazno-izlaznih portova, maksimalno 8 zahtevanih izlaznih portova po paketu i opterećenje 0,2 na ulaznim portovima (za  $U = 2$ ), VOQ**



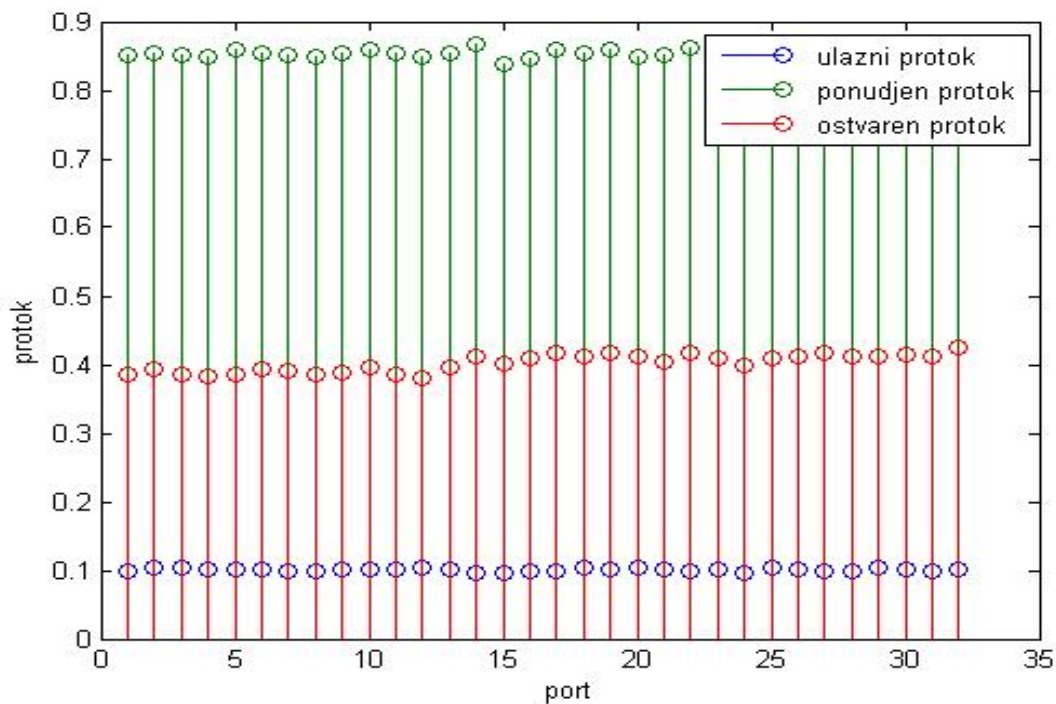
**Slika 4.1.3. 16 ulazno-izlaznih portova, maksimalno 8 zahtevanih izlaznih portova po paketu i opterećenje 0,2 na ulaznim portovima, FIFO**



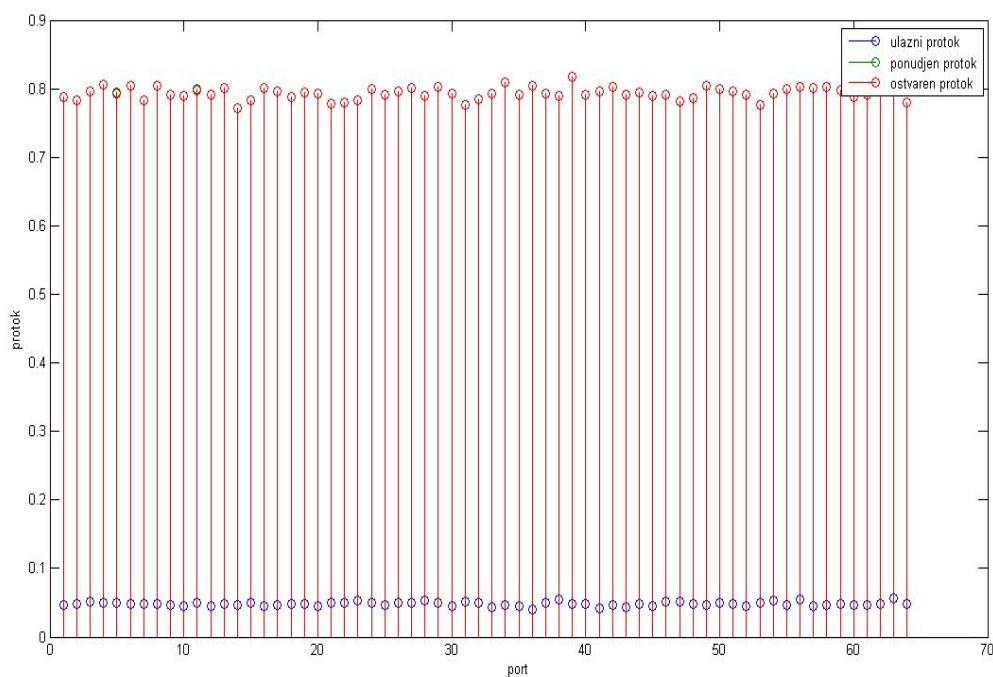
**Slika 4.1.4. 32 ulazno-izlaznih portova, maksimalno 16 zahtevanih izlaznih portova po paketu i opterećenje 0,1 na ulaznim portovima (za  $U = 1$ ), VOQ**



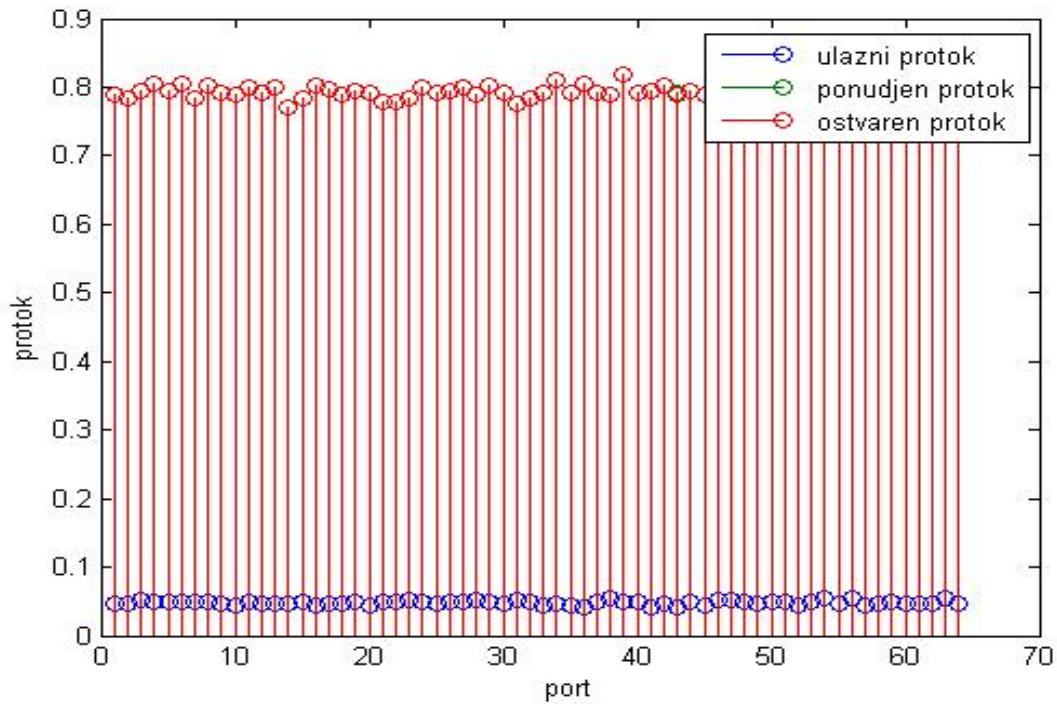
**Slika 4.1.5. 32 ulazno-izlaznih portova, maksimalno 16 zahtevanih izlaznih portova po paketu i opterećenje 0,1 na ulaznim portovima (za  $U = 2$ ), VOQ**



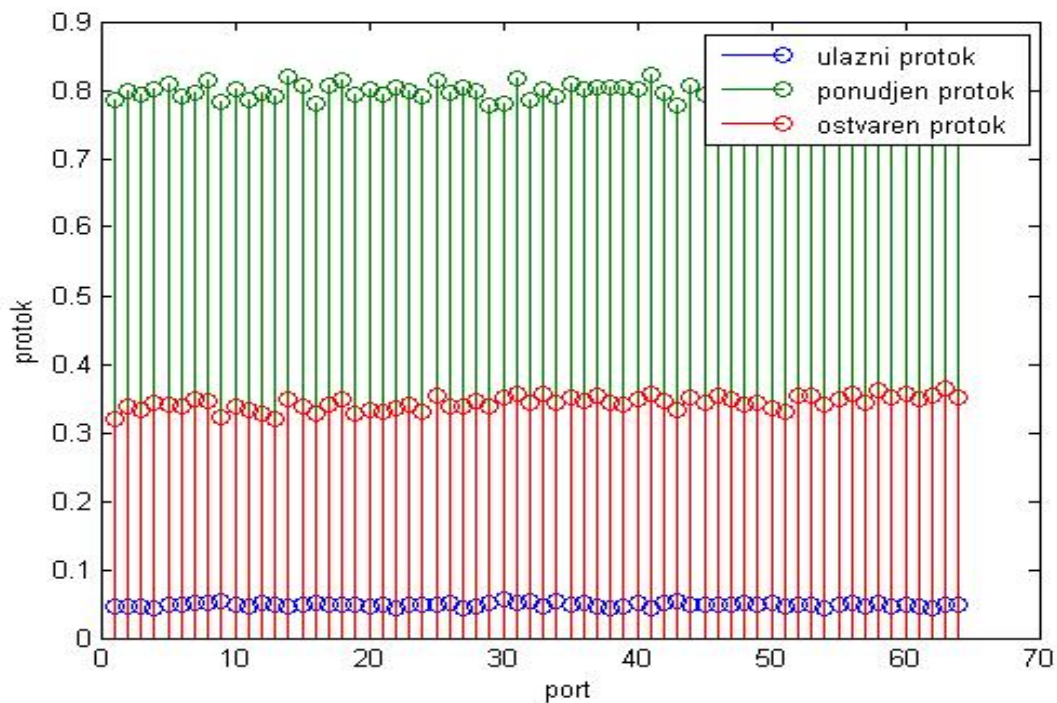
**Slika 4.1.6. 32 ulazno-izlaznih portova, maksimalno 16 zahtevanih izlaznih portova po paketu i opterećenje 0,1 na ulaznim portovima, FIFO**



**Slika 4.1.7. 64 ulazno-izlaznih portova, maksimalno 32 zahtevanih izlaznih portova po paketu i opterećenje 0,048 na ulaznim portovima (za  $U = 1$ ), VOQ**



**Slika 4.1.8.** 64 ulazno-izlaznih portova, maksimalno 32 zahtevanih izlaznih portova po paketu i opterećenje 0,048 na ulaznim portovima (za  $U = 2$ ), VOQ



**Slika 4.1.9.** 64 ulazno-izlaznih portova, maksimalno 32 zahtevanih izlaznih portova po paketu i opterećenje 0,048 na ulaznim portovima, FIFO

## 4.2. Analiza rezultata simulacije

Analizom dobijenih rezultata prikazanih na slikama 4.1.1. – 4.1.9., može se primetiti kako je u slučajevima korišćenja FIFO pristupa ostvareni protok manji od polovine ponuđenog. Ovo je očekivano, imajući u vidu da kod FIFO pristupa problem HOL (*Head of Line*) blokade obara ostvarenu propusnost. Sa druge strane, korišćenjem VOQ pristupa ne nastaje problem HOL blokade, dok parcijalno slanje paketa optimizuje iskorišćenost, jer se izbegava da se sa ulaza ništa ne pošalje kada nijedan paket ne može u potpunosti da pošalje. Analizom rezultata pokazano je da kombinacija VOQ pristupa i parcijalnog slanja paketa obezbeđuje maksimalni kapacitet rutiranja od po jednog paketa, na svakom izlaznom portu, u svakom pojedinačnom slotu.

Ubrzanje  $U$  obezbeđuje mogućnost slanja do  $U$  paketa iz bafera svakog od ulaznih portova ka izlaznim portovima. S obzirom da je kombinacija pretrage celokupnog bafera i delimičnog slanja osigurala maksimalan mogući protok, odnosno da je za  $U = 1$  postignuta maksimalna propusnost, u ovom slučaju povećanje ubrzanja (za  $U = 2$ ) nije uticalo na ostvareni protok, ali bi u nekom opštem slučaju povećanje moglo da poveća propusnost.

Analizom ispitivanog algoritma raspoređivanja paketa za slanje pokazano je da je algoritam veoma dobar i efikasan za uniformni saobraćaj sa srednjom veličinom maksimuma *fanout*-a (broj izlaza kojima je namenjen multikast paket). Međutim, neophodna su ispitivanja i za druge vrste saobraćaja kako bi se postigla kompletna ocena kvaliteta raspoređivača i kako bi se utvrdilo za koje tipove saobraćaja ispitivani algoritam daje slabe rezultate.

## 5. ZAKLJUČAK

IP Multikast omogućava efikasniji saobraćaj pri slanju sa jednog izvora ka više odredišta. Njegova efikasnost veoma zavisi od načina rutiranja, što je i pokazano rezultatima simulacije dobijenim u ovom radu.

Međutim, svaki pristup ima svoja ograničenja, i njegova primena zavisi od slučaja do slučaja. Iako se VOQ pristup pokazao veoma efikasnijim od FIFO pristupa, VOQ pristup sa parcijalnim slanjem paketa je, recimo, nemoguće koristiti u slučaju saobraćaja za koji je potrebno rutiranje paketa u redosledu u kakvom su i stigli u bafer, kao što je *real time* saobraćaj koji se koristi za govornu ili video i govornu komunikaciju. Za takav saobraćaj FIFO pristup ispunjava zahtevane kriterijume. VOQ pristup je pogodan za podatke kojima nije važan poredak niti vreme stizanja paketa. Parcijalno slanje paketa dovodi do situacije u kojoj neće svako od odredišta dobiti pakete istovremeno, niti po istom redosledu. To, takođe, dovodi do problema pri opsluživanju *real time* aplikacija, kao što su govor i video *streaming*. Korišćenjem kombinacije VOQ pristup i delimičnog slanja bilo bi nemoguće opslužiti konferencijski razgovor, govorni ili govorni i video *streaming*, dok je situacija u kojoj bi bila pogodna, npr., primanje *torrent* paketa. Rezultati dobijeni analizom algoritma u ovom radu pokazali su da kombinacija VOQ pristupa i parcijalnog slanja paketa obezbeđuje maksimalni kapacitet rutiranja od po jednog paketa, na svakom izlaznom portu, u svakom pojedinačnom slotu.

Analizom ispitivanog algoritma raspoređivanja paketa za slanje pokazano je da je algoritam veoma dobar i efikasan za uniformni saobraćaj sa srednjom veličinom maksimuma *fanout*-a. Nesumnjivo je da su neophodna ispitivanja i za druge vrste saobraćaja, za postizanje kompletne ocene kvaliteta raspoređivača i utvrđivanje za koje tipove saobraćaja ispitivani algoritam daje slabije rezultate, kako bi se izveo sveobuhvatniji zaključak o podobnosti i efikasnosti ispitivanog algoritma.

## LITERATURA

- [1] D. Makofske, K. Almeroth, *Multicast Sockets Practical Guide for Programmers*, The Practical Guide Series, Elsevier Science, USA, 2003.
- [2] J. Kim, „Revisiting IP Multicast“, *Proc. of "Internet Routing"*, Technische Universitaet Berlin, WS 2006/2007, January 17, 2008.
- [3] D. Minoli, *IP Multicast with applications to IPTV nad mobile DVB-H*, Wiley Interscience, A John Wiley & Sons, Inc., 2008.
- [4] Z. Čiča, Širokopoljasne telekomunikacione mreže, Poglavlje 5 [Online]. Available: [http://telekomunikacije.etf.rs/predmeti/te4ks/docs/STM/STM\\_05.pdf](http://telekomunikacije.etf.rs/predmeti/te4ks/docs/STM/STM_05.pdf)

# A. KOD SIMULACIJE REALIZOVAN U MATLAB-U

```
clc, clear all, close all

Niz = inputdlg('Unesite broj portova rutera'); %broj portova rutera
Podaci=str2num(Niz{1});
N = Podaci(1);

Niz = inputdlg('Unesite maksimalan broj izlaza kojima paket moze biti
namenjen'); %maksimalan broj izlaza kojima paket moze biti namenjen
Podaci = str2num(Niz{1});
Nmax = Podaci(1);

Niz = inputdlg('Unesite verovatnocu pojavljivanja paketa na ulazu');
%verovatnoca pojavljivanja paketa na ulazu
Podaci = str2num(Niz{1});
P = Podaci(1);

Niz = inputdlg('Unesite ubrzanje'); %verovatnoca pojavljivanja paketa na
ulazu
Podaci = str2num(Niz{1});
U = Podaci(1);

slotovi = 100000; %broj slotova simulacije

matrica_ulazni_protok = zeros(N,1); %matrica protoka paketa na ulazu
matrica_ponudjen_protok = zeros(N,1); %matrica ponudjenog protoka paketa na
izlazu
matrica_ostvaren_protok = zeros(N,1); %matrica ostvarenog protoka paketa

brojac = 0;

bafer = zeros(N,N,U*slotovi); %trodimenzionalni niz dimenzija N/N/slotovi,
kome se u svakom slotu na poziciji :,cnt_slot upisuje matrica_portova
%matrica_portovi predstavlja matricu dimenzija N,N u kojoj se na pozicijama
%N,1,2,3...N nalaze sve nule ako na ulazni port u tom slotu nije stigao
%paket, a jedinice ili nule u slucaju ako je stigao. Ako je paket pristigao
%na ulazni port, recimo, 1, a namenjen je izlaznim portovima 2 i 3, dok ceo
%ruter ima 4 porta, sadrzaj matrice bi za prvi red izgledao ovako 0 1 1 0
%svaka vrsta trodimenzionalnog niza bafer predstavlja bafer za neki port,
%tako je vrsta (1,:,1) prva pozicija bafera za prvi port, (2,:,1) je prva
%pozicija bafera za drugi port, (1,:,2) je druga pozicija bafera za prvi
%port, i tako dalje. U baferu se nalaze paketi koji su predstavljeni nizom
%nula i jedinica, u kome jedinice predstavljaju portove za koje su
%namenjeni. Kad je paket poslat na izlaze, brise se iz bafera tako sto se
%u vrstu u kojoj se nalazio taj paket upisuju nule. Bafer prakticno pocinje
%od prve vrste u kojoj se nalaze neke vrednsti, sto je uradjeno zbog
%prakticnosti resenja
for cnt_slot = 1:slotovi %simuliranje za svaki slot
    brojac = brojac + 1;
    clc
    display(brojac/slotovi*100,'complete');
    matrica_portova = zeros(N,N); %matrica u kojoj se belezi kom izlazu je
koji port uputio paket za trenutni slot
    for cnt_port = 1:N %simuliranje za svaki port
        pom_port = rand(1); %odlucivanje da li je paket stigao na ulazni
port
```



```

    if pom_port <= P
        br_izlaza = randi(Nmax); %broj izlaza na koje se paket upucuje
        if br_izlaza == N %za slucaj da je broj izlaza jednak broju
portova, smanjuje se za jedan jer je jedan port ulazni
            br_izlaza = br_izlaza - 1;
        end

        if cnt_port == N
            izlazi = randperm(N-1,br_izlaza); %za slucaj da je paket
stigao na poslednji port, bira se br_izlaza izlaza od 1 do N-1
        else
            izlazi = randperm(N,br_izlaza+1); %za slucaj da paket nije
stigao na poslednji port, bira se br_izlaza + 1 izlaza za paket
            if ismember(cnt_port,izlazi) %u slucaju da je za izlazni
port izabran port na koji je pristigao paket, mesto njega uzima se za
izlazni port ovaj dodatni port koji je izabran za br_izlaza + 1
                p = find(izlazi == cnt_port); %u slucaju da nije za
izlazni port izabran port na koji je pristigao paket, br_izlaza + 1 port se
ne racuna

                izlazi(p)= izlazi(length(izlazi)); %posto u matlabu
permutacije je moguće odraditi samo na intervalu 1 do N, u slucaju kad
zelim izbaciti ulazni port iz permutacija, moram birati jedan port viska,
za slucaj da je za izlazni izabran bas ulazni port
            end
        end
        for cnt_izlazi = 1:br_izlaza
            matrica_portova(cnt_port,izlazi(cnt_izlazi)) = 1; %za gore
izabran broj izlaza se odredjuje na koje izlaze se paket salje
        end
    end
end

bafer(:, :, cnt_slot) = matrica_portova; %trodimenzionalni niz kome treca
vektor_slobodnih_izlaza = zeros(1,N); %niz koji predstavlja slobodne
portove, oznacene nulom, i zauzete, oznacene jedinicom
for cnt_port = 1:N
    for pom_U = 1:U
        pom_bafer = sum(bafer,2); %sabiraju se svi clanovi u vrstama
bafera, kako bi se odredilo na kom ulaznom portu postoje paketi (vrednost
ce biti veca od 0)
        %pom_bafer je matrica od tri dimenzije i to N/1,slotovi
        [x,y,z] = find(pom_bafer(cnt_port,1,:)); %traze se pozicija na
kojima se nalaze nenulte vrednosti, i bira se prva od njih jer ona
predstavlja prvo mesto u baferu tog porta
        %x je broj vrste u kojoj se nalazi nenulta vrednost, y je
vrednost
        %trece dimenzije na kojoj se nalazi nenulta vrednost, a z je
%vrednost te nenulte vrednosti
        if sum(vektor_slobodnih_izlaza) ==
length(vektor_slobodnih_izlaza)
            break
        end

        pom = 0; %pomocna promenljiva u kojoj se cuva najveći broj
slobodnih izlaza
        pom_poz = 0; %pomocna promenljiva u kojoj se cuva port s
najvecim brojem slobodnih izlaza
        for cnt = 1:length(y)
            a = find(bafer(cnt_port,:,y(cnt))); %predstavlja portove na
koje je poslat taj paket iz bafera
            pom_z = 0;

```

```

        pom_a_pr = 0;
        for cnt_pom = 1:length(a)
            if vektor_slobodnih_izlaza(1,a(1,cnt_pom)) ~= 1
%isptivanje da li su svi izlazi za koje je paket namenjen slobodni
                pom_z = pom_z + 1; %ovaj brojac se inkrementira
                svaki put kad je izlaz slobodan
                pom_a_pr(pom_z) = a(cnt_pom);
            end
        end

        if pom_z == z(cnt) %u slucaju kad je nadjen paket s
maksimalnim poklapanjem, zaustavlja se pretragav
            pom_poz = y(cnt);
            pom_a = pom_a_pr;
            break
        end
        if pom_z > pom %u slucaju ako paket ima vece poklapanje od
prethodnog paketa s najvećim poklapanjem, pamti se nov paket
            pom = pom_z;
            pom_poz = y(cnt);
            pom_a = 0;
            pom_a = pom_a_pr;
        end
    end
end

if isempty(x) == 0 && pom_poz ~= 0
    for cnt = 1:length(pom_a)
        vektor_slobodnih_izlaza(1,pom_a(cnt)) = 1; %upisuju se
jedinice u vektor slobodnih izlaza sto znaci da su ti izlazi zauzeti
        bafer(cnt_port,pom_a(cnt),pom_poz) = 0; %upisuju se
nule u bafer, sto znaci da je paket s te pozicije poslat
        matrica_ostvaren_protok(pom_a(cnt),1) =
matrica_ostvaren_protok(pom_a(cnt),1) + 1/slotovi;
    end
end
end
end

paketi_uk = sum(matrica_portova,2); %u svakom slotu se odredjuje koliko
paketa je poslato ka izlazima sa ulaznog porta
paketi_uk_izlaz = sum(matrica_portova,1); %u svakom slotu se odredjuje
koliko paketa je namenjeno izlazu
for cnt_paketi = 1:N
    if paketi_uk(cnt_paketi,1) ~= 0 %ako je taj broj nula, to znaci da
na ulazni port nije ni stigao paket
        matrica_ulazni_protok(cnt_paketi,1) =
matrica_ulazni_protok(cnt_paketi,1) + 1/slotovi; %ukupan broj paketa koji
su stigli na ulazne portove
    end
    matrica_ponudjen_protok(cnt_paketi,1) =
matrica_ponudjen_protok(cnt_paketi,1) +
paketi_uk_izlaz(1,cnt_paketi)/slotovi; %ukupan broj paketa koji su poslali
na izlaze
end
end

%graficko predstavljanje rezultata
figure

```

```
stem([matrica_ulazni_protok matrica_ponudjen_protok
matrica_ostvaren_protok])
xlabel('port')
ylabel('protok')
    legend('ulazni protok', 'ponudjen protok', 'ostvaren protok')
```