

**ELEKTROTEHNIČKI FAKULTET UNIVERZITETA U BEOGRADU**



**ANALIZA MULTIKAST KONTROLERA ZA MEŠANI MULTIKAST I  
UNIKAST SAOBRAĆAJ**

– Diplomski rad –

Kandidat:

Olga Konstantinov 128/2011

Mentor:

doc. dr Zoran Čiča

Beograd, Oktobar 2015.

# SADRŽAJ

SADRŽAJ.....	2
1. UVOD.....	3
2. IP MULTIKAST.....	4
2.1 OSNOVNI PODACI O IP MULTIKASTU.....	4
2.2. PROBLEMI U IP MULTIKASTU.....	5
2.3. PROTOKOLI IP MULTIKASTA.....	5
2.3.1 Protokoli za prijavu/odjavu na/sa sesije.....	6
2.3.2 Protokoli rutiranje.....	6
3. OPIS APLIKACIJE.....	8
4. REZULTATI SIMULACIJE.....	10
5. ZAKLJUČAK.....	18
LITERATURA.....	20
PRILOG - PROGRAMSKI KOD.....	21

# 1. UVOD

Tradicionalan način komunikacije predstavlja komunikaciju između dva pojedinačna korisnika - **unicast** ili od jednog korisnika ka svim ostalim - **broadcast**. Takođe u početku se komunikacija obavljala tehnikom komutacije kola.

Komutacija kola je predstavljala zauzimanje jedne unapred definisane putanje za potrebe jedne veze. Zaključeno je da u toku jedne veze postoje neiskorišćeni resursi usled pauza u komunikaciji. Zato je uvedena nova tehnologija - komutacija paketa. Komutacija paketa je tehnika zasnovana na principu da se pri komunikaciji korisnika informacije šalju u vidu paketa odnosno podaci koji se šalju od pošiljaoca ka primaocu se pakuju u pakete i prosleđuju dalje. Paketi kroz mrežu putuju tako što preko linkova dolaze do mrežnih čvorova tj. komutatora i zatim prolaze kroz njih. Uloga komutatora je da na ulazu odredi na koji će izlaz paket biti prosleđen.

**IP (internet protocol) multikast** predstavlja nov koncept slanja podataka. U ovom slučaju podaci se šalju od jednog korisnika ka grupi drugih. Korisnici kojima se šalju podaci predstavljaju podgrupu svih korisnika.

Multikast se najčešće primenjuje u multimedijalnim aplikacijama u realnom vremenu kao što su video konferencije, IP/TV, video po zahtevu, učenje na daljinu, javne konferencije i predavanja, razne vrste onlajn učenja, studentska internet televizija ili radio su samo neke od mogućnosti. Broj multikast aplikacija postaje sve veći i time udeo multikast saobraćaja u mreži raste i potrebni su mrežni čvorovi koji efikasno procesiraju i prosleđuju multikast saobraćaj.

U okviru ove teze posmatraćemo komutaciju mešanog multikast i unicast paketskog saobraćaja i posmatrati kako njihov odnos utiče na efikasnost komutatora ako se koristi *greedy* raspoređivač.

U drugom poglavlju teze biće navedene ukratko osnove multikasta: šta je multikast, koji su osnovni problemi i protokoli sa kojima se susrećemo. Zatim će u trećem poglavlju biti dat opis osnovnog principa rada aplikacije (generalne ideje) i njenih glavnih funkcija. Biće opisane uloge glavnih promenljivih i detaljan opis samih funkcija. U četvrtom poglavlju će biti dati rezultati simulacije u vidu grafika sa komentarima šta iz tih grafika možemo zaključiti. Peto, ujedno i poslednje poglavlje, će biti rezime svega urađenog u tezi i zaključci doneseni prilikom izrade celokupne aplikacije i simulacije.

## 2. IP MULTIKAST

U ovom poglavlju biće date osnove IP multikast tehnologija: šta je multikast tehnologija, koje su osnovne prednosti i mane ovakvog načina komunikacije i koji su osnovni protokoli koji se koriste.

### 2.1 Osnovni podaci o IP multikastu

U današnjim IP mrežama sve je veći udeo multikast saobraćaja. Pod IP multikast saobraćajem se podrazumevaju tri vrste komunikacija:

1. *One to many* - kada jedan predajnik šalje podatke grupi prijemnika. Aplikacije karakteristične za ovaj vid IP multikasta su: slanje mrežnog vremena grupi mrežnih uređaja, onlajn učenje (prezentacije, predavanja itd.), odašiljanje televizijskog signala, radio signala itd.

2. *Many to many* - Kada više predajnika šalje podatke ka više prijemnika. Aplikacije tipične za ovaj vid IP multikasta su: onlajn predavanja sa mogućnošću da učesnici postavljaju pitanja, onlajn igranje sa više igrača itd.

3. *Many to one* - Više predajnika šalje podatke samo jednom prijemniku. Aplikacije tipične za ovaj vid IP multikasta je npr. sakupljanje ponuda ili različitih vrsta podataka. Ono što je posebno bitno naglasiti da je ovaj vid IP multikasta najproblematičniji jer može doći do zagušenja prijemnika u slučaju prevelike količine pristiglih podataka.

IP multikast je tehnologija kojom se omogućuje efikasniji saobraćaj prilikom slanja informacija velikom broju korisnika. Multikast omogućava ostvarivanje ponuđenog saobraćaja (saobraćaja koji je krenuo od pošiljaoca) ka većem broju primaoca sa mnogo manjim opterećenjem izvora nego druge konkurentne tehnologije. Generalno IP multikast saobraćaj bi mogao da bude organizovan kao skup unicast veza. Ovo bi bilo izvedeno tako što bi predajnik sa svakim od prijemnika uspostavio vezu i poslao mu kopiju paketa. Ovako bi predajnik morao da, ako ima npr. 1000 zainteresovanih prijemnika, pošalje 1000 kopija paketa. Takođe, mreža bi morala da prenese ovih 1000 paketa. Primetimo da bi u međuvremenu linkovi koji vode do komutatora gde će se saobraćaj podeliti prenosili 999 kopija koje su nepotrebne. Na ovaj način smo nepotrebno opteretili i predajnik i mrežu. Zato koristimo IP multikast. Prednost ove vrste komunikacije je u tome što predajnik šalje samo jednu kopiju paketa na adresu multikast sesije (grupe) a zatim ga komutatori umnožavaju i prosleđuju tamo gde je potreban. Na ovaj način rasterećujemo predajnik i on ne generiše nepotrebne kopije paketa. Takođe, prijemnici ne komuniciraju direktno sa predajnikom već sa lokalnim mrežnim čvorovima. Prijemnici se i prijavljuju i odjavljuju sa sesije odgovarajućim lokalnim mrežnim čvorovima.

Da bi multikast komunikacija dobro funkcionisala moraju biti definisana dva mehanizma:

1. Prvi je mehanizam za prijavljivanje i odjavljivanje iz grupe. Kod ovog mehanizma u pojedinim slučajevima izuzetno je važno da je kratko vreme odjavljivanja i prijavljivanja jer korisnici ne smeju osetiti kašnjenja u prijemu podataka. Primer za ovo je televizijski program gde korisnik dok menja kanale ne sme imati veliko vreme čekanja.

2. Drugi mehanizam je mehanizam kreiranja stabla prosleđivanja paketa među komutatorima. Stablo predstavlja promenljivu vezu linkova i komutatora. Na krajevima stabla nalaze se predajnik i prijemnik. Stablo se formira za određenu grupu (sesiju) i ažurira se svaki put kada se neki prijemnik odjavi ili prijavi na grupu.

## 2.2. Problemi u IP multikastu

U IP multikastu postoje i određeni problemi implementacije od kojih su dva glavna, ali ne i jedina.

1. Problem konstruisanja optimalnog stabla. Postavlja se pitanje kako konstruisati stablo tako da ono bude optimalno. Bilo bi logično da želimo da paket putuje od predajnika do prijemnika najkraćom putanjom. Ali ovakav način razmišljanja tj. implementacije nije uvek najbolji jer se može desiti da tako paket putuje preko čvorova i linkova koji su dosta opterećeni. Drugi način je da biramo putanju tako da na svakom čvoru saobraćaj približno podjednako raspoređen. Ovo naravno nisu jedini parametri koje uzimamo u obzir jer postoji i kašnjenje, zauzeti propusni opseg itd. Sve ovo uzeto u obzir vodi do zaključka da formiranje optimalnog stabla nije nimalo jednostavno i ta oblast se i dalje istražuje i traga se za mehanizmima koji bi pomogli lakšem formiranju optimalnog stabla.

2. Problem prosleđivanja unutar samog komutatora. Na putu između predajnika i prijemnika paket prolazi kroz veliki broj komutatora. Neki od ovih komutatora su takvi da ne podržavaju multikast saobraćaj već kad paket dođe do njih, oni ga umnože i proslede na izlaze definisane multikast stablom. Ovo samim tim povećava opterećenost komutatora. Takođe, čak i komutatori koji podržavaju multikast nekad imaju problema u prosleđivanju pakete. Naime, multikast saobraćaj i jeste koncipiran tako da imamo veliki broj predajnika i prijemnika, a u današnje vreme kada su protoci veliki radi se o izuzetno velikom broju paketa koji se prosleđuju. Komutatori uglavnom imaju bafere (memorije u kojima se paketi nizu redom kako pristižu i bivaju prosleđeni dalje kada dođu na red) na izlazu ali ti baferi unose dodatno kašnjenje. Ovaj problem se intenzivno rešava jer zahtevi za protocima postaju sve veći a samim tim i performanse komutatora moraju biti sve bolje.

Ovi problemi nisu jedini koji se javljaju u slučaju multikast tehnologija. Javlja se i problem kreiranja pouzdanog i kvalitetnog prenosa kada se veza prostire preko nekoliko nezavisnih sistema. U tom slučaju se mora postići određeni dogovor između provajdera koji učestvuju u formiranju veze i na taj način obezbediti globalnu pokrivenost multikastom.

## 2.3. Protokoli IP multikasta

Kao što je ranije navedeno u tezi, postoje dve glavne komponente da bi multikast tehnologija dobro funkcionisala.

Prva je prijavljivanje na sesiju i odjavljivanje sa iste. Protokoli koji se koriste za to su: u IPv4 mrežama je IGMP (*Internet Group Management Protocol*) pri čemu su u upotrebi tri verzije IGMPv1, IGMPv2 i IGMPv3. MLD (*Multicast Listener Discovery*) protokol se koristi u IPv6 mrežama, pri čemu postoje verzije MLDv1 i MLDv2.

Druga komponenta je kreiranje multikast stabla i protokoli koji se koriste za to su: PIM (*Protocol Independent Multicast*), ali pored njega su definisana i proširenja postojećih unicast protokola rutiranja poput MBGP (*Multicast Border Gateway Protocol*), MOSPF (*Multicast Open*

*Shortest Path First*) i drugih. U nastavku će biti reči o PIM (*Protocol Independent Multicast*) protokolu kao najpoznatijem i najčešće korišćenom multikast protokolu.

### 2.3.1 Protokoli za prijavu/odjavu na/sa sesije

**1. IGMPv2** (*Internet Group Management Protocol*) protokol je definisan u RFC 2236. Princip rada ove vrste protokola je taj da IGMP komutator (ovako nazivamo svaki komutator koji podržava IGMP protokol) periodično šalje upit na sve mreže korisnika za sve sesije za koje on zna. Korisnici šalju svoj odgovor na upit i tako obavestavaju IGMP komutator u kojoj sesiji žele da učestvuju. Na ovaj način IGMP komutator stalno ažurira sesije i tako zna kome treba da šalje podatke. On podatke grupi šalje samo ako ta grupa ima bar jednog aktivnog člana. Kada želi da se odjavi sa sesije korisnik šalje zahtev direktno IGMP komutatoru. Isto tako i kad želi da se prijavi. U slučaju da je na korisničku mrežu povezano više IGMP komutatora, tada će samo jedan slati upite (onaj čija je IP adresa najmanja). IGMPv2 je kompatibilan sa prethodnom (IGMPv1) verzijom. Glavna nadogradnja u odnosu na verziju 1 je ta što korisnici šalju poruke odjave direktno IGMP komutatoru i tako ubrzavaju proces prestanka slanja sadržaja ukoliko to više nije potrebno za tu grupu (u slučaju da je član koji se odjavio bio poslednji aktivni u grupi).

**2. IGMPv3** protokol je definisan u RFC 3376 i RFC 4604. Gotovo je isti kao i njegovi prethodnici sa tom izmenom da ova verzija može da deklarise listu multikast izvorišta od kojih želi da prima sadržaj i listu multikast izvorišta od kojih ne želi da prima sadržaj. Ova verzija je kompatibilna sa prethodnim verzijama ovog protokola.

**3. MLD** protokol je ekvivalent IGMP protokolu u IPv6 mrežama, pri čemu su definisane dve verzije MLDv1 i MLDv2. MLDv1 verzija je definisana u RFC 2710 i implementira funkcionalnosti IGMPv2 protokola samo u IPv6 mrežama, a MLDv2 verzija je definisana u RFC preporukama RFC 3810 i RFC 4604 i implementira funkcionalnosti IGMPv3 protokola samo u IPv6 mrežama. Princip rada MLDv1 protokola je funkcionalno identičan radu IGMPv2 protokola. Razlike postoje samo u tehničkim detaljima zbog razlika između IPv4 i IPv6 mreža.

### 2.3.2 Protokoli rutiranja

Glavni i najčešće korišćeni protokol je **PIM protokol** koji se koristi za kreiranje multikast stabla p o k m će se p osledivati multikast paketi neke grupe. PIM poruke se šalju direktno upakovane u IP datagrame (kao ICMP (*Internet Control Message Protocol*) poruke). PIM poruka se sastoji od PIM zaglavlja i sadržaja PIM poruke. Postoji više varijanti PIM protokola:

1. PIM-SM (*PIM Sparse Mode*) - trenutno najčešći u praksi. PIM-SM varijanta je definisana u RFC 4601. Ova verzija protokola je razvijena sa idejom da je uvek broj zainteresovanih korisnika multikast grupe (aktivnih članova) mnogo manji nego ukupan broj članova pa je odatle i naziv sparse. Ima dobru skalabilnost tj. dobro se prilagođava povećanom protoku saobraćaja.
2. PIM-DM (*PIM Dense Mode*) - varijanta je definisana u RFC 3973. Koristi se kada je najveći deo mreže tj. većina podmreža zainteresovana (aktivna) da primi multikast saobraćaj. Nema dobru skalabilnost pa zato nije pogodan za velike mreže.
3. PIM-SSM (*PIM Source Specific Multicast*) – je varijanta gde je koren stabla samo jedan izvor i na taj način je omogućena sigurnija i skalabilnija veza za ograničen broj aplikacija.

4. bidirekcionni PIM – Formira bidirekcionno stablo. Nikad se ne formira po najkraćoj putanji između predajnika i prijemnika pa zato može da unosi veća kašnjenja nego ostale varijante.

### 3. OPIS APLIKACIJE

U okviru aplikacije, simulirana je kombinacija unikast i multikast saobraćaja u programskom paketu MATLAB za komutator kojega konfigurira raspoređivač koji radi po *greedy* principu. Simulira se komutator sa ulazno-izlaznim portovima, dolazak paketa na ulaze njegovih portova, njihovo skladištenje u bafer i komutacija kroz komutator ka izlazima portova. Izlazi simulacije su grafička predstava ponudnog, ostvarenog i ulaznog protoka po portovima komutatora. Ulazi u simulaciju su:

1. broj ulazno-izlaznih portova  $N$
2. verovatnoća pojavljivanja paketa na ulazu portova  $P$
3. maksimalan broj portova na čije izlaze je upućen paket  $N_{max}$
4. udeo unikast saobraćaja u ukupno generisanom saobraćaju  $P_u$
5. ubrzanje simulacije  $U$

Simulacija se vrši u 100 000 vremenskih slotova kako bi se dobila što bolja statistika iz koje ćemo izvesti zaključke o performansama sistema.

U svakom vremenskom slotu se simulira prispeće novih paketa, upoređivanjem slučajne vrednosti sa  $P$ . Zatim se određuje da li će pristigli paket biti unikast ili multikast tipa, upoređivanjem slučajne vrednosti sa  $P_u$ , kao i određivanje portova na koje je paket upućen. Ovo se obavlja za sve portove komutatora, a zatim se svi paketi pristigli u vremenskom slotu skladište u bafer. Potom se obrađuju podaci u baferu, traže se paketi koji će biti poslani ka željenim izlazima, brišu iz bafera, i čuva se trenutna statistika.

Paket je vektor dužine  $N$ , čije kolone predstavljaju svaki od  $N$  portova komutatora. Ako je paket upućen na izlaz nekog porta, u koloni koja predstavlja taj port je upisan broj 1, na svim ostalim pozicijama je broj 0. Svi paketi jednog slotu se nalaze u matrici *matrica\_portova*, koja je veličine  $N*N$  čiji red predstavlja port na čiji je ulaz prispeo paket, a kolone portove za čije je izlaze taj paket namenjen.

Struktura *bafer* je trodimenzionalna matrica veličine  $N*N*(U*slotovi)$  kojom su predstavljeni baferi na portovima komutatora. U svakom slotu se u bafer upisuje celokupna matrica *matrica\_portova* na novu poziciju treće dimenzije *bafer-a*.

Vektor *vektor\_slobodnih\_izlaza* je dužine  $N$ , i u njemu se beleži na izlaze kojih portova su komutirani paketi u trenutnom slotu. Ako je već određen paket koji će biti prosleđen na izlaz nekog porta, u koloni koja predstavlja taj port će biti broj 1. U kolonama portova, čiji su izlazi još slobodni, biće broj 0.

Vektori *matrica\_ponudjen\_protok*, *matrica\_ulazni\_protok* i *matrica\_ostvaren\_protok* su veličine  $N$  i sadrže statistike protoka po portovima.

U svakom slotu se, po generisanju i skladištenju paketa u bafer, pristupa komutiranju. Po principu FIFO, uzima se prvi paket u baferu, i ispituje da li su mu sve zahtevane destinacije slobodne. Taj paket se briše iz bafera tako što se na svim njegovim pozicijama upisuje broj 0. Pošto se obrade baferi svih portova, ili *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 komutirani paketi.



Ubrzanje  $U$  predstavlja koliko puta će biti obavljena funkcija komutiranja tokom jednog slot.

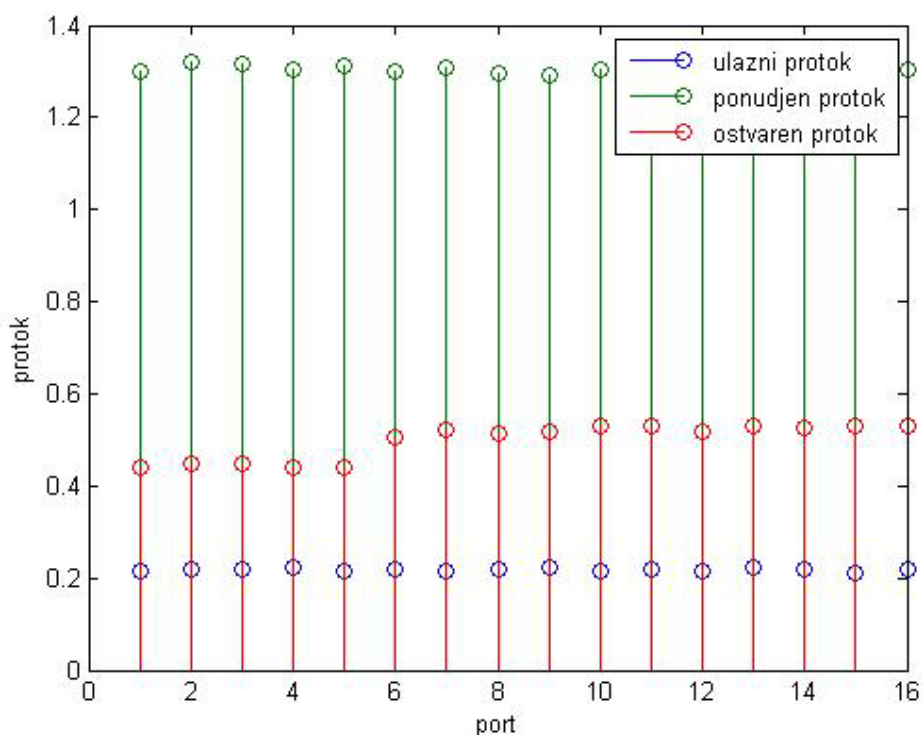
Kompletan MATLAB kod korišćen za simulaciju dat je na kraju teze u poglavlju pod nazivom Prilog-Programski kod.

## 4. REZULTATI SIMULACIJE

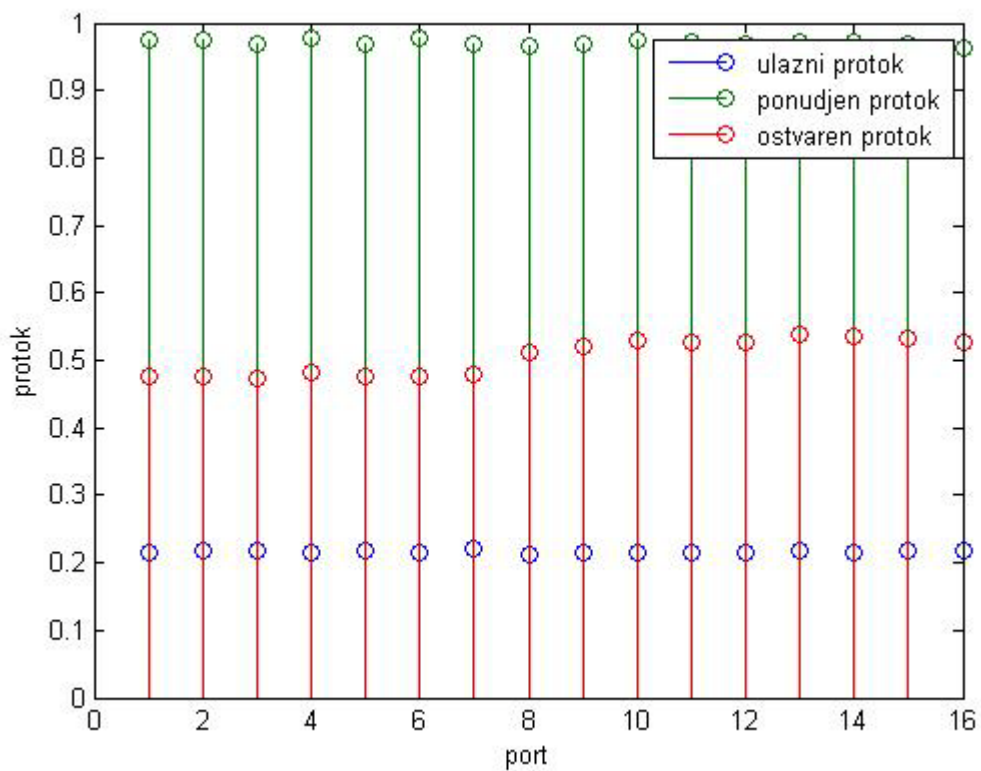
U ovom poglavlju biće prikazani rezultati simulacije i objašnjenja dobijenih rezultata.

Simulacija je rađena za različite odnose multikast i unicast saobraćaja za komutatore sa 16, 32 i 64 ulazno-izlazna porta. Parametri koji su zadati u simulaciji su sledeći (a biće i navedeni za svaku sliku pojedinačno, ispod slike):

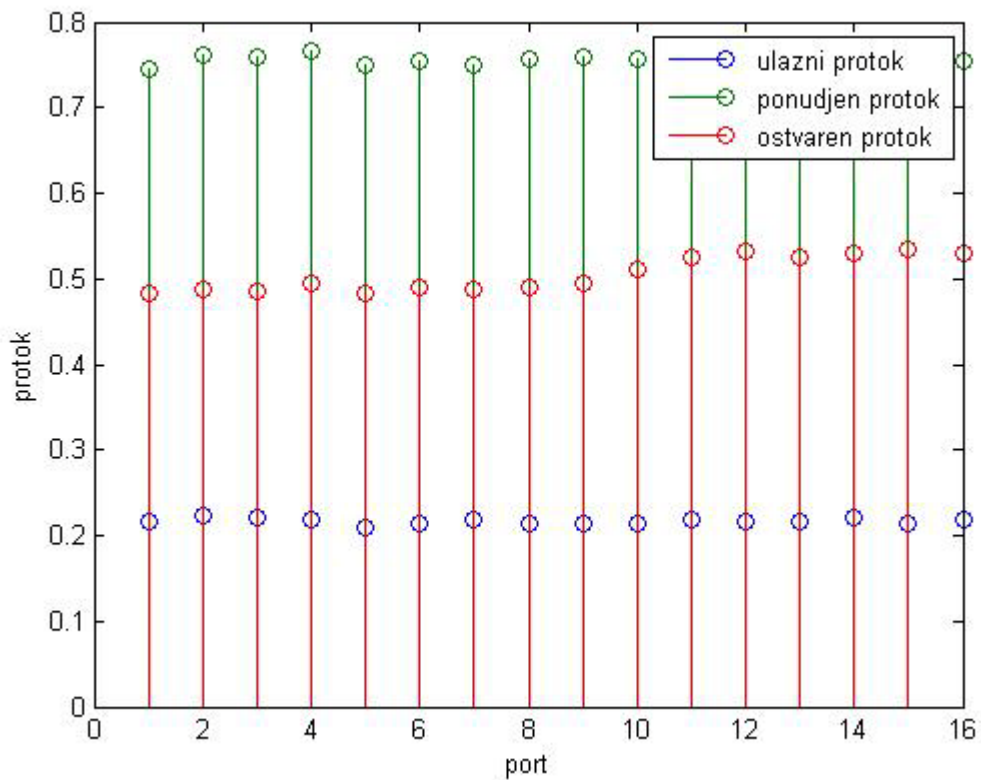
- 16 ulazno-izlaznih portova, maksimalno 10 zahtevanih izlaznih portova po paketu, verovatnoća pojavljivanja paketa na ulaznim portovima 0.217, a udeo unicast saobraćaja u ukupnom ima tri vrednosti: 0, 0.3 i 0.5.
- 32 ulazno-izlazna porta, maksimalno 16 zahtevanih izlaznih portova po paketu, verovatnoća pojavljivanja paketa na ulaznim portovima 0.11, a udeo unicast saobraćaja u ukupnom saobraćaju ima tri vrednosti: 0, 0.22 i 0.5.
- 64 ulazno-izlazna porta, maksimalno 32 zahtevana izlazna porta po paketu, verovatnoća pojavljivanja paketa na ulaznim portovima 0.1, a udeo unicast saobraćaja u ukupnom saobraćaju ima tri vrednosti: 0, 0.3 i 0.54.



Slika 4.1.  $N=16$ ,  $N_{max}=10$ ,  $P=0.217$ ,  $P_u=0$ ,  $U=1$

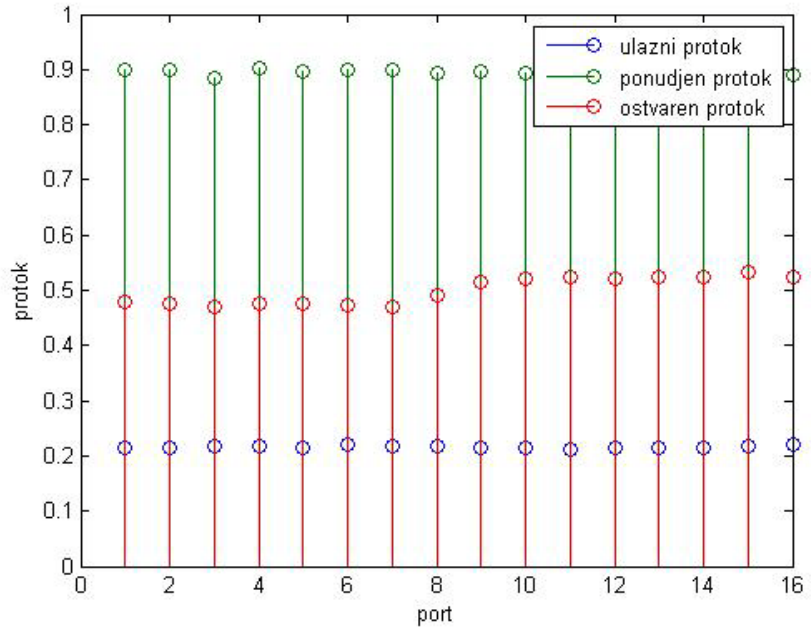


Slika 4.2.  $N=16, N_{max}=10, P=0.217, P_u=0.3, U=1$

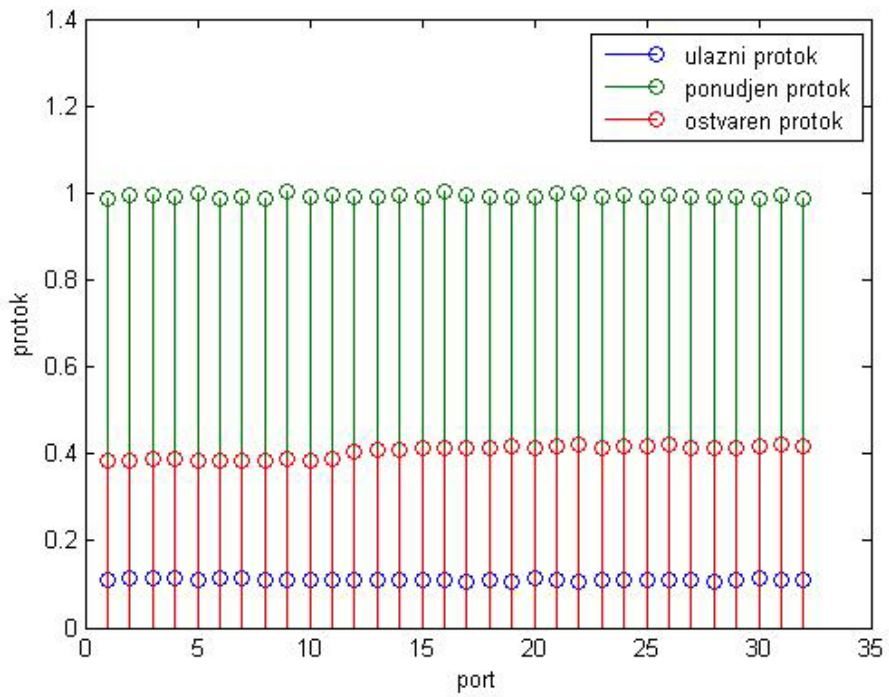


Slika 4.3.  $N=16, N_{max}=10, P=0.217, P_u=0.5, U=1$

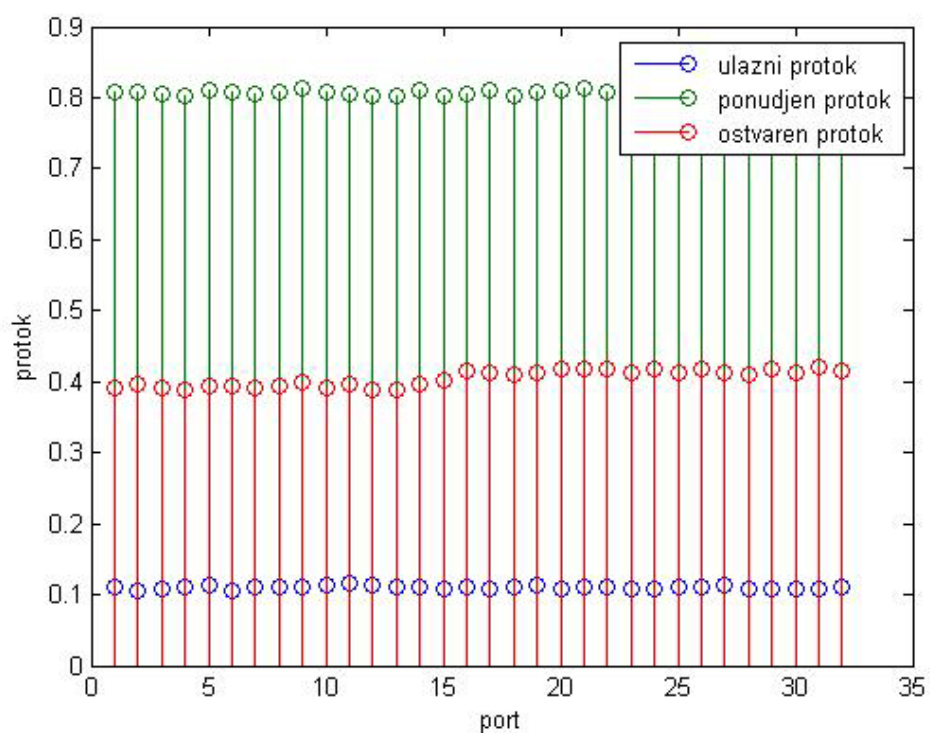
Ostvaren protok je približno isti u sva tri slučaja. Ostvaren protok je relativno nizak jer se paket sa početka bafera šalje samo ako su svi odgovarajući izlazi slobodni što može izazvati HOL (*head of line*) blokadu, tj. upotreba FIFO bafera dovodi do pojave HOL blokade i obaranja propusnosti simuliranog rešenja. Ponudjen protok raste kako se smanjuje udeo unikast saobraćaja. Povećanje ubrzanja nije značajno uticalo na povećanje ostvarenog protoka usled efekta HOL blokade.



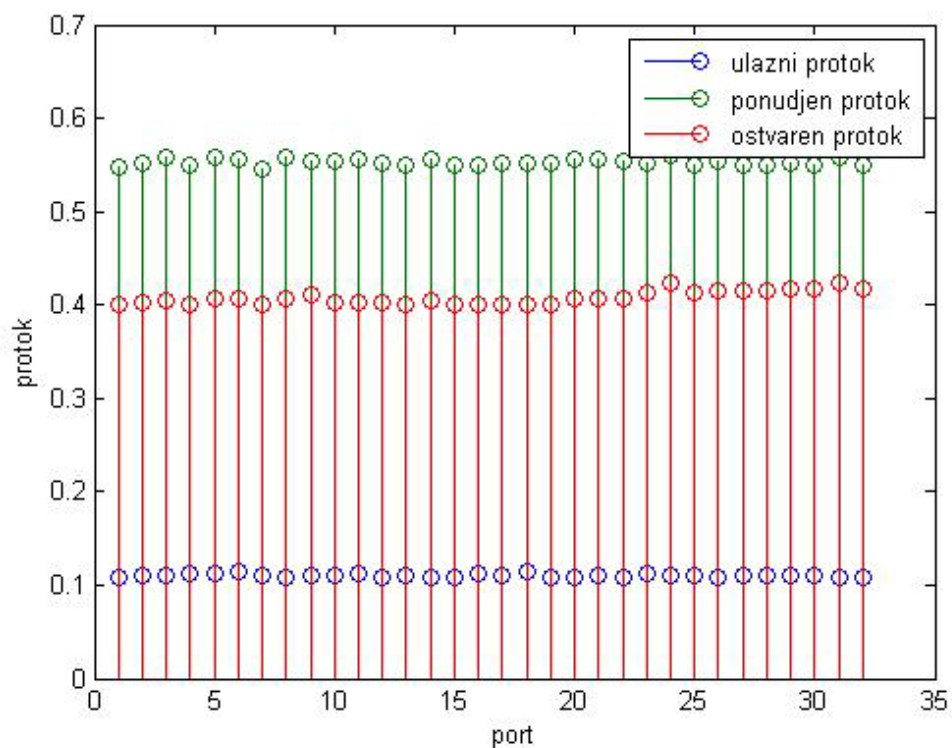
Slika 4.4.  $N=16, N_{max}=9, P=0.217, P_u=0.5, U=2$



Slika 4.5.  $N=32, N_{max}=16, P=0.11, P_u=0, U=1$

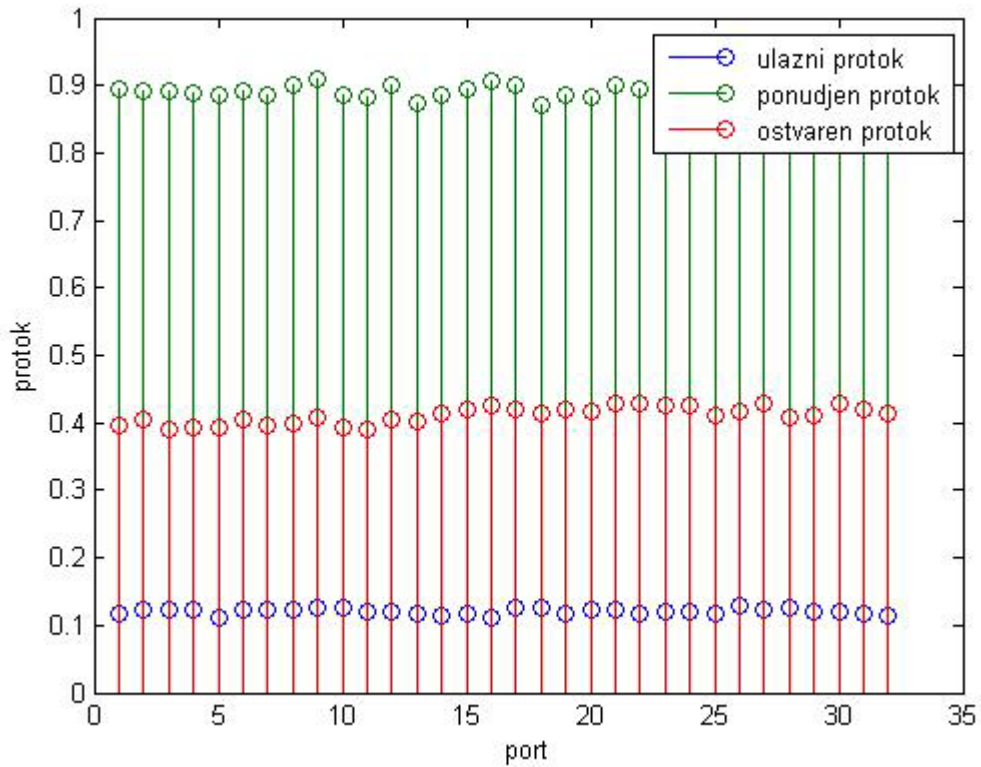


Slika 4.6.  $N=32, N_{max}=16, P=0.11, Pu=0.22, U=1$



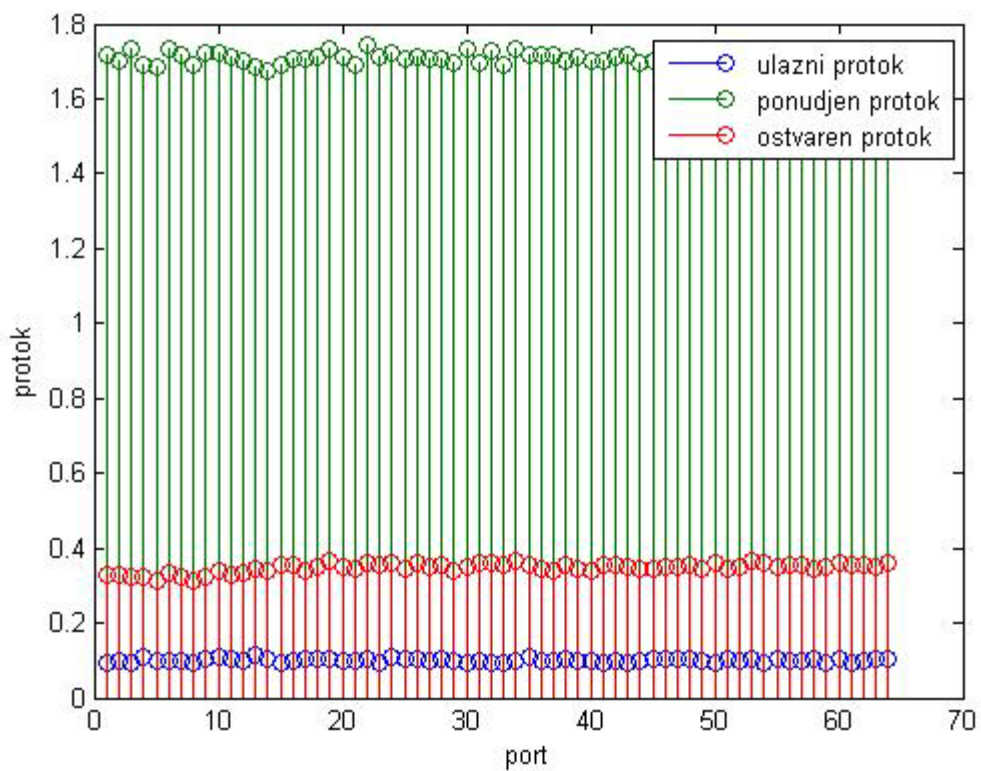
Slika 4.7.  $N=32, N_{max}=16, P=0.11, Pu=0.5, U=1$

Ostvaren protok je približno isti u sva tri slučaja. Primetimo da je opet ostvaren protok relativno nizak jer se koristi FIFO bafer koji dovodi do pojave HOL blokade i obaranja propusnosti simuliranog rešenja. Ponuđen protok raste kako se smanjuje udeo unikast saobraćaja.

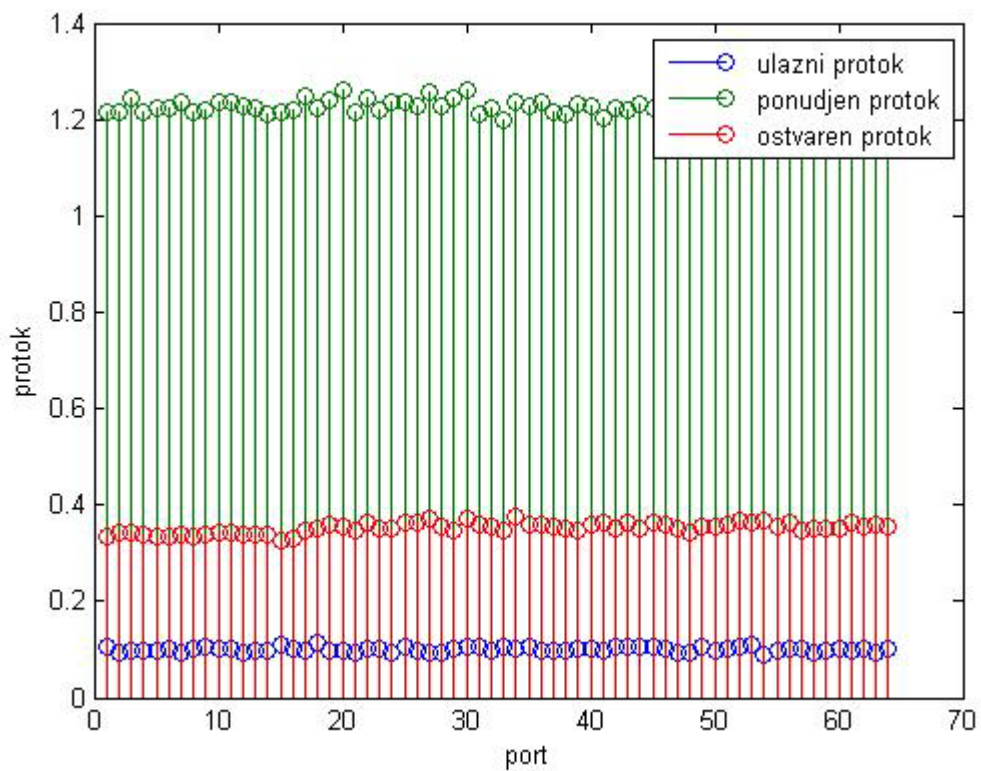


Slika 4.8.  $N=32, N_{max}=16, P=0.11, P_u=0.22, U=2$

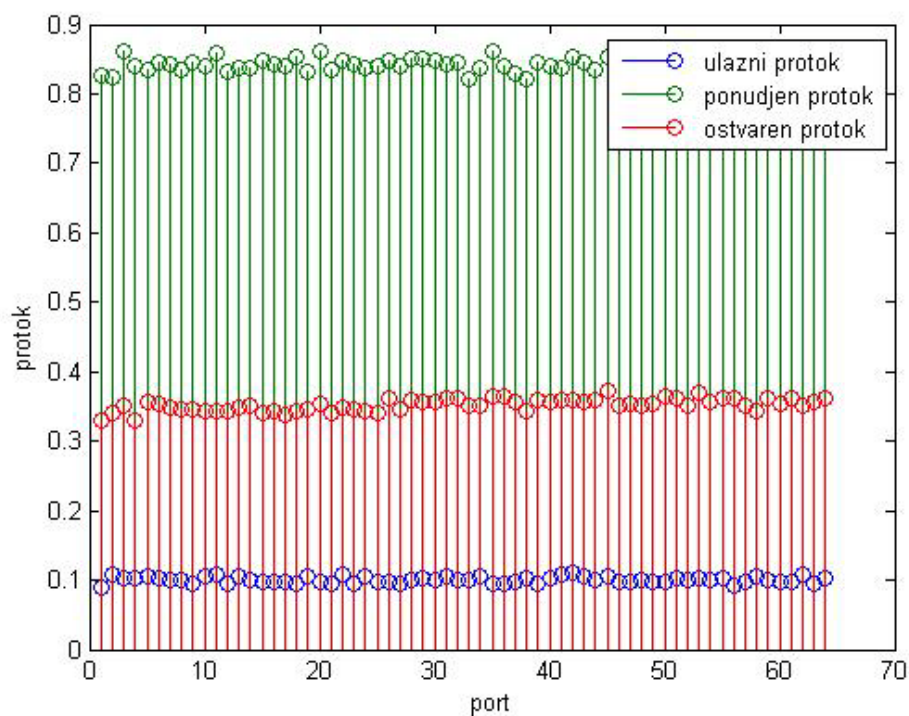
Povećanje ubrzanja nije značajno uticalo na povećanje ostvarenog protoka usled uticaja HOL blokade.



Slika 4.9.  $N=64, N_{max}=32, P=0.1, P_u=0, U=1$

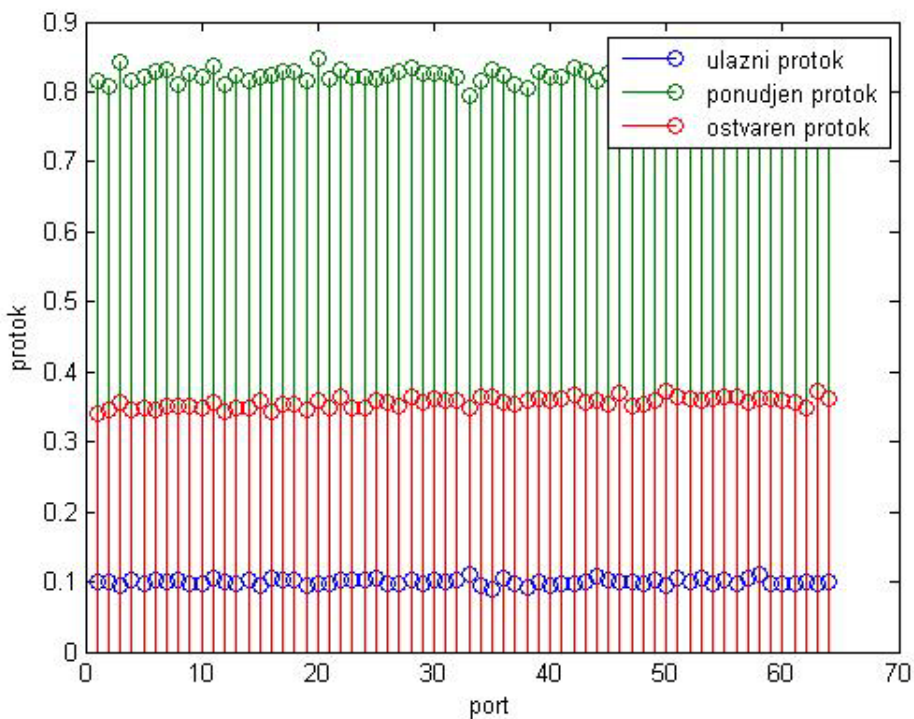


Slika 4.10.  $N=64, N_{max}=32, P=0.1, P_u=0.3, U=1$



Slika 4.11.  $N=64, N_{max}=32, P=0.1, P_u=0.54, U=1$

I ovde ostvaren protok ostaje približno isti za sva tri slučaja. Ponudjen protok raste kako se smanjuje udeo unikast saobraćaja u ukupnom saobraćaju.



Slika 4.12.  $N=64, N_{max}=32, P=0.1, P_u=0.54, U=2$



Povećanje ubrzanja ni ovde nije značajno uticalo na povećanje ostvarenog protoka usled efekta HOL blokade.

Vidi se kako u rezultatima simulacije prikazanim na graficima ponuđen protok opada sa porastom udela unikast saobraćaja u celokupnom saobraćaju. Ovo i jeste očekivano s obzirom na to da je odlika unikast paketa da budu namenjeni samo jednom izlaznom portu u toku jednog vremenskog slotu, dok je kod multikasta podrazumevano da paket ide na više izlaznih portova. Kod multikasta samim tim što paket sa ulaza ide na više izlaznih portova veća je šansa da paket bude blokiran jer se u ovoj varijanti zahteva da svi odgovarajući izlazi budu slobodni. Takođe, kada je ponuđen protok iznad 1 to znači da je svič preopterećen. Ovo važi za sve konfiguracije komutatora (16,32,64) testirane prilikom simulacije.

Ostvareni protok je ostao približno isti za sve konfiguracije komutatora (16, 32, 64) i sve odnose unikast i multikast saobraćaja. Ovakav ostvareni protok je povezan sa tipom bafera koji je korišćen u okviru koda (simulacije), a to je FIFO bafer (*first in first out*). To je vrsta bafera gde se prvi pristigli paket sa ulaza prosleđuje prvi na izlaz i tako redom do poslednjeg. FIFO bafer šalje paket sa prve pozicije samo ako su svi izlazi kojima je taj paket namenjen slobodni. Ovakav način obrade paketa dovodi do HOL (*head of line*) blokade i to obara performanse tj. propusnost komutatora.

Vrednost ubrzanja  $U$  obezbeđuje da se sa svakog ulaznog porta iz bafera pošalje do  $U$  paketa ka izlaznim portovima u jednom slotu. Malo povećanje ubrzanja (sa 1 na 2 što odgovara realnim situacijama) ne utiče mnogo na povećanje ostvarenog protoka u ovom slučaju. Ovakav rezultat je posledica FIFO principa. Bolje performanse bi se dobile kada bi se omogućilo da se dozvoli portovima da umesto da šalju paket samo ako su sva odgovarajuća odredišta slobodna, da mu se dozvoli parcijalno slanje na samo deo odredišta (tzv. *fanout splitting*). Takođe, mogao bi se primeniti VOQ (*virtual output queueing*) jer bi se na taj način povećala šansa za slanje paketa sa porta.

## 5. ZAKLJUČAK

Analizom metoda prenosa podataka u obliku paketa navedenih u prethodnim poglavljima zaključujemo:

Kada bismo unicast tehnikom prenosa pokušali da prosledimo pakete velikom broju korisnika javila bi se dva glavna problema:

1. Predajnik bi bio preopterećen jer bi generisao mnogo kopija istog paketa koji treba proslediti.
2. Mreža bi prenosila mnogostruko više paketa nego što je neophodno.

Multicast se odlikuje sledećim karakteristikama:

Prednosti:

1. Pri prenosu multicast tehnikom omogućava se ista pouzdanost kao i pri prenosu unicast tehnikom, ali je iskorišćenost veća u odnosu na unicast
2. Prijavlivanje i odjavljivanje sa grupe se obavlja dinamički i tako se poboljšavaju performanse u smislu da se paketi šalju samo na one grupe koje imaju aktivne korisnike
3. Podržava da pojedinačni korisnici budu članovi jedne ili više grupa
4. Može da prenosi više tokova do iste grupe
5. Predajnik (ili komutator u slučaju da je paket već unutar mreže) ne poseduje informaciju o broju prijemnih odredišta

Nedostaci:

1. Nema garancije pouzdanog prenosa
2. Nema kontrole zagušenja (sto je veliki nedostatak jer se danas radi sa velikim protocima i brojem paketa)
3. Neka se dobijaju paketi u pogrešnom redosledu, a za određene vrste servisa je bitno da paketi stižu onim redosledom kojim su slati sa izvora
4. Neka možemo primiti duplikate paketa (tada moramo uvesti algoritam za odbacivanje ovakvih paketa)

Zato se uvodi tehnologija IP multicast komunikacije. Njom se odgovornost za umnožavanje paketa, njihovo prosleđivanje kroz mrežu do određene grupe korisnika prebacuje na komutatore u mreži. IP multicast omogućuje da se rastereti izvor informacija i da se smanji nepotrebno umnožavanje paketa.

Problem koji se javlja kod prenosa paketa IP multicast tehnologijom je taj što na putu kroz mrežu paketi nailaze na komutatore koji ne podržavaju IP multicast tehnologije. U tom slučaju deo

prenosa vrši se multikast tehnikom prenosa, a deo unicast metodom prenosa.

Da bi analizirali ovaj problem vršili smo simulaciju komutatora različitih veličina i parametara kroz koje prolaze i multikast i unicast paketi. Zaključili smo da vrednost ostvarenog protoka zavisi od vrste bafera koji se nalazi na ulazu komutatora. U okviru naše simulacije korišćen je FIFO bafer. Ovaj bafer prvi 'najstariji' pristigli paket prosleđuje prvi na sve njemu namenjene izlaze samo ako su svi izlazi slobodni, što može dovesti do HOL (*Head of Line*) blokade i smanjenja performansi kao što je i viđeno kroz rezultate simulacije. Buduće analize bi trebale da ispituju ponašanje sviča kada se koriste VOQ baferi, kao i slučaj kada je dozvoljeno parcijalno slanje paketa na odredišta (tzv. *fanout splitting*).

Dalji razvoj IP multikast tehnologija usmeren je u tri pravca:

1. Zamena komutatora u mreži koji ne podržavaju IP multikast onima koji će ovu tehnologiju podržavati
2. Razvoj softvera i hardvera komutatora kako bi se omogućio rad bez zagušenja pri velikim protocima
3. Globalna pokrivenost IP multikast tehnologijom.

## LITERATURA

- [1] doc. dr Zoran Čiča , Komutacioni sistemi-Paketski komutatori (poglavlje 9), [http://telekomunikacije.etf.rs/predmeti/te4ks/docs/KS/KS\\_09.pdf](http://telekomunikacije.etf.rs/predmeti/te4ks/docs/KS/KS_09.pdf)
- [2] doc. dr Zoran Čiča, Širokopoljasne telekomunikacione mreže-IP multikast (poglavlje 5), [http://telekomunikacije.etf.rs/predmeti/te4ks/docs/STM/STM\\_05.pdf](http://telekomunikacije.etf.rs/predmeti/te4ks/docs/STM/STM_05.pdf)
- [3] prof. dr Zoran Jovanović dr Pavle Vuletić dr Slavko Gajin Marina Vermezović, Multikast, <http://rti.etf.bg.ac.rs/rti/ir4roi/Prezentacije/04%20-%20Multicast.pdf>
- [4] Telecom ABC , Multicast, <http://www.telecomabc.com/m/multicasting.html>
- [5] Telecom ABC, Packet switching, <http://www.telecomabc.com/p/packet-switching.html>
- [6] Wikipedia Free Encyclopedia, PIM protocol, [https://en.wikipedia.org/wiki/Protocol\\_Independent\\_Multicast](https://en.wikipedia.org/wiki/Protocol_Independent_Multicast)

## A. PROGRAMSKI KOD

```
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 udeo unicast paketa'); %verovatnoca pojavljivanja paketa
na ulazu
Podaci = str2num(Niz{1});
Pu = 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); %trodimenzijski niz dimenzija N/N/slotovi, kome
se u svakom slotu na poziciji :,cnt_slot upisuju 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
            pom_Pu = rand(1);
            if pom_Pu <= Pu
                br_izlaza = 1; %broj izlaza na koje se paket upucuje

                izlaz = randperm(N,br_izlaza);
                while izlaz == cnt_port
                    izlaz = randperm(N,br_izlaza);
                end
                matrica_portova(cnt_port,izlaz) = 1; %za gore izabran broj izlaza se
odredjuje na koje izlaze se paket salje
            else
                br_izlaza = 1; %broj izlaza na koje se paket upucuje
                while br_izlaza == 1
                    br_izlaza = randi(Nmax); %broj izlaza na koje se paket upucuje
                end
                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
                    end
                end
            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 cnt_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 veća od 0)
        %pom_bafer je matrica od tri dimenzije i to N/1,slotovi
        [x,y,z] = find(pom_bafer(cnt_port,1,:)); %traže 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
        %treće 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

        if isempty(x) == 0
            a = find(bafer(cnt_port,:,y(1,1))); %predstavlja portove na koje je
poslat taj paket iz bafera
            pom_z = 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
                else
                    break
                end
            end
            if pom_z == z(1) %u slucaju da je pom_z jednako vrednosti z to znaci da
su svi portovi kojima je paket namenjen slobodni
                for cnt = 1:length(a)
                    vektor_slobodnih_izlaza(1,a(1,cnt)) = 1; %upisuju se jedinice u
vektor slobodnih izlaza sto znaci da su ti izlazi zauzeti
                end
                for cnt_izlaz_pom = 1:N
                    if bafer(cnt_port,cnt_izlaz_pom,y(1,1)) == 1
                        matrica_ostvaren_protok(cnt_izlaz_pom,1) =
matrica_ostvaren_protok(cnt_izlaz_pom,1) + 1/slotovi;
                    end
                end
                bafer(cnt_port,:,y(1,1)) = zeros(1,N,1); %upisuju se nule u bafer, sto
znaci da je paket s te pozicije poslat
            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
end

```

```
    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 poslani 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')
```