

Analiza performansi VOQ crossbar komutatora paketa sa LPF baziranim algoritmima

Milutin Radonjić, *Member, IEEE*, Igor Radusinović, *Member, IEEE*

Sadržaj — U ovom radu su prezentovani rezultati do kojih se došlo u procesu projektovanja scheduler-a VOQ (Virtual Output Queueing) krosbar komutatora paketa za implementaciju pomoću FPGA kola. Analiziran je iLPF (iterative Longest Port First) algoritam i njegove varijante, od kojih se najbolje pokazao algoritam LIPFwTM (Longest Input Port First with Throughput Maximization). Kao parametri performansi praktično realizovanih scheduler-a posmatrani su srednje kašnjenje, propusnost i verovatnoća gubitka, pri različitim uslovima saobraćaja. Dobijeni rezultati su upoređeni sa rezultatima računarskih simulacija pri kojima nema ograničenja diktiranih praktičnom realizacijom.

Glavne reči — Krosbar komutator paketa, kašnjenje, propusnost, verilog, verovatnoća gubitka, scheduler.

I. UVOD

SA intenzivnim povećavanjem saobraćaja na Internetu uloga switch-eva i router-a dobija na značaju u procesu komunikacija. Ova činjenica uzrokovala je i intenzivna istraživanja u domenu projektovanja komutacionih polja, kako bi se postigla što veća propusnost podataka. U literaturi se mogu naći različite varijante komutacionih polja, kao i algoritama koji diktiraju način prosleđivanja paketa kroz komutaciono polje.

U ovom radu je opisana implementacija scheduler-a krosbar komutacionog polja veličine 4x4 i to sa VOQ (Virtual Output Queue) metodom baferovanja [1] - [3]. Za upravljanje scheduler-om upotrebljen je iLPF (iterative Longest Port First) algoritam [3], [4]. Pošto u pojedinim uslovima saobraćaja osnovni iLPF algoritam ne postiže najbolje performanse, razmatrali smo i dve njegove varijante: LOPFwTM (Longest Output Port First with Throughput Maximization) i LIPFwTM (Longest Input Port First with Throughput Maximization) [2], [4], od kojih je ovaj drugi pokazao bolje performanse.

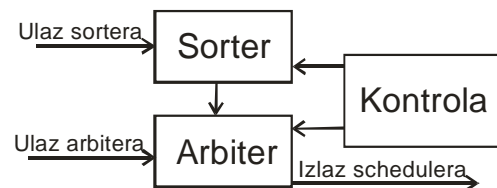
U poglavlju II dat je kratak opis hardverskih komponenti uređaja. Nakon toga, u poglavlju III, prikazani su rezultati simulacija za razne uslove saobraćaja. Zatim su, u poglavlju IV, objašnjeni neki aspekti praktične realizacije sistema. Na kraju su date zaključne napomene sa daljim pravcima istraživanja.

M. Radonjić, Elektrotehnički fakultet u Podgorici, Crna Gora; (telefon: 382-67-224872; faks: 382-20-245873; e-mail: m.radonjic@ieee.org).

I. Radusinović, Elektrotehnički fakultet u Podgorici, Crna Gora; (e-mail: igorr@cg.ac.yu).

II. HARDVERSKA REALIZACIJA UREĐAJA

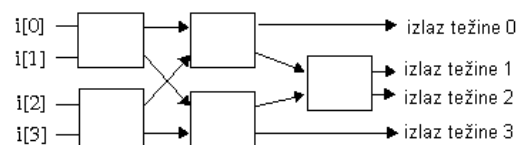
Scheduler krosbar komutatora paketa je projektovan u Verilog-u, koristeći Xilinx-ovo razvojno okruženje. Blok šema uređaja je prikazana na Sl. 1.



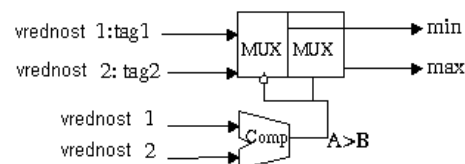
Sl. 1. Blok šema scheduler-a

Na ulaz sortera, kod iLPF algoritma, dovode se informacije o trenutnoj zauzetosti ulaznih i virtuelnih izlaznih bafera (koristi se i termin "portova") u datom vremenskom slotu. Kod LIPFwTM algoritma, takođe se dovodi dužina ulaznog bafera, ali se umesto dužine izlaznog bafera dovodi broj nenulih članova po VOQ za svaki port.

Izlaz iz sortera je, u oba slučaja, sortirana vrednost ulaznih parametara. Kao sorter je korišćena jedna varijanta Batcher mreže za sortiranje [6]. Izgled mreže za sortiranje prikazan je na Sl. 2, a detaljna šema jednog čvora iste mreže na Sl. 3.



Sl. 2. Mreža za sortiranje (sorter)

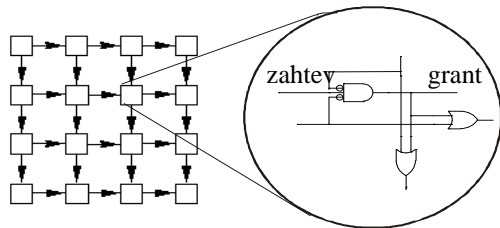


Sl. 3. Jedan čvor mreže za sortiranje

Pošto je potrebno sortirati dve grupe nezavisnih podataka (dužine ulaznih i VOQ bafera po svakom portu komutatora) bila bi potrebna i dva kola za sortiranje, da bi se proces mogao obavljati istovremeno. Međutim, ovim radom se predlaže da se zbog jednostavnosti implementacije i manjeg utroška logičkih ćelija implementira samo jedan sorter kojim će se sortirati obe grupe ulaznih podataka u vremenski pomerenim trenucima.

To se postiže stavljanjem memorijskih elemenata (latcheva) na ulaz i izlaz sortera.

Ulaz u arbiter predstavlja matrica zahteva za prenos podataka. Ta matrica nosi informaciju koji ulazni port ima na čekanju pakete i za koje su izlazne portove namenjeni. Nad tom matricom prvo se izvrše permutacije saglasno izlazima sortera, tako da ulazni port sa najvećom dužinom dođe u red sa indeksom nula, a ulazni port sa najmanjom dužinom u red sa indeksom 3. Na isti način se izvrši permutacija po kolonama. Tako permutovana matrica se dovodi na ulaz kola za arbitražu, čiji izgled je prikazan na Sl. 4.



Sl. 4. Kolo za arbitražu

Izlaz kola za arbitražu je matrica grantova saglasna permutovanoj matrici zahteva koja je na njegovom ulazu. Da bi se dobila prava matrica grantova potrebno je izlaz kola za arbitražu ponovo permutovati u obrnutom smeru i po vrstama (ulaznim redovima) i po kolonama (virtuelnim izlaznim redovima). Tako dobijena matrica grantova predstavlja izlaz kompletnog scheduler-a.

III. REZULTATI SIMULACIJE

U procesu validacije projektovanog uređaja vršene su simulacije njegovog ponašanja u raznim uslovima dolaznog saobraćaja. U tu svrhu, posmatrani su uniformni i Interrupted Bernoulli Process (IBP), za razne vrednosti njihovih parametara [5].

Prilikom inicijalnih testiranja iLPF algoritma, za uniformni model saobraćaja, primećeno je da se matrica grantova na izlazu uređaja generiše na neoptimalan način. Naime, matrice nisu uvek imale maksimalni mogući broj grantova u odnosu na ulazne zahteve. Zato su analizirani potencijalni algoritmi koji bi bolje iskoristili komutacionu matricu u svakom trenutku. Pokazalo se da takvu osobinu poseduje LIPFwTM algoritam, pa su zato u procesima simulacije upoređivane performanse ova dva algoritma.

Pomenuti algoritmi se razlikuju samo u tome koji se podaci dovode na ulaz, pa je praktična realizacija scheduler-a identična za oba slučaja.

Parametri koji su interesantni za analizu u ovakvim slučajevima su *srednje kašnjenje* i *srednja propusnost sistema* [5]. Rezultati dobijeni za srednju propusnost neće biti ilustrovani jer se pokazalo da je ona u većini slučajeva gotovo prava linija, odnosno razmatrani algoritmi u uslovima posmatranih dolaznih saobraćaja obezbeđuju gotovo 100% propusnost.

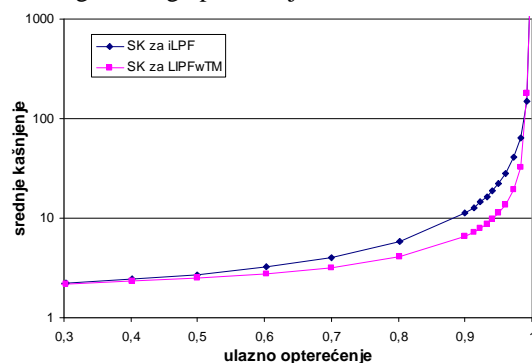
Međutim, kod praktične realizacije se mora unapred odrediti veličina ovog bafera, a za algoritme koji su korišćeni mora se definisati broj ulaznih linija preko kojih će scheduler primati podatke o dužini zauzetog bafera. Zato je posmatran i parametar *verovatnoća gubitka* kao i

parametri *maksimalna dužina ulaznog i izlaznog porta*.

Maksimalna dužina ulaznog odnosno izlaznog porta je naziv za najveću vrednost veličine zauzetosti bafera koja je prosledjena scheduleru tokom simulacije, a koja je posledica generisanog saobraćaja. Na osnovu ovih vrednosti se može dobiti slika o mogućim gubicima za određeni saobraćaj, prilikom ograničenja broja ulaznih linija. Takođe, ovi rezultati mogu pomoći prilikom izbora veličine bafera za ulazne podatke.

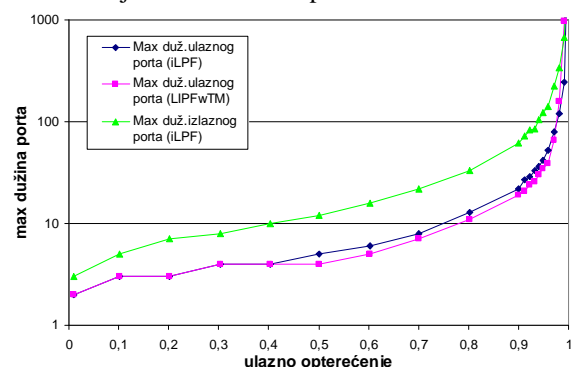
A. Analiza za uniformni saobraćaj

Na Sl. 5 prikazan je dijagram srednjeg kašnjenja za oba algoritma u slučaju neograničenog ulaznog bafera, a za uniformni saobraćaj. Simulacija je obavljena na 100 miliona vremenskih slotova. Sa dijagrama se može uočiti da LIPFwTM algoritam ima manje kašnjenje sve do blizu jediničnog ulaznog opterećenja.



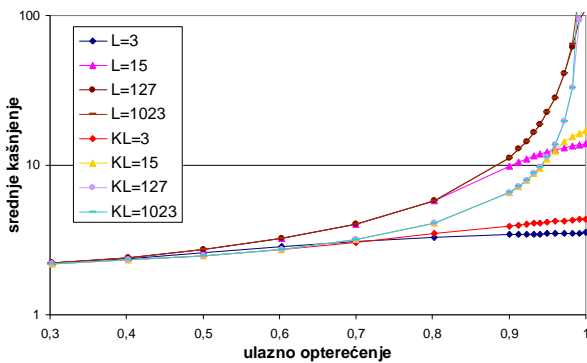
Sl. 5. Srednje kašnjenje za uniformni saobraćaj

Na Sl. 6 su prikazane maksimalne vrednosti dužine portova (sinonim za dužinu zauzetog dela bafera, tj. reda) pri istoj simulaciji. Kod iLPF algoritma posmatramo i ulazni i (virtuelni) izlazni red, dok se kod LIPFwTM algoritma posmatra samo ulazni red jer se umesto dužine virtuelnog izlaznog reda uređaju prosleđuje broj nenultih članova u VOQ-ovima. Ta vrednost može maksimalno da bude jednaka broju ulaznih portova u komutacionoj matrici. Kao što se sa slike može videti, i u ovom aspektu LIPFwTM algoritam pokazuje bolje performanse, jer zahteva manje dužine ulaznih portova.



Sl. 6. Maksimalne dužine ulaznih portova

Nakon ovih rezultata pristupilo se ispitivanju kako se sistem ponaša u slučaju ograničene dužine bafera, odnosno ograničenog broja linija preko kojih se podaci o dužini portova prosleđuju scheduler-u. Rezultati ovih ispitivanja prikazani su na sl. 7.



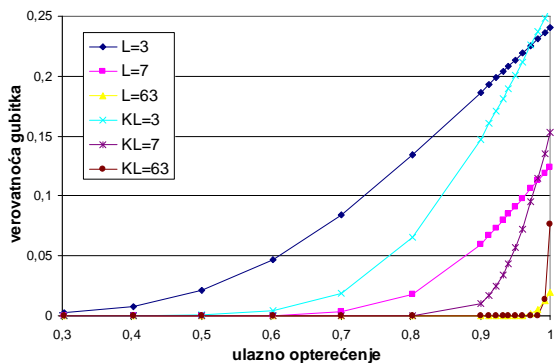
Sl. 7. Srednje kašnjenje za ograničenu dužinu portova

Sa oznakom "L" označena je maksimalno dopuštena vrednost dužine porta pri kojoj je dijagram dobijen. Ako je npr. $L=3$, to znači da je za prijem informacija kod scheduler-a predviđeno dve linije po ulaznom portu. Za $L=1023$ bi bilo predviđeno deset linija. Kod LIPFwTM algoritma stavljen je prefiks "K", da se lakše razlikuje od iLPF algoritma.

Posmatranjem ovih dijagrama možemo uočiti da su rezultati slični kod oba algoritma i da imaju istu tendenciju pri promenama opterećenja i ograničenja broja linija na ulazu. Ono što se može primetiti, a što je i bilo očekivano, je da pri povećanju maksimalne dozvoljene vrednosti dužine portova dobijamo grafik koji sve više liči grafiku sa Sl. 5. I sa ovog dijagrama možemo videti da LIPFwTM algoritam ima nešto bolje rezultate, osim za jako nisko ograničenje dužine porta ($L=3$).

Površnim pogledom na dijagrame sa Sl. 7 moglo bi se zaključiti da najbolje karakteristike ima sistem sa najrestriktivnijim ograničenjem u pogledu dužine ulaznog bafera. Zaista, u tom slučaju imamo najmanje kašnjenje, ali izračunato na bazi paketa koji su prihvaćeni u ulaznom baferu. Međutim, pošto neće svi pristigli paketi biti prihvaćeni, ovde se mora posmatrati i koliko će paketa, zbog ograničenja bafera, biti odbijeno.

Na Sl. 8 je prikazan dijagram verovatnoće gubitka za razne vrednosti ulaznog opterećenja i ograničenja dužine portova kod iLPF i LIPFwTM algoritma. Kao što se može videti, i kod verovatnoće gubitka bolje se pokazao LIPFwTM algoritam, sve do vrlo visokih vrednosti ulaznog opterećenja ($p=0,98$).



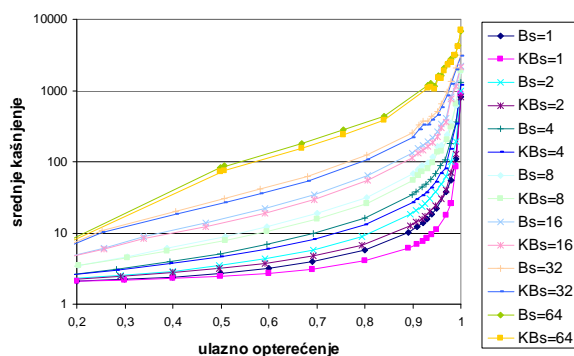
Sl. 8. Verovatnoća gubitka kod iLPF i LIPFwTM za neka ograničenja dužine portova

Iz svega navedenoga može se zaključiti da, u slučaju uniformnog saobraćaja, LIPFwTM algoritam pokazuje bolje karakteristike kod crossbar scheduler-a koji smo izabrali za implementaciju.

B. Analiza za slučaj IBP modela saobraćaja

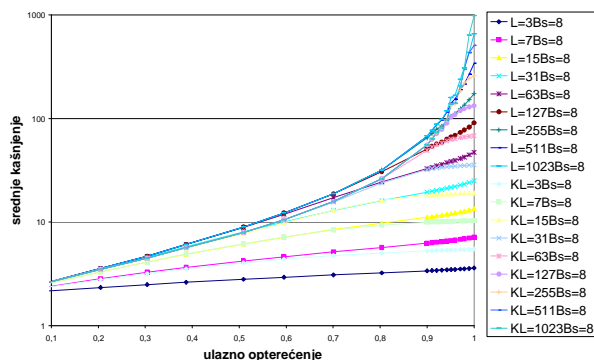
I za ovaj tip saobraćaja posmatrali smo iste parametre kao kod uniformnog saobraćaja, za različita opterećenja i različite vrednosti burst-a (B_s) na milion vremenskih slotova. U [5] je objašnjen ovaj model i način na koji se generiše ulazni saobraćaj. Za naše potrebe koristili smo sledeće vrednosti burst-a: 1, 2, 4, 8, 16, 32 i 64.

Na Sl. 9 se može videti kako srednje kašnjenje zavisi od ulaznog opterećenja i parametra B_s . I ovde je sa "K" označen LIPFwTM algoritam. Slično uniformnom saobraćaju, pokazuje se da LIPFwTM algoritam ima nešto niže vrednosti srednjeg kašnjenja u odnosu na iLPF algoritam.



Sl. 9. Srednje kašnjenje kod IBP modela saobraćaja

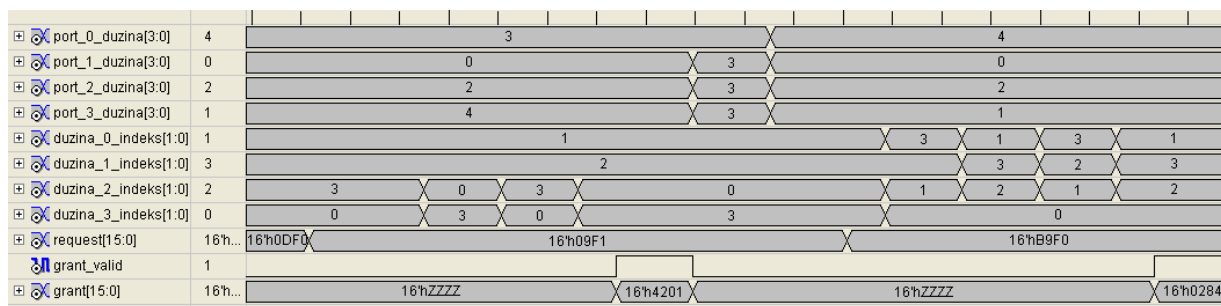
Ako se sada uzmu u obzir i ograničenja u dužini ulaznog bafera (broja ulaznih linija), dobija se veliki broj dijagrama. Kao ilustracija ponašanja sistema, prikazani su dijagrami srednjeg kašnjenja za vrednost $B_s=8$, na Sl. 10.



Sl. 10. Srednje kašnjenje za $B_s=8$

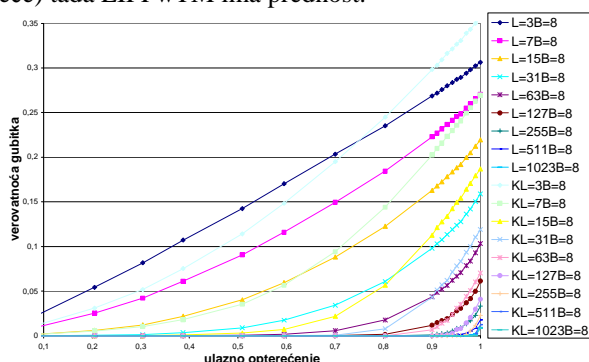
Uz ovaj dijagram, saglasno već rečenome kod uniformnog saobraćaja, mora se posmatrati i verovatnoća gubitaka. Pošto i tih dijagrama ima mnogo, zbog ograničenog prostora biće dati rezultati samo za $B_s=8$ (Sl. 11).

Analizirajući rezultate za različite vrednosti parametra B_s može se primetiti da se za male vrednosti B_s sistem ponaša slično uniformnom saobraćaju i da LIPFwTM algoritam ima bolje performanse. Međutim, kako vrednost



Sl. 12. Neki signali nakon završetka simulacije

burst-a raste performanse sistema zavise od toga koliko je ograničenje na ulazu. Za restriktivnija ograničenja iLPF algoritam se bolje pokazao. Na primer, za $B_s=8$, pri ograničenju na ulazu (L) od 3, 7 i 15, dobija se da iLPF algoritam ima manju verovatnoću gubitka. Međutim, ako se posmatraju ograničenja koja su manje restriktivna (63 i veće) tada LIPFwTM ima prednost.



Sl. 11. Verovatnoća gubitka za $B_s=8$

IV. PRAKTIČNA IMPLEMENTACIJA

Nakon analize za dva modela saobraćaja možemo zaključiti da algoritam LIPFwTM ima bolje karakteristike od iLPF algoritma, u najvećem broju slučajeva. Pošto je naš cilj bio da analiziramo i ponašanje sistema za potrebe praktične realizacije, izabrali smo da ograničenje ulaznog bafera bude $L=15$, tj četiri linije će prenositi informaciju o dužini svakog porta. Kod uniformnog saobraćaja ovakav sistem daje sasvim dobre performanse, a za IBP model saobraćaja predstavlja kompromis između kompleksnosti realizacije i performansi u radu.

Nakon realizacije scheduler-a u verilogu, izvršena je simulacija njegovog ponašanja za različite uslove saobraćaja. Na Sl. 12 prikazan je izgled nekih od signala nakon završetka simulacije za uniformni saobraćaj od 10 hiljada vremenskih slotova, pri ulaznom opterećenju od $p=0,8$. Signal *request* je matrica zahteva za propuštanjem paketa koja se formira na osnovu stanja ulaznog bafera. On predstavlja ulaz arbitera (Sl.1). Signali *port(0,1,2,3)duzina* su dužine portova koje se prosleđuju scheduler-u. To su, dakle, signali koji se dovode na ulaz sortera (Sl. 1). Signale *duzina(0,1,2,3)indeks* prizvodi sorter i prosleđuje ih arbitru. Oni predstavljaju indekse ulaznih portova sortirane prema dužini pripadajućih portova. Signal *grant_valid* signalizira kada je arbitražna završena za tekući vremenski slot i kada je spremna matrica grantova, predstavljena signalom *grant*.

V. ZAKLJUČAK

Nakon prikazanih rezultata možemo zaključiti da je pristup projektovanju crossbar scheduler-a koji smo izabrali bio ispravan. Rezultati koje smo dobili za scheduler sa četiri ulazne linije po portu, preko kojih se prenose dužine portova, pokazuje odlične performanse za slučaj uniformnog saobraćaja i sasvim solidne performanse za slučaj IBP modela saobraćaja. Ispitivanja su vršena i za nebalansirani uniformni model saobraćaja, ali pošto rezultati nisu doveli do novih saznanja, usled nedostatka prostora, isti su izostavljeni.

Dalji pravci istraživanja ići će u pravcu analize performansi FDRR algoritma raspoređivanja [7].

LITERATURA

- [1] N. McKeown, "Scheduling algorithms for input-queued cell switches", PhD. Thesis, University of California at Berkeley, 1995.
- [2] N. H. Liu, K. L. Yeung, D. C. W. Pao, "Scheduling Algorithms for Input-queued Switches with Virtual Output Queueing", *Proc. Of ICC'2001*, Helsinki, Finland, June 2001.
- [3] A. Mekikittikul, N. McKeown, "A Practical Scheduling Algorithm to Achieve 100% Throughput in Input-Queued Switches", *IEEE Infocom 98*, San Francisco, April 1998, Vol 2, pp. 792-799.
- [4] I. Radusinović, M. Pejanović, Z. Petrović, "A Performance Comparison of Different Multi-input Queueing ATM Switches", *Proc. TELSIS 2001*, Nis, Yugoslavia, September 2001.
- [5] I. Radusinović, Z. Veljović, M. Pejanović, Z. Petrović, "Impact of Scheduling Algorithms on Performances of Buffered Crossbar Switch Fabric", *IEEE ICC 2002*, New York, USA, April 2002.
- [6] K. E. Batcher, "Sorting networks and their applications", *Spring Joint Computer Conference*, AFIPS Proc. vol. 32, pp 307-314, 1968.
- [7] D. Banović, I. Radusinović, "Scheduling algorithm for VOQ switches", *AEU - International Journal of Electronics and Communications*, Vol. 62, Issue 6, Jun 2008., pp. 455-458, ISSN 1434-8411.

ABSTRACT

In this paper, VOQ crossbar switch scheduler implementation through FPGA circuits is explained. It is constructed for iLPF (iterative Longest Port First) algorithm, as well as its variant LIPFwTM (Longest Input Port First with Throughput Maximization). Parameters for evaluating device performance (average delay, throughput and loss probability) are evaluated under different traffic conditions. Results of behavioral simulations on implemented circuits are compared with simulations on modeled circuits that are not limited as implemented ones.

PERFORMANCE ANALYSIS OF LPF BASED VOQ CROSSBAR SWITCHES

Milutin Radonjić, Igor Radusinović