Analiza performansi mreže elementarnih Markovljevih podvorbenih sustava

Pašalić, Luka

Master's thesis / Diplomski rad

2020

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **University of Zagreb, Faculty of Transport and Traffic Sciences / Sveučilište u Zagrebu, Fakultet prometnih znanosti**

Permanent link / Trajna poveznica: https://urn.nsk.hr/urn:nbn:hr:119:012210

Rights / Prava: In copyright / Zaštićeno autorskim pravom.

Download date / Datum preuzimanja: **2024-04-19**

Repository / Repozitorij:

Faculty of Transport and Traffic Sciences - Institutional Repository

SVEUČILIŠTE U ZAGREBU
FAKULTET PROMETNIH ZNANOSTI

Luka Pašalić

ANALIZA PERFORMANSI MREŽE ELEMENTARNIH MARKOVLJEVIH PODVORBENIH SUSTAVA

DIPLOMSKI RAD

Zagreb, 2020.

Sveučilište u Zagrebu
Fakultet prometnih znanosti

DIPLOMSKI RAD

ANALIZA PERFORMANSI MREŽE ELEMENTARNIH MARKOVLJEVIH PODVORBENIH SUSTAVA

PERFORMANCE ANALYSIS OF A NETWORK OF ELEMENTARY MARKOVIAN QUEUING SYSTEMS

Mentor: doc. dr. sc. Marko Matulin

Student: Luka Pašalić, bacc. ing. traff.
JMBAG: 013538479

Zagreb, srpanj 2020.

ANALIZA PERFORMANSI MREŽE ELEMENTARNIH MARKOVLJEVIH PODVORBENIH SUSTAVA

SAŽETAK

Svrha istraživanja u ovome radu je opisati ulogu čvora u mreži, približiti čitatelju teoriju matematičkih zakonitosti, kojima se može lakše razumjeti i analizirati podvorbene sustave te kroz simulacije doći do zaključaka o ponašanju mreže u ovisnosti o veličini određenih parametara i rasporedu čvorova u mreži. Konfiguriranje i pokretanje simulacije mreže s jednim, dva i tri čvora odvija se u MatLab-u, odnosno njegovom grafičko-programskom sučelju za modeliranje, Simulink-u. Cilj rada je analizirati te mreže na temelju simulacija, koje daju realne statističke podatke o vremenu čekanja i posluživanja u mreži. U posljednjim stranicama rada, predstavljeni su rezultati, kako grafički tako i tablično, na temelju kojih se dolazi do konkretnih zaključaka o utjecaju pojedinih parametara na performanse mreže.

KLJUČNE RIJEČI: čvor; podvorbeni sustav; simulacija; Simulink; vrijeme posluživanja

SUMMARY

The purpose of this thesis is to describe the role of the node in a network, make it easier for a reader to understand and analyze queuing systems through mathematical laws, and with the help of simulations, come to the conclusion about network behavior in dependency of certain parameters and node arrangement. Configuring and starting a simulation with 1, 2, and 3 nodes is done through the Simulink, which is MatLabs graphical-programming interface for modeling. The objective of this thesis is to analyze these networks, based on the results from simulations, which gives realistic statistic data about waiting and serving time. In the last pages of this thesis are the network performance results, which are presented through the graphs and tables, and based on that results, the conclusions are made.

KEY WORDS: node; queuing system; simulation; Simulink; serving time

SADRŽAJ

1. UVOD 1

2. OPIS RADA I ULOGA TELEKOMUNIKACIJSKOG ČVORA 3

2.1. Upravljanje usmjeravanjem 3

2.1.1. Statičko umjeravanje 4

2.1.2. Dinamičko usmjeravanje 4

2.1.3. Zadano usmjeravanje 5

2.2. Kontrola prometnog toka 6

2.2.1. Softverska kontrola toka 8

2.2.2. Stani-i-pričekaj kontrola toka 9

2.2.3. Kontrola toka klizajućeg prozora 10

2.3. Kontrola pogreške 11

2.3.1. Otkrivanje pogreške 12

2.3.1.1. Provjera pariteta 12

2.3.1.2. Shema ponavljanja 12

2.3.1.3. CRC 13

2.3.1.4. *Checksum* 13

2.3.2. Ispravak pogreške 13

2.4. Zaključna razmatranja o radu i ulozi čvora 14

3. MATEMATIČKI OPIS PODVORBENIH SUSTAVA 15

3.1. Jednoposlužiteljski Markovljevi podvorbeni sustavi s beskonačnim izvorištem 16

3.1.1. M/M/1/∞/∞/FCFS 16

3.1.2. M/M/1/K/∞/FCFS 19

3.2. Višeposlužiteljski Markovljevi podvorbeni sustavi s beskonačnim izvorištem 22

3.2.1. M/M/C/∞/∞/FCFS 22

3.2.2. M/M/C/C/∞/FCFS 24
4. IZRAČUN PERFORMANSI SUSTAVA NA IZOLIRANOM DIJELU MREŽE 27
5. PRIKAZ RAČUNALNOG MODELA 31
6. ANALIZA PERFORMANSI MREŽE U OVISNOSTI O NJENOJ VELIČINI 35
6.1. Usporedba rezultata ručno izračunatih zadataka s rezultatima simulacije 35
6.2. Analiza performansi mreže s dva čvora 37
6.3. Analiza performansi mreže s tri čvora 42
7. ZAKLJUČAK 51
LITERATURA 52
POPIS KRATICA 54
POPIS SLIKA 55
POPIS TABLICA 56
POPIS GRAFIKONA 57

1. UVOD

Podvorbeni sustavi kao jedna od ključnih komponenti analiziranja raznih grana industrije i znanosti, a poglavito telekomunikacija, predstavlja područje kojim se bavi ovaj diplomski rad. Kako bi sustav telekomunikacija, a tako i šire poznati termin Internet, pravilno obavljali zadatke za koje su namijenjeni, potrebno je sustav dimenzionirati kako ne bi došlo do prevelikih zagušenja i čekanja, a opet s druge strane potrebno je racionalno koristiti resurse, tj. potrebno je odrediti optimalnu konfiguraciju čvora ili mreže u cjelini.

U razradi tematike ovoga rada korišteni su znanstveni radovi, knjige namijenjene za proučavanje teorije čekanja i podvorbenih sustava, portali koji se bave tematikom koji predstavljaju interesno područje ovoga diplomskog rada te sveučilišna predavanja koja pružaju jasan uvod u navedeno područje. Za daljnju analizu i zaključke, korišten je MatLab, odnosno njegovo grafičko-programsko sučelje za modeliranje, Simulink. Simulink pruža mogućnost simulacije podvorbenih sustava te je u ovome radu tako stvorena virtualna mreža čvorova, u kojoj se s pomoću izmjerenih parametara, analizirao čvor i mreža s dva i tri čvora.

Rad se sastoji od sedam poglavlja:

1. Uvod
2. Opis rada i uloga telekomunikacijskog čvora
3. Matematički opis podvorbenih sustava
4. Izračun performansi sustava na izoliranom dijelu mreže
5. Prikaz računalnog modela
6. Analiza mreže performansi u ovisnosti o njenoj veličini
7. Zaključak.

U drugom poglavlju se opisuje telekomunikacijski čvor te koje su njegove funkcije u mreži.

Treće poglavlje matematički opisuje podvorbene sustave, prikazuje formule potrebne kako bi se došlo do određenih bitnih veličina te vidjele razlike između osnovnih Poisson-ovskih podvorbenih sustava.

Četvrto poglavlje prikazuje izračun parametara u podvorbenom sustavu, bez upotrebe računalnih alata.

U petom poglavlju opisane su osnovne zadaće objekata, koji se nalaze u preddefiniranom modelu potrebnom za provođenje simulacije, kako bi poslije bilo moguće proširiti model na više čvorova.

Šesto poglavlje donosi opis procesa provođenja simulacije te njene rezultate, prikazane grafički i tablično, kako bi se jasnije dobio osjećaj o utjecaju određenih parametara na mrežu.

Sedmo poglavlje predstavlja zaključak te završna mišljenja autora o rezultatima simulacije i analizi podvorbenih sustava.

2. OPIS RADA I ULOGA TELEKOMUNIKACIJSKOG ČVORA

Kako bi se definirao opis rada i uloga telekomunikacijskog čvora, potrebno je definirati sami termin čvor. Čvor je bilo koji fizički uređaj u mreži, koji je sposoban poslati, primiti ili proslijediti informacije. Primjer često korištenih čvorova su modemi, usmjerivači, hub-ovi, poslužitelji, pisači, kao i ostali uređaji koji se povezuju putem Wi-Fi-a ili Ethernet-a.

Međutim, ovo poglavlje neće se baviti funkcijom nekog od navedenih čvorova pojedinačno, već općenitom ulogom telekomunikacijskog čvora, tj. što bi on trebao raditi. Prema [1], uloga čvora u mreži može se podijeliti u tri osnovne procesa, nužna za uspješno obavljanje prijenosa informacija u telekomunikacijama:

- upravljanje usmjeravanjem (engl. *routing control*),
- kontrola prometnog toka (engl. *flow control*),
- kontrola pogrešaka (engl. *error control*).

2.1. Upravljanje usmjeravanjem

Usmjeravanje je proces odabira puta kojim podatci mogu biti preneseni s izvora na odredište. Usmjeravanje se obavlja uređajem koji se naziva usmjerivač (engl. *router*), te navedeni uređaj prosljeđuje podatke, tj. pakete prema informacijama kojima se nalaze u zaglavlju paketa (engl. *packet header*) i informacijama koje se nalaze u tablicama usmjeravanja (engl. *forwarding table*). Algoritmi usmjeravanja (engl. *routing algorithms*) se koriste za usmjeravanje paketa te su oni zapravo softveri odgovorni za odabiranje optimalnog puta kojim će paket stići na odredište, [2].

Protokoli usmjeravanja koriste metriku za određivanje najboljeg puta za isporuku paketa. Metrika je standard mjerenja poput broja skokova (engl. *hop count*), širine pojasa (engl. *bandwidth*), kašnjenja (engl. *delay*), trenutnog opterećenja na putu (engl. *current load on the path*) itd., koji algoritam usmjeravanja koristi za određivanje optimalnog puta do odredišta. Algoritam usmjeravanja inicijalizira i održava tablicu usmjeravanja za postupak određivanja puta. Usmjeravanje se, prema [2], može podijeliti u tri kategorije:

- statičko usmjeravanje,
- dinamičko usmjeravanje,
- zadano usmjeravanje.

2.1.1. Statičko umjeravanje

Statičku tablicu usmjeravanja stvara, održava i ažurira mrežni administrator, ručno. Za potpuno povezivanje, na svakom usmjerivaču mora se konfigurirati statička ruta do svake mreže. To osigurava detaljnu razinu kontrole nad usmjeravanjem, ali postaje nepraktično na velikim mrežama. Usmjerivači neće međusobno dijeliti statičke rute, smanjujući na taj način upotrebu CPU-a/RAM-a te raspoloživi prijenosni pojas. Međutim, statičko usmjeravanje nije otporno na greške, jer svaka promjena infrastrukture (poput ispada linka ili dodavanja nove mreže) zahtijeva ručnu intervenciju. Usmjerivači koji rade u čisto statičkom okruženju ne mogu jednostavno odabrati bolju rutu, ako link postane nedostupan, [3].

Prema [4], dolazi se do zaključaka kada bi se trebalo koristiti statičko usmjeravanje:

- u manjim mrežama jer pruža lakoću održavanja tablice usmjeravanja,
- u mrežama kada usmjerivač nema više od jednog susjednog izlaza,
- kada se koristi jedan put do bilo koje mreže, koja nema određenije podudaranje sa drugim putem u tablici usmjeravanja.

2.1.2. Dinamičko usmjeravanje

Dinamičku tablicu usmjeravanja stvara, održava i ažurira protokol usmjeravanja koji djeluje na usmjerivaču. Primjeri protokola usmjeravanja uključuju RIP (engl. *Routing Information Protocol*), EIGRP (engl. *Enhanced Interior Gateway Routing Protocol*) i OSPF (engl. *Open Shortest Path First*), [3].

Dinamičko usmjeravanje se koristi kada se protokoli, zvani protokoli usmjeravanja, koriste za izgradnju tablica usmjeravanja u mreži. Korištenje protokola usmjeravanja lakše je od statičkog usmjeravanja i zadanog usmjeravanja, ali je skuplje u pogledu upotrebe CPU-a i propusnosti. Svaki protokol usmjeravanja definira vlastita pravila komunikacije između usmjerivača i odabira najbolje rute, odnosno puta, [5]. U tablici 1 mogu se vidjeti osnovne razlike između dinamičkog i statičkog usmjeravanja.

Protokoli usmjeravanja su klasificirani kao IGP (engl. *Interior Gateway Protocols*) i EGP (engl. *Exterior Gateway Protocols*). IGP protokoli se koriste za razmjenu informacija unutar mreža, koje spadaju u jednu administrativnu domenu (još zvanu autonomni sustavi). EGP protokoli se koriste za razmjenu informacija o usmjeravanja između različitih autonomnih sustava, [5].

Važno je napomenuti da se ne smije poistovjetiti protokole usmjeravanja (engl. *routing protocol*) s protokolom preusmjeravanja (engl. *routed protocol*). *Routed* protokol obavlja funkcije trećeg sloja OSI modela te dodjeljuje logičke adrese uređajima i usmjerava podatke između mreža (npr. IP), a *routing* protokol dinamički slaže informacije o mreži, topologiji i sljedećem skoku u tablice usmjeravanja (npr. RIP, EIGRP i dr.), [3].

Tablica 1. Usporedba statičkog i dinamičkog usmjeravanja

	Statičko usmjeravanje	Dinamičko usmjeravanje
Upotreba CPU/RAM-a	Minimalna	Dodatna
U slučaju pada linka	Mreža postaje neupotrebljiva	Odabire se druga ruta
Infrastrukturne promjene	Ručno	Automatski
Velike mreže (konfiguracija)	Nepraktično	Jednostavno

Izvor: [3]

2.1.3. Zadano usmjeravanje

Zadano usmjeravanje može se smatrati posebnom vrstom statičkog usmjeravanja. Razlika između normalne statičke rute i zadane rute je što se zadana ruta koristi za slanje paketa namijenjenih bilo kojem nepoznatom odredištu na jednu sljedeću skočnu adresu, [5].

Zadano usmjeravanje je tehnika u kojoj je usmjerivač konfiguriran za slanje svih paketa na isti hop uređaj, a nije važno pripada li određenoj mreži ili ne. Paket se prenosi na uređaj za koji je konfiguriran u zadanom usmjeravanju. Zadano usmjeravanje koristi se kada mreže rade s jednom izlaznom točkom, [2].

Zadano usmjeravanje također je korisno kada većinu odredišnih mreža treba usmjeriti na jedan uređaj. Pri dodavanju zadane rute, potrebno je osigurati da uređaj koji slijedi, može dalje proslijediti paket ili će sljedeći uređaj ispustiti paket, tj. bit će izgubljen. Još jedna stvar koju je potrebno napomenuti je da kada u tablici usmjeravanja postoji određena ruta do odredišta, usmjerivač će koristiti tu rutu, a ne zadanu rutu. Jedino kada usmjerivač koristi zadanu rutu je kada određena ruta ne postoji, [5].

2.2. Kontrola prometnog toka

U stvarnim sustavima, ponekad se dogode situacije u kojima se primatelj ne može „nositi“ s količinom podataka koje mu šalje pošiljatelj: poslužitelj ne može tako brzo zapisati na disk podatke koje mu šalju računala ili se primateljev međuspremnik popuni dok su resursi zauzeti nekom drugom radnjom. Kako bi se izbjeglo preplavljivanje međuspremnika, mora postojati rješenje kojim je moguće pauzirati tu aktivnost. Takav proces je poznat pod imenom – kontrola prometnog toka. Uobičajeno se obavlja na transportnom i/ili podatkovnom sloju, [7].

Kontrola prometnog toka je tehnika koja osigurava da pošiljatelj koji šalje podatke ne preplavi primatelja s prevelikom količinom podataka. Kontrola toka je skup procedura koja govori pošiljatelju koliko podataka može poslati prije nego što pričeka potvrdu od primatelja, [6]. Postoje tri primarna tipa kontrole prometnog toka u uobičajenoj uporabi:

- softverska kontrola toka,
- stani-i-pričekaj kontrola toka (engl. *stop-and-wait flow control*),
- kontrola toka klizajućeg prozora (engl. *sliding window flow control*).

Glavne funkcije kontrole toka u mreži s paketnim prijenosom podataka su:

- prevencija degradacije propusnosti i gubitka efikasnosti zbog preopterećenja,
- izbjegavanje zastoja,
- ravnopravna dodjela resursa između korisnika,
- odgovarajuća brzina između mreže i korisnika koji su povezani s mrežom.

Degradacija propusnosti i zastoj se događaju iz razloga što promet koji je već prihvaćen u mrežu, premašuje odgovarajući kapacitet. Kako bi se izbjegla pretjerana dodjela resursa, kontrola toka uključuje skup ograničenja (na međuspremnike koji se ne mogu dodijeliti, izvanredne pakete, razine transmisije i dr.) koji mogu efektivno ograničiti promet kako bi pristupio mreži, odnosno određenim dijelovima mreže. Takvi skupovi ograničenja mogu biti fiksni ili dinamički prilagođeni, ovisni o stanju mreže, [8].

Osim zahtjeva propusnosti, mrežni resursi moraju biti ravnopravno raspodijeljeni među korisnicima. Jedna od funkcija kontrole toka je sprječavanje neravnopravnosti postavljanjem selektivnih ograničenja na količinu resursa koje se svakom korisniku (ili grupi korisnika) mogu dodijeliti, usprkos negativnom učinku koji ta ograničenja mogu imati na dinamičko dijeljenje resursa te na ukupnu učinkovitost propusnosti. Prema [8], kontrola protoka može se provoditi na različitim razinama u paketnoj mreži:

- razina skoka (engl. *hop level*) - ova razina kontrole toka pokušava održavati nesmetan protok prometa između dva susjedna čvora u računalnoj mreži, izbjegavajući lokalna zagušenja u međuspreminicima i zastoje,
- razina od-ulaza-do-izlaza (engl. *entry-to-exit level*) – ova razina kontrole toka je uobičajeno implementirana kao protokol između izvorišnog i odredišnog preklopnika (engl. *switch*) te ima svrhu prevencije preopterećenosti u međuspremniku na izlaznom preklopniku,
- razina pristupa mreži (engl. *network access level*) - cilj ove razine je prigušivanje vanjskih ulaza na temelju mjerenja unutarnjih (za razliku od odredišta) zagušenja mreže,
- transportna razina (engl. *transport level*) - ovo je razina kontrole protoka povezana s protokolom na transportnom sloju, tj. protokolom koji osigurava pouzdanu isporuku paketa na virtualnu vezu između dva udaljena procesa. Njegova je glavna svrha sprječavanje zagušenja korisničkih međuspremnika na razini procesa (tj. izvan mreže).

2.2.1. Softverska kontrola toka

Softverska kontrola toka, ponekad još zvana XON/XOFF[1] kontrola nakon česte implementacije u RS-232[2] serijskim vezama, dopušta slanje signala koje će signalizirati pošiljatelju da nastavi slati podatke (XON signal) ili da stane i pričeka (XOFF signal), dok se ne pošalje XON signal. Tehnika je vrlo jednostavna i često korištena te postoje ASCII kodovi koji su konvencijom dodijeljeni XON-u (0x11) i XOFF-u (0x13)[3]. Softverska kontrola toka je vrlo učinkovita vrsta kontrole toka. U slučaju da je primatelj u mogućnosti prihvatiti podatke u maksimalnoj mjeri koju pošiljatelj može poslati, nema potrebe za slanje ikakve dodatne poruke ili signala, iako navedena kontrola nije potpuno otporna na pogreške, [7].

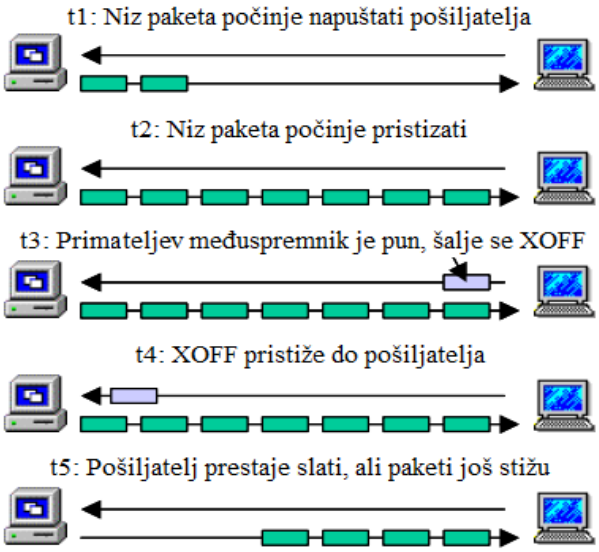
Rukovanje (engl. *handshake*) se uglavnom koristi za sprječavanje preplavljenja međuspremnika primatelja, međuspremnika u memoriji koji se koristi za spremanje nedavno primljenih bajtova. Ako dođe do preplavljenja, to utječe na način na koji se upravlja novim paketima na komunikacijskom kanalu. U najgorem slučaju kada je softver loše dizajniran, ti se paketi bacaju bez provjere. Ako je takav paket XOFF ili XON, tok komunikacije može biti ozbiljno narušen. Pošiljatelj će kontinuirano dostavljati nove podatke, ako se izgubi XOFF ili nikad neće slati nove podatke, ako nije primljen XON, [9].

Postoji još jedan problem s ovakvim pristupom kontroli toka, a problem je taj da ne radi dobro u slučaju kada su pošiljatelj i primatelj na velikoj udaljenosti. Problem se javlja s kašnjenjem u mreži. Ako se primateljev međuspremnik počne puniti, primatelj pošalje XOFF poruku pošiljatelju, tok neće stati istog trenutka. Prvo XOFF poruka mora stići do pošiljatelja te tijekom tog vremena pošiljatelj će kontinuirano slati pakete. Također, kada pošiljatelj i prestane slati pakete, na linkovima može biti još podataka koji stižu primatelju, [7].

Potpuno vrijeme u kojemu će primatelj nastaviti primati pakete, nakon što pošalje XOFF poruku je RTT (engl. *Round Trip Time*) linka – vrijeme potrebno XOFF-

[1] *X označava pošiljatelja.*
[2] *RS-232 (Recommended Standard 232) – standard u telekomunikacijama za prijenos podataka u serijskim komunikacijama.*
[3] *Striktno gledano, ti ASCII kodovi su definirani kao „Device control 1“ i „Device Control 3“, ali se najčešće koriste za slanje XON i XOFF signala u softverskoj kontroli toka.*

u da stigne do pošiljatelja te zadnje poslani paket koji treba stići do primatelja, kao što je prikazano na slici 1.



Slika 1. Softverska kontrola toka, [7]

2.2.2. Stani-i-pričekaj kontrola toka

Kako bi se riješili problemi s kašnjenjem duž linka, primatelja se mora staviti u poziciju s većom kontrolom razine informacija koje će se razmijeniti. Naime, ako primatelj bude u mogućnosti da uz informaciju koja daje signal pošiljatelju kada slati pakete, pošalje i informaciju koliko paketa smije poslati, riješio bi se određeni dio problema koji se događa u softverskoj kontroli toka. Upravo navedeno radi stani-i-pričekaj kontrola toka. Postoji cijena koja se treba platiti u smislu efikasnosti, ali dobro dizajnirani protokoli mogu eliminirati mogućnosti gubitka paketa, zbog prelijevanja međuspremnika, [7]

Stani-i-pričekaj je najjednostavniji oblik kontrole toka, gdje pošiljatelj prenosi podatkovni okvir (engl. *data frame*). Nakon primitka okvira, primatelj pokazuje spremnost da prihvati drugi podatkovni okvir tako što vraća ACK[4] podatkovni okvir, koji potvrđuje upravo primljeni podatkovni okvir. Pošiljatelj mora pričekati dok ne primi ACK okvir prije nego što pošalje sljedeći podatkovni okvir. To se ponekad naziva *ping-pong*

[4] *Podatkovni okvir koji se vraća pošiljatelju kako bi se potvrdio primitak okvira.*

ponašanjem, zahtjev/odgovor jednostavan je za razumijevanje i jednostavan za implementaciju, ali nije vrlo učinkovit, [10].

U LAN okruženju s brzim vezama, to ne predstavlja veliku brigu, ali WAN veze provest će većinu svog vremena u praznom hodu, pogotovo ako je potrebno nekoliko skokova. Uz upotrebu više okvira za jednu poruku, navedena kontrola toka može biti neadekvatna. Problem je u tome što samo jedan podatkovni okvir može biti u prijenosu. Za vrlo visoke brzine podataka ili za vrlo velike udaljenosti između pošiljatelja i primatelja, kontrola protoka stani-i-pričekaj ne omogućava učinkovito korištenje linka, [6].

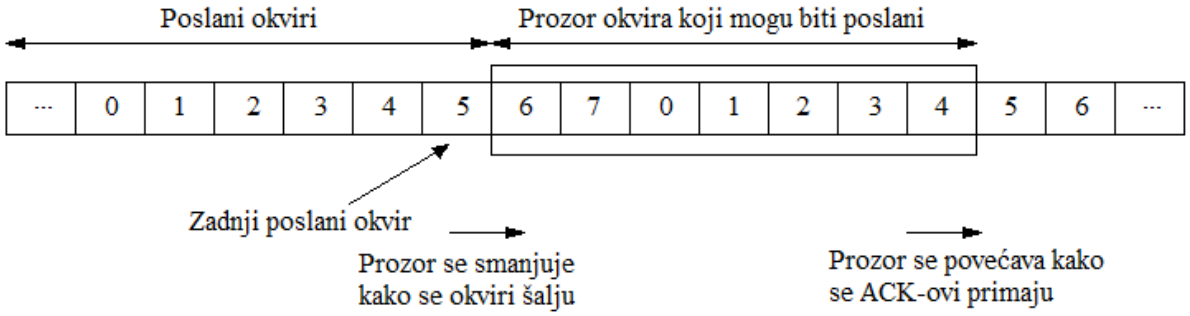
2.2.3. Kontrola toka klizajućeg prozora

Korištenjem više podatkovnih okvira za jednu poruku, stani-i-pričekaj kontrola toka ne daje dobre rezultate. Samo jedan podatkovni okvir istovremeno može biti u prijenosu. U kontroli toka stani-i-pričekaj, ako je vrijeme propagacije dulje od vremena transmisije, onda znači da je pošiljatelj završio sa slanjem cijelog podatkovnog okvira, a prvi bitovi još nisu stigli do primatelja, tada rezultiraju ozbiljne neučinkovitosti. Učinkovitost se može uvelike poboljšati dopuštajući istodobni prolaz više okvira. Učinkovitost se također može poboljšati korištenjem *full-duplex*[5] linije, [10].

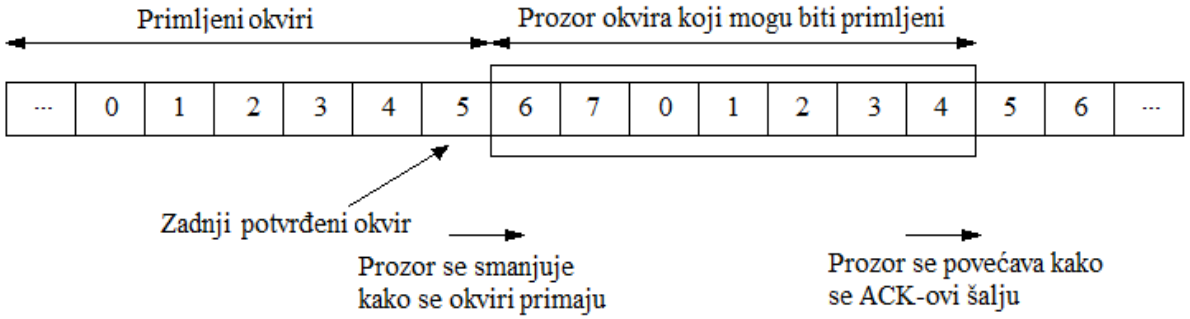
Kontrola toka klizajućeg prozora omogućuje transmisiju više podatkovnih okvira, odjednom. Kontrola dodjeljuje svakom podatkovnom okviru *k-bit* brojevni niz i raspon brojevnog niza je $[0 \dots 2^k - 1]$. Ova kontrola radi tako što su dvije strane (pošiljatelj i primatelj), povezani *full-duplex* vezom. Primatelj dodjeljuje kapacitet međuspremnika za *n* okvira. Dakle, primatelj može prihvatiti *n* okvira, a pošiljatelju je dopušteno slanje *n* okvira bez čekanja na potvrde. Kako bi se pratilo koji su okviri primljeni, svaki je označen rednim brojem. Primatelj potvrđuje primitak podatkovnog okvira, slanjem potvrde koja uključuje redni broj očekivanog sljedećeg okvira. Ova potvrda također podrazumijeva da je primatelj spreman primiti sljedećih *n* okvira, počevši od navedenog broja, [6].

[5] *Komunikacija koja se može istovremeno odvijati u dva smjera.*

Na slikama 2 i 3 može se vidjeti niz okvira u klizajućoj kontroli toka, koje prikazuju dinamičko mijenjanje veličine prozora okvira koji se mogu poslati, sa stajališta pošiljatelja i primatelja.



Slika 2. Okviri koji se šalju sa stajališta pošiljatelja, [6]



Slika 3. Okviri koji se šalju sa stajališta primatelja, [6]

2.3. Kontrola pogreške

Kada se podatci prenose putem komunikacijskog kanala, uvijek postoji vjerojatnost da će se neki od bitova promijeniti (oštetiti) zbog šuma, izobličenja signala ili prigušenja. Ako se dogode pogreške, neki će se bitovi mijenjati s 0 na 1 ili s 1 na 0. Kontrola pogrešaka omogućuje primatelju da obavijesti pošiljatelja o bilo kojem podatkovnom okviru koji je izgubljen ili oštećen u prijenosu te koordinira ponovni prijenos tih okvira od strane pošiljatelja. Kontrola pogreške podijeljena je, prema [11], u dvije glavne kategorije:

- otkrivanje pogreške (engl. *error detection*) – dopušta primatelju provjeru primljenih podataka pa ako nisu vjerodostojni može se zatražiti retransmisija,

- ispravak pogreške (engl. *error correction*) – dopušta primatelju da rekonstruira originalnu informaciju, ako je bila iskvarena u procesu transmisije.

2.3.1. Otkrivanje pogreške

Nekoliko shema postoji kako bi se postignulo otkrivanje pogreške. Svi kodovi za otkrivanje pogreške (engl. *error detection code*), u koje spadaju svi kodovi za otkrivanje i ispravak pogreške, prenose više bita nego što je to bilo u originalnom podatku. Većina kodova djeluje sustavno; pošiljatelj šalje fiksan broj bita originalnog podatka, praćeni s fiksnim brojem bitova za provjeru (u literaturi ih se često naziva redundantnim), koji su izvedeni od podatkovnih bita od strane nekih deterministčkih algoritama, [12].

Primatelj primjenjuje isti algoritam na primljene podatke i uspoređuje izlaz s primljenim bitovima za provjeru; ako vrijednost ne odgovara, pogreška se dogodila na nekoj točki u procesu transmisije. U sustavima koji koriste ne-sustavne kodove, podatkovni bitovi se pretvaraju u što manje bitova koda pa pošiljatelj šalje samo te bitove, [12]. Načini na koje se može otkriti pogreška su opisani u sljedećim potpoglavljima.

2.3.1.1. Provjera pariteta

Najjednostavnija shema otkrivanja pogrešaka je dodavanje bita pariteta na kraj niza podataka. Tipičan primjer je ASCII prijenos u kojem je paritetni bit spojen sa svakim 7-bitnim ASCII znakom. Vrijednost ovog bita je odabrana tako da znak ima parni broj jedinica (parni paritet) ili neparni broj jedinica (neparni paritet), [11].

2.3.1.2. Shema ponavljanja

Shema ponavljanja (engl. *repetition scheme*) je način gdje se u strujanju podataka koji trebaju biti poslani događa raspodjela na blokove, te se svaki blok šalje neki predodređeni broj puta. Kako se blokovi šalju više puta, uspoređuju se isti blokovi međusobno. Ovaj način nije toliko efikasan te može biti podložan problemima ako se dogodi pogreška na istom bloku, više puta, ali je vrlo jednostavna, [12].

2.3.1.3. CRC

Složenije metode otkrivanja (i ispravljanja) pogreške koriste svojstva ograničenih polja i polinoma nad takvim poljima. CRC (engl. *Cyclic Redundancy Check*) smatra blok podataka koeficijentom polinoma, a zatim se dijeli fiksnim, unaprijed određenim polinomom. Koeficijent rezultata dijeljenja se uzima kao redundantni podatkovni bitovi, CRC. Nakon primitka, može se ponovno izračunati CRC, iz bitova u paketu koji predstavljaju podatke koji se žele prenijeti na prvom mjestu (engl. *payload*) i usporediti s tim CRC-om koji je primljen. Neslaganje ukazuje na to da se dogodila pogreška, [12].

2.3.1.4. *Checksum*

U ovom načinu otkrivanja pogreške, podatci se dijele u *k* segmenata, u svakom segmentu po *m* bitova. Na pošiljateljevoj strani, pomoću aritmetike komplementa jedinica, segmentu se dodaje vrijednost kako bi se dobila suma. Sume se zbrajaju kako bi se dobio *cheksum. Cheksum* se šalje zajedno sa segmentima, a na pošiljateljevoj strani svi primljeni segmenti dobivaju dodanu vrijednost aritmetikom komplementa jedinica te je suma upotpunjenja. U slučaju da je rezultat nula, podatci se prihvaćaju, u suprotnom se odbacuju, [13].

2.3.2. Ispravak pogreške

Prema [12], postoje dva osnovna načina za dizajniranje koda kanala i protokola za sustav ispravka pogreške:

- ARQ (engl. *Automatic-Repeat Request*) – pošiljatelj šalje podatke te kôd za otkrivanje pogrešaka, koji onda primatelj koristi da bi provjerio postojanje grešaka te zatražio retransmisiju podataka s pogreškom. U mnogo slučajeva zahtjev za ponovno slanje je implicitan; primatelj šalje ACK za podatke koji su ispravno primljeni, a pošiljatelj tada ponovno šalje one podatke za koje nije dobio ACK.
- FEC (engl. *Forward Error Correction*) – pošiljatelj dodaje dodatne bitove tako što kodira podatke po preddefiniranom algoritmu s ECC (engl. *Error-Corecting Code*) kodom te ih šalje. Primatelj analizira dodatne bitove te locira i ispravlja pogreške te se nakon toga dodatni bitovi brišu, a ostatak

predstavlja početni sadržaj. Kôd ne zahtijeva *handshake* pa je pogodan za *broadcast* (s jednog izvora na više odredišta) komunikaciju.

Moguće je kombinirati dva navedena načina, na način da se male pogreške isprave bez ponovnog slanja, a velike pogreške se otkriju i zatraži se ponovno slanje. Kombinacija se zove hibridni ARQ (engl. *hybrid Automatic-Repeat Request*).

2.4. Zaključna razmatranja o radu i ulozi čvora

Kroz prethodna potpoglavlja može se vidjeti koje su osnovne zadaće čvora u telekomunikacijskoj mreži. Sve te operacije zahtijevaju vrijeme, a kako podatci rijetko kada prolaze kroz samo jedan čvor, navedeno vrijeme može se samo povećati. Razvojem telekomunikacija, pojavile su se nove aplikacije i usluge koje pružaju informacije u stvarnom vremenu (strujanje videosadržaja, *instant* poruke, VoIP pozivi i dr.) te takve aplikacije i usluge ne mogu beskonačno „trpiti“ vrijeme potrebno da informacija stigne od izvorišta do odredišta, tj. kvaliteta takvih aplikacija biva narušena u slučaju da je vrijeme potrebno za prijenos podataka predugo.

Iz navedenih razloga, bitno je napraviti analizu posluživanja u čvorištima i mreži čvorišta, kako bi se došlo do zaključka u kakvom rasporedu i s kakvim kapacitetom mrežu treba dimenzionirati, da bi se dobilo optimalno vrijeme prijenosa podataka. Takvim zaključcima i analizama se bavi grana primijenjene teorije vjerojatnosti pod nazivom – podvorbeni sustavi. U sljedećim poglavljima opisati će se podvorbeni sustavi, ručno riješiti primjeri podvorbenih sustava te provesti simulacije na mreži čvorova, kako bi se izvršila analiza.

3. MATEMATIČKI OPIS PODVORBENIH SUSTAVA

Podvorbeni sustavi su grana primijenjene teorije vjerojatnosti koja se koristi za opisivanje specijaliziranijih matematičkih modela za redove čekanja. Oni koriste modele čekanja za predstavljanje različitih vrsta sustava čekanja koji se pojavljuju u praksi. Modeli omogućuju pronalaženje odgovarajuće ravnoteže između cijene usluge i vremena čekanja. Koncept teorije čekanja razvijen je velikim dijelom u kontekstu inženjeringa telefonskog prometa, koji je stvorio A. K. Erlang 1909. godine, [14].

Podvorbeni sustavi se sastoje od korisnika (koji traži uslugu/podvorbu, ne moraju biti ljudi), poslužitelja (koji obavlja posluživanje), repa čekanja (u kojemu se nalaze korisnici koji čekaju na posluživanje) te podvorbene stege (zbirka pravila po kojima se obavlja posluživanje). Primjeri podvorbenih sustava mogu se pronaći u svakodnevnom životu:

- niz elektroničkih pošti koje čekaju na procesiranje na poslužitelju,
- korisnici u redu u poštanskom uredu, koji čekaju uslugu,
- vozila koja čekaju na naplatnim postajama,
- zrakoplovi koji čekaju svoj red za slijetanje.

Kendall je 1953. godine predstavio oznaku koja opisuje karakteristike podvorbenog sustava, a oznaka izgleda:

$$A/B/C/K_q/K_i/P.S. \tag{1}$$

U Kendall-ovoj oznaci A predstavlja oznaku za razdiobu međudolaznih vremena korisnika, B oznaku za razdiobu trajanja posluživanja, C broj poslužitelja, K_q kapacitet sustava, K_i kapacitet izvora te $P.S.$ podvorbenu stegu (npr. prvi došao prvi poslužen, FCFS (engl. *First Come First Served*)). Razdiobe trajanja posluživanja te razdiobe međudolaznih vremena korisnika mogu biti razne (eksponencijalna, konstantna, opća, hipereksponencijalna itd.), [15].

Kako ovo poglavlje opisuje podvorbene sustave kojima je ulazni tok opisan Poisson-ovim procesom, tj. razdioba međudolaznih vremena i razdioba trajanja posluživanja ravna se po eksponencijalnoj razdiobi, opisivat će se samo takvi

podvorbeni sustavi. Takvi sustavi se još nazivaju i Poisson-ovski ili Markovljevi podvorbeni sustavi. Kendall-ova oznaka za te sustave uvijek započinje s M/M/…, jer slovo M u toj oznaci označava eksponencijalnu razdiobu.

Poisson-ov slučajni proces opisuje vjerojatnost dolazaka korisnika, koji se pojavljuju brzinom λ. Drugi naziv za takvu pojavu je rođenje, stoga nije čudno što se velik broj pojava u astronomiji, biologiji i medicini mogu opisati pomoću navedenoga procesa. Osim procesa dolazaka korisnika u sustava, Poisson-ov slučajni proces može opisivati i pojavu odlazaka korisnika iz sustava, tj. posluživanje korisnika koji se odvijaju brzinom μ pa se takva pojava može nazivati umiranjem. Dakle, slučajni proces rađanja i umiranja nastaje proširenjem Poisson-ovog slučajnog procesa, koji istodobno obuhvaća dolaske i odlaske korisnika iz sustava, uvjetujući da isti ovise o stanju podvorbenog sustava, [15].

U sljedećim potpoglavljima bit će prikazani izrazi za izračunavanje vjerojatnosti da se sustav nađe u nekom od stanja, izrazi za izračunavanje mjerila izvedbe i izrazi za izračunavanje razdiobe vremena čekanja i boravka u sustavu te njihovih prosječnih vrijednosti, a svi ti izrazi su preuzeti iz [1].

3.1. Jednoposlužiteljski Markovljevi podvorbeni sustavi s beskonačnim izvorištem

Jednoposlužiteljski Markovljevi podvorbeni sustavi s beskonačnim izvorištem koji će se opisati u ovom potpoglavlju bit će M/M/1/∞/∞/FCFS i M/M/1/K/∞/FCFS.

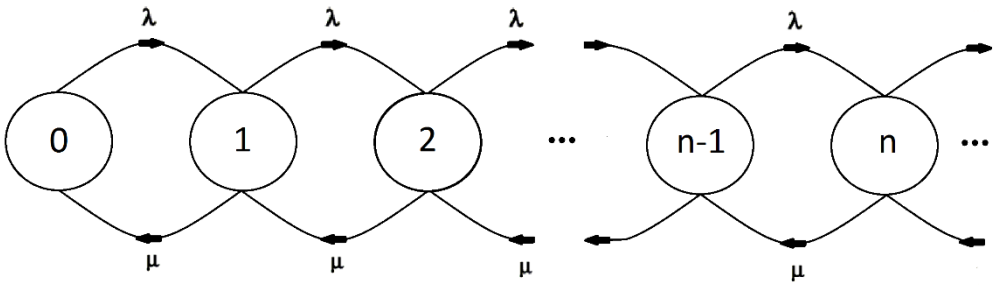
3.1.1. M/M/1/∞/∞/FCFS

Navedeni podvorbeni sustav opisuje sustav čekanja i posluživanja s međudolaznim vremenima i vremenima trajanja posluživanja koja se ravnaju po eksponencijalnoj razdiobi, s jednim poslužiteljem, neograničenom veličinom reda s *FCFS* disciplinom posluživanja i neograničenim izvorom. M/M/1/∞/∞/FCFS je jedan od najranijih sustava koji se analizira. Poslužiteljsko mjesto sadrži jednog poslužitelja, koji može posluživati jednog korisnika istovremeno. Ako poslužitelj nije zaposlen, korisnik se odmah poslužuje. Inače, korisnik koji dolazi, stiže u red čekanja. Taj će korisnik dobiti svoju uslugu kasnije, kada izađe iz reda čekanja. Kada poslužitelj dovrši posluživanje korisnika, korisnik odlazi, [14].

Za rješavanje Markovljevih podvorbenih sustava, prema [16], potrebna su tri koraka:

- izračunavanje razdiobe stanja sustava,
- izračunavanje mjerila izvedbe, koje se izračunavaju pomoću ove razdiobe (T_q – prosječno vrijeme zadržavanja u sustavu, T_w – prosječno vrijeme čekanja, L_q - prosječan broj korisnika u sustavu, L_w – prosječan broj korisnika u redu)
- izračunavanje razdiobe vremena čekanja i boravka u sustavu te njihovih prosječnih vrijednosti.

Kako se promatra samo stacionarno stanje, potrebno je poslužiti se gotovim rezultatima razmatranja stacionarnog stanja slučajnog procesa rađanja i umiranja, stoga je potrebno pronaći λ_n i μ_n, odnosno brzine nailazaka korisnika u sustav i brzinu posluživanja korisnika, [15]. Prijelaz među stanjima za ovaj sustav vidljiv je na slici 4 (stanja označavaju broj korisnika u sustavu).



Slika 4. Dijagram prijelaza među stanjima za M/M/1/∞/∞/FCFS

Prvi korak u rješavanju ovakvog modela je određivanje λ_n i μ_n, a za ovaj model vrijedi $\lambda_n = \lambda$; $\mu = \mu_n$, za sve *n*. Izraz koji povezuje dva navedena parametra te koji vrijedi uvijek, neovisno o modelu posluživanja, gdje a predstavlja ponuđeni promet, a T_s prosječno vrijeme posluživanja, glasi:

$$a = \lambda \cdot T_s = \frac{\lambda}{\mu}. \tag{2}$$

Izraz kojim će se izračunati vjerojatnost da je u sustavu *n* korisnika glasi:

$$Q_n = P(n_q = n) = a^n \cdot (1 - a). \quad (3)$$

Navedeni izraz je izraz za funkciju gustoće vjerojatnosti geometrijske slučajne varijable, drugim riječima n_q ima geometrijsku razdiobu s $p = 1 - a$ i $q = a$. Za model M/M/1/∞/∞/FCFS, Q_n predstavlja funkciju gustoće vjerojatnosti, [15].

Drugi korak u rješavanju ovog modela je određivanje mjera izvedbe, koje se izračunavaju pomoću razdiobe stanja, koja se računa u prvom koraku. Prosječan broj korisnika u sustavu, koji se označava s L_q, računa se sljedećim izrazom:

$$L_q = \frac{a}{1-a} = \frac{\lambda}{\mu-\lambda}. \quad (4)$$

Sljedeća mjera izvedbe koja je usko povezana s prethodnom je prosječna duljina reda, koja se označava s L_w, te izraz za tu mjeru glasi:

$$L_w = \frac{a^2}{1-a}. \quad (5)$$

Navedene dvije mjere izvedbe su povezane izrazom:

$$L_q = L_w + L_s \quad (6)$$

gdje L_s predstavlja očekivani broj korisnika na posluživanju te u zbroju s prosječnim brojem korisnika koji čeka L_w, predstavlja L_q, tj. prosječan broj korisnika u sustavu. Sljedeći izrazi predstavljaju vremena provedena u sustavu te ti izrazi vrijede za sve modele podvorbenih sustava. Navedeni izrazi su povezani s mjerama izvedbe čije su formule ranije prikazane (L_w, L_q). Prosječno vrijeme koje korisnik provede u redu, odnosno čekajući na posluživanje označava se s T_w te računa prema izrazu (7).

$$T_w = \frac{L_w}{\lambda}. \quad (7)$$

Prosječno vrijeme zadržavanja u sustavu, u koje spada vrijeme od trenutka kada korisnik stigne u red čekanja i sve dok ne izađe s poslužiteljskog mjesta, označava se s T_q, a izraz za računanje glasi:

$$T_q = \frac{L_q}{\lambda}. \quad (8)$$

Sljedeća veličina koja se analizira je prosječno vrijeme posluživanja, koje prođe od trenutka kada korisnik izađe iz repa na posluživanje te traje sve dok ne izađe s poslužiteljskog mjesta, a označava se s T_s. Izraz s kojim se može izračunati prosječno vrijeme posluživanja te povezuje dvije prethodne veličine glasi:

$$T_s = T_q - T_w. \quad (9)$$

Treći korak u analizi rješava se pomoću razdiobe vremena čekanja i boravka u sustavu. Izraz koji opisuje takvu razdiobu glasi:

$$Q(t) = P[\tau_q \leq t] = 1 - e^{-(\mu-\lambda)\cdot t}. \quad (10)$$

Iz navedene formule dobiva se izvod s kojim se u ovome sustavu može izračunati prosječno vrijeme zadržavanja korisnika u sustavu:

$$T_q = \frac{1}{\mu-\lambda}. \quad (11)$$

Kada se želi izračunati vjerojatnost da u sustavu nema reda, prvo se mora analizirati koji su to slučajevi kada u sustavu nema reda. Naime, u sustavu su moguće dvije situacije koje dovode do zaključka da u sustavu nema reda. Prva situacija je kada u sustavu uopće nema korisnika, a druga kada je u sustavu točno jedan korisnik pa je taj korisnik već na posluživanju, iz tog razloga u sustavu i u toj situaciji nema reda. Zbroj vjerojatnosti dviju navedenih situacija daje vjerojatnost da u sustavu nema reda:

$$W_0 = Q_0 + Q_1 \quad (12)$$

za ostale slučajeve općenito vrijedi:

$$W_i = Q_{i+1}. \quad (13)$$

3.1.2. M/M/1/K/∞/FCFS

M/M/1/K/∞/FCFS podvorbeni sustav je sustav u kojemu se međudolazna vremena korisnika i vremena posluživanja ravnaju po eksponencijalnoj razdiobi, jedan

poslužitelj obavlja posluživanje korisnika, podvorbena stega je *FCFS*, izvor je neograničen, a jedina razlika u odnosu na prethodno opisani podvorbeni sustav je taj da ovaj sustav ima ograničenje kapaciteta korisnika *K*. To znači da kada god je u sustavu *K* korisnika (1 na posluživanju, *K* - 1 u redu), svi dodatni pokušaji dolaska u sustav se odbijaju.

Takvi korisnici se nazivaju gubitak ili se za njih kaže da su blokirani. Ovaj podvorbeni sustav je primjer općeg sustava s gubitcima te se takvi sustavi s ograničenim brojem korisnika ponekad nazivaju i *truncated*, [16]. U ovakvom podvorbenom sustavu potrebno je razlikovati korisnike koji dolaze pred sustav od onih koji u njega stvarno ulaze. Oni koji dolaze pred sustav čine „brzinu dolazaka pred sustav λ, a oni koji ulaze čine brzinu ulaska u sustav λ' te se oni ponekad nazivaju stvarnom brzinom dolazaka“, [15]. Iz navedenog se može zaključiti da je brzina ulazaka u sustav zapravo onaj dio dolazećih korisnika koji ulaze u sustav i bivaju posluženi, [16].

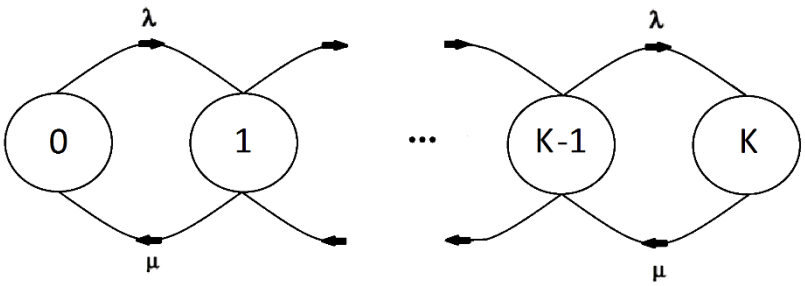
U takvom slučaju vrijedi:

$$\lambda' = \lambda(1 - Q_K) \tag{14}$$

i

$$a' = a(1 - Q_K). \tag{15}$$

Ograničenje λ_n za $n = K$ je ono što razlikuje ovaj sustav od prethodnog. Sustav je takav da stanje veće od *K* nikada neće biti postignuto, čak i u slučaju kada je $\lambda \geq \mu$, [17]. Na slici 5 prikazan je dijagram prijelaza među stanjima za M/M/1/K/∞/FCFS.



Slika 5. Dijagram prijelaza među stanjima za M/M/1/K/∞/FCFS

Postupak rješavanja ovakvog sustava odvija se pomoću slučajnog procesa rađanja i umiranja:

$$\lambda(n) = \begin{cases} \lambda, & n = 0,1,2,\dots(K-1) \\ 0, & n \geq K \end{cases}, \tag{16}$$

$$\mu(n) = \begin{cases} \mu; & n = 0,1,\dots K \\ 0; & n > K \end{cases}. \tag{17}$$

Izraz kojim će se izračunati vjerojatnost da je u sustavu *n* korisnika glasi:

$$Q_n = P(n_q = n) = \frac{(1-a)\cdot a^n}{1-a^{K+1}}. \tag{18}$$

U slučaju da je broj korisnika *n* jednak kapacitetu sustava K, zaključuje se da je Q_K vjerojatnost gubitka ovakvog sustava ili se može reći da je Q_K vjerojatnost da je spremnik kapaciteta (*K - 1*) pun, [16]. Ako je poznat Q_K može se izračunati K, tj. dimenzionirati kapacitet spremnika:

$$K = \frac{\ln[\frac{Q_K}{1-a(1-Q_K)}]}{\ln a}. \tag{19}$$

Prosječan broj korisnika u sustavu izračunava sljedećim izrazom:

$$L_q = \frac{a}{1-a} - \frac{(K+1)\cdot a^{K+1}}{1-a^{K+1}}. \tag{20}$$

Očekivani broj korisnika na posluživanju glasi:

$$L_s = a' = a(1-Q_K). \tag{21}$$

Prosječno vrijeme zadržavanja u sustavu:

$$T_q = \frac{L_q}{\lambda'} = \frac{L_q}{\lambda(1-Q_K)}. \tag{22}$$

Izraz za prosječno vrijeme čekanja korisnika na poslugu:

$$T_w = \frac{L_w}{\lambda'} = \frac{L_w}{\lambda(1-Q_K)}. \tag{23}$$

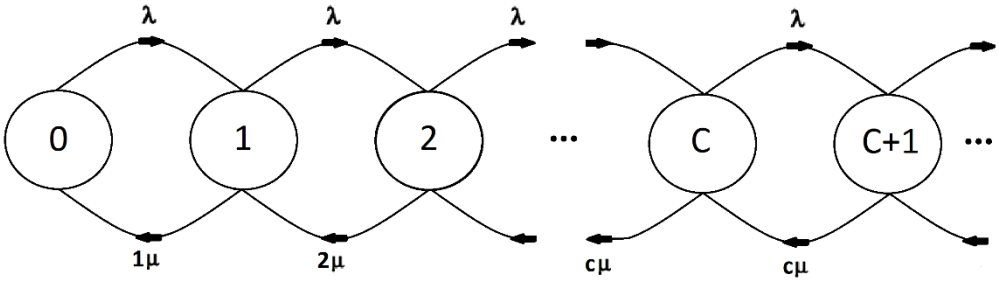
3.2. Višeposlužiteljski Markovljevi podvorbeni sustavi s beskonačnim izvorištem

Višeposlužiteljski podvorbeni sustavi korisni su u modeliranju telekomunikacijskih sustava. Uobičajeno je da u takvim sustavima postoji nekoliko kanala koji se koriste za komunikaciju. Takvi kanali se smatraju paralelnim poslužiteljima u podvorbenom sustavu, [18]. Kroz ovo poglavlje opisati će se dva modela podvorbenih sustava, koji su višeposlužiteljski. Ti modeli su M/M/C/∞/∞/FCFS i M/M/C/C/∞/FCFS.

3.2.1. M/M/C/∞/∞/FCFS

Ovaj podvorbeni sustav kao i oni opisani u prethodnom poglavlju, ima svojstvo da se međudolazna vremena i vremena posluživanja ravnaju po eksponencijalnoj razdiobi. Razlika u odnosu na prethodne modele je ta da postoji c poslužitelja koji poslužuju korisnike koji se nalaze u jednom redu. Primjer takvog sustava može se pronaći na šalterima u banci, gdje u slučaju da u sustavu ima manje korisnika nego poslužitelja, svi korisnici su na posluživanju. Suprotno navedenome, ako u sustavu ima više korisnika nego poslužitelja, broj korisnika u redu se dobije na način da se oduzme broj poslužitelja od broja korisnika u sustavu, [17].

Svojstva ovog modela su da poslužitelj uvijek poslužuje maksimalno jednog korisnika i da prilikom dolaska, korisnici neposredno ulaze najbližem slobodnom poslužitelju. U ovakvom sustavu ne postoji mogućnost da jedan poslužitelj poslužuje više korisnika. Pravilo izlaska iz repa i u ovom modelu je *FCFS*. U dijagramu stanja ovog modela beskonačno mnogo je stanja, s obzirom da je red neograničen (vidljivo na slici 6). Najveća brzina posluživanja događa se u onom trenutku kada su svi poslužitelji zauzeti, tj. kada je u sustavu broj korisnika jednak broju poslužitelja ili veći, [16].



Slika 6. Dijagram prijelaza među stanjima za M/M/C/∞/∞/FCFS

Iz razloga što u ovome modelu ima više poslužitelja, ponuđeni promet razlikuje se od jakosti prometa, stoga izraz za jakost prometa u ovom modelu glasi:

$$\rho = \frac{a}{c} = \frac{\lambda}{c\mu}. \tag{24}$$

Ovakav sustav nema gubitaka, stoga obavljeni promet a, jednak je ponuđenom a', a jakost prometa ρ, jednak je faktoru iskoristivosti ρ', [15]. Izraz kojim se računa vjerojatnost da je u sustavu *n* korisnika, dijeli se na dva slučaja, a oni glase:

$$Q_n = P(n_q = n) = \begin{cases} \frac{a^n}{n!} \cdot Q_0;\ n = 0,1,2,\dots(c-1) \\ \frac{a^n}{c! \cdot c^{n-c}} \cdot Q_0;\ n \geq c \end{cases}. \tag{25}$$

U prethodnom izrazu može se primijetiti pojavljivanje vjerojatnosti da je sustav prazan, odnosno da u sustavu nema korisnika Q_0. Ta veličina računa se na način da se dobije njezina inverzna vrijednost, koji glasi:

$$Q_0^{-1} = \sum_{n=0}^{c-1} \frac{a^n}{n!} + \frac{a^c}{c!(1-\rho)}. \tag{26}$$

Važna izlazna veličina u ovom modelu svakako je vjerojatnost čekanja. Budući da ovakav sustav ima c poslužitelja, čekanje na podvorbu u ovom sustavu dogoditi će se samo u slučaju da je u sustavu c ili više korisnika, stoga će se zbrojem tih dviju vjerojatnosti, dobiti vjerojatnost čekanja, [16]. Izraz za vjerojatnost čekanja u ovom sustavu još se naziva druga Erlang-ova formula, a ona glasi:

$$P(n_q \geq c) = E_{2,c}(a) = \sum_{j=c}^{\infty} Q_j = \frac{\frac{a^c}{c!} \cdot Q_0}{1-\rho} = \frac{\frac{a^c}{c!}}{(1-\rho) \cdot \sum_{j=0}^{c-1} \frac{a^j}{j!} + \frac{a^c}{c!}}. \tag{27}$$

Važnost navedene izlazne veličine očituje se u tome što se i ostale izlazne veličine izračunavaju pomoću nje:

$$L_w = \frac{\rho \cdot E_{2,c}\,(a)}{1-\rho}, \tag{28}$$

$$L_q = L_w + a, \tag{29}$$

$$T_w = \frac{L_w}{\lambda} = \frac{E_{2,c}\,(a) \cdot T_s}{c(1-\rho)}, \tag{30}$$

$$T_q = \frac{L_q}{\lambda} = T_s + \frac{E_{2,c}\,(a) \cdot T_s}{c(1-\rho)}. \tag{31}$$

Funkcija distribucije vremena čekanja u ovome modelu dijeli se na dva izraza. U prvom izrazu računa se vjerojatnost da korisnik čeka manje ili jednako određenom vremenu, a u drugom izrazu se računa vjerojatnost da korisnik čeka duže od nekog vremena. Izrazi za te dvije veličine glase:

$$W_t = P[\tau_w \leq t] = 1 - E_{2,c}(a) \cdot e^{-\mu \cdot t \cdot (c-a)}, \tag{32}$$

$$W_t = P[\tau_w > t] = E_{2,c}(a) \cdot e^{-\mu \cdot t \cdot (c-a)}. \tag{33}$$

Posljednja izlazna veličina u ovome modelu je T_d, odnosno prosječno vrijeme čekanja korisnika koji stvarno čekaju:

$$T_d = \frac{T_w}{E_{2,c}(a)} = \frac{T_s}{c(1-\rho)}. \tag{34}$$

3.2.2. M/M/C/C/∞/FCFS

Ovakav podvorbeni sustav se razlikuje od prethodnog u tome što ne postoji mogućnost stvaranja repa. Ovakav podvorbeni sustav je temeljni podvorbeni sustav s gubitcima. Podvorbeni sustav s gubitcima je sustav u kojemu u trenutku dolaska pred sustav, korisnik nailazi na sve poslužitelje zauzete, uključujući i rep čekanja te nepovratno odlaze iz sustava. Korisnici koji nailaze na sve poslužitelje zauzete nazivaju se blokiranim. Potrebno je razlikovati temeljni od općeg sustava s gubitcima, koji se spominjao u jednom od prethodnih potpoglavlja, [15].

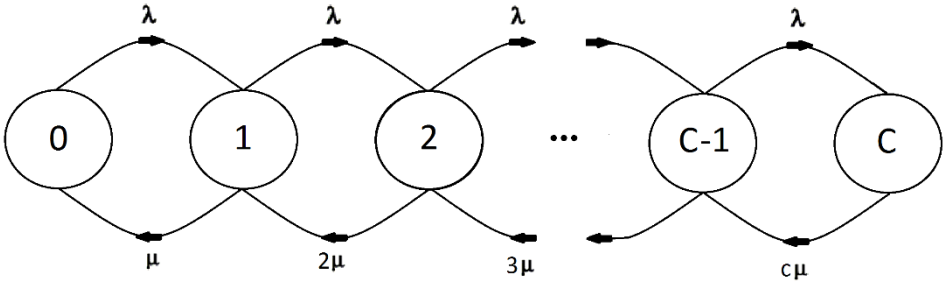
Naime, u općem sustavu s gubitcima, u sustavu gdje postoji vremenski ili prostorno ograničeni rep, blokirani su oni korisnici koji nailaze na sve poslužitelje

zauzete, ali mogu stati u rep. Gubitak u takvom sustavu je onaj korisnik koja nailazi na popunjene i poslužitelje i rep sustava, koji je na neki način ograničen. U temeljnom sustavu s gubitcima, blokirani korisnici predstavljaju gubitak, iz razloga što kada bivaju blokirani (jer su svi poslužitelji zauzeti), nema repa u kojemu se može čekati, [15].

Ovaj sustav ima c paralelnih poslužitelja, kapacitet izvora je neograničen, podvorbena stega je *FCFS*, a kapacitet sustava iznosi c (kao što se vidi na slici 7), što znači da nema mogućnosti stvaranja repa, kao što je već navedeno. Ulazne veličine ovog modela su:

$$A(t) = 1 - e^{-\lambda t} \rightarrow T_a = \frac{1}{\lambda}, \tag{35}$$

$$B(t) = 1 - e^{-\mu t} \rightarrow T_s = \frac{1}{\mu}. \tag{36}$$



Slika 7. Dijagram prijelaza među stanjima za M/M/C/C/∞/FCFS

Vjerojatnost da je u sustavu *n* korisnika u ovom modelu računa se preko izraza:

$$Cn = \frac{a^n}{n!} \rightarrow Q_n = \frac{\frac{a^n}{n!}}{\sum_{j=0}^{c} \frac{a!}{j!}}. \tag{37}$$

U ovom sustavu bitna je Erlang-ova formula za gubitke, jer se preko nje računa obavljeni promet. Navedena formula je vjerojatnost da je u sustavu točno *c* korisnika, te taj izraz glasi, [16]:

$$E_{1,c}(a) = P[n_q = c] = \frac{\frac{a^c}{c!}}{\sum_{j=0}^{c} \frac{a!}{j!}} \tag{38}$$

Koristeći taj izraz, može se izračunati obavljeni promet:

$$a' = a(1 - E_{1,c}(a)). \tag{39}$$

Iz razloga što u ovom sustavu nema čekanja, izlazne veličine L_w i T_w jednake su nuli. Broj korisnika u sustavu jednak je obavljenom prometu, a vrijeme provedeno u sustavu jednako je vremenu posluživanja:

$$L_q = a', \tag{40}$$

$$T_q = T_s = \frac{L_q}{\lambda'} \tag{41}$$

gdje λ' predstavlja brzinu ulaska u sustava, a izraz za računanje glasi:

$$\lambda' = \lambda(1 - E_{1,c}(a)). \tag{42}$$

Treba napomenuti da je Erlang-ova formula za gubitke $E_{1,c}(a)$ jednaka vjerojatnosti da su svi poslužitelji zauzeti te da će dolazeći korisnik doći pred sustav u trenutku kada su svi zauzeti. Navedene dvije vjerojatnosti su jednake samo kada se radi o Poisson-ovim slučajnim procesima, [16].

4. IZRAČUN PERFORMANSI SUSTAVA NA IZOLIRANOM DIJELU MREŽE

Kako bi se dobio osjećaj za veličine te da opis podvorbenih sustava prije simulacija (koje su analizirane u sljedećem poglavlju) ne bude samo zasnovan na teoriji i izrazima, bez numeričkih veličina, u ovome poglavlju će se izračunati mjerila izvedbe za elementarni Markovljev podvorbeni sustav, tj. M/M/1/∞/∞/FCFS. Za zadatak će se uzeti proizvoljan primjer u kojemu će biti zadane određene veličine, a pojašnjavati će se, korak po korak, način na koji se dolazi do određenih rezultata.

U ovom primjeru poznata je veličina ponuđenog prometa $a = 0{,}6\ Erl$ te intenzitet posluživanja $\mu = 150\ korisnika/h$. Odredit će se vjerojatnost da je podvorbeni sustav u nekom od stanja. Odredit će se vjerojatnosti za $0 \leq n_q \leq 5$. Izraz je već napisan u prethodnom poglavlju pa izračun da je u sustavu 0 korisnika glasi:

$$Q_0 = P(n_q = 0) = a^n \cdot (1 - a) = 0{,}6^0 \cdot (1 - 0{,}6) = 0{,}4\ ili\ 40\%.$$

Vjerojatnost da je u sustavu 1 korisnik:

$$Q_1 = P(n_q = 1) = a^n \cdot (1 - a) = 0{,}6^1 \cdot (1 - 0{,}6) = 0{,}24\ ili\ 24\%.$$

Vjerojatnost da su u sustavu 2 korisnika:

$$Q_2 = P(n_q = 2) = a^n \cdot (1 - a) = 0{,}6^2 \cdot (1 - 0{,}6) = 0{,}144\ ili\ 14{,}4\%.$$

Vjerojatnost da su u sustavu 3 korisnika:

$$Q_3 = P(n_q = 3) = a^n \cdot (1 - a) = 0{,}6^3 \cdot (1 - 0{,}6) = 0{,}0864\ ili\ 8{,}64\%.$$

Vjerojatnost da su u sustavu 4 korisnika:

$$Q_4 = P(n_q = 4) = a^n \cdot (1 - a) = 0{,}6^4 \cdot (1 - 0{,}6) = 0{,}05184\ ili\ 5.184\%.$$

Vjerojatnost da je u sustavu 5 korisnika:

$$Q_5 = P(n_q = 5) = a^n \cdot (1 - a) = 0{,}6^5 \cdot (1 - 0{,}6) = 0{,}031104\ ili\ 3{,}1104\%.$$

Iz navedenih rezultata jasno se može zaključiti da je sustav u većini slučajeva prazan ili je u sustavu jedan korisnik (64%), uz ovakav ponuđeni promet. U slučaju da se u ovakvom modelu želi izračunati da je u sustavu maksimalno 2 korisnika, zbrojile bi se vjerojatnosti da je sustav u nekom od stanja manjem od 2 korisnika, uključujući i to stanje:

$$P(n_q \leq 2) = \sum_{i=0}^{2} Q_{ni} = Q_0 + Q_1 + Q_2 = 0{,}4 + 0{,}24 + 0{,}144 = 0{,}784\ ili\ 78{,}4\%.$$

Kada se želi izračunati vjerojatnost da je u sustavu više od 2 korisnika, koristi se suprotna vjerojatnost:

$$P(n_q > 2) = \sum_{i=3}^{\infty} Q_{ni} = 1 - P(n_q \leq 2) = 1 - 0{,}784 = 0{,}216\ ili\ 21{,}6\%.$$

Ako se želi izračunati prosječni broj korisnika u sustavu L_q, moguće je do rješenja doći na dva načina. Prvi način je uz poznavanje veličine prometa:

$$L_q = \frac{a}{1-a} = \frac{0.6}{1-0{,}6} = 1{,}5\ korisnik.$$

Drugi način računanja je uz poznate podatke o intenzitetu dolazaka i intenzitetu posluživanja, a budući da u ovom slučaju nema zadanog intenziteta posluživanja, izračunati će se po izrazu:

$$\lambda = a \cdot \mu = 0{,}6 \cdot 150 = 90\ kor/h$$

te sada kada je poznata veličina intenziteta posluživanja, može se na drugi način izračunati prosječni broj korisnika u sustavu:

$$L_q = \frac{\lambda}{\mu-\lambda} = \frac{90}{150-90} = 1{,}5\ korisnik.$$

Standardno odstupanje σ, za broj korisnika u sustavu računa se na sljedeći način:

$$\sigma = \sqrt{Var\{n_q\}} = \frac{\sqrt{a}}{1-a} = \frac{\sqrt{0{,}6}}{1-0{,}6} = 1{,}9365\ korisnika.$$

Izračun za vjerojatnost da u sustavu nema reda, po već prethodno opisanom izrazu i već poznatim veličinama:

$$W_0 = Q_0 + Q_1 = 0{,}4 + 0{,}24 = 0{,}64\ ili\ 64\%$$

te s takvom poznatom vjerojatnosti da u sustavu nema reda, pomoću suprotne vjerojatnosti može se izračunati da u sustavu postoji red, gdje n_w predstavlja veličinu reda:

$$P(n_w > 0) = 1 - P(n_w = 0) = 1 - 0{,}64 = 0{,}36\ ili\ 36\%.$$

U drugačijim situacijama, kada se ne traži ima li ili nema reda u sustavu, tj. kada se traži da je u redu točno određeni broj korisnika (npr. 4), izračun glasi:

$$W_i = Q_{i+1} \rightarrow W_4 = Q_5 = 0{,}031104\ ili\ 3{,}1104\%$$

a u slučajevima kada se računa da u redu sustava ima najviše 2 korisnika:

$$P(n_w \leq 2) = P(n_w = 0) + P(n_w = 1) + P(n_w = 2) =$$

$$= W_0(Q_0 + Q_1) + W_1(Q_2) + W_2(Q_3) = 0{,}64 + 0{,}144 + 0{,}0864 =$$

$$= 0{,}8704\ ili\ 87{,}04\%.$$

Prosječna duljina reda L_w za ovaj model i ovakav ponuđeni promet:

$$L_w = \frac{a^2}{1-a} = \frac{0{,}6^2}{1-0{,}6} = 0{,}9\ korisnika.$$

Sljedeća vjerojatnost koja će se izračunati je vjerojatnost da korisnik u sustavu boravi najviše 2 minute:

$$Q(t) = P\left[\tau_q \leq t\right] = 1 - e^{-(\mu-\lambda)\cdot t} \rightarrow Q(t = 2\ min) =$$

$$= 1 - e^{-(150-90)\cdot\frac{2}{60}} = 0{,}8647\ ili\ 86{,}47\%.$$

Prosječno vrijeme zadržavanja korisnika T_q u sustavu računa se sljedećim izrazom te će za ovaj primjer iznositi:

$$T_q = E\{\tau_q\} = \frac{1}{\mu-\lambda} = \frac{1}{150-90} = 0{,}01667\ h\ \rightarrow 1\ min\ \rightarrow 60\ s.$$

Prosječno vrijeme čekanja na posluživanje, odnosno prosječno vrijeme provedeno u redu, može se izračunati preko prosječne duljine reda:

$$T_w = \frac{L_w}{\lambda} = \frac{0,9}{90} = 0,01\ h\ \rightarrow 0,6\ min\ \rightarrow 36\ s$$

a prosječno vrijeme posluživanja može se izračunati preko prethodne dvije veličine:

$$T_s = T_q - T_w = 0,01667 - 0,01 = 0,00667 \rightarrow 0,4\ min\ \rightarrow 24\ s.$$

Vjerojatnost da će korisnik pri ulasku u sustav naići na slobodan poslužitelj može se izračunati na dva načina. Prvi način je izračun preko vjerojatnosti da u sustavu nema korisnika:

$$Q_0 = P(n_q = 0) = a^n \cdot (1 - a) = 0,6^0 \cdot (1 - 0,6) = 0,4\ ili\ 40\%$$

a drugi način izračunom vjerojatnosti da korisnik neće čekati:

$$W(t) = P[\tau_w \leq t] = 1 - a \cdot e^{-\mu \cdot (1-a) \cdot t} \rightarrow W(0) = 1 - 0,6 \cdot e^{-150 \cdot (1-0,6) \cdot 0} =$$

$$= 1 - a = 1 - 0,6 = 0,4\ ili\ 40\%.$$

Prosječan broj korisnika koji će biti poslužen tijekom vremena zaposlenosti:

$$L_c = E\{n_c\} = \frac{1}{1-a} = \frac{1}{1-0,6} = 2,5\ korisnika.$$

Izračun gornje granice čekanja za 85% korisnika rješava se pomoću funkcije distribucije vremena čekanja:

$$W(t) = 1 - a \cdot e^{-\mu \cdot (1-a) \cdot t} \rightarrow$$

$$1 - a \cdot e^{-\mu (1-a) \cdot tx} = 0,85 \rightarrow$$

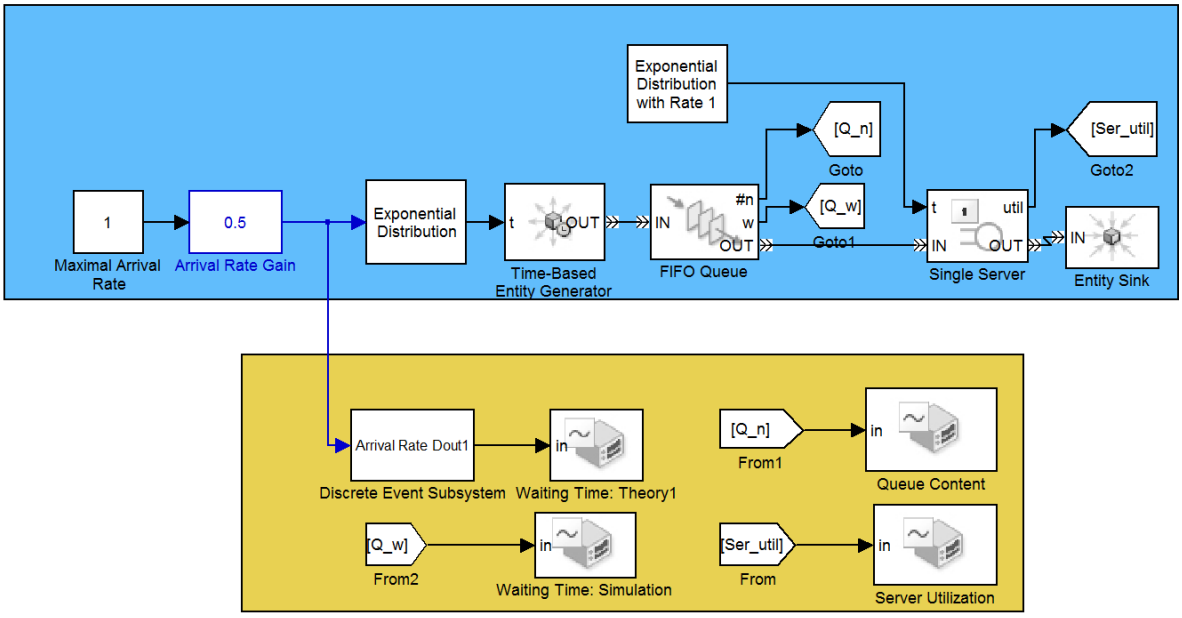
$$e^{-\mu (1-a) \cdot tx} = \frac{0,15}{a} \rightarrow$$

$$-\mu (1 - a) \cdot tx = \ln \frac{0,15}{a} \rightarrow$$

$$tx = \frac{\ln \frac{0,15}{a}}{-\mu (1-a)} = \frac{\ln \frac{0,15}{0,6}}{-150 (1-0,6)} = 0,223\ h \rightarrow 1,39\ min.$$

5. PRIKAZ RAČUNALNOG MODELA

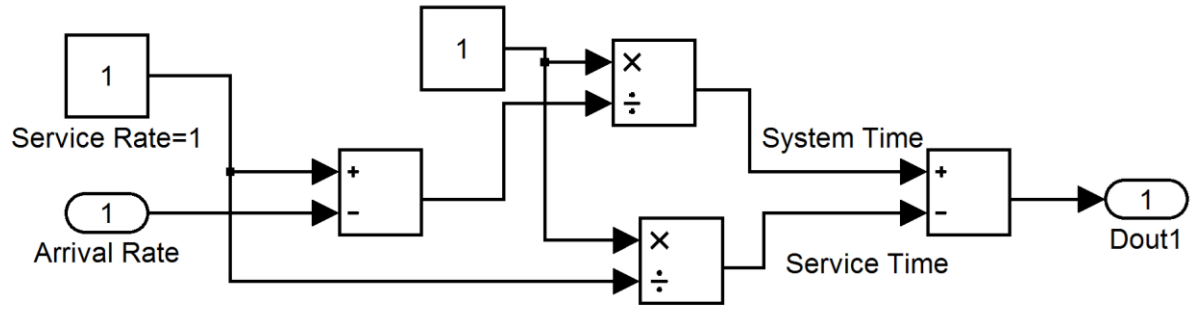
Kako bi se dodatno analizirali elementarni Markovljevi podvorbeni sustavi, provest će se simulacija takvih sustava u računalnom alatu MatLab, odnosno njegovom programskom okruženju za modeliranje Simulink-u. Simulink već ima preddefinirani M/M/1/∞/∞/FCFS model te je izgled takvog modela prikazan na slici 8.



Slika 8. Prikaz sheme preddefiniranog računalnog modela za elementarni Markovljev podvorbeni sustav u Simulink-u

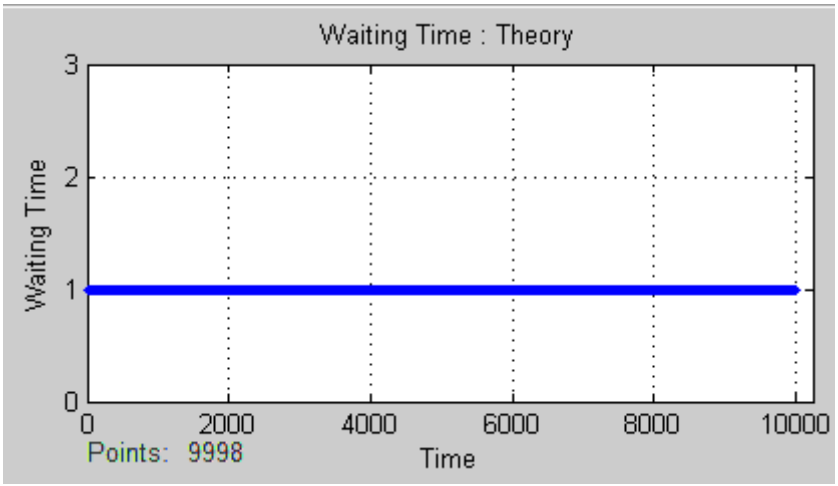
Svaki od prikazanih objekata bit će pojašnjen, a rezultati simulacije i usporedbe simulacija (s različitim brojem čvorova) biti će obrazložene u sljedećem poglavlju.

Prva dva objekta *Maximal Arrival Rate* i *Arrival Rate Gain* u umnošku daju intenzitet nailaska korisnika u sustav. Njihov izlaz predstavlja ulaz za dva objekta. Prvi objekt kojemu umnožak *Maximal Arrival Rate-a* i *Arrival Rate Gain-a* predstavlja ulaz je objekt *Discrete Event Subsystem,* koji zapravo predstavlja podsustav koji teoretski računa vrijeme čekanja u sustavu. Na slici 9 može se vidjeti unutarnja konfiguracija navedenog podsustava te kako računa vrijeme čekanja, u teoriji.



Slika 9. *Discrete Event Subsystem* - konfiguracija

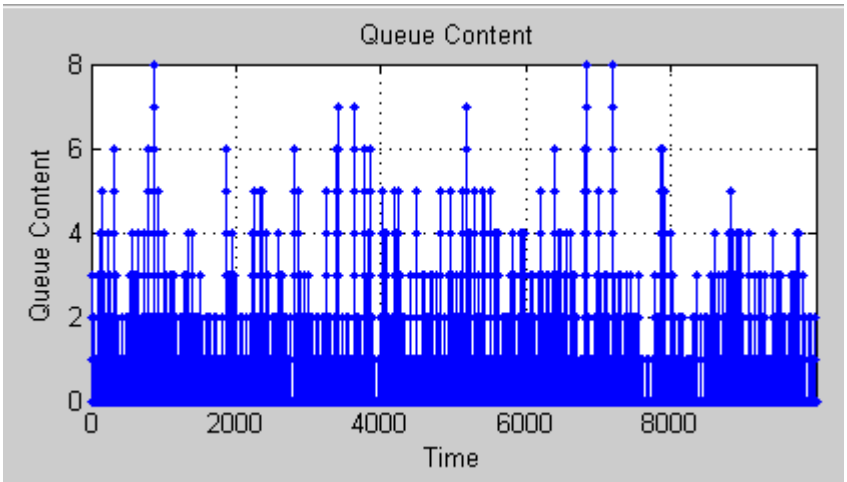
Sljedeći objekt kojemu izlaz ovog podsustava predstavlja ulaz je objekt *Waiting Time: Theory1*, koji prilikom simulacije grafički prikazuje teoretsko vrijeme čekanja. Na slici 10 može se vidjeti grafički prikaz teoretskog rezultata vremena čekanja, neke jednostavne simulacije.



Slika 10. Grafički prikaz rezultata simulacije za teoretsko vrijeme čekanja entiteta u Simulink-u

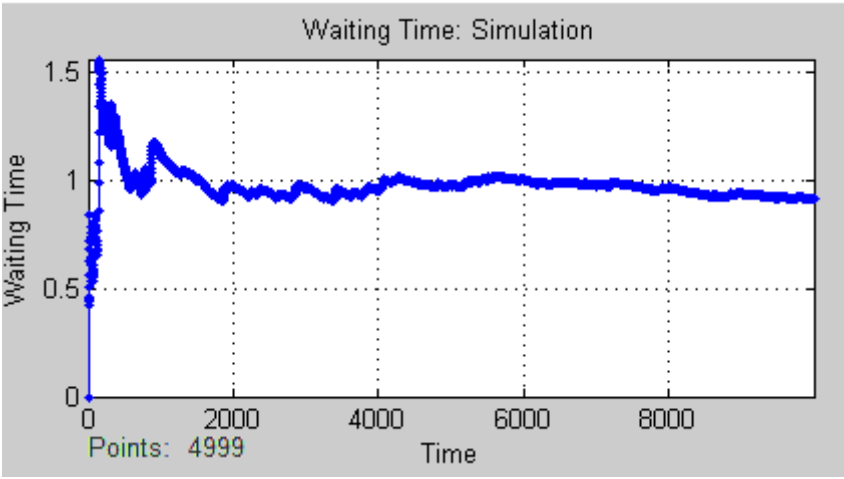
Drugi objekt kojemu je umnožak *Maximal Arrival Rate-a* i *Arrival Rate Gain-a* ulaz, je objekt *Exponential Distribution*, koji uz takav ulaz generira brojeve, koji odgovaraju intenzitetu nailaska jednakom umnošku *Maximal Arrival Rate-a i Arrival Rate Gain-a,* a da se ti brojevi ravnaju po eksponencijalnoj razdiobi. Takvi brojevi ulaze u sljedeći blok pod nazivom *Time-Based Entity Generator*, koji na temelju tih vremenskih zakonitosti, generira korisnike čija su međudolazna vremena eksponencijalna te se ti korisnici spremaju pristupiti sustavu.

Sljedeći objekt kojemu navedeni korisnici predstavljaju ulaz je sami red čekanja *FIFO Queue*, koji uz jedan ulaz ima i tri izlaza. Podvorbena stega može se iščitati iz samog naziva, a ona je prvi došao, prvi poslužen. Prvi izlaz pod oznakom *#n* predstavlja broj entiteta koji se nalaze u redu te se on šalje na sljedeći objekt naziva *Goto*, koji se potom preko objekta *From1* šalje do *Queue Content* kako bi se grafički prikazao broj korisnika u sustavu, u određenom trenutku. Na slici 11 može se vidjeti grafički prikaz rezultata simulacije, za broj korisnika u sustavu u nekom trenutku.



Slika 11. Grafički prikaz rezultata simulacije za broj korisnika u sustavu u Simulink-u

Drugi izlaz iz istog objekta je pod oznakom *w*, koji predstavlja simulacijsko vrijeme koje je korisnik čekao na posluživanje. Simulacijsko vrijeme čekanja na posluživanje se preko objekata *Goto1* i *From2* šalje do objekta *Waiting Time: Simulation*, koje grafički prikazuje rezultate, kao što je vidljivo na slici 12.

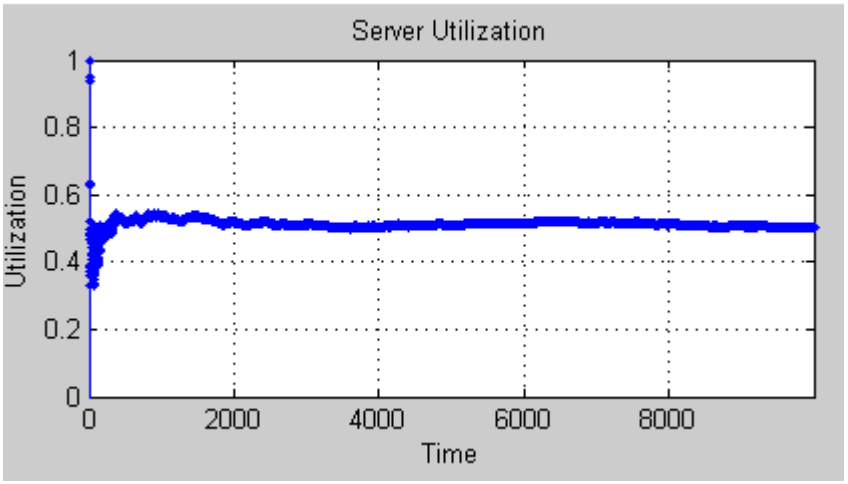


Slika 12. Grafički prikaz rezultata simulacije za vrijeme čekanja u Simulink-u

Treći, ujedno i zadnji izlaz iz *FIFO Queue* objekta je izlaz pod oznakom *OUT,* koji šalje korisnike u objekt *Single Server*, koji je zapravo i jedini poslužitelj u ovome podvorbenom sustavu. Treba napomenuti kako se u postavkama izlazi mogu dodati ili isključiti iz uporabe, a takve preinake sustava kao i neke složenije će se odraditi u sljedećem poglavlju.

Single Server objekt koji ima ulogu poslužitelja u ovom podvorenom sustavu, prima parametre iz dva ulaza i rezultate posluživanja šalje na dva izlaza. Prvi ulaz je već naveden (*OUT* iz *FIFO Queue-a*), a drugi je objekt *Exponential distribution with Rate 1,* koji predstavlja vrijeme posluživanja s eksponencijalnom razdiobom i s prosječnim vremenom posluživanja jednakim 1, koji se može dodatno prilagođavati.

Prvi od dva izlaza (*util*) predstavlja ulaz za objekt *Goto2* te se te vrijednosti preko objekta *From* šalju do objekta *Server Utilization,* koji je zapravo grafički prikaz iskorištenosti poslužitelja, kao što se vidi na slici 13. Drugi izlaz je označen s *OUT* te predstavlja ulaz *za Entity Sink,* koji ima mogućnost prihvata ili blokiranja korisnika, ali navedeni objekt u ovoj konfiguraciji nema bitnu funkciju te bi sustav mogao raditi i bez njega. *OUT* izlaz je jako bitan, jer će u simulaciji s više čvorova on predstavljati ulaz za sljedeći čvor.



Slika 13. Grafički prikaz rezultata simulacije za iskorištenost poslužitelja u Simulink-u

6. ANALIZA PERFORMANSI MREŽE U OVISNOSTI O NJENOJ VELIČINI

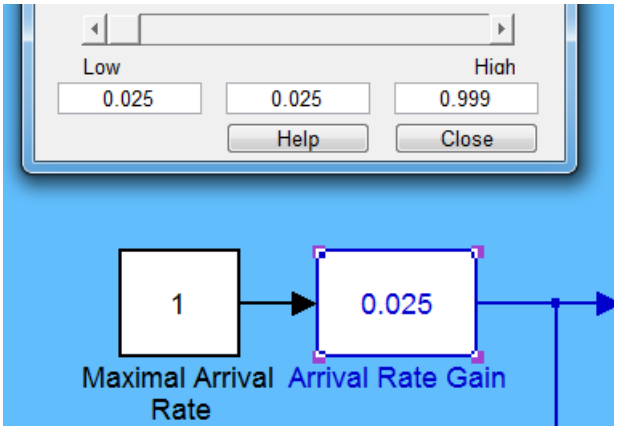
U ovome poglavlju, koje zapravo predstavlja i najbitniji dio ovog diplomskog rada, bit će prikazane konfiguracije i analize mreže s dva i tri čvora. Uz navedeno, na primjeru će se usporediti rezultati ručno izračunatih zadataka, s rezultatima iz simulacije, kako bi se potvrdila vjerodostojnost simulacijskog alata i njegovih rezultata.

6.1. Usporedba rezultata ručno izračunatih zadataka s rezultatima simulacije

Kako bi se usporedili rezultati, iskoristiti će se rješenja iz četvrtog poglavlja. Simulacija će se konfigurirati s već poznatim parametrima te će se rezultati prikazati grafički. Prvi korak je podešavanje intenziteta nailaska korisnika u sustav. Kako je već spomenuto, prva dva objekta *Maximal Arrival Rate* i *Arrival Rate Gain* u umnošku daju intenzitet nailaska korisnika na sustav. Intenzitet nailaska je već izračunat te on iznosi $\lambda = 90\ kor/h$. Kako simulacija po preddefiniranim postavkama omogućuje postavljanje vrijednosti *Maximal Arrival Rate-a* i *Arrival Rate Gain-a* samo između nula i jedan, intenzitet će se pretvoriti u nailaske po sekundi:

$$\lambda = \frac{90\ kor/h}{3600} = 0{,}025\ kor/s.$$

Na slici 14 vidljivo je podešavanje intenziteta nailaska korisnika u sustav u simulaciji te da umnožak dvaju objekata iznosi $0{,}025$.

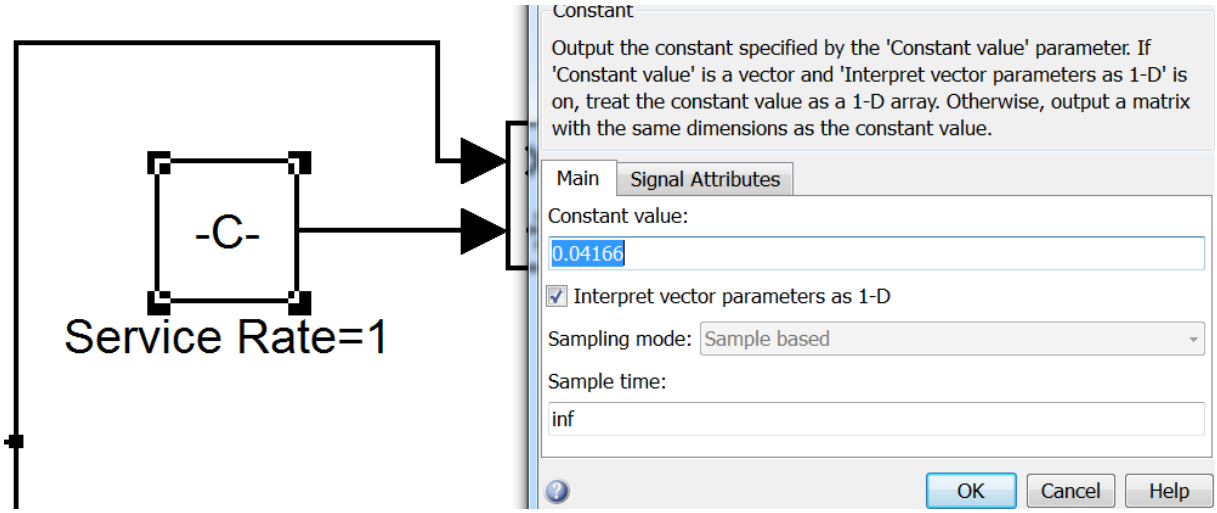


Slika 14. Podešavanje intenziteta nailaska korisnika na sustav u Simulink-u

Drugi korak je podešavanje prosječnog vremena posluživanja korisnika u sustavu, a taj je postupak malo složeniji. Naime, nije moguće direktno podesiti prosječno vrijeme posluživanja, ali je moguće podesiti *service rate* (intenzitet posluživanja) u objektu *Exponential distribution with Rate 1.* Kako je poznato da je prosječno vrijeme posluživanja T_s obrnuto proporcionalno intenzitetu posluživanja β, može se na taj način izračunati i podesiti (prosječno vrijeme posluživanja T_s je izračunato u četvrtom poglavlju):

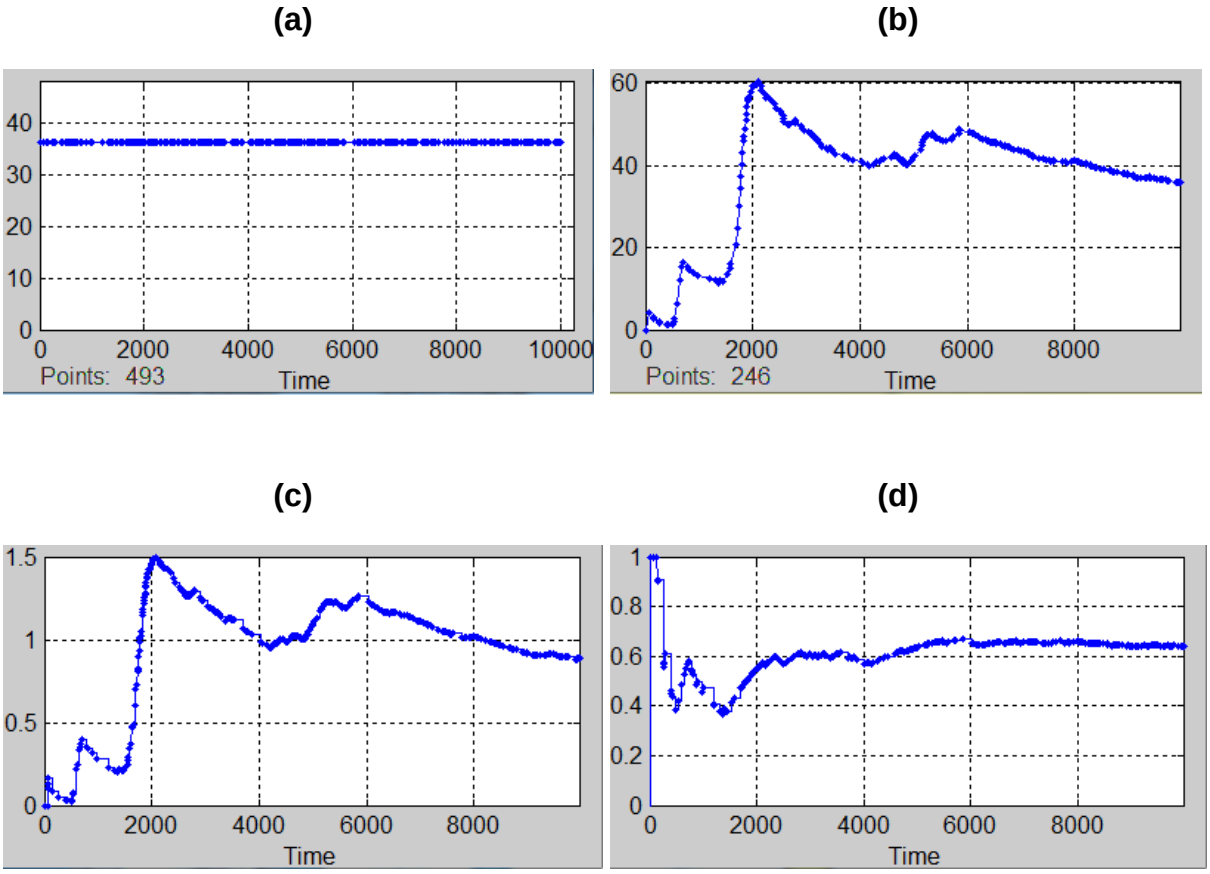
$$\beta = \frac{1}{T_s} = \frac{1}{24} = 0{,}04166\ s^{-1}.$$

Desnim klikom na objekt *Exponential distribution with Rate 1* te odabirom opcije iz izbornika *Look Under Mask,* podešava se intenzitet posluživanja na odgovarajuću vrijednost, kao što je vidljivo na slici 15.



Slika 15. Podešavanje intenziteta posluživanja u Simulink-u

Treći korak je podešavanje intenziteta posluživanja na objektu *Discrete Event Subsystem* te je postupak isti kao na prethodnom objektu. Prije samog pokretanja simulacije treba napomenuti da su napravljene preinake u odnosu na preddefinirane postavke te da je isključen izlaz za *Queue Content* i da je ubačen *Average Queue Length*. Na slici 16 vidljivi su izlazni rezultati simulacije koji odgovaraju izračunatim vrijednostima. Treba napomenuti da *server utilization* (jakost prometa) u ovom modelu (M/M/1) ima istu vrijednost kao i promet, koji u zadatku u četvrtom poglavlju iznosi $a = 0{,}6\ Erl$.



Slika 16. Rješenja simulacije u Simulink-u: **a** vrijeme čekanja (teoretsko), **b** vrijeme čekanja (simulacija), **c** prosječna duljina reda, **d** jakost prometa

Ovo potpoglavlje je bitno iz razloga što omogućuje nastavak rada u kojemu će se analize mreže temeljiti na simulacijama, koje su teško izvedive ručno, a ovi rezultati potvrđuju vjerodostojnost simulacije.

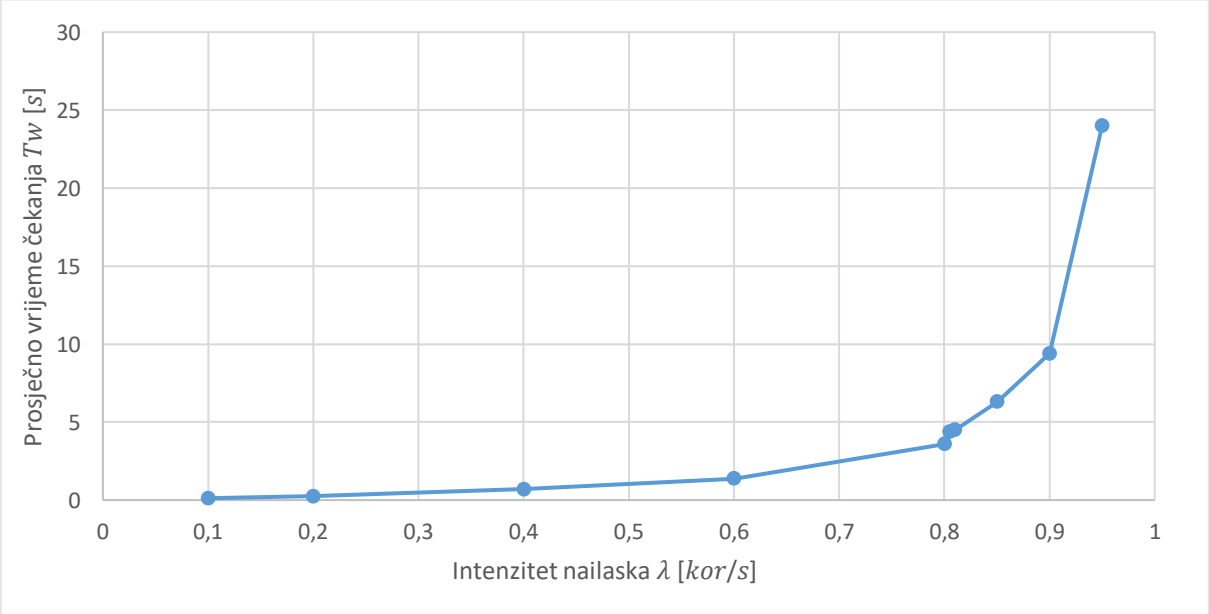
6.2. Analiza performansi mreže s dva čvora

U ovom potpoglavlju analizira se mreža s dva čvora. Mreža je podešena na način da izlaz prvog čvora predstavlja ulaz drugom čvoru. U odnosu na preddefinirani model dodani su objekti *FIFO Queue 2* i *Single Server 2*, koji predstavljaju red čekanja i poslužitelj na drugom čvoru. Dobro je napomenuti kako se ovdje radi o serijski spojenim poslužiteljima, a ne paralelnim, što znači da će svaki korisnik morati biti poslužen na oba poslužitelja, bez prava biranja kojemu će poslužitelju prvo pristupiti.

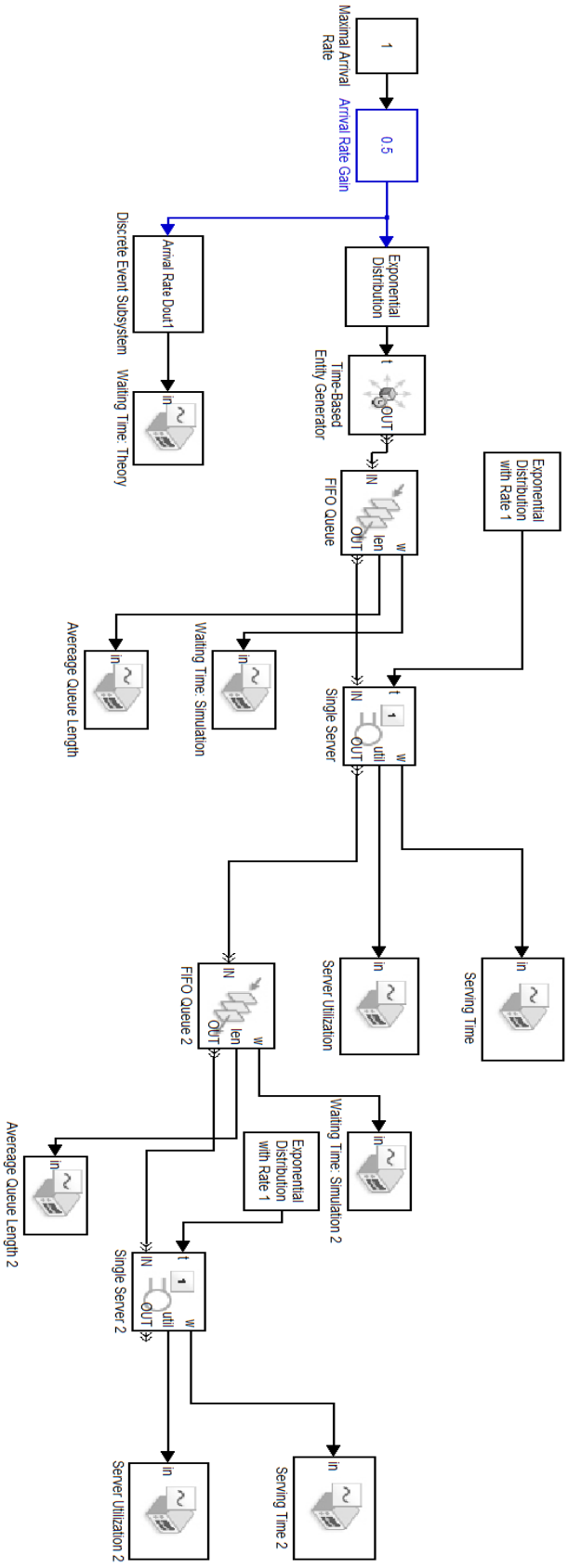
Ostali objekti koji su dodani su *Waiting Time: Simulation 2, Serving Time 2, Average Queue Lentgth 2* i *Server Utilization 2* koji služe za grafički prikaz rješenja simulacije te *Exponential Distribution With Rate 1* koji služi da poslužitelj poslužuje korisnike po zakonitostima eksponencijalne razdiobe. Sve promjene i shema modela je vidljiva na slici 17.

Mreža će se analizirati na način da će se mijenjati intenzitet nailaska korisnika te će se donositi zaključci na temelju rezultata. Isto tako, mreža će se analizirati i promjenom prosječnog vremena čekanja kako na prvom, tako i na drugom čvoru.

Pokretanjem simulacije s različitim intenzitetima dolazaka korisnika λ i istim vremenom posluživanja T_s, dobio se niz rezultata koji su vidljivi na grafikonima 1 i 2. Na grafikonu 1 može se vidjeti kako prosječno vrijeme čekanja na posluživanje na prvom čvoru raste s porastom intenziteta nailaska. Treba napomenuti kako je jakost prometa jednaka intenzitetu nailaska iz razloga što je vrijeme posluživanja $T_s = 1\ s$.

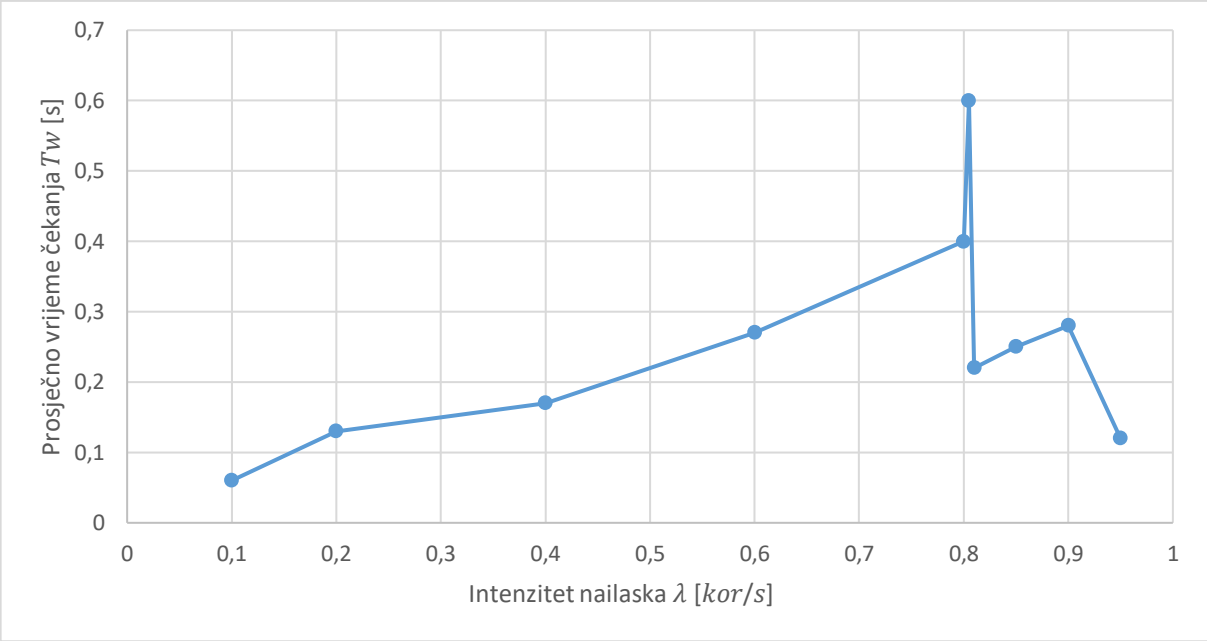


Grafikon 1. Prosječno vrijeme čekanja na posluživanje na prvom poslužitelju



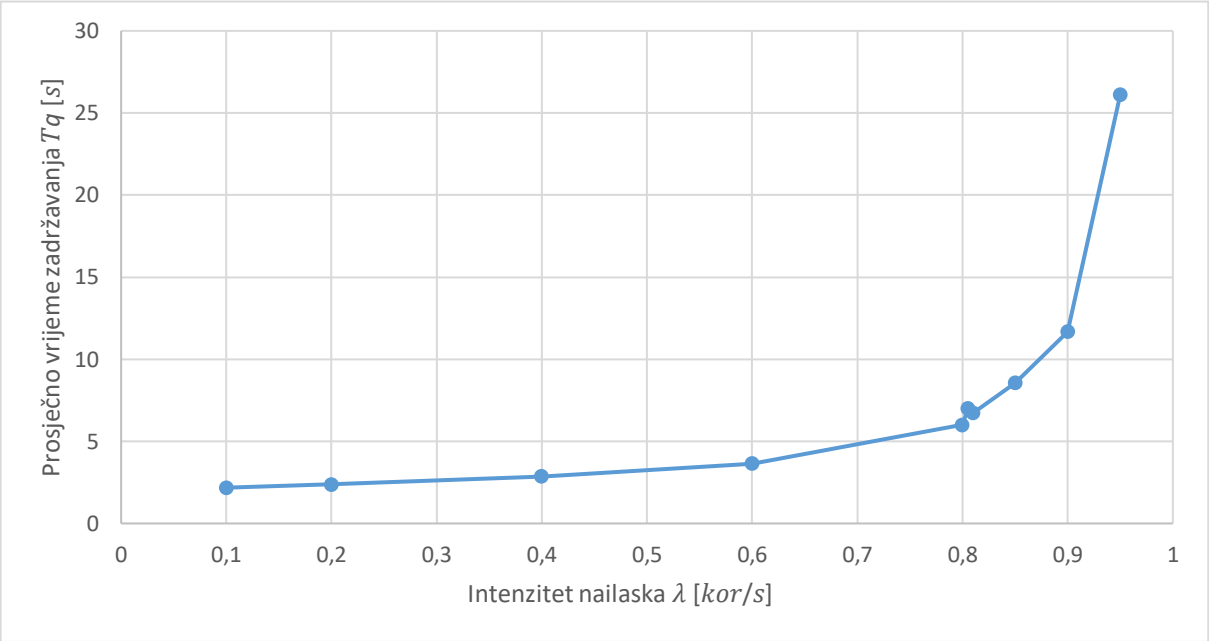
Slika 17. Prikaz sheme računalnog modela za mrežu s dva čvora u Simulink-u

Na grafikonu 2 se vidi kako pri najvećem porastu prosječnog vremena čekanja na prvom poslužitelju, prosječno vrijeme čekanja na drugom poslužitelju znatno se smanjuje. Takva situacija je i logična iz razloga što prvi čvor toliko dugo zadržava korisnike u sustavu da intenzitet nailaska na drugom čvoru postaje vrlo mali. Izuzetak je kada intenzitet nailaska iznosi $\lambda = 0{,}805\ kor/s$ te u tom slučaju prosječno vrijeme čekanja na drugom poslužitelju doseže svoj vrhunac nakon čega pada.



Grafikon 2. Prosječno vrijeme čekanja na posluživanje na drugom poslužitelju

Iz ovakve simulacije i rezultata dolazi se do zaključka da u mreži s dva serijska poslužitelja, koja poslužuju po vremenskim zakonitostima eksponencijalne razdiobe, intenzitet nailaska, odnosno jakost prometa ne bi trebala prelaziti $\rho = 0{,}81$; iz razloga što prosječno vrijeme čekanja u redu počinje naglo rasti i sustav postaje prezasićen, kao što je vidljivo na grafikonu 3, koji prikazuje ovisnost prosječnog vremena zadržavanja korisnika u cijelom sustavu o intenzitetu nailaska korisnika.



Grafikon 3. Prosječno vrijeme zadržavanja korisnika u cijeloj mreži s dva čvora

U sljedećoj simulaciji mijenjaju se parametri vremena posluživanja na prvom i drugom poslužitelju. Vrijeme posluživanja mijenjat će se na način da će se odrediti dvije brzine posluživanja, odnosno dva vremena posluživanja te će nakon provedene simulacije zamijeniti mjesta, odnosno vrijeme koje je bilo podešeno na prvom poslužitelju bit će na drugom i obratno.

Na ovakav pristup simulaciji može se doći do zaključka koji poslužitelj treba biti na kojem mjestu u seriji posluživanja, gdje je jedan poslužitelj „sporiji“ od drugoga, kako bi se sustav optimizirao. U tablici 2 prikazana su vremena koja korisnik provede u sustavu, u odnosu na vrijeme posluživanja.

Tablica 2. Prosječno vrijeme provedeno u sustavu za korisnika u mreži s dva čvora, s različitim vremenima čekanja

	T_s na prvom poslužitelju $[s]$	T_s na drugom poslužitelju $[s]$	T_w na prvom poslužitelju $[s]$	T_W na drugom poslužitelju $[s]$	T_q za cijeli sustav $[s]$
Simulacija 1.	1,5	1	4,3	0,08	6,88
Simulacija 2.	1	1,5	1,1	4,3	7,9
Simulacija 3.	2	0,5	122	0	124,5
Simulacija 4.	0,5	2	0,17	220	222,67
Simulacija 5.	1	0,5	1,08	0,022	2,602
Simulacija 6.	0,5	1	0,175	0,93	2,605
Simulacija 7.	0,5	0,33	0,17	0,02	1,02
Simulacija 8.	0,33	0,5	0,07	0,15	1,05

U navedenoj tablici je provedeno osam simulacija, gdje se T_s izmjenjivao između poslužitelja. U prvim simulacija (neparnim), „sporiji“ poslužitelj bi uvijek bio prvi u seriji posluživanja te bi takav raspored davao bolje rezultate, odnosno ukupno prosječno vrijeme zadržavanja u sustavu za korisnika bi iznosilo manje, nego u slučaju kada je „sporiji“ poslužitelj na drugom mjestu. Iako je ovo premali broj simulacija da bi se došlo do sigurnog zaključka da je ovakva konfiguracija uvijek bolja, sigurno predstavlja veću vjerojatnost da će sustav biti optimiziraniji, ako daljnja istraživanja nisu moguća.

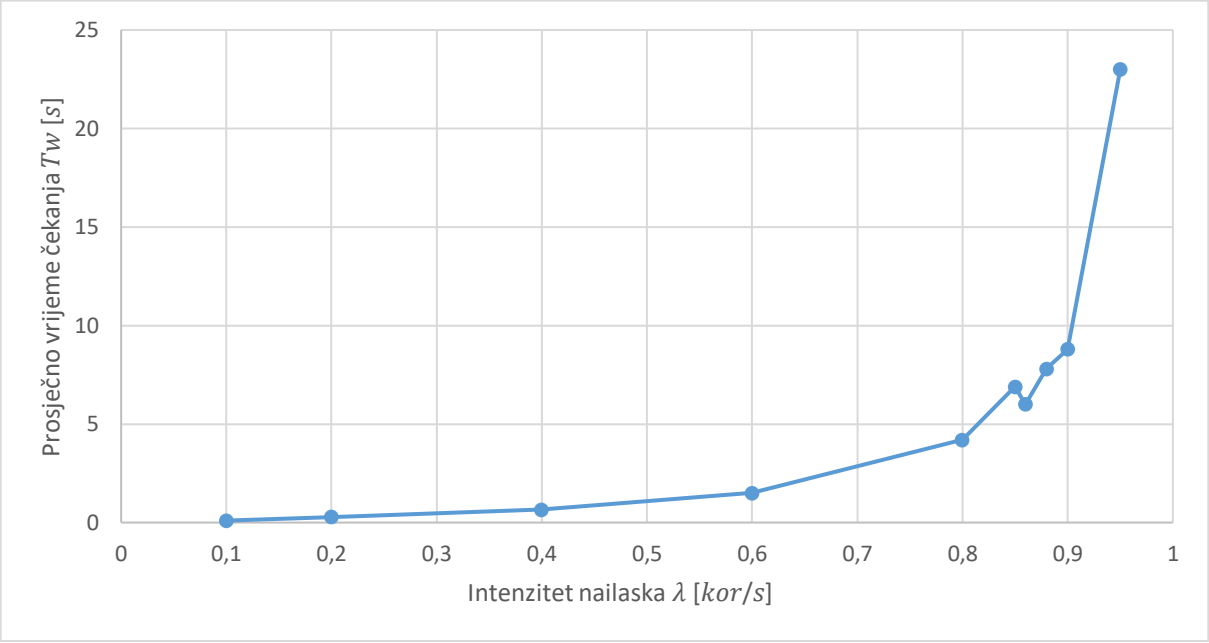
6.3. Analiza performansi mreže s tri čvora

Analiza performansi mreže s tri čvora provesti će se na sličan način kao i mreža s dva čvora, samo što se u ovom slučaju mijenja shema računalnog modela, dodajući mreži još jedan čvor u odnosu na mrežu s dva čvora. Objekti koji su dodani su isti kao

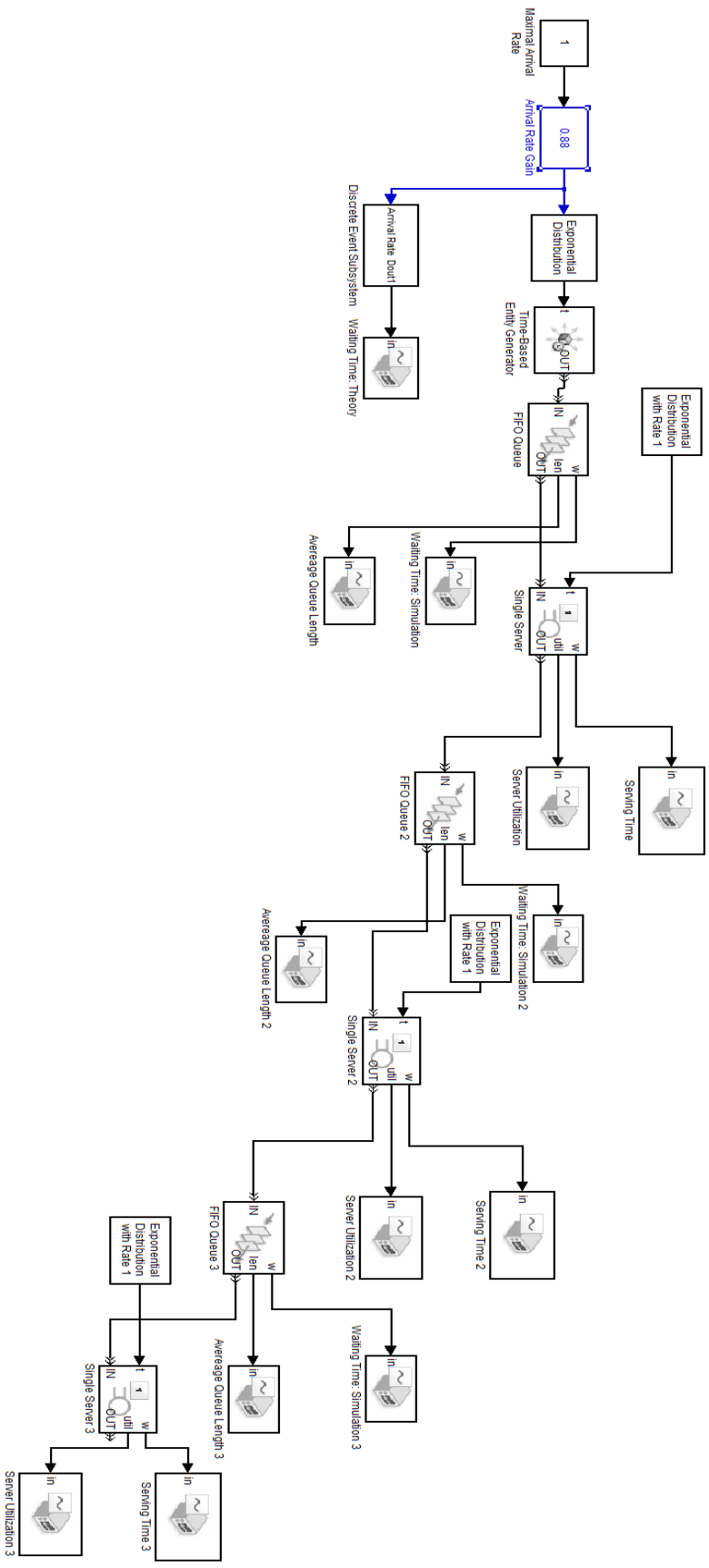
i pri dodavanju objekata za drugi čvor, stoga se neće ponovno navoditi i u ovom potpoglavlju. U ovoj shemi opet se radi o serijskoj izvedbi mreže te svaki korisnik mora proći sve redove i poslužitelje. Shema je vidljiva na slici 18.

Mreža će se analizirati na isti način kao i u prethodnom slučaju, kada se mreža sastojala od dva čvora. Pokretanjem simulacije s različitim intenzitetima posluživanja λ i istim vremenom posluživanja T_s, dobio se niz rezultata koji su vidljivi na grafikonima 4, 5, 6 i 7. I u ovome slučaju je jakost prometa jednaka intenzitetu nailaska iz razloga što je vrijeme posluživanja $T_s = 1\ s$.

Na grafikonu 4 prikazana je ovisnost prosječnog vremena čekanja na prvom poslužitelju o intenzitetu nailaska te se ponavlja zakonitost iz prethodnog poglavlja, gdje vrijeme provedeno u redu za prvi poslužitelj raste kako raste intenzitet nailaska, odnosno jakost prometa raste. Izuzetak je na jakosti prometa $\rho = 0{,}86$; gdje prosječno vrijeme čekanja pada, pa nakon toga opet raste, kao što je vidljivo na grafikonu 4.

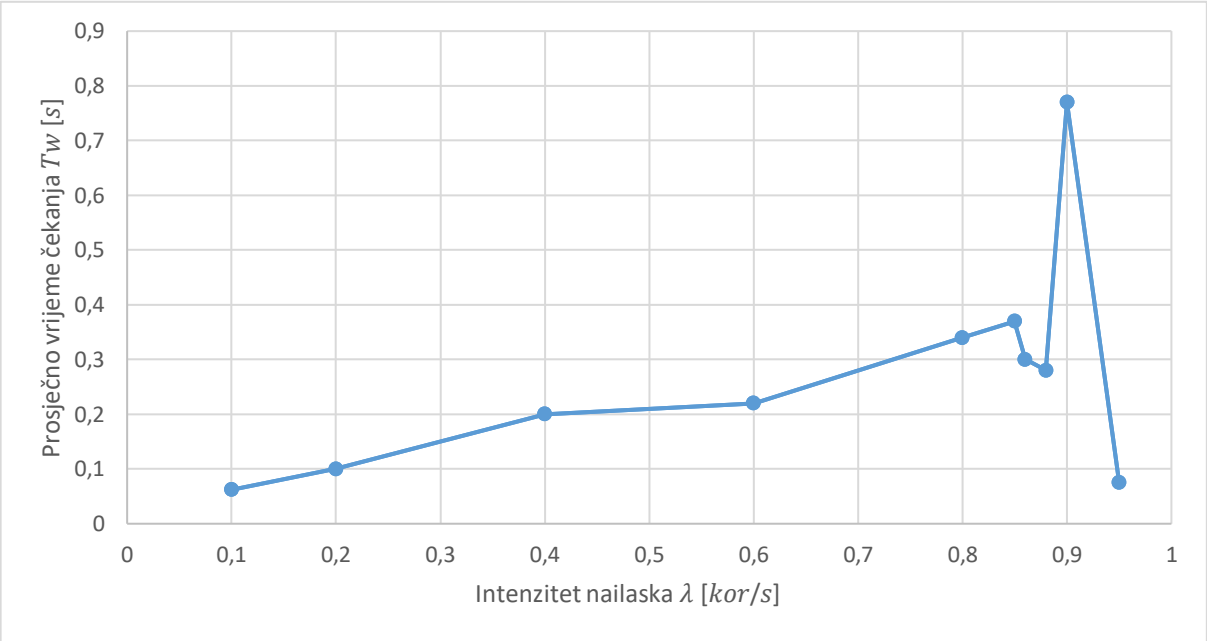


Grafikon 4. Prosječno vrijeme čekanja na posluživanje na prvom poslužitelju



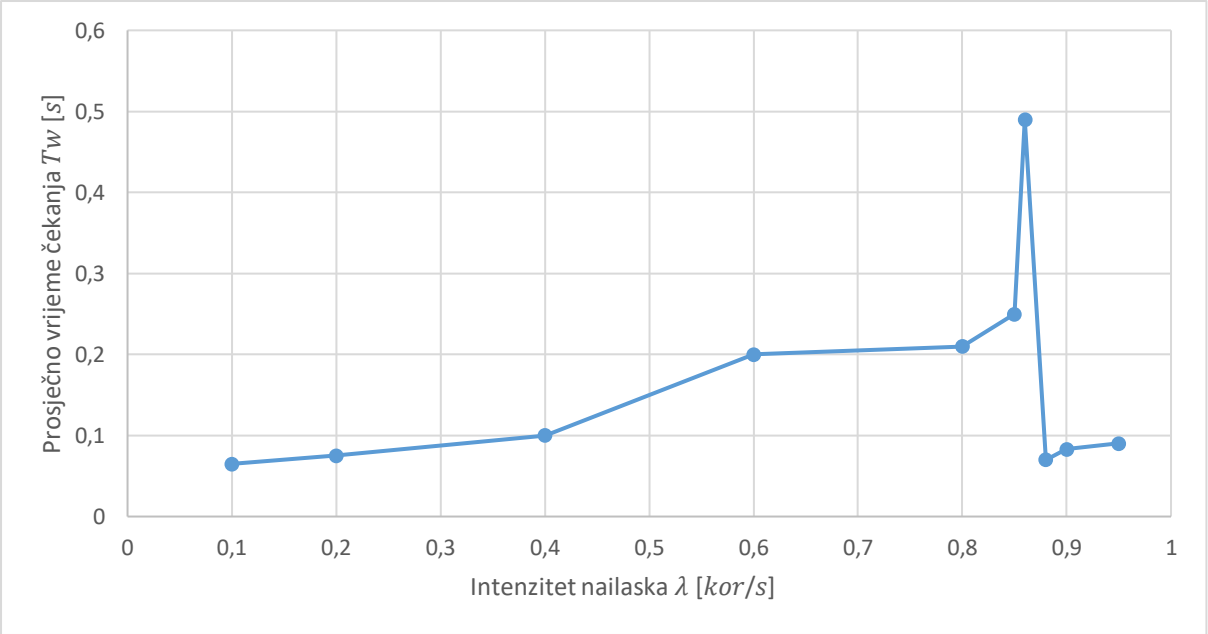
Slika 18. Prikaz sheme računalnog modela za mrežu s tri čvora u Simulink-u

Na grafikonu 5 vidi se ovisnost prosječnog vremena čekanja na posluživanje u drugom čvoru o intenzitetu nailaska korisnika i jakosti prometa. Prosječno vrijeme čekanja raste sve do $\rho = 0{,}85$; kada se počinje smanjivati te se smanjuje sve do $\rho = 0{,}88$; kada opet počinje rasti te se dostiže vrhunac vremena čekanja na $\rho = 0{,}9$; nakon čega znatno pada.



Grafikon 5. Prosječno vrijeme čekanja na posluživanje na drugom poslužitelju

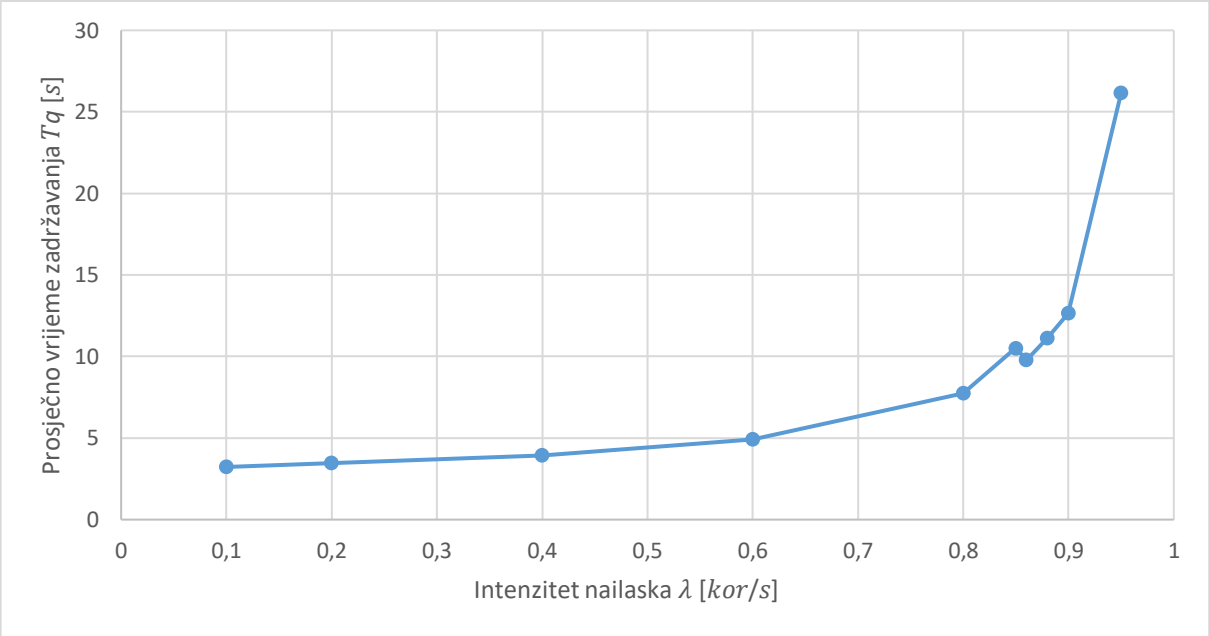
Na grafikonu 6 prikazana su vremena čekanja za treći čvor, koji je najzanimljiviji iz razloga što u prethodnom potpoglavlju nije bilo trećeg čvora. Ono što je razlika u ovom čvoru u odnosu na prva dva, je to da se u prethodna dva prije dosezanja vrhunca vremena čekanja dogodio pad, dok u ovome, trećem čvoru toga nema. Ovaj čvor bilježi rast vremena čekanja cijelo vrijeme dok ne dođe do maksimuma, nakon čega pada na najniže vrijednosti i na takvim približnim vrijednostima se zadržava.



Grafikon 6. Prosječno vrijeme čekanja na posluživanje na trećem poslužitelju

Na grafikonu 7 prikazano je ukupno prosječno vrijeme zadržavanja korisnika u sustavu. Krivulja je slična kao i u mreži s dva čvora te obje bilježe nagli rast kada jakost prometa prijeđe $\rho = 0{,}8$; s izuzetkom da kod mreže s dva čvora, blagi pad se događa pri jakosti prometa $\rho = 0{,}81$; nakon čega opet raste, a kod mreže s tri čvora taj se pad događa pri jakosti prometa $\rho = 0{,}86$; nakon čega opet raste.

Treba napomenuti i da pri opterećenjima mreže kada jakost prometa prelazi $\rho = 0{,}9$; mreža s tri čvora skoro pa i ne zadržava korisnike više nego mreža s dva čvora, a u situaciji $\rho = 0{,}95$; razlika je zanemarivo mala. Iz ovakvih rezultata dolazi se do zaključka da pri visokim opterećenjima dodatni poslužitelj u seriji neće znatno utjecati na produljenje boravka korisnika u sustavu, ako je vrijeme posluživanja jednako na svim poslužiteljima.



Grafikon 7. Prosječno vrijeme zadržavanja korisnika u cijeloj mreži s tri čvora

U sljedećim simulacijama mijenjaju se parametri vremena posluživanja na prvom, drugom i trećem poslužitelju. U prvih šest simulacija mijenja se vrijeme posluživanja kao i u mreži s dva čvora, tj. mijenja se tako da se odrede tri vremena posluživanja i simuliraju sve moguće kombinacije, tako da se pokuša doći do zaključka kakav raspored odgovara mreži s tri čvora, odnosno koji raspored daje najbolje rezultate.

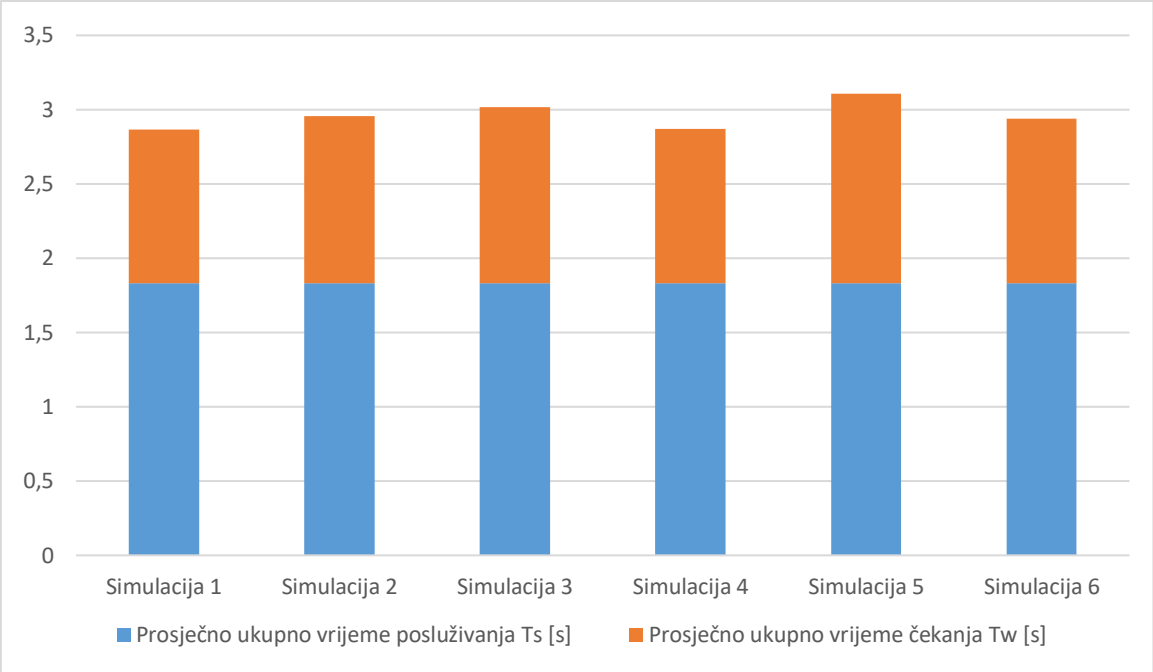
Drugih šest simulacija (od 7. do 12. simulacije) provode se na način da dva poslužitelja imaju isto vrijeme posluživanja, a jedan ima nižu vrijednost vremena posluživanja (od 7. do 9. simulacije) ili dva poslužitelja imaju isto vrijeme posluživanja, a jedan ima višu vrijednost vremena posluživanja (od 10. do 12. simulacije).

U tablici 3 vidljiva su prosječna vremena zadržavanja korisnika u sustavu. Ono što je zanimljivo za prvih šest simulacija (gdje su sva vremena posluživanja različita), je da su dvije simulacije s najkraćim vremenom bile one, u kojima je najkraće vrijeme posluživanja bilo na prvom ili trećem čvoru, a simulacija s najvećim prosječnim vremenom zadržavanja korisnika je ona, kada je najkraće vrijeme posluživanja bilo u srednjem, drugom čvoru.

Tablica 3. Prosječno vrijeme provedeno u sustavu za korisnika u mreži s tri čvora, s različitim vremenima čekanja

	T_s na prvom poslužitelju $[s]$	T_s na drugom poslužitelju $[s]$	T_s na trećem poslužitelju $[s]$	T_W na prvom poslužitelju $[s]$	T_W na drugom poslužitelju $[s]$	T_W na trećem poslužitelju $[s]$	T_q za cijeli sustav $[s]$
Simulacija 1.	1	0,5	0,33	0,96	0,027	0,05	2,87
Simulacija 2.	1	0,33	0,5	0,98	0,008	0,14	2,96
Simulacija 3.	0,5	1	0,33	0,155	0,98	0,005	2,97
Simulacija 4.	0,33	1	0,5	0,06	0,96	0,02	2,87
Simulacija 5.	0,5	0,33	1	0,16	0,017	1,1	3,11
Simulacija 6.	0,33	0,5	1	0,07	0,14	0,9	2,94
Simulacija 7.	2	2	1	122	0,004	0	127,004
Simulacija 8.	2	1	2	190	0,001	125	320,001
Simulacija 9.	1	2	2	1	190	4	200
Simulacija 10.	1	1	2	1,1	0,3	190	195,4
Simulacija 11.	1	2	1	0,93	38	0,001	42,931
Simulacija 12.	2	1	1	130	0	0,74	134,74

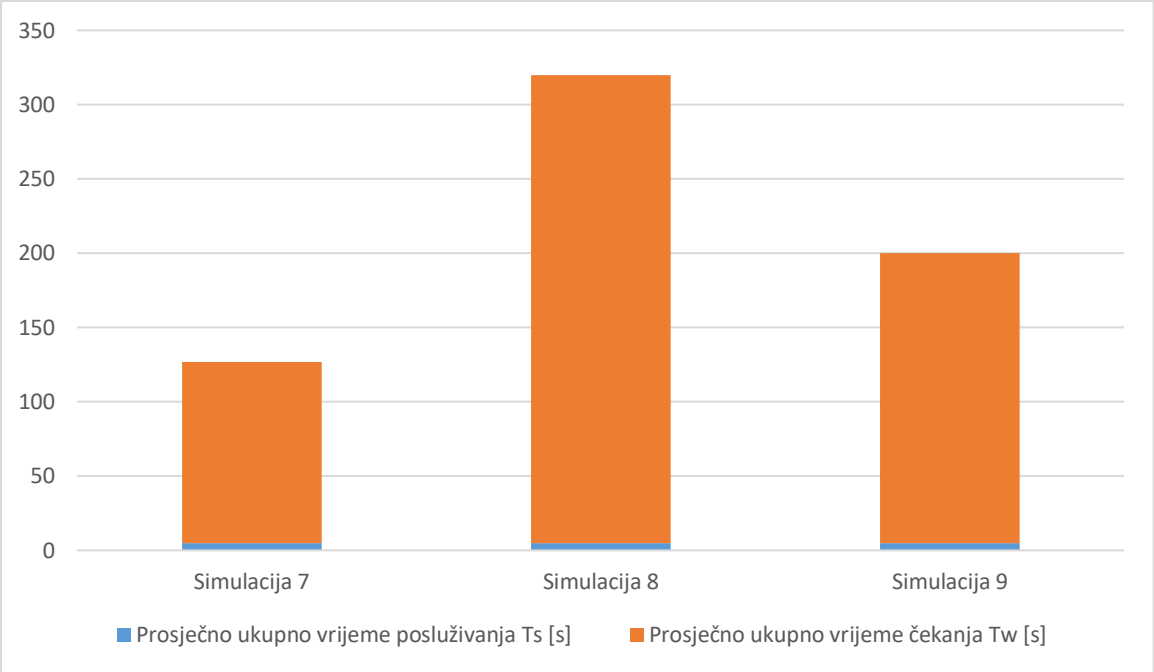
Također se i ovdje ponavlja situacija, kao i u mreži s dva čvora, da je najveća vrijednost prosječnog ukupnog vremena zadržavanja korisnika imala ona simulacija, kojoj je najsporiji poslužitelj bio na zadnjem mjestu. Na grafikonu 8 mogu se vidjeti rezultati prvih šest simulacija grafički, kako bi se vizualno dobio osjet o vremenskim veličinama.



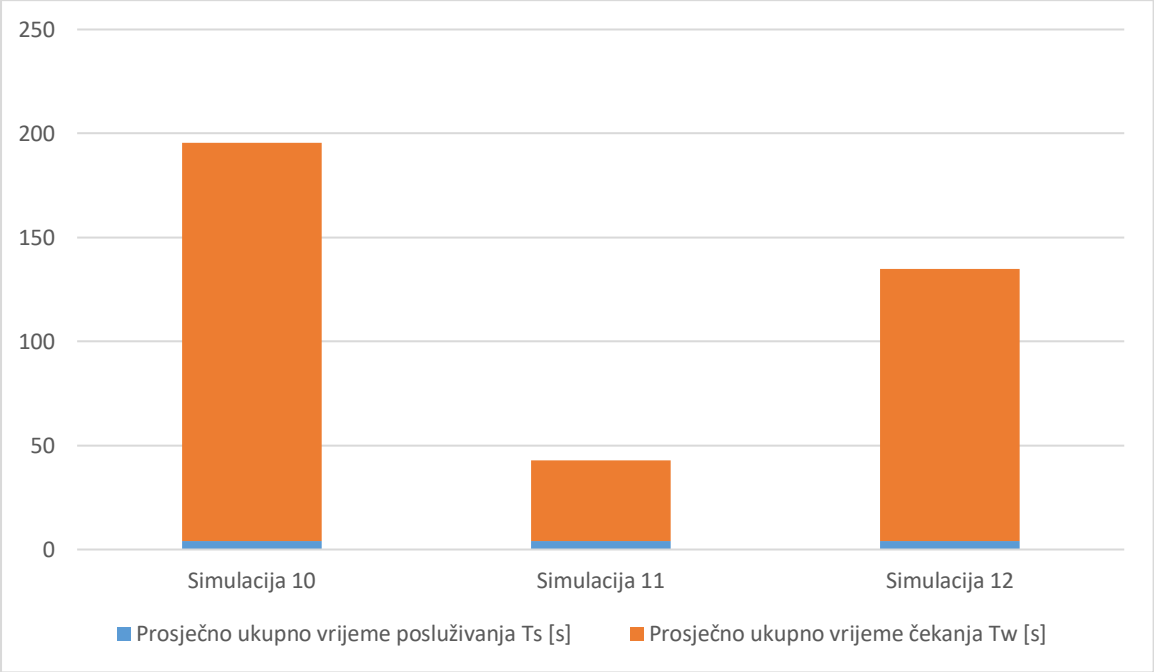
Grafikon 8. Prikaz prosječnog ukupnog vremena posluživanja i prosječnog ukupnog vremena čekanja za prvih šest simulacija (zbroj dvaju navedenih daje prosječno ukupno vrijeme zadržavanja korisnika u sustavu T_q)

Sljedeće tri simulacije (od 7. do 9. simulacije) prikazane su i na grafikonu 9, a to su simulacije gdje dva poslužitelja imaju isto vrijeme posluživanja, a treći ima kraće. Navedene simulacije opet potvrđuju tezu da u mreži s tri čvora, najbrži poslužitelj ne smije biti u sredini jer takav raspored najdulje zadržava korisnika. Oni rasporedi u kojemu je najbrži poslužitelj na ulaznom ili izlaznom mjestu daju bolje rezultate, a najbolji je rezultat kada je najbrži poslužitelj na zadnjem mjestu.

Zadnje tri simulacije (od 10. do 12. simulacije) prikazane su i na grafikonu 10, a to su simulacije gdje dva poslužitelja imaju isto vrijeme posluživanja, a treći ima dulje. S ovakvim vremenima posluživanja, najduže vrijeme zadržavanja korisnika je opet kada se najsporiji poslužitelj postavi na zadnje, treće mjesto, a najkraće vrijeme zadržavanja korisnika je onda kada je najsporiji poslužitelj u sredini.



Grafikon 9. Prikaz prosječnog ukupnog vremena posluživanja i prosječnog ukupnog vremena čekanja za simulacije 7,8 i 9 (zbroj dvaju navedenih daje prosječno ukupno vrijeme zadržavanja korisnika u sustavu T_q)



Grafikon 10. Prikaz prosječnog ukupnog vremena posluživanja i prosječnog ukupnog vremena čekanja za simulacije 10, 11 i 12 (zbroj dvaju navedenih daje prosječno ukupno vrijeme zadržavanja korisnika u sustavu T_q)

7. ZAKLJUČAK

Telekomunikacijski čvor, koji uz poveznicu između čvorova čini mrežu, predstavlja osnovni element telekomunikacijske ili računalne mreže, bez koje bi današnji sustavi otežano funkcionirali, bez obzira na koju granu industrije se odnosi. U ovome radu se opisom uloge čvora pokušalo približiti koje su njegove osnovne zadaće, a opisom podvorbenih sustava i teorije čekanja ukazati na važnost optimizacije čvora i mreže, kako bi čvor bez poteškoća obavljao svoje zadatke.

Upravo iz takvih razloga se provodila simulacija, čiji su rezultati donijeli niz zaključaka o rasporedu i konfiguraciji mreže. Iako broj provedenih simulacija nije popriličan, zaključci kao što su da čvor s najduljim vremenom posluživanja ne bi trebao biti na zadnjem mjestu te da intenzitet nailaska korisnika ne bi trebao prelaziti 80% sposobnosti posluživanja, daju osnove za dimenzioniranje mreže.

Također, od svih dvanaest simulacija provedenih u navedenoj analizi, vrijeme posluživanja razlikuje se samo u tri slučaja, ali se raspored u tim slučajevima mijenja. Temeljem toga, simulacije br. 5, 8 i 10 su simulacije s najduljim ukupnim prosječnim vremenom zadržavanja korisnika u sustavu (svaka u svom slučaju), a navedenim simulacijama je zajedničko to da je poslužitelj s najkraćim vremenom posluživanja bio na drugom mjestu, a poslužitelj s najduljim vremenom posluživanja bio je na zadnjem, trećem mjestu.

Stječe se dojam da se s parametrima koji se podešavaju na ulazu (T_s i λ) i dobivaju na izlazu (T_w, ρ i L_w), modelira ponašanje čvora, kako bi čvor ispunio svoju ulogu u telekomunikacijama ili računalnim mrežama. Ovakvim modeliranjem, lakše je optimizirati stvarne sustave u kojima čvor može predstavljati neke druge elemente u nekim drugim sustavima, kao što je CPU/RAM u računalnom sustavu, gdje će se sporijim procesiranjem ili čitanjem podataka povećavati vrijeme posluživanja i čekanja ili gdje će razni kompleksniji protokoli za kontrolu pogrešaka, povećati vrijeme posluživanja, a to će se na kraju odraziti na kvalitetu usluge. Ovakva analiza daje podlogu za daljnja istraživanja te se ovakva metoda simulacije pokazala kao dobar alat za analizu performansi, posebice u slučajevima kada bi izračunavanje performansi ručno bilo znatno složenije i zahtijevalo puno vremena zaključaka.

LITERATURA

[1] Mrvelj Š, Matulin M. Autorizirana predavanja iz kolegija Tehnologija TK prometa 1. 2016. Dostupno na: https://moodle.srce.hr/2017-2018/pluginfile.php/1210660/mod_resource/content/2/komutacijski%20i%20transmisijski%20sustavi%20za%20objavu.pdf (pristupljeno: ožujak 2020.)

[2] **Javatpoint** portal. Dostupno na: https://www.javatpoint.com/computer-network-routing (pristupljeno: ožujak 2020.)

[3] **Balchunas A**. Static vs. Dynamic routing. 2007. Dostupno na: https://www.routeralley.com/guides/static_dynamic_routing.pdf (pristupljeno: ožujak 2020.)

[4] **Cisco**. Routing and switching essentials. 2008. Dostupno na: https://mars.merhot.dk/w/images/2/2d/RSE6_Chapter2_Student_Version.pdf (pristupljeno: ožujak 2020.)

[5] **CCNA Study guide**. Static, default and dynamic routing. Dostupno na: https://www.freeccnastudyguide.com/study-guides/ccna/ch4/routing/ (pristupljeno: ožujak 2020.)

[6] **Budhathoki D**. Flow control. 2013. Dostupno na: http://www.idc-online.com/technical_references/pdfs/data_communications/Flow_Control.pdf (pristupljeno: ožujak 2020.)

[7] **Pearce D.** Getting strarted with communications engineering. 2007. Dostupno na: https://www-users.york.ac.uk/~dajp1/Introductions/GSW_Flow_Control.pdf (pristupljeno: ožujak 2020.)

[8] **Gerla M, Kleinrock L.** Flow control: a comparative survey. Travanj 1980. Dostupno na: http://nrlweb.cs.ucla.edu/nrlweb/publication/download/583/80jnl-Flow_Control-A_Comparative_Survey.pdf (pristupljeno: ožujak 2020.)

[9] **Bies L.** RS232 flow control and handshaking. Studeni 2019. Dostupno na: https://www.lammertbies.nl/comm/info/rs-232-flow-control (pristupljeno: ožujak 2020.)

[10] **Kharagpur.** Dana link control. Dostupno na: https://nptel.ac.in/content/storage2/courses/106105080/pdf/M3L3.pdf (pristupljeno: ožujak 2020.)

[11] **Budhathoki D**. Error control and detection. 2013. Dostupno na: http://www.idc-online.com/technical_references/pdfs/dana_communications/Error_Control_and_Detection.pdf (pristupljeno: ožujak 2020.)

[12] **Virtual university of Pakistan.** Error detection and correction. Dostupno na: https://vulms.vu.edu.pk/Courses/CS501/Downloads/Error%20detection%20and%20correction.pdf (pristupljeno: ožujak 2020.)

[13] **Geeksforegeeks** portal. Dostupno na: https://www.geeksforgeeks.org/error-detection-in-computer-networks/ (pristupljeno: ožujak 2020.)

[14] **Benard T, Langat R, Gichengo J.** Markovian queuing models. 2012. Dostupno na: https://www.ijsr.net/archive/v3i11/T0NUMTQ3ODg=.pdf (pristupljeno: travanj 2020.)

[15] **Begović M.** Podvorbeni sustavi. Zagreb, Fakultet prometnih znanosti, 2006.

[16] **Mrvelj Š, Matulin M.** Autorizirana predavanja iz kolegija Podvorbeni sustavi. 2018. Dostupno na: https://moodle.srce.hr/2018-2019/course/view.php?id=31516 (pristupljeno: travanj 2020.)

[17] **Berry R**. Queuing theory. Dostupno na: https://www.whitman.edu/documents/Academics/Mathematics/berryrm.pdf (pristupljeno: travanj 2020.)

[18] **Attahiru S.** Queuing theory for telecommunications. London 2010.

POPIS KRATICA

CPU	Central Processing Unit
RAM	Random Access Memory
RIP	Routing Information Protocol
EIGRP	Enhanced Interior Gateway Routing Protocol
OSPF	Open Shortest Path First
IGP	Interior Gateway Protocols
EGP	Exterior Gateway Protocols
OSI	Open Systems Interconnection
IP	Internet Protocol
RS-232	Recommended Standard 232
ASCII	American Standard Code for Information Interchange
RTT	Round Trip Time
ACK	Acknowledgement
LAN	Local Area Network
WAN	Wide Area Network
CRC	Cyclic Redundancy Check
ARQ	Automatic-Repeat Request
FEC	Forward Error Correction
ECC	Error-Corecting Code
VoIP	Voice Over Internet Protocol
FCFS	First Come First Served
FIFO	First In First Out

POPIS SLIKA

Slika 1. Softverska kontrola toka 9
Slika 2. Okviri koji se šalju sa stajališta pošiljatelja 11
Slika 3. Okviri koji se šalju sa stajališta primatelja 11
Slika 4. Dijagram prijelaza među stanjima za M/M/1/∞/∞/FCFS 17
Slika 5. Dijagram prijelaza među stanjima za M/M/1/K/∞/FCFS 20
Slika 6. Dijagram prijelaza među stanjima za M/M/C/∞/∞/FCFS 23
Slika 7. Dijagram prijelaza među stanjima za M/M/C/C/∞/FCFS 25
Slika 8. Prikaz sheme preddefiniranog računalnog modela za elementarni markovljev podvorbeni sustav u simulink-u 31
Slika 9. *Discrete event subsystem* - konfiguracija 32
Slika 10. Grafički prikaz rezultata simulacije za teoretsko vrijeme čekanja entiteta u simulink-u 32
Slika 11. Grafički prikaz rezultata simulacije za broj korisnika u sustavu u simulink-u 33
Slika 12. Grafički prikaz rezultata simulacije za vrijeme čekanja u simulink-u 33
Slika 13. Grafički prikaz rezultata simulacije za iskorištenost poslužitelja u simulink-u 34
Slika 14. Podešavanje intenziteta nailaska korisnika na sustav u simulink-u 35
Slika 15. Podešavanje intenziteta posluživanja u simulink-u 36
Slika 16. Rješenja simulacije u simulink-u 37
Slika 17. Prikaz sheme računalnog modela za mrežu s dva čvora u simulink-u 39
Slika 18. Prikaz sheme računalnog modela za mrežu s tri čvora u simulink-u 44

POPIS TABLICA

Tablica 1. Usporedba statičkog i dinamičkog usmjeravanja 5
Tablica 2. Prosječno vrijeme provedeno u sustavu za korisnika u mreži s dva čvora, s različitim vremenima čekanja 42
Tablica 3. Prosječno vrijeme provedeno u sustavu za korisnika u mreži s tri čvora, s različitim vremenima čekanja 48

POPIS GRAFIKONA

Grafikon 1. Prosječno vrijeme čekanja na posluživanje na prvom poslužitelju 38
Grafikon 2. Prosječno vrijeme čekanja na posluživanje na drugom poslužitelju 40
Grafikon 3. Prosječno vrijeme zadržavanja korisnika u cijeloj mreži s dva čvora 41
Grafikon 4. Prosječno vrijeme čekanja na posluživanje na prvom poslužitelju 43
Grafikon 5. Prosječno vrijeme čekanja na posluživanje na drugom poslužitelju 45
Grafikon 6. Prosječno vrijeme čekanja na posluživanje na trećem poslužitelju 46
Grafikon 7. Prosječno vrijeme zadržavanja korisnika u cijeloj mreži s tri čvora 47
Grafikon 8. Prikaz prosječnog ukupnog vremena posluživanja i prosječnog ukupnog vremena čekanja za prvih šest simulacija (zbroj dvaju navedenih daje prosječno ukupno vrijeme zadržavanja korisnika u sustavu Tq) 49
Grafikon 9. Prikaz prosječnog ukupnog vremena posluživanja i prosječnog ukupnog vremena čekanja za simulacije 7,8 i 9 (zbroj dvaju navedenih daje prosječno ukupno vrijeme zadržavanja korisnika u sustavu Tq) 50
Grafikon 10. Prikaz prosječnog ukupnog vremena posluživanja i prosječnog ukupnog vremena čekanja za simulacije 10, 11 i 12 (zbroj dvaju navedenih daje prosječno ukupno vrijeme zadržavanja korisnika u sustavu Tq) 50