

UNIVERZITET CRNE GORE
ELEKTROTEHNIČKI FAKULTET

mr Slavica Tomović

**ANALIZA PERFORMANSI NOVIH TEHNIKA ZA INŽENJERING
SAOBRAĆAJA U ISP MREŽAMA NAREDNE GENERACIJE**

– Doktorska disertacija –

Podgorica, 2019. godine

UNIVERSITY OF MONTENEGRO
FACULTY OF ELECTRICAL ENGINEERING

MSc Slavica Tomović

**PERFORMANCE ANALYSIS OF NEW TRAFFIC ENGINEERING
TECHNIQUES FOR NEXT GENERATION ISP NETWORKS**

– PhD thesis –

Podgorica, 2019.

PODACI O DOKTORANDU, MENTORU I ČLANOVIMA KOMISIJE

DOKTORAND

Ime i prezime: **Slavica Tomović**

Datum i mjesto rođenja: **05.02.1991. godine, Nikšić, Crna Gora**

Naziv završenog postdiplomskog studijskog programa: **Elektrotehnički fakultet, odsjek *Elektronika, telekomunikacije, računari*, smjer *Telekomunikacije – magistarske studije***

Godina završetka: **2015.**

MENTOR:

dr Igor Radusinović, redovni profesor, Univerzitet Crne Gore, Elektrotehnički fakultet

KOMISIJA ZA ODBRANU DOKTORSKE DISERTACIJE:

dr Zoran Čiča, vanredni profesor, Univerzitet u Beogradu, Elektrotehnički fakultet

dr Igor Radusinović, redovni profesor, Univerzitet Crne Gore, Elektrotehnički fakultet

dr Mirko Sužnjević, docent, Univerzitet u Zagrebu, Elektrotehnički fakultet

dr Vesna Popović Bugarin, vanredni profesor, Univerzitet Crne Gore, Elektrotehnički fakultet

dr Gordana Gardašević, vanredni profesor, Univerzitet Banja Luka, Elektrotehnički fakultet

DATUM ODBRANE:

14.11.2019.

Posvećeno uspomeni na oca Slavka i đeda Miluna

Zahvalnica

Zahvaljujem se svom mentoru prof. dr Igoru Radusinoviću na ukazanom povjerenju i slobodi koju mi je dao u istraživanju, na strpljenju i pažljivom intelektualnom usmjeravanju u najvažnijim momentima. Veoma cijenim vrijeme koje je izdvojio za ovo istraživanje, stručne savjete i kreativne ideje koje je podijelio sa mnom da bi moje iskustvo na doktorskim studijama učinio produktivnim i stimulativnim. Entuzijazam koji ima za kontinuirano profesionalno usavršavanje i istraživanje najaktuelnijih trendova u oblasti telekomunikacija i šire, uz divergentni stil razmišljanja prilikom analize različitih problema, bili su veoma zarazni i inspirativni, čak i u teškim periodima tokom izrade teze.

Zahvaljujem svim članovima komisije za stručnu pomoć i profesionalne sugestije koje su doprinijele poboljšanju kvaliteta ove disertacije.

Na kraju, najveću zahvalnost dugujem svojoj porodici na razumijevanju i безусловnoj podršci koju su mi pružili tokom čitavih studija.

Podgorica, maj 2019. godine

Slavica Tomović

PODACI O DOKTORSKOJ DISERTACIJI

Naziv doktorskih studija: **Doktorske studije elektrotehnike**

Naslov doktorske disertacije: **Analiza performansi novih tehnika za inženjering saobraćaja u ISP mrežama naredne generacije**

Ključne riječi: Softverski definisane mreže, inženjering saobraćaja
kvalitet servisa, balansiranje saobraćaja

Datum prijave doktorske teze: **04.05.2017. godine**

Datum sjednice Senata UCG na
kojoj je prihvaćena teza: **16.10.2017. godine**

Naučna oblast: **Telekomunikacije**

Uža naučna oblast: **Telekomunikacione mreže**

REZIME:

Napredak širokopojsnih tehnologija zajedno sa trendom poznatim kao *Internet of Things* (IoT) uticao je na značajan porast saobraćaja na Internetu i diverzifikaciju Internet servisa. Postojeći mehanizmi za upravljanje ISP (*Internet Service Provider*) mrežama pokazali su brojna ograničenja u pogledu mogućnosti garancije kvaliteta servisa i efikasnog inženjeringa saobraćaja (*Traffic Engineering* - TE). Imajući ova ograničenja u vidu, ISP provajderi obično primjenjuju visok stepen redudanse u svojim mrežama, koristeći linkove 2-3 puta većeg kapaciteta u odnosu na uobičajeno saobraćajno opterećenje. Stoga, ISP provajderi se suočavaju sa velikim troškovima ulaganja u infrastrukturu, dok su mrežni resursi neiskorišćeni najveći dio vremena. Tehnologija softverski-definisanog umrežavanja (*Software Defined Networking* - SDN) razvijena je sa ciljem fleksibilnijeg upravljanja mrežom, poboljšanja iskorišćenosti mrežnih resursa i smanjenja operativnih troškova. Prednosti SDN-a u odnosu na tradicionalne mreže proizilaze iz prisustva centralizovane kontrolne platforme u mreži koja može da kontinuirano prati stanje mrežnih resursa i dinamički reprogramira uređaje za prosleđivanje saobraćaja. U ovoj disertaciji predloženo je više originalnih TE algoritama za SDN ISP mreže koji se mogu iskoristiti za unapređenje BoD (*Bandwidth on Demand*) servisa. Predloženi algoritmi omogućavaju pouzdan prenos podataka uz garanciju kašnjenja i propusnosti. Pri tome, obrada korisničkih zahtjeva vrši se u skoro realnom vremenu, a prisutvo SDN uređaja zahtijeva se samo na ivici mreže. Doprinos u odnosu na postojeća istraživanja u TE oblasti predstavlja i analiza kompromisa između optimalnosti šeme rutiranja i potrebnog broja rekonfiguracija. Dobijeni rezultati pokazuju da predložena rešenja ostvaruju značajna poboljšanja u odnosu na konvencionalne TE tehnike i *state-of-the-art* tehnike koje su bazirane na više-kriterijumskoj optimizaciji. Najefikasnija od predloženih rešenja implementirana su na OpenFlow kontroleru i eksperimentalno verifikovana. Eksperimentalna analiza potvrdila je zadovoljavajuću skalabilnost SDN kontrolne ravni i mogućnost pravovremene reakcije na iznenadne saobraćajne distorzije.

INFORMATION ON DOCTORAL DISSERTATION

PhD study program:	PhD studies in Electrical Engineering
Dissertation title:	Performance analysis of new traffic engineering techniques for next generation ISP networks
Keywords:	Software-defined networking, traffic engineering, quality of service, load balancing
Thesis application date:	04.05.2017.
Date of session of UoM Senate where the thesis is accepted:	16.10.2017.
Scientific area:	Telecommunications
Specific scientific area:	Telecommunication networks

ABSTRACT: The advancements in broadband technology along with the evolution of Internet of Things (IoT) paradigm have caused a significant increase in Internet traffic demand and the diversification of Internet services. The existing ISP (*Internet Service Provider*) network management mechanisms have shown numerous limitations in terms of quality of service (QoS) provisioning and efficiency of traffic engineering (TE). Having those limitations in mind, ISPs typically over-provision network links 2-3 times relative to the offered load, in order to avoid QoS implications. Therefore, ISPs face with a high cost of infrastructure investments, while the network resources are poorly utilized most of the time. Software Defined Networking (SDN) technology emerged with the goal of enabling more flexible network management, improving network utilization and reducing operating costs. The advancements of SDN versus traditional networks arise from the presence of a centralized control platform that can continuously monitor the network state and dynamically reprogramme traffic forwarding devices. In this thesis, several original TE algorithms for SDN-based ISP network have been proposed. Those algorithms could be used to enhance a traditional *Bandwidth on Demand* service by enabling reliable, bandwidth-delay guaranteed data transfers on request. Thereby, user's requests are processed in real-time, and SDN capability is required only at the network edge. An important contribution of this thesis is the analysis of the routing optimality - reconfiguration overhead trade-off. The obtained results show that the proposed solutions outperform conventional TE techniques, which tend to minimize the maximum link utilization, and state-of-the-art TE techniques that are based on multi-objective optimization models. The most effective of the proposed solutions have been implemented on OpenFlow controller and verified experimentally. Experimental analysis confirmed satisfactory scalability of the implemented SDN control plane, as well as its ability to timely react to traffic distortions.

Predgovor

Ova doktorska disertacija je rezultat višegodišnjeg naučno-istraživačkog rada sprovedenog u Laboratoriji za telekomunikacije na Elektrotehničkom fakultetu, Univerziteta Crne Gore. Prihvativši sugestiju prof. dr Igora Radusinovića, koji mi je dao mnoge korisne savjete vezane za izradu doktorske disertacije, svoj istraživački rad sam usmjerila u pravcu razvoja novih rešenja za inženjering saobraćaja i garanciju kvaliteta servisa u ISP mrežama naredne generacije.

Tokom poslednje decenije, softverski definisano umrežavanje (SDN - *Software Defined Networking*) se pozicioniralo kao revolucionarna paradigma umrežavanja, i privuklo pažnju kako industrije tako i akademskih zajednica. Iako još uvijek postoje neusaglašenosti između standardizacionih tijela oko definicije SDN arhitekture, postoji generalni konsenzus o tome da se ključni doprinosi SDN-a očekuju u oblasti inženjeringa saobraćaja. Tokom istraživanja i proučavanja literature, uvidjela sam da je primjena SDN-a u ISP mrežama nedovoljno obrađena. Naime, postojeća SDN rešenja za inženjering saobraćaja dominantno su razvijana za potrebe mreža data-centara i kampus mreža. Iako na prvi pogled djeluje da se ova rešenja uz blage modifikacije mogu jednostavno primijeniti u ISP mrežama, detaljnija analiza pokazala je suprotno. Nedostaci postojećih SDN TE rešenja u pogledu skalabilnosti i podrške za garanciju kvaliteta servisa motivisali su predlog teme za ovu doktorsku disertaciju. Uz svesrdnu podršku mentora, od prve godine doktorskih studija počela sam trasirati put ka zacrtanom cilju, kroz radove koji su objavljeni u međunarodnim časopisima i konferencijama, i učešćem u više naučno-istraživačkih projekata. Neki od mojih radova koji su nastali kao osnova disertacije su ubrzo postali citirani, što me je dodatno učvrstilo u nastavku sprovođenja istraživanja jer je ukazivalo na oskudnost naučnog doprinosa u razmatranoj problematici. Glavni doprinosi disertacije su publikovani u više od 20 radova u međunarodnim časopisima i na međunarodnim konferencijama.

Nadam se da će prezentovani rezultati podstaći nova istraživanja u ovoj oblasti i da će biti od koristi, kako naučnoj zajednici, tako i inženjerskoj praksi.

Podgorica, maj 2019. godine

mr Slavica Tomović

Izvod iz teze

Tehnike za inženjering saobraćaja su od krucijalne važnosti za provajdere Internet servisa (ISP - *Internet Service Providers*) jer omogućavaju optimizaciju mrežnih performansi, efikasno korišćenje mrežnih resursa i smanjenje operativnih troškova. Podrška za inženjering saobraćaja (*Traffic Engineering* - TE) uvedena je još u ATM (*Asynchronous Transfer Mode*) i IP/MPLS (*Internet Protocol/ MultiProtocol Label Switching*) mrežama. Međutim, pokazalo se da ove tehnologije i odgovarajuće TE tehnike ne mogu ispuniti zahtjeve mreža naredne generacije, iz dva razloga. Prvo, savremene Internet aplikacije zahtijevaju da kontrolna ravan mreže donosi i sprovodi kontrolne odluke u skoro realnom vremenu, i da bude dovoljno skalabilna da se izbori sa ogromnom količinom saobraćaja. Mrežna arhitektura bi trebalo da bude u stanju da klasifikuje saobraćaj različitih aplikacija, i da pruži odgovarajući tip servisa svakoj od klasa u kratkom vremenskom periodu (npr. reda milisekundi). Drugo, današnjim mrežama su neophodni inteligentniji i efikasniji alati za balansiranje saobraćajnog opterećenja kako bi odgovorile na izazove koje nameću rapidan razvoj Cloud tehnologija i *Internet of Things* servisa. Međutim, konvencionalne TE tehnologije bazirane su na zatvorenom i nefleksibilnom dizajnu mrežne arhitekture, gdje su kontrolna ravan i ravan podataka integrisane na svakom uređaju. Distribuirana kontrolna ravan onemogućava dinamičku analizu i regulaciju stanja u mreži, usled čega mreža nije stanju da se prilagodi rastućim i veoma varijabilnim obrascima saobraćaja.

Softverski-definisano umrežavanje (SDN - *Software Defined Networking*) predstavlja novi model mrežne arhitekture kod koje je kontrolna ravan razdvojena od ravni podataka i logički centralizovana na kontrolerskom uređaju. SDN kontroler predstavlja pogodnu platformu za implementaciju sofisticiranih TE algoritama u sklopu centralizovanih aplikacija koje imaju globalni uvid u stanje mreže i mogu dinamički mijenjati mrežnu konfiguraciju u cilju poboljšanja sveukupnih performansi. Potencijal SDN mrežne arhitekture za poboljšanje iskorišćenosti mrežnih resursa već je uspješno demonstriran u DC (*Data Center*), *inter-DC* i kampus mrežama. Međutim, primjena SDN-a u ISP mrežama počela je da privlači pažnju tek nedavno, i za sada predstavlja nedovoljno istraženu oblast u kojoj se očekuju novi naučno-istraživački doprinosi.

U ovoj disertaciji analiziran je problem inženjeringa saobraćaja u softverski-definisanim ISP mrežama. Specifičnost ovog problema ogleda se u tome što u ISP mrežama nije moguće

kontrolisati dolazne saobraćajne tokove na način koji je uobičajen u DC i *inter-DC* mrežama, gdje je SDN prvobitno našao svoju primjenu. Naime, ISP provajderi su u obavezi da korisnicima pruže kvalitet servisa (QoS - *Quality of Service*) koji je definisan odgovarajućim SLA (*Service Level Agreement*) ugovorima. Drugi izazov je skalabilnost centralizovane kontrolne ravni, koja bi trebalo da podrži obradu velikog broja korisničkih zahtjeva u kratkom vremenskom intervalu, kontinuirano prati promjene saobraćaja i događaje u mreži, i reaguje na iste na što optimalniji način. U disertaciji je predloženo originalno SDN kontrolno okruženje za ISP mreže, za koje je karakteristična organizacija logike za upravljanje saobraćajem u *online* i *offline* modul. *Online* modul nudi interfejs prema krajnjim korisnicima, prihvata njihove BoD (*Bandwidth on Demand*) zahtjeve, sprovodi kontrolu pristupa na osnovu trenutne dostupnosti resursa u mreži, i instalira nove rute ukoliko je neophodno. Osnovni kriterijum koji *online* modul mora da ispuni je donošenje odluke o prihvatanju/rutiranju novih saobraćajnih tokova praktično u realnom vremenu. Optimizacija iskorišćenosti mrežnih resursa delegirana je *offline* modulu, koji periodično vrši inženjering saobraćaja uzimajući u obzir QoS ograničenja u pogledu propusnosti, kašnjenja i pouzdanosti, za svaki pojedini korisnički zahtjev. Na ovaj način djelimično je „žrtvovana” optimalnost rutiranja u cilju smanjenja vremena odziva na BoD zahtjev. Bitno je napomenuti da su istraživanja orijentisana ka unapređenju servisnog portfolia ISP provajdera, koji je danas dominantno ograničen na dugoročno iznajmljivanje privatnih linija (virtuelnih linkova) sa rezervisanim kapacitetom. Predloženi dizajn SDN kontrolera podržava fleksibilni BoD servis, koji omogućava korisnicima da dinamički skaliraju QoS parametre svojih konekcija u skladu sa trenutnim potrebama, na potpuno automatizovan način.

Posebna pažnja posvećena je analizi uticaja dinamičkog inženjeringa saobraćaja na stabilnost mreže. Predložen je novi TE algoritam baziran na više-kriterijumskom optimizacionom modelu koji teži da minimizuje dvije konfliktne optimizacione funkcije: „cijenu” rutiranja i broj rekonfiguracija. Prioritizacija optimizacionih funkcija vrši se dinamički, primjenom Lyapunov *drift-plus-penalty* algoritma, koji kao ulazni parametar koristi zadato ograničenje u pogledu prosječne učestanosti rekonfiguracija. Predloženi algoritam dopušta veći broj rekonfiguracija od granične vrijednosti u slučaju kada šema rutiranja značajno odstupa od optimalne. Međutim, posmatrano na dužem vremenskom intervalu, rekonfiguracije su pažljivo kontrolisane tako da njihov broj nikada značajnije ne prelazi vrijednost koju preferira mrežni operator. Dobijeni simulacioni rezultati su pokazali da predloženo rešenje sa manjom računskom kompleksnošću ostvaruje bolje performanse od *state-of-the-art* konkurentskih TE rešenja iz literature, a pri tome ne zahtijeva poznavanje saobraćajnih statistika.

TE algoritmi koji su se u simulacionoj analizi pokazali kao najskalabilniji i najefikasniji u pogledu servisnog kapaciteta, realizovani su i testirani u eksperimentalnom SDN okruženju.

Thesis overview

Traffic engineering (TE) techniques are of crucial importance for Internet Service Providers (ISPs) because of their ability to optimize network performance, resource utilization and operational expenditures. TE support was firstly introduced in ATM (*Asynchronous Transfer Mode*) and IP/MPLS (*Internet Protocol/MultiProtocol Label Switching*) networks. However, it has been shown that those technologies, as well as corresponding TE techniques, fail to meet requirements of the next generation networks, for two reasons. First, modern Internet applications require scalable control plane that can handle a huge amount of traffic, and which is able to make and conduct control decision in near real-time. The network architecture should support fine-grained classification of traffic generated by diverse applications, and provide adequate service for each traffic class within a short time (e.g. order of milliseconds). Second, today's networks need more intelligent and efficient tools for load balancing in order to respond to challenges imposed by the rapid evolution of Cloud technologies and *Internet of Things* services. However, conventional TE technologies are based on closed and inflexible network architecture, where the control plane and the data plane are tightly integrated in each device. Distributed control plane prevents dynamic analysis and regulation of the network state. Consequently, the network is not able to adapt to rising and highly variable traffic patterns.

Software-defined networking (SDN) introduces a new model of network architecture where the control plane is decoupled from the data plane and logically centralized at the controller device. SDN controller has been widely recognized as a suitable platform for the implementation of sophisticated TE applications, which have global view of the network state and can dynamically adjust the network configuration in order to improve overall performance. The potential of SDN for improving resource utilization has been already successfully showcased in DC (*Data Center*), *inter*-DC and campus networks. The use of SDNs in ISP networks has begun to attract attention relatively recently, and for the time being represents prominent, but insufficiently explored field, where new scientific/research contributions are expected.

This thesis analyses the problem of traffic engineering in software-defined ISP networks. The specificity of this problem reflects in the fact that ISP networks are not allowed to control incoming traffic flows in a way that is common in DC and *inter*-DC networks. Namely, ISP providers are required to provide users with the quality of service (QoS) defined in the

appropriate SLAs (*Service Level Agreements*). Another challenge is the scalability of the centralized control plane, which is expected to handle a large number of user requests over a short time interval, to continuously monitor network conditions and react to sudden events in the most optimal way. Motivated by the aforementioned challenges, this thesis proposes a new SDN control framework for ISP networks that implements traffic management logic in two separate modules: *online* and *offline* module. The *online* module processes *Bandwidth on Demand* (BoD) requests, performs admission control based on the current availability of the network resources, and installs new routes for incoming traffic flows if necessary. The key requirement that must be met by this module is making admission control decisions in near real-time. A task of optimizing resource utilization is delegated to the *offline* module, which periodically performs traffic engineering subject to QoS constraints in terms of bandwidth, delay and reliability for each individual user request. In this way, some routing optimality is "sacrificed" for the sake of reducing the service response time. The presented research was oriented towards improving the service portfolio of traditional ISP networks, which is mostly limited to long-term leasing of virtual-links (private lines) with reserved capacity. The proposed SDN controller design supports flexible BoD service, which allows users to dynamically scale QoS parameters of their connections up and down, according to the current needs, in a fully automated manner.

A special attention is paid to the impact of dynamic traffic engineering on the network stability. A new TE algorithm is proposed, which strives to minimize two conflicting optimization functions: a total routing "cost" and the number of reconfiguration. Priority of the optimizations functions is determined dynamically, using the Lyapunov *drift-plus-penalty* algorithm, which takes into account a given constraint in terms of the time-average reconfiguration rate. The proposed algorithm allows more reconfigurations when the routing scheme deviates significantly from the optimal one. However, over a longer time interval, reconfigurations are carefully controlled so that the time average reconfiguration rate does not exceed the critical value, which is defined by the network operator. The presented simulation analysis shows that the proposed solution outperforms *state-of-the-art* competitive TE solutions. Thereby, it has much lower complexity and does not require knowledge of traffic statistics.

The most scalable and the most efficient of the proposed TE solutions have been implemented on OpenFlow controller and experimentally validated in a virtualized network environment.

Sadržaj

Lista slika	i
Lista tabela	iv
Skraćenice	v
Uvod	1
1 Izazovi u razvoju ISP mreža naredne generacije	6
1.1 5G sistemi: motiv i zahtjevi	7
1.1.1 Servisne grupe i ključni indikatori performansi	8
1.2 Nedostaci tradicionalne mrežne arhitekture	11
1.2.1 Ograničenja distribuirane kontrolne ravni	11
1.2.2 Ograničenja hardverske implementacije mrežnih funkcija	12
1.3 Softverizacija mreže	13
1.3.1 NFV	13
1.3.2 Softverski definisano umrežavanje	15
1.3.3 OpenFlow	16
1.4 Primjena SDN-a u ISP mrežama naredne generacije	20
1.4.1 Bandwidth on Demand (BoD) servis u ISP mrežama	20
1.4.2 Virtuelizacija transportne mreže	21

1.4.3	Više-nivovska optimizacija trasnportne mreže	21
2	Upravljanje saobraćajem	23
2.1	IP TE	24
2.1.1	SPF rutiranje	25
2.1.2	ECMP rutiranje	27
2.2	MPLS TE	28
2.2.1	Offline MPLS TE	30
2.2.2	Online MPLS TE	31
2.2.3	Motiv za razvoj novih TE rešenja	32
2.3	Ograničenja savremenih TE tehnologija	33
2.4	Modeli saobraćaja	36
2.5	Formulacija optimizacionih TE algoritama za različite modele saobraćaja	37
2.5.1	Model mreže	38
2.5.2	Formulacija MLU minimizacije za <i>pipe</i> model	38
2.5.3	Formulacija MLU minimizacije za <i>hose</i> model	39
2.5.4	Formulacija MLU minimizacije za HSDT model	40
2.6	Višekriterijumska TE optimizacija	41
2.6.1	WSM metod	42
2.6.2	Metod ϵ -ograničenja	43
3	Tehnike za garanciju kvaliteta servisa	45
3.1	IntServ	46
3.2	DiffServ	48
3.3	QoS u OpenFlow mrežama	49
3.3.1	Baferi	49

3.3.2	OpenFlow <i>meter</i> tabele	51
3.4	QoS algoritmi rutiranja: analiza kompleksnosti	52
3.4.1	QoS rutiranje sa jednom metrikom	53
3.4.2	Rutiranje na osnovu više QoS kriterijuma	54
4	Predlog kontrolnog okruženja za inženjering saobraćaja u SD-ISP mrežama	56
4.1	Model sistema	58
4.1.1	Modul za <i>online</i> rutiranje	60
4.1.2	Modul za <i>offline</i> TE	62
4.1.3	Alternativne definicije optimizacione funkcije	67
4.2	Simulaciona metodologija	68
4.3	Simulacioni rezultati	70
4.3.1	Poređenje sa konvencionalnim TE rešenjima	70
4.3.2	Poređenje sa SDN TE rešenjima	72
4.3.3	Analiza uticaja optimizacione funkcije	80
5	Predlog TE rešenja sa kontrolisanim rekonfiguracijama: RO-RO	81
5.1	RO-RO optimizator	82
5.1.1	Generisanje Pareto optimalnih rešenja	83
5.1.2	Izbor kompromisnog rešenja	87
5.2	RO-RO heuristika	88
5.3	RORO-MC: kompromisno rešenje za minimizaciju cijene rutiranja	91
5.4	Simulaciona analiza	94
5.4.1	Performanse RO-RO pristupa u uslovima odsustva QoS3 zahtjeva	95
5.4.2	Performanse RORO-MC pristupa u uslovima odsustva QoS3 saobraćaja	101
5.4.3	Performanse RORO-MC pristupa u uslovima sa QoS3 saobraćajem	104

6	Eksperimentalni rezultati	110
6.1	Prototip SDN kontrolnog okruženja za inženjering saobraćaja u ISP mre- žama	110
6.1.1	Monitoring resursa	111
6.1.2	API za BoD servis i instalaciju tunela	113
6.1.3	Obrada BoD zahtjeva	113
6.2	Eksperimentalna validacija performansi	117
	Zaključak	123
	Literatura	127

Spisak slika

1.1	5G servisne grupe.	9
1.2	Značaj indikatora performansi za različite servisne grupe.	10
1.3	Tradicionalna i SDN mrežna arhitektura.	11
1.4	Virtuelizacija mrežnih funkcija.	14
1.5	SDN arhitektura.	15
1.6	Elementi OpenFlow mreže.	17
2.1	Offline inženjering saobraćaja.	24
2.2	Rutiranje najkraćom putanjom.	26
2.3	Optimizacija težinskih faktora kod SPF rutiranja.	26
2.4	Princip rutiranja na osnovu MPLS labela.	29
2.5	Primjer kreiranja „interferencije” prilikom izbora ruta.	32
2.6	Optimizacija ruta u MPLS i SDN mrežama.	35
2.7	Poređenje različitih modela saobraćaja.	37
2.8	Pareto front za optimizacione funkcije μ_1 i μ_2	43
3.1	RSVP protokol.	47
3.2	Tabele bafera i QoS tabele OpenvSwitch-a.	50
3.3	Elementi OF-Config protokola.	51
3.4	Format zapisa u <i>meter</i> OpenFlow tabeli.	52

4.1	Razmatrana arhitektura mreže.	58
4.2	<i>Online</i> algoritam rutiranja.	61
4.3	Primjer sortirajuće mreže (a) i pronalaženja dva najveća elementa <i>bubble-sort</i> pristupom [3].	64
4.4	Sprint ISP topologija.	68
4.5	Promjena intenziteta saobraćaja u toku dana.	70
4.6	Broj odbijenih jedinica BoD zahtjeva za SDN-LB, CSPF-TE i Hose-TE. .	71
4.7	Srednja MLU vrijednost za SDN-LB, CSPF-TE i Hose-TE.	72
4.8	Odbijene BoD jedinice u simulacionom scenariju sa QoS1 i QoS2 zahtjevima.	75
4.9	Prosječna MLU vrijednost u simulacionom scenariju sa QoS1 i QoS2 zahtjevima.	75
4.10	Odbijene BoD jedinice u drugom simulacionom scenariju.	76
4.11	Prosječna MLU vrijednost u drugom simulacionom scenariju.	77
4.12	Uticaj parametra T na broj odbijenih BoD zahtjeva.	79
4.13	Uticaj parametra T na srednju MLU vrijednost.	80
4.14	Uticaj optimizacione funkcije na broj odbijenih BoD zahtjeva.	80
5.1	Cijena linka u funkciji iskorišćenosti linka.	92
5.2	Princip rada RORO-MC optimizatora.	94
5.3	Odbijeni BoD zahtjevi.	96
5.4	Uticaj T parametra na performanse RORO-H pristupa. Sufiksi u legendi indiciraju vrijednost T parametra koja je korišćena u simulacijama.	99
5.5	Uticaj T parametra na performanse RORO i EWMOP TE pristupa. Sufiksi u legendi indiciraju korišćenu vrijednost T parametra.	100
5.6	Uticaj prosječnog trajanja toka (FD - <i>Flow Duration</i>) parametra na performanse RORO-H i SDN-LB pristupa.	101

5.7	Odbijene jedinice BoD zahtjeva i srednji MLU u simulacionoj analizi performansi RORO-MCH pristupa.	102
5.8	Poređenje RORO-MC i EWMOP optimizacionih modela.	103
5.9	Uticaj T parametra na performanse RORO-MCH pristupa.	104
5.10	Uticaj R_{avg} ograničenja na performanse RORO-MCH pristupa u uslovima bez QoS3 saobraćaja.	105
5.11	Performanse analiziranih TE algoritama u funkciji procenta QoS3 zahtjeva u ukupnom generisanom saobraćajnom opterećenju.	107
5.12	Uticaj R_{avg} ograničenja na performanse RORO-MCH pristupa.	108
5.13	Uticaj q parametra na broj odbijenih BoD zahtjeva.	108
5.14	Uticaj q parametra na prosječnu učestanost rekonfiguracija.	109
6.1	Princip detekcije topologije razmjenom BDDP poruka.	112
6.2	Primjer konfiguracije tabele tokova i <i>group</i> tabele za svič na ulazu u mrežu (a-b), u jezgru mreže (c) i izlazu mreže (d).	114
6.3	Ilustracija odabira <i>bucket</i> -a na osnovu <i>hash</i> funkcije.	115
6.4	Grafički interfejs Floodlight kontrolera i mrežna topologija korišćena u eksperimentima.	119
6.5	Odbijene jedinice BoD zahtjeva u Mininet eksperimentima. Analizirani algoritmi označeni su na sa: A = RORO-MCH, B = SDN-CM (T=5min), C = SDN-CM (T=1h), D = SDN-LB (T=5min), E = SDN-LB (T=1h). Indeksi {1, 2, 3, 4} označavaju različite saobraćajne scenarije sa udjelom QoS2 zahtjeva od {0, 40, 60, 100}%.	120
6.6	Prosječna učestanost rekonfiguracija u Mininet eksperimentima bez QoS3 zahtjeva.	120
6.7	Eksperimenatalni (E) i simulacioni (S) rezultati u pogledu odbijenih BoD jedinica. Analizirani algoritmi označeni su indeksima: 1 = RORO-MCH, 2 = SDN-CM (T=5min) i 3 = SDN-CM (T=1h).	121
6.8	Prosječna učestanost rekonfiguracija u Mininet eksperimentima sa QoS3 zahtjevima.	122

Spisak tabela

1.1	Poređenje IMT-2020 i IMT-Advanced sistema	10
3.1	Zahtjevi različitih aplikacija u pogledu kašnjenja i brzine prenosa.	46
3.2	Računska kompleksnost kompozitnih problema QoS rutiranja.	55
4.1	Analiza skalabilnosti/kompleksnosti različitih SDN-TE pristupa - Scenarij I. Broj rekonfiguracija (Rekonfig), broj generisanih kontrolnih poruka (Flow-mod) i broj zapisa u tabelama tokova (FT zapisi).	78
4.2	Analiza skalabilnosti/kompleksnosti različitih SDN-TE pristupa - Scenarij II. Broj rekonfiguracija (Rekonfig), broj generisanih kontrolnih poruka (Flow-mod) i broj zapisa u tabelama tokova (FT zapisi).	79
5.1	Notacija za RO-RO optimizacioni model	84
5.2	Konfiguracija simulacija	95
5.3	Poređenje RO-RO pristupa sa konkurentskim TE rešenjima u pogledu prosječnog broj rekonfiguracija u minuti.	97
5.4	Poređenje RORO-MCH pristupa sa konkurentskim TE rešenjima u pogledu prosječnog broj rekonfiguracija u minuti.	102
5.5	Uticaj parametra N_{max} na vrijeme izvršavanja (CT - <i>Computational Time</i>) i odbijene jedinice BoD zahtjeva (RB - <i>Rejected Bandwidth</i>).	105
6.1	Konfiguracija eksperimenata.	118
6.2	Prosječni i maksimalni procenat izgubljenih paketa (<i>APLR</i> i <i>MPLS</i>), i prosječni i maksimalni <i>jitter</i> (<i>Avg</i> i <i>Max jitter</i>).	119

Skraćenice

AF Assured Forwarding

APLR Average Packet Loss Rate

ATM Asynchronous Transfer Mode

BFD Bidirectional Forwarding Detection

BoD Bandwidth on Demand

CS Class Selector

CSPF Constrained Shortest Path First

DC Data Center

DDoS Distributed Denial of Service

DiffServ Differentiated Services

DORA Dynamic Online Routing Algorithm

DS Delay Sensitive

DSCP Differentiated Services Code Point

ECMP Equal Cost MultiPath

eMBB Enhanced Mobile Broadband

EF Expedited Forwarding

EWMOP Equally Weighted Multi-objective OPTimization

FD Flow Duration

FEC Forwarding Equivalence Class

HSDT Hose model with bounds of Source-Destination Traffic demands

FTP File Transfer Protocol

IETF Internet Engineering Task Force

IntServ Integrated Services

IO Izvorišno-Odredišni

IoT Internet of Things

IP Internet Protocol

IS-IS Intermediate System to Intermediate System

ISP Internet Service Provider

ITU International Telecommunication Union

LP Linear Programming

LSR Label Switching Router

LTE Long Term Evolution

MATE MPLS adaptive Traffic Engineering

MCF Multi Commodity Flow

MIRA Minimum Interference Routing Algorithm

MLU Maximum Link Utilization

mMTC Massive Machine Type Communications

MPLR Maximum Packet Loss Rate

MPLS Multi Protocol Label Switching

MOP Multi-objective OPTimization

NFV Network Function Virtualization

NI Northbound Interface

NP Nondeterministic Polynomial time

OSPF Open Shortest Path First

PCE Path Computation Element

PHB Per-Hop Behavior

PoP Point of Presence

PPV Path Potential Value

QoS Quality of Service

RC Routing Cost

REST REpresentational State Transfer

RO-RO Routing Optimality - Reconfiguration Overhead balancing

RORO-H RORO-Heuristic

RORO-MC RORO Minimizing routing Cost

RORO-MCH RORO-MC Heuristic

RSVP Resource Reservation Protocol

SD Software Defined

SDN Software Defined Networking

SDN-LB Software Defined Networking with Load Balancing

SDN-CM Software Defined Networking with routing Cost Minimization

SI Southbound Interface

SNMP Simple Network Management Protocol

SPF Shortest Path First

SWP Shortest Widest Path

TCP Transmission Control Protocol

TE Traffic Engineering

TED Traffic Engineering Database

ToS Type of Service

UDP User Datagram Protocol

uRLLC Ultra Reliable Low Latency Communication

VLAN Virtual Local Area Network

VNF Virtual Network Function

VoIP Voice over Internet Protocol

VPN Virtual Private Network

VTN Virtual Transport Network

WAN Wide Area Network

WFQ Weighted Fair Queuing

WSP Widest Shortest Path

Uvod

Uslijed ekspanzije *Internet of Things* (IoT) sistema u automobilske industriji, zdravstvu i energetici, kao i sve veće popularnosti aplikacija proširene i virtualne stvarnosti, garancija ultra-pouzdanosti komunikacije sa malim kašnjenjem postaje ključni izazov u mrežama Internet Servis Provajdera (ISP). Kako bi održali korak za pomenutim trendovima i pružili adekvatan kvalitet servisa u uslovima velikog saobraćajnog opterećenja, ISP provajderi primjenjuju visok stepen redundantnosti u svojim mrežama. Naime, u okosnici mreže koriste se linkovi 2-3 puta većeg kapaciteta u odnosu na njihovo prosječno opterećenje [1]. S obzirom da ovakav pristup nije ekonomski održiv, istraživačka zajednica ulaže velike napore u razvoj naprednih tehnika za inženjering saobraćaja (*Traffic Engineering - TE*), čiji je zadatak optimalno balansiranje saobraćaja u mreži.

U uobičajenoj klasifikaciji TE algoritma ističu se dvije komplementarne forme: *online* i *offline* [2]. *Online* algoritmi reaguju na promjene u mreži u realnom vremenu. *Offline* algoritmi se izvršavaju periodično, u dužim vremenskim intervalima. Iako su obje navedene forme TE algoritama široko istraživane u kontekstu ATM (*Asynchronous Transfer Mode*) i IP/MPLS (*Multi Protocol Label Switching*) mreža, praksa je pokazala da se složeniji optimizacioni modeli ne mogu efikasno implementirati na distribuiranoj kontrolnoj ravni, koja je karakteristična za ove mreže. MPLS mreže, na primjer, posjeduju mehanizme za praćenje opterećenosti linkova, reaktivno rerutiranje saobraćajnih tokova i autoregulaciju dodijeljenog propusnog opsega [3]. Međutim, rezultirajući sistem je previše složen i destabilizuje mrežu [4]. Takođe, kontrolne odluke donose se bez globalnog uvida u stanje mreže, čime se ne uspijeva obezbijediti „globalna” optimalnost šeme rutiranja [1].

Tehnologija softverski-definisiranog umrežavanja (*Software Defined Networking - SDN*) u potpunosti je redefinisala tradicionalnu mrežnu arhitekturu razdvajajući kontrolu ravan od ravni podataka [5]. SDN kontrolna ravan je logički centralizovana na resursno moćnoj platformi - SDN kontroleru, koji vrši apstrakciju svih mrežnih resursa u cilju fleksibil-

nijeg upravljanja mrežom. Pri tome, programski interfejs kontrolera može simultano koristiti veći broj aplikacija sa potencijalno konfliktnim zahtjevima [6]. U SDN mrežama, kontrola saobraćaja obavlja se na nivou „toka” paketa, koji se može definisati na osnovu sličnosti paketa u pogledu bilo kojeg podskupa polja u zaglavlju (od nivoa linka do aplikacionog nivoa) [7]. Na ovaj način, SDN uvodi novi koncept mrežne virtuelizacije, koji pored mogućnosti dijeljenja resursa fizičke infrastrukture omogućava korišćenje različitih kontrolnih protokola u svakoj od virtuelnih mreža. Tako, na primjer, saobraćaj aplikacija osjetljivih na kašnjenje može se tretirati na potpuno drugačiji način od saobraćaja za koji je propusnost ključna metrika kvaliteta servisa [8]. Ovo je veliki iskorak u odnosu na konvencionalne IP (*Internet Protocol*) mreže koje nude samo *best-effort* servis, tretirajući kompletan saobraćaj na isti način. Uvođenje kontrole na nivou „toka” takođe otvara prostor za realizaciju dinamičkog *multipath* rutiranja, koje značajno može doprinijeti povećanju kapaciteta mreže. Reprezentativan primjer mogućnosti SDN-a je Google-ova WAN (*Wide Area Network*) mreža, u kojoj je iskorišćenost linkova dovedena do nivoa od 95% [9].

SDN je do danas primjenu dominantno našao u mrežama savremenih *data* centara (DC) [10–12] i WAN (*Wide Area Networks*) mrežama koje povezuju *data* centre na geografski udaljenim lokacijama (tzv. *inter-DC* mreže) [1, 9, 13, 14]. Iako na prvi pogled djeluje da se postojeći SDN TE mehanizmi koji su dizajnirani za DC i *inter-DC* mreže mogu u potpunosti prilagoditi potrebama ISP mreža, detaljnija analiza otkriva da ključni zahtjevi ISP mreža na ovaj način ne mogu biti ispunjeni [15–17]. Prije svega, u ISP mrežama nije moguće proizvoljno oblikovati saobraćajne tokove zbog SLA (*Service Level Agreement*) ugovora koje ISP provajderi sklapaju sa korisnicima, i koji provajderu nameću obavezu garancije određenog nivoa kvaliteta servisa (npr. propusnosti). Takođe, DC i *inter-DC* rešenja nisu dovoljno skalabilna za ISP mreže, s obzirom da je većina njih dizajnirana za optimizaciju agregiranih saobraćajnih tokova između manjeg broja lokacija [15]. Topologije većih ISP mreža sadrže desetine hiljada mrežnih uređaja, što ponekad rezultira milionima ograničenja i varijabli u formulacijama TE optimizacionih problema. Da bi osigurao optimalan rad mreže, SDN kontroler mora reagovati na događaje u mreži (odlaske i dolaske saobraćajnih tokova, otkaze linkova i SDN svičeva) u skoro realnom vremenu. Ovo zahtijeva rešavanje složenih optimizacionih modela u veoma uskim vremenskim okvirima i učestale rekonfiguracije SDN svičeva. Prilikom svake rekonfiguracije SDN kontroler mora da ukloni stara pravila rutiranja i instalira nova. Ažuriranja pravila rutiranja ne izvršavaju se na svim SDN svičevima u isto vrijeme, pa prilikom svake re-

konfiguracije postoji rizik od stvaranja petlji rutiranja i privremenog zagušenja linkova [18, 19]. U ISP mrežama navedeni problemi su posebno izraženi uslijed velikih razlika u propagacionim kašnjenjima na linkovima. Stoga, jedan od bitnih izazova je pronalaženje efikasnog kompromisa između optimalnosti rutiranja i učestanosti rekonfiguracija.

Centralna tema ove doktorske disertacije su TE algoritmi za garanciju kvaliteta servisa (*Quality of Service* - QoS) u softverski-definisanim ISP mrežama. Istraživanje je motivirano potrebom ISP provajdera za novim servisnim modelima i efikasnijim mehanizmima za alokaciju virtuelnih linkova, čija je uloga prenos podataka između određenih lokacija u mreži u skladu sa unaprijed definisanom QoS politikom. Kako garancija kvaliteta servisa ima svoju cijenu - ISP provajder mora da dodijeli dio fizičkih reursa svakom od klijenata, od interesa je da se primjenom adekvatnih TE algoritama mrežni resursi iskoriste na što efikasniji način. Različiti profili korisnika imaju različite QoS zahtjeve. Korisnicima interaktivnih multimedijalnih servisa ograničavanje kašnjenja u mreži je od primarnog značaja. Sa druge strane, konvencionalni *Web*, *FTP (File Transfer Protocol)* i *Email* saobraćaj nije osjetljiv na kašnjenje, ali je i dalje potrebno rezervirati određeni kapacitet u mreži s obzirom da se virtuelnim linkom prenosi agregirani saobraćaj iz nekog domena. U literaturi je predloženo više TE tehnika koje teže da minimizuju vjerovatnoću blokiranja QoS zahtjeva [20–24]. Međutim, predložena rešenja dominantno su dizajnirana za klasični *Bandwidth on Demand* (BoD) servis, čiji je fokus na garanciji propusnosti. Sa druge strane, garancija kašnjenja je nedovoljno istražena, uprkos značaju koji ima za veliki broj aplikacija [25–27]. U istraživanjima čiji su rezultati izloženi u disertaciji se polazi od hipoteze da softverski-definisane ISP mreže mogu ponuditi širi portfolio SLA opcija od tradicionalnih MPLS ISP mreža, uključujući garanciju kašnjenja, propusnosti i pouzdanosti, uz prihvatljivu brzinu izvršavanja kontrolnih algoritma. Do sada je osmišljeno, simulirano i praktično implementirano više originalnih TE rešenja, a rezultati istraživanja su publikovani u naučnim časopisima i prezentovani na više naučnih konferencija. Fokus istraživanja nije samo na razvoju matematičkih optimizacionih modela koji simultano razmatraju više QoS metrika, već i na prilagođavanju istih tako da njihova praktična primjena u ISP mrežama bude izvodljiva. Stoga, pored rešenja za optimizaciju iskorišćenosti resursa u SDN ISP mrežama, predložena su i rešenja koja koriste dva konfliktna optimizaciona kriterijuma: (i) minimizaciju „troškova” rutiranja, i (ii) minimizaciju broja rekonfiguracija. Ova rešenja nude operatorima mogućnost definisanja ograničenja u pogledu prosječne učestanosti rekonfiguracija, na osnovu kojeg se kasnije donosi odluka o najefikasnijem kompromisnom rešenju za oba razmatrana optimizaciona kriterijuma.

Doktorska disertacija je organizovana u šest poglavlja.

U Glavi 1 dat je kratak osvrt na ključne izazove u razvoju ISP mreža naredne generacije. Predstavljeni su novi servisi koji su istakli nedostatke tradicionalne mrežne arhitekture, kao i koncept mrežne softverizacije na kojem će biti bazirane mreže naredne generacije. Posebna pažnja posvećena je OpenFlow SDN protokolu [7], koji je iskorišćen za realizaciju predloženih TE pristupa. Na kraju poglavlja diskutovane su mogućnosti SDN-a za unapređenje i proširenje servisnog portfolia ISP provajdera.

U Glavi 2 dat je pregled TE rešenja iz literature koja su dizajnirana za tradicionalne MPLS i IP mreže. Rešenja su klasifikovana po kriterijumu dinamike izvršavanja na *online* i *offline*. Posebno su istaknuti nedostaci ovih rešenja koje je moguće prevazići SDN mrežnom arhitekturom. Takođe, predstavljeni su različiti saobraćajni modeli koji se koriste prilikom definisanja optimizacionih TE problema. Za svaki od analiziranih saobraćajnih modela formulisan je problem minimizacije maksimalne iskorišćenosti linka (*Maximum Link Utilization* - MLU), koji ima veliki praktični značaj u ISP mrežama. Prezentovani su i različiti pristupi rešavanju višekriterijumskih TE problema.

Princip rada *IntServ* [28] i *DiffServ* [29] QoS arhitektura, i QoS mogućnosti OpenFlow protokola razmatrani su u Glavi 3. Izvršena je klasifikacija osnovnih QoS problema rutiranja i diskutovana je njihova računaska kompleksnost.

U Glavi 4 predstavljen je originalni dizajn SDN kontrolera, kod kojeg je u cilju poboljšanja skalabilnosti logika za kontrolu saobraćaja organizovana u dva dijela: (i) modul za *online* rutiranje i kontrolu pristupa, i (ii) *offline-TE* modul, koji periodično optimizuje parametre za balansiranje saobraćaja. Predstavljen je matematički model *offline-TE* algoritma koji optimizacionu funkciju bazira na minimizaciji MLU, a kao optimizaciona ograničenja koristi zahtjeve saobraćajnih tokova u pogledu propusnosti, kašnjenja i pouzdanosti. Predloženo rešenje je upoređeno sa konvencionalnim i *state-of-the-art* rešenjima slične namjene u pogledu procenta odbijenih QoS zahtjeva, prosječne MLU vrijednosti i generisanog kontrolnog *overhead*-a. Algoritmi su upoređeni po više kriterijumima i pri različitim parametrima sistema, kako bi se izveli što potpuniji zaključci o njihovim performansama i ponašanju u različitim uslovima implementacije.

U Glavi 5 identifikovani su problemi koji nastaju kao rezultat rekonfiguracija mreže za potrebe inženjeringa saobraćaja. Predložen je bi-kriterijumski TE optimizacioni model koji teži simultano da minimizuje „troškove” rutiranja i učestanost rekonfiguracija. Pri tome, za razliku od analize u [19], polazna pretpostavka je da statistički model mrežnih

događaja nije unaprijed poznat. Predloženo rešenje nije bazirano na klasičnom rešavanju bi-kriterijumskog optimizacionog problema, gdje se kompromisno rešenje iz Pareto seta [30] bira statičkim težiranjem vrijednosti optimizacionih funkcija, kao što je to slučaj u [18]. Težiranje optimizacionih funkcija vrši se na osnovu ograničenja u pogledu srednje dozvoljene učestanosti rekonfiguracija, koje zadaje mrežni operator. Ovo ograničenje modelovano je virtuelnim baferom koji se vremenom popunjava kada srednja učestanost rekonfiguracija pređe kritičnu vrijednost. Za izbor najefikasnijeg Pareto optimalnog rešenja primijenjen je Lyapunov *drift-plus-penalty* algoritam [31], koji teži da minimizuje zauzetost virtuelnog bafera i „troškove” rutiranja. Na ovaj način, praćenjem zauzetosti virtuelnog bafera, indirektno je ostvareno dinamičko težiranje primarnih optimizacionih funkcija. S obzirom da analizirani bi-kriterijumski optimizacioni problem pripada klasi *NP-complete* problema, predložena je i heuristika koja omogućava efikasno rešavanje velikih problemskih instanci.

Rešenja predložena u ovoj disertaciji koja su se po više kriterijuma pokazala pogodnim za praktičnu primjenu, implementirana su na OpenFlow/SDN kontroleru. Detalji implementacije i rezultati eksperimentalne evaluacije dati su u Glavi 6.

U zaključku su date završne konstatacije i zapažanja sa osvrtom na realizovano istraživanje, njegov značaj, kao i mogućnosti rada na budućim istraživanjima u ovom pravcu.

Glava 1

Izazovi u razvoju ISP mreža naredne generacije

Mobilne i fiksne mreže, zajedno sa platformama za isporuku servisa, moraju odgovoriti na rapidan razvoj širokopojsnih pristupnih tehnologija, masovnu upotrebu podataka i zahtjeve vremenski-kritičnih aplikacija. Pri tome, treba da obezbijede fleksibilne i brže prihode od servisa, uz smanjene kapitalne i operativne troškove. Ekspanzija *Internet of Things* i mrežno-integriranih Cloud servisa dodatno ističe potrebu za novim računarskim paradigmama i servisno orijentisanim mrežama opšte namjene, koje karakteriše pouzdanost, malo kašnjenje, velika propusnost i obrada podataka u realnom vremenu [32]. Svi ovi zahtjevi su dio transformacije ka mrežama pete (5G) i narednih generacija [33].

Mreže pete generacije biće orijentisane prije svega na podršku industrijskog razvoja, a tek onda na krajnje korisnike, tj. stanovništvo [34]. U fokusu su različite grane industrije kao što su: proizvodnja, automobilska industrija, energetika, zdravstvo, pametni gradovi itd. Da bi efikasno podržale rastuću potražnju za postojećim servisima i širok spektar novih industrijskih aplikacija na istoj infrastrukturi, 5G mreže zahtijevaju arhitekturne izmjene u odnosu na trenutne implementacije. Prema trenutnim 5G tehničkim specifikacijama, transformacija mrežne arhitekture ići će u pravcu mrežne softverizacije [35]. Ključne tehnologije koje će omogućiti ovaj trend su SDN [36], NFV [37] (*Network Function Virtualisation*), Fog [38] i *Edge computing* [39].

U ovom poglavlju dat je pregled novih servisnih scenarija koji su motivisali 5G inicijativu. Takođe, predstavljeni su QoS zahtjevi koje 5G mreže treba da ispune. Zatim, objašnjen je trend mrežne softverizacije kao i uloga SDN i NFV tehnologija u 5G sistemima.

Na kraju, identifikovane su mogućnosti za poboljšanje servisnog portfolia u softverski definisanim ISP (SD-ISP) mrežama naredne generacije.

1.1 5G sistemi: motiv i zahtjevi

Očekivanja su da će 5G mreže odigrati ključnu ulogu u globalnoj optimizaciji pokrivanja potrebe različitih tržišta na ekonomičan i efikasan način. Početak komercijalne upotrebe 5G mreža predviđen je za 2020. godinu [33]. Iako je njihova budućnost još uvijek neizvjesna, postoji konsenzus o tome da 5G mreže neće biti samo prirodna evolucija današnjih mobilnih mreža. Među trendovima koji motivišu razvoj 5G tehnologija posebno se ističu:

- Drastičan porast mobilnog saobraćaja. CISCO predviđa da će se ukupni mobilni saobraćaj povećati oko 7 puta u periodu od 2017. do 2022. godine [40].
- Usvajanje IoT rešenja u različitim granama industrije. Predviđanja su da će broj povezanih IoT uređaja do 2023. godine iznositi 31.4 milijardi [41].
- Kontinuitet pojavljivanja novih servisa (npr. 3D video visoke rezolucije, proširena realnost, mobilni Cloud, aplikacije za taktilni Internet) i komunikacionih scenarija (npr. *Machine-to-Machine* komunikacije u ultra gustim mrežama sa visokim stepenom mobilnosti [42]).

Navedeni trendovi nameću veliki broj izazova na koje je neophodno odgovoriti povećanjem kapaciteta mreže, poboljšanjem energetske efikasnosti i iskorišćenosti spektra, kao i skalabilnijim metodama upravljanja mrežnim resursima. S tim u vezi, definisano je više tehničkih zahtjeva za mreže naredne generacije, uzimajući u obzir perspektivu korisnika, mreže i servisa [43]:

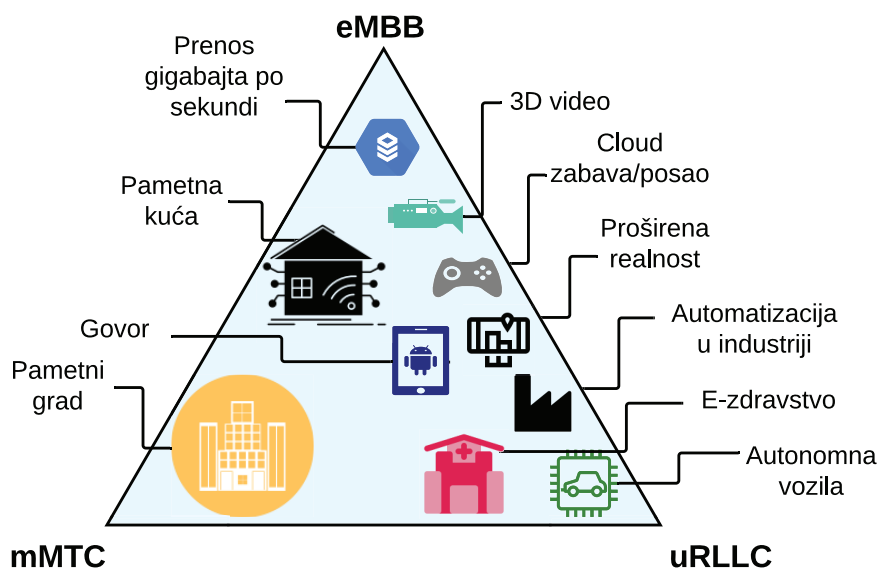
1. **Ultra velike brzine prenosa i ultra malo kašnjenje** - Oko 1000 puta veće brzine prenosa od onih koje pruža LTE (*Long Term Evolution*) mreža, i ultra mala kašnjenja reda nekoliko milisekundi.
2. **Masovna konektivnost** - Podrška za 1000 puta veći broj uređaja u odnosu na današnje mreže.

3. **Fleksibilna i inteligentna mreža** - Softverski-definisana mrežna arhitektura, mogućnost analize podataka u realnom vremenu i pružanje personalizovanih i inteligentnih servisa.
4. **Pouzdan rad** - Obezbijediti dostupnost i pouzdanost mreže na nivou od 99%, kao i mogućnost automatske rekonfiguracije u slučaju kvara.
5. **Energetski i ekonomski isplativa infrastruktura** - 5G sistemi bi trebalo da budu 50-100 puta energetski efikasniji od LTE sistema, i ekonomičniji u pogledu potrebnih ulaganja u infrastrukturu.

1.1.1 Servisne grupe i ključni indikatori performansi

5G tehnologije imaju ambiciju da podrže širok spektar servisa i aplikacija različitih zahtjeva. Više standardizacionih tijela [35, 44, 45] definisalo je pionirske scenarije upotrebe 5G sistema, koji su poslužili kao osnova za određivanje 5G zahtjeva. Iako je za 5G sisteme generalno lako naći ogroman broj primjena, 5G servisi se obično mogu kategorizovati u sledeće tri grupe [33]:

- **Poboljšanje mobilne širokopojasne mreže** (eMBB - *Enhanced Mobile Broadband*). eMBB ima za cilj da zadovolji potrebe ljudi za digitalnim načinom života, i fokusira se na aplikacije koje imaju visoke zahtjeve za propusnim opsegom, kao što su ultra HD (*High Definition*) video, virtuelna i proširena stvarnost [46].
- **Ultra pouzdana komunikacija malog kašnjenja** (uRLLC - *Ultra Reliable Low Latency Communication*). Cilj uRLLC servisne grupe je da podrži digitalnu transformaciju industrije. Fokusira se na servise osjetljive na kašnjenje, kao što su bezbjednost u asistiranim i autonomnim vozilima, monitoring i upravljanje u realnom vremenu, udaljene medicinske intervencije itd. Današnje mreže ne podržavaju ove servise.
- **Masovne mašinske komunikacije** (mMTC - *Massive Machine Type Communications*). Mašinske komunikacije se odnose na uređaje koji komuniciraju bez ljudske intervencije. mMTC grupa teži da ispuni zahtjeve IoT aplikacija koje periodično prikupljaju manje količine podataka iz baterijski napajanih senzorskih mreža, i obrađuju ih sa ciljem unaprijeđenja nekog proizvodnog procesa ili kvaliteta života



Slika 1.1: 5G servisne grupe.

u određenoj sredini (npr. pametni gradovi ili pametna agrikultura). Današnje mreže nisu optimizovane za pružanje ove klase servisa, naročito ako se uzme u obzir njihovo ograničenje u pogledu maksimalnog broja povezanih uređaja po jedinici površine.

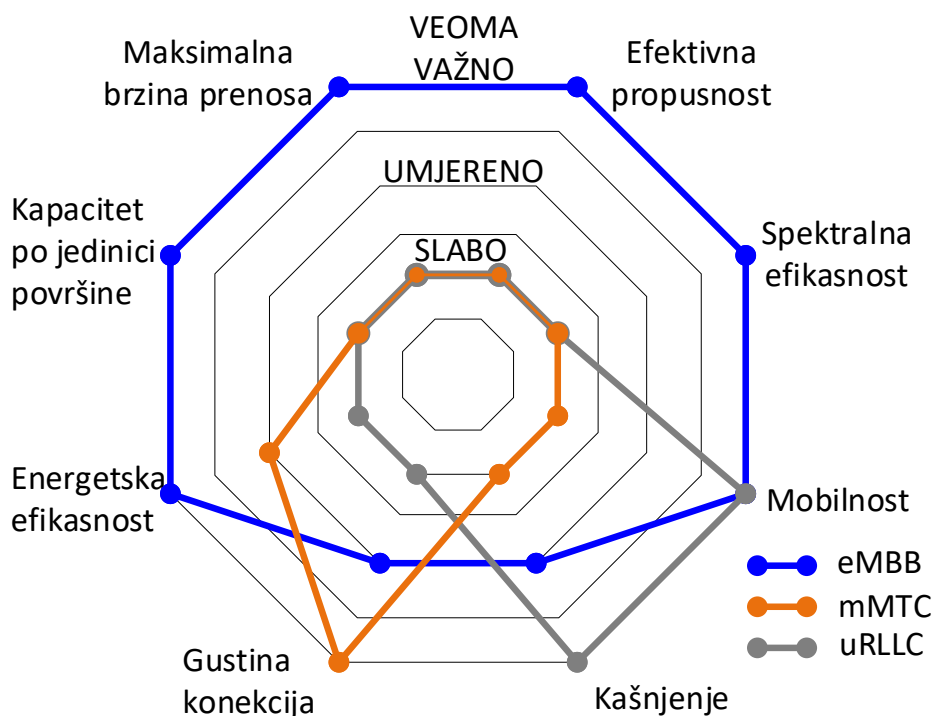
Navedene servisne grupe imaju različite i suprotstavljene zahtjeve koji problematizuju njihovu koegzistenciju [47]. Iz tog razloga, integracija pomenutih servisnih grupa na zajedničkoj infrastrukturi smatra se jednim od ključnih izazova za mreže naredne generacije [33]. Na Slici 1.1 prikazani su primjeri servisnih scenarija za svaku od grupa [35].

Na osnovu predloženih servisnih scenarija, Međunarodna unija za telekomunikacije (*International Telecommunications Union* - ITU) definisala je ključne indikatore performansi i njihove ciljne vrijednosti za 5G mreže. U Tabeli 1.1, 5G (IMT-2020) i 4G (IMT-Advanced) mreže upoređene su u pogledu ciljnih performansi. Sa Slike 1.2 se može primijetiti da je značaj različitih indikatora performansi različit za svaku od servisnih grupa [48].

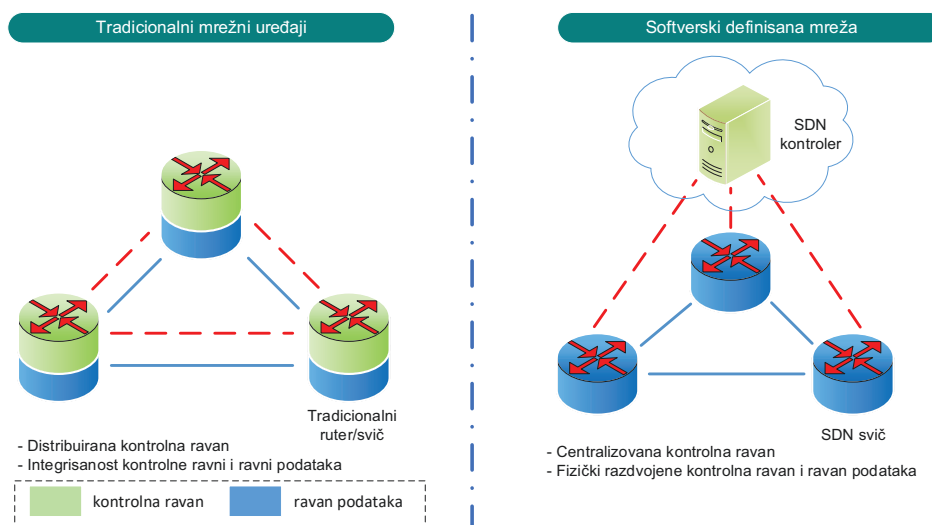
Pored toga što treba da pruži definisane performanse, od 5G tehnologija se očekuje veća fleksibilnost i skalabilnost. Na putu ka ostvarenju ovih ciljeva, softverizacija mreže će imati primarnu ulogu [49, 50].

Tabela 1.1: Poređenje IMT-2020 i IMT-Advanced sistema

KPI	IMT-2020	IMT-Advanced
Maksimalna brzina prenosa (<i>Gbps</i>)	20	1
Kapacitet po jedinici površine (<i>Mbps/m²</i>)	10	0.1
Gustina konekcija (<i>br. uređaja /km²</i>)	10 ⁶	10 ⁵
Kašnjenje (<i>ms</i>)	1	10
Mobilnost (<i>km/h</i>)	500	350
Propusnost (<i>Mbps</i>)	100-1000	10
Spektralna efikasnost	3x	1x
Energetska efikasnost	100x	1x



Slika 1.2: Značaj indikatora performansi za različite servisne grupe.



Slika 1.3: Tradicionalna i SDN mrežna arhitektura.

1.2 Nedostaci tradicionalne mrežne arhitekture

Današnje mrežne arhitekture nisu u stanju da odgovore na stroge QoS zahtjeve servisnih scenarija diskutovanih u prethodnoj sekciji uslijed ograničavajućih svojstava distribuirane kontrole ravni i integrisanosti mrežnih funkcija sa hardverom. Uticaj ovih ograničenja detaljnije je analiziran u nastavku ove sekcije.

1.2.1 Ograničenja distribuirane kontrolne ravni

Funkcionalnosti mreže organizovane su u ravni podataka, kontrole i upravljanja. Ravan podataka sastoji se od hardvera koji je zadužen za efikasno prosleđivanje saobraćaja. Kontrolnu ravan čine protokoli koji vrše popunjavanje tabela prosleđivanja elemenata u ravni podataka. Ravan upravljanja uključuje softverske servise, kao što su SNMP (*Simple Network Management Protocol*) alati, koji se koriste za udaljeni monitoring i konfigurisanje kontrolnih funkcija [36]. Mrežna politika definiše se u ravni upravljanja, kontrolna ravan sprovodi tu politiku, a ravan podataka je izvršava prosleđujući saobraćaj na odgovarajući način.

U tradicionalnim IP mrežama, kontrolna ravan i ravan podataka su objedinjene unutar mrežnih uređaja, i čitava arhitektura je veoma decentralizovana (Slika 1.3). Takvo rešenje se smatralo pogodnim za početni dizajn Interneta, jer se činilo kao najbolji način za garantovanje pouzdanosti prenosa podataka, koja je bila ključni cilj dizajna. Štaviše,

ovaj pristup se pokazao prilično efikasnim kada su performanse mreže u pitanju, i uspio je da podrži rapidan rast u brzinama prenosa. Međutim, krajnji rezultat danas je veoma kompleksna i relativno statična arhitektura sa aspekta upravljanja [51], koja nije u stanju da se izbori sa strogim i često suprotstavljenim zahtjevima modernih aplikacija, trendovima širenja virtuelizacije i Cloud servisa. Upravljanje se obavlja na vrlo niskom nivou, koji zahtijeva dobro poznavanje mrežne opreme. Tako, na primjer, svako dodavanje ili uklanjanje uređaja iz mreže zahtijeva manuelnu rekonfiguraciju velikog broja svičeva, rutera, *firewall*-a, web portala za autentifikaciju, kao i ažuriranje fajlova za kontrolu pristupa, VLAN (*Virtual Local Area Network*) opcija, QoS politika i ostalih mehanizama zasnovanih na protokolima. Prilikom rekonfiguracije potrebno je razmotriti mrežnu topologiju i model uređaja. Za komplikovanje zadatke potrebno je i detaljnije znanje o mreži. Mehanizmi za automatsku rekonfiguraciju gotovo ne postoje. Greške usled manuelne konfiguracije su veoma česte i mogu imati katastrofalne posledice. Jedan pogrešno podešen uređaj može da uzrokuje neželjeno ponašanje čitave mreže, uključujući između ostalog gubitke paketa, petlje rutiranja, uspostavljanje neželjenih putanja ili kršenje servisnih ugovora [25].

Usled pomenutih ograničenja provajderi servisa suočavaju se sa velikim izazovima kada je skaliranje mreže u pitanju. Servisi kompanija kao što su Google, Microsoft i Facebook, često su zasnovani su na korišćenju kompleksnih algoritama za paralelnu obradu podataka, koji su distribuirani preko većeg broja fizičkih servera. Kako se djelokrug aplikacija krajnjih korisnika povećava (na primjer, pretraživanje i indeksiranje čitavog web-a i vraćanje rezultata pretrage nazad), broj računarskih elemenata koji učestvuje u obradi korisničkog zahtjeva naglo raste, a podaci koje tom prilikom međusobno razmijene mogu se mjeriti u petabajtima. Servisima ovog tipa potrebne su hiperskalabilne mreže koje mogu podržati efektivno i ekonomično povezivanje stotina hiljada, potencijalno miliona, fizičkih servera. Taj nivo skalabilnosti ne mogu pružiti tradicionalne mrežne arhitekture u kojima se donošenje kontrolnih odluka i sprovođenje administrativnih politika vrši na nivou pojedinačnih uređaja.

1.2.2 Ograničenja hardverske implementacije mrežnih funkcija

Današnje mreže sadrže veliki broj hardverskih uređaja na kojima su vertikalno integrisane različite mrežne funkcije (npr. rutiranje, balansiranje opterećenja, dubinska inspekcija paketa, *firewall*). Softver koji je implementiran na ovim uređajima je zatvorenog

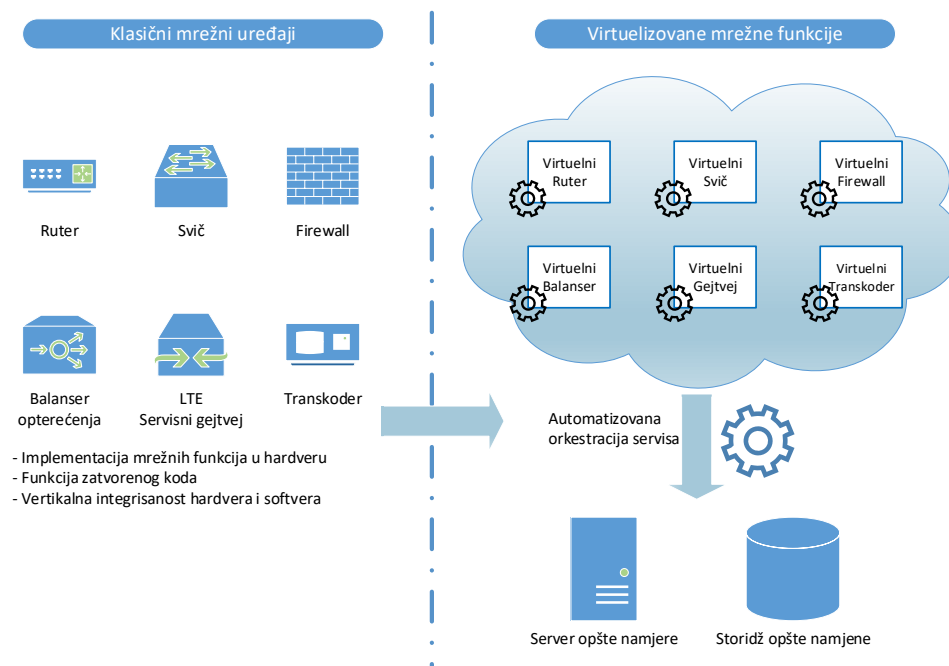
koda, specifičan za različite proizvođače, i nudi vrlo ograničene mogućnosti za udaljenu rekonfiguraciju. Vertikalna integritetnost hardvera i softvera sputava razvoj i implementaciju novih mrežnih funkcionalnosti [37]. U situaciji kada mrežni operator identifikuje potrebu za poboljšanjem neke mrežne funkcionalnosti neophodno je izvršiti nabavku nove opreme. Iz ovih razloga, resursi u današnjim mrežama se statički dimenzionišu u fazi planiranja tako da je mreža u stanju da podrži maksimalno saobraćajno opterećenje predviđeno za period od nekoliko narednih godina. Jednom kada se saobraćajno opterećenje približi mrežnom kapacitetu (npr. 70% iskorišćenosti), vrši se *upgrade* opreme kako bi mreža mogla da odgovori na budući zahtjeve. Stoga, operatori su prinuđeni na velika ulaganja u infrastrukturu iako su mrežni resursi neiskorišćeni najveći dio vremena. Sa druge strane, bilo kakva nepredviđena distorzija saobraćaja može izazvati probleme u funkcionisanju mreže.

1.3 Softverizacija mreže

Koncept softverizacije mreže predstavlja specifičan pristup dizajnu, implementaciji, upravljanju i održavanju mrežne opreme ili njenih komponenti programiranjem softvera. Osmišljen je sa ciljem stvaranja uslova za jednostavan redizajn mreže i servisnih arhitektura, optimizaciju mrežnih performansi i troškova. Softverizacijom mreže vrši se radikalna transformacija konvencionalne mrežne arhitekture, pa je usvajanje ovog koncepta jedan od najrelevantnijih tehničkih izazova u telekomunikacionoj industriji i naučnoj zajednici. U ovoj sekciji predstavljene su dvije ključne tehnologije na kojima je ovaj pristup baziran: NFV i SDN.

1.3.1 NFV

Ključna ideja NFV tehnologije je razdvajanje mrežnih funkcija od hardvera, i stvaranje uslova za izvršavanje mrežnih funkcija u formi softverskih komponenti na serverima opšte namjene [37, 52] (Slika 1.4). Serveri mogu biti smješteni na mrežnim čvorištima ili u Cloud-u. U ovom kontekstu NFV se oslanja na virtualizaciju servera. Za razliku od virtualizacije servera, virtualne mrežne funkcije (eng. *Virtual Network Functions* – VNFs) se mogu sastojati od jedne ili više virtualnih mašina na kojima se izvršavaju različiti softveri i procesi koji zamjenjuju ulogu određenih hardverskih uređaja. Po pravilu, više virtualnih



Slika 1.4: Virtualizacija mrežnih funkcija.

mrežnih funkcija se koristi u sekvenci kako bi se korisnicima pružili servisi od interesa. Centralni element NFV mrežnog okruženja je platforma za orkestraciju servisa, koja omogućava efikasno instaliranje i monitoring virtuelnih mrežnih funkcija (npr. transkoder, NAT, *firewall*, keširanje) na NFV infrastrukturi [53, 54]. Da bi se ispunili određeni *real-time* zahtjevi, za realizaciju nekih mrežnih funkcija koriste se ipak specijalizovane hardverske komponente. Ove komponente vrše složene računske operacije i vremenski kritične zadatke koje nije moguće realizovati na serverima opšte namjene.

Ključne prednosti NFV tehnologije su:

1. **Smanjeni kapitalni troškovi.** Smanjenje kapitalnih troškova rezultat je primjene servera opšte namjene i fleksibilnog modela naplate servisa (tzv. *pay-as-you-go* model). Serveri opšte namjene su, obzirom na njihovu masovnu proizvodnju, značajno jeftiniji od specijalizovanog hardvera. Sa druge strane, uz *pay-as-you-go* model naplate za IaaS (*Infrastructure as a Service*) servis Cloud provajdera, mrežni operatori mogu izbjeći ogromna ulaganja u infrastrukturu prilikom implementacije ili nadogradnje mreže. Takođe, mrežni resursi ne moraju više biti predimenzionisani, čime se postiže njihova bolja iskorisćenost.
2. **Smanjeni operativni troškovi.** NFV omogućava operatorima da potpuno automa-

tizuju upravljačke zadatke kao što su kreiranje i skaliranje mrežnih servisa i funkcija. Takođe, NFV doprinosi smanjenju potrošnje energije prilagođavanjem mrežne opreme radnom opterećenju.

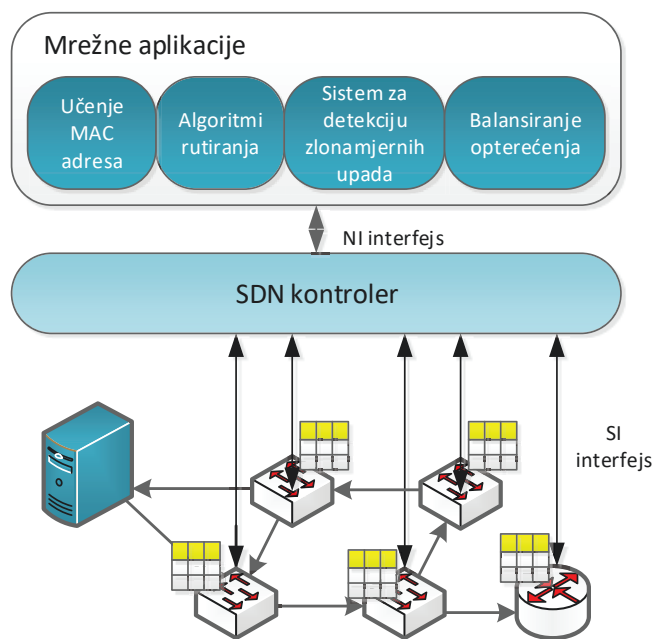
3. **Fleksibilnost upravljanja i skalabilnost.** Operatori mogu dinamički prilagođavati resurse saobraćajnom opterećenju, a pri tom garantovati da zahtijevane performanse neće biti narušene. Pored toga, pozicije mrežnih funkcija mogu se fleksibilno mijenjati. Na primjer, operator može kreirati instancu mrežnog servisa na svakom pristupnom čvoru mreže u cilju poboljšanja kvaliteta servisa, tj. smanjenja propagacionog kašnjenja. Servisi (u vidu sekvence više mrežnih funkcija) mogu se dodavati i brisati relativno brzo, u roku od nekoliko minuta, bez ljudske intervencije.

1.3.2 Softverski definisano umrežavanje

Logička šema SDN arhitekture prikazana je na Slici 1.5. Fundamentalna ideja ovog rešenja je razdvajanje kontrolne ravni od ravni podataka. Mrežna inteligencija je centralizovana na programabilnom SDN kontroleru koji se kao softverska platforma može izvršavati na serveru opšte namjene. Sama mreža je programabilna, i njeno ponašanje se kontroliše aplikacijama koje se izvršavaju na kontroleru. Ovo je osnovna odrednica SDN-a, koja se smatra njegovom glavnom prednošću.

Važno je naglasiti da SDN arhitektura podrazumijeva samo logičku centralizaciju kontrolne ravni. Kontrolne funkcije se mogu izvršavati na više fizički razdvojenih kontrolera u cilju pružanja odgovarajućeg nivoa performansi, skalabilnosti i pouzdanosti [55, 56]. Prednosti logičke centralizacije kontrolnih funkcija su mnogostruke:

- Sprovođenje mrežne politike je znatno jednostavnije i manje sklono greškama;
- Kontrolni programi automatski reaguju na promjene u mreži;
- Mrežni uređaji više ne moraju da izvršavaju ogroman broj protokola, već samo prihvataju instrukcije SDN kontrolera;
- SDN kontroler ima uvid u stanje svih mrežnih resursa, što omogućava razvoj naprednijih mrežnih funkcija, servisa i aplikacija.

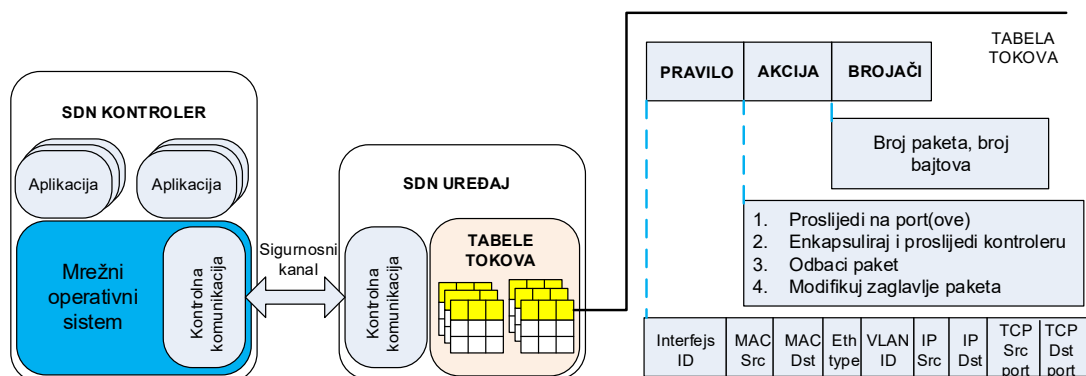


Slika 1.5: SDN arhitektura.

Kao što je prikazano na Slici 1.5, SDN kontroler podržava dva API-ja (*Application Programming Interface*): SI (*Southbound Interface*), koji definiše protokol za komunikaciju sa uređajima u ravni podataka, i NI (*Northbound Interface*) koji se koristi za razvoj samih aplikacija. Tipično, NI interfejs apstrahuje set instrukcija koje koristi SI interfejs za programiranje mrežnih uređaja. Uz otvoreni NI interfejs, SDN aplikacije mogu se dizajnirati i pokretati bez uvida u detalje implementacije mrežnih servisa i mogućnosti koje koriste. Kada je SI interfejs u pitanju, na raspolaganju je više rešenja, uključujući: OpenFlow [7], ForCES [57], POF [58], [59] i OpenState [60]. Danas se u komercijalnim mrežama i za potrebe istraživanja dominantno koristi OpenFlow, što je i izbor napravljen za potrebe ove disertacije.

1.3.3 OpenFlow

OpenFlow je prvi standardizovani protokol za komunikaciju između kontrolne ravni i ravni podataka SDN arhitekture. Omogućava direktan pristup ravni podataka i manipulaciju mrežnim uređajima poput svičeva i rutera, bilo fizičkih bilo virtuelnih. OpenFlow se izvršava na obje strane interfejsa između uređaja za prosleđivanje saobraćaja i mrežnog kontrolera. Kontroler upravlja saobraćajem na nivou individualnih ili agregiranih saobraćajnih tokova. Uz kontrolu visoke granularnosti moguće je dinamički adaptirati



Slika 1.6: Elementi OpenFlow mreže.

konfiguraciju mreže zahtjevima različitih aplikacija.

OpenFlow svič

Na Slici 1.6 prikazana je arhitektura OpenFlow mreže čiji su glavni elementi OpenFlow svičevi i mrežni kontroler. OpenFlow svič predstavlja generalizaciju Ethernet sviča kod kojeg je ravan podataka odvojena od kontrolne ravni i apstrahovana tabelom tokova. Eksterna kontrola svičeva se obavlja putem sigurnosnog kanala.

OpenFlow koristi tzv. tabele tokova za reprezentaciju tabela prosleđivanja svičeva različitih proizvođača. Svaki od zapisa u tabeli tokova sastoji se od tri dijela:

- *Match* pravila – skup polja iz zaglavlja paketa koji definiše karakteristike toka;
- Akcije – određuju postupak obrade paketa;
- Brojača – čuvaju informacije o broju paketa ili bajtova prenesenih na nivou svakog interfejsa i toka, kao i informacije o vremenu proteklom od pristizanja poslednjeg paketa toka.

OpenFlow svič može imati više tabela tokova. Polja zaglavlja svakog primljenog paketa upoređuju se sa *match* poljima u tabeli tokova. Ukoliko se ustanovi poklapanje, preduzima se specificirana akcija (na primjer prosleđivanje paketa na određeni port ili odbacivanje paketa). U suprotnom, paket se prosleđuje kontroleru, koji definiše sledeće korake obrade na osnovu implementiranog kontrolnog algoritma.

Iako OpenFlow svič omogućava procesuiranje paketa linijskim brzinama, ne omogućava kontrolu na nivou paketa. Naime, razdvajanjem kontrolne ravni od ravni podataka izgubljena je mogućnost brzog procesuiranja kontrolnih akcija. Stoga, kontrolne akcije ne preduzimaju se za svaki paket, već se u skladu sa različitim vrijednostima u zaglavlju paketi grupišu u tokove, i kontrola se vrši na nivou toka. Kada OpenFlow svič primi paket koji ne pripada ni jednom od tokova definisanih u tabeli, prosleđuje ga kontroleru. Kontroler određuje kako taj paket treba biti tretiran i dodaje odgovarajuće instrukcije u tabele tokova OpenFlow svičeva. Svaki sledeći paket istog toka neće se prosleđivati kontroleru već će se obrađivati u skladu sa primljenim instrukcijama. Na ovaj način napravljen je kompromis između fleksibilnosti i granularnosti kontrole. Ipak, ovaj kompromis predstavio je novi važan koncept – tok paketa. Za definiciju saobraćajnog toka kontroler može da koristi različita polja zaglavlja koja uključuju parametre drugog, trećeg i četvrtog TCP/IP nivoa. Na Slici 1.6 prikazan je set polja koji podržava OpenFlow verzija 1.0. Novije verzije protokola direktno podržavaju znatno širi spektar parametara, dok je na indirektnan način, primjenom tzv. eksperimentalnih *match* polja, moguće vršiti kontrolu na osnovu proizvoljne grupe bajtova paketa [7]. *Match* polja zajedno obezbjeđuju definiciju toka, ali pored toga nude mogućnosti agregacije tokova i separacije različitih tipova saobraćaja. Agregacija se ostvaruje tako što se prilikom kreiranja zapisa u tabeli neka od *match* polja ostave praznim. U tom slučaju, smatra se da bilo koja vrijednost tog polja u zaglavlju paketa zadovoljava uslov poklapanja. Na primjer, tok definisan određenom izvorišnom IP adresom obuhvatiće sav saobraćaj koji potiče od uređaja sa tom IP adresom. Ukoliko se dodatno specificira 80 kao broj TCP porta, ovaj tok obuhvataće HTTP saobraćaj koji generiše taj uređaj. Na isti način i bilo koji drugi tip saobraćaja koji se uvijek očekuje na određenom portu jednostavno se može definisati kao poseban tok.

Kada su akcije koje se sprovode nad paketima u pitanju, OpenFlow trenutno podržava sledeće mogućnosti:

1. Prosleđivanje na određeni port. Na ovaj način se paketi rutiraju kroz mrežu.
2. Enkapsulacija i prosleđivanje paketa kontroleru.
3. Odbacivanje paketa toka. Može se koristiti u sigurnosne svrhe za sprečavanje DDoS (*Distributed Denial-of-Service*) napada i kontrolisanje *broadcast* saobraćaja.
4. Modifikacija polja u zaglavlju paketa.
5. Dodavanje i uklanjanje MPLS labela.

6. Prosleđivanje paketa u određenu *group* tabelu. Koncept *group* tabele donosi novi set mogućnosti za prosleđivanje saobraćaja, kao što su *multicast* prenos, rerutiranje u slučaju ispada linka, agregacija linkova, balansiranje opterećenja itd. Na ovaj način moguće je efikasno sprovoditi isti set akcija nad grupom tokova.
7. Monitoring dolazne brzine toka.
8. Plavljenje, tj. slanje kopije paketa na sve portove osim onog na kojem je paket primljen.
9. Mapiranje paketa u određeni bafer izlaznog porta.

Sigurnosni kanal

Sigurnosni kanal je interfejs putem kojeg OpenFlow svič razmjenjuje poruke sa kontrolerom. Ove poruke moraju zadovoljavati format definisan OpenFlow protokolom, i prenose se TCP konekcijom u binarnom formatu. OpenFlow definiše tri tipa kontrolnih poruka: kontroler-svič poruke, asinhronone i simetrične poruke. Kontroler-svič poruke generiše kontroler, i one ne zahtijevaju uvijek odgovor sviča. Koriste se za konfigurisanje sviča, upravljanje tabelom tokova, dobijanje informacija o njenom sadržaju i karakteristikama sviča. Asinhronone poruke šalje svič kontroleru da bi ga obavijestio o nekom događaju u mreži. Jedna od takvih situacija je kada svič primi paket koji ne pripada ni jednom od tokova definisanih tabelom. Tada se asinhronom porukom specijalog tipa (*Packet-in* poruka) kontroleru šalje paket (ili samo jedan njegov dio) na ispitivanje. Slanje asinhronih poruka može biti inicirano i isticanjem roka važenja zapisa u tabeli tokova, promjenom stanja nekog od portova ili nekim drugim problemom u mreži. Konačno, simetrične poruke šalju se u oba smjera i koriste se za detekciju problema konekcije između OpenFlow sviča i kontrolera.

OpenFlow kontroler

Kontroler je ključni uređaj OpenFlow arhitekture koji sprovodi kontrolnu politiku tako što distribuira odgovarajuće instrukcije mrežnim uređajima. Odgovoran je za donošenje odluka o načinu procesuiranja paketa, dodaje i briše zapise u tabeli tokova preko sigurnosnog kanala. Obično se instalira na serveru povezanom na mrežu, pri čemu se može

konfigurisati da radi u centralizovanom ili distribuiranom modu. Prva konfiguracija podrazumijeva da jedan kontroler upravlja svim svičevima u mreži, dok su u drugom slučaju dva ili više kontrolera zaduženi za kontrolu različitih segmenata mreže. Centralizovana konfiguracija je problematična jer se u slučaju otkaza kontrolera sve operacije u mreži prekidaju. Kod distribuirane konfiguracije svaki od kontrolera mora imati identičnu „sliku” mreže u realnom vremenu kako ne bi došlo do degradacije mrežnih performansi.

Trenutno je dostupno više različitih *open-source* implementacija kontrolera koji su bazirani na Python, C, C++ ili Java programskom jeziku [61–66].

1.4 Primjena SDN-a u ISP mrežama naredne generacije

U ovoj sekciji razmotrene su mogućnosti koje SDN nudi za unaprijeđenje i proširenje servisnog portfolia ISP mreža. Identifikovana su tri nova servisna scenarija: (i) *Bandwidth on Demand* (BoD); (ii) virtuelizacija transportne mreže; i (iii) višenivovska optimizacija IP/optičke transportne mreže [67].

1.4.1 Bandwidth on Demand (BoD) servis u ISP mrežama

Pod uticajem sve češćih *big data* projekata, Cloud umrežavanja i *ad-hoc* među-kompanijskih kolaboracija, za ISP mreže karakteristične su velike varijacije saobraćaja, sa pikovima koji mogu trajati od nekoliko minuta do nekoliko sati. Na SDN infrastrukturi ISP provajderi mogu ponuditi fleksibilni BoD servis, koji omogućava korisnicima da dinamički rezervišu i skaliraju resurse za svoje konekcije po potrebi. Pri tome, korisnici mogu specificirati zahtjeve u pogledu propusnosti, kašnjenja i pouzdanosti. Dodatno, može se zahtijevati uspostavljanje/skaliranje konekcije u realnom vremenu ili u nekom unaprijed definisanom terminu, potencijalno sa različitim tarifama naplate u cilju efikasnijeg korišćenja resursa. SDN BoD servis donosi obostrane benefite mrežnim operatorima i njihovim korisnicima. Korisnicima su na raspolaganju velike brzine prenosa na privremenoj osnovi, kada i gdje im je to potrebno, čime se eliminiše potreba za iznajmljivanjem zasebnih privatnih linija. Sa druge strane, ovaj poslovni model odgovara i operatorima, jer mogu iznajmljivati iste mrežne resurse većem broju klijenata u različitim vremenskim intervalima.

1.4.2 Virtuelizacija transportne mreže

Virtuelizacijom transportne mreže (VTN – *Virtual Transport Network*) ISP provajderi mogu napraviti dodatni iskorak u odnosu na BoD servis, nudeći klijentima mogućnost upravljanja saobraćajem na rezervisanim mrežnim resursima. Na primjer, direktnom kontrolom virtuelne transportne mreže klijent može dinamički preraspodjeljivati dodijeljeni kapacitet različitim aplikacijama. SDN kontroler bi u tom slučaju bio zadužen za kreiranje apstrakcije zasebene virtuelne mrežne topologije za svakog od klijenata, kao i mapiranje njihovih kontrolnih odluka na fizičku infrastrukturu. Step en apstrakcije virtuelne topologije, kontrolne privilegije klijenta i performanse konekcija (npr. kašnjenja, propusnost) definiše administrativna politika operatora. Provajder distribuiranih Cloud servisa može koristiti VTN servis za orkestraciju BoD zahtjeva između klijentskih uređaja i svojih *data* centara. Na primjer, u vrijeme nekog sportskog događaja, Cloud provajder može dodijeliti veći kapacitet tokovima koji prenose video sadržaj iz sportske dvorane do *data* centra gdje se video obrađuje i odakle se dalje distribuira korisnicima [67]. Iz perspektive ISP provajdera, ključne prednosti VTP servisa su smanjeni troškovi ulaganja u infrastrukturu usled mogućnosti dinamičkog dijeljenja mrežnih resursa (linkova, optičkih vlakana, talasnih dužina, itd.) između više klijenata, kao i mogućnost sticanja novih prihoda s obzirom na relevantnost ovog servisa za veliki broj poslovnih korisnika.

1.4.3 Više-nivovska optimizacija transportne mreže

ISP mreže se obično sastoje iz dva dijela: IP/MPLS mreže i pasivne optičke mreže. Trenutno se IP/MPLS mreže koriste za većinu servisa zbog mogućnosti fleksibilne kontrole rutiranja i kvaliteta servisa. Sa druge strane, optičke mreže predstavljaju ekonomičnu nadgradnju mrežnih kapaciteta, koja sve više dobija na značaju usled drastičnog porasta intenziteta i varijabilnosti saobraćaja. U konvencionalnim ISP mrežama, IP i optički nivo transportne mreže se dizajniraju i implementiraju nezavisno. Upravljanje ovim mrežama vrše odvojeni mrežni entiteti, između kojih ne postoji automatska koordinacija prilikom donošenja odluka o rutiranju ili inženjeringu saobraćaja. SDN predstavlja obećavajuće rešenje za integraciju kontrolne ravni IP i optičke transportne mreže. Više-nivovska kontrolna ravan omogućava ISP provajderima da automatizuju mnoge procese u IP i optičkom domenu, kao i da drastično smanje vrijeme potrebno za pružanje servisa. Takođe, omogućava razmjenu informacija između nivoa, čime se postiže dodatni nivo zaštite za IP mreže u slučaju kvara, dok se u optičkom nivou obezbjeđuju uslovi za

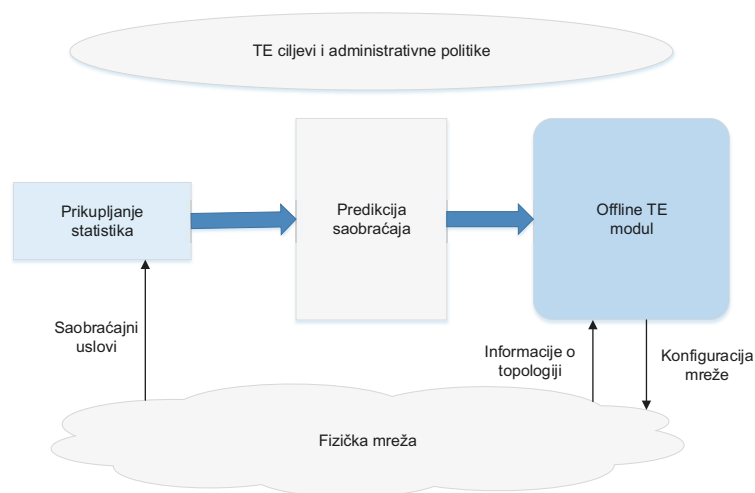
restauraciju bez degradacije performansi. SDN kontroler predstavlja pogodnu platformu za implementaciju više-nivovskih optimizacionih algoritama čiji je zadatak pronalaženje optimalnih mrežnih konfiguracija koje maksimizuju iskorišćenost resursa u oba domena istovremeno [5]. Na ovaj način smanjuje se potreba za dodavanjem novih fizičkih resursa u obje mreže, i povećava obim servisa koje postojeći mrežni resursi mogu da podrže.

Glava 2

Upravljanje saobraćajem

Današnji Internet pruža *best-effort* model servisa za mnoge popularne aplikacije kao što su web, e-mail, fajl transfer, IP-TV, online igrice i video *streaming*. Kako se Internet razvija u pogledu broja povezanih uređaja i složenosti, upravljanje saobraćajem postaje sve važniji i izazovniji zadatak. Dok QoS algoritmi rutiranja pokušavaju da pruže bolji servis tokovima u uslovima zagušenja, TE tehnike raspoređuju saobraćajne tokove tako da se izbjegne zagušenje usled neravnomjerne iskorišćenosti mreže [68]. Dakle, TE tehnike su fokusirane više na poboljšanje performansi i kapaciteta mreže. Inženjering saobraćaja podrazumijeva kontrolu saobraćajnog opterećenja svake rute u mreži, što se direktno odražava na nivo zagušenja i efikasnost korišćenja resursa. Zbog svoje praktične važnosti, aktivno je područje istraživanja već dugi niz godina. U ovom poglavlju dat je osvrt na trenutnu praksu inženjeringa saobraćaja u ISP mrežama i najznačajnija istraživanja u ovoj oblasti.

U Sekcijama 2.1 i 2.2 predstavljena su TE rešenja koja odgovaraju različitim tehnološkim pristupima: MPLS TE i IP TE, redom. TE tehnike su dodatno klasifikovane po kriterijumu dinamike izvršavanja na *online* i *offline* tehnike. Osnovna razlika između *offline* i *online* TE tehnika je (ne)dostupnost saobraćajne matrice i vremenski okvir manipulacije saobraćajem. *Offline* TE rešenja periodično vrše optimalno mapiranje prognozirane matrice saobraćaja na fizičku topologiju (Slika 2.1). Optimizacija mrežne konfiguracije vrši se u dužim vremenskim okvirima, tipično jednom dnevno, nedeljno ili mjesečno. Glavna slabost ovog pristupa je što nije moguće adaptirati konfiguraciju mreže dinamici saobraćaja i događajima kao što su ispadi linkova i rutera. *Online* TE tehnike dinamički kontrolišu mrežu ne zahtijevajući informacije o očekivanoj matrici saobraćaja. Osnovni zadatak



Slika 2.1: Offline inženjering saobraćaja.

online algoritama je optimalno balansiranje opterećenja linkova prilikom rutiranja novih saobraćajnih tokova (jedan po jedan) tako da mreža može zadovoljiti proizvoljnu buduću saobraćajnu potražnju bez ulaska u zagušenje. U nekim slučajevima, *online* TE algoritmi vrše rerutiranje postojećih tokova u mreži u cilju ispunjenja QoS zahtjeva novih tokova. Međutim, rerutiranje je obično ograničeno na mali broj tokova kako se ne bi ugrozila mrežna stabilnost i kvalitet servisa. S obzirom da kompletni saobraćajni uslovi nisu poznati, *online* TE rešenja ne mogu obezbijediti optimalnu konfiguraciju ruta.

Na kraju Sekcije 2.2 sumirani su nedostaci postojećih TE rešenja, koji ujedno predstavljaju motiv istraživanja ove disertacije. Takođe, objašnjeno je na koji način SDN može doprinijeti poboljšanju TE mogućnosti u odnosu na konvencionalne mrežne tehnologije (Sekcija 2.3).

U Sekciji 2.4 predstavljena su tri modela saobraćaja koji se koriste u definiciji optimizacionih TE problema: *pipe* model, *hose* model i HSDT (*Hose model with bounds of Source-Destination Traffic demands*) model. Za svaki od saobraćajnih modela formulisano je TE problem minimizacije maksimalne iskorišćenosti linkova (*Maximum Link Utilization* - MLU) u Sekciji 2.5. Pristupi rešavanju višekriterijumskih TE problema objašnjeni su u Sekciji 2.6.

2.1 IP TE

Rutiranje saobraćaja u IP mrežama vrši se isključivo na osnovu destinacione IP adrese paketa. Samim tim, mogućnosti saobraćajnog inženjeringa su veoma ograničene. Među-

tim, jednostavnost IP arhitekture ima i svojih prednosti, a to su:

- **Skalabilnost:** Mreža funkcioniše uz vrlo skromnu razmjenu kontrolnih informacija između uređaja, što povlači da više kapaciteta ostaje na raspolaganju za prenos korisničkog saobraćaja.
- **Brz oporavak od greške:** Nije potrebno uspostavljanje *backup* rute. Saobraćaj se automatski preusmjerava na alternativnu rutu u slučaju ispada linka.

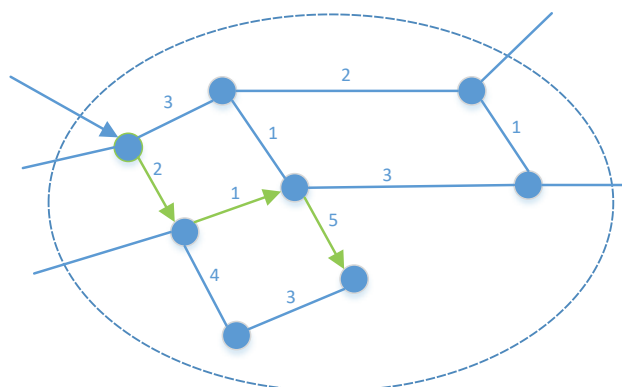
Stoga, mehanizmi za inženjering saobraćaja u IP mrežama dugo su privlačili pažnju istraživačke zajednice i nakon razvoja naprednijih mrežnih tehnologija. Većina razvijenih rešenja [69–71] bazirana je na jednostavnim šemama rutiranja kao što je SPF (*Shortest Path First*), i šemama ravnomjernog balansiranja opterećenja preko ruta sa identičnim težinskim faktorom (*Equal Cost Multipath routing* - ECMP).

2.1.1 SPF rutiranje

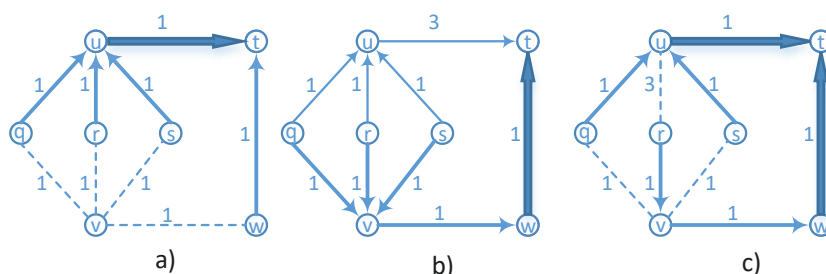
Najzastupljeniji protokoli rutiranja u IP mrežama, kao npr. OSPF (*Open Shortest Path First*) i IS-IS (*Intermediate System - Intermediate System*) računaju rute u mreži na osnovu statičkih težinskih faktora koji odražavaju karakteristike linkova. Ruteri razmjenjuju informacije o težinskim faktorima svojih linkova, i na taj način kreiraju sliku topologije unutar autonomnog sistema, kao što je prikazano na Slici 2.2. Zatim, svaki ruter računa najkraće putanje ka ostalim ruterima u mreži, pri čemu se pod *dužinom* putanje smatra suma težinskih faktora linkova duž putanje. Informacije o izračunatim putanjama čuvaju se u tabelama rutiranja, koje kontrolišu prosleđivanje IP paketa na svakom ruteru u mreži.

Osnovna ideja SPF TE rešenja je podešavanje težinskih faktora linkova u skladu sa unaprijed poznatim optimizacionim kriterijumom. Na Slici 2.3 prikazan je primjer kontrole saobraćaja podešavanjem težinskih faktora. Sva tri prikazana dijagrama odnose se na istu mrežu. Pretpostavljeno je da su linkovi istog kapaciteta, koji je izražen u apstraktnim jedinicama, a čvorovi q , r , s i w šalju jednu jedinicu saobraćaja prema čvoru t . Kao optimizacioni kriterijum usvojena je minimizacija MLU.

- **Inicijalna konfiguracija težinskih faktora:** Prvi dijagram prikazuje ishod rutiranja u slučaju kada svi linkovi imaju težinski faktor 1. Kompletan saobraćaj čvorova q , r i s usmjerava se preko čvora u , stvarajući opterećenje od 3 jedinice kapaciteta na linku (u, t) .



Slika 2.2: Rutiranje najkraćom putanjom.



Slika 2.3: Optimizacija težinskih faktora kod SPF rutiranja.

- **Lokalna promjena težinskog faktora zagušenog linka:** Naivni pristup relaksiranju opterećenja na linku (u, t) je povećavanje njegovog težinskog faktora. Na drugom dijagramu težinski faktor ovog linka je postavljen na 3. Ova konfiguracija rezultira opterećenjem od 4 jedinice na linku (w, t) .
- **Globalna optimizacija težinskih faktora:** Optimizacija svih težinskih faktora na nivou mreže rezultira stanjem prikazanim na trećem dijagramu, gdje ni jedan link nije opterećen sa više od 2 jedinice saobraćaja. Distribucija saobraćaja je optimalna sa aspekta maksimalne iskorišćenosti linkova. S obzirom da postoji ukupni saobraćajni zahtjev od 4 jedinice prema čvoru t , a taj čvor ima samo dva linka, jasno je da bolje rešenje nije moguće ostvariti nijednom drugom šemom rutiranja.

Optimalnost težinskih faktora uslovljena je tačnošću informacija o mrežnoj topologiji i zauzetosti linkova. Ove informacije prikupljaju se SMTP protokolom. Za potrebe inženjeringa saobraćaja, obično je potrebno izvršiti i estimaciju saobraćaja između svakog para mrežnih čvorišta [2].

Iako navedeni primjer pokazuje da je podešavanjem težinskih faktora moguće efika-

sno balansirati mrežno opterećenje, praktičnu implementaciju ovog pristupa prate brojni izazovi, uslijed kojih se u današnjim mrežama dominantno koriste statički težinski faktori [4]. Promjene težinskih faktora odražavaju se na rutiranje svih sabračajnih tokova, pa samim tim ugrožavaju stabilnost mreže. Kvarovi u mrežama (npr. ispadi linkova) mogu unijeti drastične promjene u distribuciji saobraćaja i izazvati zagušenje nekih njenih segmenata. Sa druge strane, podešavanjem težinskih faktora uglavnom nije moguće realizovati optimalnu konfiguraciju ruta zbog ograničenja koja nameće rutiranje najkraćom putanjom [3].

Pored pomenutog *offline* TE pristupa, u literaturi je predloženo nekoliko *online* TE algoritama za IP mreže koji definišu težinske faktore linkova kao funkciju nekog dinamičkog, promjenjivog parametra, kao što je iskorišćenost linka ili kašnjenje na linku [72]. Ova rešenja ne vrše rerutiranje postojećih tokova u mreži. Međutim, ruteri moraju periodično da razmjenjuju informacije o svojim težinskim faktorima. Prilikom tog procesa može doći do kreiranja petlji rutiranja usled dugog vremena konvergencije tabela rutiranja [73].

2.1.2 ECMP rutiranje

Kod ECMP rutiranja [74] ulazno saobraćajno opterećenje se ravnomjerno dijeli preko više ruta sa jednakim težinskim faktorom. Dijeljenje saobraćaja se realizuje na nivou saobraćajnog toka da bi se izbjegli problemi sa neredoslednim prijemom paketa. Istraživanja [3, 69, 70] su pokazala da je optimizacijom OSPF/IS-IS težinskih faktora (u cilju balansiranja opterećenja) moguće poboljšati servisni kapacitet mreže za 50-110% u odnosu na konvencionalni pristup konfiguracije težinskih faktora inverzno proporcionalno kapacitetu linka. Osnovna ideja predloženih algoritma je podešavanje težinskih faktora na izlaznim linkovima mrežnih čvorišta u cilju kreiranja većeg broja putanja sa jednakim težinskim faktorom prema destinacionom čvoru. Kao rezultat toga, saobraćaj se umjesto preko jedne putanje prenosi preko više putanja sa jednakim OSPF/IS-IS težinskim faktorom. Predloženi model optimizacije težinskih faktora je *NP-hard*, što ga čini nepromjenjivim za dinamički (*online*) inženjering saobraćaja. Samim tim, mapiranje tokova na ECMP rute može se smatrati statičkim [75], jer ne uzima u obzir trenutnu iskorišćenost linkova ili veličinu tokova. Posledično, dolazi do kolizija u mreži i degradacije cjelokupnih mrežnih performansi [10].

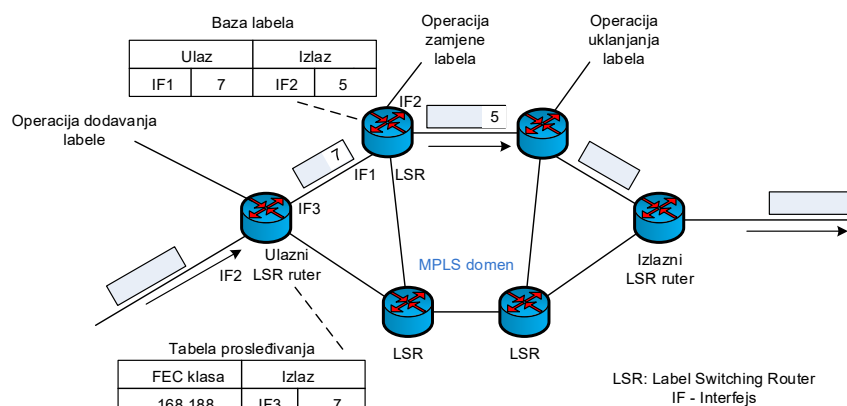
2.2 MPLS TE

Prednosti MPLS [76] TE rešenja u odnosu na IP TE rešenja dominantno proističu iz mogućnosti eksplicitnog rutiranja, koje nije obavezno bazirano na destinacionoj IP adresi. Koncept eksplicitnog rutiranja omogućava neravnomjernu distribuciju saobraćaja preko više putanja (*multipath* rutiranje). Prednosti takvog modela rutiranja su brojne, uključujući sledeće:

- **Prilagođavanje zahtjevima aplikacija** - Aplikacije imaju različite QoS zahteve. Ukoliko postoji više ruta između izvorišnog i destinacionog čvorišta, vremenski-kritične aplikacije (npr. taktilni IoT, VoIP) mogu koristiti rute sa malim kašnjenjem, dok se prenos fajlova može vršiti rutama velike propusnosti [53].
- **Veća pouzdanost prenosa** – U slučaju ispada rutera ili linka na nekoj od ruta saobraćaj se može brzo preusmeriti na alternativne rute. Robustnost na otkaze u mreži je posebno izražena ukoliko su rute nepreklapajuće.
- **Izbjegavanje zagušenih ruta** – Saobraćajno opterećenje može se balansirati preko više ruta u cilju izbegavanja zagušenja linkova. Istraživanja su pokazala da je moguće značajno poboljšati iskorišćenost mrežnih resursa korišćenjem samo dvije rute između svakog para čvorišta [77].

Multipath rutiranje ima posebno veliki potencijal u ISP mrežama, za koje je tipičan veliki diverzitet ruta. Čak 90% ISP topologija analiziranih u [78] imaju najmanje 4 nepreklapajuće rute između svakog PoP (*Point of Presence*) para. Iako se Internet saobraćaj danas dominantno prenosi rutama sa najmanjim težinskim faktorom, 30% - 80% vremena postoje alternativne rute u mreži koje nude manje kašnjenje ili gubitke [79].

U MPLS mrežama podrška za *multipath* rutiranje ostvarena je uvođenjem novog modela prosleđivanja paketa na osnovu labela. Labele predstavljaju kratke identifikacione oznake fiksne dužine (32 bita) koje se dodjeljuju paketima prilikom ulaska u mrežu. One identifikuju putanju (MPLS tunel [80]) kojom će se paket prosleđivati kroz mrežu, kao i odgovarajuće servisne attribute. Informacije iz zaglavlja paketa analiziraju se samo jednom - na ulazu u mrežu (Slika 2.4), gdje se donosi odluka o rutiranju, a dalje se postupak usmjeravanja paketa zasniva samo na labelama. Labele se konačno uklanjaju iz paketa od strane ivičnog rutera na izlazu iz MPLS domena. Iz primjera prikazanog na Slici 2.4 treba primijetiti da labele imaju lokalni značaj. Logika za dodjeljivanje labela bazirana



Slika 2.4: Princip rutiranja na osnovu MPLS labela.

je na konceptu FEC (*Forwarding Equivalence Class*) klasa. RFC 3031 standard definiše FEC klasu kao skup paketa čija odredišna IP adresa ima isti mrežni prefiks, ali je moguće po potrebi koristiti i druge definicije koje uključuju polja transportnih i IP zaglavlja. FEC klasa se vezuje za MPLS labelu, pa svi paketi iste FEC klase koriste identičan tunnel i opslužuju se na identičan način, pod uslovom da ulaze u MPLS domen na istom mjestu. Određivanje pripadnosti paketa FEC klasi se vrši samo na ulazu u MPLS domen, dok se u ostalim MPLS ruterima vrši samo procesiranje labela.

MPLS tuneli predstavljaju logičke linkove između dva rutera ili hosta. Kreiraju ih MPLS ruteri na ivici mreže na osnovu informacija o mrežnoj topologiji, matrici saobraćaja i dostupnosti resursa (npr. dostupni kapacitet na linkovima). Ove informacije prikupljaju se uz pomoć protokola rutiranja, kao što su OSPF i IS-IS, dok se uspostavljanje tunela vrši signalizacijom koju definiše RSVP (*Resource Reservation Protocol*) protokol [81]. Trenutna praksa u MPLS ISP mrežama je ravnomjerno balansiranje saobraćaja preko dostupnih tunela, ukoliko nisu definisani QoS zahtjevi [1]. Za rutiranje tokova koji imaju QoS zahtjeve u pogledu propusnosti, odgovarajući tunnel/ruta se bira izvršavanjem CSPF (*Constrained Shortest Path First*) algoritma. CSPF algoritam predstavlja ekstenziju klasičnog SPF algoritma, koja računa putanju sa najmanjim težinskim faktorom razmatrajući pri tome QoS ograničenja kao što su propusnost, kašnjenje, itd.

U nastavku poglavlja dat je pregled *state-of-the-art* MPLS TE tehnika. S obzirom da se ove tehnike mogu primijeniti uz izvjesne izmjene i u SDN mrežama, identifikovani su njihovi nedostaci koji su motivisali istraživanja u sklopu disertacije.

2.2.1 Offline MPLS TE

Problem optimizacije ruta u MPLS mrežama može se formulirati kao MCF (*Multi-Commodity Flow*) problem [82]. Međutim, ovaj pristup je nepraktičan za implementaciju zbog velike računske kompleksnosti i ogromnog broja tunela koji su potrebni za realizaciju optimalne šeme rutiranja [3].

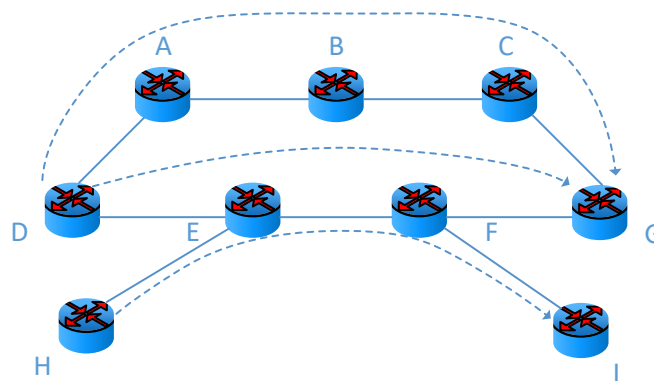
Imajući u vidu probleme u pogledu skalabilnosti MCF pristupa, predloženo je više rešenja koja su bazirana na jednostavnom CSPF rutiranju [83] i koja ne zahtijevaju koordinaciju između individualnih sabračajnih tokova [84]. CSPF rutiranje se obično vrši na nivou agregiranih tokova (tzv. makro tokova) koji predstavljaju skup više individualnih tokova sa sličnim QoS zahtjevima i istim izvorišnim i destinacionim mrežnim čvorištem. Princip rada CSPF algoritama je sledeći. Prije uspostavljanja tunela za određeni agregirani tok, svi linkovi nedovoljnog kapaciteta se uklanjaju iz mrežne topologije. Na tako formiranom mrežnom grafu izvršava se SPF algoritam rutiranja u cilju pronalaženja najkraće validne rute. Identična procedura se zatim ponavlja za sve ostale agregirane tokove. WSP (*Widest Shortest Path*) i SWP (*Shortest Widest Path*) tehnike [85] su predložene kao alternative klasičnom CSPF rutiranju koje teže da povećaju dostupni kapacitet *bottleneck* linkova na odabranim rutama. WSP algoritam iz seta najkraćih ruta koje mogu da ispune QoS zahtjev bira „najširu”, tj. onu sa najviše slobodnog kapaciteta. Na ovaj način pokušava se napraviti kompromis između dva oprečna zahtjeva: balansiranja opterećenja i potrošnje resursa. Komplementarna vrsta kompromisa napravljena je sa SWP algoritmom, koji od svih mogućih ruta bira najširu. Ukoliko postoji više takvih ruta, optimalnom se smatra najkraća od njih. Problem pronalažanja najšire rute konceptualno je sličan problemu rutiranja najkraćom putanjom. Jedina razlika je u tome što se umjesto rute sa najmanjim brojem hopova bira ona najvećeg kapaciteta [25].

Veliki broj offline TE algoritama iz literature baziran je na ideji minimizacije MLU [86]. Ovakva definicija optimizacione funkcije omogućava distribuciju saobraćajnog opterećenja zagušenih dijelova mreže preko manje iskorišćenih mrežnih segmenata. Na taj način povećava se servisni kapacitet mreže, odnosno vjerovatnoća prihvatanja novih saobraćajnih tokova. Samim tim, mreža je u stanju da podrži potencijalni budući porast saobraćaja u toku dužeg perioda bez potrebe za rekonfiguracijom postojećih ruta. Ovo naročito važi u slučajevima kada saobraćaj raste proporcionalno trenutnom obrascu.

2.2.2 Online MPLS TE

Online MPLS TE rešenja mogu se klasifikovati u dvije kategorije. U prvu kategoriju spadaju rešenja za dinamičko podešavanje parametara za balansiranje opterećenja preko više unaprijed instaliranih statičkih tunela [77, 87], dok drugu kategoriju čine rešenja koja dinamički instaliraju tunele za saobraćajne tokove po njihovom ulasku u mrežu. Tipičan primjer prve kategorije je MATE [87], koji kontinualno vrši monitoring i analizu zagušenosti tunela, i adaptivno podešava parametre za balansiranje saobraćaja na ulaznim MPLS ruterima, sa ciljem minimizacije ukupnog mrežnog opterećenja (tzv. troškova rutiranja). Druga kategorija *online* MPLS TE rešenja često je bazirana na primjeni CSPF, WSP i SWP algoritama. Takođe, razvijeno je više rešenja koja razmatraju problem “interferencije” između MPLS tunela [20, 21]. Naime, primijećeno je da uspostavljanje ruta samo na osnovu informacija o mrežnoj topologiji i slobodnom kapacitetu linkova, a bez razmatranja pozicija svih izvorišno-odredišnih parova mrežnih čvorova, često dovodi do zagušenja “kritičnih” linkova koje koristi veliki broj tunela. Uticaj alokacije resursa za novi tok na dostupni kapacitet ostalih tunela u mreži definisan je kao “interferencija”. Na Slici 2.5 ilustrovan je primjer kreiranja “interferencije” prilikom izbora ruta. Pretpostavimo da u mrežu sa slike prvo dolazi saobraćajni tok kojeg treba prenijeti od čvora D do čvora G . Ukoliko se za ovaj tok odabere najkraća putanja ($D - > E - > F - > G$), budući saobraćajni zahtjevi između H i I biće blokirani ukoliko “kritični” link (E, F) nema dovoljno kapaciteta da podrži ukupno opterećenje. Iz ovog primjera može se primijetiti da je pozicija kritičnih linkova određena pozicijom individualnih parova MPLS rutera koji predstavljaju tačke ulaska/izlaska saobraćaja. Stoga, ukoliko su pozicije ulaznih/izlaznih MPLS rutera poznate, moguće je efikasno umanjiti “interferenciju” izbjegavanjem kritičnih linkova prilikom rutiranja. MIRA (*Minimum Interference Routing Algoritam*) [20] algoritam razvijen je sa ciljem minimizacije opterećenja “kritičnih” linkova. Pod kritičnim linkovima MIRA smatra linkove čije smanjenje kapaciteta, u bilo kojoj mjeri, direktno izaziva smanjenje ukupnog dostupnog kapaciteta za jedan ili više izvorišno-odredišnih parova u mreži. Pri tome, kritičnost linka raste sa brojem parova na čiji dostupni kapacitet utiče. Računanje ruta svodi se na računanje putanje sa najmanjim težinskim faktorom u grafu gde svaki link ima težinski faktor proporcionalan nivou svoje kritičnosti.

DORA algoritam [21] takođe je baziran na ideji minimizacije interferencije između tunela. Predložen je u cilju prevazilaženja dva osnovna nedostataka MIRA algoritma: računске kompleksnosti i nedovoljno balansiranog korišćenja mrežnih resursa. DORA sprovodi inženjering saobraćaja u dvije faze: *online* i *offline*. *Offline* faza izvršava se sa-



Slika 2.5: Primjer kreiranja „interferencije” prilikom izbora ruta.

mo prilikom inicijalizacije mreže i u slučaju promjene u topologiji. Linkovima se tom prilikom dodjeljuju skalarne vrijednosti (*Path Potential Value* – PPV) koje indiciraju učestanost pojavljivanja linka u tunelima izvorišno-odredišnih parova. *Online* faza algoritma izvršava se u trenutku pristizanja saobraćajnog toka, i na osnovu informacija o trenutnoj zauzetosti resursa u mreži bira jedan od tunela kojim će se tok prenositi. Računanje optimalne rute vrši se jednostavnim SPF algoritmom na mrežnom grafu čiji su težinski faktori skalirana funkcija PPV vrijednosti i slobodnog kapaciteta na linku.

2.2.3 Motiv za razvoj novih TE rešenja

Za zadatu matricu saobraćaja optimalnu konfiguraciju mreže moguće je postići samo primjenom *offline* TE rešenja koja podrazumijevaju izvršavanje kompleksnih optimizacionih modela. Kada su ISP mreže u pitanju, optimizacioni modeli sadrže ogroman broj varijabli i ograničenja, pa se njihovo rešavanje vrši uglavnom periodično, u dužim vremenskim intervalima. Iako je u nekim situacijama moguće izvršiti estimaciju budućih saobraćajnih zahtjeva, dinamika saobraćaja je uglavnom nepredvidljiva, pa efikasnost *offline* TE konfiguracije rapidno slabi sa vremenom. Kod rešenja baziranih na CSPF algoritmu napravljen je kompromis između optimalnosti šeme rutiranja i računске kompleksnosti. Međutim, rezultujuća konfiguracija može zahtijevati preusmjeravanje svih tokova u mreži, što povlači generisanje velike količine kontrolnog saobraćaja i degradaciju kvaliteta servisa. Da bi izbjegli ove nedostatke, *online* TE algoritmi dinamički određuju rute za nove saobraćajne tokove na osnovu trenutnih mjerenja različitih parametara mrežnih performansi, ne mijenjajući rute već uspostavljenih tokova. S obzirom da ova rešenja ne razmatraju ukupno saobraćajno opterećenje mreže, već samo pojedinačne zahtjeve, ne

uspjevaju da obezbijede optimalnost rutiranja na globalnom nivou.

Jedna od hipoteza na kojoj su bazirana istraživanja sprovedena u okviru disertacije je da se adekvatnim kombinovanjem *offline* TE mehanizama i *online* rutiranja može značajno poboljšati servisni kapacitet mreže u odnosu na slučaj kada se primjenjuje samo *offline* ili samo *online* TE pristup. Pri tome, poželjna strategija je da *offline* TE algoritmi periodično optimizuju mrežnu konfiguraciju na kontrolisan način, i da pružaju potrebne smjernice *online* algoritmima rutiranja kako odabrane rute ne bi stvarale “interferenciju” budućim saobraćajnim zahtjevima. Takođe, za razliku od *state-of-the-art* rešenja iz literature koja su dominantno dizajnirana za *best-effort* servis, ili eventualno podrazumijevaju garanciju propusnosti, TE rešenja predstavljena u ovoj disertaciji omogućavaju fleksibilnu diferencijaciju saobraćaja na osnovu QoS zahtjeva u pogledu propusnosti, kašnjenja i pouzdanosti. Predloženi TE pristupi inspirisani su principom balansiranja saobraćaja u MPLS mrežama sa višestrukim tunelima između izvorišno-odredišnih čvorova. Međutim, TE problem je riješen u kontekstu softverski definisanog umrežavanja - tehnologije u usponu koja prevazilazi brojne nedostatke MPLS-a, usled kojih je inženjering saobraćaja u današnjim ISP mrežama i dalje usvojen samo u primitivnom obliku [88].

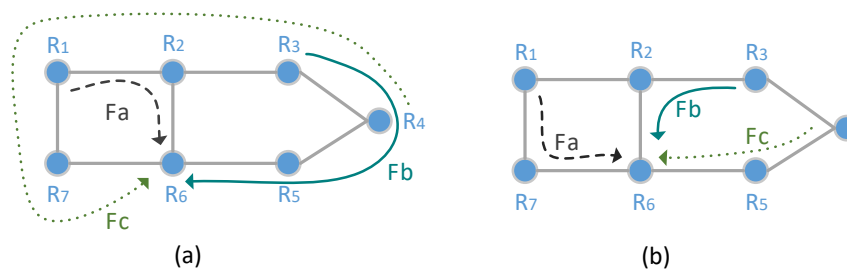
2.3 Ograničenja savremenih TE tehnologija

S obzirom da su tema ove disertacije SDN TE rešenja, u cilju obrazloženja sprovedenih istraživanja važno je identifikovati nedostatke drugih TE tehnologija koje su trenutno zastupljene u ISP mrežama. Preciznije, u ovoj sekciji analizirana su ograničenja MPLS tehnologije, na kojoj su dominantno bazirane današnje ISP mreže, kao i ograničenja PCE (*Path Computation Element*) arhitekture koja se smatra najnovijim TE pristupom u mrežama ovog tipa [89]. Takođe, objašnjeno je i kako SDN doprinosi prevazilaženju identifikovanih nedostataka MPLS/PCE mrežne arhitekture.

- **Neprecizno dijeljenje saobraćaja:** Strategije za izbjegavanje zagušenja u mreži često podrazumijevaju *multipath* rutiranje. Međutim, aktuelni mehanizmi za dijeljenje saobraćaja preko više putanja imaju nekoliko ograničenja. Sa jedne strane, dijeljenje saobraćaja na nivou paketa rezultuje neredoslednim prijemom paketa, što je posebno nepoželjno za TCP (*Transmission Control Protocol*) aplikacije. Neredosledan prijem paketa TCP toka trigeruje nepotrebno aktiviranje TCP mehanizama za kontrolu zagušenja, što rezultuje smanjenom brzinom prenosa i degradacijom

cjelokupnih mrežnih performansi [90]. Takođe, usled varijacije kašnjenja (*jitter*) prilikom prenosa, na prijemnoj strani su potrebni veliki baferi za privremeno skladištenje dolaznog saobraćaja. Problemi neredoslednog prijema paketa mogu se izbjeći dijeljenjem saobraćaja na nivou tokova. Međutim, granularnost dijeljenja saobraćaja uslovljena je mogućnostima uređaja za prosleđivanje i *hash* funkcijom koja se koristi prilikom mapiranja tokova na rute. Realizovana granularnost dijeljenja često se ne podudara sa onom koju zahtijeva TE rešenje, što dovodi do distribucije saobraćaja u neodgovarajućem odnosu na svakoj od ruta. Stoga, mrežne performanse mogu prilično odstupati od optimalnih. Takođe, mehanizmi za balansiranje saobraćaja preko više putanja, kao što je ECMP, ne uzimaju u obzir moguće zagušenje na rutama sa najmanjim težinskim faktorom (preko kojih se distribuiraju tokovi), što dodatno doprinosi degradaciji kvaliteta servisa [91]. SDN omogućava prevazilaženje navedenih problema zahvaljujući većoj granularnosti dijeljenja saobraćaja koju nude SDN svičevi.

- **Neoptimalna alokacija resursa:** MPLS TE tehnike alokacije resursa takođe imaju brojne nedostatke. U radu [92] identifikovan je problem velike varijacije kašnjenja na linkovima u MPLS mreži. Analizirana su zastupljena MPLS-TE rešenja i utvrđeno je da su fluktuacije kašnjenja posledica MPLS *autobandwidth* mehanizma [76], koji kontinualno razmatra mogućnosti za poboljšanje postojećih ruta izvršavanjem CSPF algoritma. Ovaj mehanizam nude mnogi proizvođači MPLS mrežne opreme kao rešenje za automatsko podešavanje rezervisanog propusnog opega zavisno od saobraćajnog opterećenja. Distribuirani model rutiranja u klasičnim MPLS mrežama je generalno veoma neefikasan. Na Slici 2.6 prikazan je primjer mreže u kojoj svaki link može podržati samo jedan tok [13]. Ukoliko tokovi dolaze po redosledu F_A , F_B , i F_C , resursi će biti alocirani kao što je prikazano na Slici 2.6a. Tok F_A je dodijeljen jednoj od najkraćih ruta. U trenutku dolaska toka F_B , instalira se najkraća ruta sa najviše dostupnog kapaciteta (CSPF). Na isti način tretira se i tok F_C . Ovakva distribucija saobraćaja nije optimalna ni sa aspekta količine zauzetih resursa ni sa aspekta dužine alociranih ruta. SDN prevazilazi ove probleme izvršavanjem algoritama rutiranja na logički centralizovanom kontroleru, koji ima globalni uvid u stanje mreže. Na Slici 2.6b prikazana je optimalna šema rutiranja koja zahtijeva ne-lokalne promjene ruta, npr. pomjeranje toka F_A u trenutku dolaska F_B . Iako je konfiguraciju sa Slike 2.6b moguće realizovati i u MPLS mrežama, npr. dijeljenjem toka F_A preko gornje i donje rute, a zatim primjenom *preemption* strategije [1] koja



Slika 2.6: Optimizacija ruta u MPLS i SDN mrežama.

preferira kraće rute za tokove F_B i F_C , to i dalje ne eliminiše fundamentalni problem lokalnog donošenja odluka (na nivou rutera) o alokaciji resursa [93].

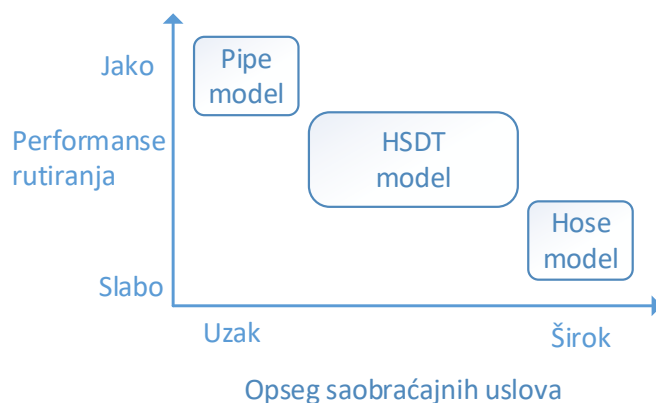
- **TE baze podataka ne odražavaju trenutno stanje u mreži:** PCE mrežna arhitektura u odnosu na klasičnu MPLS arhitekturu dodaje namjenski element (PCE server) koji je zadužen za izračunavanje ruta, što omogućava primjenu složenijih i sofisticiranijih TE algoritama. Iako PCE arhitektura na ovaj način unapređuje klasični MPLS TE u mnogim aspektima (npr. efikasnost alokacije resursa i rutiranja), takođe predstavlja nova ograničenja. Kod PCE arhitekture računanje ruta se obavlja na osnovu TE informacija koje su skladištene u TED (*TE database*) bazi. TED baza sadrži popis resursa dostupnih u mreži i ostale informacije koje PCE kontroler koristi prilikom računanja ruta. Međutim, prema RFC 4655 standardu [94], TED baza ne odražava uvijek stanje u mreži u realnom vremenu. Kada TED baza nije pravilno sinhronizovana sa stanjem mreže, dolazi do instalacije neadekvatnih ruta. Sa druge strane, ovaj problem nije izražen u SDN mrežama uslijed centralizovane kontrolne ravni.
- **Veliko vrijeme konvergencije distribuiranih protokola:** Još jedno važno ograničenje MPLS/PCE arhitekture je njena zavisnost od RSVP-TE protokola. Kada mrežni uređaj zahtijeva uspostavljanje rute, PCE kontroler odgovara sa informacijom o izračunatoj ruti, a uređaj dalje tu informaciju prenosi kroz mrežu koristeći distribuirani RSVP-TE protokol za signaliziranje. Uspostavljanje rute može trajati značajno duže nego u slučaju kada to vrši centralizovana kontrolna ravan sa zasebnim kanalom za kontrolnu komunikaciju (*out-of-bound*). Ovo se direktno odražava na mrežnu stabilnost (RSVP-TE je *inbound* protokol) i performanse postojećih saobraćajnih tokova. SDN pomaže u prevazilaženju ovih problema pružajući programabilnu ravan podataka koja se može kontrolisati *out-of-band* komunikacijom.

Sve u svemu, TE rešenja su značajno evoluirala prethodnih godina. Od ranih MPLS TE rešenja do sofisticiranijih rešenja prilagođenih PCE arhitekturi bilježi se stalan porast u poboljšanju iskorišćenosti mrežnih resursa. Međutim, MPLS TE rešenja prati i veliki broj problema koji su identifikovani u ovom poglavlju. Usled toga, istražuju se novi pristupi, a rešenja bazirana na SDN-u se nameću kao prirodni put koji treba slijediti.

2.4 Modeli saobraćaja

Saobraćajno opterećenje mreže sa N čvorišta uobičajeno se modeluje matricom saobraćaja $T\{d_{uv}\}$, dimenzija $N \times N$, čiji elementi d_{uv} predstavljaju saobraćaj između izvorišnog čvora u i destinacionog čvora v . Ukoliko su svi elementi matrice T poznati, riječ je o *pipe* modelu saobraćaja. Poznavanje matrice saobraćaja preduslov je za računanje optimalne konfiguracije ruta. Međutim, uslijed rapidnog razvoja novih Internet aplikacija, kao što su proširena realnost, VoIP i *peer-to-peer*, saobraćaj u ISP mrežama je postao veoma varijabilan i komplikovan za estimiranje. Sa tog aspekta, poželjno je da se TE algoritam izvršava što češće, kako bi se kompenzovala greške estimacije. Estimacija matrice saobraćaja može se vršiti na osnovu periodičnih mjerenja saobraćajnih zahtjeva d_{uv} , ili na osnovu mjerenja opterećenja svakog linka u mreži [95]. U oba slučaja, proces prikupljanja saobraćajnih statistika traje izvjesno vrijeme, u toku kojeg se matrica saobraćaja može drastično promijeniti. Čak i kada je moguće prikupiti ove informacije relativno brzo, u mrežama većih razmjera rešavanje optimizacionih TE problema rutiranja traje dosta vremena, pa se adaptiranje šeme rutiranja ne može vršiti u realnom vremenu.

Dinamičke adaptacije tabela rutiranja generalno komplikuju rad mreže [96]. U cilju izbjegavanja ovih komplikacija, ISP ponekad preferira protokole rutiranja koji se ne prilagođavaju dinamički promjenama saobraćaja, već koriste statičke, *a priori* izračunate tabele rutiranja prilikom prosleđivanja paketa. To je dovelo do interesovanja za *hose* model saobraćaja [97], gdje je pretpostavka da je poznato maksimalno ulazno i izlazno opterećenje svakog mrežnog čvorišta, ali da informacije o trenutnoj matrici saobraćaja nisu dostupne. Dok estimaciju matrice saobraćaja prate brojni problemi, mrežni operatori relativno jednostavno određuju *hose* model saobraćaja. Ukupni izlazni saobraćaj čvora u može se predstaviti sa $\sum_v d_{uv} \leq \alpha_u$, gdje je α_u maksimalna količina saobraćaja koju čvor u može poslati u mrežu. Ukupni izlazni saobraćaj čvora v modeluje se sa $\sum_u d_{uv} \leq \beta_v$, gdje je β_v maksimalna količina saobraćaja koju čvor v može da primi iz mreže. *Hose* model saobraćaja kao ograničenja uzima samo α_u i β_v . TE algoritmi koji vrše optimizaciju ruta



Slika 2.7: Poređenje različitih modela saobraćaja.

na osnovu *hose* modela saobraćaja u literaturi su poznati kao *oblivious* algoritmi [96,97]. Prednost primjene ovih rešenja ogleda se u robustnosti šeme rutiranja na iznenadne fluktuacije saobraćaja.

Rutiranje bazirano na *hose* modelu ne može biti efikasno kao rutiranje na osnovu *pipe* modela, s obzirom da u prvom slučaju informacije o matrici saobraćaja nisu dostupne. Stoga, poželjno je da operatori suze opseg saobraćajnih uslova obuhvaćenih *hose* modelom na osnovu prethodnih iskustava i saobraćajnih statistika. Prvo, moguće je estimirati odstupanje izmjerene matrice saobraćaja od realne u datom trenutku. Na primjer, utvrđeno je da relativna greška estimacije matrice saobraćaja na osnovu zauzetosti linkova može iznositi 20% i više [98]. Drugo, mogu se predvidjeti fluktuacije saobraćaja iz istorije rada mreže. Treće, mrežni operatori znaju trendove saobraćajnih zahtjeva. Na primer, saobraćajni zahtjevi velikih gradova su mnogo veći od onih u malim gradovima. Stoga, moguće je definisati dodatna ograničenja sa $\delta_{uv} \leq d_{uv} \leq y_{uv}$ za svaki par izvorišnog čvora u i odredišta v , gdje δ_{uv} i y_{uv} predstavljaju gornju i donju granicu za d_{uv} , respektivno. *Hose* model dopunjen sa ovim ograničenjima poznat je kao HSDT model [99]. Karakteristike *pipe*, *hose* i HSDT modela su šematski prikazane na Slici 2.7 [100].

2.5 Formulacija optimizacionih TE algoritama za različite modele saobraćaja

Šema rutiranja ima veliki uticaj na efikasnost korišćenja mrežnih resursa i ukupnu mrežnu propusnost. Izborom adekvatnih ruta suzbija se zagušenje mreže i povećava ro-

bustnost na fluktuacije saobraćaja koje su teško predvidljive. Jedan od korisnih pristupa poboljšanju performansi rutiranja je minimizacija MLU metrike, odnosno stope zagušenosti mreže. Ovaj pristup primijenjen je u nastavku sekcije, gdje je razmatran problem optimalnog rutiranja za tri saobraćajna modela: *pipe*, *hose* i HSDT.

2.5.1 Model mreže

U definicijama optimizacionih problema rutiranja biće korišćena sledeća terminologija. Mreža je modelovana kao usmjereni graf $G(V, L)$, gdje V predstavlja skup čvorova, a L skup svih linkova. Sa $N \subseteq V$ označen je set ivičnih čvorova koji generišu saobraćaj u mrežu. Za link između čvorova $i \in V$ i $j \in V$ usvojena je notacija $(i, j) \in L$. Svakom linku $(i, j) \in L$ dodijeljen je kapacitet c_{ij} , dok je saobraćajno opterećenje linka obilježeno sa y_{ij} . Saobraćajni zahtjev između čvorova u i v je označen sa d_{uv} . Dio zahtjeva d_{uv} koji se rutira linkom $(i, j) \in L$ označen je sa x_{ij}^{uv} . Ukoliko je $x_{ij}^{uv} > 0$, onda se link (i, j) nalazi na jednoj od ruta koju koristi d_{uv} , pri čemu je $x_{ij}^{uv} < 1$ indikator *multipath* rutiranja za d_{uv} . Maksimalna iskorišćenost linkova definisana je kao:

$$r = \max_{(i,j) \in L} \frac{y_{ij}}{c_{ij}} \quad (2.1)$$

gdje je $0 \leq r \leq 1$. Minimizacijom varijable r maksimizuje se servisni kapacitet mreže. Ovo proističe iz činjenice da je na svakom od linkova $(i, j) \in L$ slobodno $(1 - r) \cdot c_{ij}$ kapaciteta.

2.5.2 Formulacija MLU minimizacije za *pipe* model

Formulacija problema optimalnog rutiranja na osnovu pipe modela saobraćaja predložena je u [86]. Optimalna MLU vrijednost utvrđuje se rešavanjem sledećeg LP (*Linear Programming*) problema:

$$\text{minimize } r \quad (2.2a)$$

$$\text{subject to } \sum_{j:(i,j) \in L} x_{ij}^{uv} - \sum_{j:(j,i) \in L} x_{ji}^{uv} = 1 \quad \forall u, v \in N, i = u \quad (2.2b)$$

$$\sum_{j:(i,j) \in L} x_{ij}^{uv} - \sum_{j:(j,i) \in L} x_{ji}^{uv} = 0 \quad \forall u, v \in N, i (\neq u, v) \quad (2.2c)$$

$$\sum_{u,v \in N} x_{ij}^{uv} \cdot d_{uv} \leq c_{ij} \cdot r \quad \forall (i, j) \in L \quad (2.2d)$$

$$0 \leq x_{ij}^{uv} \leq 1 \quad \forall u, v \in N, (i, j) \in L \quad (2.2e)$$

$$0 \leq r \leq 1 \quad (2.2f)$$

Ulazni parametri optimizacionog modela su matrica saobraćaja $T = \{d_{uv}\}$, a izlazne varijable r i x_{ij}^{uv} . Optimizaciona funkcija (2.2a) minimizuje MLU. Jednačine (2.2b) i (2.2c) definišu ograničenja konzervacije toka. Prema (2.2c) količina saobraćaja koja ulazi u čvor mora da bude jednaka količini saobraćaja koja izlazi iz čvora, osim u slučaju izvorišnog i odredišnog čvora. Za izvorišni čvor $i (= u)$ toka d_{uv} važi da svih d_{uv} jedinica toka izlaze iz čvora, što je formulisano sa (2.2b). Ukoliko su zadovoljena ograničenja (2.2b) i (2.2c), konzervacija toka je zadovoljena i na odredišnom čvoru. Jednačina (2.2d) ukazuje da je zbir frakcija saobraćajnih zahtjeva prenijetih preko linka $(i, j) \in L$ manji ili jednak od proizvoda MLU vrijednosti i kapaciteta linka.

2.5.3 Formulacija MLU minimizacije za *hose* model

Problem optimalnog rutiranja sa *hose* modelom saobraćaja može se takođe formulirati jednačinama (2.2a)-(2.2f), gdje su r i x_{ij}^{uv} optimizacione varijable a d_{uv} i c_{ij} poznati parametri. Međutim, zahtjevi između izvorišno-odredišnih čvorova (d_{uv}) su ograničeni na sledeći način:

$$\sum_{u \in N} d_{uv} \leq \alpha_u, \quad u \in N \quad (2.3)$$

$$\sum_{v \in N} d_{uv} \leq \beta_v, \quad v \in N \quad (2.4)$$

Jednačine (2.3) i (2.4) definišu skup svih matrica saobraćaja $\{T\}$ koje zadovoljavaju ograničenja *hose* modela. Iz perspektive mrežnog operatora, u cilju stabilnog rada mreže, šema rutiranja ne bi trebalo da se rekonfiguriše ukoliko matrica T varira u opsegu $\{T\}$. Zadatak optimizacionog algoritma je pronalaženje šeme rutiranja koja minimizuje MLU (r) za sve moguće matrice saobraćaja is skupa $\{T\}$. Iako su jednačine (2.2a)-(2.2f) izražene u formi LP problema, u slučaju *hose* modela do rešenja se ne dolazi jednostavno, jer je potrebno riješiti optimizacioni model za svaku validnu kombinaciju $T = \{d_{uv}\}$ koja

zadovoljava (2.3) i (2.4). S obzirom da d_{uv} uzima realne vrijednosti, broj takvih kombinacija je beskonačno veliki. Međutim, primjenom teoreme dualnosti, ovaj problem se može transformisati u standardni LP problem relativno male računske kompleksnosti [99]:

$$\text{minimize } r \quad (2.5a)$$

$$\text{subject to } \sum_{j:(i,j) \in L} x_{ij}^{uv} - \sum_{j:(j,i) \in L} x_{ji}^{uv} = 1 \quad \forall u, v \in N, i = u \quad (2.5b)$$

$$\sum_{j:(i,j) \in L} x_{ij}^{uv} - \sum_{j:(j,i) \in L} x_{ji}^{uv} = 0 \quad \forall u, v \in N, i(\neq u, v) \in V \quad (2.5c)$$

$$\sum_{u \in N} \alpha_u \pi_{ij}(u) + \sum_{u \in N} \beta_u \lambda_{ij}(u) \leq c_{ij} \cdot r \quad \forall (i, j) \in L \quad (2.5d)$$

$$x_{ij}^{uv} \leq \pi_{ij}(u) + \lambda_{ij}(v) \quad \forall u, v \in N, (i, j) \in L \quad (2.5e)$$

$$\pi_{ij}(u), \lambda_{ij}(u) \geq 0 \quad \forall u \in N, (i, j) \in L \quad (2.5f)$$

$$0 \leq x_{ij}^{uv} \leq 1 \quad \forall u, v \in N, (i, j) \in L \quad (2.5g)$$

$$0 \leq r \leq 1 \quad (2.5h)$$

U odnosu na originalnu definiciju problema, ograničenja (2.2d), (2.3) i (2.4) su zamijenjena sa (2.5d)-(2.5f). Uvođenjem novih varijabli $\pi_{ij}(u)$ i $\lambda_{ij}(u)$ ograničenja *hose* modela su uzeta u obzir.

2.5.4 Formulacija MLU minimizacije za HSDT model

Kod HSDT modela zahtjevi između izvorišno-odredišnih parova formulisani su na sledeći način:

$$\sum_{u \in N} d_{uv} \leq \alpha_u, \quad u \in N \quad (2.6)$$

$$\sum_{v \in N} d_{uv} \leq \beta_v, \quad v \in N \quad (2.7)$$

$$\delta_{uv} \leq d_{uv} \leq \gamma_{uv}, \quad u, v \in N \quad (2.8)$$

Pronalaženje optimalne šeme rutiranja koja minimizuje MLU za najgori slučaj iz skupa saobraćajnih matrica $T\{d_{uv}\}$ koje zadovoljavaju (2.6)-(2.8) zahtijeva sličnu proceduru

kao kod *hose* modela. Računska kompleksnost problema može se značajno smanjiti eliminacijom d_{uv} varijabli, kao što je predloženo u [99]:

$$\min \quad r \quad (2.9a)$$

$$\text{s.t.} \quad \sum_{j:(i,j) \in L} x_{ij}^{uv} - \sum_{j:(j,i) \in L} x_{ji}^{uv} = 1 \quad \forall u, v \in N, i = u \quad (2.9b)$$

$$\sum_{j:(i,j) \in L} x_{ij}^{uv} - \sum_{j:(j,i) \in L} x_{ji}^{uv} = 0 \quad \forall u, v \in N, i (\neq u, v) \in V \quad (2.9c)$$

$$\sum_{u \in N} \alpha_u \pi_{ij}(u) + \sum_{u \in N} \beta_u \lambda_{ij}(u) + \sum_{u,v \in N} [\gamma_{uv} \eta_{ij}(u, v) - \delta_{uv} \theta_{ij}(u, v)] \leq c_{ij} \cdot r \quad \forall (i, j) \in L \quad (2.9d)$$

$$x_{ij}^{uv} \leq \pi_{ij}(u) + \lambda_{ij}(v) + \eta_{ij}(u, v) - \theta_{ij}(u, v) \quad \forall u, v \in N, (i, j) \in L \quad (2.9e)$$

$$\pi_{ij}(u), \lambda_{ij}(u), \eta_{ij}(u, v), \theta_{ij}(u, v) \geq 0 \quad \forall u, v \in N, (i, j) \in L \quad (2.9f)$$

$$0 \leq x_{ij}^{uv} \leq 1 \quad \forall u, v \in N, (i, j) \in L \quad (2.9g)$$

$$0 \leq r \leq 1 \quad (2.9h)$$

Izlazne varijable optimizacionog modela su $r, x_{ij}^{uv}, \pi_{ij}(u), \lambda_{ij}(u), \eta_{ij}(u, v)$ i $\theta_{ij}(u, v)$, dok su $c_{ij}, \alpha_u, \beta_u, \delta_{uv}$ i γ_{uv} zadati parametri. Uvođenjem novih varijabli $\pi_{ij}(u), \lambda_{ij}(u), \eta_{ij}(u, v)$ i $\theta_{ij}(u, v)$ optimizacija rutiranja sa HSOT modelom svodi se na klasičan LP problem koji može riješiti standardni LP solver.

2.6 Višekriterijumska TE optimizacija

Performanse TE algoritama mogu se ocjenjivati po više kriterijuma. Kao što će se vidjeti u narednim poglavljima, optimizacija po jednom kriterijumu može dovesti do drastične degradacije performansi po drugim kriterijuma relevantnim za mrežnog operatora. Rešenja koja simultano optimizuju performanse po više kriterijuma veoma često ne postoje. Na primjer, TE algoritam koji minimizuje MLU može biti modifikovan tako da smanjuje ukupnu količinu zauzetih resursa u mreži, ali u tom slučaju rezultujuća MLU vrijednost će biti veća od optimalne. Mrežni operatori obično teže da nađu najbolji balans između konfliktnih optimizacionih ciljeva. Stoga, neophodno je definisati novi koncept optimalnosti za višekriterijumske optimizacione probleme (*Multi Objective Optimisation* - MOP).

Za razliku od klasičnih optimizacionih problema sa jednom optimizacionom funkcijom, koji imaju samo jedno rešenje, višekriterijumski optimizacioni problemi imaju više podjednako efikasnih rešenja, koja su poznata kao Pareto optimalna rešenja [30]. Sa druge strane, korisniku je obično potrebno samo jedno rešenje, pa MOP procedura tipično uključuje sledeća dva koraka:

1. Pronalaženje kompromisno-optimalnih rešenja koja su maksimalno različita (*maximally spread* koncept) u pogledu odgovarajućih vrijednosti optimizacionih funkcija.
2. Izbor jednog od dobijenih rešenja na osnovu dodatnih informacija o samoj problematici.

Ovaj pristup je manje subjektivan od pristupa gdje se optimizacione funkcije priorizuju na samom početku. MOP algoritmi koriste koncept dominacije za poređenje dva rešenja. Smatra se da rešenje s_1 dominira rešenjem s_2 ukoliko su ispunjeni sledeći uslovi:

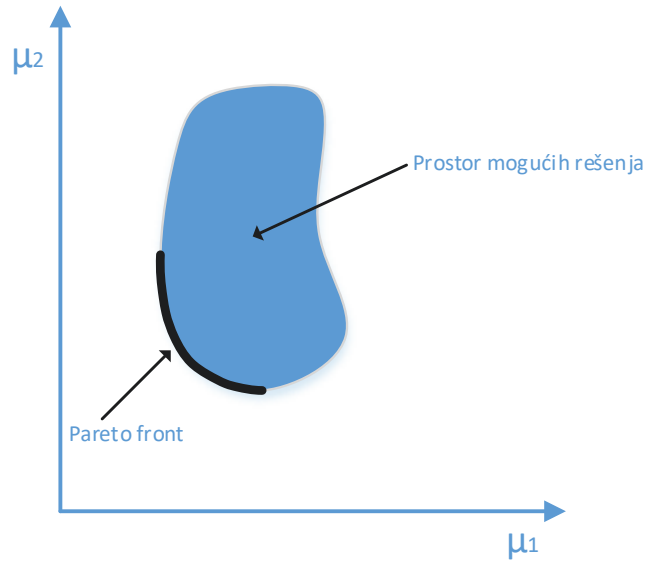
1. Rešenje s_1 nije gore od rešenja s_2 po svim kriterijumima;
2. Rešenje s_1 je striktno bolje od s_2 najmanje po jednom optimizacionom kriterijumu.

Ukoliko informacije o prioritetu optimizacionih funkcija nisu jasno definisane, ne može se tvrditi da je jedno rešenje bolje od drugog ukoliko ono nije dominantno u tom poređenju. Skup „nedominiranih” rešenja čini tzv. Pareto front (Slika 2.8). Zajednička karakteristika ovih rešenja je da nije moguće dodatno poboljšati vrijednost jedne od optimizacionih funkcija a pri tom ne degradirati vrijednost ostalih.

Postoji nekoliko metoda koji omogućavaju rešavanje MOP problema korišćenjem jednokriterijumske aproksimacije. U nastavku sekcije opisani su neki od njih.

2.6.1 WSM metod

WSM (*Weighted Sum Method*) metod dodjeljuje ne-negativne težinske faktore w_μ svakoj od M optimizacionih funkcija $f_\mu(x)$. Suma težinskih faktora je tipično 1. Matematička



Slika 2.8: Pareto front za optimizacione funkcije μ_1 i μ_2 .

definicija ovog metoda data je jednačinama:

$$\text{minimize} \quad F(x) = \sum_{\mu=1}^M w_{\mu} f_{\mu}(x) \quad (2.10a)$$

$$\text{subject to} \quad g_j(x) \geq 0 \quad j = 1, 2, \dots, J \quad (2.10b)$$

$$h_k(x) = 0 \quad k = 1, 2, \dots, K \quad (2.10c)$$

gde $g_j(x)$ i $h_k(x)$ predstavljaju optimizaciona ograničenja. Variranjem težinskih faktora moguće je pronaći različita rešenja koja pripadaju Pareto frontu. Međutim, ovaj metod ima i nekoliko nedostataka [101]: (i) ne postoji efikasan (*a priori*) metod za određivanje težinskih faktora kojim se može garantovati da će dobijeno rešenje biti prihvatljivo; (ii) nije moguće pronaći rešenja koja pripadaju nekonveksnim regijama Pareto fronta, (iii) variranje težinskih faktora često ne rezultira jednako distribuiranom i tačnom/potpunom reprezentacijom Pareto fronta.

2.6.2 Metod ϵ -ograničenja

Metod ϵ -ograničenja vrši minimizaciju/maksimizaciju samo jedne optimizacione funkcije - $f_m(x)$, dok se ostale koriste kao ograničenja:

$$\text{minimize } f_m(x) \quad (2.11a)$$

$$\text{subject to } f_\mu(x) \leq \epsilon_\mu \quad \mu = 1, 2, \dots, M \quad \mu \neq m \quad (2.11b)$$

$$g_j(x) \geq 0 \quad j = 1, 2, \dots, J \quad (2.11c)$$

$$h_k(x) = 0 \quad k = 1, 2, \dots, K \quad (2.11d)$$

Jedna od poteškoća kod ovog metoda je odabir ϵ_μ vrijednosti, s obzirom da najbolje vrijednosti nisu unaprijed poznate. Ukoliko se ϵ -vrijednosti povećavaju u suviše malim koracima, algoritam će imati veliki broj redundantnih koraka. Sa druge strane, ukoliko su koraci promjene previše veliki, neka od Pareto optimalnih rešenja neće biti pronađena. Potrebna je dodatna informacija od korisnika u cilju efikasnog pronalaženja rešenja u nekonveksnoj regiji. Međutim, metod ϵ -ograničenja ima nekoliko prednosti u odnosu na WSM metod:

1. WSM pronalazi ekstremume samo linearnih modela, pa je veliki broj iteracija redundantan i rezultira istim rešenjem. Metod ϵ -ograničenja pronalazi efikasna rešenja koja nisu obavezno ekstremumi.
2. Omogućava pronalaženje rešenja u nekonveksnoj Pareto regiji.
3. Moguće je kontrolisati broj efikasnih rešenja, što nije jednostavno kod WSM-a.

Glava 3

Tehnike za garanciju kvaliteta servisa

Termin „kvalitet servisa” (QoS) u literaturi se upotrebljava u kontekstu performansi koje mrežna infrastruktura garantuje mrežnom saobraćaju. To mogu biti garancije u pogledu dodijeljenog propusnog opsega, ograničenja kašnjenja, varijacije kašnjenja, vjerovatnoće gubitaka paketa ili samo dodjela većeg prioriteta pri prenosu. Različiti tipovi garancija potrebni su za mnoge moderne aplikacije da bi ispravno funkcionisale (Tabela 3.1) [102]. Garancija kvaliteta servisa zahtijeva implementaciju QoS mehanizama na nivou mrežnog uređaja i na nivou čitave mreže. U zavisnosti od klase saobraćaja kojoj pripadaju, paketi se smještaju u različite bafere. Na nivou uređaja mora se implementirati odgovarajući algoritam koji vrši opsluživanje bafera u skladu sa zahtjevima različitih saobraćajnih klasa u pogledu propusnosti i/ili kašnjenja. Pod pretpostavkom da mrežni uređaji imaju podršku za ovakve mehanizme, aplikacijama se mogu garantovati i performanse na nivou čitave mreže izvršavanjem QoS algoritama rutiranja i komunikacionih protokola.

S obzirom da je fokus ove disertacije na analizi TE tehnika sa QoS podrškom, u ovom poglavlju razmatrane su uglavnom tehnike za pružanje QoS garancija na nivou mreže. Objašnjene su dvije ključne QoS arhitekture koje je standardizovao IETF: IntServ [28] i DiffServ [29], a zatim su identifikovane dodatne QoS mogućnosti koje nudi OpenFlow protokol u SDN mrežama.

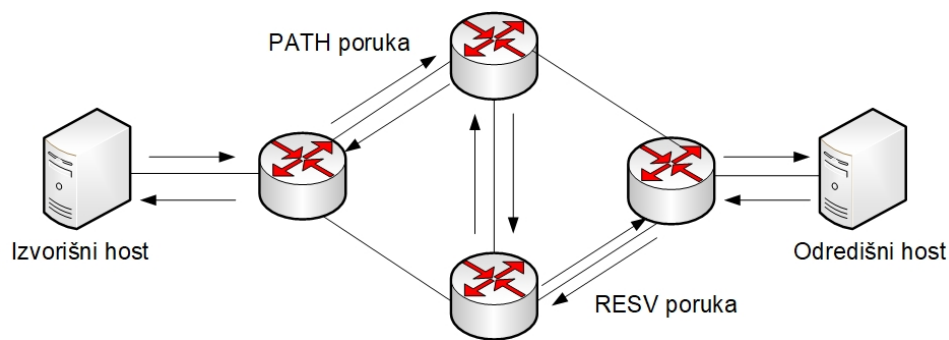
Tabela 3.1: Zahtjevi različitih aplikacija u pogledu kašnjenja i brzine prenosa.

Aplikacija	Brzina prenosa (b/s)	Kašnjenje
Virtuelna realnost	1 Gb/s	1 ms
Industrijske aplikacije za automatizaciju proizvodnje	10 Mb/s	0.25-10 ms
Inteligentni transportni sistemi	10-700 Mb/s	10-100 ms
Udaljena medicinska dijagnoza/ intervencija	100 Mb/s	1 ms
Taktilni Internet	1 Gb/s	5 ms
Pametna mreža	10-1500 kb/s	1-20 ms
Robotika i teleprisustvo	100 Mb/s	1 ms
Transfer fajlova	1-10 Mb/s	10 s
Email	10 kb/s	4 s
VOIP	64 kb/s	150 ms
Video konferencija	128 kb/s - 6 Mb/s	150 ms

3.1 IntServ

IntServ arhitektura omogućava pružanje apsolutnih QoS garancija na nivou konekcije koristeći mehanizme za rezervaciju resursa (propusni opseg, baferi). Kod ovog QoS modela, aplikacija prije slanja podataka koristi RSVP [81] protokol da signalizira svakom mrežnom uređaju na odabranoj ruti svoje servisne zahtjeve u pogledu propusnog opsega ili kašnjenja. Sa prenosom podataka se počinje samo ukoliko svaki uređaj na ruti može rezervirati zahtijevani propusni opseg. Na Slici 3.1 su ilustrovane bazične operacije RSVP protokola koje se izvršavaju posredstvom kontrolnih poruka PATH i RESV [81]. Izvorišni host šalje PATH poruku u kojoj specificira karakteristike saobraćajnog toka. Nakon prijema PATH poruke, RSVP modul svakog rutera pamti adresu rutera od kojeg je poruka primljena. Prijemni host odgovara RESV porukom koja se prosleđuje istim putem unazad, i rezervira resurse na svim ruterima duž rute. Konekcija se uspostavlja ukoliko je zahtjev prihvaćen od strane svih rutera na putanji, ili odbija ukoliko bilo koji od rutera nema dovoljno resursa na raspolaganju [25].

Pored *best-effort* servisa, IntServ pruža još dvije klase servisa: *Garantovani servis* i *Servis kontrolisanog opterećenja*. *Garantovani servis* aplikaciji garantuje potrebni propusni opseg, kašnjenje i prenos bez gubitaka. Namijenjen je *real-time* aplikacijama, kao



Slika 3.1: RSVP protokol.

što su aplikacije za prenos govora ali i brojne IoT aplikacije koje obavljaju vremenski kritične zadatke. *Servisom kontrolisanog opterećenja* ne pružaju se kvantitativne QoS garancije, već samo "utisak blago opterećene mreže". Drugim riječima, ovaj servis teži da aproksimira uslove koje aplikacijama nudi *best-effort* servis u uslovima slabog intenziteta saobraćaja. Namijenjen je širokom spektru aplikacija koje su originalno dizajnirane za današnji Internet, ali su veoma osjetljive na zagušenja u mreži. Sa druge strane, smatra se da se određeni gubici i kašnjenje mogu tolerisati.

Iako IntServ omogućava klasifikaciju servisa i može pružiti apsolutne QoS garancije za prioritetne tokove, njegova primjena ograničena je samo na manje kompanijske mreže iz sledećih razloga:

- Svi ruteri moraju podržavati RSVP, kontrolu pristupa, klasifikaciju paketa i kompleksne algoritme za opsluživanje bafera. Takođe, ruteri moraju da čuvaju informacije o svakoj aktivnoj rutiranoj konekciji, što postaje veoma zahtjevno u većim mrežama.
- Prilikom uspostavljanja novih i brisanja starih ruta generiše se ogromna količina kontrolnog saobraćaja.
- Zahtjevi za rezervaciju resursa šalju se rutama koje određuju tradicionalni protokoli rutiranja kao što su OSPF i IS-IS. Stoga, može doći do odbijanja QoS zahtjeva ako na izračunatoj ruti nema dovoljno resursa, iako isti eventualno može biti zadovoljen na nekoj od alternativnih ruta.
- IntServ može pružiti garancije u pogledu maksimalnog kašnjenja u redovima čeka. Međutim, ukupno kašnjenje u mreži čine još tri komponente: propagaciono kašnjenje, kašnjenje usled prenosa i kašnjenje uslijed obrade u čvorištu.

3.2 DiffServ

Za razliku od IntServ arhitekture koja vrši *end-to-end* rezervaciju resursa za prioritetne tokove, kod DiffServ arhitekture [29] paketi se označavaju na ulazu u mrežu i klasifikuju u konačan broj saobraćajnih klasa. Identifikovanje različitih klasa tokova u agregiranom saobraćaju vrši se na osnovu DSCP (*DiffServ Code Point*) polja u IP zaglavlju. QoS servisi sprovode se nezavisno na svakom hopu i za svaki paket, tako što se vrijednost DSCP polja iz zaglavlja paketa mapira sa odgovarajućim redom čekanja, brzinom opsluživanja i vjerovatnoćom odbacivanja paketa u slučaju zagušenja. IETF je do sada standardizovao četiri različite klase servisa (PHBs - *Per Hop Behaviours*):

1. *Default* - tradicionalni *best-effort* servis;
2. EF (*Expedited Forwarding*) - Klasa najvećeg prioriteta koja se opslužuje sa najmanjim kašnjenjem, *jitter*-om i vjerovatnoćom odbacivanja paketa u slučaju zagušenja. Ukoliko agregirani EF saobraćaj postane previše obiman, može doći do stagnacije ostalog saobraćaja, pa je neophodno na osnovu trenutnih EF tokova u mreži vršiti procjenu o tome da li treba prihvatiti nove tokove ovog tipa;
3. AF (*Assured Forwarding*) klasa - obezbjeđuje pouzdan i redosledan prenos. Ova klasa dobija bolji tretman od *best-effort* klase, ali kvalitet servisa zavisi od trenutnog stanja u mreži. AF klasa je prikladna za saobraćaj koji zahtijeva veću propusnost ali nije previše osjetljiv na kašnjenje;
4. CS (*Class Selector*) klasa - uvedena je radi kompatibilnosti sa starijim sistemima koji su koristili najviša tri bita u ToS (*Type of Service*) polju IPv4 zaglavlja za označavanje prioriteta saobraćaja. S tim u vezi, definisano je 7 podklasa koje označavaju različite prioritete.

Glavna prednost DiffServ-a je njegova skalabilnost, s obzirom da ruteri održavaju informacije samo o konačnom broju klasa umjesto o pojedinačnim tokovima. Inteligencija je koncentrisana na perifernim ruterima, koji pored klasifikacije obavljaju kontrolu pristupa i oblikovanje saobraćaja u skladu sa administrativnom politikom. Samim tim, ovo rešenje je značajno jednostavnije i jeftinije za implementaciju od IntServ-a. Međutim, iako se agregacijom tokova u klase eliminiše problem skalabilnosti, na ovaj način se mogu ostvariti samo relativne QoS garancije.

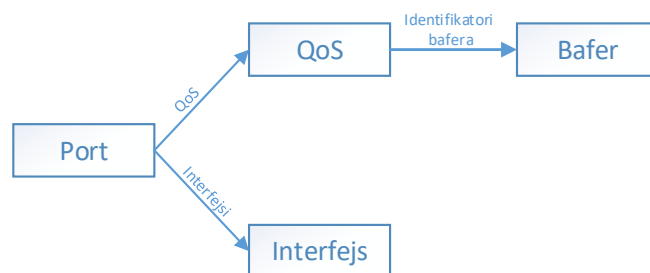
3.3 QoS u OpenFlow mrežama

QoS mogućnosti OpenFlow protokola su još uvijek veoma ograničene [103]. U ranim verzijama protokola (1.0-1.1) uvedena je podrška za mapiranje paketa u izlazne baferne i podešavanje minimalne brzine opsluživanja bafera. OpenFlow 1.2 dodao je mogućnost prikupljanja statistika na nivou svih bafera sviča. Još jedno bitno QoS unapređenje u ovoj verziji je uvođenje *max-rate* atributa bafera, koji služi za ograničavanje maksimalne brzine opsluživanja, kao i eksperimentalnih atributa koji omogućavaju proizvođačima SDN uređaja i mrežnim programerima naknadno dodavanje novih funkcionalnosti. OpenFlow 1.3 uveo je koncept *meter* tabela u cilju postizanja granularne QoS kontrole u OpenFlow mrežama. Dok izlazni baferi kontrolišu odlaznu brzinu saobraćaja, *meter* tabele se mogu koristiti za monitoring i kontrolu brzine prenosa toka prije prosleđivanja na izlazni interfejs. Sa OpenFlow verzijom 1.5 omogućeno je korišćenje više *meter* instrukcija na nivou jednog toka. OpenFlow svičevi takođe mogu da čitaju i modifikuju ToS bite u IP zaglavlju.

3.3.1 Baferi

Kao što je već pomenuto, baferi sa kontrolisanom minimalnom i maksimalnom brzinom opsluživanja dio su specifikacije OpenFlow protokola još od njegovih ranih verzija. Međutim, OpenFlow ne definiše mehanizme upravljanja baferima OpenFlow svičeva, već samo omogućava prikupljanje njihovih statistika. Upravljački zadaci (kreiranje, brisanje i modifikovanje bafera) delegirani su protokolima kao što su OF-Config [104] i OVSDB [105].

OVSDB koristi više tabela za upravljanje OpenvSwitch softverskim SDN svičevima [106], uključujući: tabele tokova, tabele portova, NetFlow [107] tabele i druge. Posebne tabele kreiraju se za upravljanje baferima (*Queue* tabele) i kvalitetom servisa (QoS tabele). Ove tabele se međusobno nezavisne, i mogu postojati čak i kada nisu referencirane nekim portom OpenFlow sviča. Veze između relevantnih tabela OpenvSwitch-a prikazane su na Slici 3.2. Tabela portova je korelisana sa interfejs tabelom i QoS tabelom. Veza između port tabele i interfejs tabele je obavezna, što znači da svaki port mora biti pridružen određenom interfejsu. Sa druge strane, veza sa QoS tabelom je opcionalna. Svaki port može biti povezan sa najviše jednom QoS politikom, dok QoS tabeli može biti pridruženo više bafera.



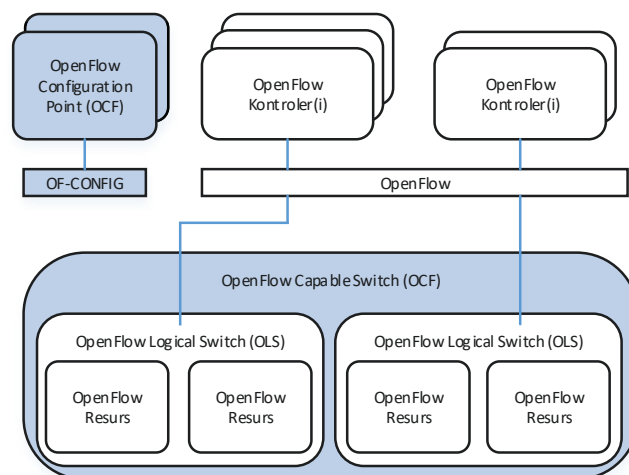
Slika 3.2: Tabele bafera i QoS tabele OpenvSwitch-a.

Za razliku od OVSDB protokola koji je podržan samo kod OpenvSwitch-eva, OF-Config protokol je podržan kod većine komercijalnih hardverskih i softverskih SDN svičeva. Dok OpenFlow generalno funkcioniše na nivou tokova kreirajući pravila za prosleđivanje saobraćaja, OF-Config omogućava OpenFlow kontroleru da upravlja resursima kao što su portovi, baferi, i da sprovodi QoS politike u ravni podataka. Ključni elementi OF-Config protokola i korišćena terminologija prikazani su na Slici 3.3:

- *OpenFlow Logical Switch (OLS)*: apstrakcija koja predstavlja OpenFlow svič.
- *OpenFlow Capable Switch (OCS)*: operativni kontekst koji sadrži jedan ili više OLS komponenti. Predstavlja fizički svič ili virtuelno mrežno okruženje koje hostuje jedan ili više logičkih svičeva.
- *OpenFlow Configuration Point (OCF)*: servis za upravljanje OCS komponentama. Može biti dio OpenFlow kontrolera ili zasebnog entiteta zaduženog za upravljanje mrežom.
- *OpenFlow resursi*: resursi OLS i OCS modula kao što su intefejsi, baferi, itd.

Nakon što se QoS politike i baferi konfiguriraju na sviču, saobraćajni tokovi se mogu usmjeriti u određeni bafer korišćenjem OpenFlow *set-queue* akcije. Ovom akcijom paketi toka prosleđuju se u bafer sa navedenim identifikatorom. Ukoliko više tokova koristi isti bafer, agregirana brzina prenosa biće kontrolisana na izlaznom interfejsu u skladu sa *min-rate* i *max-rate* atributima bafera. Ovi atributi su definisani u OpenFlow specifikaciji na sledeći način:

- *min-rate*: definiše minimalnu brzinu opsluživanja bafera. Ukoliko je *min-rate* atributu dodijeljena vrijednost, svič će prioritizovati ovaj bafer (i bilo koji tok koji se



Slika 3.3: Elementi OF-Config protokola.

prosleđuje tim baferom) u cilju ostvarivanja potrebne brzine opsluživanja, degradirajući ako je potrebno brzine prenosa drugih tokova. Ukoliko se na jednom portu sviča konfiguriše više bafera čija suma *min-rate* atributa prevazilazi kapacitet linka, brzine opsluživanja svih bafera biće penalizovane i kapacitet će biti izdijeljen proporcionalno *min-rate* atributima.

- *max-rate*: maksimalna brzina opsluživanja bafera. Ukoliko je dolazna brzina paketa u bafer veća od specificirane *max-rate* brzine, svič će baferovati pakete i/ili odbacivati ih po potrebi tako da *max-rate* zahtjev bude ispunjen.

Realizacija mehanizama za kontrolu maksimalne i minimalne brzine opsluživanja bafera nije dio OpenFlow specifikacije i zavisi od konkretne implementacije OpenFlow sviča.

3.3.2 OpenFlow *meter* tabele

Meter tabela je novi koncept uveden sa OpenFlow verzijom 1.3. Za razliku od OpenFlow bafera koji kontrolišu brzinu opsluživanja paketa na izlaznim interfejesima, *meter* tabele koriste se za nadgledanje dolazne brzine paketa. Zapisi *meter* tabele (tzv. mjerači) ne odnose se na portove sviča, već na saobraćajne tokove. Tabela tokova sadrži instrukcije o tome koji mjerač treba primijeniti nad kojim tokom. Mjerač kontroliše agregiranu

ID Mjerača	Opseg	Brojač
------------	-------	--------

Slika 3.4: Format zapisa u *meter* OpenFlow tabeli.

brzinu prenosa svih njemu pridruženih tokova. Na ovaj način moguće je implementirati različite QoS tehnike.

Zapisi *meter* tabele sadrže sledeće komponente, kao što je prikazano na Slici 3.4:

- Identifikator mjerača: 32-bitni cijeli broj koji na jedinstven način identifikuje mjerač.
- Lista opsega mjerača: svaki „opseg” definiše različitu brzinu prenosa i sadrži instrukcije o tome na koji način treba procesirati pridružene tokove koji prelaze definisanu brzinu prenosa.
- Brojači: sadrže informaciju o broju procesiranih paketa. Koristi se uglavnom za statističke svrhe.

Postoje dva tipa „opsega” koji definišu na koji način se paketi procesiraju: *drop* i *DSCP remark*. U slučaju primjene *drop* opsega, paketi koji pređu ograničenje brzine prenosa se odbacuju. Sa druge strane, *DSCP remark* opseg samo povećava vjerovatnoću odbacivanja paketa za odgovarajući tok podešavanjem DSCP polja u IP zaglavlju paketa. Na ovaj način, kombinovanjem mjerača i OpenFlow bafera, moguće je realizovati i kompleksna QoS okruženja kao što je DiffServ. Na primjer, ukoliko je na dva toka čija je brzina generisanja paketa 50 Mb/s i ToS vrijednost 64 primijenjen mjerač od 30 Mb/s, ToS biti paketa koji prelaze brzinu od 30Mb/s biće modifikovani na 32, čime se povećava vjerovatnoća njihovog odbacivanja u slučaju zagušenja.

3.4 QoS algoritmi rutiranja: analiza kompleksnosti

Jedan od definisanih zahtjeva za SDN kontrolno okruženje predloženo u ovoj disertaciji je računanje ruta u realnom vremenu. S tim u vezi, u ovoj sekciji predstavljani su osnovni problemi QoS rutiranja i diskutovana je njihova kompleksnost.

Kompleksnost QoS algoritma korelisana je sa strukturom težinskih faktora linkova, koji se formulišu u skladu sa odabranom metrikom rutiranja. Metrika rutiranja definiše tip QoS garancija koji mreža može da pruži. Drugim riječima, ukoliko se QoS zahtjevi saobraćajnog toka ne mogu mapirati sa kombinacijom metrika rutiranja koje se koriste u mreži, to znači da zahtjev ne može biti podržan. Metrike koje se obično koriste u mrežama mogu se klasifikovati u tri kategorije [25, 108]:

1. aditivne metrike - za koje važi da je težinski faktor rute $P = \{u_1, u_2, u_3 \dots u_n\}$ jednak sumi težinskih faktora linkova duž rute. Ukoliko se za težinski faktor linka $(i, j) \in L$ koji odgovara metrici k usvoji notacija $w_k(i, j)$, a za težinski faktor rute P notacija $w_k(P)$, pomenuti uslov se može formulisati na sledeći način: $w_k(P) = w_k(u_1, u_2) + w_k(u_2, u_3) + \dots + w_k(u_{n-1}, u_n)$.
2. multiplikativne metrike - kod kojih je težinski faktor rute jednak proizvodu težinskih faktora linkova duž rute, tj.: $w_k(P) = w_k(u_1, u_2) \cdot w_k(u_2, u_3) \cdot \dots \cdot w_k(u_{n-1}, u_n)$.
3. konkavne metrike - kod kojih važi: $w_k(P) = \min\{w_k(u_1, u_2), w_k(u_2, u_3), \dots, w_k(u_{n-1}, u_n)\}$.

Aditivne metrike uključuju kašnjenje, varijaciju kašnjenja, „cijene” linkova i broj hopova. Pouzdanost je multiplikativna metrika, dok je propusnost konkavna. Multiplikativne metrike se mogu svesti na aditivne tako što se težinski faktori linkova w_k i ograničenja C zamjene odgovarajućim logaritamskim vrijednostima $\log(w_k)$ i $\log(C)$.

3.4.1 QoS rutiranje sa jednom metrikom

U najjednostavnijem slučaju QoS zahtjevi toka se mogu predstaviti jednom od prethodno navedenih metrika. Problem se zatim svodi ili na optimizacioni problem ili na problem ograničenja [109]. Metrike se dalje mogu klasifikovati u rutom-ograničene i linkom-ograničene. Konkavne metrike su linkom-ograničene jer je metrika rute određena karakteristikama *bottleneck* linka. Aditivne i multiplikativne metrike su rutom-ograničene, jer metrika rute zavisi od svih linkova koji joj pripadaju [25]. U skladu sa poslednjom klasifikacijom, izdvajaju se četiri osnovna problema QoS rutiranja sa jednom metrikom:

1. **problem optimizacije linka:** Za dati graf $G(V, L)$ i težinske faktore linkova $w_k(i, j) \forall (i, j) \in L$ koji odgovaraju konkavnoj metrici k , pronaći rutu P od izvorišnog čvora u do destinacionog čvora v koja maksimizuje $w_k(P)$.

Primjer ovog problema je pronalaženje rute sa najvećim propusnim opsegom.

2. **problem ograničenja linka:** Za dati graf mreže $G(V, L)$ i težinske faktore linkova $w_k(i, j) \forall (i, j) \in L$ koji odgovaraju konkavnoj metrici k , i QoS ograničenje C , pronaći rutu P od izvorišnog čvora u do destinacionog čvora v koja zadovoljava uslov $w_k(P) \leq C$.

Pronalaženje rute koja ispunjava zahtjev u pogledu propusnosti pripada ovoj klasi problema. Problem se može riješiti računanjem optimalne rute i provjerom ispunjenosti uslova $w_k(P) \leq C$. Drugi pristup je uklanjanje linkova grafa koji ne zadovoljavaju ograničenje C , a zatim primjena SPF algoritma na novoformiranom grafu.

3. **problem optimizacije rute:** Za dati graf $G(V, L)$ i težinske faktore linkova $w_k(i, j) \forall (i, j) \in L$ koji odgovaraju aditivnoj metrici k , pronaći rutu P od izvorišnog čvora u do destinacionog čvora v koja minimizuje $w_k(P)$.

Primjeri ove klase problema su pronalaženje rute sa najmanjim brojem hopova, najmanjom cijenom ili kašnjenjem. Problem je moguće riješiti direktnom primjenom Dijkstra ili Bellman-Ford algoritma [110].

4. **problem ograničenja rute:** Za dati graf mreže $G(V, L)$ i težinske faktore linkova $w_k(i, j) \forall (i, j) \in L$ koji odgovaraju aditivnoj metrici k , i QoS ograničenje C , pronaći rutu P od izvorišnog čvora u do destinacionog čvora v koja zadovoljava uslov $w_k(P) \leq C$.

Primjer problema je pronalaženje rute sa kašnjenjem ispod određene vrijednosti.

3.4.2 Rutiranje na osnovu više QoS kriterijuma

Problemi predstavljeni u prethodnoj sekciji su male računske složenosti i mogu se izvršavati u polinomijalnom vremenu. Međutim, često je više metrika potrebno za opis zahtijevanog kvaliteta servisa. U zavisnosti od kombinacije metrika koje se koriste u ovakvim situacijama, računska kompleksnost algoritma rutiranja može rasti do nivoa nepotrebljivosti u realnim komunikacionim sistemima. Bilo koja kombinacija dvije ili više aditivnih ili multiplikativnih metrika stvara tzv. *NP-complete* problem koji se ne može riješiti u polinomijalnom vremenu u složenim mrežama [25, 109]. Polinomijalna kompleksnost moguća je samo u slučaju primjene jedne konkavne metrike (npr. propusnost)

Tabela 3.2: Računska kompleksnost kompozitnih problema QoS rutiranja.

	Optimizacija linka	Ograničenje linka	Optimizacija rute	Ograničenje rute
Optimizacija linka		Polinomijalna	-	Polinomijalna
Ograničenje linka		Polinomijalna	Polinomijalna	Polinomijalna
Optimizacija rute			-	NP-complete
Ograničenje rute				NP-complete

i jedne metrike nekonkavnog tipa, najčešće kašnjenja ili broja hopova. U ostalim slučajevima, obično se primjenjuje pristup rutiranja na osnovu jedne metrike koja predstavlja kombinaciju metrika iz polaznog problema. Osnovna ideja ovog pristupa je kombinovanje različitih QoS parametara u jednu mjeru kvaliteta koja će se koristiti prilikom donošenja odluka o rutiranju. Međutim, kombinovana metrika u najboljem slučaju može poslužiti samo kao indikator performansi i ne može se iskoristiti za pružanje apsolutnih QoS garancija. Ipak, u mrežama sa malim propagacionim kašnjenjem, korišćenjem samo metrike propusnosti moguće je zadovoljiti QoS zahtjeve i u pogledu kašnjenja. Na primjer, WFQ i WF2Q algoritmi opsluživanja bafera jasno definišu mapiranje ograničenja kašnjenja u ograničenje propusnosti [111]. Međutim, nedostatak ovog pristupa je neefikasno korišćenje mrežnih resursa, jer se za male tokove sa strogim zahtjevima u pogledu kašnjenja rezerviše znatno veći propusni opseg od potrebnog.

U Tabeli 3.2 prikazani su relevantni kompozitni problemi QoS rutiranja. Dvostruki optimizacioni problemi su izostavljeni, jer se smatraju neracionalnim. Problem dvostruke optimizacije se može aproksimativno riješiti svođenjem kompozitnog problema na dva problema sa jednom metrikom rutiranja, pri čemu se druga optimizacija vrši ukoliko postoji više od jedne optimalne rute po prvom kriterijumu. Podaci prikazani u Tabeli 3.2 ukazuju da već u slučaju primjene dvije metrike rutiranja veliki broj realnih QoS problema postaje nerešiv u realnom vremenu. Od ključne važnosti za njihove rešavanje je razvoj heuristika koje nude efikasan kompromis između optimalnosti ruta i računske kompleksnosti.

Glava 4

Predlog kontrolnog okruženja za inženjering saobraćaja u SD-ISP mrežama

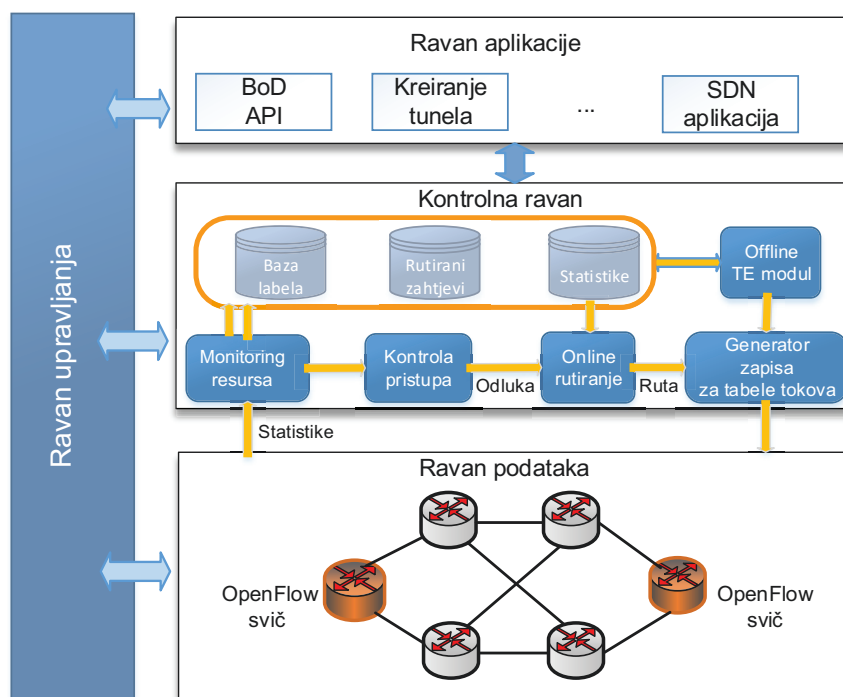
SDN kontroler predstavlja pogodnu platformu za razvoj naprednih TE tehnika u formi centralizovanih aplikacija koje imaju globalni uvid u stanje mreže i koje mogu dinamički reagovati na svaki događaj u mreži. Svoju primjenu SDN je inicijalno našao u kompanijskim i DC mrežama [10–12]. Relativno nedavno, Cloud provajderi (npr. Microsoft i Google) demonstrirali su potencijal domenskih SDN TE tehnika za poboljšanje iskorisćenosti mrežnih resursa i tolerancije na kvarove [1, 9, 13]. Pored toga, evidentan je trend primjene SDN rešenja za inženjering saobraćaja između više administrativnih domena [112, 113]. Međutim, ova rešenja ne mogu u potpunosti zadovoljiti ključne zahtjeve ISP mreža. Naime, u DC i *inter*-DC mrežama moguće je dinamički kontrolisati brzinu generisanja saobraćaja i planirati vremenski raspored saobraćajnih tokova kada je to potrebno [95, 114]. Na ovaj način, moguće je izbjeći iznenadne fluktuacije saobraćaja i spriječiti začešćenje mrežnih linkova. Sa druge strane, ISP mreže nemaju ovu privilegiju jer klijentima moraju pružiti QoS garancije specificirane u SLA ugovorima. Takođe, većina predloženih rešenja nije dovoljno skalabilna za ISP mreže, s obzirom da formulacija odgovarajućeg TE problema može sadržati milione ograničenja i varijabli.

U ovom poglavlju razmatran je jedan od fundamentalnih izazova u ISP mrežama - skalabilna alokacija resursa za konkurentne QoS zahtjeve. Predloženo je SDN kontrolno okruženje, na osnovu sledećih razmatranih ciljeva:

- Djelimična nadogradnja mreže - ISP provajderi mogu iskoristiti konvencionalnu MPLS opremu u jezgru mreže, dok su SDN uređaji neophodni samo na ivici mreže za efikasno balansiranje saobraćaja preko više MPLS tunela;
- Pružanje garancija u pogledu kašnjenja, propusnosti i pouzdanosti;
- Brza restauracija ruta u slučaju otkaza linkova u mreži;
- Sprovođenje kontrole pristupa u skoro realnom vremenu;
- Minimizacija interakcije sa svičevima u ravni podataka;
- Donošenje odluka na osnovu agregiranih saobraćajnih statistika u cilju smanjenja radnog opterećenja SDN kontrolera.

Jedna od osnovnih karakteristika predloženog rešenja je podjela kontrolne logike SDN kontrolera u *offline* i *online* modul. *Online* modul zadužen je za uspostavljanje virtuelnih linkova na zahtjev korisnika, i donošenje odluka o rutiranju i kontroli pristupa u skoro realnom vremenu. *Offline* modul periodično optimizuje distribuciju ukupnog saobraćaja preko instaliranih tunela u mreži. Ovakvom podjelom kontrolnih funkcionalnosti napravljen je kompromis između optimalnosti rutiranja i skalabilnosti kontrolne ravni.

U Sekciji 4.1 predstavljen je razmatrani model sistema i predložen dizajn SDN kontrolera za softverski-definisane (SD) ISP mreže. Performanse predloženog rešenja analizirane su simulacionom metodologijom koja je objašnjena u Sekciji 4.2. Simulacioni rezultati predstavljeni su u Sekciji 4.3 za scenarije bez i sa garancijom pouzdanosti, odvojeno, u cilju pružanja jasnijeg uvida na uticaj različitih elemenata TE algoritma na cjelokupne mrežne performanse. Prilikom analize izvršeno je poređenje sa CSPF pristupom koji je dominantno zastupljen u današnjim MPLS mrežama, *oblivious* TE pristupom baziranim na *hose* saobraćajnom modelu i TE rešenjima koja koriste pogodnosti fleksibilnog SDN rutiranja u jezgru mreže. Simulacioni rezultati pokazuju da predloženo rešenje pruža bolje performanse od klasičnog CSPF i *oblivious* TE pristupa čak i kada se *offline* TE modul izvršava periodično, u dužim vremenskim intervalima. Dobijeni rezultati takođe pokazuju da pri odgovarajućoj učestanosti TE optimizacija predloženi pristup ostvaruje servisni kapacitet relativno blizak onom koji se postiže u mrežama sa SDN jezgrom. Pri tome, značajno smanjuje kontrolni *overhead*.



Slika 4.1: Razmatrana arhitektura mreže.

4.1 Model sistema

Predmet analize je mrežna arhitektura prikazana na Slici 4.1, koja je je organizovana u ravan podataka, kontrolnu ravan, ravan aplikacije i ravan upravljanja. Ravan podataka čine uređaji u jezgri (okosnici) i na periferiji (ivici) mreže. Pretpostavka je da uređaji na periferiji mreže podržavaju OpenFlow protokol, dok jezgro može biti bazirano na klasičnoj MPLS tehnologiji. Između svakog izvorišno-odredišnog (IO) para OpenFlow svičeva konfigurisano je do K tunela za prenos saobraćaja. Da bi se u slučaju otkaza linkova degradacija performansi svela na što manju mjeru, za svaki IO par tuneli su odabrani tako da najviše q od njih dijeli isti link. U slučaju ispada linka ili sviča, svičevi na ivici mreže preusmjeravaju dolazni saobraćaj preko alternativnih tunela, proporcionalno težinskim faktorima za balansiranje opterećenja koje SDN kontroler podešava periodično. Ovaj oblik dinamičkog balansiranja saobraćaja moguće je implementirati sa OpenFlow *group* tabelama [115, 116].

SD-ISP mreža nudi BoD servis klijentima preko odgovarajućeg NI interfejsa. Preko ovog interfejsa korisnici mogu tražiti uspostavljanje virtuelnih linkova ili dinamički prilagođavati parametre svojih konekcija iznenadnim i privremenim potrebama. Parametri konekcije, između ostalog, uključuju QoS ograničenja u pogledu propusnosti, kašnjenja i

pouzdanosti. U daljoj analizi pretpostavljeno je da se BoD servisna ponuda ISP provajdera može klasifikovati na sledeće klase:

- QoS1 - Servis garancije propusnosti za konekciju.
- QoS2 - Servis kojim se garantuje propusnost i maksimalno kašnjenje ispod određene vrijednosti. Namijenjen je aplikacijama koje prenose vremenski kritične sadržaje.
- QoS3 - Servis najvećeg prioriteta kojim se pružaju garancije u pogledu propusnosti, kašnjenja i pouzdanosti. Predviđen je za ultra-pouzdanu komunikaciju koja ne toleriše velika kašnjenja. Pouzdanost je garantovana u formi robustnosti na otkaz unaprijed definisanog broja (n) linkova.

Po potrebi moguće je definisati više QoS2 i QoS3 podklasa, sa različitim ograničenjima kašnjenja, kako bi se diferencijacija servisa učinila granularnijom. Pored BoD servisa, ISP nudi i klasični *best-effort* servis za saobraćaj koji ne zahtijeva QoS garancije. Usvajanje ovakvog modela klasifikacije saobraćaja inspirisano je preporukama IETF DiffServ radne grupe [117], koje definišu tri osnovne kategorije servisa: *Premium*, *Ordinary* i *Best-effort*. U razmatranoj mrežnoj arhitekturi QoS2 i QoS3 klase odgovaraju *Premium* servisu, dok je QoS1 *Ordinary* servis koji toleriše veća kašnjenja. *Best-effort* saobraćaj koristi resurse koji nisu rezervisani za prioritetnu *Premium* i *Ordinary* klasu, bez bilo kakvih QoS garancija.

Predloženi dizajn SDN kontrolera ima dva modula koja su odgovorna za kontrolu saobraćaja: *online* modul za rutiranje i *offline* modul za TE optimizaciju. Prvi modul sprovodi kontrolu pristupa i određuje inicijalne rute za BoD zahtjeve na osnovu informacija o dostupnom kapacitetu u mreži. *Offline* TE modul periodično razmatra trenutnu šemu rutiranja tokova i optimizuje težinske faktore za balansiranje saobraćaja u cilju poboljšanja iskorišćenosti mrežnih resursa. Balansiranje saobraćaja se vrši preko (do K) unaprijed instaliranih tunela.

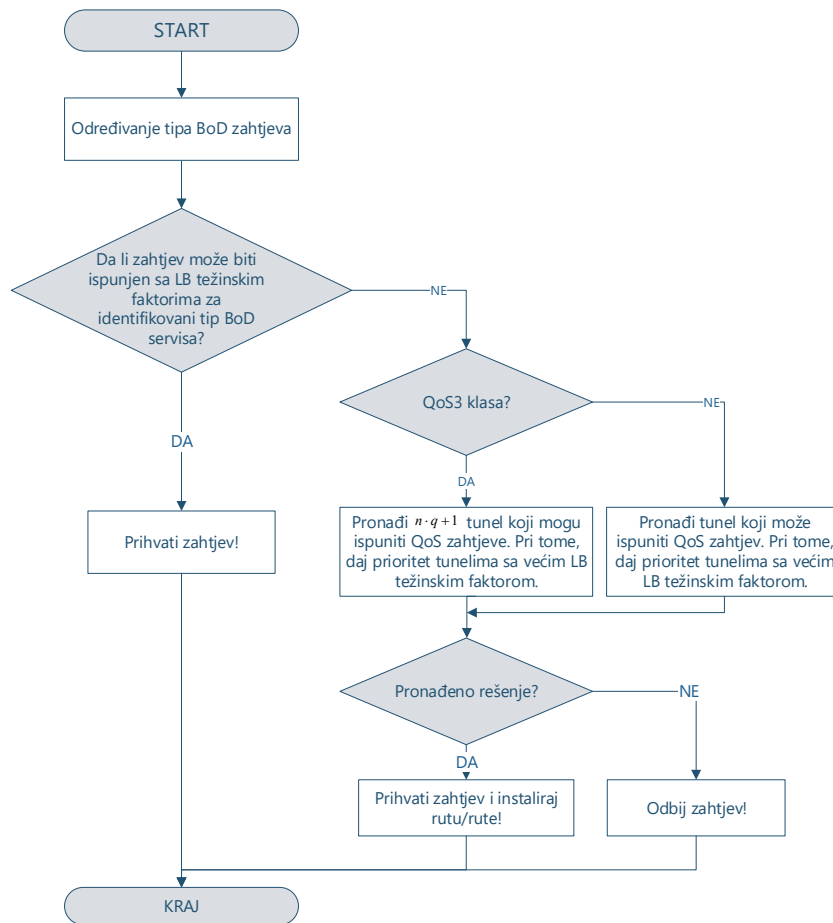
Saobraćaj se na ulazu u mrežu klasifikuje u četiri klase, koje odgovaraju pomenutim kategorijama servisa. Svaka klasa se taguje jedinstvenom labelom. Na primjer, VLAN (*Virtual Local Area Network*) identifikator ili MPLS labela se može koristiti prilikom klasifikacije [1]. Dijeljenje saobraćaja između tunela se vrši na nivou toka, primjenom *hash* funkcije nad poljima u zaglavlju paketa. S obzirom da OpenFlow svičevi mogu vršiti prosljeđivanje paketa na osnovu velikog broja različitih polja zaglavlja [7], pretpostavka je

da se dijeljenje saobraćaja može fleksibilno realizovati na ovaj način. Zapisi u tabelama tokova ivičnih SDN svičeva mapiraju dolazne pakete sa jednom od ukupno 4 *group* tabele na osnovu destinacione IP adrese i pridružene labele. Svaka *group* tabela odgovara različitoj klasi servisa i konfigurisana je sa (do) K tunela i K težinskih faktora, koji određuju udio dolaznog saobraćaja koji se prosleđuje svakim od tunela.

4.1.1 Modul za *online* rutiranje

Modul za *online* rutiranje sprovodi kontrolu pristupa prilikom prijema korisničkih zahtjeva za uspostavljanje virtuelnih linkova, i instalira nove rute po potrebi. S obzirom da virtuelni linkovi zahtijevaju QoS garancije, svaki od njih mora biti odobren od strane SDN kontrolera individualno. Stoga, ovaj modul je dizajniran tako da izvršava računski jednostavan QoS algoritam rutiranja koji teži da minimizuje instalaciju novih zapisa u tabelama tokova. Koraci izvršavanja algoritma prikazani su na Slici 4.2.

U prvom koraku algoritam određuje tip QoS zahtjeva u cilju filtriranja podskupa tunela koji zadovoljavaju definisano ograničenje u pogledu kašnjenja. Zatim, algoritam provjerava da li je moguće ispuniti zahtjev u pogledu propusnosti primjenom postojećih težinskih faktora za balansiranje opterećenja (*Load Balancing* - *LB*) koje je izračunao *offline* TE modul. Ukoliko jeste, korisnički zahtjev se prihvata. Pri tome, nije potrebno instalirati nove zapise u tabelama tokova. Ulazni svičevi mapiraju saobraćaj koji generiše korisnik sa odgovarajućom *group* tabelom, a zatim taguju i balansiraju pojedinačne tokove preko više unaprijed konfigurisanih tunela. Ukoliko zahtjev ne može biti ispunjen na ovaj način, algoritam provjerava da li neki od tunela sa dovoljno malim kašnjenjem može pružiti potrebnu propusnost, pod pretpostavkom da se saobraćaj prenosi samo jednom rutom. Ukoliko postoji više tunela koja ispunjavaju ovaj zahtjev, prioritet se daje onom sa najvećim težinskim faktorom za balansiranje ulaznog opterećenja, tj. taj tunel se bira sa najvećom vjerovatnoćom. U ovom slučaju, prosleđivanje saobraćaja neće ići preko *group* tabela, pa je potrebno dodati nove zapise u tabelama tokova ulaznih svičeva. Za QoS3 zahtjeve, rezultat algoritma je lista tunela. Jedan od ovih tunela je primarni tunel koji će prenositi saobraćaj korisnika, dok su ostali elementi liste alternativni tuneli koji služe kao *backup* u slučaju otkaza linka na primarnoj ruti. Preciznije, algoritam vraća $n \cdot q + 1$ tunela, čime se garantuje da neće doći do prekida servisa u bilo kojoj situaciji kada do n linkova otkáže. Ukoliko ne postoji tunel(i) sa dovoljno slobodnog kapaciteta da pruži potrebne QoS garancije, korisnikov zahtjev se odbija.



Slika 4.2: Online algoritam rutiranja.

Važno je napomenuti da u ovoj fazi fokus nije na optimizaciji rutiranja, već na brznoj alokaciji resursa za virtuelni link. TE problem sa ograničenjima u pogledu propusnosti i kašnjenja ne može se riješiti u realnom vremenu. Naime, kako TE tehnike obično balansiraju opterećenje mreže izborom odgovarajućih težinskih faktora linkova, a težinski faktori i kašnjenje predstavljaju aditivne metrike rutiranja, problem izbora optimalne rute postaje *NP-complete* [24]. Diskretizacijom kašnjenja na linkovima problem se može uprostiti, međutim, efikasnost TE algoritma se smanjuje u tom slučaju. Optimizacija ruta za QoS3 saobraćaj predstavlja još kompleksniji zadatak, s obzirom da je potrebno dodatno garantovati robustnost na otkaz linkova.

4.1.2 Modul za *offline* TE

Modul za *offline* TE periodično izvršava optimizacioni algoritam koji teži da minimizuje MLU pri striktnim QoS ograničenjima. Ulazni parametri TE algoritma su:

1. Graf mreže;
2. Matrica saobraćaja T ;
3. Lista tunela između IO parova.

Mreža je modelovana kao usmjereni graf $G = (V, L)$ gdje je V skup mrežnih čvorova, a L skup linkova. Usmjereni graf formiran je modelovanjem *full-duplex* linkova u formi dva *simplex* linka istih karakteristika. Svakom linku $l \in L$ su pridružena dva parametra: kapacitet c_l i propagaciono kašnjenje l_{prop} . Pretpostavka je da propagaciono kašnjenje dominantno određuje ukupno kašnjenje na ruti. Ova aproksimacija može se smatrati prihvatljivom u ISP mrežama, gdje je kašnjenje u redovima čekanja relativno malo uslijed velikih kapaciteta linkova i efekta statističkog multipleksiranja agregiranih saobraćajnih tokova [118].

Saobraćajna matrica $T\{d_{uv}\}$ modelovana je kao $|N| \times |N|$ matrica, gdje je N set Open-Flow svičeva na periferiji mreže. Element matrice d_{uv} predstavlja agregirani saobraćajni zahtjev između izvorišnog sviča u i destinacionog sviča v . Za potrebe optimizacije, matrica T je dekomponovana na četiri matrice $T_{be}, T_{qos1}, T_{qos2}$ i T_{qos3} , koje odgovaraju različitim klasama saobraćaja. Saobraćajni zahtjev klase $C \in \{be, qos1, qos2, qos3\}$, $d_{uv}^{(C)}$, rutira se preko skupa tunela P_{uv} , koji se izračunava *a-priori*. Za svaki saobraćajni zahtjev $d_{uv}^{(C)}$, *offline* algoritam računa optimalne težinske faktore $x_{p_{uv}}^{(C)}$ za dijeljenje (balansiranje) saobraćaja preko tunela $p_{uv} \in P_{uv}$, tj. dio ulaznog opterećenja koji će se rutirati preko svakog od tunela $p_{uv} \in P_{uv}$. Pri tome, dodatni resursi se rezervišu u mreži kako bi se očuvale performanse QoS3 tokova sve dok je broj linkova koji su u kvaru manji od n (konfigurabilan parametar). Optimizacioni algoritam definisan je na sledeći način:

$$\min \quad z \tag{4.1a}$$

$$\text{s.t.} \quad \sum_{p_{uv}} x_{p_{uv}}^{(C)} = 1 \quad C \in \{be, qos1, qos2\}, \forall (u, v) \in N \tag{4.1b}$$

$$\sum_{p_{uv}} x_{p_{uv}}^{(qos3)} > 1 \quad \forall (u, v) \in N \tag{4.1c}$$

$$\sum_{\forall C} \sum_{(u,v) \in N} \sum_{p_{uv}} R_{p_{uv},l} \cdot x_{p_{uv}}^{(C)} \cdot d_{uv}^{(C)} \leq z \cdot c_l \quad (4.1d)$$

$$\sum_{\forall C \neq be} \sum_{(u,v) \in N} \sum_{p_{uv}} R_{p_{uv},l} \cdot x_{p_{uv}}^{(C)} \cdot d_{uv}^{(C)} \leq c_l \quad (4.1e)$$

$$y_{p_{uv}}^{(C)} \cdot \left(\sum_{l \in p_{uv}} l_{prop} - D_{max}^{(C)} \right) \leq 0 \quad \forall p_{uv}, C \quad (4.1f)$$

$$y_{p_{uv}}^{(C)} \geq x_{p_{uv}}^{(C)} \quad \forall C, (u, v) \in N, p_{uv} \quad (4.1g)$$

$$\sum_{p_{uv} \in P_{uv}^{\mu}} x_{p_{uv}}^{(qos3)} \geq 1 \quad \forall (u, v) \in N, \mu \in U_n \quad (4.1h)$$

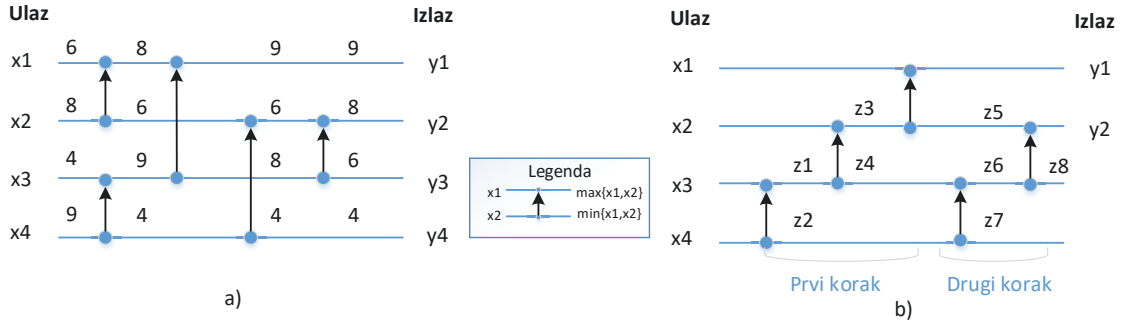
$$x_{p_{uv}}^{(C)} \geq 0 \quad \forall C, (u, v) \in N, p_{uv} \quad (4.1i)$$

$$y_{p_{uv}}^{(C)} \in \{0, 1\} \quad \forall C, (u, v) \in N, p_{uv} \quad (4.1j)$$

$$z \geq 0 \quad (4.1k)$$

Optimizacioni model vrši minimizaciju MLU varijable z koristeći ograničenja (4.1b)-(4.1k). Optimizacije varijable $x_{p_{uv}}^{(C)}$ definišu dio saobraćajnog zahtjeva $d_{uv}^{(C)}$ koji će se rutirati tunelom p_{uv} . Ograničenjima (4.1b),(4.1c) i (4.1h) garantuje se propusnost za sve QoS tokove matrice saobraćaja. Sa (4.1d) i (4.1e) definisana su ograničenja kapaciteta linka. Naime, opterećenje koje generiše QoS saobraćaj na linku mora biti manje od kapaciteta linka (4.1e). Sa druge strane, za *best-effort* tokove propusnost nije garantovana, pa z varijabla (tj. MLU) u optimizacionom modelu može uzimati vrijednost veću od 1 (4.1k). $R_{p_{uv},l}$ je binarna varijabla koja indicira da li link l pripada tunelu p_{uv} . Ukoliko pripada, $R_{p_{uv},l}$ je 1, u suprotnom 0. U (4.1f) uvedene su optimizacione varijable $y_{p_{uv}}^{(C)}$, koje služe za ograničavanje maksimalnog kašnjenja sa $D_{max}^{(C)}$. Ukoliko je kašnjenje na ruti p_{uv} veće od $D_{max}^{(C)}$, (4.1f) i (4.1g) osiguravaju da saobraćaj klase C neće biti rutiran preko p_{uv} , odnosno tada $y_{p_{uv}}^{(C)}$ mora biti 0 kako bi ograničenje (4.1g) bilo zadovoljeno. Za klase $C \in \{be, qos1\}$, pretpostavka je da $D_{max}^{(C)}$ uzima beskonačno veliku vrijednost.

Glavni uzrok velike računske kompleksnosti predstavljenog optimizacionog modela je ograničenje (4.1h), koje za QoS3 saobraćaj garantuje robustnost na otkaz do n linkova. Skup potencijalnih događaja otkaza linkova može se predstaviti vektorom $\mu = [\mu_l | l \in L]$, gdje μ_l uzima vrijednost 1 u slučaju otkaza linka l , ili 0 u suprotnom. Stoga, propusnost saobraćajnog toka neće biti ugrožena u slučaju kvara (do) n linkova, ukoliko se mreža nalazi u stanju iz skupa $U_n = \{\mu | \sum_l \mu_l \leq n\}$. U posmatranom stanju μ , za svaki saobraćajni zahtjev $d_{uv}^{(qos3)}$ moguće je odrediti set tunela P_{uv}^{μ} koji ne koriste ni jedan od linkova u kvaru. Ograničenje (4.1h) zahtijeva da ovi tuneli zbirno imaju najmanje $d_{uv}^{(qos3)}$ slobodnog kapaciteta. Jasno je da sa većim brojem alternativnih tunela ovaj zahtjev može biti lakše



Slika 4.3: Primjer sortirajuće mreže (a) i pronalaženja dva najveća elementa *bubble-sort* pristupom [3].

ispunjen. S obzirom da je u analiziranom scenariju za svaki IO par stepen preklapanja tunela na nivou linka ograničen sa q , za bilo koje stanje $\mu \in U_n$ broj alternativnih tunela $|P_{uv}^\mu|$ nije manji od $|P_{uv}| - n \cdot q$. Stoga, koristeći pristup predložen u [13], moguće je zamijeniti $|N| \cdot |N - 1| \cdot \sum_{k=1}^n \binom{|L|}{k}$ ograničenja iz (4.1h) sa:

$$\sum_{p_{uv} \in P_{uv}} x_{p_{uv}}^{(qos3)} - \sum_{k=1}^{n \cdot q} x_{uv}^{max_k} \geq 1 \quad \forall (u, v) \in N \quad (4.2)$$

gdje $x_{uv}^{max_k}$ označava k -ti po redu najveći element u $X = \{x_{p_{uv}}^{(qos3)} | p_{uv} \in P_{uv}\}$.

Da bi se optimizacioni model mogao efikasno izvršavati neophodno je uslov (4.2) formulisati u skup linearnih ograničenja. Za ove svrhe, pogodno je iskoristiti metod sortirajućih mreža [119], na način koji je predložen u [13]. Sortirajuće mreže mogu sortirati bilo koji niz od M vrijednosti jednostavnim operacijama poređenja elemenata niza i zamjene njihovih pozicija. Primjer sortirajuće mreže za 4 vrijednosti prikazan je na Slici 4.3a. Vertikalnim strelicama predstavljeni su komparatori koji vrše operacije poređenja i zamjene mjesta elemenata. Svaki komparator poredi dvije ulazne vrijednosti i pomjera veću od njih na visočiju poziciju u mreži, a manju vrijednost spušta. Jedna od bitnih osobina sortirajućih mreža je da je redosled komparatorskih operacija nezavisan od ulaza, što nije slučaj kod većine algoritama za sortiranje, gdje svako poređenje zavisi od ishoda prethodno završenog. Ova osobina omogućava kodiranje veze između originalnih varijabli i k -tog najvećeg elementa linearnim izrazima [13]. S obzirom da je za predloženi optimizacioni model od interesa samo M najvećih vrijednosti, a ne sortiranje svih varijabli, dovoljno je koristiti samo segment sortirajuće mreže. Dok kompletno sortiranje niza od K elemenata zahtijeva $\Theta(K \log^2 K)$ poređenja/zamjena, segment za određivanje M najvećih vrijednosti sadrži $\Theta(KM)$ operatora. Ova razlika je značajna imajući u vidu da je $M = n \cdot q$ znatno

manje od ukupnog broja tunela u razmatranom slučaju. Za sortiranje mreže iskorišćen je *bubble-sort* pristup iz [13], koji je ilustriran na Slici 4.3b, na primjeru za pronalaženje 2 najveće (od ukupno 4) vrijednosti. Pseudo-kod algoritma za generisanje LP izraza za M najvećih varijabli dat je Algoritmom 1. Algoritam se izvršava u M koraka, i u svakom koraku (pozivom Algoritma 2) kreira izraz za najveću od vrijednosti koje nisu sortirane u prethodnim koracima.

Algoritam 1 Pronalaženje M najvećih vrijednosti i kreiranje LP ograničenja

Ulaz: $X \rightarrow$ niz varijabli, $M \rightarrow$ broj najvećih varijabli koje treba pronaći;

Izlaz: $Y \rightarrow$ niz novih varijabli kod kojeg je $Y[i]$ ($0 \leq i \leq M$) i -ti najveći element niza X ;
 $C \rightarrow$ LP ograničenja

```

1:  $Y = \{\}, C = \{\}$ 
2: while  $|Y| < M$  do:
3:    $y^*, X, C' = \text{računanjeMax}(X)$ ;
4:    $Y = Y + \{y^*\}$ ;
5:    $C = C + \{C'\}$ ;
6: return  $Y, C$ 

```

Algoritam 2 $\text{računanjeMax}(X)$

Ulaz: $X \rightarrow$ niz varijabli;

Izlaz: $x^* \rightarrow$ varijabla koja odgovara najvećem elementu niza X ; $Y \rightarrow$ niz koji predstavlja
 X bez x^* ; $C \rightarrow$ set ograničenja koja povezuju X, Y i x^* ;

Komentar: $\text{pop}()$ funkcija uklanja prvi element niza X i vraća ga kao rezultat;

```

1:  $x^* = X.\text{pop}()$ ,  $Y = \{\}, C = \{\}$ ;
2: while  $|X| \neq \{\}$  do:
3:    $x = X.\text{pop}()$ ;
4:    $x_{\max}, x_{\min} \leftarrow$  nove varijable
5:   #kreiranje dva nova ograničenja
6:    $c_1 \leftarrow 2x_{\max} = x + x^* + |x - x^*|$ ;
7:    $c_1 \leftarrow 2x_{\min} = x + x^* - |x - x^*|$ ;
8:    $x^* \leftarrow x_{\max}$ ;  $Y \leftarrow Y + \{x_{\min}\}$ ;  $C \leftarrow C + \{c_1, c_2\}$ ;
9: return  $x^*, Y, C$ 

```

Izrazi sa apsolutnim vrijednostima u Algoritmu 2 (linije 6-7) čine optimizacioni problem nelinearnim i veoma teškim za rešavanje standardnim optimizacionim procedurama. Međutim, ukoliko je apsolutna vrijednost primijenjena nad linearnom funkcijom, ista se

može preformulisati sa dva linearna izraza, pod pretpostavkom da se koristi *simplex* metod [120] za rešavanje optimizacionog problema. Drugim riječima, $z = |f(x)|$ se u optimizacionom modelu može predstaviti sa dva linearna izraza ukoliko je $f(x)$ linearna funkcija sama po sebi. Pri tome, potrebno je uvođenje nove optimizacione varijable z' , za koju važe sledeća ograničenja:

$$f(x) \leq z' \quad (4.3a)$$

$$-f(x) \leq z' \quad (4.3b)$$

Stoga, uslov (4.2) u predloženom optimizacionom modelu formulisan je sledećim LP izrazima za $\forall d_{uv}^{(qos3)} \in T_{qos3}, (u, v) \in N$:

$$\sum_{p_{uv} \in P_{uv}} x_{p_{uv}}^{(qos3)} - \sum_{k=1}^{n \cdot q} q_{uv}^{maxk} (|P_{uv}| - k) \geq 1 \quad (4.4a)$$

$$q_{uv}^{min_0}(p_{id}) = x_p^{(qos3)} \quad \forall p_{id} \in \{1..|P_{uv}|\} \quad (4.4b)$$

$$q_{uv}^{maxk}(0) = q_{uv}^{min_{k-1}}(1) \quad \forall k \in \{1..M\} \quad (4.4c)$$

$$q_{uv}^{maxk}(p_{id}) \geq 0 \quad \forall k \in \{0..M\}, p_{id} \in \{0..|P_{uv}|\} \quad (4.4d)$$

$$q_{uv}^{min_k}(p_{id}) \geq 0 \quad \forall k \in \{0..M\}, p_{id} \in \{0..|P_{uv}|\} \quad (4.4e)$$

$$\forall k \in \{1..M\}, p_{id} \in \{1..|P_{uv}| - k\} :$$

$$2q_{uv}^{maxk}(p_{id}) = q_{uv}^{min_{k-1}}(p_{id} + 1) + q_{uv}^{maxk}(p_{id} - 1) + h_{uv}^{maxk}(p_{id}) \quad (4.4f)$$

$$2q_{uv}^{min_k}(p_{id}) = q_{uv}^{min_{k-1}}(p_{id} + 1) + q_{uv}^{maxk}(p_{id} - 1) - h_{uv}^{maxk}(p_{id}) \quad (4.4g)$$

$$-h_{uv}^{maxk}(p_{id}) \leq q_{uv}^{min_{k-1}}(p_{id} + 1) - q_{uv}^{maxk}(p_{id} - 1) \leq h_{uv}^{maxk}(p_{id}) \quad (4.4h)$$

$$h_{uv}^{maxk}(p_{id}) \geq 0 \quad (4.4i)$$

Za potrebe kodiranja uslova (4.2) uvedeni su nizovi novih varijabli: q_{uv}^{maxk} , $q_{uv}^{min_k}$ i h_{uv}^{maxk} . Sa p_{id} obilježen je redni broj tunela $p \in P_{uv}$. Nakon izvršavanja algoritma, niz optimizacionih varijabli q_{uv}^{maxk} sadrži informacije o M najvećih elemenata iz $X = \{x_{p_{uv}}^{(qos3)} | p_{uv} \in P_{uv}\}$. Preciznije, k -ti najveći element ekvivalentan je varijabli $q_{uv}^{maxk}(|P_{uv}| - k)$, tj. nalazi se na poziciji $|P_{uv}| - k$ u nizu q_{uv}^{maxk} . Pomoćne varijable $q_{uv}^{min_k}$ definišu manju vrijednost prilikom svake operacije poređenja u sortirajućoj mreži. Varijable h_{uv}^{maxk} omogućavaju kodiranje apsolutnih vrijednosti iz Algoritma 2 u vidu linearnih ograničenja, na način koji je prethodno objašnjen.

4.1.3 Alternativne definicije optimizacione funkcije

Ključni nedostatak MLU optimizacione funkcije je činjenica da na njenu vrijednost utiče samo najopterećeniji link u mreži. Stanja ostalih linkova ne utiču na optimalno rešenje. Na primjer, scenario u kojem su svi linkovi iskorišćeni 90% ima istu vrijednost optimizacione funkcije kao scenario u kojem je samo jedan link iskorišćen 90% dok su ostali slabo opterećeni. U praksi, naravno, drugi scenario je poželjniji. Stoga, u ovom poglavlju razmatrane su još dvije formulacije optimizacione funkcije za *offline* TE modul. Prva, za koju je usvojena notacija *MaxMinFlow*, teži da maksimizuje dostupni kapacitet za „najugroženije” IO parove. Definicija *MaxMinFlow* modela data je sa 4.5.

$$\max \quad \lambda \quad (4.5a)$$

$$\text{s.t.} \quad \sum_{\forall C \neq be} \sum_{(u,v) \in N} \sum_{p_{uv}} R_{p_{uv},l} \cdot x_{p_{uv}}^{(C)} \cdot d_{uv}^{(C)} \leq c_l \quad \forall l \in L \quad (4.5b)$$

$$\sum_{p_{uv}} (cap(p_{uv}) - bot(p_{uv})) \geq \lambda \quad \forall (u,v) \in N \quad (4.5c)$$

$$bot(p_{uv}) = \max_{l \in L} R_{p_{uv},l} \cdot \sum_{\forall C \neq be} \sum_{(s,t) \in N} \sum_{p_{st}} R_{p_{st},l} \cdot x_{p_{st}}^{(C)} \cdot d_{st}^{(C)} \quad (4.5d)$$

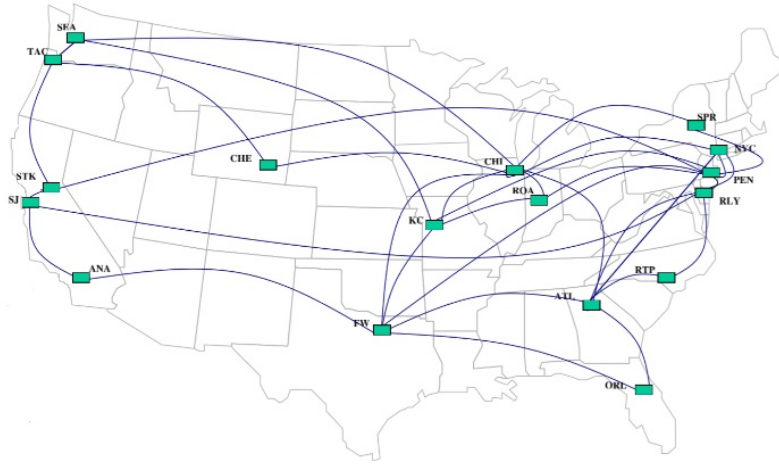
$$\lambda \geq 0 \quad (4.5e)$$

uključujući ograničenja (4.1b), (4.1c), (4.1f)-(4.1k). U optimizacionom modelu 4.5, $cap(p_{uv})$ i $bot(p_{uv})$ predstavljaju kapacitet tunela p_{uv} i opterećenje *bottleneck* linka istog tunela, respektivno.

Druga razmatrana alternativa za *offline* TE optimizacionu funkciju u nastavku poglavlja označena je sa *MaxSum*. Definisana je na sličan način kao *MaxMinFlow*, ali ne potencira ravnopravnost IO parova. Umjesto toga, fokusirana je na minimizaciju ukupnog dostupnog kapaciteta za sve IO parove zajedno. *MaxSum* optimizaciona funkcija definisana je na sledeći način:

$$\max \quad \sum_{(u,v) \in N} \sum_{p_{uv}} (cap(p_{uv}) - bot(p_{uv})) \quad (4.6a)$$

pri čemu se (4.5c) i (4.5e) ne koriste kao ograničenja.



Slika 4.4: Sprint ISP topologija.

4.2 Simulaciona metodologija

Za potrebe evaluacije predloženog TE pristupa razvijen je simulator u *Python* programskom jeziku, koji koristi CPLEX [121] za rešavanje optimizacionih problema. U simulacijama korišćena je Sprint topologija sa Slike 4.4. PoP lokacije su mapirane sa mrežnim čvorovima. Kapacitet svakog linka izražen je u apstraktnim jedinicama propusnog opsega, i podešen na 1000. Kašnjenje na linkovima izračunato je dijeljenjem „velike kružne udaljenosti” između povezanih PoP čvorišta i brzine prostiranja svjetlosti u optičkom vlaknu [95].

S obzirom da noviji trejsovi saobraćaja za ISP mreže nisu javno dostupni, u simulacijama je korišćen *gravity* model [122] za generisanje prosječne matrice saobraćaja tokom časa najvećeg opterećenja u toku dana. *Gravity* model pretpostavlja proporcionalan odnos između saobraćaja između IO para (u, v) - d_{uv} , ukupnog saobraćaja koji ulazi u mrežu preko čvora u - $d_{en}(u)$ i ukupnog saobraćaja koji napušta mrežu preko čvora v - $d_{ex}(v)$. Saobraćajna potražnja između čvorova u i v računata je po sledećem obrascu:

$$d_{uv} = \frac{d_{ex}(v) \cdot d_{en}(u)}{\sum_{v \in V} d_{ex}(v)} \quad (4.7)$$

U prethodnom izrazu $d_{en}(u)$ dobijeno je sumiranjem kapaciteta svih izlaznih linkova čvora u , a $d_{ex}(v)$ sumiranjem kapaciteta svih izlaznih linkova čvora v . S obzirom da ovako kreirana *gravity* saobraćajna matrica prevazilazi mrežne kapacitete, naknadno je izvršeno njeno skaliranje. Prvo, matrica je izdijeljena na 100 elementarnih matrica identične di-

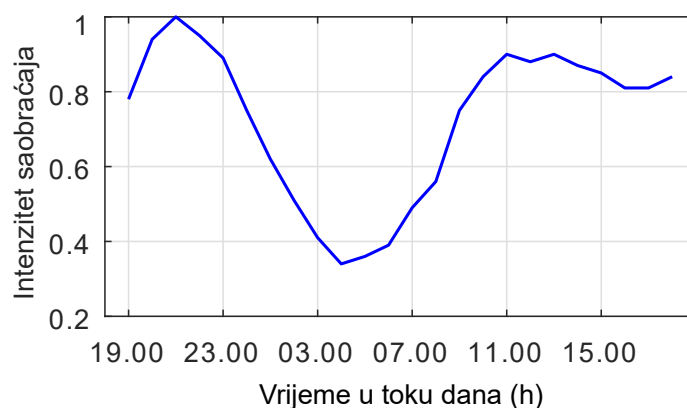
menzije. Zatim, CSPF algoritam je korišćen iterativno za rutiranje elementarnih matrica sve dok neki od linkova u mreži ne postane iskorišćen preko 70%. Konačno, matrica saobraćaja za čas najvećeg opterećenja dobijena je sumiranjem rutiranih elementarnih matrica. Na osnovu ove matrice simulirana je dinamika dolazaka korisničkih zahtjeva. Trajanje tokova modelovano je eksponencijalnom raspodjelom srednje vrijednosti 20 minuta, ukoliko nije drugačije naznačeno, dok je Poasonova raspodjela korišćena za generisanje BoD zahtjeva između IO parova. Srednja brzina Poasonovih dolazaka računata je tako da rezultujući model saobraćaja u prosjeku konvergira „skaliranoj” *gravity* matrici saobraćaja. Prosječan broj aktivnih tokova između IO parova može se posmatrati kao broj elemenata u virtuelnom $M/M/\infty$ redu čekanja [123]. U tom slučaju, prosječan broj aktivnih tokova između čvorova u i v može se dobiti kao:

$$N_{uv} = \frac{d_{uv}}{B} \quad (4.8)$$

gdje B predstavlja prosječnu vrijednost pojedinačnog QoS zahtjeva u pogledu propusnosti. S obzirom da je srednja zauzetost $M/M/\infty$ reda čekanja jednaka intenzitetu saobraćaja, srednja brzina Poasonovih dolazaka između IO parova (tokom vremena t) izvedena je kao:

$$\lambda_{uv} = \frac{N_{uv}}{t} \quad (4.9)$$

Zahtjevi u pogledu propusnosti za svaki generisani tok birani su iz seta od [3, 5, 7] jedinica, sa jednakom vjerovatnoćom. Pretpostavljeno je da svi *best-effort* tokovi dijele jedan rezervisani dio kapaciteta mreže, bez QoS garancija. Stoga, *best-effort* zahtjevi nisu razmatrani u daljoj analizi, već su generisani tokovi uvijek pridruženi jednoj od QoS klasa saobraćaja. Za QoS2 klasu, ograničenje kašnjenja postavljeno je na 35ms, što odgovara kašnjenju na najkraćoj ruti između najudaljenijeg IO para u mreži. Maksimalno dozvoljeno kašnjenje za QoS3 klasu podešeno je na 100ms. Između svakog IO para instalirano je maksimalno 15 tunela (tj. $K = 15$), dok je parametar preklapanja ruta q podešen na 3. Pretpostavljeno je da QoS3 klasa BoD servisa garantuje robustnost na otkaz jednog linka ($n = 1$). Trajanje simulacija podešeno je na jedan dan, a srednja dolazna brzina zahtjeva tokom tog vremena skalirana je u skladu sa sabračajnim obrascem sa Slike 4.5. Interval između uzastopnih izvršavanja *offline* TE modula za predloženi pristup podešen je na 20 minuta.



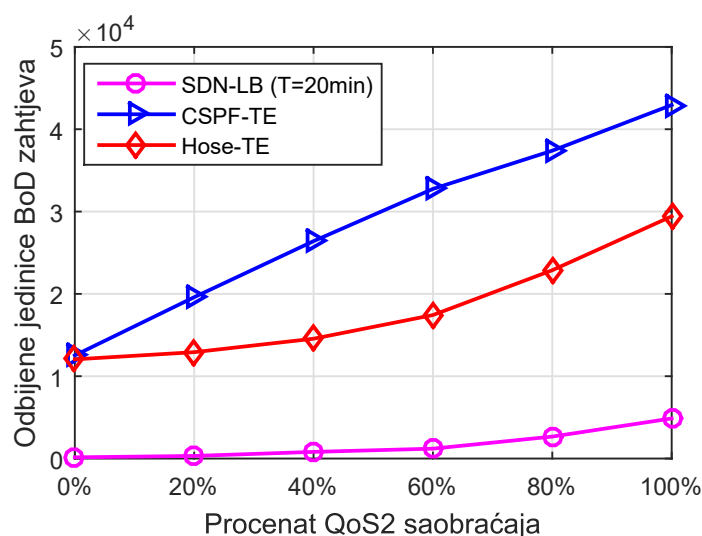
Slika 4.5: Promjena intenziteta saobraćaja u toku dana.

4.3 Simulacioni rezultati

Rezultati simulacione analize organizovani su u dvije cjeline. U prvom dijelu sekcije predstavljeni su rezultati poređenja predloženog pristupa za alokaciju virtuelnih linkova sa konvencionalnim TE pristupima: (i) CSPF TE i (ii) *oblivious* TE. Nakon toga, izvršeno je poređenje sa rešenjima koja koriste fleksibilnost SDN rutiranja u jezgru mreže.

4.3.1 Poređenje sa konvencionalnim TE rešenjima

Predloženi pristup, za koji će u nastavku biti usvojena notacija SDN-LB, u ovoj sekciji upoređen je sa dva konvencionalna TE pristupa u ISP mrežama. Prvi je baziran na CSPF rutiranju, dok je drugi baziran na *hose* modelu saobraćaja. Pod CSPF pristupom, podrazumijeva se TE bez centralizovanih optimizacija, gdje se CSPF algoritam rutiranja koristi za pronalaženje optimalnog tunela za svaki BoD zahtjev, tj. virtuelni link. Robustnost na otkaz linkova u mreži garantuje se rezervacijom potrebnog kapaciteta na više međusobno nepreklapajućih ruta. Drugi analizirani pristup podrazumijeva *oblivious* rutiranje na osnovu *hose* modela saobraćaja (Hose-TE). Hose-TE pristup je razmatran kao predstavnik optimizovanog, ali statičkog inženjeringa saobraćaja. Naime, dok SDN TE rešenja prilagođavaju šemu rutiranja promjenjivim saobraćajnim uslovima, Hose-TE izbjegava dinamičke rekonfiguracije mreže radi stabilnosti mreže i jednostavnosti upravljanja. Kao što je objašnjeno u Sekciji 2, Hose-TE određuje optimalnu konfiguraciju ruta samo na osnovu informacija o maksimalnom saobraćaju koji može ući i izaći iz mreže preko svakog od njenih čvorišta. U analizi je pretpostavljeno da su pomenuti parametri za svaki

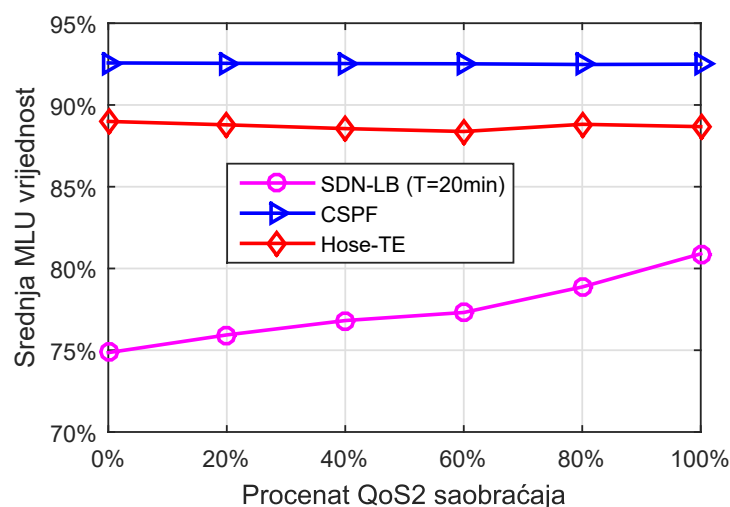


Slika 4.6: Broj odbijenih jedinica BoD zahtjeva za SDN-LB, CSPF-TE i Hose-TE.

od čvorova određeni ukupnim kapacitetom ulaznih i izlaznih linkova tog čvora [97]. S obzirom da je standardni Hose-TE pristup dizajniran za *best-effort* tip servisa, za potrebe analize neophodno je bilo uvesti određene modifikacije. Naime, u simulacijama je najprije korišćen LP metod iz [97] za rešavanje kompleksnog optimizacionog modela. Zatim, na osnovu dobijenih vrijednosti varijabli, određen je set ruta u optimalnoj šemi rutiranja za *hose* model saobraćaja. Takođe, svakoj od ovih ruta dodijeljen je težinski faktor proporcionalan količini saobraćaja koja se njom prenosi. Za svaki BoD zahtjev, analizirani Hose-TE pristup bira jednu od ruta koja može pružiti potrebni QoS. Pri tome, ukoliko više odgovarajućih ruta postoji, sa najvećom vjerovatnoćom bira se ona sa najvećim težinskim faktorom.

Poređenje SDN-LB, CSPF-TE i Hose-TE pristupa izvršeno je za scenario bez QoS3 saobraćaja (Slike 4.6-4.7) iz razloga što za Hose-TE pristup nije poznata odgovarajuća ekstenzija koja može garantovati robustnost na otkaz linkova. Kao indikatori performansi usvojeni su: (i) broj odbijenih jedinica BoD zahtjeva, i (ii) srednja vrijednost MLU.

Kao što se može primijetiti sa Slike 4.6, SDN-LB značajno efikasnije koristi mrežne resurse od CSPF-TE i Hose-TE pristupa, koji se u ovoj analizi smatraju konkurentskim. Kod CSPF-TE pristupa, za svaki BoD zahtjev bira se najkraća ruta koja može da pruži potrebni QoS. Stoga, uglavnom se forsiraju rute sa najmanjim kašnjenjem, što se može smatrati optimalnim izborom za DS (*Delay Sensitive*) aplikacije, ali je veoma je neefikasno iz perspektive balansiranja opterećenja jer linkovi koji su kritični za veliki broj parova vrlo brzo postaju zagušeni. Riječ je o linkovima sa malim propagacionim kašnjenjem.



Slika 4.7: Srednja MLU vrijednost za SDN-LB, CSPF-TE i Hose-TE.

njem, koji su obično dio ruta sa najmanjim kašnjenjem. Kao posledica nebalansiranog opterećenja linkova dolazi do odbijanja velikog broj BoD zahtjeva (Slika 4.7). Poboljšanje u performansama koje SDN-LB ostvaraje u odnosu na Hose-TE takođe je evidentno sa Slika 4.6-4.7, što indicira jasnu potrebu za dinamičkim optimizacijama saobraćaja u ISP mrežama.

4.3.2 Poređenje sa SDN TE rešenjima

S obzirom da u vrijeme pisanja ove disertacije u literaturi nisu poznata druga SDN TE rešenja koja se mogu direktno primijeniti za mapiranje virtuelnih linkova u ISP mrežama, u ovoj sekciji definisana su još dva rešenja koja će poslužiti za analizu uticaja računskih olakšica u predloženom optimizacionom modelu. Ova dva rešenja su optimalnija od predloženog u pogledu ukupne mrežne propusnosti, ali zahtijevaju implementaciju SDN infrastrukture u svim segmentima mreže, tj. i u jezgu i na periferiji mreže. Prvo rešenje, koje će biti notirano sa SDN-OPT, maksimizuje iskorišćenost mrežnih resursa izvršavanjem QoS-ograničenog MCF optimizacionog problema u trenucima kada BoD zahtjev ne može biti prihvaćen od strane modula za *online* rutiranje. Preciznije, SDN-OPT MCF problem formulisan je na sledeći način:

$$\min \quad z + \varepsilon \cdot \sum_{l \in L} \sum_{(u,v) \in N} (a_{uv,l}^{(qos3)} + \sum_{\forall C} f_{uv,l}^{(C)}) \quad (4.10a)$$

$$\text{s.t.} \quad \sum_{l \in L^+(k)} f_{uv,l}^{(C)} - \sum_{l \in L^-(k)} f_{uv,l}^{(C)} = \begin{cases} +1 & ,if \ k=u \\ -1 & ,if \ k=v \\ 0 & ,else \end{cases} \quad \forall (u,v,k) \in N, \forall C \quad (4.10b)$$

$$\sum_{l \in L^+(k)} a_{uv,l}^{(qos3)} - \sum_{l \in L^-(k)} a_{uv,l}^{(qos3)} = \begin{cases} +1 & ,if \ k=u \\ -1 & ,if \ k=v \\ 0 & ,else \end{cases} \quad (4.10c)$$

$$\sum_{(u,v) \in N} (d_{uv}^{(qos3)} \cdot a_{uv,l}^{(qos3)} + \sum_{\forall C} f_{uv,l}^{(C)} \cdot d_{uv}^{(C)}) \leq z \cdot c_l \quad \forall l \in L \quad (4.10d)$$

$$y f_{uv,l}^{(C)} \geq f_{uv,l}^{(C)} \quad \forall l \in L, \forall (u,v) \in N, \forall C \quad (4.10e)$$

$$y a_{uv,l}^{(qos3)} \geq a_{uv,l}^{(qos3)} \quad \forall l \in L, \forall (u,v) \in N \quad (4.10f)$$

$$y f_{uv,l^+}^{(qos3)} + y f_{uv,l^-}^{(qos3)} + y a_{uv,l^+}^{(qos3)} + y a_{uv,l^-}^{(qos3)} \leq 1 \quad \forall l \in L \quad (4.10g)$$

$$\sum_{l \in L} y f_{uv,l}^{(C)} \cdot l_{prop} \leq D_{max}^{(C)} \quad \forall (u,v) \in N, C \quad (4.10h)$$

$$\sum_{l \in L} a f_{uv,l}^{(qos3)} \cdot l_{prop} \leq D_{max}^{(qos3)} \quad \forall (u,v) \in N \quad (4.10i)$$

$$0 \leq f_{uv,l}^{(C)} \leq 1 \quad \forall l \in L, \forall (u,v) \in N, \forall C \quad (4.10j)$$

$$0 \leq a_{uv,l}^{(qos3)} \leq 1 \quad \forall l \in L, \forall (u,v) \in N \quad (4.10k)$$

$$y f_{uv,l}^{(C)} \in \{0, 1\} \quad \forall l \in L, \forall (u,v) \in N, \forall C \quad (4.10l)$$

$$y a_{uv,l}^{(qos3)} \in \{0, 1\} \quad \forall l \in L, \forall (u,v) \in N \quad (4.10m)$$

$$0 \leq z \leq 1 \quad (4.10n)$$

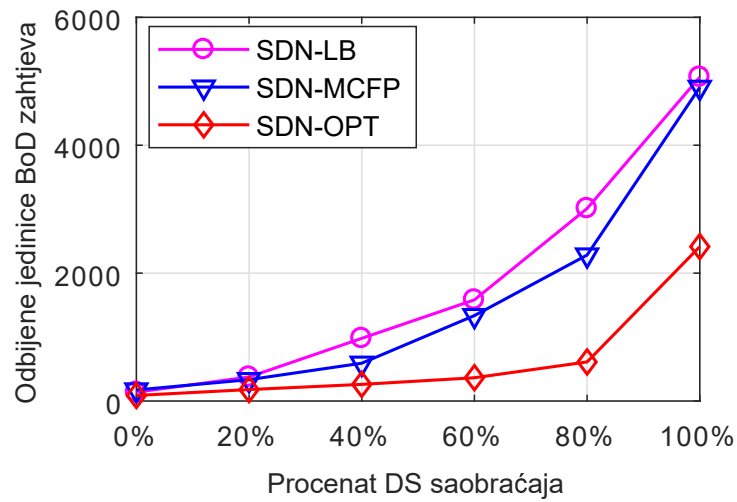
U gornjoj formulaciji varijabla z odgovara MLU vrijednosti kao i prije, $f_{uv,l}^{(C)}$ definiše dio saobraćajnog zahtjeva $d_{uv}^{(C)}$ koji se rutira preko linka l , dok $a_{uv,l}^{(qos3)}$ definiše dio $d_{uv}^{(qos3)}$ koji će se rutirati linkom l u slučaju ispada nekog od linkova na primarnoj ruti. Ukoliko se usvoji notacija $L^-(k)$ za ulazne linkove čvora k , i $L^+(k)$ za izlazne linkove k , (4.10b) predstavlja ograničenje konzervacije toka za primarne rute, dok (4.10c) definiše ograničenje konzervacije toka za *backup* rute koje se dodjeljuju QoS3 zahtjevima. Minimizacija MLU usvojena je kao glavni optimizacioni kriterijum. Drugi sabirak u (4.10a) je korišćen radi prevencije petlji rutiranja. Njegov uticaj je kontrolisan sa ε parametrom, kojem je dodijeljena veoma mala vrijednost. Ograničenje kapaciteta linka dato je sa (4.10d). Bi-

narne varijable $y_{uv,l}^{(C)}$ indiciraju da li agregirani tok $d_{uv}^{(C)}$ koristi link l . Ukoliko koristi, vrijednost ove varijable je 1, u suprotnom 0. Na sličan način, u (4.10f) uvedene su varijable $y_{uv,l}^{(qos3)}$ koje indiciraju da li link l pripada *backup* rutama, koje su predviđene za prenos saobraćaja $d_{uv}^{(qos3)}$ u slučaju kvara na primarnoj ruti. Robustnost na otkaz linkova za QoS3 saobraćaj je garantovana definisanjem uslova za nepreklapanje primarnih i *backup* ruta. S obzirom da je u simulacijama svaki *duplex* link modelovan sa dva usmjerena linka suprotne orijentacije (l i l^-), da bi uslov nepreklapanja primarnih i *backup* ruta bio ispunjen neophodno je dodati ograničenje (4.10g) u definiciju optimizacionog problema. Kašnjenje na odabranim rutama kontrolisano je ograničenjima (4.10h) i (4.10i).

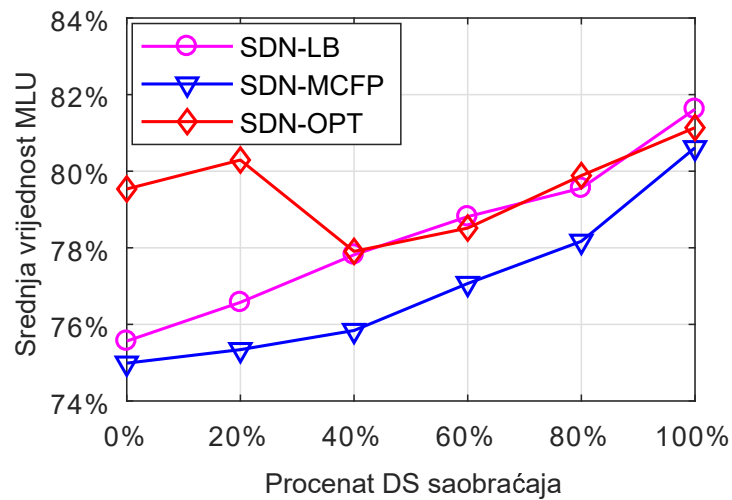
Da bi se izbjegle nepotrebne rekonfiguracije mreže (npr. pri slabom intenzitetu saobraćaja), SDN-OPT koristi jednostavan *online* algoritam rutiranja koji se izvršava u tri koraka. Prvo, algoritam provjerava da li neka od postojećih ruta može pružiti zahtijevani kvalitet servisa. Ukoliko to nije slučaj, u drugom koraku primjenjuje se CSPF algoritam za pronalaženje nove rute. Ukoliko ni CSPF ne daje odgovarajući rezultat, izvršava se MCF optimizacioni model 4.10. Dakle, optimizacije se izvršavaju reaktivno, svaki put kada *online* algoritam ne može da pronađe odgovarajuće rešenje. S obzirom da se SDN-OPT karakteriše velikom računskom kompleksnošću, u ovoj sekciji analiziran je još jedan TE pristup: SDN-MCFP, koji se razlikuje od SDN-OPT samo po tome što se MCF optimizacije izvršavaju periodično umjesto reaktivno.

Simulaciona analiza podijeljena je na dva dijela, koja odgovaraju različitim saobraćajnim scenarijima. U prvom scenariju generisani su samo QoS1 i QoS2 BoD zahtjevi (Scenario I). Dobijeni rezultati u pogledu odbijenih BoD jedinica i srednje MLU vrijednosti za ovaj scenario prikazni su na Slikama 4.8 i 4.9. U drugom simulacionom scenariju generisane su sve tri klase BoD zahtjeva (Scenario II). Odgovarajući rezultati prikazani su odvojeno na Slikama 4.10 i 4.11, kako bi se mogao jasnije uočiti uticaj različitih dijelova predložene TE šeme na cjelokupne performanse. Sa Slike 4.8 moguće je uočiti da se razlika u performansama između SDN-LB i SDN-MCFP smanjuje sa porastom QoS2 saobraćaja. To se može pripisati činjenici da u mreži ne postoji veliki broj ruta koje mogu ispuniti stroge zahtjeve QoS2 klase u pogledu kašnjenja. Stoga, iako SDN-OPT i SDN-MCFP mogu da koriste proizvoljne rute i mijenjaju ih dinamički, SDN-LB ne zaostaje previše u pogledu diverziteta ruta. U simulacijama gdje je udio QoS2 saobraćaja bio 100%, SDN-LB odbija dva puta više zahtjeva od SDN-OPT, ali u odnosu na SDN-MCFP razlika u performansama je skoro zanemarljiva.

Kada su u pitanju MLU rezultati sa Slike 4.9, primijetne su velike vrijednosti za SDN-



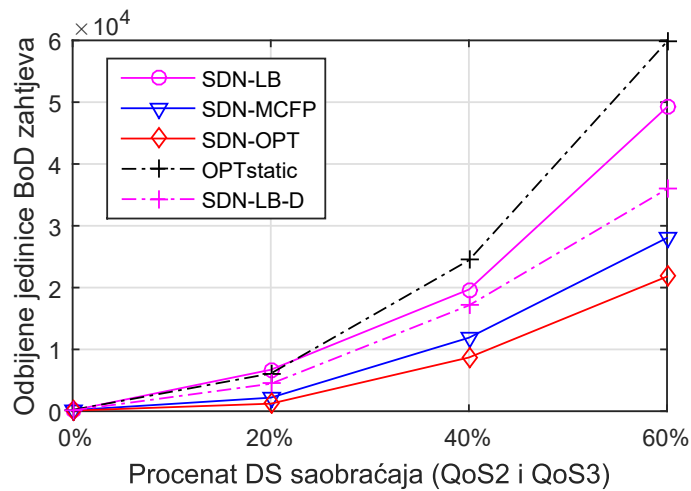
Slika 4.8: Odbijene BoD jedinice u simulacionom scenariju sa QoS1 i QoS2 zahtjevima.



Slika 4.9: Prosječna MLU vrijednost u simulacionom scenariju sa QoS1 i QoS2 zahtjevima.

OPT pri malom procentu QoS2 zahtjeva. Razlog je taj što se alokacija virtuelnih linkova u ovom slučaju uspješno vrši sa *online* modulom za rutiranje, pa se MCF optimizacije rijetko izvršavaju.

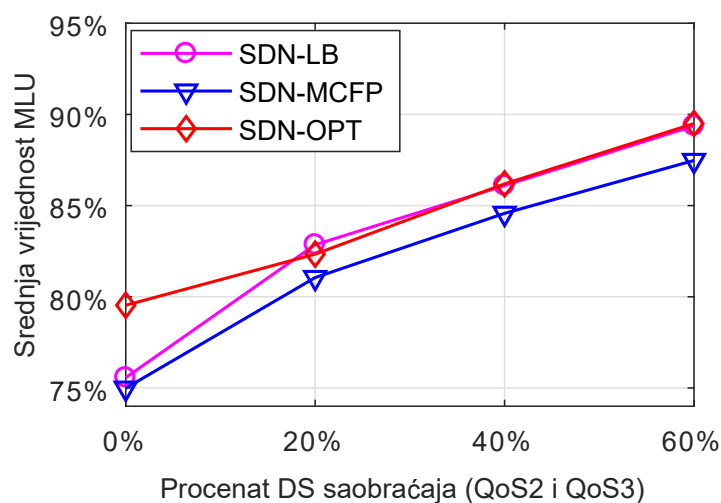
U drugom simulacionom scenariju generisana su u istim proporcijama dva tipa DS BoD zahtjeva: QoS2 i QoS3 zahtjevi. Sa Slike 4.10 može se primijetiti da procenat nealociranih virtuelnih linkova značajno raste sa uvođenjem QoS3 klase saobraćaja. U simulacijama, za ovu klasu nije definisano strogo ograničenje kašnjenja ($D_{max}^{(qos3)} = 100ms$), pa se



Slika 4.10: Odbijene BoD jedinice u drugom simulacionom scenariju.

prilikom alokacije QoS3 virtuelnih linkova na dužim rutama troši velika količina resursa. Performanse SDN-LB u pogledu odbijenih zahtjeva degradiraju se kao rezultat računske olakšice koja je uvedena sa (4.2). Pojednostavljenje optimizacionog modela uzrokovalo je alokaciju veće količine redundantnih resursa za QoS3 tokove u situacijama kada je $q > 1$. Sa druge strane, SDN-OPT i SDN-MCFP za svaki QoS3 zahtjev koriste jednu primarnu i jednu alternativnu rutu. Ove rute su međusobno nepreklapajuće, pa je dovoljno rezervisati zahtijevani kapacitet na svakoj od njih. Međutim, važno je primijetiti da SDN-LB doprinosi značajno uštedi resursa kada su tuneli između IO parova nepreklapajući (tj. $q = 1$). Naime, za $q = 1$ razlika u performansama SDN-LB i SDN-MCFP/SDN-OPT se zapravo smanjuje, pod pretpostavkom da u mreži postoji dovoljan broj nepreklapajućih tunela. U cilju dokazivanja ove tvrdnje, izvršene su simulacije za slučaj kada SDN-LB za QoS3 saobraćaj koristi samo nepreklapajuće tunele između IO parova. Dobijeni rezultati su prikazani na Slici 4.10, i označeni sa SDN-LB-D. Sa Slike 4.10 se može primijetiti da SDN-LB-D odbija manji broj BoD zahtjeva od standardnog SDN-LB pristupa, uprkos činjenici da je u simulacijama korišćena PoP topologija mreže, gdje je broj nepreklapajućih ruta relativno mali. U praksi, broj nepreklapajućih ruta je znatno veći, pa se više resursa može staviti na raspolaganje QoS3 saobraćaju.

Na Slici 4.10 takođe su prikazani rezultati dobijeni algoritmom za statičko balansiranje opterećenja između tunela, pri čemu su težinski faktori za balansiranje optimizovani za čas najvećeg opterećenja, pod pretpostavkom da je matrica saobraćaja unaprijed poznata (OPTstatic). Predstavljeni rezultati pokazuju da obje verzije SDN-LB pristupa ostvaruju



Slika 4.11: Prosječna MLU vrijednost u drugom simulacionom scenariju.

značajno bolje performanse od statičkih TE rešenja koja su uobičajena u današnjim ISP mrežama.

U simulacijama, SDN-LB, SDN-MCFP i SDN-OPT pristupi upoređeni su i u pogledu broja rekonfiguracija, kontrolnog *overhead*-a i prosječnog broja zapisa u tabelama tokova. Kao mjera kontrolnog *overhead*-a usvojen je broj *Flow-mod* OpenFlow poruka koje generiše SDN kontroler. Dobijeni rezultati su sumirani u Tabeli 4.1 za Scenario I, i u Tabeli 4.2 za Scenario II. Rezultati očigledno ukazuju na nepraktičnost SDN-OPT pristupa, uslijed ogromnog broja rekonfiguracija mrežnih parametara u uslovima velikog intenziteta saobraćaja. Ovo je problematično imajući u vidu da rekonfiguracije u razmatranom scenariju podrazumijevaju simultano ažuriranje težinskih faktora za balansiranje opterećenja na svim svičevima na ivici mreže. Kako se u opštem slučaju ne može garantovati da će se ažuriranja desiti na svim uređajima u isto vrijeme, ovo predstavlja veliki problem za stabilnost mreže. SDN-LB i SDN-MCFP obavljaju optimizacije periodično, u jednakim intervalima od 20 minuta, pa je broj rekonfiguracija u njihovom slučaju deterministički. Sa druge strane, kontrolni *overhead* za SDN-LB je 2-3 puta manji nego za SDN-MCFP. Ovi rezultati ukazuju na prednost SDN-LB pristupa u pogledu skalabilnosti. Naime, manja količina kontrolnog saobraćaja indicira da se jedan kontroler može koristiti za upravljanje većim mrežnim particijama. Interesantno je primijetiti da je u SDN-LB simulacijama broj zapisa u tabelama tokova veći nego u simulacijama druga dva pristupa. Međutim, ovo je velikim dijelom posledica korišćenja PoP topologije, gdje je na svakom čvoru konfigurisano do K statičkih tunela, za svaki od ostalih čvorova u mreži i za svaku

Tabela 4.1: Analiza skalabilnosti/kompleksnosti različitih SDN-TE pristupa - Scenarij I. Broj rekonfiguracija (Rekonfig), broj generisanih kontrolnih poruka (Flow-mod) i broj zapisa u tabelama tokova (FT zapisi).

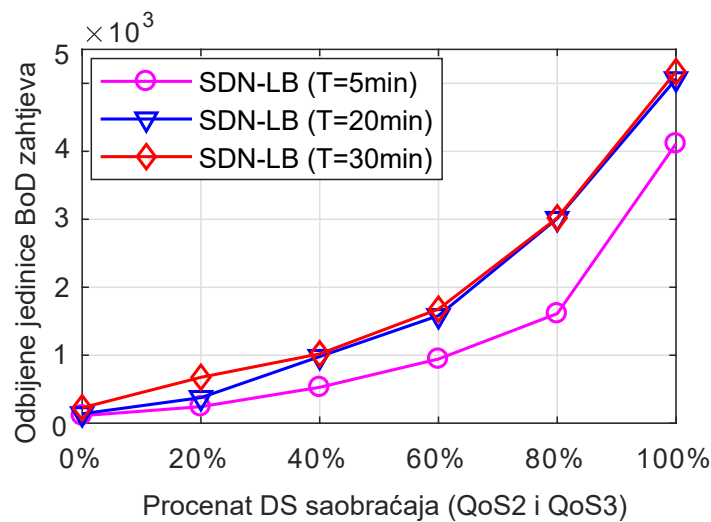
	QoS2 %	Rekonf	Flow-mod	FT zapisi
SDN-LB	0%	72	69354	9470
SDN-MCFP	0%	72	173560	2257
SDN-OPT	0%	16	51428	2394
SDN-LB	20%	72	81193	9525
SDN-MCFP	20%	72	216900	2768
SDN-OPT	20%	57	177336	3039
SDN-LB	40%	72	78033	9509
SDN-MCFP	40%	72	220199	2819
SDN-OPT	40%	77	231690	2970
SDN-LB	60%	72	75675	9497
SDN-MCFP	60%	72	222592	2827
SDN-OPT	60%	146	421560	2980
SDN-LB	80%	72	74583	9492
SDN-MCFP	80%	72	221436	2774
SDN-OPT	80%	359	993077	2967
SDN-LB	100%	72	74215	9490
SDN-MCFP	100%	72	173322	2267
SDN-OPT	100%	576	1292473	2394

klasu pojedinačno. Takođe, bitno je napomenuti da se broj zapisa u tabelama tokova može dodatno regulisati uklanjanjem nekorišćenih tunela u svakom ciklusu izvršavanja *offline* TE modula.

Imajući u vidu teško predvidljivu prirodu saobraćajnih zahtjeva u ISP mrežama, analiziran je uticaj učestanosti *offline* optimizacija na performanse SDN-LB pristupa. Simulacije su odrađene za različite periode (T) ciklusa izvršavanja *offline* TE modula: 5, 20 i 30 minuta. Dobijeni rezultati u pogledu odbijenih jedinica BoD zahtjeva prikazani su na Slici 4.12, dok su rezultati u pogledu prosječne MLU vrijednosti dati na Slici 4.13. Povećanje vrijednosti T parametra od 5 do 30 minuta rezultiralo je povećanjem odbijenih BoD zahtjeva. Degradacija performansi posledica je velikih varijacija matrice saobraćaja tokom intervala između uzastopnih izvršavanja *offline* TE modula. Stoga, parametri za balansiranje opterećenja koji su optimalni na početku tog intervala značajno odstupaju

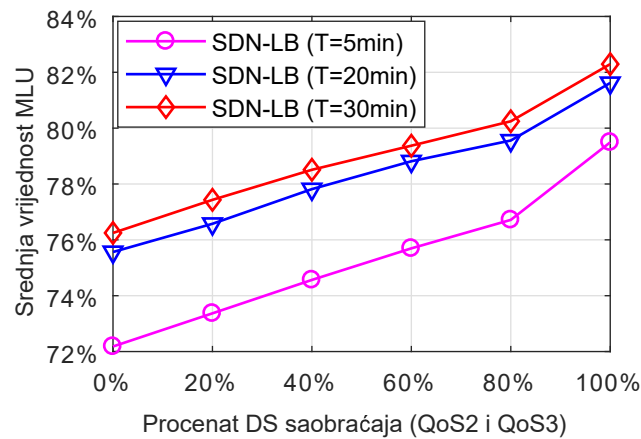
Tabela 4.2: Analiza skalabilnosti/kompleksnosti različitih SDN-TE pristupa - Scenario II. Broj rekonfiguracija (Rekonf), broj generisanih kontrolnih poruka (Flow-mod) i broj zapisa u tabelama tokova (FT zapisi).

	DS %	Rekonf	Flow-mod	FT zapisi
SDN-LB	20%	72	118077	9707
SDN-MCFP	20%	72	308036	3692
SDN-OPT	20%	472	1716335	3991
SDN-LB	40%	72	125489	9741
SDN-MCFP	40%	72	356820	4183
SDN-OPT	40%	3382	12980354	4141
SDN-LB	60%	72	142545	9822
SDN-MCFP	60%	72	387878	4447
SDN-OPT	60%	9935	39414465	4177



Slika 4.12: Uticaj parametra T na broj odbijenih BoD zahtjeva.

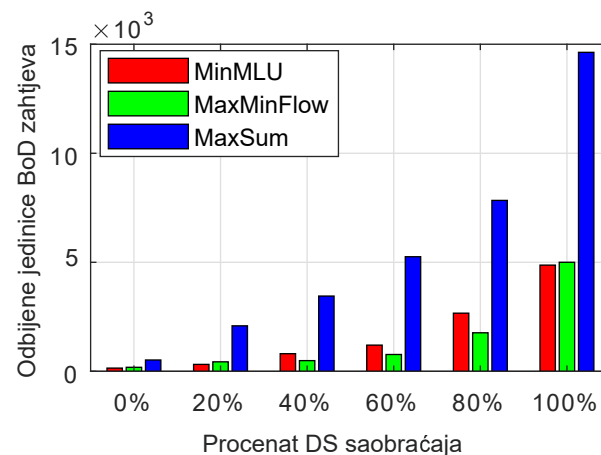
od pravog optimuma nakon nekog vremena. Međutim, u odnosu na konvencionalne TE pristupe poboljšanja su značajna i za velike vrijednosti T . Stoga, može se zaključiti da se solidne performanse mogu očekivati čak i kada matrica saobraćaja nije tačno poznata. Prilikom određivanja odgovarajuće vrijednosti T parametara potrebno je dodatno razmotriti njegov uticaj na kontrolni *overhead* i broj rekonfiguracija. Ovaj problem detaljnije je razmatran u Poglavlju 5.



Slika 4.13: Uticaj parametra T na srednju MLU vrijednost.

4.3.3 Analiza uticaja optimizacione funkcije

Na Slici 4.14 prikazani su rezultati analize uticaja optimizacionih funkcija diskutovanih u Sekciji 4.1.3 na performanse SDN-LB pristupa. Može se primijetiti da su najbolji rezultati ostvareni sa *MaxMinFlow* optimizacionom funkcijom. *MaxMinFlow* teži da izbjegne opterećivanje linkova koje koristi veliki broj tunela. Ovi linkovi su kritični za veliki broj IO parova, što objašnjava poboljšanje performansi u odnosu na standardni pristup MLU minimizacije, koji je označen sa *MinMLU*. Međutim, *MaxMinFlow* ne diferencira tunele u pogledu njihovog značaja za DS saobraćaj, pa su njegove performanse nešto lošije u scenariju sa 100%-tnim udjelom DS saobraćaja. Sa *MaxSum* optimizacionom funkcijom ostvareni su najgori rezultati u svim simulacijama. Analiza je pokazala da sa ovom funkcijom ipak nije moguće efikasno balansirati saobraćajno opterećenje.



Slika 4.14: Uticaj optimizacione funkcije na broj odbijenih BoD zahtjeva.

Glava 5

Predlog TE rešenja sa kontrolisanim rekonfiguracijama: RO-RO

Iako SDN nudi nove mogućnosti za efikasno upravljanje saobraćajem, u većim mrežama centralizovana kontrolna ravan nameće brojne izazove u pogledu skalabilnosti. Da bi obezbijedio optimalan rad mreže, SDN kontroler mora da reaguje na promjene saobraćaja i događaje u mreži u realnom vremenu. Ovo zahtijeva rešavanje TE optimizacionih problema u uskim vremenskim okvirima i rekonfiguraciju tabela tokova SDN uređaja na osnovu rezultata optimizacije. Međutim, česte rekonfiguracije negativno se odražavaju na kvalitet servisa jer kontroler mora da ukloni stare rute i uspostavi nove, rizikujući privremena zagušenja linkova i kreiranje petlji rutiranja [18]. Uobičajeni pristup rešavanju ovog problema je periodično izvršavanje TE algoritma u dovoljno velikim vremenskim razmacima. Tako, na primjer, u komercijalnim ISP mrežama TE algoritam se obično izvršava u intervalima od jednog sata ili više [124]. Manja učestanost optimizacija znači veću pouzdanost i stabilnost mreže, ali mreža koristi neoptimalnu konfiguraciju duže vremena, što može biti kritično kada se uslovi u mreži mijenjaju nepredvidljivo. U novijoj literaturi predložena su dva sofisticiranija pristupa [18], [19]. U [19], predložena je upotreba iterativnog optimizacionog solvera uz kontrolnu politiku koja definiše kada rešenje solvera treba primijeniti. Kontrolna politika je dizajnirana da minimizuje broj rekonfiguracija u skladu sa dozvoljenim „budžetom”. Međutim, ovaj pristup baziran je na pretpostavci da su karakteristike stohastičkih procesa koji odgovaraju različitim događajima u mreži (dolasci i odlasci tokova, kvarovi) unaprijed poznate, što je u praksi teško očekivati. Sa druge strane, u [18] predložen je optimizacioni model koji minimizuje MLU i broj ažuriranja tabela tokova. Primjenjivost ovog pristupa je veoma upitna, s obzirom da je predviđe-

no da se kompleksni optimizacioni problem rešava prilikom obrade svakog saobraćajnog zahtjeva.

U ovom poglavlju predložena je nova TE tehnika koja omogućava skalabilnu alokaciju resursa za virtuelne linkove u SD-ISP mrežama. Predloženo rešenje bazirano je na više-kriterijumskom optimizacionom modelu koji teži da ispuni QoS zahtjeve tokova i odredi težinske faktore za balansiranje saobraćajnog opterećenja tako da dva konfliktna cilja budu ispunjena: (i) maksimizacija servisnog kapaciteta mreže; i (ii) minimizacija ažuriranja tabela tokova. Predloženi pristup omogućava mrežnom operatoru da definiše ograničenje u pogledu srednje učestanosti rekonfiguracija, koje se koristi prilikom određivanja najefikasnijeg kompromisa između optimalnosti šeme rutiranja i broja rekonfiguracija. Ovaj postupak objašnjen je u Sekciji 5.1. Heuristika koja omogućava efikasno rešavanje optimizacionog problema u većim mrežama predložena je u Sekciji 5.2. Performanse predloženog rešenja analizirane su u Sekciji 5.3.

5.1 RO-RO optimizator

U nastavku poglavlja podrazumijeva se model SD-ISP mreže iz Sekcije 4, gdje su TE algoritam i algoritam rutiranja/kontrole pristupa implementirani u odvojenim modulima SDN kontrolera. Fokus je na unapređenju *offline TE* modula u smislu smanjenja broja rekonfiguracija. Originalno poboljšanje predstavljeno je u [16] pod nazivom RO-RO (*Routing Optimality - Reconfiguration Overhead balance*), pa će u nastavku za modul kontrolera koji izvršava RO-RO algoritam biti usvojen naziv RO-RO optimizator.

RO-RO optimizator periodično analizira stanje u mreži (svakih T sekundi) i utvrđuje set optimalnih TE akcija za unapređenje istog. Optimalnost TE odluke ocjenjuje se na osnovu dva kriterijuma: (i) rezultujuće MLU vrijednosti, i (ii) broja zahtijevanih rekonfiguracija. Očigledno je da su ova dva kriterijuma međusobno suprotstavljena, i da ne postoji jedinstveno rešenje koje simultano optimizuje obje optimizacione funkcije. Sa druge strane, višekriterijumski optimizacioni problemi obično imaju set Pareto optimalnih rešenja, kao što je objašnjeno u Poglavlju 2. Stoga se RO-RO optimizacija izvršava se u dvije faze. Prva faza je generisanje seta Pareto optimalnih rešenja (tzv. Pareto front). U drugoj fazi bira se najefikasnije kompromisno rešenje iz Pareto seta.

5.1.1 Generisanje Pareto optimalnih rešenja

Generisanje Pareto optimalnih rešenja za RO-RO optimizacioni problem vrši se primenom AUGMECON metoda [125]. U Tabeli 5.1 date se definicije parametara i varijabli koji su sastavni dio optimizacionog modela. Na samom početku, određuje se optimalna MLU vrijednost - MLU^{opt} . Nakon toga, u više iteracija optimizuje se broj ažuriranja *group* tabela *OpenFlow* svičeva, dodavanjem ograničenja: $MLU \leq MLU^*$, gdje MLU^* varira u opsegu od $MLU(t)$ do MLU^{opt} . Ovi koraci objašnjeni su detaljnije u nastavku.

MLU minimizacija (Korak 1):

Za pronalaženje optimalne MLU vrijednosti iskorišćen je model iz Poglavlja 4, pri čemu su eliminisane cjelobrojne varijable koje garantuju odabir rutra sa kašnjenjem ispod kritične vrijednosti. Pod pretpostavkom da je za svaku od analiziranih QoS klasa saobraćaja $C \in \{qos1, qos2, qos3\}$ set ruta (tunela) koje mogu ispuniti zahtjev u pogledu kašnjenja unaprijed poznat ($P_{uv}^{(C)}$), problem MLU minimizacije može se definisati bez cjelobrojnih varijabli, čime se značajno smanjuje njegova računaska kompleksnost:

$$\min \quad MLU \quad (5.1a)$$

s.t.

$$\sum_{p \in P_{uv}^{(C)}} x_{d_{uv}^{(C)}, p} = 1 \quad C \in \{qos1, qos2\}, \forall (u, v) \in N \quad (5.1b)$$

$$\sum_{p \in P_{uv}^{(C)}} x_{d_{uv}^{(qos3)}, p} \geq 1 \quad \forall (u, v) \in N \quad (5.1c)$$

$$\sum_{\forall C} \sum_{(u,v) \in N} \sum_{p \in P_{uv}^{(C)}} R_{p,l} \cdot x_{d_{uv}^{(C)}, p} \cdot d_{uv}^{(C)} \leq MLU \cdot c_l \quad \forall l \in L \quad (5.1d)$$

$$\sum_{p_{uv} \in P_{uv}^{(qos3)}} x_{p_{uv}}^{(qos3)} - \sum_{k=1}^{n \cdot q} x_{uv}^{max_k} \geq 1 \quad \forall (u, v) \in N \quad (5.1e)$$

$$x_{d_{uv}^{(C)}, p} \geq 0 \quad \forall C, \forall (u, v) \in N, \forall p \quad (5.1f)$$

$$0 \leq MLU \leq 1 \quad (5.1g)$$

Ograničenja konzervacije agregiranih tokova definisana su sa (5.1b) i (5.1c). U optimizacionom modelu razmatrani su samo QoS tokovi, koji odgovaraju BoD zahtjevima, pa ukupni rezervisani propusni opseg na linku ne smije biti veći od kapaciteta linka, što je formulisano sa (5.1d).

Tabela 5.1: Notacija za RO-RO optimizacioni model

Parametar	Opis
$P_{uv}^{(C)}$	Tuneli između čvorova u i v koji mogu ispuniti zahtjev u pogledu kašnjenja za klasu C
$d_{uv}^{(C)}$	Agregirani saobraćajni tok klase C između IO para (u,v)
$x_{d_{uv}^{(C)},p}$	Težinski faktor za balansiranje agregiranog toka $d_{uv}^{(C)}$ preko tunela p
c_l	Kapacitet linka l
l_{prop}	Propagaciono kašnjenje na linku l
$R_{p,l}$	1 ukoliko ruta p sadrži link l , 0 u suprotnom
MLU^{opt}	Optimalna MLU vrijednost
$MLU(t)$	MLU u trenutku t
MLU^*	MLU vrijednost iz opsega $MLU(t) - MLU^{opt}$
α	Broj ažuriranja <i>group</i> tabela
α_n^*	Rešenje optimizacionog modela (5.2) u n -toj iteraciji
$\alpha(t)^*$	Broj rekonfiguracija koje RO-RO optimizator trigeruje u trenutku t
$\beta_{uv,p}^{(C)}$	1 u slučaju promjene $x_{d_{uv}^{(C)},p}$, 0 u suprotnom
$\gamma_{uv,p}^{(C)}$	Realni indikator promjene $x_{d_{uv}^{(C)},p}$
$g_{uv}^{(C)}$	1 u slučaju ažuriranja <i>group</i> tabele za IO par (u,v) i klasu C , 0 u suprotnom
r	Opseg MLU varijable: $MLU(t) - MLU^{opt}$
s	<i>Surplus</i> varijabla
PF	Pareto front
R_{avg}	Ograničenje u pogledu srednjeg broja rekonfiguracija po minuti
R_{max}	Ograničenje u pogledu maksimalnog broja rekonfiguracija u jednom ciklusu izvršavanja RO-RO optimizatora
$V(t)$	Zauzetost virtuelnog bafera u trenutku t
Γ	Odstupanje trenutne MLU vrijednosti od MLU^{opt}
T	Vremenski interval između dva uzastopna izvršavanja RO-RO optimizatora

Iterativna pretraga Pareto optimalnih rešenja (Korak 2):

Rezultat prvog koraka je optimalna MLU vrijednost - MLU^{opt} . Nakon određivanja MLU^{opt} , RO-RO optimizator definiše opseg MLU vrijednosti r , sa donjom granicom MLU^{opt} i gornjom granicom $MLU(t)$. Zatim, u cilju generisanja Pareto fronta, ovaj opseg se dijeli na 10 jednako udaljenih tačaka MLU_n^* ($0 \leq n \leq 10$) koje se nalaze između MLU^{opt} i $MLU(t)$. Pri tome, primjenjuje se dodatno ograničenje da rastojanje između dvije susjedne tačke ne može biti manje od 0.02. Naime, uočeno je da korišćenje većeg broja tačaka, ili korišćenje manjeg rastojanja između tačaka, ne doprinosi značajnije poboljšanju performansi mreže. Svaka od generisanih tačaka MLU_n^* koristi se kao ograničenje u sledećem optimizacionom modelu:

$$\min \quad \alpha - \frac{\epsilon \cdot s}{r} \quad (5.2a)$$

s.t.

$$(5.1b), (5.1c), (5.1e), (5.1f)$$

$$\alpha = \sum_{(u,v) \in N} \sum_{\forall C} g_{uv}^{(C)} \quad (5.2b)$$

$$\beta_{uv,p}^{(C)} = \begin{cases} 0 & , if x(t)_{d_{uv}^{(C)},p} = x(t-T)_{d_{uv}^{(C)},p} \\ 1 & , else \end{cases} \quad (5.2c)$$

$$\forall (u,v) \in N, \forall C, \forall p \in P_{uv}^{(C)} \quad (5.2d)$$

$$g_{uv}^{(C)} \geq \beta_{uv,p}^{(C)} \quad \forall (u,v) \in N, \forall C, \forall p \in P_{uv}^{(C)}$$

$$z + s = MLU_n^* \quad (5.2e)$$

$$\sum_{\forall C} \sum_{(u,v) \in N} \sum_{p \in P_{uv}^{(C)}} R_{p,l} \cdot x(t)_{d_{uv}^{(C)},p} \cdot d_{uv}^{(C)} \leq z \cdot c_l \quad \forall l \in L \quad (5.2f)$$

$$g_{uv}^{(C)} \in \{0, 1\} \quad \forall (u,v) \in N, \forall C \quad (5.2g)$$

$$\beta_{uv,p}^{(C)} \in \{0, 1\} \quad \forall (u,v) \in N, \forall C, \forall p \in P_{uv}^{(C)} \quad (5.2h)$$

$$s \geq 0 \quad (5.2i)$$

$$0 \leq z \leq 1 \quad (5.2j)$$

Osnovna ideja predloženog rešenja je minimizacija broja rekonfiguracija α , u skladu sa MLU i QoS ograničenjima. Pod rekonfiguracijama ovdje se razmatraju samo ažuriranja *group* tabela, jer ona utiču na distribuciju agregiranih saobraćajnih tokova. MLU ograničenje transformisano je u jednakost (5.2e) uvođenjem *surplus* varijable s , koja utiče na

korelisano ograničenje (5.2f). Na ovaj način izbjegava se generisanje tzv. neefikasnih Pareto rešenja [125]. *Surplus* varijabla takođe se koristi u optimizacionoj funkciji (5.2a), ali kao sabirak manjeg prioriteta, pa se konstanti ϵ dodjeljuje mala vrijednost (tipično između 10^{-6} i 10^{-3}). Ograničenja (5.2b) - (5.2d) omogućavaju minimizaciju broja ažuriranja *group* tabela. U razmatranom scenariju, svaki OpenFlow svič u koristi 3 *group* tabele za QoS saobraćaj prema odredišnom čvoru v - po jednu za svaku od klasa $C \in \{qos1, qos2, qos3\}$. Varijabla $g_{uv}^{(C)}$ prati promjene *group* tabele za IO par (u, v) i QoS klasu C . U slučaju promjene, $g_{uv}^{(C)}$ uzima vrijednost 1, a 0 u suprotnom. Da bi se napravila razlika između novih i starih vrijednosti težinskih faktora za balansiranje opterećenja, $x_{d_{uv}^{(C)},p}$ variable su predstavljene u funkciji vremena t . U (5.2c), binarne varijable $\beta_{uv,p}^{(C)}$ su jednake 1 kada se novoizračunata vrijednost težinskog faktora $x(t)_{d_{uv}^{(C)},p}$ ne poklapa sa starom vrijednošću $x(t-T)_{d_{uv}^{(C)},p}$. U suprotnom, vrijednost ovih varijabli je 0. Ograničenje (5.2d) indicira da se *group* tabela ažurira samo ukoliko se barem jedan od njenih težinskih faktora promijenio u odnosu na svoju prethodnu vrijednost.

U (5.2c) neophodno je provjeriti da li je uslov $\left| x(t)_{d_{uv}^{(C)},p} - x(t-T)_{d_{uv}^{(C)},p} \right| > 0$ zadovoljen. Međutim, apsolutna vrijednost nije prihvatljiva za LP optimizacione solve. U cilju eliminisanja potrebe za ovim operatorom, uvode se dodatne varijable $\gamma_{uv,p}^{(C)}$:

$$\forall (u, v) \in N, \forall C, \forall p \in P_{uv}^{(C)} :$$

$$\gamma_{uv}^{(C)} \geq x(t)_{d_{uv}^{(C)},p} - x(t-T)_{d_{uv}^{(C)},p} \quad (5.2k)$$

$$\gamma_{uv}^{(C)} \geq x(t-T)_{d_{uv}^{(C)},p} - x(t)_{d_{uv}^{(C)},p} \quad (5.2l)$$

S obzirom da varijable $\beta_{uv,p}^{(C)}$ moraju biti binarne da bi reflektovale promjene težinskih faktora za balansiranje opterećenja, potrebno je definisati još jedno ograničenje:

$$\eta \cdot \gamma_{uv}^{(C)} \leq \beta_{uv,p}^{(C)} \leq \eta \cdot \gamma_{uv}^{(C)} + 1 \quad (5.2m)$$

gdje je η konstanta male vrijednosti (npr. 10^{-5}). Stoga, konačna definicija problema je: minimizacija (5.2a), pod ograničenjima (5.2b), (5.2d)-(5.2m), i QoS ograničenjima (5.1b), (5.1c), (5.1e) i (5.1f). Za rešenje optimizacionog modela (5.2) u n -toj iteraciji usvojena je notacija α_n^* , pa se rezultujući Pareto front nakon N iteracija može predstaviti na sledeći način:

$$PF = \{(MLU_1^*, \alpha_1^*), (MLU_2^*, \alpha_2^*) \dots (MLU_N^*, \alpha_N^*)\} \quad (5.3)$$

5.1.2 Izbor kompromisnog rešenja

U svakom ciklusu izvršavanja RO-RO optimizator bira jedno od Pareto optimalnih rešenja iz seta (5.3). U opštem slučaju, rešenje (MLU_n^*, α_n^*) zahtijeva α_n^* ažuriranja *group* tabela i unosi odstupanje Γ od optimalne MLU vrijednosti:

$$\Gamma = MLU_n^* - MLU^{opt} \quad (5.4)$$

Uobičajeni metodi za rangiranje Pareto optimalnih rešenja u MOP problemima su: *Weighted Sum Method* (WSM), *Weighted Product Method* (WPM) i *Weighted Average Ranking* (WAR) [126]. Ovi metodi koriste težinske faktore za definisanje kompromisa između različitih optimizacionih kriterijuma. U [18], WSM metod iskorišćen je za pronalaženje najefikasnijeg kompromisnog rešenja na osnovu dva kriterijuma: MLU i broja ažuriranja tabela tokova. Međutim, nasuprot [18], RO-RO teži da minimizuje prosječno odstupanje od optimalne MLU vrijednosti (Γ) održavajući prosječnu učestanost rekonfiguracija ispod R_{avg} - preferirane vrijednosti koju definiše mrežni operator. Treba primijetiti da ovo zahtijeva dinamičko podešavanje prioriteta optimizacionih funkcija, pa WSM, WSP i WAR metodi rangiranja nisu adekvatni za tu primjenu. U predloženom rešenju, ograničenje za učestanost rekonfiguracija modelovano je virtuelnim baferom, čije se stanje mijenja u slotovima $t \in \{0, T, 2T \dots\}$ sledećom dinamikom:

$$V(t+T) = [V(t) - R_{avg}]^+ + \alpha(t)^*/T \quad (5.5)$$

U gornjoj jednačini $V(t)$ predstavlja zauzetost bafera u posmatranom trenutku t , a $V(t+T)$ predstavlja očekivanu zauzetost bafera u trenutku $t+T$, kada će RO-RO optimizator ponovo biti aktiviran. Brzina opsluživanja bafera odgovara željenoj srednjoj brzini (tj. učestanosti) rekonfiguracija - R_{avg} . Broj dolazaka u bafer tokom vremena $[t, t+T]$ odgovara broju rekonfiguracija $\alpha(t)^*$ koje RO-RO optimizator odluči da izvrši u trenutku t . Početna zauzetost virtuelnog bafera $V(0)$ je podešena na nulu. Lako se može uočiti da iz (5.5) slijedi:

$$\frac{\alpha(t)^*}{T} - R_{avg} \leq V(t+T) - V(t) \quad (5.6)$$

Nakon primjene teoreme teleskopskih redova, dobija se:

$$\sum_{\tau=0}^{t-1} \left(\frac{\alpha(\tau)^*}{T} - R_{avg} \right) \leq V(t) - V(0) = V(t) \quad (5.7)$$

Dijeljenje prethodnog izraza sa t rezultira sa:

$$\frac{1}{t} \sum_{\tau=0}^{t-1} E \left[\left(\frac{\alpha(\tau)^*}{T} - R_{avg} \right) \right] \leq \frac{E[V(t)]}{t} \quad (5.8)$$

Dakle, ograničenje u pogledu srednje učestanosti rekonfiguracija biće zadovoljeno ako je virtuelni bafer stabilan pri prosječnoj brzini popunjavanja i opsluživanja [31], tj. kada je sledeći uslov zadovoljen:

$$\lim_{t \rightarrow \infty} \frac{E[V(t)]}{t} = 0 \quad (5.9)$$

Bafer se može stabilizovati minimizacijom Lyapunov *drift* funkcije [31], koja prati promjene zauzetosti bafera tokom vremena:

$$L(t) = V(t+T)^2 - V(t)^2 \quad (5.10)$$

Međutim, dok akcije koje minimizuju $L(t)$ pomažu u ispunjenju zahtjeva u pogledu srednje učestanosti rekonfiguracija, razlika između rezultujuće i optimalne MLU vrijednosti može postati neprihvatljivo velika. Stoga, umjesto donošenja odluke koja minimizuje $L(t)$, RO-RO optimizator bira Pareto optimalno rešenje koje minimizuje *drift-plus-penalty* funkciju:

$$A \cdot \Gamma(t) + L(t) \quad (5.11)$$

gdje je $A \geq 0$ konstanta koja naglašava relativni značaj prve optimizacione funkcije. Povećavanjem A moguće je MLU približiti optimalnoj vrijednosti na račun povećanja prosječne zauzetosti virtuelnog bafera.

5.2 RO-RO heuristika

Imajući u vidu da je generisanje Pareto optimalnih rešenja za RO-RO optimizacioni model NP-complete problem (zbog cjelobrojnih varijabli), u ovoj sekciji predložena je

heuristika za računanje aproksimacije Pareto fronta. Pseudo-kod heuristike dat je Algoritmom 3.

U prvom koraku algoritam poziva LP optimizacioni model (5.1) koji minimizuje MLU za trenutnu matricu saobraćaja. Ukoliko su tuneli koji mogu ispuniti zahtjeve u pogledu kašćenja za različite klase saobraćaja unaprijed poznati, ovaj model se može riješiti u polinomijalnom vremenu. Nakon toga, određuje se broj rekonfiguracija α koje optimalno MLU rešenje zahtijeva (linija 2). S obzirom da je u prvom koraku odrađena optimizacija po samo jednom kriterijumu, MLU^{opt} se sa velikom vjerovatnoćom može postići i sa značajno manjim brojem rekonfiguracija od α . Stoga, u narednim koracima, cilj je približiti se MLU^{opt} vrijednosti rerutiranjem što manjeg broja agregiranih tokova. Ulazni parametar algoritma je ograničenje u pogledu broja iteracija N_{max} - koje između ostalog definiše maksimalan broj izvršavanja LP modela (5.1). Rezultat dijeljenja α sa N_{max} predstavlja „rezoluciju” prostora rešenja - α_{step} . Drugim riječima, set mogućih vrijednosti *overhead*-a rekonfiguracije $A^* = \{\alpha_1^*, \alpha_2^*, \dots, \alpha_{N_{max}}^*\}$ u aproksimiranom Pareto frontu dobija se kao umnožak od α_{step} (linije 7-12). Pretpostavka je da ISP specificira ograničenje R_{max} u pogledu maksimalnog broja rekonfiguracija tokom bilo kojeg intervala T . Stoga, u skup A^* se dodaju samo multipli od α_{step} koji su manji od R_{max} . Elementi A^* se zatim sortiraju u rastući poredak, i iterativno analiziraju (linije 15-21). U svakoj iteraciji određuje se set od α_n^* ($1 \leq \alpha_n^* \leq N_{max}$) kritičnih zahtjeva za koje će težinski faktori za balansiranje biti optimizovani LP modelom (5.1) (linija 18). Ovdje, pod saobraćajnim „zahtjevom” podrazumijeva se ukupno generisano opterećenje nekog IO para, koje pripada istoj QoS klasi.

Algoritam 3 konstruiše kompletan set od α_n^* ($1 \leq n \leq N_{max}$) „kritičnih” zahtjeva (tj. agregiranih tokova) tako što prvo dodaje α_{n-1}^* zahtjeva koji su određeni u prethodnim iteracijama algoritma. Nakon toga, poziva funkciju *odrediKritičneZahtjeve()* u cilju pronalaženja dodatnih α_{step} „kritičnih” zahtjeva na osnovu informacija o zauzetosti resursa u slučaju kada se primijeni rešenje LP modela (5.1) iz prethodne iteracije. Pseudo-kod funkcije za računanje α_{step} „kritičnih” zahtjeva dat je u Algoritmu 4. U cilju pronalaženje zahtjeva koji mogu najviše doprinijeti smanjenju MLU, ova funkcija prvo identifikuje najopterećeniji link u mreži na osnovu poznatih informacija o zauzetosti resursa (linija 6). Zatim, zahtjev koji najviše doprinosi opterećenju ovog linka se proglašava „kritičnim” i dodaje u set *LPzahtjevi*, pod uslovom da u prethodnim iteracijama već nije proglašen „kritičnim” (linije 8-9). Nakon određivanja kritičnog zahtjeva, lokalna kopija informacija o zauzetosti resursa (*statistike*) se ažurira tako što se dodaju resursi koji su predviđeni za

Algoritam 3 RO-RO heuristika

Ulaz: *stats* → mrežne statistike, *zahtjevi* → agregirani tokovi između IO parova, *diferencirani* po klasama. N_{max} → maksimalni broj iteracija;

Izlaz: *PF* = aproksimirani Pareto front;

Komentar: *minimizujMLU()* rešava *optimizacioni model (5.1)*, *LBparametri* → *težinski faktori za balansiranje opterećenja*;

```
1:  $MLU^{opt}$ , LBparametri = minimizujMLU(zahtjevi)
2:  $\alpha$  = brojRekonfiguracija( $MLU^{opt}$ )
3: PF = { }
4: PF.dodaj(( $MLU^{opt}$ ,  $\alpha$ ))
5:  $\alpha_{step} = \alpha / N_{max}$ 
6:  $A^* = \{ \}$ 
7: for  $i$  in range (1,  $N_{max}$ ) do
8:   if  $\alpha_{step} \cdot i > R_{max}$  then
9:     break
10:  end if
11:   $A^*.dodaj(i \cdot \alpha_{step})$ 
12: end for
13:  $A^*.sortiraj()$  /*opadajući redosled
14: LPahtjevi = { }
15: for  $\alpha_n^*$  in  $A^*$  do
16:  kritični = odrediKriticneZahtjeve( $\alpha_{step}$ , LPzahtjevi)
17:  LPzahtjevi.add(kritični)
18:   $MLU^*$ , LBparametri = minimizujMLU(LPzahtjevi)
19:  PF.dodaj(( $MLU_n^*$ ,  $\alpha_n^*$ ))
20:  ažurirajStanjeMreže(LBparametri)
21: end for
```

Algoritam 4 Algoritam za određivanje kritičnih zahtjeva

```
1: function odrediKritičneZahtjeve( $\alpha_{step}$ , LPzahtjevi)
2:   rezultat = { }
3:   brojač = 0
4:   statistike = stats.copy()
5:   while brojač <  $\alpha_{step}$  do:
6:     kritičniLink=pronađiNajopterećenijiLink()
7:     kritični_zah_tjev=kritičniLink.najvećiZah_tjev(statistike)
8:     if kritični_zah_tjev in LPzahtjevi do: continue
9:     end if
10:    rezultat.dodaj(kritični_zah_tjev)
11:    statistike.ukloni(kritični_zah_tjev)  /*Zanemaruje saobraćaj kritičnog toka
12:    brojač+=1
13:  end while
14:  return rezultat
15: end function
```

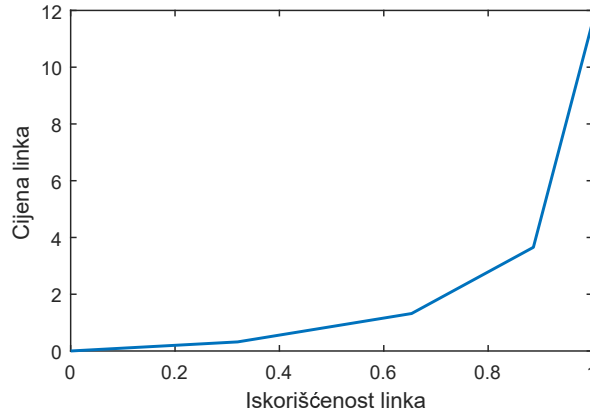
ovaj zahtjev, tj. prisustvo ovog zahtjeva se ignoriše do kraja poziva funkcije (linija 11). Prethodno opisana procedura ponavlja se α_{step} puta.

Nakon što kompletira set $LPzahtjevi$, Algoritam 3 poziva LP model (5.1) u cilju optimizacije težinskih faktora za balansiranje kritičnih tokova (linija 18). Težinski faktori ostalih („nekritičnih”) zahtjeva ostaju nepromijenjeni. Na osnovu rešenja LP modela (5.1), informacije o stanju u mreži se ponovo ažuriraju tako da reflektuju promjene koje nastaju primjenom nove šeme za balansiranje saobraćaja (linija 20). Rezultujuća MLU vrijednost (MLU_n^*), u paru sa α_n^* , se dodaje u aproksimirani Pareto front.

Najefikasnije kompromisno rešenje iz aproksimiranog Pareto fronta se bira *drift-plus-penalty* algoritmom koji je objašnjen u prethodnoj sekciji.

5.3 RORO-MC: kompromisno rešenje za minimizaciju cijene rutiranja

Na vrijednost prethodno opisane RO-RO optimizacione funkcije utiču samo broj re-konfiguracija i iskorišćenost najopterećenijeg linka. Iako se MLU minimizacija često sma-



Slika 5.1: Cijena linka u funkciji iskorišćenosti linka.

tra prirodnom i intuitivnom optimizacionom funkcijom za ISP mreže [15, 95], MLU kao metrika je osjetljiva na individualne *bottleneck* linkove, koje je ponekad teško izbjeći sa bilo kojom šemom rutiranja. Iz ovog razloga, u nastavku poglavlja razmatrana je i alternativna formulacija RO-RO optimizacione funkcije, koja teži da minimizuje ukupnu „cijenu” rutiranja i broj rekonfiguracija. Predloženo rešenje publikovano je u [16] pod nazivom RORO-MC, gdje sufiks MC predstavlja skraćenicu za *Minimizing routing Cost*. U odnosu na prethodnu definiciju RORO optimizacionog problema, RORO-MC uvodi novu formu optimizacione funkcije za LP model (5.1):

$$\min \quad MLU + w \cdot \sum_{l \in L} cost(l_a) \quad (5.12a)$$

U gornjoj formulaciji cijena linka (*cost*) je konveksna funkcija iskorišćenosti linka l_a . Iskorišćenost linka definisana je kao odnos ukupnog opterećenja linka x i kapaciteta linka c_l . Konstanta w kontroliše uticaj cijene rutiranja na vrijednost optimizacione funkcije. Eksperimentalnom analizom utvrđeno je da pri vrijednosti 10^{-4} konstanta w pruža zadovoljavajući kompromis između MLU i cijene rutiranja. Funkcija l_a definisana je kao izlomljena linearna funkcija sa Slike 5.1, koja je inicijalno predložena u [71] i često korišćena u literaturi. Generalno, ova funkcija sugerise da je relativno „jeftino” prenositi saobraćaj preko slabo iskorišćenih linkova. Cijena progresivno raste kako se iskorišćenost linka približava 100%. Preciznije, za datu vrijednost c_l , izvod cijene linka definisan je na sledeći način:

$$cost'(l_a) = \begin{cases} 1, & \text{if } 0 \leq l_a \leq 1/3, \\ 3, & \text{if } 1/3 \leq l_a \leq 2/3, \\ 10, & \text{if } 2/3 \leq l_a \leq 9/10. \\ 70, & \text{if } 9/10 \leq l_a \leq 1. \end{cases} \quad (5.13)$$

Ključni koraci RORO-MC optimizatora sumirani su na Slici 5.2. U prvom koraku pronalazi se optimalna „cijena” rutiranja (RC^{opt}), koja odgovara optimalnoj vrijednosti funkcije (5.12). Nakon toga, Pareto optimalna rešenja za RORO-MC optimizacioni problem generišu se na način objašnjen u Sekciji 5.1.1, pri čemu je optimizacioni model (5.2) modifikovan na sledeći način:

$$\min \quad \alpha - \frac{\epsilon \cdot s}{r} \quad (5.14a)$$

s.t.

(5.1b), (5.1c), (5.1e), (5.1f)

$$\alpha = \sum_{(u,v) \in N} \sum_{\forall C} g_{uv}^{(C)} \quad (5.14b)$$

$$\beta_{uv,p}^{(C)} = \begin{cases} 0 & , \text{if } x(t)_{d_{uv}^{(C)},p} = x(t-T)_{d_{uv}^{(C)},p} \\ 1 & , \text{else} \end{cases} \quad (5.14c)$$

$$\forall (u,v) \in N, \forall C, \forall p \in P_{uv}^{(C)} \quad (5.14d)$$

$$g_{uv}^{(C)} \geq \beta_{uv,p}^{(C)} \quad \forall (u,v) \in N, \forall C, \forall p \in P_{uv}^{(C)} \quad (5.14d)$$

$$z + s = RC_n^* \quad (5.14e)$$

$$\sum_{\forall C} \sum_{(u,v) \in N} \sum_{p \in P_{uv}^{(C)}} R_{p,l} \cdot x(t)_{d_{uv}^{(C)},p} \cdot d_{uv}^{(C)} \leq MLU \cdot l_{cap} \quad \forall l \in L \quad (5.14f)$$

$$MLU + \sum_{l \in L} l_{cost} \leq z \quad (5.14g)$$

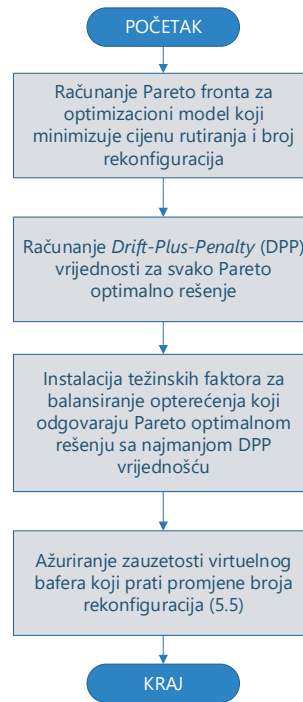
$$g_{uv}^{(C)} \in \{0, 1\} \quad \forall C, \forall (u,v) \in N \quad (5.14h)$$

$$\beta_{uv,p}^{(C)} \in \{0, 1\} \quad \forall C, \forall (u,v) \in N, \forall p \in P_{uv}^{(C)} \quad (5.14i)$$

$$s \geq 0 \quad (5.14j)$$

$$z \geq 0 \quad (5.14k)$$

Optimizacioni model (5.14) generiše Pareto optimalno rešenje (RC_n^*, α_n^*) , pri čemu je RC_n^* zadato ograničenje a α_n^* vrijednost optimizacione varijable α nakon izvršavanja



Slika 5.2: Princip rada RORO-MC optimizatora.

optimizacionog algoritma. Vrijednosti za RC_n^* biraju se iz opsega r , koji je definisan kao: $RC(t) - RC^{opt}$. Na kraju, kompromisno rešenje iz Pareto fronta bira se *drift-plus-penalty* algoritmom iz Sekcije 5.1, čija *penalty* funkcija predstavlja odstupanje trenutne cijene rutiranja od optimalne vrijednosti.

5.4 Simulaciona analiza

Za analizu performansi RO-RO TE pristupa primijenjena je metodologija iz Poglavlja 4. Konfiguracija parametara simulatora sumirana je u Tabeli 5.2. Predloženi pristup upoređen je sa SDN-LB pristupom [115], koji periodično minimizuje MLU. Pri tome, simulacije za SDN-LB urađene su za dvije različite vrijednosti periode izvršavanja MLU optimizacija: $T=5\text{min}$ i $T=1\text{h}$. Prva vrijednost je odabrana za demonstraciju prednosti i nedostataka dinamičkog SDN TE modela u ISP mrežama. Između ostalog, ova vrijednost odgovara periodi TE ciklusa u SWAN arhitekturi [1], koja je implementirana u Microsoft *inter-DC* mreži. Sa druge strane, prema [124], inženjering saobraćaja u konvencionalnim ISP mrežama obično se ponavlja u intervalima od 1h ili više. Stoga, u sprovedenoj analizi drugi scenario se smatra prihvatljivim sa aspekta učestanosti rekonfiguracija. Takođe,

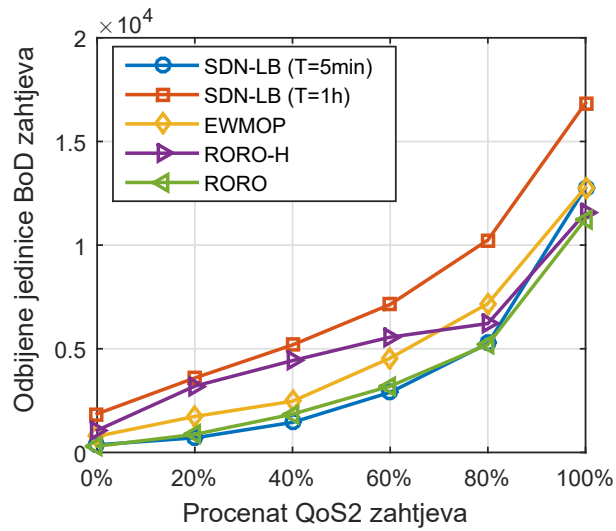
Tabela 5.2: Konfiguracija simulacija

Mrežna topologija	Sprint (na nivou PoP lokacija)
Broj čvorova	17
Broj linkova	72
Kapacitet linkova	1000 jedinica
Dolazni proces BoD zahtjeva	Poasonova raspodjela za svaki IO par (srednja vrijednost = prosječni zahtjev IO para/ prosječno trajanje zahtjeva)
BoD zahtjev u pogledu kapaciteta	[3, 5, 7] jedinica
Trajanje BoD zahtjeva	Eksponencijalna raspodjela (srednja vrijednost = 20 minuta)
Maksimalni broj tunela između IO parova (K)	10
Koeficijent preklapanja tunela IO para (q)	2
Ograničenje kašnjenja za QoS2 klasu	35ms
Ograničenje kašnjenja za QoS3 klasu	100ms
Koeficijent tolerancije na otkaz linkova (n)	1
Trajanje simulacije	1 dan (24 sata)

analizirane su performanse MOP TE pristupa koji je implementiran na osnovu ideja iz [18], gdje su MLU i *overhead* rekonfiguracija razmatrani kao jednako prioritizovane optimizacione funkcije. Za ovo rešenje usvojena je notacija EWMOP (*Equally Weighted Multi Objective oPtimization*). Simulaciona analiza podijeljena je na dva segmenta, koja odgovaraju saobraćajnim scenarijima bez i sa QoS3 BoD zahtjevima. Perioda izvršavanja RO-RO optimizatora inicijalno je podešena na 5 minuta. Heuristike predloženih algoritama označene su sufiksom 'H'. Broj iteracija RO-RO heuristike ograničen je na 100.

5.4.1 Performanse RO-RO pristupa u uslovima odsustva QoS3 zahtjeva

U cilju demonstracije mogućnosti RO-RO optimizatora, različiti TE pristupi upoređeni su u pogledu broja odbijenih jedinica BoD zahtjeva i učestanosti rekonfiguracija. Rezultati su predstavljeni u funkciji procenta QoS2 zahtjeva na Slici 5.3 i u Tabeli 5.3,



Slika 5.3: Odbijeni BoD zahtjevi.

respektivno. U simulacijama R_{max} parametar je konfigurisan na 250, a R_{avg} na 1.6 rekonfiguracija po minuti - što odgovara slučaju od prosječno 8 rekonfiguracija u svakom optimizacionom ciklusu. Konfiguracija R_{avg} parametra bazirana je na zapažanjima iz SDN-LB ($T=1h$) simulacija. Kao što je ranije objašnjeno, $T=1h$ se u ovoj analizi smatra prihvatljivom periodom optimizacija, koja je našla primjenu u realnim mrežama. Izvršavanjem simulacija za različite vrijednosti R_{avg} parametra zapaženo je da RO-RO nudi najefikasniji kompromis između srednjeg odstupanja MLU od optimalne vrijednosti i učestanosti rekonfiguracija kada je A reda 10^7 . Povećavanjem A parametra moguće je dalje smanjiti srednje odstupanje rezultujuće od optimalne MLU vrijednosti, međutim, broj rekonfiguracija može prevazići R_{avg} ograničenje u tom slučaju. Sa druge strane, identifikovano je da RORO-H realizuje sličan kompromis kada je A reda $5 \cdot 10^5$. Stoga, u nastavku poglavlja analizirane su performanse RO-RO i RORO-H algoritama samo za pomenute vrijednosti A parametra.

Sa Slike 5.3 interesantno je primijetiti da RO-RO odbija približno isti broj BoD zahtjeva kao SDN-LB ($T=5min$), dok pri tome smanjuje učestanost rekonfiguracija i do 25 puta (Tabela 5.3). Štaviše, kada su QoS2 zahtjevi dominantno zastupljeni, RO-RO nadmašuje performanse SDN-LB. Ovaj donekle neočekivani rezultat je posledica formulisanja problema optimizacije kao minimizacije MLU, usled čega jedan *bottleneck* link može da definiše stanje čitave mreže na neodgovarajući način. Takođe, optimizacioni model ne penalizuje korišćenje veoma dugih ruta, što uzrokuje intenzivnu potrošnju resursa. RO-RO algoritam je fokusiran na optimizaciju balansiranja opterećenja najaktivnijih IO parova,

Tabela 5.3: Poređenje RO-RO pristupa sa konkurentskim TE rešenjima u pogledu prosječnog broj rekonfiguracija u minuti.

QoS2 %	SDN-LB (T=5min)	SDN-LB (T=1h)	EWMOP	RORO-H	RORO
0 %	15.44	1.43	0.65	1.57	0.73
20 %	21.51	1.98	0.72	1.63	0.88
40 %	21.69	2.15	0.80	1.58	0.97
60 %	22.13	2.25	0.87	1.63	1.18
80 %	22.37	2.47	1.11	1.68	1.61
100 %	7.14	0.72	0.39	1.29	0.42

tj. onih koji generišu najveći broj zahtjeva. U nekim slučajevima ovo rezultira manjom prosječnom MLU vrijednošću iako RO-RO optimizator ne primjenjuje optimalnu šemu rutiranja u svakom ciklusu izvršavanja. Nedostaci SDN-LB pristupa su posebno izraženi u scenariju kada su svi BoD zahtjevi osjetljivi na kašnjenje (100% QoS2), jer je rutiranje ograničeno na mali broj tunela i tzv. „uska grla” na prenosnim rutama se lako kreiraju. U takvim uslovima ne postoji veliki broj validnih opcija za balansiranje QoS2 saobraćaja, pa je broj rekonfiguracija generalno manji.

U poređenju sa EWMOP pristupom, RO-RO odbija manji broj BoD zahtjeva. Pobjoljšanje servisnog kapaciteta rezultat je dinamičkog balansiranja između dva konfliktna optimizaciona kriterijuma: MLU i učestanosti rekonfiguracija. Naime, tokom procesa rangiranja Pareto optimalnih rešenja, RO-RO procjenjuje značaj MLU metrike na osnovu trenutnog stanja virtuelnog bafera koji prati devijacije broja rekonfiguracija tokom vremena u odnosu na R_{avg} ograničenje. Na ovaj način, RO-RO uspijeva da efikasno iskoristi „budžet” rekonfiguracija [19] koji definiše ISP provajder, u cilju postizanja što manje MLU vrijednosti. Sa druge strane, EWMOP pristup ne razmatra ograničenje ISP provajdera, pa je sklon precjenjivanju druge optimizacione funkcije.

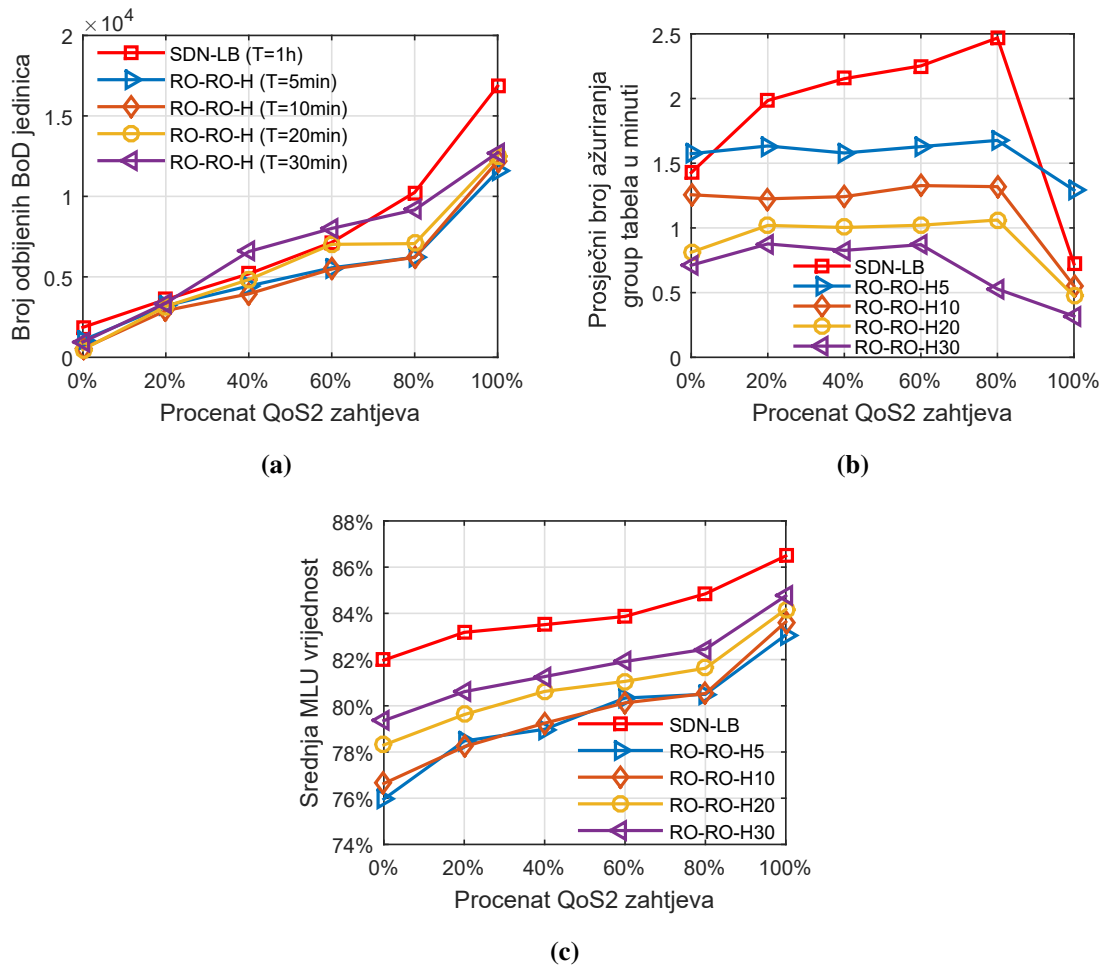
S obzirom da je RO-RO algoritam baziran na računski kompleksnom optimizacionom modelu, analiza performansi RORO-H heuristike je od većeg praktičnog značaja. Kao što se vidi sa Slike 5.3, RORO-H ostvaruje bolje performanse od SDN-LB pri sličnom broju rekonfiguracija. Ovo je primjetno iz poređenja sa SDN-LB (T=1h) pristupom, koji je u analizi deklarisan kao predstavnik periodičnih TE pristupa sa prihvatljivim *overhead*-om rekonfiguracije. U odnosu na SDN-LB (T=1h), RORO-H odbija manji broj BoD zahtjeva u svakom od simuliranih saobraćajnih scenarija. Iako je broj rekonfiguracije kod SDN-LB

pristupa moguće dalje smanjiti povećavanjem T parametra preko 1h, ovo bi rezultiralo nemogućnošću reagovanja na iznenadne distorzije saobraćaja. Sa druge strane, RORO-H posjeduje mehanizme za kontrolisanje broja rekonfiguracija, pa se može izvršavati češće u cilju efikasnije reakcije na brze promjene stanja u mreži. U simulacijama sa manje od 80% QoS2 saobraćaja, RORO-H odbija više BoD zahtjeva od EWMOP algoritma. Međutim, performanse RORO-H algoritma se i dalje mogu smatrati zadovoljavajućim, imajući u vidu da je njegova računaska kompleksnost značajno manja. Takođe, bitno je napomenuti da na performanse EWMOP algoritma značajno utiču uslovi u mreži. Na primjer, kada se Pareto optimalna rešenja veoma razlikuju u pogledu broja rekonfiguracija koje zahtijevaju, EWMOP optimizator indirektno podcjenjuje značaj odstupanja rezultujuće MLU vrijednosti od optimalne.

U nastavku sekcije performanse RO-RO i RORO-H algoritama analizirane su odvojeno. Da bi se fer analizom istakle prednosti predloženog rešenja, RORO-H upoređen je sa SDN-LB ($T=1h$), a RO-RO sa EWMOP pristupom. Naime, poređenje servisnog kapaciteta RORO-H i EWMOP pristupa ne može se smatrati fer, s obzirom da je EWMOP baziran na NP-hard optimizacionom modelu. Sa druge strane, SDN-LB ($T=1h$) i RORO-H su bazirani na LP modelima koji imaju komparabilan rekonfiguracioni *overhead*.

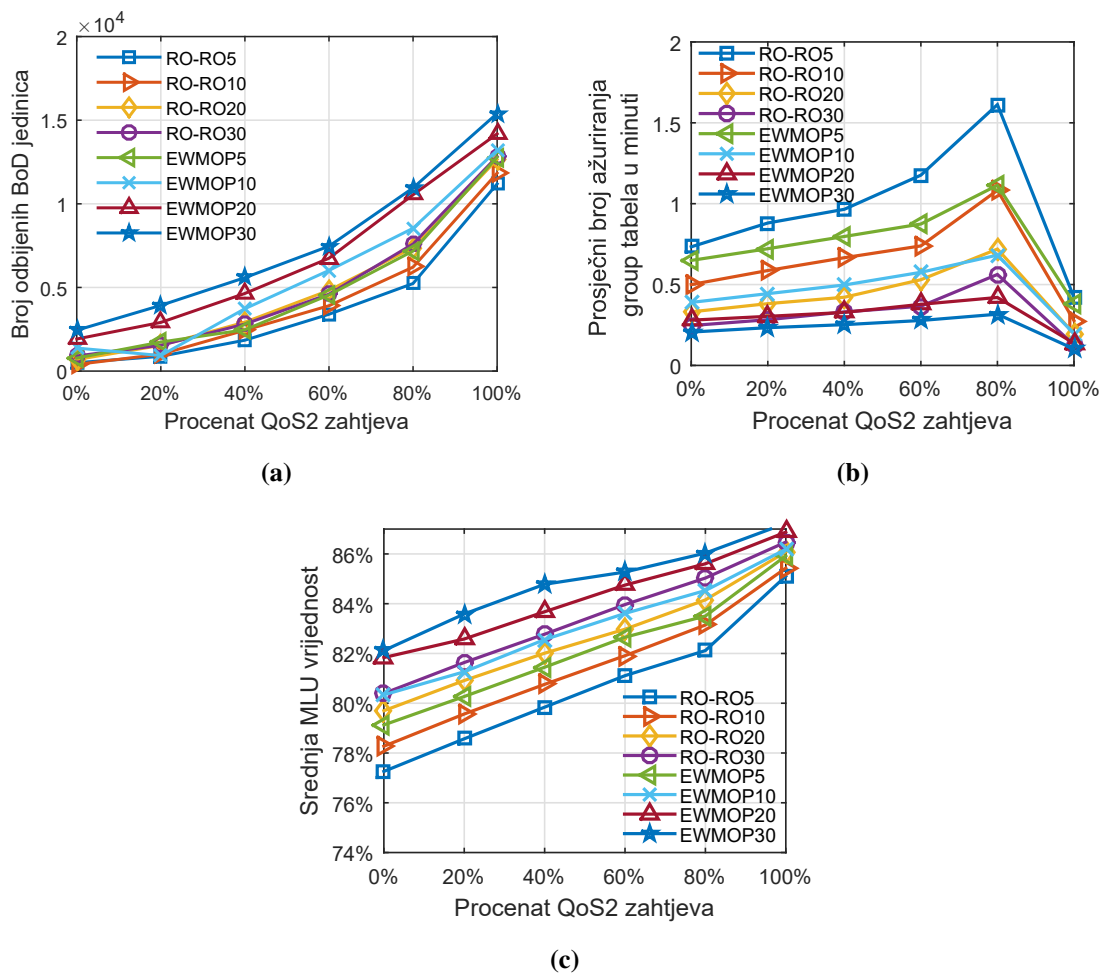
Na Slici 5.4 prikazan je uticaj periode optimizacionih ciklusa (T) na performanse RORO-H pristupa. Može se primijetiti da povećanje vrijednosti T parametra sa 5 na 10 minuta ne utiče značajnije na srednju MLU vrijednost i broj odbijenih jedinica BoD zahtjeva, dok doprinosi smanjenju broja rekonfiguracija. Sa druge strane, pri konfiguraciji $T=30$ minuta, predložena heuristika u nekim od simuliranih saobraćajnih scenarija pokazuje lošije performanse od SDN-LB tehnike koja minimizuje MLU u intervalima od 1h. Generalno, povećanje T parametra smanjuje učestanost rekonfiguracija pri istom R_{avg} ograničenju, i povećava prosječnu MLU vrijednost. Na Slici 5.5 prikazan je uticaj parametra T na performanse RO-RO i EWMOP pristupa. Na osnovu dobijenih rezultata može se zaključiti da RO-RO optimizacioni model ostvaruje poboljšanje performansi i pri relativno velikim vrijednostima T parametra. Iako 5.5b sugerise da je EWMOP pristup pogodniji sa aspekta broja rekonfiguracija, to se u ovoj analizi ne može smatrati prednošću jer RO-RO održava srednju učestanost rekonfiguracija u skladu sa graničnom vrijednošću koju definiše mrežni operator. Ovdje je bitno napomenuti da *drift-plus-penalty* funkcija (5.11) ne garantuje da će prosječna učestanost rekonfiguracija biti ispod R_{avg} . Međutim, uz odgovarajuću vrijednost A parametra, koja je korišćena u simulacionoj analizi, srednja učestanost rekonfiguracija nikada ne odstupa previše od R_{avg} jer *drift* funkcija naglo raste

kada RO-RO optimizator vrši više od $R_{avg} \cdot T$ rekonfiguracija.

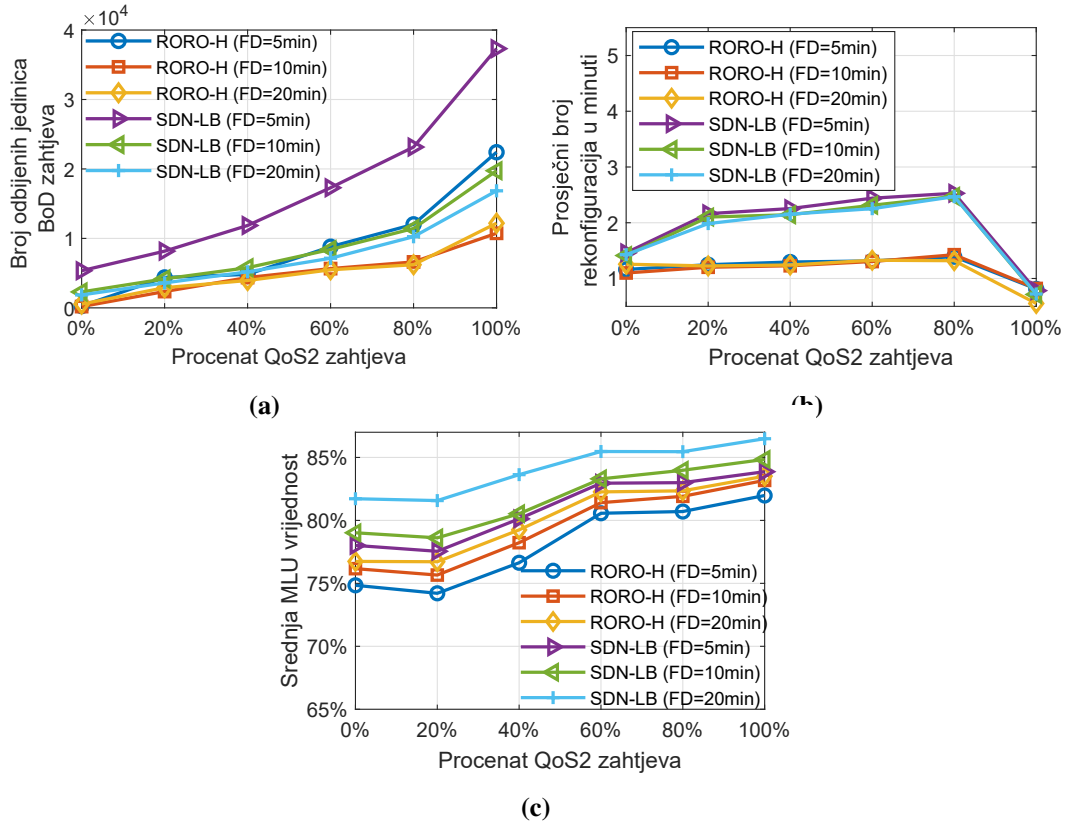


Slika 5.4: Uticaj T parametra na performanse RORO-H pristupa. Sufiksi u legendi indiciraju vrijednost T parametra koja je korišćena u simulacijama.

U sprovedenoj analizi razmatran je i uticaj prosječnog trajanja generisanih tokova (BoD zahtjeva) na performanse SDN-LB ($T=1h$) i RORO-H ($T=10min$) pristupa. Kao što se može vidjeti sa Slike 5.6, performanse oba pristupa se degradiraju kako se prosječno trajanje toka smanjuje u odnosu na periodu izvršavanja TE optimizacija. Međutim, razlika u performansama između SDN-LB i RORO-H povećava se u scenarijima sa kraćim tokovima. Dobijeni rezultati ističu značaj dinamičkih TE optimizacija u uslovima varijabilnog saobraćajnog opterećenja.



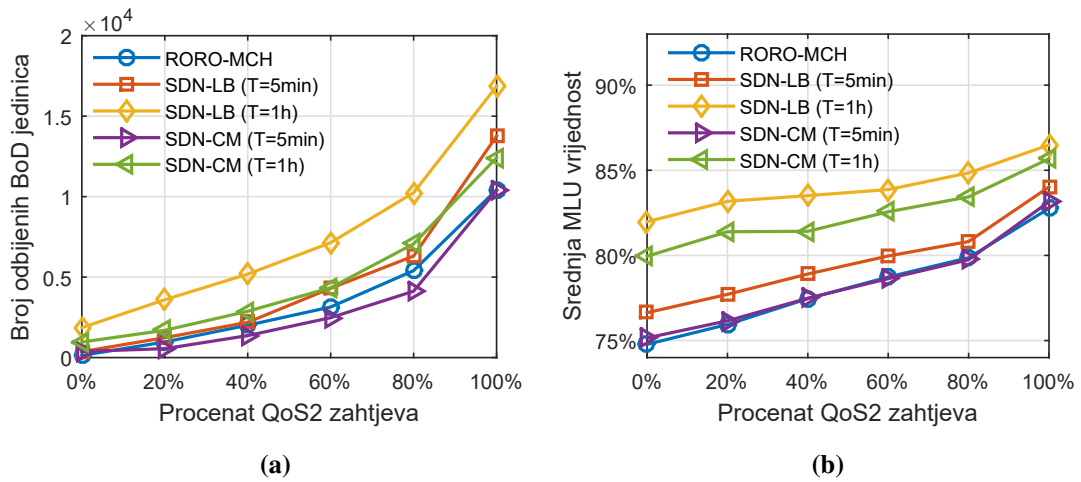
Slika 5.5: Uticaj T parametra na performanse RORO i EWMOP TE pristupa.
Sufiksi u legendi indiciraju korišćenu vrijednost T parametra.



Slika 5.6: Uticaj prosječnog trajanja toka (FD - *Flow Duration*) parametra na performanse RORO-H i SDN-LB pristupa.

5.4.2 Performanse RORO-MC pristupa u uslovima odsustva QoS3 saobraćaja

TE rešenja koja teže da optimizuju servisni kapacitet mreže minimizacijom MLU metrike imaju nekoliko nedostataka. U ovoj sekciji analizirano je da li se minimizacijom cijene rutiranja mogu poboljšati performanse RO-RO optimizatora (RORO-MC pristup). RORO-MC heuristika upoređena je sa SDN-LB pristupom u pogledu odbijenih jedinica BoD zahtjeva i prosječne MLU vrijednosti na Slici 5.7. U cilju fer poređenja, analizirane su performanse TE tehnike koja periodično minimizuje optimizacionu funkciju (5.12a). Rezultati za ovu tehniku predstavljeni su pod nazivom SDN-CM (SDN *with Cost Minimization*). Sa Slike 5.7 može se primijetiti sličnost u performansama RORO-MCH i SDN-CM ($T=5\text{min}$). U odnosu na SDN-LB ($T=5\text{min}$), RORO-MCH odbija značajno manje BoD zahtjeva. Rezultati u pogledu učestanosti rekonfiguracija dati su u Tabeli 5.4. Za isto T , RORO-MCH smanjuje rekonfiguracioni *overhead* do 14 puta u poređenju sa SDN-LB pristupom, i do 25 puta u poređenju sa SDN-CM.



Slika 5.7: Odbijene jedinice BoD zahtjeva i srednji MLU u simulacionoj analizi performansi RORO-MCH pristupa.

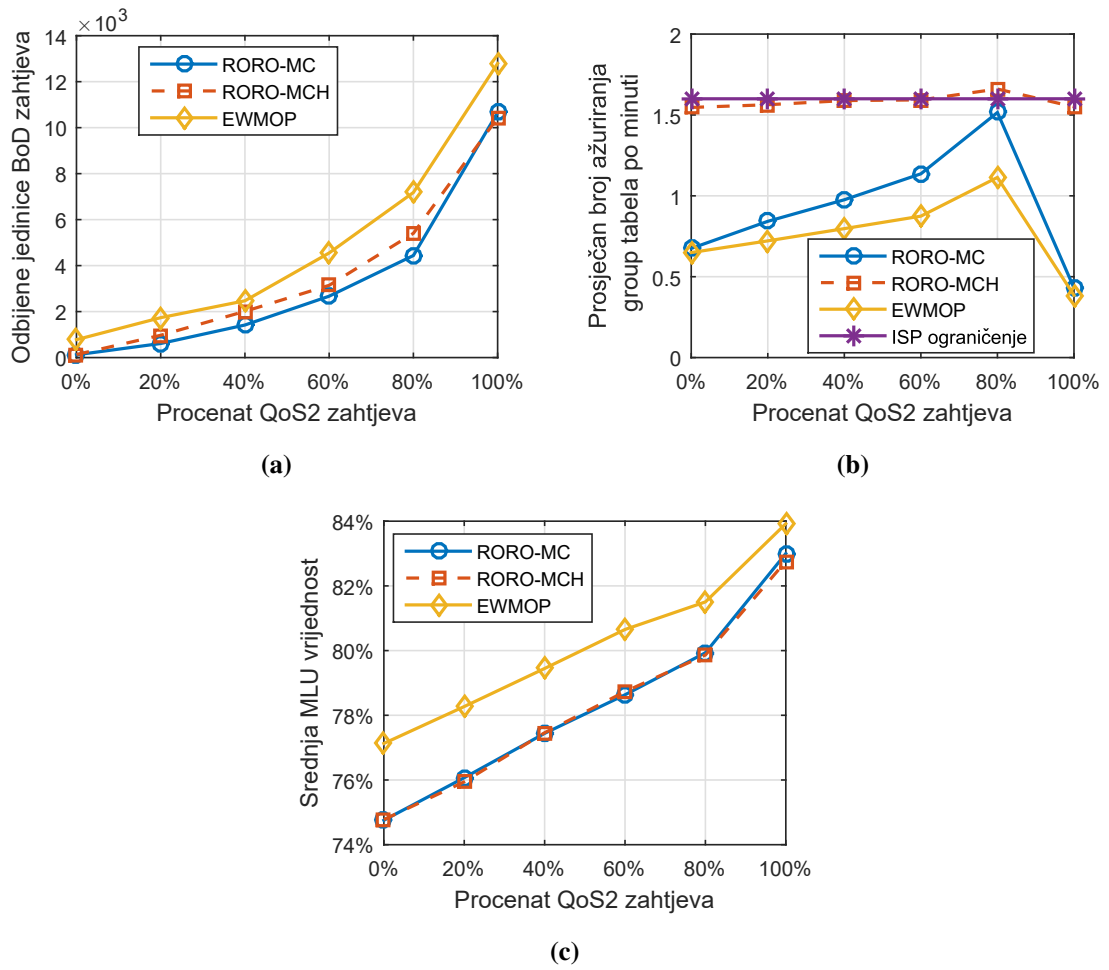
Tabela 5.4: Poređenje RORO-MCH pristupa sa konkurentskim TE rešenjima u pogledu prosječnog broj rekonfiguracija u minuti.

QoS2 %	RORO-MCH (T=5min)	SDN-LB (T=5min)	SDN-LB (T=1h)	SDN-MC (T=5min)	SDN-MC (T=1h)
0%	1.55	15.44	1.43	24.14	2.08
20%	1.56	21.51	1.98	37.74	3.34
40%	1.59	21.69	2.15	39.63	3.46
60%	1.59	22.13	2.25	40.06	3.62
80%	1.66	22.37	2.47	38.58	3.63
100%	1.55	7.14	0.72	18.26	1.63

Na Slici 5.8 prikazani su rezultati poređenja RORO-MC i EWMOP pristupa. Rezultati potvrđuju prednosti RORO-MC pristupa u pogledu servisnog kapaciteta i u pogledu prosječne MLU vrijednosti. Takođe, interesantno je primijetiti da se poboljšanja performansi ostvaruju i sa RORO-MC heuristikom (RORO-MCH).

Uticaj T parametra na ponašanje RORO-MCH analiziran je na Slici 5.9. Za razliku od RORO-H, RORO-MCH nije osjetljiv na promjene T . Međutim, manje vrijednosti T parametra se mogu smatrati poželjnijim zbog mogućnosti efikasnijeg reagovanja na iznenadne distorzije saobraćaja.

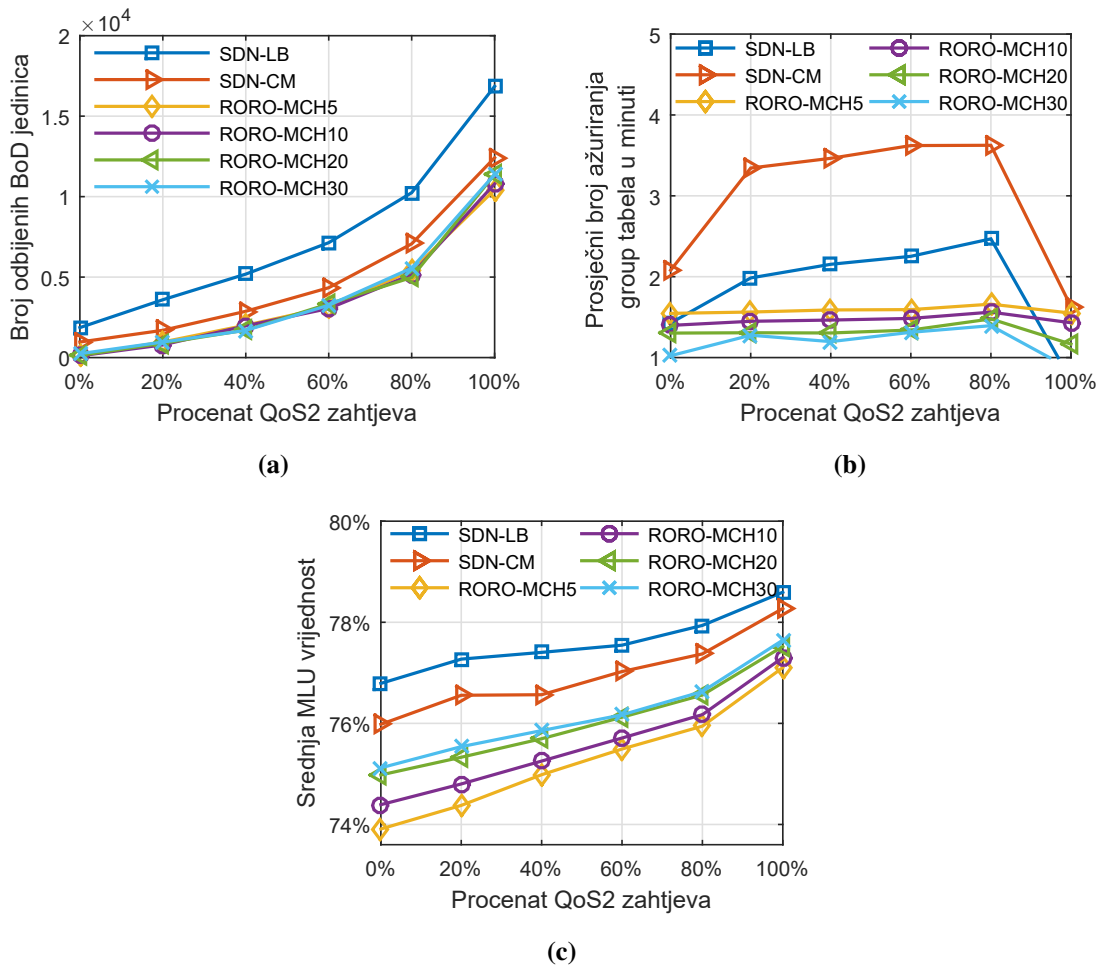
Na Slici 5.10 predstavljeni su simulacioni rezultati za tri različite vrijednosti R_{avg} ograničenja: 1.2, 1.6 i 2. Kao što je očekivano, sa porastom R_{avg} poboljšava se iskori-



Slika 5.8: Poređenje RORO-MC i EWMOP optimizacionih modela.

šćenost mrežnih resursa. Ovo se može pripisati činjenici da kontroler ima slobodu da izvršava rekonfiguracije mrežnih parametara češće. U simulacionim scenarijima gdje R_{avg} uzima malu vrijednost, gubici u pogledu odbijenih jedinica BoD zahtjeva su očigledni za RORO-MCH. Međutim, Slika 5.10b potvrđuje da se ovo dešava kao posledica prilagođavanja kontrolnih odluka ograničenju u pogledu srednje učestanosti rekonfiguracija. Prosječni broj rekonfiguracija u minuti RORO-MCH optimizatora je u najgorem slučaj vrlo malo iznad R_{avg} praga.

Računska kompleksnost RORO-MCH algoritma zavisi od veličine mreže. Međutim, kompleksnost algoritma je moguće kontrolisati ograničavanjem broja LP iteracija - N_{max} . U Tabeli 5.5 su predstavljeni rezultati analize uticaja N_{max} i R_{avg} parametara na vrijeme izvršavanja RORO-MCH algoritma. Rezultati su dobijeni na Windows mašini sa 16GB RAM-a i i7 procesorom. Očigledno, vrijeme izvršavanja će povećava sa brojem iteracija.

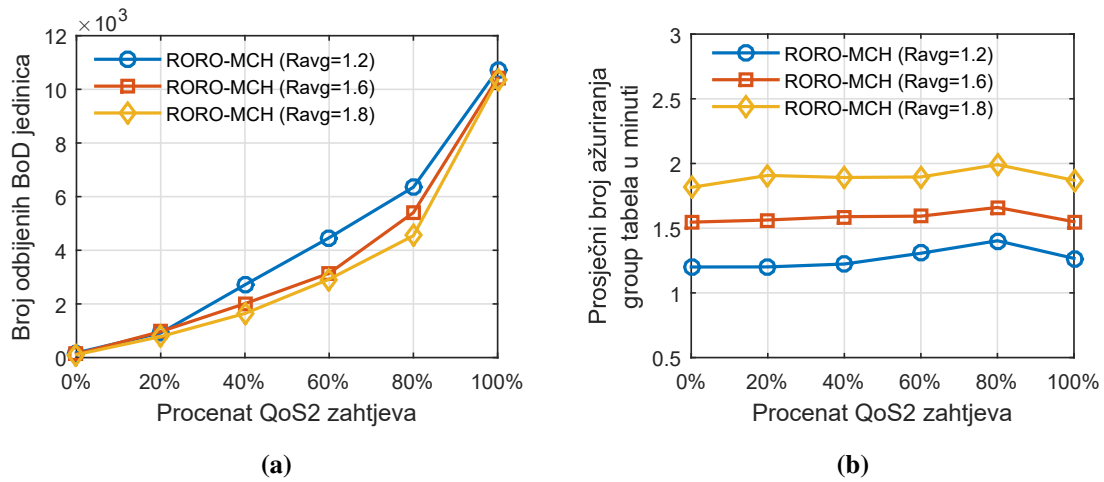


Slika 5.9: Uticaj T parametra na performanse RORO-MCH pristupa.

Stoga, od velike je važnosti odrediti efikasnu, kompromisnu vrijednost za N_{max} . Iz Tabele 5.5 se može primijetiti da uticaj broja iteracija algoritma na broj odbijenih jedinica BoD zahtjeva slabi sa povećanjem R_{avg} . U svim analiziranim scenarijima, 100 iteracija ($N_{max} = 100$) je bilo dovoljno da RORO-MCH dostigne skoro optimalne performanse. Pri ovoj konfiguraciji, vrijeme izvršavanja RORO-MCH algoritma iznosilo je oko 15s u najgorem slučaju.

5.4.3 Performanse RORO-MC pristupa u uslovima sa QoS3 saobraćajem

S obzirom da je prethodno pokazano da RORO-MC doprinosi poboljšanju servisnog kapaciteta mreže u odnosu na RO-RO algoritam inženjeringa saobraćaja, simulaciona



Slika 5.10: Uticaj R_{avg} ograničenja na performanse RORO-MCH pristupa u uslovima bez QoS3 saobraćaja.

Tabela 5.5: Uticaj parametra N_{max} na vrijeme izvršavanja (CT - *Computational Time*) i odbijene jedinice BoD zahtjeva (RB - *Rejected Bandwidth*).

Nmax	RORO-MCH (Ravg=1.6)		RORO-MCH (Ravg=4)		RORO-MCH (Ravg=6)	
	CT	RB	CT	RB	CT	RB
10	1.94s	12767	2.18s	3560	2.33s	2454
20	3.14s	11868	3.57s	2753	3.9s	2509
100	12.84s	3142	12.83s	2546	15.64s	2485
150	16.78s	2920	19.36s	2421	23.19s	2423

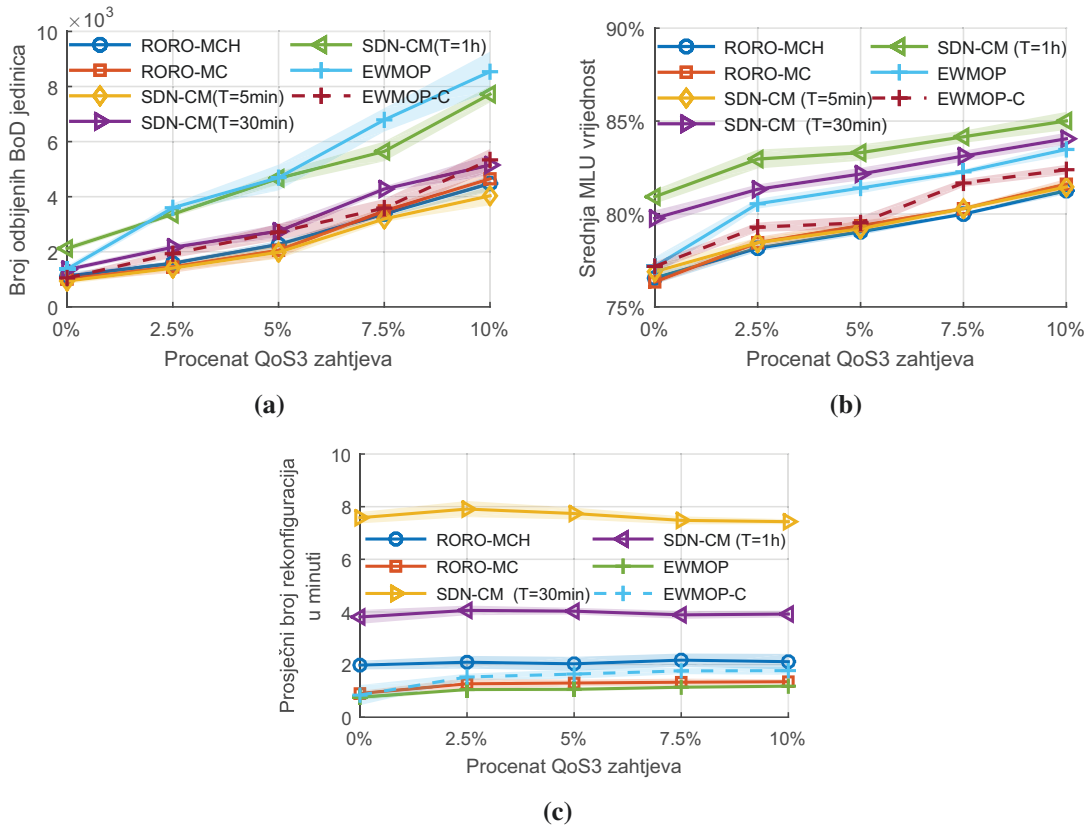
analiza je zaključena evaluacijom performansi RORO-MC pristupa u uslovima kada se generišu sve tri klase BoD zahtjeva. S tim u vezi, izvršeno je poređenje sa SDN-CM i EWMOP TE algoritima pri zadatom ograničenju učestanosti rekonfiguracija od $R_{avg}=3$ rekonfiguracije po minuti. U odnosu na analizu iz prethodnih sekcija ovog poglavlja, R_{avg} ograničenje je povećano iz razloga što uvođenje QoS3 klase saobraćaja utiče na porast rekonfiguracionog *overhead*-a. Vrijednost je odabrana tako da bude manja od učestanosti rekonfiguracija koju nameće SDN-CM ($T=1h$) pristup. Procenat DS zahtjeva (QoS2 i QoS3) u ukupnom broju generisanih BoD zahtjeva fiksiran je na 50%. Intenzitet saobraćaja je smanjen u odnosu na prethodno razmatrani saobraćajni scenario jer QoS3 tokovi zahtijevaju rezervaciju dodatnih kapaciteta u mreži. Koeficijent preklapanja tunela IO para na nivou linka (q) definisan je odvojeno za svaku klasu. Za QoS1 i QoS2 klasu zadržan je koeficijent $q = 2$, dok je rutiranje QoS3 tokova ograničeno samo na nepreklapajuće tu-

nele ($q = 1$), ukoliko nije drugačije naznačeno. QoS3 tokovima garantovana je robustnost na otkaz jednog linka ($n = 1$). Dobijeni rezultati prikazani su na Slici 5.11 sa intervalom pouzdanosti od 95 %.

Kao što se može vidjeti sa Slike 5.11, primjenom RORO-MC algoritma ostvaruje se približno isti servisni kapacitet kao sa SDN-CM ($T=5\text{min}$) algoritmom, pri čemu je broj rekonfiguracija značajno smanjen. Rekonfiguracioni *overhead* SDN-CM algoritma u svim simulacijama je bio neprihvatljivo veliki za bilo kakvo praktično razmatranje (42-44 rekonfiguracija u minuti), i kao takav nije prikazan na Slici 5.11b radi bolje vidljivosti ostalih rezultata. U odnosu na EWMOP, RORO-MCH odbija značajno manje BoD zahtjeva. Slabe performanse EWMOP algoritma velikim dijelom se mogu pripisati činjenici da se balansiranje saobraćaja optimizuje minimizacijom MLU metrike. U cilju sticanja jasnijeg uvida u druge aspekte EWMOP algoritma, na Slici 5.11 prikazani su i rezultati ostvareni modifikovanom verzijom EWMOP algoritma (označeni sa EWMOP-C) koja umjesto optimizacione funkcije za minimizaciju MLU koristi funkciju (5.12a). U odnosu na EWMOP, EWMOP-C odbija manji broj BoD zahtjeva. Međutim, njegove performanse su i dalje lošije od računski značajno jednostavnijeg RORO-MCH algoritma. Ovi rezultati sugerišu ograničenja TE pristupa koji daje statičke i jednake prioritete optimizacionim funkcijama za minimizaciju cijene rutiranja i *overheada* rekonfiguracije.

Analizirana TE rešenja upoređena su u pogledu prosječne MLU vrijednosti na Slici 5.11b. Približno isti rezultati ostvareni su sa RORO-MC, RORO-MCH i SDN-CM algoritmima pri istoj učestanosti optimizacionih ciklusa. Stoga, može se zaključiti da je optimizacijom težinskih faktora za balansiranje „kritičnih” tokova moguće postići približno optimalnu iskorišćenost resursa.

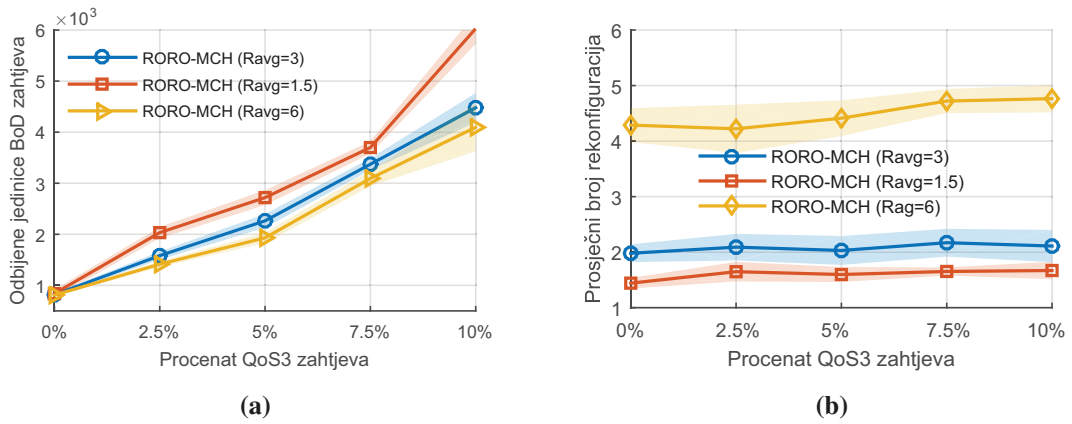
Na Slici 5.12 analiziran je uticaj R_{avg} ograničenja na performanse RORO-MCH algoritma. Simulacije su odrađene za tri različita R_{avg} ograničenja: 1.5, 3 i 6 rekonfiguracija po minuti. Kao i u scenariju bez QoS3 saobraćaja, broj odbijenih BoD zahtjeva se smanjuje kada mreža može da podrži veći broj rekonfiguracija (tj. sa povećanjem R_{avg}). Prosječna učestanost rekonfiguracija u scenarijima $R_{avg} = 3$ i $R_{avg} = 6$ je značajno ispod granične vrijednosti, jer dalje povećanje broja rekonfiguracija ne doprinosi značajnijem poboljšanju cijene rutiranja. Manje od 5 rekonfiguracija po minuti, u prosjeku, je bilo dovoljno za realizaciju skoro optimalnih performansi. Prosječna učestanost rekonfiguracija za RORO-MCH ($R_{avg} = 3$) iznosila je oko 2 rekonfiguracije po minuti, iako je sa Slike 5.12 evidentno da postoji prostor za poboljšanje cijene rutiranja. Ovaj rezultat je najvjerojatnije posljedica toga što poboljšanje cijene rutiranja nije bilo moguće ostvariti sa manje



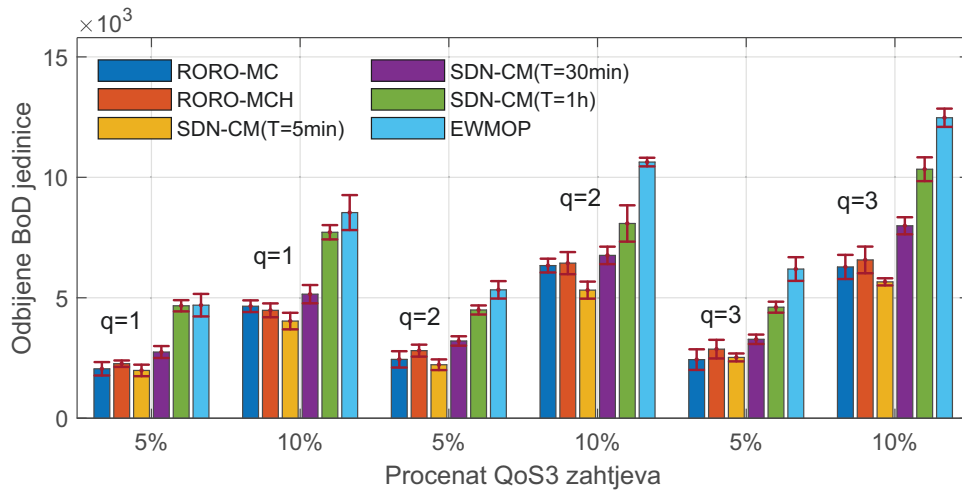
Slika 5.11: Performanse analiziranih TE algoritama u funkciji procenta QoS3 zahtjeva u ukupnom generisanom saobraćajnom opterećenju.

od 3 rekonfiguracije u minuti. RORO-MCH ($R_{avg} = 1.5$) probija ograničenje mrežnog operatora u uslovima velikog saobraćajnog opterećenja. Međutim, kao što se vidi sa Slike 5.12, odstupanje od R_{avg} je relativno malo.

Garantovanje robustnosti na otkaz linkova je računski kompleksan zadatak. Kao što je već objašnjeno u Poglavlju 4, da bi se smanjila ova kompleksnost neophodno je definisati ograničenje u pogledu maksimalno dozvoljenog preklapanja tunela IO para na nivou linka (q). Stoga, razmatran je fundamentalni problem izbora adekvatne q vrijednosti za QoS3 saobraćaj. Veće q generalno znači veći diverzitet ruta, tj. više dostupnog kapaciteta. Međutim, u razmatranom scenariju takođe povlači da više resursa mora biti rezervisano za QoS3 tokove u cilju garancije robustnosti na otkaz linkova. Stoga, usvojeno je kompromisno rešenje koje koristi dvije vrijednosti q parametra: jednu za QoS3 tokove, i jednu za QoS1 i QoS2 tokove (q'). Analiza je pokazala da vrijednosti q' parametra veće od 2 ne doprinose značajno poboljšanju servisnog kapaciteta mreže. Uticaj q parametra na broj odbijenih BoD zahtjeva i prosječni MLU ilustrovan je na Slikama 5.13-5.14. Pretposta-

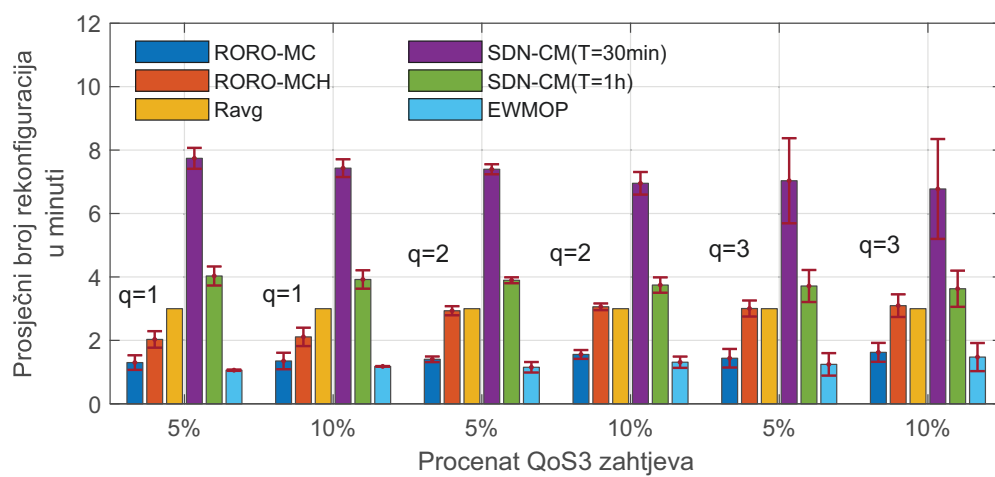


Slika 5.12: Uticaj R_{avg} ograničenja na performanse RORO-MCH pristupa.



Slika 5.13: Uticaj q parametra na broj odbijenih BoD zahtjeva.

vljeno je da svi analizirani TE algoritmi koriste isti mehanizam zaštite QoS3 tokova od kvarova u mreži. Najbolje performanse su ostvarene sa $q = 1$, tj. kada su nepreklapajući tuneli korišćeni za rutiranje QoS3 saobraćaja između IO parova. Stoga, može se zaključiti da efikasnost korišćenja mrežnih resursa dominantno zavisi od količine dodatnih resursa koji se rezervišu za QoS3 tokove u cilju očuvanja kvaliteta servisa u slučaju otkaza nekog od linkova na primarnim rutama. U prilog predloženom rešenju ide činjenica da je broj nepreklapajućih tunela u realnim ISP topologijama relativno veliki.



Slika 5.14: Uticaj q parametra na prosječnu učestanost rekonfiguracija.

Glava 6

Eksperimentalni rezultati

SDN predstavlja tehnologiju u razvoju, pa u vrijeme pisanja ove doktorske disertacije primjeri efikasnih, skalabilnih i inteligentnih aplikacija za upravljanje saobraćajem nisu javno dostupni. U ovom poglavlju predstavljen je prototip kontrolnog okruženja za inženjering saobraćaja u ISP mrežama. Detalji implementacije prototipa objašnjeni su u Sekciji 6.1. U Sekciji 6.2 predstavljeni su eksperimentalni rezultati evaluacije performansi predloženog RORO-MCH TE algoritma. Rezultati potvrđuju zaključke simulacione analize kao i izvodljivost predloženog TE pristupa sa postojećim tehnologijama i protokolima.

6.1 Prototip SDN kontrolnog okruženja za inženjering saobraćaja u ISP mrežama

Jedan od glavnih ciljeva SDN kontrolnog okruženja predloženog u Poglavlju 4 je omogućavanje visoko-granularne diferencijacije saobraćaja i pružanje fleksibilnog BoD servisa konekcijama koje imaju QoS zahtjeve. Tradicionalna mrežna infrastruktura nije pogodna za ove namjene zbog svoje statičke i kompleksne prirode. Ovo je posledica distribuirane kontrolne ravni, koja otežava automatizaciju kontrolnih procesa. Kao rezultat toga, mreža nije u stanju da se dinamički prilagođava promjenjivim zahtjevima korisnika i aplikacija.

Predloženi dizajn SDN kontrolera prati arhitekturu sa Slike 4.1. Na Slici su istaknuti sledeći ključni elementi kontrolera:

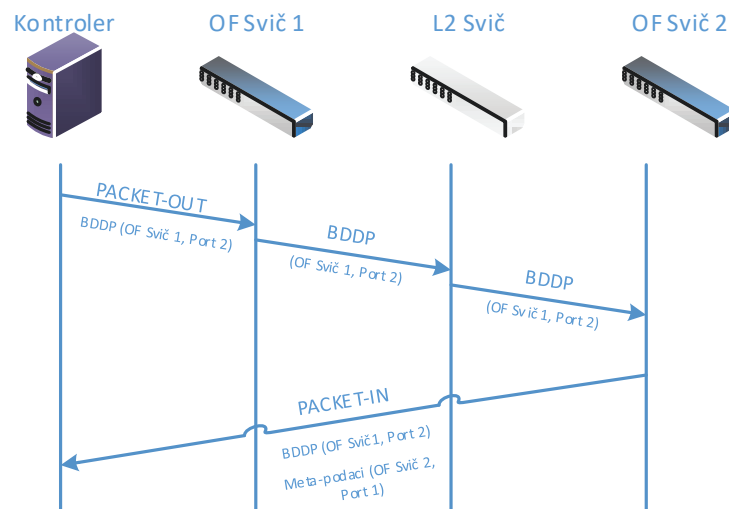
- Monitoring resursa
- API za BoD servis i instalaciju ruta
- Moduli za *online* rutiranje i *offline* TE

U nastavku poglavlja biće diskutovana implementacija svake od navedenih funkcionalnosti pojedinačno.

6.1.1 Monitoring resursa

Modul za monitoring resursa odgovoran je za prikupljanje i održavanje informacija o trenutnom stanju u mreži. Jedan od fundamentalnih zadataka ovog modula je detekcija topologije, što podrazumijeva detekciju OpenFlow svičeva, linkova, i hostova u mreži. Kada OpenFlow svič uspostavi TCP konekciju sa kontrolerom, kontroler šalje *Feature-request* poruku sviču i čeka odgovor. Iz poruke odgovora, kontroler izvlači informacije o karakteristikama sviča, kao što su identifikator sviča (*datapath ID*), lista portova, itd. Detekcija linkova u mreži vrši se posredstvom LLDP (*Link Layer Discovery Protocol*) i BDDP (*Broadcast Domain Discovery Protocol*) paketa. LLDP paketi se prepoznaju u mreži na osnovu specijalne MAC adrese i vrijednosti 0x08cc u *EtherType* polju zaglavlja. Proces detekcije topologije inicira kontroler, koji enkapsulira LLDP pakete u *Packet-out* poruke, i periodično ih šalje svim povezanim OpenFlow svičevima. *Packet-out* poruka sadrži instrukcije za prosleđivanje primljenih LLDP paketa preko odgovarajućeg interfejsa. Kada susjedni svič primi paket ovog tipa, vrši pretraživanje tabele tokova. Kako za LLDP pakete ne postoji odgovarajući zapis, paket se šalje kontroleru kao dio *Packet-in* poruke koja je definisana OpenFlow specifikacijom. Analizirajući *Packet-in* poruke sa LLDP paketima, kontroler može uspješno identifikovati linkove između direktno povezanih OpenFlow svičeva. Ove informacije su konstantno dostupne ostalim modulima koji se izvršavaju na kontroleru.

Floodlight kontroler nudi i mogućnost detekcije *multi-hop* linkova između OpenFlow svičeva u hibridnim mrežama koje se sastoje od tradicionalnih i OpenFlow svičeva. Ova funkcionalnost bazirana je na slanju BDDP paketa. Za razliku od LLDP paketa koji imaju specijalnu *multicast* MAC adresu, BDDP paketi koriste *broadcast* adresu, pa se mogu prosleđivati i tradicionalnim svičevima koji pripadaju istom *broadcast* domenu kao



Slika 6.1: Princip detekcije topologije razmjenom BDDP poruka.

i OpenFlow svičevi. Druga bitna razlika u odnosu na LLDP pakete je vrijednost *EtherType* polja, koja je kod BDDP paketa 0x8999. Kontroler periodično šalje BDDP pakete svim povezanim OpenFlow svičevima. Po jedan paket namijenjen je svakom od izlaznih interfejsa sviča. Informacije o identifikatoru sviča i interfejsa kojem je paket namijenjen sadržani su u *payload*-u paketa. OpenFlow svič prepoznaje BDDP pakete na osnovu specifične vrijednosti *EtherType* polja, i prosleđuje ih na odgovarajući izlazni interfejs. Ukoliko susjedni svič podržava OpenFlow protokol, odmah će prosljediti paket kontroleru kao dio *Packet-in* poruke. U slučaju da je susjed tradicionalni L2 svič, prosleđivanje će biti izvršeno u skladu sa destinacionom MAC adresom (ff:ff:ff:ff:ff:ff), tj. paket će biti poslat na sve izlazne interfejse. Na ovaj način, paketi dolaze do drugih OpenFlow svičeva u istom *broadcast* domenu, i vraćaju se kontroleru u formi *Packet-in* poruke, slično kao što je to slučaj sa LLDP paketima. *Packet-in* poruka sadrži meta-podatke koji su potrebni kontroleru za detekciju *multi-hop* linka. Ovaj postupak ilustrovan je na Slici 6.1.

LLDP i BDDP paketi se mogu iskoristiti i za estimaciju kašnjenja na linkovima. Na Floodlight kontroleru implementirana je funkcija koja u LLDP i BDDP pakete upisuje *timestamp* sa vremenom slanja paketa. Kada OpenFlow svič sa druge strane linka vrati LLDP/BDDP paket kontroleru, na osnovu razlike vremena prijema paketa i upisane *timestamp* vrijednosti moguće je izračunati ukupno vrijeme prenosa paketa. Ovo vrijeme uključuje i kašnjenje u komunikaciji između svičeva i kontrolera. Međutim, Floodlight kontroler posjeduje mehanizme za estimaciju kašnjenja u komunikaciji sa svičevima, pa se na ovaj način jednostavno estimira i kašnjenje na linkovima.

Ukoliko informacije o povezanim podmrežama nisu unaprijed poznate, detekcija lokacija hostova se vrši reaktivno. Kada host prvi put generiše saobraćaj u mrežu, pod pretpostavkom da OpenFlow svičevi nemaju instrukcije za obradu njegovih paketa, paketi će biti proslijeđeni kontroleru. Kontroler zatim mapira adresu hosta sa identifikatorom OpenFlow sviča i porta na koji je paket stigao, i instalira potrebne rute u mreži.

6.1.2 API za BoD servis i instalaciju tunela

Na kontroleru je implementiran REST (*Representational State Transfer*) interfejs za prihvatanje BoD zahtjeva. BoD zahtjevi definišu relevantne parametre konekcije, kao što su: tip BoD servisa (QoS1, QoS2 ili QoS3), IO par i trajanje zahtjeva. U eksperimentima je pretpostavljeno da svaki OpenFlow svič služi za pristup unaprijed definisanom opsegu IP adresa, koji je definisan *inter-area* politikom rutiranja. S obzirom da korisnici nisu svjesni identifikatora svičeva, IO parovi su indicirani adresom izvorišne i odredišne mreže između kojih je potrebno uspostaviti konekciju (virtuelni link) sa garantovanim kvalitetom servisa.

Funkcija za uspostavljanje tunela između IO parova takođe je implementirana u formi REST modula. Prilikom prijema zahtjeva koji specificira K i q parametre, ovaj modul instalira K q -nepreklapajućih ruta između svakog IO para. Za računanje ruta koristi se *Yen*-ov algoritam [127].

6.1.3 Obrada BoD zahtjeva

BoD zahtjevi prosleđuju se modulu za *online* rutiranje. Kao što je objašnjeno u Poglavlju 4, ovaj modul prvo provjerava da li se zahtjev može ispuniti bez instalacije dodatnih zapisa u tabelama tokova, oslanjajući se na trenutnu šemu balansiranja opterećenja. Ukoliko na taj način nije moguće ispuniti QoS zahtjev, modul za *online* rutiranje bira jednu od instaliranih ruta za prenos svih tokova koji se odnose na pristigli zahtjev, pod uslovom da ruta može da pruži potrebni QoS.

Offline TE modul implementiran je tako da se izvšava u okviru posebne Java niti. Balansiranje opterećenja između IO parova realizovno je pomoću *SELECT group* tabela na ulaznim svičevima. Format tabele tokova za svičeve na ulazu, izlazu i jezgru mreže, ilustrovan je na Slici 6.2, na primjeru jednog IO para. U tabelama tokova ulaznih svičeva

a) Tabela tokova ulaznog sviča

Cookie	Match fields	Actions
0	Eth_type=0x0800, IP_src, IP_dst, ToS=0x10	Go to Group X
0	Eth_type=0x0800, IP_src, IP_dst, ToS=0x08	Go to Group Y
0	Eth_type=0x0800, IP_src, IP_dst, ToS=0x04	Go to Group Z
10	Eth_type=0x0800, IP_src, IP_dst, ToS=0x04	set_VlanID =2100, set_queue=1, out_port=1

Tri grupe za
svaki IO par

Privremeno pravilo
instalirano od strane
Online modula za
rutiranje

b) Group tabela ulaznog sviča

ID	Type	Bucket 0 (B0)	B0 weight	B0 watch-port	Bucket 1 (B1)	B1 weight	B1 watch-port	Bucket 2 (B2)	B2 weight	B2 watch-port
X	SELECT	Set VlanID = 2100, set_queue=1, outPort=1	10	1	Set VlanID = 2101, set_queue=1, outPort=2	40	2	Set VlanID = 2102, set_queue=1, outPort=3	50	3
Y	SELECT	Set VlanID = 2100, set_queue=1, outPort=1	40	1	Set VlanID = 2101, set_queue=1, outPort=2	60	2	Set VlanID = 2102, set_queue=1, outPort=3	0	3
Z	SELECT	Set VlanID = 2100, set_queue=1, outPort=1	100	1	Set VlanID = 2101, set_queue=1, outPort=2	0	2	Set VlanID = 2102, set_queue=1, outPort=3	0	3

Jedan bucket prema
svakom od *offline*
instaliranih tunela

c) Tabela tokova sviča u jezgru mreže

Cookie	Match fields	Actions
0	Eth_type=0x0800, IP_src, IP_dst, VlanID=2101	set_queue=1, out_port=1

Statički zapis

d) Tabela tokova izlaznog sviča

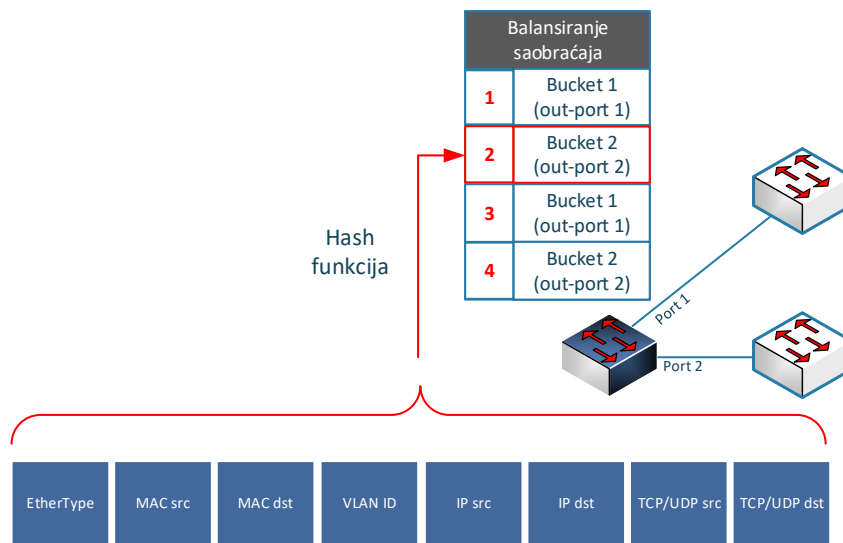
Cookie	Match fields	Actions
0	Eth_type=0x0800, IP_src, IP_dst, VlanID=2101	strip_VlanID, out_port=1

Statički zapis

Slika 6.2: Primjer konfiguracije tabele tokova i *group* tabele za svič na ulazu u mrežu (a-b), u jezgru mreže (c) i izlazu mreže (d).

su konfigurisana po tri statička zapisa za QoS saobraćaj prema svakom izlaznom sviču. Zapisi u tabeli identifikuju IO par za saobraćajni tok na osnovu destinacione IO adrese, i usmjeravaju pakete toka u odgovarajuću *group* tabelu (zapisi koje instalira *offline* TE modul) ili direktno na odgovarajući izlazni interfejs kao rezultat *online* rutiranja. Za mapiranje saobraćajnih tokova sa QoS klasama korišćeno je *Type of Service* (ToS) polje u IP zaglavlju, na sledeći način:

- ToS 0x10 -> QoS1
- ToS 0x08 -> QoS2
- ToS 0x04 -> QoS3



Slika 6.3: Ilustracija odabira *bucket*-a na osnovu *hash* funkcije.

Svaka *group* tabela ima jedinstven identifikator (na nivou sviča) i sadrži set *bucket*-a koji definišu akcije koje je potrebno izvršiti nad pristiglim paketima. Tip akcije zavisi od tipa tabele. OpenFlow protokol za sada definiše četiri tipa *group* tabele: ALL, SELECT, INDIRECT i FAST-FAILOVER [7]. U predložem modelu sistema, SELECT *group* tabele su korišćene za balansiranje saobraćaja. Kao što je prikazano na Slici 6.2b, svakom *bucket*-u SELECT *group* tabele pridružen je težinski faktor, koji omogućava konfigurisanje neuniformnog balansiranja opterećenja između *bucket*-a. Dio saobraćaja koji je namijenjen određenom *bucket*-u definisan je udjelom težinskog faktora tog *bucketa* u sumi svih težinskih faktora *group* tabele [7]. Algoritam za izbor *bucket*-a zavisi od modela sviča. U eksperimentima čiji su rezultati predstavljeni u nastavku sekcije korišćeni su OpenvSwitch-evi [106], koji računaju jednostavnu *hash* funkciju nad skupom unaprijed definisanih polja u zaglavlju paketa, i dodjeljuju pakete *bucket*-ima na osnovu izračunate vrijednosti i konfigurisanih težinskih faktora. *Hash* funkcija je simetrična u smislu da zamjena izvorišne i odredišne adrese u paketu ne utiče na izbor *bucket*-a. U slučaju da OpenFlow verzija 1.5 (ili novija) nije podržana na kontroleru, OpenvSwitch-evi kao argument *hash* funkcije koriste samo IP i MAC adrese. Kako je to slučaj sa Floodlight kontrolerom, za potrebe eksperimenata modifikovan je izvorni kod OpenvSwitch-a tako da *hash* funkcija uključuje L4 polja, kao što je ilustrovano na Slici 6.3. Na ovaj način, stvoreni su uslovi za diferenciranje saobraćajnih tokova čak i kada su oni generisani između istog para hostova.

Svaki *bucket* *group* tabele obavlja različite akcije. U razmatranom scenariju, broj *buc-*

ket-a u *group* tabeli odgovara broju ruta instaliranih između IO para za odgovarajuću klasu saobraćaja. Osnovna funkcija svakog *bucket*-a je prosleđivanje paketa na jednu od instaliranih ruta. Pored ove akcije, sprovode se još dvije: markiranje saobraćaja i izbor bafera. S obzirom da rute između IO parova nisu obavezno nepreklapajuće, svakoj od njih pridružuje se specifična labela kako bi se izbjeglo kreiranje konfliktnih zapisa u tabelama tokova u jezgru mreže. Na primjer, ukoliko dvije rute imaju zajednički segment, a OpenFlow svičevi u jezgru mreže vrše prosleđivanje saobraćaja samo na osnovu zadatog opsega za izvorišne i destinacione IP adrese, zapisi u tabelama tokova za ove dvije rute mogu biti međusobno suprotstavljeni. Da bi se izbjegao ovaj problem, saobraćajni tokovi koji je prenose različitim rutama između istog IO para markiraju se različitim labelama, a OpenFlow svičevi u jezgru mreže su konfigurisani da vrše prosleđivanje na osnovu izvorišne i destinacione IP adrese, i dodijeljene labele. Markiranje tokova na nivou rute vrši se dodavanjem odgovarajuće vrijednosti u VLAN ID polje zaglavlja. VLAN ID dodjeljuje se na ulazu u mrežu i uklanja na izlazu iz mreže. Kao što je prikazano na Slici 6.2c, svičevi u jezgru mreže vrše jednostavne akcije prosleđivanja paketa koje su podržane i na konvencionalnim MPLS ruterima. U slučaju da su MPLS ruteri zastupljeni u jezgru mreže, ulazni OpenFlow svičevi mogu markirati saobraćaj dodjeljivanjem odgovarajuće MPLS labele u skladu sa trenutnom šemom rutiranja.

Kada trenutna šema za balansiranje saobraćaja ne može ispuniti QoS zahtjeve korisnika, SDN kontroler kreira dodatna pravila u tabeli tokova ulaznog sviča (*online* rutiranje). Ova pravila imaju veći prioritet od statičkih, i primijenjuju se samo nad onim paketima čija destinaciona IP adresa pripada opsegu IP adresa korisnika. Paketima koji zadovoljavaju pravila *online* rutiranja dodjeljuje se odgovarajuća labela (npr. VLAN ID), a zatim se isti prosleđuju na zadati izlazni interfejs. Sa Slike 6.2a se može primijetiti da kontroler dodjeljuje različite *cookie* vrijednosti [7] statičkim i *online* pravilima u tabeli tokova. Na taj način se pojednostavljuje postupak uklanjanja *online* pravila nakon optimizacije težinskih faktora za balansiranje opterećenja.

U cilju sprečavanja gubitaka kontrolnog saobraćaja (npr. LLDP i BDDP paketa) uslijed prevelike opterećenosti linkova, po dva bafera konfigurisana su na svakom mrežnom interfejsu:

- bafer 0 - koristi se za kontrolni saobraćaj;
- bafer 1 - opslužuje generisano saobraćajno opterećenje brzinom koja je ograničena na 90% kapaciteta linka.

Watch-port parametar *bucket*-a ima važnu ulogu za efikasan oporavak OpenFlow mreže u slučaju otkaza nekog od linkova (Slika 6.2b). Ovaj parametar indicira port (interfejs) sviča čija aktivnost uslovljava korišćenje *bucket*-a. Drugim riječima, *bucket* se smatra aktivnim ukoliko je njegov *watch-port* aktivan. U slučaju kvara lokalnog linka, svi saobraćajni tokovi koji su pogođeni ovim kvarom bivaju automatski rerutirani preko ostalih *bucket*-a u *group* tabeli. Međutim, s obzirom da se *group* tabele koriste samo na ulaznim svičevima, na ovaj način nije moguće reagovati u slučaju otkaza „nelokalnih” linkova. Stoga, ipak je potrebno da kontroler kontinualno prati status linkova i deaktivira određene *bucket*-e u slučaju kvarova. Isključivanje *bucket*-a može se dalje automatizovati korišćenjem BFD (*Bidirectional Forwarding Detection*) sesija na nivou rute za monitoring statusa linkova, umjesto individualnih BFD sesija na nivou svakog linka [128].

6.2 Eksperimentalna validacija performansi

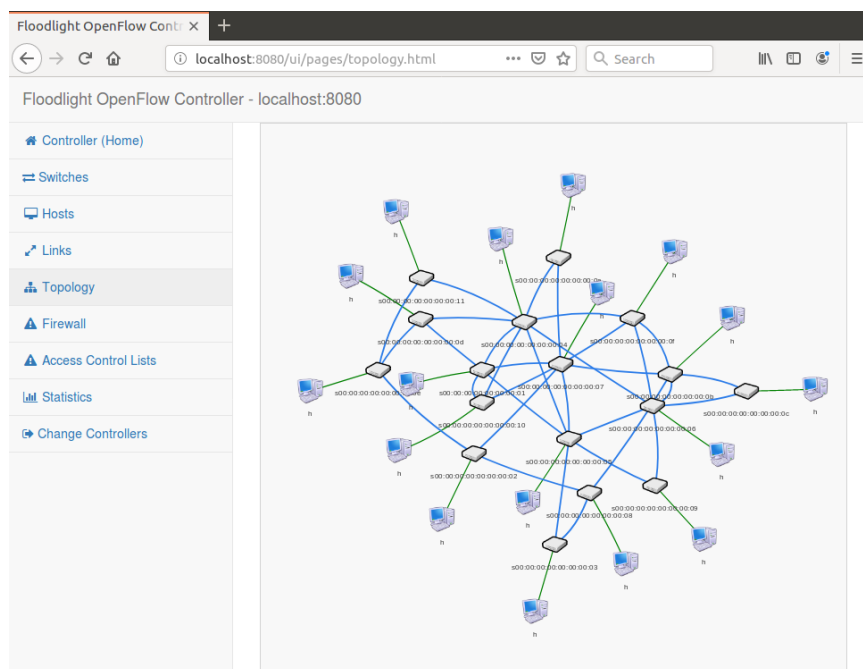
U cilju validacije predloženih TE rešenja RORO-MCH, SDN-LB i SDN-CM algoritmi implementirani su kao dio *offline* TE modula SDN kontrolera. Eksperimenti su realizovani u Mininet emulatoru sa OpenvSwitch-evima. Usled hardverskih ograničenja mašine na kojoj je emulirana virtuelna SDN mreža, konfiguracija eksperimenata je modifikovana u odnosu na konfiguraciju simulacija iz Poglavlja 4 i 5, kao što je prikazano u Tabeli 6.1. Matrica saobraćaja za čas najvećeg opterećenja generisana je istom metodologijom kao u simulacionoj analizi. Generisani su BoD zahtjevi reda 1Mb/s. U slučaju da kontroler prihvati zahtjev, odgovarajuće opterećenje generisano je u formi 5 UDP tokova, čija je brzina prenosa 200kb/s. Kontroler je konfigurisan da sprovodi kontrolu pristupa nad tokovima tako da iskorišćenost linkova ne prelazi 90%. Međutim, linkovi i dalje mogu biti preopterećeni usled nepreciznog dijeljenja saobraćaja preko više ruta. Na Slici 6.4 predstavljen je grafički interfejs Floodlight kontrolera sa prikazom topologije koje je kreirana u Mininet emulatoru.

Eksperimenti su izvršeni za više saobraćajnih scenarija, koji se međusobno razlikuju u pogledu udjela QoS2 i QoS3 saobraćaja u ukupnom generisanom opterećenju. Kao indikatori kvaliteta servisa razmatrani su procenat odbijenih paketa i *jitter*, čije je mjerenje podržano kod Iperf [129] generatora saobraćaja. U Tabeli 6.2 predstavljeni su rezultati ostvareni RORO-MCH pristupom u eksperimentima gdje QoS3 BoD zahtjevi nisu generisani. Ovaj saobraćajni scenario je analiziran jer više doprinosi opterećenju linkova. Naime, za QoS3 zahtjeve rezervišu se redundantni resursi, koji se koriste samo u sluča-

Tabela 6.1: Konfiguracija eksperimenata.

Mrežna topologija	Sprint (PoP lokacije)
Kapacitet linkova	100 Mb/s
Kapacitet linka koji je rezervisan za QoS saobraćaj	80Mb/s
Kapacitet linka koji je rezervisan za kontrolni saobraćaj	10Mb/s
Proces generisanja BoD zahtjeva	Poasonova raspodjela za svaki IO par (srednja vrijednost = prosječni zahtjev IO para/ prosječno trajanje zahtjeva)
BoD zahtjev u pogledu propusnosti	1 Mb/s
Trajanje BoD zahtjeva	Eksponencijalna raspodjela (srednja vrijednost = 20 minuta)
Maksimalni broj ruta za jedan IO par (K)	10
Koeficijent preklapanja ruta IO para (q)	1
Ograničenje kašnjenja za QoS2 klasu	35ms
Ograničenje kašnjenja za QoS3 klasu	100ms
Trajanje eksperimenta	1h
R_{avg}	2.4/6 rekonfiguracija u minuti za scenario bez/sa QoS3 saobraćajem

ju kvara, pa su linkovi posledično slabije opterećeni. Rezultati predstavljeni u Tabeli 6.2 su relevantni iz razloga što je u simulacionoj analizi pretpostavljeno da svičevi na ulazu mreže mogu precizno da realizuju balansiranje opterećenja u skladu sa odlukom koju donese kontroler. Međutim, na performanse algoritma za izbor *bucket*-a utiču razni faktori, uključujući granularnost tokova, *hash*-funkciju, itd. Stoga, nije moguće garantovati QoS bez alokacije dodatnih resursa koji će kompenzovati pomenute nedostatke balansiranja na nivou toka. Samim tim, jasno je da efikasnost predloženog rešenja zavisi od količine resursa koju je potrebno rezervisati za ove svrhe. Rezultati iz Tabele 6.2, međutim, idu u prilog predloženom rešenju implementacije. Naime, maksimalni procenat izgubljenih paketa (MPLR - *Maximum Packet Loss Rate*) nije prelazio 1%, dok je prosječni broj izgubljenih paketa (APLS - *Average Packet Loss Rate*) veoma mali u svakom od sprovedenih eksperimenata. Bitno je napomenuti da su ovi rezultati ostvareni u relativno „nepovoljnim” uslovima, gdje se za svaki BoD zahtjev generiše svega 5 tokova. Sa granularnijim



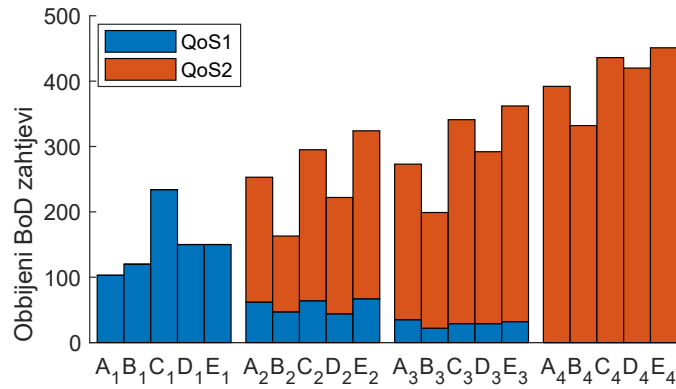
Slika 6.4: Grafički interfejs Floodlight kontrolera i mrežna topologija korišćena u eksperimentima.

Tabela 6.2: Prosječni i maksimalni procenat izgubljenih paketa (*APLR* i *MPLS*), i prosječni i maksimalni *jitter* (*Avg* i *Max jitter*).

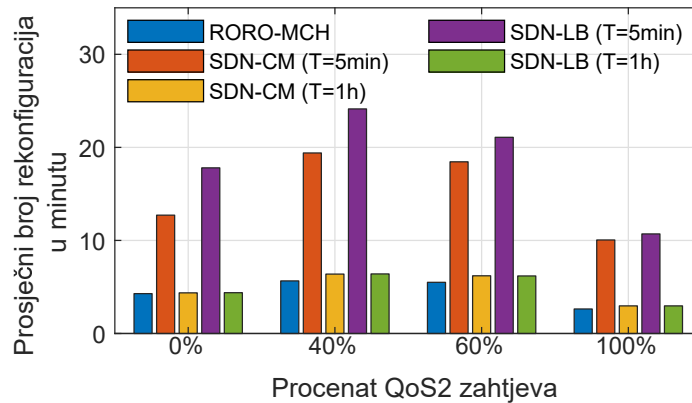
DS %	APLR	MPLR	Avg jitter	Max jitter
0%	0.011%	0.49%	0.82ms	7ms
40%	0.089%	0.61%	0.77ms	4.20ms
60%	0.021%	0.67%	0.68ms	4.33ms
100%	0.079%	0.38%	0.65ms	5.15ms

agregiranim tokovima može se očekivati skoro optimalna preciznost dijeljenja saobraćaja. U eksperimentima je korišćena politika za kontrolu pristupa koja svodi iskorišćenost linkova na nivo ispod 90%. U praksi se primjenjuju i znatno strožije politike u cilju izbjegavanja kašnjenja usled baferovanja. Rezultati u pogledu maksimalnog i prosječnog *jitter*-a iz Tabele 6.2 mogu poslužiti kao indikator kašnjenja usled baferovanja pri visokom intenzitetu saobraćaja.

Na Slikama 6.5 i 6.6 upoređeni su rezultati dobijeni sa RORO-MCH, SDN-LB i CMLU-PTE algoritmima u pogledu broja odbijenih BoD zahtjeva i prosječnog broja rekonfiguracija u minuti. Rezultati se odnose na scenario bez QoS3 saobraćaja, sa R_{avg}



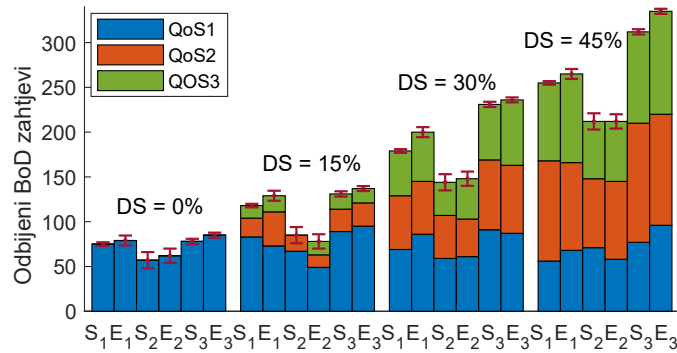
Slika 6.5: Odbijene jedinice BoD zahtjeva u Mininet eksperimentima. Analizirani algoritmi označeni su na sa: A = RORO-MCH, B = SDN-CM (T=5min), C = SDN-CM (T=1h), D = SDN-LB (T=5min), E = SDN-LB (T=1h). Indeksi {1, 2, 3, 4} označavaju različite saobraćajne scenarije sa udjelom QoS2 zahtjeva od {0, 40, 60, 100}%.



Slika 6.6: Prosječna učestanost rekonfiguracija u Mininet eksperimentima bez QoS3 zahtjeva.

ograničenjem od 2.4 rekonfiguracije po minuti. Kao i u simulacionoj analizi, vrijednost R_{avg} ograničenja je birana tako da bude ispod prosječne učestanosti rekonfiguracija u slučaju primjene SDN-LB i SDN-CM algoritama u intervalima od jednog sata ($T = 1h$). Rezultati potvrđuju zaključke simulacione analize.

Eksperimentalni rezultati ("E") i simulacioni rezultati ("S") u pogledu broja odbijenih BoD zahtjeva upoređeni su na Slici 6.7. U eksperimentima su generisane sve tri klase BoD zahtjeva. Procenat DS zahtjeva variran je između 0% i 45%, pri čemu je odnos QoS2 i QoS3 zahtjeva iznosio je 2:1. Performanse RORO-MCH algoritma upoređene su sa SDN-CM algoritmom, kao najefikasnijim od analiziranih jednokriterijskih optimizacionih

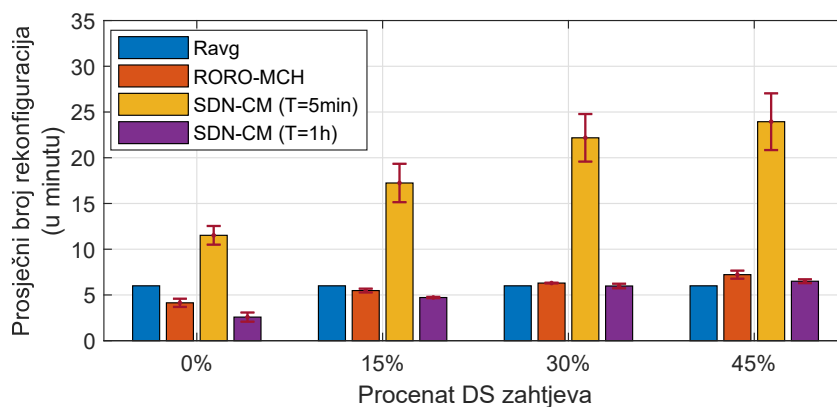


Slika 6.7: Eksperimentalni (E) i simulacioni (S) rezultati u pogledu odbijenih BoD jedinica. Analizirani algoritmi označeni su indeksima: 1 = RORO-MCH, 2 = SDN-CM (T=5min) i 3 = SDN-CM (T=1h).

rešenja. Kao što se može primijetiti sa Slike 6.7, simulacije oponašaju razvijeni prototip SDN kontrolera na zadovoljavajući način. Određene razlike u rezultatima postoje, ali se one mogu smatrati očekivanim kada se sledeće činjenice uzmu u obzir:

- BoD zahtjevi generisani su od strane hostova koji su povezani na periferne Open-Flow svičeve. Obrada BoD zahtijeva u tom slučaju podrazumijeva komunikaciju hosta sa kontrolerom, čekanje poruke potvrde od kontrolera i inicijalizaciju *Iperf* klijentskog procesa. S obzirom da ovaj proces traje izvjesno vrijeme, koje zavisi od iskorišćenosti CPU resursa, vrijeme između dolazaka BoD zahtjeva u Mininet eksperimentu i simulacijama generalno se razlikuje.
- Usled velikog broja generisanih *Iperf* sesija, broj aktivnih niti u eksperimentima dostizao je vrijednosti preko 20000, što se značajno odražava na ponašanje kontrolera. Iz tog razloga interval pouzdanosti za eksperimentalne rezultate veći je od istog za simulacione rezultate.
- Kontroler provjera vrijeme isticanja BoD zahtjeva periodično, svake sekunde.

Eksperimentalni rezultati u pogledu prosječne učestanosti rekonfiguracija u scenariju gdje su generisane sve tri klase BoD zahtjeva predstavljeni su na Slici 6.8. Optimizacija cijene rutiranja izvršena je nakon petog minuta od strane oba analizirana algoritma, u cilju fer poređenja i inicijalizacije težinskih faktora za balansiranje opterećenja. Rekonfiguracije koje su izvršene tom prilikom nisu ubrojane u predstavljene rezultate. Matrica saobraćaja je skalirana u odnosu na prethodni scenario, kao što je rađeno i u simulacionoj



Slika 6.8: Prosječna učestanost rekonfiguracija u Mininet eksperimentima sa QoS3 zahtjevima.

analizi. Sa Slike 6.8 primjetno je da se broj rekonfiguracija za oba algoritma povećava sa procentom DS zahtjeva. U eksperimentima sa 30% i 45% DS zahtjeva, RORO-MCH značajno zaostaje za SDN-CM (T=5min) algoritmom u pogledu optimalnosti šeme rutiranja. Međutim, prosječna učestanost rekonfiguracija je i dalje približna R_{avg} , iako je mreže izložena velikom saobraćajnom opterećenju tokom čitavog trajanja eksperimenata.

Zaključak

Usled konstantne evolucije mrežnih servisa i Internet aplikacija, saobraćaj u ISP mrežama postaje sve kompleksniji i varijabilniji. Konvergencija servisa za prenos podataka, govora i televizijskih sadržaja na *all-IP* infrastrukturi direktno se odražava na porast saobraćaja i njegovu dinamiku. Kako se uslovi u mreži dinamički mijenjaju, tako se mijenja i kvalitet servisa za saobraćajne tokove. Uobičajena strategija ISP provajdera za garanciju kvaliteta servisa je optimizacija ruta za „najgori scenario” saobraćajnih uslova, kako bi se eliminisala potreba za rekonfiguracijama mreže. Kao rezultat ove strategije, mrežni linkovi se obično dimenzionišu tako da je njihov kapacitet 2-3 puta veći od prosječnog opterećenja. S obzirom da takav pristup nije ekonomski održiv na duže staze, u ovoj disertaciji analizirane su mogućnosti za poboljšanje servisnog kapaciteta ISP mreža primjenom naprednih TE tehnika.

U disertaciji su istaknuti nedostaci tradicionalnih TE tehnologija i QoS arhitektura, koji su motivisali razvoj SDN TE rešenja. Predloženo je SDN kontrolno okruženje za ISP mreže, čija je logika za upravljanje saobraćajem oraganizovana u dva modula: *online* modul za rutiranje i *offline* TE modul. Osnovna ideja predloženog rešenja je pružanje fleksibilnog BoD servisa, koji zahtijeva obradu korisničkih zahtjeva u realnom vremenu, uz što efikasnije korišćenje mrežnih resursa. Za razliku od tradicionalnih servisa ISP provajdera, koji su uglavnom ograničeni na iznajmljivanje privatnih virtuelnih linkova sa rezervisanim kapacitetom, predloženi model BoD servisa omogućava korisnicima da dinamički prilagođavaju parametre svojih konekcija iznenadnim i privremenim potrebama. BoD zahtjevi pored željene propusnosti specificiraju ograničenje kašnjenja za konekcije i zahtjeve u pogledu pouzdanosti. *Online* modul za rutiranje i *offline* TE modul dizajnirani su prema različitim kriterijumima, koji su definisani u skladu sa njihovim ulogama u procesu upravljanja saobraćajem. Primarni zadatak *online* modula za rutiranje je brzo donošenje odluke o prihvatanju/odbijanju BoD zahtjeva na osnovu informacija o trenutnoj zauzetosti mrežnih resursa. Stoga, ovaj modul je baziran na računski jednostavnom

QoS algoritmu rutiranja, koji teži da izbjegne dodavanje novih zapisa u tabelama tokova OpenFlow svičeva. S obzirom da SDN uređaji tipično koriste TCAM (*Ternary Content Addressable Memory*) memoriju manjeg kapaciteta za implementaciju tabele tokova, minimizacija broja zapisa predstavlja važan zahtjev [130, 131]. Za optimizaciju iskorišćenosti mrežnih resursa zadužen je *offline* TE modul. *Offline* TE algoritam (SDN-LB) se odlikuje većom kompleksnošću, i izvršava se periodično. Zadatak algoritma je računanje optimalne distribucije matrice saobraćaja preko više tunela koji su u mreži unaprijed instalirani. Na osnovu rezultata *offline* TE algoritma, kontroler ažurira parametre za balansiranje opterećenja u tabelama tokova SDN svičeva. Simulaciona analiza pokazala je da predloženi pristup postiže značajno bolje performanse od CSPF-baziranog rešenja koje je dominantno zastupljeno u današnjim ISP mrežama. Prednosti dinamičkog inženjeringa saobraćaja demonstrirane su i poređenjem sa *oblivious* tehnikom rutiranja, koja je optimizovana za *hose* model saobraćaja. Poboljšanja performansi evidentna su i kada se optimizacioni TE algoritam izvršava u dužim vremenskim intervalima. Samim tim mogu se očekivati zadovoljavajuće performanse i kada informacije o matrici saobraćaja nisu precizno poznate.

Predloženi TE pristup podrazumijeva prisustvo SDN uređaja na periferiji mreže, dok u jezgru mreže koristi statičke tunele koje je moguće implementirati i tradicionalnim MPLS ruterima. Mogućnost integracije sa MPLS uređajima veoma je bitna iz ekonomske perspektive, jer tranzicija sa MPLS na SDN tehnologiju zahtijeva značajna ulaganja u infrastrukturu. Međutim, odsustvo SDN uređaja u jezgru mreže, odnosno korišćenje statičkih tunela, ograničava servisni kapacitet ISP provajdera. Iz ovog razloga, u disertaciji je izvršeno poređenje predloženog pristupa sa dva kompleksnija rešenja koja koriste fleksibilnost SDN rutiranja i u jezgru mreže. Simulacionom analizom utvrđeno je da su gubici u pogledu servisnog kapaciteta relativno mali, dok se primjenom statičkih tunela značajno smanjuje kontrolni *overhead* i kompleksnost TE algoritma.

U Poglavlju 5 napravljen je novi iskorak u odnosu na postojeće TE pristupe. Predložen je RO-RO optimizacioni algoritam za *offline* TE modul. Za razliku od većine postojećih TE algoritama koji su fokusirani na optimizaciju iskorišćenosti mrežnih resursa, RO-RO teži da obezbijedi stabilan rad mreže kontrolisanjem broja rekonfiguracija. Predlog rešenja motivisan je naglom degradacijom performansi SDN-LB algoritma, ali i drugih periodičnih TE algoritama baziranih na jedno-kriterijumskom optimizacionom modelu, sa povećanjem periode izvršavanja. Sa druge strane, često izvršavanje ovih algoritama povlači veliki broj rekonfiguracija koje se negativno odražavaju na stabilnost mreže. RO-

RO algoritam baziran je na optimizacionom modelu koji teži simultano da minimizuje dva suprotstavljena kriterijuma: MLU i broj rekonfiguracija. Za generisanje seta Pareto optimalnih rešenja primijenjen je AUGMECON metod [125], dok je za izbor najefikasnijeg kompromisnog rešenja iz Pareto seta predložen Lypaunov *drift-plus-penalty* algoritam [31], kojem se kao ulazni parametar zadaje ograničenje u pogledu prosječne učestanosti rekonfiguracija. Predloženi algoritam dopušta privremeno prekoračenje zadanog ograničenja u slučaju kada konfiguracija težinskih faktora za balansiranje opterećenja značajno odstupa od optimalne. Međutim, suprostavljene optimizacione funkcije su dinamički priorizovane, uzimajući u obzir broj rekonfiguracija koji je izvršen u prethodnim ciklusima, tako da posmatrano na dužem vremenskom intervalu učestanost rekonfiguracija rijetko prelazi kritičnu vrijednost koju definiše mežni operator. Kako je na ovaj način broj rekonfiguracija prilikom svakog izvršavanja RO-RO algoritma kontrolisan, SDN kontroler može vršiti inženjering saobraćaja češće, u cilju pravovremene reakcije na iznenadne saobraćajne distorzije i kvarove u mreži. Stoga, iako je u disertaciji modul kontrolera koji izvršava TE algoritam označen kao *offline* modul, primjenom RO-RO algoritma on dobija više *online* prirodu, jer se može izvršavati u relativno malim vremenskim razmacima bez negativnog uticaja na stabilnost mreže.

Poboljšanja ostvarena RO-RO algoritmom postignuta su na račun povećanja računske kompleksnosti. Za razliku od SDN-LB algoritma koji je baziran na LP optimizacionom modelu, RO-RO optimizacioni problem je *NP-complete*, što ograničava njegovu primjenu na manje mrežne segmente. S ciljem da se smanji računaska složenost RO-RO algoritma, predložena je njegova heuristika RORO-H, koja u više iteracija izvršava LP model za minimizaciju MLU. Prezentovani rezultati simulacija potvrdili su da RO-RO heuristika doprinosi efikasnijem korišćenju mrežnih resursa od *state-of-the-art* konkurenskog rešenja iz literature ([18]), koje je takođe bazirano na bi-kriterijumskom optimizacionom modelu. Poboljšanja su ostvarena i pored manje računske složenosti. U disertaciji je razmatrano više različitih definicija optimizacione funkcije za problem balansiranja saobraćajnog opterećenja. Pokazano je da se performanse RORO i RORO-H algoritama mogu dodatno unaprijediti ukoliko se optimizaciona funkcija koja minimizuje MLU proširi tako da minimizuje i ukupnu „cijenu” rutiranja (RORO-MCH algoritam).

Prototip predloženog SDN kontrolnog okruženja implementiran je na Floodlight Open-Flow kontroleru. U sklopu okruženja implementirani su RORO-MCH i SDN-LB algoritmi. Eksperimenti u Mininet virtuelizovanom mrežnom okruženju potvrdili su zaključke simulacione analize.

TE algoritmi koji su prezentovani u ovoj disertaciji mogu se primijeniti i za alokaciju virtuelnih mreža u 5G sistemima. Naime, problem alokacije resursa za virtuelne topologije može se direktno svesti na problem alokacije višestrukih virtuelnih linkova, koji je razmatran u disertaciji. U tom slučaju korisno je razmotriti drugačije modele za specifikaciju QoS zahtjeva, u cilju uštede resursa. Interesantan pravac za nastavak istraživanja su adaptivni TE mehanizmi za dinamičko skaliranje BoD zahtjeva virtuelnih linkova koji su sastavni dio jedne virtuelne mrežne topologije na osnovu predikcija sobračajne matrice. Imajući u vidu da različiti BoD zahtjevi imaju različit korist za ISP provajdere, koju je u trenutku njihovog pristizanja teško procijeniti, aktuelna istraživanja su usmjerena na primjenu *reinforcement learning* algoritama ([132]) za donošenje inteligentnijih odluka o kontroli pristupa. Hijerarhijski dizajn kontrolne ravni od velike je važnosti za ISP mreže većih dimenzija, pa dalji rad može biti fokusiran i u tom pravcu.

Literatura

- [1] C.-Y. Hong *et al.*, “Achieving high utilization with software-driven wan,” ser. Proc. of SIGCOMM ’13, 2013, pp. 15–26.
- [2] I. F. Akyildiz, A. Lee, P. Wang, M. Luo, and W. Chou, “A roadmap for traffic engineering in sdn-openflow networks,” *Computer Networks*, vol. 71, pp. 1 – 30, 2014. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S1389128614002254>
- [3] N. Wang, K. H. Ho, G. Pavlou, and M. Howarth, “An overview of routing optimization for internet traffic engineering,” *IEEE Communications Surveys Tutorials*, vol. 10, no. 1, pp. 36–56, First 2008.
- [4] S. Das *et al.*, “Application-aware aggregation and traffic engineering in a converged packet-circuit network,” in *2011 Optical Fiber Communication Conference and Exposition and the National Fiber Optic Engineers Conference*, March 2011, pp. 1–3.
- [5] A. Sadasivarao, S. Syed, P. Pan, C. Liou, A. Lake, C. Guok, and I. Monga, “Open transport switch: A software defined networking architecture for transport networks,” in *Proceedings of the Second ACM SIGCOMM Workshop on Hot Topics in Software Defined Networking*, ser. HotSDN ’13. New York, NY, USA: ACM, 2013, pp. 115–120. [Online]. Available: <http://doi.acm.org/10.1145/2491185.2491192>
- [6] S. Tomovic, N. Prasad, and I. Radusinovic, “Sdn control framework for qos provisioning,” in *2014 22nd Telecommunications Forum Telfor (TELFOR)*, Nov 2014, pp. 111–114.

- [7] “Open networking foundation - openflow v1.4 specification,” <https://www.opennetworking.org> (Accessed: 2018-27-02). [Online]. Available: <https://www.opennetworking.org>
- [8] S. Tomovic, N. Lekic, I. Radusinovic, and G. Gardasevic, “A new approach to dynamic routing in sdn networks,” in *2016 18th Mediterranean Electrotechnical Conference (MELECON)*, April 2016, pp. 1–6.
- [9] S. Jain *et al.*, “B4: Experience with a globally-deployed software defined wan,” *SIGCOMM Comput. Commun. Rev.*, vol. 43, no. 4, pp. 3–14, Aug. 2013. [Online]. Available: <http://doi.acm.org/10.1145/2534169.2486019>
- [10] M. Al-Fares *et al.*, “Hedera: Dynamic flow scheduling for data center networks,” in *Proc. of the NSDI’10*. Berkeley, CA, USA: USENIX Association, 2010, pp. 19–19. [Online]. Available: <http://dl.acm.org/citation.cfm?id=1855711.1855730>
- [11] A. R. Curtis *et al.*, “Devoflow: Scaling flow management for high-performance networks,” *SIGCOMM Comput. Commun. Rev.*, vol. 41, no. 4, pp. 254–265, Aug. 2011. [Online]. Available: <http://doi.acm.org/10.1145/2043164.2018466>
- [12] M. Yu *et al.*, “Scalable flow-based networking with difane,” *SIGCOMM Comput. Comm. Rev.*, vol. 40, no. 4, pp. 351–362, Aug. 2010. [Online]. Available: <http://doi.acm.org/10.1145/1851275.1851224>
- [13] H. H. Liu *et al.*, “Traffic engineering with forward fault correction,” *SIGCOMM Comput. Comm. Rev.*, vol. 44, no. 4, pp. 527–538, Aug. 2014. [Online]. Available: <http://doi.acm.org/10.1145/2740070.2626314>
- [14] J. M. Wang *et al.*, “Sdn-based multi-class qos-guaranteed inter-data center traffic management,” in *2014 IEEE 3rd International Conference on Cloud Networking (CloudNet)*, Oct 2014, pp. 401–406.
- [15] M. Moradi, Y. Zhang, Z. M. Mao, and R. Manghirmalani, “Dragon: Scalable, flexible, and efficient traffic engineering in software defined isp networks,” *IEEE Journal on Selected Areas in Communications*, vol. 36, no. 12, pp. 2744–2756, Dec 2018.
- [16] S. Tomovic and I. Radusinovic, “Ro-ro: Routing optimality - reconfiguration overhead balance in software-defined isp networks,” *IEEE Journal on Selected Areas in Communications*, pp. 1–1, 2019.

- [17] S. Tomovic and I. Radusinovic, "An effective use of sdn for virtual-link provisioning in isp networks," *IEICE Transactions on Communications*, 2018.
- [18] U. Ashraf, "Rule minimization for traffic evolution in software-defined networks," *IEEE Communications Letters*, vol. 21, no. 4, pp. 793–796, April 2017.
- [19] S. Paris *et al.*, "Dynamic control for failure recovery and flow reconfiguration in sdn," in *12th Int. Conf. on the Design of Reliable Communication Networks*, March 2016, pp. 152–159.
- [20] K. Kar, M. Kodialam, and T. V. Lakshman, "Minimum interference routing of bandwidth guaranteed tunnels with mpls traffic engineering applications," *IEEE Journal on Selected Areas in Communications*, vol. 18, no. 12, pp. 2566–2579, Dec 2000.
- [21] W. Szeto, R. Boutaba, and Y. Iraqi, "Dynamic online routing algorithm for mpls traffic engineering," in *NETWORKING 2002: Networking Technologies, Services, and Protocols; Performance of Computer and Communication Networks; Mobile and Wireless Communications*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2002, pp. 936–946.
- [22] S. Suri, M. Waldvogel, and P. R. Warkhede, "Profile-based routing: A new framework for mpls traffic engineering," in *Proceedings of the Second International Workshop on Quality of Future Internet Services*, ser. COST 263. London, UK, UK: Springer-Verlag, 2001, pp. 138–157. [Online]. Available: <http://dl.acm.org/citation.cfm?id=646462.693730>
- [23] J. W. Guck, A. Van Bemten, M. Reisslein, and W. Kellerer, "Unicast qos routing algorithms for sdn: A comprehensive survey and performance evaluation," *IEEE Communications Surveys Tutorials*, vol. 20, no. 1, pp. 388–415, Firstquarter 2018.
- [24] S. Tomovic and I. Radusinovic, "Fast and efficient bandwidth-delay constrained routing algorithm for sdn networks," in *2016 IEEE NetSoft Conference and Workshops (NetSoft)*, June 2016, pp. 303–311.
- [25] S. Tomović, "Algoritmi rutiranja za podršku kvaliteta servisa u softverski definisanim mrežama," Podgorica, Montenegro, 2015.

- [26] S. Tomovic, M. Radonjic, and I. Radusinovic, "Bandwidth-delay constrained routing algorithms for backbone sdn networks," in *2015 12th International Conference on Telecommunication in Modern Satellite, Cable and Broadcasting Services (TELSIKS)*, Oct 2015, pp. 227–230.
- [27] S. Tomovic, I. Radusinovic, and N. Prasad, "Performance comparison of qos routing algorithms applicable to large-scale sdn networks," in *IEEE EUROCON 2015 - International Conference on Computer as a Tool (EUROCON)*, Sep. 2015, pp. 1–6.
- [28] R. T. Braden, D. D. D. Clark, and S. Shenker, "Integrated Services in the Internet Architecture: an Overview," RFC 1633, Jun. 1994. [Online]. Available: <https://rfc-editor.org/rfc/rfc1633.txt>
- [29] D. L. Black and P. Jones, "Differentiated Services (Diffserv) and Real-Time Communication," RFC 7657, Nov. 2015. [Online]. Available: <https://rfc-editor.org/rfc/rfc7657.txt>
- [30] P. Ngatchou, A. Zarei, and A. El-Sharkawi, "Pareto multi objective optimization," in *Proceedings of the 13th International Conference on, Intelligent Systems Application to Power Systems*, Nov 2005, pp. 84–91.
- [31] M. J. Neely, *Stochastic Network Optimization with Application to Communication and Queueing Systems*. Morgan and Claypool Publishers, 2010.
- [32] G. Gardašević, M. Veletić, N. Maletić, D. Vasiljević, I. Radusinović, S. Tomović, and M. Radonjić, "The iot architectural framework, design issues and application domains," *Wireless Personal Communications*, vol. 92, no. 1, pp. 127–148, Jan 2017. [Online]. Available: <https://doi.org/10.1007/s11277-016-3842-3>
- [33] M. Shafi, A. F. Molisch, P. J. Smith, T. Haustein, P. Zhu, P. De Silva, F. Tufveson, A. Benjebbour, and G. Wunder, "5g: A tutorial overview of standards, trials, challenges, deployment, and practice," *IEEE Journal on Selected Areas in Communications*, vol. 35, no. 6, pp. 1201–1221, June 2017.
- [34] M. Agiwal, A. Roy, and N. Saxena, "Next generation 5G wireless networks: a comprehensive survey," *IEEE Communications Surveys Tutorials*, vol. PP, no. 99, pp. 1–1, 2016.

- [35] ITU, “Setting the scene for 5g: Opportunities and challenges,” February 2018. [Online]. Available: https://www.itu.int/en/ITU-D/Documents/ITU_5G_REPORT-2018.pdf
- [36] D. Kreutz, F. M. V. Ramos, P. E. Veríssimo, C. E. Rothenberg, S. Azodolmolky, and S. Uhlig, “Software-defined networking: A comprehensive survey,” *Proceedings of the IEEE*, vol. 103, no. 1, pp. 14–76, Jan 2015.
- [37] Y. Li and M. Chen, “Software-defined network function virtualization: A survey,” *IEEE Access*, vol. 3, pp. 2542–2553, 2015.
- [38] S. Tomovic, K. Yoshigoe, I. Maljevic, and I. Radusinovic, “Software-defined fog network architecture for iot,” *Wirel. Pers. Commun.*, vol. 92, no. 1, pp. 181–196, Jan. 2017. [Online]. Available: <https://doi.org/10.1007/s11277-016-3845-0>
- [39] T. Taleb, K. Samdanis, B. Mada, H. Flinck, S. Dutta, and D. Sabella, “On multi-access edge computing: A survey of the emerging 5g network edge cloud architecture and orchestration,” *IEEE Communications Surveys Tutorials*, vol. 19, no. 3, pp. 1657–1681, thirdquarter 2017.
- [40] Cisco Visual Networking Index, “Global mobile data traffic forecast update, 2017–2022,” February 2019. [Online]. Available: <https://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/white-paper-c11-738429.pdf>
- [41] Ericsson, “Ericsson mobility report,” June 2018. [Online]. Available: <https://www.ericsson.com/assets/local/mobility-report/documents/2018/ericsson-mobility-report-june-2018.pdf>
- [42] N. Xia, H. Chen, and C. Yang, “Radio resource management in machine-to-machine communications—a survey,” *IEEE Communications Surveys Tutorials*, vol. 20, no. 1, pp. 791–828, Firstquarter 2018.
- [43] SK-Telekom, “Sk telekom’s view on 5g vision, architecture, technology and spectrum,” 2014. [Online]. Available: https://www.sktelecom.com/img/pds/press/SKT_5G%20White%20Paper_V1.0_Eng.pdf
- [44] NGMN Alliance. 5g white paper. [Online]. Available: https://www.ngmn.org/fileadmin/ngmn/content/images/news/ngmn_news/NGMN_5G_White_Paper_V1_0.pdf

- [45] 3GPP, “Study on new services and markets technology enablers,” August 2016.
- [46] S. Chang and W. Chen, “Does visualize industries matter? a technology foresight of global virtual reality and augmented reality industry,” in *2017 International Conference on Applied System Innovation (ICASI)*, May 2017, pp. 382–385.
- [47] S. Tomovic and I. Radusinovic, “Mapping application requirements to virtualization-enabled software defined wsn,” *Wirel. Pers. Commun.*, vol. 97, no. 2, pp. 1693–1709, Nov. 2017. [Online]. Available: <https://doi.org/10.1007/s11277-017-4650-0>
- [48] ITU-R. Imt vision – framework and overall objectives of the future development of imt for 2020 and beyond. [Online]. Available: https://www.itu.int/dms_pubrec/itu-r/rec/m/r-rec-m.2083-0-201509-i!!pdf-e.pdf
- [49] ITU-T, “Requirements of the imt-2020 network,” 2018.
- [50] S. Tomovic, M. Pejanovic-Djurisic, and I. Radusinovic, “Sdn based mobile networks: Concepts and benefits,” *Wirel. Pers. Commun.*, vol. 78, no. 3, pp. 1629–1644, Oct. 2014. [Online]. Available: <http://dx.doi.org/10.1007/s11277-014-1909-6>
- [51] H. Kim and N. Feamster, “Improving network management with software defined networking,” *IEEE Communications Magazine*, vol. 51, no. 2, pp. 114–119, February 2013.
- [52] R. Mijumbi, J. Serrat, J. Gorricho, N. Bouten, F. De Turck, and R. Boutaba, “Network function virtualization: State-of-the-art and research challenges,” *IEEE Communications Surveys Tutorials*, vol. 18, no. 1, pp. 236–262, Firstquarter 2016.
- [53] G. Davoli, W. Cerroni, S. Tomovic, C. Buratti, C. Contoli, and F. Callegati, “Intent-based service management for heterogeneous software-defined infrastructure domains,” *International Journal of Network Management*, vol. 29, no. 1, p. e2051, e2051 nem.2051. [Online]. Available: <https://onlinelibrary.wiley.com/doi/abs/10.1002/nem.2051>
- [54] S. Tomovic, W. Cerroni, F. Callegati, R. Verdone, I. Radusinovic, M. Pejanovic, and C. Buratti, “An architecture for qos-aware service deployment in software-defined iot networks,” in *2017 20th International Symposium on Wireless Personal Multimedia Communications (WPMC)*, Dec 2017, pp. 561–567.

- [55] T. Koponen *et al.*, “Onix: A distributed control platform for large-scale production networks,” in *9th USENIX Conference on Operating Systems Design and Implementation*, 2010, pp. 351–364.
- [56] S. H. Yeganeh *et al.*, “On scalability of software-defined networking,” *IEEE Communications Magazine*, vol. 51, no. 2, pp. 136–141, February 2013.
- [57] J. M. Halpern, R. HAAS, A. Doria, L. Dong, W. Wang, H. M. Khosravi, J. H. Salim, and R. Gopal, “Forwarding and Control Element Separation (ForCES) Protocol Specification,” RFC 5810, Mar. 2010. [Online]. Available: <https://rfc-editor.org/rfc/rfc5810.txt>
- [58] H. Song, “Protocol-oblivious forwarding: Unleash the power of sdn through a future-proof forwarding plane,” in *Proceedings of the Second ACM SIGCOMM Workshop on Hot Topics in Software Defined Networking*, ser. HotSDN ’13. New York, NY, USA: ACM, 2013, pp. 127–132. [Online]. Available: <http://doi.acm.org/10.1145/2491185.2491190>
- [59] M. Smith, R. E. Adams, M. Dvorkin, Y. Laribi, V. Pandey, P. Garg, and N. Weidenbacher, “OpFlex Control Protocol,” Internet Engineering Task Force, Internet-Draft draft-smith-opflex-03, Apr. 2016, work in Progress. [Online]. Available: <https://datatracker.ietf.org/doc/html/draft-smith-opflex-03>
- [60] G. Bianchi, M. Bonola, A. Capone, and C. Cascone, “Openstate: Programming platform-independent stateful openflow applications inside the switch,” *SIGCOMM Comput. Commun. Rev.*, vol. 44, no. 2, pp. 44–51, Apr. 2014. [Online]. Available: <http://doi.acm.org/10.1145/2602204.2602211>
- [61] Floodlight OpenFlow project, (Accessed: 2018-27-02). [Online]. Available: <http://www.projectfloodlight.org/floodlight/>
- [62] Onos project. (Accessed: 2019-22-03). [Online]. Available: <https://onosproject.org/>
- [63] OpenDaylight project. (Accessed: 2019-22-03). [Online]. Available: <https://www.opendaylight.org/>
- [64] Pox project. (Accessed: 2019-22-03). [Online]. Available: <https://github.com/noxrepo/pox>

- [65] Nox project. (Accessed: 2019-22-03). [Online]. Available: <https://github.com/noxrepo/nox>
- [66] Ryu project. (Accessed: 2019-22-03). [Online]. Available: <https://osrg.github.io/ryu/>
- [67] C. Janz, L. Ong, K. Sethuraman, and V. Shukla, "Emerging transport sdn architecture and use cases," *IEEE Communications Magazine*, vol. 54, no. 10, pp. 116–121, October 2016.
- [68] K. Gopalan, "Efficient provisioning algorithms for network resource virtualization with qos guarantees," Ph.D. dissertation, Stony Brook University, 2003.
- [69] B. Fortz and M. Thorup, "Internet traffic engineering by optimizing ospf weights," in *Proceedings IEEE INFOCOM 2000. Conference on Computer Communications. Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies (Cat. No.00CH37064)*, vol. 2, March 2000, pp. 519–528 vol.2.
- [70] B. Fortz, J. Rexford, and M. Thorup, "Traffic engineering with traditional ip routing protocols," *IEEE Communications Magazine*, vol. 40, no. 10, pp. 118–124, Oct 2002.
- [71] B. Fortz and M. Thorup, "Optimizing ospf/is-is weights in a changing world," *IEEE Journal on Selected Areas in Communications*, vol. 20, no. 4, pp. 756–767, May 2002.
- [72] R. A. Guerin, A. Orda, and D. Williams, "Qos routing mechanisms and ospf extensions," in *GLOBECOM 97. IEEE Global Telecommunications Conference. Conference Record*, vol. 3, Nov 1997, pp. 1903–1908 vol.3.
- [73] C. Labovitz, G. R. Malan, and F. Jahanian, "Internet routing instability," *IEEE-ACM Transactions on Networking*, vol. 6, no. 5, pp. 515–528, Oct 1998.
- [74] C. Hopps, "Analysis of an Equal-Cost Multi-Path Algorithm," RFC 2992, Nov. 2000. [Online]. Available: <https://rfc-editor.org/rfc/rfc2992.txt>
- [75] S. Kandula, D. Katabi, S. Sinha, and A. Berger, "Dynamic load balancing without packet reordering," *SIGCOMM Comput. Commun. Rev.*, vol. 37, no. 2, pp. 51–62, Mar. 2007. [Online]. Available: <http://doi.acm.org/10.1145/1232919.1232925>

- [76] A. Viswanathan, E. C. Rosen, and R. Callon, "Multiprotocol Label Switching Architecture," RFC 3031, Jan. 2001. [Online]. Available: <https://rfc-editor.org/rfc/rfc3031.txt>
- [77] S. Kandula, D. Katabi, B. S. Davie, and A. Charny, "Walking the tightrope: responsive yet stable traffic engineering," in *SIGCOMM*, 2005.
- [78] R. Teixeira, K. Marzullo, S. Savage, and G. M. Voelker, "Characterizing and measuring path diversity of internet topologies," *SIGMETRICS Perform. Eval. Rev.*, vol. 31, no. 1, pp. 304–305, Jun. 2003. [Online]. Available: <http://doi.acm.org/10.1145/885651.781069>
- [79] S. Savage, A. Collins, E. Hoffman, J. Snell, and T. Anderson, "The end-to-end effects of internet path selection," *SIGCOMM Comput. Commun. Rev.*, vol. 29, no. 4, pp. 289–299, Aug. 1999. [Online]. Available: <http://doi.acm.org/10.1145/316194.316233>
- [80] D. Awduche, J. Malcolm, J. Agogbua, M. O'Dell, and J. McManus, "Requirements for traffic engineering over mpls," United States, 1999.
- [81] R. T. Braden, L. Zhang, S. Berson, S. Herzog, and S. Jamin, "Resource ReSerVation Protocol (RSVP) – Version 1 Functional Specification," RFC 2205, Sep. 1997. [Online]. Available: <https://rfc-editor.org/rfc/rfc2205.txt>
- [82] S. Even *et al.*, "On the complexity of time table and multi-commodity flow problems," in *16th Annual Symposium on Foundations of Computer Science*, October 1975, pp. 184–193.
- [83] O. Younis and S. Fahmy, "Constraint-based routing in the internet: Basic principles and recent research," *IEEE Communications Surveys Tutorials*, vol. 5, no. 1, pp. 2–13, Third 2003.
- [84] B. B. A. Hannan and L. M. Ni, "Traffic engineering with mpls in the internet," *IEEE Network*, vol. 14, no. 2, pp. 28–33, March 2000.
- [85] J. Crowcroft, "Quality-of-service routing for supporting multimedia applications," *IEEE Journal on Selected Areas in Communications*, vol. 14, no. 7, pp. 1228–1234, Sep. 1996.

- [86] Y. Wang and Z. Wang, "Explicit routing algorithms for internet traffic engineering," in *Proceedings Eight International Conference on Computer Communications and Networks (Cat. No.99EX370)*, Oct 1999, pp. 582–588.
- [87] A. Elwalid, C. Jin, S. Low, and I. Widjaja, "Mate: Mpls adaptive traffic engineering," in *Proceedings IEEE INFOCOM 2001. Conference on Computer Communications. Twentieth Annual Joint Conference of the IEEE Computer and Communications Society (Cat. No.01CH37213)*, vol. 3, April 2001, pp. 1300–1309 vol.3.
- [88] S. Das, G. Parulkar, and N. McKeown, "Why openflow/sdn can succeed where gmpls failed," in *2012 38th European Conference and Exhibition on Optical Communications*, Sep. 2012, pp. 1–3.
- [89] J. Vasseur, A. Farrel, and G. Ash, "A Path Computation Element (PCE)-Based Architecture," RFC 4655, Aug. 2006. [Online]. Available: <https://rfc-editor.org/rfc/rfc4655.txt>
- [90] K. Leung, V. O. k. Li, and D. Yang, "An overview of packet reordering in transmission control protocol (tcp): Problems, solutions, and challenges," *IEEE Transactions on Parallel and Distributed Systems*, vol. 18, no. 4, pp. 522–535, April 2007.
- [91] M. Alizadeh, T. Edsall, S. Dharmapurikar, R. Vaidyanathan, K. Chu, A. Fingerhut, V. T. Lam, F. Matus, R. Pan, N. Yadav, and G. Varghese, "Conga: Distributed congestion-aware load balancing for datacenters," *SIGCOMM Comput. Commun. Rev.*, vol. 44, no. 4, pp. 503–514, Aug. 2014. [Online]. Available: <http://doi.acm.org/10.1145/2740070.2626316>
- [92] A. Pathak, M. Zhang, Y. C. Hu, R. Mahajan, and D. Maltz, "Latency inflation with mpls-based traffic engineering," in *Proceedings of the 2011 ACM SIGCOMM Conference on Internet Measurement Conference*, ser. IMC '11. New York, NY, USA: ACM, 2011, pp. 463–472. [Online]. Available: <http://doi.acm.org/10.1145/2068816.2068859>
- [93] A. Mendiola *et al.*, "A survey on the contributions of software-defined networking to traffic engineering," *IEEE Communications Surveys Tutorials*, vol. 19, no. 2, pp. 918–953, Secondquarter 2017.

- [94] J. Vasseur, A. Farrel, and G. Ash, “A Path Computation Element (PCE)-Based Architecture,” RFC 4655, Aug. 2006. [Online]. Available: <https://rfc-editor.org/rfc/rfc4655.txt>
- [95] J. L. Martins and N. Campos, “Short-sighted routing, or when less is more,” *IEEE Comm. Magazine*, vol. 54, no. 10, pp. 82–88, Oct. 2016.
- [96] M. Kodialam, T. V. Lakshman, J. B. Orlin, and S. Sengupta, “Oblivious routing of highly variable traffic in service overlays and ip backbones,” *IEEE/ACM Transactions on Networking*, vol. 17, no. 2, pp. 459–472, April 2009.
- [97] M. Kodialam *et al.*, “Traffic-oblivious routing in the hose model,” *IEEE/ACM Transactions on Networking*, vol. 19, no. 3, pp. 774–787, June 2011.
- [98] A. Medina, N. Taft, K. Salamatian, S. Bhattacharyya, and C. Diot, “Traffic matrix estimation: Existing techniques and new directions,” *SIGCOMM Comput. Commun. Rev.*, vol. 32, no. 4, pp. 161–174, Aug. 2002. [Online]. Available: <http://doi.acm.org/10.1145/964725.633041>
- [99] E. Oki and A. Iwaki, “Performance comparisons of optimal routing by pipe, hose, and intermediate models,” in *2009 IEEE Sarnoff Symposium*, March 2009, pp. 1–5.
- [100] E. Oki, R. Rojas-Cessa, M. Tatipamula, and C. Vogt, *Advanced Internet Protocols, Services, and Applications*, 1st ed. Wiley Publishing, 2012.
- [101] S. R. Hunter, E. A. Applegate, V. Arora, B. Chong, K. Cooper, O. Rincón-Guevara, and C. Vivas-Valencia, “An introduction to multiobjective simulation optimization,” *ACM Trans. Model. Comput. Simul.*, vol. 29, no. 1, pp. 7:1–7:36, Jan. 2019. [Online]. Available: <http://doi.acm.org/10.1145/3299872>
- [102] I. Parvez, A. Rahmati, I. Guvenc, A. I. Sarwat, and H. Dai, “A survey on low latency towards 5g: Ran, core network and caching solutions,” *IEEE Communications Surveys Tutorials*, vol. 20, no. 4, pp. 3098–3130, Fourthquarter 2018.
- [103] M. Karakus and A. Durresi, “Quality of service (qos) in software defined networking (sdn),” *J. Netw. Comput. Appl.*, vol. 80, no. C, pp. 200–218, Feb. 2017. [Online]. Available: <https://doi.org/10.1016/j.jnca.2016.12.019>

- [104] T. Čejka and R. Krejčí, “Configuration of open vswitch using of-config,” in *NOMS 2016 - 2016 IEEE/IFIP Network Operations and Management Symposium*, April 2016, pp. 883–888.
- [105] “Ovsdb protokol,” accessed: 2019-04-04. [Online]. Available: <http://docs.openvswitch.org/en/latest/ref/ovsdb.7/>
- [106] B. Pfaff *et al.*, “The design and implementation of open vswitch,” ser. NSDI’15 conference, CA, USA, 2015, pp. 117–130.
- [107] B. Claise, “Cisco Systems NetFlow Services Export Version 9,” RFC 3954, Oct. 2004. [Online]. Available: <https://rfc-editor.org/rfc/rfc3954.txt>
- [108] J. Crowcroft, “Quality-of-service routing for supporting multimedia applications,” *IEEE Journal on Selected Areas in Communications*, vol. 14, no. 7, pp. 1228–1234, Sep. 1996.
- [109] T. Korkmaz and M. Krunz, “Multi-constrained optimal path selection,” in *Proceedings IEEE INFOCOM 2001. Conference on Computer Communications. Twentieth Annual Joint Conference of the IEEE Computer and Communications Society (Cat. No.01CH37213)*, vol. 2, April 2001, pp. 834–843 vol.2.
- [110] N. M. Garcia, M. Freire, and P. Monteiro, “On the performance of shortest path routing algorithms for modeling and simulation of static source routed networks: an extension to the dijkstra algorithm,” in *International Conf. on Systems and Networks Communications*, vol. 1, August 2007, pp. 1–1.
- [111] J. C. R. Bennett and H. Zhang, “Wf2q: Worst-case fair weighted fair queueing,” in *INFOCOM*, 1996.
- [112] K. P. Kadiyala and J. A. Cobb, “Inter-as traffic engineering with sdn,” in *2017 IEEE Conference on Network Function Virtualization and Software Defined Networks (NFV-SDN)*, Nov 2017, pp. 1–7.
- [113] A. Gupta, L. Vanbever, M. Shahbaz, S. P. Donovan, B. Schlinker, N. Feamster, J. Rexford, S. Shenker, R. Clark, and E. Katz-Bassett, “Sdx: A software defined internet exchange,” *SIGCOMM Comput. Commun. Rev.*, vol. 44, no. 4, pp. 551–562, Aug. 2014.

- [114] X. Jin *et al.*, “Optimizing bulk transfers with software-defined optical wan,” in *Proc. of the 2016 ACM SIGCOMM Conference*, pp. 87–100.
- [115] S. Tomovic and I. Radusinovic, “Traffic engineering approach to virtual-link provisioning in software-defined isp networks,” in *2017 25th Telecommunication Forum (TELFOR)*, Nov 2017, pp. 1–4.
- [116] —, “A new traffic engineering approach for qos provisioning and failure recovery in sdn-based isp networks,” in *2018 23rd International Scientific-Professional Conference on Information Technology (IT)*, Feb 2018, pp. 1–4.
- [117] D. Blake, M. Carlson, E. Davies, Z. Wang, and W. Weiss, “An architecture for differentiated services,” 1998.
- [118] C. Fraleigh *et al.*, “Provisioning ip backbone networks to support latency sensitive traffic,” in *IEEE INFOCOM 2003*, vol. 1, March 2003, pp. 375–385 vol.1.
- [119] K. E. Batcher, “Sorting networks and their applications,” in *Proceedings of the April 30–May 2, 1968, Spring Joint Computer Conference*, ser. AFIPS ’68 (Spring). New York, NY, USA: ACM, 1968, pp. 307–314. [Online]. Available: <http://doi.acm.org/10.1145/1468075.1468121>
- [120] A. Ravindran and T. W. Hill, “Note—a comment on the use of simplex method for absolute value problems,” *Manage. Sci.*, vol. 19, no. 5, pp. 581–582, Jan. 1973. [Online]. Available: <http://dx.doi.org/10.1287/mnsc.19.5.581>
- [121] “Cplex optimization studio,” <http://www-03.ibm.com/software/>. [Online]. Available: <http://www-03.ibm.com/software/products/en/ibmilogcpleoptistud>
- [122] A. Gunnar *et al.*, “Traffic matrix estimation on a large ip backbone: A comparison on real data,” in *Proc. of the 4th ACM SIGCOMM Conference on Internet Measurement*, 2004, pp. 149–160.
- [123] P. G. Harrison and N. M. Patel, *Performance Modelling of Communication Networks and Computer Architectures (International Computer S*, 1st ed. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 1992.
- [124] J. He and J. Rexford, “Date: Distributed adaptive traffic engineering,” in *In Proc. IEEE INFOCOMM*, 2006.

- [125] G. Mavrotas and K. Florios, “An improved version of the augmented ϵ -constraint method (augmecon2) for finding the exact pareto set in multi-objective integer programming problems,” *Appl. Math. Comput.*, vol. 219, no. 18, pp. 9652–9669, May 2013. [Online]. Available: <http://dx.doi.org/10.1016/j.amc.2013.03.002>
- [126] L. J. Wismans, T. Brands, E. C. Van Berkum, and M. C. Bliemer, “Pruning and ranking the pareto optimal set, application for the dynamic multi-objective network design problem,” *Journal of Advanced Transportation*, vol. 48, no. 6, pp. 588–607. [Online]. Available: <https://onlinelibrary.wiley.com/doi/abs/10.1002/atr.1212>
- [127] J. Y. Yen, “An algorithm for finding shortest routes from all source nodes to a given destination in general networks,” *Quart. Appl. Math.*, vol. 27, pp. 526–530, 1970.
- [128] N. L. M. v. Adrichem *et al.*, “Fast recovery in software-defined networks,” in *2014 Third European Workshop on Software Defined Networks*, Sept 2014, pp. 61–66.
- [129] “Iperf3 traffic generator,” <https://iperf.fr>. [Online]. Available: <https://iperf.fr>
- [130] J. Sheu, W. Lin, and G. Chang, “Efficient tcam rules distribution algorithms in software-defined networking,” *IEEE Transactions on Network and Service Management*, vol. 15, no. 2, pp. 854–865, June 2018.
- [131] S. Tomovic and I. Radusinovic, “Traffic engineering approach to virtual-link provisioning in software-defined isp networks,” *Telfor Journal*, vol. 10, no. 1, June 2018.
- [132] R. S. Sutton and A. G. Barto, “Reinforcement learning: An introduction,” *IEEE Transactions on Neural Networks*, vol. 9, no. 5, pp. 1054–1054, Sep. 1998.

KRATKA BIOGRAFIJA

Slavica Slavka Tomović je rođena 05.02.1991. godine u Nikšiću, Republika Crna Gora. Osnovnu školu i Gimnaziju završila je u Nikšiću. Za pokazan uspjeh tokom osnovne i srednje škole nagrađena je diplomom Luča. Na Elektrotehnički fakultet u Podgorici, odsjek Elektronika, telekomunikacije i računari upisala se 2009. godine. Osnovne studije završila je sa prosječnom ocjenom A. Diplomirala je 2013. godine odbranom specijalističkog rada „Implementacija RIP i OSPF protokla rutiranja na Quagga softverskoj platformi“ sa prosječnom ocjenom A. Tokom osnovnih studija bila je korisnik stipendija Opštine Nikšić i stipendija Ministarstva nauke Crne Gore za talentovane studente. Na postdiplomске studije na Elektrotehničkom fakultetu u Podgorici, smjer Telekomunikacije, upisala se u septembru 2013. godine kod mentora prof. dr Igora Radusinovića. Magistarsku tezu pod nazivom: „Algoritmi rutiranja za podršku kvalitetu servisa u softverski definisanim mrežama“ uspješno je odbranila, sa prosječnom ocjenom A, u junu 2015. godine.

Od septembra 2014. godine angažovana je u svojstvu saradnika u nastavi na Univerzitetu Crne Gore. Učestvovala je u izvođenju računskih i laboratorijskih vježbi na više predmeta na Elektrotehničkom fakultetu i Pomorskom fakultetu.

U dosadašnjem naučnom radu bila je autor preko 30 naučnih radova objavljenih u renomiranim međunarodnim i domaćim časopisima i izlaganim u okviru programa međunarodnih i domaćih konferencija. Od toga, objavila je 8 radova u časopisima sa SCI liste, sa ukupnim impact faktorim 20.14. Član je IEEE udruženja. Obavlja dužnost recenzenta za više renomiranih međunarodnih časopisa sa SCI/SCIE liste.

Bila je angažovana na više nacionalnih i evropskih naučnih projekata.

IZJAVA O AUTORSTVU

Potpisani/a: Slavica Tomović

Broj indeksa: 2/2015

Izjavljujem

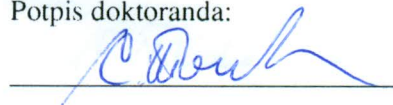
da je doktorska disertacija pod naslovom:

Analiza performansi novih tehnika za inženjering saobraćaja u ISP mrežama naredne generacije

- rezultat sopstvenog istraživačkog rada;
- da predložena disertacija ni u cjelini ni u djelovima nije bila predložena za dobijanje bilo koje diplome prema studijskim programima drugih ustanova visokog obrazovanja;
- da su rezultati korektno navedeni, i
- da nijesam povrijedio autorska i druga prava intelektualne svojine koja pripadaju trećim licima.

Podgorica,
jun 2019. godine

Potpis doktoranda:



**IZJAVA O ISTOVJETNOSTI ŠTAMPANE I ELEKTRONSKE VERZIJE
DOKTORSKOG RADA**

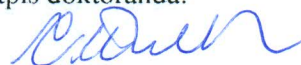
Ime i prezime autora:	Slavica Tomović
Broj indeksa/upisa:	2/2015
Studijski program:	Doktorske studije elektrotehnike
Naslov rada:	Analiza performansi novih tehnika za inženjering saobraćaja u ISP mrežama naredne generacije
Mentor:	Prof. dr Igor Radusinović
Potpisani:	Slavica Tomović

Izjavljujem da je štampana verzija mog doktorskog rada istovjetna elektronskoj verziji koju sam predao za objavljivanje u Digitalni arhiv Univerziteta Crne Gore.

Istovremeno izjavljujem da dozvoljavam objavljivanje mojih ličnih podataka u vezi sa dobijanjem akademskog naziva doktora nauka, odnosno zvanja doktora umjetnosti, kao što su ime i prezime, godina i mjesto rođenja, naziv disertacije i datum odbrane rada.

Podgorica,
jun 2019. godine

Potpis doktoranda:



IZJAVA O KORIŠĆENJU

Ovlašćujem Univerzitetsku biblioteku da u Digitalni arhiv Univerziteta Crne Gore pohrani moju doktorsku disertaciju pod naslovom:

Analiza performansi novih tehnika za inženjering saobraćaja u ISP mrežama naredne generacije

koja je moje autorsko djelo.

Disertaciju sa svim prilogima predao sam u elektronskom formatu pogodnom za trajno arhiviranje.

Moju doktorsku disertaciju pohranjenu u Digitalni arhiv Univerziteta Crne Gore mogu da koriste svi koji poštuju odredbe sadržane u odabranom tipu licence Kreativne zajednice (Creative Commons) za koju sam se odlučio.

1. Autorstvo
2. Autorstvo – nekomercijalno
3. **Autorstvo – nekomercijalno – bez prerade**
4. Autorstvo – nekomercijalno – dijeliti pod istim uslovima
5. Autorstvo – bez prerade
6. Autorstvo – dijeliti pod istim uslovima

Podgorica,
jun 2019. godine

Potpis doktoranda:



1. **Autorstvo.** Licenca sa najširim obimom prava korišćenja. Dozvoljavaju se prerade, umnožavanje, distribucija i javno saopštavanje djela, pod uslovom da se navede ime izvornog autora (onako kako je izvorni autor ili davalac licence odredio). Djelo se može koristiti i u komercijalne svrhe.
2. **Autorstvo – nekomercijalno.** Dozvoljavaju se prerade, umnožavanje, distribucija i javno saopštavanje djela, pod uslovom da se navede ime izvornog autora (onako kako je izvorni autor ili davalac licence odredio). Komercijalna upotreba djela nije dozvoljena.
3. **Autorstvo – nekomercijalno – bez prerade.** Licenca kojom se u najvećoj mjeri ograničavaju prava korišćenja djela. Dozvoljava se umnožavanje, distribucija i javno saopštavanje djela, pod uslovom da se navede ime izvornog autora (onako kako je izvorni autor ili davalac licence odredio). Djelo se ne može mijenjati, preoblikovati ili koristiti u drugom djelu. Komercijalna upotreba djela nije dozvoljena.
4. **Autorstvo – nekomercijalno – dijeliti pod istim uslovima.** Dozvoljava se umnožavanje, distribucija, javno saopštavanje i prerada djela, pod uslovom da se navede ime izvornog autora (onako kako je izvorni autor ili davalac licence odredio). Ukoliko se djelo mijenja, preoblikuje ili koristi u drugom djelu, prerada se mora distribuirati pod istom ili sličnom licencom. Djelo i prerade se ne mogu koristiti u komercijalne svrhe.
5. **Autorstvo – bez prerade.** Dozvoljava se umnožavanje, distribucija i javno saopštavanje djela, pod uslovom da se navede ime izvornog autora (onako kako je izvorni autor ili davalac licence odredio). Djelo se ne može mijenjati, preoblikovati ili koristiti u drugom djelu. Licenca dozvoljava komercijalnu upotrebu djela.
6. **Autorstvo – dijeliti pod istim uslovima.** Dozvoljava se umnožavanje, distribucija i javno saopštavanje djela, pod uslovom da se navede ime izvornog autora (onako kako je izvorni autor ili davalac licence odredio). Ukoliko se djelo mijenja, preoblikuje ili koristi u drugom djelu, prerade se moraju distribuirati pod istom ili sličnom licencom. Ova licenca dozvoljava komercijalnu upotrebu djela i prerada. Slična je softverskim licencama, odnosno licencama otvorenog koda.