Univerzitet Crne Gore
ELEKTROTEHNIČKI FAKULTET
81000 Podgorica, Dž. Vašingtona bb, tel. (020) 245 839, fax: (020) 245 873
Ž.R. 510-255-51, **PIB:** 02016702 302, **PDV:** 30/31-03951-6

UCG
Univerzitet Crne Gore

Broj: 02/1-773/1
Datum: 28.05. 2019

UNIVERZITET CRNE GORE

- Centru za doktorske studije -

- Senatu -

O V D J E

U prilogu dostavljamo Odluku Vijeća Elektrotehničkog fakulteta sa sjednice od 28.05.2019. godine i obrazac **D2**, sa pratećom dokumentacijom, za kandidatkinju MSc **Slavicu Tomović**, na dalji postupak.

D E K A N,
Prof. dr Zoran Veljović

Univerzitet Crne Gore
UCG
Elektrotehnički fakultet

ISPUNJENOST USLOVA DOKTORANDA

OPŠTI PODACI O DOKTORANDU			
Titula, ime, ime roditelja, prezime	MSc Slavica Slavko Tomović		
Fakultet	Elektrotehnički fakultet Podgorica		
Studijski program	Doktorske studije Elektrotehnike		
Broj indeksa	2/15		
NAZIV DOKTORSKE DISERTACIJE			
Na službenom jeziku	Analiza performansi novih tehnika za inženjering saobraćaja u ISP mrežama naredne generacije		
Na engleskom jeziku	Performance analysis of new traffic engineering techniques for next generation ISP networks		
Naučna oblast	Telekomunikacije		
MENTOR/MENTORI			
Prvi mentor	Prof. dr Igor Radusinović	Elektrotehnički fakultet, Univerzitet Crne Gore, Podgorica, Crna Gora	Telekomunikacije
KOMISIJA ZA PREGLED I OCJENU DOKTORSKE DISERTACIJE			
Prof. dr Milica Pejanović Đurišić		Elektrotehnički fakultet, Univerzitet Crne Gore, Podgorica, Crna Gora	Telekomunikacije
Prof. dr Igor Radusinović		Elektrotehnički fakultet, Univerzitet Crne Gore, Podgorica, Crna Gora	Telekomunikacije
Prof. dr Zoran Čiča		Elektrotehnički fakultet, Univerzitet Beograd, Srbija	Telekomunikacije

Datum značajni za ocjenu doktorske disertacije	
Sjednica Senata na kojoj je data saglasnost na ocjenu teme i kandidata	16.10.2017.
Dostavljanja doktorske disertacije organizacionoj jedinici i saglasanost mentora	23.05.2019. 23.05.2019.
Sjednica Vijeća organizacione jedinice na kojoj je dat prijedlog za imenovanje komisija za pregled i ocjenu doktorske disertacije	28.05.2019.

ISPUNJENOST USLOVA DOKTORANDA
U skladu sa članom 38 pravila doktorskih studija kandidat **je** cjelokupna ili dio sopstvenih istraživanja vezanih za doktorsku disertaciju publikovao u časopisu sa **(SCI/SCIE)/(SSCI/A&HCI)** liste kao prvi autor.
Spisak radova doktoranda iz oblasti doktorskih studija koje je publikovao u časopisima sa (upisati odgovarajuću listu)

RADOVI PUBLIKOVANI U ČASOPISIMA SA SCI LISTE:

1. S. Tomović and I. Radusinović, "RO-RO: Routing Optimality - Reconfiguration Overhead Balance in Software-Defined ISP Networks," in *IEEE Journal on Selected Areas in Communications*, vol. 37, no. 5, pp. 997-1011, May 2019.
   DOI: 10.1109/JSAC.2019.2906762
   Print ISSN: 0733-8716
   Link na rad: https://ieeexplore.ieee.org/document/8672627
   Informacija o IMPACT faktoru časopisa:
   https://ieeexplore.ieee.org/xpl/RecentIssue.jsp?punumber=49

2. S. Tomovicć and I. Radusinović "An effective use of SDN for virtual-link provisioning in ISP networks," in *IEICE Transactions on Communications*, Vol.E102-B, No.4, pp.855-864, April 2019.
   DOI: 10.1587/transcom.2018EBP3191
   ISSN: 0916-8516
   Link na rad:
   https://search.ieice.org/bin/summary.php?id=e102-b_4_855&category=-&lang=E&year=2019&abst=
   Informacija o IMPACT faktoru časopisa: http://www.ieice.org/cs/jpn/EB/index.html

3. S. Tomović and I. Radusinović, "Mapping Application Requirements to Virtualization-Enabled Software Defined WSN," *Wireless Personal Communications*, Vol. 97, Issue 2, November 2017, 1693-1709.
   DOI: https://doi.org/10.1007/s11277-017-4650-0
   ISSN: 0929-6212
   Link na rad: https://link.springer.com/article/10.1007/s11277-017-4650-0
   Informacija o IMPACT faktoru časopisa: https://link.springer.com/journal/11277

4. G. Davoli, W. Cerroni, S. Tomović, C. Buratti, C. Contoli, F Callegati, "Intent-based service management for heterogeneous software-defined infrastructure domains," *International Journal of Network Management*, 2019; e2051.
   DOI: https://doi.org/10.1002/nem.2051
   ISSN: 1055-7148
   Link na rad: https://onlinelibrary.wiley.com/doi/full/10.1002/nem.2051
   Informacija o IMPACT faktoru časopisa:
   https://onlinelibrary.wiley.com/journal/10991190

5. S. Tomović, K. Yoshigoe, I. Maljevic, I. Radusinovic, "Software-Defined Fog Network Architecture for IoT," *Wireless Personal Communications*, Vol. 92, No. 1, pp. 181-196, January 2017.
   DOI: https://doi.org/10.1007/s11277-016-3845-0
   ISSN: 0929-6212
   Link na rad: https://link.springer.com/article/10.1007/s11277-016-3845-0
   Informacija o IMPACT faktoru časopisa: https://link.springer.com/journal/11277

6. G. Gardašević, M. Veletić, N. Maletić, D. Vasiljević, I. Radusinović, S. Tomović, M. Radonjić, "The IoT Architectural Framework, Design Issues and Application Domains", *Wireless Personal* Communications, Vol. 91, No. 1, pp. 127-148, January 2017.
   DOI: https://doi.org/10.1007/s11277-016-3842-3
   ISSN: 0929-6212
   Link na rad: https://link.springer.com/article/10.1007/s11277-016-3842-3
   Informacija o IMPACT faktoru časopisa: https://link.springer.com/journal/11277

7. S. Tomović, M. Pejanovic-Djurisic, I. Radusinovic, "SDN based Mobile Networks: Concepts and Benefits", *Wireless Personal Communications*, Vol. 78, No. 3, pp. 1629-1644, 2014.
   DOI: https://doi.org/10.1007/s11277-014-1909-6
   ISSN: 0929-6212
   Link na rad: https://link.springer.com/article/10.1007/s11277-014-1909-6
   Informacija o IMPACT faktoru časopisa: https://link.springer.com/journal/11277

RADOVI PUBLIKOVANI U DRUGIM ČASOPISIMA

1. S. Tomovic, I. Radusinovic, "Traffic Engineering Approach to Virtual-link Provisioning in Software-defined ISP Networks," *Telfor Journal*, Vol. 10, No. 1, pp. 14-20, 2018, DOI: DOI:10.1109/telfor.2017.8249296, ISSN 1821-3251 (SCOPUS baza).
2. G. Gogic, S. Tomovic, I. Radusinovic, "Heterogeneous implementation of OpenFlow data-centre testbed", ETF Journal of Electrical Engineering, Vol. 23, No. 1, pp. 1-10, November 2017.

RADOVI IZLOŽENI NA KONFERENCIJAMA

1. S. Tomovic and I. Radusinovic, "A new traffic engineering approach for QoS provisioning and failure recovery in SDN-based ISP networks," 2018 23rd International Scientific-Professional Conference on Information Technology (IT), Žabljak, 2018, pp. 1-4. DOI: 10.1109/SPIT.2018.8350854, ISBN: 978-1-5386-3621-3.
2. S. Tomovic and I. Radusinovic, "Traffic engineering approach to virtual-link provisioning in software-defined ISP networks," 2017 25th Telecommunication Forum (TELFOR), Belgrade, 2017, pp. 1-4, DOI: 10.1109/TELFOR.2017.8249296, ISBN: 978-1-5386-3074-7.
3. G. Gogic. S. Tomovic, I. Radusinovic, "Performance evaluation of OpenFlow data centre network testbed", Proc. 22th Conference on Information Technologies IT 17, Žabljak, Montenegro, February 2017.
4. S. Tomovic et al., "An architecture for QoS-aware service deployment in software-defined IoT networks," 2017 20th International Symposium on Wireless Personal Multimedia Communications (WPMC), 2017, pp. 561-567, DOI: 10.1109/WPMC.2017.8301875, 978-1-5386-2769-3.
5. S. Tomovic, N. Lekic, G. Gardasevic, I. Radusinovic, "A New Approach to Dynamic Routing in SDN Networks", 2016 18th Mediterranean Electrotechnical Conference (MELECON), pp. 1-6, Lemesos, Cyprus, April 2016, DOI: 10.1109/MELCON.2016.7495433, Electronic ISBN: 978-1-5090-0058-6.

6. S. Tomovic, I. Radusinovic, "Fast and Efficient Dandwidth-delay Constrained Routing Algorithm for SDN Networks" ,IEEE NetSoft Conference, pp. 303-311, Seoul, South Korea, June 2016, DOI: 10.1109/NETSOFT.2016.7502426, Electronic ISBN: 978-1-4673-9486-4.
7. S. Tomovic, N. Prasad, I. Radusinovic, "Performance Comparison of QoS Routing Algorithms Applicable to Large-Scale SDN Networks", Proc. of IEEE Eurocon 2015, pp. 172-177, Salamanka, Spain, September 2015, DOI: 10.1109/EUROCON.2015.7313698, Electronic ISBN: 978-1-4799-8569-2.
8. S. Tomovic, M. Radonjic, I. Radusinovic, "Bandwidth-Delay Constrained Routing Algorithms for Backbone SDN Networks", Proc. of TELSIKS 2015, pp. 227-230, Niš, Serbia, October 2015, DOI: 10.1109/TELSKS.2015.7357775, ISBN: 978-1-4673-7515-3.
9. M. Ratkovic, S. Tomovic, N. Zaric, M. Radonjic, I. Radusinovic, "SDN Network Emulation with Mininet Software Tool", Proc. 20th Conference on Information Technologies IT 15, pp. 80-83, Žabljak, Montenegro, February 2015.

Obrazloženje mentora o korišćenju doktorske disertacije u publikovanim radovima

Doktorand MSc Slavica Tomović je većinu svojih istraživanja na kojima je zasnovana doktorska disertacija prezentovala kroz 7 radova, koji su publikovani u renomiranim međunarodnim časopisima sa SCI/SCIE liste, sa **kumulativnim IMPACT Factor-om 13.833**. Na pet radova kandidat je **prvi** autor. Dio rezultata je objavljen i u radovima (njih ukupno 9) koji su izloženi na međunarodnim i regionalnim konferencijama i kroz dva rada u drugim časopisima. Treba istaći da je doktorand autor i drugih radova mimo oblasti disertacije, što se može vidjeti iz bibliografije koja je data u prilogu ovog obrasca. U nastavku slijedi obrazloženje ključnih rezultata publikovanih kroz 7 radova u renomiranim međunarodnim časopisima, koji predstavljaju i temelj predmetne doktorske disertacije.

Naučni rad „*RO-RO: Routing Optimality - Reconfiguration Overhead Balance in Software-Defined ISP Networks*“ publikovan je u renomiranom časopisu *IEEE Journal on Selected Areas in Communications* sa IMPACT Factor-om **7.172**, i u njemu je predstavljen originalni algoritam za inženjering saobraćaja u mrežama Internet Servis Provajdera (ISP). Predloženi algoritam baziran je na višekriterijumskom optimizacionom modelu koji teži simultano da minimizuje maksimalnu iskorišćenost linkova i broj rekonfiguracija. Razmatrani optimizacioni kriterijumi su međusobno suprostavljeni, pa je u radu predložen i algoritam za pronalaženje najefikasnijeg kompromisnog rešenja iz Pareto fronta. Ovaj algoritam baziran je na ideji minimizacije Lyapunov *drift-plus-penalty* funkcije, pri čemu je *penalty* funkcijom predstavljeno odstupanje odgovarajuće šeme rutiranja od optimalne (u pogledu maksimalne iskorišćenosti linka), dok je *drift* funkcija korišćena za praćenje promjena u zauzetosti virtuelnog bafera koji modeluje uticaj rekonfiguracija tokom vremena. Cilj predloženog algoritma je stabilizacija virtuelnog bafera (tj. minimizacija njegovog *drift-a*) uz što optimalniju konfiguraciju ruta. Sadržaj rada je obrađen u petoj i šestoj glavi disertacije.

U naučnom radu „*An effective use of SDN for virtual-link provisioning in ISP networks*“, koji je publikovan u renomiranom časopisu *IEICE Transactions on Communications* sa IMPACT Factor-om **1.09**, predložen je originalni dizajn kontrolera za softverski definisane ISP mreže. Predloženi model SDN (*Software Defined Networking*) kontrolera omogućava fleksibilnu alokaciju virtuelnih linkova na zahtjev korisnika, uz efikasno korišćenje dostupnih mrežnih resursa. Funkcionalnosti rutiranja i inženjeringa saobraćaja organizovane su u dva odvojena modula: *online* i *offline* modul, respektivno. Modul za rutiranje baziran je na računski jednostavnom QoS (*Quality of Service*) algoritmu rutiranja, sa ciljem smanjenja vremena odziva na zahtjev korisnika. Distribucija saobraćaja je periodično optimizovana od strane *offline* modula, izvršavanjem TE (*Traffic Engineering*) algoritma koji kao ograničenja razmatra zahtjeve korisnika u pogledu propusnosti, kašnjenja i pouzdanosti. Rad sadrži opsežne simulacione rezultate, koji su verifikovani eksperimentalno. Sadržaj rada je obrađen u četvrtoj i šestoj glavi disertacije.

U radu "*Intent-based service management for heterogeneous software-defined infrastructure domains*" koji je objavljen u renomiranom časopisu *International Journal of Network Management* sa IMPACT Factor-om **1.35**, predloženo je rešenje za alokaciju virtuelnih linkova sa VNF (*Virtual Network Function*) servisnim lancima u heterogenim softverski definisanim mrežama. Predložena mrežna arhitektura bazirana je na SDN i NFV tehnologijama, kao što je objašnjeno u drugom poglavlju disertacije. SDN kontrolno okruženje koje je predstavljeno u četvrtom poglavlju disertacije nadograđeno je sa modulima za upravljanje virtuelizovanom mrežnom infrastrukturom, koji vrše apstrakciju heterogenih resursa u različitim administrativnim domenima. Na ovaj način omogućeno je kreiranje *end-to-end* servisnih lanaca sa garantovanim kvalitetom servisa, transparetno u odnosu na krajnjeg korisnika. Rad sadrži detalje vezane za implementaciju prototipa predmetne mrežne arhitetkure i rezultate analize njegovih performansi.

U radu "*Mapping Application Requirements to Virtualization-Enabled Software Defined WSN*" koji je objavljen u renomiranom časopisu *Wireless Personal Communications* sa IMPACT Factor-om **1.2**, predložen je algoritam za inženjering saobraćaja u *Internet of Things* (IoT) okruženjima sa softverski definisanim senzorskim čvorovima. Razmatran je model virtuelizovane senzorske mreže, čiji se resursi koriste od strane većeg broja IoT aplikacija sa potencijalno konfliktnim zahtjevima. Zahtjevi različitih aplikacija modelovani su acikličnim grafovima i mapirani sa resursima SDN infrastrukture u skladu sa odlukama TE algoritma koji se izvršava na mrežnom kontroleru. U drugom poglavlju disertacije diskutovan je značaj ovog servisnog scenarija za mreže naredne generacije.

Rad "*Software-Defined Fog Network Architecture for IoT*", koji je objavljen u časopisu *Wireless Personal Communications* sa IMPACT Factor-om **1.2**, predlaže IoT arhitekturu baziranu na konceptu softverski definisanog umrežavanja. Akcenat je stavljen na primjeni SDN-a za orkestraciju *Fog* servisa. Dodatno, razmatran je problem inženjeringa saobraćaja i upravljanja radio pristupnom mrežom. Značaj ovog istraživanja istaknut je u drugom poglavlju ove disertacije.

Istraživanja na kojima je zasnovana doktorska disertacija iskorišćena su i u radu "*The IoT Architectural Framework, Design Issues and Application Domains*", koji je publikovan u časopisu *Wireless Personal Communications* sa IMPACT Factor-om **1.2**. U radu su analizirani ključni izazovi u dizajnu različitih nivoa IoT arhitekture, od fizičkog nivoa do nivoa aplikacije. U jednom od segmenata rada predstavljen je problem upravljanja saobraćajem u IoT okruženjima, a kao rešenje predložena je primjena softverski definisanih mreža.

Konačno, u radu "*SDN based Mobile Networks: Concepts and Benefits*" dat je pregled rešenja za softverizaciju mobilne celularne mreže. Rad je publikovan u renomiranom časopisu *Wireless Personal Communications* sa IMPACT Factor-om **1.2**. U fokusu rada su problemi minimizacije interferencije, upravljanja saobraćajem i mrežne virtuelizacije, od kojih su poslednja dva predmet disertacije.

Jedan dio rezultata kandidata publikovan je kroz 7 radova na renomiranim međunarodnim konferencijma, uključujući i konferenciju NetSoft, koja je široko poznata kao jedna od najznačajnijih konferencija u oblasti mrežne softverizacije, zatim konferencije Eurocon, Melecon, Telfor i Telsiks. Na međunarodnoj konferenciji ISWCS (*International Symposium for Wireless*

Communication Systems), koja je 2017. godine održana u Bolonji, kandidat je dobio priznanje za najbolji demo, predstavljajući prototip SDN kontrolera koji je razvijen u toku rada na disertaciji. Dva rada objavljena su i na konferenciji *Informacione tehnologije* (IT), koja ima regionalni značaj.
Datum i ovjera (pečat i potpis odgovorne osobe)
U Podgorici, 28.05.2019. MP DEKAN

Prilog dokumenta sadrži:

1. Potvrdu o predaji doktorske disertacije organizacionoj jedinici
2. Odluku o imenovanju komisije za pregled i ocjenu doktorske disertacije
3. Kopiju rada publikovanog u časopisu sa odgovarajuće liste
4. Biografiju i bibliografiju kandidata
5. Biografiju i bibliografiju članova komisije za pregled i ocjenu doktorske disertacije sa potvrdom o izboru u odgovarajuće akademsko zvanje i potvrdom da barem jedan član komisije nije u radnom odnosu na Univerzitetu Crne Gore

Univerzitet Crne Gore
UCG
Univerzitet Crne Gore

Univerzitet Crne Gore
ELEKTROTEHNIČKI FAKULTET
81000 Podgorica, Dž. Vašingtona bb, tel. (020) 245 839, fax: (020) 245 873
Ž.R. 510-255-51, **PIB:** 02016702 302, **PDV:** 30/31-03951-6

Broj: 02/1 - 757
Datum: 27. 05. 2019

Na osnovu službene evidencije i dokumentacije Elektrotehničkog fakulteta u Podgorici, izdaje se

POTVRDA

MSc Slavica Tomović, studentkinja doktorskih studija na Elektrotehničkom fakultetu u Podgorici, dana 23.05.2019. godine dostavila je ovom Fakultetu doktorsku disertaciju pod nazivom: „Analiza performansi novih tehnika za inženjering saobraćaja u ISP mrežama naredne generacije", na dalji postupak.

UNIVERZITET CRNE GORE UCG Elektrotehnički fakultet

DEKAN,

Prof. dr Zoran Veljović

Univerzitet Crne Gore
UCG
Univerzitet Crne Gore
ELEKTROTEHNIČKI FAKULTET
81000 Podgorica, Dž. Vašingtona bb, tel. (020) 245 839, fax: (020) 245 873
Ž.R. 510-255-51, **PIB:** 02016702 302, **PDV:** 30/31-03951-6

Broj: 02/1-773
Datum: 28.05.2019

Na osnovu člana 64 Statuta Univerziteta Crne Gore, u vezi sa članom 41 Pravila doktorskih studija, na predlog Komisije za doktorske studije, Vijeće Elektrotehničkog fakulteta u Podgorici, na sjednici od 28.05.2019. godine, donijelo je

ODLUKU

I Utvrđuje se da su ispunjeni uslovi iz Pravila doktorskih studija za dalji rad na doktorskoj disertaciji **„Analiza performansi novih tehnika za inženjering saobraćaja u ISP mrežama naredne generacije"**, kandidatkinje MSc **Slavice Tomović.**

II Predlaže se Komisija za ocjenu navedene doktorske disertacije, u sastavu:

1. Dr Milica Pejanović-Đurišić, redovni profesor Elektrotehničkog fakulteta Univerziteta Crne Gore,
2. Dr Igor Radusinović, redovni profesor Elektrotehničkog fakulteta Univerziteta Crne Gore,
3. Dr Zoran Čiča, vanredni profesor Elektrotehničkog fakulteta Univerziteta u Beogradu.

Komisija iz tačke II ove Odluke podnijeće Izvještaj Vijeću Fakulteta u roku od 45 dana od dana imenovanja.

-VIJEĆE ELEKTROTEHNIČKOG FAKULTETA-

DEKAN,
Prof. dr Zoran Veljović

Univerzitet Crne Gore
UCG
Elektrotehnički fakultet

Dostavljeno:
- Senatu,
- Centru za doktorske studije,
- u dosije,
- a/a.

SPISAK RADOVA SA REZULTATIMA IZ DOKTORSKE TEZE

RADOVI PUBLIKOVANI U ČASOPISIMA SA SCI LISTE:

1. S. Tomovic and I. Radusinovic, "RO-RO: Routing Optimality - Reconfiguration Overhead Balance in Software-Defined ISP Networks," in *IEEE Journal on Selected Areas in Communications*, vol. 37, no. 5, pp. 997-1011, May 2019.
   DOI: 10.1109/JSAC.2019.2906762
   Print ISSN: 0733-8716
   Link na rad: https://ieeexplore.ieee.org/document/8672627
   Informacija o IMPACT faktoru časopisa:
   https://ieeexplore.ieee.org/xpl/RecentIssue.jsp?punumber=49

2. S. Tomovic and I. Radusinovic, "An effective use of SDN for virtual-link provisioning in ISP networks," in *IEICE Transactions on Communications*, Vol.E102-B, No.4, pp.855-864, April 2019.
   DOI: 10.1587/transcom.2018EBP3191
   ISSN: 0916-8516
   Link na rad:
   https://search.ieice.org/bin/summary.php?id=e102-b_4_855&category=-&lang=E&year=2019&abst=
   Informacija o IMPACT faktoru časopisa: http://www.ieice.org/cs/jpn/EB/index.html

3. S. Tomovic and I. Radusinovic, "Mapping Application Requirements to Virtualization-Enabled Software Defined WSN," *Wireless Personal Communications*, Vol. 97, Issue 2, November 2017, 1693-1709.
   DOI: https://doi.org/10.1007/s11277-017-4650-0
   ISSN: 0929-6212
   Link na rad: https://link.springer.com/article/10.1007/s11277-017-4650-0
   Informacija o IMPACT faktoru časopisa: https://link.springer.com/journal/11277

4. G. Davoli, W. Cerroni, S. Tomovic, C. Buratti, C. Contoli, F Callegati, "Intent-based service management for heterogeneous software-defined infrastructure domains," *International Journal of Network Management*, 2019; e2051.
   DOI: https://doi.org/10.1002/nem.2051
   ISSN: 1055-7148
   Link na rad: https://onlinelibrary.wiley.com/doi/full/10.1002/nem.2051
   Informacija o IMPACT faktoru časopisa: https://onlinelibrary.wiley.com/journal/10991190

5. S. Tomovic, K. Yoshigoe, I. Maljevic, I. Radusinovic, "Software-Defined Fog Network Architecture for IoT," *Wireless Personal Communications*, Vol. 92, No. 1, pp. 181-196, January 2017.
   DOI: https://doi.org/10.1007/s11277-016-3845-0
   ISSN: 0929-6212
   Link na rad: https://link.springer.com/article/10.1007/s11277-016-3845-0
   Informacija o IMPACT faktoru časopisa: https://link.springer.com/journal/11277

6. G. Gardasevic, M. Veletic, N. Maletic, D. Vasiljevic, I. Radusinovic, S. Tomovic, M. Radonjic, "The IoT Architectural Framework, Design Issues and Application Domains", *Wireless Personal* Communications, Vol. 91, No. 1, pp. 127-148, January 2017.
   DOI: https://doi.org/10.1007/s11277-016-3842-3
   ISSN: 0929-6212
   Link na rad: https://link.springer.com/article/10.1007/s11277-016-3842-3
   Informacija o IMPACT faktoru časopisa: https://link.springer.com/journal/11277

7. S. Tomovic, M. Pejanovic-Djurisic, I. Radusinovic, "SDN based Mobile Networks: Concepts and Benefits", *Wireless Personal Communications*, Vol. 78, No. 3, pp. 1629-1644, 2014.
   DOI: https://doi.org/10.1007/s11277-014-1909-6
   ISSN: 0929-6212
   Link na rad: https://link.springer.com/article/10.1007/s11277-014-1909-6
   Informacija o IMPACT faktoru časopisa: https://link.springer.com/journal/11277

RADOVI PUBLIKOVANI U DRUGIM ČASOPISIMA

1. S. Tomovic, I. Radusinovic, "Traffic Engineering Approach to Virtual-link Provisioning in Software-defined ISP Networks," *Telfor Journal*, Vol. 10, No. 1, pp. 14-20, 2018, DOI: DOI:10.1109/telfor.2017.8249296, ISSN 1821-3251 (SCOPUS baza).
2. G. Gogic, S. Tomovic, I. Radusinovic, "Heterogeneous implementation of OpenFlow data-centre testbed", ETF Journal of Electrical Engineering, Vol. 23, No. 1, pp. 1-10, November 2017.

RADOVI IZLOŽENI NA KONFERENCIJAMA

1. S. Tomovic and I. Radusinovic, "A new traffic engineering approach for QoS provisioning and failure recovery in SDN-based ISP networks," 2018 23rd International Scientific-Professional Conference on Information Technology (IT), Žabljak, 2018, pp. 1-4. DOI: 10.1109/SPIT.2018.8350854, ISBN: 978-1-5386-3621-3.
2. S. Tomovic and I. Radusinovic, "Traffic engineering approach to virtual-link provisioning in software-defined ISP networks," 2017 25th Telecommunication Forum (TELFOR), Belgrade, 2017, pp. 1-4, DOI: 10.1109/TELFOR.2017.8249296, ISBN: 978-1-5386-3074-7.
3. G. Gogic. S. Tomovic, I. Radusinovic, "Performance evaluation of OpenFlow data centre network testbed", Proc. 22th Conference on Information Technologies IT 17, Žabljak, Montenegro, February 2017.
4. S. Tomovic et al., "An architecture for QoS-aware service deployment in software-defined IoT networks," 2017 20th International Symposium on Wireless Personal Multimedia Communications (WPMC), 2017, pp. 561-567, DOI: 10.1109/WPMC.2017.8301875, 978-1-5386-2769-3.
5. S. Tomovic, N. Lekic, G. Gardasevic, I. Radusinovic, "A New Approach to Dynamic Routing in SDN Networks", 2016 18th Mediterranean Electrotechnical Conference (MELECON), pp. 1-6, Lemesos, Cyprus, April 2016, DOI: 10.1109/MELCON.2016.7495433, Electronic ISBN: 978-1-5090-0058-6.
6. S. Tomovic, I. Radusinovic, "Fast and Efficient Dandwidth-delay Constrained Routing Algorithm for SDN Networks" ,IEEE NetSoft Conference, pp. 303-311, Seoul, South Korea, June 2016, DOI: 10.1109/NETSOFT.2016.7502426, Electronic ISBN: 978-1-4673-9486-4.

7. S. Tomovic, N. Prasad, I. Radusinovic, "Performance Comparison of QoS Routing Algorithms Applicable to Large-Scale SDN Networks", Proc. of IEEE Eurocon 2015, pp. 172-177, Salamanka, Spain, September 2015, DOI: 10.1109/EUROCON.2015.7313698, Electronic ISBN: 978-1-4799-8569-2.
8. S. Tomovic, M. Radonjic, I. Radusinovic, "Bandwidth-Delay Constrained Routing Algorithms for Backbone SDN Networks", Proc. of TELSIKS 2015, pp. 227-230, Niš, Serbia, October 2015, DOI: 10.1109/TELSKS.2015.7357775, ISBN: 978-1-4673-7515-3.
9. M. Ratkovic, S. Tomovic, N. Zaric, M. Radonjic, I. Radusinovic, "SDN Network Emulation with Mininet Software Tool", Proc. 20th Conference on Information Technologies IT 15, pp. 80-83, Žabljak, Montenegro, February 2015.

IEEE Journal on Selected Areas in Communications

7.172	0.04435	2.716
Impact Factor	Eigenfactor	Article Influence Score

IEEE Journal on Selected Areas in Communications focuses on all telecommunications, including telephone, telegraphy, facsimile, and point-to-point television, by electromagnetic propagation. View Full Aims & Scope

IEEE Journal on Selected Areas in Communications focuses on all telecommunications, including telephone, telegraphy, facsimile, and point-to-point television, by electromagnetic propagation. View Full Aims & Scope

Author Resources

- Submission Guidelines
- Submit Your Manuscript
- Author Center
- Become a Reviewer
- Open Access Publishing Options

Meet the Editor

Editor-in-Chief
Raouf Boutaba

David R. Sheraton School of Computer Science
University of Waterloo
200 University Avenue West
Waterloo, Ontario, N2L 3G1, Canada
1 (519) 888-4820

rboutaba@uwaterloo.ca

Popular Articles | Latest Published Articles

What Will 5G Be?
Jeffrey G. Andrews ; Stefano Buzzi ; Wan Choi ; Stephen V. Hanly ; Angel Lozano ; Anthony C. K. Soong ; Jianzhong Charlie Zhang

C2TCP: A Flexible Cellular TCP To Meet Stringent Delay Requirements
Soheil Abbasloo ; Yang Xu ; H. Jonathan Chao

5G: A Tutorial Overview Of Standards, Trials, Challenges, Deployment, And Practice
Mansoor Shafi ; Andreas F. Molisch ; Peter J. Smith ; Thomas Haustein ; Peiying Zhu ; Prasan De Silva ; Fredrik Tufvesson ; Anass Benjebbour ; Gerhard Wunder

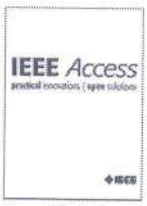
A Survey On Non-Orthogonal Multiple Access For 5G Networks: Research Challenges And Future Trends
Zhiguo Ding ; Xianfu Lei ; George K. Karagiannidis ; Robert Schober ; Jinhong Yuan ; Vijay K. Bhargava

Millimeter Wave Channel Modeling And Cellular Capacity Evaluation
Mustafa Riza Akdeniz ; Yuanpeng Liu ; Mathew K. Samimi ; Shu Sun ; Sundeep Rangan ; Theodore S. Rappaport ; Elza Erkip

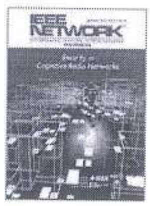
View All

Related Magazines or Journals

Journal of Communications and Networks



IEEE Access



IEEE Network

IEICE Communications Society

IEICE Communications Society

Transactions on Communications

| Trans. Commun. Home | Information for Authors | Search |

English / Japanese

For Readers

- Search
- Open Access Papers
- Transactions Online
- Terms of Use for Individual
- Terms of Use for Institutions

For Authors

- Information for Authors
- Manuscript Registration Form for the Initial Submission
- LaTeX style/class files for manuscripts preparation
- Word template for manuscripts preparation
- A Professional Plain English Handbook (Only for members)

News

Impact Factor: 1.090 (2017)

POSITION PAPER, a new category

From Editorial Committee

Editor's Message

Committee Members

Call For Papers (Special Section)

The IEICE Transactions on Communications originates from the IEICE Transactions (Japanese language) that started in 1918, and took the current form in 1991 as one of the four English-language Transactions. It receives about 600 paper-submissions every year, of which over 60 percent come from countries outside Japan. Among those contributions, only high quality and outstanding papers are published on a monthly basis. The peer-review process is careful but yet efficient; it typically takes less than 12 months from submission to publication.

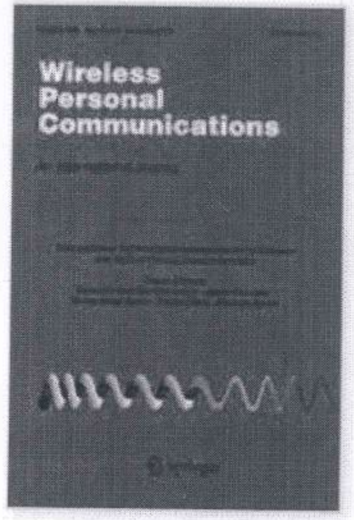
In addition to the regular section, the IEICE Transactions on Communications features Special Sections/Issues providing comprehensive coverage of timely topics nearly every month, and there invites papers from leading experts in specialist fields as a complete and authoritative overview of key areas of research. Specially selected papers, including invited papers, are available online to the public free of charge under the open access paper policy.

The IEICE Transactions on Communications covers the following research fields: Fundamental Theories for Communications, Fiber-Optic Communications,

Wireless Personal Communications

An International Journal

ISSN: 0929-6212 (Print) 1572-834X (Online)



Description

Wireless Personal Communications is an archival, peer reviewed, scientific and technical journal addressing mobile communications and computing. It investigates theoretical, engineering, and experimental aspects of radio communications, voice, data, images, and multimedia.

A partial list of topics includes propagation, system models, speech and image coding, multiple access techniques, protocols performance evaluation, radio ... show all

Browse Volumes & Issues

Impact Factor	Available
1.200	1994 - 2019
Volumes	Issues
106	385
Articles	Open Access
8,186	107 Articles

Latest Articles

OriginalPaper

Design Optimization of EDFA for 16 × 10 Gbps Data Rate DWDM System

RO-RO: Routing Optimality - Reconfiguration Overhead Balance in Software-Defined ISP Networks

Slavica Tomovic, *Student Member, IEEE*, and Igor Radusinovic, *Member, IEEE*

***Abstract*—Software-defined networking provides a promising solution for traffic engineering (TE) by utilizing a centralized controller to remotely configure programmable network switches. In order to achieve optimal traffic distribution and meet the Quality of Service (QoS) requirements, the SDN controller must frequently solve complex optimization models. Applying an optimization model in ISP (Internet Service Provider) networks implies abundant routing reconfigurations, which adversely impact the network stability and QoS parameters. Namely, since routing updates usually cannot be applied on all switches at the same time, transient congestion and loops often happen as a result. Here, in this paper, we propose a new control framework that strives to maximize the network throughput and provide QoS with minimal reconfiguration costs. In contrast to the conventional TE approaches, which perform the network optimizations periodically and control the side effects of reconfigurations by carefully choosing the period length between the optimization cycles, we propose a multi-objective optimization model which jointly minimizes the routing cost function and the reconfiguration overhead. The Pareto frontier of the optimization model is generated by the augmented ϵ-constrained method, whereas a Lyapunov drift-plus-penalty function is used to select the best compromise solution from the Pareto set. Since the reconfiguration overhead is reduced, the network controller could be allowed to optimize resource allocation more frequently, in order to respond quickly and efficiently to the network changes. Considering that the proposed optimization model is computationally intractable in large-scale networks, we also propose a heuristic algorithm to efficiently solve large instances of the problem. In simulations and Mininet experiments, we show that our solution brings performance improvement over the conventional periodic TE techniques. Moreover, the proposed heuristic achieves better QoS request acceptance ratio than the state-of-the-art multi-objective optimization solution, despite the significantly reduced computational complexity.**

***Index Terms*—Routing, SDN, traffic Engineering, QoS.**

I. Introduction

With the expansion of Internet of Things (IoT) and the increasing confluence of augmented reality and artificial intelligence, providing the Quality of Service (QoS) guarantees becomes one of the crucial challenges for Internet Service Providers (ISPs). In today's practice, ISPs avoid QoS issues by over-provisioning backbone links 2 to 3 times relative to the offered load [1], [2]. Since this approach cannot be economically justified in the long run, various Traffic Engineering (TE) mechanisms have been proposed to realize enhanced network performance through an efficient usage of resources [3]. The goal of TE is to optimally map traffic demand into the network. This is hard to achieve with traditional IP/MPLS routers which make greedy routing decisions that fail to provide an overall network optimality [1]. The emergence of Software Defined Networking (SDN) redefined the network architecture by separating the control plane from the data plane. This created opportunity to implement sophisticated TE techniques at powerful centralized control platform (SDN controller) that is able to track the network state and reconfigure data-plane accordingly. This has been already showcased in Data-Centre (DC) and inter-DC networks [1], [2], [4]–[9].

Despite numerous advantages of the SDN concept, there are a lot of performance and scalability concerns when it comes to large-scale network implementations. In order to ensure an optimal network operation, SDN controller must respond to traffic and network changes in the real time. This implies solving complex traffic optimization problems within tight time constraints and reconfiguring flow-tables of SDN switches based on the optimization result. However, frequent network reconfigurations adversely impact performance since the controller has to remove old routing rules and to set up new ones, risking transient instability and congestion [10], [11]. Thus, there is a key problem to balance between the routing optimality and the reconfiguration rate. A common way of handling the above problem is to perform TE periodically (offline) and to use a greedy QoS routing algorithm for online path computation [12]. Increasing the period of TE phases means better reliability and stability, but the network maintains a sub-optimal configuration longer, which might be very costly when the network conditions change unpredictably. New approaches have been proposed in [10] and [11]. Paris *et al.* [10] proposed the control policy that decides whether to apply the solution of iterative optimization solver or not. The control policy is designed to minimize the total flow-allocation cost while respecting the network reconfiguration "budget". An important limitation of this approach is the assumption that network events (flow arrivals and departures) occur according to the stochastic process of known characteristics. In [11], an exact model that optimizes traffic distribution while minimizing

Manuscript received October 11, 2018; revised March 10, 2019; accepted March 13, 2019. *(Corresponding author: Slavica Tomovic.)*

The authors are with the Faculty of Electrical Engineering, University of Montenegro, 81000 Podgorica, Montenegro (e-mail: slavicat@ucg.ac.me; igorr@ucg.ac.me).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/JSAC.2019.2906762

the number of flow-table updates is proposed. However, the applicability of this solution is questionable, since this complex optimization problem is expected to be solved in response to each demand arrival.

Motivated by the problems in changing routing rules, we propose a new technique for scalable QoS provisioning and TE in SDN-based ISP networks. In the proposed approach, the controller periodically reconsiders the current load-balancing pattern and solves a bi-objective optimization problem to improve it. The optimization model involves two conflicting goals: i) maximizing the throughput by minimizing the routing cost function, and ii) minimizing flow-table updates. Our solution allows ISP operator to define the limitation in terms of the time-average reconfiguration rate, which is used to guide the search for the most efficient trade-off between routing optimality and the reconfiguration overhead at each optimization instant. More in detail, the main contributions of this paper are as follows:

1) The proposed Routing Optimality - Reconfiguration Overhead (RO-RO) balancing scheme, maximizes the network throughput subject to the pre-defined QoS and time-average reconfiguration rate constraints. For this purpose, a new bi-objective optimization model has been developed which dynamically adjusts the weights of the two optimization functions involved. In this way, the routing pattern is prevented to deviate significantly from the optimal one. Thereby, our solution does not require any knowledge of flow arrival statistics, in contrast to [13]. We obtain the set of Pareto efficient solutions for the considered problem by using the augmented ϵ-constrained method [14], and choose the most efficient solution from the Pareto set with the drift-plus-penalty algorithm [15] that greedily balances the penalty in terms of the routing optimality gap and the stability (drift) of a virtual queue that captures the price of reconfiguration overhead as it evolves over time [13].
2) Since the proposed multi-objective optimization model is NP-hard, we have developed a new heuristic algorithm to efficiently solve large instances of the TE problem.
3) The performance of the proposed technique was thoroughly examined via simulations. It was found that our solution, for the given reconfiguration budget, reduces QoS rejection ratio when compared to conventional periodic TE schemes. Moreover, it is able to meet very stringent ISP's requirements in terms of the time-average reconfiguration rate without increasing the rejected bandwidth demand significantly. In addition, we found that the proposed heuristic, despite the significantly lower complexity, outperforms the state-of-the art bi-objective optimization technique [11] under the wide range of the analyzed network load scenarios.
4) We implemented the proposed heuristic as an application on top of Floodlight OpenFlow controller [16], and evaluated its performance in the emulated Mininet network.

The rest of the paper is organized as follows. Section II provides a background and explains the motivation for our work. Section III presents the system model and the proposed SDN controller design. The results of simulation analysis are provided in Section IV. In Section V, we explain the prototype implementation and present experimental results. The conclusion remarks are given in Section VI.

II. Background and Motivation

In order to use network capacity efficiently, ISPs mostly rely on MPLS technology, which offers TE capability. The main goal of TE is an even traffic load spreading across the network. MPLS ingress routers typically use Equal Cost MultiPath (ECMP) routing to split traffic demands equally across multiple tunnels towards the same egress router. If ISP needs to provide QoS guarantees, Constrained Shortest Path First (CSPF) routing algorithm is used, which computes the shortest path subject to QoS constraints [3].

Although MPLS has many attractive features, today's ISP networks still suffer from poor efficiency. This could be attributed to the fact that each ingress router installs LSPs independently, without coordination with other routers. The overall result is a suboptimal traffic distribution [1]. Also, the practice has shown that MPLS networks slowly adapt to network changes and cannot support real-time network reconfigurations [17]. To overcome the limitations of classic MPLS TE, the IETF proposed the Path Computation Element (PCE) architecture, where a dedicated element is in charge of a centralized path computation [18]. However, PCE-based architecture also suffers from slow path establishment issues, since it relies on distributed Resource Reservation Protocol (RSVP-TE) to set up, maintain and tear down traffic tunnels. Moreover, from the TE perspective, there is an issue with per-flow traffic splitting granularity in multipath routing, which is determined by the forwarding element capabilities and might deviate significantly from that assumed by the TE solution [3]. This motivated the emergence of SDN architecture with out-of-band programming capabilities and higher granularity of traffic control [3].

The SDN-based TE solutions mostly gained attention in data-center (DC) networks [4]–[6], [19]. Also, several inter-DC SDN network designs, such as Google B4 [2] and Microsoft SWAN [1], [9], have been proposed to improve link utilization and fault tolerance. In order to improve the network QoS, SWAN centrally controls when and how much traffic each DC service can send. B4 has a similar architecture, but additionally enables integration of existing routing protocols in SDN networks. In contrast to [1], [2], which provide soft delay guarantees for interactive flows, MCTEQ TE algorithm [7] includes explicitly both propagation and queuing delay in the TE optimization problem definition. In [9], TE and failure recovery are jointly considered in order to ensure better performance in networks prone to link and switch failures. This TE approach guarantees congestion-free failure recovery if some of the a priori defined fault models are detected.

TE algorithms designed for DC and inter-DC networks cannot be directly applied to ISP networks. Namely, in ISP networks, incoming traffic cannot be scheduled and controlled due to Service Level Agreements (SLAs). Therefore, in [12], [20]–[22], intra-domain TE algorithms for Software-Defined (SD) ISP networks have been proposed. FlowAR algorithm [21] enables a fast and efficient QoS constrained routing by adopting a dynamic flow-migration approach. A trade-off between scalability of SDN controller and the optimality of routing decisions was firstly investigated in [20]. However, use-cases with traffic differentiation were not analyzed. In our previous work [12], [22], we considered the problem of QoS-guaranteed virtual-link provisioning. In particular, we proposed a SDN controller design with the online routing and the offline TE components. The focus was on the optimization of network throughput and failure recovery, but the impact of the reconfiguration overhead was neglected. Therefore, in this paper, we highlight the destructive effects of reconfigurations on a network stability, and propose the SDN-based TE approach that strives to meet the QoS and TE goals with a minimal reconfiguration cost.

The most streamlined way to reduce the reconfiguration overhead during the system operation time is to perform TE periodically with a sufficiently long period between TE cycles [12]. More advanced solutions have been proposed in [10], [11]. In [10], a specially designed SDN control policy is used to filter decisions of iterative routing solver. The solution of the solver is applied only if it could bring a throughput improvement larger than a threshold value. The threshold is defined as a function of the current network configuration optimality and the number of applied reconfigurations. However, this approach is a probabilistic one, i.e. flow arrivals and other network events are assumed to be stochastic processes of known characteristics. The approach from [11] uses a multi-objective optimization model to jointly minimize MLU and the number of flow-table updates. The optimization objectives are always weighted equally in [11], which limits the TE efficiency. Our solution is also based on multi-objective optimization, but it dynamically changes the weights of the objective functions in order to satisfy the time-average reconfiguration rate that is pre-defined by the ISP.

Similarly to our work, Guck *et al.* [23] propose a function split between QoS-constrained routing and TE in industrial networks. A reactive routing algorithm was used to quickly find routes for delay sensitive traffic, while link parameters are configured offline. In contrast to [23], we perform offline optimizations of the load balancing in ISP networks, considering bandwidth and delay as QoS constraints.

The incremental deployment of SDN in an ISP network has been discussed in [24]. In order to reduce the costs of network capacity upgrades over time, multiple strategies for migrating IP network to SDN have been proposed. TE and network performance issues in the hybrid (IP/SDN) networks were discussed in [25]–[28]. Our work could be classified in the category of TE solutions for the hybrid networks because SDN capability is required just at the network edge. However, tunnelling is required in the network core (e.g. MPLS) so that traffic could be routed more flexibly and efficiently. Also, our work addresses TE issue with the focus on QoS provisioning and reconfiguration rate control problems, not covered in [24]–[28]. SDN control framework for joint optimization of intra-domain and inter-domain routing has been proposed in [29]. This framework has a hierarchical structure that can support TE for large-scale ISP networks with thousands of switches. That work is partially complementary to ours, since we propose intra-domain TE solution that (optionally) could rely on the framework from [29] in order to provide better scalability. It should be noted that [29] addresses many requirements that the previous SDN-based TE designs failed to simultaneously satisfy for ISP networks. However, unlike our work, authors of [29] did not analyze the problems related to a high reconfiguration overhead.

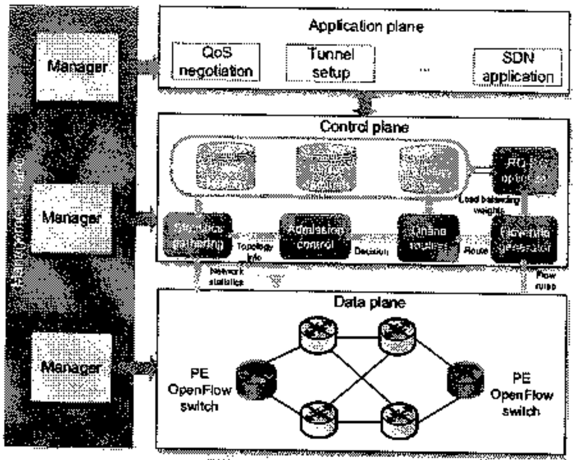


Fig. 1. The system architecture.

III. The System Model

We consider the system architecture from Fig. 1. Following the terminology from MPLS/IP networks, the data plane consists of the Core (CR) and Provider Edge (PE) devices. PE devices are OpenFlow-enabled [30], while SDN capability is not necessarily required at CR devices. Each IE pair of PE devices uses a set of up to K tunnels for traffic delivery. In order to minimize the capacity loss in case of a link failure, tunnels of each IE pair are selected to be "q-disjoint", i.e. at most q of them can share the same link.

We assume that ISP offers flexible BoD service to users. This means that users can use self-provisioning tools to dynamically scale the connectivity requirements up and down, according to their needs. This service becomes increasingly important in the era of ad-hoc inter-enterprise collaborations, cloud computing and big data [31], where high-bandwidth data transfers happen only occasionally, and thus it is beneficial for both users and ISPs to avoid traditional long-term contracts. SDN-based ISP network is able to quickly provide BoD service by offering adequate northbound interface towards users [32]. Through this interface, users can request (or adjust) connectivity service for urgent and temporary needs, subject to QoS constraints, such as bandwidth, delay, reliability, etc. In the further analysis, we assume that the ISP's BoD offering is broadly classified into the following two classes:

- QoS1 BoD - a connectivity service with bandwidth guarantees.
- QoS2 BoD - a connectivity with the guaranteed bandwidth and delay upper bound, intended for time-critical applications.

Thereby, multiple QoS2 BoD subclasses with different delay bounds could be defined to make service differentiation more granular. ISP can additionally allow users to choose between immediate and time-scheduled service provisioning with potentially different pricing, in order to optimize resource utilization [31]. SDN-based BoD model promises substantial economical savings for enterprises, since they can lease network resources based on real-time priorities, without the burden of operating an expensive private line. For example, an enterprise can issue QoS1 service for a data backup application, which requires a high bit-rate but is not affected by the network delay. The request may relate to only a few hours at night, when backup transfers are performed. On the other side, QoS2 BoD service can help enterprises to efficiently handle video conferencing and other delay-sensitive events. From the ISP's perspective, SDN-based BoD creates new revenue opportunities, since they can rent unused network resources on a time-shared basis.

The proposed SDN controller design has two modules responsible for traffic control: *Online-Routing* and *RO-RO optimizer*. The *Online-Routing* enforces admission control policy and determines the initial routes for the BoD requests based on the current bandwidth availability in the network. The *RO-RO optimizer* module periodically re-considers the flow allocation and optimizes the load balancing in the network subject to QoS constraints. During this process, the controller makes a trade-off between the routing optimality and the reconfiguration overhead. Ingress switches classify incoming traffic in two classes that correspond to two different BoD service types. Each traffic class is tagged in a unique way. The traffic splitting across the tunnels is flow-based, performed by applying a hash function on packets' header fields. Unequal traffic splitting ratios could be implemented by using the SELECT group-tables [30] in the processing pipeline of OpenFlow switches. In ingress flow-tables, packets are mapped to one of the total two group-tables based on the destination address and the label attached. Each group-table corresponds to a different BoD service class (QoS1 or QoS2), and it is configured with a set of up to K tunnels and K weights, which determine the ratio of the matching traffic which is to be sent to each of the tunnels. In the rest of this Section, we explain the controller's *Online-Routing* and *RO-RO optimizer* modules in more detail.

A. Online-Routing

This module performs the admission control for user's BoD request by running a computationally simple QoS routing algorithm. The work-flow of the algorithm is shown in Fig. 2. In the first step, the algorithm determines the type of BoD request, in order to filter out the subset of tunnels that can meet a user's delay requirement. Then, the algorithm checks whether a bandwidth requirement can be met when splitting demand over multiple paths, using load balancing weights determined by the *RO-RO optimizer* module in the last running cycle. If so, the request is accepted and no new flow-table rules are installed in the network. The ingress switches will direct the flow to an appropriate group-table, and distribute its sub-flows over multiple pre-configured tunnels. If bandwidth requirement cannot be met with the current load-balancing weights, the algorithm seeks for an eligible tunnel that can

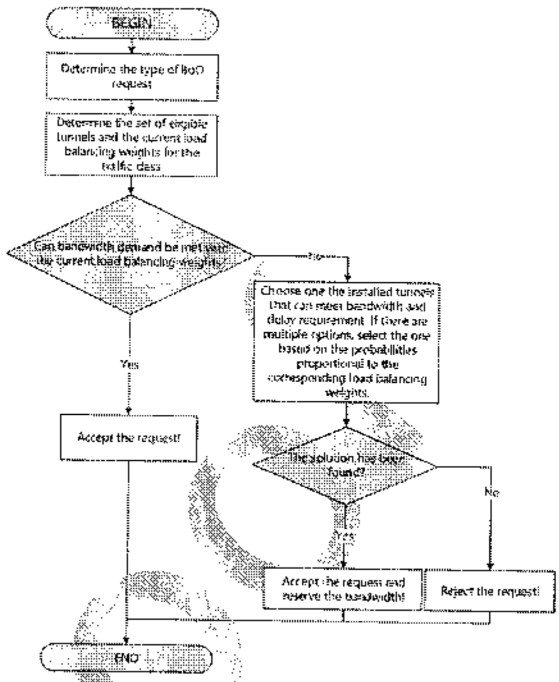


Fig. 2. The online routing process.

carry a total user's demand without traffic splitting. If multiple tunnels are eligible, the preference is given to the one with the largest load-balancing weight according to the *RO-RO optimizer*. In this case, a new routing rules must be added in the network. It is important to note that for online routing the goal is not to optimize the routing decision, but rather to find a quick solution to allocate resources for the user.

B. RO-RO Optimizer

The RO-RO optimizer periodically analyzes the network state (every T seconds) and identifies the optimal set of actions to improve it. The optimality of TE decision is affected by two criteria: i) the resulting MLU, and ii) the number of reconfigurations required. Clearly, those two criteria are conflicting, and there is no unique solution that simultaneously optimizes both of them. Instead, multiple-objective optimization problems usually have a set of so-called Pareto optimal solutions, that lie on the optimality curve [33]. A solution satisfies Pareto optimality criterion if it is not possible to improve any of the objectives without deteriorating the others. Therefore, the RO-RO optimizer module operates in two phases. In the first phase, a set of Pareto optimal solutions (known as Pareto frontier) is generated. In the second phase, the appropriate Pareto optimal solution is found.

1) The Pareto Frontier Generation: We use the AUGMECON method [14] to compute Pareto optimal solutions for our bi-objective optimization problem. Table I explains the parameters and decision variables involved in the optimization model. Firstly, we find optimal MLU solution - MLU^{opt}. Then, in several iterations, we optimize the number of group-table updates α by using the constraint: $MLU \leq MLU^{*}$, where MLU^{*} is varied in the range from $MLU(t)$ to MLU^{opt}. Each of these steps is explained below.

TABLE I
THE NOTATION SUMMARY

Param	Description
$P_{ij}^{(C)}$	Paths between nodes i and j that can meet delay requirement of traffic class C
$d_{ij}^{(C)}$	Aggregated traffic demand of class C between IE pair (i,j)
$x_{d_{ij}^{(C)},p}$	The splitting ratio of $d_{ij}^{(C)}$ over path p
c_l	Capacity of link l
pd_l	Propagation delay of link l
$R_{p,l}$	1 if path p contains link l otherwise 0
MLU^{opt}	Optimal MLU
$MLU(t)$	MLU at time t
MLU^*	MLU sample in range between $MLU(t)$ and MLU^{opt}
α	Number of group-table updates
α_n^*	The solution of the optimization model (2a) in n-th iteration
$\alpha(t)^*$	The number of reconfiguration performed by RO-RO optimizer at time t
$\beta_{ij,p}^{(C)}$	1 if $x_{d_{ij}^{(C)},p}$ has changed, 0 otherwise
$\gamma_{ij,p}^{(C)}$	Non-binary indicator of $x_{d_{ij}^{(C)},p}$ change
$g_{ij}^{(C)}$	1 if group-table for IE pair (i,j) and class C has changed, 0 otherwise
r	Range of MLU variable: $MLU(t)$ - MLU^{opt}
s	Surplus variable
PF	Pareto frontier
R_{avg}	Time average constraint on reconfiguration rate
R_{max}	Upper-bound on reconfiguration rate
$V(t)$	Virtual queue size at time t
Γ	MLU optimality gap
T	Time interval between consecutive runs of RO-RO optimizer

a) MLU Minimization (step 1): The network is modelled as a directed graph $G = (V, L)$, where V is a set of nodes and L is a set of edges. Each link is described by capacity c_l and propagation delay pd_l. We assume that total path delay is predominantly determined by the propagation delay. This assumption can be justified in the ISP scenario, because low queuing delays are expected due to high-speed links and the effect of statistical multiplexing of the aggregated traffic flows [34]. A traffic matrix TM is modelled as $|N|x|N|$ matrix, where N is the set of edge OpenFlow switches. Matrix element d_{ij} refers to aggregated traffic demand between ingress switch i and egress switch j. In the optimization model, matrix TM is decomposed into two matrices TM_{qos1} and TM_{qos2}, one for each traffic class. A traffic-class demand $d_{ij}^{(C)}$, $C \in \{qos1, qos2\}$ splits across the set of tunnels $P_{ij}^{(C)}$, which is computed a priori to satisfy delay requirement of class C. For each traffic demand $d_{ij}^{(C)}$, we compute the optimal splitting ratios $x_{d_{ij}^{(C)},p}$ for tunnels $p \in P_{ij}^{(C)}$, i.e. portions of the traffic demand that would be routed over each tunnel from the set $P_{ij}^{(C)}$. More specifically, an optimal MLU value is found by solving the following optimization model:

$$\min MLU \quad (1a)$$

$$\text{s.t.} \sum_{p \in P_{ij}^{(C)}} x_{d_{ij}^{(C)},p} = 1 \quad \forall C, \forall (i,j) \in N \quad (1b)$$

$$\sum_{\forall C} \sum_{(i,j) \in N} \sum_{p \in P_{ij}^{(C)}} R_{p,l} \cdot x_{d_{ij}^{(C)},p} \cdot d_{ij}^{(C)} \leq MLU \cdot c_l \quad (1c)$$

$$0 \leq x_{d_{ij}^{(C)},p} \leq 1 \quad \forall C, \forall (i,j) \in N, \forall p \in P_{ij}^{(C)} \quad (1d)$$

$$0 \leq MLU \leq 1 \quad (1e)$$

The constraint (1b) ensures that all traffic demands are routed. The link capacity constraints are given by (1c). $R_{p,l}$ is a binary variable which is 1 when a path p includes a link l, and 0 otherwise. Since the controller should provide the bandwidth guarantees, a total amount of reserved bandwidth must not exceed a link capacity, as stated by (1e).

b) Iterative Search of Pareto-Optimal Solutions (step 2): The result of the first step is MLU^{opt}. We define the range of MLU values r, with MLU^{opt} as a lower bound and $MLU(t)$ as an upper bound. Then, in order to generate Pareto frontier, we divide this range by using at most 10 equidistant points MLU_n^* $(0 \leq n \leq 10)$ placed between MLU^{opt} and $MLU(t)$. In addition, we ensure that the distance between points is not smaller than 0.02. We noticed that increasing the number of points above 10, or using smaller distance between the points, does not lead to noticeable performance improvement. Each point MLU_n^* is used as a constraint in the following single-objective optimization model:

$$\min \ \alpha - \frac{\epsilon \cdot s}{r} \quad (2a)$$

s.t.

$$(1b), (1d), and: \alpha = \sum_{(i,j) \in N} \sum_{\forall C} g_{ij}^{(C)} \quad (2b)$$

$$\beta_{ij,p}^{(C)} = \begin{cases} 0 & , if x(t)_{d_{ij}^{(C)},p} == x(t-T)_{d_{ij}^{(C)},p} \\ 1 & , else \end{cases} \quad (2c)$$

$$\forall (i,j) \in N, \forall C, \forall p \in P_{ij}^{(C)}$$

$$g_{ij}^{(C)} \geq \beta_{ij,p}^{(C)} \quad \forall (i,j) \in N, \forall C, \forall p \in P_{ij}^{(C)} \quad (2d)$$

$$z + s = MLU_n^* \quad (2e)$$

$$\sum_{\forall C} \sum_{(i,j) \in N} \sum_{p \in P_{ij}^{(C)}} R_{p,l} \cdot x(t)_{d_{ij}^{(C)},p} \cdot d_{ij}^{(C)} \leq z \cdot c_l \quad \forall l \in L \quad (2f)$$

$$g_{ij}^{(C)} \in \{0, 1\} \quad \forall (i,j) \in N, \forall C \quad (2g)$$

$$\beta_{ij,p}^{(C)} \in \{0, 1\} \quad \forall (i,j) \in N, \forall C, \forall p \in P_{ij}^{(C)} \quad (2h)$$

$$s \geq 0 \quad (2i)$$

$$0 \leq z \leq 1 \quad (2j)$$

The main idea of the above optimization model is to minimize the number of reconfigurations α, subject to MLU and QoS constraints. The MLU constraint is transformed into equality (2e) by introducing a surplus variable s, that affects the related constraint (2f). This helps in avoiding weakly efficient Pareto solutions [14]. The surplus variable is also used in the optimization function (2a), but as a low-priority (second) term since ϵ is a very small constant (typically between 10^{-6} and 10^{-3}). Basically, with (2b) – (2d) we minimize the number of group-table updates. In the considered scenario, each OpenFlow switch i maintains two group-tables per egress node j - one for each traffic class C. With $g_{ij}^{(C)}$ we track whether a group-table for IE pair (i, j) and QoS class

C must be updated ($g_{ij}^{(C)} = 1$) or not ($g_{ij}^{(C)} = 0$). In order to make distinction between new and the current load balancing weights, we show $x_{d_{ij}^{(C)},p}$ as a function of time t. In (2c), the binary decisions variables $\beta_{ij,p}^{(C)}$ are equal to 1 when the newly computed load-balancing weight $x(t)_{d_{ij}^{(C)},p}$ is different from the old one $x(t-T)_{d_{ij}^{(C)},p}$, or equal to 0 otherwise. The constraint (2d) indicates that a group-table is updated if and only if at least one of its load balancing weights has changed with regard to its previous value. In (2c), we need to check if $\left|x(t)_{d_{ij}^{(C)},p} - x(t-T)_{d_{ij}^{(C)},p}\right| > 0$ holds, whereas the modulus operator is not acceptable for Linear Programming (LP) solvers. In order to eliminate the need for the modulus operator, we introduce the additional variables $\gamma_{ij,p}^{(C)}$:

$\forall (i,j) \in N, \forall C, \forall p \in P_{ij}^{(C)}$:

$$\gamma_{ij}^{(C)} \geq x(t)_{d_{ij}^{(C)},p} - x(t-T)_{d_{ij}^{(C)},p} \quad (2k)$$

$$\gamma_{ij}^{(C)} \geq x(t-T)_{d_{ij}^{(C)},p} - x(t)_{d_{ij}^{(C)},p} \quad (2l)$$

Since $\beta_{ij,p}^{(C)}$ must be binary in order to reflect the changes of load-balancing weights, we add one more constraint:

$$\eta \cdot \gamma_{ij}^{(C)} \leq \beta_{ij,p}^{(C)} \leq \eta \cdot \gamma_{ij}^{(C)} + 1 \quad (2m)$$

where η is a small value (e.g. 10^{-5}). Therefore, the final problem definition becomes: minimize (2a), subject to (2b), (2d)-(2m), and the QoS constraints (1b) and (1d). We denote the solution of the optimization model (2a) in n-th iteration with α_n^*. Thus, the resulting Pareto frontier (PF) after N iterations becomes:

$$PF = (MLU_1^*, \alpha_1^*), (MLU_2^*, \alpha_2^*)...(MLU_N^*, \alpha_N^*) \quad (3)$$

2) *Decision Making:* At each running cycle, the *RO-RO optimizer* selects one of the Pareto optimal solutions from (3). In general, a solution (MLU_n^*, α_n^*) requires α_n^* updates of group-tables and introduces the MLU optimality gap:

$$\Gamma = MLU_n^* - MLU^{opt} \quad (4)$$

The common Pareto ranking methods used for decision making in multi-objective optimization are: Weighted Sum Method (WSM), Weighted Product Method (WPM) and the Weighted Average Ranking (WAR) [35]. In all of those methods, weights are used to define the trade-offs between the objectives. In [11], the WSM method is applied to find the best compromise solution based on two criteria: MLU and the number of flow-table updates. However, in contrast to [11] we are interested in minimizing an average optimality gap Γ while keeping the time-average reconfiguration rate below R_{avg} - which is an input parameter provided by the ISP. Note that this requires adjustments of the optimization functions' priorities (i.e. weights), thus WSM, WPM and WAR ranking methods are unsuitable for our needs. In order to achieve the time-average guarantees on the reconfiguration rate, we model the reconfiguration rate constraint as a virtual queue, with dynamics over time slots $t \in \{0, T, 2T...\}$ as follows:

$$V(t+T) = \lceil V(t) - R_{avg} \rceil^+ + \alpha(t)^*/T \quad (5)$$

In the above equation, $V(t)$ is a queue size at a given time t, while $V(t+T)$ is expected queue size at the moment $t+T$ when the RO-RO optimizer will run again. The service rate of the queue is the desired time-average reconfiguration rate R_{avg}. The number of queue arrivals during the time interval $[t, t+T]$ corresponds to the number of reconfigurations $\alpha(t)^*$ that RO-RO optimizer decides to perform at time t. The initial queue size $V(0)$ is set to 0. It is easy to see that (5) implies:

$$\frac{\alpha(t)^*}{T} - R_{avg} \leq V(t+T) - V(t) \quad (6)$$

After applying the low of telescoping sums, we obtain:

$$\sum_{\tau=0}^{t-1} \left(\frac{\alpha(t)^*}{T} - R_{avg}\right) \leq V(t) - V(0) = V(t) \quad (7)$$

Dividing by t and applying expectation yields:

$$\frac{1}{t}\sum_{\tau=0}^{t-1} E\left[\left(\frac{\alpha(t)^*}{T} - R_{avg}\right)\right] \leq \frac{E[V(t)]}{t} \quad (8)$$

Therefore, the time average constraint on the reconfiguration rate is satisfied when virtual queue is "mean rate stable" [15], i.e. whenever the following statement holds:

$$\lim_{t \to \infty} \frac{E[V(t)]}{t} = 0 \quad (9)$$

It has been proven that such virtual queue could be stabilized by minimizing the Lyapunov drift function [15], which captures the changes in the queue backlog:

$$L(t) = V(t+T)^2 - V(t)^2 \quad (10)$$

However, while taking actions to minimize $L(t)$ will help to meet the average time constraint for $\alpha(t)$, the resulting average MLU optimality gap might be unacceptably large. Thus, instead of choosing the Pareto optimal solution that minimizes $L(t)$, the *RO-RO optimizer* chooses the one which minimizes the following drift-plus-penalty function:

$$A \cdot \Gamma(t) + L(t) \quad (11)$$

where $A \geq 0$ is a constant which emphasizes the importance of the first objective function. By increasing A we can push routing optimality gap towards 0. at the expense of increasing an average queue size linearly proportional to A.

C. RO-RO Heuristic

Since generating Pareto optimal solutions for the RO-RO optimization model is NP-complete problem (due to integer decision variables), here we propose a heuristic for computing approximation of Pareto frontier. The pseudo-code of the heuristic is provided in Algorithm 1. Firstly, we solve the LP optimization model (1) that minimizes the MLU for the current traffic matrix. If delay-matching tunnels for both traffic classes are known in advance, this model can be solved in polynomial time. After that, we determine the number of reconfigurations α required by the optimal MLU solution (line 2). Since only a single objective optimization was performed, the MLU^{opt} most likely could be accomplished with a number of reconfigurations smaller than α. Therefore, in the

Algorithm 1 RO-RO Heuristic

Input: *networkInfo* = network statistic, *demands* = aggregated demands between IE pairs, N_{max}=maximum number of iterations, R_{max} = upper bound on the reconfiguration rate;
Output: *PF* = aproximated Pareto frontier;
Comment: *minimizeMLU() solves the optimization model (1), LBweights stands for the load balancing weights;*

1: MLU^{opt}, LBweights = *minimizeMLU*(demands)
2: α = *countReconfigs*(MLU^{opt})
3: PF={ }
4: PF.*add*((MLU^{opt}, α))
5: $\alpha_{step} = \alpha / N_{max}$
6: $A^* = \{\}$
7: **for** i **in range** $(1, N_{max})$ **do**
8:   $A^*.add(i \cdot \alpha_{step})$
9: **end for**
10: $A^*.sort\_ascending()$
11: prevDemands={ }
12: **for** α_n^* **in** A^* **do**
13:   **if** $\alpha_n^* > R_{max} \cdot T$ **then**
14:     **break**
15:   **end if**
16:   LPdemands = prevDemands
17:   cDemands = *getCriticalDemands*(α_{step}, LPdemands)
18:   LPdemands.*add*(cDemands)
19:   MLU^*, LBweights = *minimizeMLU*(LPdemands)
20:   PF.add((MLU_n^*, α_n^*))
21:   *updateNetworkState*(LBweights)
22:   prevDemands = LPdemands
23: **end for**
24: **function** *getCriticalDemands*(α_{step}, LPdemands)
25:   results ={ }
26:   dNum = { }
27:   stats=networkInfo.*copy*()
28:   **while** dNum < α_{step} **do:**
29:     worstLink=*getMostLoadedLink*()
30:     critical_demand=worstLink.*getMaxDemand*(stats)
31:     **if** critical_demand **in** LPdemands **do: continue**
32:     **end if**
33:     results.*add*(critical_demand)
34:     stats.*remove*(critical_demand) {*Neglect load of critical demand*}
35:     dNum+=1
36:   **end while**
37:   **return** results
38: **end function**

next steps, we try to approach the MLU^{opt} by rerouting as few as possible traffic flows. The input parameter of the algorithm is the bound on the number of iterations N_{max} - which limits the maximum number of the LP model executions. By dividing α with N_{max}, we define the "resolution" of the solution space - α_{step}, i.e. the set of possible reconfiguration overheads $A^* = \{\alpha_1^*, \alpha_2^*, ...\alpha_{N_{max}}^*\}$ in the approximated Pareto frontier is obtained as a multiple of α_{step} (lines 7–9). Then, we iterate over A^* arranged in the ascending order, and use its elements as input arguments of *getCriticalDemands()* function (line 19). We assume that ISP specifies the upper-bound on the reconfiguration rate over any time interval T, which we denote with R_{max}. Therefore, the elements of $A*$ that imply the reconfiguration rate higher than R_{max} are excluded from further examination.

For the given input α_n^* $(1 \leq n \leq N_{max})$, *getCriticalDemands()* function determines α_n^* traffic demands that will be reconsidered for rerouting by the LP MLU minimization model (1). The goal is to choose those demands that can contribute the most to reducing the network MLU. Here, "traffic demand" refers to the total offered load of a given IE pair that shares the same QoS policy. Our strategy to approach the mentioned goal is as follows. At the beginning of each iteration, the solution set *LPdemands* inherits the list of demands selected in the previous iterations - *prevDemands*. Therefore, in each call, the function *getCriticalDemands()* actually search for α_{step} "critical demands", while $\alpha_n^* - \alpha_{step}$ critical demands are already known. In order to find a new "critical demand", we initially identify the most loaded link based on the network state information (line 29). Then, the demand which loads this link the most is proclaimed "critical" and added to the *LPdemands* set, provided that it has not been already chosen as "critical" in the previous iterations (line 31–33). After a new "critical" demand is determined, the network state information is updated by adding the bandwidth occupied by the chosen demand, i.e. the existence of this demand in the network is ignored till the end of function call (line 34). When the *LPdemands* set is complete, the LP model is used to minimize the MLU by redistributing the "critical" demands (line 19). The resource allocation for non-critical demands is left unchanged. Based on the solution of the LP model (1), the network state information is updated again so as to reflect the network state in case when the computed load balancing scheme is applied (line 21). The resulting MLU value MLU_n^*, in pair with α_n^*, makes an approximated Pareto optimal solution. Once the approximated Pareto frontier is generated, the most suitable solution is selected with drift-plus-penalty algorithm described in the previous subsection.

D. Discussion on the Optimization Function

The above-described RO-RO optimization function is affected only by the number of reconfigurations and by the load of the most burdened link. Although minimizing the MLU is a natural and intuitive objective, which is commonly adopted in ISP networks [20], [29], this metric is sensitive to individual bottleneck links that may be hard to avoid under any routing solution. As a result, for example, the scenario in which all links are 90% utilized has the same optimal solution value as a scenario in which only one link has utilization of 90%, while all the other links are underutilized, if we assume that both solutions require the same number of flow-table updates. In practice, the second scenario is more preferable. For this reason, we propose an alternative formulation of the objective function for the RO-RO optimizer, which strives to minimize MLU, routing cost and the number of reconfigurations. In the rest of the paper, we refer to this solution as RORO-MC

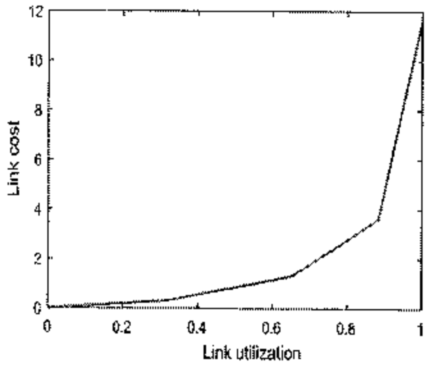


Fig. 3. A link cost in function of the link utilization.

(MC stands for Minimizing routing Cost). With regard to the previous problem definition, RORO-MC requires modifying optimization function (1a) as:

$$\min \ MLU + w \cdot \sum_{i \in L} \text{cost}(l_a) \tag{12a}$$

In the above formulation, link cost is a convex function of the link utilization l_a, computed as ratio of the offered load x and the link capacity c_l, while w is a constant used to control the impact of the total routing cost on the solution value. Experimentally, we found out that setting w to 10^{-4} provides a very satisfying trade-off between the MLU and the total routing cost. We have adopted a piece-wise linear link cost function from Fig. 3, which was initially proposed in [36] and extensively used in literature. Generally, it is "cheap" to send traffic over a link with a small utilization. The cost increases progressively as the utilization approaches 100%. More precisely, for a given value of c_l, derivative of the cost function is defined as follows:

$$cost'(l_a) = \begin{cases} 1, & if\ 0 \le l_a \le 1/3, \\ 3, & if\ 1/3 \le l_a \le 2/3, \\ 10, & if\ 2/3 \le l_a \le 9/10. \\ 70, & if\ 9/10 \le l_a \le 1. \end{cases} \tag{13}$$

The processing steps of the RORO-MC optimizer are summarized in Fig. 4.

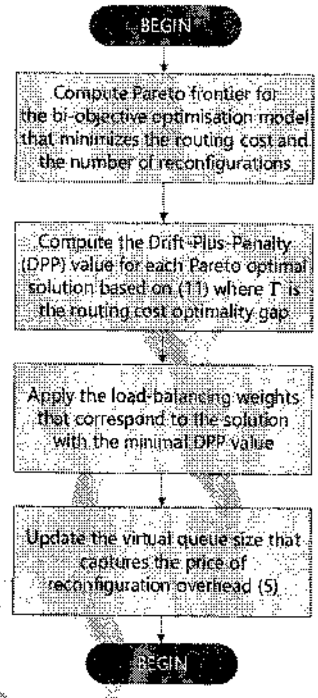


Fig. 4. The workflow of RORO-MC optimizer.

IV. Simulation Results

We have developed a flow-level simulator in Python to evaluate performance of the proposed TE solution. The CPLEX [37] library has been used to solve optimization problems. In simulations, we used POP-level Sprint ISP topology [38]. The link capacities were set to 1000 units, while delay on each link was derived from a great circle distance between the connected POPs and the speed of light in the fiber. Since for most ISP networks real traffic traces are not available, we used a well known gravity model [39] to generate an average traffic matrix for the hour of the maximum load during a day. By scaling the gravity model matrix, we derived "admissible" traffic matrix, which brings MLU to at most 70%. Based on an "admissible" traffic matrix, the dynamic evolution of the BoD request arrivals was simulated. We used exponential distribution with the mean $t = 20min$ to model the durations of traffic demands and Poisson distribution to model arrivals of BoD requests for each IE pair. The mean rate of the BoD request arrivals during the hour of the maximum load is computed so that the resulting traffic model in average converges to the "admissible" traffic matrix. For simplicity, the bandwidth requirement of each BoD request was chosen randomly from the set [3, 5, 7] bandwidth units. The delay bound for QoS2 requests was set to 35ms, which corresponds to delay on the shortest path between the mutually most distant IE pair on the Sprint topology. The number of tunnels between each IE pair (*K*) was bounded by 10. Those tunnels are selected to be "2-disjoint" (i.e. *q*=2). For the considered network topology we found out that using more than 10 shortest paths between IE pairs results in negligible increase of throughput. Duration of simulations was set to one day, and the request arrival rate during that period was scaled according to the daily traffic pattern from [40]. The simulation setting is summarized in Table II.

We compared the RORO TE approach with the TE approach that minimizes MLU periodically (MLU-PTE) [12]. Thereby, we run simulations for two different values of the period between MLU optimizations: T=5 min and T=1h. The first value of T is chosen to demonstrate pros and cons of adopting highly dynamic SDN-based TE model in ISP networks. In particular, this value corresponds to the period of TE cycles in SWAN architecture [1] which is deployed in real-world inter-DC networks. On the other side, according to [41], TE in ISP networks is usually performed at the timescale of hours. Thus, we consider the later scenario reasonable in terms of the reconfiguration overhead imposed. In addition, we analyzed the performance of the multi-objective TE approach which is implemented based on the ideas from [11], where MLU and the reconfiguration overhead optimization objectives are always weighted equally. We use the notation

TABLE II
THE SIMULATION SETTING

Network topology	Sprint (PoP level)
The number of nodes	17
The number of links	72
Link capacity	1000 units
BoD request arrival process	Poisson for each IE pair (mean = IE average demand/ average request duration)
The bandwidth demand of BoD requests	[3, 5, 7] units
The duration of BoD requests	Exponential distribution (mean = 20 minutes)
The max. number of paths between an IE pair (K)	10
Link disjointness parameter for IE pair (q)	2
QoS1 delay bound	35ms
Simulation duration	1 day (24 hours)

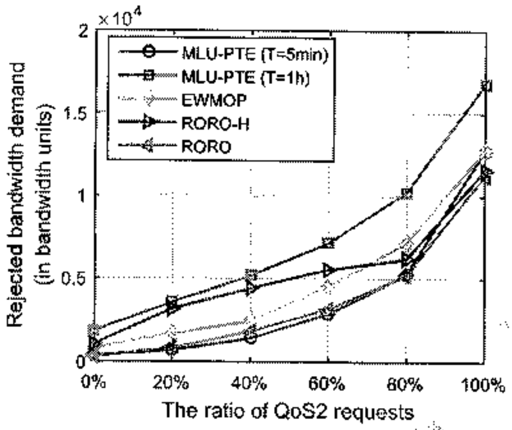


Fig. 5. Rejected bandwidth demand.

EWMOP (Equally Weighted Multi Objective oPtimization) as a representation for it. We divide the simulation analysis in two parts. Firstly, we analyze the performance of RO-RO optimizer explained in Section III (B). After that, we evaluate the benefits of using RORO-MC optimizer discussed in Section III (D). The period between the running cycles for both optimizers was set to 5 minutes by default. We denote the heuristic version of the solutions with the suffix "H". For the heuristics, we limit the number of iterations to 100.

A. Performance Analysis of RO-RO-Based TE

As the goal of this evaluation is to show that the RO-RO optimizer is able to efficiently balance between the routing optimality and the reconfiguration overhead, we show the results in terms of rejected bandwidth demand and the average reconfiguration rate in Fig. 5 and Table III, respectively. We measured the reconfiguration rate as the rate of group-table updates initiated by the RO-RO optimizer, as those updates affect aggregated traffic demands. The results are shown as a function of the percentage of QoS2 requests. In all of the simulation scenarios, R_{max} was set to 150 and R_{avg} was set to 1.6 reconfigurations per minute - which corresponds to 8 reconfigurations in each optimization cycle in average. The setting for R_{avg} was done based on the observations from MLU-PTE (T=1h) simulations. As mentioned before, here we

TABLE III
THE AVERAGE RECONFIGURATION RATE

QoS2 %	MLU-PTE (T=5min)	MLU-PTE (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

consider MLU-PTE (T=1h) approach as a representative of a conventional TE in ISP networks. By running the simulation for different values of R_{avg}, we found out that for $A = 10^7$ the RO-RO TE offers the most efficient trade-off between the average MLU optimality gap and the reconfiguration rate. By increasing the A parameter, we can further reduce the average MLU optimality gap but the reconfiguration overhead could break the R_{avg} constraint in that case. On the other side, we identified that RORO-H achieves the similar trade-off for $A = 5 \cdot 10^5$. Therefore, we analyze performance of RO-RO and RORO-H only for the mentioned values of A.

From Fig. 5 it is interesting to note that RO-RO causes almost the same BoD request rejection ratio as MLU-PTE (T=5min), even though the reconfiguration rate is reduced up to 25 times (Table III). Moreover, when QoS2 traffic is predominant, RO-RO slightly outperforms MLU-PTE. This "unexpected" improvement is a consequence of formulating optimization problem as minimizing MLU. This kind of optimization model suffers from allowing a single bottleneck link to define the whole network picture. Also, it does not penalize using very long detours which cause intensive resource consumption. In these situations, RO-RO performs better because it is focused on optimizing the load balancing for the most demanding IE pairs, on which the largest number of BoD requests refers to. In some cases, this results in a lower time-averaged MLU even though the routing configuration that yields an optimal MLU is not established in each run of the RO-RO optimizer. The limitations of MLU-PTE are especially pronounced in the scenario where all BoD requests are delay-sensitive (100% of QoS2 traffic), because the routing is limited to small number of low-latency paths and the transmission bottlenecks are easily created. Since there are no many valid options for redistributing the offered QoS2 load of IE pairs, the drop of MLU-PTE's reconfiguration rate occurs (Table III).

When compared to EWMOP, RO-RO rejects smaller number of BoD requests. Here, the capacity improvement is a result of the dynamic balancing between the two conflicting optimization goals: MLU and the reconfiguration rate. Namely, during the Pareto ranking process, RO-RO evaluates the importance of MLU objective value based on the current state of the virtual queue that captures deviation of the time-average reconfiguration rate with regard to R_{avg}. In this way, RO-RO efficiently exploits the reconfiguration "budget" given by the ISP to minimize the MLU optimality gap. On the other side, EWMOP is unaware of this reconfiguration "budget", and thus suffer from overvaluing the second optimization function. Since the RO-RO uses a computationally hard optimization

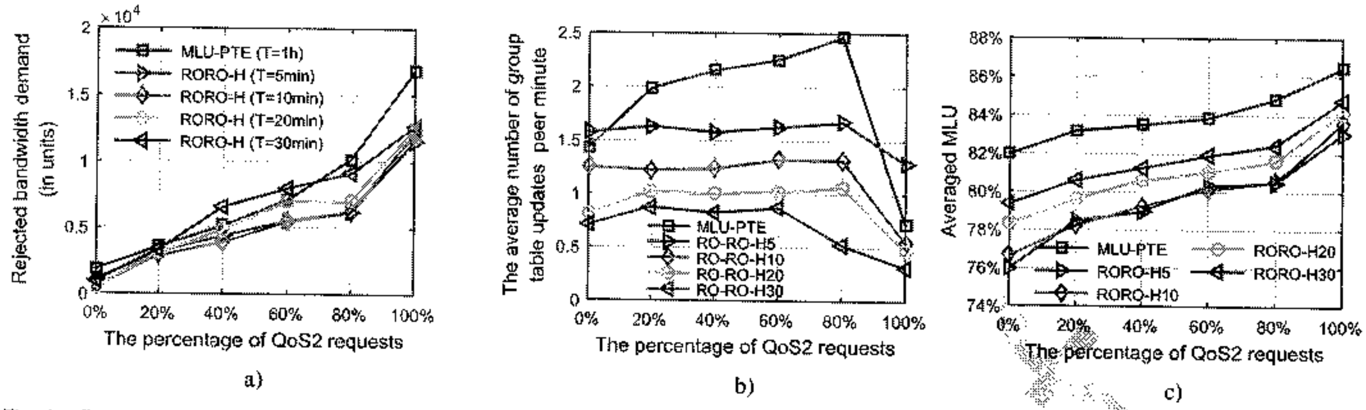


Fig. 6. The impact of T on RORO-H performance. The suffixes in the legend denote the **T** value used in simulations.

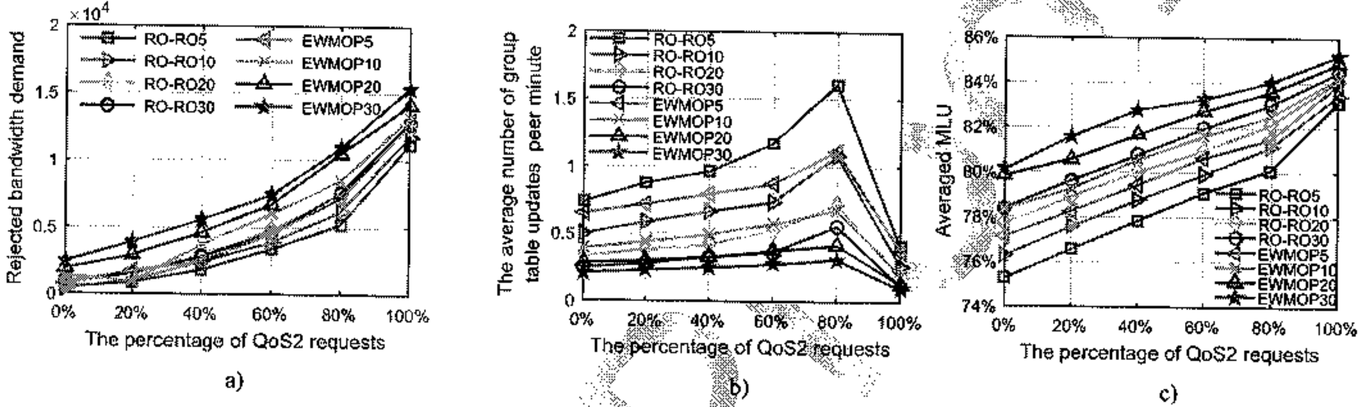


Fig. 7. The impact of T on RORO and EWMOP performance. The suffixes in the legend denote the **T** value used in simulations.

model, the performance of RORO-H is of greater practical importance. As illustrated in Fig. 5, RORO-H clearly outperforms MLU-PTE for the same (or even lower) level of reconfiguration overhead. This is evident from the comparison with MLU-PTE (T=1h), which we have declared here as a conventional TE scheme with a reasonable reconfiguration overhead. It can be seen from Fig. 5 that RORO-H has a lower BoD rejection ratio in all of the simulated traffic-load scenarios. Moreover, it provides a lower rate of group-table updates. The reconfiguration overhead of MLU-PTE could be further reduced by increasing T above 1h, however, this would result in the inability to respond to sudden traffic distortions. On the other hand, RORO-H has means to control the reconfiguration overhead, so the controller can reallocate the network resources more frequently in order to better track the rapid changes of the network state. When the ratio of QoS2 traffic is below 80%, RORO-H rejects a higher amount of bandwidth demand than EWMOP. However, the performance is still satisfying considering that RORO-H complexity is significantly lower. It should be noted that the performance of the EWMOP is strongly impacted by the network conditions. For example, when Pareto optimal solutions in EWMOP optimization model strongly differ in terms of the number of reconfigurations required, the optimizer could underestimate the importance of the MLU optimality gap.

In the rest of the paper, we analyze the performance of the RORO and RORO-H separately. In order to fairly clarify the value of our proposal, we compare RORO-H with MLU-PTE (T=1h), and RO-RO with EWMOP. Namely, it is unfair to compare BoD rejection ratio of RORO-H and EWMOP, since EWMOP is based on NP-hard optimization model. On the other side, MLU-PTE (T=1h) and RORO-H are both based on LP models and impose comparable reconfiguration overheads.

Figure 6 depicts the impact of the period length (T) between the optimization cycles on RORO-H performance. It can be observed that increasing T from 5 to 10 minutes does not impact the average MLU and BoD rejection performance significantly, while the total system reconfiguration overhead is reduced. On the other side, when T is 30 minutes, our heuristic under-performs the MLU-PTE technique (which minimizes MLU hourly) in some of the traffic-load scenarios analyzed. In general, increasing T leads to reduction of reconfiguration overhead, although the same R_{avg} setting is used, and to increase in average MLU optimality gap. Fig. 7 shows the influence of T on RO-RO and EWMOP performance. The obtained results allow to conclude that RO-RO optimization model provides performance improvement regardless of the T value. Fig. 7(b) suggests that EWMOP introduces a lower reconfiguration overhead. However, this cannot be considered as a benefit because RO-RO keeps its time-average reconfiguration rate below the given threshold.

B. Performance Analysis of RORO-MC

As discussed in Section III (D), TE solutions that strive to optimize the network throughput by minimizing MLU suffer from several limitations. Thus, we have investigated the

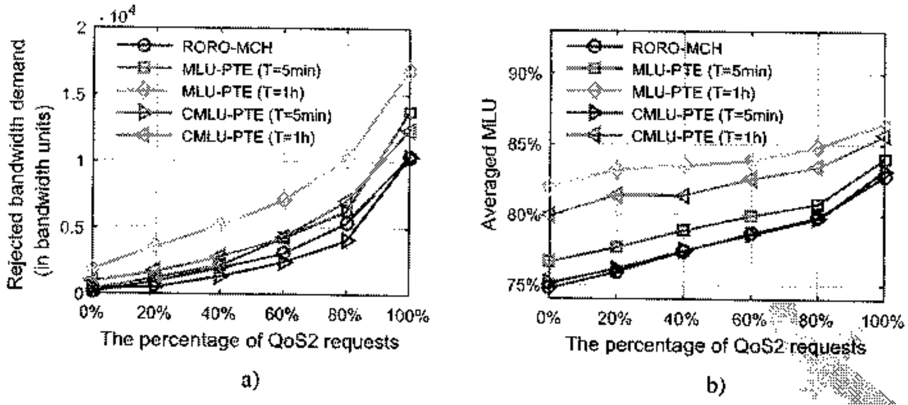


Fig. 8. Rejected bandwidth demand and average MLU in RORO-MCH simulation analysis.

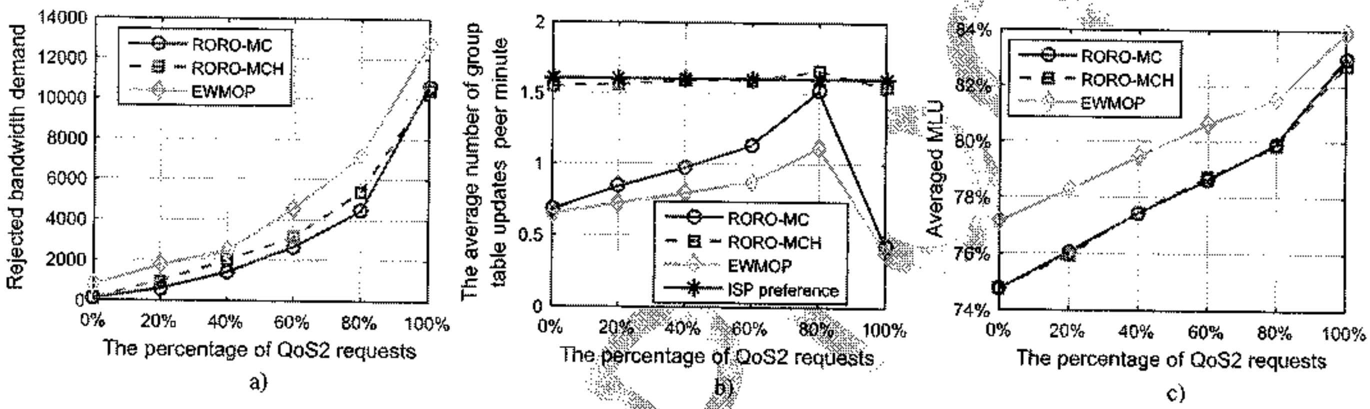


Fig. 9. The comparison of RORO-MC and EWMOP optimization models.

TABLE IV
THE AVERAGE RECONFIGURATION RATE

QoS2%	RORO-MCH (T=5min)	MLU-PTE (T=5min)	MLU-PTE (T=1h)	CMLU-PTE (T=5min)	CMLU-PTE (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

effects of taking into account the routing costs in the RO-RO optimization function (RORO-MC). RORO-MC heuristic (RORO-MCH approach) was compared with MLU-PTE technique in terms of rejected bandwidth demand and average MLU in Fig. 8. In order to ensure a fair comparison, we also provide results for the periodic TE scheme (CMLU-PTE) that minimizes the objective function (12a). It can be observed that our heuristic exhibits almost the same behavior as CMLU-PTE (T=5min) and outperforms MLU-PTE (T=5min) noticeably. The results in terms of the average reconfiguration rate are given in Table IV. For the same T, RORO-MCH reduces reconfiguration overhead up to 14 times in comparison to MLU-PTE, and up to 25 times in comparison to CMLU-PTE.

We compare RORO-MC and EWMOP in Fig. 9. The results confirm a favorable behavior of RORO-MC. Moreover, it is interesting to note that RORO-MCH provides a noticeable improvement over EWMOP.

The impact of T parameter on the RORO-MCH behavior was analyzed in Fig. 10. In contrast to RORO-H, RORO-MCH is not sensitive to the changes of T. However, a lower T is preferable for efficient handling of sudden network distortions.

In Fig. 11, we show the simulation results for three different values of R_{avg} (1.2, 1.6 and 2). As expected, increasing R_{avg} leads to better utilization of the network resources. This could be explained by the fact that the controller is allowed to perform reconfigurations more frequently. In case when R_{avg} value is low, the loss in terms of rejected bandwidth demand is noticeable for RORO-MCH. However, Fig. 11b confirms that this happens because the proposed heuristic adjusts its decisions to the time-average reconfiguration constraint. The reconfiguration rate of RORO-MCH, in the worst scenario, just a slightly exceeds the R_{avg} threshold.

The computational complexity of RORO-MCH algorithm depends on the network size. However, it can be controlled by limiting the number of LP iterations - N_{max}. We have evaluated the computational time of RORO-MCH as a function of N_{max} and R_{avg}. The results are summarized in Table V. All the results were obtained on Windows machine with 16GB of RAM and i7 processor. As expected, the computational time increases with the number of iterations. Therefore, it is important to determine a good compromise value for N_{max}. From Table V, it can be observed that the impact of N_{max} on BoD rejection ratio diminishes as R_{avg} increases. In all of the analyzed scenarios, setting N_{max} to 100 was sufficient for near

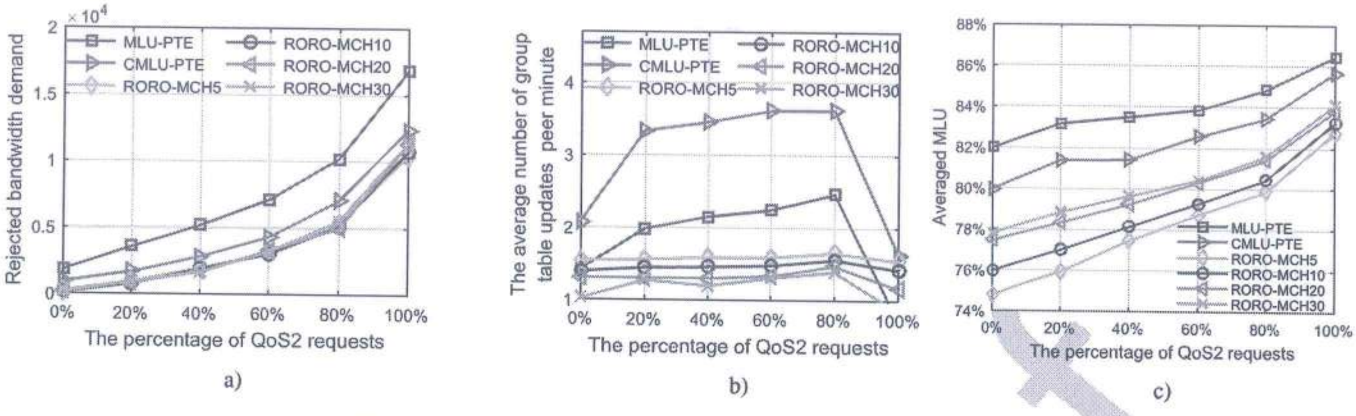


Fig. 10. The impact of T on RORO-MCH performance.

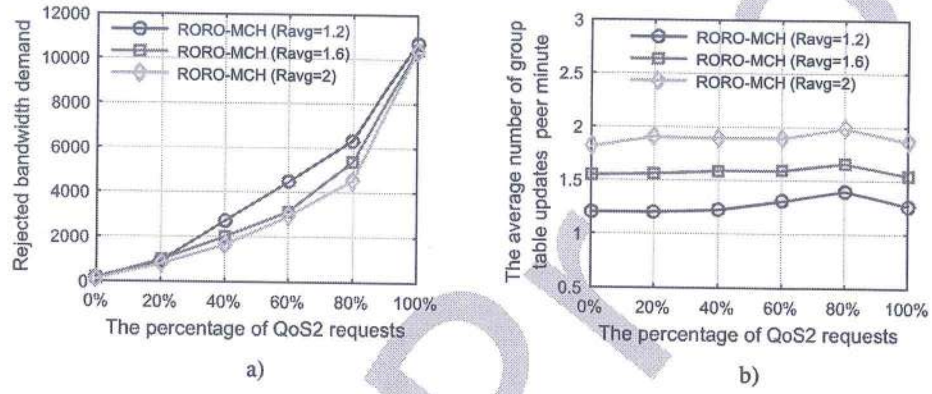


Fig. 11. The impact of R_{avg} on RORO-MCH performance.

TABLE V
THE IMPACT OF N_{max} ON COMPUTATIONAL TIME (CT) AND REJECTED BANDWIDTH (RB) DEMAND

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

TABLE VI
THE EXPERIMENT SETTING

Network topology	Sprint (PoP level)
Link capacity	100 Mbit/s
Reservable link capacity	80 Mbit/s
Link capacity reserved for control traffic	10 Mbit/s
BoD request arrival process	Poisson for each IE pair (mean = IE average demand/ average request duration)
The bandwidth demand of BoD requests	1 Mbit/s
The duration of BoD requests	Exponential distribution (mean = 20 minutes)
The max. number of paths between an IE pair (K)	10
Link disjointness parameter for IE pair (q)	1
QoS1 delay bound	35ms
Experiment duration	1h
R_{avg}	2.4 reconfig. per minute

optimal RORO-MCH performance. For that configuration, computational time was around 15s in the worst case.

V. EXPERIMENTAL VALIDATION

In order to validate the proposed solution, we implemented RORO-MCH control application on Floodlight OpenFlow controller [16]. The experiments were carried out in Mininet emulator with OVS switches. Due to the Mininet's scalability limitations, the parameters of the experimental set-up were modified with regard to the simulation set-up, as shown in Table VI. To generate traffic matrix for the hour of the maximum load we used the methodology from Section IV. The bandwidth demand of each BoD request was 1Mb/s. If the request is accepted by the controller, traffic load for the request is generated in the form of 5 UDP sub-flows, with the data rate of 200 Kbps per flow. The controller enforces the admission control policy that prevents link-utilization over 90%. However, the links still could become over-utilized as a result of inaccurate traffic splitting over multiple paths.

A. Implementation Details

The system prototype is developed according to the architecture shown in Fig. 1. Our controller exposes the REST interface for BoD negotiation. The BoD requests are sent in JSON format, indicating relevant parameters such as: BoD type, IE

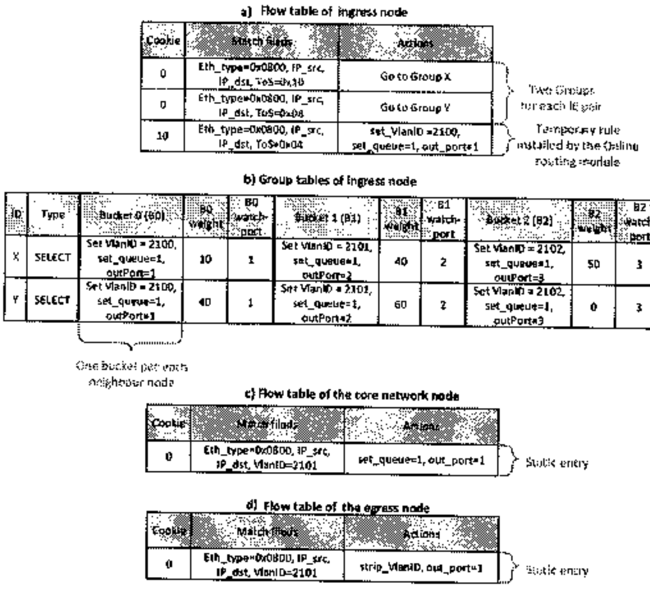


Fig. 12. Example of flow table and group table configurations for ingress node (a, b), core (c) and egress (d) node.

pair and the expiration time for the request. The tunnel setup is delegated to the another REST module which, for the given input parameters K and q, installs K least-delay q-disjoint paths between each IE pair. *RO-RO optimizer* is scheduled to run RORO-MCH algorithm periodically (every 5 minutes) within a separate thread.

The flow-table format for the ingress, core and egress node of a given IE pair is illustrated in Fig. 12. Ingress nodes maintain two static flow-table entries per each egress node. Those entries are linked with group-tables that are responsible for handling traffic flows of the particular type (QoS1 or QoS2 flows). In our experiments, we assume that each egress node is used to reach a distinct range of a priori known IP addresses. Thus, based on the packet destination address, ingress node is able to identify the egress node for the incoming flow. The flow is then processed either by the appropriate group-table (load-balancing rules determined by the RO-RO optimizer), or directly forwarded to the out-port based on the online routing rules. In our experiments, we used Type of Service (ToS) field in IP header to map a traffic flow with QoS type.

In the proposed system, SELECT group-tables are used to implement load-balancing over multiple routes. Each group-table has a unique identifier (at node level) and contains action buckets which specify set of actions to be executed on entering packets. Each bucket forwards packets to the one of the routes installed between the IE pair. As shown in Fig. 12b, buckets are associated with a weight parameter, which defines the bucket's share of the traffic processed by the group. More precisely, the amount of traffic assigned to a bucket is defined by the individual bucket's weight divided by the sum of the bucket weights in the group [30]. If bucket's weight is zero, it is not used by the group. The bucket selection algorithm, in general, depends on the switch model. OVS switches compute a simple hash function on a user-configurable tuple of L2-L3 packet header fields. Since address-based hashing

TABLE VII
PACKET LOSS AND JITTER RESULTS

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

(supported by OVS [42]) is not suitable for our experiment where hardware capabilities of the host machine impose limits on maximum number of emulated devices, we modified a source code of OVS switch to include L4 port numbers in hash computation.

Since routes between IE pair are not necessarily link-disjoint, each of them is associated with a specific label in order to avoid conflicting rules in the core switches. For example, if two routes have routing segment in common, and core OpenFlow switches make forwarding decision just based on source IP and destination IP address ranges, rules installed for these two routes might conflict each other. To avoid this situation, traffic flows using different routes between a common IE node pair are tagged with different labels. We used VLAN ID to differentiate flows traversing distinct routes between an IE pair. As shown in Fig. 12c, core switches perform just simple forwarding actions which could be supported by ordinary MPLS routers. In case that MPLS is used in the network core, ingress nodes can simply tag incoming packets with the appropriate MPLS label (instead of VLAN IDs).

In order to prevent loss of control traffic due to high traffic load, two queues are configured on each network interface: i) default queue 0 - which is used for control traffic, and ii) queue 1 - which accommodates the generated traffic load, with maximum service rate limited to 90% of link capacity.

When the QoS requirements of the BoD request cannot be met with current load balancing scheme, the *Online-Routing* module of the controller installs additional flow-table rules at ingress switches. Those rules are of higher priority. They match only the IP address range specified in the BoD request, tag the matching packets with the appropriate VLAN ID (or MPLS label) and forward them to the given out-port directly.

B. Experimental Results

We have conducted several experiments with different percentages of QoS2 BoD requests (0%, 40%, 60% and 100%). With regard to QoS indicators for traffic flows, here we limit our analysis to packet loss rate and jitter, which are reported by Iperf [43] traffic generator. The obtained results in terms of average and maximum packet loss rate (denoted as APLR and MPLR respectively) and jitter are shown in Table VII. These results are valuable because in the simulation analysis we assumed that ingress switches are able to accurately realize load-balancing according to the decisions made by the RO-RO optimizer. However, the performance of bucket selection algorithm is affected by many factors (e.g flow granularity, hash-function, etc.). Therefore, the QoS cannot be guaranteed without leaving some spare capacity on the links

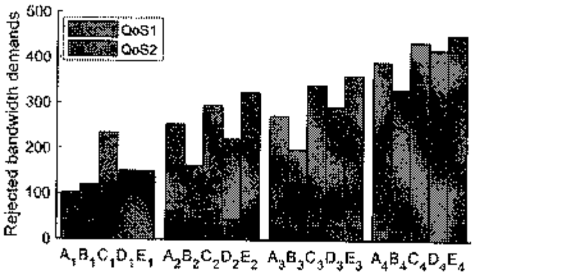


Fig. 13. Rejected bandwidth demand in Mininet experiments. The analyzed algorithms are denoted as follows: A = RORO-MCH, B = CMLU-PTE (T=5min), C = CMLU-PTE (T = 1h), D = MLU-PTE (T = 5min), E = MLU-PTE (T = 1h). The indexes {1, 2, 3, 4} denote different traffic load scenarios with {0, 40, 60, 100}% of QoS2 requests.

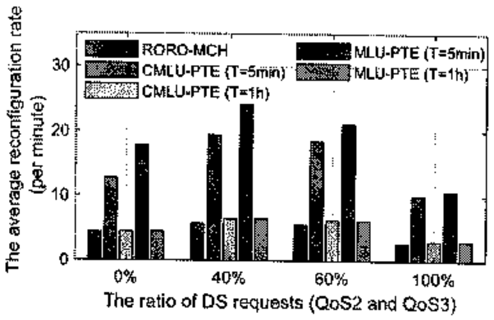


Fig. 14. Average reconfiguration rate in Mininet experiments.

that should compensate the mentioned limitations of flow-level load balancing. This makes the proposed implementation approach arguable. However, our experimental results alleviate concerns in that regard. Namely, although aggregated traffic demand of a single request is generated with only 5 traffic flows, MPLR did not exceed 1%, while APLR is very low. These results are achieved with the admission control policy that maintains link utilization below 90%. In practice, even more strict policies are considered reasonable due to queuing delay implications. The jitter measurements from our experiment (Table VII) are the indicator of queuing delay introduced by a high traffic load.

In order to validate the proposed solution, we also implemented MLU-PTE and MLU-CPTE algorithms on Floodlight controller. The obtained results in terms of the rejected bandwidth demand and average reconfiguration rate are shown in Figs. 13 and 14. Obviously, the experimental results confirm a favorable behavior of RORO-MCH.

VI. Conclusion

The main contribution of this paper is a proposal of a new technique for BoD service provisioning in SD-ISP networks which addresses the problem of a frequent flow-table updates due to evolving traffic demands. The problem was mathematically formulated and an optimal scheme, called RO-RO, was proposed to solve the corresponding optimization problem. The RO-RO strives to dynamically balance the resource consumption in the network in order to maximize the BoD request acceptance ratio, considering as a constraint the ISP's preference in terms of the time-average reconfiguration rate. Since the computational complexity of the proposed solution is high, we have proposed a heuristic method to dynamically find an efficient trade-off between the routing optimality gap and the reconfiguration overhead. The obtained simulation results show that the heuristic maintains an acceptable routing optimality gap, while the reconfiguration overhead is reduced significantly compared to conventional periodic TE schemes. We also proposed the RO-RO heuristic which adopts minimization of the routing cost as one of the TE objectives (RORO-MCH). It has been shown that RORO-MCH outperforms the state-of-the art TE solution [11] that jointly minimizes MLU and the reconfiguration cost. The proposed solution has been implemented on Floodlight OpenFlow controller and tested in Mininet.

In our future work, we will analyze the suitability of different evolutionary algorithm (such as [44]–[46]) for the implementation of RO-RO heuristic, with the goal of further reducing the optimality gap and potentially the worst case response time. Further on, the hierarchical structure of the control framework will be developed, in order to meet requirements of large-scale ISP networks.

References

[1] C.-Y. Hong *et al.*, "Achieving high utilization with software-driven wan," in *Proc. SIGCOMM*, Aug. 2013, pp. 15–26.
[2] S. Jain *et al.*, "B4: Experience with a globally-deployed software defined WAN," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 43, no. 4, pp. 3–14, 2013.
[3] A. Mendiola, J. Astorga, E. Jacob, and M. Higuero, "A survey on the contributions of software-defined networking to traffic engineering," *IEEE Commun. Surveys Tuts.*, vol. 19, no. 2, pp. 918–953, 2nd Quart., 2016.
[4] M. Al-Fares *et al.*, "Hedera: Dynamic flow scheduling for data center networks," in *Proc. NSDI*, Berkeley, CA, USA: USENIX, Apr. 2010, p. 19. [Online]. Available: http://dl.acm.org/citation.cfm?id=1855711.1855730
[5] M. Yu, J. Rexford, M. J. Freedman, and J. Wang, "Scalable flow-based networking with DIFANE," *SIGCOMM Comput. Commun. Rev.*, vol. 40, no. 4, pp. 351–362, Oct. 2010.
[6] A. R. Curtis, J. C. Mogul, J. Tourrilhes, P. Yalagandula, P. Sharma, and S. Banerjee, "DevoFlow: Scaling flow management for high-performance networks," *Comput. Commun. Rev.*, vol. 41, no. 4, pp. 254–265, Aug. 2011.
[7] J. M. Wang, Y. Wang, X. Dai, and B. Bensaou, "SDN-based multi-class QoS-guaranteed inter-data center traffic management," in *Proc. IEEE 3rd Int. Conf. Cloud Netw. (CloudNet)*, Oct. 2014, pp. 401–406.
[8] X. Jin *et al.*, "Optimizing bulk transfers with software-defined optical wan," in *Proc. ACM SIGCOMM Conf.*, Aug. 2016, pp. 87–100.
[9] H. H. Liu *et al.*, "Traffic engineering with forward fault correction," *SIGCOMM Comput. Commun. Rev.*, vol. 44, no. 4, pp. 527–538, Aug. 2014. doi: 10.1145/2740070.2626314.
[10] S. Paris, A. Destounis, L. Maggi, G. S. Paschos, and J. Leguay, "Controlling flow reconfigurations in SDN," in *Proc. IEEE INFOCOM*, Apr. 2016, pp. 1–9.
[11] U. Ashraf, "Rule minimization for traffic evolution in software-defined networks," *IEEE Commun. Lett.*, vol. 21, no. 4, pp. 793–796, Apr. 2017.
[12] S. Tomovic and I. Radusinovic, "Traffic engineering approach to virtual-link provisioning in software-defined ISP networks," in *Proc. 25th Telecommun. (TELFOR)*, Nov. 2017, pp. 1–4.
[13] S. Paris *et al.*, "Dynamic control for failure recovery and flow reconfiguration in sdn," in *Proc. 12th Int. Conf. Design Reliable Commun. Netw.*, Mar. 2016, pp. 152–159.
[14] 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, 2013.
[15] M. J. Neely, *Stochastic Network Optimization with Application to Communication and Queueing Systems*. New York, NY, USA: Morgan Publishers, 2010.

[16] (2018). *Floodlight Openflow Controller.* [Online]. Available: http://www.projectfloodlight.org/floodlight/

[17] S. Das *et al.*, "Application-aware aggregation and traffic engineering in a converged packet-circuit network," in *Proc. Opt. Fiber Commun. Conf. Expo. Nat. Fiber Optic Eng. Conf.*, Mar. 2011, pp. 1–3.

[18] A. Farrel, J. P. Vasseur, and J. Ash, "A Path Computation Element (PCE)-Based Architecture," IETF, Fremon, CA, USA: Tech. Rep. RFC 4655, Aug. 2006. [Online]. Available: www.ietf.org/rfc/rfc4655.txt

[19] A. Ghosh, S. Ha, E. Crabbe, and J. Rexford, "Scalable multi-class traffic management in data center backbone networks," *IEEE J. Sel. Areas Commun.*, vol. 31, no. 12, pp. 2673–2684, December 2013.

[20] J. L. Martins and N. Campos, "Short-sighted routing, or when less is more," *IEEE Communi. Mag.*, vol. 54, no. 10, pp. 82–88, Oct. 2016.

[21] S. Tomovic and I. Radusinovic, "Fast and efficient bandwidth-delay constrained routing algorithm for SDN networks," in *Proc. IEEE NetSoft Conf. Workshops (NetSoft)*, Jun. 2016, pp. 303–311.

[22] S. Tomovic and I. Radusinovic, "A new traffic engineering approach for QOS provisioning and failure recovery in SDN-based ISP networks," in *Proc. 23rd Int. Sci.-Prof. Conf. Inf. Technol.(IT)*, Feb. 2018, pp. 1–4.

[23] J. W. Guck *et al.*, "Function split between delay-constrained routing and resource allocation for centrally managed qos in industrial networks," *IEEE Trans. Ind. Inform.*, vol. 12, no. 6, pp. 2050–2061, Dec 2016.

[24] M. Caria, A. Jukan, and M. Hoffmann, "A performance study of network migration to sdn-enabled traffic engineering," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Dec 2013, pp. 1391–1396.

[25] Y. Guo, Z. Wang, X. Yin, X. Shi, and J. Wu, "Traffic engineering in hybrid sdn networks with multiple traffic matrices," *Comput. Netw.*, vol. 126, pp. 187–199, Oct. 2017. [Online]. Available: [Online]. Available: http://www.sciencedirect.com/science/article/pii/S138912861730289X

[26] Y. Hu, W. Wang, X. Gong, X. Que, Y. Ma, and S. Cheng, "Maximizing network utilization in hybrid software-defined networks," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Dec 2015, pp. 1–6.

[27] S. Agarwal, M. Kodialam, and T. V. Lakshman, "Traffic engineering in software defined networks," in *Proc. INFOCOM*, April 2013, pp. 2211–2219.

[28] Y. Guo, Z. Wang, X. Yin, X. Shi, and J. Wu, "Traffic engineering in sdn/ospf hybrid network," in *Proc. IEEE 22nd Int. Conf. Netw. Protocols*, Oct 2014, pp. 563–568.

[29] M. Moradi, Y. Zhang, Z. M. Mao, and R. Manghirmalani, "Dragon: Scalable flexible, and efficient traffic engineering in software defined isp networks," *IEEE J. Sel. Areas Commun.*, vol. 36, no. 12, pp. 2744–2756, Dec 2018.

[30] (2018). *Open Networking Foundation—Openflow v1.4 Specification.* [Online]. Available: https://www.opennetworking.org

[31] C. Janz, L. Ong, K. Sethuraman, and V. Shukla, "Emerging transport sdn architecture and use cases," *IEEE Commun. Mag.*, vol. 54, no. 10, pp. 116–121, Oct. 2016.

[32] *Open Networking Foundation: Use Cases for Carrier Grade Sdn.* Accessed: Mar. 4, 2019. [Online]. Available: https://www.opennetworking.org/

[33] P. Ngatchou, A. Zarei, and A. El-Sharkawi, "Pareto multi objective optimization," in *Proc. 13th Int. Conf., Intell. Syst. Appl. Power Syst.*, Nov. 2005, pp. 84–91.

[34] C. Fraleigh *et al.*, "Provisioning ip backbone networks to support latency sensitive traffic," in *Proc. IEEE INFOCOM*, vol. 1, Mar. 2003, pp. 375–385.

[35] L. J. Wismans, T. Brands, E. C. Van Berkum, and M. C. Bliemer, "Pruning andranking the pareto optimal set, application for the dynamic multi-objective network design problem," *J. Adv. Transp.*, vol. 48, no. 6, pp. 588–607, 2016.

[36] B. Fortz and M. Thorup, "Optimizing ospf/is-is weights in a changing world," *IEEE J. Sel. Areas Commun.*, vol. 20, no. 4, pp. 756–767, May 2002.

[37] (2018). *Cplex Optimization Studio.* [Online]. Available: http://www-03.ibm.com/software/products/en/ibmilogcpleoptistud

[38] R. Teixeira, K. Marzullo, S. Savage, and G. M. Voelker, "In search of path diversity in isp networks," in *Proc. 3rd ACM SIGCOMM Conf. Internet Meas.*, 2003, pp. 313–318. doi: 10.1145/948205.948247.

[39] A. Gunnar *et al.*, "Traffic matrix estimation on a large ip backbone: A comparison on real data," in *Proc. 4th ACM SIGCOMM Conf. Internet Meas.*, Sep. 2004, pp. 149–160.

[40] K. Thompson, G. J. Miller, and R. Wilder, "Wide-area Internet traffic patterns and characteristics," *IEEE Netw.*, vol. 11, no. 6, pp. 10–23, Nov. 1997.

[41] J. He and J. Rexford, "Date: Distributed adaptive traffic engineering," in *Proc. INFOCOMM*, Aug. 2006, pp. 1–10.

[42] B. Pfaff *et al.*, "The design and implementation of open vswitch," in *Proc. NSDI*, 2015, pp. 117–130.

[43] (2016). *Iperf3 Traffic Generator.* [Online]. Available: https://iperf.fr

[44] S.-C. Chu, P.-w. Tsai, and J.-S. Pan, "Cat swarm optimization," in *Trends in Artificial Intelligence*, Q. Yang and G. Webb, Eds. Berlin, Germany: Springer, 2006, pp. 854–858.

[45] D. Wu, F. Kong, W. Gao, Y. Shen, and Z. Ji, "Improved chicken swarm optimization," in *Proc. IEEE Int. Conf. Cyber Technol. Automat., Control, Intell. Syst. (CYBER)*, Jun. 2015, pp. 681–686.

[46] K. S. Tang, K. F. Man, S. Kwong, and Q. He, "Genetic algorithms and their applications," *IEEE Signal Process. Mag.*, vol. 13, no. 6, pp. 22–37, Nov 1996.

Slavica Tomovic received the M.Sc. degree in electronics, telecommunications and computer science from the Faculty of Electrical Engineering, University of Montenegro, Podgorica, in 2015, where she is currently pursuing the Ph.D. degree. Her main research interests are in the areas of software-defined networking, quality of service (QoS) management and architectures, and the Internet of Things.

Igor Radusinovic received the M.Sc. and Ph.D. degrees from the University of Belgrade, Serbia, in 1997 and 2003, respectively. He is currently a Full professor in telecommunications networks and switching systems with the University of Montenegro. He has published more than 120 papers in international and national scientific journals and conferences as an author or co-author. His main research interests are in the areas of software-defined networking, packet switching, and congestion control mechanisms.

IEICE TRANSACTIONS

on Communications

VOL. E102-B NO. 4
APRIL 2019

A PUBLICATION OF THE COMMUNICATIONS SOCIETY
The Institute of Electronics, Information and Communication Engineers
Kikai-Shinko-Kaikan Bldg., 5-8, Shibakoen 3chome, Minato-ku, TOKYO, 105-0011 JAPAN

PAPER

An Effective Use of SDN for Virtual-Link Provisioning in ISP Networks

Slavica TOMOVIĆ[†a)], *Nonmember and* Igor RADUSINOVIĆ[†], *Member*

SUMMARY The ability of Software Defined Networking (SDN) to dynamically adjust the network behaviour and to support fine-grained routing policies becomes increasingly attractive beyond the boundaries of Data Centre domains, where SDN has already gained enormous momentum. However, the wider adoption of SDN in ISP (Internet Service Provider) networks is still uncertain due to concerns about the scalability of a centralized traffic management in large-scale environments. This is particularly problematic when ISP offers virtual-link services, which imply a performance guaranteed data transfer between two network points. Our solution is a new approach to virtual-link mapping in SDN-based ISP networks. Within the problem's scope, we address traffic engineering (TE), QoS provisioning and failure recovery issues. In order to decrease the controller load, computational effort, and processing delay, we introduce a function split between online routing and TE. The TE functions are performed periodically, with configurable periodicity. In order to reduce the control overhead, we restrict the traffic optimization problem to load balancing over multiple static tunnels. This allows retention of the traditional MPLS routers in the network core and to achieve fast virtual-link restoration in case of physical-link failures. The online routing and admission control algorithms have been designed with the goal of low complexity, and to minimize Flow-table updates. In our simulation study, we compare the proposed virtual-link mapping solution with the solutions that exploit routing flexibility in fully SDN-enabled networks. We find that the throughput loss due to the use of static traffic tunnels is relatively small, while the control overhead is reduced significantly. A prototype of the proposed SDN control-plane is developed and validated in the Mininet emulator.
key words: ISP, SDN, OpenFlow, QoS

1. Introduction

With the expansion of Internet of Things (IoT) and the increasing confluence of augmented reality and artificial intelligence, providing reliable and low-latency communication becomes one of the crucial challenges for Internet Service Providers (ISPs). Nowadays, ISPs offer virtual-link services to business users, as an option for QoS (Quality of Service)-guaranteed data transfer between two points in the network [1]. Virtual-links are mostly used to interconnect remote company sites or transfer sensitive data to the Cloud. However, QoS guarantees in traditional ISP networks are achieved at the expense of link over-provisioning. Namely, backbone links are over-provisioned 2 to 3 times relative to the offered load [2], [3]. Since this is not a viable solution in the long run, a great amount of effort has been put into the development of high-performance traffic engineering (TE) mechanisms [4].

While TE techniques were widely investigated in the context of ATM (Asynchronous Transfer Mode) and IP/MPLS (Internet Protocol/Multi-Protocol Label Switching) networks in the past, it has been shown that the distributed control plane hardly supports complex TE operations without jeopardizing the network stability [5]. Moreover, due to the lack of complete visibility of the network state, TE decisions are made greedily, failing to provide a global optimality [2]. SDN (Software Defined Networking) has brought TE back to the light by logically centralizing the network control plane. However, when it comes to the large-scale SDN deployments, there are a lot of concerns for performance and scalability. For example, while the capability to dynamically re-optimize traffic distribution is often considered as one of the major strengths of SDN, frequent network reconfigurations degrade QoS because the controller needs to tear down the old routing paths and set up the new ones [2], [5], [6]. A full control over the all traffic flows results in the control plane bottleneck. In situations when a controller's reaches a certain threshold, the processing latency makes a non-negligible factor in total end-to-end latency [7]. Furthermore, a complete traffic visibility might require hundreds of thousands of flow table entries at each SDN switch, while the current state-of-the-art limitation is 16K entries per switch [8].

The adoption of SDN in ISP networks imposes at least two more challenges: i) a huge capital expenditures for upgrading the backbone routers [9]; ii) most of the existing SDN-based TE solutions are designed for Data Centre (DC) and inter-DC networks, and cannot be directly applied in ISP networks where Service Level Agreements (SLAs) must be guaranteed. Therefore, we propose a solution for virtual-link mapping with the following objectives in mind:

- Support for gradual SDN upgrade - allow ISPs to integrate traditional MPLS technology with SDN by using SDN capabilities at the provider edge to optimize load balancing over a static set of MPLS tunnels;
- Provide reliable, low-latency, bandwidth-guaranteed data transfers on-demand;
- Make admission control decisions in near real-time;
- Minimize control plane - data plane communication.

In order to achieve the aforementioned objectives we organize the SDN controller's logic into the *online* and the *offline* module. The *online* module makes routing and admission control decisions on virtual-link requests. Since each virtual-

Manuscript received June 29, 2018.
Manuscript revised September 19, 2018.
Manuscript publicized October 18, 2018.
[†]The authors are with the Faculty of Electrical Engineering, University of Montenegro, Dzordza Vasingtona bb, 81000, Podgorica, Montenegro.
a) E-mail: slavicat@ucg.ac.me
DOI: 10.1587/transcom.2018EBP3191

link must be approved by the controller individually, this module runs a low complexity routing algorithm that tends to avoid installation of new Flow-table rules. The *offline* module periodically optimizes the traffic distribution over a set of preconfigured traffic tunnels in the network. This way, we trade some routing optimality for the sake of the control plane scalability. Our simulations have shown that ISP networks with SDN-enabled edge, using the proposed solution, could achieve virtual-link rejection ratio close to that achievable in a fully SDN-enabled network when periodic traffic optimizations are performed. On the other side, the reduction in the control overhead is significant. We also present an experimental prototype of the proposed solution in order to validate the use of OpenFlow-based technology.

The rest of the paper is organized as follows. In Sect. 2 we provide background and explain the motivation for our work. Section 3 explains the system model. Section 4 presents the simulation methodology and the obtained results. Sections 5 explains the prototype implementation and experimental results. The paper is concluded in Sect. 6.

2. Background and Motivation

In order to use the network capacity efficiently, ISPs today mostly use MPLS technology, which offers TE capability. MPLS ingress routers typically use equal cost multipath (ECMP) routing to split traffic equally across multiple tunnels to the same egress. If ISP needs to provide QoS guarantees, CSPF routing algorithm is used [4]. Although MPLS has many attractive features, ISP networks still suffer from poor efficiency. This could be attributed to the fact that each ingress router installs LSPs independently, without coordination with other routers. The overall result is a suboptimal traffic distribution. Also, the practice has shown that MPLS networks slowly adapt to network changes [10]. With an emergence of SDN, a simpler, centralized control system is being developed. However, in large-scale network deployments, the centralized architecture faces serious scalability challenges. In order to improve the scalability of the SDN control plane, two research directions were mainly followed [11]. The first is to design a decentralized control architecture, composed of multiple controllers that share the load by cooperating together [12]. This approach does not impose any new requirements on SDN switches but suffers from consistency problems when the controllers are widely geographically distributed [13]. The other approach is to delegate some control tasks to SDN switches [14]–[16]. This could reduce control overhead significantly, but the modifications of a switch hardware are necessary. In contrast to these approaches, we exploit the existing OpenFlow capabilities [17] to improve the SDN scalability in a scenario where complex QoS and TE goals should be met.

The early SDN works on TE solutions were mostly focused on DC networks. There also exist several inter-DC network designs such as Google B4 [18] and Microsoft SWAN [2] that use SDN to dynamically control the routing and rate allocation. SWAN improves the network utilization while meeting QoS policy goals, by centrally controlling when and how much traffic each DC service can send. B4 shares similar high-level architecture but mainly deals with the integration of the existing routing protocols in SDNs. While both SWAN and B4 provide only soft delay guarantees for delay-sensitive (DS) flows, MCTEQ TE algorithm [19] provides strict delay guarantees by considering queuing delay in the definition of TE optimization problem. However, TE algorithms designed for DC and inter-DC networks cannot be directly applied to ISP networks, where the incoming traffic cannot be scheduled or suppressed [2]. Recent works [1], [5], [20], [21] take into account this limitation and propose TE algorithms for SDN-based ISP networks. In [20], a flow migration approach is proposed in order to dynamically manage network resources. A trade-off between SDN control plane scalability and the optimality of routing decisions has been investigated in [5]. However, the traffic differentiation was not considered. Our previous works [1], [21] propose QoS-aware mechanisms for virtual-link provisioning and compare their performance with the current practice MPLS-based solutions. In this paper, we analyse the scalability gains and performance limitations of the proposed mechanism that exploits SDN capabilities only at the network edge with regard to the solutions that benefit from a complete SDN deployment. Also, we have implemented the proposed solution on OpenFlow controller in order to validate its functionality.

A function split between delay-constrained routing and TE has already been proposed for industrial networks [22]. Namely, a reactive routing algorithm is used to quickly find the routes that meet a hard real-time QoS, while the link parameters (e.g. transmission bit rate and delay budget of different priority queues) are configured offline. In contrast to this work, we tend to perform offline optimizations of the load balancing parameters in ISP networks, considering not only the delay but also the bandwidth and the reliability as QoS constraints.

Maintaining a reliable end-to-end connection is an elementary task of an ISP network. New versions of OpenFlow protocol introduced a so-called fast-failover mechanism, which uses Group-tables to enable a quick local migration of traffic flows in response to the link failures [17]. However, these migrations do not account for the link capacity constraints. On the other hand, SDN controller could find suitable tunnels to carry disrupted demands, but a controller-initiated path restoration cannot meet the strict ISP requirement for the reconfiguration delay below 50ms [23]. In [24], TE and failure recovery are jointly considered in order to ensure a more efficient routing after the failures. An additional bandwidth is reserved for traffic flows in order to guarantee a failure recovery without congestion when some of the a priori defined fault models are detected. We exploit this approach and extend it with QoS-awareness in terms of delay and bandwidth.

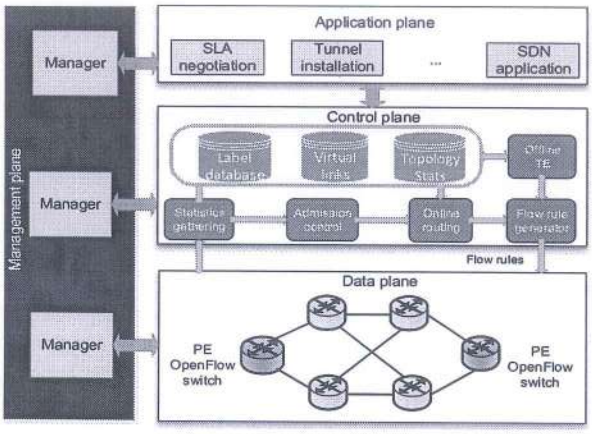


Fig. 1 The system architecture.

3. The System Model

We consider the network model from Fig. 1 [1]. The data plane consists of the Core (CR) and Provider Edge (PE) routers. CR routers are MPLS-enabled and carry the traffic to and from the OpenFlow-enabled PE switches. The application plane offers three types of SLAs for virtual-links:

1. QoS1 - bandwidth-guaranteed connectivity service.
2. QoS2 - guarantees very low delay bound and the required bandwidth for time-critical data transfers.
3. QoS3 - guarantees bandwidth, delay, and robustness to a pre-defined number (n) of link failures. It is intended for latency-sensitive ultra-reliable communication.

The decisions on virtual-link requests are made by the control plane, which consists of the following modules:

- *Statistics gathering* - pulls statistics from the network devices at regular intervals.
- *Admission-control* - makes a decision whether to accept or not a virtual-link request based on the current availability of the network resources.
- *Online routing* - computes the appropriate route(s) for virtual-link requests in accordance with the specified QoS requirements.
- *Offline-TE* - periodically optimizes the load balancing in the network.
- *Flow rule generator* - installs Flow-table rules.

Up to K tunnels are used for load-balancing between each ingress-egress (IE) pair of switches. In order to minimize the capacity loss in case of a link failure, tunnels of each IE pair are "q-disjoint", i.e. at most q of them can share the same link. When a link failure occurs, ingress switches redistribute the traffic traversing a failed link over the unaffected tunnels proportionally to the load balancing weights that are proactively installed by the controller. Such rebalancing is feasible with OpenFlow Group-tables [17].

The traffic is classified into three classes that correspond to the three different SLA types. Each traffic class is tagged in a unique way at the network entrance. Virtual-link requests are initially handled by the *Online routing* module of the SDN controller. The *Offline TE* module runs periodically, in order to optimize the network performance by redistributing the traffic over a set of pre-configured tunnels. The traffic splitting across the tunnels is performed on a flow-level, by applying a hash function on packets' header fields. Since virtual-links accommodate aggregated traffic demands (not individual flows) and the OpenFlow protocol supports a wide range of traffic classifiers [17], a fine-granularity of traffic splitting could be expected. Unequal traffic splitting ratios could be implemented by using the SELECT Group-tables [17] in the processing pipeline of OpenFlow switches. An ingress switch maps an incoming packet based on its destination and the label attached to one of the total three Group tables. Each Group-table corresponds to a different traffic class, and it is configured with (up to) K tunnels and K weights, which determine the ratio of the matching traffic which is to be sent to each of the tunnels.

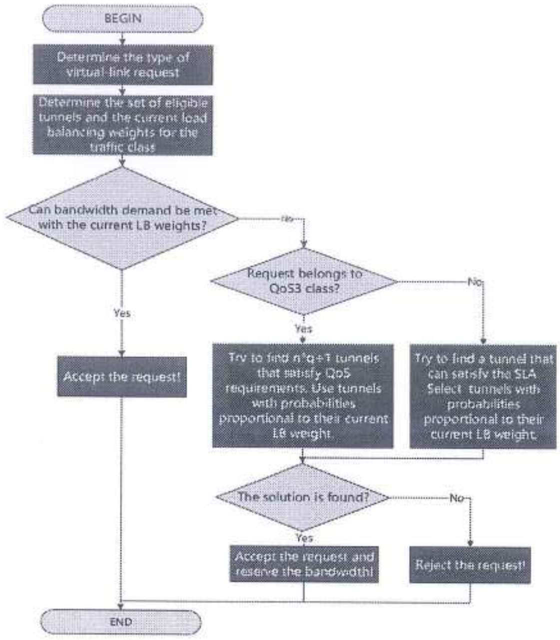


Fig. 2 The online routing algorithm. LB stands for load balancing.

3.1 Online Routing Module of the SDN Controller

This module performs the admission control for a virtual-link request by running computationally undemanding QoS routing algorithm. The workflow of the algorithm is shown in Fig. 2. In the first step, the algorithm checks the type of the virtual-link request in order to determine a subset of tunnels that can meet the delay requirement. Then, the algorithm checks whether the bandwidth requirement can be met when using the load balancing weights determined by the *Offline TE* module in its last running cycle. If yes, the request is accepted and no new Flow-table rules need to be installed in the network. The ingress switches will direct the flow to the appropriate Group-table, and tag and balance its sub-flows over multiple pre-configured tunnels. Otherwise, the algorithm seeks for an eligible tunnel that can carry the whole virtual-link without traffic splitting. If multiple tunnels are eligible, the preference is given to the one with the largest load-balancing weight according to the *Offline TE* module, i.e. that tunnel is selected with the highest probability. In this case, an ingress switch will not use the Group-tables to forward the flow's packets, thus a new Flow-table rule must be added. In case of QoS3 requests, the algorithm outputs a list of multiple tunnels. One of them is a primary tunnel that will be used to carry the traffic demand. The others are alternative tunnels on which the bandwidth will be reserved in order to ensure the robustness to link failures. In particular, the algorithm outputs a list of $n \cdot q + 1$ tunnels. This guarantees that the service disruption will not happen in any situation when up to n link fail. If the algorithm cannot find a tunnel(s) that can fulfill guarantees specified in the corresponding SLA, the request is rejected. It is important to note that in this stage the goal is not to optimize the routing decision, but rather to find a quick solution to allocate the virtual-links. This is because a bandwidth-delay-reliability constrained TE would result in a computationally hard problem that cannot be solved in real-time [20].

3.2 Offline TE Module of the SDN Controller

The *Offline TE* module of the controller periodically runs the optimization algorithm, which tends to minimize maximum link utilization (MLU) subject to the QoS constraints. The algorithm models ISP network as a directed graph $G = (V, L)$ where V is a set of nodes and L is a set of edges. Each link is described by two parameters: capacity c_l and propagation delay d_l. We assume that the propagation delay predominantly determines the total path delay. This assumption fits the ISP scenario, where low queuing delays are expected due to high-speed links and the effect of statistical multiplexing of the aggregated traffic flows [25].

A traffic matrix TM is $|N|x|N|$ matrix, where N is the set of edge OpenFlow switches. Matrix element t_{ij} refers to aggregated traffic demand between origin switch i and destination switch j. For the purpose of the network optimization, matrix TM is decomposed into three matrices TM_{qos1}, TM_{qos2} and TM_{qos3}, one for each traffic class. A traffic-class demand $t_{ij}^{(C)}$, $C \in \{qos1, qos2, qos3\}$ splits across the set of tunnels P_{ij}, which is computed a priori. For each traffic demand $t_{ij}^{(C)}$, the offline algorithm computes optimal splitting ratios $x_{p_{ij}}^{(C)}$ for tunnels $p_{ij} \in P_{ij}$, i.e. portions of the traffic demand that would be routed over each tunnel from the set P_{ij}. In order to ensure that QoS3 traffic demands will not experience congestion as long as a total number of link failures is less or equal to n (configurable bound), additional resources are reserved. The optimization algorithm is formulated as follows:

$$\min \quad z \tag{1a}$$

s.t.

$$\sum_{p_{ij}} x_{p_{ij}}^{(C)} = 1 \quad C \in \{qos1, qos2\}, \forall (i,j) \in N \tag{1b}$$

$$\sum_{p_{ij}} x_{p_{ij}}^{(qos3)} > 1 \quad \forall (i,j) \in N \tag{1c}$$

$$\sum_{\forall C} \sum_{(i,j) \in N} \sum_{p_{ij}} R_{p_{ij},l} \cdot x_{p_{ij}}^{(C)} \cdot t_{ij}^{(C)} \le z \cdot c_l \tag{1d}$$

$$y_{p_{ij}}^{(C)} \cdot \Big(\sum_{l \in p_{ij}} d_l - D_{max}^{(C)}\Big) \le 0 \quad \forall p_{ij}, C \tag{1e}$$

$$y_{p_{ij}}^{(C)} \ge x_{p_{ij}}^{(C)} \quad \forall C, (i,j) \in N, p_{ij} \tag{1f}$$

$$\sum_{p_{ij} \in P_{ij}^{\mu}} x_{p_{ij}}^{(qos3)} \ge 1 \quad \forall (i,j) \in N, \mu \in U_n \tag{1g}$$

$$x_{p_{ij}}^{(C)} \ge 0 \quad \forall C, (i,j) \in N, p_{ij} \tag{1h}$$

$$y_{p_{ij}}^{(C)} \in 0, 1 \quad \forall C, (i,j) \in N, p_{ij} \tag{1i}$$

$$0 \le z \le 1 \tag{1j}$$

The above optimization model minimizes MLU z subject to the set of constraints (1b) - (1j). The decision variable $x_{p_{ij}}^{(C)}$ defines a fraction of traffic demand to be routed over a path p_{ij}. The constraints (1b) and (1c) ensure that all traffic demands are routed. The link capacity constraint is given with (1d). $R_{p_{ij},l}$ is a binary value which indicates whether a path p_{ij} includes a link l. If so, $R_{p_{ij},l}$ is 1, otherwise 0. Since all QoS classes require bandwidth guarantees, z must be lower or equal to 1. In (1e), we have introduced decision variables $y_{p_{ij}}^{(C)}$, which guarantee delays for virtual links. This is necessary because total path delay for traffic classes $C \in \{qos2, qos3\}$ should not exceed $D_{max}^{(C)}$, which stands for the maximum tolerable delay for the class. If p_{ij} cannot meet this requirement for a traffic class, (1e) and (1f) ensure that no traffic of that class is routed over p_{ij}, i.e. $y_{p_{ij}}^{(C)}$ must be zero in order to satisfy (1e) and $x_{p_{ij}}^{(C)}$ equals 0 because of (1f). Here, we assume that $D_{max}^{(qos1)}$ is infinitely large.

The computational challenge is imposed by (1g), which ensures that QoS3 traffic will maintain the required throughput when up to n link fail. A case of link faults can be represented by a vector $\mu = [\mu_l | l \in L]$, where μ_l is 1 when link l has failed or 0 otherwise. Therefore, a traffic demand is robust to n link failures if its bandwidth requirement can be met under a set of failure cases $U_n = \{\mu | \sum_l \mu_l \le n\}$. Given a fault case μ, for each traffic demand $t_{ij}^{(qos3)}$, we can determine a set of residual tunnels P_{ij}^{μ} that do not traverse any failed link. The constraints (1g) require these tunnels are able to handle $t_{ij}^{(qos3)}$. Clearly, a larger number of residual tunnels entails greater network throughput. Since tunnels between an IE pair are q-disjoint, for any link fault case $\mu \in U_n$, the number of residual tunnels $|P_{ij}^{\mu}|$ is no less than $|P_{ij}| - n \cdot q$. Thus, following the approach from [24], we can replace $|N| \cdot |N-1| \cdot \sum_{k=1}^{n} \binom{|L|}{k}$ constraints from (1g) with:

$$\sum_{k=1}^{|P_{ij}|} x_{p_{ij}}^{(qos3)} - \sum_{k=1}^{n \cdot q} x_{p_{ij}}^{maxk^{(qos3)}} \ge 1 \quad \forall (i,j) \in N \tag{2}$$

where $x_{p_{ij}}^{maxk^{(qos3)}}$ denotes the k-th largest element in $X =$

$\{x_{p_{ij}}^{(qos3)} | p_{ij} \in P_{ij}\}$. The method for encoding M largest variables as linear constraints is explained in [24].

4. The Simulation Results

To evaluate the performance of the proposed TE model, we have developed a simulator in Python that uses CPLEX [26] to solve optimization problems. In simulations, we used the POP-level Sprint ISP topology [1]. Link capacities were set to 1000 units, while delay on each link was derived from a great circle distance between the connected POPs and speed of light in the fiber. Since for most ISP networks real traffic traces are not publicly available, we used the well-known gravity model [27] to generate an average value of traffic matrix TM for the hour of the maximum load during a day. By scaling the gravity model matrix, we derived "admissible" traffic matrix, which brings MLU to at most 70% when flows are not delay-sensitive (DS). Based on an "admissible" traffic matrix, the dynamic evolution of the network traffic has been simulated. We used exponential distribution with the mean $t = 20\,min$ to model virtual-link durations and Poisson distribution to model arrivals of virtual-link requests for each IE pair. The mean rate of request arrivals during the hour of the maximum load is computed so that the resulting traffic model in average converges to the "admissible" traffic matrix. For simplicity, the bandwidth requirement of each virtual-link was chosen randomly from the set [3, 5, 7] units. The QoS2 delay bound was set to 35ms, which corresponds to delay on the shortest path between the mutually most distant IE pair on the Sprint topology. Maximum tolerable delay for QoS3 traffic class was set to 100ms. Unless otherwise specified, the period between offline TE phases was set to 20 minutes. The number of tunnels between each IE pair (K) was bounded by 15, while the path disjointness parameter q was set to 3. For simplicity, we assumed that QoS3 SLA guarantees robustness to a single link failure ($n = 1$). Duration of simulations was set to one day, and a request arrival rate during that period was scaled as illustrated in Fig. 3.

Since, as far as we know, in the literature there are no SDN-based TE methods that could be directly applied for virtual-link mapping in ISP networks, herein, we define two more solutions in order to evaluate the pros and cons of the scalability remediations introduced in our paper. These two solutions are better optimized in terms of throughput, but require fully SDN-enables network infrastructure that

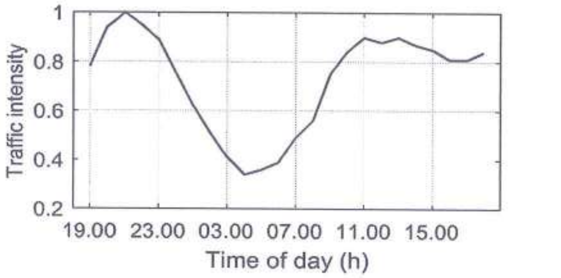


Fig. 3 A daily traffic load pattern derived from research results in [28].

allows routing flexibility in the network core. In the rest of the paper, we refer to them as SDN-OPT and SDN-MCFP. We denote the proposed load balancing TE method with SDN-LB. The SDN-OPT maximizes the network utilization by performing QoS-constrained multi-commodity flow (MCF) optimization [29] each time when a virtual-link cannot be successfully accommodated with the online routing algorithm. The MCF problem is formulated as follows:

$$\min \quad z + \varepsilon \cdot \sum_{l \in L} \sum_{(i,j) \in N} (a_{ij,l}^{(qos3)} + \sum_{\forall C} f_{ij,l}^{(C)}) \tag{3a}$$

s.t.

$$\sum_{l \in L^+(k)} f_{ij,l}^{(C)} - \sum_{l \in L^-(k)} f_{ij,l}^{(C)} = \begin{cases} +1 & ,if\ \text{k=i} \\ -1 & ,if\ \text{k=j} \\ 0 & ,else \end{cases} \tag{3b}$$

$\forall (i, j, k) \in N, \forall C$

$$\sum_{l \in L^+(k)} a_{ij,l}^{(qos3)} - \sum_{l \in L^-(k)} a_{ij,l}^{(qos3)} = \begin{cases} +1 & ,if\ \text{k=i} \\ -1 & ,if\ \text{k=j} \\ 0 & ,else \end{cases} \tag{3c}$$

$$\sum_{(i,j) \in N} (t_{ij}^{(qos3)} \cdot a_{ij,l}^{(qos3)} + \sum_{\forall C} f_{ij,l}^{(C)} \cdot t_{ij}^{(C)}) \le z \cdot c_l \quad \forall l \in L \tag{3d}$$

$$yf_{ij,l}^{(C)} \ge f_{ij,l}^{(C)} \quad \forall l \in L, \forall (i,j) \in N, \forall C \tag{3e}$$

$$ya_{ij,l}^{(qos3)} \ge a_{ij,l}^{(qos3)} \quad \forall l \in L, \forall (i,j) \in N \tag{3f}$$

$$yf_{ij,l^+}^{(qos3)} + yf_{ij,l^-}^{(qos3)} + ya_{ij,l^+}^{(qos3)} + ya_{ij,l^-}^{(qos3)} \le 1 \tag{3g}$$

$$\sum_{l \in L} yf_{ij,l}^{(C)} \cdot d_l \le D_{max}^{(C)} \quad \forall (i,j) \in N, C \tag{3h}$$

$$\sum_{l \in L} af_{ij,l}^{(qos3)} \cdot d_l \le D_{max}^{(qos3)} \quad \forall (i,j) \in N \tag{3i}$$

$$0 \le f_{ij,l}^{(C)}, a_{ij,l}^{(qos3)} \le 1 \quad \forall l \in L, \forall (i,j) \in N, \forall C \tag{3j}$$

$$yf_{ij,l}^{(C)}, ya_{ij,l}^{(qos3)} \in 0, 1 \quad \forall l \in L, \forall (i,j) \in N, \forall C \tag{3k}$$

$$0 \le z \le 1 \tag{3l}$$

In the above formulation, z denotes MLU, $f_{ij,l}^{(C)}$ is a fraction of traffic demand $t_{ij}^{(C)}$ that traverses a link l, while $a_{ij,l}^{(qos3)}$ is the fraction of $t_{ij}^{(qos3)}$ routed over link l in case when some link of the primary paths fails. If we adopt notation $L^-(k)$ for incoming links of node k, and $L^+(k)$ for outgoing links of k, (3b) defines flow conservation constraints for all primary paths, and (3c) defines flow conservation constraints for alternative paths that are assigned to QoS3 demands. We adopt minimization of MLU for the main optimization objective. The addend in (3a) is used only to prevent routing loops. Its impact is diminished by ε parameter, which has a very low value. The link capacity constraint is given with (3d). The binary variables $yf_{ij,l}^{(C)}$ indicate whether a traffic demand $t_{ij}^{(C)}$ traverses a link l. Similarly, in (3f) we introduce the variables $ya_{ij,l}^{(qos3)}$ to indicate if link l belongs to the alternative paths that should carry $t_{ij}^{(qos3)}$ in case of a link failure. The robustness to a link failure for the QoS3 traffic class is ensured by defining link-disjointness constraint for the primary and alternative paths. Since we model each network link as two contrary directed links (l and l^-), the complete link-disjointness is guaranteed only if (3g) is satisfied. Delay bounds are guaranteed with (3h) and (3i).

In order to avoid unnecessary network reconfigurations under low traffic loads, SDN-OPT uses a simple online routing algorithm that operates as follows. First, the algorithm checks whether one of the existing routes could provide the required QoS. If not, the CSPF algorithm is used to find a new one. If both attempts fail, the QoS-constrained MCF optimization is performed. Thus, the optimization stages are triggered sequentially by the failure of the online algorithm to find a solution. Considering that SDN-OPT introduces a high computation cost when the network is heavily loaded, we analysed one more approach: SDN-MCFP, that performs MCF optimizations periodically instead of reactively.

Figures 4 and 5 show the obtained results in terms of rejected bandwidth demand for two different simulation scenarios. In the first scenario, only QoS1 and QoS2 requests are generated, whereas in the second scenario all three virtual-link classes are present. As explained in Sect. 3, by DS requests we consider QoS2 and QoS3 requests, and in the second scenario they are generated in the same proportions. From Fig. 4, which refers to the first simulation scenario, it can be seen that the performance gap between SDN-LB and SDN-MCFP is minimal when only QoS2 requests are generated. In that case, SDN-LB still rejects twice as many requests as SDN-OPT, but virtually matches the performance of SDN-MCFP. This could be attributed to the fact that there are few routes in the network that can satisfy the strict delay requirements of the QoS2 class. Thus, although SDN-OPT and SDN-MCFP have the capability to use arbitrary routes and to change them dynamically, SDN-LB does not lose much on the routing diversity. With the introduction of QoS3 traffic in the second simulation scenario, the performance gap between SDN-MCFP and SDN-LB increases, as shown in Fig. 5. This is because the computational simplification (2) increases the bandwidth over-provisioning for QoS3 flows when $q > 1$. On the other hand, SDN-OPT and SDN-MCFP always over-provision bandwidth for QoS3 traffic by a factor of 2. However, it is important to note that the resource allocation scheme used by SDN-LB achieves minimal over-provisioning when tunnels between IE pairs are link-disjoint ($q = 1$). Thus, if q=1, the performance gap between SDN-LB and SDN-MCFP/SDN-OPT actually decreases, provided that a sufficient number of link-disjoint tunnels exists. To prove this claim, we performed a set of simulations in which SDN-LB method was restricted to use only link-disjoint tunnels for QoS3 traffic. The obtained results are shown in Fig. 5, denoted as SDN-LB-D. From Fig. 5, it can be seen that SDN-LB-D achieves lower virtual-link rejection ratio than standard SDN-LB approach, despite the fact that we used the POP-level ISP topology where the number of link-disjoint paths between IE pairs is very small. In real-world networks, such paths are more numerous, and thus more resources are available for QoS3 virtual-links. Additionally, in Fig. 5 we show results for static load balancing approach, which is optimized for an hour of the maximum load, assuming that TM is known in advance (OPTstatic). These results allow us to conclude that both versions of SDN-LB significantly outperform the static TE solutions that are common in today's networks.

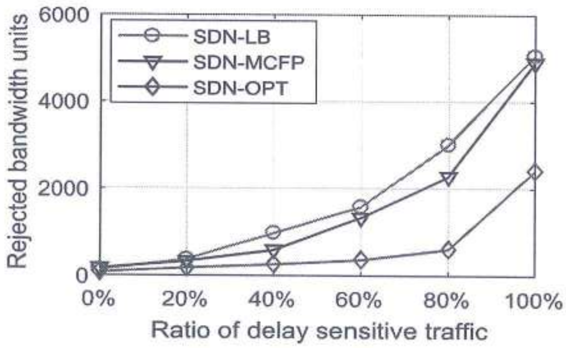


Fig. 4 Rejected bandwidth units for Scenario I.

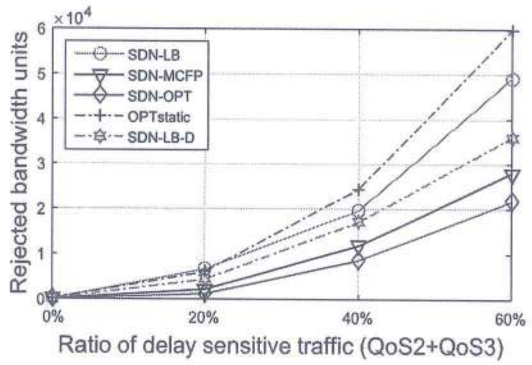


Fig. 5 Rejected bandwidth units for Scenario II.

In the second simulation scenario, we have also compared SDN-LB, SDN-MCFP, and SDN-OPT in terms of the number of network-wide reconfigurations performed, the control overhead and the average number of Flow-table entries used in the network. To evaluate the control overhead, we measured the number of Flow-mod OpenFlow messages generated by the SDN controller. The results are reported in Table 1. Due to lack of space, we denote the measured parameters with: *Reconfs*, *C-overhead* and *FT-entries*, respectively. The numbers prove non-practicality of the SDN-OPT approach due to a huge number of network-wide reconfigurations involved when the traffic load is high. This is problematic because network reconfigurations could cause transient instability and congestion due to synchronization issues. The SDN-LB and SDN-MCFP perform TE optimizations periodically, in equal intervals of 20 min, thus the number of reconfigurations is deterministic for both of them. On the other hand, the control overhead of SDN-LB is 2-3 times lower than that of SDN-MCFP. The obtained results indicate that SDN-LB is more scalable than the other two

Table 1 The scalability analysis of different TE approaches - Scenario II.

	DS %	Reconfs	C-overhead	FT-entries
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

approaches since the controller generates control messages less frequently and thus is able to control larger network partitions. It is interesting to note that the maximum Flow-table size in SDN-LB simulations was larger than in the simulations of the other two approaches. However, this is mainly because we used only POP-level ISP topology, where each node is configured with up to K static tunnels towards all the other nodes in the network, for each traffic class. The number of Flow-table entries could be reduced if unused tunnels are removed in each running cycle of the Offline-TE module.

5. Experimental Prototype

In order to validate the feasibility of the proposed virtual-link solution, the control plane prototype has been implemented on Floodlight controller [30]. When it comes to the main controller's modules from Fig. 1, we have implemented *SLA negotiation* as a REST module that receives virtual-link requests in JSON format. A request indicates the SLA type, IE pair, and duration of the virtual-link. Routing paths in the network are installed pro-actively, by the *Tunnel installation* module, which runs the K-shortest path Yen's algorithm. *Online routing* module is called reactively, upon each virtual-link request arrival. *Offline TE* is scheduled to run periodically, within a separate Java thread.

Figure 6 illustrates Flow table configuration at the ingress node, core node, and the egress node for a given IE pair. Flow table of the ingress node contains three static entries per each egress node. We assume that each egress node is used to reach a distinct range of IP addresses which is known in advance. Thus, Flow table identifies IE pair for incoming flows based on the destination IP address and directs packets either to the appropriate Group-table based on QoS requirements (offline rules for aggregated traffic demands), or directly to the out-port as a result of the online routing. In experiments, we used Type of Service field in IP header to map the flow to the corresponding QoS type.

Each Group-table has a unique identifier (at node level) and contains action buckets which specify a set of actions that can be executed on the entering packets. In the proposed system, SELECT Group-tables are used, which are primarily designed for load balancing. As shown in Fig. 6b), each bucket in a SELECT group is associated with a weight, which is used to support unequal load balancing among the Group's buckets. The bucket's share of the traffic processed by the

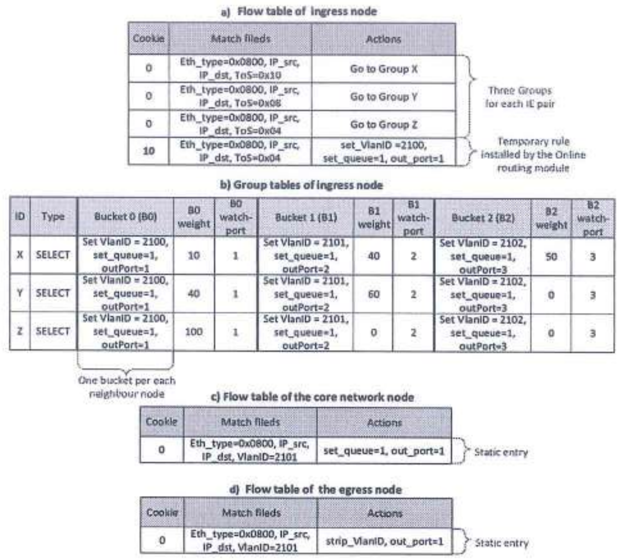


Fig. 6 Example of flow table and group table configurations for ingress node (a,b), core (c) and egress (d) node.

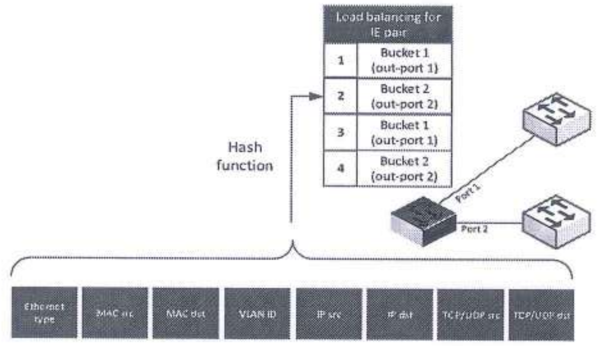


Fig. 7 Hash-based bucket selection in SELECT Group-tables.

Group is defined by the individual bucket's weight divided by the sum of the bucket weights in the Group [17]. The bucket selection algorithm, in general, depends on the switch model. In our experiments, we used OpenvSwitches [31], reprogrammed to calculate a simple hash function on L2-L4 packet header fields and assign packets to buckets according to the hash value and the bucket weights, as shown in Fig. 7. Each bucket can execute different and distinct actions. In our scenario, Group-table of ingress node contains as many buckets as the number of routes installed for the IE pair and the traffic class the Group refers to. Therefore, the primary function of each bucket is to forward flows to one of the multiple routes that are installed a priori. Before this, bucket performs two more actions: flow tagging and queue selection. Since IE pair's routes are not necessarily link-disjoint, each of them is marked with a specific label in order to avoid creation of conflicting rules in Flow-tables of core OpenFlow switches. In our experimental deployment, VLAN ID is used to differentiate flows traversing distinct IE pair routes. VLAN ID is assigned on the ingress node and removed on the egress node. In the network core, support for the OpenFlow protocol is not required and MPLS routers could be used instead. In that scenario, ingress nodes should tag packets with a MPLS label based on the active routing scheme in the network core.

Besides the static Flow-table entries, the controller dynamically installs additional Flow-table rules at ingress nodes when the current network configuration cannot meet QoS requirements of the received virtual-link requests (online routing). These rules have a higher priority than the static ones, match only IP address range indicated by the request, tag the matching packets with the appropriate VLAN IDs (or MPLS labels) and forward them to the given outport directly. As can be seen from the example in Fig. 6a), the controller assigns a different (non-default) cookie value [17] to these flows. This simplifies the flow removal upon periodical execution of the Offline-TE functions.

In order to prevent loss of control traffic due to high traffic load, two queues are configured on each network interface: i) queue 0 - used for control traffic, and ii) queue 1 - which accommodates the generated traffic load, with maximum service rate limited to 90% of the link capacity.

To validate the functionality of the controller prototype, we have performed emulations in Mininet [32]. The experimental scenario is similar to the one explained in Sect. 4, with the following adjustments done:

- Link capacity is set to 100 Mbps.
- Traffic matrix for an hour of the maximum load is used.
- The bandwidth requirement of each virtual-link request is set to 1 Mbps. If the request is accepted by the controller, traffic load for the virtual-link is generated in the form of five 200 Kbps UDP flows.
- Link-disjoint routes are used for each IE pair
- Admission control policy prevents virtual-links to bring link-utilization over 90%. Note that this still may happen because the controller is not aware of the accuracy of traffic splitting over multiple routes.

We have conducted several experiments with different proportions of DS virtual-link requests (0%, 25%, %50 and %75). QoS2 and QoS3 requests, which belong to the DS category, are generated in equal amounts. For the sake of scalability-performance trade-off analysis, each of these experiments has been run using three different period values for *Offline TE* executions (T=5, 20, and 30 min). The obtained results in terms of average and maximum packet loss rate (denoted as APLR and MPLR respectively) are shown in Table 2. These results are important because in the simulation analysis from Sect. 4 we have assumed that ingress switches are able to precisely realize load-balancing according to the decisions made by the *Offline TE* module. The same assumption is adopted in the *Online routing* module of the SDN controller. On the other side, the performance of the bucket selection algorithm is not deterministic but affected by many factors, such as flow granularity and hash-function used. Therefore, the QoS cannot be guaranteed without leaving some spare capacity on links, in order to compensate for the mentioned limitations of flow-level load balancing. This makes the feasibility of the proposed approach arguable. However, our experimental results alleviate concerns in that regard. Namely, although aggregated traffic demand of a singe virtual-link is generated with only 5 traffic flows, MPLR did not exceed 1.4% for any of the flows, while APLR is very low in each of the experimental scenarios. These results are achieved with the admission control policy that tends to keep link utilization below 90%. In practice, even more strict policies are considered reasonable due to queuing delay implications.

Table 2 Average (APLR) and maximum (MPLR) packet loss rate.

DS %	APLR (T=5)	APLR (T=20)	APLR (T=30)	MPLR (T=5)	MPLR (T=20)	MPLR (T=30)
0%	0.006%	0.008%	0.008%	0.38%	0.88%	1.4%
25%	0.011%	0.011%	0.009%	0.49%	0.36%	0.38%
50%	0.012%	0.01%	0.006%	0.64%	0.51%	0.23%
75%	0.094%	0.008%	0.007%	0.36%	0.4%	0.31%

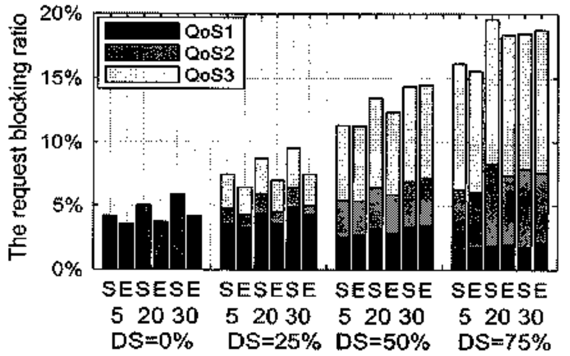


Fig. 8 Comparisson of simulation (S) and Mininet emulation (E) results in terms of the request blocking ratio in function of periodicity of offline TE optimizations (T=5min, T=20min and T=30min).

Figure 8 compares the request blocking ratio measured in Mininet experiments (denoted with "E") and our simulator (denoted with "S"). The results are shown in function of T and percentage of DS traffic. The presented results allow to conclude that the simulations imitate the controller's behaviour relatively well. Differences in performance exist, but are expected if the following facts are taken into account:

- In the experiments, virtual-link requests are generated by hosts attached to ingress network nodes. Thus, when a host requests a virtual-link, it communicates with the controller, waits for the controller's approval, and then initialize the Iperf client process [33] to generate traffic. As this takes some time, the inter-arrival times of traffic flows in the Mininet experiment differ from those in simulations. Due to a large number of Iperf session involved, the number of active threads in experiments exceeds 20000.
- Our controller extends the validation time of each virtual-link for 2s. This is done in order to avoid situations in which controller considers virtual-link expired while its traffic is still in the network.

- The controller performs checking for the expired virtual-links periodically (every second).

6. Conclusions

The contribution of this paper is to detail the scalable and effective use of SDN for virtual-link provisioning in ISP networks, especially when SDN devices are deployed only at the provider's edge. Within the scope of the virtual-link provisioning problem, we address traffic engineering, QoS and failure recovery issues. The computational scalability of the proposed solution reflects in the specific SDN controller design that adopts a function split between on-demand routing (i.e. resource allocation for virtual-links) and traffic engineering. Further, in order to decrease the controller-switch interaction, the TE is restricted to load-balancing optimization over static set of pre-configured traffic tunnels, while online routing logic is designed to exploit the existing routing rules as much as possible. The results of our simulation study have shown that the proposed solution efficiently compensates the limited routing flexibility in the network core, and reduces the control overhead up to 3 times compared with more optimized approaches for fully SDN-enabled networks. In that sense, the proposed solution alleviates scalability concerns for adoption of SDN in ISP networks, and has potential to reduce the operating costs for ISPs. The proposed system has been implemented and tested in Mininet.

References

[1] S. Tomovic and I. Radusinovic, "Traffic engineering approach to virtual-link provisioning in software-defined isp networks," 2017 25th Telecommunication Forum (TELFOR), pp.1–4, Nov. 2017.
[2] C.Y. Hong, S. Kandula, R. Mahajan, M. Zhang, V. Gill, M. Nanduri, and R. Wattenhofer, "Achieving high utilization with software-driven wan," Proc. SIGCOMM'13, pp.15–26, 2013.
[3] X. Jin, Y. Li, D. Wei, S. Li, J. Gao, L. Xu, G. Li, W. Xu, and J. Rexford, "Optimizing bulk transfers with software-defined optical wan," Proc. 2016 ACM SIGCOMM Conference, pp.87–100, 2016.
[4] A. Mendiola, J. Astorga, E. Jacob, and M. Higuero, "A survey on the contributions of software-defined networking to traffic engineering," IEEE Commun. Surveys Tuts., vol.19, no.2, pp.918–953, Secondquarter 2017.
[5] J.L. Martins and N. Campos, "Short-sighted routing, or when less is more," IEEE Commun. Mag., vol.54, no.10, pp.82–88, Oct. 2016.
[6] S. Paris, G.S. Paschos, and J. Leguay, "Dynamic control for failure recovery and flow reconfiguration in SDN," 12th Int. Conf. on the Design of Reliable Communication Networks, pp.152–159, March 2016.
[7] A. Tootoonchian et al., "On controller performance in software-defined networks," Proc. Hot-ICE'12 conference, pp.10–10, Berkeley, CA, USA, USENIX Association, 2012.
[8] "Broadcom trident ii series," Accessed: 2018-27-02.
[9] K. Poularakis, G. Iosifidis, G. Smaragdakis, and L. Tassiulas, "One step at a time: Optimizing SDN upgrades in ISP networks," IEEE INFOCOM 2017 - IEEE Conference on Computer Communications, pp.1–9, May 2017.
[10] S. Das, Y. Yiakoumis, G. Parulkar, N. McKeown, P. Singh, D. Getachew, and P.D. Desai, "Application-aware aggregation and traffic engineering in a converged packet-circuit network," 2011 Optical Fiber Communication Conference and Exposition and the National Fiber Optic Engineers Conference, pp.1–3, March 2011.
[11] J. Hu, C. Lin, X. Li, and J. Huang, "Scalability of control planes for software defined networks: Modeling and evaluation," IEEE 22nd International Symposium of Quality of Service (IWQoS), pp.147–152, May 2014.
[12] T. Koponen et al., "Onix: A distributed control platform for large-scale production networks," 9th USENIX Conference on Operating Systems Design and Implementation, pp.351–364, 2010.
[13] S.H. Yeganeh, A. Tootoonchian, and Y. Ganjali, "On scalability of software-defined networking," IEEE Commun. Mag., vol.51, no.2, pp.136–141, Feb. 2013.
[14] M. Yu, J. Rexford, M.J. Freedman, and J. Wang, "Scalable flow-based networking with difane," SIGCOMM Comput. Comm. Rev., vol.40, no.4, pp.351–362, Aug. 2010.
[15] A.R. Curtis, J.C. Mogul, J. Tourrilhes, P. Yalagandula, P. Sharma, and S. Banerjee, "DevoFlow: Scaling flow management for high-performance networks," SIGCOMM Comput. Commun. Rev., vol.41, no.4, pp.254–265, Aug. 2011.
[16] M. Al-Fares et al., "Hedera: Dynamic flow scheduling for data center networks," Proc. NSDI'10, p.19, Berkeley, CA, USA, USENIX Association, 2010.
[17] "Open networking foundation - openflow v1.4 specification," https://www.opennetworking.org (Accessed: 2018-27-02).
[18] S. Jain, M. Zhu, J. Zolla, U. Hölzle, S. Stuart, A. Vahdat, A. Kumar, S. Mandal, J. Ong, L. Poutievski, A. Singh, S. Venkata, J. Wanderer, and J. Zhou, "B4: Experience with a globally-deployed software defined wan," SIGCOMM Comput. Commun. Rev., vol.43, no.4, pp.3–14, Aug. 2013.
[19] J.M. Wang, Y. Wang, X. Dai, and B. Bensaou, "SDN-based multi-class QoS-guaranteed inter-data center traffic management," 2014 IEEE 3rd International Conference on Cloud Networking (CloudNet), pp.401–406, Oct. 2014.
[20] S. Tomovic and I. Radusinovic, "Fast and efficient bandwidth-delay constrained routing algorithm for SDN networks," 2016 IEEE NetSoft Conference and Workshops (NetSoft), pp.303–311, June 2016.
[21] S. Tomovic and I. Radusinovic, "A new traffic engineering approach for QoS provisioning and failure recovery in SDN-based ISP networks," 23rd Int. Scientific-Professional Conf. on IT, pp.1–4, Feb. 2018.
[22] J.W. Guck, M. Reisslein, and W. Kellerer, "Function split between delay-constrained routing and resource allocation for centrally managed QoS in industrial networks," IEEE Trans. Ind. Informat., vol.12, no.6, pp.2050–2061, Dec. 2016.
[23] P. Thorat, R. Challa, S.M. Raza, D.S. Kim, and H. Choo, "Proactive failure recovery scheme for data traffic in software defined networks," 2016 IEEE NetSoft Conference and Workshops (NetSoft), pp.219–225, June 2016.
[24] H.H. Liu, S. Kandula, R. Mahajan, M. Zhang, and D. Gelernter, "Traffic engineering with forward fault correction," SIGCOMM Comput. Comm. Rev., vol.44, no.4, pp.527–538, Aug. 2014.
[25] C. Fraleigh, F. Tobagi, and C. Diot, "Provisioning ip backbone networks to support latency sensitive traffic," IEEE INFOCOM 2003, vol.1, pp.375–385, March 2003.
[26] "Cplex optimization studio," http://www-03.ibm.com/software/
[27] A. Gunnar, M. Johansson, and T. Telkamp, "Traffic matrix estimation on a large ip backbone: A comparison on real data," Proc. 4th ACM SIGCOMM Conference on Internet Measurement, pp.149–160, 2004.
[28] K. Thompson, G.J. Miller, and R. Wilder, "Wide-area Internet traffic patterns and characteristics," IEEE Netw., vol.11, no.6, pp.10–23, Nov. 1997.
[29] S. Even, A. Itai, and A. Shamir, "On the complexity of time table and multi-commodity flow problems," 16th Annual Symposium on Foundations of Computer Science, pp.184–193, Oct. 1975.
[30] "Floodlight openflow controller," http://www.projectfloodlight.org/ (Accessed: 2018-27-02).
[31] B. Pfaff et al., "The design and implementation of open vswitch,"

NSDI'15 conference, pp.117–130, CA, USA, 2015.
[32] "Mininet emulator," http://mininet.org/
[33] "Iperf3 traffic generator," https://iperf.fr

Slavica Tomović received MSc degree in Electronics, Telecommunications and Computer Science from Faculty of Electrical Engineering in Podgorica (2015), University of Montenegro. Currently, She is Ph.D. student at the Faculty of Electrical Engineering, University of Montenegro. Her main research interests are in the areas of software-defined networking, quality of service (QoS) management and architectures and Internet of Things.

Igor Radusinović received the MSc (1997) and Ph.D. (2003) degree by the University of Belgrade, Serbia. He is a Full professor in Telecommunications networks and Switching systems at the University of Montenegro. He has published as an author or co-author more than 120 papers in international and national scientific journals and conferences. His main research interests are in the areas of software-defined networking, packet switching and congestion control mechanisms.

Received: 2 March 2018 | Revised: 8 August 2018 | Accepted: 12 September 2018
DOI: 10.1002/nem.2051

SPECIAL ISSUE PAPER

WILEY

Intent-based service management for heterogeneous software-defined infrastructure domains

Gianluca Davoli[1] | Walter Cerroni[1] | Slavica Tomovic[2] | Chiara Buratti[1] | Chiara Contoli[3] | Franco Callegati[4]

[1]DEI, University of Bologna, Bologna, Italy
[2]Faculty of Electrical Engineering, University of Montenegro, Podgorica, Montenegro
[3]CIRI ICT, University of Bologna, Bologna, Italy
[4]DISI, University of Bologna, Bologna, Italy

Correspondence
Walter Cerroni, DEI, University of Bologna, via Cesare Pavese 50, 47522 Cesena, FC, Italy.
Email: walter.cerroni@unibo.it

Funding information
Italian Ministry of Education, University and Research (MIUR), PRIN 2015 "GAUChO" project, Grant/Award Number: 2015YPXH4W_004; European Cooperation in Science and Technology (COST) Action CA15104-IRACON; European Commission, Horizon 2020 "EuroCPS" project, Grant/Award Number: 644090

Summary

One of the main challenges in delivering end-to-end service chains across multiple software-defined networking (SDN) and network function virtualization (NFV) domains is to achieve unified management and orchestration functions. A very critical aspect is the definition of an open, vendor-agnostic, and interoperable northbound interface (NBI) that should be as abstract as possible and decoupled from domain-specific data and control plane technologies. In this paper, we propose a reference architecture and an intent-based NBI for end-to-end service management across multiple technological domains. The general approach is tested in a heterogeneous OpenFlow/Internet-of-Things (IoT) SDN test bed, where the proposed solution is applied to a rather complex service provisioning scenario spanning three different technological domains: an IoT infrastructure deployment, a cloud-based data collection, processing, and publishing platform, and a transport domain over a geographic network interconnecting the IoT domain and the data center hosting the cloud services.

1 | INTRODUCTION

Service provisioning in today's communication infrastructures is being revolutionized by the unprecedented central role of software-based networking solutions, following the recent innovations brought about by cloud computing and resource virtualization.[1,2] In particular, the network function virtualization (NFV) paradigm fosters flexible and cost-effective service provisioning by deploying network functions as pieces of software running on vendor-independent hardware platforms, bringing the benefits of cloud computing to network infrastructure management.[3] At the same time, software-defined networking (SDN) decouples software-based network control and management planes from the hardware-based forwarding plane, turning traditional vendor locked-in infrastructures into communication platforms that are fully programmable via a standardized, open, southbound interface (SBI).[4]

In this framework, the term *service function chaining (SFC)* is used to describe the deployment of composite services that are obtained from a concatenation, ie, a *chain*, of one or more basic services typically provided by a single network function implemented in some form of virtualized environment (eg, virtual machine [VM] and container). The SFC* is

*In this paper, the SFC acronym will be used to refer to both service function *chaining* and service function *chain*, depending on the context.

fundamentally the series of service functions that a packet or a traffic flow must traverse from its source to its destination. Thanks to the capabilities offered by SDN and NFV, the SFC can be dynamically controlled and modified over a relatively small time scale, increasing the service provisioning flexibility while significantly reducing the management burden compared with traditional network architectures.

Within a single technological and administrative domain, such as a single data center, the SFC-related operations, ie, composition, maintenance, and modification, can be successfully achieved with the help of the native domain management system, as recently demonstrated by specific proof-of-concept implementations.[5] However, end-to-end services and the related SFCs must very often be provided to customers across different network administrative and/or technological domains. Guaranteeing specific functionality and quality of service (QoS) has always been a challenging task in multidomain environments.[6-8] One of the most critical issues to achieve unified management and orchestration of end-to-end services across multiple domains is the definition of an open, vendor-agnostic, and interoperable northbound interface (NBI), through which applications are allowed to control the underlying heterogeneous NFV and SDN infrastructures and take advantage of dynamic SFC. Although a standard NBI definition is still under discussion, a commonly accepted approach is to adopt a so-called *intent-based* interface that allows to declare service outcomes and high-level operational goals rather than specify detailed networking mechanisms.[9]

In this paper, we present a reference architecture and define a related intent-based NBI for end-to-end service management and orchestration across multiple technological domains, extending our preliminary work on heterogeneous OpenFlow/IoT SDN domains.[10] In particular, we consider the use case of a rather complex service provisioning scenario spanning different technological domains:

1. an Internet-of-Things (IoT) infrastructure deployment, representing the first technological domain;
2. a cloud-based data collection, processing, and publishing platform, representing the second technological domain;
3. a transport domain over a geographic network interconnecting the IoT domain and the data center hosting the cloud services, enhanced with proper SDN control capabilities to implement dynamic SFC.

The data "produced" in the IoT domain are "consumed" inside a cloud domain where different data streams traverse different SFCs. The goal is to dynamically differentiate the QoS of different data streams representing different end-to-end services. This is achieved by means of an effective integration of computing and networking resource management in a cloud infrastructure.[5] The infrastructure is fully automated both in the cloud deployment of a given set of network functions and in the capability of reacting to changes in the overall network conditions, safeguarding the service level agreement. The conditions of the network infrastructure underlying the set of functions and implementing the required SFC can be monitored and modified to pursue the QoS objectives for the various active end-to-end services.

The remainder of the paper is organized as follows. In Section 2, we present existing work related to different aspects of our approach. Then we propose our reference architecture and define the intent-based NBI in Sections 3 and 4, respectively. We provide specific examples and technical details related to IoT, cloud, and transport domains in Sections 5 to 7, respectively. We report the experimental validation in Section 8 and finally conclude the paper in Section 9.

2 | RELATED WORK

2.1 | Intent-based networking

The concept of *intent-based networking* has recently gained increasing attention from both industry and academia. One of the earliest definitions of an intent-based NBI came from the industry[11] and included the following features: invariance, portability, composability, scalability, and context awareness. Then the first step towards standardization in the SDN context was made by the Open Networking Foundation (ONF),[12] which defined an intent-based NBI as nonprescriptive, provider independent, and declarative. ONF also specified that a set of mechanism, named *mappings*, are required to translate intent NBI requests into forms that lower-level entities can understand, thus making consumer and provider systems separately implemented but able to communicate in terms that are "natural" to each. As discussed in Section 4, we follow the ONF approach in our definition of intent-based NBI, which must allow an abstract yet flexible definition of a service chain, without knowledge of technology-specific details.

Other examples of adoption of an intent-based approach include abstraction for virtualized network management in a multitenant data center environment[13]; high-level specification of network slicing requirements and automated configuration in an SDN infrastructure[14]; definition of a service-oriented architecture for service composition based on microservices[15]; and scalable label-based abstraction of policy requirements for large cloud computing environments.[16] However, each of those solutions focuses on a single specific domain, whereas our approach takes advantage of the powerful abstraction level offered by an intent-based NBI to manage end-to-end services across multiple technological domains.

The use of intent-based networking in multidomain scenarios is still an open research issue. An intent-based mobile backhauling interface for 5G networks has been proposed and prototyped in Subramanya et al.[17] The specific characteristics of 5G mobile networks require the integrated management of multiple technological domains in the radio access and backhaul segments, including Wi-Fi access points and OpenFlow switches. The platform design accounts for several service scenarios, including mobility management, uplink/downlink decoupling, and fine-grained packet processing. However, the intent-based interface defined in Subramanya et al[17] is not completely decoupled from the underlying infrastructures, as it requires knowledge of low-level details such as switch IDs, port numbers, and medium access control (MAC) addresses, making it difficult to extend it to heterogeneous technological domains, such as IoT. More recently, the problem of multidomain intent decomposition into local intent graphs for each domain has been addressed in Arezoumand et al.[18] The latter work is complementary to our experimental study reported in this paper, as we do not focus on the intent decomposition problem, but provide a generalized intent-based definition of service function chains to be applied to each domain involved.

The most widely used SDN control platforms offer some sort of intent-based NBIs. OpenDaylight and OpenStack Neutron offer the Group-based Policy (GBP) tool, which allows to specify communication policies (or contracts) between groups of end points (ie, VMs, containers, and ports).[19] For instance, a typical GBP specification could be such as "allow web traffic to web server end point group," which will automatically reconfigure the firewall security groups in order to allow access to the requested service. However, differently from our approach, GBP was not intended for specifying complex SFCs without the support of ad hoc configuration tools specific to the OpenStack platform. OpenDaylight also provides Network Intent Composition (NIC), which basically allows to specify connectivity requirements between end points with redirection, such as "connect endpoint A to endpoint B redirecting through C."[20] Similarly, Open Network Operating System (ONOS) offers the Intent Framework, which is also a way to express connectivity requirements, specifying (even multiple) end points or connect points, eg, "connect endpoint A to endpoints B and C."[21] However, differently from our approach, both solutions still require knowledge of low-level details such as switch IDs, port numbers, virtual local area network (VLAN) IDs, and MAC addresses.

Finally, it is worth to mention that intent-based networking is often perceived as similar to the concept of high-level policy-based network management. This is particularly true in the case of policy refinement techniques, aimed at deriving (or refining) lower-level policies from higher-level, goal-oriented specifications.[22] In our approach, we share the point of view recently expressed by Internet Engineering Task Force (IETF) concerning the conceptual differences between "intent" and "policy."[9] Both terms refer to high-level abstractions for managing networks without delving into device-specific details. However, a policy typically involves a set of rules used to define what to do under what circumstances (events, conditions, and actions), but it does not necessarily specify a desired outcome. Differently, an intent is used to define network-wide outcomes and high-level operational goals, without the need to enumerate specific events, conditions, and actions. In this sense, policy refinement can be considered equivalent to the ONF mappings needed to translate intents into lower-level policies.

2.2 | Standards for SFC

The implementation of a given SFC that spans several network domains with nonhomogeneous forwarding technologies is very challenging and is usually solved by means of some form of network overlay (eg, tunneling). This problem was addressed by the IETF, which suggests that the service-specific overlay can be obtained by applying suitable packet encapsulation.[23] One option being considered by IETF is the so-called *network service header* (NSH) standard,[24] which intends to provide a flexible, dynamic, and transport-independent SFC solution for the data plane. The NSH standard focuses on data plane aspects only, and very little has been said about a possible SFC control plane solution. To the best of our knowledge, the only document that attempts to do so is an IETF draft that, at the time of writing, has already expired.[25] Therefore, here we adopt a possible implementation of a NSH-aware control plane inspired by the concepts discussed in the IETF draft.[26] It is based on the use of SDN-like technology inside NSH nodes and on the adoption of the OpenFlow protocol for the communication between the SFC control plane and the NSH enabled nodes. This allows for a seamless integration of the related NBI with the NBI adopted in the IoT and cloud domains.

2.3 | Software-defined IoT

IoT facilitates billions of devices to be enabled with network connectivity to collect and exchange information for providing different services. IoT should allow connected devices to be controlled and accessed remotely, in an efficient manner.

However, traditional network infrastructures in many cases cannot satisfy IoT requirements, and new approaches, based on the application of the SDN concept, have been recently proposed.[27] Different works discuss advantages of applying SDN to IoT.[28-30] From the *network management* viewpoint, SDN may enable traffic control and load balancing. As an example, in Baddeley et al,[31] it is shown that the SDN controller may steer traffic in noninterfered area, improving network performance. When it comes to *resource utilization*, the SDN approach allows viewing nodes as resource providers and to efficiently map the users' requests into the proper resources. Multiple applications could run concurrently on different WSNs by optimizing their resource use according to availability and other cost metrics.[32] Regarding *energy management*, a solution focused on both centralized device and topology management was proposed,[33] allowing to switch on/off devices to reduce energy consumption compared with a decentralized solution.

The papers mentioned above report theoretical analysis or discussion, while few practical deployments are available. SDN-WISE is a stateful SDN solution pursuing the reduction of the amount of information exchanged between sensor nodes and the SDN controller.[30] However, no resource management or QoS-based network management is considered. An extension of the 6LoWPAN protocols stack was proposed to implement SDN,[31] demonstrating performance improvement with respect to the Routing Protocol for Low Power and Lossy Networks (RPL)–based decentralized solution. Results are obtained via a platform emulating TI CC2420 devices and not using real devices as we do in our paper.

In contrast, in this paper, we propose an SDN-based IoT architecture and validate it with an experimental test bed integrated with our end-to-end service management platform. The architecture allows to jointly implement resource management and network management, and to characterize performance at different planes (data, control, and management plane), as never done in previous literature to the best of our knowledge. The proposed SDN IoT solution builds on existing work,[30,34] where however resource management was not allowed and where the multidomain integration was not present.

3 | REFERENCE NETWORK ARCHITECTURE

The reference multidomain SDN/NFV architecture considered in this paper is shown in Figure 1. Although our approach to intent-based service management can be generalized to any SDN/NFV technology domain, the domains included in Figure 1 are those involved in the use case considered in this paper: Data collected from sensor and actuator devices of a software-defined IoT domain are dispatched across the transport network to reach a set of suitable consumers, implemented by means of virtualized network functions (VNFs) and deployed within a cloud computing domain.

Considering the purpose of our study and the nature of the orchestration features we are interested in, our reference architecture is inspired by the ETSI NFV specifications, with particular reference to the Management and Orchestration

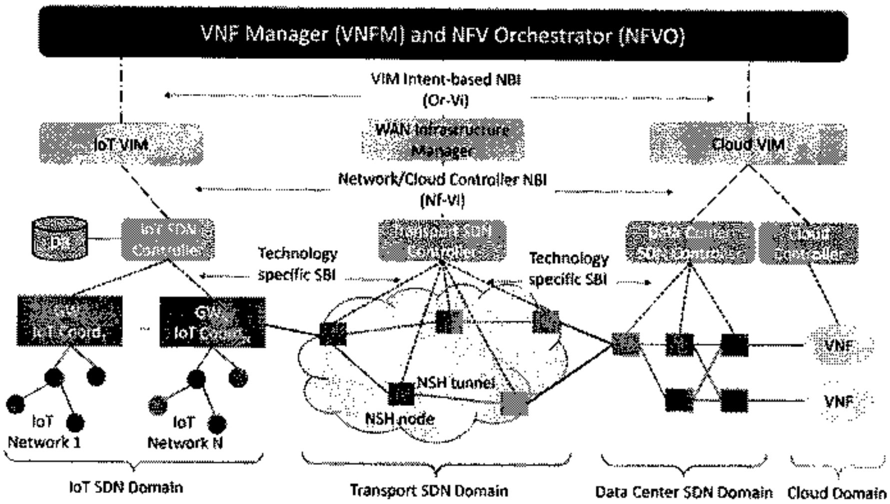


FIGURE 1 Reference multidomain SDN/NFV architecture. Three different technological domains are displayed here, including an IoT domain, a data center and cloud domain, and a geographical transport network domain. GW, gateway; IoT, Internet-of-Things; NFV, network function virtualization; NSH, network service header; SBI, southbound interface; SDN, software-defined networking; VIM, Virtualized Infrastructure Manager; VNF, virtualized network function; WAN, wide area network

(MANO) framework,[35] although our approach is focused on an end-to-end service perspective. The rationale behind this choice is that, on the one hand, the proposed architecture has the advantage to be consistent with the most relevant NFV standard initiative to date and, on the other hand, the architecture itself can be seamlessly extended to include any further SDN/NFV domain and technology as part of the underlying virtualized infrastructure.

Each SDN/NFV domain in Figure 1 consists of a technology-specific infrastructure, including the following:

- data plane components, such as IoT nodes and gateways (GWs), NSH network nodes, SDN switches, VMs running in cloud computing nodes, physical and virtual interconnecting links; these components provide the network, compute, and storage resources to be orchestrated;
- control plane components, such as SDN and cloud controllers with related data stores and interfaces; these components are responsible for proper VNF deployment and traffic steering across VNFs and domains;
- management plane components, such as Virtualized Infrastructure Managers (VIMs) and WAN Infrastructure Manager (WAN-IM), for managing resources in the technological domains; based on the available implementations, some of these components could be in charge of multiple domains,[36] as in the case of the cloud VIM in Figure 1.

The overarching VNF Manager (VNFM) and NFV Orchestrator (NFVO) components are responsible for programming the underlying VIMs/WAN-IM and infrastructure controllers in order to implement and maintain the required service chains in a consistent and effective way, for both intradomain and interdomain scenarios. While technology-specific and domain-specific NBI and SBI are used inside each domain to efficiently control and manage the relevant components, the design of the overarching VNFM and NFVO should be as technology-agnostic as possible, so that a service chain to be deployed can be specified by a customer using a high-level, intent-based description of the service itself. This would also allow the proposed architecture to be more general and capable of being extended to different SDN technologies and domains.

In order to achieve such generality in the high-level management and orchestration components, we argue that *the act of decoupling service abstractions from the underlying technology-specific resources should be performed mainly by the infrastructure managers (VIMs and WAN-IM).* Therefore, we extend the concept of interactions on the basis of intents to the NBI offered by the VIMs/WAN-IM, which should be defined as an open and abstract interface, independent of the specific technology used in the underlying domains. This approach could also allow different administrative domains to expose only service abstractions without disclosing sensitive details related to the underlying infrastructures.

4 | VIM NORTHBOUND INTERFACE

In general, the definition of an open, vendor-agnostic, and interoperable interface will foster improved and standardized procedures for service specification to the underlying multidomain NFV and SDN platforms. In particular, the powerful abstraction level offered by an intent-based NBI allows to specify service outcomes and high-level operational goals rather than mechanisms, by taking advantage of formalism close to the customer's natural language.[9,12] Therefore, in our architecture, we assume that some kind of intent-based interface is offered to the customer by the overarching VNFM and NFVO components.

When a given service request is received, the high-level management and orchestration functions must convert that request into a set of suitable service chains and pass them to the relevant VIMs[†] in charge of the underlying infrastructures and domains involved in the service composition. Then each VIM must coordinate the respective controllers in order to

- verify availability and location in the cloud infrastructure of the VNFs required to compose the specified service, instantiating new ones if needed;
- program traffic steering rules in the network infrastructure to deploy a suitable network forwarding path.

The NBI exposed by the VIMs should allow an abstract yet flexible definition of the service chain, without knowledge of the technology-specific details such as devices, ports, and addresses. This means that a request sent to the VIMs should specify not only the sequence but also the nature of the different VNFs to be traversed, which is strictly related to the service component they implement, as well as other peculiar characteristics of the service itself, such as QoS metrics and

[†]For the sake of simplicity, in the remainder of the paper, we use the term VIM to refer also to the WAN-IM in charge of managing the transport domain, considering that the underlying NSH-based network overlay exposes a sort of virtualized infrastructure to the orchestrator.

thresholds. In particular, the NBI should allow an abstract representation of the QoS features for the requested service and the topological characteristics of each VNF to be applied in the service chain.

A possible definition of the VIM NBI is presented here, considering the following service and function abstractions.

- A *QoS feature* is defined in qualitative terms relevant to the specified service, eg, guaranteed bit rate or limited delay.
- A *QoS threshold* can be specified for the metric of interest, eg, a minimum bit rate or a maximum delay value.
- A VNF can be *terminating* or *forwarding* a given traffic flow. For instance, a deep packet inspection (DPI) function usually terminates a mirrored copy of a given flow, whereas a network address translator (NAT) forwards incoming flows.
- A forwarding VNF can be *port-symmetric* or *port-asymmetric*, depending on whether or not it can be traversed by a given traffic flow regardless of which port is used as input or output. For instance, a NAT is port-asymmetric, because it must receive inbound and outbound traffic from a port connected to a public and private network, respectively. A basic IP routing function can be considered port-symmetric, as it forwards packets on the basis of the destination address.
- A VNF can be *path-symmetric* or *path-asymmetric*, depending on whether or not it must be traversed by a given flow in both upstream and downstream directions. For instance, an intrusion detection system (IDS) is typically path-symmetric, because it needs to analyze packets in both directions of a given flow. A traffic shaper can be considered path-asymmetric if it must limit only outbound traffic.

In order to implement the aforementioned abstractions, we define an SFC template adopting the well-known JSON format. This template should be coupled with other deployment templates defined by the ETSI MANO specifications in order to complete service provisioning. However, in this work, we focus only on the SFC aspects of the NBI. A service chain is therefore defined as follows:

```
{
  "src": "node_value",
  "dst": "node_value",
  "qos": "qos_type",
  "qos-thr": "qos_value",
  "vnfList": [vnf],
  "dupList": [dup]
}
```

where `src` and `dst` represent the end point nodes of the service chain, either global or limited to a given VIM domain; `node_value` is a text string that contains a high-level unique identifier of a node known to both orchestrator and VIMs, eg, by means of some form of mapping mechanism as defined in previous work[12]; `qos` represents the QoS feature to be provided with the service chain; `qos_type` is a text string that contains a high-level unique identifier of a QoS metric known to both orchestrator and VIM; `qos-thr` represents the QoS threshold to be applied to the specified metric; `qos_value` is the actual value assigned to the threshold; `vnfList` is the ordered list of VNFs to be traversed according to the specified service; `dupList` is the list of VNFs towards which the traffic flow must be duplicated; each VNF included in `dupList` must be also included in `vnfList` to specify at which stage of the SFC the traffic must be mirrored.

Each VNF is described in terms of its topological abstractions with the following template:

```
vnf ::= {
  "name": "node_value",
  "terminal": "bool_value",
  "port_sym": "bool_value",
  "path_sym": "bool_value"
} | ε
```

where `bool_value` is a text string representing either a Boolean or a null value and the ϵ symbol indicates the possibility that `vnf` is an empty element. Considering that some network functions (eg, DPI and IDS) require traffic flows to be mirrored, the (possibly empty) list of VNFs towards which the traffic flow must be duplicated is specified with the following template:

```
dup ::= {"name": "node_value"} | ε
```

The NBI offered by VIMs can be implemented through the mechanisms of a Representational State Transfer (REST) Application Programming Interface (API) and should provide the following methods:

- define a new service chain;
- update an existing service chain;
- delete an existing service chain.

These actions are basically in line with the operations foreseen by the ETSI MANO specifications with reference to the interface between NFVO and VIM. It is worth highlighting that the NBI description given above is indeed based on the concept of intent according to IETF and ONF definitions.[9,12] QoS metric, VNFs, and service chains are specified in a high-level, goal-oriented format, without the need to enumerate specific events, conditions, and actions and without any knowledge of the technology-specific details. A non–intent-based description of a service chain, eg, using the OpenFlow expressiveness to steer traffic flows and compose the network forwarding path, would require the customer to specify low-level policies to install multiple flow rules in each forwarding device for each traffic direction, involving technology-dependent details such as IP and MAC addresses, device identifiers, and port numbers.

The NBI defined above is used, according to the architecture in Figure 1, to specify an IoT data gathering service crossing two different SDN domains and an NFV chain, as well as to specify the characteristics that the transport service from the IoT domain to the cloud domain should provide. For the use case considered here, the high-level QoS features offered by the SDN/NFV platform include "delay-sensitive" and "loss-sensitive" services, with the possibility to specify a threshold for the relevant metric. Such quantitative QoS objectives are not as easy to obtain in the transport network, where most of the infrastructure may not be under the control of the final user; therefore, it is assumed that the QoS objectives are given in a more qualitative way, with the same syntax as above but without specifying any quantitative threshold. This means that the service expects the transport domain to do "its best" with reference to that particular QoS aspect.

Although the above intent-based NBI definition is common to all VIMs considered in our use case, the orchestrator must specify different content for each VIM depending on the specific resources to be programmed and the specific segment of the service chain to be deployed in each domain. This approach allows the definition of the NBI to be more flexible, facilitating new technological extensions and new domains integration.

5 | IOT SDN DOMAIN

The IoT SDN domain included in the architecture of Figure 1 is composed of (i) a VIM able to manage components and resources in the IoT domain; (ii) an *IoT SDN controller* (IoTC), implementing the software-defined control plane of the IoT domain; and (iii) a set of IoT networks, where different devices send the measured data via multihop paths to a coordinator node that forwards them to the final consumer. Since the different IoT networks will possibly use different technologies (eg, Zigbee, LoraWAN, and 6LowPAN), each IoT coordinator will be connected to a specific GW in charge of forwarding data outside the IoT domain.

When a service request is received from the high-level management and orchestration functions, the IoT VIM accesses the IoTC, including a database that stores information about devices of the different networks, such as the IP address of the corresponding GW, the service provided, and the related QoS feature that could be guaranteed. The VIM tries to map the incoming request with the resource knowledge available in the database, in order to select the proper IoT device to forward the request to (the details of this operation are presented in Section 5.1). According to the decision taken, the IoTC will (i) program the selected IoT network to make sure that the requested QoS would be guaranteed and (ii) forward the request to the identified GW. More details about the different components are provided in the rest of this section.

5.1 | The IoT VIM and database

The VIM is capable of handling requests containing either the particular IoT device to be queried or a high-level description of the service requested by the customer, together with some other possible specification related to the QoS (in terms of maximum latency). Let us consider the case of a customer that wishes to periodically collect temperature values in a given room and monitor them by means of a processing/publishing service called `ServP` running as a virtual function in the cloud domain. Assume that the customer is interested in having a sort of real-time monitoring of the measured temperature, thus requiring a delay-sensitive service. Then the intent-based request sent to the IoT VIM, expressed according to the JSON format specified in Section 4, could be as follows:

```
{
  "src": "ServP",
  "dst": "Temperature Room X",
  "qos": "Delay Sensitive",
  "qos-thr": "15 ms",
  "vnfList": "null",
  "dupList": "null"
}
```

In the IoT domain, following the typical IoT device query approach, `src` represents the source of the query, which is the final consumer of the data to be collected. In our example, this is the processing/publishing service in the cloud. `dst` represents the final end point of the query, which could be one or multiple IoT devices. This text string may contain (i) a unique identifier of a specific IoT device or (ii) a high-level intent-based description of the requested service. The second option is used in our example above. `qos` represents the requested QoS feature in terms of either maximum latency, expressed as data plane round-trip time (see below), or minimum loss, expressed as the probability of successfully receiving the data from that device. If needed, the user may also provide a quantitative threshold `qos-thr`. In the example above, a delay-sensitive service with a 15-millisecond threshold is requested. Finally, `vnfList` and `dupList` are not specified in the example because we assume that the orchestrator opted for VNFs located only in the cloud domain.

At this point, the VIM sends a query to the database module (located in the controller in our implementation): If the VIM is searching for a specific node, the controller notifies back to the VIM the presence or not of the node; if a service is requested instead, the controller replies with the list of nodes that can provide that service and the related QoS parameters. Finally, the VIM determines which nodes comply with the QoS requirement and sends queries to those nodes (again over the controller) by specifying their ID. The latter is possible, thanks to the information contained in the database, where an entry per IoT device is generated and each entry includes the following:

- the unique MAC address (eg, the IEEE 802.15.4 64-bit address);
- the corresponding network address (ie, the short address used in IEEE 802.15.4 at 16-bit);
- the ID of the IoT network the device belongs to;
- the service provided by the device (eg, temperature sensor and light sensor);
- the value and timestamp of the last measurement gathered from the device;
- the corresponding QoS in terms of latency: these values are computed by averaging among different measurements taken over time.

When the IoTC receives a new measurement from a device, the data are stored in the database, together with the instant in which it was received. Once a new request for the same device arrives, the VIM checks the timestamp and decides if the data should be updated or not (if not, the value is immediately returned). With reference to the QoS, it is important to underline that in case the same device could reach the IoT coordinator via different paths (eg, having different number of hops), the corresponding QoS values are stored in the database. A simplified example is reported in Table 1, where we are considering a room having two carbon monoxide sensors detecting the presence of smoke (devices 1 and 2) and a light sensor (device 3). Device 1 can reach the coordinator via three different paths, characterized by 1, 2, or 3 hops, and different resulting QoS values, namely, round-trip time RTT_i in the case of i hops. Device 2 has two possible paths, whereas device 3 has only one path. If a user asks for the level of CO in room X and wants the data in real time (delay-sensitive service), with a maximum latency (`qos-thr`) of 15 milliseconds, the VIM will select device 1 and will notify to the IoTC the topology to be used to trigger such node in order to guarantee the requested QoS feature. However, if the delay requirement is relaxed, the IoTC may decide for a longer path, possibly characterized by a lower loss probability (loss-sensitive service).[27] Once the IoTC receives the request from the VIM, it will program the IoT network according to the selected topology.

TABLE 1 Example of quality of service (QoS) values stored into the Internet-of-Things (IoT) database

Node ID	Service	RTT_1	RTT_2	RTT_3
1	Smoke Detector Room X	12 ms	24 ms	36 ms
2	Smoke Detector Room X	null	23 ms	38 ms
3	Light Room X	null	null	34 ms

5.2 | The IoT controller and network

The main functionalities of the IoT controller are gathering information from devices, maintaining a representation of the network, and establishing routing paths.

In order to achieve the decoupling of the control plane from the data plane, it is fundamental that each device can discover a path towards the coordinator. This is done during the network initialization, where the coordinator sends an *Hello* packet in broadcast; this packet is forwarded by nodes, after updating the number of hops (ie, number of hops separating the node from the coordinator). In this phase, each node selects the best next hop to be used to reach the coordinator, which is the one characterized by the lowest number of hops. Since *Hello* packets are sent in broadcast, they are also used to create neighbors tables, containing the received signal strength indicator (RSSI) received from each neighbor. These neighbors tables are periodically sent to the coordinator using the best next hop selected in the initial phase and then forwarded to the IoT controller. In this way, the controller will update the database with the current map of devices and will compute paths on the basis of the RSSI matrix (power received by each node when another is transmitting). In our implementation, we assign to each link a cost having an inverse proportionality wrt the RSSI, and then, by running Dijkstra, we select for each node the path characterized by the lowest path cost (sum of the costs of the links in the path). Moreover, to limit delays, we also impose a maximum number of hops, denoted as H; by changing H, different topologies are obtained, corresponding to possibly various performance levels, as shown in Table 1.

Requests coming from the VIM are forwarded by the IoTC to the proper IoT coordinator, together with the information about the selected path connecting the coordinator and the intended device to be set up to guarantee the requested QoS. This path is then forwarded by the coordinator to all devices belonging to the route itself (through the transmission of a packet called *Path*), in order to update the flow tables at devices. In case a device receives a packet for handling which it has no information, a *PathRequest* packet is sent through the route defined in the initial phase to the controller, that after elaborating it, will reply sending a *PathResponse*.

6 | DATA CENTER SDN AND CLOUD DOMAINS

In this section, we consider both the data center SDN domain and the cloud computing domain depicted in Figure 1, assuming that they are managed by a single VIM. The data plane topology assumed for the use case considered in this paper is shown in Figure 2. An OpenFlow-based SDN data center infrastructure is assumed to be in charge of the connectivity within the cloud domain, thus providing programmable traffic steering functionality to implement suitable SFCs. All the switches included in the topology (`s1, s2, ..., s7`) are OpenFlow-enabled devices and are governed by an SDN controller (eg, ONOS[37]), whereas the computing infrastructure is managed through a cloud platform (eg, OpenStack.[38])

Switch `s6` is an edge device that represents the ingress point to the cloud network. Incoming traffic flows, carried by means of suitable tunnels in the transport network, will be terminated here. Router `vrl` is the (virtual) edge router of the (virtual) tenant network responsible for the connectivity within the cloud domain of the requested IoT data collection service. Switches `s1` to `s5` are either physical or virtual switches used by the tenant network for VNF connectivity. Two VNFs are deployed in the cloud: `chk` performs integrity and sanity check on the collected data for improved reliability,

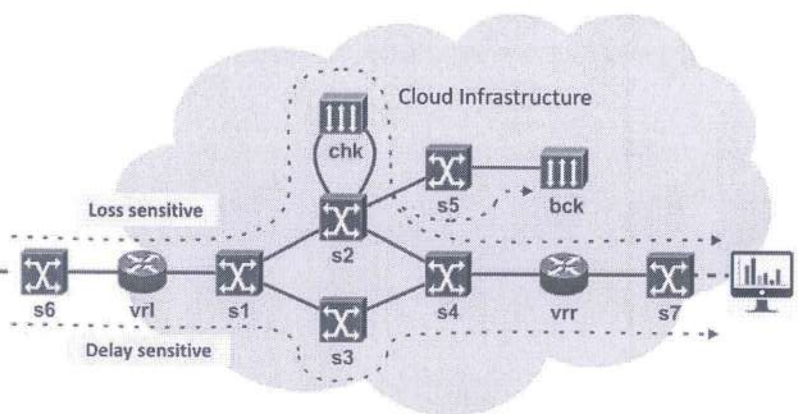


FIGURE 2 Data plane topology of the data center software-defined networking (SDN) and cloud domains considered in the use case

whereas bck is used to store backup copies of the collected data. Router vrr is the (virtual) edge router of the (possibly different) tenant responsible for the IoT data collection, processing, and publishing services. Switch s7 is a (virtual) switch in the latter tenant's network, providing layer 2 connectivity to the server ServP where collected data are processed and published.

According to the QoS features of the use case considered here, we assume that the connectivity service offers two different paths in the OpenFlow domain. One path is characterized by minimum latency, where switches have been configured with small buffers to limit queuing delay. Those switches are continuously monitored by the SDN controller to detect traffic levels that can lead to possible congestion. In addition, no VNF processing is performed along this path, which could introduce additional delay. This path is more suitable for delay-sensitive flows. The second path is dedicated to loss-sensitive traffic flows, where switches have large buffers to reduce losses, and data are processed by chk and duplicated at switch s2 in order to be stored in bck.

Therefore, depending on the QoS feature requested by the customer, the high-level management and orchestration functions can specify two different service chains. Assuming that, on the basis of the interaction between the orchestrator and the IoT VIM, incoming data will be collected from IoT network *k* and then forwarded to ServP, according to the JSON format specified in Section 4, the intent-based request to the cloud VIM NBI could be

```
{
  "src": "IoT-GW[k]",
  "dst": "ServP",
  "qos": "Delay Sensitive",
  "qos-thr": "10 ms",
  "vnfList": "null",
  "dupList": "null"
}
```

for the delay-sensitive QoS feature or

```
{
  "src": "IoT-GW[k]",
  "dst": "ServP",
  "qos": "Loss Sensitive",
  "qos-thr": "99.999
  "vnfList": [chk, bck]
  "dupList": [bck]
}

chk ::= {
  "name": "chk",
  "terminal": "false",
  "port_sym": "true",
  "path_sym": "false"
}

bck ::= {
  "name": "bck",
  "terminal": "true",
  "port_sym": "null",
  "path_sym": "false"
}
```

for the loss-sensitive QoS feature. The SDN controller must implement a data plane monitoring service to make sure that, in the former case, the minimum latency path guarantees the requested maximum delay of 10 milliseconds, whereas in the latter case, the VNFs inserted in the service chain and the more reliable path ensure the required 99.999% availability.

We developed the VIM for the data center and cloud domains as an application running on top of the ONOS platform. It is worth to note that ONOS already provides a built-in, intent-based NBI that can be used to program the SDN domain and deploy the required network forwarding paths. However, in order to specify the ONOS intents, some knowledge is required of the specific data plane technical details, while in our approach, we prefer to expose only high-level abstractions to the orchestrator. Therefore, one of the main functions of our VIM is to implement new, more general and abstract intents that can be expressed according to the NBI specification given above. Then the VIM takes advantage of the network topology features offered by the SDN/cloud controllers to discover VNF location in the cloud and relevant connectivity details, and eventually it is able to compose native ONOS intents and build more complex network forwarding paths.

The VIM can be instantiated as an ONOS service called *ChainService*, which provides the capability of dynamically handling the VNF chains through the abstract NBI defined in Section 4. To achieve extensibility and modularity, the implementation of ChainService is delegated to a module called *ChainManager*, which is in charge of executing all the required steps to translate the high-level service specifications into ONOS-native intents. The input to ChainManager can be given through either the ONOS command line interface (CLI) or a REST API. The latter is preferable because it allows remote applications to use standard protocols (eg, HTTP) to access resources and configure services. In our current implementation, the REST API provides the following service end points[‡]:

```
POST /chaining/{action}/{direction}
DELETE /chaining/flush
```

In the former end point, the `action` variable indicates the operation that the orchestrator intends to perform on a specified service chain (`add`, `update`, or `delete`), whereas in case of an update, the `direction` variable (`forth`, `back`, or `both`) defines whether the modified chain specification refers to the existing forwarding path from `src` to `dst`, the opposite way, or both directions. Parameters and identifier of the specified service chain are included (in JSON format) in the message body of the POST request method. So the basic operations of this end point are as follows:

- If the `add` action is given, this will result in defining a new service chain, based on the JSON specification included in the message body. This means that a forwarding path will be created for traffic flowing from `src` to `dst` and another one in the opposite direction. Note that the two paths are not necessarily symmetric, based on the topological abstractions defined by the NBI.
- If the `update` action is given, then the direction is taken into account and the forward path, backward path, or both paths of the specified existing service chain are changed. In fact, a user may be interested in changing only a segment of the forwarding path and only in one direction, to reduce the control plane latency and limiting the impact that a path change can have on the existing traffic flows.
- If the `delete` action is given, then both forwarding paths of the specified existing service chain are removed.

ChainService provides also the `flush` operation through another end point, thus offering the possibility of deleting in a single step the forwarding paths of all the service chains previously created.

7 | THE TRANSPORT NETWORK DOMAIN

The role of the transport domain depicted in Figure 1 is to provide interdomain connectivity between IoT and data center/cloud domains across a general geographical network, as required by the service chain to be instantiated. Although the SDN concept has been recently extended to inter-data center transport networks[39] and to flexible wide area network (WAN) interconnections,[40,41] in our study, we choose to be independent of the control capabilities offered by the transport network. As previously mentioned, an overlay approach allows to deal with heterogeneous forwarding technologies in the transport domain.

In particular, in order to keep service provisioning operations separate from and independent of the underlying transport infrastructure, we adopt the IETF SFC approach and take advantage of NSH encapsulation to properly steer traffic flows between the different domains involved in a given SFC.[24] In fact, when used in conjunction with some tunneling technoligy (eg, VXLAN), NSH can be seen as a new way to implement a network overlay enabling SFC on top of legacy transport networks. As detailed in this section, our SDN-like solution for implementing the NSH control plane enables a seamless integration of the transport infrastructure manager's NBI with the NBI adopted in the IoT and data center/cloud domains, as well as the ability to dynamically adapt traffic flow forwarding to the requirements of the SFC being deployed.

‡This is a first implementation of the API. In future versions, we will consider to modify it according to a more accurate design following the REST principles, eg, by using different HTTP methods to perform different actions and by specifying only resources in the URIs.

7.1 | The IETF service function chaining architecture

The IETF SFC architecture for a given transport domain defines the following:

- *Service function path* (SFP): a specification of the path to be followed by packets assigned to a certain SFC, ie, an abstraction of the sequence of nodes the packets will traverse;
- *SFC encapsulation* (SFC-En): a form of SFP identification that enables to follow the correct sequence of nodes in the SFC.

Moreover, the main components of the SFC architecture are as follows:

- *SFC classifiers* (SFC-Cl), which classify the incoming traffic on the basis of predefined policies, in order for the flow to be steered through the required set of network service functions; the main task for the SFC-Cl is to add the SFC-En, which is then removed by the last node in the SFP, or by a SFC-aware function that consumes the packet;
- *Service functions* (SF), which are the basic elements of a chain and are responsible for a specific treatment of received packets; they can act at different levels of the protocol stack, and they can be implemented either as virtual elements hosted by a server or as physical equipment with specialized hardware; a SF can be either SFC-aware (ie, able to act on SFC-encapsulated packets) or SFC-unaware (ie, it must receive only packets without SFC encapsulation);
- *Service function forwarders* (SFF), which are responsible for forwarding traffic to one or more connected SFs according to information carried in the SFC-En; they can also terminate the SFP;
- *SFC proxies* (SFC-Pr), which remove and insert SFC-En on behalf of SFC-unaware SFs, before and after their action, respectively.

The implementation of the architecture requires a protocol to provide SFP identification, transport-independent chaining, and packet-based network and service metadata. The NSH protocol is designed to this purpose, with the goal to be easy to implement across a range of devices, both physical and virtual, including hardware platforms. The two most important fields in the NSH header are as follows:

- *Service path identifier* (SPI): a 24-bit integer number assigned to packets by the first SFC-Cl in the SFP; all nodes taking part in that SFP must use the same SPI consistently;
- *Service index* (SI): an 8-bit integer number, used to identify the current position within the SFP; it is set to its maximum value (ie, 255) or to a value related to the length of the SFP and is decremented of one unit by all SFC-aware SFs and SFC proxies the packet traverses in the SFP.

7.2 | OpenFlow-based NSH control plane

As an example, the SFC architecture with all its building components is plotted in Figure 3. It is composed of an SFC control plane entity, a pair of SFC-Cls, an intermediate node serving as both SFF and SFC-Pr towards SFC-unaware SFs, a SFC-aware SF, and two SFC-unaware SFs.

According to this proposal, the transport network can be controller by one or multiple network operators through a generic control plane paradigm, either SDN or non-SDN. As a matter of fact, *service providers and network providers can act as completely independent entities, each adopting its favorite control plane approach.* Nonetheless, the IETF does not specify how the architecture should be implemented. In Davoli et al,[26] it was proposed to build it around an OpenFlow-capable switch (OF-S) as follows:

- SFC entities are interconnected by means of a tunneling technology (eg, VXLAN) through an underlying network infrastructure;
- the SFP (ie, a SPI/SI pair) is mapped into the ports of the employed OF-S, thus creating a SFP-to-transport mapping;
- such mapping is combined with the tunnels to deploy the SFC in the real transport network.

In summary, each NSH interface, corresponding to a specific SPI/SI pair, is bridged to a port on the node's internal OF-S, and by means of the association of SPI/SI pairs to ports on a OF-S, it is possible to have the node acting as an NSH Service Plane component while controlling it through the OpenFlow protocol from an SDN controller, which takes the role of SFC control plane entity (SFC-Co) running applications that enforce Service Plane policies. The NSH mapping tables are therefore implemented in the form of flow tables inside the OF-S. As an example, assume port *N* of the OF-S is bridged to interface `nshM` of the node. Instructing the switch to send traffic out of port *N* will result in the node sending NSH-encapsulated traffic out of interface `nshM` with the corresponding SPI/SI values. Therefore, depending on what kind

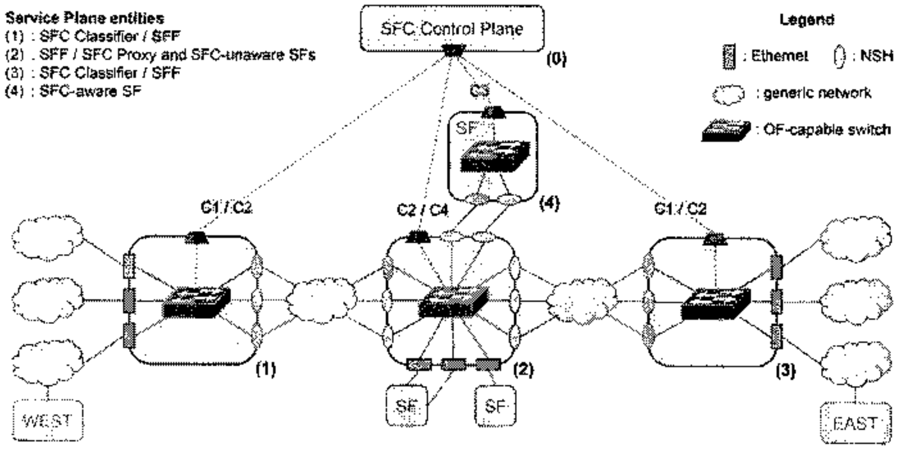


FIGURE 3 Reference NSH-based transport architecture: The role of nodes (1) to (4) is shown in the upper-left corner. NSH, network service header; SFC, service function chaining

of flow rules are installed in the internal OF-S, an SFC node can be programmed to perform different Service Plane entity functions. With reference to Figure 3, the entities are mapped to the nodes in the following way:

- Node (0) hosts the SFC-Co.
- Node (1) is responsible for adding the NSH tag to packets coming from the *WEST* domain and forwarding NSH-encapsulated packets to the first SFF in the SFP: in this role, it acts as SFC-Cl. Additionally, this node is also responsible for removing the NSH tag from packets assigned to a SFP that ends at node (1), such as packets destined to the *WEST* domain, thus acting as SFF. Following this approach, the SFC classification is as expressive as OpenFlow matching is.
- Node (2) is responsible for handling the NSH encapsulation on behalf of SFC-unaware SFs, as well as for forwarding the NSH-encapsulated packets to the following SF or SFF in the SFP. In those two tasks, node (2) acts as SFC-Pr and SFF, respectively.
- Node (3), similarly to node (1), acts both as SFC-Cl and SFF for the traffic exchanged with the *EAST* domain.
- Node (4) acts as an SFC-aware SF, as it is able to receive NSH-encapsulated packets from the SFF and process them, before sending them back to the SFF after updating the SI.

7.3 | Transport VIM and NBI

According to the ETSI MANO specifications and architecture, we can say that the transport network has its own VIM, more properly called WAN Infrastructure Manager (see Figure 1). The WAN-IM gets service specifications from the NBI and talks to the SDN controller issuing flow rules to the OF-Ses inside the SFC nodes. Again according to the JSON format specified in Section 4, the intent-based request to the WAN-IM could have the form

```
{
  "src": "IoT-Domain",
  "dst": "Cloud-Domain",
  "qos": "Delay Sensitive",
  "qos-thr": "null",
  "vnfList": "null",
  "dupList": "null"
}
```

For the transport domain, it is not possible, in general, to specify the target value of a QoS parameter in absolute quantitative terms. The domain carries a variety of customers and the transport operator does not allow the final users to control the network behavior. Therefore, we assume that what can be done in this case is just to specify a "qualitative" objective, meaning that the `qos-thr` value is either unspecified or ignored. However, the qualitative QoS class specification is still useful to the WAN-IM to decide what is the best path to route the traffic flows and enforce that choice. In the example, "qos": "Delay Sensitive" means that the service request should be accepted by finding the path with the smallest latency possible, although a minimum latency value cannot be guaranteed. The transport network controller can enforce this decision by periodically monitoring path latency and possibly rerouting the flows if a different path with smaller latency becomes available. The requirement will then be honored only on an availability basis.

8 | EXPERIMENTAL VALIDATION

8.1 | Test bed setup

As a demonstration of the feasibility of the multidomain service management solution proposed here, we developed a test bed to implement the reference architecture of the cloud-based IoT data collection service with quality differentiation illustrated in Figure 1. The complete test bed setup, including the components discussed in Sections 5 to 7, is shown in Figure 4. The customer on the top-right corner requests the service to the high-level management and orchestration functions, specifying the desired QoS feature. The orchestrator then forwards the request to the VIM REST NBIs of the relevant domains using the JSON format described in the previous sections. Each VIM performs the operations required in the respective domain and programs the underlying controllers according to the requested service and QoS feature. Data generated by the IoT devices are sent by the relevant GW via HTTP POST to the collecting/processing/publishing server in the cloud, where the customer can retrieve it (the case of delay-sensitive QoS feature is shown in the figure).

In the test bed, the data center SDN domain and the cloud domain were emulated using Mininet running in a VM.[42] The data plane topology shown in Figure 2 was built with a customized Mininet script specifying the required OpenFlow switches. Routers and VNFs were deployed as separated network namespaces in the same VM and connected to the virtual switches created by Mininet. Additional VMs were instantiated in the same physical server to deploy the data collection/processing server and the ONOS platform components needed by the SDN control plane. Those VMs were connected to the VM running Mininet through suitable virtual interfaces created inside the physical server. In order to provide the two paths with different latency, chk was configured to introduce an additional random delay uniformly distributed between 25 and 35 milliseconds, with 25% correlation between consecutive samples.

As far as the IoT domain is concerned, in our implementation, we set up an IoT network using the "European Laboratory of Wireless Communications for the Future Internet" (EuWIn) platform and, in particular, the flexible topology test bed (Flextop) facility.[43,44] The lab is composed of TI CC2530 devices,[44] compliant with IEEE 802.15.4, on top of which our

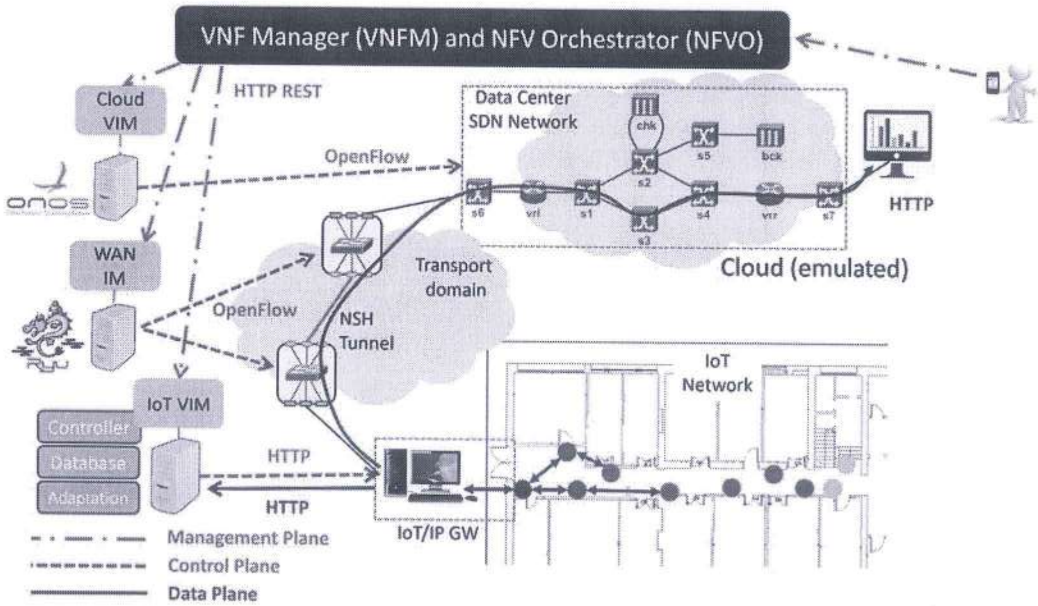


FIGURE 4 The NFV/SDN test bed setup developed to demonstrate end-to-end multidomain service management. Information flow in the management and control planes is displayed with dashed lines. Information flow in the data plane is displayed with solid lines. GW, gateway; IoT, Internet-of-Things; NFV, network function virtualization; NSH, network service header; SDN, software-defined networking; VIM, Virtualized Infrastructure Manager; VNF, virtualized network function; WAN, wide area network

SDN protocol stack is running. Nodes are located into boxes on the walls of a corridor at the University of Bologna. The map with the corresponding location of the nodes selected for the experiments (identified by colored circles) is shown in the bottom-right part of Figure 4. In particular, in the figure, we show an example of topology when setting $H = 3$ (ie, maximum three hops to reach the coordinator): The red node is the coordinator, the blue nodes are connected via one hop, the green nodes via two hops, and the yellow nodes via three hops. In our experiments, IoT data gathered by the GW were then duplicated and sent to (1) the IoTC and then to the IoT VIM and (2) to the transport domain, to be forwarded to the data center SDN domain and then to the cloud, from which the user could read the measured data.

Finally, the transport domain in our test bed was implemented on a legacy physical network, on top of which we enabled NSH encapsulation and VXLAN tunneling between pairs of NSH-capable nodes. The NSH end points serve as SFC-Cl's, as introduced in Section 7.1. An instance of the Ryu network controller[45] implements the SDN controller responsible for steering the traffic in the transport domain. It does so by means of NSH encapsulation and dynamic SPI/SI allocation.

In order to validate the adaptive traffic steering capabilities of the NSH-based transport domain, three NSH end points were deployed as ingress/egress nodes exchanging traffic with other domains. One end point was connected to the IoT domain GW located at the University of Bologna facilities. A second end point was connected to the VMs previously mentioned, where the data center SDN domain and the cloud domain were emulated with Mininet. Those VMs were deployed on a physical server located in a research-oriented computing facility in Belgium. This setup was used in a first set of experiments, where we were interested in validating the intent-based service management capabilities of the IoT, data center and cloud domains involved in our reference architecture scenario.

A second set of experiments were run after instantiating the data collection/processing server also in a VM located at the University of Bologna and connected to the third NSH end point. The latter setup was used to emulate the scenario where the required service is discovered in an edge or fog computing domain located closer to the IoT domain with respect to the remote cloud domain. In this case, the edge/fog node offering the service may not be continuously available, because of the limited and variable (eg, due to mobility) number of resources available in such kind of computing environments. However, when the required resources can be found in a local edge/fog domain, it is preferable to take advantage of them so that a delay-sensitive service can be delivered with a reduced data plane latency, resulting also in a reduced traffic load in the transport network. The adaptive traffic steering capabilities of the NSH-based transport domain allow to dynamically change the end-to-end service deployment from the cloud-based scenario to the edge/fog-based one, as sketched in Figures 5 and 6.

In the following sections, we first report the validation of the proposed intent-based service management approach in each of the different technological domains we included in our test bed. Then, we present the validation of the actual end-to-end service deployment.

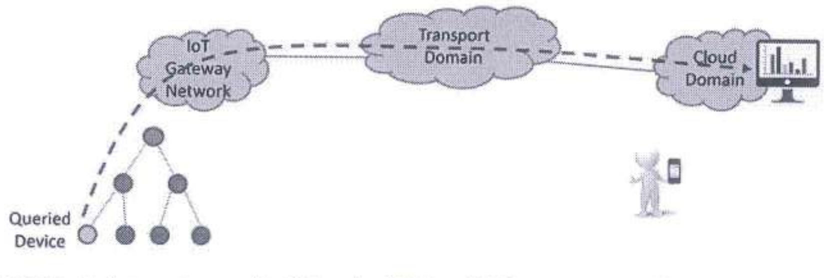


FIGURE 5 End-to-end service deployment across the Internet-of-Things (IoT), transport, and data center/cloud domains

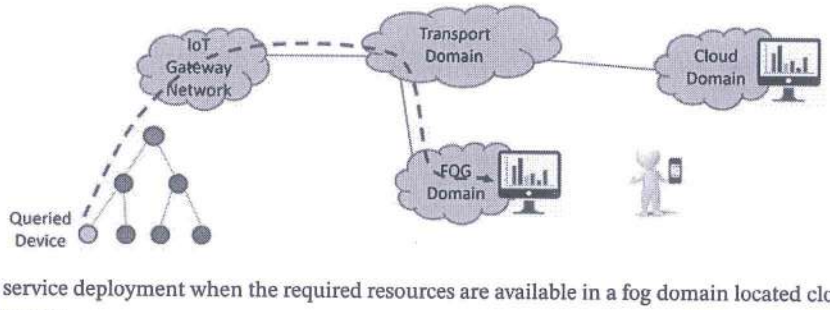


FIGURE 6 End-to-end service deployment when the required resources are available in a fog domain located closer to the user or Internet-of-Things (IoT) domain

TABLE 2 Average RTT at the Internet-of-Things (IoT) data plane (DP), control plane (CP), and Virtualized Infrastructure Manager (VIM), and controller processing time (CPT) for different quality of service (QoS) requirements

QoS	Hops	RTT at DP, ms	RTT at CP, ms	RTT at VIM, ms	CPT, ms
Delay sensitive	1 hop	12.6	516.7	522.2	239.7
Loss sensitive	3 hops	40.4	545.7	550.5	253.1

8.2 | IoT domain validation

We considered an application where the user asks for the data measured by an IoT device with a given QoS and waits for the reply. In the IoT network, both the request and the reply data frames have a payload of 10 bytes, and queries were generated by the user every second. For each query, we measured (i) the RTT at the IoT data plane, which is the interval of time between the reception of the query coming from the IoTC at the application layer of the IoT coordinator, and the reception of the reply from the target node, again at the application layer of the coordinator; (ii) the RTT at the control plane, which is the interval of time between the reception of the query coming from the VIM at the IoTC, and the reception of the reply coming from the intended GW, again at the IoTC; and (iii) the RTT measured taking the time stamp at the IoT VIM. Moreover, to better validate the control plane, we also computed the controller processing time (CPT), defined as the time instant between the reception at the controller of the query from the VIM and the instant in which the query is forwarded to the GW (this interval includes the time needed for paths computation).

Performance was evaluated by averaging over 10 000 queries generated by the user towards a node in case of a delay-sensitive service (ie, we set $H = 1$, and the node was at 1 hop from the coordinator) and a loss-sensitive service (ie, we set $H = 3$ and the node was at 3 hops from the coordinator). Results are shown in Table 2. The average RTT at the IoT data plane is mainly influenced by the number of hops, giving latency values in the order of tens of milliseconds that can be considered acceptable depending on the required QoS. As for the RTT measured at the control plane and VIM, results demonstrate that the most significant contribution depends on the IoTC response time. To better quantify the latter, the CPT is shown in the last column of the table. As can be noted, half of the RTT measured at the control plane is the time needed by the controller to process the query and generate paths. The remaining delay is due to the communication within the IoT network, for data transfer (ie, data plane RTT), plus the time needed to install the paths in the IoT network. This also validates the correct behavior of the IoT control plane from the functional point of view.

It is important to underline that paths in the IoT network were refreshed periodically and not at every query received. In particular, in our implementation, we sent one *Path* packet per device to be queried (all the nine nodes switched on in this case) every 250 seconds. As a result, the control plane RTT is not constant but presents some peaks when a new *Path* packet is generated. We measured the standard deviation of such control plane RTT that was equal to 280 milliseconds (considering all measurements taken).

8.3 | Cloud domain validation

We measured the performance within the emulated data center and cloud network when the customer requests the service specifying two traffic classes, according to the QoS features offered by the data center SDN domain: delay sensitive and loss sensitive. In this case, one-way latency in the emulated cloud network was measured by comparing timestamps of each packet captured at switches `s6` and `s7`. The capture was performed in the server hosting the Mininet VM, so the same reference clock was used for the sake of accuracy. The measurements were made by averaging over 10 000 requests.

Results are reported in Table 3 in terms of average and standard deviation of the data plane one-way latency. The numbers show the correct behavior of the data center SDN domain with respect to the requested QoS feature: Very limited latency was measured in the delay-sensitive case, whereas in the loss-sensitive case, no packet was lost and `bck` successfully stored a copy of the entire data set transmitted by the IoT GW.

We also measured the NBI response time at the VIM implemented in ONOS, ie, the time required by the VIM to process a JSON service chain specification and suitably program the SDN controller. To assess the scalability of the NBI, we generated an increasing number of requests (from 5 to 200) sent in a batch to the VIM. Each measured response time was obtained as an average over 20 runs with the same number of requests. Figure 7 shows the average NBI response time with 95% confidence intervals. The numbers show that the VIM is very responsive, in the order of tens of milliseconds.

TABLE 3 Average and standard deviation (SD) of data plane (DP) one-way latency computed at the emulated cloud network

QoS feature	Average latency, ms	SD, ms
Delay sensitive	0.3	0.28
Loss sensitive	31.7	2.41

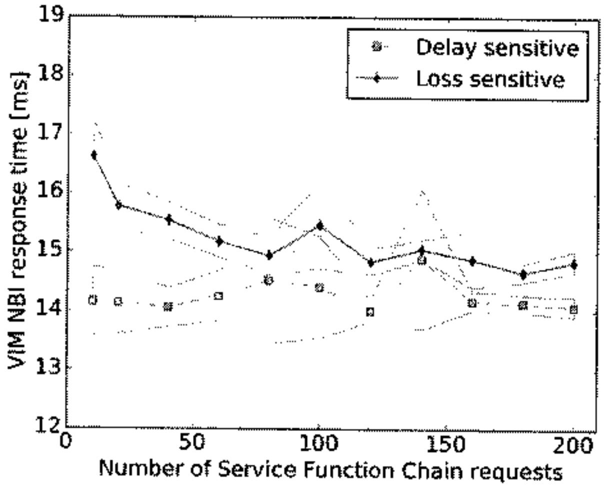


FIGURE 7 Average northbound interface (NBI) response time and 95% confidence interval at the software-defined networking (SDN)/cloud Virtualized Infrastructure Manager (VIM) with increasing number of service chain requests

TABLE 4 Average response time of the Open Network Operating System (ONOS) controller to execute the intent and flow installation in the data center software-defined networking (SDN) network

No. of vCPUs	Delay sensitive, ms	Loss sensitive, ms
2	3321.4	3468.9
4	2071.7	2984.7
8	1617.9	2866.6

The setup of loss-sensitive service chains takes slightly longer than the delay-sensitive ones because of the relatively more complex service chain to be processed.

As already discussed in Section 6, the VIM for the data center and cloud domains was developed as an application running on top of the ONOS platform and taking advantage of its connectivity-oriented, intent-based NBI. This means that the operations performed by the VIM (ie, parsing and processing a request received through its service-oriented, intent-based NBI; connecting to the ONOS NBI; programming the relevant intents) are decoupled from the ONOS-based operations (ie, installing the requested intents in its core modules and translating them into actual OpenFlow rules to be added to the controlled SDN switches). Therefore, the response time reported in Figure 7 does not include the time needed by ONOS to complete the flow rule setup. Since the latter depends on the specific SDN control technology adopted, we decided to keep it separate from the VIM response time.

However, for the sake of completeness, we report in Table 4 the time needed by ONOS to execute the intent and flow installation for the two QoS classes under different VM resource configurations in terms of number of CPUs. The results, obtained from the average over 100 SFC requests, show how the ONOS response time decreases when more resources are dedicated to it, keeping the network programming time in the order of a couple of seconds. This also proves the correct behavior of the data center and cloud domain control plane from the functional point of view. A complete functional validation of the proposed NBI and the underlying control plane was performed on a very similar experimental environment.[46]

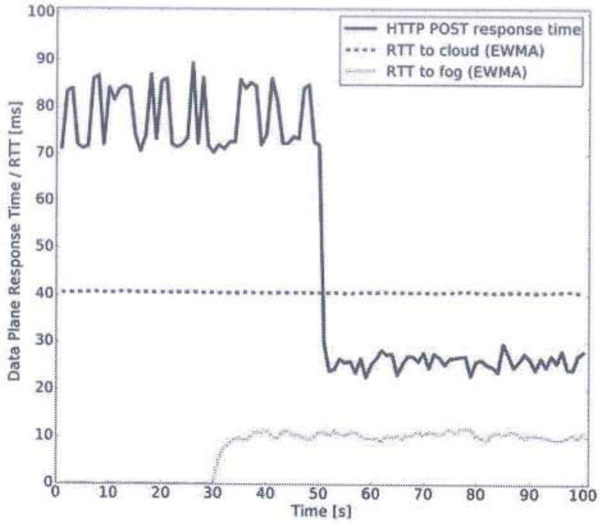


FIGURE 8 Temporal evolution of the transport data plane response time for HTTP POST requests and corresponding measured RTT values (exponential weighted moving average [EWMA] with weight $\alpha = 0.5$)

8.4 | Transport domain validation

In the transport domain, we measured the latency of the data plane between the NSH end point connected to the IoT domain and the NSH end point connected to the domain where the data "consumer" is located. To functionally validate the adaptive traffic steering capabilities of the SDN control plane adopted for the NSH-based overlay, we started with a delay-sensitive service located in the remote cloud and assumed that at some point suitable resources were discovered[§] in a fog domain located closer to the IoT domain. We used ping-based periodic RTT measurements between each pair of NSH end points (IoT-to-cloud and IoT-to-fog) to choose the domain with the minimum data plane latency. In order to stabilize the RTT measurements, we evaluated the exponential weighted moving average (EWMA) of the collected RTT samples with weight $\alpha = 0.5$. Although a single-way latency measurement may be more significant, in our setup, it was impossible to accurately assess it, as the source and destination NSH end points resided in different and remote physical machines, with nonsynchronized clock sources.

As a realistic estimation of the response time in the transport network data plane, we measured the time needed to complete a series of HTTP POST requests from end point to end point, taking into account TCP session setup, HTTP POST message request, and 200 OK response. The POST messages were generated and sent by the node serving as NSH end point connected to the IoT domain and were received and acknowledged by the node serving as the NSH end point connected to either the cloud or the fog domain. We generated 100 POST requests, sending them one per second.

In Figure 8, the temporal evolution of the transport data plane response time for HTTP POST requests is represented by the solid line, while the network-level EWMA of the RTT is represented by the dashed line for the cloud domain and by the dotted line for the fog domain. At the beginning, the HTTP traffic is sent towards the server in the cloud domain, with a quite steady RTT moving average of about 40 milliseconds. From $t = 0$ second to $t = 50$ seconds, the traffic actually reaches the cloud domain (located in Belgium), and the fluctuations in the measured response time are mainly caused by application-level delays. Meanwhile, at $t = 30$ seconds, the periodic ping measurement detects that the fog node has become available, with an RTT moving average of about 10 milliseconds, significantly lower than the RTT measured towards the cloud. After a resource/service discovery period, assumed to be completed at $t = 50$ seconds, the transport domain SDN controller steers the traffic coming from the IoT end point towards the fog domain, achieving overall better latency performances. This validates the correct behavior of the transport domain control plane from the functional point of view. The difference between the RTT values and HTTP POST response times is due to the additional overhead included in the HTTP POST transaction with respect to a simple ping packet.

[§]We did not implement a fully fledged resource discovery mechanism, as this is out of the scope of this paper. We rely on ping response to detect when the VM, representing the resource located at the edge/fog domain, becomes alive and add a 20-second interval to emulate the resource discovery phase. We consider this very simple, and definitely incomplete, resource discovery mechanism sufficient to demonstrate the correct behavior of the traffic steering in the NSH-based transport domain.

TABLE 5 Average end-to-end service deployment time, for different quality of service (QoS) features and cloud or fog domain scenarios

QoS feature	Cloud scenario, ms	Fog scenario, ms
Delay sensitive	532.3	511.8
Loss sensitive	554.0	530.1

8.5 | End-to-end service deployment validation

As a final validation, we measured the actual end-to-end service deployment time across the multidomain scenarios in Figures 5 and 6. For this analysis, it is worth reminding that the service deployment response time is due to the response time of the management plane, consisting of the VIMs orchestrating the service implementation via the NBI, the delay in the network control plane, implemented by the SDN controllers, ie, the IoTC/ONOS/Ryu platforms in this specific test bed, and the data plane latency required by the data traveling the network once the SFC is set up.

In this case, we measured the time needed for the user's request containing the intent-based service specification to reach the VIMs in the different domains, the generation of data in the IoT domain, its transmission through the transport domain to the destination server in the cloud/fog domain, and the final acknowledgment. We did not include the time needed to actually program the network control plane that was already presented in Table 4.

The measured average values, computed over 100 samples and shown in Table 5, are about half a second, the most of it due to the service management plane (orchestration, intent-based request set, and processing, etc) with just about 10% related to the network data plane latency. Nonetheless, the reduced network latency is evident when the service is "re-routed" to the fog domain. This is very important because, for all the data posted after the service setup, the network delay would be the only component (the time needed by the management plane being needed just at setup) and therefore they would experience an improvement in response time of almost 100%.

9 | CONCLUSION

In this paper, we proposed a reference architecture, inspired by the ETSI MANO framework, and an intent-based NBI for end-to-end service management across multiple technological domains. In particular, we considered the use case of a software-defined IoT infrastructure that "produces" relevant data that are processed and "consumed" at a set of cloud-based services. The IoT domain is connected to the cloud by a generic transport network. The IoT test bed, the transport network, and the cloud infrastructure are SDN enabled with specific and technology-dependent implementations of SDN controllers. An overarching orchestration service is also assumed that exploits abstractions to implement service function chains that span across the domains with a unified NBI based on the JSON syntax and service-oriented abstractions.

The paper reports the validation results that demonstrate the feasibility of the approach and the potentials of the NBI applied in real environments over a heterogeneous OpenFlow/IoT SDN test bed. The latency values measured at both data and control/management planes allowed us to get a first insight to the performance levels of the overall system, resulting in reasonable response times for service setup and QoS requirement satisfaction. Scalability tests on the ONOS-based VIM also gave promising results. The use case reported here represents a working example of a more general approach to properly define high-level interfaces and develop the related control and management components to unify orchestration capabilities across multiple SDN/NFV domains.

As future directions, we intend to test performance when multiple IoT networks are managed by the same controller, implementing a load balancing strategy. Also the case of integrating IoT networks using different technologies, such as LoRa, will be investigated. We also plan to generalize the proposed intent-based NBI in order to encompass different service scenarios that may involve multiple domains, such as 5G network slicing or multiaccess edge computing. Finally, we are also developing an original mathematical formulation of the intent mapping problem and an intent specification interpreter based on natural language.

ACKNOWLEDGMENTS

This work was partially funded by project "GAUChO—A Green Adaptive Fog Computing and Networking Architecture," funded by the Italian Ministry of Education, University and Research (MIUR) under the "PRIN 2015" program, grant no. 2015YPXH4W_004; the COST Action CA15104-IRACON; the Horizon 2020 grant no. 644090: "EuroCPS—European Network of competencies and platforms for Enabling SME from any sector building Innovative CPS products to sustain demand for European manufacturing."

ORCID

Walter Cerroni http://orcid.org/0000-0002-4629-031X

REFERENCES

1. Soares J, Goncalves C, Parreira B, et al. Toward a telco cloud environment for service functions. *IEEE Commun Mag*. 2015;53(2):98-106.
2. Peterson L, Al-Shabibi A, Anshutz T, et al. Central office re-architected as a data center. *IEEE Commun Mag*. 2016;54(10):96-101.
3. Mijumbi R, Serrat J, Gorricho JL, Bouten N, De Turck F, Boutaba R. Network function virtualization: state-of-the-art and research challenges. *IEEE Commun Surv Tutorials*. 2016;18(1):236-262.
4. Hu F, Hao Q, Bao K. A survey on software-defined network and OpenFlow: from concept to implementation. *IEEE Commun Surv Tutorials*. 2014;16(4):2181-2206.
5. Callegati F, Cerroni W, Contoli C, Cardone R, Nocentini M, Manzalini A. SDN for dynamic NFV deployment. *IEEE Commun Mag*. 2016;54(10):89-95.
6. Rosa RV, Santos MAS, Rothenberg CE. MD2-NFV: the case for multi-domain distributed network functions virtualization. In: 2015 International Conference and Workshops on Networked Systems (NetSys). Cottbus, Germany; 2015:1-5.
7. Phemius K, Bouet M, Leguay J. DISCO: distributed multi-domain SDN controllers. In: 2014 IEEE Network Operations and Management Symposium (NOMS). Krakow, Poland; 2014:1-4.
8. Sonkoly B, Czentye J, Szabo R, et al. Multi-domain service orchestration over networks and clouds: a unified approach. In: 2015 ACM SIGCOMM Conference. London, UK; 2015:377-378.
9. Clemm A, Ciavaglia L, Zambenedetti Granville L. Clarifying the concepts of intent and policy. Internet-Draft Draft-Clemm-Nmrg-Dist-Intent-01, Internet Engineering Task Force. Work in Progress; 2018.
10. Cerroni W, Buratti C, Cerboni S, et al. Intent-based management and orchestration of heterogeneous OpenFlow/IoT SDN domains. In: 2017 IEEE Conference on Network Softwarization (NetSoft). Bologna, Italy; 2017:1-9.
11. Lenrow D. Intent: don't tell me what to do! (Tell me what you want). https://www.sdxcentral.com/articles/contributed/network-intent-summit-perspective-david-lenrow/2015/02/; 2015.
12. Intent NBI. Definition and principles. 2016. https://www.opennetworking.org/sdn-resources/technical-library.
13. Cohen R, Barabash K, Rochwerger B, et al. An intent-based approach for network virtualization. In: 2013 IFIP/IEEE International Symposium on Integrated Network Management (IM). Ghent, Belgium; 2013:42-50.
14. Han Y, Li J, Hoang D, Yoo J, Hong JW. An intent-based network virtualization platform for SDN. In: 2016 12th International Conference on Network and Service Management (CNSM). Montreal, Canada; 2016:353-358.
15. Pham M, Hoang DB. SDN applications—the intent-based northbound interface realisation for extended applications. In: 2016 IEEE NetSoft Conference and Workshops (NetSoft). Seoul, Korea; 2016:372-377.
16. Kang J, Lee J, Nagendra V, Banerjee S. LMS: label management service for intent-driven cloud management. In: 2017 IFIP/IEEE International Symposium on Integrated Network Management (IM). Lisbon, Portugal; 2017:177-185.
17. Subramanya T, Riggio R, Rasheed T. Intent-based mobile backhauling for 5G networks. In: 2016 12th International Conference on Network and Service Management (CNSM). Montreal, Canada; 2016:348-352.
18. Arezoumand S, Dzeparoska K, Bannazadeh H, Leon-Garcia A. MD-IDN: multi-domain intent-driven networking in software-defined infrastructures. In: 2017 13th International Conference on Network and Service Management (CNSM). Tokyo, Japan; 2017:1-7.
19. OpenDayLight Group Based Policy User Guide. http://docs.opendaylight.org/en/stable-nitrogen/user-guide/group-based-policy-user-guide.html
20. OpenDayLight Network Intent Composition (NIC) User Guide. http://docs.opendaylight.org/en/stable-nitrogen/user-guide/network-intent-composition-(nic)-user-guide.html
21. ONOS. Intent framework. https://wiki.onosproject.org/display/ONOS/Intent+Framework.
22. Rubio-Loyola J, Serrat J, Charalambides M, Flegkas P, Pavlou G, Lafuente AL. Using linear temporal model checking for goal-oriented policy refinement frameworks. In: 6th IEEE International Workshop on Policies for Distributed Systems and Networks (POLICY'05). Stockholm, Sweden; 2005:181-190.
23. Halpern JM, Pignataro C. Service function chaining (SFC) architecture. IETF RFC 7665; 2015.
24. Quinn P, Elzur U, Pignataro C. Network service header (NSH). IETF RFC 8300; 2018.
25. Boucadair M. (SFC) Control plane components and requirements. Internet-draft draft-IETF- SFC-control-plane-08, Internet Engineering Task Force. Work in Progress.
26. Davoli G, Cerroni W, Contoli C, Foresta F, Callegati F. Implementation of service function chaining control plane through OpenFlow. In: 2017 IEEE Conference on Network Function Virtualization and Software Defined Networks (NFV-SDN). Berlin, Germany; 2017:1-4.
27. Bera S, Misra S, Vasilakos AV. Software-defined networking for Internet of Things: a survey. *IEEE Internet Things J*. 2017;4(6):1994-2008.

28. Luo T, Tan HP, Quek TQS. Sensor OpenFlow: enabling software-defined wireless sensor networks. *IEEE Commun Lett.* 2012;16(11):1896-1899.
29. Zeng D, Miyazaki T, Guo S, Tsukahara T, Kitamichi J, Hayashi T. Evolution of software-defined sensor networks. In: 2013 IEEE Conference on Mobile Ad-hoc and Sensor Networks (MSN). Dalian, China; 2013:410-413.
30. Galluccio L, Milardo S, Morabito G, Palazzo S. SDN-WISE: design, prototyping and experimentation of a stateful SDN solution for wireless sensor networks. In: 2015 IEEE Conference on Computer Communications (INFOCOM). Hong Kong; 2015:513-521.
31. Baddeley M, Nejabati R, Oikonomou G, Sooriyabandara M, Simeonidou D. Evolving SDN for low-power IoT networks. In: 2018 IEEE Conference on Network Softwarization (NetSoft). Montreal, Canada; 2018:71-79.
32. Oteafy SMA, Hassanein HS. Towards a global IoT: resource re-utilization in WSNs. In: 2012 International Conference on Computing, Networking and Communications (ICNC). Maui, Hawaii, USA; 2012:617-622.
33. Bera S, Misra S, Roy SK, Obaidat MS. Soft-WSN: software-defined WSN management system for IoT applications. *IEEE Syst J.* 2018;12(3):2074-2081.
34. Buratti C, Stajkic A, Gardasevic G, et al. Testing protocols for the Internet of Things on the EuWIn platform. *IEEE Internet Things J.* 2016;3(1):124-133.
35. Network functions virtualisation (NFV); Management and Orchestration. 2014. http://www.etsi.org/technologies-clusters/technologies/nfv.
36. The Frog 4.0 Project. https://github.com/netgroup-polito/frog4.
37. ONOS: open network operating system. http://onosproject.org.
38. OpenStack: open source cloud computing software. https://www.openstack.org.
39. Jain S, Kumar A, Mandal S, et al. B4: experience with a globally-deployed software defined WAN. *SIGCOMM Comput Commun Rev.* 2013;43(4):3-14.
40. What is software-defined WAN (or SD-WAN or SDWAN)? https://www.sdxcentral.com/sd-wan/definitions/software-defined-sdn-wan/
41. Jain R, Khondoker R. Security Analysis of SDN WAN Applications-B4 and IWAN. In: Khondoker R, ed. *SDN and NFV Security. Lecture Notes in Networks and Systems*, Vol. 30. Cham, Switzerland: Springer; 2018;111-127.
42. Mininet. http://mininet.org
43. Abrignani MD, Buratti C, Dardari D, et al. The EuWIn testbed for 802.15.4/Zigbee networks: from the simulation to the real world. In: 10th International Symposium on Wireless Communication Systems (ISWCS). Ilmenau, Germany; 2013:1-5.
44. D22.1—definition of EuWIn@CNIT/Bologna testbed interfaces and preliminary plan of activities. 2013 NoE-newcom#.
45. Ryu SDN framework. https://osrg.github.io/ryu/.
46. Callegati F, Cerroni W, Contoli C, Foresta F. Performance of intent-based virtualized network infrastructure management. In: IEEE 2017 International Conference on Communications (ICC). Paris, France; 2017:2017.

AUTHOR BIOGRAPHIES

Gianluca Davoli received his MSc degree in Telecommunication Engineering from the University of Bologna, Italy, in April 2017, and is currently a PhD student in Electronic, Telecommunications and Information Technologies at the same university. His main research interests are in the field of SDN/NFV and cloud computing, with a specific focus on modeling and design of an intent-based northbound interface for SDN controllers, as well as design, deployment and performance evaluation of service function chaining solutions leveraging NFV capabilities, and automated traffic monitoring for the enforcement of QoS policies in distributed/cloud environments.

Walter Cerroni is an assistant professor of communication networks at the University of Bologna, Italy. His research interests are related to several aspects of next generation network design, control, management, resource optimization, performance evaluation, and experimentation. The most recent topics include software-defined networking (SDN), network function virtualization (NFV), service function chaining in cloud computing platforms (eg, OpenStack), modeling and design of inter– and intra–data center interconnection networks for cloud computing infrastructures. He coauthored more than 100 papers in refereed journals and conference proceedings. He is an IEEE Senior Member.

Slavica Tomovic received MSc degree in Electronics, Telecommunications and Computer Science from the University of Montenegro in June 2015. Currently, she is a PhD student at the Faculty of Electrical Engineering, University of Montenegro. She has been working as a research and teaching assistant at Faculty of Electrical Engineering, University of Montenegro, since September 2014. Her main research interests are in the areas of software-defined networking, quality of service (QoS) management, and Internet-of-Things.

Chiara Buratti received the MSc degree in Telecommunication Engineering and the PhD degree from the University of Bologna in 2003 and 2009, respectively. Since 2011, she has been an Research Associate at the University of Bologna. She has coauthored over 80 technical papers. Her research interests are on wireless sensor networks and Internet-of-Things, with particular attention to MAC and routing protocols, and SDN solutions for IoT. She collaborated in many European projects, and she has been responsible for the Bologna site of the EuWIn platform developed within the NoE Newcom# and the Co-Chair of the EWG-Internet-of-Things of the Cost Action IRACON.

Chiara Contoli graduated in Computer Engineering from the University of Bologna, Italy in 2013. In the same year, she was research scientist at the University of California at Los Angeles, where she worked on Content Delivery Networks for automotive applications. In 2016, she was a visiting scholar at Saint Louis University, where she worked on network architecture and protocols design. She received the PhD degree in Electronic, Telecommunications and Information Technologies from the University of Bologna in 2017, where she is currently a postdoctoral researcher. Her research interests are in programmable networks, SDN, NFV, and more generally on advanced networking architectures and protocols.

Franco Callegati is an associate professor of telecommunication networks at the University of Bologna, Italy. His research interests are in the field of teletraffic modeling and performance evaluation of telecommunication networks. He is currently working on performance evaluation and experimental validation of SDN/NFV–based networking solutions. He has been active in EU-funded research projects since FP4, where he led activities and participated in various steering committees.

How to cite this article: Davoli G, Cerroni W, Tomovic S, Buratti C, Contoli C, Callegati F. Intent-based service management for heterogeneous software-defined infrastructure domains. *Int J Network Mgmt*. 2018;e2051. https://doi.org/10.1002/nem.2051

BIOGRAFIJA

Slavica 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 postdiplomske 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. Iste godine upisala je doktorske studije na Elektrotehničkom fakultetu u Podgorici. Doktorsku tezu pod nazivom „Analiza performansi novih tehnika za inženjering saobraćaja u ISP mrežama naredne generacije" prijavila je 04.05.2017. godine, a polazna istraživanja odbranila 26.06.2017. godine.

Od septembra 2015. godine angažovana je u svojstvu saradnika u nastavi na Elektrotehničkom fakultetu u Podgorici i do 2016. godine bila je angažovana na sljedećim predmetima: Poslovne računarske mreže, Mobilne radiokomunikacije, Radiotehnika, Osnove analognih telekomunikacija, Osnove digitalnih telekomunikacija, Radiokomunikacije, Računarske mreže i komunikacije, Principi mobilnih komunikacija i Teorija električnih kola. Od 2016. saradnik je na predmetima: Telekomunikacione mreže, Komutacioni sistemi, Računarske mreže, Poslovne računarske mreže, Osnove analognih telekomunikacija, Osnove digitalnih telekomunikacija, Računarske mreže i komunikacije i Teorija električnih kola. Bila je angažovana i na predmtu Pomorske komunikacije (Pomorski fakultet u Kotoru) školske 2016/2017. godine.

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 7 radova u časopisima sa SCI liste, sa ukupnim impact faktorim 13.833. Slavica Tomović takođe obavlja dužnosti recezenta u više renomiranih časopisa sa SCI liste. Oblasti u kojima je postigla najznačajnije rezultate su softverski-definisane mreže, algoritmi za inženjering saobraćaja i garanciju kvaliteta servisa u komunikacionim sistemima naredne generacije.

Učestvovala je u više nacionalnih i evropskih naučnih projekata.

Govori engleski jezik.

europass Curriculum Vitae

LIČNE INFORMACIJE

Slavica Tomović

Nikšić

+382 69 468 583

slavicat@ac.me

Datum rodjenja: 05.02.1991.

Državljanstvo: Crna Gora

Mjesto rođenja: Nikšić, Crna Gora

RADNO ISKUSTVO

2014-danas

Saradnik u nastavi

Univerzitet Crne Gore, Elektrotehnički fakultet, Podgorica

Angažovanje na predmetima:

Akademske studije:

- Računarske mreže – specijalističke studije, smjer Telekomunikacije, 2015-danas
- Računarske mreže – specijalističke studije, smjer Računari, 2015-danas
- Telekomunikacione mreže, 2015-danas
- Komutacioni sistemi, 2015-danas
- Osnove analognih telekomunikacija, 2014-2019
- Osnove digitalnih telekomunikacija, 2014-danas
- Teorija električnih kola – osnovne studije, smjer Elektronika, telekomunikacije i računari, 2014-danas
- Teorija električnih kola – osnovne studije, smjer Energetika i automatika, 2014-danas
- Mobilne radiokomunikacije, 2014-2015
- Radiotehnika, 2014-2015
- Radiokomunikacije, 2014-2015
- Električna mjerenja, smjer Energetika i automatika – 2015-2016
- Električna mjerenja, smjer Elektronika telekomunikacije i računari, 2015-2016

Studije primjenjenog računarstva:

- Poslovne računarske mreže, 2014-danas
- Računarske mreže i komunikacije, 2014-danas
- Principi mobilnih komunikacija, 2014-2015
- Računarske mreže – napredni kurs, 2015-danas

Univerzitet Crne Gore, Pomorski fakultet, Kotor

Angažovanje na predmetika:

- Pomorske komunikacije

OBRAZOVANJE

2015-danas **PhD studije**

Univerzitet Crne Gore, Elektrotehnički fakultet, Podgorica (Crna Gora)

Odsjek: Telekomunikacije

Naziv doktorske teze: „ Analiza performansi novih tehnika za inženjering saobraćaja u ISP mrežama naredne generacije "

2014 **MSc teza**

Univerzitet Crne Gore, Elektrotehnički fakultet, Podgorica (Crna Gora)

Odsjek: Telekomunikacije

Naziv magistarskog rada: "Algoritmi za podršku kvaliteta servisa u softverski definisanim mrežama"

Prosječna ocjena: A

2013 **Specijalističke studije**

Univerzitet Crne Gore, Elektrotehnički fakultet, Podgorica (Crna Gora)

Odsjek: Telekomunikacije

Naziv specijalističkog rada: "Implementacija RIP i OSPF protokola na Quagga softverskoj platformi"

Prosječna ocjena: A

2012 **BSc studije**

Univerzitet Crne Gore, Elektrotehnički fakultet, Podgorica (Crna Gora)

Odsjek: Elektronika, telekomunikacije i računari

Prosječna ocjena: A

LIČNE VJEŠTINE

Maternji jezik: Crnogorski

Ostali jezici

	RAZUMIJEVANJE		GOVOR		PISANJE
	Slušanje	Čitanje	Govorna interakcija	Govorna produkcija	
engleski	C1	C1	C1	C1	C1

Stupnjevi: A1 i A2: Početnik - B1 i B2: Samostalni korisnik - C1 i C2: Iskusni korisnik
Zajednički evropski referentni okvir za jezike

Interesovanja i ekspertiza

- Softverski-definisane mreže (SDN)
- Telekomunikacione mreže
- Programabilne hardverske platforme
- Internet of Things
- Bežične senzorske mreže
- Digitalni komunikacioni sistemi
- Komutacioni sistemi

Kompjuterske vještine

- Programski jezici: Python, Java, C, C++, C#, Perl
- Operativni sistemi Linux, Windows i prateće aplikacije.
- Verilog, VHDL, Matlab, HTML, CSS, JSF, PHP, Spring, AutoCAD.

ISKUSTVO U ISTRAŽIVAČKIM PROJEKTIMA

Projekti:

- „Razvoj Future Internet rutera na programabilnim hardverskim i *open-source* softverskim platformama", crnogorski nacionalni ICT projekat finansiran od strane Crnogorskog Ministarstva nauke, Elektrotenički fakultet, Univerzitet Crne Gore, član projekta, 2012-2017.
- „Fostering innovation based research for e-Montenegro - FORe-MONT", EU FP7 projekat, član projekta, 2014-2016.
- „Centar izvrsnosti u bioinformatici - BIO-ICT", član projekta, 2014-2017.
- „Virtuelizacija mrežnih funkcija softverski-definisane Internet of Things arhitekture za pametne gradove", Bilateralni projekat sa Univerzitetom u Bolonji, član projekta, 2016-danas.

NAGRADE I ODLIKOVANJA

- Diploma "Luča" za odlične ocjene u osnovnoj i srednjoj školi
- Stipendije ministarstva prosvjete i nauke Crne Gore za talentovane studente, 2010-2013
- Nagrada za najbolji demo na ISWCS (*International Symposium for Wireless Communication Systems*) međunarodnoj konferenciji, 2017, Bolonja, Italija.

BIBLIOGRAFIJA

RADOVI PUBLIKOVANI U ČASOPISIMA SA SCI LISTE:

1. S. Tomovic and I. Radusinovic, "RO-RO: Routing Optimality - Reconfiguration Overhead Balance in Software-Defined ISP Networks," in IEEE Journal on Selected Areas in Communications, vol. 37, no. 5, pp. 997-1011, May 2019. doi: 10.1109/JSAC.2019.2906762
2. S. Tomovic and I. Radusinovic, "An effective use of SDN for virtual-link provisioning in ISP networks," in IEICE Transactions on Communications, Vol.E102-B, No.4, pp.855-864, April 2019. DOI: 10.1587/transcom.2018EBP3191
3. S. Tomovic and I. Radusinovic, "Mapping Application Requirements to Virtualization-Enabled Software Defined WSN," Wireless Personal Communications, Vol. 97, Issue 2, November 2017, 1693-1709. DOI: https://doi.org/10.1007/s11277-017-4650-0
4. G. Davoli, W. Cerroni, S. Tomovic, C. Buratti, C. Contoli, F Callegati, "Intent-based service management for heterogeneous software-defined infrastructure domains," International Journal of Network Management, 2019; e2051. DOI: https://doi.org/10.1002/nem.2051
5. S. Tomovic, K. Yoshigoe, I. Maljevic, I. Radusinovic, "Software-Defined Fog Network Architecture for IoT," Wireless Personal Communications, Vol. 92, No. 1, pp. 181-196, January 2017. DOI: https://doi.org/10.1007/s11277-016-3845-0
6. G. Gardasevic, M. Veletic, N. Maletic, D. Vasiljevic, I. Radusinovic, S. Tomovic, M. Radonjic, "The IoT Architectural Framework, Design Issues and Application Domains", Wireless Personal Communications, Vol. 91, No. 1, pp. 127-148, January 2017.

DOI: https://doi.org/10.1007/s11277-016-3842-3

7. S. Tomovic, M. Pejanovic-Djurisic, I. Radusinovic, "SDN based Mobile Networks: Concepts and Benefits", Wireless Personal Communications, Vol. 78, No. 3, pp. 1629-1644, 2014. DOI:10.1007/s11277-014-1909-6.

RADOVI PUBLIKOVANI U DRUGIM ČASOPISIMA:

1. S. Tomovic, I. Radusinovic, "Traffic Engineering Approach to Virtual-link Provisioning in Software-defined ISP Networks," Telfor Journal, Vol. 10, No. 1, pp. 14-20, 2018.
2. S. Tomovic, I. Radusinovic, "Extending the lifetime of wireless sensor network with partial SDN deployment", Telfor Journal, Vol.8, No.1, pp. 8-13, 2016.
3. S. Tomovic, I. Radusinovic, "Energy Efficient Target Coverage in Partially Deployed Software Defined Wireless Sensor Network", Cognitive Radio Oriented Wireless Networks, Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering, vol. 172, pp.729-740, June 2016.
4. Tomovic S., Radonjic M., Radusinovic I., "Quagga Routing Platform: Application and Performance", ETF Journal of Electrical Engineering, Vol.20, No.1, pp. 126-135, September 2014, ISSN 0354-8653.
5. Tomovic S., M. Radonjic, M. Pejanovic-Djurisic, Radusinovic I., "Software-defined Wireless Sensor Networks: Opportunities and Challenges ", ETF Journal of Electrical Engineering, Vol.21, No.1, pp. 74-83, September 2015, ISSN 0354-8653.
6. G. Gogic, S. Tomovic, I. Radusinovic, "Heterogeneous implementation of OpenFlow data-centre testbed", ETF Journal of Electrical Engineering, Vol. 23, No. 1, pp. 1-10, November 2017.

RADOVI IZLOŽENI NA KONFERENCIJAMA:

1. Tomovic S., Prasad N., Radusinovic I., "SDN Control Framework for QoS Provisioning", Proc. of 22Rd Telecommunication Forum TELFOR 2014, pp. 111-114, Belgrade, Serbia, November 2014.
2. Tomovic S., Yoshigoe K., Maljevic I., Pejanovic-Djurisic M., Radusinovic I., "SDN-based Concept of QoS-aware Heterogeneous Wireless Network Operation", Proc. of 22Rd Telecommunication Forum TELFOR 2014, pp. 111-114, Belgrade, Serbia, November 2014.
3. Tomovic S., Prasad N., Radusinovic I., "Performance Comparison of QoS Routing Algorithms Applicable to Large-Scale SDN Networks", Proc. of IEEE Eurocon 2015, pp. 172-177, Salamanka, Spain, September 2015.
4. Tomovic S., Radonjic M., Radusinovic I., "Bandwidth-Delay Constrained Routing Algorithms for Backbone SDN Networks", Proc. of TELSIKS 2015, pp. 227-230, Nis, Serbia, October 2015.
5. Tomovic S., Radusinovic I., "Performance Analysis of a New SDN-based WSN Architecture", Proc. of 23Rd Telecommunication Forum TELFOR 2015, pp. 99-102, Belgrade, Serbia, November 2015.
6. Tomovic S., Lekic N., Gardasevic G., Radusinovic I.,"A New Approach to Dynamic Routing in SDN Networks", 2016 18th Mediterranean Electrotechnical Conference (MELECON), pp. 1-6, Lemesos, Cyprus, April 2016.
7. Tomovic S., Radusinovic I., "Fast and Efficient Dandwidth-delay Constrained Routing Algorithm for SDN Networks" ,IEEE NetSoft Conferenc, pp. 303-311, Seoul, South Korea, June 2016.
8. S. Tomovic and I. Radusinovic, "Allocation algorithm for handling multiple applications in software-defined WSN," 2016 24th Telecommunications Forum (TELFOR), Belgrade, 2016, pp. 1-4. doi: 10.1109/TELFOR.2016.7818740.
9. S. Tomovic et al., "An architecture for QoS-aware service deployment in software-defined IoT networks," 2017 20th International Symposium on Wireless Personal Multimedia Communications (WPMC), 2017, pp. 561-567.
10. B. Skrbic, D. Radovanovic, S. Tomovic, L. Lazovic, Z. Zecevic and I. Radusinovic, "A

decentralized platform for heterogeneous IoT networks management," 2018 23rd International Scientific-Professional Conference on Information Technology (IT), Zabljak, 2018, pp. 1-4.

11. S. Tomovic and I. Radusinovic, "A new traffic engineering approach for QoS provisioning and failure recovery in SDN-based ISP networks," 2018 23rd International Scientific-Professional Conference on Information Technology (IT), Zabljak, 2018, pp. 1-4.
12. S. Tomovic and I. Radusinovic, "Traffic engineering approach to virtual-link provisioning in software-defined ISP networks," 2017 25th Telecommunication Forum (TELFOR), Belgrade, 2017, pp. 1-4.
13. Tomovic S. Radonjic M., Radusinovic I., "Implementation of RIP and OSPF Protocols On Quagga Platform", Proc. 19th Conference on Information Technologies IT 14, pp. 64-67, Zabljak, Montenegro, February 2014.
14. Tomovic S. Radusinovic I., "OpenMont: SDN Controller for QoS Provisioning", Proc. of 58th ETRAN Conference, pp. 1-6, Vrnjacka Banja, Serbia, May 2014.
15. Ratkovic M., Tomovic S., Zaric N., Radonjic M., Radusinovic I., "SDN Network Emulation with Mininet Software Tool", Proc. 20th Conference on Information Technologies IT 15, pp. 80-83, Žabljak, Montenegro, February 2015.
16. Tomovic S., Radonjic M., Pejanovic Djurisic M., Radusinovic I., "Software Defined Wireless Sensor Networks", Proc. 20th Conference on Information Technologies IT 15, pp. 84-87, Žabljak, Montenegro, February 2015.
17. Miletic. A, Tomovic S., Radusinovic I., "Software-defined infrastructure for mobile cloud", Proc. 21th Conference on Information Technologies IT 16, pp. 137-141, Žabljak, Montenegro, February 2016.
18. Gogic G, Tomovic S., Radusinovic I., "Performance evaluation of OpenFlow data centre network testbed", Proc. 22th Conference on Information Technologies IT 17, Žabljak, Montenegro, February 2017.

Prof. dr Milica Pejanović-Djurišić

BIOGRAFIJA

Milica Pejanović-Djurišić je redovni profesor na Elektrotehničkom fakultetu Univerziteta Crne Gore u Podgorici. Nakon diplomiranja na Elektrotehničkom fakultetu u Podgorici, magistrirala je i 1987. god doktorirala na smjeru Telekomunikacije Elektrotehničkog fakulteta u Beogradu. Za rezultate postignute tokom studija nagrađivana je od strane Univerziteta Crne Gore, a dobitnik je i nagrade„19.decembar", kao i nagrade CANU iz Fonda „Petar Vukčević". Govori engleski, francuski i italijanski jezik.

Od oktobra 1982. godine je angažovana na Elektrotehničkom fakultetu Univerziteta Crne Gore, najprije u zvanju asistenta na predmetima iz grupe telekomunikacija i elektronike. Potom je u junu 1988. godine izabrana u akademsko zvanje docenta, a aprila 1994. godine u zvanje vanrednog profesora. Redovni profesor na disciplinama iz oblasti Telekomunikacija je postala u decembru 1998. godine. Usavršavala se i boravila kao gostujući profesor na univerzitetima u UK, Danskoj, SAD, Italiji, Švajcarskoj... Prof. dr Pejanović-Djurišić je u svom naučno-istraživačkom radu orijentisana na oblast mobilnih radio komunikacija, 5G sistema, kooperativnih relejnih sistema, rješenja za bežični IoT, kao i generalno tehnika i tješenja za unaprijedjivanje performansi bežičnih komunikacionih sistema gdje je postigla zapažene rezultate koji su publikovani u preko 200 naučnih radova u međunarodnim časopisima i na međunarodnim konferencijama, kao i u više od 90 naučnih i stručnih radova u domaćim časopisima i na konferencijama. Autor je tri knjige, više naučnih studija i stručnih projekata. Održala je i veliki broj predavanja i seminara po pozivu na inostranim univerzitetima i relevatnim medjunarodnim konferencijama, a bila je i mentor u izradi značajnog broja magistarskih i doktorskih disertacija u oblasti telekomunikacija. Koordinirala je brojne naučne i stručne projekte kako na nacionalnom, tako i na medjunarodnom nivou, ukjljučujući EU FP6, FP7 i H2020 programe.

Prof. Pejanović-Djurišić je je angažovana i na projektima ITU (International Telecommunication Union), organizacije UN u Ženevi, u oblasti mobilnih radio sistema novih generacija i implementacije Internet tehnologija. Značajno iskustvo je stekla radeći i kao konsultant industrijskih kompanija iz oblasti mobilnih komunikacija, kao evaluator medjunarodnih projekata i kao član globalnih profesionalnih asocijacija (IEEE, IEICE, Društva za telekomunikacije Srbije i Crne Gore) i inicijativa u oblasti info-komunikacionih tehnologija. Član je i Programskog komiteta TELFOR-a, TELSIKSa, brojnih medjunarodnih konferencija iz oblasti telekomunikacija (GWS, WPMC, EUROCON, WTS...). Obavljala je i dužnost predsjednika Upravnog odbora Telekoma Crne Gore (1999.- 2002.), kao i predsjednika Odbora direktora prvog crnogorskog Internet provajdera.

Nastavne aktivnosti Prof. Pejanović-Djurišić na Elektrotehničkom fakultetu odnose se na sledeće discipline:

- Osnove telekomunikacija (osnovne akademske studije)
- Osnove digitalnih telekomunikacija (osnovne akademske studije)
- Mobilne radiokomunikacije (MSc studije, program Telekomunikacije)
- Principi modernih telekomunikacija (MSc studije, program Telekomunikacije)
- Personalni telekomunikacioni sistemi (doktorske studije)
- Protokoli i tehnologije za bežične komunikacione sisteme (doktorske studije).

Prof. dr Milica Pejanović-Djurišić

Izabrane publikacije

KNJIGA

1. M.Pejanovic-Djurisic, E.Kocan, R.Prasad: "OFDM based relay systems for future wireless networks", River Publishers, Denmark, 2012, (ISBN: 9788792329271) https://www.riverpublishers.com/book_details.php?book_id=117

RADOVI U MEDJUNARODNIM ČASOPISIMA SA SCI(E) LISTE:

1. G. Sheng, S. Dang, Z. Zhang, **E. Kocan**, M. Pejanovic–Djurisic, *OFDM WITH INDEX MODULATION ASSISTED BY MULTIPLE AMPLIFY-AND-FORWARD RELAYS*, IEEE Wireless Communication Letters, (ISSN: 2162-2337), Jan. 2019. DOI: 10.1109/LWC.2019.2893258. https://ieeexplore.ieee.org/document/8612925
2. **E. Kocan**, B. Domazetovic, M. Pejanovic–Djurisic, *RANGE EXTENSION IN IEEE 802.11AH SYSTEMS THROUGH RELAYING*, Wireless Personal Communications (ISSN: 0929-6212), vol. 97, no 2, pp. 1889-1910, November 2017. http://dx.doi.org/10.1007/s11277-017-4334-9
3. E. Kocan, M. Pejanovic-Djurisic, PERFORMANCE IMPROVEMENT OF DUAL-HOP OFDM DECODE-AND-FORWARD RELAY SYSTEM, Wireless Personal Communications (ISSN: 0929-6212), vol. 93, no. 3, pp. 769-778, April 2017. http://dx.doi.org/10.1007/s11277-014-2227-8.
4. E. Kocan, M. Pejanovic-Djurisic, TOWARDS FUTURE LOW EXPOSURE MOBILE CELLULAR NETWORKS, Wireless Personal Communications (ISSN: 0929-6212), vol. 92, no. 1, pp. 221-235, January 2017. http://dx.doi.org/10.1007/s11277-016-3847-y
5. Y. Huang, N. Varseir, S. Niksic, E. Kocan, M. Pejanovic-Djurisic, et al., COMPARISON OF AVERAGE GLOBAL EXPOSURE OF POPULATION INDUCED BY A MACRO 3G NETWORK IN DIFFERENT GEOGRAPHICAL AREAS IN FRANCE AND SERBIA, Bioelectromagnetics (ISSN: 0197-8462), vol. 37, no.6, pp. 382-390, Sept. 2016.

   http://onlinelibrary.wiley.com/doi/10.1002/bem.21990/abstract

6. M. Pesko, M. Smolnikar, M. Vucnik, T. Javornik, M. Pejanovic-Djurisic, M. Mohorcic, SMARTPHONE WITH AUGMENTED GATEWAY FUNCTIONALITY AS OPPORTUNISTIC WSN GATEWAY DEVICE, Wireless Personal Communications (ISSN: 0929-6212), vol. 78, no. 3, pp. 1811-1826, October 2014. https://doi.org/10.1007/s11277-014-1908-7

7. U. Urosevic, Z. Veljovic, M. Pejanovic–Djurisic, "A NEW SOLUTION FOR SIMPLE COOPERATIVE RELAYING", Wireless Personal Communications, (ISSN: 0929-6212), Vol. 75, Issue 2, pp. 1235-1250, Springer, March 2014.
https://doi.org/10.1007/s11277-013-1420-5
8. M. Ilic-Delibasic, M. Pejanovic-Djurisic, MRC DUAL-DIVERSITY SYSTEM OVER CORRELATED AND NON-IDENTICAL RICEAN FADING CHANNELS, IEEE Communication Letters, Vol. 17, no. 12, pp. 2280 - 2283, December 2013.
https://ieeexplore.ieee.org/document/6656080
9. U. Urosevic, Z. Veljovic, M. Pejanovic–Djurisic, MIMO SOLUTION FOR PERFORMANCE IMPROVEMENTS OF OFDM-CDMA SYSTEM WITH PILOT TONE, Wireless Networks (ISSN: 1022-0038), Vol. 19, Issue 8, pp. 2021-2028, November 2013.
https://doi.org/10.1007/s11276-013-0586-9
10. E. Kocan, M. Pejanovic – Djurisic, D. S. Michalopoulos, G. K. Karagiannidis, PERFORMANCE EVALUATION OF OFDM AMPLIFY-AND-FORWARD RELAY SYSTEM WITH SUBCARRIER PERMUTATION, IEICE Trans. on Communications (ISSN: 0916-8516), Vol. E93-B, No.05, pp. 1216-1223, May. 2010.
https://www.jstage.jst.go.jp/article/transcom/E93.B/5/E93.B_5_1216/_article

UNIVERZITET CRNE GORE

81001 PODGORICA Cetinjski put bb, P. Fah 99, Jugoslavija
Tel: (38 81) 14-484 Fax: (38 81) 11-301

Broj: 01-1312
Podgorica, 25.12.1998.

Na osnovu člana 97. Zakona o Univerzitetu ("Sl. list RCG" br. 37/92) i člana 94. Statuta Univerziteta Crne Gore, Naučno-nastavno vijeće Univerziteta Crne Gore na sjednici održanoj 24.12.1998. donijelo je

ODLUKU
o izboru u zvanje

Dr MILICA PEJANOVIĆ-DJURIŠIĆ bira se u zvanje redovnog profesora Univerziteta Crne Gore za predmete Telekomunikacije I, Telekomunikacije II i Mobilne radiotelekomunikacije za rad na neodredjeno vrijeme sa punim radnim vremenom na Elektrotehničkom fakultetu u Podgorici.

PRAVNA POUKA: *Protiv ove Odluke lica koja smatraju da su im povrijedjena prava imaju pravo žalbe Naučno- - nastavnom vijeću Univerziteta Crne Gore u roku od 15. dana.*

REKTOR,

Prof. dr Ratko Djukanović

Prof.dr Igor Radusinović

BIOGRAFIJA

Igor Radusinović je rođen 29.07.1972. godine na Cetinju. U Podgorici je završio osnovnu i srednju školu. Nakon završetka srednje škole, 1990. godine, upisao je Elektrotehnički fakultet u Podgorici, gdje je u roku diplomirao jula 1994. godine sa prosječnom ocjenom 9,54.
Tokom obrazovanja više puta je nagrađivan, pri čemu je posebno važno napomenuti: diplome LUČA za osnovnu i srednju školu, NOVČANU NAGRADU UNIVERZITETA CRNE GORE koja se dodjeljuje najboljem studentu završne godine Elektrotehničkog fakulteta u Podgorici i PLAKETU UNIVERZITETA CRNE GORE kao najboljem studentu generacije u oblastima tehničkih i prirodnomatematičkih nauka.
Magistarske studije – oblast Telekomunikacije (Telekomunikacione mreže) na Elektrotehničkom fakultetu u Beogradu, je sa prosječnom ocjenom 10 (deset) završio 1997. godine odbranom magistarske teze. Na Elektrotehničkom fakulteta u Beogradu je 2003. godine odbranio doktorsku disertaciju pod nazivom "ANALIZA PERFORMANSI ATM KOMUTATORA SA STANOVIŠTA LOKACIJE I UPRAVLJANJA REDOVIMA ČEKANJA".
Od 01.12.1994. godine radi kao saradnik na Katedri za telekomunikacije Elektrotehničkog fakulteta u Podgorici. Biran je u zvanje asistenta za predmete Telekomunikacione mreže i Komutacioni sistemi na Katedri za telekomunikacije Elektrotehničkog fakulteta u Podgorici. U periodu od 2003. do 2008. godine je obavljao poslove docenta na predmetima Telekomunikacione mreže, Komutacioni sistemi i Računarske mreže na Katedri za telekomunikacije Elektrotehničkog fakulteta u Podgorici. Od 2008. godine do 2013. godine, je bio u zvanju vanrednog profesora na predmetima Telekomunikacione mreže, Komutacioni sistemi i Računarske mreže na Elektrotehničkom fakultetu u Podgorici. Od 2013. godine je u zvanju redovnog profesora na predmetima Telekomunikacione mreže, Komutacioni sistemi i Računarske mreže na Elektrotehničkom fakultetu u Podgorici.
Tokom svog dosadašnjeg profesionalnog angažmana na Elektrotehničkom fakultet obavljao je više značajnih funkcija od kojih su najznačajnije: šef Katedre za telekomunikacije, rukovodilac Računarskog Centra, rukovodilac studijskog programa Elektronika, Telekomunikacije i Računari, predstavnik rukovodstva za kvalitet i rukovodilac Laboratorije za mjerenje elektromagnetnih emisija.
U dosadašnjem naučnom radu objavio je preko 150 radova u renomiranim međunarodnim i domaćim naučnim časopisima, kao i na međunarodnim i domaćim konferencijama. Oblasti naučnog istraživanja su telekomunikacione mreže sa posebnim rezultatima u teoriji redova čekanja, komutaciji paketa, tehnikama kontrole zagušenja i softverski definisanim mrežama. Do sada je bio mentor pri izradi dvije doktorske disertacije, dvanaest magistarskih teza i preko 80 diplomskih i specijalističkih radova. Igor Radusinović je rukovodio ili učestvovao u izradi sedamnaest domaćih i šest međunarodnih naučnih i stručnih projekata. Učestvovao je u izradi preko 50 elaborata o uticaju zračenja baznih stanica na životnu sredinu. Učestvovao je u mjerenjima i izradi Izvještaja o ispitivanju nivoa električnog polja za preko 100 radio emisionih sistema.
Član je profesionalnih udruženja IEEE, IEICE, Inžinjerske komore Crne Gore i Društva za telekomunikacije. Član je Naučnog odbora TELFOR-a, Programskog odbora INFOFEST-a i Programskog odbora IT. Recenzent je eminentnih međunarodnih časopisa IEEE Communications Letters i IEEE/ACM Transactions on Networking.
Od oktobra 2009. do februara 2011. godine je obavljao funkciju pomoćnika ministra za nauku, istraživanja i tehnološki razvoj u Ministarstvu prosvjete i nauke Vlade Crne Gore. Od marta 2011. do februara 2013. godine je obavljao dužnost predsjednika nacionalnog Savjeta za naučnoistraživačku djelatnost. Od početka 2010. godine do februara 2013. je bio predstavnik Crne Gore u ERAC (European Reasearch Area Committee) i Bordu guvernera JRC (Joint Research Center Board of Governors). Od jula 2012. do avgusta 2016. godine je bio član Upravnog odbora Univerziteta Crne Gore. Od decembra 2011. godine je član, a od juna 2013. do jula 2017. godine i predsjednik Odbora direktora Pošte Crne Gore a.d. Podgorica. Od 2016. do aprila 2019. godine je bio senator Univerziteta Crne Gore. Govori engleski jezik. Oženjen je i ima tri sina.

DESET REFERENCI

1. S. Tomović and **I. Radusinović**, "RO-RO: Routing Optimality - Reconfiguration Overhead Balance in Software-Defined ISP Networks," in *IEEE Journal on Selected Areas in Communications*, vol. 37, no. 5, pp. 997-1011, May 2019, Print ISSN: 0733-8716, DOI: 10.1109/JSAC.2019.2906762.

   Link na rad: https://ieeexplore.ieee.org/document/8672627
   SCI lista:
   https://ieeexplore.ieee.org/xpl/RecentIssue.jsp?punumber=49

2. S. Tomović and **I. Radusinović**, "An effective use of SDN for virtual-link provisioning in ISP networks," in *IEICE Transactions on Communications*, Vol.E102-B, No.4, pp.855-864, April 2019, ISSN: 0916-8516, DOI: 10.1587/transcom.2018EBP3191-

   Link na rad:
   https://search.ieice.org/bin/summary.php?id=e102-b_4_855&category=-&lang=E&year=2019&abst=
   SCI lista: http://www.ieice.org/cs/jpn/EB/index.html

3. S. Tomović and **I. Radusinović**, "Mapping Application Requirements to Virtualization-Enabled Software Defined WSN," *Wireless Personal Communications*, Vol. 97, Issue 2, November 2017, pp. 1693-1709, ISSN: 0929-6212.

   DOI: https://doi.org/10.1007/s11277-017-4650-0
   Link na rad: https://link.springer.com/article/10.1007/s11277-017-4650-0
   SCI lista: https://link.springer.com/journal/11277

4. S. Tomović, K. Yoshigoe, I. Maljević, I. Radusinović, "Software-Defined Fog Network Architecture for IoT," *Wireless Personal Communications*, Vol. 92, No. 1, pp. 181-196, January 2017, ISSN: 0929-6212, DOI: https://doi.org/10.1007/s11277-016-3845-0

   Link na rad: https://link.springer.com/article/10.1007/s11277-016-3845-0
   SCI lista: https://link.springer.com/journal/11277

5. G. Gardašević, S. Divanović, M. Radonjić, **I. Radusinović**, "A qos-aware dual crosspoint queued switch with largest weighted occupancy first scheduling algorithm," *IEICE Transaction on Communications*, Vol.E98-B, No.01, January 2015, pp. 201-208, ISSN: 0916-8516 (print version), ISSN: 1745-1345 (Online), DOI: 10.1587/transcom.E98.B.201.

   Link na rad:
   https://search.ieice.org/bin/summary.php?id=e98-b_1_201&category=B&lang=E&year=2015&abst=
   SCI lista:
   http://mjl.clarivate.com/cgi-bin/jrnlst/jlresults.cgi?PC=MASTER&Word=*IEICE%20Transactions%20on%20Communications

6. **I. Radusinović**, M. Radonjić, A. Šimurina, I. Maljević, Z. Veljovic, "A new analytical model for the cq switch throughput calculation under the bursty traffic", *International Journal of Electronics and Communications (AEU)*, Vol. 66, No 12, December 2012, pp.1038– 1041, ISSN: 1434-8411, DOI: https://doi.org/10.1016/j.aeue.2012.05.009.

   Link na rad: https://www.sciencedirect.com/science/article/pii/S143484111200129X
   SCI lista:
   http://mjl.clarivate.com/cgi-bin/jrnlst/jlresults.cgi?PC=MASTER&Word=*AEU

7. M. Radonjić, **I. Radusinović**, "Impact of scheduling algorithms on performance of crosspoint-queued switch", *Annals of Telecommunications*, Vol 66, No 5-6, May/June 2011, pp.363-376, ISSN: 0003-4347 (print version), ISSN: 1958-9395 (electronic version), DOI: 10.1007/s12243-010-0214-y.

   Link na rad: https://link.springer.com/article/10.1007/s12243-010-0214-y
   SCI lista:

http://mjl.clarivate.com/cgi-bin/jrnlst/jlresults.cgi?PC=MASTER&Word=*Annals%20of%20Telecommunications

http://mjl.clarivate.com/cgi-bin/jrnlst/jlresults.cgi?PC=MASTER&ISSN=*0003-4347

8. D. Banović, **I. Radusinović**, "Scheduling algorithm for VOQ switches," *AEU - International Journal of Electronics and Communications*, Vol. 62, Issue 6, Jun 2008., pp. 455-458, ISSN 1434-8411, DOI: https://doi.org/10.1016/j.aeue.2007.06.004.

   Link na rad: https://www.sciencedirect.com/science/article/pii/S1434841107001288
   SCI lista:
   http://mjl.clarivate.com/cgi-bin/jrnlst/jlresults.cgi?PC=MASTER&Word=*AEU

9. **I. Radusinović**, M. Pejanović, Z. Petrović, "Performance of the dual-banyan atm switch under arbitrary traffic patterns," *European Transactions on Telecommunications (Transactions on Emerging Telecommunications Technologies)*, No.14, pp 319 – 323, November 2003, DOI: **https://doi.org/10.1002/ett.927**, ISSN: 2161-3915.

   Link na rad: https://onlinelibrary.wiley.com/doi/abs/10.1002/ett.927
   SCI lista:
   http://mjl.clarivate.com/cgi-bin/jrnlst/jlresults.cgi?PC=MASTER&Word=*Transactions%20on%20Emerging%20Telecommunications%20Technologies

10. **I. Radusinović**, M. Pejanović, Z. Petrović, "A new analytical model for the dual banyan switch throughput calculation," *IEEE Communications Letters*, Vol. 6., No. 2., pp 76-78, February 2002, ISSN: 1089-7798, DOI: 10.1109/4234.984701.

    Link na rad: https://ieeexplore.ieee.org/document/984701
    SCI lista:
    http://mjl.clarivate.com/cgi-bin/jrnlst/jlresults.cgi?PC=MASTER&Word=*IEEE%20Communications%20Letters

УНИВЕРЗИТЕТ ЦРНЕ ГОРЕ

Ул. Цетињска бр. 2
П. фах 99
81000 ПОДГОРИЦА
ЦРНА ГОРА
Телефон: (020) 414-255
Факс: (020) 414-230
E-mail: rektor@ac.me

UNIVERSITY OF MONTENEGRO

Ul. Cetinjska br. 2
P.O. BOX 99
81 000 PODGORICA
MONTENEGRO
Phone: (+382) 20 414-255
Fax: (+382) 20 414-230
E-mail: rektor@ac.me

Број: 08-2130
Датум, 26.09. 2013. г.

Ref: ______
Date, ______

Na osnovu člana 75 stav 2 Zakona o visokom obrazovanju (Sl.list RCG, br. 60/03 i Sl.list CG, br. 45/10 i 47/11) i člana 18 stav 1 tačka 3 Statuta Univerziteta Crne Gore, Senat Univerziteta Crne Gore, na sjednici održanoj 26.09.2013. godine, donio je

ODLUKU
O IZBORU U ZVANJE

Dr IGOR RADUSINOVIĆ bira se u akademsko zvanje **redovni profesor** Univerziteta Crne Gore za predmete: Telekomunikacione mreže, Komutacioni sistemi, Računarske mreže (t) i Računarske mreže (r), na studijskom programu Elektronika, Telekomunikacije i Računari, na Elektrotehničkom fakultetu.

REKTOR

Prof. dr Predrag Miranović

УНИВЕРЗИТЕТ ЦРНЕ ГОРЕ
ЕЛЕКТРОТЕХНИЧКИ ФАКУЛТЕТ
Број 02/2-1519
Подгорица, 04.10. 2013 год.

Зоран Чича

Биографија

Зоран Чича је рођен 18.03.1979. у Загребу. Основно образовање је започео у Загребу. У Сремској Митровици је завршио основну школу, а потом уписао и завршио гимназију. Након тога, 1997. године је уписао Електротехнички факултет у Београду као редован студент и дипломирао 27.05.2002. са укупном средњом оценом 9.50, на дипломском испиту оцена 10. Током студија хонорарно радио као демонстратор у Лабораторији за електронику Електротехничког факултета у Београду. Постдипломске студије - смер Телекомуникационе и рачунарске мреже на Електротехничком факултету у Београду је уписао 2002. Испите на постдипломским студијама положио је са просечном оценом 10. Магистарску тезу „Примена детерминистичке теорије сервисних система у планирању транспортних мрежа", чији је ментор био редовни професор др Гроздан Петровић, одбранио је децембра 2007. године. Докторску дисертацију „Имплементација функција пакетског процесирања у Интернет рутерима великог капацитета", чији је ментор била ванредни професор др Александра Смиљанић, одбранио је 12.07.2012. године.

У 2002. стекао је звање асистента-приправника на Електротехничком факултету у Београду. У 2008. је стекао звање асистента. У 2013. је стекао тренутно звање доцента. Држао је рачунске и лаб вежбе из више предмета на Електротехничком факултету у Београду: Телекомуникационе мреже, Пројектовање дигиталних телефонских централа, Рачунарске телекомуникације, Рачунарске основе и примена Интернета, Комутациони системи, Програмирање комуникационог хардвера, Интернет програмирање, Архитектура свичева и рутера. Такође је држао рачунске вежбе из предмета Рачунарске мреже и комуникације на Војно-техничкој академији. У звању доцента, држао је предавања из више предмета на Електротехничком факултету у Београду: Комутациони системи, Програмирање комуникационог хардвера, Мрежна администрација и програмирање, Телекомуникационе мреже за приступ, Широкопојасне телекомуникационе мреже, Теорија телекомуникационог саобраћаја, Синхронизација у телекомуникационим мрежама. Учествовао је на неколико пројеката финансираних од стране Министарства надлежног за науку у оквиру којих је био аутор/коаутор више техничких решења. Такође, Зоран Чича је био ментор више дипломских/завршних и мастер теза (више од 80 дипломских/завршних теза, више од 50 мастер теза), а био је и члан више комисија за дипломске/завршне и мастер тезе. Такође је био члан комисије за три докторске тезе, од којих је једна била двојна теза (Електронски факултет у Нишу и Norwegian University of Science and Technology у Трондхајму, Норвешка). Зоран учествује и у раду Комисије за студије II степена на Електротехничком факултету у Београду (као Председник комисије).

Зоран Чича је аутор више радова на домаћим и међународним конференцијама, као и часописима. Два рада су била награђена у категорији Најбољи радови младих аутора на конференцијама ЕТРАН 2007 и 2009. Четири рада су објављена у часописима са импакт фактором - 2 рада у IET Electronics Letters, 1 рад у Computer Networks и један рад у IEEE Computer Arhitecture Letters. Аутор је једног уџбеника (за предмет Програмирање комуникационог хардвера) који се користи у склопу наставе на Електротехничком факултету у Београду.

Зоран Чича је рецензент за међународне часописе IEEE Communication Letters, IEEE Transactions on Communications, Computer Networks, Wireless Personal Communications, Computer Communications, IEEE Journal on Selected Areas in Communications, за међународну конференцију IEEE Workshop on High Performance Switching and Routing, TELFOR и icETRAN, као и за домаћу конференцију ЕТРАН. У 2010. добио је признање *Exemplary reviewer* за часопис IEEE Communication Letters.

ДЕСЕТ РЕФЕРЕНЦИ

[1] **Z. Čiča**, A. Smiljanić, „Balanced Parallelised Frugal IPv6 Lookup Algorithm," *IET Electronics Letters*, vol.47(17), pp. 963-965, August 2011, IF=0.965, ISSN 0013-5194, DOI: 10.1049/el.2011.0966, **M23**
Линк на рад: https://ieeexplore.ieee.org/abstract/document/6004738
SCI листа:
http://mjl.clarivate.com/cgi-bin/jrnlst/jlresults.cgi?PC=MASTER&Word=*Electronics%20Letters

[2] A. Smiljanić, **Z. Čiča**, „A Comparative Review of Scalable Lookup Algorithms for IPv6," *Computer Networks*, vol.56(13), pp. 3040-3054, September 2012, IF=1.200, ISSN 1389-1286, DOI: 10.1016/j.comnet.2012.04.027, **M22**
Link na rad: https://www.sciencedirect.com/science/article/abs/pii/S1389128612001806
SCI листа:
http://mjl.clarivate.com/cgi-bin/jrnlst/jlresults.cgi?PC=MASTER&Word=*Computer%20networks

[3] **Z. Čiča**, „Non-blocking Frame Based Multicast Scheduler for IQ Switches," *IET Electronics Letters*, vol.52(4), pp. 285-287, February 2016, IF=1.155, ISSN 0013-5194, DOI: 10.1049/el.2015.3547, **M23**
Линк на рад: https://ieeexplore.ieee.org/document/7405391
SCI листа:
http://mjl.clarivate.com/cgi-bin/jrnlst/jlresults.cgi?PC=MASTER&Word=*Electronics%20Letters

[4] S. Durkovic, **Z. Čiča**, „Birkhoff-von Neumann Switch Based on Greedy Scheduling," *IEEE Computer Architecture Letters*, vol.PP(99), pp. 1-1, May 2017, IF=1.397, ISSN 1556-6056, DOI: 10.1109/LCA.2017.2707082, **M22**
Линк на рад: https://ieeexplore.ieee.org/document/7932455
SCI листа:
http://mjl.clarivate.com/cgi-bin/jrnlst/jlresults.cgi?PC=MASTER&Word=*IEEE%20Computer%20Architecture%20Letters

[5] **Z. Čiča**, „IP Lookup as a Critical Functionality of Packet Processors," *Telfor Journal*, vol. 5(1), pp. 8-13, 2013, ISSN 1821-3251, **M52**
SCOPUS: http://journal.telfor.rs/Public/Home.aspx
Линк на рад: http://journal.telfor.rs/Published/Vol5No1/Vol5No1_A2.pdf

[6] S. Durkovic, **Z. Čiča**, „Birkhoff-von Neumann Switch with Deflection Based Load Balancing," *Telfor Journal*, vol. 9(1), pp. 14-19, 2017, ISSN 1821-3251, DOI: 10.5937/telfor1701014D, **M52**
SCOPUS: http://journal.telfor.rs/Public/Home.aspx
Линк на рад: http://journal.telfor.rs/Published/Vol9No1/Vol9No1_A3.pdf

[7] M. Carević, **Z. Čiča**, „FPGA Implementation of IP Packet Segmentation and Reassembly in Internet Router," *Serbian Journal of Electrical Engineering*, vol. 6(3), pp. 399-407, December 2009, ISSN 1451-4869, DOI: 10.2298/sjee0903399c **M52**
Линк на рад: http://www.journal.ftn.kg.ac.rs/Vol_6-3/03-Carevic-Cica.pdf
SCOPUS: http://www.journal.ftn.kg.ac.rs/

[8] **Z. Čiča**, A. Smiljanić, „Frugal IP Lookup Based on a Parallel Search," *Proc. of IEEE Workshop on High Performance Switching and Routing 2009*, Paris, France, June 2009, ISBN 978-1-4244-5174-6, DOI: 10.1109/HPSR.2009.5307435, **M33**

[9] **Z. Čiča**, L.Milinković, A. Smiljanić, „FPGA Implementation of Lookup Algorithms," *Proc. of IEEE Workshop on High Performance Switching and Routing 2011*, Cartagena, Spain, July 2011, ISBN 978-1-4244-8454-6, DOI: 10.1109/HPSR.2011.5986037, **M33**

[10] V. Soskic, A. Smiljanić, **Z. Čiča**, „Implementation of the multicast LDP protocol on the FPGA chips," *Proc. of IEEE Workshop on High Performance Switching and Routing 2014*, Vancouver, Canada, July 2014, ISBN 978-1-4799-1633-7, DOI: 10.1109/HPSR.2014.6900897, **M33**

УНИВЕРЗИТЕТ У БЕОГРАДУ

Студентски трг 1, 11000 Београд, Република Србија
Тел.: 011 3207400; Факс: 011 2638912; E-mail: officebu@rect.bg.ac.rs

ВЕЋЕ НАУЧНИХ ОБЛАСТИ
ТЕХНИЧКИХ НАУКА

Београд, 26.2.2018. године
02 број: 61202-869/2-18
ЛД

На основу чл. 145. Закона о високом образовању ("Службени гласник РС", број: 88/17), чл. 47. ст. 5. тач. 1. Статута Универзитета у Београду ("Гласник Универзитета у Београду", број 186/15-пречишћени текст и 189/16), чл. 13. ст. 1. Правилника о већима научних области на Универзитету у Београду ("Гласник Универзитета у Београду", број 134/07, 150/09, 158/11, 164/11, 165/11, 180/14, 195/16 и 197/17), чл. 21. ст. 1. тач. 1. Правилника о начину и поступку стицања звања и заснивања радног односа наставника Универзитета у Београду ("Гласник Универзитета у Београду", број 142/08, 150/09, 160/11 и 196/16) и Правилника о минималним условима за стицање звања наставника на Универзитету у Београду ("Гласник Универзитета у Београду", број 192/16, 195/16, 197/17 и 199/17), а на предлог Изборног већа Електротехничког факултета, број: 2515/7 од 13. фебруара 2018. године, Веће научних области техничких наука, на седници, одржаној 26. фебруара 2018. године, донело је

О Д Л У К У

БИРА СЕ **др Зоран Чича**, у звање ванредног професора за ужу научну област Телекомуникације.

О б р а з л о ж е њ е

Универзитет у Београду – Електротехнички факултет је дана 20. децембра 2017. године у публикацији „Послови", објавио конкурс за избор у звање ванредног професора, за ужу научну област: Телекомуникације, због истека изборног периода.

Извештај Комисије за припрему извештаја о пријављеним кандидатима стављен је на увид јавности дана 15. јануара 2018. године, на сајту Факултета.

На основу предлога Комисије за припрему извештаја о пријављеним кандидатима, Изборно веће Електротехничког факултета, на седници одржаној 13. фебруара 2018. године, донело је одлуку о утврђивању предлога да се кандидат др Зоран Чича, изабере у звање ванредног професора.

Факултет је дана 19. фебруара 2018. године доставио Универзитету комплетан захтев за избор у звање на прописаним обрасцима.

Универзитет је комплетну документацију коју је доставио Факултет ставио на web страницу Универзитета, дана 19. фебруара 2018. године.

Веће научних области техничких наука, на седници одржаној дана 26. фебруара 2018. године, разматрало је захтев Факултета и утврдило да кандидат испуњава услове прописане чл. 74. и 75. Закона о високом образовању и чланом 125. Статута Универзитета у Београду, као и услове прописане Правилником о минималним условима за стицање звања наставника на Универзитету у Београду, па је донета одлука као у изреци.

ПРЕДСЕДНИК ВЕЋА

Проф. др Јован Филиповић

Доставити:
- Факултету (2),
- Архиви Универзитета (1).

ЕЛЕКТРОТЕХНИЧКИ ФАКУЛТЕТ
(назив послодавца)
БЕОГРАД, Булевар краља Александра 73
(седиште послодавца)
2515/10
(број)
11 APR 2018
(датум)

На основу чланова 24 став 1, 27, 30-33 Закона о раду ("Службени гласник РС", број 24/2005, 61/2005, 54/2009, 32/2013 и 75/2014) и члана 75 Закона о високом образовању („Службени гласник РС" 88/2017) закључује се

УГОВОР О РАДУ

1. **ЕЛЕКТРОТЕХНИЧКИ ФАКУЛТЕТ** – Булевар краља Александра 73 (у даљем тексту: (назив и седиште послодавца) послодавац) заснива радни однос са:

др Зораном Чичом, из Сремске Митровице, Марка Перичина 10/8
(име и презиме запослене, место пребивалишта, односно боравишта)

1803979330129 — доктор електротехничких наука
ЈМБГ — (стручни,академски,научни назив)

2. Радни однос се заснива за обављање послова у звању **ванредног професора**, са следећим описом послова:
- извођење предавања и вежби на I, II i III степену студија;
- припрема и извођење предавања и вежби, као и других облика наставе;
- рад на изради и осавремењавању наставних планова и програма студија које се остварују на Факултету;
- праћење и примена новина у области наставних метода;
- припрема и обављање испита;
- консултације са студентима;
- обављање осталих облика наставе који су предвиђени програмом наставног предмета;
- организовање појединачног и заједничког научног рада са студентима;
- менторство у изради дипломских радова;
- менторски рад са студентима магистарских, односно докторских студија;
- учешће у раду комисије за одобравање, оцену или одбрану магистарског рада, односно докторске дисертације;
- остваривање наставе на студијама за иновацију знања, као и на студијама за остваривање програма стручног усавршавања;
- иновације у настави;
- сарадња са сарадницима у току остваривања свих облика наставе;
- учешће у раду катедре, научно-наставног већа, изборног већа и других стручних органа и комисија Факултета
- обавља и друге послове везане за научно-наставни процес као и послове и задатке које му декан и стручни органи Факултета ставе у задатак у складу са својим знањем и потребама Факултета.

Катедра за телекомуникације
(организациона јединица)

3. Запослени ће обављати послове у Београду (седишту послодавца).
4. Запослени заснива радни однос на:
   1) неодређено време, почев од / (датум заснивања радног односа)
   2) одређено време од **15.04.2018.** до **14.04.2023.** у трајању од **5 година** (месеци, година)
5. Запослени заснива радни однос са пуним (пуним или непуним) радним временом, у трајању од **40** (бр. часова недељно)
6. Запослени је дужан да ступи на рад **15.04.2018. год.** (дан, месец и година почетка рада)
7. Запослени прихвата да у току рада може да буде распоређен на друге послове у складу са законом и општим актом о раду послодавца.

8. Запослени има право на зараду за обављени рад и време проведено на раду. Елементи који одређују зараду запосленог су:

S = 2,700 бодова,
K_1= 25 бод/год.,
K_3= 0

На дан 15.04.2018. год. запосленом се признаје Z = 0 година проведених у звању, односно струци. Основица (S*1000 + K) износи 2725 бодова. Промена Z се рачуна од 15.04.2019. године.

Послодавац се обавезује да запосленом приликом сваке исплате зараде и накнаде зараде, достави писмени обрачун, у складу са Законом.

9. Запослени има право на увећану зараду у складу са законом, колективним уговором и општим актом послодавца.

10. Запослени има право на зараду из добити Послодавца сагласно општем акту послодавца.

11. Запослени има право на накнаду зараде и друга примања у складу са законом, колективним уговором и општим актом послодавца. Запосленом се зарада и друга примања исплаћују у роковима утврђеним општим актом послодавца.

12. Запослени има право на накнаду трошкова превоза:
- за долазак и одлазак са рада,
- за време проведено на службеном путу у земљи и иностранству у складу са законом и општим актом послодавца.

13. Запослени има право на одмор у току рада, на дневни, недељни и годишњи одмор у складу са законом и општим актом послодавца. О времену коришћења годишњег одмора одлучује послодавац уз претходну консултацију запосленог.

14. Запослени има право на плаћено одсуство у случају:
- склапања брака – 5 радних дана,
- порођаја супруге – 5 радна дана,
- теже болести или смрти члана уже породице – 5 радна дана,
- добровољног давања крви – 2 узастопна дана рачунајући и дан давања крви,
- за приватне потребе по одобрењу декана – до 2 радна дана.

За дане плаћеног одсуства запослени прима накнаду у висини зараде коју би остварио да је у те дане радио.

15. Запослени је одговоран за штету коју је на раду или у вези са радом – намерно или из крајње непажње, прозроковао послодавцу.

16. Послодавац може отказати уговор о раду запосленом ако крши радне обавезе, и то:
- ако је утврђено да не остварује резултате рада;
- ако је утврђено да нема потребна знања и способности за обављање послова на којима ради;
- ако не поштује радну дисциплину, односно ако је његово понашање такво да не може да настави рад код послодавца;
- ако учини доступним или саопшти садржај испитних задатака и решења пре термина одржавања испита;
- ако учини кривично дело на раду и у вези са радом;
- ако злоупотреби право на одсуство због привремене спречености за рад;
- ако незаконито располаже средствима;
- ако повреди прописе о заштити од пожара, експлозије, елементарних непогода и штетних деловања отровних и других опасних материја, као и повреда прописа и непредузимања мера ради заштите запослених, средстава рада и животне средине;
- ако ода пословну, службену или другу тајну утврђену законом или општим актом послодавца као и одавање података из тендерске документације коју подноси факултет по расписаној јавној набавци;
- ако ода пословну, службену или другу тајну утврђену законом или општим актом послодавца;
- ако одбије да обавља послове на које је распоређен;
- ако нецелисходно и неодговорно користи средства рада;
- ако фалсификује новчана и друга документа;
- ако изазове већи неред или тучу на раду;
- ако чешће долази на рад у напитом стању или употребљава алкохол или наркотик за време рада, које смањује способност за рад или омета процес рада;
- ако не достави потврду о привременој спречености за рад у року од три дана;
- ако злоупотреби радну обавези у намери да за себе или другог прибави имовинску корист; односно свако друго незаконито и неовлашћено понашање са наведеном намером;
- ако проневери или украде имовину факултета.

17. Ако запослени претрпи повреду или штету на раду или у вези са радом, послодавац је дужан да му накнади штету, у складу са Законом и општим актом.

Послодавац одговара за штету коју одговорно лице или запослени вршећи злостављање проузрокује другом запосленом код истог послодавца, у складу са Законом о спречавању злостављања на раду.

Запослени који врши злостављање, као и запослени који злоупотреби право на заштиту од злостављања, одговоран је за непоштовање радне дисциплине, односно повреду радне дужности.

18. Послодавац се обавезује да одмах по ступању запосленог на рад поднесе прописане пријаве на обавезно социјално осигурање и да благовремено уплаћују одговарајуће доприносе, у складу са законом.

19. Запослени је дужан да се придржава прописаних мера заштите на раду.

20. Послодавац је дужан да обезбеди услове рада и организује рад којим се обезбеђује заштита живота и здравља запосленог, у складу са законом и другим прописима.

21. Запослени и послодавац прихватају сва права, обавезе и одговорности утврђене законом и општим актом послодавца.

22. Запослени може послодавцу да откаже уговор о раду у писаној форми у року од 15 дана пре дана наведеног као дан престанка радног односа (отказни рок).

23. Све спорне ситуације из овог уговора, уговорне стране ће решавати споразумно. У случају спора, надлежан је стварно надлежни суд у Београду.

24. Свака од уговорних страна може да откаже овај уговор, под условима и случајевима утврђеним законом.

25. Овај уговор је сачињен у 4 (четири) истоветна примерка, од којих 1 (један) примерак задржава запослени, а 3 (три) послодавац.

ЗАПОСЛЕНИ

ЗА ПОСЛОДАВЦА