

PASLAUGŲ KOKYBĖS UŽTIKRINIMAS NAUDOJANT *OPENFLOW* PROTOKOLĄ

Liudas DUOBA

Vilniaus Gedimino technikos universitetas, Vilnius, Lietuva
El. paštas liudas@duoba.lt

Santrauka. *OpenFlow* šiuo metu yra vienas iš nedaugelio protokolų, kuris gali padėti įgyvendinti programiškai valdomų tinklų (angl. *software defined networks*) pradininkų deklaruotas idėjas. Protokolas unikalus tuo, kad geba atskirti duomenų perdavimo įrenginio (komutatoriaus, maršrutų parinktuvo) valdymo (angl. *control plane*) ir duomenų perdavimo (angl. *data plane*) plokštumas. Visa valdymo logikos sistema perduodama numatytiems įrenginiams – valdikliams (angl. *controller*). Darbe siūlomas metodas, kaip naudojant *OpenFlow* protokolo savybes galima garantuoti teikiamų duomenų perdavimo paslaugų kokybę.

Reikšminiai žodžiai: programiškai valdomi tinklai, *OpenFlow* protokolas, kokybės valdymas.

Įvadas

Neenumaldomai didėjantis viešais ir privačiais tinklais perduodamų duomenų kiekis kartu ragina plėsti infrastruktūrą, reikalingą duomenų perdavimo tinklams sėkmingai veikti. Auganti debesų kompiuterijos technologijų taikymo tendencija (Lara *et al.* 2013) verčia telekomunikacijų ir kitas įmones statyti naujus ir plėsti esamus duomenų centrus. Savo ruožtu taip pat neatsilieka ir kompiuterių tinklai – šie plečiami ir atnaujinami, siekiant garantuoti teikiamų paslaugų kokybę bei atitikti augančios greಿತaveikos poreikius. Kiekvienas naujas duomenų centras ir didesnis duomenų tinklų perdavimo operatorius susiduria su dideliais iššūkiais:

- Kaip užtikrinti efektyvų duomenų perdavimo tinklo valdymą?
- Kaip paspartinti tinklo ir paslaugų plėtrą?
- Kaip efektyviai panaudoti tinklo resursus?

Ieškant atsakymų į visus šiuos klausimus, pradėta domėtis galimybe kompiuterių tinklų valdymą atskirti nuo įprastinių tinklo įrenginių (pvz., komutatorių, maršrutų parinktuvų) ir perkelti visą valdymo sistemą į centralizuotą įrenginį – valdiklį, kuriame veikia atitinkama programinė įranga (Casado *et al.* 2007; Nadeau, Gray 2013).

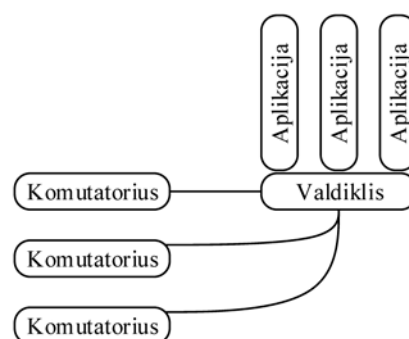
Taip visi tinklo elementai taptų valdomi programiniu būdu, o toks tinklas būtų vadinamas programine įranga arba programiškai valdomas tinklu (angl. *Software Defined Networks (SDN)*). Didžiųjų kompiuterių tinklų įrenginių ir technologijų gamintojų (pvz., *Cisco* arba *Juniper Networks*) pasiūlyti SDN sprendimai, deja, nėra suderinami tarpusavyje, todėl daugelio žvilgsnis krypta į atvirojo kodo

iniciatyva paremtą *OpenFlow* technologiją (Bakshi 2013), kuri yra suderinta su įvairių kompanijų gaminama įranga.

OpenFlow architektūra

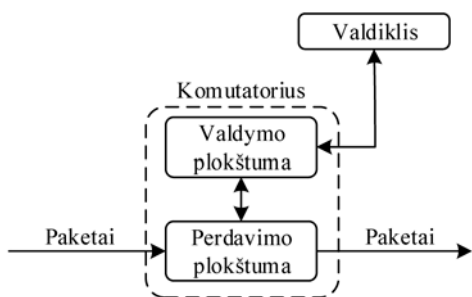
Įprastinė *OpenFlow* protokolo valdomo tinklo architektūra sudaryta iš vieno ar kelių valdiklių bei daugelio komutatorių (1 pav.), kurių kiekvienas su valdikliu užmezga sesiją, skirtą apsikeisti valdymo duomenimis.

Valdiklis sudarytas iš atskirų programų, kurios, gavusios duomenų paketus iš komutatorių, pagal tam tikrą algoritmą juos apdoroja ir parenka reikalingą instrukciją: perduoti, atmesti, pakeisti. Taigi duomenų paketas gali būti nukreipiamas į tam tikrą parinkto komutatoriaus prievadą, jam gali būti pakeista DSCP ar MAC adreso vertė arba jis tiesiog gali būti neapdorotas, nurodant jį neaptarnauti.



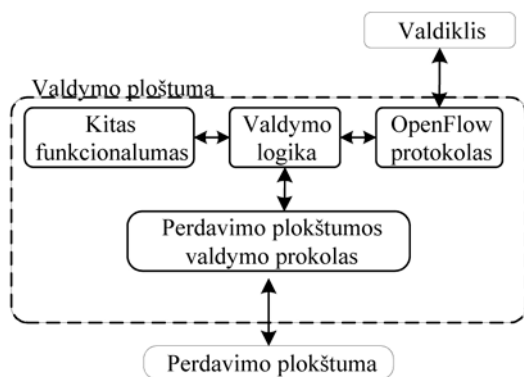
1 pav. *OpenFlow* valdomo tinklo architektūra

Fig. 1. *OpenFlow* network architecture



2 pav. *OpenFlow* komutatoriaus valdymo ir perdavimo plokštumų atskyrimas

Fig. 2. Decoupling data and control planes of *OpenFlow* enabled switch



3 pav. *OpenFlow* komutatoriaus valdymo plokštuma

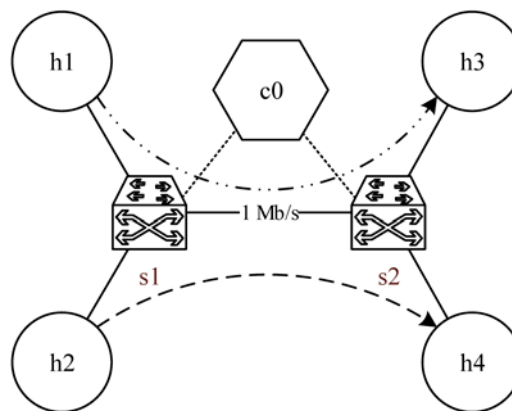
Fig. 3. Control plane of *OpenFlow* enabled network switch

OpenFlow architektūroje naudojami komutatoriai yra ypatingi tuo, jog jų valdymo ir perdavimo plokštumos yra atskirtos (2 pav.). Perdavimo plokštuma, atliekanti paketų komutavimą, yra tiesiogiai priklausoma nuo valdymo plokštumos. Komutatorius, gavęs nežinomą paketą, pirmiausiai jį persiunčia valdymo plokštumai, kuri savo ruožtu visą informaciją persiunčia valdikliui (3 pav.). Čia duomenų paketas apdorojamas ir veikiančios programos tokiam paketui komutuoti sugeneruoja taisyklių rinkinį, kuris siunčiamas atgal komutatoriui. Toks rinkinys verčiamas instrukcija ir įrašomas komutatoriaus į ASIC lustus.

Instrukcijos gali būti pačio valdiklio nuolat atnaujinamos, taip yra kuriama dinamiška duomenų perdavimo aplinka. Statinių taisyklių nebuvimas laikomas didžiausiu programiškai valdomų tinklų privalumu, leidžiančiu kurti lanksčias duomenų perdavimo taisykles bei užtikrinti lengvą plėtrą.

Kokybės garantavimas *OpenFlow* tinkluose

Auganti duomenų perdavimo apimtis verčia telekomunikacijų įmones ieškoti efektyvių būdų, kaip aptarnauti padidėjusį paslaugų poreikį ir suteikti kokybiškas paslaugas.



4 pav. Tinklo topologija: valdiklis – c0; komutatoriai – s1, s2; tarnybinės stotys – h1, h2, h3 ir h4

Fig. 4. Topology of network: controller – c0; switch – s1, s2; hosts – h1, h2, h3 ir h4

Nemaža dalis operatorių teikiamoms paslaugoms garantuoti gali taikyti gana brangų būdą – technologinį perteklumą, tai jokių žinių nereikalaujantis būdas, kai aklaui didinamas esamų resursų kiekis. Toks būdas negali būti vadinamas problemos sprendimu.

Vietoje to kokybei garantuoti reikėtų pasitelkti intelektualius metodus. Visų pirma, reikia gerai suprasti ir išsiaiškinti, kurie duomenų srautai yra svarbūs ir reikalauja kokybinių parametrų (vėlinimas, švytavimas, paketų (ne)praradimas) užtikrinimo (Van Adrichem *et al.* 2014), o kuriems užtenka minimalių resursų. Srautus identifikuoti galime naudodamiesi informacija iš IPv4 ir IPv6 antraščių: *ToS* ar *Traffic Class* reikšmės, vektorius iš IP šaltinio bei tikslo adresų, TCP ir UDP prievadų, VLAN numerio. Tam gali būti panaudojama ir išskirtinai IPv6 *Flow Label* reikšmė ar MPLS žymės (angl. *label*) reikšmė.

Identifikavus srautus derėtų garantuoti deramą resursų kiekį. Tai gali būti pasiekta apribojant (angl. *policing*) kitus, mažiau svarbius srautus arba duomenis nukreipiant kitu, tačiau kokybės reikalavimus garantuojančiu keliu.

Siekiant parodyti pagrindines kokybės valdymo funkcijas *OpenFlow* protokolo valdomuose tinkluose, modeliuota situacija virtualiame tinkle (4 pav.), kuriame veikia du duomenų srautus generuojantys šaltiniai. Modelis kuriamas *Mininet* aplinkoje naudojant *Ryu* valdiklį.

Kokybinių parametrų valdymo metodas

Siūlomas metodas turi tradiciniuose tinkluose beveik nepasitaikančią savybę – žinoti viso tinklo topologijos būseną bei turėti visą informaciją apie ją. Tai padeda automatiškai rasti ir pritaikyti geriausią perduodamų duomenų srautų konfigūraciją be sistemų administratoriaus įsikišimo (Das *et al.* 2010).

Skirtingai nuo dabar esamų kokybės valdymo mechanizmų, metodas suteikia galimybę dinamiškai reaguoti į pasikeitusias tinklo charakteristikas ir geriau išnaudoti turimus tinklo resursus.

Kadangi metodas pasinaudodamas *OpenFlow* protokolo įrankiais sudaro viso tinklo žemėlapi, siūloma atsisaityti transliavimo (angl. *broadcast*) duomenų paketų srauto. Galiniams įrenginiams reikalingą informaciją (pvz., ARP, DHCP) siūloma perduoti centralizuotai tiesiai iš valdiklio ir taip sumažinti tokio tipo duomenų paketų plitimą tinkle.

Kuriant metodą remiamasi prielaidomis, jog plečiant tinklą gali reikėti pakeisti įrangą taip, kad ji palaikytų didesnę TCAM lentelių įrašų skaičių, ir padidinti skaičiavimo resursus tinklo valdikliui.

Metodo algoritmo dalys yra šios:

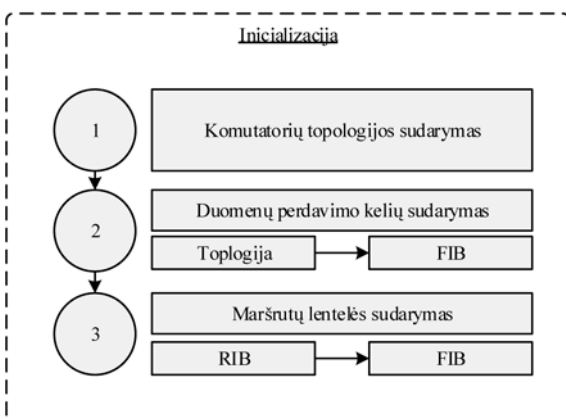
- inicializacija;
- kokybinių parametrų stebėjimas;
- duomenų perdavimas.

Inicializacijos metu (5 pav.) vienas iš pirmųjų uždavinių yra sudaryti esamo tinklo topologiją. Pirmiausiai iš visų komutatorių ir tarpusavio jungčių sudaromas grafas. Antruoju etapu prie šio grafo prijungiami informacijos siuntėjai ir gavėjai, arba, kitaip tariant, kompiuteriai.

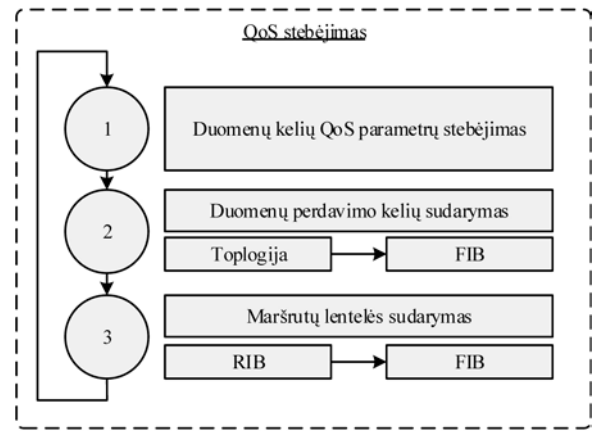
Įrašius pirmuosius paprastuosius kelius iš karto apskaičiuojami kokybiniai parametrai (6 pav.), skirti surasti keliams, geriau atitinkantiems duotųjų srautų kokybinius poreikius. Pagal gautus duomenis kiekvienam srautui sudaromi pagrindiniai ir alternatyvieji duomenų perdavimo keliai.

Remiantis šiais duomenimis ir inicializacijos metu gauta informacija, sudaromos taisyklės, skirtos duomenims perduoti atsižvelgiant į srautų kokybinius parametrus.

Kokybinių parametrų skaičiavimo procesas yra nuolatinis: stebima esama tinklo situacija bei reaguojama į įvykusius pasikeitimus. Pvz., padidėjus tam tikro sujungimo vėlinimui ir viršijus nustatytą kažkuriam srautui ribą,



5 pav. Procesai, vykstantys inicializacijos etape
Fig. 5. Processes during initialization step



6 pav. Procesai, vykstantys QoS stebėjimo etape
Fig. 6. Processes during QoS monitoring step

tam srautui bus ieškoma alternatyvaus kelio, kuris atitiks reikalaujamas kokybines charakteristikas.

Duomenų perdavimo dalyje atliekamas anksčiau paminėtų dalių apskaičiuotų kelių perkėlimas į komutatorių taisyklių lenteles naudojantis *OpenFlow* protokolu.

Kokybinių parametrų stebėjimas

Tam, kad metodas tinkamai veiktų ir būtų galima įvertinti, kaip perduoti duomenų srautus, reikalingi praktiniai parametrų matavimai, t. y. būdai, kaip gauti informaciją apie paketų vėlinimą, švytvimą ir praradimą bei duomenų srautų greitaveiką.

Bendra idėja ta, kad valdiklis generuoja matavimo paketą su laiko žyme ir siunčia į matuojamą tinklo segmentą. Galinis komutatorius, gavęs tokį paketą, jį perduoda valdikliui, kuris apskaičiuoja paketo vėlinimą. Taigi vėlinimas tiriamame tinkle apskaičiuojamas pagal formulę:

$$\Delta T = T_2 - T_1 - \frac{T_{S1} + T_{S3}}{2} - k, \quad (1)$$

čia T_1 – matavimo paketo išsiuntimo laikas; T_2 – matavimo paketo gavimo laikas; T_{S1} – S1 komutatoriaus sukuriamas vėlinimas; T_{S3} – S3 komutatoriaus sukuriamas vėlinimas; k – konstanta, apibūdinanti valdymo paketų apdorojimo trukmę komutatoriuose.

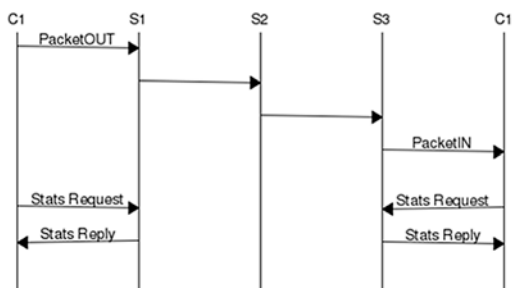
Iliustruojant vėlinimo matavimo pavyzdį, galima remtis 7 pav. pavaizduota siunčiamų paketų diagrama.

Įgyvendintas vėlinimo matavimo algoritmas ir jo palyginimas su įprasta *ping* funkcija parodytas 8 pav.

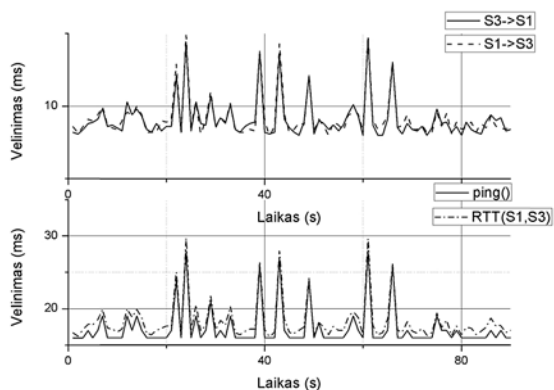
Siekiant nustatyti švytvimą parametras, panaudoti gauti viso matuojamo kelio vėlinimo duomenys. Rezultatai apskaičiuojami pagal formulę:

$$\Delta T_{sv} = T_i - T_{i-1}, \quad (2)$$

čia T_i – paskutinio matavimo vėlinimas; T_{i-1} – $i-1$ matavimo paketo gavimo laikas.



7 pav. Vėlinimo matavimo algoritmo metu siunčiami paketai
Fig. 7. Packets which are required for measuring network latency



8 pav. Vėlinimo matavimo algoritmo palyginimas su ping funkcija

Fig. 8. Comparison of results of latency measurement algorithm and ping function

Matuojamas kelio greitaveikos parametras G apskaičiuojamas pagal kiekvieno prievado, esančio kelyje, užtikrinamą greitaveiką duotuoju metu. Matuojama vertė nustatoma pagal mažiausiąją gautą vertę:

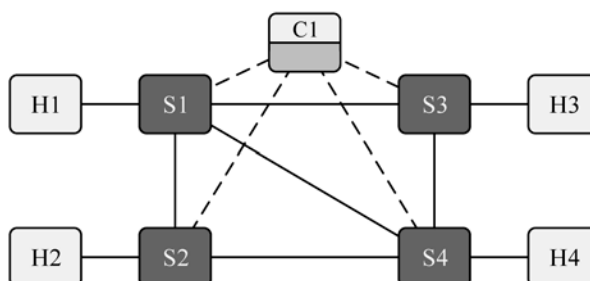
$$G = \min \left(i, \frac{n_{i2} + n_{i1}}{\Delta t} \right), \quad (3)$$

čia n_{i1} – i -tojo prievado išsiųstų bitų skaičius matavimo pradžioje; n_{i2} – i -tojo prievado išsiųstų bitų skaičius matavimo pabaigoje; Δt – laiko skirtumas tarp matavimų.

Gauti rezultatai

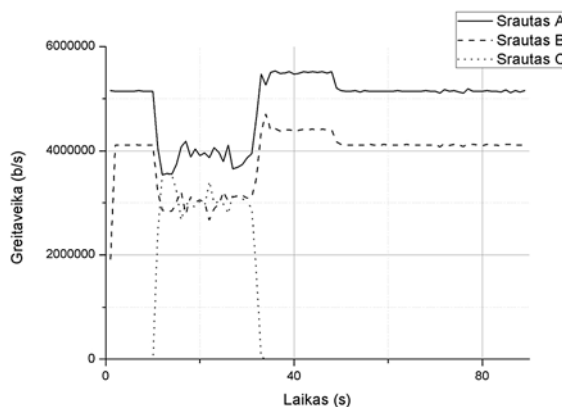
Kokybės valdymo metodo bandymai buvo vykdomi 9 pav. pavaizduotoje kompiuterių tinklo topologijos schemoje. Tinklą sudaro valdiklis C1, keturi aparatiniai HP 5406zl komutatoriai S1, S2, S3 ir S4 bei keturi kompiuteriai H1, H2, H3 ir H4. Visi komutatoriaus prievada nustatyti veikti 10 Mb/s režimu.

Testavimo eksperimentą sudaro trys dalys. Pirmosios dalies metu iš kompiuterio H3 sukuriamas 4 Mb/s srautas A į kompiuterį H1. Lygiagrečiai sukuriamas 5 Mb/s srautas



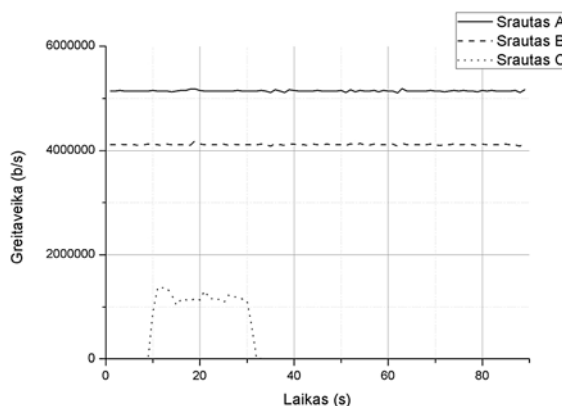
9 pav. Srautų išdėstymo pagal svarbą algoritmo veikimui testuoti taikoma topologija

Fig. 9. Network topology for testing flow prioritization algorithm



10 pav. Duomenų srautai su papildoma apkrova be kokybės valdymo mechanizmo

Fig. 10. Network flows with a load while QoS control mechanism is off



11 pav. Duomenų srautai su papildoma apkrova, taikant kokybės valdymo mechanizmą

Fig. 11. Network flows with a load while QoS control mechanism is on

B iš kompiuterio H4 į kompiuterį H1. Nuo 10 s įjungiamas papildomas srautas C iš kompiuterio H2 į kompiuterį H1, kurio greitaveika 4 Mb/s.

Analizuojant greitaveikos rezultatus 10 pav. matyti, kad srautų A ir B kokybės charakteristikos greitaveikos

atžvilgiu pablogėjusios 17–20 %, o reikalinga greitimeika nėra garantuojama.

Trečiasis eksperimentas skirtas antrojo testo scenarijui pakartoti, tik šį kartą bus įjungiamas kokybės valdymo mechanizmas, nustačius srauto A ir srauto B kokybinius reikalavimus, taip pat turi būti užtikrintos atitinkamai 4 ir 5 Mb/s greitimeikos.

Gauti trečiojo eksperimento rezultatai pavaizduoti 11 pav. Šį kartą skirtumas akivaizdus: srautų A ir B greitimeikos charakteristikos viso eksperimento metu nepakitusios, o atsiradęs srautas C gauna tik mažą dalį likusių išteklių. Tai puikiai iliustruoja kokybės valdymo mechanizmo dinaminį funkcionalumą.

Išvados

1. Sukurto kokybinių parametrų valdymo metodo tinklo topologijos duomenų bazės sudarymo algoritmas geba sukurti viso tinklo topologijos medį, pagal kurį pats metodas parenka optimalų duomenų perdavimo kelią, taikant trumpiausio ir plačiausio kelio radimo algoritmus.
2. Metodas taip pat leidžia parinkti optimalų duomenų perdavimo kelią, atsižvelgiant į jungčių kokybines charakteristikas.
3. Metodas veikia su visais rinkoje esančiais ir *OpenFlow* 1.3 protokolą palaikančiais komutatoriais. Vienintelis parametras, kurį reikia modifikuoti, yra naudojamų lentelių skaičius, kuris yra parenkamas pagal komutatoriaus technines savybes.

Literatūra

- Bakshi, K. 2013. Considerations for software defined networking (SDN): approaches and use cases, in *Aerospace Conference*, 2–9 March 2013, Montana, USA, 1–9.
- Casado, M.; Freedman, M. J.; Pettit, J.; Luo, J.; McKeown, N.; Shenker, S. 2007. Ethane: taking control of the enterprise, in *SIGCOMM Computer Communication Review*, New York, ACM, 1–12. <https://doi.org/10.1145/1282380.1282382>
- Das, S.; Parulkar, G.; McKeown, N.; Singh, P.; Getachew, D.; Ong, L. 2010. Packet and circuit network convergence with OpenFlow, in *Optical Fiber Communication Conference*, 21–25 March 2010, San Diego, USA, 1–3.
- Lara, A.; Kolasani, A.; Ramamurthy, B. 2013. Network innovation using OpenFlow: a Survey, *IEEE Communications Surveys & Tutorials* 16(1): 493–512. <https://doi.org/10.1109/SURV.2013.081313.00105>
- Nadeau, T. D.; Gray, K. 2013. *SDN: Software defined networks*. Sebastopol: O'Reilly Media. 384 p. ISBN 978–1-449-34230-2.
- Van Adrichem, N. L. M.; Doerr, C.; Kuipers, F. A. 2014. OpenNetMon: network monitoring in openflow software-defined networks, in *IEEE Network Operations and Management Symposium, (NOMS)*, 5–9 May 2014, Krakow, Poland, 1–8.

ASSURING QUALITY OF SERVICE BY USING OPENFLOW PROTOCOL

L. Duoba

Abstract

OpenFlow currently is the only one protocol that can implement ideas of the Software Defined Networking pioneers. The Protocol is unique because of ability to separate any network device control and data planes. All network intelligence is (logically) centralized in dedicated controllers. Objective of this article to propose a new method that enhances QoS in data networks by using *OpenFlow* protocol.

Keywords: Software defined networks, OpenFlow, quality of service.