

OPENFLOW KOMUTATOR U SDN MREŽAMA

Datum prijave: 22.04.2015.
Datum prihvaćanja: 29.05.2015.

UDK: 004.45
Pregledni rad

E. Ciriković, dipl.ing.

Visoka škola Virovitica

Matije Gupca 78, 33000 Virovitica, Republika Hrvatska

Telefon: +38533722037 Fax: +38533722099 E-mail: enes.cirikovic@vsmti.hr

SAŽETAK - OpenFlow kao široko prihvaćena tehnologija koncepta softverski upravljanih mreža svojim je kapacitetima omogućila realizaciju novih ideja u primjeni mrežnih aplikacija i ostalih strukturnih komponenata mrežnih arhitektura na razini istraživačke zajednice, a sve češće i u profesionalnim IT krugovima. Razdvajanjem upravljačkog i podatkovnog sloja u do tada vertikalno integriranim mrežnim arhitekturama osnova je softverski upravljanih mreža i kao takva je prirodan „partner“ OpenFlow specifikaciji kojom su definirana formalna pravila za realizaciju komunikacijskih protokola te izdvajanja upravljačke logike mrežnih uređaja iz hardverskog okruženja u neovisne softverske komponente. Rezultat primjene navedenih koncepata ogleda se u pojednostavljenom upravljanju mrežama u cjelini kao i specifičnim konfiguracijama inače programabilno ograničenih ili potpuno zatvorenih hardverskih ekvivalenata mrežnih uređaja. Tu su također i pojednostavljene mogućnosti virtualizacije računalnih mreža i podatkovnih centara te primjene novih sigurnosnih koncepata. U radu je prikazan osnovni koncept softverski upravljanih mreža te arhitektura OpenFlow protokola i komutatora kao dijela istoga koncepta. Na kraju je predstavljen jednostavan primjer implementacije OpenFlow Open vSwitch komutatora u testnoj topologiji Mininet softverskog mrežnog emulatora te OpenDaylight upravljačkog programa.

Ključne riječi: softverski upravljane mreže, OpenFlow, virtualni komutator

SUMMARY- As a widely accepted part of Software-Defined Networking, OpenFlow technology provides well defined platform for researchers and IT professionals implementing new ideas in network applications and its core components. Separation of control plane from data plane in traditional vertical network architectures enables perfect match with OpenFlow specification which defines formal rules for interlayer communication protocols, devices and hardware independent control logic. The benefits are simplified network management and open approach to programming network hardware behaviour as well as ease of network and data center virtualization. This paper presents basic Software-Defined Networking concepts as well as OpenFlow protocol and switch architectures. Whole concept is presented in Mininet virtual network environment with Open vSwitch software switch and OpenDaylight controller.

Keywords: Software Defined Networks, OpenFlow, virtual switch

1. UVOD

Zahtjevi za fleksibilnom upravljivošću mrežnih uređaja, neovisno o specifičnim softverskim i hardverskim rješenjima pojedinih proizvođača, najviše su došli do izražaja u softverski definiranim mrežama implementiranim kroz OpenFlow specifikaciju. Tako je posljednjih godina standard prerastao akademske uvjete korištenja i pronašao nišu među proizvođačima komercijalnih komutatora koji u svoju opremu ugrađuju OpenFlow API.

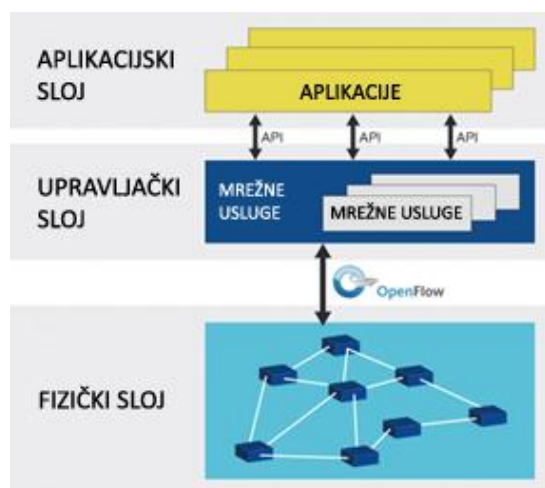
OpenFlow je otvoreni komunikacijski protokol koji služi kao sučelje za udaljeno upravljanje tablicama usmjeravanja u mrežnim uređajima kao što su bežične pristupne točke, komutatori, vatrozidi i usmjerni vači. Izvorni koncept nastao je 2008. godine na sveučilištu Stanford, a od samog početka nalazi se pod okriljem organizacije Open Networking Foundation koja se bavi promocijom softverski definiranih mreža (Software Defined Networking) putem razvoja otvorenih komunikacijskih standarda. Ideja kao takva ima potencijale za vrlo dobru implementaciju u postojećim računalnim mrežama te se može iskoristiti u istraživačke i akademske svrhe. S obzirom na softversku primjenu i povoljnu hardversku podršku današnjih računala, ovakav pristup u simulaciji novih rješenja u računalnim mrežama nalazi brojne prednosti u odnosu na neke od dosadašnjih pokušaja softverske i hardverske nadogradnje produkcijske opreme koja je

u prvom redu materijalno zahtjevna, a često i neostvariva zbog zatvorenog pristupa internoj arhitekturi takve opreme. Dok su u konvencionalnim komutatorima sve funkcije kontrole i prosljeđivanja paketa hardverski integrirane, u softverski orijentiranim rješenjima dolazi do razdvajanja razina kontrole i podataka te otvorenih mogućnosti programabilnog upravljanja. Tako je podatkovna razina i dalje pridružena preklopniku, a upravljačka razina realizirana je softverski te ju kao takvu pogoni poseban upravitelj, najčešće poseban poslužitelj, koji s podatkovnom razinom komunicira upravo putem OpenFlow protokola preko standardnih grupa poruka. Osnovni preduvjet za praktičnu realizaciju koncepta softverski definiranih mreža zadovoljeni su OpenFlow specifikacijom koja osim samog komunikacijskog protokola između SDN kontrolera i mrežnih uređaja određuje i njihovu zajedničku logičku arhitekturu. Sve skupa opisano je dokumentom *OpenFlow Switch Specification*, koji se od inačice 1.2 do trenutne 1.5 nalazi pod grupom Open Networking Foundation.

2. SOFTVERSKI DEFINIRANE MREŽE

U uvjetima eksplozije podatkovnih tokova te ekspanzije novih mrežnih aplikacija sve su veći zahtjevi na brzinu procesiranja te specifičnog upravljanja paketima u mrežnim uređajima poput usmjeritelja, komutatora, vatrozida i sl. Upravo ideja razdvajanja vertikalno integrirane mrežne infrastrukture na zasebne slojeve upravljačke logike i podatkovnog procesiranja te otvaranje mogućnosti proizvoljne programabilnosti mreža, središnji je element koncepta softverskog upravljanja mrežom, tzv. Software Defined Networking ili skraćeno SDN. U takvom sustavu, ključnu ulogu ima središnji upravljač koji je odgovoran za upravljanje uređajima pod svojom domenom, koji u tom slučaju tvore fizičku mrežnu infrastrukturu.

SLIKA 1. ARHITEKTURA SDN MREŽE



Izvor: Prilagođeno prema <https://www.opennetworking.org/images/stories/downloads/sdn-resources/white-papers/wp-sdn-newnorm.pdf>

Prema prikazanoj su arhitekturi očigledno razdvojeni upravljački od podatkovnog sloja čime je i u praktičnom smislu iz mrežnih uređaja izdvojena upravljačka funkcija te im je ostavljena operativna uloga preusmjeravanja podatkovnih paketa. Nadalje, odluke preusmjeravanja paketa transformirane su iz destinacijski orijentiranih u odluke zavisne o tokovima kao skupovima paketa s odgovarajućim pratećim vrijednosnim kriterijima i zadanim funkcijama. Sukladno tome, paketi istog toka na jednak se način interpretiraju i procesiraju u svim uređajima fizičkog sloja, čime se postiže apstrakcija tokova, kojom se omogućava generalizacija rada različitih mrežnih uređaja. Iznad fizičke komponente nalazi se tzv. mrežni operacijski sustav ili SDN upravljački sustav koji svojim resursima omogućuje programsko upravljanje uređajima kroz centralizirani pogled na mrežnu apstrakciju.

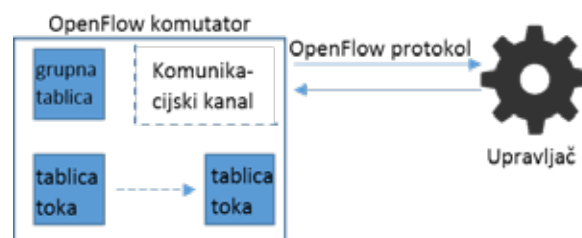
U osnovi se arhitektura prikazana slikom 1 može razmatrati kroz tri apstrakcijska sloja: prosljeđivanja, distribucije i specifikacije. Apstrakcijski sloj

prosljeđivanja, neovisan o svojstvima fizičke mrežne infrastrukture, obavlja ulogu izvršavanja podrške zahtjevima prosljeđenih od strane aplikacijske instance mrežne arhitekture. Uvid u cjelovitu sliku, inače distribuirane mrežne infrastrukture, obavlja se kroz sloj distribucije, koji uz logički centraliziranu ulogu upravljanja mrežnim uređajima te prikupljanja informacija o njihovom radu i međusobnoj povezanosti stvara jasnu osnovu za rad mrežnih aplikacija. Način na koji mrežne aplikacije realiziraju svoje primarne funkcije, bez upliva u samu implementaciju na fizičkoj razini, pripada apstrakcijskom sloju specifikacije koji na ovakve zahtjeve pruža rješenja virtualizacije te upotrebe programskih jezika.

3. OPENFLOW KOMUTATOR

Osnovna struktura OpenFlow mrežnog okruženja prikazana je slikom 2.

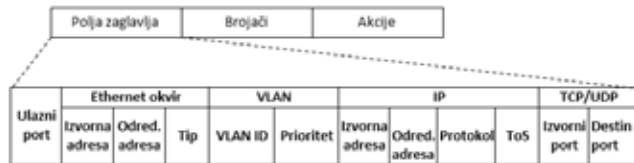
SLIKA 2. OPENFLOW MREŽNO OKRUŽENJE



Izvor: Prilagođeno prema <https://www.opennetworking.org/images/stories/downloads/sdn-resources/onf-specifications/openflow/openflow-spec-v1.4.0.pdf>

S obzirom na način izvedbe, razlikuju se dvije vrste OpenFlow preklopnika. Prvi tip zasniva se na realizaciji tablica toka i OpenFlow protokola u hardverskoj TCAM komponenti i operacijskom sustavu mrežnih usmjeritelja ili preklopnika. Drugi tip realiziran je softverski na bazi UNIX/Linux sustava. OpenFlow preklopnik u osnovi je sastavljen od jedne ili više tablica toka (flow tables) koje tvore cjevovod tablica toka i grupne tablice (group table) kojima se provode osnovne funkcije usporedbe i usmjeravanja paketa. Komunikacija s SDN kontrolerom odvija se OpenFlow protokolom putem SSL (Secure Socket Layer) protokola i sastavni je dio svakog OpenFlow preklopnika. Tablice toka (en. flow tables) uspoređuju dolazne pakete s podatkovnim tokovima te na osnovi pripadnosti primijenjuju određene funkcije. Tako se svaka tablica toka sastoji od zapisa ulaznih tokova čije nastajanje, ažuriranje ili brisanje inicira kontroler ovisno o stanju dolaznih paketa. Svaki se zapis ulaznog toka sastoji od polja za uparivanje, brojača te skupa funkcija koje se primjenjuju na pripadajuće pakete. Glavne komponente zapisa toka prikazane su slikom 3.

SLIKA 3. GLAVNE KOMPONENTE ZAPISA TOKA

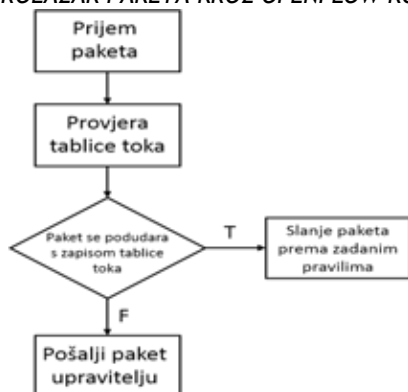


Izvor: Izradio autor

Svaki se zapis toka tablice toka sastoji od polja za usporedbu s nadolazećim paketima, koja su sastavljena od podataka o ulaznom portu dolaznog paketa, podacima izvorišne i odredišne MAC adrese, izvorišne i odredišne IP adrese te TCP i UDP porta te metapodacima preuzetih od prethodne tablice toka. Nadalje, tu su i polja za određivanje prioriteta usporedbe, te polja s brojačima koja se ažuriraju nakon svake usporedbe s dolaznim paketom, polja s zadanim instrukcijama, polje tajmera i kolačića.

Pripadnost paketa određenom toku provjerava se dolaskom do prve tablice toka i ovisno o ishodu nastavlja se s prosljeđivanjem drugim tablicama. U slučaju podudarnosti paketa s određenim tokom iniciraju se prethodno definirane specifične instrukcije, a u slučaju nepodudarnosti sa zapisima polja za uparivanje u tablicama toka provode se akcije sukladne konfiguraciji polja za nepostojeće tokove. Tako je moguće da se nepodudarajući paketi, sukladno konfiguraciji tablica toka, odbace, prosljede narednoj tablici toka ili da se putem posebne poruke vrate upravitelju na daljnju obradu. Akcije uključene u instrukcije kao sastavni dio svakog zapisa tablice toka odnose se na prosljeđivanje ili promjenu zaglavlja paketa ili daljnju obradu u grupnoj tablici toka. Drugi tip instrukcija odnose se na obradu u cjevovodu tablica tokova a uključuju prosljeđivanje paketa između tablica tokova instrukcijom *Goto-Table* te razmjenu metapodataka pri komunikaciji između tablica. Usporedba paketa s tablicama toka u cjevovodu tablica tokova zaustavlja se kada se u skupu instrukcija povezanih s poredbenim podatkovnim tokom ne nalazi pokazivač na sljedeću tablicu toka te se tada provodi preusmjeravanje ili izmjena zaglavlja samog paketa sukladno akcijama pridruženim instrukcijama pripadajućeg podatkovnog toka. Dijagram toka procesiranja paketa u OpenFlow preklopniku prikazano je slikom 4.

SLIKA 4. PROLAZAK PAKETA KROZ OPENFLOW KOMUTATOR



Izvor: Izradio autor

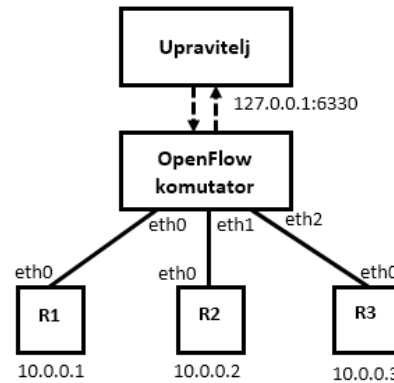
3. OPEN vSWITCH KOMUTATOR

Praktična primjena uspostave mrežnog okruženja putem demonstracije rada OpenvSwitch OpenFlow preklopnika unutar predkonfigurirane VirtualBox virtualne mašine preuzete s <http://sdnhub.org/tutorials/sdn-tutorial-vm/> na kojoj se nalazi 64-bit Ubuntu operacijski sustav sa sljedećim predinstaliranim alatima:

- SDN kontroleri: OpenDaylight, ONOS, RYU, Floodlight, Floodlight-OF1.3, POX, i Trema
- Primjeri koda za hub i L2 pametni switch
- Open vSwitch 2.3.0 s podrškom za Openflow 1.2, 1.3 i 1.4
- Mininet alat za kreiranje virtualnih mrežnih topologija

U primjeru se koristi jednostavna topologija s jednim preklopnikom i tri računala koja nastoje ostvariti međusobnu povezivost komandom ping (slika 5).

SLIKA 5. MININET TESTNA TOPOLOGIJA

Izvor: Prilagođeno prema <http://sdnhub.org/resources/useful-mininet-setups/>

Mininet testna topologiju inicirana je unosom sljedeće komande:

```
$ sudo mn --to:po single,3 --mac --switch ovsk --controller remote
```

,gdje se parametrom *-mac*, automatski definiraju MAC adrese *s1-eth[0|1|2]*, te *h[1|2|3]-eth0* koje pripadaju svim prisutnim uređajima, odnosno pripadajućim im aktivnim sučeljima. Parametrom *-topo single, 3* određuje se mrežna topologija s jednim preklopnikom s tri spojena računala. Parametrom *-controller remote* određuje se tip upravljanja preklopnikom udaljenim upraviteljem. Način rada preklopnika određen je parametrom *-switch ovsk*, a radi se o *Open vSwitch-u* u Kernel modu.

Provjera osnovne mrežne povezivosti računala 1 i 3 na mreži, provodi se pokretanjem komande *h1 ping h3*. S obzirom da u ovome trenutku nije inicijaliziran mrežni upravitelj, jasno je da nije ostvarena mrežna povezivost što je prikazano ICMP porukom „*Destination Host Unreachable*“. Za primjer je odabran pre-

dinstalirani *OpenDaylight* kontroler za čiji je rad potrebno prethodno izgraditi Java aplikaciju unosom komande *mvn clean install* u novom terminalu na lokaciji:

```
ubuntu@sdnhubvm:~/SDNHub_Opendaylight_Tutorial/commons/parent[08:12] (master)$:
```

Nakon uspješno izgrađene *Java aplikacije*, potrebno je pokrenuti aplikaciju kontrolera komandom *./run.sh* na lokaciji:

```
ubuntu@sdnhubvm:~/SDNHub_Opendaylight_Tutorial/distribution/opendaylight-osgi-adsal/target/distribution-osgi-adsal-1.1.0-SNAPSHOT-osgipackage/opendaylight
```

U ovome trenutku uspostavljeni su svi uvjeti za povezivost računala iz primjera te se ponovnim pokretanjem komande *h1 ping h3* u prozoru terminala, gdje je pokrenuta aplikacija *mininet*, pojavljuju poruke o uspješno uspostavljenoj vezi (slika 6).

SLIKA 6. PROVJERA POVEZIVOSTI RAČUNALA 1 I 3 IZ TOPOLOGIJE SA SLIKE 5

```
mininet> h1 ping h3
PING 10.0.0.3 (10.0.0.3) 56(84) bytes of data.
64 bytes from 10.0.0.3: icmp_seq=1 ttl=64 time=16.9 ms
64 bytes from 10.0.0.3: icmp_seq=2 ttl=64 time=8.81 ms
64 bytes from 10.0.0.3: icmp_seq=3 ttl=64 time=6.43 ms
64 bytes from 10.0.0.3: icmp_seq=4 ttl=64 time=9.95 ms
^C
--- 10.0.0.3 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3009ms
rtt min/avg/max/mdev = 6.430/10.538/16.955/3.917 ms
mininet> █
```

Izvor: Izradio autor

4. ZAKLJUČAK

Potencijal *OpenFlow* specifikacije na ideji softverski upravljanih mreža dobiva sve veći zamah u vremenima ekspanzije podatkovnih tokova te uređaja na fizičkoj mrežnoj razini. Mogućnosti upravljanja takvim mrežama neovisne su o proizvođačima opreme te programabilnosti njihova ponašanja s obzirom na trenutne produkcijske uloge, te omogućuju istraživačima ali i ostalim profesionalnim korisnicima povoljnu platformu za realizaciju kako postojećih tako i novih mrežnih rješenja.

Iako ideja softverski upravljanih mreža nije nova, svoju značajniju potporu dobila je tek pojavom *OpenFlow* specifikacije kada se iz primjene u zatvorenim laboratorijskim uvjetima usko specijalizirane znanstvene akademske zajednice proširila i u komercijalne primjene. Tako je danas moguće pronaći sve veći broj produkcijske opreme priznatih proizvođača, ali i praktičnu implementaciju istih u globalnim IT kompanijama. Konačni efekt primjene softverski upravljanih mreža i *OpenFlow* protokola te uređaja na bazi istih, redukcija je troškova implementacije globaliziranog upravljanja uređajima fizičkog sloja kao i testiranja novih protokola i algoritama na baznoj osnovi stvarnih podatkovnih izvora.

LITERATURA

1. Foundation, O. N. (2012). Software-defined networking: The new norm for networks. ONF White Paper.
2. Lantz, B., Heller, B., & McKeown, N. (2010, October). A network in a laptop: rapid prototyping for software-defined networks. In Proceedings of the 9th ACM SIGCOMM Workshop on Hot Topics in Networks (p. 19). ACM.
3. Al-Somaidai, M. B., & Yahya, E. B. (2014). Survey of software components to emulate OpenFlow protocol as an SDN implementation. *American Journal of Software Engineering and Applications*, 3(1), 1-9.
4. McKeown, N., Anderson, T., Balakrishnan, H., Parulkar, G., Peterson, L., Rexford, J., ... & Turner, J. (2008). OpenFlow: enabling innovation in campus networks. *ACM SIGCOMM Computer Communication Review*, 38(2), 69-74.
5. Kim, H., & Feamster, N. (2013). Improving network management with software defined networking. *Communications Magazine, IEEE*, 51(2), 114-119.
6. Lara, A., Kolasani, A., & Ramamurthy, B. (2014). Network innovation using openflow: A survey. *Communications Surveys & Tutorials, IEEE*, 16(1), 493-512.
7. Kreutz, D., Ramos, F. M., Verissimo, P. E., Rothenberg, C. E., Azodolmolky, S., & Uhlig, S. (2015). Software-defined networking: A comprehensive survey. *proceedings of the IEEE*, 103(1), 14-76.
8. OpenFlow Switch Specification v1.4, <https://www.opennetworking.org/images/stories/downloads/sdn-resources/onf-specifications/openflow/openflow-spec-v1.4.0.pdf>