Analiza izvedbe specifikacije OpenFlow u

upravljačkom uređaju OpenDaylight

Tehnička dokumentacija

Verzija <1.0>

Studentski tim: Kristijan Dubravec

Vice Ivušić

Mario Pavić

Nastavnik: Doc. dr. sc. Ognjen Dobrijević

Sadržaj

1. Opis projektnog zadatka 3

2. Programski upravljane komunikacijske mreže 4

3. OpenFlow 6

3.1 OpenFlow specifikacija 7

4. OpenDaylight 12

5. Analiza 13

5.1 Zaglavlje 13

5.2 Struktura fizičkog porta 14

5.3 OFPT\_HELLO 16

5.4 OFTP\_FEATURES\_REQUEST 16

5.5 OFTP\_FEATURES\_REPLY 17

5.6 OFTP\_ERROR 20

5.7 OFPT\_STATS\_REQUEST 21

5.8 OFPT\_STATS\_REPLY 22

5.8.1 Desc Stats Reply 22

5.8.2 Table Stats 23

5.8.3 Port Stats 24

5.9 OFPT\_GET\_CONFIG\_REQUEST 25

5.10 OFPT\_GET\_CONFIG\_REPLY / OFPT\_SET\_CONFIG 26

5.11 OFPT\_FLOW\_MOD 27

5.11.1 OFPT\_FLOW\_MOD unutar Wiresharka 2.2.3 27

5.11.2 OFPT\_FLOW\_MOD unutar Wiresharka 1.6.7 (laboratorijsko okruženje) 29

5.12 OFPT\_PACKET\_IN 31

5.13 OFPT\_PACKET\_OUT 32

6. Zaključak 33

7. Literatura 34

8. Skraćenice 35

9. Rječnik pojmova 35

# Opis projektnog zadatka

Internet je najveća i najrasprostranjenija računalna komunikacijska mreža. 1995. godine samo je 1% svjetske populacije imalo pristup internetu, dok se danas ta statistika kreće oko 40% svjetske populacije. Međutim, iako popularnost interneta i dalje raste, postojeća infrastruktura iznimno je otporna na promjene što otežava razvoj i inovativnost u području telekomunikacijskih mreža. [5]

Programski upravljane mreže (engl. Software-Defined Network, skr. SDN) nude rješenje problema tromosti mreža. Arhitektura SDN-a podijeljena je na kontrolnu razinu i podatkovnu razinu; kontrolnu razinu ostvaruje središnji upravljački uređaj dok je podatkovna razina podijeljena između mrežnih uređaja. Odvajajući logiku prosljeđivanja od samih mrežnih uređaja u zasebni upravljački uređaj, čime se zadatak mrežnih uređaja svodi jedino na prosljeđivanje tokova podataka, omogućuje se razvoj logike prosljeđivanja potpuno neovisno o konkretnoj izvedbi sklopovlja o kojemu mreža ovisi. Upravljački uređaj i mrežni uređaji moraju komunicirati; kako bi komunikacija bila moguća, format i sadržaj poruka mora biti dobro definiran i standardiziran. Jedan od najpopularnijih protokola za komunikaciju između navedenih uređaja je OpenFlow protokol, koji je opisan OpenFlow specifikacijom. [2][3]

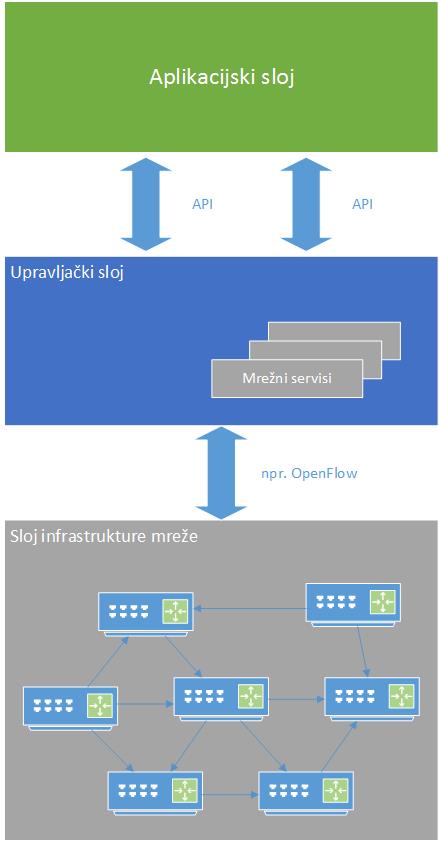
OpenFlow specifikacija sadrži mnoštvo zahtjeva od kojih neki moraju, ali neki i ne moraju biti implementirani unutar konkretnog programskog rješenja. Programska platforma otvorenog koda koja realizira jedan upravljački uređaj, te koja je korištena u sklopu ovog projekta, je OpenDaylight. Zadatak projekta je analizirati poruke poslane između OpenDaylighta i SDN mrežnih uređaja unutar laboratorijskog okruženja na Zavodu za telekomunikacije kako bi se utvrdilo do koje su mjere funkcionalnosti opisane OpenFlow specifikacijom implementirane unutar samog OpenDaylighta.

# Programski upravljane komunikacijske mreže

U tradicionalnim mrežama poput Interneta, kao najveća mreža današnjega doba, podatci se u paketima po mreži prenose pomoću brojnih mrežnih uređaja, ponajviše usmjernika. Usmjernik je uređaj koji donosi odluku o sljedećem skoku paketa na njegovom putu do odredišta tako što na temelju internih tablica i logike ("pameti") koju ima ugrađenu u sebi donosi odluku najboljeg puta po nekom parametru. Budući da je Internet mreža koja se stalno dinamički mijenja, put koji je u jednom trenutku bio optimalan u drugom trenutku može biti zagušen ili nedostupan. Tako prvi usmjernik koji postane svjestan tog problema mora obavijestiti sve druge usmjernike da ne koriste taj put prilikom usmjeravanja svojih paketa. Današnja je mreža sačinjena od velikog broja takvih uređaja te, zbog velike geografske udaljenosti od kraja do kraja mreže, slanje obavijesti o trenutnom statusu nekog puta može potrajati. Zbog takvog i mnogih sličnih problema na koje današnja arhitektura nema jednostavnih rješenja došlo se na ideju o drugačijoj mrežnoj arhitekturi, a to je programski-upravljiva mreža.

SDN je arhitektura mreže u kojem je funkcija odlučivanja, tj. logika ("pamet"), odvojena od funkcije prosljeđivanja koje su kod tradicionalne mreže obje smještene u usmjernik. Takvom separacijom funkcija odlučivanja može se fizički odvojiti od funkcije prosljeđivanja. Funkcija odlučivanja se u ovakvim arhitekturama prepušta upravljačkom uređaju.

Upravljački uređaj upravlja mrežom i predstavlja "pamet" mrežnim uređajima koji će obavljati funkciju prosljeđivanja. SDN ne definira da će u konkretnoj implementaciji postojati samo jedan upravljački uređaj (što bi za današnje mreže, s obzirom na to da se one sastoje od velikog broja takvih uređaja, bilo prezahtjevno za jedan mrežni uređaj) koji će upravljati cijelom mrežom već specificira samo da će se funkcija odlučivanja i prosljeđivanja odvojiti jedna od druge, a konkretnu ideju u vezi postojanja više upravljačkih uređaja zaduženih za neku mrežu ostavlja konkretnoj implementaciji da to riješi na svoj način. Tako se dopušta potpuna programibilnost upravljačkog uređaja i olakšavanje ugradnje drugačijeg načina usmjeravanja od dosadašnjeg (što je pomoću IP adrese i MAC adrese). Tako da istraživači i znanstvenici koji se bave osmišljavanjem novog načina po kojem bi se paketi usmjeravali po mreži jednostavnom programskom ugradnjom u upravljački uređaj imaju mogućnost korištenja drugačijih parametara po kojima bi algoritmi za usmjeravanje donosili odluke prilikom odlučivanja. [11][12]

Mrežni uređaji u SDN arhitekturi koji obavljaju funkciju prosljeđivanja daleko su jednostavniji od mrežnih uređaja koji imaju tu istu funkciju u tradicionalnoj mreži zbog toga što oni nemaju direktno u sebi ugrađenu logiku ("pamet") već je ta logika centralizirana u mrežnom uređaju koji onda donosi odluku što napraviti sa svakim paketom, dok takvi uređaji slijepo slijede odluku koju je upravljački uređaj donio. Krivo razmišljanje bilo bi da za svaki primljeni paket takav usmjernik mora pitati upravljački uređaj što napraviti s njima. Na takav bi način upravljački uređaj bio prezatrpan odlučivanjem u mreži za koju je zadužen i to ne bi bilo dobro. U konkretnoj implementaciji može biti izvedeno da takvi usmjernici imaju tablicu u kojoj imaju unaprijed pohranjene odluke za različite pakete pa usmjernik ako naiđe paket koji odgovara jednom od redaka te tablice izvrši odluku koja je definirana u tom retku, a ako usmjernik ne naiđe na zapis u tablici koji odgovara primljenom paketu kontaktira upravljački uređaj kako bi mu on rekao što napraviti s njim. Tako da bi upravljački uređaj određivao putanje za nove pakete i upisivao odluke u tablice usmjernika preko kojeg će ti paketi ići ili mijenjao konkretnu tablicu nekog usmjernika kako bi promijenio dotadašnju putanju nekog paketa u slučaju zagušenja nekog puta. [10]

Slika 1 SDN slojevi

Budući da upravljački uređaj komunicira sa svim usmjernicima mreže, on ima moć da, ako otkrije da neki put postaje zagušen ili nedostupan, ugradi nove odluke u usmjernike tako da se taj put zaobiđe i time poboljša optimalnost puteva kojim podatci prolaze po mreži. Način na koji bi upravljački uređaj došao do tih saznanja se ostavlja konkretnoj implementaciji da izvede na najbolji mogući način.

Organizacija koja promovira SDN kao način upravljanja mrežom je ONF (engl. Open networking foundation). Pod njihovim okriljem je razvijen OpenFlow standard koji su razvili i unaprjeđuju dan danas. OpenFlow standard predstavlja jedan od načina na koji bi se ideja SDN mogla implementirati te koju su mnogi distributeri mrežne opreme prihvatili i implementirali u svojim uređajima. OpenFlow je prvi SDN standard – time je pokazano da se SDN ideja može implementirati i koristiti u današnjem svijetu. [4]

# OpenFlow

Cilj programski-upravljane mreže je omogućiti administratorima mreže i "oblaka" mehanizme kako bi mogli brzo reagirati na potrebite promjene u mreži pomoću centralnog upravljačkog uređaja. SDN obuhvaća razne vrste mrežnih tehnologija dizajnirane kako bi učinile mrežu fleksibilnijom i bržom. Takav način upravljanja mrežom podupire uvelike rad virtualnih servera i infrastrukturu pohrane kod modernih podatkovnih centara. [3]

Često je mišljenje da je OpenFlow sinonim za SDN, ali to je samo jedan od elemenata cjelokupne SDN arhitekture. OpenFlow je, ugrubo, komunikacijski protokol otvorenog standarda koji omogućuje da upravljački uređaj može komunicirati s prosljeđujućom jedinicom.

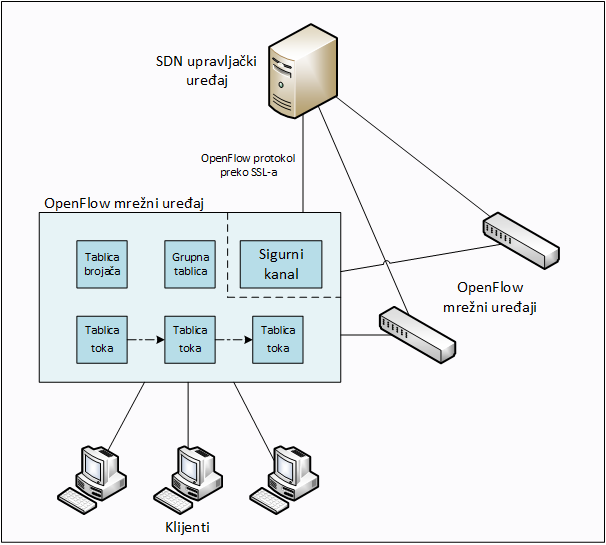
Osnovna ideja OpenFlowa je da se može spojiti više mrežnih uređaja ili čak više mreža kako bi se stvorio jedinstveni tok (engl. *flow*) – u biti VLAN na steroidima. Tada bi se moglo prema potrebi upravljati čitavom infrastrukturom, postavkama i prometom mreže. Sâm pojam toka nije dobro definiran unutar OpenFlow specifikacije. Semantički, tok je niz paketa koji putuju mrežom, a koji dijele isti skup vrijednosti unutar polja zaglavlja. Npr. jedan tok može se sastojati od svih paketa s jednakom izvorišnom i odredišnom IP adresom.

Kako je SDN tržište još u razvoju, mnoge su implementacije SDN-a konfigurirane kao nadogradnja na postojeće mreže i mnoge od trenutnih komercijalnih primjena OpenFlowa orijentirane su na proizvodnju poboljšanih verzija postojećih mrežnih usluga koristeći OpenFlow model. Trebat će vremena da mrežno tržište u potpunosti usvoji potencijal SDN-a i integrira centralizirani pristup SDN-a u njegove metode i zahtjeve. [8]

Svaka nova verzija OpenFlowa donosi nove korisne mogućnosti koje proširuju opseg i mogućnosti SDN-a. Međutim, ove nove mogućnosti mijenjaju i način kako se problemi rješavaju u podatkovnim centrima. Umjesto izgradnje mreža s fiksnim arhitekturama koje su predviđene za velika opterećenja, resursi podatkovnih centara mogu biti optimizirani preko cijele mreže, pa čak i preko domena. Prva verzija OpenFlow specifikacije – 1.0 – razvijena je na Sveučilištu Stanford. ONF je ubrzo uzeo projekt pod svoje okrilje te je nastavio razvijati specifikaciju. OpenFlow 1.3 proširuje funkcionalnosti specifikacije i vjerojatno će služiti kao temelj na koji će se buduće komercijalne implementacije oslanjati; stoga, ONF ne planira mnogo promjena u doglednoj budućnosti. [3]

Ranije primjene OpenFlowa često koriste njegove kapacitete upravljanja tokom podataka kako bi preusmjerili promet prije nego li stigne do jezgre same mreže – u biti optimizirajući rukovanje s paketima na njenim rubnim dijelovima. Tokovi se mogu spajati ili razdvajati te tako omogućuju navođenje i sigurnost podataka, kao i to da se promet koncentrira kroz manje mrežnih čvorišta što smanjuje operativne troškove te smanjuje troškove za električnu energiju i klimatizaciju.

## OpenFlow specifikacija

Kako bi se koncept SDN-a preveo u praktičnu implementaciju, dva uvjeta moraju biti zadovoljena. Prvi uvjet zahtijeva zajedničku logičku arhitekturu svih mrežnih uređaja kojima upravlja upravljački uređaj. Sama logička arhitektura može biti implementirana na različite načine u različitim uređajima; bitno je da je sučelje prema upravljačkom uređaju jednako za svaki mrežni uređaj. Drugi uvjet traži da se za komunikaciju između mrežnih uređaja koristi standardizirani, sigurni protokol. Oba uvjeta ispunjava OpenFlow koji, osim što služi kao protokol između mrežnih uređaja, također služi kao specifikacija logičke strukture svih funkcionalnosti mrežnih uređaja. 

Upravljački uređaj komunicira s OpenFlow kompatibilnim mrežnim uređajima koristeći OpenFlow protokol koji se izvodi povrh SSL-a (engl. *Secure Sockets Layer*). Svaki mrežni uređaj spojen je s drugim mrežnim uređajima i, po mogućnosti, s klijentima koji su izvor i odredište tokova podataka. Unutar svakog mrežnog uređaja nalaze se tablice tokova kojima se upravlja tokovima koji prolaze mrežnim uređajima. [3]

Slika 2 Struktura OpenFlow okruženja

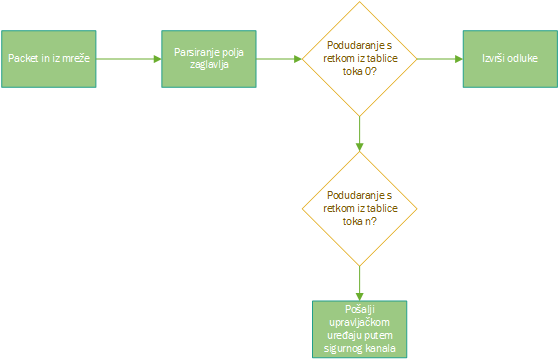
OpenFlow specifikacija definira tri vrste tablica unutar logičke arhitekture mrežnih uređaja. Tablica toka (engl. *flow table*) koristi se za identificiranje poruka koji pripadaju određenom toku i navodi akcije koje se moraju izvesti nad porukama. Moguće je izvesti akcije koje navode dvije ili više tablica toka, povezane u obliku cjevovoda. Također, tablica toka može usmjeriti tok grupnoj tablici (engl. *group table*) koja može izazvati broj drugih akcija koje se izvode nad jednim tokom, ili više njih. Postoje i tablice brojača (engl. *meter table*) koje mogu izazvati različite akcije nad tokom koje utječu na performanse.

OpenFlow protokol opisuje poruke koje upravljački uređaj i mrežni uređaji međusobno razmjenjuju. Protokol se obično implementira povrh SSL-a ili TSL-a (engl. *Transport Layer Security*) čime se osigurava kanal.

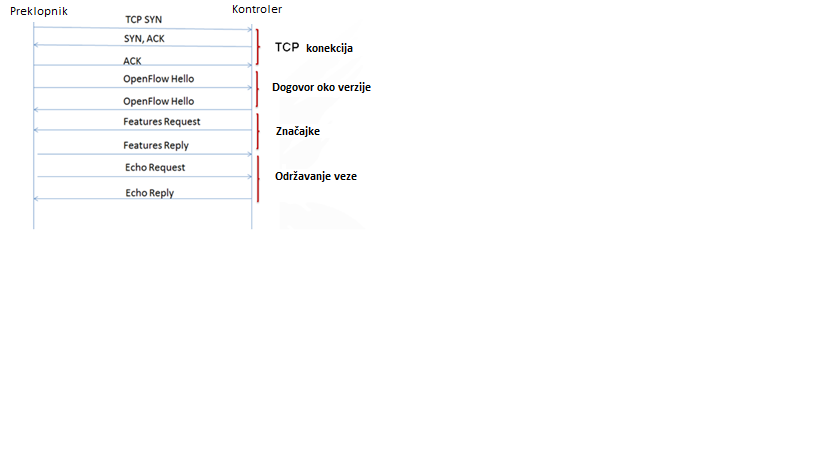
Polja svake poruke koja pristigne u OpenFlow mrežni uređaj uspoređuju se s poljima za usporedbu odgovarajućih redaka unutar tablice toka. Ako postoje odluke koje se moraju izvršiti, one se izvršavaju (npr. prosljeđivanje poruke na određeni port na temelju odredišne IP adrese). Ukoliko nije određena nijedna odluka koju mrežni uređaj mora izvršiti na temelju tablice toka, mrežni uređaj enkapsulira poruku i prosljeđuje ju upravljačkom uređaju koji analizira poruku i stvara pripadajuću tablicu toka za mrežni uređaj.

Tablica toka sastoji se od polja za zaglavlja, polja za brojače i polja za odluke koje se moraju izvršiti. Polja za zaglavlja koriste se kako bi se usporedile i pronašle one poruke koje se podudaraju s podacima u poljima. Svako polje sadržava konkretnu vrijednost, vrijednost s maskom ili oznaku da može doći bilo koja vrijednost (engl. *wildcard*). [1]

Brojači se koriste za relevantne statistike konkretne tablice toka. Svaki redak unutar tablice toka može imati nula ili više radnji koje govore komutatoru što napraviti s podudarajućim porukama.

OpenFlow protokol podržava tri vrste poruka: *controller-to-switch*, asinkrone i simetrične poruke. *Controller-to-switch­* poruke šalje upravljački uređaj kako bi izravno upravljao ili nadzirao stanje mrežnog uređaja. One mogu, ali ne moraju, zahtijevati odgovor od mrežnog uređaja. Ova grupa poruka omogućuje upravljačkom uređaju upravljanje logičkim stanjem i konfiguracijom mrežnog uređaja, kao i upravljanje recima tablica toka te grupnih tablica. Asinkrone poruke šalje komutator kako bi upravljačkom uređaju dostavio ažurirane podatke o događajima u mreži, poput pristizanja poruka, i o promijenjenom stanju samog komutatora. Simetrične poruke šalju i upravljački uređaj i komutator, npr. kad razmjenjuju *Hello* poruke pri uspostavi mreže.

Slika 3 Dijagram toka obrade paketa u OpenFlow mrežnom uređaju

Pri uspostavi OpenFlow veze, sudionici razmjenjuju *Hello* poruke u kojoj navode koja je najnovija verzija OpenFlow protokola koju podržavaju. Na taj način se određuje koja će se verzija OpenFlow protokola koristiti za daljnju komunikaciju, odnosno odabire se ona koja je najmanja zajednička uređajima.

Slika 4 Izmjena poruka između preklopnika i kontrolera

# OpenDaylight

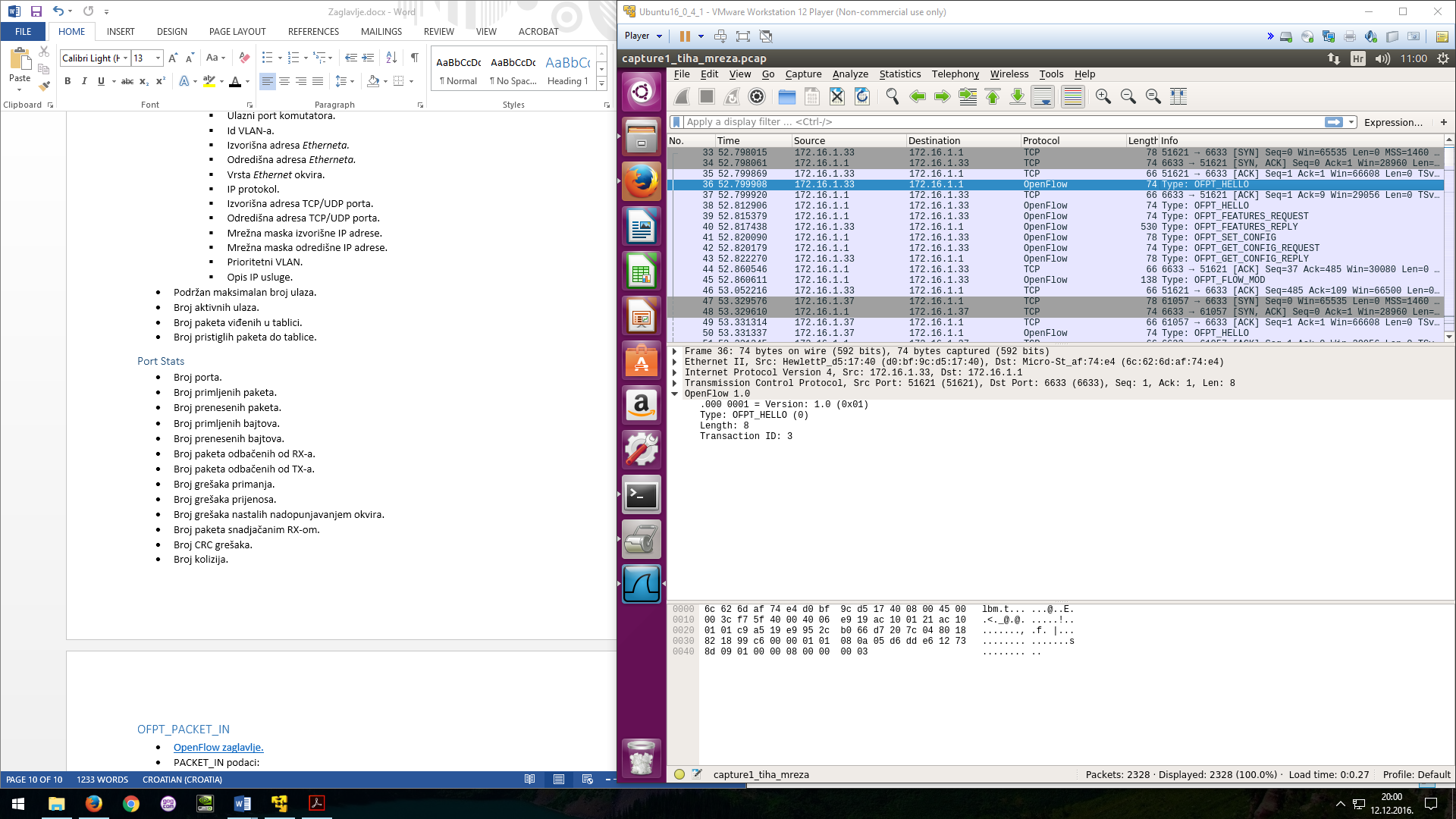
OpenDaylight je modularna SDN programska platforma otvorenog koda napisana u programskom jeziku Java. Projekt je osnovan 2013. godine s potporom tvrtki, među kojima spadaju Cisco, Brocade, Hewlett Packard, Ericsson, Microsoft i brojne druge, s ciljem da se ubrza usvajanje SDN tehnologija te da se sa što većom transparentnošću razvijaju nove tehnologije. Imenovanje izdanja prati periodičku tablicu elemenata – tako je prvo izdanje, *Hydrogen*, izašlo u veljači 2014., dok je najnovije izdanje u trenutku pisanja ovog dokumenta *Boron*. U laboratorijskom okruženju na Zavodu za telekomunikacije upravo se koristi osnovno izdanje OpenDaylighta, tj. *Hydrogen*.

OpenDaylight nudi podršku za veliki broj kako standardnih tako i nadolazećih SDN protokola, od kojih je za projekt najvažniji OpenFlow protokol. Također pruža apstraktnu platformu usluga zasnovanu na modelu koja omogućuje korisnicima da pišu aplikacije koje mogu lako raditi na raznolikom hardveru i *south-bound* protokolima. Ostvaruje ulogu SDN upravljačkog uređaja koji je zadužen za kompleksne radnje poput usmjeravanja poruka i sigurnosnih provjera. Kako bi te radnje bile moguće, koriste se OpenFlow poruke definirane OpenFlow specifikacijom. [9]

# Analiza

OpenDaylight i mrežni uređaji u laboratorijskoj mreži razmjenjuju OpenFlow poruke koje su definirane verzijom 1.0 OpenFlow specifikacije. Analizom poruka koje razmjenjuju OpenDaylight i mrežni uređaji pomoću alata Wireshark, ustanovit će se koji su svi zahtjevi specifikacije implementirani unutar OpenDaylighta. Sve poruke snimljene unutar Wiresharka 1.6.7 snimljene su na računalu u laboratorijskom okruženju na Zavodu za telekomunikacije.

## Zaglavlje

Zaglavlje OpenFlow poruke sadrži najelementarnije informacije o poruci.

Slika 5 Zaglavlje OpenFlow poruke unutar Wiresharka 1.6.7.

Svaka OpenFlow poruka počinje sa zaglavljem sa sljedećom strukturom:

* Verzija OpenFlow protokola koja se koristi.
* Vrsta poruke.
* Veličina poruke.  
  – Zaglavlje i tijelo poruke čine cijelu poruku. Veličina poruke izražena je u oktetima.
* Transakcijski identifikacijski broj.  
  – Služi za uparivanje poruka. Tipičan par poruka sastoji se od poruke koja zahtijeva neku informaciju te poruke koja šalje nazad traženu informaciju.

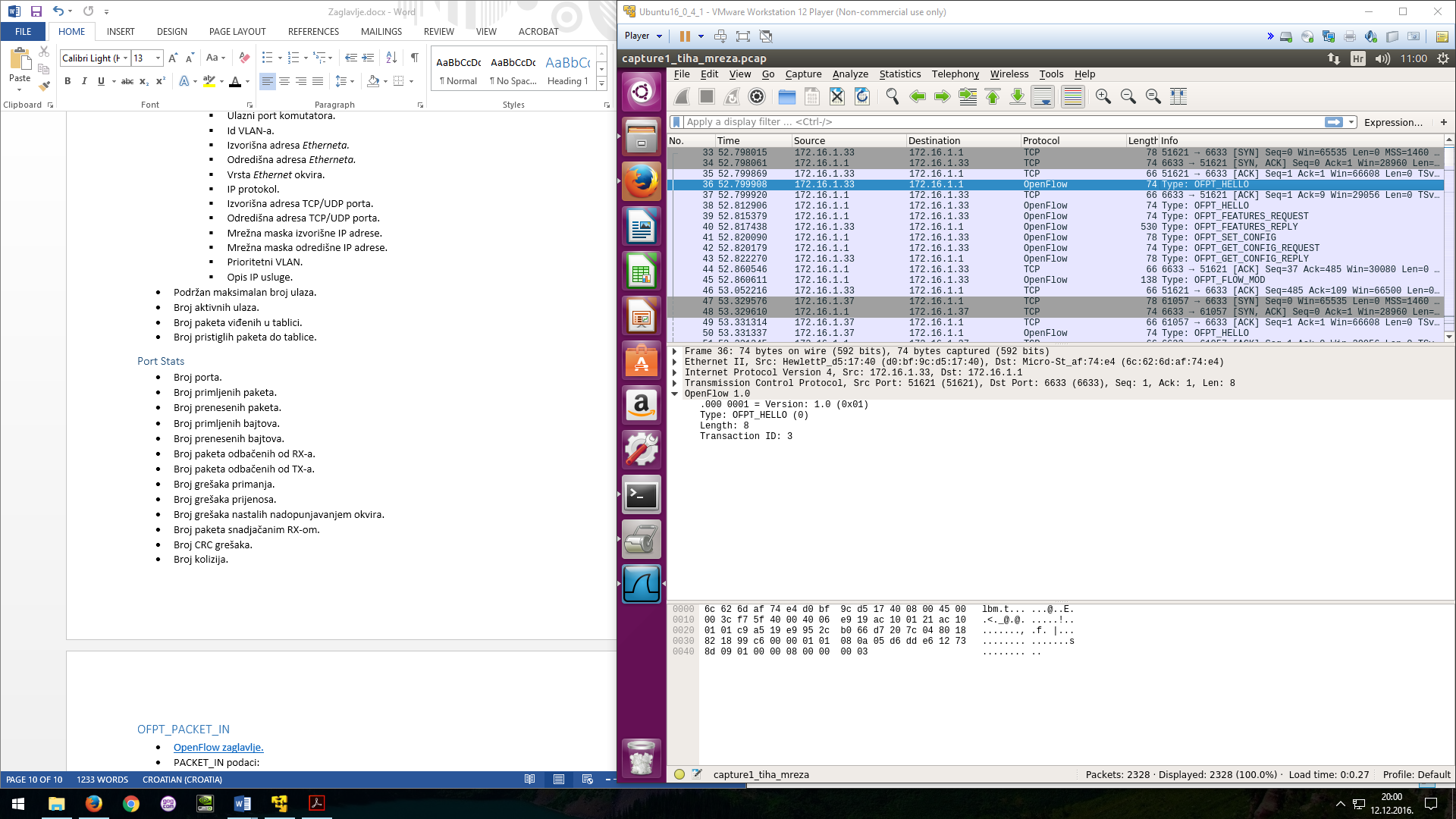
## Struktura fizičkog porta

Slika 6 Struktura fizičkog porta unutar Wiresharka 1.6.7.

Fizički port opisan je sljedećom strukturom:

* Identifikacijski broj porta.  
  – Služi za povezivanje puta podataka s fizičkim portom.
* MAC adresa porta.
* Ime porta.  
  – U čitljivom formatu; namijenjeno korisnicima.
* Ponašajna bitmapa. Sljedeće karakteristike vrijede kada je odgovarajući bit postavljen u jedinicu:
  + Port je zatvoren.
  + Onemogućen je *spanning tree* protokol.
  + Sve se poruke, osim onih poslanih *spanning tree* protokolom, odbacuju.
  + Sve se poruke poslane *spanning tree* protokolom odbacuju.
  + Port neće biti uključen prilikom preplavljivanja mreže.
  + Sve se poruke odbacuju.
  + Ne šalju se *packet-in­* poruke za poruke koje stižu na port.
* Bitmapa stanja. Sljedeće karakteristike vrijede kada je odgovarajući bit postavljen u jedinicu:
  + Prisutan je fizički podatkovni sloj.
  + (\*) Specifikacija navodi zastavice koje opisuju trenutno stanje porta tijekom izvedbe *spanning tree* protokola.
* Bitmapa mogućnosti. Sljedeće karakteristike vrijede kada je odgovarajući bit postavljen u jedinicu:
  + Podržan je 10 Mb *half-duplex*.
  + Podržan je 10 Mb *full-duplex*.
  + Podržan je 100 Mb *half-duplex*.
  + Podržan je 100 Mb *full-duplex*.
  + Podržan je 1 Gb *half-duplex*.
  + Podržan je 1 Gb *full-duplex*.
  + Podržan je 10 Gb *full-duplex*.
  + Podržan je bakreni prijenosni medij,
  + Podržana je optička niti.
  + Podržano je automatsko dogovaranje.
  + Podržano je pauziranje.
  + Podržano je asimetrično pauziranje.
* Polje prikazanih mogućnosti.  
  – Označava koje su mogućnosti vidljive vanjskom klijentu.
* Polje podržanih mogućnosti.  
  – Označava koje je mogućnosti port u stanju podržati.
* Polje mogućnosti porta s kojim je uspostavljena veza.  
  – Označava koje su mogućnosti vidljive od strane porta s kojim je uspostavljena veza.

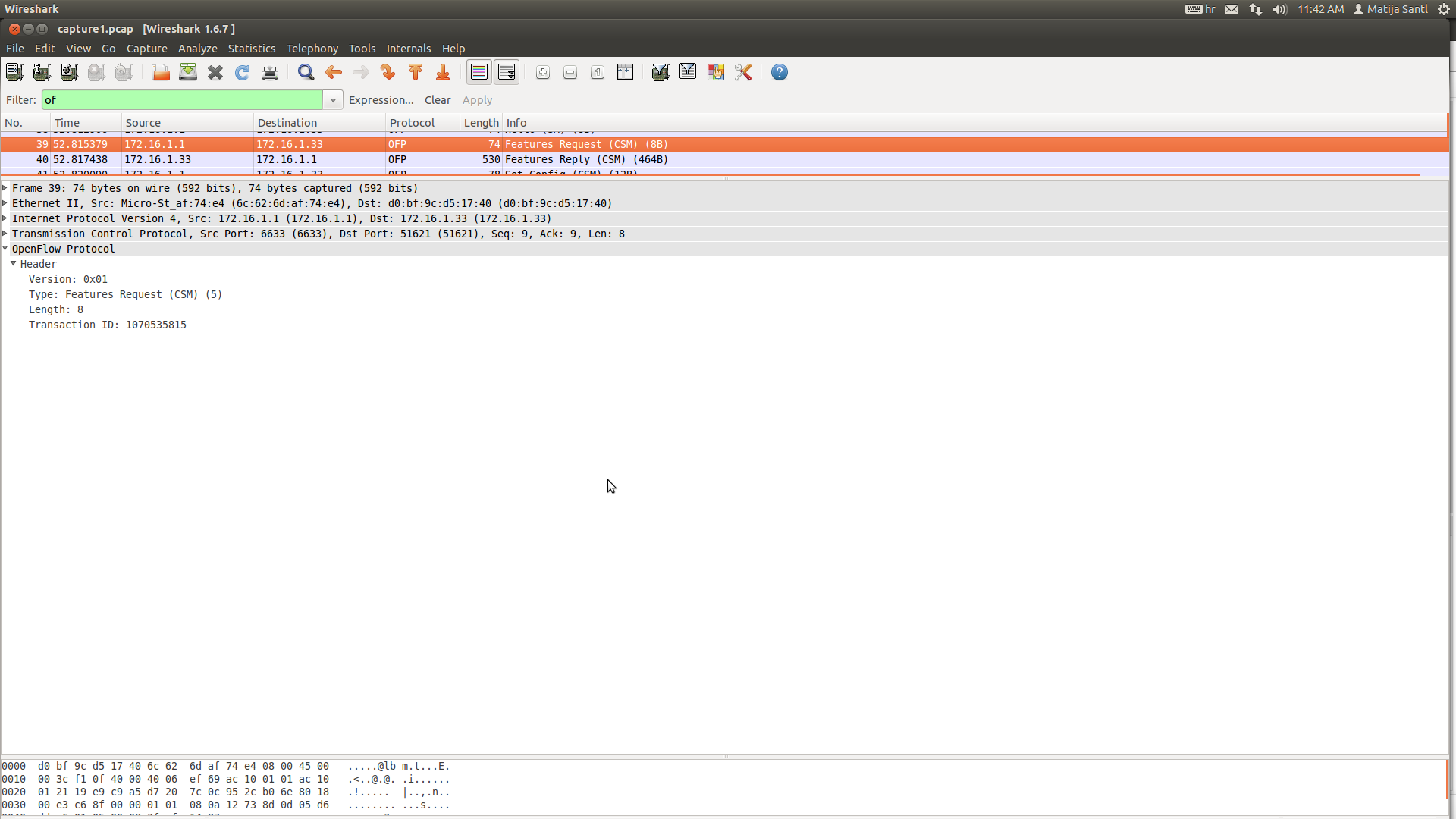
## OFPT\_HELLO

OFTP\_HELLO poruke izmjenjuju se između upravljačkog uređaja i mrežnog uređaja pri uspostavi veze. Spada pod simetrične poruke.

Slika 7 OFPT\_HELLO unutar Wiresharka 1.6.7.

Poruka se sastoji samo od [OpenFlow zaglavlja](#_OFPT_HELLO).

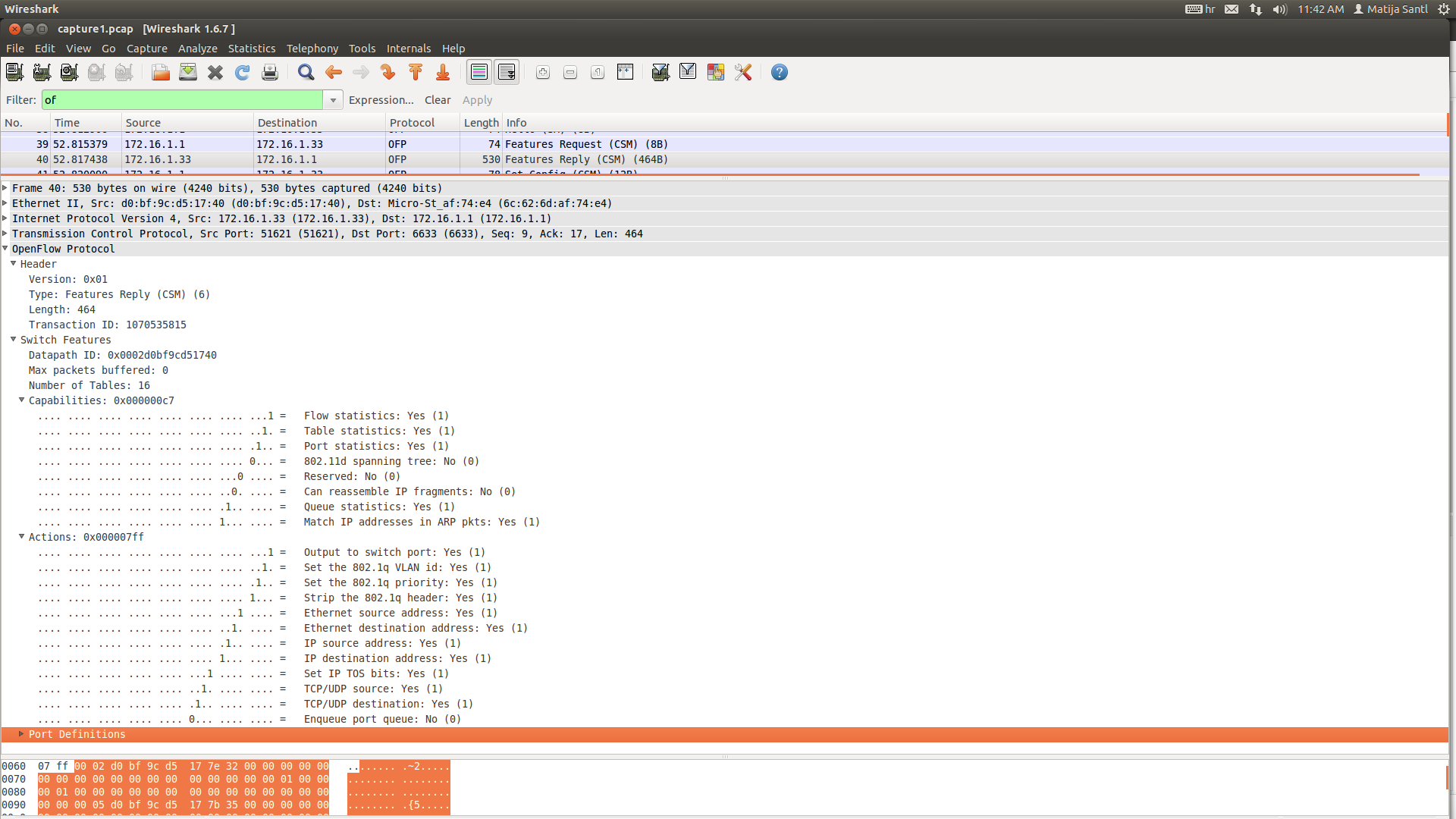
## OFTP\_FEATURES\_REQUEST

Poruka od upravljačkog uređaja mrežnom uređaju. Zahtijeva od mrežnog uređaja da mu odgovori s OFTP\_FEATURES\_REPLY porukom.

Slika 8 OFPT\_FEATURES\_REQUEST unutar Wiresharka 1.6.7.

Poruka se sastoji samo od [OpenFlow zaglavlja](#_Zaglavlje_1).

## OFTP\_FEATURES\_REPLY

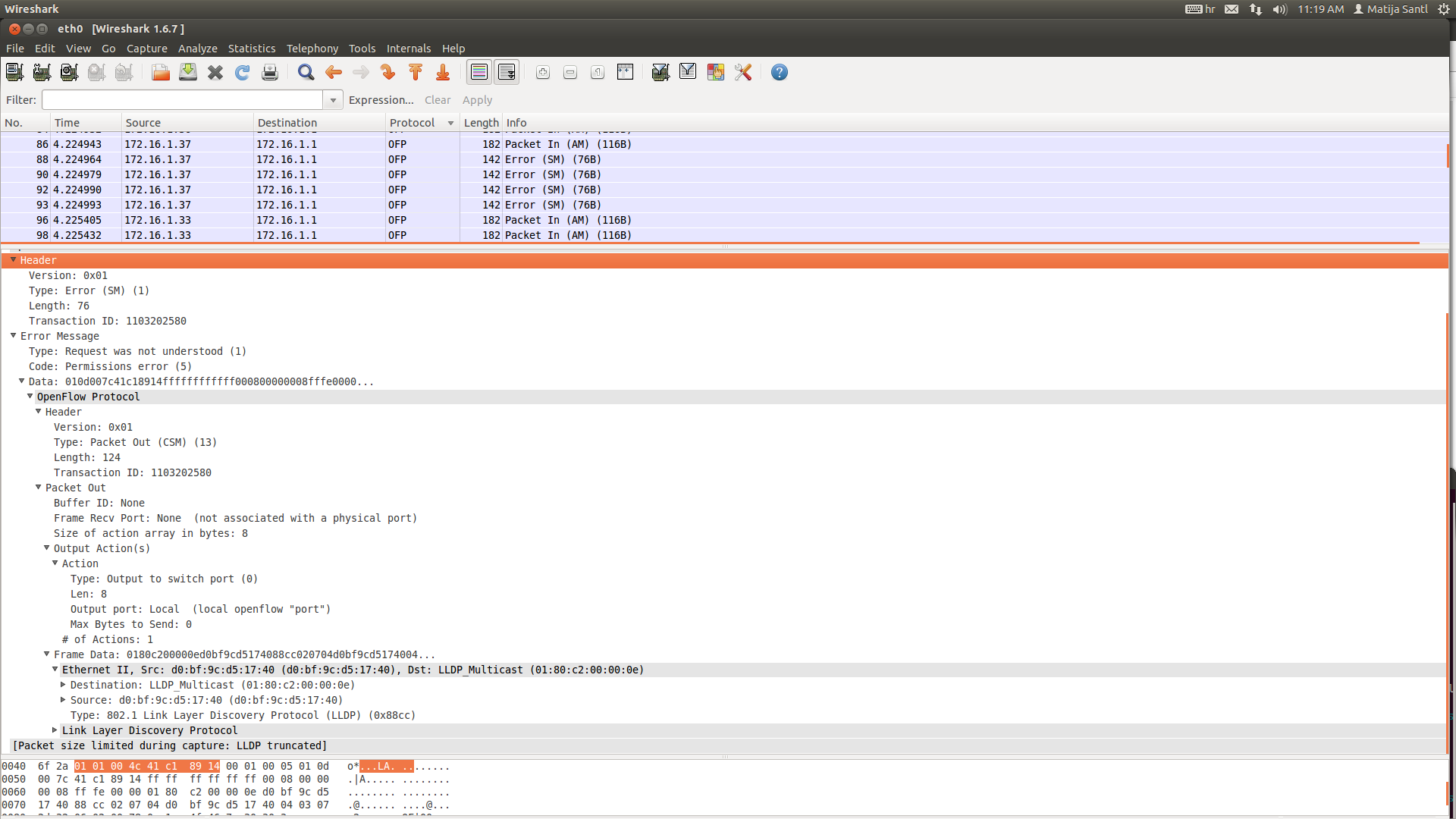
Mrežni uređaj na OFTP\_FEATURES\_REQUEST odgovara ovom porukom koja sadrži informacije o mogućnostima mrežnog uređaja.

Slika 9 OFPT\_FEATURES\_REPLY unutar Wiresharka 1.6.7.

Struktura poruke:

* [OpenFlow zaglavlje.](#_Zaglavlje)
* Mogućnosti mrežnog uređaja.
* Identifikacijski broj puta podataka.  
  – Nižih 48 bitova predstavljaju MAC adresu komutatora dok je semantika viših 16 bitova implementacijski zavisna.
* Maksimalni broj *bufferiranih* poruka.
* Maksimalni broj tablica toka.
* Bitmapa mogućnosti. Sljedeće karakteristike vrijede kada je odgovarajući bit postavljen u jedinicu:
  + - Podržano računanje statistike toka.
    - Podržano računanje statistike tablica toka.
    - Podržano računanje statistike portova.
    - Podržan je *spanning tree* protokol.
    - Bit koji mora biti nula.  
      – Ostavlja se mogućnost da se u budućim verzijama bit iskoristi u neku drugu svrhu.
    - Podržana je defragmentacija IP fragmenata.
    - Podržano računanje statistike reda čekanja (engl. *queue*).
    - Podržano podudaranje IP adresa unutar ARP poruka.
  + Bitmapa tokovnih radnji. Sljedeće karakteristike vrijede kada je odgovarajući bit postavljen u jedinicu:
    - Poruka se prosljeđuje na port.
    - Postavlja se VLAN identifikacijski broj.  
      – Ako ne postoji VLAN s danim identifikacijskim brojem, stvara se novo zaglavlje sa specificiranim identifikacijskim brojem i prioritetom od nula. Ako već postoji odgovarajuće VLAN zaglavlje, identifikacijski broj tog zaglavlja zamjenjuje se s predanom vrijednošću.
    - Postavlja se VLAN prioritet.  
      – Ako ne postoji VLAN s danim identifikacijskim brojem, dodaje se novo zaglavlje s predanim prioritetom i VLAN identifikacijskim brojem od nula. Ako VLAN zaglavlje već postoji, polje prioriteta zamjenjuje se s novim.
    - Briše se VLAN zaglavlje.
    - Izmjenjuje se izvorišna Ethernet MAC adresa.
    - Izmjenjuje se odredišna Ethernet MAC adresa.
    - Izmjenjuje se izvorišna IP adresa.
    - Izmjenjuje se odredišna IP adresa.
    - Postavljaju se IP ToS (Type of Service) bitovi.
    - Izmjenjuje se izvorišni TCP/UDP port.
    - Izmjenjuje se odredišni TCP/UDP port.
    - Poruka se prosljeđuje na red čekanja pripadajućeg porta.
* Definicije portova.
  + [Port 1 .. n.](#_Struktura_fizičkog_porta)

## OFTP\_ERROR

Poruka poslana od strane mrežnog uređaja upravljačkom uređaju koja obavještava o nastalom problemu. Spada pod nepromjenjive i simetrične poruke.

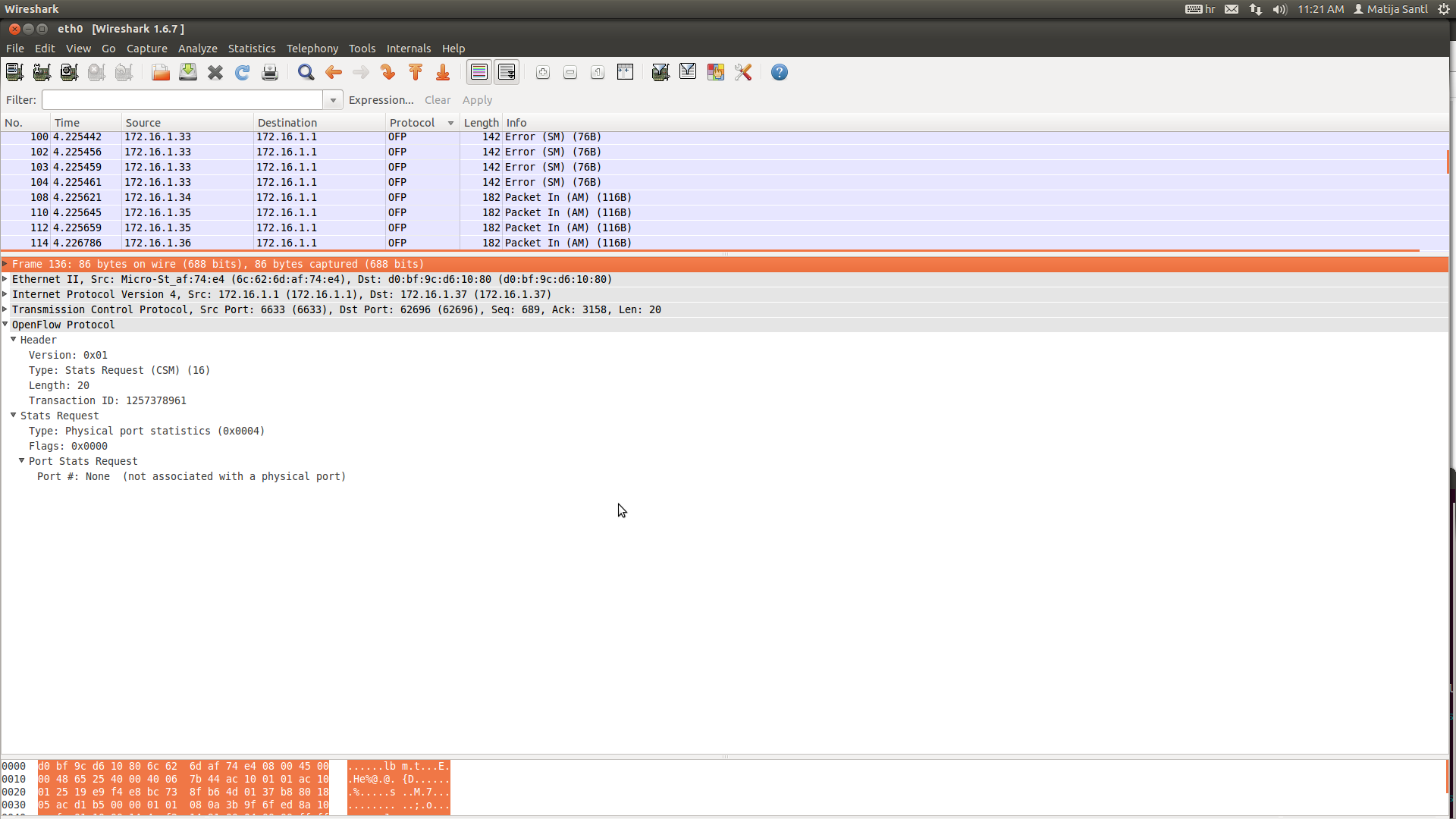
Slika 10 OFTP\_ERROR unutar Wiresharka 1.6.7.

Struktura poruke:

* [OpenFlow zaglavlje.](#_Zaglavlje)
* Vrsta poruke koja opisuje do kakve greške je došlo.
* Pobliže opisuje vrstu greške koja se dogodila, ovisno o prethodnom parametru.
* Podaci sadrže najmanje 64 okteta poruke, kod koje je došlo do greške.

## OFPT\_STATS\_REQUEST

Poruke spadaju u skupinu statičnih poruka poslanih od strane upravljačkog uređaja mrežnom uređaju. Njima se ispituje trenutno stanje toka podataka.



Slika 11 OFTP\_STATS\_REQUEST unutar Wiresharka 1.6.7.

Struktura poruke:

* [OpenFlow zaglavlje.](#_Zaglavlje)
* Određuje vrstu informacije koja se prosljeđuje i način kojim će se prikazati tijelo poruke.

– Traži se informacija o stanju na portovima.

* Zastavica nije postavljena.
* Informacije o stanju na fizičkim portovima.
  + Nije dodijeljen ni jednom fizičkom portu.

## OFPT\_STATS\_REPLY

Poslano od strane mrežnog uređaja upravljačkom uređaju kao odgovor na poruke koje zahtijevaju stanje toka podataka. Tijelo poruke se bitno razlikuje ovisno o tipu statistike koja se tražila u OFTP\_STATS\_REQUEST. Struktura poruka:

* [OpenFlow zaglavlje.](#_Zaglavlje)
* Određuje vrstu informacije koja se prosljeđuje kao odgovor i način kojim će se prikazati tijelo poruke.
* Zastavica određuju hoće li nakon ovog odgovora poruke slijediti još koji odgovor.

### Desc Stats Reply

Slika 12 OFPT\_STATS\_REQUEST za Desc Stats Reply unutar Wiresharka 1.6.7.

* Proizvođač.
* Opis *hardwarea.*
* Opis programske potpore.
* Serijski broj.
* Naziv puta podataka vidljiv korisniku.

### Table Stats

Slika 13 OFPT\_STATS\_REPLY za Table Stats unutar Wiresharka 1.6.7.

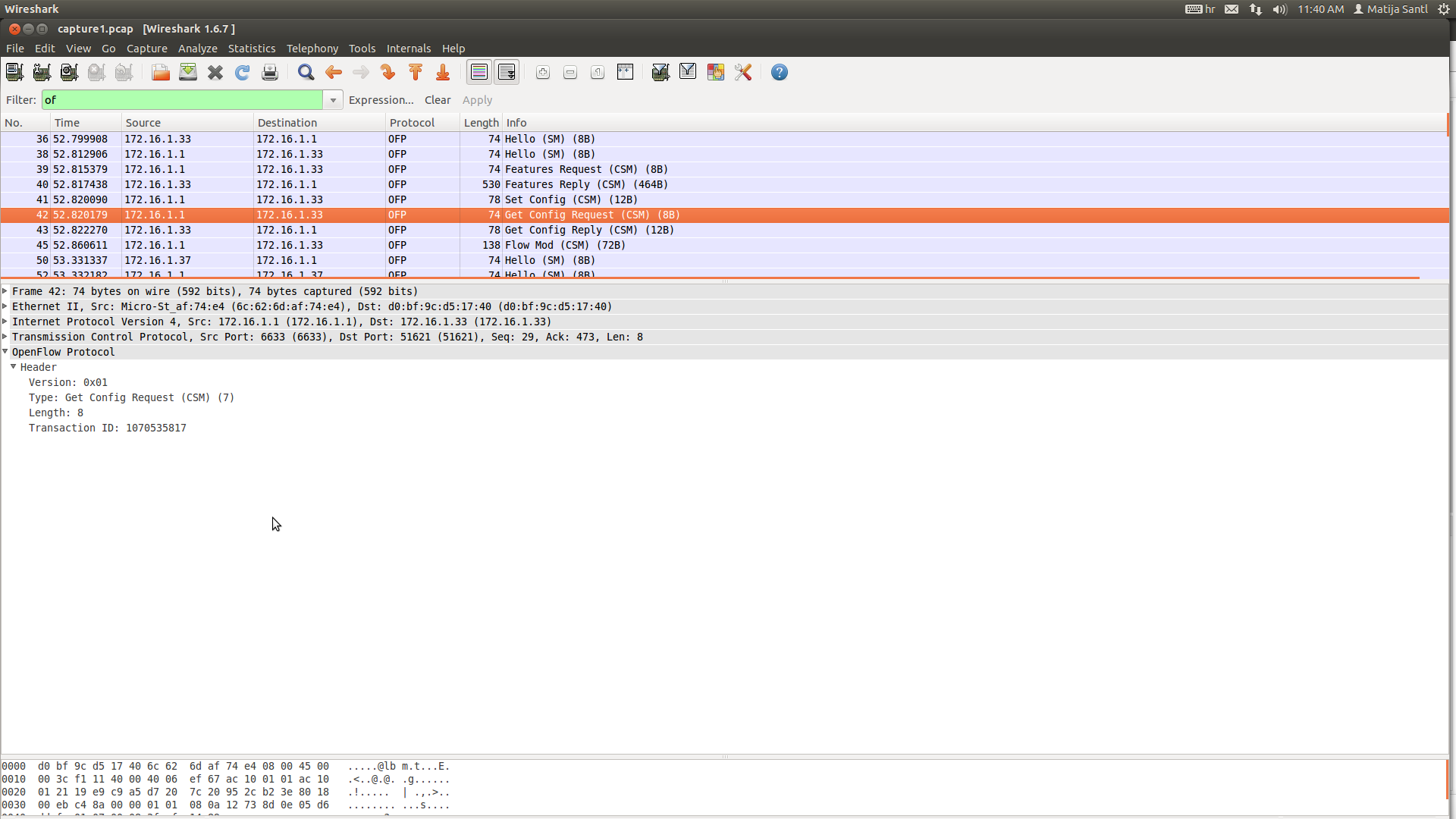
* Identifikator tablice.
* Poravnavanje do 32-bita naznačeno imenom tablice čija je to maksimalna veličina.
* Bitmapa polja za usporedbu koji su podržani u tablici:
  + Ulazni port komutatora.
  + Id VLAN-a.
  + Izvorišna adresa *Etherneta.*
  + Odredišna adresa *Etherneta.*
  + Vrsta *Ethernet* okvira.
  + IP protokol.
  + Izvorišna adresa TCP/UDP porta.
  + Odredišna adresa TCP/UDP porta.
  + Mrežna maska izvorišne IP adrese.
  + Mrežna maska odredišne IP adrese.
  + Prioritetni VLAN.
  + Opis IP usluge.
* Podržan maksimalan broj ulaza.
* Broj aktivnih ulaza.
* Broj paketa viđenih u tablici.
* Broj pristiglih paketa do tablice.

### Port Stats

Slika 14 OFPT\_STATS\_REPLY za Port Stats unutar Wiresharka 1.6.7.

* Broj porta.
* Broj primljenih paketa.
* Broj prenesenih paketa.
* Broj primljenih bajtova.
* Broj prenesenih bajtova.
* Broj paketa odbačenih od RX-a.
* Broj paketa odbačenih od TX-a.
* Broj grešaka primanja.
* Broj grešaka prijenosa.
* Broj grešaka nastalih nadopunjavanjem okvira.
* Broj paketa snadjačanim RX-om.
* Broj CRC grešaka.
* Broj kolizija.

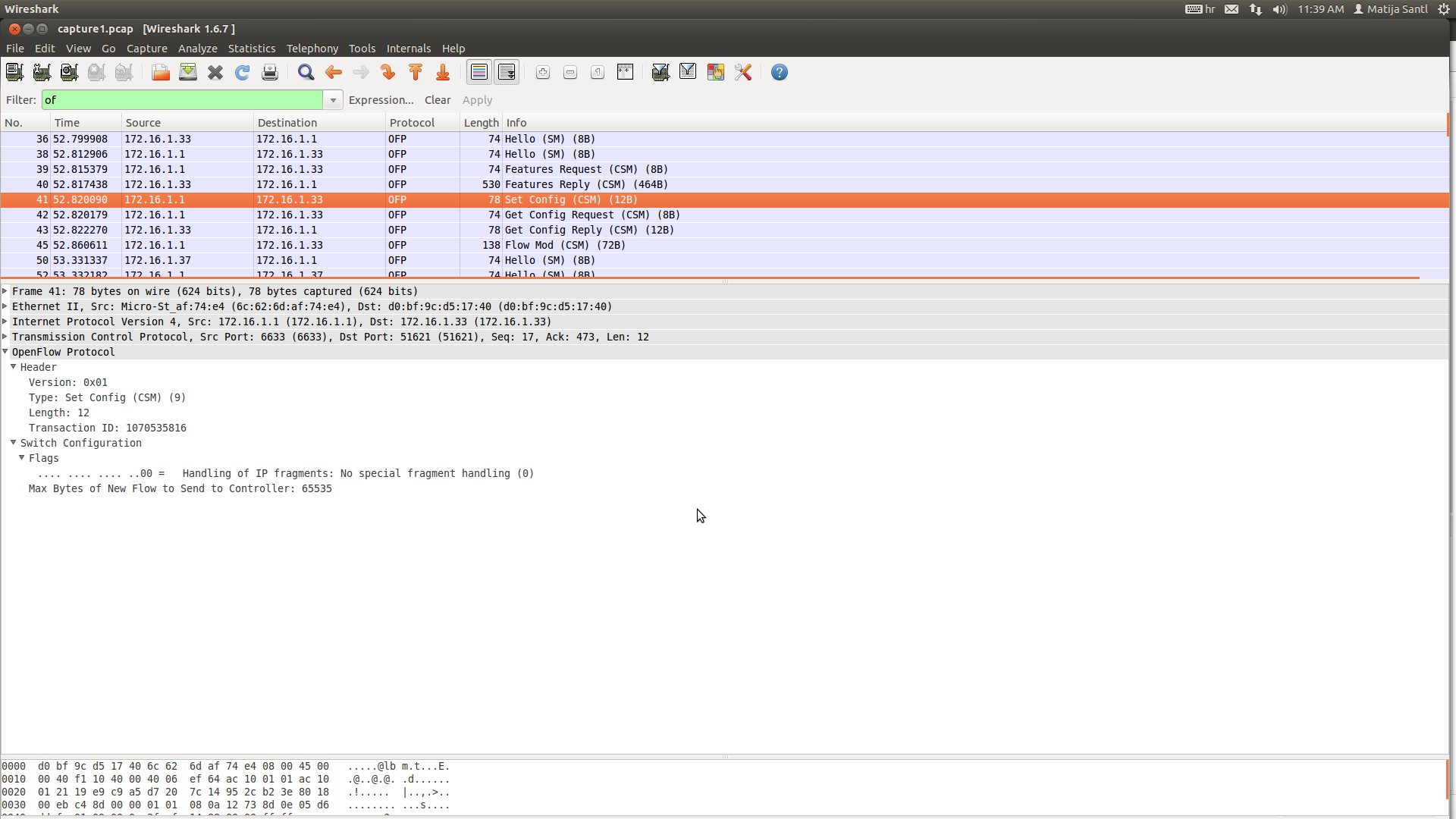
## OFPT\_GET\_CONFIG\_REQUEST

Poruka se šalje od strane mrežnog uređaja upravljačkom uređaju. Spada u skupinu poruka za konfiguraciju mrežnog uređaja.

Slika 15 OFPT\_GET\_CONFIG\_REQUEST unutar Wiresharka 1.6.7.

Poruka se sastoji samo od [OpenFlow zaglavlja](#_Zaglavlje_1).

## OFPT\_GET\_CONFIG\_REPLY / OFPT\_SET\_CONFIG

Poslano od strane kontrolera komutatoru/poslano od strane komutatora kontroleru. Spada u skupinu poruka za konfiguraciju komutatora.

Slika 16 OFPT\_GET\_CONFIG\_REPLY / OFPT\_SET\_CONFIG unutar Wiresharka 1.6.7.

Struktura poruke:

* [OpenFlow zaglavlje.](#_Zaglavlje)
* Zastavice koje određuju treba li IP fragmente tretirati normalno, odbaciti ili reasemblirati.
* Definira broj bajtova svakog paketa poslanog kontroleru koji su „pogodili“ ili „promašili“ tablicu toka, a imali su kontroler kao destinaciju.

## OFPT\_FLOW\_MOD

Spada u naredbene poruke upravljačkog uređaja. Parsiranje poruke bitno se razlikuje kad je Wireshark *screenshot* mreže otvoren u novijim verzijama Wiresharka koje imaju djelomičnu podršku za parsiranje OpenFlow poruka verzije 1.0 u odnosu na to kad je isti *screenshot* otvoren u Wiresharku 1.6.7 s instaliranim OpenFlow *dissector*-om na računalu u laboratorijskom okruženju. Zbog toga su iznesene su analize obiju verzija poruka.

### OFPT\_FLOW\_MOD unutar Wiresharka 2.2.3

Slika 17 OFPT\_FLOW\_MOD unutar Wiresharka 2.2.3.

Struktura poruke:

* [OpenFlow zaglavlje.](#_Zaglavlje)
* Polja za usporedbu sadrže:
  + Sadrži broj zastavica koje bi mogle biti postavljene.
  + Ulazni port preklopnika.
  + MAC adresa izvorišta.
  + MAC adresa odredišta.
  + Ulazni VLAN id.
  + Ulazni VLAN prioritet.
  + Nadopunjavanje do 64 bita.
  + Vrsta Ethernet okvira.
  + IP ToS (Type of Service).
  + IP protokol.
  + Nadopunjavanje do 64 bita.
  + Izvorišna IP adresa.
  + Odredišna IP adresa.
  + TCP/UDP izvorišni port.
  + TCP/UDP odredišni port.
* *Opaque controller-issued identifier.*
* Naredba koja se treba izvršit nad tokom podataka:
  + Izbriši sve podudarajuće tokove.
* Vrijeme mirovanja prije odbacivanja u sekundama.
* Maksimalno vrijeme mirovanja prije odbacivanja u sekundama.
* Razina prioriteta ulaznog toka.
* Identifikacijski broj pohranjenog paketa koji treba primijeniti.
* Izlazni port (nije povezan s fizičkim portom).
* Zastavica nisu postavljene.

### OFPT\_FLOW\_MOD unutar Wiresharka 1.6.7 (laboratorijsko okruženje)

Slika 18 OFPT\_FLOW\_MOD unutar Wiresharka 1.6.7.

Struktura poruke:

* [OpenFlow zaglavlje.](#_Zaglavlje)
* Polja za usporedbu sadrže:
  + Ulazni port komutatora.
  + Id VLAN-a.
  + Izvorišna adresa *Etherneta.*
  + Odredišna adresa *Etherneta.*
  + Vrsta *Ethernet* okvira.
  + IP protokol.
  + Izvorišna adresa TCP/UDP porta.
  + Odredišna adresa TCP/UDP porta.
  + Mrežna maska izvorišne IP adrese.
  + Mrežna maska odredišne IP adrese.
  + Prioritetni VLAN.
  + Opis IP usluge.
* *Opaque controller-issued identifier.*
* Naredba koja se treba izvršit nad tokom podataka:
  + Izbriši sve podudarajuće tokove.
* Vrijeme mirovanja prije odbacivanja u sekundama.
* Maksimalno vrijeme mirovanja prije odbacivanja u sekundama.
* Razina prioriteta ulaznog toka.
* Id pohranjenog paketa koji treba primijeniti.
* Izlazni port (nije povezan sa fizički port).
* Zastavica nisu postavljene, a mogu biti:
  + Pošalji poruku ako je tok podataka istekao ili obrisan.
  + Provjeri, ima li preklapanja prije dodavanja novog toka podataka.
  + Tok podataka se tretira kao hitan tok i koristi se za prosljeđivanje u slučaju ispada kontrolera.
* Izlazne akcije:
  + Nije određena ni jedna akcija.

## OFPT\_PACKET\_IN

Slika 19 OFPT\_PACKET\_IN unutar Wiresharka 1.6.7.

* [OpenFlow zaglavlje.](#_Zaglavlje)
* PACKET\_IN podaci:
  + ID spremnika – ID broj za identifikaciju paketa u spremniku za jednu putanju (engl. *datapath*).
  + Ukupna veličina okvira.
  + Port sa kojeg je primljen okvir.
  + Razlog zbog kojeg je paket poslan (jedan od OFPT\_\*).
  + Polje za podatke koji se prenose.

## OFPT\_PACKET\_OUT

Slika 20 OFPT\_PACKET\_OUT unutar Wiresharka 1.6.7.

* [OpenFlow zaglavlje.](#_Zaglavlje)
* PACKET\_OUT podaci:
* ID spremnika – ID broj za identifikaciju paketa u spremniku za jednu putanju (engl. *datapath*).
* Port s kojeg je primljen okvir.
* Veličina polja za akcije u bajtovima.
* Polje za akcije:
  + Zaglavlje za akcije:
    - Tip akcije – 16 bita.
    - Duljina akcije uključujući i zaglavlje – 16 bita.
    - 4 \* 8 bitova za konkretnu akciju.
* Polje za podatke koji se prenose.

# Zaključak

SDN implementiran pomoću OpenFlowa omogućuje novi pristup upravljanju kompleksnim mrežama nezavisno od proizvođača mrežnih uređaja. Moguća je programibilnost upravljačkog uređaja te konfiguriranje drugačijeg načina usmjeravanja osim onog zasnovanog na IP i MAC adresama. Vrlo je vjerojatno da će postati često korištene tehnologije unutar velikih telekomunikacijskih mreža i "cloud" infrastruktura.

Kroz samu analizu OpenFlow poruka može se utvrditi da je OpenFlow skoro u potpunosti implementiran prema specifikaciji unutar OpenDaylighta.

# Literatura

[1] Open Networking Foundation, *OpenFlow Switch Specification version 1.0.0*, 2009.

[2] N. McKeown et al., *OpenFlow: Enabling Innovation in Campus Networks*, dostupno na <http://ccr.sigcomm.org/online/files/p69-v38n2n-mckeown.pdf>

[3] W. Stallings, *Software-Defined Networks and OpenFlow*, dostupno na <http://www.cisco.com/web/about/ac123/ac147/archived_issues/ipj_16-1/ipj_16-1.pdf>

[4] O SDN-u i OpenFlowu na <http://www.opennetworking.org/>

[5] Podaci o korisnicima interneta na <http://www.internetlivestats.com/internet-users/>

[6] O razinama mreže na <https://www.opendaylight.org/news/blogs/2016/11/software-defined-networking-fundamentals-part-1-intro-networking-planes>

[7] O SDN-u na <http://www.sdxcentral.com/sdn/definitions/what-the-definition-of-software-defined-networking-sdn/>

[8] NoviFlow, *Use Cases for high capacity OpenFlow switching at the intelligent edge*, 2014. dostupno na <http://noviflow.com/resource/sdn-at-the-intelligent-edge/>

[9] O OpenDaylightu na <http://wiki.opendaylight.org>

[10] Open Networking Foundation, *SDN Architecture Overview*, 2013. dostupno na <https://www.opennetworking.org/images/stories/downloads/sdn-resources/technical-reports/SDN-architecture-overview-1.0.pdf>

[11] O SDN-u na <http://www.radio-electronics.com/info/telecommunications_networks/sdn-software-defined-networking/basics-tutorial.php>

[12] Open Networking Foundation, *SDN Security Considerations in the Data Center*m, 2013. dostupno na <http://www.opennetworking.org/images/stories/downloads/sdn-resources/solution-briefs/sb-security-data-center.pdf>

# Skraćenice

IP – *Internet Protocol*

MAC – *Media Access Control*

SDN – *Software Defined Network*

ONF – *Open Networking Foundation*

SSL – *Secure Sockets Layer*

TSL – *Transport Layer Security*

API – *Application Programming Interface*

ARP – *Address Resolution Protocol*

TCP – *Transmission Control Protocol*

UDP – *User Datagram Protocol*

VLAN – *Virtual Local Area Network*

QoS – *Quality of Service*

ToS – *Type of Service*

# Rječnik pojmova

SDN –arhitektura mreže u kojem je funkcija odlučivanja, tj. logika ("pamet"), odvojena od funkcije prosljeđivanja koje su kod tradicionalne mreže obje smještene u usmjernik.

OpenFlow – otvoreni standard koji se koristi za implementaciju komunikacijskog kanala između upravljačkog uređaj i mrežnih uređaja u SDN mreži.

OpenDaylight - OpenDaylight je modularna SDN programska platforma otvorenog koda napisana u programskom jeziku Java.

Java – objektno orijentirani programski jezik visoke razine.

**Popis slika**

[Slika 1 SDN slojevi 5](file:///F:\Desktop\FER2_Projekt_Tehnicka_Dokumentacija.docx#_Toc472075300)

[Slika 2 Struktura OpenFlow okruženja 8](file:///F:\Desktop\FER2_Projekt_Tehnicka_Dokumentacija.docx#_Toc472075301)

[Slika 3 Dijagram toka obrade paketa u OpenFlow mrežnom uređaju 10](file:///F:\Desktop\FER2_Projekt_Tehnicka_Dokumentacija.docx#_Toc472075302)

[Slika 4 Izmjena poruka između preklopnika i kontrolera 11](file:///F:\Desktop\FER2_Projekt_Tehnicka_Dokumentacija.docx#_Toc472075303)

[Slika 5 Zaglavlje OpenFlow poruke unutar Wiresharka 1.6.7. 13](file:///F:\Desktop\FER2_Projekt_Tehnicka_Dokumentacija.docx#_Toc472075304)

[Slika 6 Struktura fizičkog porta unutar Wiresharka 1.6.7. 14](file:///F:\Desktop\FER2_Projekt_Tehnicka_Dokumentacija.docx#_Toc472075305)

[Slika 7 OFPT\_HELLO unutar Wiresharka 1.6.7. 16](file:///F:\Desktop\FER2_Projekt_Tehnicka_Dokumentacija.docx#_Toc472075306)

[Slika 8 OFPT\_FEATURES\_REQUEST unutar Wiresharka 1.6.7. 16](file:///F:\Desktop\FER2_Projekt_Tehnicka_Dokumentacija.docx#_Toc472075307)

[Slika 9 OFPT\_FEATURES\_REPLY unutar Wiresharka 1.6.7. 17](file:///F:\Desktop\FER2_Projekt_Tehnicka_Dokumentacija.docx#_Toc472075308)

[Slika 10 OFTP\_ERROR unutar Wiresharka 1.6.7. 20](file:///F:\Desktop\FER2_Projekt_Tehnicka_Dokumentacija.docx#_Toc472075309)

[Slika 11 OFTP\_STATS\_REQUEST unutar Wiresharka 1.6.7. 21](file:///F:\Desktop\FER2_Projekt_Tehnicka_Dokumentacija.docx#_Toc472075310)

[Slika 12 OFPT\_STATS\_REQUEST za Desc Stats Reply unutar Wiresharka 1.6.7. 22](file:///F:\Desktop\FER2_Projekt_Tehnicka_Dokumentacija.docx#_Toc472075311)

[Slika 13 OFPT\_STATS\_REPLY za Table Stats unutar Wiresharka 1.6.7. 23](file:///F:\Desktop\FER2_Projekt_Tehnicka_Dokumentacija.docx#_Toc472075312)

[Slika 14 OFPT\_STATS\_REPLY za Port Stats unutar Wiresharka 1.6.7. 24](file:///F:\Desktop\FER2_Projekt_Tehnicka_Dokumentacija.docx#_Toc472075313)

[Slika 15 OFPT\_GET\_CONFIG\_REQUEST unutar Wiresharka 1.6.7. 25](file:///F:\Desktop\FER2_Projekt_Tehnicka_Dokumentacija.docx#_Toc472075314)

[Slika 16 OFPT\_GET\_CONFIG\_REPLY / OFPT\_SET\_CONFIG unutar Wiresharka 1.6.7. 26](file:///F:\Desktop\FER2_Projekt_Tehnicka_Dokumentacija.docx#_Toc472075315)

[Slika 17 OFPT\_FLOW\_MOD unutar Wiresharka 2.2.3. 27](file:///F:\Desktop\FER2_Projekt_Tehnicka_Dokumentacija.docx#_Toc472075316)

[Slika 18 OFPT\_FLOW\_MOD unutar Wiresharka 1.6.7. 29](file:///F:\Desktop\FER2_Projekt_Tehnicka_Dokumentacija.docx#_Toc472075317)

[Slika 19 OFPT\_PACKET\_IN unutar Wiresharka 1.6.7. 31](file:///F:\Desktop\FER2_Projekt_Tehnicka_Dokumentacija.docx#_Toc472075318)

[Slika 20 OFPT\_PACKET\_OUT unutar Wiresharka 1.6.7. 32](file:///F:\Desktop\FER2_Projekt_Tehnicka_Dokumentacija.docx#_Toc472075319)