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 specifikacija 6

OpenDaylight 8

# 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.

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 te kako bi komunikacija bila moguća, format i sadržaj poruka mora biti dobro definiran i standardiziran. Jedna od najpopularnijih specifikacija za protokol između tih uređaja je OpenFlow specifikacija.

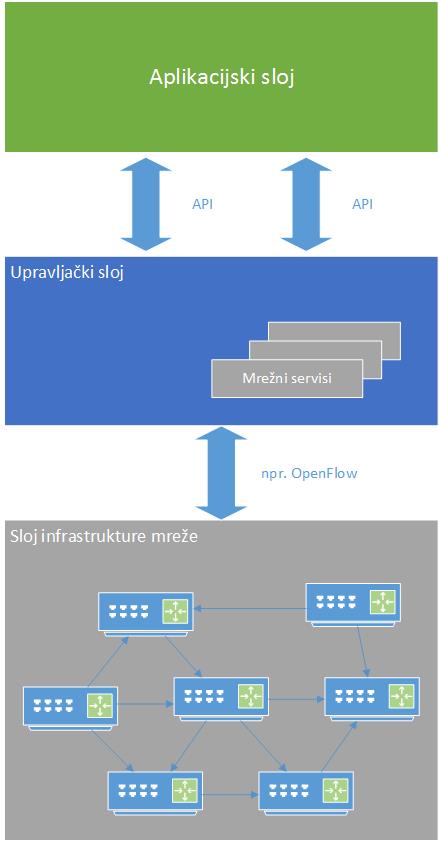
OpenFlow specifikacija sadrži mnoštvo zahtjeva od kojih neki moraju, ali neki i ne moraju biti implementirani unutar konkretnog programskog rješenja. Programska platforma otvorenog koda koja realizira jedan upravljački uređaj, te koja je korištena u sklopu ovog projekta, je OpenDaylight. Zadatak projekta 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 kontroleru.

Kontroler je uređaj koji upravlja mrežom i predstavlja "pamet" uređajima koji će obavljati funkciju prosljeđivanja. SDN ne definira da će u konkretnoj implementaciji postojati samo jedan kontroler (što bi za današnje mreže, s obzirom na to da se one sastoje od velikog broja takvih uređaja, bilo prezahtjevno za jednog kontrolera) koji će upravljati cijelom mrežom već specifira samo da će se funkcija odlučivanja i prosljeđivanja odvojiti jedna od druge, a konkretnu ideju u vezi postojanja više kontrolera zaduženih za neku mrežu ostavlja konkretnoj implementaciji da to riješi na svoj način. Tako se dopušta potpuna programabilnost kontrolera 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 kontroler imaju mogućnost korištenja drugačijih parametara po kojima bi algoritmi za usmjeravanje donosili odluke prilikom odlučivanja.

Uređaji u SDN arhitekturi koji obavljaju funkciju prosljeđivanja daleko su jednostavniji od 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 kontroler koji onda donosi odluku što napraviti sa svakim paketom, dok takvi uređaji slijepo slijede odluku koju je kontroler donio. Krivo razmišljanje bilo bi da za svaki primljeni paket takav usmjernik mora pitati kontrolera što napraviti s njima. Na takav bi način kontroler 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 kontrolera kako bi mu on rekao što napraviti s njim. Tako da bi kontroler 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.

Slika 1 SDN slojevi

Budući da kontroler komunicira sa svim usmjeriterljima 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 kontroler 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.

# 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 nudi aplikacijska sučelja (API) za komunikaciju s klijentima (fizički uređaji i virtualni komutatori). Time 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.

# OpenFlow

Cilj Programski-upravljane mreže je omogućiti administratorima mreže i „oblaka“ kako bi mogli brzo reagirati na potrebite promjene u mreži pomoću centralne kontrolne jedinice. SDN obuhvaća razne vrste mrežnih tehnologija dizajnirane kako bi učinile mrežu fleksibilnijom i bržom. Takav način upravljanja mrežom podupire uveliko rad virtualnih servera i infrastrukturu pohrane kod modernih data centara.

Često ljudi pomisle 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 kontrolna jedinica može komunicirati s prosljeđujućom jedinicom.

The basic idea with OpenFlow is you can connect multiple switches and even networks together to create a flow - essentially VLAN on steroids. You can then manage an entire infrastructure, setting policies and managing traffic type accordingly.

Osnovna ideja OpenFLowa je da se može spojiti više switchewa ili čak više mreža kako bi se stvorio jedinstveni flow(tok podataka) – u biti VLAN na sterodima. Tada bi se moglo prema potrebi upravljati čitavom infrastrukturom, postavkama i prometom mreže.

As the SDN market is still in development, many implementations of SDN/OpenFlow are configured as overlays on existing networks, and many of the current commercial deployment of OpenFlow are oriented towards providing improved versions of existing network services using an OpenFlow model. It will take time for the networking market to fully assimilate the potential of SDN and integrate the centralized intelligence approach of SDN/OpenFlow into their methods and requirements.

Kako je SDN tržište još u razvoju, mnoge implementacije SDN-a su konfigurirane kao nadogradnja na postojeće mreže i mnogi od trenutnih komercijalnih primjena OpenFLowa su orijentirani 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.

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 data centra mogu biti optimizirani preko cijele mreže, pa čak i preko domena.

Early adopters of OpenFlow are often using its flow management capabilities to redirect traffic before it reaches the core network, in essence optimizing the handling of packets at the edge. Flows can be aggregated or segregated to enable multi-homing and data security policies, as well as to concentrate traffic into fewer network nodes, reducing overprovisioning, operating costs, and even reducing power consumption and air conditioning costs.

Rane primjene OpenFLowa često koriste njegove kapacitete upravljanja tokom podataka kako bi presumjerili promet prije nego 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, smanjujući operativne troškove, pa čak i smanjenje troškova električne energije i klimatizacije.

OpenDaylight is a highly available, modular, extensible, scalable and multi-protocol controller infrastructure built for SDN deployments on modern heterogeneous multi-vendor networks. OpenDaylight provides a model-driven service abstraction platform that allows users to write apps that easily work across a wide variety of hardware and south-bound protocols.

OpenDaylight je vrlo dostupna, modularna, proširiva i prilagodljiva infrastruktura kontrolera izgrađena za SDN implementacije na suvremenim heterogenim mrežama. OpenDaylight pruža apstraktnu platformu usluga zasnovanu na modelu koja omogućuje korisnicima da pišu aplikacije koje mogu lako raditi na raznolikom hardweru i south-bound protokolima.

Slika prikazuje izmjenu poruka izmđu preklopnika i kontrolera:

## Detalji OpenFlow specifikacije

OpenFlow komutator sastoji se od tablica tokova (engl. *flow table*) prema kojima se poruke uspoređuju i dalje prosljeđuju te od sigurnog kanala koji je spojen s vanjskim upravljačkim uređajem tj., u kontekstu projektnog zadatka, s OpenDaylightom. Upravljački uređaj upravlja komutatorom preko sigurnog kanala koristeći se OpenFlow protokolom.

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

Tablica toka sastoji se od polja za zaglavlja, polja za brojače i polja za radnje. 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, maskiranu vrijednost ili *wildcard* vrijednost.

[staviti tablicu vrijednosti za polja zaglavlja]

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.

[opisati podudaranje i staviti dijagram toka]

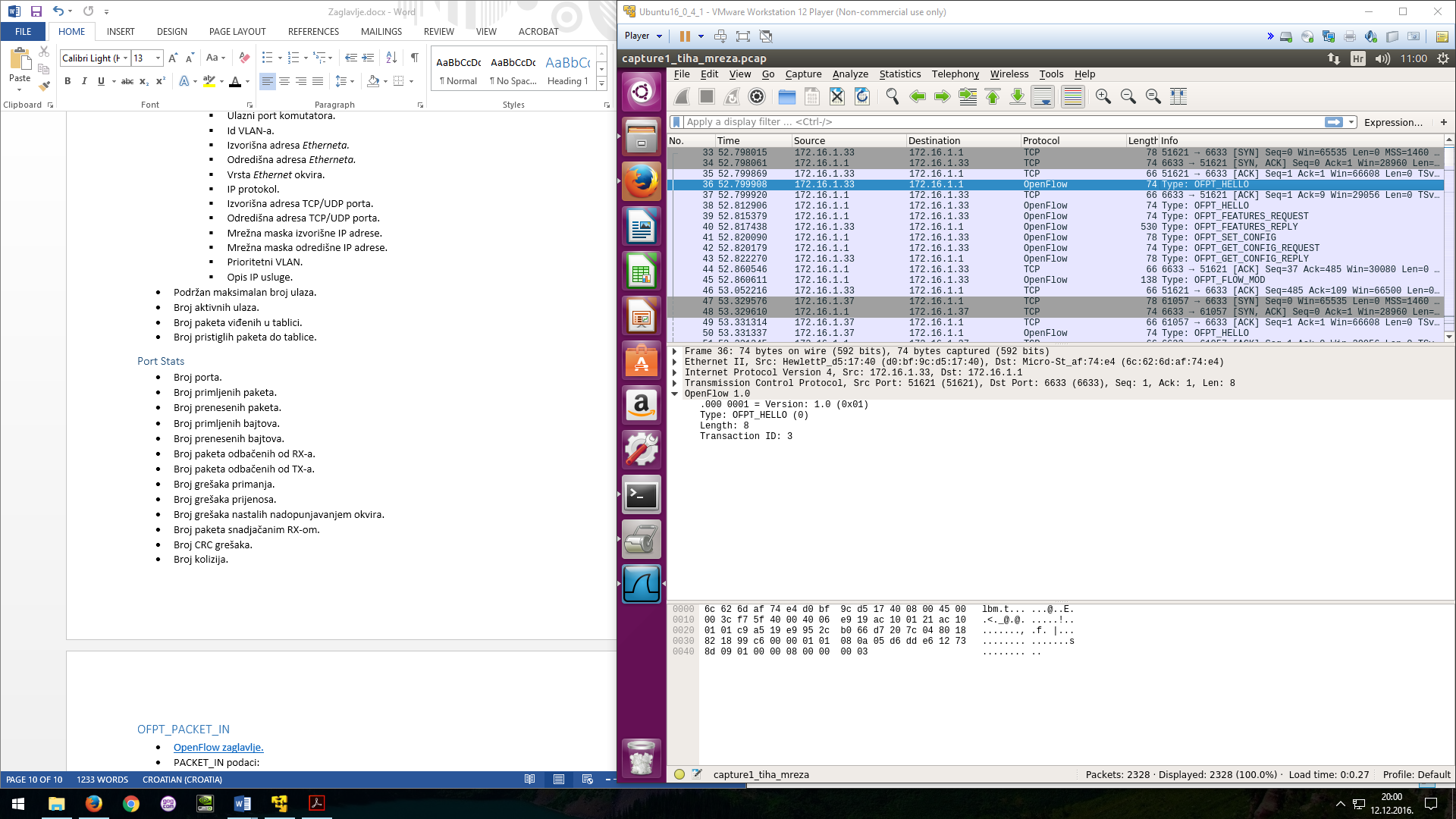
OpenFlow protokol podržava tri vrste poruka: *controller-to-switch*, asinkrone i simetrične poruke. *Controller-to-switch­* poruke šalje upravljački uređaj kako bi izravno upravljao ili nadzirao stanje komutatora. One mogu, ali ne moraju, zahtijevati odgovor od komutatora. Asinkrone poruke šalje komutator 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.

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.

# Analiza

OpenDaylight i mrežni uređaji u laboratorijskoj mreži razmjenjuju OpenFlow poruke koje su definirane verzijom 1.0 OpenFlow specifikacije.

## Zaglavlje

Zaglavlje OpenFlow poruke sadrži najelementarnije informacije o poruci.

Svaka OpenFlow poruka počinje sa zaglavljem sa sljedećom strukturom:

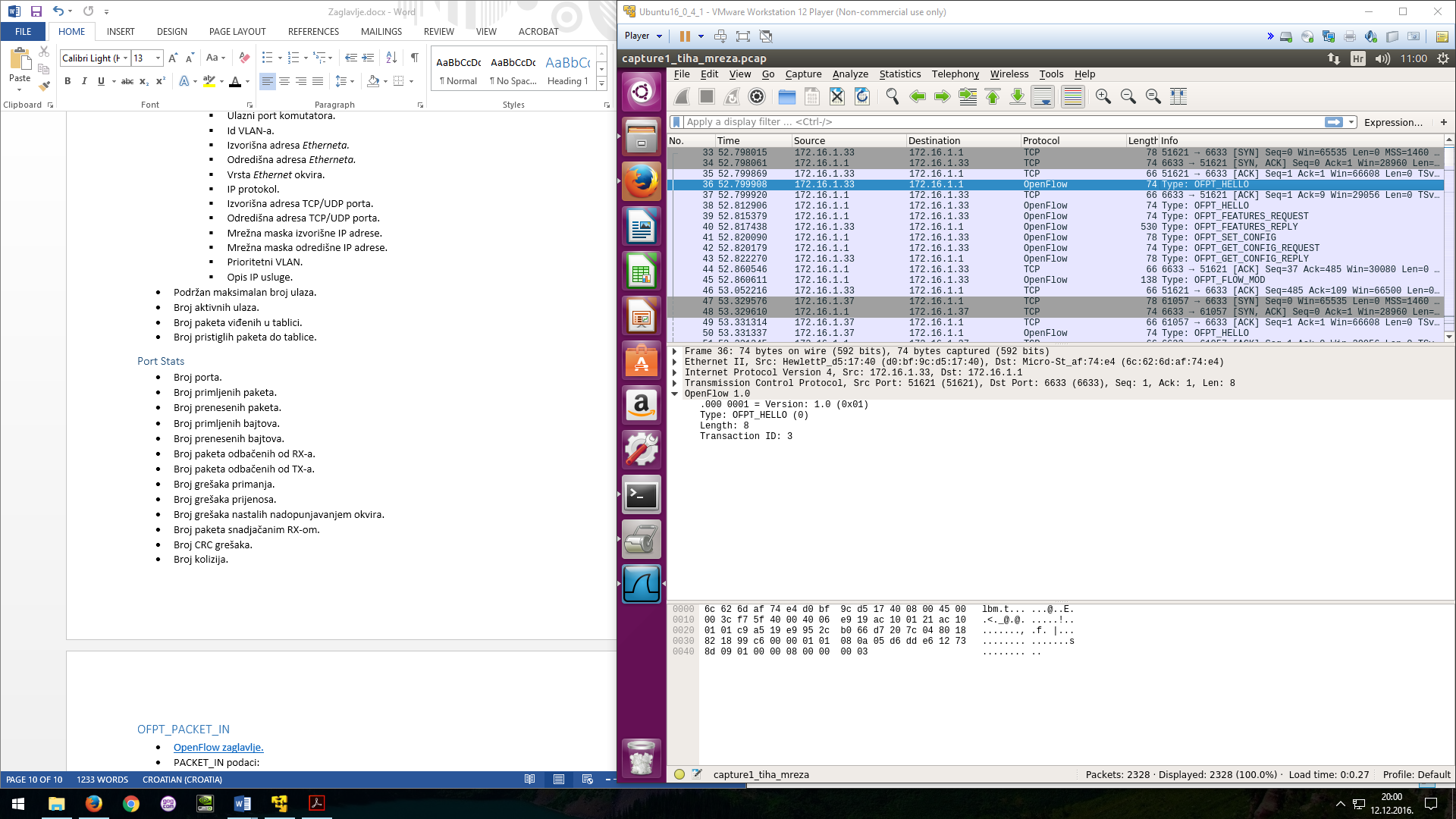
* Verzija OpenFlow protokola koja se koristi.
* Vrsta poruke.
* Veličina poruke.  
  – Zaglavlje i tijelo poruke čine cijelu poruku. Veličina poruke izražena je u oktetima.
* Transakcijski identifikacijski broj.  
  – Služi za uparivanje poruka. Tipičan par poruka sastoji se od poruke koja zahtijeva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

## Struktura fizičkog porta

Fizički port opisan je sljedećom strukturom:

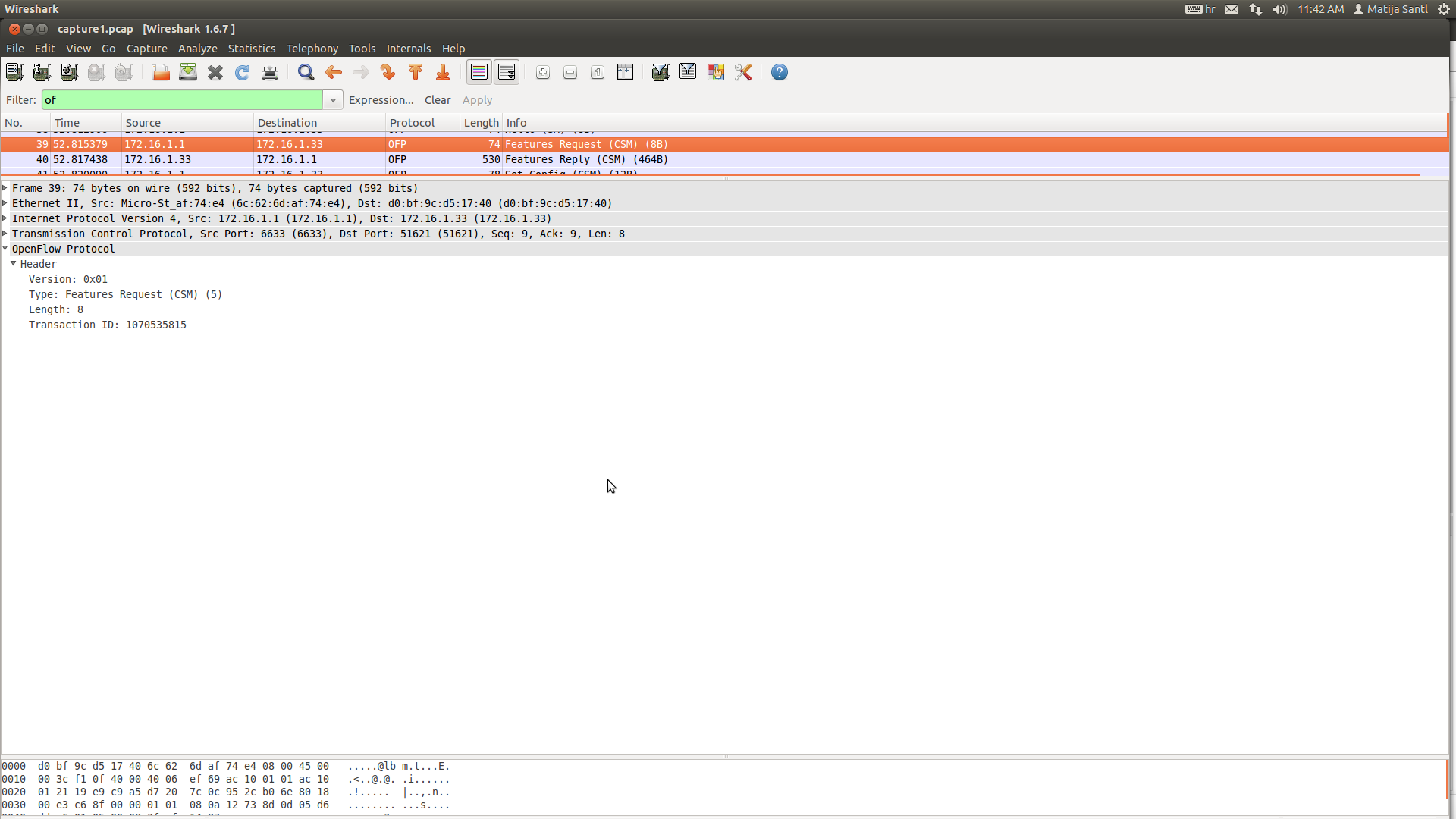
* Identifikacijski broj porta.  
  – Služi za povezivanje puta podataka s fizičkim portom.
* MAC adresa porta.
* Ime porta.  
  – U čitljivom formatu; namijenjeno korisnicima.
* Ponašajna bitmapa. Sljedeće karakteristike vrijede kada je odgovarajući bit postavljen u jedinicu:
  + Port je zatvoren.
  + 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 se uspostavljena veza.

## OFPT\_HELLO

OFTP\_HELLO poruke izmijenjuju se između kontrolera i komutatora pri uspostavi veze. Spada pod simetrične poruke.

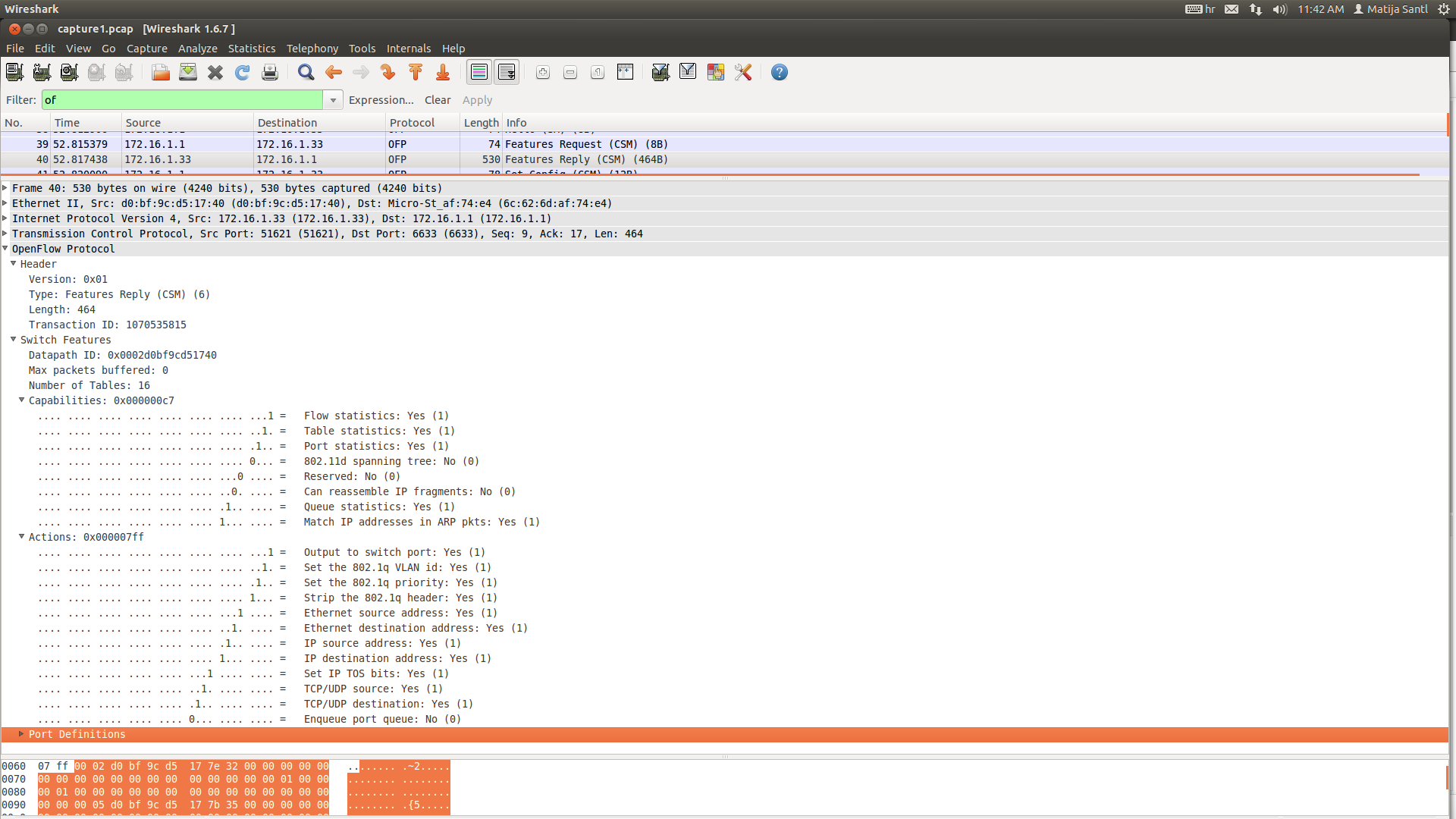
Poruka se sastoji samo od [OpenFlow zaglavlja](#_OFPT_HELLO).

## OFTP\_FEATURES\_REQUEST

Poruka od kontrolera komutatoru. Zahtijeva od komutatora da mu pošalje OFTP\_FEATURES\_REPLY.

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

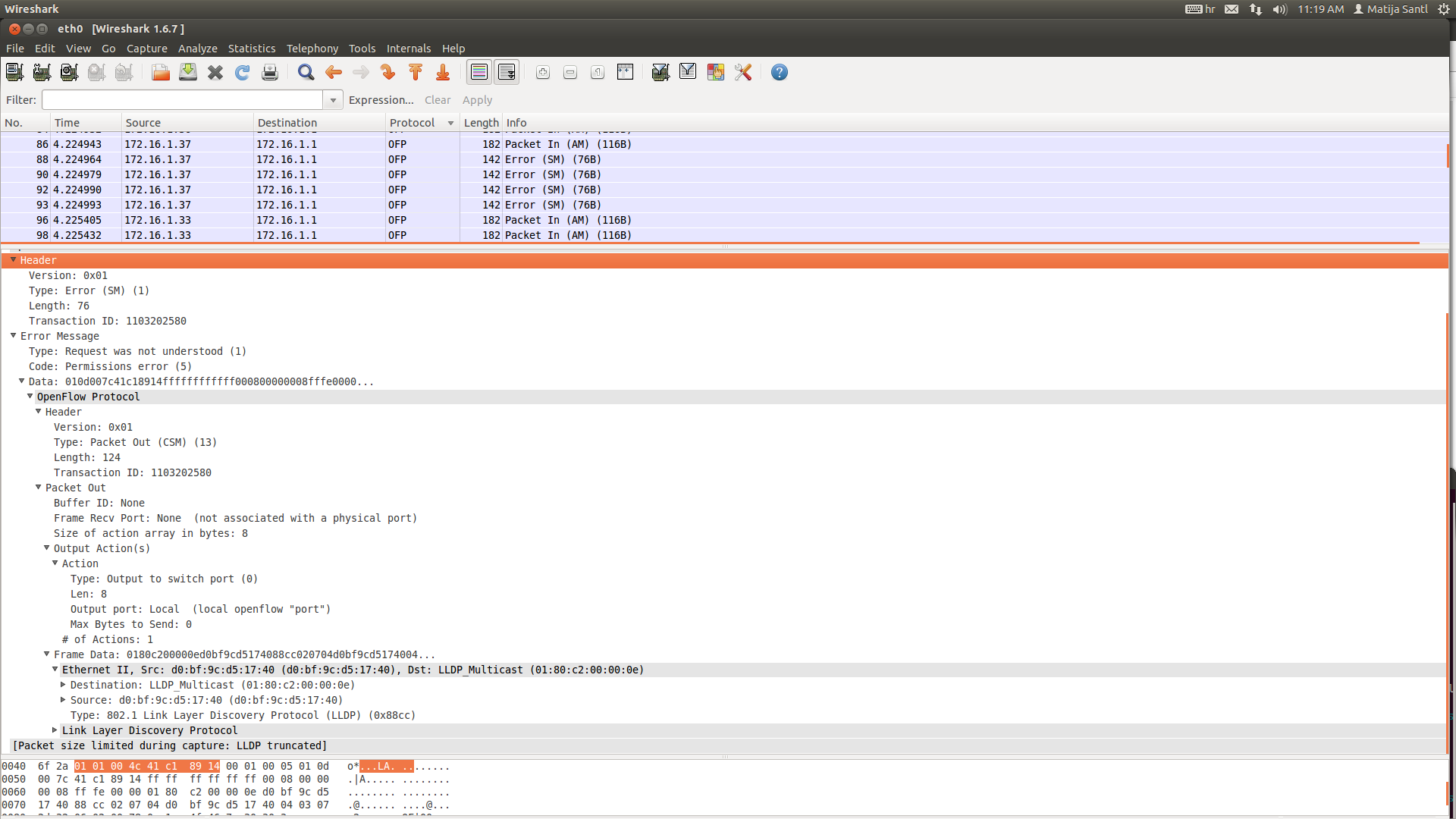
## OFTP\_FEATURES\_REPLY

Komutator na OFTP\_FEATURES\_REQUEST odgovara ovom porukom koja sadrži informacije o mogućnostima komutatora.

Struktura poruke:

* [OpenFlow zaglavlje.](#_Zaglavlje)
* Mogućnosti komutatora.
* 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

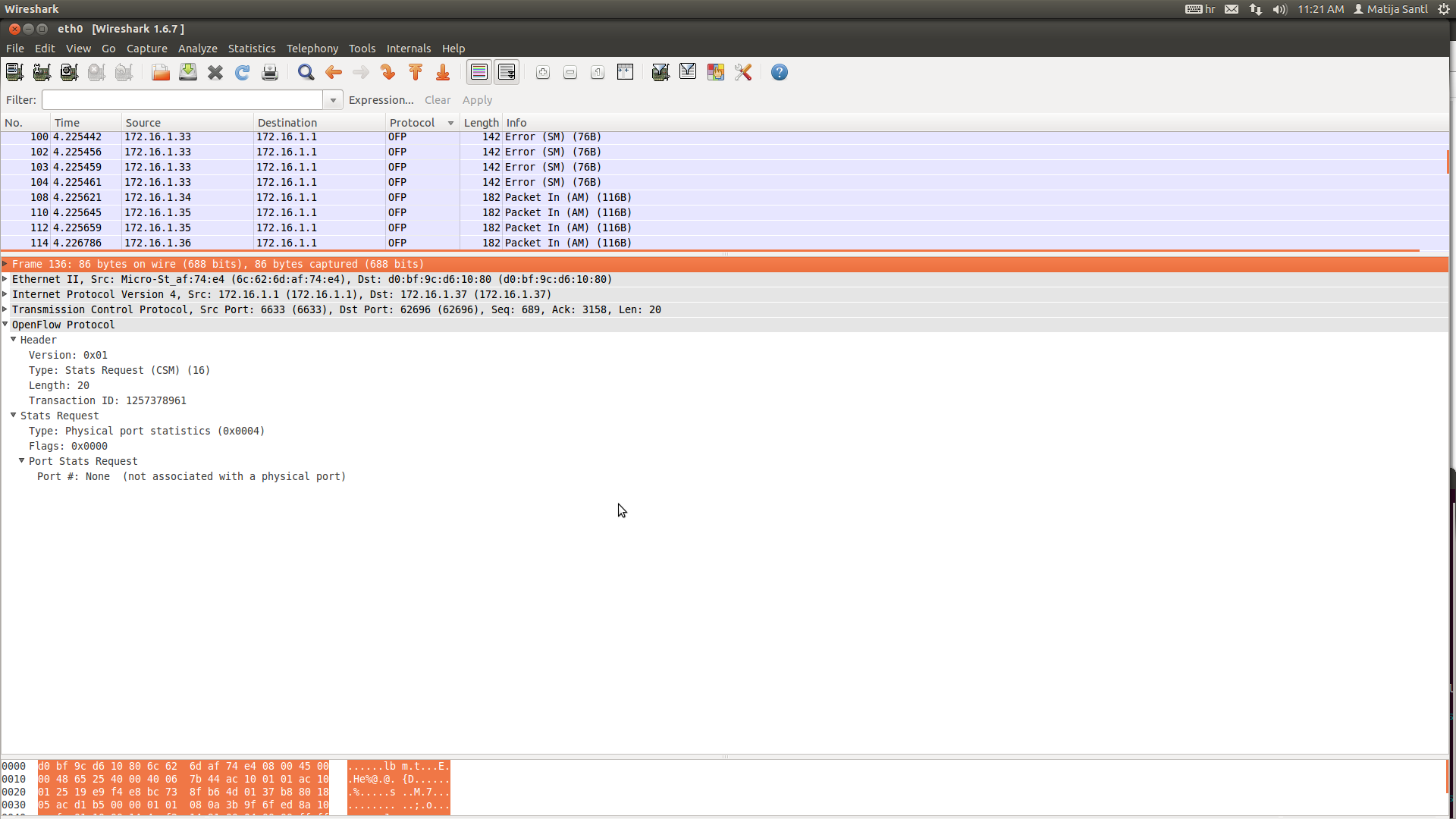
Poruka poslana od strane komutatora kontroleru koja obavještava o nastalom problemu. Spada pod nepromjenjive i simetrične poruke.

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 bajta poruke, kod koje je došlo do greške.

## OFPT\_STATS\_REQUEST

Poruke spadaju u skupinu statičnih poruka poslanih od strane kontrolera komutatoru. Njima se ispituje trenutno stanje toka podataka.



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 komutatora kontroleru 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.

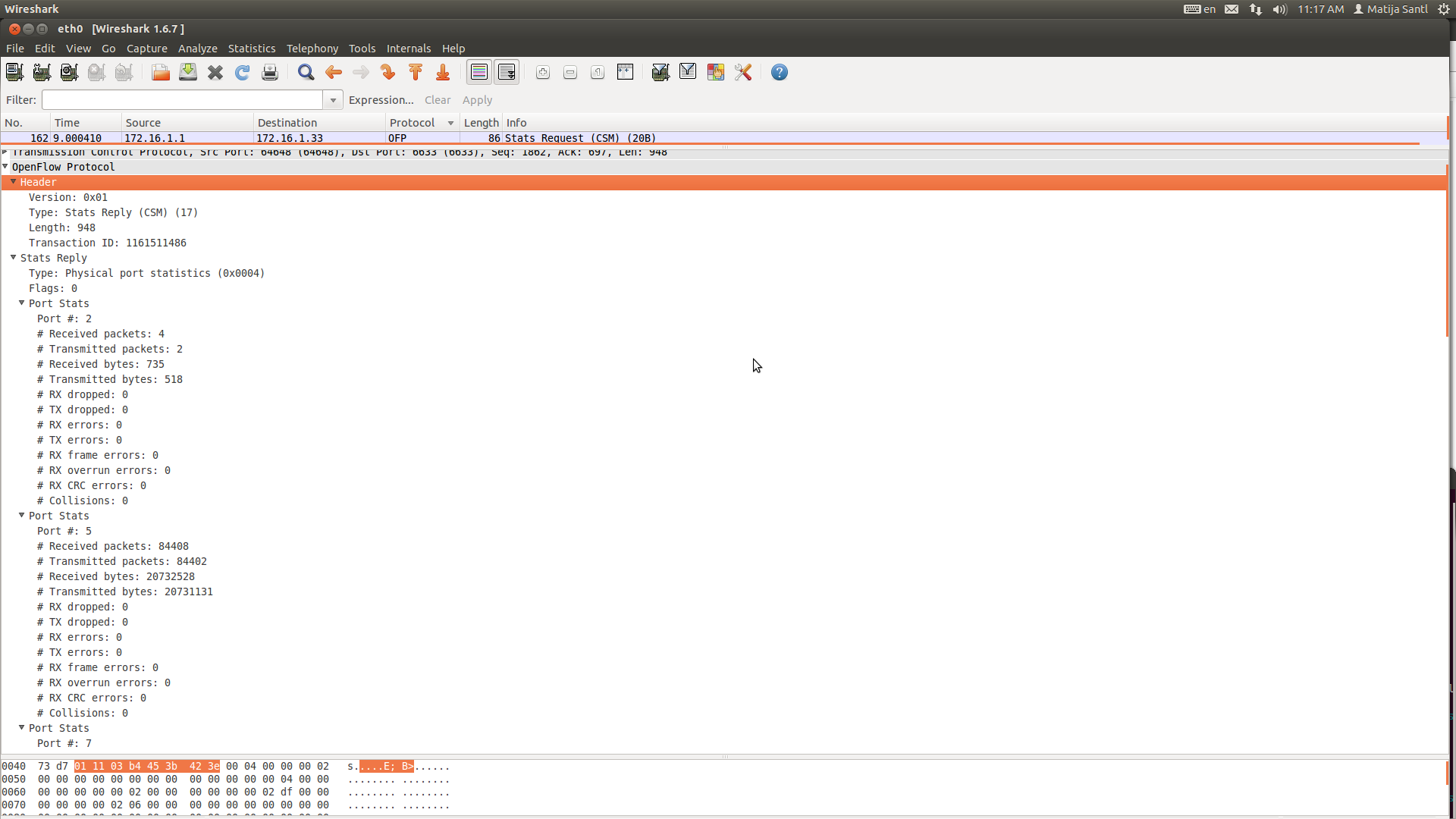
### Desc Stats Reply

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

### Table Stats

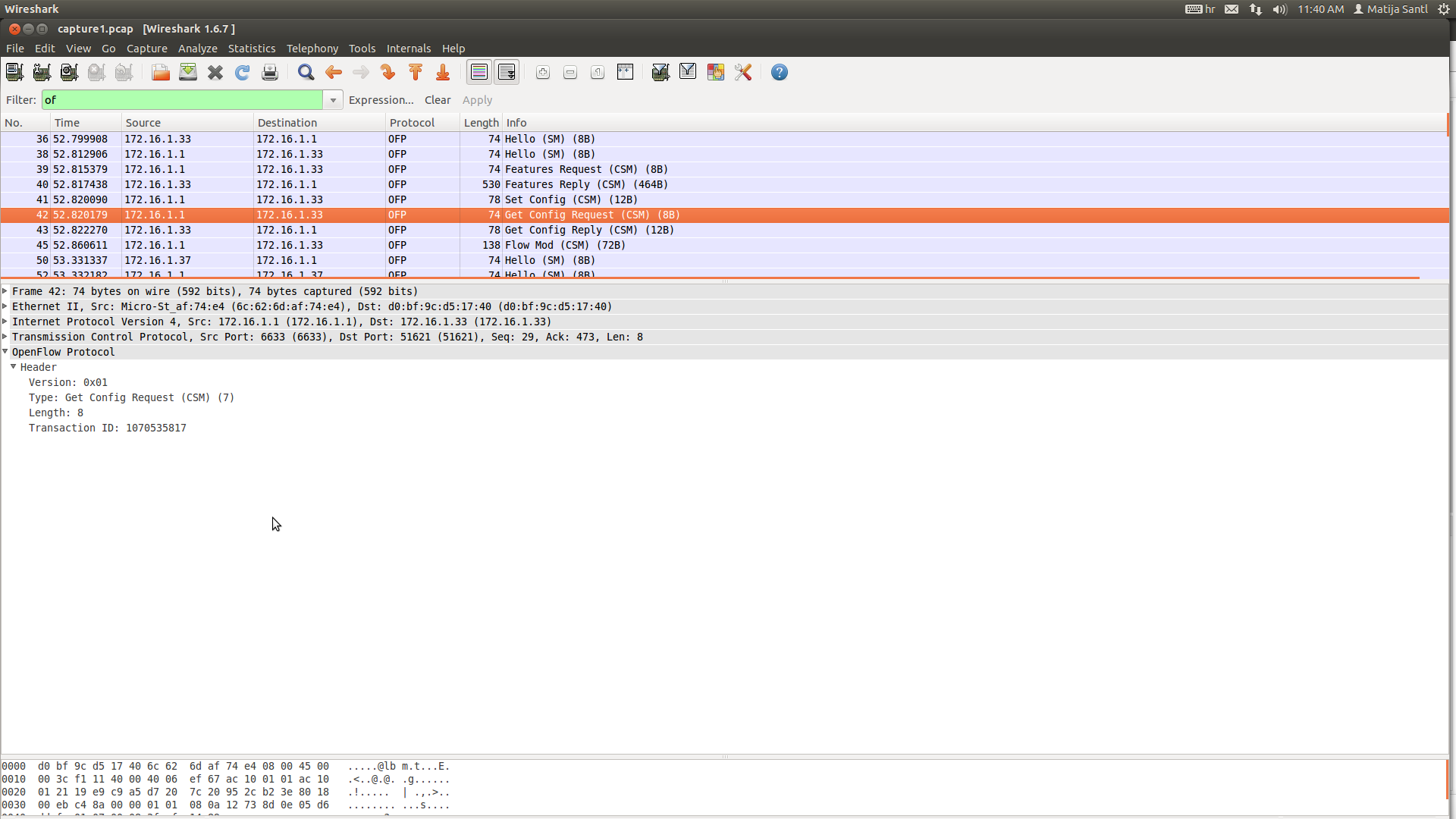
* Identifikator tablice.
* Poravnavanje do 32-bita naznačeno imenom tablice čija je to maksimalna veličina.
* Bitmapa polja za usporedbu koji su podržani u tablici:
  + Ulazni port komutatora.
  + Id VLAN-a.
  + Izvorišna adresa *Etherneta.*
  + Odredišna adresa *Etherneta.*
  + Vrsta *Ethernet* okvira.
  + 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



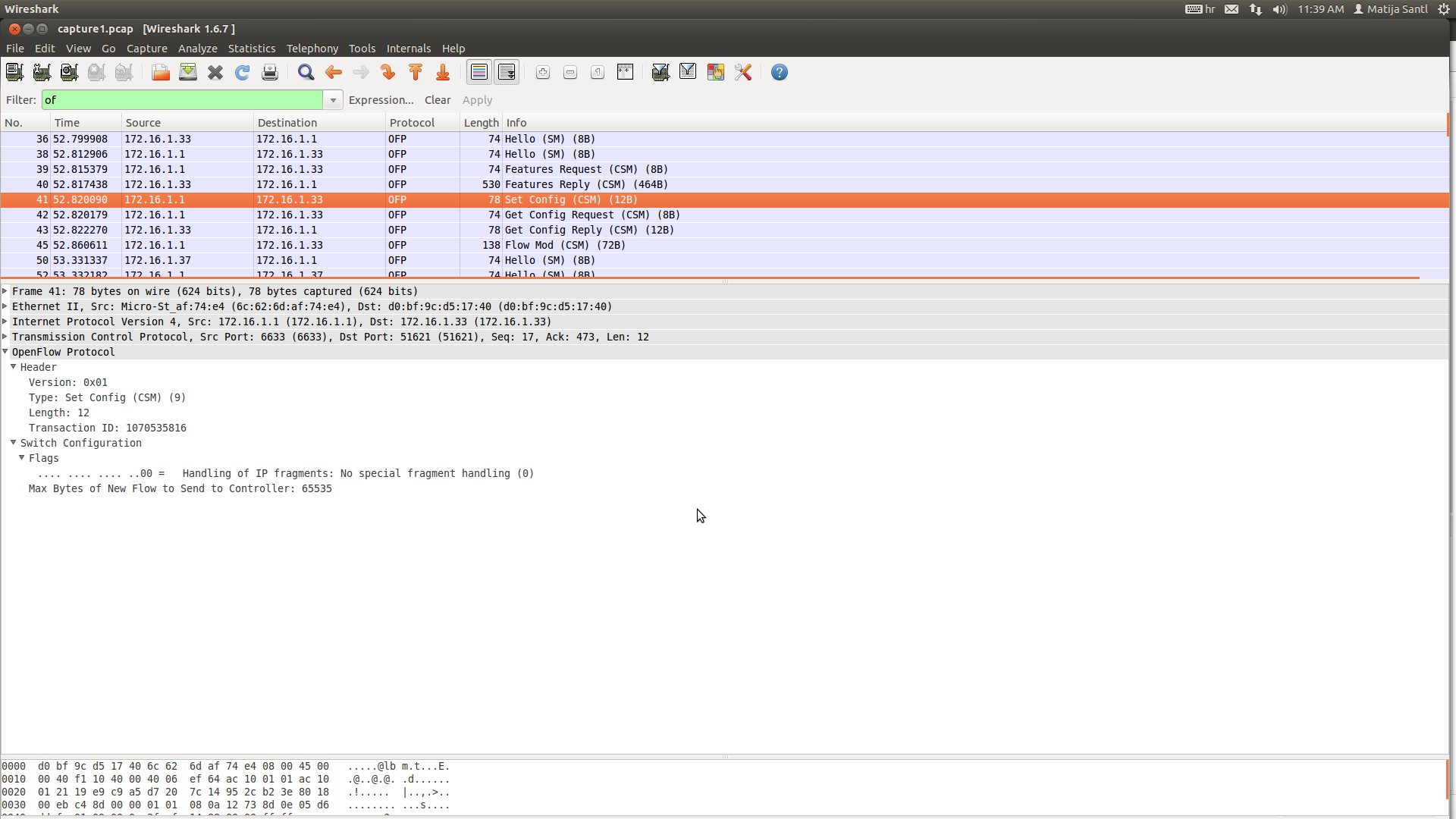
* Broj porta.
* Broj primljenih paketa.
* Broj prenesenih paketa.
* Broj primljenih bajtova.
* Broj prenesenih bajtova.
* Broj paketa odbačenih od RX-a.
* Broj paketa odbačenih od TX-a.
* Broj grešaka primanja.
* Broj grešaka prijenosa.
* Broj grešaka nastalih nadopunjavanjem okvira.
* Broj paketa snadjačanim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

## OFPT\_GET\_CONFIG\_REQUEST

Poruka se šalje od strane kontrolera komutatoru. Spada u skupinu poruka za konfiguraciju komutatora.

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.

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 kontrolera.

### OFPT\_FLOW\_MOD Wireshark 2.2.3

Struktura poruke:

* [OpenFlow zaglavlje.](#_Zaglavlje)
* Polja za usporedbu sadrže: (verzija koju domaočitam, u prilogu screenshoot)
  + 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.
* Id pohranjenog paketa koji treba primjeniti.
* Izlazni port (nije povezan sa fizički port).
* Zastavica nisu postavljene.

### OFPT\_FLOW\_MOD Wireshark 1.6.7 (laboratorijsko okruženje)

Struktura poruke:

* [OpenFlow zaglavlje.](#_Zaglavlje)
* Polja za usporedbu sadrže: (prema screenshootu iz labosa)
  + 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 primjeniti.
* 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

* [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 OFPR\_\*)
  + Polje za podatke koji se prenose

## OFPT\_PACKET\_OUT

* [OpenFlow zaglavlje.](#_Zaglavlje)
* PACKET\_OUT podaci:
* ID spremnika – ID broj za identifikaciju paketa u spremniku za jednu putanju (engl. *datapath*)
* Port sa 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

# Literatura